

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Michal Bartoš

Efektivní rozvrhování skupiny výtahů

Katedra aplikované matematiky

Vedoucí bakalářské práce: Mgr. Jakub Černý

Studijní program: Informatika, programování

2007

Rád bych poděkoval všem, co mi umožnili v klidu pracovat na této bakalářské práci. Také bych rád poděkoval svému vedoucímu Mgr. Jakobovi Černému za návrh zajímavého tématu a konzultace ohledně nejasností s vypracováváním.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne: 7.8.2007

Michal Bartoš

Obsah

1	Výtahový systém.....	6
1.1	Co budova obsahuje.....	6
1.2	Děni v budově.....	6
1.3	Změny vybavení výtahového systému.....	7
2	Simulace.....	8
2.1	Tvorba simulačního programu.....	8
2.2	Diskrétní simulace.....	8
2.3	Spojité simulace.....	8
2.4	Rozdíl simulace a reálného provozu.....	9
3	Plánování skupiny výtahů.....	10
3.1	Složitost rozhodování.....	10
3.2	Lidská povaha.....	11
3.3	Rozdílné hodnocení efektivity plánovací strategie.....	11
4	Možné způsoby rozšíření vybavení výtahové techniky.....	13
4.1	Změna systému ovládání výtahů.....	13
4.2	Sledování počtu čekajících osob.....	13
4.3	Sledování rozložení lidí v rámci budovy.....	14
5	Různé plánovací strategie.....	16
5.1	Výtahový algoritmus.....	16
5.2	Vyřízení požadavků dle pořadí jejich vzniku.....	17
5.3	Obsluha výtahem, který může přijet nejrychleji (bez změn volby již rozhodnutých přiřazení výtahů).....	18
5.4	Obsluha výtahem, který může přijet nejrychleji (průběžně měnící volbu přiřazeného výtahu).....	19
5.5	Provoz směrem z přízemí (Up-peak).....	20
5.6	Provoz směrem do přízemí (Down-peak).....	21
5.7	Seskupování podle oblastí.....	21
5.8	Kyvadlová doprava (Sabbath service).....	22
6	Simulační program.....	24
6.1	Hlavní okno.....	24
6.2	Menu hlavního okna.....	25
6.3	Okno: Data source.....	25
6.4	Formát souboru se vstupními daty.....	26
6.5	Okno: Settings.....	27
6.6	Okno: Statistic.....	28
6.7	FIFO_algorithm.dll.....	28
6.8	nearest_algorithm.dll.....	29
6.9	nearest_recalc_algorithm.dll.....	29
6.10	Elevator_algorithm.dll.....	30
7	Srovnání jednotlivých plánovacích algoritmů.....	31
7.1	Standardní budova.....	31
7.2	Zvýšený provoz v rámci budovy.....	32
7.3	Standardní budova bez suterénu.....	33
7.4	Mrakodrap.....	33
7.5	Přetížený výtahový systém.....	34
7.6	Přetížený výtahový systém s většími výtahy.....	34

7.7 Zhodnocení výsledků měření.....	35
Literatura.....	36
Použitý software.....	36
Obsah příloženého CD.....	36

Název práce: Efektivní rozvrhování skupiny výtahů
Autor: Michal Bartoš
Katedra: Katedra aplikované matematiky
Vedoucí bakalářské práce: Mgr. Jakub Černý
e-mail vedoucího: Jakub.Cerny@mff.cuni.cz

Abstrakt: Tato práce se zabývá efektivností plánovacích algoritmů řídících skupinu výtahů. K porovnání jednotlivých plánovacích algoritmů byl vytvořen simulační program testující jejich vlastnosti vzhledem k nastavení parametrů výtahového systému a budovy do níž je umístěn. Výsledky testování jsou porovnány podle naměřených hodnot a určena vhodnost nasazení jednotlivých plánovacích algoritmů v závislosti na nastavených parametrech. Schopnosti plánovacího algoritmu závisí také na informacích, které je mu schopen poskytnout výtahový systém. Z tohoto důvodu se práce zabývá i samotným výtahovým systémem a navrhuje jeho změny dovolující plánovacím algoritmům pracovat efektivněji.

Klíčová slova: výtahový systém, plánovací algoritmus, simulace

Title: Effective planing of group of elevators
Author: Michal Bartoš
Department: Department of Applied Mathematics
Supervisor: Mgr. Jakub Černý
Supervisor's e-mail address: Jakub.Cerny@mff.cuni.cz

Abstract: This work deals with effectiveness of planing algorithms controlling group of elevators. For comparing planing algorithms was created simulation software, which test its properties with regarding to setting of parameters of elevator system and building where is elevator system placed. Results of testing are compared by measured values and determinate pertinence of use of particular planing algorithm in dependency on setting of parameters. Capabilities of planing algorithm also depend on informations which can elevator system provide to it. For this reason this work also deals with elevator system itself and suggest its changes allowing planing algorithms work more effective.

Keywords: elevator system, planing algorithm, simulation

1 Výtahový systém

Jedná se o ucelený pojem zahrnující vše související s výtahy. Na první pohled by se mohlo zdát, že se jedná pouze o výtahy a jejich řízení, ale to zahrnuje i chování lidí, rozložení budovy a schodišť v ní a další aspekty, které nejsou na první pohled zřejmé.

1.1 Co budova obsahuje

Budova se skládá z určitého množství poschodí, přičemž přízemí má specifický význam z důvodu jakým se lidé v budově pohybují (viz. kapitola 1.2 Dění v budově) a skupiny výtahů skládající se alespoň z jednoho výtahu, ale ve větších budovách jich je typicky více.

Výtahy mají v každém poschodí jedno společné ovládání, kterým se dají přivolat. Ovládání se může lišit podle různých druhů výtahového systému, ale typicky obsahuje dvojici tlačítek, která slouží pro výběr jakým směrem chce člověk jet, přičemž volbou směru je výtah automaticky přivolán. Tento systém ovládání s dvojicí tlačítek vznikl v době, kdy byly výtahy řízeny takzvaným Výtahovým algoritmem (viz. kapitola 5.1 Výtahový algoritmus) a toto byla jediná informace podle které se rozhodoval. Další možností je panel, který umožní přímo výběr poschodí, kam se chce člověk dostat.

Každý výtah je vybaven ovládáním pro určení cílového poschodí, kam se chtějí cestující dostat, pokud se ovšem nejedná o výtahový systém, který vyžaduje znalost cílového poschodí již při přivolávání výtahu. V takovém případě bývá již výtah vybaven pouze tlačítky pro okamžité uzavření dveří, otevření dveří a prvky potřebnými pro oznámení poruchy (zaseknutí výtahu mezi poschodími, ...) prostřednictvím telefonu, nebo reproduktoru s mikrofonom. V každém výtahu je také panel informující cestující o aktuální poloze výtahu a poschodích, která byla cestujícími zvolena jakožto cílová.

V jednotlivých poschodích může být dále panel zobrazující informace určené pro čekající lidi. V přízemí bývá zobrazeno, v jakém poschodí se nalézají jednotlivé výtahy, zatím co v ostatních poschodích bývá pouze informace o směru pohybu jednotlivých výtahů. Mohou se ovšem zobrazovat i další informace v závislosti na konkrétním výtahovém systému. Například doba za jakou výtah přijede, nebo který výtah pro člověka přijede. Tyto informace mohou lidem pomoci rozhodnout se, zda nejlépe raději po schodišti, nebo připravit se před správný výtah.

1.2 Dění v budově

Po budově se pohybují lidé a pro přepravu mezi jednotlivými poschodími využívají výtahového systému, případně pokud se jim zdá, že výtah příliš dlouho nepřijíždí, nebo z jiných důvodů, mohou využít schodiště. Lidé přivolávají výtahy vzhledem k výtahovému systému v budově, ale typicky ve chvíli, kdy přijdou k výtahům zvolí směr (nahoru, nebo dolů) jakým chtějí jet, čímž výtah přivolají.

Lidé se do budovy potřebují někudy dostat a také ji opustit. V závislosti na povaze budovy mohou k tomuto účelu sloužit suterénní podlaží, jelikož zde mohou být umístěna podzemní parkoviště. Hlavním vstupním bodem budovy ovšem zůstává přízemí, kudy přichází a také odchází většina lidí (případně všichni, podle typu budovy). Převážná většina lidí tedy využívá výtahový systém k přepravě z, nebo do

přízemí.

Průběh cesty jednoho člověka výtahovým systémem, za předpokladu že při přivolávání výtahu člověk volí směr cesty, od příchodu k výtahům v poschodí odkud chce jet až po příjezd na cílové poschodí:

- 1) přivolá výtah zvolením směru jakým chce jet, pokud ovšem není již výtah v tomto směru přivolán dříve někým jiným
- 2) čeká dokud nepřijede prázdný výtah, nebo výtah jedoucí směrem požadovaným při přivolávání výtahu
- 3) po otevření bezpečnostních dveří nastoupí do výtahu
- 4) zadá poschodí, do kterého chce být přepraven
- 5) uzavření bezpečnostních dveří
- 6) zrychlování výtahu (pokud výtah pojede pouze krátkou vzdálenost, nemusí dojít k dosažení maximální rychlosti výtahu, bod 6) tedy nenastane)
- 7) pohyb výtahu plnou rychlostí
- 8) zpomalování výtahu, tak aby zastavil v cílovém poschodí
- 9) otevření bezpečnostních dveří
- 10) nachází-li se výtah v poschodí, kam se chtěl člověk dostat, vystoupí, jinak se vrací k bodu 4)

1.3 Změny vybavení výtahového systému

Z důvodu bezpečnosti byla většina starých výtahů a všechny nové vybaveny bezpečnostními dveřmi, které oddělují prostor kabiny výtahu od výtahové šachty. Došlo tedy k zakrytí štěrbin, kterou mohly propadnout menší věci, ale především představovala riziko pro menší děti a zvířata, jelikož vlivem pohybu výtahu mohli být zachyceni o pohybující se stěnu výtahové šachty do této štěrbině skřípnuti.

Jednou z lidmi nejkładněji přijatých změn, bylo přidání zrcadel do kabin výtahů. Ačkoliv na provoz výtahového systému to nemá žádný vliv, cestující mají možnost zkontrolovat svůj zevnějšek, nebo jej částečně poupravit, což spousta lidí uvítala. Díky této možnosti určité činnosti lidem připadá cesta kratší, než kdyby pouze čekali a sledovali, jak dlouho cesta skutečně trvá.

V některých budovách může být nutnost strojovny, kde jsou umístěny motory pohánějící výtahový systém, značně nevhodná. Nebo, pokud je do budovy výtahový systém doděláván až dodatečně, za předpokladu, že je v budově nedostatek prostoru, zcela nemožná. Proto některé novější výtahové systémy řeší pohon nikoliv strojovnou, ale menším pohonem umístěným přímo ve výtahové šachtě. Nevýhodou takového řešení ovšem zůstává menší nosnost a především nelze použít ve výškových budovách, jelikož je koncipován pro nižší zdvih. Další informace je možno získat viz. Literatura [2].

Pro tažení výtahů se využívají splétaná ocelová lana, v nichž je uprostřed sisalové jádro sloužící k mazání lana z důvodu zabránění korozi. K tomuto systému existuje novější alternativa v podobě plochého lana, kde jsou jednotlivé prameny namísto stočení kolem jádra rozprostřeny do šířky a zapouzdřeny do polyuretanového obalu. Jelikož ocelové prameny lana jsou kryty, nedochází ke styku kovu o kov na hnacích kolech, což by mělo snižovat hlučnost a jiné negativní vlastnosti. Tento systém využívá společnost Otis viz. Literatura [3].

2 Simulace

Simulace je způsob zkoumání vlastností nějakého reálného děje pomocí experimentů s jeho matematickým modelem typicky implementovaným počítačovým programem.

2.1 Tvorba simulačního programu

Nejprve je potřeba navrhnout model, který co možná nejvíce odpovídá realitě, nebo alespoň vlastnostem které jsou pro simulaci podstatné. Tedy popis reálného děje pomocí matematických rovnic popisující simulovaný děj. Model se následně převádí do simulačního programu. Ačkoliv model by měl být tvořen tak aby byl programově implementovatelný, může se v průběhu programování simulátoru ukázat, že některá rozhodnutí byla nevhodná a je tedy zapotřebí model upravit.

2.2 Diskrétní simulace

Jedná se o způsob, jakým simulovat průběh děje, ve kterém je možno nalézt význačné události a jejich zpracováním určit průběh dění až po další význačnou událost (čas, kdy k ní dojde, jakého bude typu a jak se změní stav simulovaného děje). Další informace je možno nalézt viz. Literatura [6].

Hlavními součástmi diskretní simulace je plánovací kalendář a cyklus, který z něj vybírá události a zpracovává je, nebo předává dále ke zpracování. Z důvodu efektivnosti bývá typicky plánovací kalendář implementován pomocí haldy, která umožňuje velmi rychle vybírat nejmenší prvek (událost s nejmenším časem), což je hlavní způsob získávání událostí z plánovacího kalendáře. Výjimečně může dojít k potřebě najít v plánovacím kalendáři jinou událost, než tu s nejmenším časem (tu, kterou je zapotřebí zpracovat jako první), ale to je pouze za okolností, že je zapotřebí provést nějaké velmi specifické změny a typicky se to nedělá.

Na začátku simulace jsou nastaveny počáteční hodnoty a především vložena alespoň jedna událost do plánovacího kalendáře, která simulaci odstartuje. Poté se postupně z plánovacího kalendáře vybírají jednotlivé události podle času, kdy k nim má dojít a zpracovávají se. Dochází tedy k výpočtu pouze určité části simulovaného děje a v různých časových rozestupech, na rozdíl od spojitě simulace, kde dochází k přepočtu všech simulovaných objektů. Zpracování události má typicky za následek vznik nové události navazující na právě vybranou událost a její vložení do plánovacího kalendáře. Nová událost je spočtena z informací obsažených v právě zpracovávané události. Zpracování jedné události ovšem může vyvolat vznik i více nových událostí, nebo také žádné. Nevznikne-li při zpracovávání nová událost, znamená to, že daná část simulovaného děje skončila. Simulace končí ve chvíli, kdy je plánovací kalendář vyprázdněn a neexistuje žádná další událost ke zpracování.

2.3 Spojitá simulace

Slouží k simulování průběhu reálných dějů, rozdělením po pevně daných časových úsecích a krokováním. Je vhodná pro simulování dějů, kdy uživatele zajímá stav simulace i mimo význačné události, jako v diskretní simulaci. Hlavními důvody nutnosti využít spojitě simulace jsou, pokud ve sledovaném ději nelze nalézt vhodné významné události, podle kterých diskretní simulace probíhá, nebo pokud je zapotřebí průběh simulace vykreslovat uživateli. Lidé se totiž mnohem lépe orientují

v plynulém průběhu, nežli v přeskokách, které nemají pevný časový rozestup. Důležitým rozhodnutím při tvorbě spojitě simulace je volba času mezi jednotlivými kroky. Zkracování tohoto času má za následek zvyšování náročnosti na výpočetní kapacitu, jelikož přibývá kroků, které je zapotřebí vypočítat. Prodlužování má naopak za následek snižování přesnosti simulace, jelikož dochází k opomíjení událostí, které nastanou v průběhu jednoho simulačního kroku. Je tedy velmi důležité zvolit čas mezi jednotlivými kroky tak, aby co nejlépe vyhovoval požadavkům na spolehlivost simulace, ale aby se zároveň simulace nestala výpočetně příliš náročnou a tedy obtížněji využitelnou.

Na počátku simulace je zadán počáteční stav a nastaveno chování simulovaného děje. Tyto informace musí být dostatečné, aby bylo možno určit, jak bude děj pokračovat z každého stavu, do kterého se dostane. Simulace tedy podle nastaveného chování přepočte aktuální stav, jako by děj probíhal po dobu mezi jednotlivými kroky, čímž určí následující stav. Tomuto výpočtu se říká krok simulace. Při každém kroku simulace ovšem musí dojít k přepočtení celého aktuálního stavu i pro objekty, které do aktuální chvíle nebyly aktivní, jelikož neexistuje nic, co by určovalo, kdy začnou být aktivní a je tedy třeba ověřit, zda to není při právě zpracovávaném kroku. Toto je hlavní důvod, proč je spojitá simulace náročnější na výpočetní kapacitu, nežli diskrétní simulace. Výjimkou může být případ, kdy by byl čas mezi jednotlivými kroky dlouhý a děj se odehrával s velkým množstvím změn, což by zapříčinilo veliké množství událostí v diskrétní simulaci. Takto nastavená spojitá simulace by ovšem byla velmi nepřesná.

Po odstartování simulace probíhá zpracovávání jednotlivých kroků, čímž dochází k posunu v čase a tedy běhu simulace. Je-li známo, jak dlouhý čas reálného děje se bude simulovat, lze předem určit počet kroků, který bude simulace potřebovat. Díky tomu lze částečně odhadnout, jak dlouho bude simulace probíhat, ale z důvodu různě dlouhých výpočtů při jednotlivých krocích nelze dopředu přesně určit jak dlouho bude simulace probíhat.

2.4 Rozdíl simulace a reálného provozu

Simulace by měla co nejvěrněji napodobovat reálný provoz. Toho lze ovšem velmi těžko dosáhnout, jelikož by simulace musela obsahovat veliké množství aspektů. Takovouto komplikací jsou například netrpěliví lidé, kteří po určité době čekání na výtah vzdají a jdou raději po schodech nebo čas potřebný na nástup lidí. Lze sice při zadávání jednotlivých požadavků určit, jak dlouhý čas bude daný člověk potřebovat k nástupu do výtahu, ale z tohoto údaje se nedá určit, jak dlouhý čas bude potřeba pro celou skupinu lidí, jelikož lidé nenastupují po jednom, nýbrž jako skupina naráz. Tímto se dostáváme k hlavnímu problému, kterým jsou vhodná vstupní data popisující chování výtahů a především popisující pohyb osob po budově a jejich požadavků na využití výtahu pro přepravu mezi jednotlivými patry.

Výhodou simulace je možnost odzkoušet vhodnost volby řešení výtahového systému v určité budově ještě před jeho nákladnou realizací. Stále si ovšem musíme být vědomi toho, že řešení ideálně fungující v simulaci se může v reálném provozu ukázat jako nedostačující nebo nevhodné. Hlavním důvodem takového chybného předpokladu jsou nejpravděpodobněji chybná vstupní data, jelikož budova ještě není realizována a tudíž je nutno je nějakým způsobem odhadnout. Proto lze k simulaci přihlížet jakožto k doporučení, ale nikoliv jako k záruce vhodnosti volby nějakého konkrétního řešení.

3 Plánování skupiny výtahů

Plánováním skupiny výtahů se rozumí algoritmus určující kdy se má který výtah přesunout na jaké podlaží. Algoritmus může sledovat dění v budově nepřetržitě a podle aktuálního stavu budovy řídit pohyb jednotlivých výtahů. Typicky se ovšem jedná o abstrakci kdy řídicí algoritmus reaguje pouze na určité události které nelze triviálně rozhodnout na úrovni jednoho samotného výtahu a vydává povely, které jsou dále zpracovány samotnými výtahy. Řídicí algoritmus tedy neovládá jednotlivé motory pohybující kabinami výtahů nebo ovládajícími bezpečnostní dveře kabiny výtahu, ale určí výtahu do jakého poschodí se má dostat a o zbytek (rychlosti jednotlivých motorů pohybující výtahem a další nezbytné úkony) se postará výtah sám prostřednictvím ovládacích mechanismů k tomu určených.

3.1 Složitost rozhodování

Ačkoliv se tento problém může zdát triviální, nalezení optimálního naplánování pohybu jednotlivých výtahů je velice komplikované. Jelikož ve chvíli, kdy dochází k určení kam který výtah pojede, nejsou ještě známy požadavky od lidí, kteří teprve přijdou, ale měl-li by plánovací algoritmus tyto informace dopředu, mohl by se rozhodnout vhodněji. Nalezení optimálního naplánování je tedy bez této znalosti téměř nemožné. Pro každý algoritmus řídicí pohyb výtahů v budově bez znalosti budoucích požadavků, lze nalézt takové pořadí požadavků, aby výsledné naplánování nebylo optimální. Z tohoto důvodu se využívá heuristik, díky kterým lze částečně omezit vliv neznalosti budoucích požadavků jejich odhadnutím a připravením se na situaci, pokud by skutečně nastaly. Heuristika ovšem vychází z pravděpodobnosti výskytu takovýchto událostí, ale nemůže zaručit, že k nim skutečně dojde. Pokud dění v budově probíhá tak, jak bylo předpokládáno při tvorbě použité heuristiky, pak by měly odhadnuté události napomoci zefektivnění plánování. Použitím nevhodné heuristiky ovšem může zapříčinit zhoršení funkčnosti plánovacího algoritmu, jelikož se bude připravovat na příliš mnoho událostí, ke kterým následně nedojde.

Pomineme-li ovšem problém s neznalostí budoucích požadavků, jedná se stále o velice komplikovanou úlohu. Je totiž poměrně často zapotřebí zhodnotit stav celé budovy, zahrnující polohy a stav všech výtahů, počty lidí v jednotlivých poschodích budovy a výtazích, jejich požadovaný směr pohybu a na základě těchto informací rozhodnout další povely výtahům. Z tohoto důvodu se většinou plánovací algoritmy nerozhodují na základě všech dostupných informací a nepropočítávají neoptimálnější naplánování, ale využívají nějakého předpokladu, který má zajistit co nejefektivnější chod s využitím jednoduššího výpočtu.

Přehled jednotlivých komplikací při tvorbě plánovacího algoritmu:

- neznalost požadavků (přivolání výtahu), které teprve vzniknou, ale měly by ovlivnit již aktuální naplánování
- příliš mnoho informací mající vliv na naplánování (výpočetní náročnost)
- neznalost počtu lidí čekající v jednotlivých poschodích na výtah
- nevhodné chování lidí (přivolání výtahu volbou obou směrů současně, vzdání čekání na výtah a využití schodiště bez zrušení požadavku, ...)
- nevhodný popis dění v budově sloužící pro tvorbu heuristiky

3.2 Lidská povaha

Dalším omezením při návrhu výtahového systému je lidská povaha. Lidé nemají kompletní přehled o stavu výtahového systému a především neuvažují o efektivnosti naplánování vzhledem ke všem požadavkům, ale pouze z hlediska jak dlouhou dobu bude trvat přeprava jim samotným.

Mějme například výtahový systém, kde každý uživatel při příchodu zadá i patro kam pojede a do přízemí přijde skupina lidí větší, než pojme jeden výtah, ale polovina jich chce zhruba do poloviny budovy a vejdou se do jednoho výtahu, zatím co zbytek, který se také vejde do jednoho výtahu, bude chtít do horních pater budovy. Plánovací algoritmus rozhodne o poslání dvou výtahů, pro skupinu jedoucí do poloviny budovy a pro skupinu jedoucí do horních pater budovy, pro obslužení těchto požadavků. Do přízemí dorazí nejprve výtah určený pro lidi jedoucí do středních pater budovy a získali by také informaci o tom, že pro ty, kteří chtějí do horních pater budovy, přijede další výtah za chvíli. Takovéto rozhodnutí, ačkoliv by ve výsledku bylo nejrychlejší pro všechny přítomné, by nejspíše skupinka lidí jedoucích do horních pater budovy považovala za chybné a začala by nastupovat již do prvního výtahu. Lidé totiž většinou neuvažují o zdržení vlivem zastávek v poschodích mezi místem, kde nastupují a vystupují a proto by měli pocit, že čím dříve se dostanou do výtahu, tím dříve budou na místě, kam měli namířeno.

Další povahovou vlastností lidí je, že mají rádi přehled a pocit kontroly nad tím, co se okolo nich děje. Tomuto lépe odpovídají jednodušší plánovací algoritmy, které jsou z větší části řízeny přímo pokyny cestujících. Jejich nevýhodou ovšem je, že bývají méně efektivní a lidé tedy musejí čekat déle na příjezd výtahu, což jim nevyhovuje.

Jelikož navrhujeme výtahový systém tak, aby co nejvíce vyhovoval jeho uživatelům, je třeba zvolit jistý kompromis, kdy lidem ponecháme pocit kontroly nad výtahy, ale současně budeme minimalizovat čekací doby na jednotlivých poschodích.

3.3 Rozdílné hodnocení efektivnosti plánovací strategie

Plánovací algoritmy se snaží minimalizovat určitou hodnotu, jako například průměrnou dobu čekání lidí na příjezd výtahu, který je odveze kam potřebují. Minimalizovat podle minimální doby by nebylo příliš vhodné, jelikož tento údaj by mohl být velmi významně ovlivněn náhodným výskytem určité události, jako například, že výtah je přítomen na poschodí, kam přijde člověk s požadavkem a jeho čekací doba tedy bude nulová. Pro sledování časů je tedy vhodnější použít průměr, maximum, nebo medián. Medián se tolik nepoužívá, jelikož k jeho zjištění by bylo potřebné ukládat všechny příslušné časy za celou dobu sledování, ale není tolik ovlivněn výjimečnými výkyvy jako průměr, tedy za určitých okolností by byl vhodnější. Pro velké počty sledovaných hodnot se ovšem medián a průměr většinou příliš neliší a vzhledem k tomu, že získání průměru je snazší, používá se spíše průměrná hodnota.

Hodnoty získané v průběhu simulace nebo případně sledováním již realizovaného výtahového systému lze okamžitě zpracovávat a tím získat aktuální statistické hodnoty. Pro celkové hodnocení efektivnosti plánovacího algoritmu jsou ovšem podstatnější hodnoty zjištěné až po ukončení celé simulace nebo v určitých specifických časových bodech. Takovýmito časovými body může být kupříkladu půlnoc, jelikož lze předpokládat, že tou dobou bude výtahový systém v klidu

(statistické údaje tedy neobsahují žádné rozpracované údaje, které mohou zkreslit výsledky) a lze tedy zhodnotit průběh za právě proběhlý den. Dalšími takovými časovými body mohou být okamžiky, kdy dochází k přepnutí mezi režimy Up-peak, Down-peak a normálním plánovacím algoritmem (viz. kapitola 5 Různé plánovací strategie). Zde je ovšem potřeba odlišit hodnoty získané pouze z právě dokončeného režimu, případně z období které je pro nás z nějakého důvodu zajímavé a byl kvůli němu příslušný časový bod vybrán.

Pro uživatele výtahového systému by se mohl zdát nejdůležitější průměrný, nebo maximální čas od doby příchodu po dopravení do požadovaného podlaží. Lidé ovšem hodnotí efektivnost výtahů především podle doby čekání na výtah a samotný čas cesty výtahem již nepřipisují vlivu plánovače, ale jako něco neovlivnitelného, pokud se výtah ovšem nezačne pohybovat opačným směrem, než oni potřebují. Chceme-li tedy vyvolat dojem efektivního návrhu výtahového systému, je třeba minimalizovat dobu čekání na výtah a vyhnout se přibírání lidí z jiných pater, než ve směru cesty lidí, kteří jsou již ve výtahu. Také je vhodné soustředit se spíše na maximální dobu než na průměrnou, jelikož lidé mají tendenci pamatovat si pouze čekají-li dlouho.

Některé algoritmy minimalizují průměrný kvadratický čas čekání lidí na příjezd výtahu. Jedná se o období průměrné hodnoty s rozdílem, že se nenačítají přímo získané hodnoty, ale jejich kvadráty. Výhodou oproti průměrné hodnotě je snížení vlivu krátkých časů, zatím co dlouhá doba čekání se projeví mnohem více. Tento způsob jakoby zvyšuje důležitost nového údaje s jeho zvětšující se hodnotou, což odpovídá lidskému vnímání doby po níž čekají na příjezd výtahu. Čím déle jsou lidé nuceni čekat, tím se pro ně tato doba stává důležitější. Krátké doby tedy nejsou tolik podstatné, ale je zapotřebí se vyvarovat dlouhých časů od přivolání výtahu po jeho příjezd. Ačkoliv tato hodnota na první pohled nevypovídá o žádném údaji zajímavém pro běžného uživatele, pro fungování některých plánovacích algoritmů je mnohem vhodnější nežli jiné hodnotící údaje.

Pro provozovatele výtahů může být naopak zajímavý údaj o spotřebě energie spojené s provozem výtahového systému, která může být například nepříznivě ovlivněna pravidelným rozmístováním nevyužitých výtahů v rámci budovy z důvodu rychlejší reakce při přivolání výtahu nebo jinými rozhodnutími, která byla učiněna z důvodu optimalizace plánovacího rozvrhu podle jiného kritéria.

Efektivnost plánování lze hodnotit i podle počtu nepřepravených osob při překročení přepravních schopností výtahového systému nebo počtu lidí čekajících na příjezd výtahu déle, než určitou dobu. Tato hodnocení se ovšem hodí pouze pro teoretické simulace testující volbu konkrétního výtahového systému, jelikož při reálném provozu by k takovýmto situacím nemělo nikdy dojít.

4 Možné způsoby rozšíření vybavení výtahové techniky

Tato kapitola navrhuje možné změny vybavení výtahového systému, které nejsou běžně používané a mohly by napomoci k lepšímu fungování výtahového systému.

4.1 Změna systému ovládání výtahů

V dnešní době se používá dvojice tlačítek na každém poschodí pro celou skupinu výtahů současně, určující, zda cestující chce jet směrem nahoru nebo dolů. Tento systém ovšem neumožňuje plánovacím algoritmům rozhodovat se řádně již při požadavku na výtah. Alternativou k tomuto způsobu je systém kdy lidé při příchodu do podlaží zadají přímo kam chtějí být přepraveni, viz. Literatura [1, 4].

Mějme situaci, kdy člověk z prvního patra chce jet do pátého patra a ještě než pro něj přijede výtah, přijde člověk, který chce jet z pátého patra do přízemí. Jsou-li výtahy v přízemí, bude nejlogičtější poslat pro člověka do pátého patra druhý výtah, jelikož člověk z prvního patra může chtít jet třeba až do desátého. Pokud by ovšem plánovací algoritmus již věděl, že člověk z prvního patra chce jet do pátého, nemusel by druhý výtah posílat, jelikož by věděl, že v pátém patře se první výtah vyprázdní a bude tedy moci obsloužit i druhého člověka.

Je tedy vidět, že volba čísla cílového podlaží, namísto volby směru pohybu, by dovolila plánovacímu algoritmu pracovat efektivněji. Tento systém ovládání ovšem vyžaduje od všech lidí na podlaží zadat cílové podlaží. Přináší ovšem také komplikaci. Pro možnost využít této informace v případě většího počtu lidí na jednom podlaží by bylo zapotřebí nějakého způsobu, jak určit, kdo má nastoupit do kterého výtahu, jinak by bylo nutno počítat s tím, že kdokoliv může nastoupit do kteréhokoliv výtahu. Zde je ovšem problém s ochotou lidí podřídit se určitému naplánování (viz. kapitola 3.2 Lidská povaha).

Systém přivolávání výtahu volbou cílového podlaží se již používá. V této kapitole je zmíněn jelikož v Evropě se zatím příliš nerozšířil a jeho zavedení by tedy představovalo jistou modernizaci výtahových systémů.

4.2 Sledování počtu čekajících osob

Typicky výtahový systém neví kolik lidí čeká na výtah v jednotlivých poschodích, výjimkou je případ, kdy každý člověk přivolává výtah volbou podlaží kam se chtějí dostat. Tato informace by ovšem pomohla řídicímu algoritmu při příchodu většího počtu lidí na jedno poschodí zefektivnit naplánování pohybu výtahů.

Za normálního provozu se nepředpokládá, že by z jediného poschodí odjíždělo více lidí než uveze jeden výtah najednou. Pokud by ovšem například skončila nějaká konference, opouštělo by určité poschodí hodně lidí najednou. Výtahový systém ovšem pouze ví o požadavku z daného poschodí směrem do přízemí, ale nepozná že je zapotřebí poslat několik výtahů. Dojde tedy k poslání jednoho výtahu, který po naplnění odjede a lidé přivolávají další. Tento proces se může opakovat i vícekrát, jedná-li se o velkou konferenci v budově s několika výtahy. Věděl-li by ovšem výtahový systém kolik lidí čeká v daném poschodí, mohl by poslat rovnou příslušný počet výtahů a tím zkrátit čekání lidí opouštějících konferenci.

Počet lidí čekající v jednotlivých poschodích lze zjišťovat buďto tak, že jej zadají

lidé nebo si jej zjišťuje výtahový systém sám. Oba způsoby poskytují možnost zjištění, že člověk vzdal čekání na příjezd výtahu a šel raději po schodišti nebo z jiného důvodu požadavek ruší.

Výhodou možnosti, kdy lidé sami zadávají kolik jich v daném poschodí čeká, je jednodušší realizace, jelikož stačí přidat ovládací prvek, kterým to lidé zadají nebo pokud by výtahový systém počítal počet stisknutí přivolávacího tlačítka. Výhodou je také možnost zadání lidí, kteří se k výtahům teprve blíží, ale jsou již dostatečně blízko, aby bylo zřejmé, že do příjezdu výtahu dorazí. Výraznou nevýhodou je ovšem chování lidí. Spousta lidí se totiž domnívá, že pokud stisknou přivolávací tlačítko vícekrát, přijede výtah dříve. Paradoxně, pokud by výtahový systém tímto způsobem zjišťoval počet lidí čekajících v jednotlivých poschodích, pravděpodobně by výtah skutečně přijel dříve, jelikož by si výtahový systém myslel že v daném poschodí je spousta lidí. Narušilo by to ovšem chod výtahového systému jakožto celku, jelikož by docházelo k zasílání přílišného počtu výtahů nebo nevhodným přeplánováním. Jakmile by se tedy lidé dozvěděli, že zadáním většího počtu čekajících lidí, než jich skutečně v daném poschodí čeká, přijede výtah rychleji, začal by tento údaj postrádat na věrohodnosti a nedal by se tedy použít.

Výhodou sledování počtů lidí na jednotlivých poschodích výtahovým systémem je vyšší věrohodnost získaných údajů, za cenu nemožnosti rezervace míst pro lidi, kteří se k výtahům teprve blíží někým kdo již u výtahů čeká ví, kam pojedou. Hlavní nevýhodou je složitá implementace takového řešení a neznalost rozdělení kolik lidí chce jet, kterým směrem. Možným způsobem by bylo sledování prostoru před výtahy kamerou a vyhodnocováním získaných záběrů zjišťovat počet lidí. Buďto podle změn oproti záběru prázdné chodby nebo počítáním lidí přicházejících a odcházejících ze zorného pole kamery. Výhodou řešení prostřednictvím kamer je možnost jejich využití i k zabezpečení budovy. Zjednodušenou možností počítání lidí přicházejících před výtahy by mohly být dvojice optických bran, které by podle pořadí přerušení určily zda člověk přichází nebo odchází. Problém ovšem nastává pokud půjde více lidí vedle sebe a tudíž přeruší bránu pouze jednou. Další možností zjišťování počtu lidí čekajících u výtahů je pomocí senzorů umístěných v podlaze. Jelikož výtahy jsou dimenzovány podle váhové nosnosti, nabízí se možnost umístit pod podlahu senzor zjišťující váhu lidí před výtahy. Toto řešení se zdá poměrně vhodné, vyžaduje ovšem speciální podlahu před výtahy, což může být nevhodné z důvodu designu budovy.

4.3 Sledování rozložení lidí v rámci budovy

Již v dnešní době jsou některé budovy vybaveny elektronickými zámky na čipovou kartu nebo jiný elektronický senzor namísto klasických mechanických zámků. Pokud by se každý člověk nějak registroval při průchodu několika klíčovými dveřmi, kupříkladu senzory ve dveřích by byly dostatečně citlivé aby zaregistrovaly i lidi pouze procházející již otevřenými dveřmi. Tento systém by pomohl jakožto zabezpečení budovy před nežádanými návštěvníky, ale také by mohl poskytnout výtahovému systému poměrně přesný přehled o rozložení lidí v rámci budovy. Tato informace by se mohla využít při plánování rozmístování výtahů po budově za účelem zkrácení doby čekání na příjezd výtahu. Je zbytečné takto umístit výtah do blízkosti pater, kde je již pouze minimum lidí, naopak je výhodné jej umístit do blízkosti pater, kde je lidí více.

Náklady na zbudování systému sledujícího pohyb lidí v rámci budovy pouze za

účelem zefektivnění plánování pohybu výtahů by se nejspíše nevyplatily. Je-li ovšem budova něčím podobným vybavena z jiných důvodů (například zabezpečení nebo jakožto alternativa ke klasickým klíčům), lze tuto informaci využít pro výtahový plánovací algoritmus. Informace o rozložení lidí má ovšem pouze informativní charakter, jelikož z ní nelze přesně určit kdy, kam a zda vůbec budou lidé chtít přepravit. K optimálnějšímu naplánování pohybu výtahů tedy přispěje především v době minimálního zatížení výtahového systému, jelikož při vyšší zátěži je potřeba především obsloužit požadavky, které již nastaly.

5 Různé plánovací strategie

Skupinu výtahů lze řídit pomocí různých plánovacích algoritmů. Jednotlivé plánovací algoritmy se od sebe liší způsobem svého fungování, ze kterého vyplývá pro jaké druhy budov (počty poschodí, předpokládané počty přepravených osob, ...) je vhodnější a pro které naopak nebylo plánováno jeho nasazení a tedy není příliš vhodný. Dalším významným rozdílem mezi jednotlivými plánovacími algoritmy jsou jejich požadavky na vybavení výtahového systému a informace, které od něj potřebuje získat pro svoji správnou funkčnost (způsob přivolání výtahu: tlačítka nahoru/dolu nebo volba cílového podlaží, detekce počtu lidí čekajících v jednotlivých poschodích na příjezd výtahu, ...).

Jednotlivé plánovací strategie se od sebe také liší tím, zda mění již hotový plán nebo nikoliv. Výhodou strategií neměnicích své naplánování je, že mohou lidé čekající v jednotlivých podlažích i výtazích přesně informovat o časech příjezdu výtahu nebo který výtah obslouží jejich požadavek, aby měli možnost se k němu připravit. Oproti tomu strategie, které mají možnost průběžně měnit již hotové naplánování dalších akcí, mohou lépe reagovat na nově vzniklé události a tím dosahovat optimálnějšího chodu výtahového systému. Cenou za tuto dynamičnost je nutnost přepočítávat již hotový plán a tedy vyšší výpočetní náročnost. Plánující strategie mající možnost měnit své naplánování se ovšem mohou chovat jako by tuto možnost neměli. Pokud plánovací strategie nehodlá využívat možnosti změn v plánu, měla by být přímo koncipována bez této možnosti, čímž se sníží výpočetní náročnost a přibude možnost informování o přesných časech příjezdu výtahů na jednotlivá poschodí.

5.1 Výtahový algoritmus

Základní plánovací algoritmus používaný pro řízení výtahů. Princip jeho fungování je dobře známý, zmiňuje se o něm například internetová encyklopedie Wikipedia, viz. Literatura [5]. Tento algoritmus byl navržen pro řízení jednoho výtahu a není tedy příliš vhodný pro řízení skupiny výtahů. Jeho samostatné použití na jednotlivé výtahy výtahového systému má za následek, že výtahy nebudou spolupracovat a může tedy docházet k tomu, že jeden požadavek pojedou vyřídit více výtahů současně, což je značně neefektivní (požadavek vyřídí výtah, který přijede první a ostatní jely zbytečně).

Výtah vyrazí obsloužit první požadavek (určení, kterým směrem se má začít pohybovat, existují-li požadavky v obou směrech závisí na konkrétní implementaci) a nadále zastavuje ve všech poschodích ve směru svého pohybu, kde je zaregistrován požadavek na výtah směřující stejným směrem kam se výtah pohybuje nebo tam, kde chce někdo z cestujících, které veze, vystoupit. Ve chvíli, kdy nemůže takto pokračovat dále, buďto zastaví, pokud neexistují jiné požadavky nebo změni směr pohybu a vydá se obsloužit nejbližší požadavek v novém směru pohybu.

Pro využití tohoto plánovacího algoritmu k řízení skupiny výtahu by bylo nutno jej výrazněji upravit z důvodu jeho zefektivnění, což by mělo za následek spíše vznik nového plánovacího algoritmu, nežli pouhou modifikaci. Výtahový algoritmus ovšem lze uplatnit v jiných plánovacích algoritmech, jakožto pomocný algoritmus pro rozhodování v určitých specifických případech. Ke svému fungování totiž kromě informací o stavu budovy potřebuje pouze údaj o směru jakým se jednotlivé výtahy naposledy pohybovaly. Příkladem využití výtahového algoritmu, jakožto součástí

jiného plánovacího algoritmu, může být jeho využití v algoritmu obsluhujícím požadavky výtahem, který je schopen se dostat nejrychleji do poschodí jeho vzniku (viz 5.4 Obsluha výtahem, který může přijet nejrychleji (průběžně měnící volbu přiřazeného výtahu)).

Dnes se tento algoritmus ve zjednodušené formě používá pro plánování pohybu čtecích hlav v počítačových harddiscích. Neexistuje zde ovšem ekvivalent pro směr pohybu jakým chce člověk jet z patra, kde se právě nalézá, ani pro určení cílového patra. Algoritmus tedy musel být upraven, aby tyto informace nevyužíval. Tímto došlo ke zjednodušení, jelikož vyřízení požadavku nemá za následek vznik nového požadavku (bylo zapotřebí zastavit v cílovém patře právě naloženého cestujícího) a také odpadla kontrola směru požadavku, jinak zůstal algoritmus stejný.

5.2 Vyřízení požadavků dle pořadí jejich vzniku

Jedná se o plánovací algoritmus postavený na jednoduché úvaze obsluhování jednotlivých požadavků v pořadí v jakém přichází. Tento princip se díky své jednoduchosti běžně využívá na mnoha místech pod různými názvy: fronta, FIFO (First In, First Out) nebo také lidově „kdo dřív přijde, ten dřív mele“. Ne vždy je jeho využití optimální (například pro výtahový systém), ale na mnoha místech se osvědčil (například fronta v samoobsluze).

V základní podobě by tento plánovací algoritmus fungoval tak, že každý požadavek na výtah by se zařadil na konec fronty společné pro celou budovu. Jednotlivé výtahy by vždy po skončení obsluhy právě zpracovávaného požadavku (cesta na podlaží kde vyzvedne cestující nebo jejich rozvážení do poschodí, kam chtějí dopravit) z této fronty vybraly první požadavek a vydaly by se jej obsloužit (dojel by na poschodí, kde požadavek vznikl a nechal nastoupit cestující). Tento způsob je ovšem značně neefektivní, jelikož výtahy mohou za určitých podmínek již cestou k novému požadavku obsloužit některé další. Pro potřeby výtahového systému je tedy zapotřebí tento plánovací algoritmus rozšířit oproti standardnímu chování v několika ohledech. Nejpodstatnější změnou je dřívější vyřízení požadavků, které se nalézají na cestě mezi jeho aktuální pozicí a poschodím kam má namířeno. Musí ovšem směřovat stejným směrem jako se výtah aktuálně pohybuje (nebo bude pohybovat, neopustil-li ještě poschodí) a kromě zdržení způsobeného zastavením v podlaží(ch) jinak ovlivnit pohyb výtahu. Tato kontrola s sebou nese komplikaci především v ověření, zda výtah, který je již v pohybu zvládne dobrzdit do nového poschodí a jel-li výtah obsloužit nový požadavek z fronty, je zapotřebí tento požadavek někam odložit, jelikož ve frontě již není (byl vybrán ke zpracování) a nesmí se ztratit. Tyto částečně odsunutě požadavky lze buďto ukládat do speciálních front ke každému výtahu zvlášť nebo je vracet na začátek (jelikož jsou právě zpracovávány, musely být vybrány ze začátku fronty) společné fronty.

Jelikož jeden požadavek může představovat i více lidí čekajících v jednom podlaží na příjezd výtahu, je zapotřebí vždy, když výtah opouští poschodí ověřit, zda se do výtahu vešli všichni lidé, kteří zde čekali nebo zda tam někdo zůstal. Jestliže se do výtahu vešli všichni lidé není zapotřebí provádět žádné speciální akce, ale pokud v poschodí někdo zůstal, je zapotřebí obsloužit i ty, kdož se do výtahu napoprvé nevešli. Typicky výtahový systém nemá jak zjistit, že v poschodí někdo zůstal a lidé, kteří v poschodí zůstali si výtah musí přivolat znovu. Má-li ovšem výtahový systém jak zjistit, že v poschodí čekalo více lidí, nežli se do výtahu vešlo, může další výtah přivolat sám automaticky. Nejpřirozenějším způsobem vyřízení zbylých lidí

čekajících v poschodí je přidáním nového požadavku do fronty k ostatním požadavkům, odkud je standardní cestou vyřízen. Zbývá pouze zvolit, kam do fronty tento nový požadavek vložit, aby zůstal zachován princip vyřizování lidí v pořadí v jakém přicházel. Jelikož ovšem výtahový systém neví, jak dlouho čeká nejdéle čekající člověk který se do výtahu nevešel, je zapotřebí zvolit, jakým způsobem bude nový požadavek do fronty zařazen bez ohledu na pořadí, v jakém lidé přicházel. Zařazení tohoto požadavku na začátek fronty má za následek její rychlejší vyřízení a omezuje tedy hromadění lidí v jednom nebo několika málo poschodích. Naopak zařazení tohoto požadavku na konec fronty lépe odpovídá výtahovému systému, který nemá jak zjistit, zda se všichni lidé čekající v poschodí, do výtahu vešli. Tento plánovací algoritmus je poměrně snadný a tedy nenáročný na výpočetní kapacitu. Vznikají-li ovšem požadavky nevhodně, kupříkladu od nižších poschodí k vyšším a vždy směrem dolů, bude docházet k obsluze každého požadavku jednotlivě, což je značně neefektivní. Je tedy vidět, že tento plánovací algoritmus se za určitých podmínek chová značně neoptimálně.

5.3 Obsluha výtahem, který může přijet nejrychleji (bez změn volby již rozhodnutých přiřazení výtahů)

Myšlenka tohoto plánovacího algoritmu spočívá v minimalizování doby čekání na příjezd výtahu, čehož dosahuje výběrem výtahu, který bude nejdříve schopen nový požadavek vyřídit, tedy přijet do tohoto poschodí a pokračovat ve směru požadavku. Vlivem nových požadavků a neznalosti cílového poschodí, kam budou chtít jet lidé, kteří ještě nenastoupily do výtahu, ovšem může dojít k tomu, že výtah dříve vybraný jakožto nejvhodnější, již zdaleka nejvhodnější nebude. Z tohoto důvodu dochází k rozdělení tohoto algoritmu do dvou variant podle toho, zda přehodnocují již hotový plán, či nikoliv.

Plánovací algoritmus vždy při novém požadavku na výtah spočte čas, za jaký jsou schopny jednotlivé výtahy tento požadavek obsloužit. Výtah, který je schopen požadavek obsloužit nejrychleji, je k tomuto požadavku přiřazen a do jeho plánu je přidáno zastavení v podlaží, odkud nový požadavek směřuje. Výpočet času, za jaký je výtah schopen obsloužit nový požadavek, musí brát v úvahu stav, ve kterém se výtah aktuálně nalézá a naplánování jeho budoucích akcí. Kontrola aktuálního stavu je důležitá pro případ, že nový požadavek je mezi výtahem a jeho další zastávkou a pokud by byl výtah schopen dojet do poschodí s novým požadavkem, mohl by jej rovnou obsloužit. Také je zapotřebí kontrolovat zaplněnost výtahu, zda by byl schopen nabrat další cestující nebo je nutno nejprve nechat některé vystoupit. Zatím co zkoumání naplánovaných budoucích akcí je potřebné, jelikož výtah před dosažením podlaží s novým požadavkem bude muset zastavit v několika jiných podlažích, což typicky trvá déle, nežli cesta mezi jednotlivými podlažími.

Tento algoritmus má možnost přidat před již naplánované zastávky další a to pokud splňují podmínky pro jejich přijetí, což má za následek prodloužení doby, která byla dříve spočtena jakožto potřebná k dosažení dalších zastávek. Také ovšem není známo, kam chtějí jet lidé, pro které byl výtah již vyslán, ale ještě nenastoupili a tedy není známo, kdy vystoupí a výtah bude moci změnit směr, aby obsloužil požadavky v opačném směru. Je tedy zapotřebí odhadnout jak daleko bude muset výtah před svým vyprázdněním dojet. Jednou z možností je předpokládat nejdelší případ, kdy lidé budou chtít jet až do nejvyššího (nejnižšího) poschodí budovy (použito v testovaném plánovacím algoritmu). Chtějí-li ovšem lidé vystoupit dříve, dojde k

rychlejšímu uvolnění výtahu, což má za následek zkrácení doby, která byla dříve spočtena jakožto potřebná k dosažení dalších zastávek. Výpočet doby dosažení určitého podlaží tedy není absolutní, jelikož se může časem zvětšovat i zmenšovat, což může způsobit nevhodnost volby přiřazení výtahu k poschodí s požadavkem. Tento algoritmus není optimální, ani nehledá neoptimálnější naplánování určitého z údajů dostupných výtahovému systému. Snaží se minimalizovat dobu, po jakou lidé čekají na příjezd výtahu, při zachování menší zátěže na výpočetní kapacitu, která je potřebná pro běh plánovacího algoritmu.

5.4 Obsluha výtahem, který může přijet nejrychleji (průběžně měnící volbu přiřazeného výtahu)

Jedná se o modifikaci plánovacího algoritmu popsaného v předchozí kapitole (viz. 5.3 Obsluha výtahem, který může přijet nejrychleji (bez změn volby již rozhodnutých přiřazení výtahů)). Důvodem změn je nemožnost tvorby plánu na delší dobu dopředu z důvodu neustálého vývoje stavu budovy zapříčiněného příchody nových lidí nebo zadáním cílového podlaží až v době nastoupení do výtahu.

Tato varianta plánovacího algoritmu, na rozdíl od varianty neměnící volby výtahů, které mají obsloužit jednotlivé požadavky, určuje další akce vždy znovu. Tento způsob omezuje vliv neznalosti budoucích událostí tím že zkrátí dobu, na jakou dopředu určuje dění v budově a nové události jsou zpracovány již při prvním volání plánovacího algoritmu. Nevýhodou je zvýšení výpočetní náročnosti, jelikož je zapotřebí pokaždé zkoumat všechny požadavky znovu a určit, který výtah má obsloužit který požadavek nebo, zda dojde k jejich obsluze teprve po vyřízení jiných požadavků.

Tato varianta plánovacího algoritmu vyžaduje způsob, jak určit pořadí v jakém se mají jednotlivé požadavky zpracovávat, jelikož při volbě výtahu, který požadavek obslouží, tato rozhodnutí neukládá do plánu, který by pořadí definoval. Volba výtahu, který požadavek obslouží, je dána tím, který výtah je schopen jej obsloužit nejdříve. Jsou-li ve výtahu cestující je směr pohybu výtahu jednoznačně určen směrem, kterým chtějí jet a má možnost, má-li volné místo zastavit cestou v dalších poschodích, kde čekají lidé chtějící jet stejným směrem. Je-li ovšem výtah prázdný, je zapotřebí určit, který požadavek má obsloužit s ohledem na neporušení pravidla, že požadavek obsluhuje výtah, který jej obslouží nejrychleji.

Nejjednodušší možností volby dalšího požadavku k obslužení je zvolit ten, který výtah obslouží nejrychleji a neporuší při tom pravidlo, že je výtahem, který jej obslouží nejrychleji. Ačkoliv se tento způsob nabízí jakožto nejlépe odpovídající k tomuto algoritmu, jelikož stejně jako při výběru výtahu k požadavku funguje na principu volby nejrychlejšího, může být tento způsob značně nevhodný. Například byly-li by všechny výtahy v nejnižším poschodí a požadavky směřovaly dolů, pak by výtahy vždy obsloužily nejnižší z nich. Pro každý požadavek by tedy přijel samostatný výtah (jeden výtah postupně i pro více). Vyjel-li by ovšem výtah rovnou pro požadavek nejvýše, mohl by cestou dolů obsloužit všechny ostatní požadavky při této jediné cestě. Druhou nevýhodou je možnost, že některé požadavky na výtah budou muset čekat příliš dlouho, což se velmi nepříznivě projeví na kvadratickém průměrném času čekání na příjezd výtahu. Například, pokud by se větší množství lidí (takové, že je zapotřebí všech výtahů k jejich obsluze) pohybovalo mezi dolními poschodími budovy, nedošlo by k obslužení požadavku z horního poschodí budovy. Nezbyvá totiž žádný nevyužitý výtah, který by tento požadavek obsloužil, jelikož

vždy existuje nějaký bližší požadavek k obslužení a žádný cestující nechce jet do vyšších pater budovy, čímž by se tam nějaký výtah přiblížil.

Další možností volby dalšího požadavku k obslužení je tedy možnost, kdy výtah obslouží požadavek nejbližší ke své aktuální pozici ve směru od něj nebo naopak nejvzdálenější požadavek od jeho aktuální pozice ve směru k němu. Zde ovšem začíná být důležité vhodně odhadnout počty lidí opouštějící jednotlivá poschodí a kam směřují, aby nedocházelo k tomu, že lidé nemohou nastoupit, jelikož se již do výtahu nevejdou. Také je zapotřebí zvážit, zda jet až k nejvzdálenějšímu požadavku, jelikož v jeho blízkosti se může nalézat jiný výtah, který aktuálně nemůže tento požadavek obsloužit, ale v dohledné době se uvolní a mohl by převzít část zátěže. Z tohoto důvodu je pro tento způsob volby dalšího požadavku vhodné využívání heuristik, které pozměňují běžné chování tohoto způsobu a zajišťují lepší spolupráci jednotlivých výtahů.

K volbě dalšího požadavku k obslužení je možno využívat i principů jiných plánovacích algoritmů. Je ovšem zapotřebí vědět, jaké s sebou nesou komplikace a zda je lze zanedbat, nějakým vhodným způsobem omezit jejich vliv nebo dokonce zcela odstranit. Plánovacím algoritmem, který takto může být použit je Výtahový algoritmus (viz. 5.1 Výtahový algoritmus). Je volán, jestliže je výtah prázdný (po skončení obsluhy všech cestujících) a je zapotřebí určit, který požadavek má dále obsloužit. Z principu fungování Výtahového algoritmu vyplývá, že další požadavek, který má výtah obsloužit, bude první ve směru jeho dosavadního pohybu. Princip hlavního plánovacího algoritmu ovšem dává podmínku, že požadavek má obsloužit výtah který se k němu dostane nejrychleji. Složením těchto dvou principů vzniká pravidlo které určuje, že další požadavek, který má výtah obsloužit, bude první ve směru jeho dosavadního pohybu, ke kterému se právě zkoumaný výtah dostane nejrychleji ze všech výtahů. Tímto způsobem se rozšíří všechna pravidla Výtahového algoritmu o podmínku nejkratší doby obsluhy a takto vzniklá pravidla se použijí pro určení požadavku, který má výtah dále obsloužit. Nevýhodou Výtahového algoritmu je, že výtahy nespolupracují, ale konkurují si, což vede k neefektivnímu plánování. Díky podmínce na výběr výtahu podle nejkratšího času obsluhy ovšem problém s konkurencí výtahů odpadá. Využití Výtahového algoritmu také zabraňuje vzniku požadavků, které nejsou dlouhodobě vyřízeny z důvodu obsluhy požadavků soustředěných do určité oblasti budovy, jelikož výtah nejprve obslouží požadavky ve směru svého pohybu a teprve poté může změnit směr (což je nutné k zůstání v oblasti, kde se soustředilo větší množství požadavků).

5.5 Provoz směrem z přízemí (Up-peak)

Jedná se o specializovaný plánovací algoritmus nahrazující běžně použitý plánovací algoritmus v době kdy většina požadavků směřuje z přízemí budovy do ostatních pater (v ranních hodinách, kdy lidé přichází do práce), viz. Literatura [5].

Tento režim je poměrně nenáročný na plánování. Výtahy čekají v přízemí a při dostatečném naplnění nebo po uplynutí určité doby od nástupu prvního pasažéra odjíždí do nejbližšího cílového patra odkud pokračují dále, dokud nedojde k vyprázdnění výtahu. Jakmile jsou všichni pasažéři dopraveni, vrací se výtah zpět do přízemí, kde čeká na další cestující.

Tento režim ovšem nepočítá s jinými požadavky, než z přízemí do ostatních pater a je tedy nutno zajistit obslužení ostatních požadavků jiným způsobem. Například po dopravení všech lidí jedoucích z přízemí, nevracet výtah zpět do přízemí, pokud

existují jiné požadavky a do přízemí se vrátit až po jejich obslužení.

5.6 Provoz směrem do přízemí (Down-peak)

Jedná se o specializovaný režim, v němž může výtahový systém fungovat přepnutím plánovacího algoritmu na jiný k tomuto typu provozu uzpůsobený. Slouží v případě kdy většina požadavků směřuje do přízemí (v odpoledních hodinách, kdy lidé odchází z práce), viz. Literatura [5].

Existuje více různých plánovacích algoritmů pro tento režim, jelikož se jedná o komplikovanější problém, než v případě Up-peak režimu. Je totiž zapotřebí rozhodnout, v jakém patře mají výtahy čekat na požadavky a jak rozdělit, které požadavky vyřídí který výtah. Výtahy totiž nemají neomezenou kapacitu a je tedy zapotřebí sledovat kolik osob je ještě možno nabrat, než se výtah zaplní. Také se zde vyskytuje problém s požadavky jinými, než směřujícími do přízemí, které se musí řešit nějakým jiným způsobem.

Jedna z možností řešení tohoto režimu shromažďuje výtahy v nejvyšším patře budovy a když nastane nějaký požadavek směrem do přízemí, je vyslán výtah, který jej obsluží. Cestou do přízemí, pokud je ve výtahu ještě volné místo, vyzvedává další lidi směřující stejným směrem. Po uvolnění výtahu v přízemí se opět vrací do nejvyššího patra budovy a začíná další cyklus.

5.7 Seskupování podle oblasti

Tento plánovací algoritmus vychází z myšlenky, že čím méně zastávek bude muset výtah během své cesty učinit, tím kratší dobu mu bude tato cesta trvat a bude dříve možné jej využít k obsluze dalších požadavků, viz. Literatura [1, 4]. Nesmí ovšem toto činit na úkor počtu přepravených cestujících při jedné cestě výtahu. Snížení počtu zastávek, které musí výtah během své cesty učinit, lze tedy docílit pouze tak, že lidé budou nastupovat ve větším množství najednou a budou směřovat do stejných poschodí.

Jestliže má ovšem plánovací algoritmus nějakým způsobem rozdělovat lidi mezi jednotlivé výtahy, je nutné, aby měl jak lidem oznámit, do kterého výtahu mají nastoupit a především musí znát kolik lidí chce přepravit do jakého poschodí dříve, nežli nastoupí do samotných výtahů. Tento plánovací algoritmus tedy nelze využít u běžných výtahových systémů, kde je výtah přivoláván pouze volbou směru, jakým se nalézá poschodí, do kterého chce cestující přepravit.

Tento princip se výrazněji projeví teprve při vyšším zatížení výtahového systému, kdy se v jednom podlaží (typicky přízemí budovy) shromáždí více lidí a ti jsou rozdělováni mezi více výtahů, které je rozváží do podlaží, kam chtějí přepravit. Naopak, při běžném provozu, kdy je zátěž nižší a požadavky na výtah rovnoměrněji rozprostřeny v rámci budovy, nepřináší sdružování požadavků takové zlepšení, jelikož nedochází k naplnění výtahu a snížení počtu jejich zastávek je podstatně menší. Tento plánovací algoritmus je tedy vhodné nasadit v ranních hodinách, kdy směřuje větší množství lidí z přízemí budovy do ostatních poschodí (viz. 5.5 Provoz směrem z přízemí (Up-peak)) a dochází k jejich sdružování a možnosti výběru skupin se stejným cílovým podlažím, které jsou umístovány do jednotlivých výtahů. Zatím co po pomnutí této špičky je vhodnější zvolit jiný plánovací algoritmus, lépe uzpůsobený slabšímu provozu.

Jednotlivé požadavky jsou sdružovány podle poschodí, do kterého směřují a tyto

skupinky jsou dále umísťovány do jednotlivých výtahů. Z důvodu zkrácení cesty výtahu jsou do jednoho výtahu umísťovány skupinky s blízkými cílovými podlažími, díky čemuž přes celou budovu jede pouze jeden výtah a ostatním se postupně délka trasy, kterou musí ujet k obslužení všech pasažérů, zkracuje. Dojde k jakémusi rozdělení budovy na oblasti, do kterých jednotlivé výtahy jezdí. Rozsah těchto oblastí může být pevně určen nebo se může dynamicky měnit podle aktuální situace. Obdobně přidělení výtahů k jednotlivým oblastem může být pevné nebo dynamické. Jsou-li výtahy přiřazeny pevně k jednotlivým oblastem, je zapotřebí uzpůsobit rozsah jednotlivých oblastí jejich vzdálenosti od přízemí, jelikož výtahu v oblasti blíže k přízemí bude trvat jeden cyklus kratší dobu a tedy zvládne obsloužit více požadavků. Přiřadíme-li oblastem pravděpodobnost, že nový požadavek bude směřovat do podlaží z jejich rozsahu, bude se tato pravděpodobnost se vzdáleností od přízemí snižovat. Takto se sníží počet požadavků do vzdálenějších poschodí, čímž dojde k dorovnání delšího času potřebného na cestu do této oblasti. Při dynamickém přiřazování výtahu k oblastem, je zapotřebí určit oblast, do které bude v době příjezdu výtahu do přízemí, čekat nejvíce cestujících a přiřadit jej k této oblasti. Je ovšem zapotřebí odhadnout pomocí heuristiky, která oblast bude takto nejdříve zaplněna, jelikož již člověk, který přijde první a měl by jet tímto výtahem, musí být správně informován o výtahu, který jej obslouží. Pro dynamické přidělování výtahů k oblastem je tedy vhodnější přiřazovat jednotlivým požadavkům identifikaci cílové oblasti a u jednotlivých výtahů zobrazovat, do které oblasti směřuje.

Záleží, zda je člověk informován o výtahu, který jej obslouží, již při zadání poschodí, do kterého směřuje, což mu umožňuje se k tomuto výtahu připravit, ale omezuje možnosti plánovacího algoritmu, jelikož z důvodu následujících požadavků se může tato volba stát neoptimální. Druhou možností je sdělit lidem, do kterého výtahu mají nastoupit až v době příjezdu příslušného výtahu do poschodí, kde lidé čekají. Výhodou této možnosti je, že plánovací algoritmus může vhodněji rozdělit jednotlivé požadavky bez ohledu na pořadí jejich vzniku, ovšem lidé jsou nuceni neustále sledovat, do kterého výtahu mají nastoupit a nemají možnost se nějakým způsobem dopředu připravit.

Je-li o přiřazení výtahu, který nový požadavek obslouží rozhodnuto již při jeho vzniku, musí plánovací algoritmus odhadnout podle pravděpodobností s jakými jsou voleny jednotlivá cílová podlaží, kolik požadavků ještě vznikne, nežli zvolený výtah tuto skupinu požadavků obslouží a rezervovat podle tohoto odhadu místo pro nové požadavky. Správnost tohoto odhadu daného heuristikou vzniklou na základě sledování dění v budově nebo odhadem určuje míru zaplnění jednotlivých výtahů a tedy výslednou efektivitu plánovacího algoritmu. Je také nutné nějakým způsobem obsloužit požadavky, které by logicky měly spadat do nějakého výtahu, ale již by se do něj nevešly a to jejich převedením do jiného výtahu, který veze cestující do nejbližší oblasti a má ještě volnou kapacitu.

5.8 Kyvadlová doprava (*Sabbath service*)

Tato plánovací strategie se od výše popsaných liší tím, že nikterak nevyhodnocuje stav, v jakém se budova nalézá, ale pracuje neustále stejně. Využívá se v případech s vysokým počtem požadavků rovnoměrněji rozložených do většiny poschodí budovy (nerozlišuje se, zda se jedná o poschodí vzniku požadavku nebo poschodí, kam chce být člověk přepraven). Základní informace o fungování tohoto plánovacího algoritmu pochází z internetové encyklopedie Wikipedia, viz. Literatura [5].

Výtahy jezdí bez přestání mezi nejnižším a nejvyšším poschodím budovy a cestou zastavují ve všech poschodích. Při startu provozu jsou výtahy rozmístěny tak, aby do jednotlivých podlaží v jednom směru přijížděly výtahy v pravidelných rozestupech. Lidé mohou volně nastupovat i vystupovat, akorát musí trvale hlídat, zda výtah již nedorazil do poschodí, kam směřují a při nastupování, zda výtah jede směrem, který jim vyhovuje. Jelikož výtahy zastavují ve všech poschodích budovy a nezkontrolují její stav, je zbytečné, aby výtahový systém využívající tuto plánovací strategii, měl ovládací panely k přivolání výtahu nebo volbě cílového podlaží.

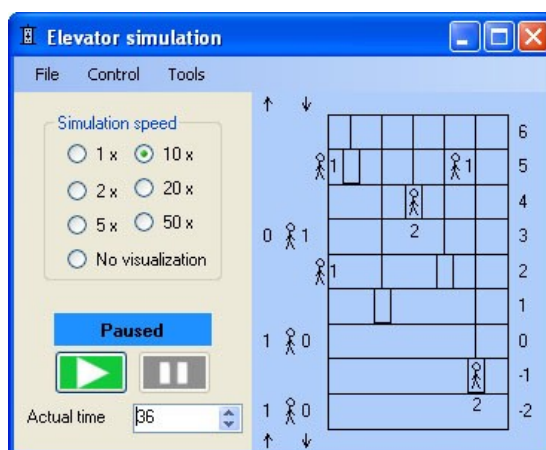
Je-li výtahový systém složen ze sudého počtu výtahů, lze je díky podmínce na pravidelnost zastávek výtahů v jednotlivých poschodích vždy rozdělit do párů, ve kterých je první výtah oproti druhému zpožděn o polovinu cyklu. Výtahy v tomto páru se pohybují vždy opačnými směry a ve chvíli, kdy je jeden z nich v nejvyšším poschodí, je druhý v nejnižším poschodí. Díky tomuto chování lze pro cestu vzhůru jednoho výtahu v páru využívat energii získanou klesáním druhého, zatímco druhý výtah bude naopak brzděn prvním výtahem. Podmínkou k využití tohoto způsobu šetření energie potřebné pro provoz výtahů je stejná vzdálenost mezi jednotlivými poschodími, aby při zastavení jednoho výtahu v podlaží nestál druhý výtah mezi podlažími a tedy nemohl otevřít dveře.

Výhodou této plánovací strategie je možnost lidí čekající v jednotlivých poschodích přesně informovat, kdy přijede výtah jejich směrem, který to bude a také doba cesty výtahem je předem odhadnutelná. Cestující tedy vědí za jak dlouho se dostanou do jimi požadovaného poschodí a mohou se rozhodnout, zda raději nejít po schodišti. Nevýhodou ovšem je, že výtahy zastavují i v podlažích, kde by jinak zastavovat nemuseli a doba cesty výtahem se tedy prodlouží. Výtahy se také nemohou pohybovat vyššími rychlostmi, jelikož mají poměrně krátkou dráhu na rozjezd a brzdění (vzdálenost mezi sousedními podlažími).

6 Simulační program

K testování jednotlivých plánovacích algoritmů byl použit simulační program vycházející z ročníkového projektu viz. Literatura [7] Simulátor výtahů, který byl rozšířen a upraven pro potřeby jednotlivých plánovacích algoritmů a jejich testování. Program běží ve dvou synchronizovaných vláknech a to grafickém mající za úkol zobrazování informací uživateli a výpočetním, které obstarává výpočet průběhu simulace. Jednotlivé plánovací algoritmy jsou obsaženy v dynamicky linkovaných knihovnách a načítány za běhu programu na žádost uživatele.

6.1 Hlavní okno



Obrázek 1 - Hlavní okno

V levé části se nachází ovládací prvky řídící grafické znázornění průběhu simulace. Rámeček označený *Simulation speed* slouží k ovládní rychlosti průběhu simulace a to v násobcích reálné rychlosti (1 sekunda zobrazování odpovídá 1 sekundě simulace). Jedinou výjimkou je volba *No visualization*, která dokončí simulaci bez zobrazování jejího průběhu nejvyšší možnou rychlostí. Je-li tato volba aktivní v průběhu simulace, jsou ostatní volby rychlosti uzamčeny.

Dvě ovládací tlačítka se standardními symboly pro přehrání a pozastavení slouží ke spuštění simulace, jejímu pozastavení a pokračování, byla-li nejprve pozastavena. Nad těmito tlačítky je vypisován aktuální stav, ve kterém se simulace nalézá (neodstartována, probíhá, pozastavena, dokončena).

Pole označené *Actual time* je přístupné pouze je-li simulace pozastavena, jinak zobrazuje čas, ve kterém se simulace nalézá. V době, kdy je simulace pozastavena slouží toto pole k posunu v čase simulace. Změnou hodnoty dojde k posunu simulace do určeného času, což je přímo viditelné v pravé části *Hlavního okna*. Krok, po kterém lze čas nastavovat závisí na nastavení rychlosti simulace tak, že při rychlostech větších než 10x, se krok prodlužuje.

V pravé části *Hlavního okna* je animován průběh simulace. Jelikož při velikém počtu poschodí a menší velikosti *Hlavního okna* by bylo zapotřebí vykreslovat postavy velice malé, jsou postavy od určité velikosti vykreslovány pouze jakožto červené elipsy. Z důvodu přehlednosti zobrazení stavu budovy není také každá osoba vykreslována zvlášť, ale jsou sdružovány do skupin podle umístění a je zobrazován počet lidí, který tato skupina představuje.

6.2 Menu hlavního okna

File / New simulation – slouží ke spuštění simulace, pokud již není spuštěna

File / Data source – zobrazí dialog pro nastavení vstupních dat s požadavky na výtah (viz. kapitola 6.3 Okno: Data source)

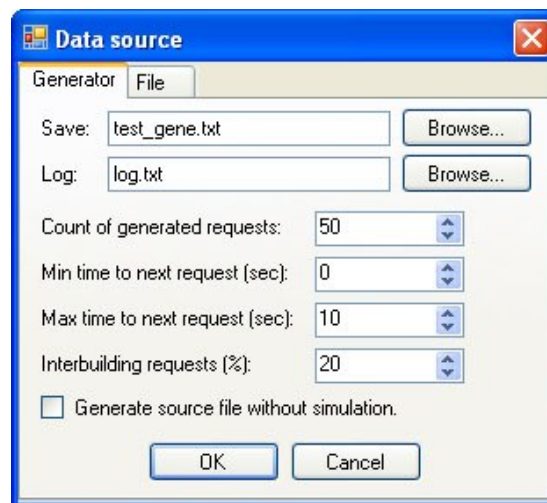
File / Exit – slouží k ukončení programu

Control – obsahuje příkazy duplicitní k ovládacím prvkům umístěným v levé části *Hlavního okna*

Tools / Open algorithm – zobrazí dialog umožňující výběr .dll souboru obsahujícího plánovací algoritmus, který bude použit pro plánování nově odstartované simulace

Tools / Settings – zobrazí dialog pro nastavení parametrů ovlivňujících simulaci, jako počet poschodí budovy a další (viz. kapitola 6.4 Okno: Settings)

6.3 Okno: Data source



Obrázek 2 - Data source (Generator)

Toto okno slouží k nastavení zdroje jednotlivých požadavků na výtah. Okno obsahuje dvě záložky *Generator* a *File*, které určují, odkud se mají data získávat. Volbou *OK* jsou provedená nastavení aplikována, naopak uzavřením okna tlačítkem *Cancel* nebo křížkem v horním pravém rohu jsou veškeré změny anulovány.

Je-li vybrána záložka *Generator*, budou požadavky získávány z generátoru umístěného v simulačním programu. Tento generátor generuje požadavky oběma směry se stejnou pravděpodobností a lze provést základní nastavení pomocí zobrazených ovládacích prvků.

Save – určuje cestu k souboru, kam se budou průběžně ukládat vygenerované hodnoty. Výsledný soubor má stejný formát jako soubor se vstupními daty a lze jej tedy použít jakožto vstupní soubor při získávání dat ze souboru. (lze zadat pomocí sousedního tlačítka *Browse* prostřednictvím dialogu pro ukládání souborů)

Log – určuje cestu k souboru, kam se bude ukládat textová podoba průběhu simulace (lze zadat pomocí sousedního tlačítka *Browse* prostřednictvím dialogu pro ukládání souborů)

Count of generated requests – určuje počet lidí který bude vygenerován

- Min time to next request (sec)* – určuje nejkratší dobu, za jakou může vzniknout další požadavek na výtah
- Max time to next request (sec)* – určuje dobu, za jakou nejpozději vznikne další požadavek na výtah
- Interbuilding requests* – určuje kolik procent požadavků nezahrnuje přízemí budovy, jakožto zdrojové ani cílové poschodí
- Generate source file without simulation* – je-li zaškrtnuto, nebude probíhat simulace, ale pouze se vygeneruje soubor, který lze využít jakožto vstupní soubor při získávání dat ze souboru



Obrázek 3 - Data source (File)

Je-li vybrána záložka *File*, budou požadavky získávány ze zadaného souboru.

Load – určuje cestu k souboru, ze kterého se budou získávat požadavky pro běh simulace (lze zadat pomocí sousedního tlačítka *Browse* prostřednictvím dialogu pro otevírání souborů)

Log – určuje cestu k souboru, kam se bude ukládat textová podoba průběhu simulace (lze zadat pomocí sousedního tlačítka *Browse* prostřednictvím dialogu pro ukládání souborů)

6.4 Formát souboru se vstupními daty

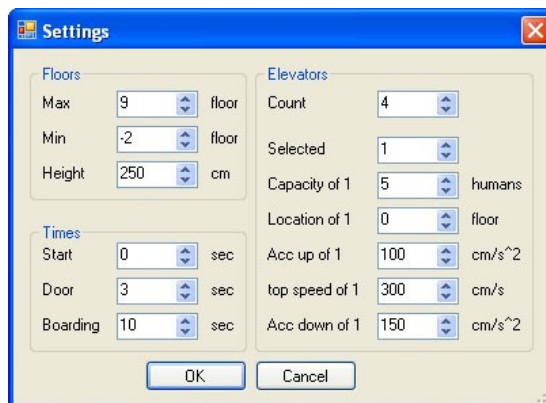
time	from	to
12	5	1
17	2	4

Text 1 - Formát datového souboru

Soubor je uvozen jednořádkovou hlavičkou popisující obsah souboru, která je programem ignorována. Každý požadavek je reprezentován samostatným řádkem skládajícím se ze trojice hodnot. První hodnota udává čas vzniku požadavku v sekundách, druhá hodnota určuje podlaží vzniku požadavku a poslední hodnota určuje podlaží, do kterého chce člověk přepravit. Jednotlivé požadavky musí být vzestupně seřazeny podle času jejich vzniku.

Na příkladě viz. Text 1 je vidět soubor obsahující dva požadavky. Druhý řádek udává, že v čase 12 sekund přijde člověk do 5. patra a bude chtít do 1. patra. Obdobně třetí řádek říká, že v čase 17 sekund přijde člověk do 2. patra a bude chtít do 4. patra.

6.5 Okno: Settings



Obrázek 4 - Settings

Toto okno slouží k nastavení vlastností budovy a výtahového systému, na němž bude probíhat simulace. Ovládací prvky jsou rozděleny do tří skupin podle povahy vlastností, které nastavují. Skupina *Floors* se týká budovy, *Times* časů potřebných pro různé události a *Elevators* jednotlivých výtahů obsažených ve výtahovém systému. Volbou *OK* jsou provedená nastavení aplikována, naopak uzavřením okna tlačítkem *Cancel* nebo křížkem v horním pravém rohu jsou veškeré změny anulovány.

Floors / Max – udává číslo nejvyššího podlaží v budově

Floors / Min – udává číslo nejnižšího podlaží v budově

Floors / Height – udává vzdálenost dvou sousedních podlaží (v centimetrech)

Times / Start – udává čas, ve kterém simulace začíná (v sekundách)

Times / Door – udává čas potřebný na otevření nebo uzavření bezpečnostních dveří (v sekundách)

Times / Boarding – udává čas, po který jsou dveře otevřené, aby mohli lidé nastupovat a vystupovat (v sekundách)

Elevators / Count – udává počet výtahů v budově (při zvýšení počtu výtahů přebírají nové výtahy nastavení aktuálně vybraného výtahu podle položky *Selected*)

Elevators / Selected – slouží k přepínání mezi jednotlivými výtahy, aby bylo možné každý nastavit samostatně

Elevators / Capacity of x – udává počet lidí, který je schopen výtah najednou pojmout (*x* je číslo výtahu)

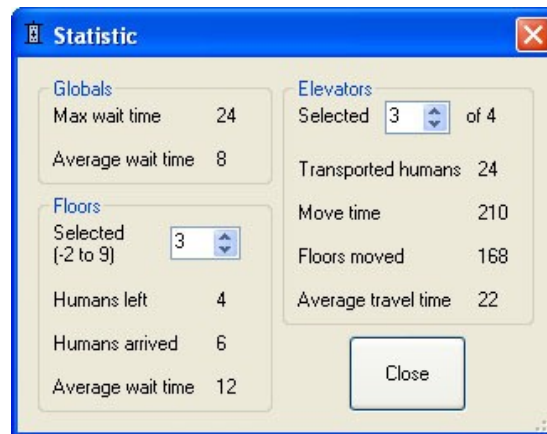
Elevators / Location of x – udává poschodí, kde se výtah nalézá na začátku simulace (*x* je číslo výtahu)

Elevators / Acc up of x – udává zrychlení výtahu směrem nahoru (*x* je číslo výtahu)

Elevators / top speed of x – udává maximální rychlost jakou je výtah schopen se pohybovat (*x* je číslo výtahu)

Elevators / Acc downv of x – udává zrychlení výtahu směrem dolů (*x* je číslo výtahu)

6.6 Okno: *Statistic*



Obrázek 5 - *Statistic*

Toto okno slouží k zobrazení informací o průběhu simulace a je automaticky zobrazeno po skončení simulace. Informace jsou rozděleny do tří skupin podle povahy informace, kterou udávají. Skupina *Globals* udává souhrnné informace o časech čekání za celou budovu, *Floors* udává informace o jednotlivých podlažích a *Elevators* o jednotlivých výtazích. Okno lze uzavřít tlačítkem *Close* nebo křížkem v horním pravém rohu okna.

Globals / Max wait time – udává nejdelší dobu čekání na příjezd výtahu za všechna poschodí budovy (v sekundách)

Globals / Average wait time – udává průměrnou dobu čekání na příjezd výtahu za všechna poschodí budovy (v sekundách)

Floors / Selected – slouží k výběru poschodí, o kterém budou zobrazené informace ve skupině *Floors*

Floors / Humans left – udává počet lidí, kteří opustili zvolené poschodí

Floors / Humans arrived – udává počet lidí, kteří přijeli do zvoleného poschodí

Floors / Average wait time – udává jak průměrně dlouho lidé čekali na příjezd výtahu ve zvoleném podlaží (v sekundách)

Elevators / Selected – slouží k výběru výtahu, o kterém budou zobrazené informace ve skupině *Elevators*

Elevators / Transported humans – udává počet cestujících přepravených zvoleným výtahem

Elevators / Move time – udává dobu, po jakou byl zvolený výtah v pohybu (v sekundách)

Elevators / Floors moved – udává kolik poschodí zvolený výtah celkem urazil, bez ohledu na směr pohybu

Elevators / Average travel time – udává průměrnou dobu jízdy jednoho člověka ve zvoleném výtahu (v sekundách)

6.7 *FIFO_algorithm.dll*

Tato knihovna obsahuje plánovací algoritmus popsany v kapitole 5.2 Vyřízení požadavků dle pořadí jejich vzniku.

Jsou zde implementovány tyto vylepšení oproti standardnímu chování fronty:

- Přibírání nových cestujících do již jedoucího výtahu, chtějí-li jet ve vhodném

- směru a pokud je výtah schopen dobrzdit do podlaží jejich vzniku.
- Požadavky, které jsou odsunuty k pozdějšímu vyřízení z důvodu obslužení požadavků cestou výtahu, jsou navraceny na začátek fronty všech čekajících požadavků. Jelikož v době jejich vybrání z fronty byly na jejím začátku a tedy první v pořadí k vyřízení.
- Pro obsluhu nového požadavku je z výtahů, které jej mohou obsloužit, vybrán výtah, který to má nejbliže. Tato volba zvyšuje efektivnost plánovacího algoritmu a také šetří energii potřebnou k provozu výtahového systému.

6.8 nearest_algorithm.dll

Tato knihovna obsahuje plánovací algoritmus popsany v kapitole 5.3 Obsluha výtahem, který může přijet nejrychleji (bez změn volby již rozhodnutých přiřazení výtahů).

Jsou zde implementovány tyto vlastnosti:

- K přiřazení výtahu, který požadavek obslouží, dochází již při jeho vzniku a dále se nemění.
- Pro každý nový požadavek je spočtena doba za jakou jsou jednotlivé výtahy schopny jej obsloužit a zvolen nejrychlejší výtah.
- Je-li požadavek v opačném směru, nežli v jakém se výtah aktuálně pohybuje, je čas, za jaký jej je výtah schopen obsloužit, počítán jako by měl výtah jet až do krajního poschodí budovy. Toto rozhodnutí je z důvodu, že cestou může výtah přijmout další požadavky v jeho aktuálním směru, které budou mít přednost a mohou směřovat maximálně do jednoho z krajních poschodí budovy.
- Je kontrolováno, zda se do výtahu vešli všichni lidé čekající v podlaží odkud odjíždí. Jestliže v podlaží někdo zůstal, je automaticky obnoven jeho požadavek na výtah.

6.9 nearest_recalc_algorithm.dll

Tato knihovna obsahuje plánovací algoritmus popsany v kapitole 5.4 Obsluha výtahem, který může přijet nejrychleji (průběžně měnící volbu přiřazeného výtahu).

Jsou zde implementovány tyto vlastnosti:

- Při vzniku nového požadavku jsou zjištěny všechny výtahy, které jej mohou obsloužit bez zastávek v jiných podlažích a z nich je vybrán výtah, který se vzhledem ke svému aktuálnímu stavu dostane k požadavku nejrychleji. Neexistuje-li žádný takovýto výtah, je požadavek uložen k pozdějšímu vyřízení.
- Při vyřizování uloženého požadavku je k jeho obsluze určen výtah, který jej může obsloužit bez zastávek v jiných poschodích a dostane se k němu nejrychleji.
- Jsou-li ve výtahu nějací pasažéři, výtah jede přednostně do pochodí, kam směřují. Cestou zastavuje, pouze má-li volnou kapacitu, v poschodích, kde někdo čeká ve směru jakým se aktuálně pohybuje a byl-li určen k jejich obsluze.
- Při určování podlaží, do kterého se má výtah přesunout, jsou zkoumána jednotlivá poschodí budovy, na něž směřuje některý z pasažérů výtahu nebo někdo čeká na příjezd výtahu. Vybráno je poschodí, do kterého se aktuálně

zkoumaný výtah dostane nejrychleji a byl určen k obsluze požadavků na tomto poschodí.

6.10 *Elevator_algorithm.dll*

Tato knihovna obsahuje plánovací algoritmus popsáný v kapitole 5.1 Výtahový algoritmus implementovaný dle pravidla popsáného v posledním odstavci kapitoly 5.4 Obsluha výtahem, který může přijet nejrychleji (průběžně měnící volbu přiřazeného výtahu) jakožto algoritmus pro určení dalšího požadavku k obslužení. Implementován je přidáním principů Výtahového algoritmu do knihovny 6.9 `nearest_recalc_algorithm.dll`.

7 Srovnání jednotlivých plánovacích algoritmů

Jednotlivé plánovací algoritmy je možno srovnávat podle různých kritérií a měřit je různými způsoby. V této práci jsou jednotlivé plánovací algoritmy hodnoceny na základě hodnot získaných jejich během v simulačním programu *Simulátor výtahů* (viz. kapitola 6 Simulační program). Simulátoru jsou nastavovány různé parametry budovy (počet poschodí, pořadí v jakém vznikají požadavky na výtah, ...) a výtahového systému (počet, rychlosti, kapacita a počáteční umístění jednotlivých výtahů). Pro každé nastavení je vygenerován soubor se vstupními daty obsahující pořadí požadavků na výtah, na němž je simulace postupně spuštěna pro všechny plánovací algoritmy implementované v dynamických knihovnách. Pro každou kombinaci nastavení a plánovacího algoritmu jsou zjištěny výsledné hodnoty shrnující průběh simulace zobrazené po jejím skončení.

Není-li u testovaného nastavení řečeno jinak, jsou používány tyto hodnoty nastavení simulačního programu:

Rozsah podlaží: -2 ~ 20	Počet výtahů: 4
Výška podlaží: 250 cm	Kapacita výtahů: 5 lidí
Doba otevírání dveří výtahu: 3 sec	Zrychlení směrem nahoru: 100 cm/sec ²
Doba pro nastupování: 10 sec	Maximální rychlost: 300 cm/sec
Rozestup požadavků: 0 ~ 20 sec	Zrychlení směrem dolů: 150 cm/sec ²
Počet vygenerovaných požadavků: 200	
Šance že požadavek bude směřovat z/do přízemí budovy: 80%	

Použité zkratky v tabulkách a textu dále v této kapitole:

Výtahový – plánovací algoritmus popsáný v kapitole 6.10 Elevator_algorithm.dll

FIFO – plánovací algoritmus popsáný v kapitole 6.7 FIFO_algorithm.dll

Nejrychlejší (s plánem) – plánovací algoritmus popsáný v kapitole 6.8 nearest_algorithm.dll

Nejrychlejší (přepočít) – plánovací algoritmus popsáný v kapitole 6.9 nearest_recalc_algorithm.dll

prům. čas – průměrný čas čekání lidí na příjezd výtahu (v sekundách)

max. čas – nejdelší čas po jaký někdo čekal na příjezd výtahu (v sekundách)

doba pohybu – doba po jakou byl průměrně jeden výtah v pohybu při obsluze požadavků v průběhu simulace (v sekundách)

ujetých podlaží – počet podlaží, které průměrně ujel jeden výtah při obsluze požadavků v průběhu simulace

7.1 Standardní budova

Použitý soubor se vstupními daty (obsažen na CD): simulation_1.txt

Změněné nastavení simulačního programu:

Rozsah podlaží: -2 ~ 15

	Výtahový	FIFO	Nejrychlejší (s plánem)	Nejrychlejší (přepočten)
prům. čas	16	18	18	19
max. čas	107	89	187	467
doba pohybu	536	526	490	508
ujetých podlaží	467	457	428	437

Tabulka 1: Simulace 1

Vzhledem k *prům. času* vychází nejlépe *Výtahový algoritmus*, jelikož rovnoměrně obsluhuje jednotlivé požadavky, díky čemuž také udržuje i nízký *max. čas*. Nejkratší *max. čas* má ovšem *FIFO*, jelikož výtahový systém není příliš zatížen a dochází k pravidelnému zpracovávání požadavků z fronty podle času jejich vzniku.

V této simulaci dopadly algoritmy minimalizující dobu potřebnou k obsluze následujícího požadavku poměrně špatně, jelikož dlouhodobě odkládaly požadavky ze suterénu budovy. K tomuto odkládání docházelo, jelikož při volbě dalšího požadavku k obsluze po vystoupení všech lidí z výtahu, již někdo čekal v přízemí budovy a výtah nacházející se v některém z vyšších poschodí byl schopen obsloužit požadavek z přízemí rychleji, nežli ze suterénu. O něco lépe dopadl *Nejrychlejší (s plánem)*, který si tvoří pro jednotlivé výtahy plán požadavků, které má vyřídit, díky čemuž nemohly být požadavky ze suterénu budovy dlouhodobě odsouvány.

7.2 Zvýšený provoz v rámci budovy

Použitý soubor se vstupními daty (obsažen na CD): simulation_2.txt

Změněné nastavení simulačního programu:

Rozsah podlaží: -2 ~ 15

Šance, že požadavek bude směřovat z/do přízemí budovy: 40%

	Výtahový	FIFO	Nejrychlejší (s plánem)	Nejrychlejší (přepočten)
prům. čas	17	20	23	18
max. čas	95	84	127	146
doba pohybu	554	535	501	491
ujetých podlaží	467	453	424	407

Tabulka 2: Simulace 2

Tato simulace byla provedena za účelem zjištění, jak se u jednotlivých plánovacích algoritmů projeví, když nebude většina lidí přicházet nebo opouštět budovu, ale zvýší se počet lidí, kteří se pohybují uvnitř budovy.

Při srovnání se simulací provedenou na budově se stejnými parametry a stejným výtahovým systémem, ale větší zátěží v přízemí (viz. 7.1 Standardní budova) se u všech plánovacích algoritmů, s výjimkou *Nejrychlejšího (přepočten)*, projevil nárůst *prům. času*. Tento nárůst je dán vyšší měrou rozprostření požadavků po budově, což má za následek, že výtahy nejsou tolik zaplněny a musí častěji zastavovat, kvůli čemuž se sníží jejich efektivita. Tomuto efektu se *Nejrychlejší (přepočten)* vyhnul, jelikož při zastavení výtahu hledá nejbližší možné poschodí, jakožto další zastávku a neřídí se plánem, který by jej nutil jezdit větší vzdálenosti. U všech plánovacích algoritmů také došlo ke snížení *max. času*, jelikož výtahy jsou volnější (častěji nikoho nevezou), což jim umožňuje obsloužit požadavky, které by jinak musely nadále čekat.

7.3 Standardní budova bez suterénu

Použitý soubor se vstupními daty (obsažen na CD): simulation_3.txt

Změněné nastavení simulačního programu:

Rozsah podlaží: 0 ~ 20

	Výtahový	FIFO	Nejrychlejší (s plánem)	Nejrychlejší (přepočten)
prům. čas	20	17	18	16
max. čas	144	138	123	107
doba pohybu	651	632	572	581
ujetých podlaží	593	579	520	524

Tabulka 3: Simulace 3

Jelikož algoritmy minimalizující dobu potřebnou k obsluze následujícího požadavku si uměly špatně poradit s požadavky směřujícími ze suterénu budovy (důvod viz. kapitola 7.1 Standardní budova), byla provedena simulace na budově bez suterénu. Podle předpokladu měla tato změna výrazný vliv na výsledek simulace.

Při srovnání plánovacích algoritmů podle *prům. času* i *max. času* vychází nejlépe *Nejrychlejší (přepočten)*. *Prům. čas* vychází nejkratší, jelikož výtahy vždy jedou obsloužit nejbližší požadavek, díky čemuž obslouží více požadavků za stejnou dobu, ale také je nezbytné, aby byly výtahy pravidelně rovnoměrně rozmíst'ovány v rámci budovy. K tomuto rozmíst'ování dochází díky obsluze požadavků směřujících z přízemí budovy a to do všech oblastí budovy, jelikož tato budova nemá suterén, do kterého směřovalo málo požadavků. *Max. čas* vychází nejkratší, především díky konkrétnímu pořadí, v jakém byly vygenerovány jednotlivé požadavky pro tuto simulaci, ale také díky rovnoměrné obsluze požadavků v rámci celé budovy.

Ke zhoršení všech hodnot naopak došlo u *Výtahového algoritmu*, které ovšem není způsobeno nepřítomností suterénu, ale zvýšením počtu poschodí v budově a tím prodloužením doby potřebné na jednu cestu výtahu.

7.4 Mrakodrap

Použitý soubor se vstupními daty (obsažen na CD): simulation_4.txt

Změněné nastavení simulačního programu:

Rozsah podlaží: 0 ~ 100

Počet výtahů: 10

Počet vygenerovaných požadavků: 500

Rozestup požadavků: 0 ~ 15 sec

	Výtahový	FIFO	Nejrychlejší (s plánem)	Nejrychlejší (přepočten)
prům. čas	26	28	20	20
max. čas	191	307	267	117
doba pohybu	2133	2119	1688	1905
ujetých podlaží	2351	2356	1854	2085

Tabulka 4: Simulace 4

Jedná se pouze o teoretickou simulaci, jelikož z technických důvodů se takovýto výtahový systém nedělá. Účelem této simulace bylo prověřit chování jednotlivých plánovacích algoritmů v budově s velkým počtem poschodí a větším množstvím výtahů k jejich obsluze.

Z důvodu velikého počtu poschodí nejlépe v této simulaci dopadly algoritmy minimalizující dobu potřebnou k obsluze následujícího požadavku, jelikož čas potřebný k dosažení cílového poschodí je ve srovnání s časem potřebným k nastoupení a vystoupení lidí v podlaží mnohem podstatnější, nežli u menších budov. Plánovací algoritmy, které neberou v potaz vzdálenost dalšího požadavku od aktuální pozice výtahu, spotřebují zbytečně mnoho času na přesun prázdných výtahů, což snižuje jejich efektivnost.

7.5 Přetížený výtahový systém

Použitý soubor se vstupními daty (obsažen na CD): simulation_5-6.txt

Změněné nastavení simulačního programu:

Rozsah podlaží: 0 ~ 20

Rozestup požadavků: 0 ~ 10 sec

	Výtahový	FIFO	Nejrychlejší (s plánem)	Nejrychlejší (přepočten)
prům. čas	59	78	68	79
max. čas	296	697	366	465
doba pohybu	256	266	244	249
ujetých podlaží	207	217	190	190

Tabulka 5: Simulace 5

Tato simulace byla provedena za účelem testování chování jednotlivých plánovacích algoritmů při vysoké zátěži výtahového systému. Z naměřených hodnot je patrné, že testovaný výtahový systém je pro budovu s takovou zátěží nedostatečný, jelikož doby čekání na příjezd výtahu jsou příliš vysoké.

Při takto silném zatížení vykazuje nejlepší výsledky *Výtahový algoritmus*, jelikož jednotlivé výtahy rovnoměrněji projíždí celou budovou a nevznikají oblasti, kde by se hromadilo větší množství čekajících lidí. Na první pohled by se mohlo zdát zvláštní, že největší *max. čas* má *FIFO*, ale toto je dáno tím, že výtahy jsou trvale zaplněny a plánovací algoritmus nemá možnost zpracovávat požadavky z fronty podle pořadí jejich vzniku.

7.6 Přetížený výtahový systém s většími výtahy

Použitý soubor se vstupními daty (obsažen na CD): simulation_5-6.txt

Změněné nastavení simulačního programu:

Rozsah podlaží: 0 ~ 20

Kapacita výtahů: 5 lidí

Rozestup požadavků: 0 ~ 10 sec

	Výtahový	FIFO	Nejrychlejší (s plánem)	Nejrychlejší (přepočten)
prům. čas	60	60	48	49
max. čas	403	311	218	202
doba pohybu	245	247	218	247
ujetých podlaží	193	199	165	195

Tabulka 6: Simulace 6

Tato simulace zkoumá vliv zvýšení kapacity výtahů při přetížení výtahového systému. Zachovává stejný seznam požadavků na výtah, nastavení budovy i

výtahového systému jako při simulaci viz. kapitola 7.5 Přetížený výtahový systém. Jedinou výjimku tvoří kapacita jednotlivých výtahů, která je zvýšena.

Ze srovnání se simulací viz. kapitola 7.5 Přetížený výtahový systém vyplývá, že *Výtahový algoritmus* není zvětšením kapacity výtahů téměř ovlivněn, jelikož požadavky jsou shromažďovány jedním výtahem, zatímco ostatní jezdí prázdnější a nárůst *max. času* je dán především konkrétním pořadím vyřizování a vzniku nových požadavků. Výrazné zlepšení lze ovšem pozorovat u ostatních plánovacích algoritmů, kde vlivem zvýšení kapacity výtahů dochází k podstatnému zrychlení obsluhy jednotlivých požadavků. Hlavním důvodem tohoto zrychlení je, že výtahy zvládnou obsloužit více požadavků najednou, což šetří cesty ostatním výtahům.

7.7 Zhodnocení výsledků měření

Z jednotlivých měření vyplývá, že *max. čas* může být zavádějící, jelikož velmi záleží na konkrétním pořadí vygenerovaných požadavků a tedy se příliš nehodí k porovnávání jednotlivých plánovacích algoritmů. Je možno k němu přihlížet jakožto k informativní hodnotě, ale pro srovnání různých plánovacích algoritmů je zapotřebí jiného srovnávacího kritéria.

Má-li být výtahový systém optimalizován podle spotřeby energie, lze z naměřených hodnot *ujetých podlaží* a *doba pohybu* říci, že nejvýhodněji vychází plánovací algoritmus *Nejrychlejší (s plánem)* a *Nejrychlejší (přepočet)* se mu velmi blíží. Tento závěr bylo možno očekávat, jelikož tyto dva plánovací algoritmy vychází z minimalizace ujeté dráhy, což má za následek zkrácení doby, po jakou musí být pohony výtahů v činnosti a tím také šetří energii potřebnou pro fungování výtahového systému.

Typicky je ovšem výtahový systém optimalizován dle času čekání lidí na příjezd výtahu, což je v Tabulkách 1-6 shrnujících průběh provedených simulací zachyceno hodnotou *prům. čas*. Jejich porovnáním a určením, jak který z testovaných plánovacích algoritmů dopadl v jednotlivých provedených simulacích vychází nejhůře *FIFO*. Tento fakt ukazuje, že *FIFO* není příliš vhodným kandidátem při výběru plánovacího algoritmu pro nový výtahový systém. Žádný z testovaných plánovacích algoritmů ale také nevychází univerzálně nejvýhodněji a vždy tedy záleží na parametrech budovy, pro kterou je výtahový systém koncipován.

Jakožto nejvhodnější pro nový výtahový systém navrhovaný pro běžnou budovu se zdá být plánovací algoritmus *Nejrychlejší (přepočet)*, jelikož tvoří kompromis mezi úsporou energie potřebné k provozu výtahového systému a dobou čekání lidí na příjezd výtahu.

Literatura

- [1] http://www.otis.com/corp/flash/8_18_06/compass.html
- [2] <http://www.otis.com/corp/k2-elevators.html>
- [3] <http://www.otisgen2.com/>
- [4] http://www.us.schindler.com/sec_index/sec_tech/sec_tech_destination/sec_tech_m10_time-2.htm
- [5] Wikipedia (20.4.2007):
http://en.wikipedia.org/wiki/Elevator#Controlling_elevators
- [6] Wikipedia (3.4.2007):
http://en.wikipedia.org/wiki/Discrete_event_simulation
- [7] Michal Bartoš (2006/2007): Ročníkový projekt Simulátor výtahů

Použitý software

- Microsoft® Visual Studio® 2005
- OpenOffice.org 2.2.0

Obsah příloženého CD

- Elektronická verze této práce ve formátu PDF
- instalační balík simulačního programu obsahující i knihovny s algoritmy
- simulační program přeložený pro .NET 2.0
- přeložené knihovny s jednotlivými plánovacími algoritmy
- soubory se vstupními daty obsahující požadavky na výtah použité při simulacích viz. kapitoly 7.1 – 7.6
- zdrojové kódy v jazyce C++ pro simulační program a jednotlivé knihovny s plánovacími algoritmy
- projekt pro Microsoft® Visual Studio® 2005 zahrnující simulační program a jednotlivé knihovny s plánovacími algoritmy