VŠB - Technická univerzita Ostrava, Fakulta bezpečnostního inženýrství

doc. Ing. Pavel Šenovský, Ph.D.

Modelování rozhodovacích procesů

skripta

4. vydání

Ostrava, 2015

Modelování rozhodovacích procesů
4. rozšířené vydání
tento text neprošel jazykovou úpravou

Vysoká škola báňská - Technická univerzita Ostrava, Fakulta bezpečnostního inženýrství

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Vážený studente, dostává se Vám do rukou učební text předmětu *Modelování rozhodovacích procesů*. Mým cílem při psaní tohoto textu bylo, aby student získal základní přehled v oblasti rozhodování a metod, které lze využít pro jeho podporu.

Teorie rozhodování ale není probírána pouze v tomto předmětu. Respektive existuje řada dalších předmětu, kde jste již mohli získat některé znalosti použitelné pro podporu rozhodován, možná aniž byste si to uvědomili, např. *statistické metody* jsou velmi dobře použitelné pro lepší pochopení rozhodovací situace.

V předmětu *Expertní systémy*, pokud si jej zvolíte budete moci získat znalosti o některých metodách inspirovaných studiem živých organizmů a způsobu jakým fungují - např. *neuronové sítě*. V navazujícím studiu Fakulta bezpečnostního inženýrství nabízí ke studiu předmět *Bezpečnostní informatika 3*, kde studenti mohou získat znalosti z oblasti dataminingu, což opět může být přínosné pro rozhodování.

Organizace textu

Pro zpříjemnění čtení jsem se také rozhodl zpracovat tento text formou vhodnou pro „distanční vzdělávání“, tak aby práce s ním byla co možná nejjednodušší. Z tohoto důvodu je text jednotlivých kapitol segmentován do bloků.

Každá kapitola začíná náhledem kapitoly, ve kterém se dozvíte, o čem budeme v kapitole mluvit a proč. V bodech se pokusím shrnout, co byste po prostudování kapitoly měli znát a kolik času by Vám studium mělo zabrat. Mějte Prosím na paměti, že tento časový údaj je pouze orientační, nebuďte proto prosím smutní nebo naštvaní, když ve skutečnosti budete kapitole věnovat o něco méně nebo více času.

Za kapitolou následuje shrnutí, ve kterém budou zdůrazněny informace, které byste si rozhodně měli zapamatovat (určitě Vám ale neuškodí, pokud si jich zapamatujete více).

To, že jste správně pochopili probíranou látku, si budete moci ověřit pomocí kontrolních otázek a testů, které by Vám měly poskytnout dostatečnou zpětnou vazbu k rozhodnutí, zdali jít dále nebo si vyhradit delší čas na opakování.

Pro zjednodušení orientace v textu jsem zavedl systém ikon:

Průvodce studiem
Slouží pro seznámení studentů s látkou, která bude v kapitole probírána.

Čas nutný ke studiu
Představuje odhad doby, který budete potřebovat k prostudování celé kapitoly. Jedná se pouze o orientační odhad, neznepokojujte se proto, pokud Vám studium bude trvat o něco déle nebo budete hotovi rychleji.

V novém vydání skript jsem se rozhodl pro trošičku jiný způsob přípravy skript a celá jsem je přepsal v desktop publishing (DTP) systému LaTeX. Důvodem jsou některé schopnosti, které je s

Vysvětlení, definice, poznámka
U této ikony najdete vysvětlující text, poznámku k probíranému tématu, která problém uvede do širších souvislostí, popřípadě důležitou definice.

Kontrolní otázky
Na závěr každé kapitoly je zařazeno několik otázek, které prověří, zda jste problematice kapitoly dostatečně porozuměli. Pokud nebudete vědět odpověď na některou otázku, je to signál pro Vás, abyste se ke kapitole vrátili.

Příklad
Příklady obsahují praktické demonstrace diskutovaného problému.

Přestávka
Po obtížné části textu, nebo prostě občas jenom tak je nutné si udělat krátkou přestávku, načerpat síly k novému studiu.

běžnými textovými procesory je možné dosáhnout pouze stěží a také to, že řada z vás bude studovat tento text přímo v počítači (tabletu, čtečce elektronických knih nebo mobilním telefonu) a v takovém případě budete chtít využít nejspíše všech schopností, která Vám tato zařízení poskytují.

Kolikrát jste si pomysleli - „jaké by to třeba bylo, kdybych mohl klepnout na jednu z těch divných zkratek (které informatici tak milují) a ona by mě přesměrovala automaticky na seznam zkratek“? Nebylo by lepší kdyby na daný literární pramen bylo možné se dostat přímo klepnutím na jeho číslo v textu, nebo aby jste nemuseli vybranou pasáž hledat přes čísla stránek, ale postačovalo by kliknout na jméno kapitoly v obsahu?

Mě jako studentovy by se to líbilo a proto doufám, že je oceníte i Vy, protože všechny výše uvedené možnosti skripta ve formátu Portable Document Format (PDF) obsahují. Aktivní odkazy jsou v textu zvýrazněny červenou (a v případě odkazů na literaturu zelenou) barvou.

Na konec skript byl přidán také rejstřík pojmů. Doporučuji, abyste jej v rámci přípravy na zkoušku prošli - zamyslete se nad tím, zda všechny pojmy, které jsem do něj zařadil, chápete a jste je schopni dát do souvislostí. Pokud ne, je vedle pojmu odkaz na číslo stránky, kde je pojem probrán a Vy můžete rychle zaplnit případné mezery ve svých znalostech předmětné problematiky.

Přeji Vám, aby čas, který strávíte s tímto textem, byl co možná nejpříjemnější a abyste jej nepovažovali za ztracený.

doc. Ing. Pavel Šenovský, Ph.D.

Poznámka autora:
Právě držíte v rukou třetí rozšířené vydání skript. Je možné, že právě studujete na zkoušku, nebo jste se ke skriptům dostali pouze náhodou po delší době. Z tohoto důvodu by se Vám mohlo hodit stručné shrnutí změn mezi jednotlivými vydáními těchto skript.

Novinky ve 4. vydání skript
1. opraveny drobné chybě v sazbě
2. doplněna kapitola věnovaná provádění dotazníkových šetření

Novinky ve 3. vydání skript
1. sazba v LaTeX
2. doplněna problematika diagramů vlivu k pojednání o rozhodování
3. doplněny některé informace o analýze citlivosti pro rozhodovací stromy
4. přepracován příklad použití multikriteriální analýzy
5. doplněny některé informace o práci s hierarchiemi kritérií v multikriteriálních analýzách
6. část věnována projektovému řízení a metodě kritické cesty (Critical Path Method (CPM)) byla doplněna o softwarovou podporu práce.
7. doplněny výpočty některých problémů v sítích pomocí software SciLab a R
8. problematika bilančních modelů doplněna o surovinový a kapacitní model

Kapitola 1

Úvod do rozhodování

Náhled kapitoly
V této kapitole se dozvíte něco o *rozhodování*. Protože rozhodování je součástí našeho každodenního života - dalo by se předpokládat, že lidé v něm budou opravdu dobří. Praxe ale ukazuje, že tomu tak není a při rozhodnutích proto často děláme chyby. Pokud uděláme špatné rozhodnutí v našem osobním životě, budeme to primárně my, kdo ponese následky. Špatné rozhodnutí firmy nebo úřadu ale může ovlivnit mnohem větší okruh lidí, nést s sebou podstatně větší náklady. Z tohoto důvodu pro složitá rozhodnutí je potřeba používat metody, které nám pomohou vybrat si správně.

Po přečtení této kapitoly budete
vědět

- co je to rozhodování

schopni

- rozlišit mezi různými rozhodovacími situacemi

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně hodinu.

1.1 Rozhodování

Každý den, každou hodinu, možná dokonce takřka každý moment doby, kdy jsme „vzhůru“ provádíme vědomě nebo nevědomě nespočet rozhodnutí. Většina z nich je natolik malých, nevýznamných, že je provádíme automaticky, aniž bychom o nich nějak uvažovali - např. se během chůze vyhneme překážce, kývneme na pozdrav na kolemjdoucího, kterého známe apod.

Některá rozhodnutí jsou náročnější - např. jsme hladoví a musíme se rozhodnout kdy a zejména co si dáme třeba na oběd. Ačkoliv taková rozhodnutí mohou být někdy značně obtížná, obvykle pro provedení rozhodnutí není vyžadováno nasazení sofistikovaných nástrojů pro podporu rozhodování. Použití takových nástrojů si vyhrazujeme pro rozhodovací situace, které jsou ještě výrazně složitější - např. rozhodování mezi různými výrobky s vysokou pořizovací cenou, různými technologiemi, různými ochrannými opatřeními apod.

Všechna rozhodnutí bez ohledu na to o čem rozhodujeme mají některé společné znaky. Prvním z nich je variantnost řešení. Rozhodujeme se tedy mezi různými *variantami řešení*, které jsou charak-

terizovány odlišnými vlastnostmi řešení. Naším cílem je přitom maximalizovat pozitivní (přínosné) charakteristiky řešení a zároveň minimalizovat charakteristiky negativní. Z tohoto pohledu je rozhodování tedy *optimalizační problém*.

Rozdílné výsledky rozhodování a jejich predikovatelnost (schopnost je předpovědět) je také první charakteristikou, kterou lze použít pro rozlišování mezi rozhodovacími situacemi. Podle predikovatelnosti výsledku rozhodnutí proto můžeme rozlišovat mezi rozhodováním za jistoty a za nejistoty. V ideálním světe, ačkoliv možná trochu nudném, bychom věděli přesně, jaké následky bude každé z našich rozhodnutí mít. V takovém případě hovoříme o *rozhodování za jistoty* a o tzv. *deterministické volbě*.

Svět kolem nás je ale plný nejistot, většinou proto přesně nejsme schopni predikovat, jaký výsledek bude naše rozhodnutí mít. To neznamená, že proces rozhodování ponecháváme čistě na náhodě, nebo že bychom byli úplně neschopni odhadnout výsledek. Odhadovaný výsledek je v takovém případě však zatížen nejistotou - existuje tedy pouze určitá pravděpodobnost toho, že rozhodnutí povede k určitému výsledku. V takovém případě hovoříme o *rozhodování za nejistoty* a používáme zejména *stochastické modely*, tedy modely pracující s pravděpodobností.

Otázkou zůstává - proč bychom měli vůbec nějaké formalizované metody pro podporu rozhodování používat, když očividně k rozhodnutí lze dospět i bez jejich použití. Odpověď je jednoduchá - přijetí rozhodnutí je snadné, přijetí správného (optimálního) rozhodnutí, ale snadné není. Metody používané pro podporu rozhodování nám umožňují přistoupit k rozhodnutí objektivně, bez předsudků (bias) a dospět k závěru, který je vnitřně konzistentní a zejména je odůvodnitelný. Analytik tedy nemusí spoléhat čistě na své zkušenosti a intuici, ačkoliv i tyto schopnosti jsou velmi důležité, ale může se opřít o dobře dokumentovanou metodu.

Při použití metod pro podporu rozhodování nemusí být řešený problém také řešen úplně od nuly - zvolená metoda poskytuje pracovní postup vedoucí k výsledku (postup tedy není nemusíme znova vyvíjet). Použití podpůrných metod proto umožňuje dosažení vyšší efektivity z hlediska vynaložených zdrojů a to jak časových nebo finančních, tak nároků na lidské zdroje.Výhodou použití metod je také to, že výsledek vyjde v očekávaném formátu a jsou pro něj známa případná omezení.

Vraťme se zpět k variantám rozhodnutí a rozdílům mezi nimi. Právě tyto rozdíly mohou posloužit jako další kritérium pro rozdělení rozhodovacích situacích. Takové rozdělení rozlišuje mezi volbami charakterizovanými jediným cílovým kritériem (*monoktriteriální*) a těmi, které jsou charakterizovány více cílovými kritérii (*multikriteriální*). Z tohoto pohledu tedy lze metody podpory rozhodování rozdělit na monokriteriální a multikriteriální.

Při použití jediného cílového kritéria často využíváme jeho finanční vyjádření - např. očekávaná finanční hodnota, zisk apod. Finanční vyjádření je z mnoha pohledů logické, protože finanční hodnotu lze určit pro celou řadu věcí: náklady, přínosy, cena majetku, budov, zdraví nebo dokonce života. Dobrým představitelem monokriteriální metody jsou *rozhodovací stromy*, které jsou podrobněji rozebrány v následující kapitole. *Multikriteriální analýza* (multicriterial analysis (MCA)) je pak metodou sama o sobě. I touto metodou se budeme v těchto výukových textech zabývat.

Samotný způsob, kterým jednotlivá kritéria vyjadřujeme (kvantifikujeme) může vést k použití odlišných metod pro podporu rozhodování. Kritéria lze v zásadě vyjádřit třemi způsoby:

1. *číselná hodnota* - ve smyslu spojité numerické veličiny např. věk, čas do úplného nabití, cena,
2. *ordinální hodnota* - tato hodnota může být taktéž číslem, avšak nemá charakter spojité veličiny, jedná se spíše o výběr z omezené množiny možností, např. 0 - ne, 1 - ano
3. *tvořená odpověď* - odpověď je tvořena volně zapsaným textem.

Odlišné typy kritérií nutně vedou k použití odlišných metod, které s nimi jsou schopny pracovat. Například číselné hodnoty jsou dobře zpracovatelné pomocí statistických metod jako je např. *regresní analýza*, zatímco ordinální hodnoty jsou zpracovatelné lépe pomocí metod jako jsou třeba *rozhodovací stromy* nebo *rozhodovací pravidla*.

V těchto skriptech je rozebrána řada metod, které se typologicky hodí pro řešení různých rozhodovacích situacích. Lze tedy také říci, že typologie problému samotného může být použita jako rozlišující kritérium metod podpory rozhodování.

Na začátku této kapitoly jsme rozebírali klasické rozhodnutí, tedy rozhodnutí mezi různými variantami jako např. výběr nějakého produktu nebo služby. Co ale kdyby podstata rozhodnutí byla odlišná. Rozhodnutí by např. nespočívalo ve výběru produktu, ale např. nastavení parametrů stroje. I v tomto případě se jedná o rozhodnutí. Z hlediska klasifikace problémů hovoříme v tomto případě často o tzv. *optimalizačních problémech*.

Pozor!
V předchozím textu byly zmíněny rozhodovací stromy. Všimněte si, že existují dvě různé metody, které se takto jmenují. V následující kapitole budou vysvětleny rozhodovací stromy jako monokriteriální metoda pro podporu rozhodování, zatímco „ty druhé“ rozhodovací stromy se používají v dataminingu a pracují nejlépe s ordinálními hodnotami. Jelikož tento text není zaměřen na datamining - nebudeme se tímto pojetím rozhodovacích stromů zabývat.

Také si povšimněte, že existuje celá řada dalších metod ať už založených na statistice nebo umělé inteligenci, které také v tomto textu nebudou podrobněji probírány, ačkoliv se jedná o metody, které jsou velmi užitečné pro podporu rozhodování. FBI nabízí celou řadu předmětů, které Vám umožní si znalosti v této oblasti doplnit, lze zmínit především předměty: Statistika, Expertní systémy, Bezpečnostní informatika 3.

Průvodce
Při výkonu své profese budete nuceni čelit mnoha různým rozhodovacím situacím, které budou vyžadovat použití odlišných přístupů. Z tohoto důvodu je výhodné budovat si portfolio metod, se kterými budete důvěrněji obeznámeni, a které budete schopni prakticky použít. Přitom platí, že čím více metod ovládáte, tím větší je šance, že se mezi nimi bude nacházet „ta pravá“, která vám umožní rychle vyřešit problém úplně a napoprvé.

Ve skutečnosti podstatou rozhodování jako takového je optimalizace cílového parametru nebo parametrů řešení, ale u problémů uvedených v předchozím odstavci je optimalizace viditelná na první pohled, a proto je označujeme speciálním názvem.

Např. pokud máme technologii a hledáme její optimální nastavení, můžeme použít jednu z metod *matematického programování* jako je třeba *lineární programování*, řešitelné např. pomocí *primárního algoritmu* známého též pod názvem *simplexova metoda*. Problematice lineárního programování bude věnována samostatná kapitola.

V případě, že budeme rozhodovat o nejlepším místě, kam je možno umístit zařízení vysílací infrastruktury operátora mobilní sítě, bude se jednat opět o optimalizační problém, ale řešení bude vyžadovat nasazení odlišné metody - tzv. *lokalizačních modelů*.

Odlišné problémy tedy vyžadují nasazení odlišných podpůrných metod.

1.2 Hierarchie a struktura rozhodování

Již víme, co je to rozhodování a že pro jeho efektivní provedení je nutné snažit se dosáhnout určitého *cíle nebo cílů rozhodování*. Na první pohled tvrzení předchozí věty vypadá jednoduše, bezproblémově, jenže pouze na první pohled. Cíle totiž mohou být dosahovány různými metodami, přičemž některé se budou vzájemně podporovat, zatímco jiné si mohou vzájemně odporovat.

Cíle se také liší podle toho v jakém rozhodovacím kontextu jsou vedeny. Poněkud odlišné zájmy a preference přitom budou mít např. různé úrovně řízení. Abychom lépe pochopili rozhodovací situaci můžeme provést její grafické vyjádření pomocí *hierarchie cílů rozhodnutí*.

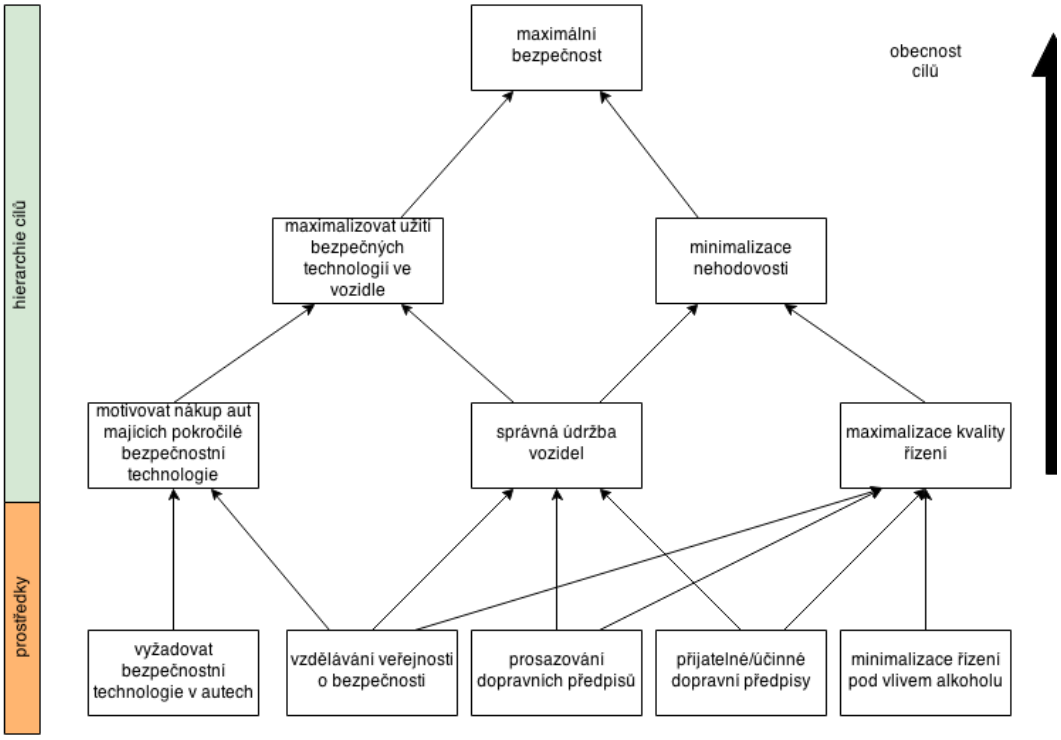
Definujme si problém, na kterém bychom mohli celou situaci jednoduše demonstrovat: *rozhodujeme se o opatřeních, který přinesou maximální bezpečnost v silničním provozu.*

Hierarchii cílů i prostředky pro jejich dosažení můžeme znázornit třeba jako na obr. 1.1.

Hierarchie cílů na obrázku naznačuje rozdílnost v pojetí cílů. Čím výše se v hierarchii cílů pohybujeme, tím jsou cíle obecnější. To odpovídá systému řízení obecně - čím výše ve vedení jdeme tím méně se řízení soustředí na detaily a také tím více se soustředí na celkový obrázek situace.

Rozdíly v pojetí cílů také lze vztáhnout k případným skupinám uživatelů - tedy tomu, kdo rozhodování provádí. V příkladu z obrázku může být tím, kdo rozhoduje:

1. běžný občan, který se snaží maximalizovat svou osobní bezpečnost,
2. výrobce automobilů, který se snaží zjistit jaké vlastnosti mají mít auta, která vyrábí,
3. ministerstvo dopravy, které má ve své gesci legislativu upravující pravidla silničního provozu,
4. apod.



Obrázek 1.1: Hierarchie cílů a prostředků pro jejich dosažení (adaptováno z [28])

Každá z výše uvedených skupin bude mít trochu jiné cíle, byť v obecné rovině může hierarchie cílů zůstat stejná. Tedy preference jednotlivých uživatelů se mohou ve stejné rozhodovací situaci, při stejně definovaných cílech výrazně lišit. Tento moment je nutné při rozhodování zohlednit a přímo souvisí s nutností poznat *kontext rozhodování*.

I pokud se zaměříme na „profesionální rozhodování“ např. ve firmách, je nutné se zamyslet nad tím, jaké je postavení jednotlivých skupin pracovníků na procesu rozhodování. Rozhodnutí samotné je vždy v něčí kompetenci. Tato kompetence vyplývá třeba z pracovní náplně nebo také může být stanovena nějakým předpisem a to ať už vnitřním (vnitropodnikový předpis) nebo v případě státní správy a samosprávy i předpisem legislativní povahy.

To, že rozhodnutí je v kompetenci určité osoby neznamená, že celý proces rozhodování je čistě na této osobě. Ve skutečnosti, pokud se daná osoba (manager) rozhoduje úplně sama, tedy profesionálně o složitých problémech, je to spíše výjimkou, která obvykle nekončí dobře. Manažer tedy má především manažerskou odpovědnost - provedení analýz, přípravu podkladů pro rozhodování a návrhy řešení situace může a měl by delegovat na své podřízené, kteří disponují patřičnou odborností.

Manažer pak kontroluje, zda navržená řešení odpovídají zadání, jsou zpracována v odpovídající kvalitě a doporučené řešení je prakticky realizovatelné. Pokud identifikuje jakékoliv problémy v analýzách nebo jejich závěrech dává podkladové materiály zpět k přepracování, v opačném případě přijme rozhodnutí a zahájí jeho realizaci.

K tomuto účelu ale manager musí disponovat odpovídajícími znalostmi v oblasti rozhodování a metod, které byly použity pro jeho podporu. V opačném případě bude manager pouze ve vleku svých podřízených, kteří však nenesou odpovědnost za rozhodnutí samotné.

Délka rozhodování a velikost podpůrného týmu je přímo úměrná složitosti problému. Jednoduché problémy tedy mohou zabrat řekněme třeba hodiny a vyžádat si použití běžných nástrojů, jako jsou např. tabulkové procesory, složité problémy mohou vyžadovat práci desítek lidí po řadu měsíců a využití specializovaných podpůrných nástrojů.

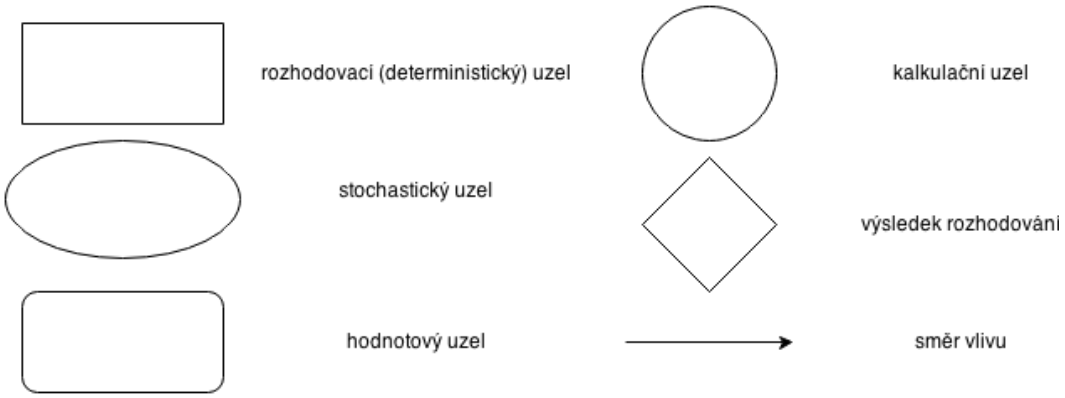
Zejména ve složitých rozhodovacích situacích je často před samotným řešením nutné lépe pochopit faktory, které do optimality výsledku rozhodování zasahují. K tomuto účelu lze použít řadu různých metod, z nichž s některými jste se možná již setkali, např. *SWOT analýza* nebo *Ishikawův diagram*, metodami *brainstormingu* a *myšlenkovými mapami* se budeme ještě v těchto skriptech zabývat. V tomto okamžiku se však zaměříme na jinou metodu, která se velmi dobře hodí pro rozebrání rozhodovací situace z hlediska různých faktorů, které v ní působí - zaměříme se na *diagramy vlivu* (influence

diagrams).

Jak název napovídá, jsou vlivové diagramy metodou grafickou, která se snaží zachytit a popsat charakter jednotlivých faktorů, u kterých se očekává, že jsou z hlediska rozhodování signifikantní. Toto očekávání může a nemusí být přitom prakticky naplněno, přičemž není možné předem rozhodnout, zda daný faktor má ve skutečnosti vliv nebo. Rozhodnutí o signifikantnosti faktoru je možné přijmout pouze na základě buďto statistických testů nebo analýzou citlivosti řešení na změny zkoumaného faktoru.

Zápis diagramu je po technické stránce snadný, protože diagram tvoří běžné tvary dostupné v řadě programů pro tvorbu schémat jako je Dia [1], Microsoft Visio [38], SmartDraw [22], Draw.io [4] a řada dalších.

Graficky lze jednotlivé prvky diagramu znázornit pomocí symbolů na obr. 1.2.

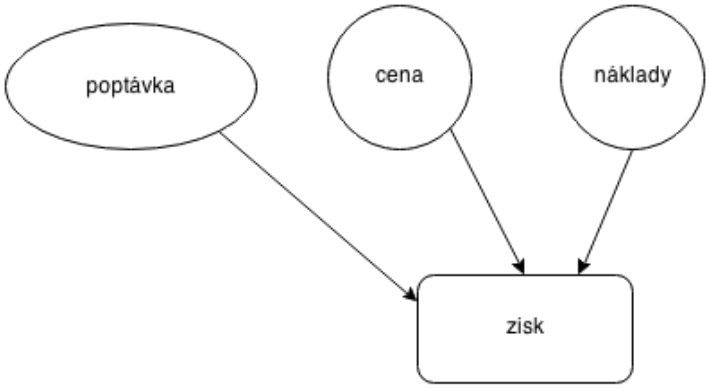


Obrázek 1.2: Základní konstruktory diagramů vlivu

Podívejme se na jednotlivé stavební prvky diagramů vlivu. První z nich je *rozhodovací uzel*. Tento uzel slouží k zachycení situace, kdy podstata (dílčího) rozhodnutí je čistě deterministická, tedy panuje jistota o následcích, plynoucích z rozhodnutí.

Stochastický uzel proti tomu je jiný - prvek jistoty zde neexistuje. Pomocí stochastických uzlů proto zachycujeme jevy, které nemáme pod kontrolou, které jsou náhodné apod.

Kalkulační uzel slouží pro pro zachycení veličin používaných ve výpočtech. Jako příklad bychom mohli uvést zisk, viz obr. 1.3.



Obrázek 1.3: Diagram vlivu pro stanovení zisku - zjednodušeně

Do tvorby zisku, tak jak je popsán na obrázku, vstupuje cena výrobku a náklady, které jsou technickými koeficienty a v znázorněné rozhodovací situaci se nemění. To je také důvod, proč v diagramu pro ně používáme kalkulační uzly. Poptávka oproti tomu je přímo závislá na řadě dalších faktorů, které nejsou pod naší kontrolou - proto použijeme stochastický uzel. Kdyby diagram tvorby zisku na obrázku byl diagramem úplným a nikoliv pouze fragmentem, musel by obsahovat minimálně jeden deterministický uzel a také jeden nebo více výsledků rozhodování.

Příklad
Zkuste experimentovat s diagramy vlivu. Vyberte si vhodný nástroj pro tvorbu grafiky a zkuste zpracovat jednoduchý rozbor nějakého rozhodnutí. Co má např. vliv na Vaše rozhodnutí při volbě volitelných předmětů pro příští semestr?

Shrnutí
Proces rozhodování předpokládá existenci různých variant řešení, které se liší svou užitečností. Rozhodování může být zatíženo nejistotou, tedy ne vždy lze předem přesně odhadnout, jaké budou následky rozhodnutí. Obvykle jsme však schopni do určité míry takovou nejistotu kvantifikovat.

V rámci rozhodování hraje obzvláště důležitou roli kontext rozhodování - ten totiž určuje systém preferencí rozhodovatele, které vedou k volbě odlišných cest vedoucích k řešení.

Strukturu rozhodovací situace a popis různých vlivů, které jsou ve hře, je možné graficky znázornit pomocí diagramů vlivu.

Otázky

1. Co charakterizuje rozhodovací situaci?
2. Jaký je rozdíl mezi stochastickým a deterministickým rozhodováním?
3. Jaké jsou typy kritérií z hlediska hodnot, kterých nabývají?
4. Může mít vliv podstata problému, o kterém rozhodujeme na metody vhodné pro jeho řešení - a proč? Můžete demonstrovat svou úvahu na příkladu?
5. Projděte v rychlosti předměty které jste dosud absolvovali a identifikujte metody, které lze použít pro podporu rozhodování (o čemkoliv, třeba i práci s riziky).
6. K čemu slouží diagram vlivu?
7. Pokuste se specifikovat důvod nutnosti zohlednění kontextu rozhodovací situace.

Kapitola 2

Rozhodovací stromy

Náhled kapitoly
V této kapitole se naučíme využívat jednoho z nejjednodušších nástrojů pro podporu rozhodování – *rozhodovací stromy.*

Po přečtení této kapitoly budete vědět

1. co jsou to rozhodovací stromy
2. jak a kdy je použít
3. co je to analýza citlivosti a jak se provádí

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně 2 hodiny, zejména pokud budete počítat přiložení příklady.

2.1 Monokriteriální pojetí rozhodovacích stromů

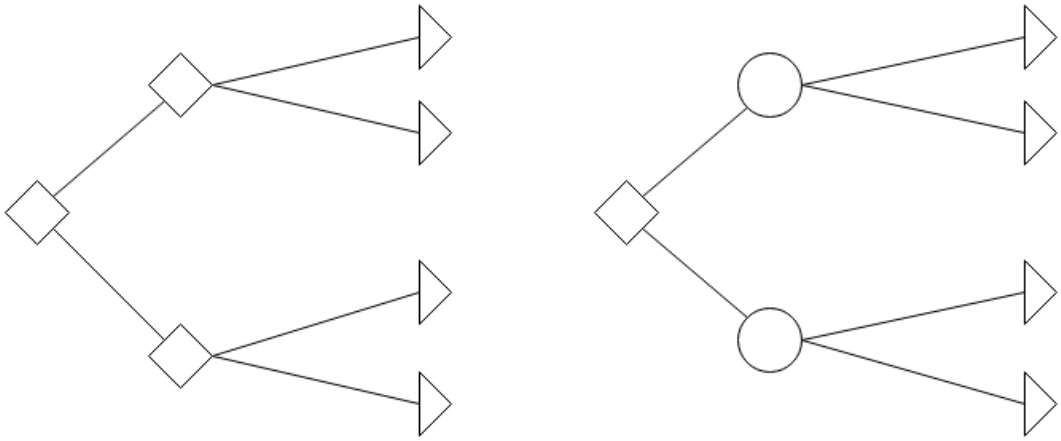
Jedním z nejjednodušších nástrojů, které lze použít pro podporu rozhodování jsou tzv. *rozhodovací stromy* (decision tree). Jedná se o orientované grafy, které svým vzhledem připomínají strom. V rozhodování nám pomáhají tak, že rozhodovací situaci a všechny varianty řešení vizualizují do formy větví, pro které vypočítáváme užitnost. Srovnáním užitností jednotlivých variant můžeme vybrat tu nejlepší pro řešení situace.

Jak již víme z předchozí kapitoly rozlišujeme mezi rozhodováním za jistoty a nejistoty. Rozhodovací stromy je možno použít pro oba případy, hovoříme pak o tzv. *deterministických a stochastických stromech.*

Deterministické stromy jsou tvořeny pouze deterministickými a listovými uzly. Stochastické stromy pak obsahují jeden nebo více stochastických uzlů. Platí přitom, že stochastický strom obvykle obsahuje také deterministické uzly. V případě, že by strom by tvořen pouze stochastickými uzly nejednalo by se ve skutečnosti o rozhodovací situaci, protože výsledek by byl čistě v rukou vrtkavé náhody.

Jednoduché schéma deterministického a stochastického stromu je dostupné na obr. 2.1.

Rozhodovací stromy jsou tvořeny uzly a hranami. Deterministické uzly značíme kosočtverci, v některé literatuře (např. [35]) se pro značení uzlů používají čtverce. Stochastické uzly v rozhodovacích stromech označujeme kolečkem. Ze stochastického uzlu vycházejí obvykle dvě nebo více hran reprezentujících možný následek volby. Každá z těchto hran musí být ohodnocena pravděpodobností to, že daný následek nastane. Součet pravděpodobností hran vycházejících ze stochastického uzlu musí být roven 1 (musí tedy pokrýt celý vějíř statisticky signifikantních následků).



Obrázek 2.1: Deterministické a stochastické stromy

Uzly představují rozhodnutí nebo události (u stochastických uzlů), které mají vliv na předmět rozhodnutí. K rozhodnutí přitom budeme potřebovat kritérium, které budeme optimalizovat. Tímto kritériem může to být zisk, náklady nebo jakákoliv jiná veličina.

Rozhodovací stromy jsou ve většině případů využívány pro monokriteriální rozhodování. Není to tím, že strom jako takový by nebylo možné použít v případě existence více cílových kritérií, ale spíše tím, že dodatečný aparát pro uvažování vícekriteriálnosti efektivně eliminuje výhody proti metodám, které pracují „nativně“ s více cílovými kritérii, jako je třeba metoda MCA. Použitím rozhodovacích stromů pro multiktriteriální rozhodování je stručně popsáno na konci této kapitoly.

Jednotlivé hrany grafu představují variantu, následek rozhodnutí nebo události. Délka hrany by měla do určité míry korespondovat s obdobím, pro které je konstruován tak, aby uzly pod sebou se z hlediska modelování udály v přibližně stejnou dobu. Správné vykreslování umožňuje rozhodovací kritérium optimalizovat i časově.

Aby byl strom lépe interpretovatelný, velmi často se uzly opatřují čísly, která můžeme usnadni orientaci v poznámkovém aparátu analýzy. Odkazem k číslu uzlu lze vyřešit slovní komentáře, složitější výpočty apod. Alternativně lze krátký popis varianty, popřípadě výpočet připojit přímo do stromu k jednotlivým hranám. Je ovšem potřeba zdůraznit, že čím větší množství informací je v rámci stromu zpracováváno, tím je následně horší orientace ve stromu a tím pádem je také mnohem snadnější v něm udělat chybu.

Demonstrujme postup tvorby rozhodovacího stromu na jednoduchých příkladech. Příklady jsou úmyslně zvoleny tak, aby byly ekonomického charakteru (pro snadné pochopení), nicméně rád bych zdůraznil, že rozhodovací stromy jsou obecné a lze je tedy využít prakticky pro jakoukoliv oblast rozhodování. Zároveň je potřeba říci, že uvedené příklady jsou „školní“, tedy mající za účel demonstrovat metodu, nikoliv reálně řešit nějaký problém.

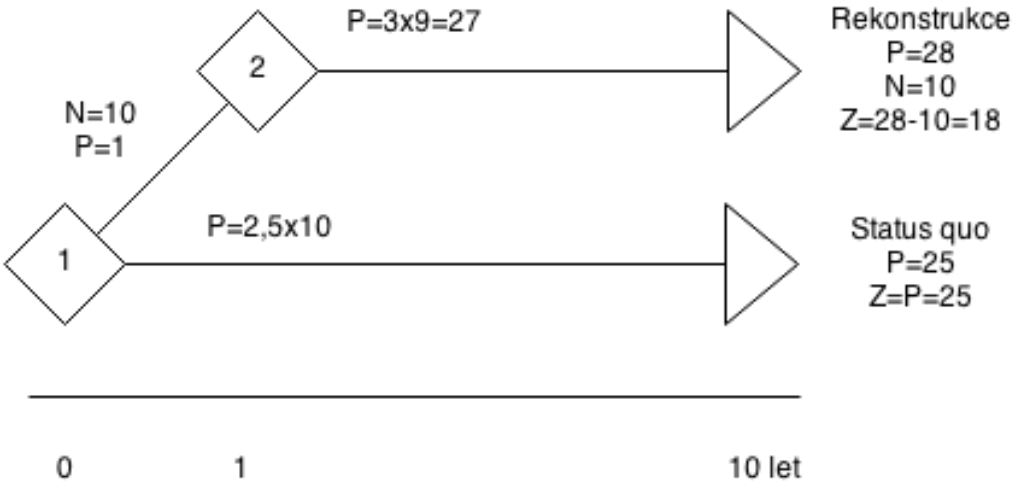
Strom 1: Deterministický strom
Firma XXX stojí před rozhodnutím, zda investovat do rekonstrukce své provozovny. V současné době je na provozovně schopna prodat výrobků ročně za 2,5 mil. Kč, rekonstrukcí by došlo k dočasnému přestěhování po dobu rekonstrukce do náhradních prostor s omezenou kapacitou, kde by bylo možno prodat zboží pouze za 1 mil. Kč. Doba rekonstrukce bude 1 rok a bude stát 10 mil. Kč. V nových prostorách bude firma XXX schopna prodat výrobků za 3 mil. Kč.

Doporučte, zda rekonstruovat nebo ne v horizontu 10-ti let.

Řešení
Zadání vypadá relativně složitě, ale ve skutečnosti tomu tak není. Máme zde pouze dvě varianty:

1. rekonstruovat,
2. nebo nerekonstruovat.

Řešení je možno konstruovat se třemi nebo čtyřmi uzly v závislosti na tom, jakým způsobem budeme problém chápat (viz obr. 2.2).



Obrázek 2.2: Grafické řešení příkladu deterministického stromu (příklad Strom 1)

Uzel
1 rozhodnutí zda rekonstruovat nebo ne
2 ukončena rekonstrukce

Počítáme přínosy a náklady spojené s každou z variant. Varianta 1 - rekonstrukce má přínosy (P) 1 mil. Kč za prodej v náhradní provozovně, a devět let prodeje v nové provozovně po 3 mil. Kč. Přínosy celkem jsou tedy 28 mil. Náklady na realizaci varianty jsou 10 mil. Kč a zisk celkem po realizaci této varianty tedy činí 18 mil. Kč.

Alternativní variantou je nerekonstruovat (tedy nulová varianta), která s sebou nese stálý příjem 2,5 mil. (po dobu 10-ti let) bez dodatečných nákladů. Zisk plynoucí s realizace této varianty tedy činí 25 mil. Kč.

Protože se snažíme **maximalizovat zisk** musíme **preferovat nulovou variantu**, protože zisk je v ní větší.

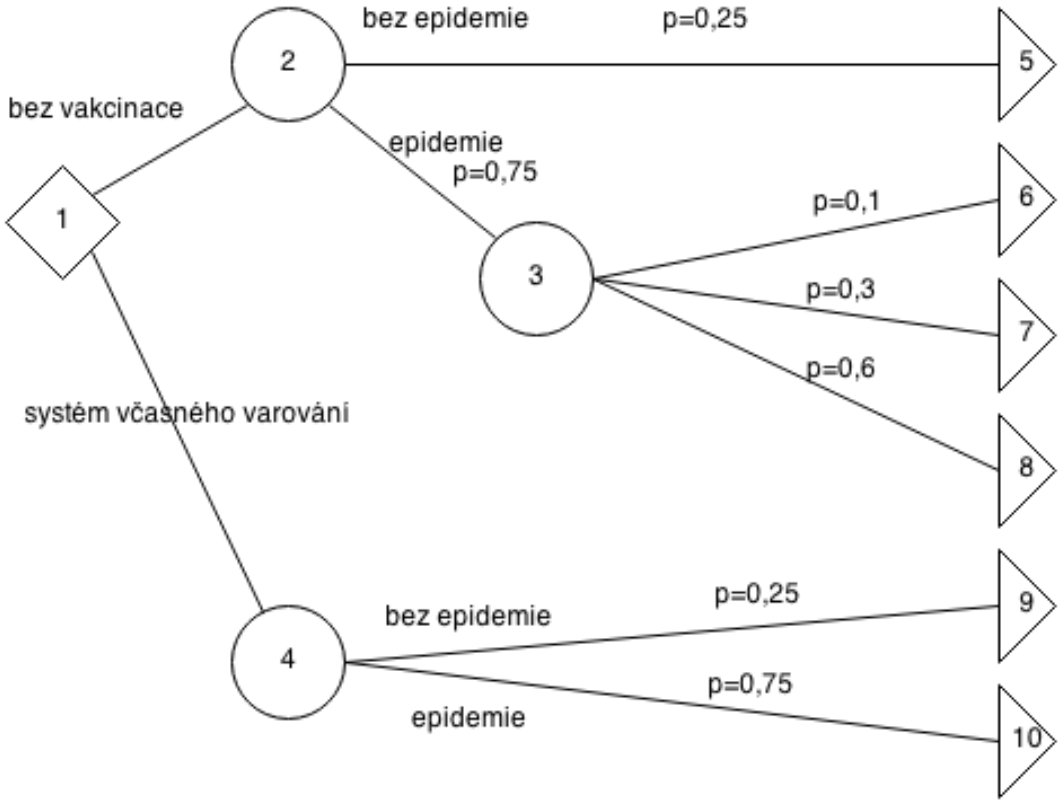
Strom 2: Stochastický strom
Poradní výbor zvažuje ekonomický přínos preventivního programu očkování proti chřipce. Pro takový program výbor zvažuje systém „včasného varování“, který by stál 120 mil. EUR a umožnil by detekovat včas nástup chřipkové epidemie. Po zjištění počátku epidemie by se začalo s očkováním.

Pokud by program očkování nebyl realizován, odhaduje se, že náklady na léčbu by dosáhly 280 mil. EUR s pravděpodobností 10%, 400 mil. EUR s pravděpodobností 30% a 600 mil. EUR s pravděpodobností 60%. Samotný program očkování by stál 400 mil. EUR a pravděpodobnost toho, že chřipková epidemie přijde je odhadována na 75%.

Předpokládáme, že očkování je okamžitě účinné a bude 100% efektivní.

Poslední věta příkladu Stromu 2 o okamžité účinnosti a 100 procentní efektivitě je určena ke zjednodušení příkladu tak, ať se nám dobře počítá. Pokusme se sestavit stochastický strom.

Proces konstrukce stromu by měl být jasně patrný z obr. 2.3, takže prostě zkusme vypočítat užitek jednotlivých variant řešení. Proces výpočtu, alespoň v první fázi bude totožný s výpočtem deterministického stromu – v první fázi výpočtu tedy ignorujeme pravděpodobnosti. Všimněte si ale, jakým způsobem jsou pravděpodobnosti pro varianty stanovovány. **Součet pravděpodobností uzlů** vycházejících z jakéhokoliv stochastického uzlu **musí být roven 1**! Samotné grafické řešení stromu tohoto příkladu naleznete na obr. 2.3.



Obrázek 2.3: Grafické řešení příkladu stochastického stromu (příklad Strom 2)

Toto omezení pravděpodobností nám zaručí, že z hlediska rozhodování budeme mít pokryt celý vějíř možných událostí, které mohou vyplynout z našeho rozhodnutí. Ale teď se už vraťme k samotnému výpočtu.

Náklady:
Bez vakcinace, epidemie nenastala: 0 EUR (1-2-5)
Bez vakcinace a epidemie s malými náklady: -280 EUR (1-2-3-6)
Bez vakcinace a epidemie se středními náklady: -400 EUR (1-2-3-7)
Bez vakcinace a epidemie s vysokými náklady: -600 EUR (1-2-3-8)
Systém včasného varování a bez epidemie: -120 EUR (1-4-9)
Systém včasného varování a epidemie: -520 EUR (1-4-9)

Pokud by rozhodovací strom byl deterministický, stačilo by nám nyní vzít z našeho hlediska nejvhodnější variantu - tedy tu s nejmenšími náklady (což by byla varianta bez systému včasného varování a vakcinace) a prohlásit ji za výsledek. Bohužel, tento příklad pracuje s pravděpodobnostmi, které jsme dosud ve výpočtu nezohlednili.

Pro zohlednění pravděpodobností budeme muset zpětně přepočítat užitnost od jednotlivých „listových uzlů“ až ke kořenu stromu. Užitnost stochastických uzlů vypočítáme tak, že užitnosti hran z něho vycházejících vynásobíme pravděpodobnostmi, které se k nim vztahují a sečteme je.
Náklady uzel 3: (-280 * 0,1) + (-400 * 0,3) + (-600 * 0,6) = -508 mil. EUR
Náklady uzel 2: 0 * 0,25 + (-508 * 0,75) = -381 mil. EUR
Náklady uzel 4: (-120 * 0,25) + (-520 * 0,75) = -420 mil. EUR

Uzel 1 je deterministický, vybíráme tedy variantu, která je pro nás výhodnější, srovnáváme přitom užitnosti uzlů 2 (-381 mil. EUR) a 4 (-420 mil. EUR). Logicky volíme v tomto případě nižší náklady - tedy **doporučujeme zamítnutí realizace systému včasného varování**.

Další příklady můžete zkusit vypočítat sami (příklady byly převzaty z [24]):

Úkol 1
MDG je společnost zaměřená na provádění geologických průzkumů sloužících k ověření toho, zda se na zkoumaném pozemku nacházejí ložiska kovů, které by bylo možné dále komerčně zužitkovat. MDG má možnost koupit pozemek za 3 mil. $.

Pokud MDG zakoupí pozemek, provede geologický průzkum. Předchozí zkušenosti ukazují, že pro tento typ pozemku geologický průzkum stojí přibližně 1 mil. $ a přináší následující výsledky:

1. mangan 1% šance
2. zlato 0.05% šance
3. stříbro 0.2% šance

Pokud je nalezen jeden kov, není šance nalézt žádný další kov.

Pokud je nalezen mangan, pak může být pozemek prodán za 30 mil. $, v případě zlata za 250 mil. $ a stříbra 150 mil. $.

MDG také může zaplatit 750 000 $ za práva provést třídenní předběžný průzkum, předtím než se rozhodne. Předchozí zkušenosti ukazují, že třídenní průzkum stojí 250 000 $ a zvyšuje jistotu, že nějaké ložisku kovu na pozemku je na 50% (toho že se průzkum povede realizovat). Šance na nalezení jednotlivých kovů je následující:

1. mangan - 3%
2. zlato - 2%
3. stříbro - 1%

Pokud třídenní test neprokáže přítomnost ložiska, šance, že ložiska zájmových kovů jsou přítomna, je velmi malá:

1. mangan - 0.75%
2. zlato - 0.04%
3. stříbro - 0.175%

Co doporučíte firmě MDG?

Řešení úkol 1
3-denní test a pokud dopadl úspěšně koupit pozemek

Úkol 2
Firma se rozhoduje, zda se zúčastní určitého výběrového řízení nebo ne. Odhadují, že pouhopouhá příprava na výběrové řízení bude stát 10 000 $. Pokud se řízení zúčastní odhadují, že je 50%-ní šance, že se dostanou do užšího výběru. V užším výběru bude potřeba dodat podrobnější informace (odhadovaná cena 5 000 $).

Společnost odhaduje, že náklady na práci a materiál spojené s realizací případného kontraktu bude činit 127 000 $. Firma uvažuje o třech možných cenách pro nabídku - 155 000 $, 170 000 $ a 190 000 $. Pravděpodobnost úspěchu pře těchto cenách je následující: 0.9, 0.75 a 0.35.

Má se firma zúčastnit výběrového řízení a pokud ano s jakou cenou?

Řešení úkol 2
zúčastnit se výběrového řízení, pokud projde do dalšího kola vybrat střední cenovou hladinu nabídky (170 000 $)

Další příklady
Nezapomeňte, že další příklady máte k dispozici v systému Moodle: `http://lms.vsb.cz` [5].

2.2 Analýza citlivosti

Jednou z metod, jak určit, zda zkoumaná proměnná má na rozhodování vliv nebo ne je provedení tzv. *analýzy citlivosti*. Tento typ analýz je obecný, není tedy vyloženě spojen s metodou rozhodovacích stromů. Základní princip metody je v zásadě jednoduchý - měníme zkoumanou proměnnou a sledujeme jaký vliv má na výsledek rozhodování - ostatní veličiny zůstávají statické (neměnné).

Postup analýzy je následující:

1. určení možného rozsahu proměnných (určení horní a dolní meze),
2. kvantifikace následků změny hodnoty parametru,
3. (nepovinně) vizualizace citlivosti pomocí grafů,
4. interpretace výsledků.

Jak tedy probíhá analýza citlivosti v praxi. Vraťme se k jednoduchému příkladu stochastického stromu z předchozí podkapitoly (příklad Strom 2). V rámci tohoto příkladu je provedena kvantifikace jednotlivých hran a to jak z hlediska pravděpodobnosti, tak z hlediska hodnot sledovaného parametru - v našem případě nákladů.

Tyto náklady jsou specifikovány pomocí jediného čísla - to je nutné, aby bylo možné vypočítat strom. Z praktického hlediska se ale jedná spíše o odhad, resp. střední hodnotu očekávaných nákladů. V praxi bude tedy tato hodnota o něco větší nebo menší. Pokud jsou k dispozici dostatečné údaje, lze použít statistických metod k výpočtu jednotlivých mezí při určité požadované míře pravděpodobnosti. Ve statistice se často jako mez používá hodnota 95% - tedy meze se volí tak, aby ohraničovaly oblast proměnné pokrývající 95% výskytů této proměnné.

Plnohodnotné statistické vyhodnocení však vyžaduje identifikaci rozdělení pravděpodobnosti proměnné a to zase vyžaduje poměrně kvalitní data, na základě kterých bude statistická analýza provedena. Taková data často v praxi nebývají dostupná. Stejně je tomu i v našem případě. Jelikož nemáme dostupná data - určíme meze expertním odhadem (viz tab. 2.1).

Tabulka 2.1: Odhad horní a dolní meze proměnných příkladu Strom 2 pro účely provedení analýzy citlivosti

Označení	**Proměnná**	**Střední hodnota** [mil. EUR]	**Dolní mez** [mil. EUR]	**Horní mez** [mil. EUR]
A	Epidemie - malé náklady	280	250	310
B	Epidemie - střední náklady	400	360	440
C	Epidemie - velké náklady	600	520	680
D	systém včasného varování	120	100	140
E	vakcinace (po varování)	400	360	440

Podle údajů v tab. 2.1 provedeme výpočet. Nejprve pro porovnání spočteme s použitím středních hodnot uzly 2 a 4 (jejich užitná hodnota rozhodne o optimalitě řešení, viz obr. 2.3). Tento výpočet jsme provedli již v předchozí podkapitole, můžeme proto pouze přepsat výsledky - uzel 2: **381** mil. EUR a uzel 4: **420** mil. EUR.

Tyto hodnoty vedou k rozhodnutí, že výhodnější nenasadit systém včasného varování před chřipkou (minimalizujeme náklady). Výše uvedené hodnoty pro nás mají také význam v tom, že tyto údaje budou sloužit jako test citlivosti řešení. Pro proměnné A - C tak budeme sledovat, zda výsledné náklady nepřesáhnou hodnotu 420 mil. EUR, zatímco pro proměnné D a E budeme zjišťovat zda náklady neklesnou pod hodnotu 381 mil. EUR.

Zkusme provést výpočet pro horní a dolní mez stanovenou pro variantu A.

$$naklady = 0,75 \cdot (0,1x + 0,3 \cdot 400 + 0,6 \cdot 600) \tag{2.1}$$

V rovnici (2.1) je x sledovanou proměnnou. V případě proměnné A dosazením hodnot 310 a 250 mil. EUR vyjde výsledek 383,35 a 378,75 mil. EUR. Z tohoto pohledu tedy rozhodnutí není citlivé na změnu v odhadu střední hodnoty nákladů léčení při malé epidemii, protože hodnota spočtených nákladů nepřesáhla 420 mil. EUR.

Analogicky můžeme podle obr. 2.3 sestavit obdobné rovnice jako (2.1) i pro ostatní analyzované proměnné. Výsledky jsou zachyceny v tab. 2.2. Pro pozdější využití je v tabulce přidány i rozdíl mezi horní a dolní mezí nákladů.

Tabulka 2.2: Výpočet celkových nákladů při analýze citlivost proměnných příkladu Strom 2

Označení	**Proměnná**	**Náklady dolní mez** [mil. EUR]	**Náklady horní mez** [mil. EUR]	**Δ nákladů** [mil. EUR]
A	Epidemie - malé náklady	378,75	383,25	4,5
B	Epidemie - střední náklady	372	390	18
C	Epidemie - velké náklady	345	417	72
D	systém včasného varování	400	440	40
E	vakcinace (po varování)	390	435	45

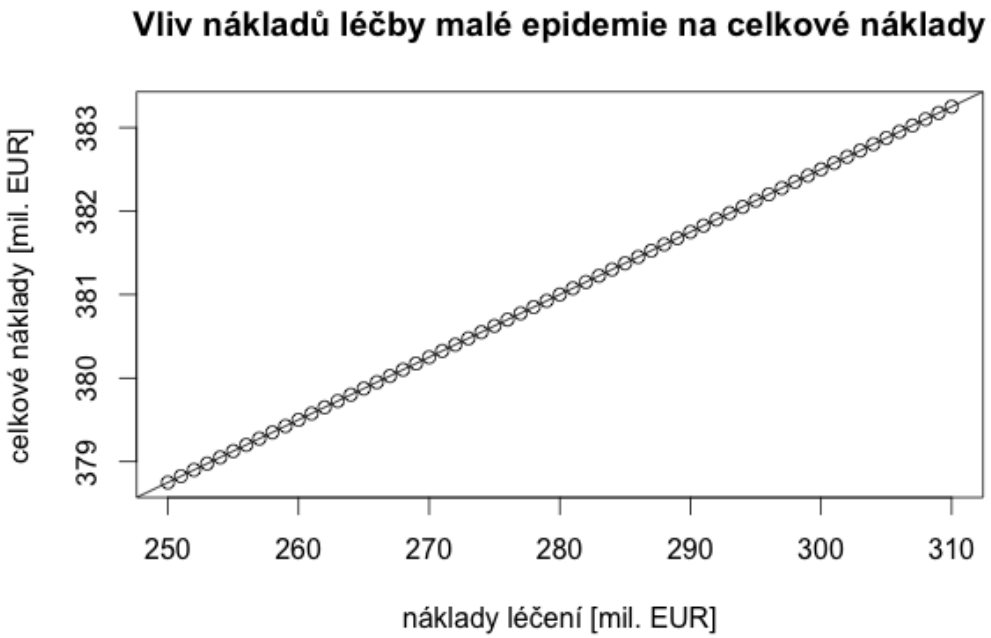
Z tab. 2.2 vyplývá, že rozhodovací situace stromu 2 není příliš citlivá na na změny v jednotlivých proměnných.

Výše uvedené skutečnosti je možné vizualizovat i graficky. Pro proměnnou A je vliv změny nákladů léčení vizualizován na obr. 2.4. Pro vygenerování grafu byl použit program R [19]. Skript řešení je v příloze. Podobným způsobem bychom mohli provést vykreslení všech analyzovaných proměnných, v tomto případě to ale nemá smysl.

Grafické znázornění citlivosti z grafu 2.4 není jediným možným typem znázornění. Pro vizualizaci citlivosti více analyzovaných proměnných se často používá tzv. *tornádový graf*. Tornádový graf se říká tomuto grafu podle typického trychtýřovitého tvaru. Jedná se o speciální druh sloupcového grafu, kde na ose y jsou vizualizovány jednotlivé analyzované proměnné a na ose x pak cílová proměnná. Analyzované proměnné jsou seřazeny sestupně podle rozsahu dopadů proměnné (pro náš příklad podle Δ nákladů z tab. 2.2).

Pro náš příklad je Tornádový graf zpracován na obr. 2.5.

Citlivost lze počítat taktéž pro jednotlivé pravděpodobnosti, způsob zpracování však z prostorových důvodů v těchto výukových textech zpracovávat nebudeme. Pokud máte zájem o podrobnější studium problematiky, můžete použít knihu Clemen & Reilly [28], která je pro tyto účely považována celosvětově za autoritativní studijní pramen.



Obrázek 2.4: Graf závislosti celkových nákladů na vývoji nákladů na léčbu malé epidemie (příklad Strom 2)

Výpočet a vizualizace citlivosti proměnných
Vyberte si některou z „neřešených“ proměnných (tedy B - E) a zkuste provést výpočet citlivosti podle mezí z tab. 2.1. Zkontrolujte, že výsledky odpovídají výsledkům v tab. 2.2.

Zkuste vytvořit graf závislosti celkových nákladů na zvolené veličině.

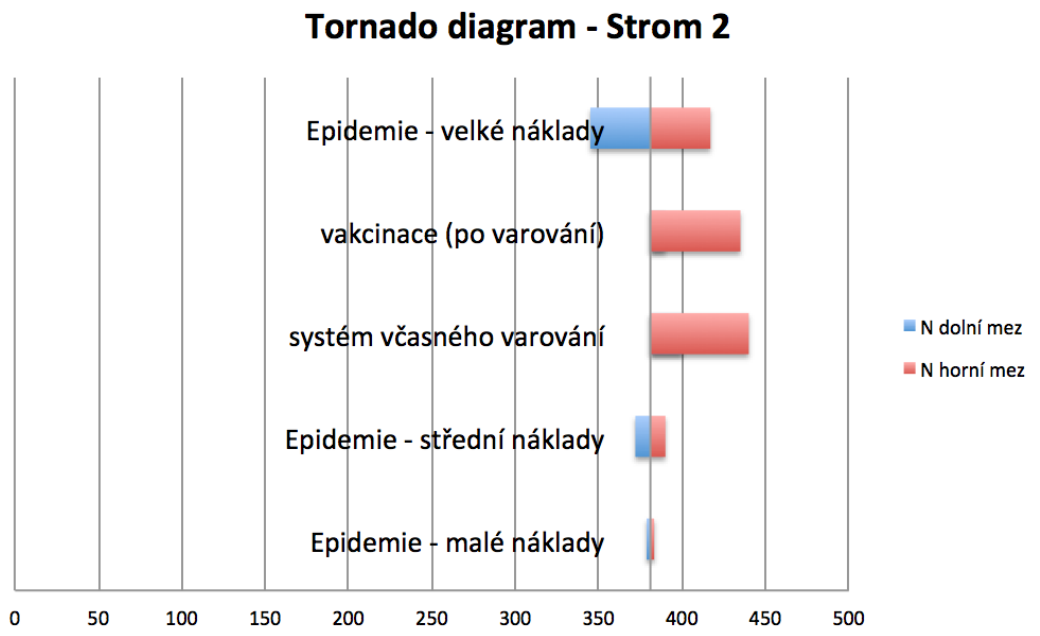
Softwarová podpora analýzy citlivost
Analýzy citlivosti je možné provádět prakticky v jakémkoliv nástroji, např. také tabulkovém procesoru MS Excel, je možné ale postupovat také jinak. Použít pokročilejší nástroje jako např. R, MathLab, SciLab a řada dalších, které se ovládají pomocí krátkých skriptů. Z tohoto důvodu se může na první pohled zdát, že použití takových nástrojů je pracnější a z počátku tomu tak může skutečně být, na druhou stranu poskytují tyto nástroje nesrovnatelně vyšší kontrolu nad všemi aspekty práce s daty a proto určitě má smysl si některý z těchto nástrojů vybrat a nastudovat jej.

R a SciLab jsou open source projekty a tudíž je můžete použít bezplatně prakticky k jakémukoliv účelu.

2.3 Multikriteriální pojetí rozhodovacích stromů

Rozhodovací stromy lze použít taktéž pro účely provedení multikriteriálního hodnocení. Postup, alespoň na počátku hodnocení je podobný jako v případě monokriteriálního použití rozhodovacích stromů. Tedy postupujeme následovně:

1. odvození struktury rozhodovacího stromu,
2. kvantifikace pravděpodobností výsledků jednotlivých dílčích rozhodnutí,
3. ocenění užitku ve *všech sledovaných* kritériích,
4. výpočet užitnosti v listovém uzlu *všech sledovaných kritérií*,
5. *odvození vah jednotlivých kritérií*,



Obrázek 2.5: Tornádový diagram proměnných a jejich vlivu na celkové náklady (příklad Strom 2)

6. *přepočet kritérií vyjádřených v naturálních jednotkách do podoby bezrozměrné (např. procento splnění představ v daném kritériu, nebo blízkost k ideálu)*,
7. *aplikace vah na kritéria a následně jejich součet*,
8. další výpočet probíhá standardně - na vypočtené bezrozměrné číslo z předchozího kroku aplikujeme běžným způsobem váhy, vypočítáváme užitnosti stochastických uzlů, rozhodujeme se pro výhodnější variantu v uzlu deterministickém.

Rozdíly proti běžnému stochastickému rozhodovacímu stromu jsou v odrážkách zvýrazněny kurzívou. Při výpočtech je potřeba obzvláště dbát na zvolenou převodní škálu. V případě stromu 2 např. počítáme s náklady - optimální varianta rozhodnutí bude pak taková, jejíž náklady budou nejmenší. Pokud ale převodní škála pro kritérium celkových nákladů bude blízkost k nějaké ideální variantě např. procenty, pak se měřítko jakoby „obrací“. Toto je nutné zohlednit v deterministických uzlech a konečně i při aplikaci váhových koeficientů a součtu užitku plynoucího z jednotlivých variant.

Záludnost stanovení vah spočívá v tom, že váhy při rozhodování odrážejí do určité míry subjektivní postoje hodnotitele. Váhy tedy nejsou objektivní, ale subjektivní. Velkou pozornost je proto nutné věnovat způsobu, kterým je odvodíme. S některými metodami pro mapování preferencí budeme pracovat v následujících kapitolách.

Shrnutí

Za určitých okolností, lze rozhodovací situace převést do podoby stromu. Takovému stromu říkáme *rozhodovací strom*. V praxi se střetáváme zejména s variantou tzv. *stochastického rozhodovacího stromu*, která analytikovi umožňuje zachytit nejistoty spojené s rozhodováním odhadem pravděpodobností následků.

Ačkoliv je je rozhodovací stromy možné použít i pro multikriteriální rozhodování, je obvykle tato metoda využívána pro výpočty spojené s rozhodováním o jediném kritériu, často vyjádřeného peněžní formou.

Po provedení analýzy je často nutné ještě zhodnotit nakolik je provedená analýza náchylná ke změnám vyvolaných drobnými změnami hodnot parametrů začleněných do rozhodování. K tomuto účelu se využívá *analýza citlivosti*.

Otázky

1. Jaká jsou pravidla pro vazby jsoucí ze stochastického uzlu?
2. Jaka jsou pravidla pro stanovení užitnosti deterministického uzlu?
3. Co je a k čemu slouží tornádový diagram?
4. Co je účelem analýzy citlivosti?
5. Ve skriptech výše je popsán způsob tvorby grafu závislosti cílové proměnné na změně jednoho vstupního parametru. Zkuste se zamyslet nad tím, zda by podobným způsobem nešlo zpracovávat graficky více proměnných najednou - co by v takovém případě bylo výstupem?

Kapitola 3

Multikriteriální analýza

Náhled kapitoly
V této kapitole se dozvíme něco o různých metodách analýzy rozhodovací situace, kdy optimalizujeme rozhodnutí z hlediska několika rozhodovacích kritérií. Zaměříme se přitom na multikriteriální rozhodovací analýzu, kterou si rozebereme jak teoreticky, tak na praktickém příkladu,.

Po přečtení této kapitoly budete
umět

1. vytvořit multikriteriální rozhodovací analýzu

znát

1. metodu párového srovnávání
2. základy práce s hierarchiemi kritérií a jak se jim vyhnout

Čas pro studium
Pro prostudování kapitoly budete potřebovat přibližně 2 hodiny.

3.1 Rozhodovací analýza

Rozhodovací analýza je jedním z nástrojů, který pomůže s rozhodnutím v případě, že existuje několik kritérií, podle kterých hodnotíme jednotlivé varianty řešení. V takovém případě, nám nástroje, jako jsou rozhodovací stromy, příliš nepomohou – ty zvládají jedno nebo několik málo kritérií (a to ne příliš pohodlně). Při práci s více kritérii se u těchto metod podstatně zvyšuje komplexita zpracování analýzy.

Multikriteriální rozhodovací analýza je proto k takovému hodnocení vhodnější. Počet kritérií zde není omezen.

Úkolem rozhodovací analýzy je přehledně zpracovat dostupné informace a pokud možno objektivně je zanalyzovat s cílem doporučit jednu, nebo několik málo variant k realizaci. Rozhodovací analýza je podklad pro rozhodování managera, tedy obvykle jiné osoby než analýzu vytváří, často dokonce z jiného oboru než je předmět rozhodnutí.

Jak tedy rozhodovací analýzu vytvořit. Hodnocení provádíme v několika krocích:

1. stanovení problému,
2. analýza informací,
3. hodnocení užitku variant,
4. hodnocení rizik variant,

5. finální verdikt – doporučení.

Jako první krok je potřeba vymezit problém. Správné vymezení problému nám umožní vymezit hodnotící kritéria, podle kterých můžeme hodnotit vhodnost realizace dané varianty.

Analýzou informací rozumíme shromáždění a vyhodnocení především *analytických informací*, tedy informacích o jednotlivých variantách řešení, kritériích, způsobech možného hodnocení, zkušenosti jiných lidí s řešením podobných rozhodovacích problémů apod. Kromě toho shromažďujeme také informace *námětové*.

Námětovými informacemi rozumíme informace vedoucím k fundamentálně odlišnému systému rozhodování, tedy vedou ke změně v chápání samotné rozhodovací situace. Nejlépe je asi demonstrovat tento typ informací na nějakém příkladu: mějme problém koupě nového auta – analytické informace se budou týkat jednotlivých značek, hodnotit se bude podle vlastností, námětovou informací by ale mohly být technické a finanční parametry spojené generálkou stávajícího auta, možnost koupě ojetého vozu nebo třeba outsourcing dopravy.

K realizaci některé „námětové“ varianty by se mohlo přistoupit v okamžiku, kdy zjistíme, že na nové auto nejsou v daném okamžiku dostupné zdroje.

3.2 Hodnocení užitku variant

Abychom mohli hodnotit užitek jednotlivých variant musíme již „napevno“ stanovit kritéria hodnocení a tato kritéria nějak vyčíslit. Kritéria i jejich kvantifikace probíhá v naturálních jednotkách daného kritéria a měla vyplynout se shromážděných analytických informací. Vhodné je tyto informace zpracovat do formy tabulky, viz tab. 3.1.

Tabulka 3.1: Kvantifikace kritérií v naturálních jednotkách pro jednotlivé varianty

kritérium	**jednotka**	**V1**	**V2**	**...**	**Vn**
K1	[min]	10	11		
K2	-	dobrý	výborný		
...					
Kn	[kg]	0,5	0,37		

Kritéria pro jednotlivé varianty kvantifikujeme v jejich přirozených jednotkách, ať už se jedná o jakékoliv jednotky. Mohou existovat i kritéria, která hodnotíme podle nějaké stupnice, jako třeba známkování ve škole, v takovém případě s nimi nemusí být spojena žádná jednotka. Tedy při kvantifikaci kritérií máme relativně vysokou volnost. Takto vyčíslená kritéria však nejsou přímo srovnatelná, neboť nelze porovnávat kritéria vyčíslená v odlišných jednotkách.

Dalším problémem mohou být také odlišnosti v kritériích z pohledu optimalizace - některá kritéria je potřeba v analýze maximalizovat, zatímco jiná je nutno minimalizovat. Odlišné přístupy pak tvoří další přirozenou překážku přímého porovnání kritérií.

I při známkování, nebo kterékoliv jiné zvolené stupnici obvykle preferujeme nějaké hodnocení nad jiným. Informace o interpretaci je nutno specifikovat do komentáře následujícího za výše uvedenou tabulkou.

Analytik provádějící rozhodovací analýzu by také neměl předpokládat, že údaje v tabelární formě jsou dostatečné z hlediska interpretace výsledků analýzy managerem, v jehož kompetenci rozhodování je. Z tohoto důvodu by jednotlivá kritéria i způsoby hodnocení měly být taktéž rozebrány v komentáři k tabulce.

Hodnota kritérií v naturálních jednotkách je pro nás přímo nepoužitelná, protože nelze přímo srovnávat ona slavná „jablka a hrušky“. Z tohoto důvodu je nutno kritéria kvantifikovaná v naturálních jednotkách převést na jednotnou hodnotící škálu. Taková škála může být v podstatě jakákoliv - nejsnadnější (a také nejvíce intuitivní) je ale převod hodnot do podoby procenta spokojenosti s danou hodnotou kritéria varianty – tedy rozhodnutí na kolik se výše daného kritéria blíží nějaké pomyslné ideální variantě (100 %). Tímto přepočtem získáme *matici prostých užitností*, jako je v tabulce 3.2.

Takto získané hodnoty jsou již srovnatelné. Navíc pro ně vždy platí, že čím je číslo větší, tím je naše spokojenost se výsledkem hodnoceného kritéria dané varianty větší - máme tedy zjednodušenou interpretaci hodnot.

Tabulka 3.2: Matice prostých užitností

kritérium	**V1**	**V2**	**...**	**Vn**
K1	80	10	...	100
K2	75	50	...	100
...	...	...	...	...
Kn	60	100	...	50

Problémem však je, že při rozhodování neklademe obvykle stejný důraz na všechna kritéria. Musíme tedy vytvořit také systém vah, kterými budeme transformovat matici prostých užitností. Velikost vah jednotlivých kritérií můžeme určit prakticky libovolně, velmi často se používají metody založené na metodách tzv. *párového srovnání*, jako je třeba metoda *Fullerova trojúhelníku* a další.

Principiálně jde o to, že postupně porovnáváme každou možnou dvojici kritérií mezi sebou a rozhodujeme, které z nich preferujeme. Svou preferenci zapíšeme a pokračujeme v porovnávání. Po ukončení porovnávání pak spočteme počty preferencí jednotlivých kritérií a vytvoříme pořadí. Toto pořadí nám může posloužit jako podklad pro navržení vah, protože častěji preferovaná kritéria budou očividně důležitější než tak která byla preferována méně, nebo dokonce vůbec.

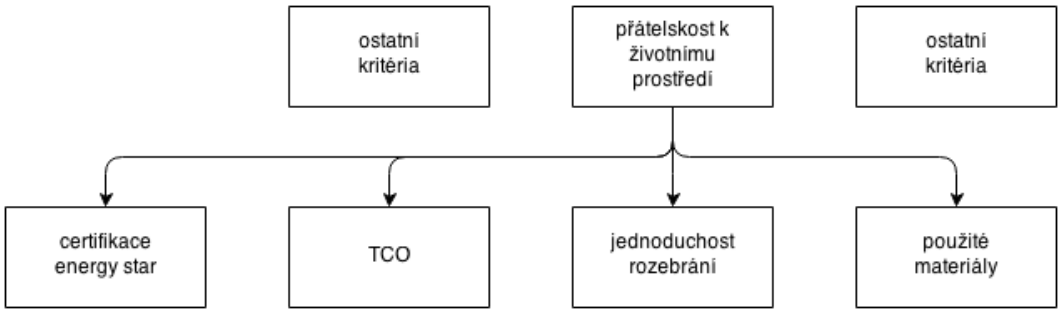
Tento postup je možný pouze v případě, že všechna kritéria jsou svým postavením na stejné úrovni. Stejnou úrovní se myslí především to, že kritéria netvoří tzv. *hierarchii*. Jak si představit hierarchii kritérií a v čem spočívá problém?

Mějme problém: potřebujeme zakoupit nový monitor a rozhodujeme se o značce a parametrech, které by takový monitor měl splňovat. Mimo jiné nás zajímají ekologické parametry monitoru. Při zpracování analytických informací jsme zjistili, že u monitorů se sledují následující charakteristiky, relevantní pro hodnocení „ekologičnosti“ monitoru:

- certifikace energy star
- certifikace TCO
- možnost jednoduchého rozebrání
- použité materiály
- environmentálně přátelský výrobek (environmental friendly)

Jelikož víme, že se snažíme demonstrovat hierarchii kritérií, lze vzájemné závislosti mezi kritérii snadno identifikovat. Základním kritériem může být přátelskost k životnímu prostředí. Míra této přátelskosti je určena velikostí spotřeby (certifikace energy star), vyzařování (TCO), jednoduchostí rozebrání a nepoužíváním materiálů, které by zatěžovaly životní prostředí více než je nezbytně nutné.

Tuto hierarchii můžeme znázornit také graficky, viz obr. 3.1.



Obrázek 3.1: Hierarchie ekologických kritérií pro rozhodnutí o koupi monitoru

Teď když vidíme problém, můžeme začít pracovat na jeho odstranění. Řešení jsou v zásadě možná dvě - buďto změníme metodu výpočtu a použijeme některou, která podporuje práci s hierarchiemi nebo dokonce sítěmi (např. metody *Analýza hierarchie procesu* (Analytic Hierarchy Process (AHP)) nebo *Analýza síťového procesu* (Analytic Network Process (ANP)), nebo upravíme systém kritérií tak, aby neobsahoval hierarchii.

Nejjednodušší způsob na vypořádání se s hierarchií je převedením uzlů stromu hierarchie do podoby listových (koncových) uzlů. V příkladu demonstrovaném na obr. 3.1 by to vypadalo tak, že z uzlu přátelskost k životnímu prostředí bychom udělali listový uzel a vše v nižších úrovních bychom zahodili.

Kritérium přátelskosti pak může v sobě může integrovat jakékoliv další - subkritéria. Jejich použití a stanovení užitnosti ale bude probíhat předem - mimo rozhodovací analýzu samotnou.

Svádět k použití by nás mohlo také opačné řešení založené na myšlence, pokud se přátelskost k životnímu prostředí skládá z TCO, materiálů atd., když použji tato subkritéria místo kritéria hlavního vyřeším problém taktéž. Ano ... za určitých okolností tomu tak skutečně může být, s tímto způsobem řešení je ale spojeno jedno velmi významné riziko a to konkrétně že neudržíme „správný poměr kritérií“.

Podstatu problému si můžeme lépe přiblížit pokud si představíme výsledek multikriteriální analýzy (MCA) jako index. Pro problém výběru monitoru pak mohu požadovat, aby určitou část indexu tvořila cena, další část technické parametry a pak také přátelskost k životnímu prostředí. Pokud místo jednoho ekologického parametru budu mít najednou čtyři, bude pro mě, jako hodnotitele, podstatně obtížnější zajistit, aby se podíl ekologických kritérií nezměnil.

Změny lze přitom uvažovat oběma směry. Rozdělením na subkritéria totiž mohu rozdrobit „sílu kritéria“ naopak subkritéria se za určitých okolností mohou také vzájemně podporovat, což může vést k vyšší než žádoucí výši váhy kritéria.

Řešení těchto problémů, jak vidno, není snadné a je plně v režii hodnotitele (analytika).

Metody pro hierarchie a sítě kritérií
Z prostorových důvodů se jednotlivým metodám pro práci s hierarchiemi a nebo sítěmi nemůžeme zabývat. Dobrou zprávou pro Vás je, že jelikož tyto metody jsou velmi známé, existuje velké množství teoretických výkladů metod i studií, které tyto metody prakticky aplikují.

Pro metodu ANP, lze doporučit knihu autora metody Saatyho Decision Making with the Analytic Network Process: Economic, Political, Social and Technological Applications with Benefits, Opportunities, Costs and Risks (Saaty [44]).

Pro metodu AHP je možné použít knihu autora (Saaty a Vargas [43]), existuje řada textů, které metodu rozebírají v dostatečné kvalitě, např. Bhusham a Kanwal [26] a další.

Vraťme se zpět k metodě běžného párového srovnání. Zkusme demonstrovat toto srovnání na variantách K1 – K5.

K1
K2 K2
K1 K2 K4
K2 K3 K2 K1
K4 K2
K3 K4
K5
K5

Jak si předchozí pyramidu vysvětlit? Postupujeme po sloupcích. *První sloupec* obsahuje přehled všech kritérií. Ve *druhém sloupci* srovnávám K1 a K2, preferuji K2, K2 a K3, preferuji K2, K3 a K4, preferuji K4 a konečně K4 a K5, preferuji K5.

Ve *třetím sloupci* hodnotím ob kritérium, srovnávám tedy K1 a K3, K2 a K4, K3 a K5, v dalších sloupcích hodnotím ob 2 kritéria, 3 ... podle počtu kritérií. Logicky čím více kritérií, tím rozsáhlejší bude konstruovaná pyramida. Nyní vypočteme pořadí. Při výpočtu lze vynechat první sloupec pyramidy, který obsahuje všechna kritéria a tudíž nám nepomůže při zhodnocení celkové významnosti kritéria.

Výpočet vah je demonstrován v tab. 3.3.

V tabulce 3.3 jsou váhy vypočtené jako obrácené pořadí, tedy nejvíce preferovaná varianta dostává největší váhu. Pro srovnání některých kritérií se může zdát takové určení vah jako příliš rigidní. Abychom získali větší rozestup mezi kritérii můžeme takové váhy násobit (2x, 10x). Je ale potřeba pamatovat na to, že váhy nám ovlivní zásadním způsobem výsledek, takže bychom měli být připraveni zdůvodnit, proč byla daná výše vah zvolena právě takovým způsobem.

Tabulka 3.3: Výpočet vah

Kritérium	**výskyt**	**pořadí**	**váha**
K1	2	2	2
K2	4	1	3
K3	1	3	1
K4	2	2	2
K5	1	3	1

V případě, že některá kritéria se dělí o stupně vítězů (třeba K1 a K4), lze přistoupit k jemnějšímu hodnocení, mohu stanovit třeba váhu K2 na 3,1 a K4 na 2,9 s tím, že během párového srovnání jsem K1 preferoval nad K4.

V každém případě by mělo být splněno, že více preferované kritérium nesmí mít menší váhu než méně preferované kritérium, například K2 nesmí mít menší váhu než K1 nebo K3.

Metoda Fullerova trojúhelníku dává dohromady párové srovnání a výpočet vah. Tento typ párového porovnání je znázorněn v tab. 3.4. Je však potřeba poznamenat, že oba postupy vedou k totožným výsledkům, takže používejte takový, který Vám připadá intuitivnější.

Tabulka 3.4: Fullerův trojúhelník pro výpočet vah

Porovnání kritérií				**kritérium**	**výskyt**	**pořadí**	**váha**
1	**1**	1	**1**	K1	2	2	2
2	3	**4**	5				
	2	**2**	**2**	K2	4	1	3
	3	4	5				
		3	**3**	K3	1	3	1
		4	5				
			4	K4	2	1	2
			5				
				K5	1	3	1

Preference kritéria je v tab. 3.4 zvýrazněna tučně.

Na základě stanovení vah můžeme sestavit *matici vážených užitností*, viz tab. 3.5.

Tabulka 3.5: Matice vážených užitností

kritérium	**váha**	**V1**	**V2**	**M**
K1	2	79 x 2 = 158	174	200
K2	3	120	300	300
K3	1	...		
K4	3	...		
K5	1	...		
$\sum$		278	474	500
U	%	55,6	94,8	100

Kontroverze okolo váhových koeficientů

V tab. 3.4 a 3.3 jsou váhy stanoveny v intervalu 1 - 3, to však není jediný možný způsob stanovení vah, neboť škály vah mohou být v zásadě libovolné za předpokladu, že bude zachována praktická proporcionalita hodnot vah, tedy např. že kritérium s váhou 3 je 3x významnější než kritérium s váhou 1. Po praktické stránce se často doporučuje používání vah v intervalu 0 - 1, což v podstatě odpovídá procentům a tudíž takové váhy mohou být významově snadněji interpretovatelné.

Při stanovení vah je vždy nutné přihlížet ke způsobu, jakým s nimi bude manipulováno a na základě toho je potřeba přizpůsobit škály.

Matici vážených užitností sestavíme tak, že hodnotu kritérií z tabulky prostých užitností vynásobíme vahou k danému kritériu přináležejícímu. Kromě toho stanovujeme také ideální variantu M. Hodnotu kritérií ideální varianty stanovíme buď jako maximum kritéria z přítomných variant nebo jako maximální teoreticky možná hodnota kritéria, tedy váha x 100. Oba přístupy vedou ke stejnému pořadí variant při hodnocení podle užitku, mohou však vést ale k odlišnostem při stanovení optimální varianty s přihlédnutím k užitku i k rizikům.

Z praktického hlediska pokud M variantu stanovíme jako maximum z existujícího, budou jednotlivé varianty v hodnocení procentně blíže variantě M než kdyby M byla nastavena jako ideální varianta. Stejnou volbu pro variantu M je následně nutné použít i pro hodnocení rizik.

V tabulce 3.5 jsem zvolil druhý způsob - tedy varianta M je ideální.

Vážené hodnoty kritérií sečteme a vypočteme nakolik se blíží ideální variantě. Součet hodnot kritérií ideální varianty mi představuje 100 % a z toho tento údaj už jednoduše dopočítáme.

3.3 Hodnocení rizika

Hodnocení rizik variant je nutné provést samostatně. Riziko se obvykle udává pravděpodobností nebo frekvencí vzniku nežádoucího efektu za určité období. Z hlediska hodnocení je výhodnější použít pravděpodobnostní charakteristiku rizika s tím že období, pro které jsou rizika určena, musí být pro všechna rizika stejná. Výši rizik můžeme přehledně zobrazit v *matici prostých rizik* (viz tabulka 3.6).

Tabulka 3.6: Matice prostých rizik

Riziko	**V1**	**V2**	**Vn**
R1	10 %	8 %	...
R2	15 %	25 %	...
Rm	...	...	...

Podobně jako u kritérií používaných při hodnocení užitnosti variant ani rizika, pokud je srovnáme mezi sebou, nebudou mít stejnou váhu. Váhu můžeme určit podobně jako u kritérií užitnosti. Aplikací vah na matici prostých rizik získáme *matici vážených rizik* (viz tabulka 3.7).

Tabulka 3.7: Matice vážených rizik

Riziko	**váha**	**V1**		**V2**		**Vmax**	
		p	**SO**	**p**	**SO**	**p**	**SO**
R1	1	0,1	0,1	0,08	0,08	1	1
R2	2	0,15	0,3	0,25	0,5	2	2
$\sum$		-	0,4	-	0,58	-	3
SO	[%]		13,3		19,33		100

V matici vážených rizik pracujeme s tzv. stupněm ohrožení (stupěň ohrožení (SO)), jedná se o vážené riziko, tedy hodnota vypočtená jako váha x pravděpodobnost vzniku negativního efektu. V matici také konstruujeme speciální variantu Vmax, v rámci které stanovujeme maximálně nepříznivou variantu z hlediska rizik.

Pro stanovení Vmax lze použít dva přístupy analogicky k stanovená M varianty při posuzování užitku. Tady můžeme pravděpodobnost realizace rizika stanovit na 100%, získáme tak maximálně nepříznivou variantu. Alternativně je možné vzít existující maximum z některého SO reálné varianty. Následně můžeme zhodnotit nakolik se reálné varianty blíží variantě Vmax.

Výpočtem matice vážených rizik jsme shromáždili všechny potřebné údaje pro přijetí rozhodnutí, je nutné je pouze konsolidovat do přehledné formy. Nejprve srovnejme pořadí variant z hlediska užitku a rizik, které jsme získali výše (v tabulkách 3.5 a 3.7).

Shrnutí je dostupné v tabulce 3.8.

Pamatujte, že užitek se snažíme maximalizovat, zatímco riziko minimalizujeme!

Tabulka 3.8: Užitek vs riziko

Pořadí	**1**	**2**
užitek	V2	V1
riziko	V1	V2

3.4 Výsledný efekt a finální hodnocení

Kromě takového pořadí můžeme vypočítat také tzv. *výsledný efekt*, který chápeme jako rozdíl užitku a rizika nebo podíl užitku a rizika. Výsledek výpočtu je dostupný v tabulce 3.9.

Tabulka 3.9: Výsledný efekt

	V1	**V2**
Užitek (U)	55,6	94,8
Riziko (R)	13,3	19,33
Výsledný efekt (U-R)	42,3	75,47
Výsledný efekt (U/R)	4,2	4,9

Proč pohlížet na výsledný efekt z různých pohledů. Alternativní pohledy mohou přinášet celkem zajímavé srovnání. Zatímco U - R měří absolutní rozdíl ve vnímání rozdílů mezi užitkem a rizikem, U/R měří kolikrát je užitek větší než riziko - takový scénář pak očividně preferuje spíše malé hodnoty rizika.

Při hodnocení variant můžeme přijmou v zásadě tři různé strategie preference, které se výrazně liší svou deklarovanou averzí k riziku:

1. maximální
2. minimální
3. optimální

U **strategie maximální** se snažíme maximalizovat užitnost bez ohledu na případná rizika. Použitím **minimální strategie** naopak minimalizujeme rizika bez ohledu na užitnost variant. **Strategie optimální** zkoumá jak riziko, tak užitek.

Výsledek by se tedy dal interpretovat na našem příkladu následovně:

1. strategie maximální – realizovat V2
2. strategie minimální – realizovat V1
3. strategie optimální – realizovat V2

Celkově by tedy měla být doporučena varianta V2. Při větším množství variant, je pravděpodobné, že rozdíly mezi nimi nebudou tak velké. Lze tedy při doporučení zmínit i alternativní návrhy, které jsou pouze o něco horší než varianta vítězná.

Časté chyby v MCA

Pokud se ohlédnete zpět do kapitoly, je proces zpracování MCA poměrně přímočarý. Struktura analýzy je pevně daná a pro výpočet používáme pouze základní matematické operace, takže šance, že uděláme chybu jsou malé, že? Zkušenosti bohužel ukazují, že tomu tak není - chyby se dějí a udělat je není vůbec těžké. Chyby v rámci MCA přitom mohou být jak logické, tak numerické.

Typickou logickou chybou by mohlo být maximalizace rizik při výběru. Jedná se o chybu z nepozornosti - při zpracování analýzy se člověk někdy přepne do automatického režimu, při kterém ne úplně používá mozek. Pak když se dělá pořadí - vytvoří pořadí pro rizika od nejvyššího po nejmenší, ačkoliv z hlediska logického se jedná o očividný nesmysl.

Dobrou ochranou proti těmto chybám je tisk analýzy a její přečtení. Zkušenosti ukazují, že čtení textu na obrazovce většinou nevede k odhalení stejného množství chyb jako v případě, že totožný text čteme ve vytištěné formě.

Numerické chyby plynou z chybného převodu hodnot kritérií v naturálních jednotkách do podoby procent. Často se objevují také chybné výpočty procent v tabulce vážených užitností (nebo jiných tabulkách). Tento typ chyb lze omezit použitím tabulkových procesorů, jako je např. MS Excel, LibreOffice Calc a další, které umožňují část výpočtů automatizovat a provázat, takže v případě, že se provede změna v jedné tabulce, může být propagována dále - minimalizuje se šance, že změněnou hodnotu zapomeneme někde použít.
Oba typy chyb plynou z podcenění rozhodovací analýzy ve stylu, když na analýzu máme šablonu - musí to být jednoduché, ne? Rozhodování je ale obvykle obtížné, **takže si to ještě neztěžujme podceňováním MCA**.

Shrnutí

V MCA postupujeme následovně:

1. shromažďování analytických a námětových informací
2. stanovení kritérií a rizik, která budou použita pro rozhodování a stanovení variant řešení
3. sestavení matice užitností pro jednotlivé varianty řešení v naturálních jednotkách, její převod do podoby procent spokojenosti s hodnotou kritéria
4. zjištění nebo odvození vah a jejich aplikace na zjištěnou užitnost variant
5. vyhodnocení rizik a výpočet SO jednotlivých variant
6. výpočet výsledného efektu a stanovení maximální, minimální a optimální varianty řešení
7. finální zhodnocení rozhodování

Otázky

1. V čem spočívá problém s hierarchiemi při stanovování vah kritérií?
2. Jakým způsobem lze určit váhy kritérií?
3. Jak fungují metody párového srovnání?

Kapitola 4

Případová studie MCA

Příklad
Mějme jednoduchý příklad, na kterém budeme prakticky demonstrovat způsob zpracování rozhodovací analýzy. Zvoleným problémem je **výběr čtečky elektronických dokumentů**.

Čtečka by měla být určena pro náruživé čtenáře vyžadující maximální komfort při čtení. Čtečka by proto měla být založena na technologii elektronického inkoustu, který poskytuje maximální kontrast pro čtení i na přímém slunci. Preferované jsou také fyzická tlačítka pro otáčení stránek.

Poznámka k organizaci příkladu
Jelikož je tento příklad školní, je potřeba si uvědomit, že systém kritérií je pro něj zjednodušen (ne však příliš zjednodušen). Také je potřeba počítat s tím, že každá technologie zastarává, v době, kdy budete tento text číst, proto mohou parametry řešení vypadat poněkud archaicky.

Text analýzy je také opatřen vysvětlujícími poznámkami. Tyto poznámky jsou určeny k lepšímu pochopení způsobu tvorby analýzy a nejsou součástí *standardní analýzy*. Texty poznámek jsou odlišeny od textu analýzy kurzívou.

4.1 Informace

V současnosti je na trhu možná několik desítek různých druhů čteček využívajících elektronický inkoust, které se výrazně odlišují svou velikostí i dalšími vlastnostmi. Co do velikostí se čtečky dodávají zejména:

1. menší - uhlopříčka 5 palců
2. standard - uhlopříčka 6 palců
3. větší - uhlopříčka 9,7 palců

Pro účely analýzy budou vzhledem k požadavkům „náruživého čtenáře“ preferovány čtečky o úhlopříčce 6 palců (5 palců je málo, 8 palců je ideál odpovídající formátu A5 1:1, ale takové čtečky nejsou na trhu, vetší zařízení jsou zase příliš těžká a používají obvykle starší technologie pro display).

Z hlediska technologií se u v současnosti prodávaných zařízení objevují tři typy zobrazovacích technologií. Zejména u starších výběhových typů se používá display Vizplex, novější pak používají Pearl a Carta. Technologie Carta je však v současnosti v podstatě nedostupná pro zařízení jiných výrobců než je společnost Amazon.

Jelikož Vizplex technologie je používána u spíše výběhových zařízení, zaměří se analýza především na zařízení používající Pearl a Carta displaye.

Z hlediska použití je velmi důležitá ergonomie ovládání zařízení. Ovládání se pohodlnější pomocí dotykového display, ale pro samotné čtení může být výhodnější použití hardwarových tlačítek, aby se

omezilo ušpinění obrazovky. Komfort čtení taktéž zvyšuje nasvícení obrazovky pomocí led diod, čímž dochází k výraznému zvýšení kontrastu textu.

Dle způsobu pořizování může mít velký význam případná podpora různých formátů elektronických knih. Vyšší podpora dává čitateli větší volnost při volbě služeb elektronických knihkupectví. Podpora formátů, ale vzhledem k existenci relativně mocných konverzních nástrojů může být také považována pouze za podpůrné kritérium.

Přítomnost audio-jack může naznačovat možnost použití čtečky pro jiné účely něž čtení, např. poslech hudby nebo audio knih. Kvalita přehrávání je ale u těchto zařízení obvykle tristní. Rozšiřující slot pro SD karty může mít svůj význam v případě, že čitatel hodlá přenášet svou knihovnu z místa na místo, pro běžné čtení je kapacita většiny zařízení vzhledem k průměrné velikosti knihy okolo 1 MB dostatečná.

Naopak výhodná je případná podpora Wi-Fi, umožňující synchronizaci, nákup apod. bez nutnosti použít USB kabel a počítač.

Na základě výše zmíněných úvah byla do rozhodovací analýzy zařazena následující zařízení:

- V1 - Kobo Glo
- V2 - Sony Reader PRS-T3
- V3 - Kindle Paperwhite 2
- V4 - Pocketbook Touch Lux
- V5 - Bookeen Cybook Odyssey HD FrontLight

Všimněte si komentářů výše - specifikace případných parametrů použitých v rozhodování je do velké míry závislá na účelu použití. Zadání přitom v některých oblastech může být dosti neurčité - v takovém případě máme možnost buď odhadnout co se tím pravděpodobně myslelo, nebo lze jít zpět do zadání a požádat zadavatele o doplnění. Proces rozhodování tedy nutně nemusí být jednosměrný, lze se vracet k předchozím krokům a na základě nových poznatků je přepracovávat.

V případě, že pořízení specializované čtečky nebude možné realizovat, nabízí se možnost koupě z hlediska funkcí univerzálnějšího tabletu. Je potřeba poznamenat, že tablety jsou obvykle těžší, v řadě případů dražší a jejich baterie umožňuje provoz na nesrovnatelně menší dobu.

K analytickým informacím je potřeba přidat také některé námětové informace sloužící pro specifikaci alternativních problémů, které mohou být později zajímavé.

4.2 Užitnost

Tabulka 4.1: Kritéria v naturálních jednotkách

Varianty	**jednotky**	**V1**	**V2**	**V3**	**V4**	**V5**
K1 Display	-	Pearl	Pearl	Carta	Pearl	Pearl
K2 Váha	g	185	200	206	198	180
K3 dotykové ovládání	A/N	A	A	A	A	A
K4 hardware tlačítka	A/N	N	A	N	A	A
K5 front light	A/N	A	N	A	A	A
K6 baterie	počet otočení stran	2 100	1 700	1 700	1 700	1 900
K7 Wi-Fi	A/N	A	A	A	A	A
K8 cena	Kč	3 400	3 800	3 300	3 400	4 200

Legenda

Display - Novější Carta oproti starším Pearl displayům poskytuje vyšší kontrast. Z hlediska užitnosti bude proto Carta hodnocena 100 % a Pearl 80 %.

Váha - váha znamená pohodlnost pohodlnost při dlouhodobém čtení. Nicméně rozdíl ve vahách v tomto případě je minimální, nejvyšší hodnota váhy je ještě velmi přijatelná, proto je 206 g hodnoceno jako 85 %.

Dotykové ovládání - je přítomno na všech zařízeních, mezi kterými probíhá rozhodování. Toto kritérium proto nebude dále zasahovat do výběru (nepřispívá k rozhodnutí).

Hardware tlačítka - jedná se o tlačítka pro otáčení stránek. Nepřítomnost tlačítek je penalizována 50 %, neboť se jedná o vážný zásah do pohodlnosti práce se čtečkou.

Front light - nasvícení displaye pomocí led diod, kterou se výrazně zvyšuje kontrast při čtení, umožňuje čtení v tmavších prostorách. Nepřítomnost nasvícení je velkou vadou v užitnosti, proto se hodnotí pouze 20 %.

Počet otočení stran: každý výrobce uvádí životnost baterie jiným způsobem, např. Pro Kobo Glo uvádí výrovce výdrž 70 hodin, za předpokladu čtení 1 stránky za minutu, což vede k výslednému 2100 stran. Předpokládá se měření při vypnutém Wi-fi.

Samotný parametr otočení stran byl zvolen z toho důvodu, že display založený na technologii elektronického papíru odebírá energii pouze při překreslování stránky.

Wi-fi - je přítomno na všech zařízeních, toto kritérium proto nepřispívá k rozhodování a bude z další analýzy vyřazeno.

Cena - je podstatným parametrem, který je použit při rozhodování. Nejvyšší hodnota 4 200 je považována z hlediska užitnosti za vhodnou pouze z 50 %.

Tabulka 4.2: Užitnost kritérií

Varianty	**V1**	**V2**	**V3**	**V4**	**V5**
K1 Display	80	80	100	80	80
K2 Váha	97	88	85	90	100
K4 hardware tlačítka	50	100	50	100	100
K5 front light	100	20	100	100	100
K6 baterie	100	70	70	70	85
K8 cena	94	70	100	94	50

Všimněte si, že užitnosti nutně nemusí být pro varianty stanovovány v intervalu 0 - 100. Hodnoty přijatelnosti jsou odvozovány ze subjektivního vnímání přijatelnosti užitností spojených s hodnoceným kritériem.

Náběh užitnosti také nemusí mít lineární charakter v celém uvažovaném intervalu kritéria. Např. cena 4 200,- Kč u varianty V5 je hodnocena užitností 50 %, cena 5 000,- Kč by ale mohla mít např. pouze 10 %. Rozhodnutí o způsobu vyhodnocení je pouze na analytikovi, ten pak ale musí být schopen svůj postoj zdůvodnit, popř. obhájit.

Váhové koeficienty této rozhodovací analýzy byly odvozeny na základě měření preferencí zadavatele metodou párového srovnání. Zaznamenané preference jsou zachyceny níže.

Tabulka 4.3: Párové srovnání

K1 Display					
	K1				
K2 Váha		K1			
	K4		K5		
K4 hardware tlačítka		K5		K1	
	K5		K2		K1
K5 front light		K4		K2	
	K5		K4		
K6 baterie		K5			
	K6				
K8 cena					

4.3 Riziko

Při pořizování čteček se jeví jako rizikové následující:

- R1 - ukončení podpory výrobcem
- R2 - odmítnutí reklamace display

Tabulka 4.4: Odvození váhových koeficientů

Kritérium	**počet**	**pořadí**	**váha**
K1 Display	4	1	5
K2 Váha	2	3	2
K4 hardware tlačítka	3	2	4
K5 front light	4	1	5
K6 baterie	1	4	3
K8 cena	0	5	1

Tabulka 4.5: Matice vážených užitností

Kritéria	**váha**	**V1**	**V2**	**V3**	**V4**	**V5**	**M**
K1 Display	5	400	400	500	400	400	500
K2 Váha	2	194	176	170	180	200	200
K4 hardware tlačítka	4	200	400	200	400	400	400
K5 front light	5	500	100	500	500	500	500
K6 baterie	3	300	210	210	210	255	300
K8 cena	1	94	70	100	94	50	100
$\sum$		1 688	1 356	1 680	1784	1 805	2 000
užitek (U)	%	84,4	67,8	84	89,2	90,25	100

- R3 - ukončení činnosti e-shopu integrovaného ve čtečce (ztráta možnosti manipulace se zakoupenými knihami)

Tabulka 4.6: Matice rizik

Rizika	**V1**	**V2**	**V3**	**V4**	**V5**
R1 ukončení podpory	40	30	5	10	10
R2 odmítnutí reklamace	30	30	5	40	10
R3 ukončení činnosti e-shop	40	40	0	30	40

Tabulka 4.7: Párové srovnání rizik

R1 ukončení podpory		
	R2	
R2 odmítnutí reklamace		R3
	R2	
R3 ukončení činnosti e-shop		

Tabulka 4.8: Odvození váhových koeficientů rizik

Riziko	**počet**	**pořadí**	**váha**
R1 ukončení podpory	0	3	1
R2 odmítnutí reklamace	2	1	3
R3 ukončení činnosti e-shop	1	2	2

4.4 Výsledný efekt a vyhodnocení

Podle rozhodovací analýzy vychází jako zajímavé varianty V3 a V5, což jsou čtečky *Amazon Kindle Paperwhite 2 (V3)* a *Bookeen Cybook Odyssey HD Frontlight (V5)*. Kindle lze považovat za sázku na jistotu s velmi dobrý servisem a podporou ze strany výrobce i poskytovatelů obsahu. Cybook oproti tomu boduje zajímavými technickými vlastnostmi, investici do něj je však možno vidět jako o něco

Tabulka 4.9: Matice vážených rizik

Riziko	**váha**	**V1**		**V2**		**V3**		**V4**		**V5**		**Vmax**	
		p	**SO**	**p**	**SO**	**p**	**SO**	**p**	**SO**	**p**	**SO**	**p**	**SO**
R1 ukončení podpory	1	0,4	0,4	0,3	0,3	0,05	0,05	0,1	0,1	0,1	0,1	1	1
R2 odmítnutí reklamace	3	0,3	0,9	0,3	0,9	0,05	0,05	0,4	1,2	0,1	0,3	1	3
R3 ukončení činnosti e-shop	2	0,4	0,8	0,4	0,8	0	0	0,3	0,6	0,4	1,2	1	2
$\sum$			2,1		2		0,1		1,9		1,6		6
SO			35		33,3		1,7		31,7		26,7		100

Tabulka 4.10: Výsledný efekt

	V1	**V2**	**V3**	**V4**	**V5**
Užitek (U)	84,4	67,8	84	89,2	90,25
Riziko (R)	35	33,3	1,7	31,7	26,7
Výsledný efekt (U-R)	49,4	34,5	82,3	57,5	63,3
Výsledný efekt (U/R)	2,4	2	49,4	2,8	3,4

Tabulka 4.11: Vyhodnocení

	1	**2**	**3**	**4**	**5**
podle užitku (strategie maximální)	V5	V4	V1	V3	V2
podle rizika (strategie minimální)	V3	V5	V4	V2	V1
strategie optimální (U-R)	V3	V5	V4	V1	V2
strategie optimální (U/R)	V3	V5	V4	V1	V2

rizikovější, jelikož francouzská společnost Bookeen je společností malou (ve srovnání s Amazon) - s tím pak mohou být spojeny problémy s reklamacemi apod.

Kapitola 5

Další metody použitelné v obecných rozhodovacích situacích

Náhled kapitoly
V této kapitole se seznámíme s některými dalšími metodami, které mohou poskytnout cenné informace pro přijetí správného rozhodnutí v prakticky kterékoliv rozhodovací situaci.

Po přečtení této kapitoly budete
vědět

- jak vytvářet myšlenkové mapy,
- co je to Delfská metoda a jak se používá

znát

- jaké problémy jsou spojeny se shromažďováním a vyhodnocováním údajů z dotazníkových šetření a jak je minimalizovat

Čas pro studium
Pro prostudování této kapitoly budete potřebovat minimálně hodinu, více pokud si zkusíte vyřešit prakticky úkoly pomocí softwarových nástrojů.

5.1 Delfská metoda

Delfská metoda byla vyvinuta společností RAND v padesátých letech minulého století a jako vedlejší produkt výzkumu financovaného letectvem USA známém jako „Projekt Delfy“. Tento projekt byl zaměřen na snahu o odhad průmyslových cílů hypotetického Sovětského útoku na USA. Základním problémem byl předpokládaný nedostatek informací - tedy pracuje se s určitým vnímáním situace, podpořeným tvrdými daty tam, kde je to možné, ale se silným vlivem intuice lidského faktoru.

Delfská metoda se v průběhu času stala velmi oblíbenou a často nasazovanou pro různé rozhodovací situace. Tato metoda se také poměrně výrazně v čase vyvíjela, dnes proto existuje řada variant této metody lišících se způsobem použití expertů, zpracováváním jejich odpovědí apod.

Podívejme se ale na metodu prakticky - v čem vlastně spočívá. Delfská metoda je založena na vyhodnocování expertních odhadů. Používá se zejména tam, kde se rozhoduje o specifických, z odborného hlediska složitých záležitost, v situacích, kdy vývoj do budoucna není jistý a nejsou dostupná „tvrdá data“ použitelná pro podporu rozhodování.

Metoda se realizuje obvykle v několika krocích:

1. specifikace problému
2. výběr expertů
3. vytvoření dotazníku pro získání informací od expertů
4. vytvoření dokumentace k dotazníku
5. rozeslání dotazníku s dokumentací k expertům
6. vyhodnocení odpovědí
7. v případě nejasností opakovat od kroku 3, s tím, že se v dalších iteracích zasílá i celkové zhodnocení již odevzdaných dotazníků
8. opakování může probíhat libovolně dlouho (opakujeme dokud nedostaneme výsledky, které přinášejí jasnou odpověď - pozor ne nutně odpověď, kterou bychom si přáli, nebo dokonce odpověď zaručeně správnou).
9. konečné zhodnocení

Přestože výše uvedený seznam je dlouhý, na první pohled se nezdá, že by byl příliš složitý. Bohužel, praktické zkušenosti ukazují, že se jedná o metodu, která je časově i finančně velmi nákladná a výsledky, kterých je dosaženo mohou být snadno znehodnoceny zaujatostí použitých expertů, nevhodnou skladbou otázek nebo nevhodným složením expertů.

Metoda samotná však přináší také celou řadu výhod. Funguje i tam, kde buďto nejsou dostupná data vůbec nebo jich je dostupných naopak tolik, že je není možné efektivně zpracovat pomocí jiných např. statistických metod. Metoda je velmi dobře schopna eliminovat extrémní názory, které jsou v přímém rozporu s širším odborným konsenzem.

Eliminace extrémních názorů může však být vnímána také jako nevýhoda - to že je názor extrémní nemusí nutně znamenat, že je také chybný.

Jedním z určujících faktorů rozhodujícím o úspěchu nebo neúspěchu nasazení metody je výběr expertů. Na experty máme poměrně velké požadavky:

- musí být experty zaměřenými určitý aspekt problému nebo odborníci zaměření více obecně na řešený problém (v oslovených expertech by měli být zástupci obou skupin)
- experti musí být nezávislí
- experti musí být ochotni a schopni odpovědět na předložené otázky

Požadavek na odbornost expertů a její různé aspekty jsou intuitivně jasné - zjednodušeně se chceme vyhnout situaci, kdy pro samé stromy neuvidíme les (a opačně kdy uvidíme les, ale nedokážeme rozlišit stromy).

Obtížnější je pochopení otázky *nezávislosti expertů*. Pokud vyslovíme slovo závislost směrem k člověku, pak nás možná napadne celá řada různých interpretací. Závislost může být finančního charakteru ve smyslu *koho chleba jíš, toho píseň zpívej*, závislosti mohou nabývat také mnohem sofistikovanějších podob.

Obecně pro účely delfské metody můžeme rozlišovat mezí závislostí ve smyslu zájmu na určitém výsledku rozhodování (bias - viz dále) a také ve smyslu závislosti na stejných nebo podobných informacích. V situacích za akutního nedostatku informací se často pracuje s takovými informacemi, které jsou k dispozici, i když nejsou přesné nebo dokonce správné. Pokud jsou data zatížena nejistotou - bude s jejich použitím spojena systematická chyba, se kterou je nutno počítat. Problémem je, že tuto chybu si expert nutně nemusí uvědomovat.

Jde o to, že každý člověk se snaží utvrdit správnost svých názorů, z tohoto důvody předmětnou problematiku diskutuje, studuje odbornou literaturu a na základě takto zjištěných údajů změní třeba názor a tento názor dále šíří a utvrzuje v něm další odborníky. Z hlediska odborníka se tak stokrát opakovaná lež může stát „pravdou“. Bohužel se nejedná o pravdu objektivní – jinými slovy i odborník může žít v bludu.

V literatuře [30] jsem se setkal s poměrně silným pojmenováním výše uvedeného. Autor takovýto kolektivní blud označuje jako datový incest. Popisuje situaci několika nezávislý vzájemně komunikujících robotů - agentů, *z nichž jeden s pravděpodobností 50 % našel cíl a posílá tuto informaci agentovi 2, agent 2 získává informaci, že agent 1 možná vidí cíl a posílá informaci agentovi 3, agent 3 věří agentovi 2, že cíl je na určitých souřadnicích a tuto skutečnost sdělí agentovi 1, který posiluje své přesvědčení o lokaci cíle na 100 % a to přesto, že se neobjevily žádné nové informace, které by tuto domněnku nějak potvrdily.*

Způsob vyhodnocování otázek se odvíjí od typů odpovědí, odpovědi lze typově rozdělit na:

1. numerické

2. ordinální (výběr z několika, předem daných, odpovědí)
3. tvořená odpověď

Je očividné, že různé typy odpovědí vyžadují zpracování odlišným způsobem. Numerické odpovědi reprezentují číselné vyjádření určité spojité veličiny, proto při zpracování hodnotitelé budou preferovat základní statistické vlastnosti jako je průměr, medián, rozložení odpovědí apod. Extrémní hodnoty mají potenciál silně vychýlit některé statistiky, srovnejte například průměrnou mzdu a medián mzdy v ČR. To je také důvodem, proč jsou v rámci vyhodnocování často extrémní hodnoty vyřazovány.

Nabízí se zajímavá otázka - *co je možné považovat za extrémní hodnotu*? Statistika má nástroje pro identifikaci hodnot v rámci statistického souboru, problém je, že statistika předpokládá, že „naměřené" hodnoty jsou náhodné, což v případě mapování názorů odborníků nebude nejspíše pravda. Z tohoto důvodu je rozhodnutí o hranici extrémů čistě na nás jako hodnotitelích. Určitá vodítka lze hledat ve studiích, které použily Delfskou metodu v minulosti. Některé z nich jako extrémní považovaly v horním a dolním kvartilu souboru odpovědí - tedy dolních a horních 25 % odpovědí.

V některých případech ale tato hranice může být příliš úzká. Posuzování proto musí probíhat **případ od případu**.

V případě, že odpovědi jsou ordinálního charakteru - respondent tedy vybírá z omezené množiny možných, předem stanovených odpovědí, většina statistik používaných pro vyhodnocení numerických odpovědí pozbývá smysl. V tomto případě by proto vyhodnocování mělo být zaměřeno spíše na zkoumání četností odpovědí.

- „Vyčnívá" některá odpověď z hlediska četnosti nebo jsou odpovědi rozloženy rovnoměrněji?
- Existuje některá odpověď, která byla preferována minimálně nebo vůbec? Pokud ano můžeme ji v dalších úvahách např. ignorovat.
- apod.

Grafický lze rozložení odpovědí dobře vizualizovat s použitím např. koláčových grafů nebo histogramů.

Tvořené odpovědi jsou z hlediska vyhodnocování nejsložitější. Složitost spočívá především v tom, že každá odpověď je v podstatě originální a vyžaduje ruční vyhodnocení. Nemůžeme si tedy pomoci statistickými metodami, grafy apod.

Vyhodnocování především hledá v odpovědích společné znaky, které naznačují to, kde se nachází konsenzus odborníků. Předmětem zájmu jsou také zajímavé myšlenky, které se nemusí v odpovědích opakovat. Tyto údaje nám mohou pomoci pochopit lépe rozhodovací situaci nebo nás mohou navést na úplně nový směr výzkumu, tedy identifikovat mezeru ve zjišťovaných údajích.

Nevýhodou je také to, že náročnost na vyhodnocení odpovídá náročnosti zpracování takové odpovědi - tedy odpověď je časově náročná i pro respondenta samotného.

Vzhledem k náročnosti na zpracování i vyhodnocování by otázky vedoucí ke tvořeným odpovědím měly tvořit pouze malou část dotazníku, pokud by se vůbec měly použít.

I když se nám podaří vybrat experty tak, aby byla zajištěna jejich nezávislost, odpovědi na otázky mohou za určitých okolností být stále zavádějící. Tento problém může být způsoben osobním spojením expertů k problematice šetření - tento problém nazýváme **bias**.

Rozlišujeme dva druhy biasu:

- hypotetický,
- strategický.

Hypotetický bias nastává tehdy, když osoba odpovídající na otázku vnímá situaci, otázkou navozenou, jako čistě hypotetickou, která se v praxi nemůže stát. V takovém případě může respondent mít tendenci odbýt odpověď, podcenit závažnost apod.

Proti tomuto typu biasu je možno se efektivně bránit tak, že otázky jsou formulovány jasně, s příklady z reálného světa, které respondenta navodí přesně do situace. Pokud není možné takového cíle dosáhnout pomocí formulace otázek, používá se často průvodní dokumentace, která není součástí dotazníku, a která popisuje důvody přítomnosti otázky, co si od ní slibuje zadavatel apod. Účelem takové dokumentace je indoktrinovat respondenta.

V případě, že otázky jsou složitější může být pro zadavatele účelné zorganizovat indoktrinační seminář.

Pro respondenta je nastudování průvodní dokumentace a případná účast na semináři povinná, což může omezit dostupnost expertů pro účely dotazníku, zejména pokud funkce respondenta není

honorovaná. Z pozice organizátora jsou se zpracováním průvodní dokumentace a organizací seminářů spojeny další dodatečné náklady.

Strategický bias nastává v přesně opačném případě, než bias hypotetický, tedy v situaci, kdy respondent nejen, že věří, že situace může nastat, ale také že určitým typem odpovědí může získat určitou výhodu. Motivace odpovědět určitým způsobem může být přitom různá. Může to být např. naha zalíbit se - expert odpovídá tak, aby naplnil očekávání zadavatele. Určité odpovědi mohou také vést ve výsledku k určitým akcím.

Např. může probíhat šetření o vnímání rizik spojených s povodní. Šetření bude probíhat mezi obyvatelstvem v ohrožené oblasti. Cílem je zjistit, jak obyvatelstvo hodnotí ohrožení svého života a majetku ve zkoumané oblasti. Zadavatel potřebuje zjistit, zda má smysl realizovat dodatečná ochranná opatření proti povodním. Respondenti, pokud jim bude znám účel zkoumání, mohou záměrně zvelič̆ovat vnímanou hrozbu, aby donutili zadavatele jednat.

Příprava dotazníkového šetření je tedy obtížná. Kromě samotného znění otázek je nutné také řešit, zda bude potřeba vytvářet podpůrné, školící materiály, pořádat pro respondenty školení apod. Důležité rozhodnutí je také o formě, kterou bude šetření probíhat. Budou formuláře rozesílány v papírové podobě, nebo elektronickou formou. Zvolit lze také osobní konzultace a to formou telefonickou nebo přímo tváří v tvář.

Metody založené na měření mezního užitku

V textu jsme hovořili o Delfské metodě a provádění dotazníkových šetření. Na podobném principu fungují i další metody, které lze použít pro podporu rozhodování. Metody CVM, TCM (a další) jsou založeny na měření mezního užitku a všechny používají pro shromažd̆ování údajů dotazníkové šetření. Toto šetření ale neprobíhá u vybraných „expertů“, ale probíhá u široké veřejnosti, obvykle v nějaké zájmové oblasti (např. oblasti ohrožené zkoumaným zdrojem rizika).

Výše uvedené metody jsou založeny na měření ochoty platit (Willigness to Pay (WTP)) nebo ochoty strpět (Willigness to Accept (WTA)). WPT zjišťuje kolik jsou obyvatelé ochotni zaplatit, aby se vyhnuli nebo minimalizovali možnost realizace rizika. Čím větší je přitom pocit ohrožení u obyvatel, tím více budou ochotni zaplatit, aby se mu vyhnuli. Metoda měření ochoty strpět (WTA) je založena na obdobném principu - zjišťuje se však výše prostředků, které jsou obyvatelé ochotni přijmout jako kompenzaci za existenci rizika (např. úlevu na nájmu, slevu na dani, příspěvek na ochranné opatření apod.).

Metoda podmíněného hodnocení (Contingent Value Method (CVM)) se nasazuje především pro měření tzv. *neobchodovatelných statků*, může nám proto pomoci v úvahách týkajících se rizik, různých otázek životního prostředí, apod. Metoda cestovních nákladů (Travel Cost Method (TCM)) je určena především pro určení „ceny“ rekreačních oblastí, přírodních rezervací apod. Cenu odhaduje na základě měření ochoty případných zájemců zaplatit si cestu do zkoumané lokality. Opět čím více je zájemce ochoten zaplatit, tím je lokalita cennější.

Těmto metodám se dále z prostorových důvodů nebudeme věnovat, existuje však celá řada studií, které problematiku nasazení metod dopodrobna zkoumají, viz např. [32, 42].

Otázky

1. Jak pracuje Delfská metoda?
2. Jaký druh odpovědí dotazníkových šetření zpracováváme a jak (základní principy zpracování)?
3. Co je to *datový incest* a jak je možné se mu vyhnout?
4. Jaký je rozdíl mezi hypotetickým a strategickým biasem?

5.2 Brainstorming a myšlenkové mapy

Brainstorming je jednou z nejpoužívanějších metod pro analyzování (nejen) rozhodovacích situací. Tato metoda funguje nejlépe pro menší týmy, které se snaží identifikovat všechny relevantní informace a směry vztahující se k analyzované situaci. Metoda jako taková je proto obzvláště užitečná v počátečních fázích rozhodování.

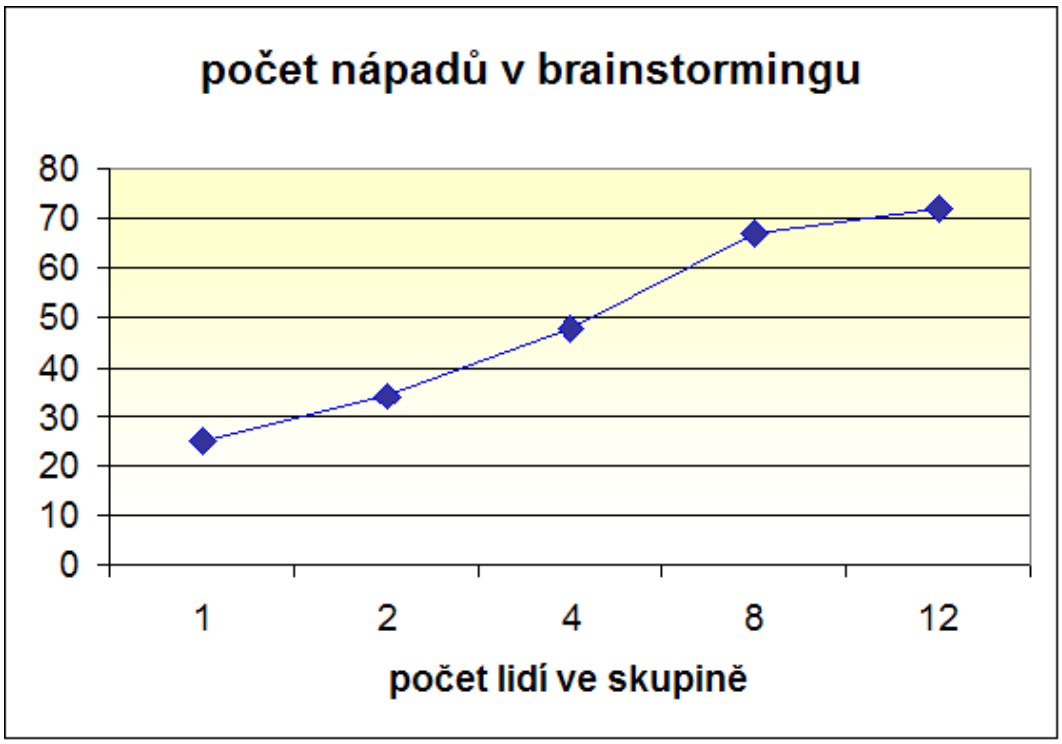
Metoda samotná není použitelná k identifikaci optimálních řešení problémů, ale spíše k jeho lepšímu popisu nebo identifikaci jeho důležitých aspektů. Řešení samotné tedy musí být dosaženo použitím jiných analytických metod.

Průběh brainstormingového sezení vždy řídí jedna předem určená osoba, která přítomné uvede do problému, stanoví centrální téma a dále zapisuje názory, nápady a myšlenky, které zazní. Postupně tak vzniká propojená síť pojmů se stanoveným tématem uprostřed. Složení týmu pro brainstorming by mělo být dostatečně bohaté tak, aby tým odborně pokrýval, pokud možno, celou oblast řešeného problému.

Během sezení se jednotlivé nápady zapisují všechny, bez hodnocení. Hodnocení návrhů, pokud by probíhalo, by mělo totiž negativní vliv na počet návrhů. Navíc i nápady, které na první pohled nejsou příliš užitečné, nebo mohou působit více exoticky mohou přinášet neotřelý, inspirativní pohled na problematiku, takže ve skutečnosti mohou být velmi prospěšné pro inspirování dalších nápadů.

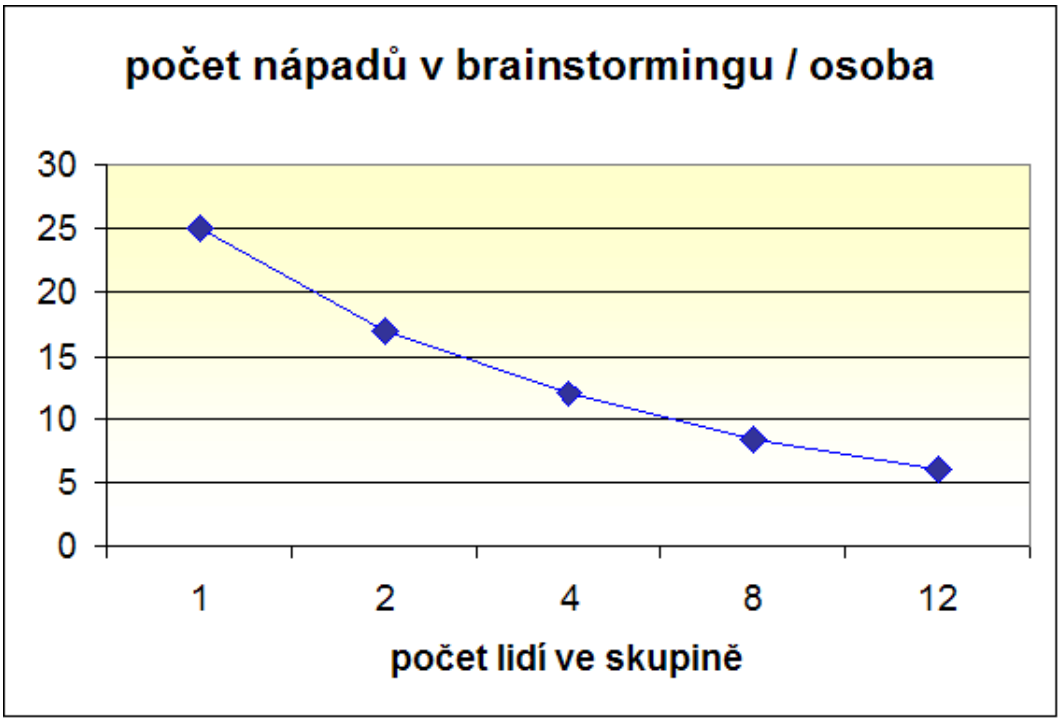
Všichni v místnosti mají stejná práva, tedy nikdo není preferován. Omezujícím faktorem je pak leader, který zapisuje nápady na tabuli/papír/do počítače (dle zvoleného záznamového média) - rychlost zápisu je tedy omezena. Aby se zabránilo chaosu musí být diskuze moderovaná, i to je úkolem leadera. Z praktického hlediska existuje souvislost mezi počtem návrhů a počtem členů týmu participujících na brainstormingu.

Obecně lze říci, že čím více lidí se podílí na brainstormingu, tím více nápadů se vygeneruje. Nárůst počtu nápadů však není rovnoměrný, od určitého počtu lidí poměrně výrazně klesá - s narůstajícím počtem účastníků, tedy jednotliví účastníci dostávají stále menší a menší prostor pro seberealizaci, což může být pro výsledek nepříznivé. Graficky lze znázornit závislost počtu lidí a počtu generovaných nápadů na obr. 5.1 a 5.2.



Obrázek 5.1: Závislost počtu nápadů na velikosti skupiny podílející se na brainstormingu (převzato z Osuský, Fajmontová [2])

Rozhodnutí o velikosti skupiny podílející se na brainstormingu může tedy výrazně ovlivnit výsledky. Jelikož neexistuje nějaká pevná hranice ... připravte se na to, že možná při použití budete muset



Obrázek 5.2: Závislost počtu nápadů na účastníka a velikosti skupiny podílející se na brainstormingu (převzato z Osuský, Fajmontová [2])

experimentovat. Nejste spokojeni s výsledkem s použitím větší skupiny? Použijte menší skupiny a následně můžete použít výstupy session jako vstup pro větší skupinu, kterou tím lépe usměrníte. Zajímavé může být také rozdělení skupiny podle oborů a jejich řešení samostatně. Výstupy je pak nutné integrovat do jediného multioborového pohledu.

Vzhledem ke způsobu shromažďování údajů někdy o výsledku brainstormingu hovoříme jako o tzv. *myšlenkové mapě*. Autor pojmu myšlenková mapa Tony Buzan [27] však myšlenkové mapy chápe spíše jako univerzální prostředek pro každodenní použití - pro organizaci vlastních nebo cizích myšlenek, pro tvoření poznámek z přednášek, organizaci informací k nastudování apod.

Myšlenkové mapy je možno dělat ručně nebo pomocí výpočetní techniky. Ručně vytvářené mapy mohou mít dokonce do určité míry i uměleckou hodnotu, viz obr. 5.3.

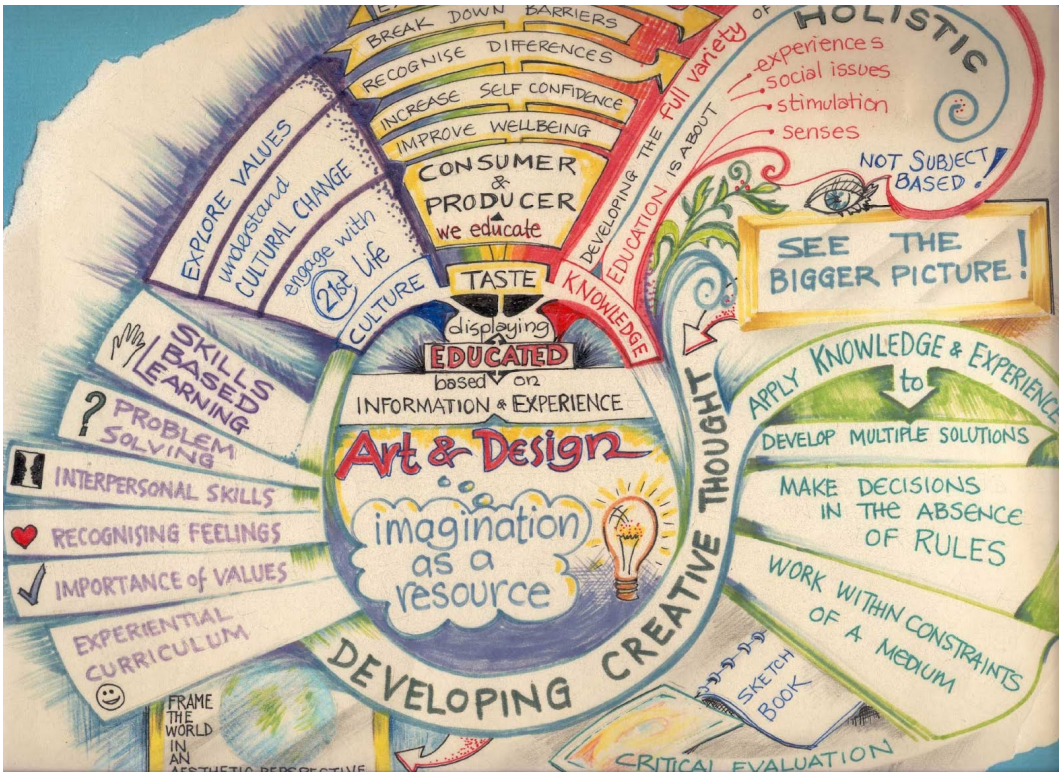
Je ale očividné, že nakreslení takové mapy „ručně“ je časově náročné a vyžaduje určité schopnosti kreslit. Většina z nás méně nadaných v oblasti kreslení proto raději sáhne po programech jako je FreeMind [23] (open source) nebo MindManager [39] (proprietální software). Výstupem těchto programů jsou myšlenkové mapy, které vypadají unifikovaně, ale jejich vytváření je pohodlné a především rychlé.

Implementace v software také přináší některé další velmi podstatné výhody jako je možnost kooperace více uživatelů nad jedinou mapou, možnost mapování dalších zdrojů do mapy, jako jsou dokumenty, obrázky nebo dokonce další mapy. K jednotlivým uzlům mapy lze také přidat dodatečnou dokumentaci.

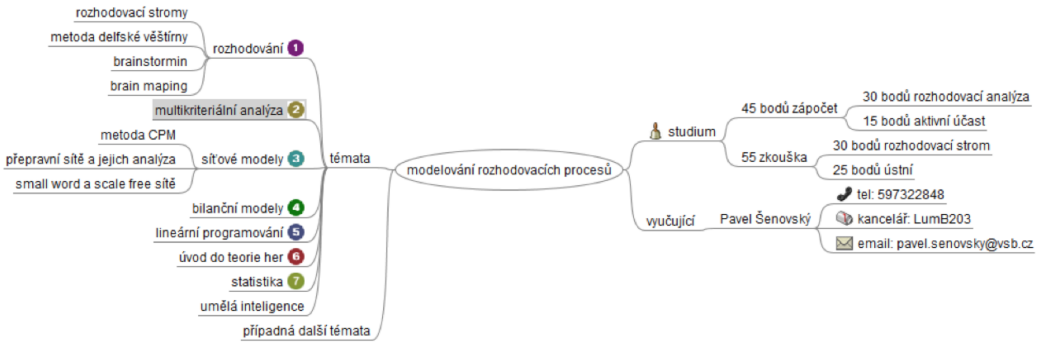
Příklad myšlenkové mapy vytvořené v programu FreeMind na téma *předmět Modelování rozhodovacích procesů* naleznete na obr. 5.4.

Obr. 5.4 byl vytvořen v programu Freemind [23]. Tento nástroj byl vytvořen v programovacím jazyku JAVA a proto je možné jej provozovat na prakticky jakémkoliv operačním systému. Jeho další výhodou je to, že se jedná o open source a proto si jej můžete vyzkoušet sami bezplatně.

Komerční alternativou je program Mind Manager [39]. Tento program máte také možnost vyzkoušet a to konkrétně na počítačové učebně LumA54, kde je instalován.



Obrázek 5.3: Myšlenková mapa Ard and Design (převzato z [14])



Obrázek 5.4: Myšlenková mapa předmět Modelování rozhodovacích procesů

Samostatný úkol

Ve zvoleném nástroji pro tvorbu myšlenkových map analyzujte zvolený problém nebo situaci. Tématem může být například *povodeň* nebo *únik neznámé látky.*

Otázky

1. K čemu je dobrá myšlenková mapa?
2. Kdy v průběhu rozhodování je nejlepší použít metody brainstormingu popř. myšlenkové mapy?
3. Specifikujte roli týmového leadera během brainstormingu?
4. Jaké výhody má „ruční“ zpracování myšlenkové mapy proti zpracování pomocí výpočetní techniky?

Shrnutí

Delfská metoda je jednou z metod, kterou lze použít v případě, že analytik nemá k dispozici tvrdá data, o která by své rozhodnutí mohl opřít. Delfská metoda je založena na expertním hodnocení. Analytik tedy specifikuje otázky, charakterizující řešený problém, sestaví skupinu expertů, kteří mu na otázky odpovědí. Tímto způsobem analytik zmapuje názory na danou problematiku a pokusí se v ní identifikovat názorový konsenzus přínosný z hlediska řešení problému.

Metoda *brainstormingu* umožňuje zejména v počátku rozhodování lépe specifikovat problém, pochopit různé aspekty rozhodovací situace. Jedná se o týmovou metodu, v rámci které zaznamenává leader názory a nápady týkající se nastoleného tématu/problému.

Myšlenkové mapy slouží pro organizaci myšlenek, poznámek apod. Každá mapa má (uprostřed) určité centrální téma které se postupně rozvíjí do stran do podoby několika stromů rozebírajících různé aspekty tématu. Myšlenkové mapy lze použít i pro formální zaznamenání výsledků brainstormingových sezení.

Z důvodu rychlosti zápisu myšlenkové mapy a jejího jednotném vzhledu se pro tvorbu používají často softwarové nástroje.

Kapitola 6

Průzkumy a dotazníková šetření

Náhled kapitoly
V této kapitole se seznámíme se způsobem provádění dotazníkových šetření a jejich vyhodnocování pomocí statistických metod. Průzkumy a dotazníková šetření jsou totiž jednou ze základních metod sloužících ke shromažďování informací prakticky o čemkoliv, napříč obory. Se způsobem tvorby a vyhodnocení byste se proto měli seznámit také Vy.

Po prostudování této kapitoly budete vědět
- co jsou dotazníková šetření a jak se tvoří
- s jakými problémy se setkáváme při jejich vyhodnocování a
- jaké metody použít pro vyhodnocení takových šetření

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně 3 hodiny, pokud ale tápete v základních pojmech statistiky, připravte se na to, že Vaše studium této kapitoly se může protáhnout a možná bude potřeba také nahlédnout do skript předmětu *Statistika*.

S dotazníkovými šetřeními se setkáváme prakticky každodenně - objevují se ve zpravodajství, pracuje s nimi statistický úřad a v neposlední řadě se velmi často používají jako základní vstupní údaj pro bakalářské a diplomové práce. Pokud se taková šetření provádějí tak často, musí být jejich realizace snadná, že? Ačkoliv vytvoření dotazníku a jeho korektní vyhodnocení není žádná věda, je možné v něm provést řadu chyb, které snadno mohou znehodnotit a zpochybnit výsledky analytických činností. Konečně, pokud by to bylo tak snadné neexistovaly by firmy, které se specializují čistě na provádění průzkumů a vyhodnocování výsledků z nich.

6.1 Úvod to tvorby dotazníků

Základní překážkou zpracování dotazníků je zejména to, že řada problémů s vyhodnocováním není na první pohled patrná - intuitivně odvoditelná, zejména pokud nemáte praktické zkušenosti s prováděním šetření. Zkusme nejprve definovat některé základní pojmy.

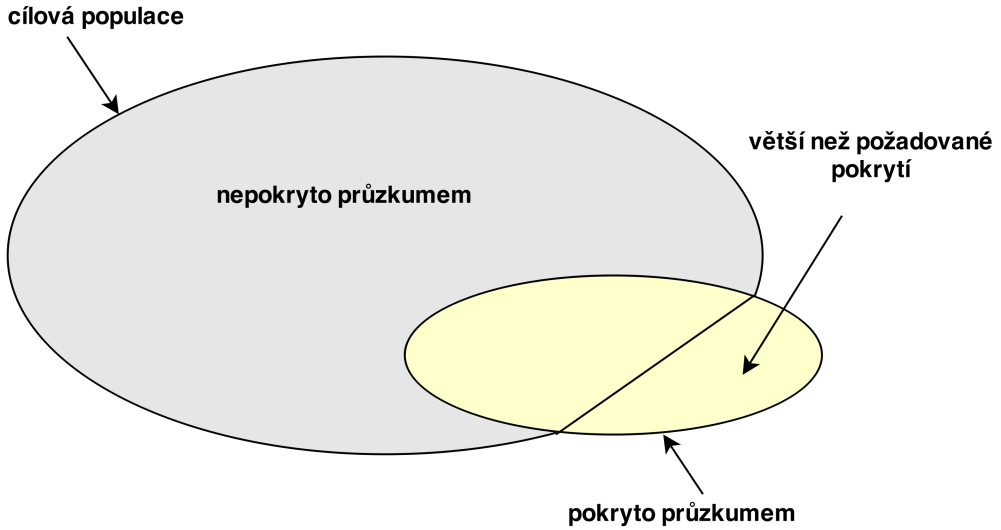
Populace - pojmem populace je v kontextu šetření poněkud odlišný, než je obvyklé. Obvykle rozumíme populací obyvatelstvo, které se nachází v určeném čase na specifikovaném místě, např. státu (populace státu), kontinentu apod. V kontextu dotazníkových šetření a průzkumů obecně však populací rozumíme tu část obyvatelstva, jejíž vlastnosti hodláme zkoumat, tedy nejedná se o úplnou populaci.

Šetření samotné je také obvykle tzv. *výběrové* - to znamená, že údaje nejsou shromažďovány u zájmové populace jako celku, ale pouze u jejího vzorku, který označujeme jako *representativní*. Není to tak, že by úplný vzorek populace nebylo možné změřit (pomocí dotazníku), ale dělá se to pouze v

omezených případech, např. pokud zájmová populace je malá, aby provedení výzkumu bylo finančně zvládnutelné (např. průzkum mezi krajskými řediteli HZS). Úplné šetření populace státu provádí např. Český statistický úřad (ČSÚ) v rámci *Sčítání lidu, domů a bytů* [48]. Jedná se ale o proces natolik náročný, že je možné jej realizovat pouze jedenkrát za 10 let.

Pro účely těchto skript budeme reprezentativním vzorkem rozumět takový vzorek zájmové populace, jehož sledované charakteristiky vykazují statisticky významnou shodu s populací jako celkem. Tedy na základě analýzy vybraného vzorku populace můžeme analogicky odhadovat vlastnosti zájmové populace.

Graficky si lze situaci představit podobně jako na obr. 6.1.



Obrázek 6.1: Vzorek obyvatelstva pokrytý průzkumem (adaptováno z [25])

Z obr. 6.1 vyplývají také některé problémy. Šetření musí být navrženo tak, aby měření probíhalo pouze v cílové populaci. Např. provádíme průzkum o povědomí o bezpečnosti obyvatelstva vybrané čtvrti města. Pokud je průzkum realizován náhodným výběrem respondentů „na ulici“ v zájmové oblasti, může se stát, že oslovíme člověka, který se v tomto území nachází náhodně - prochází, nebo jde o jednorázové nákupy apod. Takový člověk nepatří do cílové populace. Zavedením odpovědí takového člověka do vyhodnocení by se snížila vypovídající schopnost průzkumu jako celku.

Setkat se lze také s opačným případem, kdy naopak vybraný vzorek populace nepokrývá dostatečně všechny zájmové charakteristiky cílové populace. Tomuto problému čelíme použitím dostatečně velkého vzorku respondentů.

Šetření samotné provádíme obvykle v několika krocích:

1. návrh šetření
2. shromažďování dat (provedení průzkumu)
3. úprava dat
4. korekce dat (vypořádání se s chybějícími odpověďmi)
5. provedení analýzy
6. publikace výsledků šetření

Návrh šetření

Šetření provádíme účelově, prakticky to znamená, že šetření vždy sleduje určitý cíl. U každého šetření je tedy nutné si položit otázku *co je potřeba šetřením zjistit*. Tento cíl slouží jako základ pro návrh *proměnných*, jejichž hodnoty má šetření zjistit. Za tímto účelem je potřeba vhodně zformulovat otázky dotazníku, kterými tyto proměnné změříme.

Některá šetření preferují před započetím samotného výzkumu také specifikaci vstupních hypotéz, které jsou pak šetřením potvrzeny nebo naopak vyvráceny. Hypotézy před šetřením není nutné formulovat vždy - někdy má třeba šetření sloužit jako průzkum názorů, znalostí, vybavení apod. O to větší nároky jsou ale v takovém případě kladeny na analytickou část šetření, aby shromážděná data byla vhodně „vytěžena“.

Proměnné rozlišujeme dvojího druhu:

- cílové a
- doplňkové

Cílové proměnné jsou přímo spojeny s cílem šetření - jejich získání je tedy podmínkou nutnou pro úspěšně vyhodnocení šetření v souladu se stanovenými cíli šetření. *Doplňkové proměnné* nemají přímou vazbu na cíl šetření, mohou ale k úspěchu šetření výrazně přispět tím, že umožní stratifikaci šetřením získaného datového souboru, nebo nám umožní identifikovat respondenty nespadající do cílové populace.

Typickými představiteli doplňkových proměnných jsou věk, pohlaví, rodinný stav, identifikace místa provedení průzkumu atd.

Otázky je potřeba formulovat tak, aby byly krátké a jednoznačně vyložitelné i pro respondenta - neodborníka. Na tento požadavek, je potřeba dát pozor zejména v okamžiku, kdy respondent vyplňuje dotazník sám - neasistovaně. Typickým příkladem je použití elektronických formulářů dostupných přes World Wide Web (WWW) rozhraní. Respondent se v takovém případě nemá prostě koho zeptat - to ale také znamená, že není záruka, že respondent bude chápat otázku stejně jako tvůrce dotazníku. Tato nejistota se musí nutně přenést do odpovědi.

Bias
Při formulaci otázek je potřeba brát v úvahu také možnost, že respondent nebude vůči otázce (a odpovědi na ni) nestranný. S takovou situací jsme se už setkali při výkladu *Delfské metody*, takže pokud Vám výraz *bias* nic neříká, je vhodný čas vrátit se zpět do předchozí kapitoly a tyto informace si dostudovat.

Pro správné vyplnění dotazníku mohou být rozhodující i takové drobnosti jako je např. pořadí otázek. Dotazník jako celek by totiž měl působit konzistentně. To znamená, že otázky by podle tématu měly být seskupovány a společně tvořit určitý „příběh“. Je potřeba počítat s tím, že předchozí otázku do určité míry ovlivní respondenta, třeba tak, že navodí určitou situaci. To může být žádoucí - může to však být také zdroj chyb, pokud se s touto situací nepočítá.

Při tvorbě dotazníku je potřeba počítat také s tím, že vyplňování tohoto dotazníku bude vnímáno jako práce navíc - s délkou dotazníku pak obvykle klesá ochota respondentů jej vyplnit. Respondenti jsou také tradičně méně ochotni vyplňovat otázky, které jsou nepříjemné, osobní nebo citlivé - např. věk, příjem, zdravotní stav apod. Proto se často tento typ otázek dává až na konec dotazníku.

Shromažďování dat

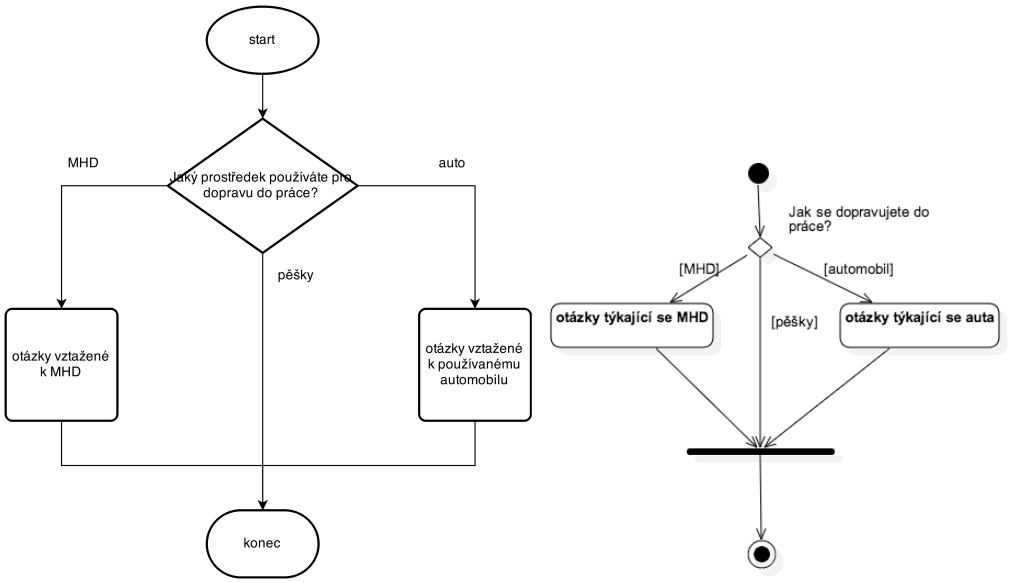
V této fázi šetření se vezme hotový dotazník a začne se s přípravou na jeho nasazení v praxi. Příprava může být krátká a sestávat se např. pouze z tisku formulářů nebo zveřejnění elektronické verze formuláře na WWW stránkách. V případě, že ale bude vyplňování dotazníku asistované - tedy bude je provádět nějaká pověřená osoba nebo osoby, je nutné tyto zaškolit, tak aby byla zajištěna konstantní kvalita vracejících se dotazníků.

Zaškolení se může kromě významu jednotlivých otázek a možných odpovědí týkat také samotného způsobu vyplnění otázek. Ve smyslu pořadí v jakém se mají vyplňovat. U složitějších dotazníků není výjimkou, že některé odpovědi mohou vyloučit celou řadu dalších otázek - např. pokud se v odpovědi dozvíme, že respondent dojíždí do práce pomocí vlastního auta, nemá smysl se jej ptát na kvalitu městské hromadné dopravy, jelikož ji prostě nepoužívá.

Nastavením možných cest průchodů dotazníkem má také pozitivní vlastnost v tom, že efektivně zmenšuje počet otázek, které respondent musí vyplnit. Při nevhodném použití lze ale docílit stavu, kdy respondent bude odpovídat na jiné otázky než by měl - a výsledky pak nebudou validní.

Pro účely kontroly průchodu lze postup dotazníkem vizualizovat. K vizualizaci se přistupuje zejména v situaci, kdy je průchod dotazníkem složitý a existuje tak reálná šance, že bude docházet při jeho vyplňování k chybám. K vizualizaci lze použít vývojový diagram [9] nebo diagram činností jazyka UML [41]. Zjednodušené příklady průchodu dotazníkem vizualizované pomocí vývojového diagramu a pomocí aktivity diagramu jazyka UML jsou znázorněny na obr. 6.2.

Jak vyplývá z obr. 6.2 je možno activity diagram jazyka UML chápat jako logické rozšíření vývojových diagramů, UML je ale také mnohem modernějším standardem, podporujícím řadu dalších typů diagamů pokrývajících odlišné aspekty systémové analýzy.



(a) Vývojový diagram (b) Activity diagram

Obrázek 6.2: Vývojový diagram vs activity diagram jazyka UML)

UML
S analytickým jazykem Universal Modeling Language (UML) se můžete seznamit v předmětu *Expertní systémy*, který si můžete zvolit v rámci svého studia ve čtvrtém ročníku.

Editace dat
Procesem editace dat se mohou myslet různé věci. Můžeme si pod ním představit proces, kdy udaje z dotazníků vyplněných v papírové podobě, převádíme do elektronické podoby pro pozdější zpracování. Editací ale také rozumíme opravu detekovaných chyb.

Rozlišujeme přitom tři základní druhy chyb:

1. chyby rozsahu
2. chyby v konzistenci
3. chyby v průchodu dotazníkem

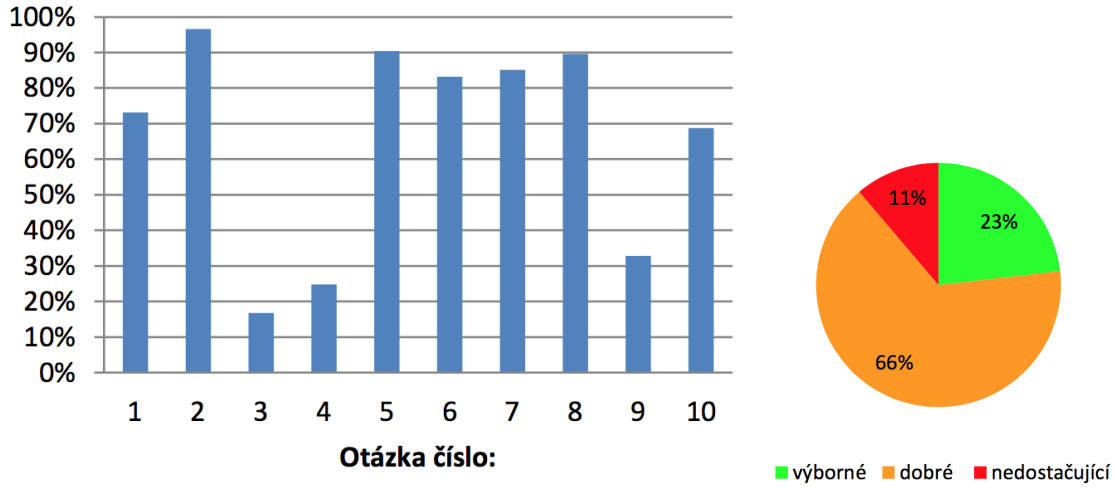
Validitu některých veličin lze kontrolovat vůči stanoveným mezím, pokud měřená chyba se těmto mezím vymyká hovoříme o *chybě v rozsahu*. Např. věk nad 100 let je u respondenta nepravděpodobný ale možný, věk nad 130 nebo záporný věk, ale již možný není. Pokud si respondent mohl vybrat s omezeného množství odpovědí, kontrolujeme, zda tak skutečně učinil a nevymyslel např. odpověď další.

Chybami v konzistenci rozumíme situaci, kdy je dotazník sice vyplněn, ale jednotlivé odpovědi jsou v vzájemném rozporu. Např. věk - 8, rodinný stav - ženatý, v podmínkách ČR nepravděpodobná kombinace. Tyto rozpory je nutné detekovat a vypořádat.

Chyby v průchodu dotazníkem vznikají chybným odhadem toho, na které otázky má respondent vlastně odpovídat. Výsledkem je situace, že ne všechny otázky na které se mělo odpovědět jsou skutečně zodpovězeny a naopak zodpovězena může být řada otázek, které měly zůstat bez odpovědi.

Detekované chyby je nutno vypořádat. Pro vypořádání můžeme přijmou různé strategie - můžeme se např. pokusit tyto údaje doplnit opětovným dotázáním se respondenta, pokud na něj máme kontakt, alternativně lze odpověď označit jako nevyplněnou.

Upozornění - příklady vyhodnocování
Příklady uvedené v této kapitole jsou pouze ilustrační - slouží tedy pouze pro demonstraci postupů procesu vyhodnocení.



(a) Sloupcový graf (b) Koláčový graf

Obrázek 6.3: Sloupcový graf a koláčový graf (Převzato z [29])

6.2 Základy vyhodnocování dotazníků

V okamžiku, kdy dostaneme za úkol vyhodnocení dotazníku si většina studentů představí něco podobného jako je znázorněno na obr. 6.3 [29].

Na použití grafů jako takovém není nic špatného - ba právě naopak. Grafy z obr. 6.3 jsou převzaty z diplomové práce Ing. Dratvy [29]. Průzkum byl realizován za účelem zjištění úrovně znalostí v oblasti ochrany obyvatelstva a krizového řízení mezi studenty středních škol a mezi běžnými občany.

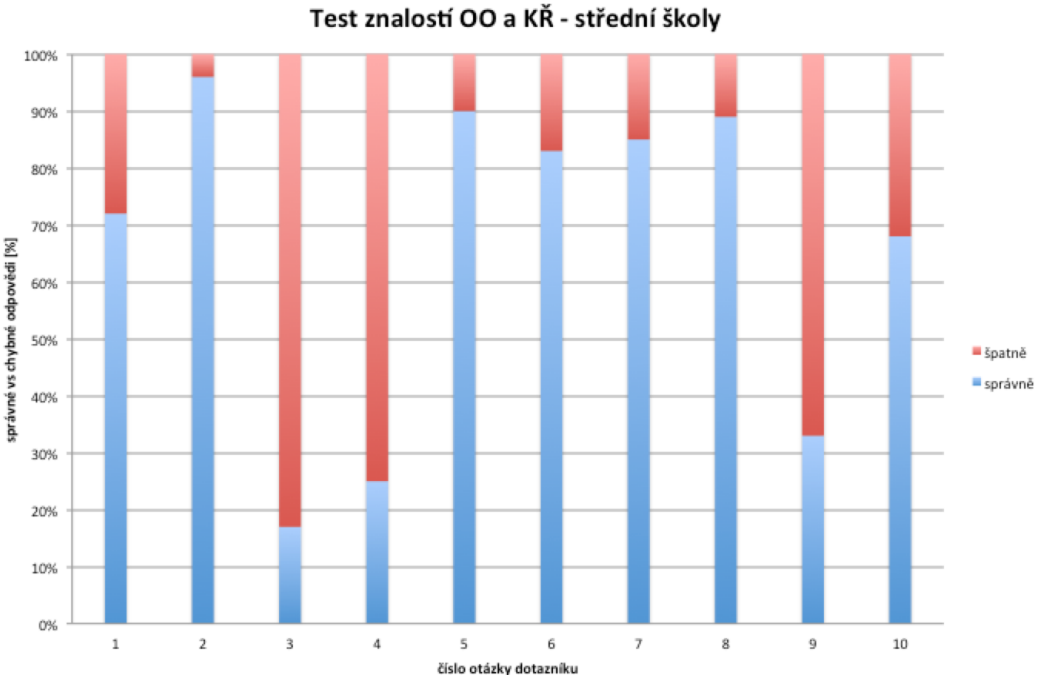
Sloupcový graf v tomto případě představuje procento správných odpovědí v jednotlivých otázkách. V čem je tedy problém? Graf jako takový by měl být, pokud to alespoň trochu půjde, samonosný. Samonosností se v tomto případě rozumí, že pokud se podíváme na graf samostatně, tedy bez studia okolního textu (kontextu grafu), mělo by být na první pohled patrné, co graf znázorňuje. K tomuto účelu máme k dispozici pouze omezené množství nástrojů:

- popisek grafu (např. *Obr. XY: Výsledky hodnocení znalostí o oblasti ochrany obyvatelstva a krizového řízení mezi studenty středních škol 2014*)
- nadpis nad grafem (např. *Test znalostí OO a KŘ - střední školy*
- popisky os grafu (např. osa X - *číslo otázky*, osa Y - *procento správných odpovědí [%]*)
- legenda (u koláčového grafu nebo grafů obdobných)

V kontextu toho, co víme o provedeném průzkumu bychom sloupcový graf 6.3 mohli nahradit odlišným typem sloupcového grafu, viz obr. 6.4. Tento typ grafu je možné použít i v případě, kdy je potřeba rozčlenit výsledek do většího množství cílových tříd než dvě, jako v našem demonstračním případě.

Grafická forma by však neměla být jedinou dostupnou formou výsledků - údaje by měly být uvedeny také v tabelární podobě. Výsledky průzkumů a jejich vyhodnocení, jsou totiž obvykle to nejzajímavější, co se v dané práci objeví. Vyhodnocení samotné je však obvykle úzce spojeno s účelem průzkumu. Čtenář, ale může chtít výsledky aplikovat odlišným způsobem a odvodit tak další informace. Pokud data nejsou k dispozici v tabelární podobě, ale pouze podobě grafické, je sice možno data odečíst, ale tímto způsobem se zbytečně do takto získaných dat může vnést chyba.

Např. u otázky 1 na obr. 6.4 správně odpovědělo 72 % nebo 73 % respondentů?



Obrázek 6.4: Výsledky hodnocení znalostí o oblasti ochrany obyvatelstva a krizového řízení mezi studenty středních škol 2014 (data: Dratva [29])

Pokud spočteme a vizualizujeme výsledky šetření samostatně pro střední školy a dospělé obyvatelstvo (obdobně jako na obr. 6.3 nebo 6.4) je možné usoudit na úroveň znalostí v oblasti ochrana obyvatelstva (OO) a krizové řízení (KŘ) v zájmové populaci, která je popsána použitými daty (pro kterou byl vzorek populace vybrán). Vyhodnocení je tedy možné samostatně pro středoškolské studenty a samostatně dospělé obyvatelstvo. Co ale dělat v případě, že potřebujeme zkoumat vlastnosti obou skupin dohromady?

První, co nás určitě napadne je oba vzorky sloučit a vyhodnotit je tak, jako by pocházely z jednoho datového souboru - tedy prostě je sečíst a přepočítat procentní podíl. Ilustrační data z příkladu jsou v tab. 6.1.

Tabulka 6.1: Vyhodnocení studenti vs dospělí (data: Dratva [29])

hodnocení	**studenti [-]**	**studenti [%]**	**dospělí [-]**	**dospělí [%]**	**celkem [-]**	**celkem [%]**
výborné	29	23	8	31	37	24,6
dobré	82	66	13	52	95	63
dostatečné	14	11	4	17	18	12

Problém je, že položky celkem, tak jak jsou znázorněny v tab. 6.1, nedávají ze statistického hlediska smysl. Pro správné odvození celkových vlastností obou skupin dohromady je nutné brát v úvahu:

1. velikost měřeného vzorku populace (v našem případě 125 studentů a 25 dospělých)
2. velikost skutečné populace, jejíž celkové vlastnosti odvozujeme (žáci ve vzdělávání s maturitní zkouškou - střední školy 2013/2014: *313 130* [40], počet dospělých v ČR ke konci roku 2014 je dle ČSÚ *8 665 578* [47]

Podle těchto údajů jsme tedy průzkumem pokryli 0,04 % studentů středních škol a 0,0003 % dospělé populace. Populace studentů středních škol tvoří přibližně 3,5 %, ale dospělí tvoří přibližně 96,5 %. Za předpokladu, že respondenti obou skupin představují reprezentativní vzorek své části zájmové populace, musíme tyto skutečnosti zohlednit, viz rovnice (6.1).

$$V = \frac{\sum_{i=1}^{N} w_i v_i}{\sum_{i=1}^{N} w_i} \tag{6.1}$$

kde V - je hodnota sledované proměnné pro celou populaci, N - celkový počet hodnocených vzorků, w - váha vzorku v cílové populaci.

Demonstrovat výpočet můžeme třeba pro zjištění předpokládaného podílu lidí s výbornými znalostmi v oblasti OO a KŘ:

$$V = \frac{3,5 \cdot 23 + 96,5 \cdot 31}{100} = 30,6 \tag{6.2}$$

Výsledek se tedy spíše blíží naměřenému výsledku pro skupinu dospělých.

Ve výkladu výše jsme provedli několikrát předpokládali, že statistický vzorek je reprezentativní, tuto skutečnost neprokázali. Vlastně jsme se vůbec nezabývali otázkou toho, jak velký má být statistický vzorek, aby jej bylo možné považovat za reprezentativní? Související otázka je, pokud mám určitý vzorek, jako úroveň přesnosti mohu vlastně očekávat.

Je potřeba si uvědomit, že cokoliv bude v rámci šetření „naměřeno“, se bude lišit od skutečné hodnoty, ať už je to průměr, nebo cokoliv vypočteného. Nepřesnost je dána tím, že pracujeme pouze se vybraným vzorkem populace a nikoliv s úplnou populací. Jako vodítko lze použít intervalový odhad chyby střední hodnoty, viz rovnice (6.3).

$$E = z_{\frac{\alpha}{2}} \cdot \frac{\sigma}{\sqrt{n}} \tag{6.3}$$

kde E - chyba, z - kritická hodnota normálního rozdělení pro úroveň důvěryhodnosti α, σ - směrodatná odchylka, n - počet naměřených hodnot.

Z této rovnice lze jednoduše odvodit velikost vzorku dat n, viz rovnice (6.4).

$$n = \left(\frac{z_{\frac{\alpha}{2}}\sigma}{E}\right)^2 \tag{6.4}$$

Výpočet se (6.4) se hodí pro numerické hodnoty v případě, že je známa směrodatná odchylka nebo je možné ji odhadnout.

Pokud pracujeme ale data jsou výčtového typu (taková data jsme zpracovávali v našem demonstračním příkladu), je nutné postupovat trošku odlišným způsobem, viz rovnice (6.5).

$$E = z_{\frac{\alpha}{2}} \sqrt{\frac{p(1-p)}{n}} \tag{6.5}$$

kde p je odhad správné hodnoty hledané veličiny, ostatní symboly jsou významově shodné s rovnicí (6.3). Odtud lze jednoduše odvodit potřebnou velikost vzorku n (viz rovnice (6.6)).

$$n = \frac{z^2_{\frac{\alpha}{2}} p(1-p)}{E^2} \tag{6.6}$$

Zkusme aplikovat tento postup na odhadu počtu dospělých s výbornými znalostmi v oblasti OO a KŘ, tedy v intencích příkladu, se kterým pracujeme v této kapitole. Naším požadavkem je, aby výsledek šetření byl v intervalu spolehlivosti 95 %, s očekávanou chybou maximálně 4 %. Pro odhad veličiny p použijeme hodnotu 31 % (tak, jak vyšla v původním šetření).

$$n = \frac{1,96^2 \cdot 0,31(1-0,31)}{0,04^2} \doteq 514 \tag{6.7}$$

Tedy vyšlo nám, že pro hodnocení znaku, který v populaci je zastoupen 31 %, v intervalu spolehlivosti 95 % a očekávané chybě 4 %, je nutné oslovit 514 respondentů (za předpokladu, že všichni oslovení respondenti odpoví, a data od nich získaná budou bez neodstranitelných chyb).

Experimentováním s jednotlivými členy výše uvedených rovnic pak lze odhadnout např. přesnost u vzorku určité velikosti.

Statistika
Pokud si nejste jistí, jak jsme k výše uvedeným závěrům došli, je nyní ten správný čas vrátit se ke svým poznámkám, studijním materiálům předmětu *Statistika*.

Shrnutí
Provádění dotazníkových šetření a jejich vyhodnocování není procesem úplně snadným. Při přípravě šetření je potřeba především věnovat pozornost stanovení cíle šetření a návrhu otázek, jejichž odpovědi nás k cíli dovedou. Otázky by přitom měly být krátké, jasně vyložitelné.

Při provádění šetření samotného je potřeba si uvědomit, že dotazování probíhá pouze u vybraného vzorku cílové populace - proto musí tento vzorek být dostatečně velký, aby bylo možné výsledky považovat za statisticky významné. Při volbě velikosti vzorku populace je nutno zohlednit požadovaný interval spolehlivosti, očekávanou chybu „měření“ a také zastoupení hledaného statistického znaku v populaci.

Vizualizaci výsledku je možné provést pomocí vhodného grafu, např. pomocí sloupcového nebo koláčového grafu. Výsledky šetření by měly však být dostupné taktéž v tabelární formě ať už přímo v textu nebo jeho příloze. Pokud je statistický soubor rozdělen pomocí určitého znaku, např. vzdělání, pohlaví, apod. je při interpretaci výsledku nutno zohlednit skutečné zastoupení takového znaku v cílové populaci.

Numerické veličiny je potřeba potřeba dále statisticky zpracovávat ve smyslu odhadu směrodatné odchylky a tuto dále zohledňovat při interpretaci získaných údajů.

Otázky

1. Co rozumíme pojmem cílová populace?
2. Definujte co je to reprezentativní vzorek?
3. Vysvětlete postup adaptace naměřeného vzorku populace na populaci cílovou.
4. Vysvětlete jak ovlivní volba intervalu spolehlivosti a očekávané chyby velikost výběrového vzorku.
5. V čem spočívá chyba průchodu dotazníkem a jak ji lze řešit?

Kapitola 7

Síťové modely

Náhled kapitoly
V této kapitole se seznámíme s aplikací síťových modelů pro zachycení návaznostní jednotlivých činností v rámci realizace projektu jako základního nástroje managera pro identifikaci problémových míst v rámci projektu a jejich sanaci.

Po prostudování této kapitoly budete vědět

- co je to síťový graf
- harmonogram
- metoda CPM

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně 3 hodiny.

7.1 Metoda CPM v projektovém řízení

Síťové modely jsou jedny z nejpoužívanějších modelů pro analýzy tzv. *kritické cesty* (CPM). Síťové modely se využívají pro vizualizaci časového průběhu projektů, pro analýzy přepravních soustav apod.

U přepravních soustav je konfigurace síťového modelu dána fyzickou konfigurací modelované sítě, u síťových modelů časového průběhu projektů je konfigurace modelu dána činnostmi, které se v rámci projektu dějí, a jejími návaznostmi.

Předtím než se pustíme do podrobnějšího výkladu konstrukce síťového modelu, zkusme si nadefinovat několik pojmů, které se v dalším textu objeví. Prvním z nich je *management*. Slovo management je odvozeno z francouzského manéž, tedy kruhová aréna určená pro drezúru koní. Někdy se původ tohoto slova odvozuje také od francouzského managere – tedy řízení. Pro management se objevuje celá řada definicí, např. že management se dosahování cílů prostřednictvím jiných. Objevuje se také definice: Management je mobilizace všech zdrojů podniku za účelem dosažení vytyčených cílů.

Management jako vědní disciplínu řadíme do oblasti humanitní, proto je jeho úplné zvládnutí pomocí tvrdých, technických prostředků nemožné – ty jsou však ale použitelné (a vysoce užitečné) jako podpůrné prostředky. Jejich použití podpůrných metod automaticky nezajišťuje manažerovi úspěch, ale zvyšuje jeho šanci.

Dalším pojmem, který budeme často diskutovat je *projekt*. Pojmem projekt rozumíme sled činností, které vedou k vymezenému cíli, s jasně daným začátkem a koncem a přesně danými zdroji použitelnými pro splnění cílů projektu.

Projekt jako takový lze vymezit činnostmi, které v jeho průběhu musí proběhnout, v určité časové návaznosti. Takové návaznosti můžeme velmi dobře vyjádřit tabulkově. Demonstrujme si tento přístup

na zjednodušeném příkladu stavby domu (viz tabulka 7.1).

Tabulka 7.1: Průběh činností projektu v čase

	činnost	**následníci**	**předchůdci**	**Délka [dny]**
A	Výkop základů	B	-	2
B	Betonování základů	C	A	10
C	Hrubá stavba	D	B	17
D	Zastřešení	E,F,G,H	C	5
E	Elektroinstalace	I	D	5
F	Potrubí (voda, plyn)	I	D	5
G	Topení	I	D	5
H	Okna	I	D	14
I	Omítky, podlahy	J	E,F,G,H	21
J	Kolaudace	-	I	1

Návaznosti můžeme popsat prostřednictvím specifikace následníků činností nebo předchůdců činnosti. *Následníci* jsou činnosti, které musí následovat po dokončení právě hodnocené činnosti. *Předchůdci* jsou činnosti, které musí být dokončeny předtím než mohu započít právě hodnocenou činnost.

Jedná se vlastně o ekvivalentní zápisy, liší se pouze směr, kterým provádíme hodnocení. U následníků postupujeme od počáteční činnosti směrem k cíli projektu, u předchůdců začínáme u poslední činnosti a propracováváme se k začátku projektu.

Některé software pro projektové řízení např. Microsoft Project [13], open source Project Libre [17] nebo Open Workbench [16] umožňují volit si způsob navazování dle potřeb a dokonce jej měnit, s tím že se následníci a předchůdci operativně přepočítají.

Softwarová podpora

Vytváření diagramů ručně je poměrně náročné - zabere to spoustu času a udělat chybu je snadné. Jelikož projektové řízení je aktivitou, která je realizována často, vznikla v průběhu doby řada pokročilých softwarových nástrojů schopných usnadnit nám činnosti plánování projektů a také kontroly jejich realizace.

Defacto leaderem v oblasti projektového řízení je společnost Microsoft se svým nástrojem MS Project [13]. Jedná se o proprietální software, který sám Microsoft řadí do „širší“ rodiny MS Office. Tento nástroj se vyznačuje celkovou uživatelskou přívětivostí a možností integrace s dalšími technologiemi Microsoftu jako MS Exchange pro možnost propagace pracovních rozvrhů jednotlivým pracovníkům projektu přímo do jejich kalendářů zobrazovaných pomocí MS Outlook (nebo webového rozhraní).

Existuje také celá řada dalších softwarových nástrojů, které jsou přímo open source. Příkladem by mohl být třena Project Libre [17] (dříve známý pod jménem OpenProj). Project Libre je dostupný na prakticky všech platformách od Windows až po OS X. Jako alternativu lze zmínit produk Open Workbench [16], který je však dostupný pouze pro MS Windows.

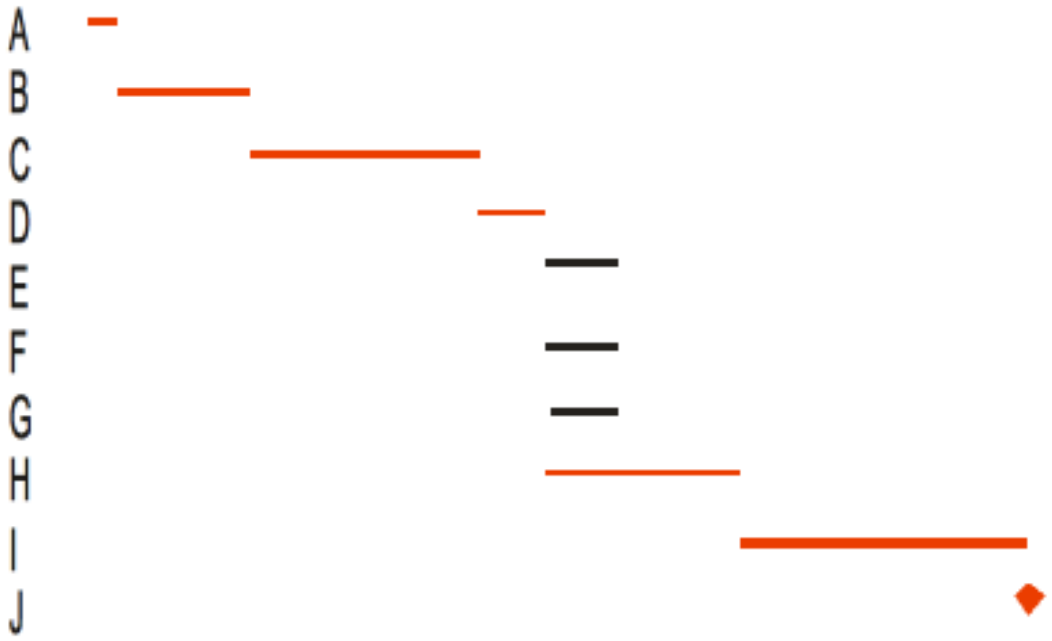
Praktické ukázky, které máte možnost vidět v následující kapitole byly vytvořeny pomocí Project Libre.

Seznam aktivit v tabulární formě není úplně dobře použitelný pro operativní řízení průběhu jednotlivých aktivit projektu, plánování zdrojů apod. Z tohoto důvodu často používáme grafickou interpretaci informací o časovém charakteru aktivit a jejich vzájemné vazbě pomocí *harmonogramu* někdy také nazývaného *Ganttův diagram* a také pomocí *síťového diagramu*.

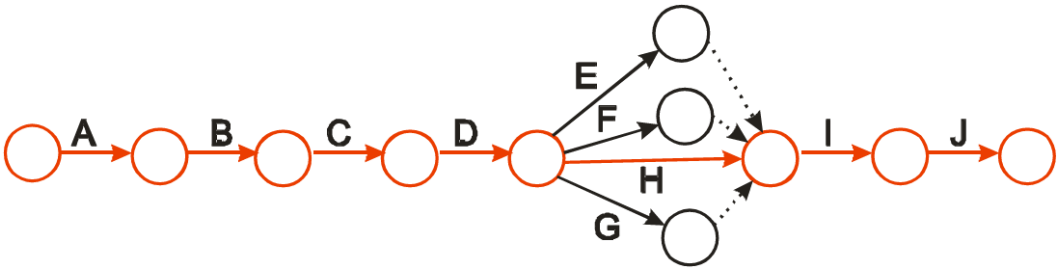
Jak vidno z obr. 7.1, červeně je zvýrazněna tzv. *kritická cesta*. Kritickou cestou rozumíme posloupnost cest od začátku do konce pro kterou platí, že překročení plánovaného času (aktivita trvá déle než bylo plánováno) automaticky vede k prodloužení délky projektu jako celku. Účelem identifikace

kritické cesty proto je identifikovat aktivity, na které si manager musí z hlediska řízení dát obzvláště velký pozor.

To samozřejmě neznamená, že by manager mohl ostatní aktivity úplně ignorovat, pouze to, že u nich existuje jistá časová rezerva. Tuto rezervu lze využít při řízení - např. z nekritické činnosti uvolnit lidské nebo materiálové zdroje a posílit činnost kritickou tak, aby se zajistilo její dokončení v čas. Odebrání zdrojů z nekritické činnosti povede taktéž k posunu začátku nebo jejímu prodloužení, pokud je tento posun nebo prodloužení dostatečně malý, aby délka činnosti nepřesáhla limity stanovené existující časovou rezervou - nezasáhne tato změna negativně do celkového obrazu projektu (nedojde k jeho prodloužení).



Obrázek 7.1: Zjednodušený harmonogram projektu



Obrázek 7.2: Síťový graf projektu

V *síťovém grafu* jsou aktivity představovány hranami. Hrany E, F, G, H představují činnosti (aktivity), které mohou probíhat zároveň. Všechny tyto aktivity musí být dokončeny předtím, než bude zahájen činnost I. Abychom tyto vazby byli schopni zachytit v síťovém grafu přidáváme do něj virtuální aktivity (viz tečkované čáry na obr. 7.2).

Tyto virtuální činnosti neodpovídají žádným činnostem reálným a jejich délka z pohledu časového je rovna nule. Jejich úkolem je tedy pouze zpřehlednění výsledného grafu. K tomu je potřeba také dodat, že notace použitá na obr. 7.2 odpovídá notaci používané v původní způsobu zaznačení sítě používaném při ručním zpracování. Softwarové nástroje používají, jak později uvidíme, trochu jinou notaci.

I v síťovém grafu je možno vyznačit kritickou cestu, na obr. 7.2 je tato cesta opět vyznačena červenou barvou.

Možná Vás napadne otázka, proč k jednomu úkolu (identifikace kritické cesty) použít dva různé grafy. Různé grafy používáme, protože identifikace kritické cesty je pouze jedním z úkolů, které tyto

grafy plní. Primárním úkolem síťového grafu je přehledně vizualizovat závislosti (návaznosti) mezi jednotlivými činnostmi. Primárním úkolem harmonogramu je vizualizovat činnosti přesně z hlediska časového, za účelem optimalizace využití zdrojů pro nekritické činnosti včetně jejich případného časového posunu.

Přínosy harmonogramů/síťových grafů projektů:

- identifikace kritické cesty
- optimalizace využití zdrojů pro vykonání nekritických činností (včetně posunutí termínů v mantinelech vytyčených kritickou cestou)
- kontrola realizovaných činností, podklady pro efektivní řízení

Příklad
Spočtěte délku kritické cesty pro příklad, se kterým jsme pracovali v této kapitole.

Řešení
Možné cesty
1 – A,B,C,D,E,I,J = 2 + 10 + 17 + 5 + 5 + 21 + 1 = 61
2 – A,B,C,D,F,I,J = 2 + 10 + 17 + 5 + 5 + 21 + 1 = 61
3 – A,B,C,D,H,I,J = 2 + 10 + 17 + 5 + 14 + 21 + 1 = 70
4 – A,B,C,D,G,I,J = 2 + 10 + 17 + 5 + 5 + 21 + 1 = 61

Kritická cesta je cesta 3 (A,B,C,D,H,I,J), projekt bude trvat 70 dní.

Kontrolní otázky

1. Co je to metoda CPM?
2. Vysvětlete pojetí návazností z hlediska následníků resp. předchůdců.
3. Proč používáme harmonogram i síťový graf, nestačil by pro řešení všech problémů pouze jeden z nich?
4. Proč klademe takový důraz na hledání kritické cesty, k čemu je to dobré?

7.2 Project Libre - softwarová podpora projektového řízení

Průvodce studiem
V této kapitole budeme prakticky demonstrovat použití softwaru pro podporu řízení na softwarovém balíku Project Libre, většina z toho, co se zde naučíte je ale také přímo aplikovatelná v jiných softwarech pro podporu řízení projektů, ačkoliv se tyto balíky budou do určité míry zejména grafickým uživatelským rozhraní (Graphical User Interface (GUI)) lišit. Pokud tedy máte na počítači instalován jiný softwarový produkt určený pro projektové řízení, nemusíte jej nahrazovat Project Libre, tedy pokud nechcete.

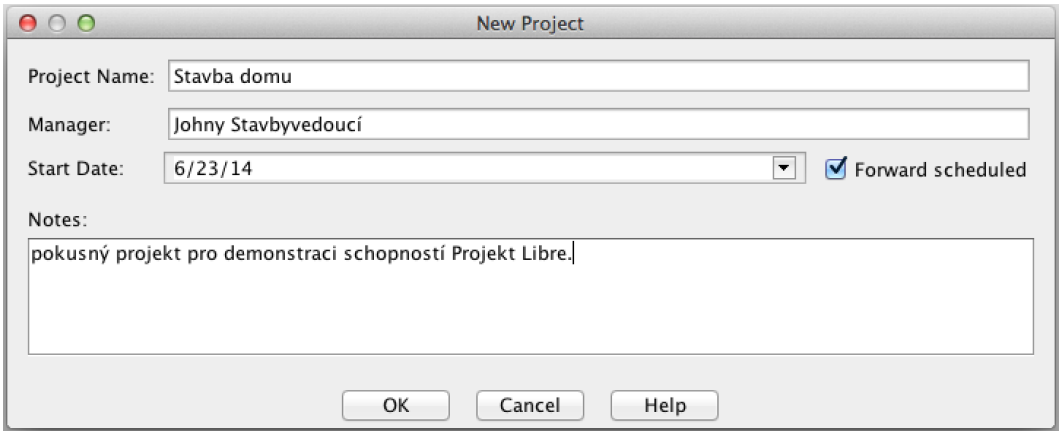
Před použitím je nutné software stáhnout z jeho domácích stránek [17]. Začátečníkům bych doporučoval stáhnout si přímo binární distribuci Project Libre určenou pro jejich operační systém. Jelikož se ale jedná o open source, máte možnost také stáhnout zdrojové kódy a manipulovat i přímo s nimi.

Pro provoz balíku budete potřebovat mít instalován také Java Runtime Environment (JRE). Pokud je na svém počítači nemáte ještě nainstalováno, můžete si je stáhnout ze stránek společnosti Oracle, která JRE vyvíjí: `http://www.oracle.com/technetwork/java/javase/downloads/index.html`.

Dejte si pozor, aby jste zvolili poslední verzi JRE. Nezapomeňte také, že budete muset JRE v průběhu času udržovat. JRE samo detekuje nové verze - je tedy pouze potřeba povolit stažení a instalaci, až Vás k tomu JRE vyzve. Z bezpečnostních důvodů doporučuji také zakázat použití JRE ve webových prohlížečích - stačí deaktivovat plugin, pokud některá z aplikací, které na internetu využíváte přítomnost podpory Javy nevyžaduje.

Dvojklikem na staženém souboru zahájíte instalační proces. V čase psaní těchto textů (červen 2014) byla poslední dostupnou verzí Project Libre verze 1.5.9. Po dokončení instalace je možno program začít používat.

Po spuštění programu je uživatel vyzván, aby založil nový projekt nebo otevřel projekt existující. Jelikož právě s programem začínáme, nemáme k dispozici žádný existující projekt, budeme proto muset vytvořit projekt nový. Při založení nového projektu je nutné zadat jméno projektu, projektového managera, datum počátku projektu. Projekt můžeme doplnit poznámkami, ale to není povinné. Obrazovka s definicí nového projektu je na obr. 7.3.



Obrázek 7.3: Založení nového projektu v Project Libre

Různé softwarové balíky pro projekty podporují různé vlastnosti. Např. MS Project umožňuje specifikovat druh kalendáře, který se má v projektu využívat (např. bussiness hours - 8-mi hodinový pracovní den, 5 dní v týdnu, nebo třeba třísměnný provoz 7 dní v týdnu).

V rámci našeho experimentování s Project Libre budeme realizovat příklad z předchozí kapitoly. Budeme tedy stavět rodinný dům. Kliknutím na OK vytvoříme projekt.

Nový projekt je automaticky otevřen v režimu Ganttova diagramu, který nám na jedné straně umožňuje definovat základní vlastnosti jednotlivých aktivit a na straně druhé je rovnou vizualizuje do podoby harmonogramu. Mezi základní vlastnosti řadíme především:

- jméno aktivity
- délka trvání aktivity
- datum počátku a konce práce na aktivitě (při definici návazností se přepočítává automaticky podle návazností a délky trvání aktivity)
- předchůdci
- zdroje

Jednotlivé aktivity projektu jsme již specifikovali v předchozí kapitole, konkrétně v tabulce 7.1, proto údaje pouze přepíšeme do GUI programu. Začneme definicí jmen projektových činností. Všimněte si, že při vytváření Project Libre automaticky doplňuje některé informace - předpokládá délku trvání 1 den s počátkem totožným se startem projektu.

Informace zadané do GUI se také přímo zaznamenávají v pravé části obrazovky v harmonogramu. Předpokládaná kritická cesta je zvýrazněna červenou barvou. Další sloupce pro aktivity zatím vyplněné nejsou - ty budeme muset definovat sami.

Nejprve provedeme změnu délky jednotlivých činností projektu. Ty jsou aktuálně nastaveny na 1 den a my z tabulky 7.1 víme, že délky řady činností budou odlišné. Délku můžeme specifikovat ve dnech, týdnech nebo měsících, Project Libre přitom předpokládá, že zadáváme délku ve dnech. 5 v položce délka je tedy interpretováno jako 5 dní (v případě, že se používá běžný kalendář bussiness hours, pak se jedná také o ekvivalent 1 týdne). Délkovou jednotku můžeme přidat přímo za číselnou specifikaci délky, např. 3d (tři dny), 2w (dva týdny), 1m (jeden měsíc). Zkratky časových jednotek jsou odvozeny z angličtiny, tedy d = day, w = week, m = month.

Mějte na paměti, že týden může znamenat také něco jiného než si myslíte, v závislosti na kalendáři, který se použije, implicitně přitom máme přednastaven kalendář bussiness hours. Kalendáře ale lze měnit na úrovni jednotlivých činností - tedy každá činnost může mít jiný kalendář a proto se definice času může poněkud zkomplikovat

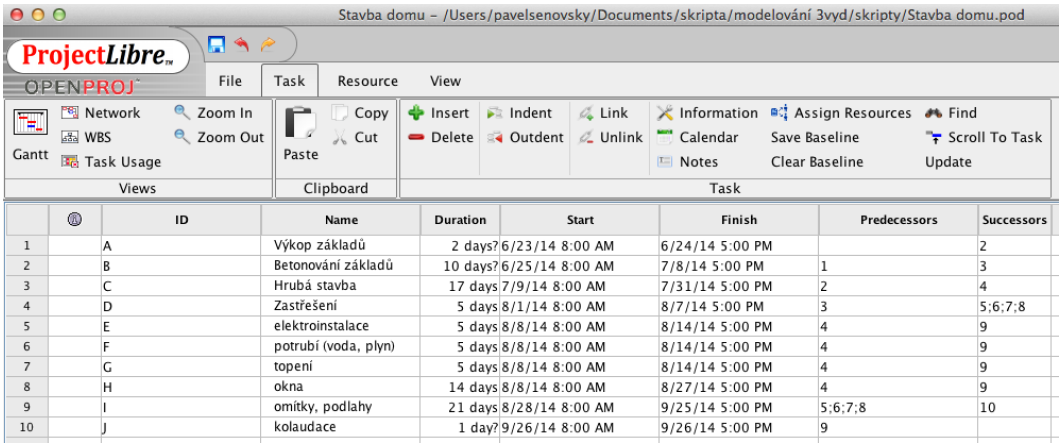
Informace o jednotlivých činnostech z tab. 7.1 se také liší v jiné podstatné charakteristice. V tabulce jsme jednotlivé činnosti měli identifikované pomocí písmen A - J, v Project Libre jsou ale identifikovány číslem řádku, na kterém jsou zapsány (1-10). Abychom zajistili kompatibilitu (a jednoduchost definice návazností), přidáme další sloupec, do kterého zapíšeme identifikační písmena.

Přidání nového sloupce je snadné. Klikneme prostě pravým tlačítkem myši na záhlaví sloupce, kam chceme nový sloupec přidat a v kontextovém menu zvolíme vložit sloupec (Insert Column ...). Kontextové menu nám také umožňuje sloupce odebírat. Tady doporučuji jistou zdrženlivost - nechejte si zobrazené alespoň základní sloupce.

Volba vložit sloupec spustí dialogové okno pro výběr typu sloupce. Typy jsou seřazené v přehledném seznamu. Z některých názvů je jasně patrné, k čemu sloupec slouží, u jiných tomu tak není. Pro náš účel potřebujeme sloupec typu Text - protože je jich tam více, zvolíme *Text1*. Jedná se o obecný sloupec, umožňující vložit jakýkoliv text. Po vložení sloupce změníme z jeho kontextového menu jeho název (volba Rename). Nové jméno bude ID. Do nově vytvořeného sloupce pak můžeme z tabulky 7.1 přepsat identifikátory.

Takže zbývá doplnění chybějících údajů - tedy návazností. V Ganttově diagramu již máme dostupný sloupec pro předchůdce, pokud ale preferujete zadaní spíše následníků - můžete, je ale potřeba přidat nový sloupec typu následník (successor). Project Libre mezi předchůdci a následníky přepočítává automaticky, takže musíme definovat pouze jedno.

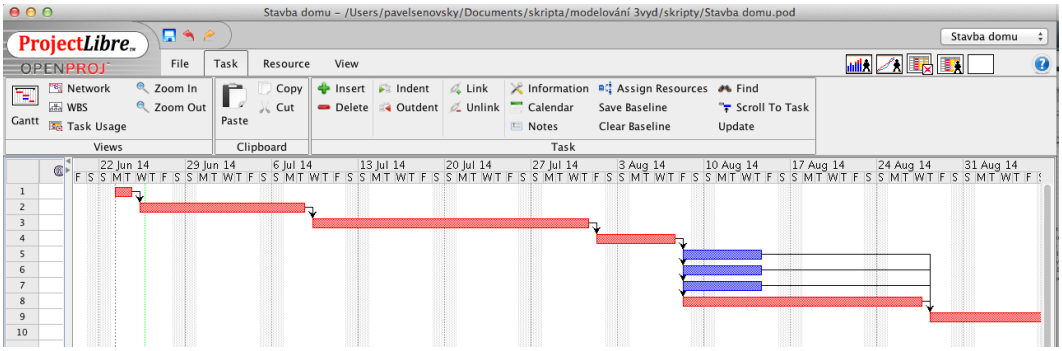
V tomto případě použiji předchůdce. Nezapomeňte, že identifikátor v Project Libre je číslo řádku, námi zadaný sloupec ID tedy slouží pro jednodušší překlad mezi číslem řádku a původním označením činnosti. Při zadávání většího množství např. předchůdců, oddělujeme jednotlivé předchůdce středníkem. Ve výsledku získáte zápis podobný tomu na obr. 7.4.



		ID	Name	Duration	Start	Finish	Predecessors	Successors
1		A	Výkop základů	2 days?	6/23/14 8:00 AM	6/24/14 5:00 PM		2
2		B	Betonování základů	10 days?	6/25/14 8:00 AM	7/8/14 5:00 PM	1	3
3		C	Hrubá stavba	17 days	7/9/14 8:00 AM	7/31/14 5:00 PM	2	4
4		D	Zastřešení	5 days	8/1/14 8:00 AM	8/7/14 5:00 PM	3	5;6;7;8
5		E	elektroinstalace	5 days	8/8/14 8:00 AM	8/14/14 5:00 PM	4	9
6		F	potrubí (voda, plyn)	5 days	8/8/14 8:00 AM	8/14/14 5:00 PM	4	9
7		G	topení	5 days	8/8/14 8:00 AM	8/14/14 5:00 PM	4	9
8		H	okna	14 days	8/8/14 8:00 AM	8/27/14 5:00 PM	4	9
9		I	omítky, podlahy	21 days	8/28/14 8:00 AM	9/25/14 5:00 PM	5;6;7;8	10
10		J	kolaudace	1 day?	9/26/14 8:00 AM	9/26/14 5:00 PM	9	

Obrázek 7.4: Project Libre - definice Ganttova diagramu

Zadáním předchůdců/následovníků jsme také nastavili spojení mezi jednotlivými činnostmi, proto harmonogram v Ganttově diagramu již začíná dávat trošku lepší smysl, viz obr. 7.5.



Obrázek 7.5: Project Libre - harmonogram

V harmonogramu jednotlivým činnostem odpovídají čáry, s délkou odpovídající době trvání činnosti. Všimněte si šedých svislých sloupců, které odpovídají dnům pracovního klidu. Činnosti jsou

také rozlišeny barevně - činnosti na kritické cestě jsou zvýrazněny červenou barvou, zatímco nekritické činnosti jsou znázorněny barvou modrou. Vzájemná návaznost je zachycena pomocí šipek.

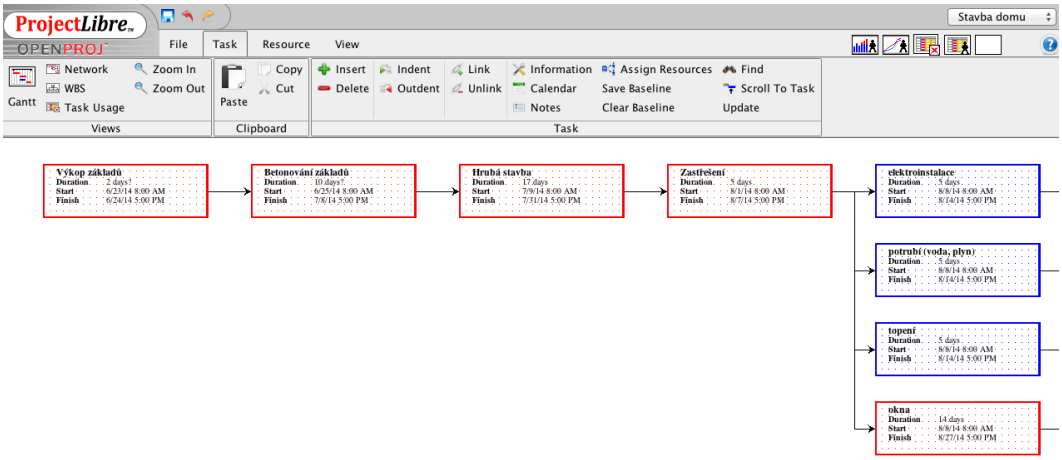
V našem případě jsou činnosti seřazeny, ale harmonogram toto nevyžaduje - stejnou funkčnost dosáhneme i když budou činnosti v seznamu činností zaznamenány na přeskáčku.

V případě, že bychom vytvářeli harmonogram ručně, museli bychom se na tomto místě zastavit, protože další manipulace by prostě technicky nebyla možná. Softwarová realizace harmonogramu dává uživateli ale další možnosti. Pomocí tažení myší lze celou činnost posunout v čase. Jelikož tažení samo o sobě nemění strukturu projektu (návaznost činností), termíny navazujících činností se přepočtou automaticky. Tažením počátku nebo konce lze změnit délku trvání činnosti. Konečně lze sledovat procento splnění další činnosti, což je nedocenitelná funkce pro účely řízení průběhu projektu.

Pro změnu procenta splnění najedeme myší na začátek činnosti - kurzor se změní do podoby %> (procento s šipkou doprava). Procento splnění měníme kliknutím a tažením. Vizuálně pak bude možné procento splnění činnosti identifikovat pomocí vyplnění činnosti černou barvou proporcionálně odpovídající procentu splnění.

Alternativním pohledem na činnosti a jejich vzájemnou vazbu je síťový diagram. V síťovém diagramu jednotlivé činnosti jsou reprezentovány uzly sítě (na rozdíl od ručního zpracování síťového grafu, kde činnosti byly reprezentovány hranami sítě). Spojení pak umožňují identifikovat jednoduše, které činnosti musí být dokončeny předtím, než mohou ostatní započít. Síťový diagram je generován automaticky na základě základních časových vlastností činností.

I v síťovém grafu je možno vizuálně identifikovat kritickou cestu - zvýraznění je provedeno opět červenou barvou. V Project Libre se síťový pohled aktivuje Task tab -> tool Network. Pro náš příklad je síťový graf zachycen na obr. 7.6.



Obrázek 7.6: Project Libre - síťový graf

Máme tedy k dispozici dva různé pohledy - harmonogram/Ganttův diagram a síťový diagram, které zachycují projekt a jeho průběh v čase, stejně jako kritickou cestu. Možná Vás napadla otázka - proč je to potřeba? Pro pochopení rozdílu je potřeba se zaměřit na rozdíly mezi oběma pohledy. Síťový diagram umožňuje, konsolidovat činnosti (a některé základní informace o nic) a jejich návaznosti na relativně malém prostoru. Harmonogram obvykle zabírá prostor větší, umožňuje ale lepší časové pochopení průběhu projektu a umožňuje snadnější manipulaci s časovými, lidskými i materiálovými zdroji (rezervami), stejně jako sledování průběhu projektu.

V předchozím odstavci jsme zmínili zdroje - může nám Project Libre pomoci i s nimi? Ano, může - zdroje, dostupné pro projekt, lze spravovat na záložce zdrojů (Resource).

Vytvořme tři zdroje - pracovník 1 a 2 a písek. Pro zdroje je potřeba specifikovat jejich typ. Typem se přitom rozumí rozlišení mezi zdroje lidskými a materiálovými. Oba druhy zdrojů mají trochu odlišné vlastnosti, se kterými se dále pracuje.

Pro lidské zdroje vybereme typ pracovní (work). Těmto zdrojům je možné přiřadit e-mail, seskupit je do pracovních týmů apod. Lidským zdrojům je nutno přiřadit také maximální možné využití (max. units). V případě lidí pracujeme s procenty, které si lze představit jako úvazek. 100 % proto znamená, že pracovníka můžeme plně využít pro projekt, 50 %, že můžeme využít pouze polovinu jeho kapacity (poloviční úvazek). Lidské zdroje nemá smysl kapacitně využívat na více než 150 %.

S každým pracovníkem jsou spojeny některé náklady - jedná se o hodinovou mzdu (standard rate) a sazbu za přesčasy (overtime rate) - tedy za dobu, kdy by došlo k překročení „nasmlouvaného“ pracovního výkonu, jak je specifikován procentem úvazku. Přesčasová sazba je obvykle vyšší než sazba standardní.

Náklady na použití (cost per usage) jsou náklady přímo spojené s nasazením pracovníka, které ale nelze chápat jako mzdu. Např. zde může být náklad na dopravu dělníků na místo stavby z ubytovny apod.

Náběh nákladů (Accrue at) umožňuje specifikovat, kdy náklad fyzicky vznikne - kdy bude nutné náklad reálně zaplatit. V tomto případě máme tři možnosti:

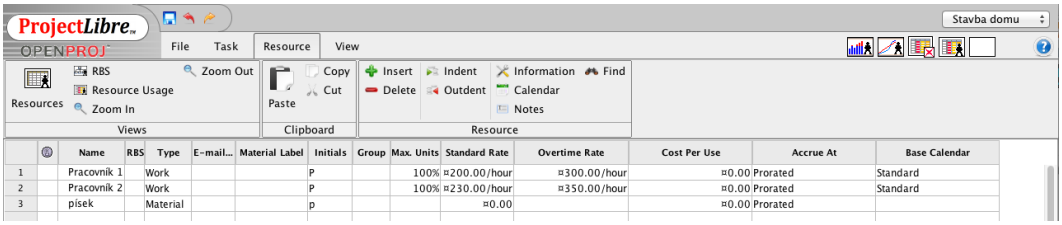
- na začátku,
- na konci,
- průběžně.

Všechny tři způsoby mohou být v případě lidských zdrojů použity. Do průběžného financování spadá běžný pracovní poměr - tedy pracovník je kmenovým zaměstnancem firmy a dostává běžnou mzdu splatnou v dohodnutých termínech. Financování na konci volíme v případě, že potřebujeme získat kontrolu nad určitým pracovním výkonem. Placení se provede až po jeho dokončení (převezmeme práci a zaplatíme). Placení na začátku předpokládá nutnost finanční investice nutné pro provedení prací - např. že součástí je nákup něčeho (např. materiálu), který je nutný pro výkon práce. V tomto případě, ale takový materiál již nezavádíme do materiálových zdrojů. Tento typ financování se používá spíše pro konraktory (subdodávky).

Každému pracovníkovi je možné nastavit jeho vlastní kalendář a to standardní (business hours), 24 hodin a noční směny (night shift). Nastavení kalendáře rozhoduje o tom, jak a kdy bude práce odvedena.

Pro materiálové zdroje řada sloupců nemá smysl, takže je nepoužíváme, řada sloupců má ale trošku jiný smysl než v případě zdrojů lidských. Standardní sazba (Standard rate) nemá charakter hodinové sazby mzdy, ale cena, za kterou pořizujeme jednotku zdroje. Jednotky v tomto případě mohou být jakékoliv - tuny, kilogramy, litry, cena za kus, čtvereční metr apod. Cena za užití je interpretačně podobná - i materiálový zdroj je potřeba dostat na místo, popř. nějak zpracovat.

Příklad definic zdrojů naleznete na obr. 7.7.



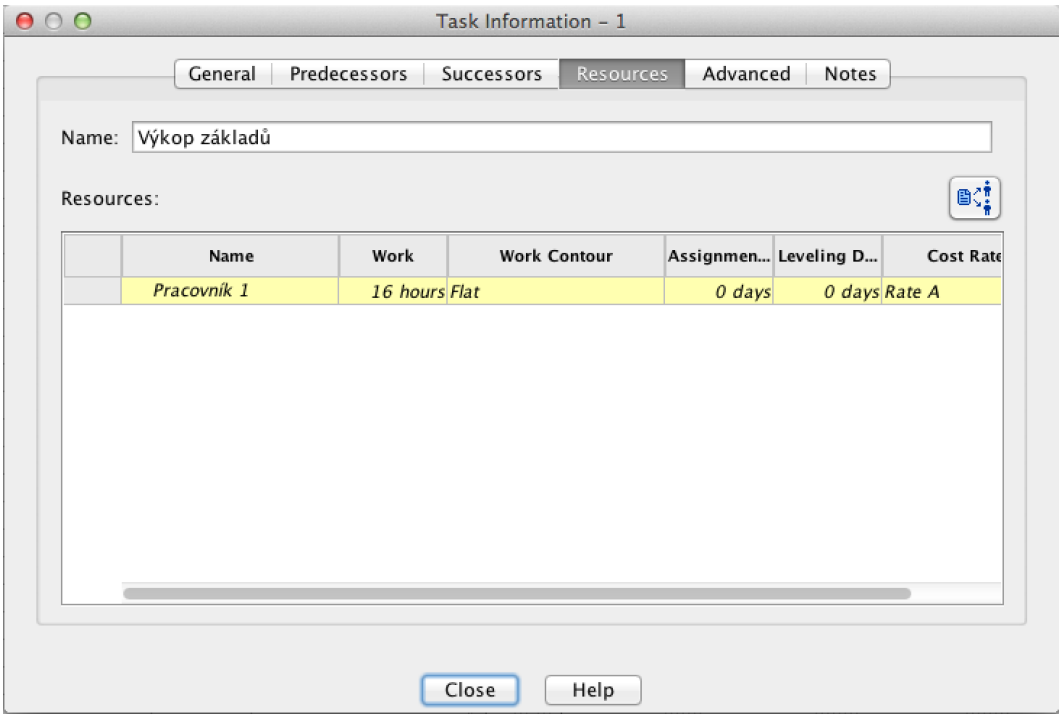
Obrázek 7.7: Project Libre - definice zdrojů

Vraťme se zpět do Ganttova diagramu - právě zde totiž můžeme přiřadit zdroje k jednotlivým úkolům. Přiřazení je možné provést dvojím způsobem, buď klikneme myší do sloupce jméno zdroje (resource name) nebo ve vlastnostech jednotlivých činnosti (spustí se dvojklikem na dané vlastnosti). Dialogové okno s podrobnostmi činnosti umožňuje nastavování celé řady dalších zajímavých vlastností vztahujících se k činnostem je na obr. 7.8.

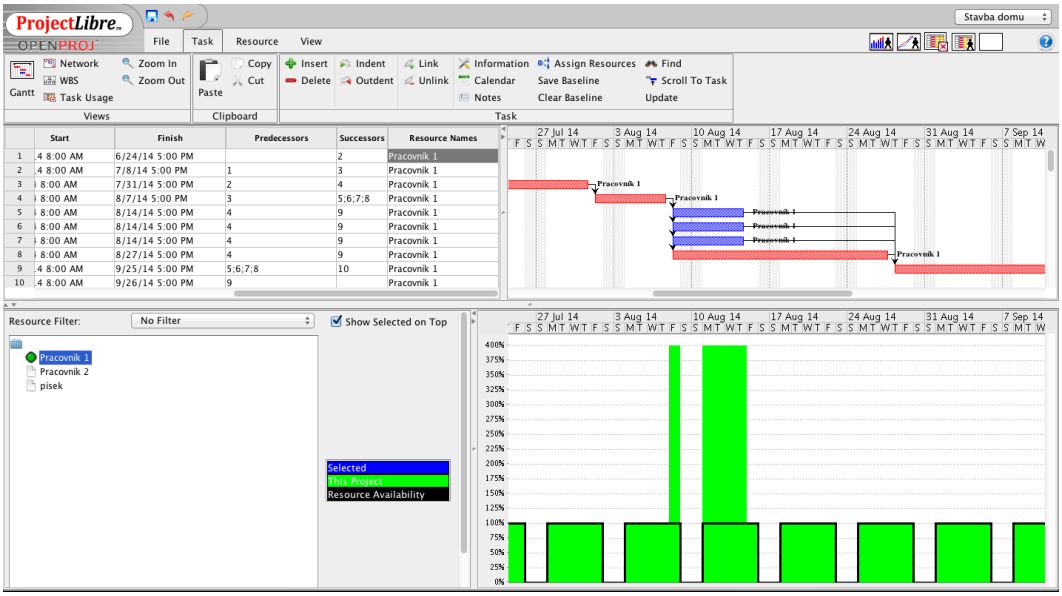
Zkusme pokusně přiřadit všem činnostem pracovníka 1. Celý proces můžete uspíšit prostým kopírováním jména zdroje ve sloupci jmen zdrojů na záložce Ganttova diagramu.

Podívejme se, jak jsou naše zdroje využívány. Nejprve prozkoumáme histogram zdrojů. Histogram můžeme spustit kliknutím na ikonu histogramu v pravém horním rohu okna programu. Pro náš příklad je histogram zachycen na obr. 7.9.

Histogram je spojen s harmonogramem, tedy posunuje se časově tak, aby odpovídal aktuální pozici harmonogramu. Z našeho hlediska je nejzajímavější období projektu spojené s činnostmi 5 - 8 (E - H). Z histogramu nám vyplývá, že zdroj Pracovník 1 bude mít pekelný týden, kdy jeho pracovní den bude mít 32 hodin. Naše využití tohoto zdroje je tedy na 400 %, což je samozřejmě logický nesmysl. Všimněte si, že Project Libre odlišuje vizuálně mezi běžnou dobou a přesčasy - to nám umožňuje jednoduše vizuálně identifikovat jakákoliv, z hlediska využití zdrojů, problémová místa - a to i v případě, že nejsou tak extrémní, jako v našem příkladu.



Obrázek 7.8: Project Libre - podrobnosti o činnosti



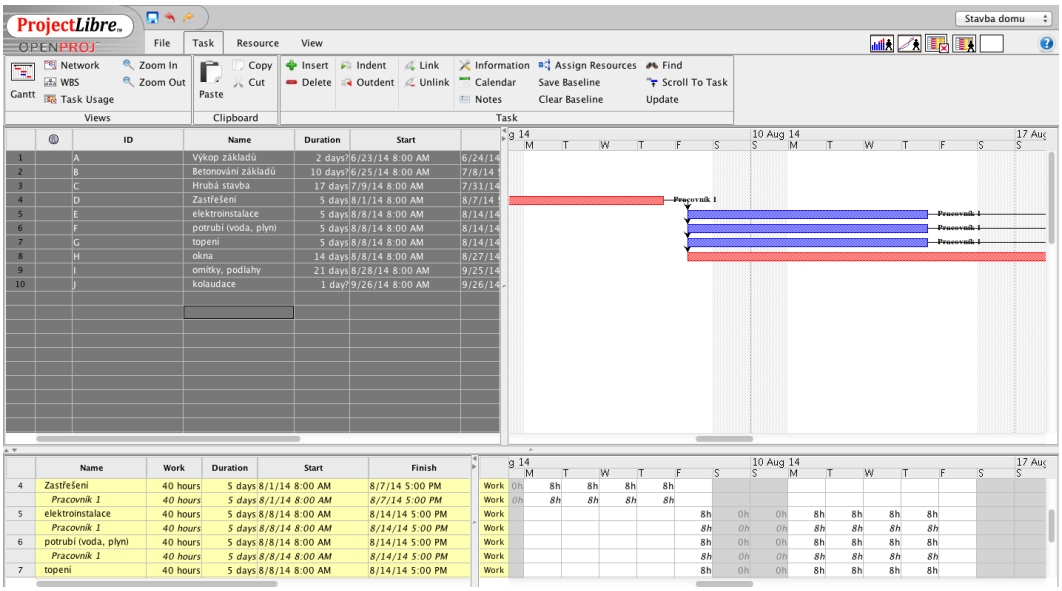
Obrázek 7.9: Project Libre - histogram zdrojů

Můžeme se také podívat na zdroje z hlediska pracovních hodin - volbou využití zdrojů (resource usage), viz obr. 7.10. Software samotný tedy disponuje celou řadou nástrojů, které nám umožní identifikovat podstatu problému a umožnit nám tak něco s ním udělat.

7.3 Optimalizace v ostatních síťových modelech

Projektový management má určité specifické potřeby, proto zkušenost se síťovými modely není do sítí fyzických úplně jednoduše přenositelná. V rámci fyzických sítí, jako jsou sítě přepravní (silniční, železniční), produktovody (ropa, voda, plyn), čelíme různým problémům odlišného charakteru.

Např. u silniční sítě nás může zajímat nejkratší cesta z místa A do místa B, což by filozoficky odpo-



Obrázek 7.10: Project Libre - užití zdrojů

Management projektu
Vytvořte projekt podle vlastního zadání, pokud se Vám nechce přemýšlet - vytvořte projekt popisující Vaše studium v tomto semestru. Pozornost v takovém případě věnujte termínům semestrálních projektů, případných testů, termínům zkoušek apod.

vídalo kritické cestě, jak ji známe z projektového řízení, stejně často nás však zajímá cesta nejrychlejší a tedy ne nutně nejkratší. Hledání nejrychlejší cesty pak souvisí s hodnocením kapacit dostupných cest.

S jakými problémy se tedy při analýzách fyzických sítí setkáváme:

- hledání nejkratší cesty mezi dvěma body,
- hledání cesty s největší kapacitou (průtokem),
- výpočet celkové kapacity sítě,
- hledání úzkých hrdel v síti,
- návrh optimální sítě mezi předem stanovenými a lokalizovanými uzly.

Portfolio problémů je tedy poměrně bohaté. Podobně bohaté je také portfolio různých algoritmů, které používáme pro jejich řešení. Není to tedy tak, že bychom měli jeden univerzální algoritmus/nástroj, který by byl schopen vyřešit všechny typy problémů. Většina z těchto algoritmů je ale principiálně podobná - tedy je vytvořena na základě obdobných předpokladů a technik výpočtu.

Síťová teorie zaznamenala svůj největší rozmach v 50. a 60. letech minulého století. Algoritmy používané pro výpočty nutně tedy nepotřebují výpočetní techniku, avšak při jejím použití se vše výrazně zrychluje a zjednodušuje.

Z hlediska předpokladů je nejdůležitějším to, že síť je uzavřená - tedy neexistují v síti žádná spojení vedoucí mimo modelovanou síť. Toto omezení je potřeba pečlivě uplatňovat a to z toho důvodu, že většina modelovaných sítí není uzavřená přirozeně, musíme tedy obvykle vybrat určitou část sítě a tu uzavřít.

Dalším omezením je požadavek, aby tok, který vstupuje do počátečního uzlu, také z posledního uzlu vystupoval. Obvykle se tedy předpokládá, že v rámci sítě nedochází ke ztrátám. To neznamená, že úlohy, které ztráty zohlednit potřebují, nejsou řešitelné, pouze algoritmy které se pro řešení používají, jsou však složitější a proto se jimi v tomto předmětu (a tomto učebním textu) zabývat nebudeme.

Podívejme se na možné řešení nějakého jednoduchého problému z hlediska algoritmu, který je možné použít. Pokusme se vyřešit problém **hledání maximálního toku v přepravní síti mezi dvěma zadanými uzly**. Základní překážkou pro řešení tohoto typu problémů je fakt, že jednotlivé

hrany sítě obvykle mají odlišné přepravní kapacity. Pokud bychom takový problém řešili pomocí kombinatoriky, tak bychom nepochybně neuspěli, protože v takovém případě komplexnost problému roste geometrickou řadou s počtem uzlů a spojení mezi nimi.

Z tohoto důvodu používáme pro řešení těchto problémů spíše iterační metody, pomocí kterých redukujeme analyzovanou síť na strom s kořenem v počátečním uzlu, který pak spočteme. Pro případ hledání maximálního možného toku mezi uzly často používáme **Ford-Fulkersonův algoritmus**, který pracuje následovně (převzato z [7]).

Vstupy: graf G s kapacitou toku c, stanoveným počátečním uzlem s a koncovým uzlem t
Výstup: tok f z s do t, který je maximální

1. $f(u,v) \leftarrow 0$ pro všechny hrany (u,v)
2. dokud existuje cesta p z s do t v G_f, taková že $c_f(u,v) > 0$ pro všechny hrany $(u,v) \in p$:
   (a) najdi $c_f(p) = min\{c_f(u,v)|(u,v) \in p\}$
   (b) pro každou hranu $(u,v) \in p$
      i. $f(u,v) \leftarrow f(u,v) + c_f(p)$ (pošli tok po hraně)
      ii. $f(v,u) \leftarrow f(v,u) - c_f(p)$(tok může být později „navrácen“)

Ford-Fulkersonův algoritmus ve skutečnosti není složitý, pokud si uvědomíte, jak funguje, ale k tomu je potřeba si postup algoritmu vizuálně představit. Jednu z možných ukázek funkce je možné najít v Ford-Fullkerson Demu [46].

Softwarová podpora pro výpočty fyzických sítí
Výpočet fyzických sítí s použitím Ford-Fullkersonova algoritmu (i algoritmů jiných) bez použití výpočetní techniky je realtivně zdlouhavé. Naštěstí existuje celá řada nástrojů, které jsou schopny nasadit tyto algoritmy automatizovaně a tím nás „odstínit“ od nutnosti zabývat se algoritmem jako takovým. Nalezneme je v systémech GPS (Global Positioning System (GPS)) a v řadě dalších
My pro výpočty použijeme program pro numerické výpočty *SciLab*. Použití pro výpočty ve fyzických sítích naleznete v následující kapitole.

Dalším typem problémů, které jsme nuceni často řešit je **hledání nejkratší cesty mezi dvěma uzly**. Pro řešení tohoto typu problémů se často používá **Dijkstrův algoritmus** pojmenovaný po Nizozemském matematikovi, který jej v 50. letech vyvinul. Dijkstrův algoritmus pracuje následovně (převzato z [3]):

1. vytvořme seznam vzdáleností, seznam minulých uzlů, seznam navštívených uzlů a současný uzel
2. všechny hodnoty v seznamu vzdáleností jsou nastaveny na nekonečno vyjma počáteční uzlu, který je nastavena na 0.
3. všechny hodnoty v seznamu navštívených uzlů nastavíme na hodnotu nepravda.
4. všechny hodnoty v seznamu minulých uzlů nastavíme na hodnotu indikující, že zatím nebyly definovány - může to být hodnota null nebo třeba Nothing podle zvoleného programovacího jazyka.
5. Současný uzel je nastavena jako počáteční uzel.
6. Současný uzel označíme jako navštívený.
7. aktualizujeme vzdálenosti a seznam minulých uzlů na základě uzlů, které mohou být navštíveny přímo ze současného uzlu.
8. Aktualizujeme současný uzel na uzel zatím nenavštívený, který může být dosažen nejkratší cestou z počáteční hrany.
9. opakujeme (od kroku 6), dokud nenavštívíme všechny uzly.

Existuje celá řada dalších algoritmů, které jsou používány pro různé typy úloh. Jelikož řada z těchto algoritmů je již staršího data (což však neznamená, že by byly nepoužitelné, nebo nevhodné k použití) existuje pro ně obvykle rozsáhlá dokumentace včetně implementací v řadě výpočetních prostředí nebo programovacích jazyků. Dobrým výchozím bodem pro začátečníka v této oblasti je Wikipedie a její článek List of Algorithms [11], který slouží jako rozcestník pro jednotlivé algoritmy, resp. jejich hesla na Wikipedii.

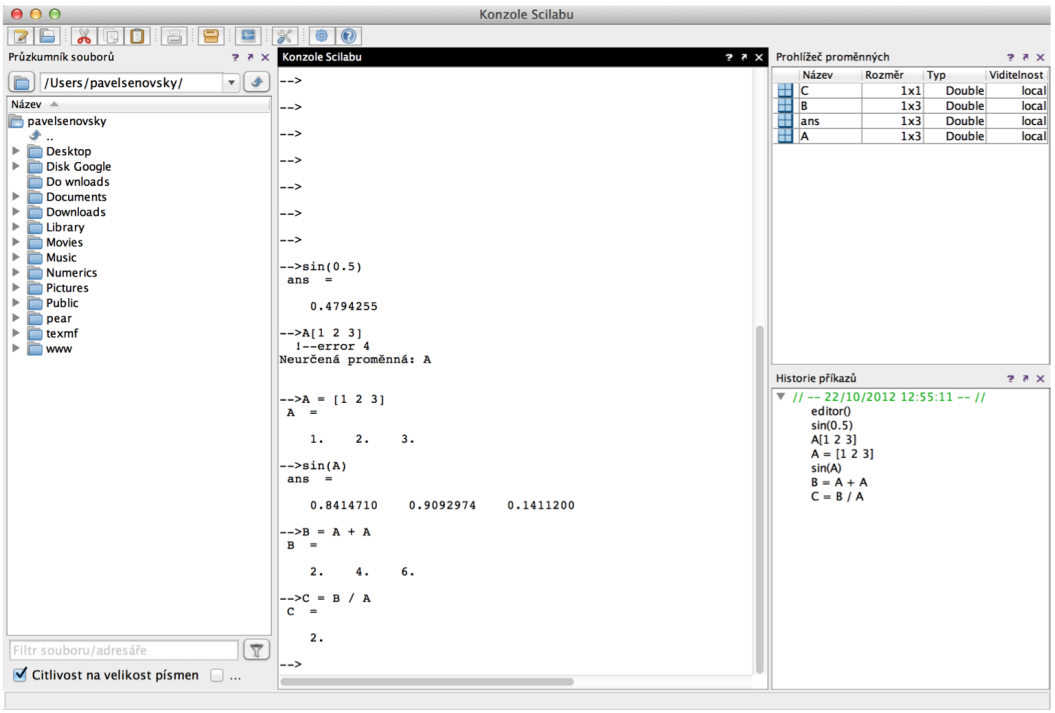
Kontrolní otázky

1. Jaká jsou omezení modelů fyzických sítí?
2. Jaké typy problémy řeší síťové modely?
3. Popište rozdíly mezi Ford-Fullkersonovým a Dijstrovým algoritmem?
4. Jaké jsou jejich společné znaky?

7.4 Numerické výpočty pomocí SciLab

Pro realizaci výpočtů v počítači si nejprve musíme připravit prostředí. Výpočty budeme provádět v nástroji SciLab [21], což je open source nástroj určený pro numerické výpočty, filozoficky podobný MathLabu. Ovládá se integrovaným programovacím jazykem, který je z větší části kompatibilní s jazykem MathLab. Verzi pro operační systém, který používáte, můžete stáhnout bezplatně `http://scilab.org`.

SciLab je dostupný pro operační systémy Windows, Linux a Apple OS X. Pro tyto operační systémy je SciLab dostupný v binární formě, kterou je jednoduché nainstalovat - prostě následujte pokyny průvodce instalací. Práce se SciLab je buď v režimu interaktivním - uživatel zadá příkaz a sleduje odezvu v konzoli programu, nebo je možné připravit dávku příkazů (SCE soubor), která se bude chovat jako malý program. Rozhraní SciLab je zachyceno na obr. 7.11.



Obrázek 7.11: SciLab - grafické uživatelské rozhraní

Pokud nemáte předchozí zkušenosti se softwary jako je MathLab, budete potřebovat krátký úvod do filozofie programu. Bohužel v těchto skriptech není prostor základy SciLabu probrat. Na Internetu ale existuje celá řada kvalitních zdrojů, doporučit je možné např. A Short Introduction to SciLab od Terence Leung Ho Yin a Tsing Nam Kiu [34]. Podrobnější úvod do problematiky SciLabu poskytuje Katedra chemického inženýrství ve SciLab Primer [33].

SciLab Primer pokrývá některá pokročilá témata, proto doporučuji začít krátkým úvodem (nebo jakýmkoliv jiným úvodem). Jednou z výhod SciLabu je, že je poměrně oblíbený a tak existuje celá řada webových sídel, které se zabývají různými aspekty funkcionality SciLab - takže pokud Vám nevyhovují materiály výše, existuje jistě celá řada dalších - stačí hledat.

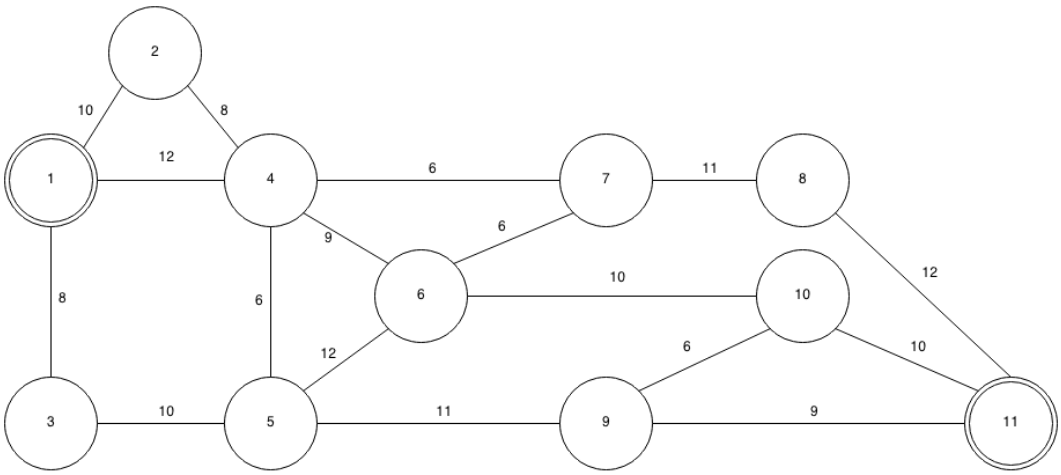
Nyní, když již máme určitou jistotu v užití programu, se můžeme ve výkladu posunout k řešení samotných optimalizačních algoritmů. Jedinou překážkou použití je fakt, že podpora těchto algoritmů

Základy SciLabu
Věnujte čas zvládnutí základů SciLabu. Minimálně potřebujete pochopit proměnné, vektory a matice a také operace s nimi.

není přímou součástí SciLab - je nutné ji doplnit pomocí samostatného výpočetního balíčku *Numerics* [15] vyvinutého na Fakultat fur Mathematic Ruhr Universitat Bochum. Balíček Numerics je dostupný ke stažení z webových stránek univerzity: `http://www.rub.de/num1/softwareE.html`. K balíčku je dostupná i dokumentace v anglickém a německém jazyce.

Balíček samotný podporuje větším množství algoritmů, než které budou probírány v těchto skriptech, takže pokud Vás problematika zaujala, může být studium dokumentace dobrým výchozím bodem dalšího studia.

Balíček stáhněte a rozbalte do složky, dle vlastního výběru - bylo by ale dobré vědět, která to byla :-), jelikož tuto informaci budeme potřebovat pro zprovoznění balíčku ve SciLab. Pro práci budeme potřebovat také síť. Pro náš příklad použijeme síť převzatou z knihy *Kvantitativní metody v manažerském rozhodování* [31]. Grafické znázornění sítě je dostupné na obr. 7.12.



Obrázek 7.12: Základní síť (převzato Gros [31])

Interpretace sítě je odlišná podle typu problémů, které budeme řešit. Pro Ford-Fulkersonův algoritmus číslo u spojnic znamená kapacitu. Pro Dijkstrův algoritmus by ale interpretace byla odlišná - jednalo by se o délku jednotlivých cest.

Nejprve se zaměříme na řešení Ford-Fulkersonova algoritmu - tedy budeme řešit problém kapacity mezi uzly 1 a 11.

Pro dosažení řešení potřebujeme vytvořit síťovou strukturu pro naše výpočty. Numerics balík umožňuje definovat síť dvěma způsoby - jako čtvercovou matici nebo seznam vektorů. Začneme s maticí, pro demonstraci použijeme prvních pět uzlů - viz tab. 7.2:

Tabulka 7.2: Specifikace sítě pomocí matice

	1	2	3	4	5	...
1	-%inf	10	8	12	-%inf	
2	10	-%inf	-%inf	8	-%inf	...
3	-%inf	-%inf	-%inf	-%inf	10	...
4	12	8	-%inf	-%inf	6	...
5	-%inf	-%inf	10	6	-%inf	...
...	...	...	...	...	...	...

Jak vidíte, většina spojení mezi uzly v naší síti neexistuje - pro neexistující spojnice vyplníme -

%inf konstantu (záporné nekonečno), pro existující spojení specifikujeme skutečnou kapacitu. Pro větší množství uzlů roste velikost matice exponenciálně - pro naši síť by proto matice měla $11 \cdot 11 = 121$ prvků. Zápis tedy může být poměrně zdlouhavý. To je také důvodem proč preferuji spíše seznam vektorů pro definici sítě, kdy se popisují pouze existující spojení ve formátu [počáteční uzel, koncový uzel, kapacita]. V našem případě by zápis ve SciLab pro uzly 1 - 5 byl následující:

```
list([1, 2, 10], [2, 4, 8], [1, 4, 12], [1, 3, 8], [3, 5, 10], [4, 5, 6])
```

Co do množství informací je tento postup jednodušší. Teď přišel čas vzít rozum do hrsti a použít naše znalosti pro provedení výpočtu.

```
mylib=lib("/Users/pavelsenovsky/Numerics")
net = list([1,2,10], [2,4,8], [1,4,12], [1,3,8], [3,5,10], [4,5,6], [4,7,6], [4,6,9], [5,6,12], [7,8,11], [8,11,12],
     [6,10,10], [10,11,10], [5,9,11], [9,10,6], [9,11,9])
first=1
last=11
ford_fulkerson(net,first,last)
```

Výše uvedený program není složitý, volá funkci ford_fulkerson s parametry specifikující strukturu sítě (net), číslo počátečního (first) a koncového (last) uzlu. Do proměnné mylib načítáme externí knihovnu Numerics pomocí funkce lib s parametrem cesty. V mém případě byl pro výpočet použit OS X, balík Numerics je rozbalen v mém domácím adresáři, odtud cesta /Users/pavelsenovsky/. V případě použití operačního systému Windows by cesta mohla vypadat následovně: c:\Users\pavelsenovsky\. Všimněte si, že je nutné specifikovat celou cestu.

Proměnná net obsahuje definici sítě s použitím vektorové notace. Proměnné first a last v našem případě není ani nutné uvádět, protože funkce předpokládá, že má spočítat kapacity mezi prvním a posledním uzlem - specifikace je nutná, pouze pokud tento předpoklad neplatí.

Po spuštění se provede výpočet - výsledek je v našem případě **25**.

Analogicky můžeme spočítat problém nejkratší cesty mezi dvěma uzly, pro zjištění nejkratší cesty mezi uzly 1 a 11 stačí změnit jméno funkce ford_fulkerson na dijkstra:

```
mylib=lib("/Users/pavelsenovsky/Numerics")
net = list([1,2,10], [2,4,8], [1,4,12], [1,3,8], [3,5,10], [4,5,6], [4,7,6], [4,6,9], [5,6,12], [7,8,11], [8,11,12],
     [6,10,10], [10,11,10], [5,9,11], [9,10,6], [9,11,9])
first=1
last=11
dijkstra(net,first,last)
```

Pokud spustíme program, zjistíme, že nejkratší cesta mezi uzly 1-3-5-9-11 s délkou 38.

Vyzkoušejte výpočet
Specifikujte síť z [46] a vypočtěte nejkratší cestu a kapacitu mezi dvěma zvolenými uzly.

7.5 Numerické výpočty v R

Pro výpočty a také vykreslení síťových grafů můžeme použít také výpočetní prostředí R. I zde nejsou analýzy sítí integrální součástí jádra systému, ale existují rozšiřující balíky, které umožňují tento typ výpočtů provádět a obsahují také nástroje na vizualizaci sítě.

Pro R je k dispozici balík *igraph*. Instalace balíků probíhá ale ve srovnání se SciLabem jednodušeji - pouze se prostředí vydá pokyn pro instalaci a R si balík stáhne a sám nainstaluje.

```
install.packages("igraph")
```

Načtení grafu se provádí pomocí funkce read.graph specifikací externího souboru obsahujícího definici sítě a specifikaci formátu souboru. Podporována je (ve verzi 0.7) řada formátů: edgelist, pajek, graphml, hml, ncol, lgl, dimacs a graphdb. Pokud se parametr formátu neuvede předpokládá se edgelist.

Edgelist je velmi jednoduchý formát specifikuje se počáteční koncový uzel a váha hrany. Naše síť by v tomto formátu vypadala následovně (viz níže). Všimněte si, že formát nevyžaduje, aby jednotlivé hrany byly na samostatných řádcích, všechny hrany tak mohou být na jediném řádku odděleny pouze mezerou. Čitelnost takového zápisu je ale horší.

```
1 2 10
2 4 8
1 4 12
1 3 8
3 5 10
4 5 6
4 7 6
4 6 9
5 6 12
7 8 11
8 11 12
6 10 10
10 11 10
5 9 11
9 10 6
9 11 9
```

Očividně nemáme prostor, abychom rozebrali všechny možné formáty proto probereme už pouze zajímavý formát *GraphML*, základní dokumentace ostatních formátů s odkazy na podrobnější informace je dostupná v dokumentaci funkce read.graph [20].

Formát GraphML je zajímavý z několika různých pohledů. Jedná se o formát založený na XML (Extensive Markup Language (XML)), proto je do určité míry samodokumentující, jednoduše interpretovatelný a má některé poměrně zajímavé schopnosti.

Naše síť by v GraphML vypadala následovně:

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
     http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
    <key id="d1" for="edge" attr.name="weight" attr.type="double"/>
    <graph id="G" edgedefault="undirected">
        <node id="n1"/>
        <node id="n2"/>
        <node id="n3"/>
        <node id="n4"/>
        <node id="n5"/>
        <node id="n6"/>
        <node id="n7"/>
        <node id="n8"/>
        <node id="n9"/>
        <node id="n10"/>
        <node id="n11"/>
        <edge source="n1" target="n2">
            <data key="d1">10.0</data>
        </edge>
        <edge source="n1" target="n4">
            <data key="d1">12.0</data>
        </edge>
        <edge source="n1" target="n3">
            <data key="d1">8.0</data>
        </edge>
        <edge source="n2" target="n4">
            <data key="d1">8.0</data>
        </edge>
        <edge source="n3" target="n5">
            <data key="d1">10.0</data>
        </edge>
        <edge source="n4" target="n5">
```

```
35          <data key="d1">6.0</data>
36        </edge>
37        <edge source="n4" target="n6">
38          <data key="d1">9.0</data>
39        </edge>
40        <edge source="n5" target="n6">
41          <data key="d1">12.0</data>
42        </edge>
43        <edge source="n4" target="n7">
44          <data key="d1">6.0</data>
45        </edge>
46        <edge source="n6" target="n7">
47          <data key="d1">6.0</data>
48        </edge>
49        <edge source="n7" target="n8">
50          <data key="d1">8.0</data>
51        </edge>
52        <edge source="n8" target="n11">
53          <data key="d1">12.0</data>
54        </edge>
55        <edge source="n6" target="n10">
56          <data key="d1">10.0</data>
57        </edge>
58        <edge source="n10" target="n11">
59          <data key="d1">10.0</data>
60        </edge>
61        <edge source="n5" target="n9">
62          <data key="d1">11.0</data>
63        </edge>
64        <edge source="n9" target="n10">
65          <data key="d1">8.0</data>
66        </edge>
67        <edge source="n9" target="n11"/>
68    </graph>
69 </graphml>
```

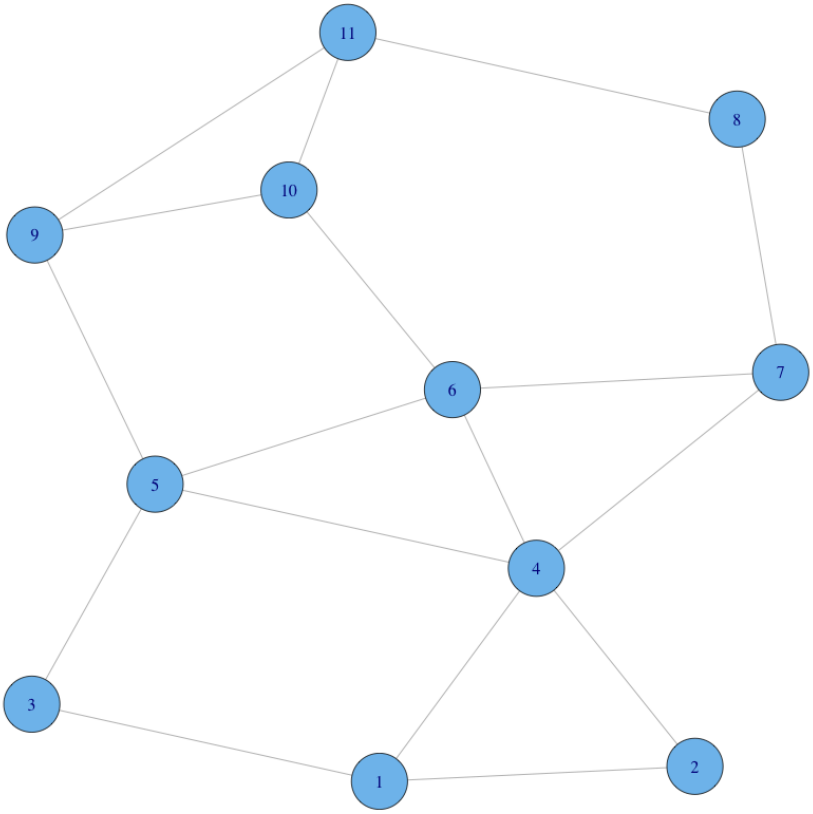
Načtení a vykreslení sítě je pak otázkou načtení definice sítě pomocí funkce read.graph a její vykreslení pomocí funkce plot. Vygenerovaný obrázek sítě ale nebude odpovídat našemu ručně vytvořenému (obr. 7.12), viz obr. 7.13.

V obou případech se jedná o strukturálně totožnou síť. Automaticky generovaná síť ale bere v úvahu lépe délku cest. Výpočet délky nejkratší cesty provedeme voláním funkce shortest.paths. Tato funkce obsahuje tři parametry - síť, která se má analyzovat, počáteční a koncový uzel. Kromě délky, nás může zajímat také cesta samotná - tu získáme použitím funkce get.all.shortest.paths, která má stejné parametry jako shortest.paths, ale vrací posloupnost uzlů odpovídající identifikované nejkratší cestě. Tato funkce zase neposkytuje informaci o celkové délce cesty.

Ford-Fulkersonův algoritmus igraph přímo nepodporuje, obsahuje ale podporu jiných algoritmů pro řešení otázek toku. Např. funkce graph.maxflow je schopna spočítat maximální kapacitu cesty mezi dvěma uzly. Definice sítě, ale musí obsahovat specifikaci kapacity, což v našem případě není splněno.

Celý příklad by mohl vypadat následovně:

```
1 library(igraph)
2 sit <- read.graph("/Users/pavelsenovsky/Documents/skripta/modelovani 3vyd/skripty/sit.graphml", "graphml")
3 plot(sit)
4 shortest.paths(sit, 1, 11)
5 get.all.shortest.paths(sit, 1, 11)
```



Obrázek 7.13: Síť generovaná automaticky na základě GraphML definice pomocí R (balík igraph)

Graphviz
V předchozím vydání skript v této kapitole byl také rozebrán program Graphviz, který slouží pro vizualizace grafů. Vzhledem k možnostem R mi ale výklad software určeného pouze pro vykreslování grafů připadal poněkud nadbytečný. Proto je výklad programu pouze zaveden jako příloha 2 pro zájemce o síťovou problematiku.

Shrnutí
Problematika sítí je poměrně komplexní, z toho důvodu rozlišujeme přísně jaký problém vlastně řešíme - potřebujeme řídit projekt, nebo nás zajímá analýza „fyzické“ sítě, jako je třeba síť silniční nebo nějaký produktovod?

Pro podporu projektového řízení je výhodné použít specializovaný software jako je MS Project, Project Libre a podobné, které jsou schopny pomoci v sledování průběhu projektu, časovém plánování jednotlivých aktivit projektu a také plánování a management zdrojů.

Sítě fyzické je nutné analyzovat pomocí odlišných nástrojů. I v tomto případě často spoléháme na podporu software, v tomto případě se však jedná spíše o obecné programy pro numerické výpočty jako je např. SciLab nebo R.

Z obecného pohledu volíme výpočetní algoritmus podle typu úlohy kterou řešíme. Z nejčastěji používaných algoritmů lze zmínit algoritmůs Dijsktrův pro hledání nejkratší cesty a Ford-Fulkersonův algoritmus pro zjištění maximální kapacity mezi dvěma uzly sítě.

Analyzované sítě mají určitá omezení - především to, že se jedná o sítě uzavřené, a že v síti nedochází ke ztrátám. Principiálně fungují oba algoritmy obdobně - redukují síť do podoby stromu, který je z hlediska výpočtu zvládnutelný.

Příklad
Definujte vlastní síť, vygenerujte ve zvoleném softwarovém nástroji její graf a spočtěte nejkratší cestu mezi dvěma zvolenými uzly.

Kapitola 8

Bilanční modely

Náhled kapitoly
Při rozhodování během krizí by se mělo vycházet z hlubokých, předem připravených znalostí/informací o dané organizaci, firmě. Jednou ze základních funkcí firem je produkce, tedy vytváření produktů za účelem dalšího prodeje.

Bilanční modely, kterými se budeme zabývat v této kapitole, slouží právě k poznání tohoto výrobního řetězce. Umožňuje nám popsat, jak se jednotlivé suroviny transformují v polotovary a ty zas ve finální výrobky.

Informace, které nám bilanční model může poskytnout, pak můžeme využít k hodnocení následků mimořádných událostí z hlediska schopnosti firmy vyrábět.

Po přečtení této kapitoly budete vědět

- co jsou to bilanční modely
- jakým způsobem se bilanční model konstruuje
- jak je možné tento model vyřešit

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně hodinu .

Bilanční modely slouží pro přesný popis návazností výrobních technologií, umožňuje tedy popsat jakým způsobem se transformují vstupy (suroviny, polotovary) na výstupy (výrobky). Bilanční modely se v podnikové ekonomice používají pro plánování výrobních kapacit a surovinových potřeb pro plánovanou produkci finálních výrobků.

„Technicky“ bilanční modely vychází z modelů vyvinutích v šedesátých letech americkým ekonomem ruského původu Vasiliem Leontiefem, které jsou označovány jako modely vstup-výstup (input output). Leontief model používal k tomu, aby popsal sektorové návaznosti v ekonomice - výstup jednoho odvětví (sektoru) je vstupem jiného. V roce 1973 byl za tuto myšlenku Leontief oceněn Nobelovou cenou za ekonomii.

Dnes existuje celá řada aplikovaných tzv. *leontiefovských modelů*. Bilanční modely jsou jejich zástupcem. Obdobné modely lze použít např. pro zkoumání/odhad následků vyřazení klíčových prvků kritické infrastruktury (kritická infrastruktura (KI)). Pro účely modelování v KI je ale nutné zpracování velkého množství informací o ekonomice jednotlivých regionů, což z prostorových důvodů nelze ve skriptech zvládnout. Při výkladu se proto zaměříme na modely bilanční, které pracují „jen“ na úrovni podniku. I v tomto případě však můžeme získat z hlediska bezpečnosti cenné informace.

Například v havarijním plánování můžeme s úspěchem využít pro modelování následku havárií, především jak se projeví vyřazení části infrastruktury podniku na jeho schopnosti udržet výrobu v chodu. Dle namodelovaných výsledků lze naplánovat způsob efektivního zotavení z následků mimořádných událostí. Vztahy mezi surovinami, polotovary a výrobky lze vyjádřit následovně (8.1):

$$x = y + Ax \tag{8.1}$$

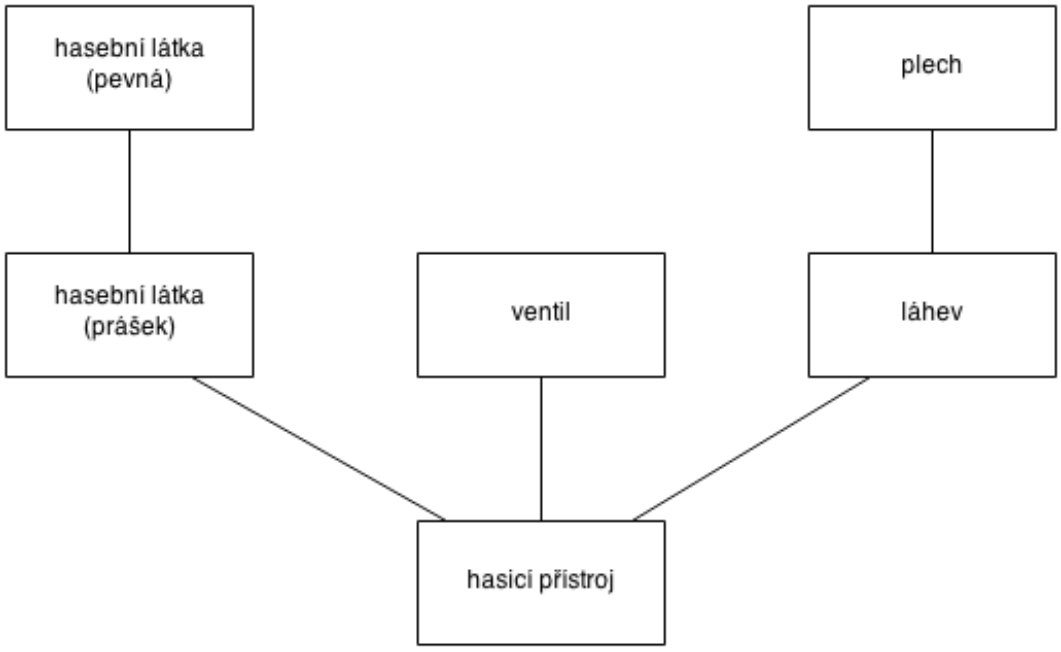
kde
x celková produkce výrobku
y produkce výrobku expedovaná mimo modelovaný systém
Ax vlastní spotřeba uvnitř modelovaného systému
A matice měrných spotřeb polotovarů

Zpracování modelu můžeme demonstrovat na zjednodušeném příkladu výroby hasicího přístroje. Abychom mohli model sestavit, musíme znát jakým způsobem se transformují vstupy výrobního procesu na výstupy. Jinými slovy musíme identifikovat všechny:

- suroviny - čisté vstupy,
- polotovary - výrobky, které se prodávají samostatně, které jsou však zároveň spotřebovávány v rámci výrobního procesu a
- finální výrobky - ty se již nespotřebovávají ve výrobě.

Pokud hovoříme o spotřebě ve výrobním procesu máme tím na mysli spotřebu v modelované části výrobního procesu. Jinými slovy surovinu lze chápat také jako finální výrobek jiného výrobního procesu, který vstupuje z vnějšku na námi modelovaného výrobního procesu a opačně.

Graficky si náš příklad s výrobou hasicího přístroje můžete představit následovně viz obr. 8.1.



Obrázek 8.1: Výroba práškového hasicího přístroje

Model výrobního procesu na obr. 8.1 je očividně zjednodušený, aby bilanční model bylo možné jednoduše demonstrovat.

Rovnice (8.1) je zapsána maticovou formou. Pokud tedy v uvažovaném příkladu z obr. 8.1 označíme:

- x_1 hasební látka (pevná)
- x_2 hasební látka (prášek)
- x_3 ventil
- x_4 plech
- x_5 láhev
- x_6 hasicí přístroj

Abychom, mohli bilanční model vyčíslit, je nutné stanovit měrné spotřeby jednotlivých polotovarů, viz tabulka 8.1.

Tabulka 8.1: Měrné spotřeby polotovarů

Výrobek	**Polotovar**	**Měrná spotřeba**
Hasicí přístroj	Hasební látka (prášek)	10 kg/1ks
	ventil	1 ks/1ks
	láhev	1 ks/1ks
Hasební látka (prášek)	Hasební látka (pevná)	1 kg/kg
Láhev	Plech	0,25 plátu

Vyjádřeno rovnicemi (8.2):

$$\begin{aligned} x_6 &= y_6 + 10x_2 + x_3 + x_5 \\ x_2 &= y_2 + x_1 \\ x_5 &= y_5 + 0,25x_4 \\ x_1 &= y_1 \\ x_3 &= y_3 \\ x_4 &= y_4 \end{aligned} \tag{8.2}$$

Takováto soustava rovnic je dobře řešitelná pokud si uvědomíme, že proměnné y jsou dány nasmlouvaným objemem (objednávkami) a polotovary můžeme kompletně substituovat jejich bilančními rovnicemi. Jednoduché výrobní systémy jsou dokonce pohodlně řešitelné ručně, u složitějších je však preferován maticový zápis a řešení pomocí tabulkového kalkulátoru.

Rovnici (8.1) tak můžeme rozepsat pro náš příklad následovně (8.3):

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,25 & 0 & 0 \\ 0 & 10 & 1 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \tag{8.3}$$

Bilanční model výše, ačkoliv Vám to možná tak nepřipadá, je velmi jednoduchý. V praxi je výrobní proces natolik složitý, že není efektivní řešit jej pomocí jediného modelu - je jej často nutno rozdělit na několik menších. I tyto „menší“ modely obvykle obsahují desítky až stovky proměnných. V našem modelu je jeden finální výrobek, dva meziprodukty a tři suroviny, tedy celkem šest proměnných.

Kromě základního bilančního modelu pro výpočet potřeb výrobků a polotovarů se často používají rozšíření zaměřené na výpočet surovinových potřeb, energetických nároků výroby a výpočet bilancí technologických omezení.

Všechny tyto dodatečné bilance pro výpočet vyžadují vypočtené výrobkové bilance - tedy výpočet bilančního modelu začíná vždy výpočtem 8.1. Surovinový model tak získáme z celkového počtu výrobků (a polotovarů) a měrných spotřeb surovin nutných pro jejich výrobu, viz (8.4).

$$s = Bx \tag{8.4}$$

kde
x vektor celkové produkce daného výrobku (polotovaru)
s vektor celkové potřeby surovin nutných pro výrobu všech výrobků x
B matice technických koeficientů popisujících měrnou spotřebu suroviny nutné pro výrobu produktu x

Podobně můžeme sestavit kapacitní model. Kapacitní model, viz 8.5) opět vychází z množství výrobků a polotovarů. Celková nutná kapacita se spočte z celkového počtu výrobků a kapacit, které jsou nutné pro jejich vyrobení.

$$f = Cx \tag{8.5}$$

kde
x vektor celkové produkce daného výrobku (polotovaru)
f vektor požadavků na kapacity nutné pro výrobu všech výrobků a polotovarů x
C matice koeficientů popisujících potřeby na kapacity daného stroje pro produkci jednotky výrobku x

Jak vidno, model neodpoví na otázku zda kapacity výrobního procesu jsou dostatečné z hlediska potřeb firmy. Výsledky modelů je tedy potřeba interpretovat -získáme pouze informaci o tom, jaké kapacity jsou potřeba a je na nás, abychom určili, zda tyto kapacity máme pokryty nebo ne. Při získávání informací z modelu jsme omezeni pouze naší fantazií. Je možno zpracovávat alternativní scénáře. Výsledky je možno zapracovávat to jiných modelů nebo je používat pro plánování, apod.

Oproti analýzám sítí se také není potřeba využívat specializovaný výpočetní software, postačuje použití běžného tabulkového procesoru.

Příklad
Příklad bilančního modelu je nahrán na http://lms.vsb.cz ve formátu MS Excel. Tento příklad je vypočítán pomocí dvou různých metod, které jsou ale z pohledu výsledků totožné.

Shrnutí
Bilanční modely jsou aplikací modelů vstup-výstup používaných pro modelování vztahů mezi vstupy a výstupy uvnitř výrobního procesu. Bilanční model obsahuje

- výrobkový model,
- surovinový model a
- kapacitní model.

Jako první se vypočítává výrobkový model, ostatní modely se vypočítávají na jeho základě.

Pro účely havarijního plánování je možné tento typ modelů použít pro kvantifikaci důsledků výpadků v části výrobního procesu způsobené nějakou mimořádnou událostí.

Otázky

1. K čemu jsou dobré bilanční modely?
2. Vysvětlete bilanční rovnici výrobků.
3. Můžete specifikovat části bilančního modelu (co může být vypočteno bilančním modelem)?
4. Jak interpretujeme model (jaké informace nám umožňuje získat a jak je můžeme využít)?

Kapitola 9

Lineární programování

Náhled kapitoly
Lineární programování je často používaným nástrojem pro řešení optimalizačních problémů, které jsme schopni omezit pomocí podmínek vyjádřitelných jako rovnice nebo nerovnice.

Po přečtení této kapitoly budete vědět

- Jak vyřešit optimalizační problém pomocí simplexovy metody (primárního algoritmu).

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně 30 minut, pokud si vyzkoušíte příklady vypočítat také prakticky, počítejte však přibližně se 4 hodinami.

Lineární programování řadíme do skupiny algoritmů pro matematické řešení optimalizačních problémů zvané *matematické programování*. Lineární programování nazýváme takovým způsobem proto, že optimalizační problém řešíme prostřednictvím soustavy lineárních rovnic popřípadě nerovnic.

V lineárním programování pracujeme s jednou základní tzv. *účelovou funkcí*, kterou optimalizujeme, tedy obvykle maximalizujeme nebo minimalizujeme a soustavou tzv. *omezujících podmínek*, stanovených jako soustava rovnic nebo nerovnic, pomocí kterých omezujeme prostor možných řešení.

Účelovou funkci můžeme obecně vyjádřit dle rovnice (9.1).

$$optimz = \sum_{j=1}^{n} c_j x_j \tag{9.1}$$

kde
z účelová funkce
optim požadovaná optimalizace max nebo min
c_j ocenění proměnných v účelové funkci (např. cena, variabilní náklady apod.)
x_j optimalizovaná proměnná

Omezující podmínky formulujeme následovně (9.2).

$$\begin{array}{rcll} \sum_{j=1}^{n} a_{ij}x_j & \leq & b_i & i = 1, 2, ..., k \\ \sum_{j=1}^{n} a_{ij}x_j & = & b_i & i = k+1, k+2, ..., k+p \\ \sum_{j=1}^{n} a_{ij}x_j & \geq & b_i & i = k+p+1, k+p+2, ..., k+p+s \\ x_j & \geq & 0 & j = 1, 2, ..., n \end{array} \tag{9.2}$$

kde
a_{ij} technické koeficienty modelu, nastavují se pro každý model samostatně a pak se již nemění
x_j optimalizovaná proměnná
b_i pravé strany omezení, opět se volí v závislosti na modelu (kapacitní omezení apod.)

Pro optimalizované proměnné přitom musí platit, že musí být nezáporné. Tato podmínka vychází z toho, že nelze vyrábět záporný počet výrobků apod. Při připuštění záporných optimalizovaných proměnných bychom získali nestandardní optimalizační problém, který by nebyl řešitelný pomocí lineárního programování (minimálně pomocí metody, kterou budeme popisovat dále).

Nerovnost je menší nebo rovno ($\leq$) popisuje situaci, kdy řešení je problému je omezeno dostupností nějakého zboží, výrobních kapacit, materiálu. Toto omezení musí být předem známo, např. víme že pro výrobu je dostupné jenom XY jednotek materiálu X. Rovnost ($=$) se v omezujících podmínkách používá pro situace zachycující známé výrobní procesy a obdobné situace, kdy je znám určitý potřebný vzájemný poměr např. surovin nutných pro produkci jednotky výrobku.

Konečně nerovnost je větší nebo rovno ($\geq$) v omezujících podmínkách používáme v situaci, kdy do modelu potřebujeme zavést nějaký druh závazku - např. již jsme se zavázali, že dodáme určité množství zboží, toto dohodnuté množství pak při řešení problému musí být vyrobeno.

Jaké problémy pomocí lineárního programování lze řešit? Primárně se tato metoda využívá v ekonomii pro optimalizace výrobního programu (za účelem maximalizace zisku), ale jedná se obecnou metodu, která je využitelná pro celou řadu jiných problémů, např. pro optimalizaci odsávání jednoho zdroje více odsávacími jednotkami, pro optimalizaci přepravy čehokoliv pomocí různých druhů přepravních prostředků apod.

Nyní jak postupovat při formulaci problému a jeho řešení. V prvním kroku je nutné formulovat model v intencích rovnic (9.1) a (9.2), tedy jako soustavu účelové funkce a jejích omezujících podmínek, jako je např. následující:

$$maxz = 2x_1 - 3x_2 + 4x_3 \tag{9.3}$$

$$\begin{array}{rcl} 4x_1 - 3x_2 + x_3 & \leq & 3 \\ x_1 + x_2 + x_3 & \leq & 10 \\ 2x_1 + x_2 - x_3 & \leq & 10 \end{array} \tag{9.4}$$

Takovouto soustavu můžeme řešit pomocí celé řady dostupných algoritmů jako je např.:
- primární algoritmus (simplexova metoda)
- duální algoritmus
- ...

My se budeme dále zabývat pouze simplexovou metodou jako nejznámějším algoritmem pro řešení tohoto typu úloh.

Následující text pro řešení úlohy pomocí simplexovy metody je poněkud náročnější, proto doporučuji, abyste buď absolvovali cvičení prakticky ukazující řešení tohoto problému, nebo alespoň prošli dostupný tutoriál Štefana Wanera [45], který je sice anglicky, ale interaktivně vás krok po kroku vede ke zvládnutí této metody.

Pusťme se tedy do řešení problému pomocí simplexovy metody.

Prvním krokem je převedení omezujících podmínek v nerovnicích do formy rovnic. To převedeme tak, že při znaménku $\leq$ doplníme tzv. doplňovací proměnnou d_i a nerovnici přepíšeme do formy rovnice (nerovnice $\leq$ a $\geq$ jsou mezi sebou formálně převoditelné násobením celé rovnice -1).

Převod na rovnici vychází z prosté úvahy - pokud o levé straně nerovnice víme že je menší než strana pravá musí existovat takové d_i, které tuto nerovnost vyrovná. Hodnotu d_i přitom samozřejmě neznáme. Vyrovnáním nerovností tedy zavádíme do problému další proměnné, které budeme muset taktéž vypočítávat.

Nerovnice uvedené výše by tak do podoby rovnic mohly být převedeny následovně:

$$\begin{array}{rcl} 4x_1 - 3x_2 + x_3 + d_1 & = & 3 \\ x_1 + x_2 + x_3 + d_2 & = & 10 \\ 2x_1 + x_2 - x_3 + d_3 & = & 10 \\ -2x_1 + 3x_2 - 4x_3 + z & = & 0 \end{array} \tag{9.5}$$

Cílem řešení je přidané proměnné d_i minimalizovat ideálně tak, aby byly rovny nule. Může se však stát, že se toto nepovede a tedy doplňkové proměnné vyjdou jako nenulové. V takovém případě je lze interpretovat např. ve smyslu množství zdroje b_i (viz (9.1)), které nebude využito, spotřebováno.

Pro ruční výpočet vytvoříme počáteční pracovní tabulku 9.1.

Tabulka 9.1: Počáteční pracovní tabulka

x_1	x_2	x_3	d_1	d_2	d_3	z	
4	-3	1	1	0	0	0	3
1	1	1	0	1	0	0	10
2	1	-1	0	0	1	0	10
-2	3	-4	0	0	0	1	0

S každou tabulkou je asociováno právě jedno tzv. *bazické řešení*. Jedná se o jedno z nekonečně mnoha řešení, které splňuje všechny omezující podmínky. Bazické řešení samo o sobě nemusí být řešením finálním.

Abychom našli bazické řešení hledáme tzv. vyčištěné sloupce tedy sloupce, ve kterých je nenulová hodnota pouze na jediném řádku. Proměnným v těchto sloupcích říkáme, že jsou aktivní, zatímco ostatní jsou neaktivní. Bazické řešení naší tabulky 9.1 by mohlo být následující.

Tabulka 9.2: Počáteční pracovní tabulka – bazické řešení

x_1	x_2	x_3	d_1	d_2	d_3	z		**Bazické řešení**
4	-3	1	**1**	0	0	0	3	$d_1 = 3/1 = 3$
1	1	1	0	**1**	0	0	10	$d_2 = 10/1 = 10$
2	1	-1	0	0	**1**	0	10	$d_3 = 10/1 = 10$
-2	3	-4	0	0	0	1	0	$z = 0/1 = 0$

Jak je patrné z tabulky 9.2, pouze proměnné d_1, d_2, d_3 a z jsou aktivní, všechny ostatní aktivní nejsou (x_1, x_2 a x_3) a proto je položíme rovny nule. Aktivní proměnné nám poslouží pro formulaci bazického řešení. Např. hodnotu proměnné d_1 v bazickém řešení spočteme dělením pravé strany rovnice hodnotou ve sloupci d_1, takže $d_1 = 3/1 = 3$.

Bazické řešení je tedy:

$$\begin{aligned} d_1 &= 3 \\ d_2, d_3 &= 10 \\ x_1, x_2, x_3, z &= 0 \end{aligned} \tag{9.6}$$

Takto získané řešení je však očividně daleko od hledaného optima. Všechny naše optimalizované proměnné x jsou rovny nule, stejně jako hodnota účelové funkce! To je dáno tím, že simplexova metoda je metodou iterační. Řešení tedy dostáváme v jednotlivých iteracích - krocích, přičemž s každým krokem je spojeno právě jedno bazické řešení. S každou iterací se také stále více blížíme k optimálnímu řešení problému.

Řešení je tedy nutné dále upravovat. Úpravu provedeme podle zvoleného středu. Střed přitom určujeme na základě několika pravidel:

1. výběr sloupce (ve kterém bude střed) – vybíráme vždy sloupec proměnné, ve kterém je jevíce záporné číslo účelové funkce (poslední řádek tabulky) – v našem případě je to číslo -4 a tedy sloupec x_3.
2. Střed musí být vždy kladné číslo, které má největší testovací poměr. Testovací poměr spočteme jako poměr pravé strany rovnice a proměnné daného sloupce na daném řádku. Podívejme se na výpočet v tabulce 9.3.

Hledaný střed je na průsečíku vybraného sloupce a řádku.

Tabulku upravujeme kromě řádku středu tak, aby se sloupec středu vyčistil (viz tab. 9.4)

Protože v posledním řádku se již nevyskytují žádná záporná čísla, dospěli jsme k optimálnímu řešení. V opačném případě bychom postup s hledáním středu a úpravami tabulky museli opakovat.

Tabulka 9.3: výpočet středu - testovací poměr

x_1	x_2	x_3	d_1	d_2	d_3	z		**testovací poměr**
4	-3	**1**	1	0	0	0	3	$3/1 = 3$
1	**1**	***1***	**0**	**1**	**0**	**0**	**10**	$\mathbf{10/1 = 10}$
2	1	**-1**	0	0	1	0	10	
-2	3	**-4**	0	0	0	1	0	

Tabulka 9.4: Iterace 2

x_1	x_2	x_3	d_1	d_2	d_3	z		**úprava**
3	-4	0	1	-1	0	0	-7	- řádek 2
1	1	1	0	1	0	0	10	
3	2	0	0	1	1	0	20	+ řádek 2
2	7	0	0	4	0	1	40	+ 4 * řádek 2

Výsledné řešení:

$$\begin{aligned} x_1, x_2 &= 0 \\ x_3 &= 10 \\ d_1 &= -7 \\ d_3 &= 20 \\ d_4 &= 40 \end{aligned} \tag{9.7}$$

Samozřejmě podobně jako v předchozích případech, optimalizační problémy neřešíme ručně, ale využíváme softwarovou podporu pro řešení. Z velkých programových balíku, které zvládají řešit problémy lineárního programování můžeme zmínit Mathematicu, Mathlab nebo open sourcový SciLab. Existuje také celá řada menších utilit, které můžete s úspěchem použít pro řešení některých jednodušších úkolů (na úrovni úkolů, které budou následovat dále). Jednou z nich je například Finite mathematics utility: simplex method tool [6].

Problém zde definujeme intuitivně do textového pole a výpočet je proveden pomocí JavaScriptového enginu přímo na Vašem PC. Rozhraní vypadá podobně jako na obr. 9.1.

Do horní části okna vypisujeme účelovou funkci, přitom stanovujeme, zda ji chceme maximalizovat nebo minimalizovat a podrobujeme ji (subject to) omezením. Řešení se spustí kliknutím na tlačítko „Solve“. Jednotlivé iterace řešení se vypíší ve spodní části obrazovky a optimální řešení se zapíše do samostatného řádku řešení (Solution).

Co obecné programy pro numerické výpočty jako je SciLab nebo R, mohou nám pomoci? Odpověď je samozřejmě ano. SciLab má pro lineární programování implementován algoritmus karmakar, který byl vyvinut v roce 1967 Dikinem a v roce 1986 pak znovuobjeven Barnesem a kol. (viz [10]). Pro řešení problémů lineárního programování pro SciLab existuje také toolkit Quapro, viz [18].

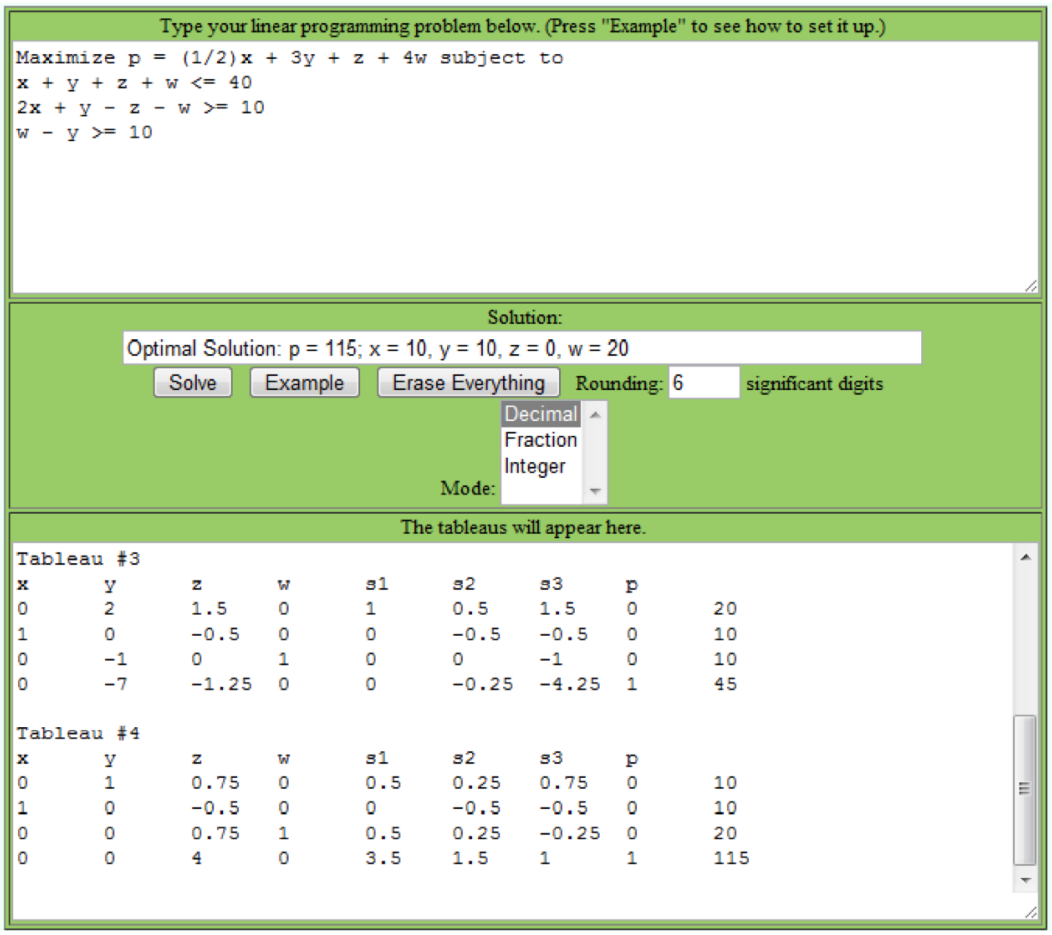
Zkusme vypočítat příklad z [10] s použitím algortimu karmarkar a pro srovnání také Simplex Method Tool [6].

$$\begin{aligned} minimizez &= -x_1 - x_2 \\ x_1 - x_2 &= 0 \\ x_1 + x_2 + x_3 &= 2 \\ x_{1,2,3} &\geq 0 \end{aligned} \tag{9.8}$$

Simplex Method Tool říká, že výsledek účelové funkce je -2 s $x_{1,2} = 1$.

Skript pro SciLab pro algoritmus karmakar by vypadal následovně (převzato z [10]):

```
Aeq = [
   1 -1 0
   1 1 1
];
beq = [0;2];
c = [-1;-1;0];
[xopt,fopt,exitflag,iter,yopt]=karmarkar(Aeq,beq,c)
```



Obrázek 9.1: Simplex tool [6]

Jak máme tento skript interpretovat? Proměnná *Aeq* obsahuje matici koeficientů levých stran omezení. Jednotlivé řádky matice odpovídají jednotlivým omezením. Máme tedy dvě omezení, což odpovídá dvěma řádkům matice *Aeq*.

Proměnná *beq* je sloupcovým vektorem pravých stran omezení. Konečně c je sloupcový vektor koeficientů účelové funkce.

Funkce karmakar využívá bere parametry *Aeq*, *beq* a c a na základě nich vrací celkem pět výsledků formou řádkového vektoru. Poslední řádek skriptu vrací výsledkem, který je bez zaokrouhlení $z = -1,999989$ pro účelovou funkce a proměnné $x_{1,2} = 0,9999948$ a $x_3 = 0,00001$. Po zaokrohlení tak dostaneme stejný výsledek jako při užití Simplex Method Tool.

Na první pohled je použití Simplex Method Tool jednodušší než použití nástrojů jako je SciLab. Jednoduchost užití ale přichází na úkor univerzálnosti řešení. SciLab je proto schopen řešit mnohem složitější problémy a nikoliv pouze pomocí jediného algoritmu. SciLab je také podstatně lepší z hlediska možnosti načítání dat z datových souborů a také z hlediska práce s výsledky.

Příklad

Dva studenti Anna a Karel pracují x a y hodin týdně. Dohromady mohou pracovat maximálně 40 hodin týdně. Podle pravidel pro brigádníky může Anna pracovat maximálně o 8 hodin více než Karel, ale Karel může pracovat maximálně o 6 hodin déle než Anna. Dodatečné omezení je $18 \leq 2y + x$. Najděte optimální řešení, pro:

1. Pokud Anna dostane 15 $ a Karel 17 $ za hodinu, nalezněte maximální kombinovaný příjem
2. Pokud Anna dostane 17 $ a Karel 15 $ za hodinu, nalezněte maximální kombinovaný příjem
3. Pokud Anna i Karel dostanou 16 $ za hodinu, nalezněte maximální kombinovaný příjem

Příklad
Společnost vyrábí dva výrobky (X a Y) a k tomu používá dva stroje (A a B). Pro výrobu každé jednotky produkce výrobku X je nutné 50 minut času stroje A a 30 minut času na stroji B. Každá jednotka výrobku Y vyžaduje 24 minut na stroji A a 33 minut na stroji B.

Na začátku týdne je v zásobě 30 jednotek X a 90 jednotek Y. Dostupný čas na stroji A se odhaduje na 40 hodin a na stroji B na 35 hodin.

Poptávka po výrobku X na tento týden se odhaduje na 75 jednotek a pro Y na 95 jednotek. Podniková politika říká, že se maximalizuje kombinmovaná suma jednotek X a Y v zásobě na konci týdne.

- formulujte problém pro rozhodnutí o tom kolik každého produktu vyrobit tento týden

Příklad
Společnost produkuje 2 výrobky (X a Y). K výrobě jsou používány dva zdroje - stroje, pro automatizovanou výrobu, a dělníci pro dodělávací práce.

Stanovení podrobností o produkčních potřebách pro jednotlivé výrobky naleznete v tabulce 9.5.

Společnost pro výrobu může získat pro následující týden 40 hodin času na stroji, ale pouze 35 hodin dělnické práce. Náklady na práci stroje jsou 10 $ za hodinu a dělníků 2 $ za hodinu. Čas bez práce (kdy nevyrábějí) se dělníkům neproplácí. Cena výrobku X je 20 $ a Y je 30 $. Společnost má kontrakt pro výrobu 10-ti výrobků X. Formulujte (a vyřešte) problém lineárního programování.

Příklad
Květinářství Azalka dostalo zakázku vyrobit svatební květinovou výzdobu. Květiny budou dvou druhů, přičemž celkem jich má být maximálně 30 ks. Výroba první květiny trvá 20 min. a květinářství přináší zisk 56 Kč (účtovaná cena 145 Kč), výroba druhé květiny trvá 15 min. a přináší květinářství zisk 43 Kč (účtovaná cena 130 Kč).

Náklady na výzdobu přitom nesmějí přesáhnout 4100 Kč a příprava nesmí trvat déle než 9 hod. Květinářství si klade otázku, jaká zvolit poměr vázaných květin, aby vzhledem k délce trvání uvázání maximalizovalo svůj zisk.

Květinářství má pro oba druhy květin dostatek zdrojů a květinářky umí stejně dobře vázat oba druhy květin.

Příklad
Výrobce čajů má k dispozici 3 kg sušené máty a 1,5 kg sušené třezalky. Má možnost vyrábět dva druhy bylinných čajů, a to čistý mátový čaj nebo směs máty a třezalky v poměru 3:2, byliny jsou plněny do nálevových sáčků po 10 g. Při výrobě je nutné počítat s odparem u máty 5 % a u třezalky 8 %. Čistého mátového čaje se prodá maximálně 100 sáčků. Zisk u jednoho sáčku čisté máty je 2 Kč a z jednoho sáčku směsi 3 Kč. Kolik sáčků každého druhu má výrobce vyrobit, aby zisk byl maximální.

Disponibilní množství čistých bylin je:

- máta ... 3000 g
- třezalka ... 1500 g

Tabulka 9.5: Produkční potřeby pro jednotlivé výrobky pro *příklad 2*

	Stroj	**Dělník**
výrobek X	13	20
výrobek Y	19	29

Shrnutí
Lineární programování slouží pro řešení optimalizačních problémů vyjádřených pomocí optimalizované *účelové funkce* a jejich *omezujících podmínek* vyjádřených soustavou rovnic a nerovnic.

Nejčastěji řešíme problém lineárního programování použitím tzv. *primárního algoritmu* známého též pod názvem *simplexova metoda*. Jedná se o iterační algoritmus, kdy s každou iterací je spojeno jedno tzv. *bazické řešení*. S každou iterací se řešení zpřesňuje.

Pro výpočet samotný obvykle používáme podporu software.

Kapitola 10

Základy teorie her

Náhled kapitoly
V situacích, kdy výsledek rozhodovací situace je přímo ovlivněn rozhodnutími dalších lidí, kteří působí proti našim zájmům, techniky, které jsme zatím probrali, příliš nepomohou. Právě v těchto situacích se používá teorie her.

Po přečtení této kapitoly budete vědět
vědět

- co je to teorie her a k čemu slouží
- jak se řeší jednoduché problémy teorie her

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně 45 minut.

Rozhodování, respektive přípravu pro rozhodnutí jsme zatím vždy konstruovali na základě objektivních, poznatelných charakteristik, kdy proti rozhodnutí nestál protivník, který by si ze všeho nejvíc přál hatit Vaše plány. Jinými slovy nebylo rozhodování činěno proti živému člověku, který má vlastní zájmy, které se také samozřejmě snaží maximalizovat.

Teorií, která popisuje právě tyto rozhodovací situace je *teorie her*. Rozhodovací situace je v této teorii vnímána jako hra, do které vstupují jednotliví účastníci rozhodování – hráči. Pro demonstraci východisek této teorie je používán často následující příklad, tzv. *dilema vězně*.

Dilema vězně
Ve věznici sedí na samotce dva obžalovaní, čekají na soud a rozhodují se o tom, co vypoví. Pokud vězeň A obviní vězně B a vězeň B bude mlčet pak vězeň A vyvázne a vězeň B dostane přísný trest. Pokud se vězni A a B budou vzájemně obviňovat, oba dostanou přísný trest. Pokud oba vězni budou mlčet, dostanou oba mírný trest, protože dokazování bude velmi obtížné. Pokud vězeň B obviní vězně A a vězeň A bude mlčet, pak vězeň B vyvázne a vězeň A dostane přísný trest.

Uvedenou situaci lze zapsat i tabulkovou formou (viz tab. 10.1).

Obdobným způsobem bychom mohli formulovat tabulku pro vězně B, pro naše účely to ale není nutné.

Cílem obou vězňů je maximalizovat svůj, vlastní užitek v situaci, kdy nemají informaci o tom, kterou strategii zvolí jejich „oponent“. Oba vězni tedy provádějí volbu na základě toho, co si myslí, že udělá jejich spoluvězeň. Takové rozhodnutí není úplně racionální.

Cílem teorie her je pomoci nám pochopit tyto situace a kultivovat tak „hru“ ve smyslu odklonu od snahy maximalizovat vlastní prospěch bez ohledu na ostatní hráče, což však často vede k cíli, k

Tabulka 10.1: Následky volby strategií vězeň A

		vězeň A	
		mlčet	**mluvit**
vězeň B	**mlčet**	běžný trest	mírný trest
	mluvit	tvrdý trest	tvrdý trest

optimalizaci užitku - tedy volbě optimální strategie.

Kdybychom slovní hodnocení převedli do číselné podoby, pak by tato čísla tvořila matici výher A o které bychom mohli napsat (10.1)

$$\max_i \min_j a_{ij} \leq \min_i \max_j a_{ij} \tag{10.1}$$

Gros píše o optimální strategii následující [31] ... *největší z minimálním výher, které se může první hráč zajistit, nemůže překročit největší výhru, kterou chce druhý hráč minimalizovat.*

V situaci, kdy existuje bod, který reprezentuje rovnost obou stran nerovnice (10.1), říkáme, že matice výher A má tzv. *sedlový bod*. Tento sedlový bod reprezentuje optimální strategii, kterou bych měl volit s ohledem na své vyhlídky a vyhlídky soupeře.

Zkusme demonstrovat tento přístup na našem příkladě. Ohodnoťme bodově možné výsledky strategií: mírný trest 50b, svoboda 100b, přísný trest 0b.

Tabulka 10.2: Hledání sedlového bodu

		vezeň A		min_j
		mlčet	**mluvit**	
vezeň B	**mlčet**	**50**	100	50
	mluvit	0	0	0
max_i		50	100	

Sedlový bod, který jsme takto zjistili, doporučuje vězni A mlčet. Pokud totiž vězeň B bude znát teorii her, bude volit právě tuto strategii, ze které budou profitovat oba. Tento výsledek však předpokládá racionální chování obou zúčastněných.

Takto jednoduše řešitelným hrám říkáme *hry dvou hráčů v normálním tvaru*. Bohužel ne všechny úlohy mají sedlový bod, v případě, že jej nemají, pak hovoříme o *smíšeném rozšíření hry dvou hráčů*. V tomto případě nehledáme jedinou, optimální strategii, ale **pravděpodobnosti, se kterými mají hráči střídat strategie tak, aby v průměru dosáhli maximální možné výhry**. Hledají se tedy tzv. *smíšené strategie*. Je třeba podotknout, že každou hru, vyjádřitelnou maticovou formou (o kterých se tady bavíme) má řešení ve smíšených strategiích.

Smíšené hry předpokládají, že rozhodovací situace se opakují, tedy že dochází k opakovanému střetu hráčů a opakovanému volení strategií. Teprve v takovém případě mají smíšené strategie smysl.

Hru pak převádíme do formy řešitelné pomocí metod lineárního programování.

Pokud by všechny rozhodovací situace měly sedlový bod, přínosy teorie her by i pro nás byly jasně patrné. Pro reálně situace, kdy proti nám působí větší množství hráčů majících k dispozici velké množství strategií představa, že nasadíme teorii her je ne příliš reálná. *Takže proč se tady vlastně teorií her vůbec zabýváme*?

Zjednodušeně bychom to mohli shrnout: *abychom změnili způsob Vás studentů, jakým vnímáte rozhodovací situace*. Abychom to udělali podíváme se na některé dobře dokumentované hry se známým výsledkem. Existuje celá řada modelových situací, ale v těchto skriptech se zaměříme pouze na dvě z nich:

1. hra na kuře (chicken game) a
2. hon na vysokou (stag hunt).

Hra na kuře v teorii her je založena na velmi nebezpečné hře hrané „naživo“ zejména velmi neodpovědnou mládeží. Hra zahrnuje dvě silná auta, jedou proti sobě, kdo první strhne řízení prohrál. Ovšem pokud žádný z hráčů nestrhne řízení dojde k čelní srážce obou aut a nejspíše i úmrtí obou hráčů.

V modelové hře mají oba hráči stejné auto, jsou stejně dobří řidiči a oba chtějí vyhrát závod. Zkusme kvantifikovat následky volby strategií.

- vyhrát závod = 10
- prohrát závod = -10
- bez vítězů = 1
- havárie = -100

S omezenými strategiemi: strhnout řízení nebo jet dále můžeme vytvořit matici výsledků viz tab. 10.3. Jelikož pracujeme pouze se dvěma hráči zaznamenám výsledky obou hráčů do jediné tabulky. Pohledy obou hráčů jsou odděleny čárkou. Pohled na výsledky obou hráčů zároveň by nám měl usnadnit pochopení celé situace.

Tabulka 10.3: Hra na kuře

		závodník A	
		strhnout řízení	**závodit**
závodník B	**strhnout řízení**	1, 1	-10, 10
	závodit	10, -10	-100, -100

Je jasné, že oba hráči vyhrát nemohou, ale oba mohou zahynout, což je něco, čemu bychom se za normálních okolností rádi vyhnuli. Optimální strategie tedy spočívá v tom donutit soupeře aby to hru vzdal. Bohužel obvykle hráč nemůže garantovat, že bude schopen donutit soupeře uhnout, takže optimální strategie maximalizující užitek neexistuje.

Za těchto podmínek by bylo pro oba hráče logickou volbou strhnout řízení. Také existuje alternativní řešení označované jako symetrická Nashova rovnováha - za předpokladu, že oba hráči volí strategii náhodně. V této hře jsou následky rozhodnutí poměrně extrémní, typově ale existují i jiné příklady, které řadíme ke hře na kuře, ale následky tak extrémní nejsou.

Např. vyjednávání odborů. Pokud odbory i zaměstnavatel tlačí na změnu platu (odbory nahoru a zaměstnavatel dolů) - dohoda nebude možná a zaměstnanci půjdou do stávky. Stávka v tomto případě je řešení bolestné pro zaměstnavatele - nepracuje se i pro zaměstnance - během stávky totiž nedostává od zaměstnavatele mzdu. Pokud tato situace bude trvat dlouho může skončit až krachem podniku a propuštěním všech zaměstnanců - všichni prohrají.

Pokud ale obě strany ustoupí ze svých požadavků - zaměstnanci získají něco málo k platu a zaměstnavateli se o něco málo zvednou náklady na pracovní sílu. Ostatní možnosti vedou k vítěství jedné strany a prohře strany druhé (management nebo odbory), která pro druhou stranu obvykle není přijatelná.

Jiným typem hry je **hon na vysokou**. Tato hra je odvozena z hypotetické situace honu na vysokou, kdy lovec na posedu čeká až se ukáže jeho kořist. Ta se ale ne a ne ukázat. Pokud bude lov úspěšný, všichni lovci se z úlovku nasytí, jenomže není zaručeno, že se vysoká vůbec ukáže. Pak se ale objeví králík. Pokud jej lovec střelí, bude pouze jeho, ale ostatní lovci půjdou domů s prázdnou. Takže co má lovec dělat - být sobec a střelit králíka, nebo kooperovat s ostatními lovci a mít šanci na velký úlovek?

Zformulujme matici výsledků. Zjednodušíme si situaci, takže budeme uvažovat pouze dva lovce. V matici opět zohledníme pohledy obou lovců, viz tab. 10.4.

Tabulka 10.4: Lov na vysokou

		lovec A	
		spolupracovat	**zradit**
lovec B	**spolupracovat**	2, 2	0, 3
	zradit	3, 0	, 1

Pokud lovci budou spolupracovat, oba se nají, pokud jeden z nich zradí (a střelí králíka), tak ten, co zradil získá všechny benefity a druhý lovec odejde s prázdnou. Pokud se oba rozhodnou zradit, oba získají určité benefity, ale menší než pokud by spolupracovali.

Situaci honu na jelena v reálu rozumíme takovou situaci, kdy dva nebo více hráčů musí kooperovat, aby dosáhli společného cíle. Tento cíl přitom není možné dosáhnout „samostatně“. Např. ochrana kriticky ohroženého druhu, který se vyskytuje na hranici mezi dvěma státy. Státy se zavážou, že podniknou akce na záchranu tohoto druhu - pokud svůj závazek oba státy splní, bude druh zachráněn.

Pokud však jeden z nich podporu projektu záchrany druhu stáhne podporu, bude inkasovat určité benefity, např. bude moci vynaložit finanční prostředky jiným způsobem. Druhý stát bude ale v projektu dál utápět peníze, aniž by měl reálnou šanci dosáhnout požadovaného výsledku.

Podobným způsobem, jako v případě honu na vysokou nebo hry na kuře je popsána celá řada dalších her. Pokud Vás tato problematika zajímá je dobrým výchozím bodem článek seznamu her [12].

Co myslíte?
Jaký typ hry by vystihoval situaci finanční krize mnoha zemí Eurozóny, které jsme byly svědky mezi léty 2008 - 2012, a zavedením Evropského stabilizačního mechanismu? Jedná se spíše o hon na vysokou nebo hru na kuře? Jaké jsou nejdůležitější momenty, které Vás k rozhodnutí vedly?

Tato otázka nemá jediné „správné“ řešení/odpověď, takže můžete popustit uzdu své fantazii

Otázky

1. Jaký je rozdílů mezi hrou typu hon na vysokou a hrou na kuře?
2. Co je to sedlový bod a jak jej najdeme?
3. Mají všechny hry sedlový bod a pokud ne, co je řešením takových her?
4. Popište tzv. vězňovo dilema.

Kapitola 11

Lokalizační modely

Náhled kapitoly
Dalším z optimalizačních problémů je problém fyzické lokalizace objektů v prostoru.

Po přečtení této kapitoly budete vědět

- co jsou lokalizační modely
- jak řešit jednoduché lokalizační problémy
- jak přistoupit k řešení složitějších problémů

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně 30 minut.

Lokalizační modely existují, aby nám pomohly řešit problémy vyžadující umístění nějakého objektu v prostoru. Umisťování obvykle neprobíhá náhodně, ale je prováděno tak, aby byla splněna určitá optimalizační kritéria obvykle odvozená z efektivity např. investice. Typickým problémem pro lokalizační modely by mohl být např. *máme vymezené území, které je nutné pokrýt mobilním signálem, kam a kolik stanic BTS (Base Transceiver Station (BTS))je potřeba umístit tak, aby byla nutná finanční investice co možná nejmenší*?

Způsob řešení problému se odvíjí od definice problému - umisťujeme jediný objekt (přidáváme další uzel do existující sítě) nebo je potřeba jich umístit více? Liší se také řešení podle toho, zda je možné pracovat ve 2D nebo je nutné zohlednit 3D charakter prostoru. Můžeme být také omezeni místy, kam lze umístit objekt. Konečně víme kolik objektů má být v prostoru umístěno, nebo je jejich počet taktéž předmětem optimalizace?

O jakých problémech se tady bavíme:

1. Společnost již má v provozu tři výrobny, generální ředitel společnosti (Chief Executive Officer (CEO)) požaduje, abychom nalezli vhodné místo pro vybudování skladu, kam budou sváženy výrobky jednotlivých továren k dočasnému uskladnění.
2. Po posledních povodních v oblasti, vláda rozhodla o nutnosti lepšího pokrytí území prvky jednotného systému varování a vyrozumění (jednotný systém varovaní a vyrozumění (JSVV)). Naším úkolem je najít vhodná místa pro umístění prvků JSVV. Navrhovaná investice by taktéž měla být efektivní z hlediska nákladů, dosaženo by tedy mělo být co možná největšího pokrytí s co možná nejmenšími náklady.
3. ...

Pro nalezení řešení je nutné specifikovat optimalizační kritérium. Takové kritérium obvykle souvisí

s náklady na spojení mezi existujícími a nově umisťovanými objekty. Tyto náklady N_{ij} je nutno vypočítat pro všechna spojení mezi objekty i a j.

Náklady samotné jsou obvykle funkcí přepravní vzdálenosti d_{ij}. Za předpokladu, že přeprava např. zboží na delší vzdálenosti stojí víc než přeprava stejného zboží na vzdálenost kratší. Množství přepraveného x_{ij} zboží je také důležité z hlediska odhadu nákladů. Třetí a poslední hodnotou, která ovlivňuje náklady je typ přepravního prostředku, který je používán pro transport - c_{ij}. S různými typy přepravních prostředků jsou spojeny odlišné náklady na použití.

Matematicky můžeme vyjádřit vztah mezi náklady přepravy z a do existujících objektů a objektů nově umístěných pomocí rovnice (11.1).

$$N_{ij} = x_{ij} d_{ij} c_{ij} = d_{ij} w_{ij} \tag{11.1}$$

Jelikož hodnoty x a c jsou z pohledu výpočtu konstantami, můžeme je nahradit hodnotou váhy daného spojení w. Optimalizace se tedy bude týkat pouze přepravní vzdálenosti d.

Spojení novými objekty (pokud potřebujeme umístit více než jeden) může být vyjádřena pomocí rovnice (11.2).

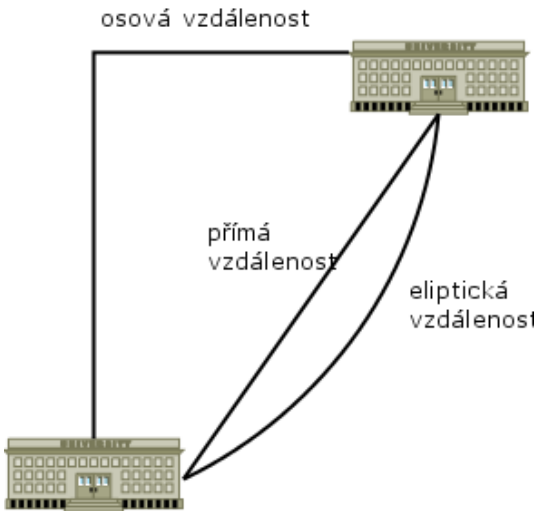
$$N_{ik} = x_{ik} d_{ik} c_{ik} = d_{ik} w_{ik} \tag{11.2}$$

Nákladová funkce pak může být vyjádřena jako suma nákladů, viz (11.3).

$$minz = \sum_{i=1}^{m} \sum_{j=1}^{n} N_{ij} + \sum_{i=1}^{n} \sum_{k=1}^{n} N_{ik} \tag{11.3}$$

Pokud umisťujeme pouze jeden objekt, celkový součet nákladů N_{ik} je roven nule, což nám výrazně zjednodušuje výpočet.

Jak jsme si řekli výše - kritickým bodem celé optimalizace je vzdálenost, jenomže jaká vzdálenost? Existují tři základní typy vzdáleností, se kterými můžeme počítat a které se výrazně liší svými vlastnostmi a způsobem výpočtu. Různé typy vzdáleností jsou demonstrovány na obr. 11.1.



Obrázek 11.1: Tři základní typy vzdáleností

Matematicky lze vzdálenosti vyjádřit pomocí rovnic (11.4 – 11.7).

Když si představíme vzdálenost, aniž bychom nad tím příliš přemýšleli, bude to nejspíše *vzdálenost přímá*. V lokalizačních modelech tento typ vzdáleností ve skutečnosti ale není až tak úplně běžný, pokud přeprava má probíhat po zemi. Např. terén je zvlněný, je nutné obcházet některé překážky apod. Možná, pokud bychom pracovali na „makro“ měřítku tyto odchylky by bylo možné zanedbat a vzdálenost považovat za přímou.

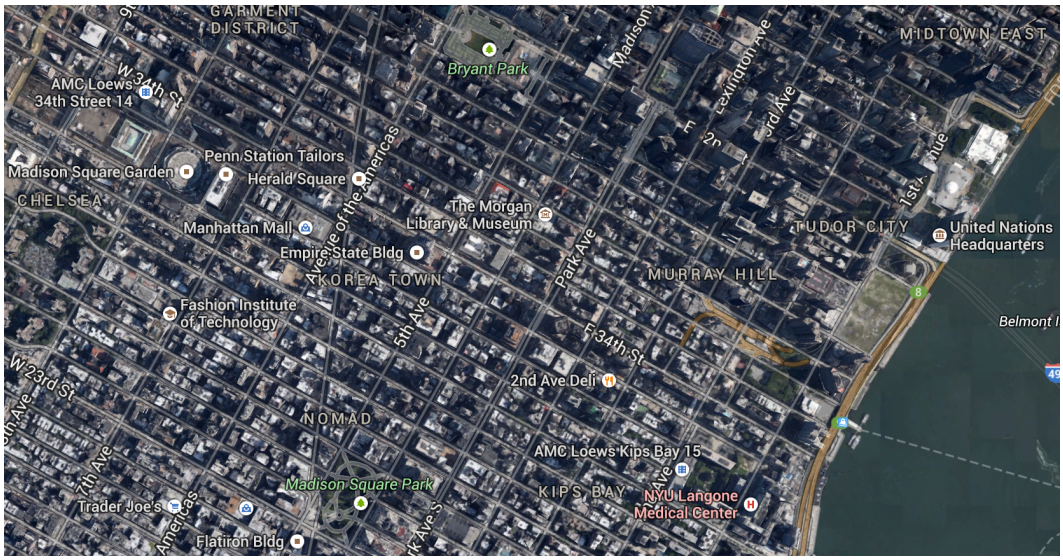
Použití přímé vzdálenosti je běžné pro bezdrátovou komunikace, kde nejsme limitování členitostí terénu.

Rovnice pro výpočet přímé vzdálenosti je notoricky známa, viz rovnice (11.4). K přímé vzdálenosti lze také přidat konstantu k, pomocí které lze započítat určité projektované prodloužení cest vyrovnávající drobnější nerovnosti terénu, kroutící se cestu apod., viz (11.5).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{11.4}$$

$$d_{ij} = k\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{11.5}$$

Osová vzdálenost může být použita v situaci, kdy se pracuje se systémem cest. To je typické pro silniční sítě v novějších čtvrtích měst. Podívejte se např. na silniční síť New Yorku, viz obr. 11.2. Taková vzdálenost může být vypočítána pomocí rovnice (11.6).



Obrázek 11.2: Silniční sít New Yorku (*zdroj: Google Maps*)

$$d_{ij} = |x_i - x_j| + |y_I - y_j| \tag{11.6}$$

Konečně můžeme použít kvadratickou vzdálenost, kterou vypočítáme pomocí rovnice (11.7).

$$d_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2 \tag{11.7}$$

Nejjednodušším příkladem, na kterém lze demonstrovat řešení lokalizačního problému je lokalizace jediného objektu v 2D s použití osové vzdálenosti. V takovém případě bychom optimalizovali vzdálenosti vyjádřené rovnicí (11.8).

$$minz = \sum_{j=1}^{n} |x - x_j| + |y - y_j| \tag{11.8}$$

Suma součtů může být jednoduše vyjádřena jako součet sum, což nám umožňuje oddělit koordináty x a y, viz (11.9).

$$minz = \sum_{j=1}^{n} w_j|x - x_j| + \sum_{j=1}^{n} w_j|y - y_j| \tag{11.9}$$

Geometricky by řešení mělo být ve středu. Jelikož pracujeme s existujícím systémem silniční sítě, hledáme takové koordináty, pro které suma vah w napravo a nalevo (pro x-ovou souřadnici) a níže a výše (pro y-ovou souřadnici) jsou přibližně stejné.

Toho lze jednoduše dosáhnout v tabulární formě. Pokusme se najít optimální umístění nového objektu, tak aby efektivně komunikoval s existujícími objekty A-D, viz tab. 11.1 pro souřadnice objektu a váhy s nimi spojené.

Tabulka 11.1: Lokalizace existujících objektů a určení vah (převzato z [31])

Objekt	**x**	**y**	**w**
A	1	8	50
B	4	2	100
C	9	2	80
D	8	5	70

Tabulka 11.2: Lokalizace existujících objektů - seřazeno podle X a Y (převzato z [31])

seřazeno podle X				
objekt	**x**	**y**	**w**	$\sum w$
A	1	8	50	50
B	4	2	100	150
D	8	5	70	**220**
C	9	2	80	300
seřazeno podle Y				
objekt	**x**	**y**	**w**	$\sum w$
B	4	2	100	100
C	9	2	80	**180**
D	8	5	70	250
A	1	8	50	300

Nyní, je nutné pro provedení výpočtu seřadit objekty podle souřadnic X a Y a vybrat takovou souřadnici, která přesahuje polovinu celkového součtu jejich vah, viz tab. 11.2.

Za daný okolností by proto nový objekt měl být umístě na souřadnicích X = 8 a Y = 2.

Při použití kvadratické vzdálenosti účelová funkce, kterou budeme optimalizovat vypadá následovně (11.10).

$$minz = \sum_{j=1}^{n} w_j((x - x_j)^2 + (y - y_j)^2) \tag{11.10}$$

Lokální minima funkce získáme parciální derivací rovnice (11.10) podle jednotlivých proměnných souřadnic, viz (11.11) a položení je rovny nule. Rovnice (11.11) má analytické řešení (11.12), které je možno jednoduše spočítat pomocí jakéhokoliv tabulkového procesoru.

$$\begin{aligned} \frac{\partial z}{\partial x} &= -2\textstyle\sum_{j=1}^{n} w_j(x - x_j) = 0 \\ \frac{\partial z}{\partial y} &= -2\textstyle\sum_{j=1}^{n} w_j(y - y_j) = 0 \end{aligned} \tag{11.11}$$

$$\begin{aligned} x &= \frac{\sum_{j=1}^{n} w_j x_j}{\sum_{j=1}^{n} w_j} \\ y &= \frac{\sum_{j=1}^{n} w_j y_j}{\sum_{j=1}^{n} w_j} \end{aligned} \tag{11.12}$$

Konečně pro přímou vzdálenost optimalizujeme rovnici (11.13). Bohužel parciální derivace (11.14) nemá analytické řešení, takže výpočet je nutné provést pomocí numerických metod.

$$minz = \sum_{j=1}^{n} w_j \sqrt{(x - x_j)^2 + (y - y_j)^2} \tag{11.13}$$

$$\begin{aligned} \frac{\partial z}{\partial x} &= \textstyle\sum_{j=1}^{n} \dfrac{x - x_j}{\sqrt{(x - x_j)^2 + (y - y_j)^2}} = 0 \\ \frac{\partial z}{\partial y} &= \textstyle\sum_{j=1}^{n} \dfrac{y - y_j}{\sqrt{(x - x_j)^2 + (y - y_j)^2}} = 0 \end{aligned} \tag{11.14}$$

Pro efektivní výpočet souřadnic použijeme substituci (11.15).

$$f_j(x,y) = \frac{w_j}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + \varepsilon}} \tag{11.15}$$

ε v rovnici (11.15) nám umožňuje obejít možnou rovnost souřadnic, která by mohla vést k situaci, kdybychom dělili nulou.

V rámci řešení začínáme v místě optimálního řešení kvadratické vzdálenosti. V jednotlivých iteracích se pak postupně blížíme k optimálnímu umístění vzdálenosti přímé. Výpočet provádíme pomocí rovnice (11.16).

$$x^{(k+1)} = \frac{\sum_{j=1}^{n} x_j f_j(x^{(k)}, y^{(k)})}{\sum_{j=1}^{n} f_j(x^{(k)}, y^{(k)})}, y^{(k+1)} = \frac{\sum_{j=1}^{n} y_j f_j(x^{(k)}, y^{(k)})}{\sum_{j=1}^{n} f_j(x^{(k)}, y^{(k)})} \tag{11.16}$$

S použitím starých (k) a nových $(k+1)$ souřadnic vypočteme nakolik iterace přispěla k zlepšení přesnosti výpočtu řešení, viz 11.17.

$$\Delta z = z^{(k)} - z^{k+1} \tag{11.17}$$

S každou iterací je zlepšení menší a menší. Takže zjednodušeně řešeno nejprve nám iterace zpřesní umístění např. o kilometry, potom o desítky metrů, potom o centimetry, ... Obvykle není efektivní snažit se dosáhnout co možná nejvyšší přesnosti, jelikož přínos takového zpřesnění by byl zanedbatelný.

Souřadnice vypočítané v poslední iteraci jsou souřadnicemi optimálními.

Při výpočtu umístění více nových objektů je zpracování ruční prakticky nemožné, takže spíše využíváme softwarových nástrojů, kterém nám s výpočty mohou výrazně pomoci.

Shrnutí

Lokalizační modely, jsou modely optimalizační, které nám umožňují nalézt optimální místo pro umístění nových objektů. Optimalizuje se kritérium obvykle odvozené z přepravní vzdálenosti mezi uvažovanými objekty.

Výpočet se liší podle toho kolik objektů umisťujeme, jestli pracujeme v 2D nebo 3D a typu vzdáleností, které uvažujeme.

Otázky

1. Co jsou lokalizační modely?
2. Jaký typ problémů jimi řešíme?
3. Existují rozdíly v pohledu na vzdálenosti v těchto modelech?
4. Jak obecně přistupujeme k řešení lokalizačních modelů při umístění jednoho objektu?
5. Můžete popsat iterační metodu pro výpočet optimálního umístění objektu s využitím přímé vzdálenosti?

Kapitola 12

Regresní modely

Náhled kapitoly
Další možností jak získat cenné informace k rozhodnutí je analýza numerických údajů popisujících předchozí vývoj a na základě těchto údajů odhadnout vývoj budoucí. Predikce budoucího vývoje pak umožňuje přijmout kvalitní rozhodnutí.

Tato kapitola je určena pro zasazení problematiky rozhodování do širšího kontextu dalších věd, v tomto případě statistiky.

Statistika samotná je podrobněji rozebrána v předmětu Statistika, který se vyučuje v 3. ročníku.

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně 45 minut.

Tato kapitola je spíše kapitolou motivační. Pro rozhodování lze totiž použít velké množství postupů, které jsou dobře známy ze statistiky. Statistika je ale natolik obsáhlá, že řešení pomocí jediné kapitoly není možné. Z tohoto důvodu berte prosím tuto kapitolu jako určitý most k předmětu *Statistika*, kde máte možnost se s touto problematikou seznámit podrobněji. Věřte tomu, že to stojí zato.

V této kapitole se zaměříme na základy *regresní analýzy*. V rámci regresní analýzy, analyzujeme n k-tic vysvětlujících proměnných, za účelem zjištění, zda nepřispívají k vysvětlení hledané vysvětlované veličiny. Převedeno do běžné češtiny – máme řadu údajů a ty se snažíme proložit křivkou, tak aby je co možná nejpřesněji kopírovala.

Nalezení křivky (regresní funkce) nám umožní vypočítat hodnoty i mezi těmito naměřenými údaji. Každý regresní model je obecně vyjádřitelný dle (12.1).

$$y = \eta + \epsilon \tag{12.1}$$

kde
y vysvětlovaná proměnná
η nenáhodná složka odhadu
ϵ náhodná složka odhadu

Matematicky jsme schopni odvodit pouze nenáhodnou složku odhadu η. O náhodné složce ϵ obvykle předpokládáme, že se řídí normálním rozdělením a z tohoto důvodu předpokládáme, že se chyby způsobené těmito náhodnými vlivy pro dostatečný počet analyzovaných údajů vzájemně vyváží.

Pro η volíme vhodný regresní model, ať už lineární regresní modely jako např. $\eta = \beta_0 + \beta_1 x_1$ nebo $\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ apod. nebo nelineární. My se z důvodu prostorových omezení budeme zabývat pouze modely lineárními.

Obecně můžeme lineární modely zapsat následovně (12.2).

$$\eta = \sum \beta_i f_i \tag{12.2}$$

kde
β regresní parametr
f regresor (funkce vysvětlujících proměnných)

Hlavním úkolem regresní analýzy je odhadnutí regresních parametrů. Jednou z nejpoužívanějších metod k zjištění odhadu regresních parametrů je metoda nejmenších čtverců. Při jejím použití nahrazujeme regresní parametry β jejich odhady b. Hledáme takovou regresní funkci jejíž součet odchylek je nejmenší. Pokud se snažíme nalézt parametry přímky, pak můžeme zapsat (12.3):

$$Y = b_0 + b_1 x_1 \tag{12.3}$$

Součet odchylek spočteme srovnáním dosaženým pomocí regresního modelu a hodnot, které jsme skutečně naměřili.

Matematicky regresní parametry vypočteme tak, že součet (12.4) parciálně derivujeme podle jednotlivých regresních parametrů, čímž dostaneme soustavu tzv. normálních rovnic, kterou dále klasicky řešíme.

$$Q = \sum_{j=1}^{n} (y_j - Y_j)^2 = \sum_{j=1}^{n} (y_j - b_0 - b_j x_j)^2 \tag{12.4}$$

Poté, co úspěšně zjistíme regresní parametry, je nutné rozhodnout, zda výsledný model dostatečně přesně zobrazuje realitu. K tomuto účelu můžeme využít tzv. *determinanční index* (12.5) nebo *korelační index* (12.6).

$$I^2 = \frac{s_T}{s_Y} = \frac{\sum (Y_j - \bar{y})^2}{\sum (y_j - \bar{y})^2} \tag{12.5}$$

$$I^2 = 1 - \frac{s_R}{s_Y} = 1 - \frac{\sum (y_j - Y_j)^2}{\sum (y_j - \bar{y})^2} \tag{12.6}$$

Determinanční index nabývá hodnot 0 – 1 a udává hodnotu v procentech, nakolik zvolený regresní model vysvětluje rozptyl naměřených hodnot vysvětlované proměnné y.

Vhodnost zvoleného modelu lze hodnotit také pomocí celkového *F-testu* a *dílčích t-testů*. Pomocí celkového F-testu, testujeme nulovou hypotézu tvrdící, že kromě parametru b_0, jsou všechny zvolené regresní funkce nulové, tedy nevýznamné pro vysvětlení proměnné y. Pro ověření této hypotézy použijeme následující testovací kritérium (12.7).

$$F = \frac{\dfrac{s_T}{p-1}}{\dfrac{s_R}{n-p}} \tag{12.7}$$

Nulovou hypotézu H_0 na *hladině významnosti* α zamítneme, pokud $F > F_{1-\alpha}$.

Dílčími t-testy hodnotíme jednotlivé regresní parametry, zda není daná (jedna) regresní funkce nulová. Použijeme testovací kritérium (12.8).

$$t_i = \frac{b_i}{s(b_i)} \tag{12.8}$$

Otázky

1. Vysvětlete podstatu testování hypotéz.
2. Co rozumíme pojmem lineární regrese?

Kapitola 13

Umělá inteligence

Náhled kapitoly
Použití prostředků umělé inteligence může také přinést důležité informace, které zefektivní rozhodovací proces. V této kapitole se zaměřím pouze na základní informace o *expertních systémech a neuronových sítích.*

Tato kapitola je zde opět vložena jako určité zasazení problematiky rozhodování do širšího rámce. Podrobněji je oblast umělé inteligence rozebrána v předmětu *Expertní systémy.*

Čas pro studium
Pro prostudování této kapitoly budete potřebovat přibližně 45 minut.

Použití prostředků umělé inteligence může také přinést důležité informace, které zefektivní rozhodovací proces. V této kapitole se zaměřím pouze na základní informace o *expertních systémech a neuronových sítích.*

Expertní systémy slouží pro nahrazení znalostí experta v určité oblasti počítačovým systémem. Hlavním cílem je poskytnout kvalifikované expertní informace v situaci, kdy expert člověk není přítomen a rozhodnutí je nutné přijmout velmi rychle nebo dokonce okamžitě.

Informace, kterou takový systém poskytne samozřejmě kvalitativně svým rozsahem a hloubkou nemůže v současné době nahradit experta - člověka. Může však pomoci učinit rámcově správné rozhodnutí, které se po příchodu experta může doladit.

Pro naplnění expertního systému je nutná úzká spolupráce experta v dané oblasti a také *znalostního inženýra.* S touto spoluprací jsou samozřejmě problémy:

1. je nutné, aby expert nechal nahlédnout znalostního inženýra a potažmo i uživatele expertního systému „pod pokličku“, což je věc, kterou někteří experti nesou nelibě,
2. znalosti experta je nutné vyjádřit v přísně formalizované podobě, tak aby si s nimi expertní systém poradil, tedy aby byly vyčísleny, porovnány apod. Člověk, však uvažuje jinak než počítač a obvykle nevyžaduje takto přísně formalizované údaje. Rozhodnutí experta člověka je totiž tvořeno řekněme tvrdou, vyčíslitelnou, měřitelnou částí (a v počítači dobře popsatelnou částí) a zkušenostmi, intuicí, které se zachycují velmi špatně. Často se stává, že expert přijde k problému a „ví“ řešení, aniž by příliš zkoumal celou situaci. Této úrovně „simulace“ experta pomocí expertních systémů ani dosáhnout nelze.

Znalosti v expertních systémech dělíme na, znalosti reprezentované deklarativně a znalosti reprezentované procedurálně. Deklarativní znalosti nazýváme také fakta nebo poznatky. Poznatkem pro expertní systém rozpoznávání zvířat může být: pes má čtyři nohy.

Procedurální znalosti definují, jakým způsobem bude probíhat odvozování a poznávání. Procedurální znalosti mají často charakter pravidel, pravidla přitom využívají definovaných poznatků: pokud je tvor pes, pak má čtyři nohy.

Expertní systém pak pomocí sledu otázek získává potřebná fakta o rozhodovací situaci a porovnává je s poznatky uloženými v jeho bázi znalostí. Teprve po získání těchto údajů podá nějaké vysvětlení.

Neuronové sítě pracují na jiném principu. Implementací umělých neuronových sítí se snažíme programově přiblížit fungování běžných neuronů mozku. Snažíme se zejména napodobit schopnost mozku učit se a vybavovat si naučené poznatky.

Neuronovou síť z hlediska modelování chápeme jako černou skříňku, která po naučení dává nějaké výsledky. V podnikové praxi se poměrně často využívají neuronové sítě spolu s regresními funkcemi pro odhady hodnot sledovaných proměnných.

Neuronová síť pracuje tak, že transformuje vstupy na výstupy přes různá spojení mezi neurony. Tento výsledek je porovnán s výsledkem skutečným naměřeným (trénovací množina) a v závislosti na tom, jak přesný byl výsledek se spojení mezi neurony posílí nebo naopak oslabí.

Proces učení tak probíhá po iteracích (po krocích), ve kterých jsou upravovány váhy spojení mezi neurony, až celková chyba klesne pod stanovenou, přijatelnou hladinu. Teprve poté je možné takto natrénovanou síť konzultovat – tedy předkládat vstupy modelované situace a odečítat výstupy.

Schopnost neuronové sítě naučit se přímo závisí na zvoleném typu neuronové sítě a její konfiguraci. Obecně se dá říci, že čím více vrstev neuronové sítě a neuronů v nich obsažených, tím větší je naděje, že neuronová síť bude schopna se naučit s přijatelnou přesností. Na druhou stranu doba nutná pro učení se s množstvím neuronů, respektive spojení mezi nimi zvyšuje a to obvykle exponenciálně. Naučení neuronové sítě nějakého „velkého“ problému tak může zabrat hodiny, dny nebo nemusí být při současném stavu výpočetní techniky řešitelné.

Některé pokročilé dataminingové nástroje k tomuto problému přistupují tak, že preferují rychlost nad přesností, s tím že nástroj zároveň umožňuje vyzkoušet i jiné metody analýzy dat, třeba pomocí regresní analýzy a všechny modely srovná – pro rozhodnutí se použije, ten s nejlepšími výsledky.

Očividně oblast umělé inteligence je velmi široká a značně přesahuje rámec, kterým se zabýváme v tomto předmětu. Zájemci o hlubší poznání této disciplíny mohou použít např. literaturu [36,37] nebo si zvolit předmět *Expertní systémy*.

Otázky

1. Zkuste vysvětlit, jak funguje proces učení neuronové sítě.
2. Zkuste vysvětlit, jak probíhá naplňování expertního systému.

Literatura

[1] *Apps/Dia - GNOME Wiki!* [online]. [cit. 2014-06-6]. Dostupné z: https://wiki.gnome.org/Apps/Dia/

[2] *Brainstorming* [online]. [cit. 2014-06-22]. Dostupné z: http://cs.wikipedia.org/wiki/Brainstorming

[3] *Dijsktra Algorithm* [online]. [cit. 2011-08-15]. Dostupné z: http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

[4] *Draw.io* [online]. [cit. 2014-06-6]. Dostupné z: https://www.draw.io/

[5] *Elektronický výukový systém* [online]. [cit. 2014-06-6]. Dostupné z: http://lms.vsb.cz

[6] *Finite mathematics utility: simplex method tool* [online]. [cit. 2014-07-8]. Dostupné z: http://www.zweigmedia.com/RealWorld/simplex.html

[7] *Ford-Fulkerson Algortihm* [online]. [cit. 2011-08-15]. Dostupné z: http://en.wikipedia.org/wiki/Ford-Fulkerson_algorithm

[8] *Graphviz - Graph Visualization Software* [online]. [cit. 2014-07-4]. Dostupné z: http://www.graphviz.org/

[9] *ISO 5807:1985 Information processing -- Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts.*

[10] *Karmakar* [online]. [cit. 2014-07-8]. Dostupné z: http://help.scilab.org/docs/5.5.0/en_US/karmarkar.html

[11] *List of Algorithms* [online]. [cit. 2011-08-15]. Dostupné z: http://en.wikipedia.org/wiki/Graph_Algorithm#Graph_algorithms

[12] *List of games in game theory* [online]. [cit. 2014-07-8]. Dostupné z: http://en.wikipedia.org/wiki/List_of_games_in_game_theory

[13] *Microsoft Project* [online]. [cit. 2014-06-22]. Dostupné z: http://office.microsoft.com/cs-cz/project/

[14] *Mind Mapping* [online]. [cit. 2014-06-22]. Dostupné z: http://alevelproduct.blogspot.cz/2012/09/mind-mapping.html

[15] *Numerics* [online]. [cit. 2014-07-3]. Dostupné z: http://www.rub.de/num1/softwareE.html

[16] *Open Workbench* [online]. [cit. 2014-06-22]. Dostupné z: http://sourceforge.net/projects/openworkbench/

[17] *Project Libre* [online]. [cit. 2012-06-22]. Dostupné z: http://www.projectlibre.org/

[18] *Quapro* [online]. [cit. 2014-07-8]. Dostupné z: http://atoms.scilab.org/toolboxes/quapro

[19] *The R Project for Statistical Computing* [online]. [cit. 2014-06-9]. Dostupné z: http://www.r-project.org/

[20] *read.graph {igraph}* [online]. [cit. 2014-04-7]. Dostupné z: http://www.inside-r.org/packages/cran/igraph/docs/UCINET

[21] *SciLab* [online]. [cit. 2011-08-15]. Dostupné z: http://www.scilab.org.

[22] *SmartDraw* [online]. [cit. 2014-06-6]. Dostupné z: http://www.smartdraw.com/

[23] *FreeMind* [online]. 2011. [cit. 2011-08-15]. Dostupné z: http://freemind.sourceforge.net/wiki/index.php/Main_Page.

[24] BEASLEY, J. E. *OR-Notes* [online]. [cit. 2014-06-6]. Dostupné z: http://people.brunel.ac.uk/~mastjjb/jeb/or/decmore.html

[25] BETHLEHEM, Jelke. *Applied Survey Methods - A Statistical Perspective*. New Jersey: Wiley, 2009. 392 s. ISBN 978-0470373088.

[26] BHUSHAM, N. I., KANWAL, R. *Strategic Decision Making: Applying the Analytic Heirarchy Process*. Londýn, Velká Británie: Springer - Verlag, 2004. 200 s. ISBN 1-8523375-6-7.
[27] BUZAN, Tony, BUZAN, Barry. *Myšlenkové mapy*. 2 vyd. Brno: Abatros Media a. s., 2012. 203 s. ISBN 978-80-265-0030-8.
[28] CLEMEN, Robert T., REILLY, Terrence. *Making Hard Decisions with Decision Tools*. Cengage Learning, 2004. 752 s. ISBN 978-0495015086.
[29] DRATVA, Vít. *Návrh koncepce vzdělávání v oblasti ochrany obyvatelstva a krizového řízení*. Ostrava: VŠB-TU Ostrava, Fakulta bezpečnostního inženýrství, 2015. 55 s.
[30] GEEWAH, Ng. *Intelligent Systems - Fusion, Tracking and Control*. Herfordshire: Research Studies Press, LTD., 2003. 264 s. ISBN 0-86380-277-X.
[31] GROS, Ivan. *Kvantitativní metody v manažerském rozhodování*. Praha: Grada Publishing a.s., 2003. 432 s. ISBN 80-247-0421-8.
[32] IAMTRAKUL, Pawince, HOKAO, Kozunori, TEKNOMO, Kardi. Public Park Valuation Using Travel Cost Method. *Proceedings of the Eastern Asia Society for Transportation Studies*. 2005, roč. 5, č. 1, s. 1249–1264. ISSN 1881-1132.
[33] KAISARE, Niket S. *Scilab Primer* [online]. [cit. 2014-07-3]. Dostupné z: http://www.shaastra.org/2011/media/main/events/Pentathlon/files/ScilabPrimer.pdf
[34] LEUNG, Terence, KIU, Tsing Nam. *A Short Introduction to Scilab* [online]. [cit. 2014-07-3]. Dostupné z: http://hkumath.hku.hk/course/MATH2601/www2601/IntroToScilab.pdf
[35] LOŠŤÁKOVÁ, H., MACHAČ, O. *Sbírka cvičení z předmětu Podniková ekonomika a management II*. Pardubice: Univerzita Pardubice, 2004. 21 s.
[36] MAŘÍK, V. *Umělá inteligence (1)*. Praha: Academia, 1993. 264 s. ISBN 80-200-0496-3.
[37] MAŘÍK, V. *Umělá inteligence (2)*. Praha: Academia, 1997. 373 s. ISBN 80-200-0504-8.
[38] MICROSOFT. *MS Visio* [online]. [cit. 2014-06-6]. Dostupné z: http://office.microsoft.com/cs-cz/profesionalni-software-pro-praci-s-diagramy-microsoft-visio-FX103472299.aspx
[39] MINDJET. *MindManager* [online]. MindJet, 2011. [cit. 2011-08-15]. Dostupné z: https://www.mindjet.com/mindmanager.
[40] MŠMT. *Výkonová data o školách a školských zařízeních – 2003/04–2013/14* [online]. [cit. 2015-06-6]. Dostupné z: http://www.msmt.cz/vzdelavani/skolstvi-v-cr/statistika-skolstvi/vykonova-data-o-skolach-a-skolskych-zarizenich-2003-04-2013
[41] OMG. *Unified Modeling Language (UML) Version 2.5 - Beta 2* [online]. [cit. 2015-06-2]. Dostupné z: http://www.omg.org/spec/UML/2.5/Beta2/
[42] ROACH, Brian, WADE, William W. Policy Evaluation of Natural Resource Injuries, Using Habitat Equivalency Analysis. *Ecological Economics*. 2006, roč. 58, č. 2, s. 421–433. doi: 10.1016/j.ecolecon.2005.07.019. ISSN 0921-8009.
[43] SAATY, Thomas L., VARGAS, Luis G. *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*. 2 vyd. Springer, 2012. 346 s. ISBN 978-1461435969.
[44] SAATY, Thomas L., VARGAS, Luis G. *Decision Making with the Analytic Network Process: Economic, Political, Social and Technological Applications with Benefits, Opportunities, Costs and Risks*. 2. vydání vyd. Springer, 2013. 363 s. International Series in Operations Research & Management Science 195. ISBN 978-1461472780.
[45] WANER, Stephan. *Tutorial for the simplex method* [online]. [cit. 2014-07-8]. Dostupné z: http://www.zweigmedia.com/RealWorld/tutorialsf4/frames4_3.html
[46] WAYNE, Kevin. *Ford-Fullkerson Demo* [online]. [cit. 2014-06-30]. Dostupné z: http://www.cs.princeton.edu/courses/archive/spring13/cos423/lectures/07DemoFordFulkerson.pdf
[47] ČSÚ. *Složení obyvatelstva podle pohlaví a jednotek věku (04-02) k 31.12.2014* [online]. [cit. 2015-06-6]. Dostupné z: http://vdb.czso.cz/vdbvo/tabparam.jsp?cislotab=04-02&&kapitola_id=368&voa=tabulka
[48] ČSÚ. *Sčítání lidu, domů a bytů 2011* [online]. [cit. 2015-06-2]. Dostupné z: https://www.csso.cz/csu/sldb

Slovník

AHP Analytic Hierarchy Process.

ANP Analytic Network Process.

BTS Base Transceiver Station.

CEO Chief Executive Officer.

CPM Critical Path Method.

CVM Contingent Value Method.

DTP desktop publishing.

FBI Fakulta bezpečnostního inženýrství.

GPS Global Positioning System.

GUI Graphical User Interface.

JRE Java Runtime Environment.

JSVV jednotný systém varovaní a vyrozumění.

KI kritická infrastruktura.

KŘ krizové řízení.

MCA multicriterial analysis.

OO ochrana obyvatelstva.

PDF Portable Document Format.

SO stupěň ohrožení.

TCM Travel Cost Method.

TCO Total Costs of Ownership.

UML Universal Modeling Language.

WTA Willigness to Accept.

WTP Willigness to Pay.

WWW World Wide Web.

XML Extensive Markup Language.

ČSÚ Český statistický úřad.

Přílohy

Příloha 1 - Analýza citlivosti Strom 2

Vykreslení grafu citlivosti celkových nákladů plynoucích z rozhodnutí na změny v očekávaných nákladech na léčení „malé“ epidemie chřipky

```
x <- 250:310
N <- 0.75 * (0.1 * x + 0.3 * 400 + 0.6 * 600)
plot(x, N, type="b", pch=21,
     main="Vliv nakladu lecby male epidemie na celkove naklady",
     xlab="naklady leceni [mil. EUR]",
     ylab="celkove naklady [mil. EUR]")
abline(lm(N~x))
```

Příloha 2 - Graphviz

Jednou z možností, jak vizualizovat sítě, je využít open source nástroj Graphviz [8]. Struktura sítě a její vlastnosti důležité z hlediska vykreslení sítě musí být předem známy. Použití obecných programů pro kreslení je sice teoreticky možné, avšak ruční rozmístění uzlů je zdlouhavé a výsledek většinou strukturálně neodpovídá úplně přesně zadání (srovnejte ručně vytvořený obr. 7.12 a automaticky generovaný 7.13).

To je důvodem pro nasazení specializovaných programů, které síť budou schopny vygenerovat automaticky. Graphviz je jedním z takových programů - jedná se o open source, který je dostupný pro řadu operačních systémů, ne pro všechny z nich je ale dostupná poslední stabilní verze programu. Program samotný poskytuje řadu nástrojů, které je možno ovládat z příkazové řádky a také jednoduché „klikací“ GUI, které umožní pohodlnou práci s programem.

Strukturu sítě definujeme tak, že zapisujeme sekvence uzlů spojených hranami např. 1 -> 2 -> 4 -> 7 -> 8 -> 11 a tyto cesty ohraničíme identifikací, že se jedná a definici grafu.

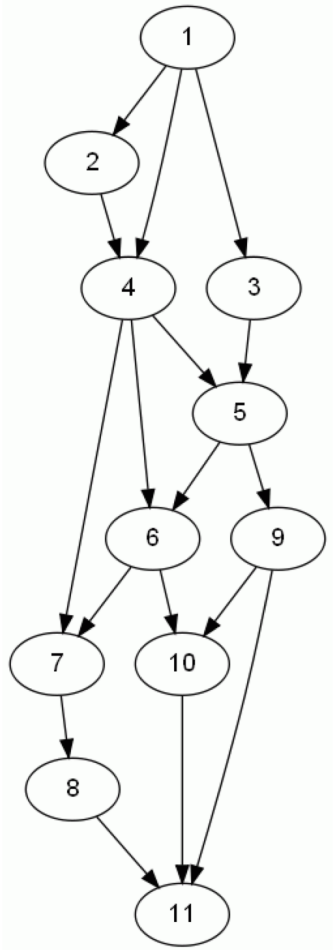
Pokud nechceme, aby graf byl orientovaný pak místo symbolu šipky „->“ použijeme „–“.

Celá definice naší sítě bude pro síť 7.12 vypadat následovně:

```
digraph G{
  1 -> 2 -> 4 -> 7 -> 8 -> 11;
  1 -> 4 -> 6 -> 7;
  1 -> 3 -> 5 -> 6 -> 10 -> 11;
  4 -> 5;
  5 -> 9 -> 10;
  9 -> 11;
}
```

Relativně jednoduché, že ano (vykreslování bylo provedeno pomocí knihovny dot). Výsledek našeho snažení bude vypadat, ale trochu jinak než na obr. 7.12, jelikož Graphviz provádí rozložení uzlů sám. To nám vadit nemusí, protože vlastnosti sítě zůstávají stejné. Výsledek našeho snažená najdete na obr. 13.1.

Pro řešení některých problémů, jako např. problému hledání nejkratší cesty, ale potřebujeme graf neorientovaný, takže výše uvedenou definici přepíšeme následovně. Výsledek je znázorněn na obr. 13.2. Vykreslování bylo provedeno pomocí knihovny neato.

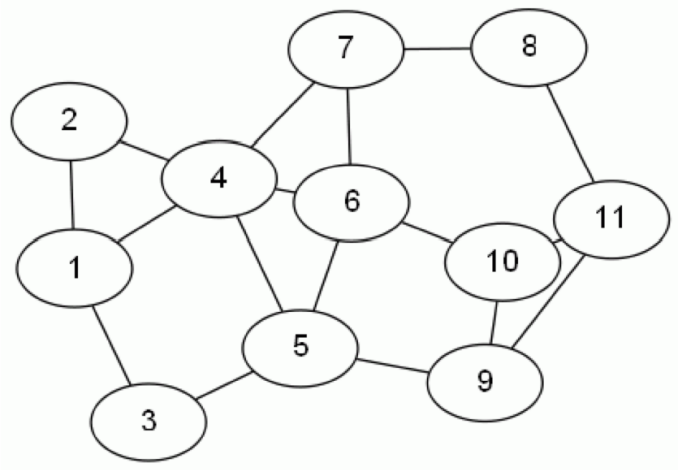


Obrázek 13.1: Orientovaná síť pomocí DOT jazyka

```
graph G{
  1 -- 2 -- 4 -- 7 -- 8 -- 11;
  1 -- 4 -- 6 -- 7;
  1 -- 3 -- 5 -- 6 -- 10 -- 11;
  4 -- 5;
  5 -- 9 -- 10;
  9 -- 11;
}
```

Ani v této formě ale není možné spočítat nejkratší cestu. Jednotlivé hrany budeme muset ohodnotit, co do délky. Upravený zápis bude vypadat následovně:

```
graph G{
  1 -- 2 [len=10];
  2 -- 4 [len=8];
  4 -- 7 [len=6];
  7 -- 8 [len=11];
  8 -- 11 [len=12];
  1 -- 4 [len=12];
  4 -- 6 [len=9];
  6 -- 7 [len=6];
  1 -- 3 [len=8];
  3 -- 5 [len=10];
  5 -- 6 [len=12];
  6 -- 10 [len=10];
  10 -- 11 [len=10];
  4 -- 5 [len=6];
  5 -- 9 [len=11];
  9 -- 10 [len=6];
  9 -- 11 [len=9];
}
```



Obrázek 13.2: Neorientovaná síť, generování pomocí knihovny neato

Rejstřík