

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

OPTIMALIZACE EXPEDICE ZBOŽÍ Z REGÁLOVÉHO SYSTÉMU LOGISTICKÉHO SKLADU

OPTIMIZATION OF DISPATCH OF GOODS FROM RACK SYSTEM OF LOGISTICS
WAREHOUSE.

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JIŘÍ CHLUP

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. Ing. MILOŠ ŠEDA, Ph.D.

BRNO 2009

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2008/2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Jiří Chlup

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Optimalizace expedice zboží z regálového systému logistického skladu

v anglickém jazyce:

Optimization of dispatch of goods from rack system of logistics warehouse.

Stručná charakteristika problematiky úkolu:

Logistická společnost skladuje na ploše 8000 m² převážně v regálových systémech zboží různých obchodních nebo výrobních firem (svých klientů). Klienti, zasílají denně objednávky k distribuci zboží svým odběratelům dle aktuálního prodeje. Objednané zásilky k distribuci mají být expedovány na oddělení dopravy v objednaný den do 17:30 hod.. Rychlost zpracování objednávek a optimální využití zdrojů (manipulační technika, lidé, čas) je stěžejní pro efektivní zvládnutí procesu expedice zboží z regálových systémů.

Cíle diplomové práce:

1. Analyzovat stávající systém expedice objednávek, využití zdrojů a tvorby plánu.
2. Navrhnout a implementovat prakticky použitelný systém, který bude vykazovat vyšší efektivitu expedice s použitím stávajících zdrojů.
3. Provést a vyhodnotit srovnávací testy.

Seznam odborné literatury:

1. Michalewicz, Z. and Fogel, D.B.: How to Solve It. Modern Heuristics. Springer-Verlag, Berlin 2000.
2. Barbucha, D. and Jedrzejowicz, P.: An Agent-Based Approach to Vehicle Routing Problem International Journal of Applied Mathematics and Computer Science, vol. 4, no. 1, 2007, pp. 19-23.

Vedoucí diplomové práce: doc. RNDr. Ing. Miloš Šeda, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2008/2009.

V Brně, dne

L.S.

doc. RNDr. Ing. Miloš Šeda, Ph.D.
Ředitel ústavu

doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

ABSTRAKT

Tato práce řeší problematiku optimalizace expedice zboží z regálového systému logistického skladu. Jedná se o řešení úlohy z praxe, které optimalizuje trasu průchodu regálovým systémem a ohodnocuje jednotlivé objednávky pro optimální tvorbu expediční dávky. Na počátku je analyzován stávající systém expedice, využití zdrojů a tvorby časového plánu. Následuje popis obecných praktických řešení hledání nejkratší cesty průchodu grafem přes zadané uzly. Poté je představeno zvolené řešení a jeho praktická implementace. Závěrem jsou provedeny srovnávací testy se stávajícím systémem a jejich vyhodnocení.

ABSTRACT

The diploma thesis submitted solves problems of optimization of goods expedition from rack system of a logistic store. It concerns solution of a practical task which optimizes a passage route through the rack system and evaluates individual orders for creation of optimal expedition dispatching batch. The existing dispatching system, use of sources and creation of time schedule is being described at the beginning. Description of general practical solutions of seeking the shortest way through the graph through appointed nodes follows. Chosen solution and its practical implementation are introduced afterwards. Comparative tests with existing system and their evaluations are mentioned at the end of the thesis.

KLÍČOVÁ SLOVA

TSP , algoritmus, graf, cesta, logistika, regál, expedice, objekt.

KEYWORDS

TSP, algorithm, graph, path, logistics, rack, expedition, object.

Poděkování

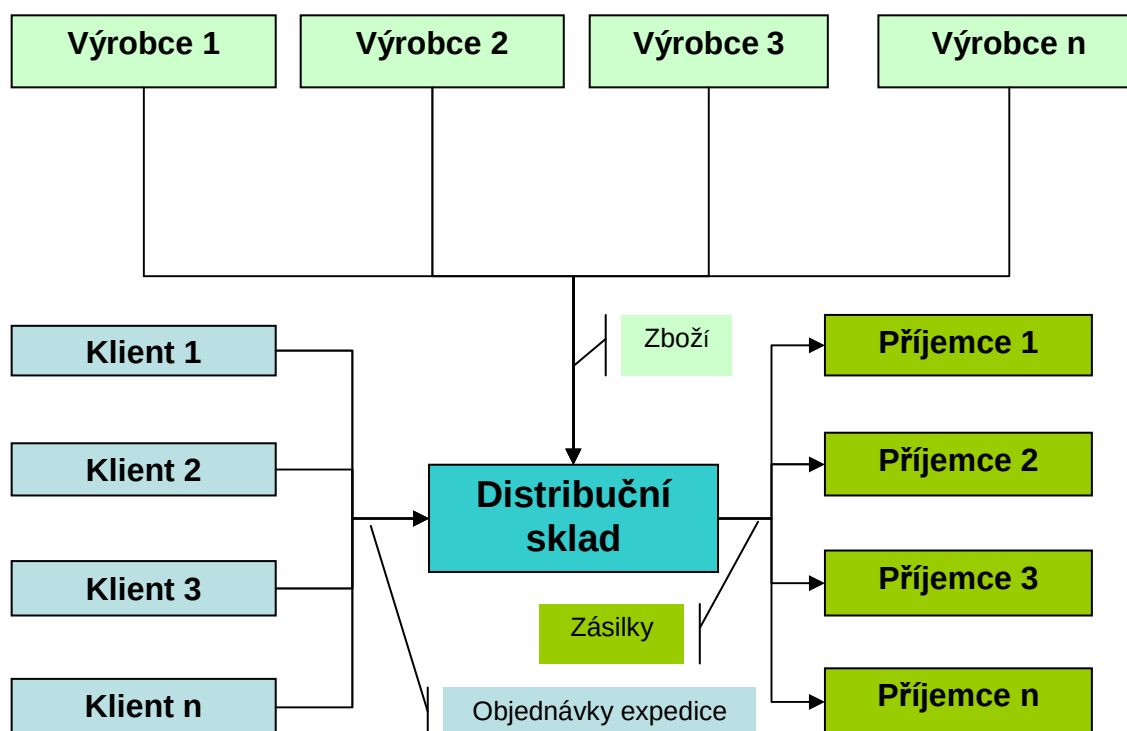
Poděkování za pomoc a podnětné připomínky patří vedoucímu diplomové práce panu doc. RNDr. Ing. Miloši Šedovi, Ph.D., a v neposlední řadě také rodině i všem blízkým a přátelům, kteří mi svou trpělivostí a tolerancí vytvořili podmínky k vypracování.

OBSAH

1. Úvod.....	3
1.1 Popis úlohy	4
1.2 Zvolené cíle.....	4
1.3 Přehled kapitol	5
2. Popis a rozbor stávajícího procesu expedice	7
2.1 Příjem objednávek k expedici.....	9
2.2 Definice a tvorba expediční dávky	9
2.3 Systém adresace pozic	11
2.4 Časová posloupnost expedice	12
2.5 Sběr položek a expediční plocha	13
3. Analýza vzorku dat a stavového prostoru	15
3.1 Metrika, kalkulace a stupně volnosti	19
3.2 Hodnoty reprezentativního vzorku dat	21
4. Teoretický přístup k řešení problematiky	25
4.1 Problém obchodního cestujícího.....	25
4.2 Historie TSP.....	26
4.3 Principy řešení	28
4.4 Prohledávací algoritmy	29
4.5 Některé vybrané optimalizační algoritmy užívané pro řešení TSP	31
4.6 Minimální kostra grafu	35
5. Praktické řešení aplikace	37
5.1 Vývojové prostředí	37
5.2 Konceptuální model aplikace.....	38
5.3 Grafy a jejich definice.....	43
5.4 Funkce ohodnocení hrany	44
5.5 Strategie	46
5.6 Prohledávací algoritmus	48
5.7 Ohodnocení objednávek	49
5.8 Vstupní a výstupní rozhraní.....	50
5.9 Uživatelské rozhraní a obsluha	51
5.10 Podmínky provozu	58
6. Testy.....	59
6.1 Optimální trasa a náročnost expediční dávky	59
6.2 Přesnost aproximativního ohodnocení objednávek	61
6.3 Test ideálního řešení	63
6.4 Časová náročnost výpočtů	63
7. Závěr	67
Seznam použité literatury	69
Seznam obrázků.....	71

1. Úvod

Motivací k žádosti o zpracování tématu této diplomové práce bylo nejen zdárné zakončení studia, ale také potřeba nalezení řešení praktického problému z oboru, v němž pracuji již bezmála deset let a s nímž se v mnoha obměnách často potýkám. Pracuji na brněnském terminále, dnes již nadnárodní logistické společnosti, která se zabývá zejména distribuční a zásobovací logistikou pro své klienty. Na našem terminále provozujeme kromě cross-dockového (800 m²) i distribuční sklad o celkové rozloze 7 200 m². Prostor distribučního skladu je rozdělena na regálový systém a volnou skladovací plochu. Regálový systém má kapacitu 6 367 paletových míst (Euro paleta 120 × 80 cm) a volná plocha je dalších 3 000 m² skladovací kapacity. V distribučním skladě poskytujeme našim klientům, kteří se etablovali zejména z obchodních a výrobních firem a kterým dovážíme komponenty či polotovary do výroby, případně spotřební zboží k prodeji od jejich dodavatelů z celého světa, především tyto základní služby: manipulaci, skladování, evidenci, třídění, balení a etiketování.



Obr. 1. Blokové schéma toku zboží a objednávek

Aktuální požadavky na moderní logistiku vycházející z potřeb našich klientů jsou, souhrnně vzato, v zásadě dva. V první řadě maximálně rychle obsloužit všechny aktuální objednávky expedice zboží, případně komponentů do výroby, k příjemcům a v druhé řadě provést celý proces, započatý přijetím objednávky k transportu a končící předáním zásilky příjemci, za minimální náklady, nejlépe zdarma. Zdánlivě dvě protichůdné úlohy. První maximalizační a druhá minimalizační, mezi nimiž existuje úzká vazba vzájemného ovlivnění. Na poli poskytovatelů komerčních logistických

služeb existuje široká řada metod, jak řešit předmětné úlohy. Na zvolené metodě, resp. výsledku řešení, je přímo závislá úspěšnost v tendrech na logistického partnera, případně výše vyprodukovaného profitu, který se při špatné kalkulaci může dostat do záporných čísel.

1.1 Popis úlohy

Jak již bylo výše uvedeno, v následujících kapitolách budou řešeny dvě hlavní úlohy, které zásadním způsobem ovlivňují ekonomičnost provozu a výkon expedičního procesu ve vztahu ke klientovi.

První řešenou úlohou je vybrat z množiny přijatých objednávek k expedici zboží právě takové objednávky, které mají nejvyšší nároky na zdroje, a tímto pomoci definovat optimální dávku ke zpracování. Dávka ke zpracování se rovná konsolidovanému souhrnu skladových položek z více objednávek přes jednotlivá umístění. Objednávky se obecně skládají z definice jednotlivých skladových položek a jejich množství k expedici. Jedna objednávka se rovná jedna expedovaná zásilka k jednomu příjemci. Jedna objednávka může obsahovat libovolný počet skladových položek. Libovolná skladová položka může být obsažena na libovolném počtu objednávek.

Druhou řešenou úlohou je pro vybranou množinu, dávku objednávek k expedici, navrhnout takovou vychystávající trasu, aby byla minimalizována spotřeba zdrojů potřebných k expedici z regálového systému. Definice vychystávající cesty právě jedné dávky objednávek je podkladem pro obsluhu manipulační techniky, která se pohybuje mezi regály a vyzvedává jednotlivé skladové položky z jejich umístění.

1.2 Zvolené cíle

Cílem této diplomové práce je analyzovat stávající systém expedice objednávek, využití zdrojů a tvorby expedičního plánu. Následně navrhnout prakticky využitelné řešení, které bude vykazovat vyšší efektivitu expedice objednávek z regálového systému distribučního skladu. Takovéto řešení implementovat a provést srovnávací testy se stávajícím řešením.

První úloha má za cíl nalezení právě takových objednávek, jejichž vyloučením z expedičního plánu maximalizujeme objem zvládnutých, expedovaných objednávek a zároveň minimalizujeme množství odložených, opožděných. Jinými slovy výhodnějším řešením je neuspokojit jednotlivé příjemce než-li kvantum.

Cílem druhé úlohy je najít v časově přijatelné době nejkratší vychystávající cestu, tedy minimalizovat ujetou vzdálenost v rámci vychystávající trasy v regálovém systému, a tím zvýšit množství zpracovaných objednávek. Vyšší efektivita je očekávána díky úspoře času na přejezdy a tím rychlejší sběr položek. V tomto případě jde o úlohu z oblasti třídy NP- těžkých.

V neposlední řadě je mým cílem také navázat na moji bakalářskou práci, a to ve smyslu tvorby prakticky využitelného systému pro konkrétní zadání. Výsledek mé bakalářské práce byl s povděkem aplikován několik let až do konce projektu, pro který byl navržen.

1.3 Přehled kapitol

Po úvodu do řešené problematiky následuje kapitola popisující a analyzující stávající proces expedice objednávek z regálového systému. Následuje obecný popis řešení předmětné problematiky, volba vhodného algoritmu zpracování, rozbor a návrh metody implementace vybraného inovačního postupu. Poté je představen konceptuální model provedení vybraného zlepšujícího řešení a jeho definice. Technika zpracování a popis postupu tvorby aplikace je pokračováním, mířícím ke zvolenému cíli. Poslední částí této diplomové práce jsou testy, srovnání a vyhodnocení výsledků na vzorových datech. Závěrem je diskutován získaný výsledek a provedeno hodnocení dosažených cílů včetně případného doporučení k možnému dalšímu rozšíření a úpravě pro potřeby praktického nasazení v prostředí stávajícího provozu na distribučním skladě.



Obr. 2. Pohled mezi regálové řady



Obr. 3. Pohled na regálové řady



Obr. 4. Pohled na paletové pozice



Obr. 5. Ukázka adresování pozic

2. Popis a rozbor stávajícího procesu expedice

- a) V následujících řádcích bude popsán praktický způsob expedice zásilek z distribučního skladu. Expedici jako takovou lze rozdělit do několika následujících fází: příjem objednávek k expedici zásilek;
- b) zpracování objednávek v rámci informačního systému;
- c) definice a tvorba expediční dávky;
- d) tisk vychystávací trasy;
- e) proces sběru položek z regálového systému;
- f) balení a etiketace na expediční ploše;
- g) uzavření výdeje jednotlivých zásilek v informačním systému;
- h) přesun hotových zásilek z expediční plochy na cross-dock.

Vybrané z uvedených fází procesu expedice budou dále diskutovány v jednotlivých podkapitolách. Nejprve ovšem ke specifikaci konstantního nastavení, tedy výchozích podmínek a využitelných zdrojů.

V rámci zpracovávaného úkolu, optimalizace procesu expedice zboží z regálového systému distribučního skladu, neuvažují problematiku řízení procesu naskladňování položek do pozic, možnost přestavby regálových konstrukcí, možnost navýšení množství manipulační techniky, změnu počtu obsluhujícího personálu a v neposlední řadě také jakékoliv narušení sjednaných časových oken vyhrazených k expedici.

Pro potřeby řešené úlohy uvažují stávající množství expediční manipulační techniky, kterou reprezentují dva vychystávací vozíky a jeden nízkozdvižný motorový paletovací vozík. Vychystávací vozíky jsou zařízení, kterými obsluhující personál projíždí definovanou trasu v regálovém systému a které umožňují zdvih osoby s celou řídicí kabinkou k jednotlivým pozicím, a to až do nejvyšších pater ve směru vertikálním pomocí hydrauliky a ve směru horizontálním pomocí pojezdu vpřed a vzad. Vozíky se vykazují velmi vysokou manévrovatelností, což jim umožňuje jedno hnací a zároveň ovládací otočné kolo v úhlu 360 °. Po příjezdu k žádané adrese manipulát přímo z kabinky vybírá potřebné položky uložené na paletě v regálové pozici a tyto odkládá na paletu vezenou k tomuto účelu. Tato paleta je uložena na posuvných vidlích, které jsou umístěny na zadní části kabinky. Po vyčerpání kapacity odkládací palety tuto obsluha odloží na aktuální pozici a nabere novou. Odvoz plných a přistavení prázdných palet zajišťuje obsluha nízkozdvižného motorového paletovacího vozíku. Tato má za úkol vyvážet z regálového systému vychystané plné palety na expediční plochu a zároveň z expediční plochy naváží zpět palety prázdné k vychystávajícím vozíkům do regálového systému. Komunikace probíhá na základě zvukové signalizace, kdy v případě potřeby výměny palety, nebo přesněji těsně před ní, obsluha vychystávacího vozíku přivolá pomocí klaksonu motorový nízkozdvih, slangově nazývaný motorka. Motorka je velmi rychlý a efektivní prostředek pro posun palet po ploše skladu. Obsluha stojící na plošině v zadní části vozíku pomocí řidítek ovládá pojezd i zdvih paletovacích vidlí v přední části. Zdvih je omezen do výše 50 cm. Jedna motorka zvládá bez obtíží obsluhovat dva vychystávací vozíky v regálovém systému, a to plynule bez zbytečných prodlev. Vychystávací vozíky se po celou dobu průjezdu definované trasy pohybují jen v regálovém systému.



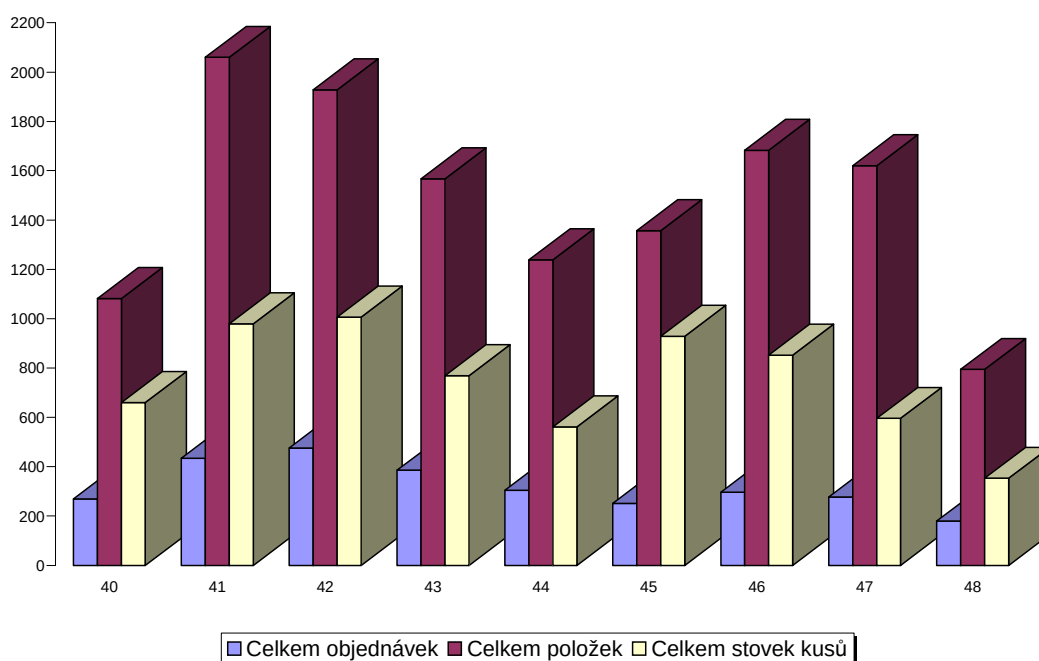
Obr. 6. Motorový nízkozdvižný vozík zvaný motorka



Obr. 7. Vychystávací vozík

2.1 Příjem objednávek k expedici

Objednávky k expedici zásilek jsou přijímány v expediční kanceláři v zásadě dvěma způsoby, a to buď datovými přenosy z informačního systému klienta, kdy tato data jsou přenášena přímo do skladového informačního systému, nebo formou e-mailu, kdy dochází k ručnímu přepisu objednávky do skladového informačního systému. Každá jedna objednávka se rovná jedné zásilce k jednomu konkrétnímu příjemci a zároveň obsahuje seznam položek včetně počtu kusů, které mají být tomuto příjemci expedovány. Klienti mají limitní omezení co do objemu zboží, které distribuční sklad garantuje denně zmanipulovat (obvykle 100 m³) a dále časové omezení, tedy do kdy mohou zasílat objednávky, které opět distribuční sklad garantuje vyexpedovat v den přijetí. Obvyklým hraničním časem je 15:00 hod. Objednávky přijaté po tomto čase jsou v případě nedostačující volné kapacity automaticky přesunuty k zpracování na následující den.

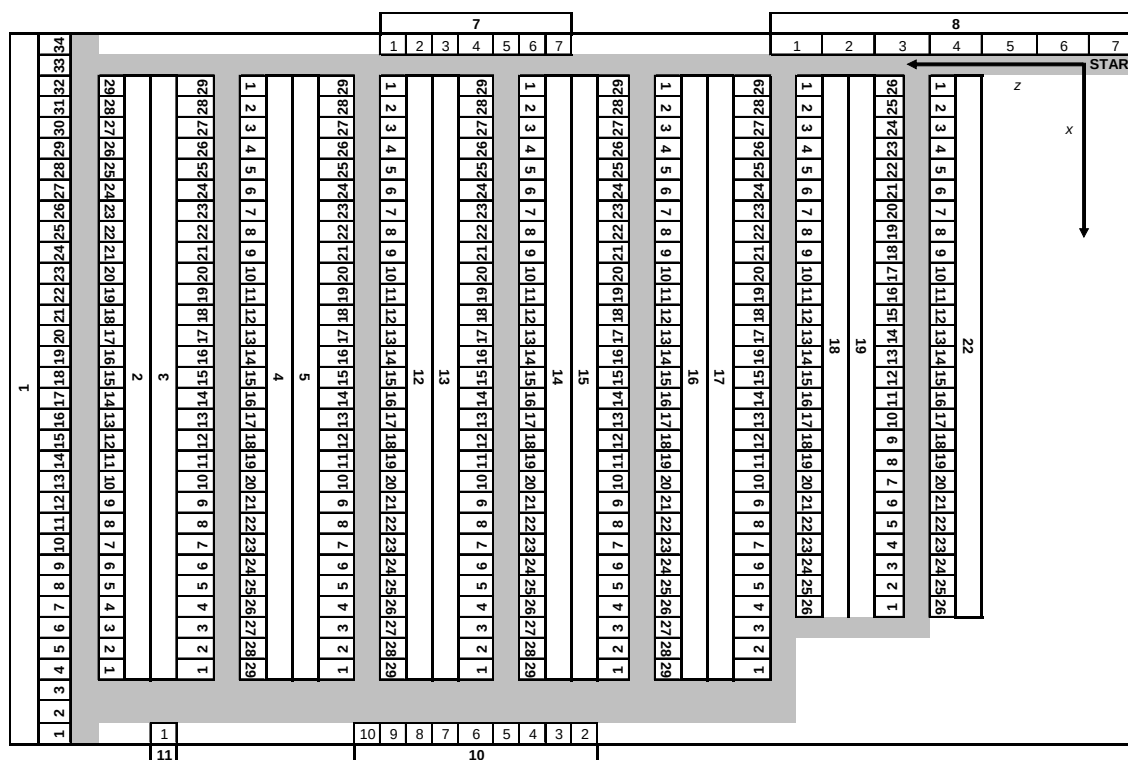


Obr. 8. Graf počtu objednávek, položek a kusů na sledovaném vzorku dat za období 1. 10–28. 11. 2008, seskupeno po týdnech

2.2 Definice a tvorba expediční dávky

Expediční dávka je množina objednávek, jejich položek, množství kusů od jednotlivých položek a umístění, na kterých se tyto položky vyskytují v regálovém systému. Umístění, ze kterých mají být položky vybrány, jsou informačním systémem optimalizovány dle parametrů sjednaných s klientem, jako například systém FIFO, prioritní šarže atp. Expediční dávka je konsolidována přes položky a jejich umístění v regálovém systému. Dále je utvořena vychystávací trasa pro obsluhu manipulační techniky, a to v současné době prostým seřazením dle adres pozic umístění. Výsledkem

takového prostého seřazení je posloupnost regálových řad, sloupců, pater a v nich jednotlivých pozic. Jelikož, jak je vidno na obr. 9 není rozložení adres na mapě regálového systému logicky seříděno (vzniklo postupnou dostavbou dalších řad), je tato metoda společně s principem sekvenčního zpracování jednotlivých řad a jejich sloupců značně neefektivní.



Obr. 9. Mapa regálového systému – půdorys

Rozložení jednotlivých regálových řad, jak je patrné z obr. 9, který znázorňuje vždy číslo řady 1–22 a dále ke každé řadě čísla sloupců 1–34, není provedeno smysluplně s ohledem na sekvenční zpracování prostého seřazení. Například v případě řady 1 a 2 nejvyšší číslo sloupce v řadě 1 nenavazuje v řadě 2 nejnižším číslem sloupce, tak jak u ostatních řad, které tvoří společnou uličku, takže v tomto případě by měl vychystávací vozík po ukončení průjezdu řady 1 přejet na nejnižší pozici řady 2 a ve zpětném směru zpracovávat pozice řady 2. Zde může dojít ke zbytečnému přejezdu až o celou délku řady. Šedá barva zobrazuje trasu obslužné cesty jednotlivých pozic. Ač v některých případech by nemusela být cesta pravoúhlá, tedy existuje kratší spojnice úhlopříčkou (26 pozice řady 22 na 1 pozici řady 17), jde o úsporu zanedbatelnou a pro řešení problematiky nepodstatnou.

Expediční dávka, tedy množství objednávek zahrnutých do jedné vychystávací trasy, tak aby dávka byla vychystána v potřebném čase, je v současnosti tvořena expertním odhadem pracovníků expedice, které posuzují náročnost objednávek, a v případě, že se dávka jeví nezvládnutelná, vyjmou z ní objednávku, kterou považují za nejkomplikovanější. Dalo by se hovořit o tzv. „selském rozumu“, kdy se na první pohled nabízí vyřadit objednávku s nejvíce položkami, při hlubším zamyšlení objednávku, která vyžaduje navštívit nejvíce pozic, ještě sofistikovanější by bylo vyloučit objednávku, která obsahuje nejvíce položek s nejmenším opakováním z hlediska celé dávky, tedy tu s nejvíce jedinečnými pozicemi, což nelze jen tak

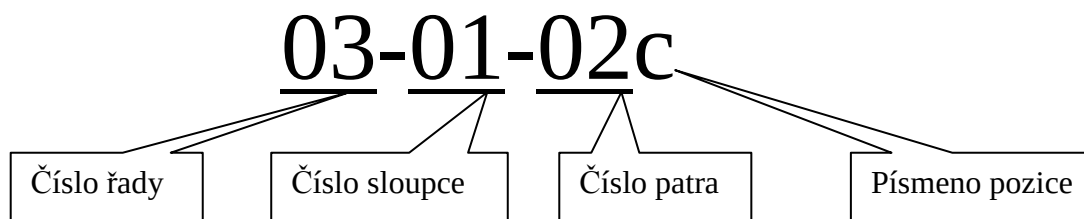
jednoduše odhadnout. Nicméně ani to, že je vyloučen doklad, který nejvíce navýší počet navštívených pozic, neznamená nejvyšší úsporu, neboť příkladně doklad s náročností pět po sobě jdoucích neopakovaných pozic v jedné řadě je jednodušší pro obsluhu než doklad se dvěma neopakovanými pozicemi s lokací v rozích skladu diagonálně proti sobě. Z uvedeného plyne, že odhad je velmi obtížný a tedy nepřesný. Jedinou přesnou metodou, jak určit doklad s nejvyšší náročností, je výpočet celkové hodnoty náročnosti expediční dávky, která je reprezentována délkou hamiltonovské kružnice (není podmínkou projet přes každý uzel pouze jednou) přes regálové pozice a postupným vyloučením jednotlivých dokladů a znovu spočtením její hodnoty lze dojít k rozdílu, který vypovídá o případné hodnotě úspory. Nejvyšší hodnota úspory vypovídá o nejnáročnějším dokladu z hlediska náročnosti obsluhy. Výpočetně velmi náročná úloha, která vzhledem k tomu, že samotné stanovení nejlevnější hamiltonovské kružnice je NP těžká úloha, může být neproveditelná v konečném polynomiálním čase. Zde se otvírá široký prostor pro nasazení heuristiky, tedy umění objevovat řešení pomocí znalostí řešeného systému. Nalezení řešení této úlohy za pomoci znalostí vlastností konkrétního systému a limitních podmínek je jedním z cílů této diplomové práce.

2.3 Systém adresace pozic

Každá paletová pozice v rámci regálového systému je jednoznačně určena nezaměnitelným identifikátorem. Tato jedinečná identifikační adresa má v sobě zakódovány informace o příslušnosti k řadě, sloupci, podlaží a pozici, tak jak je demonstrováno na obr. 10 a 11. Etikety s vytištěnou adresou pozice v čitelné velikosti jsou nalepeny na každém paletovém umístění pro dobrou orientaci obsluhy. Z praxe je známo, že šikovní obsluha regálového systému již natolik dobře zná adresní mapu pozic, že s naprostou bravurou a bez rozmyslu pojíždí od jedné pozice ke druhé. V rektilineárním systému, který regálový systém představuje, považuji za velmi vhodné využít kódování adres k relativně jednoduchému určování vzdáleností mezi jednotlivými uzly, a to pouze a jen na základě jejich adresy, tedy dvou textových řetězců.

		Řada 3												
Patra	Pozice													
		5	a	b	c	a	b	c	a	b	c	a	b	c
		4	a	b	c	a	b	c	a	b	c	a	b	c
		3	a	b	c	a	b	c	a	b	c	a	b	c
		2	a	b	c	a	b	c	a	b	c	a	b	c
		1	a	b	c	a	b	c	a	b	c	a	b	c
Sloupce:		1			2			3			4			

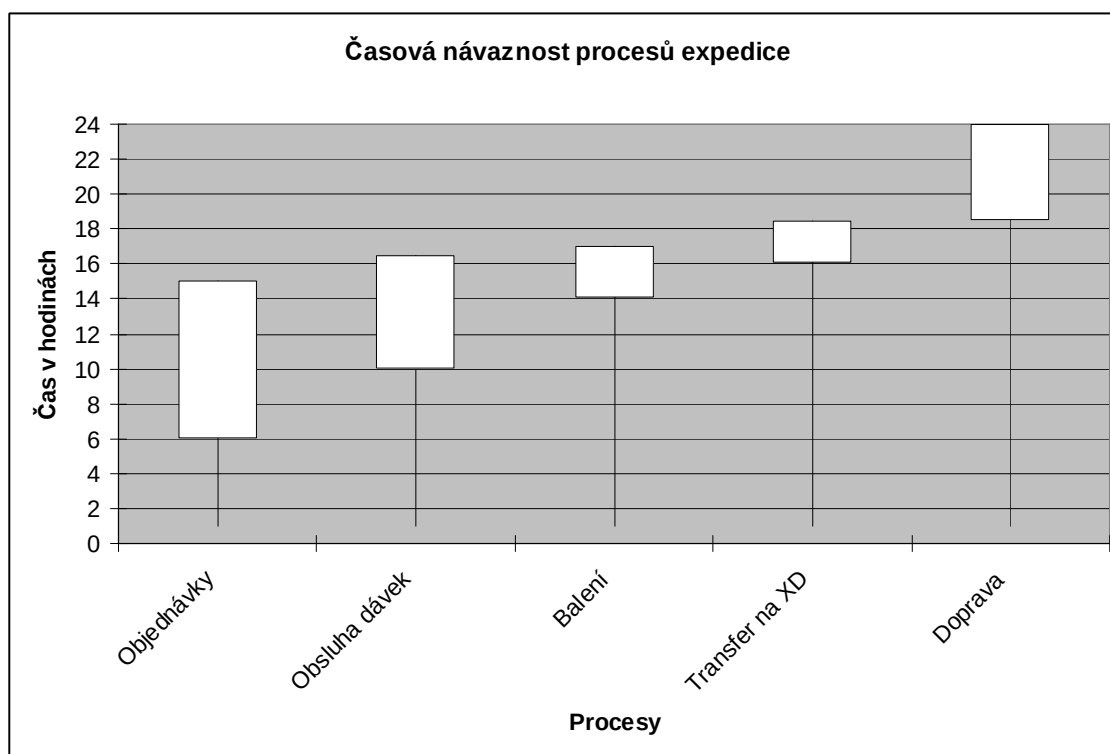
Obr. 10. Demonstrace adresování na části čelního pohledu regálové řady 3



Obr. 11. Příklad konkrétní adresy regálové pozice z řady 3

2.4 Časová posloupnost expedice

Organizaci procesu expedice objednávek z distribučního skladu z pohledu časové osy je možno vidět na obr. 12. Rozvrh začíná příjmem objednávek od klientů a jejich zpracováním v informačním systému. Následně pokračuje definicí a tvorbou expediční dávky spolu s tiskem specifikace trasy pro obsluhu vychystávajících vozíků. Dále pokračuje část fyzického sběru položek z regálového systému a jejich přesun na expediční plochu. Zde dojde ke kontrole správnosti sebraných položek a balení jednotlivých konkrétních objednávek na zásilky. Posledním krokem je předání hotových zásilek na cross-dockový sklad 24hodinového dopravního systému distribuce zásilek po celé České republice.



Obr. 12. Časový průběh procesů expedice

Pro přesnější pochopení využití denního časového fondu v rámci operací celého distribučního skladu je nutno sdělit, že v časové ose jsou popsány pouze souhrnné procesy expediční, nikoliv však příjmové, inventurní, konsolidační, úklidové, doplňkových služeb a nutná časová rezerva pro překlenutí špiček. Teprve všechny tyto procesy tvoří celkové využití časového fondu jednoho pracovního dne.

Prolínání jednotlivých procesů odráží průběžný dávkový systém zpracování objednávek, kdy paralelně probíhá více operací na různých expedičních dávkách. Jediná ostře navazující operace je doprava, což vyjadřuje souhrnné zpracování všech zásilek dopravou. Pochopitelně nelze na odchozí terminálová auta (spoje mezi depy) naložit jen zásilky některých dávek. Podmínkou je tedy mít v čase nakládky prvního terminálového auta všechny zásilky ze všech dávek na cross-dockovém skladě k dispozici.

Předchozí řádky naznačily problematiku časových limitů. Prvním důležitým časovým limitem, který je stanoven smluvně mezi klientem a distribučním skladem, je limit příjmu objednávek do expedičních dávek v jeden den. Tento limit je sjednán individuálně dle náročnosti skladované komodity každého klienta a pohybuje se od 13:00 do 15:00 hod. Objednávky přijaté nad tímto limitním časem mohou být expedovány následující den. Dalším důležitým limitním omezením je tvorba poslední expediční dávky tak, aby byly zpracovány všechny objednávky obdržené před tímto limitním časem, což je 15:00 hod. V praxi to znamená vytvoření poslední povinné expediční dávky těsně po třetí hodině, a to tak, že je nutné odhadnout zdali bude expediční dávka zpracovatelná kompletně až po přesun na cross-dock ve stanoveném čase, jinak by byla ohrožena expedice všech objednávek z dávky v objednaný den. Definice dávky je podrobněji popsána v kapitole 2.2. První dávku je možno zpracovat v deset hodin, a to z nezpracovaných objednávek předešlého dne a prvních objednávek dne aktuálního.

2.5 Sběr položek a expediční plocha

Jak již bylo zmíněno výše, ke sběru položek z regálového systému jsou k dispozici dva vychystávací vozíky. Přístupy jejich využití ke zpracování dávek jsou v zásadě dva. První je uplatněn prioritně vždy v případě současného začátku vychystávání oběma vozíky a spočívá v průjezdu definované trasy v protisměrné orientaci. Znamená to, že vozíky nikdy nebudou muset přejíždět mezi navzájem zpracovávány úseky. Nedojde k předjíždění jednoho druhým a k případnému zablokování průjezdu odloženými plnými paletami vychystaných položek. Navíc tento postup nevyžaduje žádné speciální požadavky na tiskovou sestavu vychystávací trasy, kdy jedna sestava postačuje pro oba vozíky. Jedním je zpracovávána od začátku ke konci a druhým od konce k začátku. Konec vychystávání nastává v bodě střetu protisměrného postupu vozíků vychystávací trasou. Druhým přístupem je tisk extra dávek pro každý vozík, což je ovšem výjimečné a jen v případě, že z nějakého důvodu nemohou být nasazeny oba vozíky současně. Tento postup nezajišťuje ošetření případných kolizí v průchodu systémem.

Tisková sestava expediční dávky, dle které je sběr položek prováděn, je definicí vychystávací trasy a tedy posloupností pozic, které je třeba postupně navštívit. Současně je u každé pozici definována položka kódem PNC, stručným popisem a jejím počtem k vyzvednutí. Šipka u umístění naznačuje přesun položky z a do umístění neboli pozice. Příklad části permutace pozic expediční dávky je na obr. 13 (nejedná se o vzor

užívaného tiskového výstupu). Tisková sestava trasy společně s dílčími potvrzenými příkazy výdeje jednotlivých objednávek po ukončení sběru z regálového systému a zabalení jednotlivých objednávek na expediční ploše putuje zpět k expedičnímu pracovníkovi, který na tomto základě provede potvrzení výdeje položek z informačního systému. Fyzické položky jsou po sběru z regálů seskupeny na expediční ploše, kde dochází k balení jednotlivých objednávek na přepravní jednotky a jejich etiketaci štítky s daty potřebnými k identifikaci kompletních zásilek v průběhu přepravního řetězce. Nakonec jsou zásilky přesunuty z expediční plochy přes hranici distribučního skladu do skladu dopravy, cross-docku, kam se zásilkami přechází i odpovědnost za jejich další osud na cestě k příjemci.

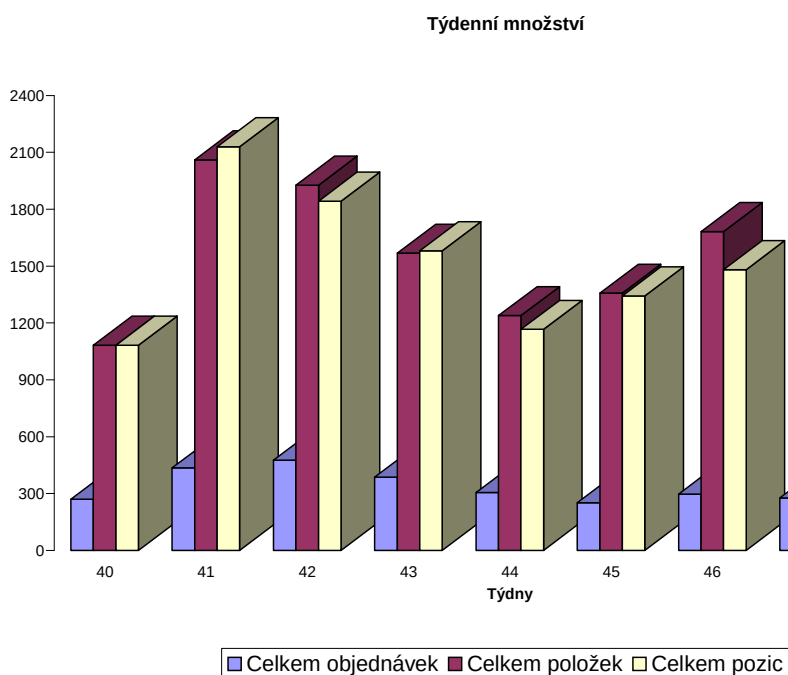
Kód (PNC)	Název	Umístění >	> Umístění	Počet	Odebráno
44312	Ping pong set s jedním míčkem	15-01-01c	Expedice	2	
39569	Zajíc růžový velký	15-01-04b	Expedice	2	
39567	Zvířátka 3 druhy	15-02-02a	Expedice	2	
38160	Medvídek, 2 barvy, srdíčka	15-02-02c	Expedice	2	
38143	Medvěd Valentin růžový a bílý	15-02-03b	Expedice	2	
39570	Kačenka	15-02-04a	Expedice	2	
44324	Badminton set s košíčkem	15-03-01c	Expedice	2	
39568	Králík růžový menší	15-03-02a	Expedice	2	
44316	Tenisová raketa s pouzdem	15-03-02c	Expedice	2	
44317	Tenisová raketa	15-03-02c	Expedice	2	
44318	Squashová raketa s pouzdem	15-03-02c	Expedice	2	
39566	Zvířátka ass 3 druhy	15-03-03b	Expedice	2	
44433	Čtyřkolka šlapací	15-03-04b	Expedice	2	
44424	Sliz v pařezu 12 ks v display boxu	15-04-02a	Expedice	24	
44355	Pokladnička auto	15-04-03a	Expedice	2	
44372	Ještěrka 24 ks v display boxu	15-04-03b	Expedice	2	
44425	Kostlivec 12 ks v display boxu	15-04-03b	Expedice	24	
44343	Míčky na pingpong 40mm žluté 6ks	15-04-03c	Expedice	2	
44365	Pokladnička obchod	15-06-04c	Expedice	2	
44368	Pokladnička domeček s hodinami	15-06-05b	Expedice	2	
44421	Sliz v kbelíku 12 ks v display boxu	15-07-02a	Expedice	24	
44426	sliz 12 ks v display boxu	15-07-02a	Expedice	24	
44331	Badmintonové košíčky 6pcs	15-07-03a	Expedice	2	
44310	Badminton set 2 rakety	15-07-04a	Expedice	2	
44390	Koule se sítkou 12 ks v display boxu	15-07-05b	Expedice	24	
44362	Pokladnička míč ragby	15-08-02a	Expedice	2	
44364	Pokladnička medvídci kabelka	15-08-03a	Expedice	2	
44322	badminton set s košíčkem	15-08-04a	Expedice	2	
44351	Basketbalový míč číslo 7	15-08-05c	Expedice	2	
44326	Badmintonové košíčky 6ks v tubě	15-09-02a	Expedice	2	
44349	Basketbalový míč číslo 7	15-09-02c	Expedice	2	
44370	Pokladnička budík s hodinami	15-09-04a	Expedice	2	
44309	badminton set 2 rakety	15-09-05b	Expedice	2	
44311	Badminton set 2 rakety	15-09-05b	Expedice	2	
44389	Lebka 12 ks v display boxu	15-10-02c	Expedice	24	
44320	Karimatka s flíí	15-10-03b	Expedice	2	
44380	Netopýr 48 ks v display boxu	15-10-03c	Expedice	96	
44386	Koule 12 ks v display boxu	15-10-03c	Expedice	24	
44388	Želvy v display boxu 12ks	15-10-03c	Expedice	24	
44391	Lebka 12 ks v display boxu	15-10-03c	Expedice	24	
44319	Karimatka	15-10-05c	Expedice	2	
44387	Alien svítili ve tmě 16 ks v display boxu	15-11-02a	Expedice	32	
44393	Mimozemšťan 16 ks v display boxu	15-11-02a	Expedice	32	
44371	Pijavice 48 ks v display boxu	15-11-03b	Expedice	2	
44376	Dinosauri 24 ks v display boxu	15-11-03b	Expedice	48	
44377	Krysa v display boxu 24 ks	15-11-03b	Expedice	48	
44354	Dřevěný badminton	15-11-04c	Expedice	2	

Obr. 13. Část posloupnosti pozic stávající tiskové sestavy vychystávající trasy

3. Analýza vzorku dat a stavového prostoru

Po úvodu do problematiky a popisu procesu expedice z regálového systému expedičního skladu přecházím k analýze historického vzorku dat, na kterém bude demonstrována složitost a zároveň budou získána data, která budou dále využita. Tímto bude splněn jeden z cílů této diplomové práce, kterým je porovnání navrženého a aplikovaného systému zpracování expediční dávky a tvorby vychystávací trasy. Exaktní změření zisku zpracovaného řešení oproti stávajícímu stavu bude provedeno závěrem. Vzorek dat je záměrně vybrán z nejnáročnějšího období v rámci roku, tak aby maximálně zatížil navrhovanou aplikaci. Tímto obdobím je měsíc říjen a listopad roku 2008.

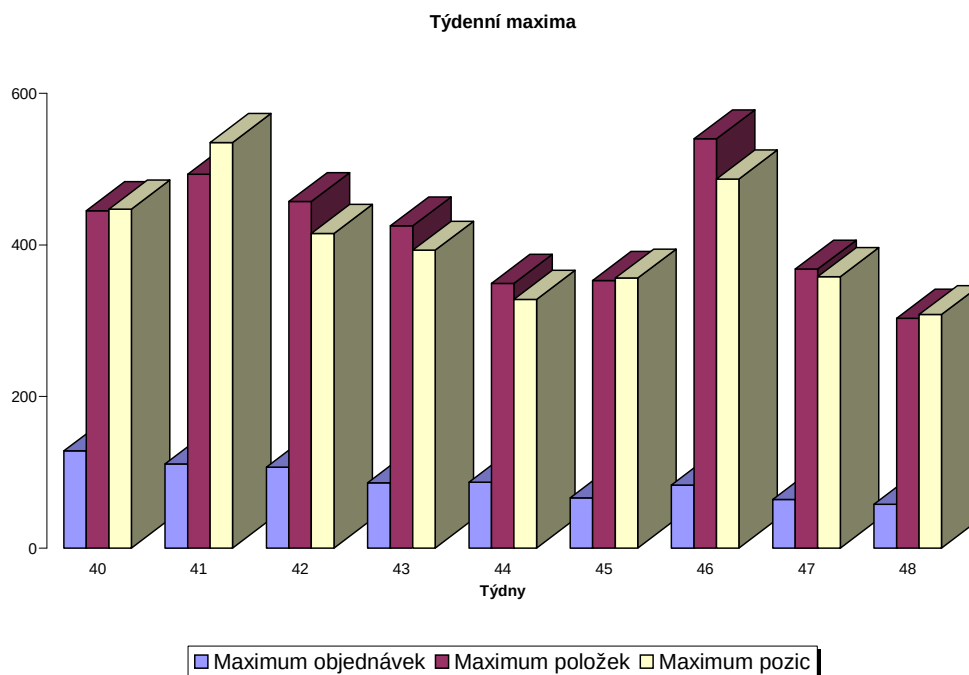
Nejdříve se na data podíváme přes množství objednávek, položek, pozic a kusů, a to přes grafy, které nejvhodněji interpretují. První graf reprezentuje množství objednávek, v nich obsažených položek a umístění těchto položek v regálových pozicích, to vše vztaženo na kalendářní týdny.



Obr. 14. Graf týdenního množství

Týdenní množství představuje, jaký je poměr mezi objednávkami a množstvím položek v nich obsažených a rovněž poměr mezi množstvím položek a množstvím navštívených pozic. Pozvolný trend v logistickém byznysu vede k objednávkám s vyšším počtem položek o méně kusech, což je dáno snahou odběratelů, obvykle maloobchodníků, minimalizovat náklady na skladování a tedy udržení optimálních minimálních zásob. Navíc je obvyklé, že náklady na dopravu hradí prodejce, tedy odesílatel. Poměr mezi množstvím položek a regálových pozic vyjadřuje náročnost expedice, která je ovlivňována zejména prioritními výběry z pozic (například FIFO) a dále také vlastnostmi položky, zejména rozměry, kdy čím větší položka, tím více zabere

místa a logicky tím více pozic musí být navštíveno i při menším počtu kusů takovéto položky. Řádově z 90 % v našem distribučním skladě platí, že jedna pozice se rovná jedna položka.



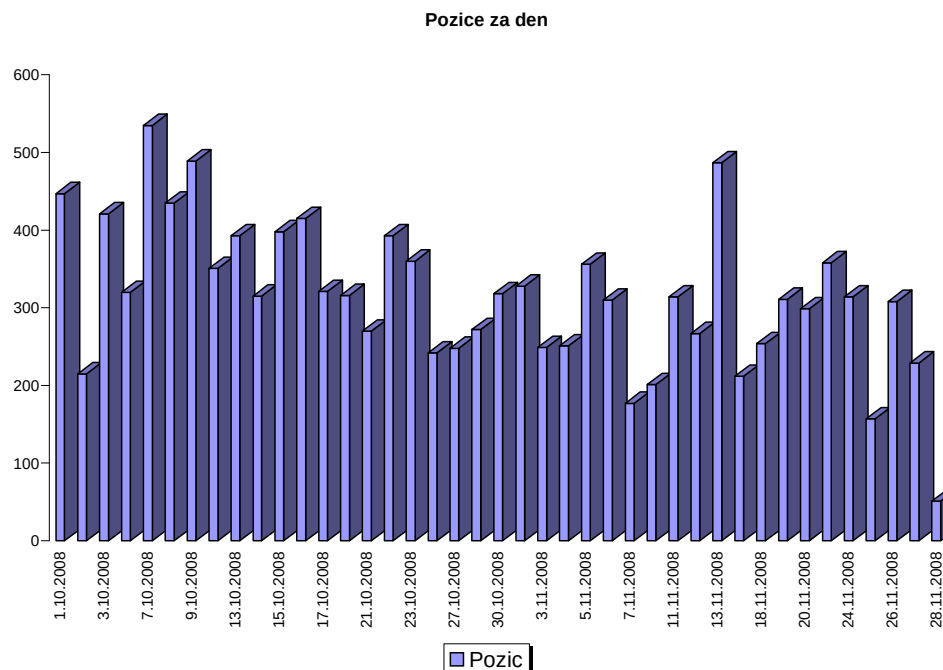
Obr. 15. Graf týdenních maxim

Graf na obr. 15 interpretuje týdenní maxima objednávek, položek a pozic. Je možno z něj vysledovat jistý harmonický průběh hodnot sledovaných dat, který vypovídá o cyklickém střídání náročných a volnějších období. Vždy na začátku a konci měsíce je období nižších objemů, kdežto uprostřed měsíce nastávají maxima odbavovaného množství.

	Objednávek	Položek	Pozic	Kusů
Průměrná denní hodnota	70	325	315	16351
Střední denní hodnota	70	334	314	15386
Maximální denní hodnota	128	540	535	34360
Minimální denní hodnota	20	52	51	2538

Obr. 16. Tabulka sledovaných hodnot

Tabulka na obr. 16 nám vypovídá v hodnotách o náročnosti expedice, a to na všech úrovních její činnosti. Tato náročnost přímo naznačuje, s jak velkým množstvím dat se bude muset potýkat navrhovaná optimalizační aplikace, když bude chtít uspět v přijatelném čase. Jak je patrné, je rozdíl mezi minimem a maximem více než desetinásobný, přičemž střední hodnota se společně s průměrnou drží mírně nad hladinou třech stovek regálových pozic, což nás z hlediska výpočetního zajímá nejvíce. Jelikož časová složitost problému je úměrná počtu permutací nad pozicemi, která je dána výrazem $O(n!)$ [2], dělá to v našem případě při střední hodnotě 314 pozic úctyhodné číslo mimo řád běžné představitelnosti, vyjádřitelné jako ∞ .



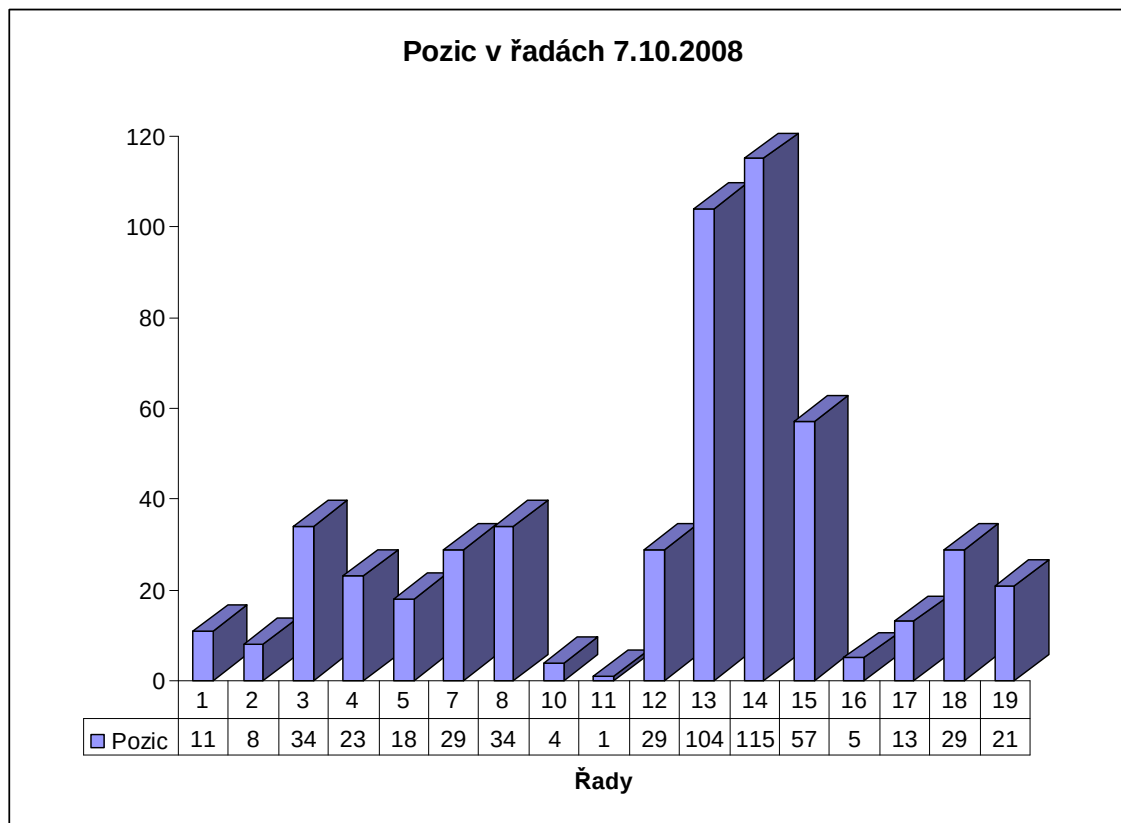
Obr. 17. Denní přehled množství pozic

Obr. 17 představuje denní množství regálových pozic, které je nutno navštívit, aby byly vychystány všechny objednávky v den, kdy jsou klienty objednány k expedici. Přehled nezahrnuje časová limitní omezení, tedy uvažuje všechny objednávky daného data. Nejnáročnějším dnem na sledovaném vzorku je 7. 10. 2008 s 535 pozicemi. Nejméně náročným dnem je 28. 11. 2008 s 51 pozicemi. Reprezentantem střední hodnoty je 11. 11. 2008 s 314 regálovými pozicemi. Tyto tři dny, resp. hodnoty náročnosti vychystávací cesty souhrnu objednávek za každý tento den, dále považuji za reprezentativní vzorek možného denního maxima, minima nebo střední hodnoty a tedy na systém budu nadále nahlížet v souvislosti s těmito daty.

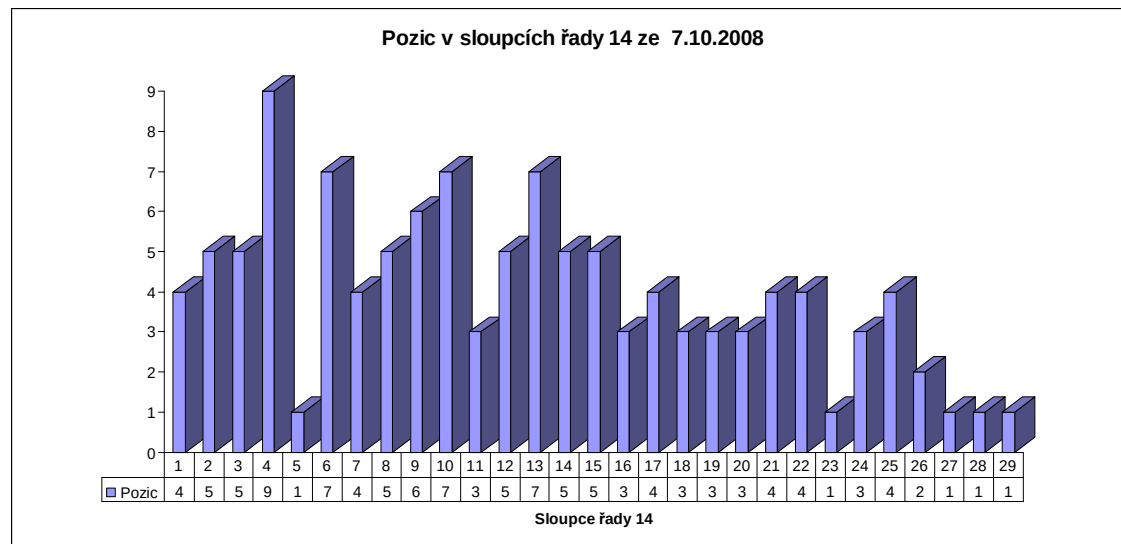
Datum	Objednávek	Položek	Pozic	Kusů
28.11.2008	20	52	51	2538
11.11.2008	64	404	314	21978
7.10.2008	111	458	535	25776

Obr. 18. Reprezentativní vzorek dat

Z tabulky na obr. 18 budou dále podrobněji rozvedena data maxima, tedy dne 7. 10. 2008. Nejprve pro kvalitnější představu rozložení přes regálové řady a následně nejobsazenější regálovou řadu přes sloupce. Jak je patrné z grafu na obr. 19, který reprezentuje 535 pozic rozložených přes regálové řady, nejvytíženější řadou je řada 14 se 115 pozicemi. Sousedem v uličce je řada 13, která je se 104 pozicemi zároveň druhou nejnáročnější řadou expediční dávky dne 7. 10. 2008. Optimalizační úloha průchodu uličkou mezi řadou 13 a 14, kdy bude třeba navštívit 219 pozic, bude opravdovou zatěžkávací zkouškou algoritmu zkonstruované aplikace. Naopak průchod uličkou mezi regálovými řadami 1 a 2 zahrnuje návštěvu pouze 19 pozic, což by se mohlo jevit v případě větších rozstupů mezi pozicemi v ose jako úloha triviální.



Obr. 19. Reprezentace 535 pozic přes regálové řady



Obr. 20. Reprezentace 114 pozic řady 14 přes jednotlivé sloupce

Graf na obr. 20 vyjadřuje rozložení 114 pozic přes sloupce řady 14. Každý sloupec obsahuje 4–6 pater po třech pozicích, tedy v maximu 18 a minimu 12 možných pozic na patro. Jak je patrné na první pohled, není vynechán jediný sloupec, přičemž minimum pozic ve sloupci je jedna a maximum devět. Tímto bych uzavřel kapitolu analýzy množství na reprezentativním vzorku a jeho rozložení přes regálový systém.

3.1 Metrika, kalkulace a stupně volnosti

Metrika užitá v rektilineárním regálovém systému je odvozením pro trojrozměrný prostor od tzv. manhattanské metriky („...po Manhattanu v New Yorku se chodí v síti pravoúhlých ulic, které jsou rovnoběžné s osami souřadného systému. Takže vzdálenost p_1 a p_2 je $|x_1 - x_2| + |y_1 - y_2|$ “) [9], tedy vzdálenost uzlu p_1 a p_2 je dána $|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2|$. Hodnota, dle které bude měřena složitost jednotlivých tras, vychází z celkové ujeté vzdálenosti při průjezdu mezi všemi zadanými uzly. Tato hodnota bude rozhodující pro posouzení úspěšnosti navrženého a realizovaného řešení touto diplomovou prací oproti nynějšímu stavu zkoumanému na reprezentativním vzorku dat. Ostatní parametry popsané dále nevstupují do hry, neboť je zde předpoklad jejich konstantního chování vzhledem k porovnáváním historickým trasám.

Pro určení časové náročnosti a tedy definování optimální expediční dávky vstupují do hry mimo celkové vzdálenosti další tři parametry, a to doba odbavení uzlu, doba potřebná k výměně palety v průběhu průjezdu trasou a počet výměn palet při průjezdu trasou. U prvního parametru předpokládám stejnou časovou náročnost na odbavení jednoho uzlu, která je reprezentována střední hodnotou odbavení jednoho uzlu, která je na počátku určena odhadem a dále může být zpřesněna na základě odklonu reality od vypočteného odhadu. Z uvedeného plyne, že první parametr bude vstupovat do měřené složitosti jako časová konstanta. U druhého parametru rovněž kalkuluji se stejnou časovou náročností pro vychystávacím vozíkem spotřebovaný čas na výměnu sesbírané palety za prázdnou v průběhu průjezdu libovolnou trasou, tedy opět konstanta určená odhadem. Třetí parametr se odvíjí od objemu balení položek, jejich počtu a způsobu ložení na paletu při odběru z pozice. Jelikož první informaci nemáme k dispozici a poslední je závislá na obsluze a její dovednosti, je tento parametr opět určen odhadem s tím, že jeho upřesnění je dále možno v závislosti na odklonu modelu od reality. Z výše uvedeného je patrné, že jediným parametrem vstupujícím do výpočtu, který není konstantou a tak bude mít přímý vliv na změnu náročnosti expediční dávky, je změna celkové délky vychystávací trasy vyřazením některé z objednávek, čímž je naplněn hlavní cíl této diplomové práce, kterým je optimalizace délky vychystávací trasy v regálovém systému logistického skladu.

Závěrem rozpravy o metrice a kalkulaci hodnot poznámka ke vstupu celkové délky do výpočtu časové náročnosti expediční dávky. Jelikož délka trasy pro vyjádření času je dělena průměrnou rychlostí pohybu vozíku mezi jednotlivými uzly, je nasnadě, že průměrná rychlost má zásadní vliv na celkový čas potřebný k průjezdu trasy. Největší vliv na průměrnou rychlost průjezdu trasou nemají konstrukční možnosti vozíku, nýbrž pouze a jen lidská obsluha, pro kterou je nejnáročnějším úkolem rychlá orientace v adresném prostoru regálového systému a tedy rychlé vyhodnocení polohy následujícího uzlu a jeho, pokud možno, co nejrychlejší dosažení nejkratší cestou, se kterou kalkuluje navržený a implementovaný systém.

Značení:

D – celková délka trasy	[m]
P – vrchol => pozice	
H – hrana	
n – počet vrcholů, pozic	
S – start, cíl => vrchol začátku a konce	
O – čas odbavení expediční dávky	[s]

U – doba odbavení jednoho uzlu	[s]
W – doba výměny palety	[s]
q – počet výměn palet na expediční dávku	
v – průměrná rychlost vychystávacího vozíku	[ms ⁻¹]

Vzorce:

$$D = \sum_i^n H_i \quad (3.1)$$

Rovnice 3.1 nám udává výpočet celkové délky trasy, který je, slovně vyjádřeno, roven součtu všech hran mezi navštívenými uzly otevřené hamiltonovské cesty, jež je definována permutací pozic.

$$O = \frac{D}{v} + U * n + W * q \quad (3.2)$$

Rovnice 3.2 je vyjádřením náročnosti zpracování expediční dávky, což je, slovně vyjádřeno, součet tří dílčích časových náročností. Prvně dobou průjezdu regálovým systémem, která je vyjádřena podílem délky trasy průměrnou rychlostí vozu. Dále střední hodnotou doby obsluhy jednoho uzlu násobeno počtem uzlů v trase a naposledy střední hodnotou doby výměny palety v průběhu průjezdu trasou násobeno počtem výměn.

Konstanty:

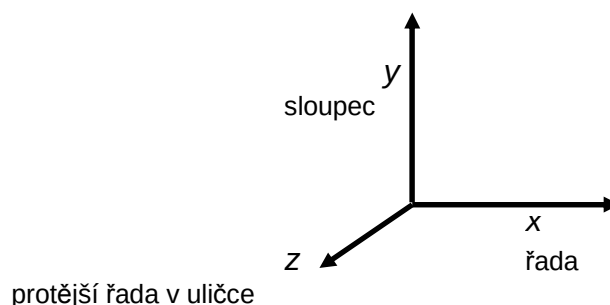
U – 60	[s]
W – 45	[s]
q – $n/10$	
v – 0,2	[ms ⁻¹]

Prostor regálového systému, ve kterém se pohybujeme sběrným vozíkem, je pochopitelně pravoúhlého charakteru. Možnost pohybu je tedy pouze v osách x , y a z . Pro modelování pohybu vozíku budeme tento považovat za nehmotný bod, který se pohybuje po hranách pravoúhlé grafové struktury od vrcholu k vrcholu dle předepsané permutace pozic. Ohodnocení hran, tedy vzdálenost mezi jednotlivými vrcholy grafu, je konstantní v rámci každé řady. Celkově lze definovat jedenáct konstantních délek hran dle typu spojnice, které úplně definují vzdálenosti v grafu:

1. hrana v regálové řadě ve směru osy x ;
2. hrana v regálové řadě ve směru osy y ;
3. hrana v regálové řadě ve směru osy z ;
4. hrana přechodu mezi dvojitou řadou regálu ve směru osy z ;
5. horní hrana přechodu mezi svislými a vodorovnými řadami ve směru osy x ;
6. dolní hrana přechodu mezi svislými a vodorovnými řadami ve směru osy x ;
7. hrana, souřadnice start a první pozice vodorovné řady 7 ve směru osy z ;

8. hrana, souřadnice start a první pozice vodorovné řady 8 ve směru osy z ;
9. hrana, souřadnice start a první pozice vodorovné řady 10 ve směru osy z ;
10. hrana, souřadnice start a první pozice vodorovné řady 11 ve směru osy z ;
11. hrana, souřadnice start a první pozice svislé řady regálů.

Každý uzel neorientovaného grafu G reprezentujícího regálový systém uvnitř některé z ulic může být maximálně stupně pět a minimálně stupně dva. Mezi regálovými řadami je tedy možno se pohybovat z každého uzlu v ose x vpřed nebo vzad, rovněž v ose y nahoru či dolů, případně v ose z z jedné strany regálové uličky na druhou. Uzly na vstupu do regálových ulic, přesahující v řadě 1, případně z některé ze čtyř podélných řad, jsou tzv. hraniční a mohou mít společnou hranu s libovolným dalším hraničním uzlem, kterých je dohromady 97.



Obr. 21. Definice souřadného systému

Hrana	Délka v metrech
1	0,93
2	1,75, (1)*
3	3,2
4	2,75
5	4,3
6	6
7	23,6
8	0
9	29,2
10	45,2
11	12,3

Obr. 22. Tabulka hran jednoznačně určujících vzdálenosti v regálovém systému

*) Výjimku tvoří regálová řada 8, 7, 10 a 11 s délkou hrany 1 m.

3.2 Hodnoty reprezentativního vzorku dat

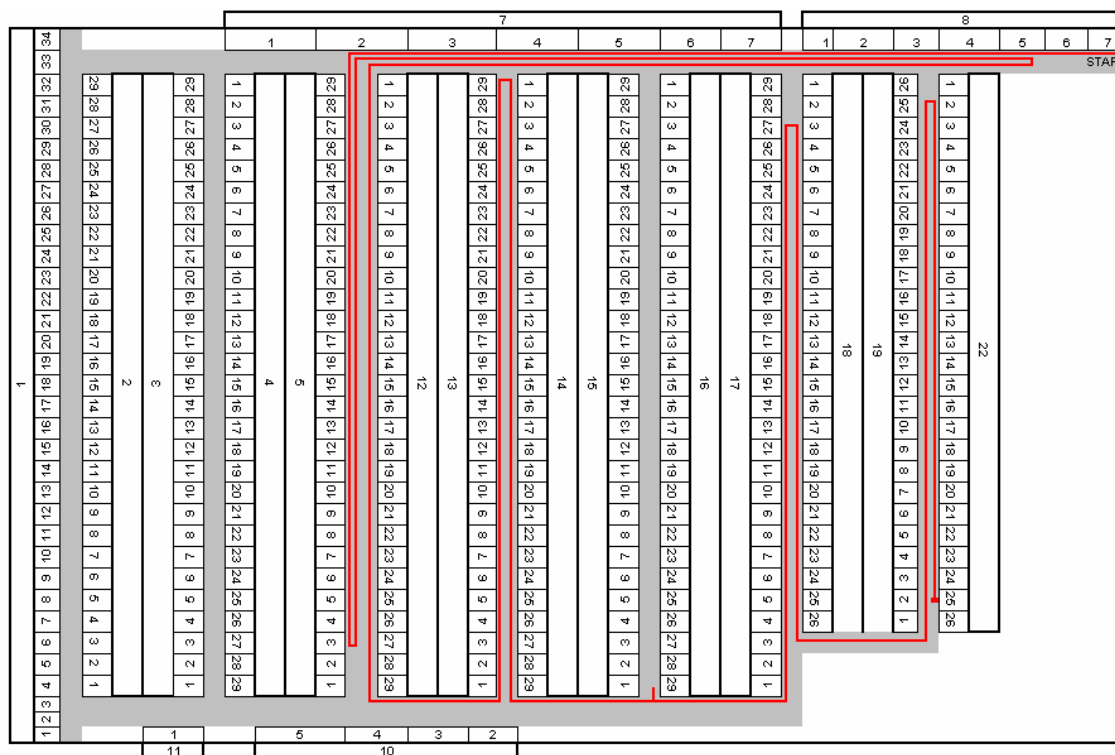
Závěrem kapitoly analyzující současný stav a metriku přináším hodnoty D a O tří reprezentativních expedičních dávek vybraných na začátku této kapitoly. Hodnoty jsou stanoveny na základě vztahu 3.1 a 3.2. Společně tvoří základ k závěrečnému

vyhodnocení úspěšnosti navrženého a implementovaného inovačního řešení v následujících kapitolách.

Hodnota	Nejmenší dávka	Střední dávka	Největší dávka
Celková délka D v metrech	869	2024	2760
Celková náročnost O v min.	127	506	805

Obr. 23. Tabulka hodnot reprezentativního vzorku dat dle stávajícího způsobu zpracování expediční dávky

Hodnoty z tabulky na obr. 23 jsou kalkulovány dle stávajícího systému generování trasy průchodu regálovým systémem s počátečním bodem na pozici 08-07-01c (viz mapa systému na obr. 9), který je vždy stejný pro všechny dávky. Na obr. 24 je naznačen půdorysně současný průchod regálovým systémem pro případ nejmenší expediční dávky.



Obr. 24. Půdorys trasy nejmenší dávky

Metoda průchodu systémem je velmi neefektivní, což je patrné na první pohled vyznačenou stopou na obr. 24, která vyjadřuje postup od bodu start přes všechny následující po cílový bod, kterým je v tomto případě pozice v řadě 22 ve sloupci 25.

Naznačení průchodu regálovou řadou v ose x , y (bokorys) v případě zpracování modelem prostého seřazení a při plném obsazení je patrné z obr. 25. Opět je na první pohled patrná neefektivita této metody.

Principiálně se nejedná o zcela zavrženíhodný přístup, neboť jde o nejrychlejší možnou variantu z hlediska zpracování v informačním systému, nicméně praktické užití by (v případě nenáročného uživatele) muselo splňovat dvě podmínky. Regálový systém by měl být vystavěn a adresován stejnou strategií průchodu, případně by informační systém

měl mít implementovánu extra řadicí specifikaci. Vše, aby byla dosažena alespoň eliminace pravidelných návratů z nejvyššího patra do nejnižšího a byl zajištěn rozumně logický průchod přes regálové řady.



Obr. 25. Průchod regálovou řadou při plném obsazení metodou seřazení

4. Teoretický přístup k řešení problematiky

Úkol zní relativně jednoduše. Navštívit konkrétních n uzlů z množiny M možných, a to tak, aby celková délka trasy byla minimální. Množinu M v pojetí řešeného problému reprezentují všechny pozice regálového systému, jež jsou vymezením trojrozměrného stavového prostoru, a cesta mezi n uzly je součet délek hran, spojnic mezi jednotlivými konkrétními n uzly.

Uvedený úkol je obdoba úlohy obecně nazývané „Problém obchodního cestujícího“ (TSP Traveling Salesman Problem). Obdoba v tom ohledu, že není důsledné omezení pro násobný průjezd jedním uzlem a zároveň není požadován návrat do výchozí pozice. Pro vyřešení netriviální úlohy neexistuje exaktní algoritmus pracující v polynomiálním čase, proto úloha patří do třídy úplných NP-těžkých.

4.1 Problém obchodního cestujícího

„Problém obchodního cestujícího“ (TSP) je obtížný diskretní optimalizační problém, matematicky vyjadřující a zobecňující úlohu nalezení nejkratší možné cesty procházející všemi zadanými městy na mapě. Na mapě je zvoleno N měst a jejich vzájemná vzdálenost pro každou z dvojic. Problém obchodního cestujícího je určit takové pořadí návštěv jednotlivých měst, aby každé město bylo navštíveno právě jednou, výsledná délka cesty byla co nejkratší a cestující se vrátil zpět do počátečního místa. V tomto případě vždy existuje nějaké ideální řešení (případně více, ale se stejnou cenou), které můžeme lehce najít tak, že prostě spočítáme délku všech možných cest a vybereme tu nejkratší.

Formálně lze problém obchodního cestujícího definovat pomocí teorie grafů. Uvažujme souvislý vážený graf. Uzly tohoto grafu reprezentují jednotlivá města a hrany reprezentují silice mezi jednotlivými městy. Cenu cesty (vzdálenost, kvalita silnice atd.) mezi jednotlivými městy označují váhy těchto hran. Řešením tohoto problému je nalézt hamiltonovský cyklus s minimálním součtem jednotlivých vah (cenou). Lze dokázat, že požadavek na návrat do místa, kde cesta začíná, nemá vliv na výpočetní složitost tohoto problému.

Náš problém náleží do třídy NP-těžkých problémů, pro které je typická vysoká výpočetní náročnost. Že jde o nedeterministicky polynomiální (NP) problém, je patrné z toho, že nedeterministický počítač, umožňující v každém kroku rozvětvit výpočet na libovolný počet větví, by mohl začít v některém „městě“, rozdělit propočet délky trasy na tolik větví, kolik vede z města silnic, a v každém z cílových měst postupovat stejně – s výjimkou tras vedoucích do již navštívených měst. Tak by prohledal všechny možné trasy v n výpočetních krocích, pokud počet měst činí N , a rozvětvil by se maximálně do $(N - 1)!$ větví.

Přestože neexistuje reálný nedeterministický počítač, lze tento problém velice jednoduše řešit pomocí jeho deterministické simulace. Tato metoda se také někdy nazývá metoda „hrubé síly“, při níž prostě sekvenčně prozkoumáme všechna možná řešení daného problému. Časová složitost této metody může být snadno odvozena z této myšlenky: „Mějme daný počet měst n a známou cenu cesty mezi každými dvěma městy. Velikost prohledávaného stavového prostoru pro symetrický TSP problém je poté $(n - 1)!/2$, pro $n > 2$ “ [10].

Aby bylo řešení efektivně nalezeno v přijatelném čase, je nutné zvolit některou z aproximativních metodu řešení za pomoci heuristiky. O takovémto výsledném řešení nelze prohlásit s určitostí, že se jedná o optimální řešení, nicméně toto se od optimálního nebude příliš lišit a hlavně bude použitelné pro reálné aplikace.

4.2 Historie TSP

Původ názvu úlohy ani počátky objevu nejsou známy. Známé matematické formulace související s problémem obchodního cestujícího sahají do 18. století k irskému matematiku siru Williamu Rowanu Hamiltonovi (obr. 26) a britskému matematiku Thomasi Penyngtonu Kirkmanovi. Jednou z prvních trénovacích úloh byla Hamiltonova hra Icosian, jejímž úkolem bylo vytvořit cestu spojující dvacet uzlů hracího pole po předem definovaných spojnicích (obr. 27). Zajímavá diskuze na téma TSP v práci Hamiltona a Kirkmana je publikována v knize *Graph Theory 1736-1936* od N. L. Biggse, E. K. Lloyd, a R. J. Wilsona, Clarendon Press, Oxford, 1976 [11].

„Obecná forma problému TSP byla poprvé studována matematiky z Vídně a Harwardu, především Karlem Mengerem ve 30. letech 20. století. Od té doby vzniklo nepřehledné množství různých algoritmů, snažících se nalézt optimální cestu obchodního cestujícího“ [10].



Obr. 26. Sir William Rowan Hamilton (1805–1865) [11]

Není divu, že od dob Hamiltona a Kirkmana se problematikou zabývalo a stále zabývá velké množství badatelů, neboť motivací k optimalizaci průchodu ohodnoceného grafu je z reálného světa stále veliké množství. Například všechny logistické úlohy, jako vyskladňování, naskladňování, distribuce, dále úlohy spojené

s optimalizací pohybu robotů a automatizačních techniky a v neposlední řadě silniční, železniční, komunikační a obecně veškeré sítě a jejich prvky.

Úspěchy ve vývoji algoritmů řešících problematiku, přesněji řečeno zásadní milníky ve vývoji řešení, představuje tabulka na obr. 28.



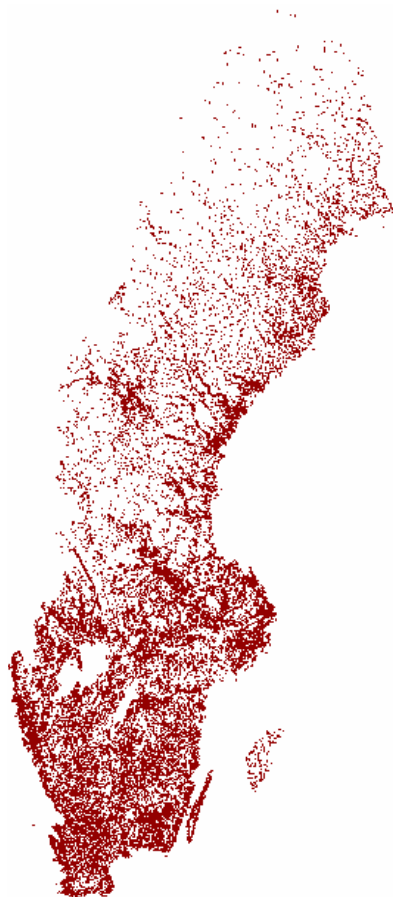
Obr. 27. Hamiltonova hra Icosian [11]

Rok	Výzkumný tým	Velikost instance	Název
1954	G. Dantzig, R. Fulkerson, and S. Johnson	49 měst	dantzig42
1971	M. Held and R.M. Karp	64 měst	64 random points
1975	P.M. Camerini, L. Fratta, and F. Maffioli	67 měst	67 random points
1977	M. Grötschel	120 měst	gr120
1980	H. Crowder and M.W. Padberg	318 měst	lin318
1987	M. Padberg and G. Rinaldi	532 měst	att532
1987	M. Grötschel and O. Holland	666 měst	gr666
1987	M. Padberg and G. Rinaldi	2 392 měst	pr2392
1994	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	7 397 měst	pla7397
1998	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	13 509 měst	usa13509
2001	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	15 112 měst	d15112
2004	D. Applegate, R. Bixby, V. Chvátal, W. Cook, and K. Helsgaun	24 978 měst	sw24798

Obr. 28. Tabulka vývoje řešení TSP dle velikosti instance [11]

Ve srovnání s instancemi v tabulce je problém řešený touto prací ohraničen množstvím pozic na regálové systému, které je 6 367, což tedy v teoretickém maximu odpovídá problematice vyřešené přibližně v roce 1994. Praktické maximum úlohy je 535 pozic, což přibližně odpovídá úloze vyřešené v roce 1987. Z uvedeného srovnání vyplývá, že podobná problematika by ještě v době před dvaceti lety byla velmi obtížně řešitelnou, ne-li neřešitelnou v konečném čase. Ovšemže zejména z hlediska

technických omezení. Pro zajímavost nejnáročnější řešení z roku 2004 propojilo 24 978 švédských měst trasou dlouhou 72 500 km.



Obr. 29. Propojení 24 978 švédských měst [11]

4.3 Principy řešení

Společným znakem všech algoritmů pro optimalizační úlohy typu TSP je prohledávání stavového prostoru přípustných řešení a výběr optimálního. Exaktní metody obvykle vyhodnotí řešení přímo, nebo použijí k vyhodnocení částečné, případně přibližné řešení dané úlohy. Přímé kompletní řešení TSP předpokládá znalost všech přípustných permutací z každého města posloupnosti, což je v praxi neproveditelná výpočetní úloha již pro malý počet měst. Časová složitost problému je dána výrazem $O(n!)$.

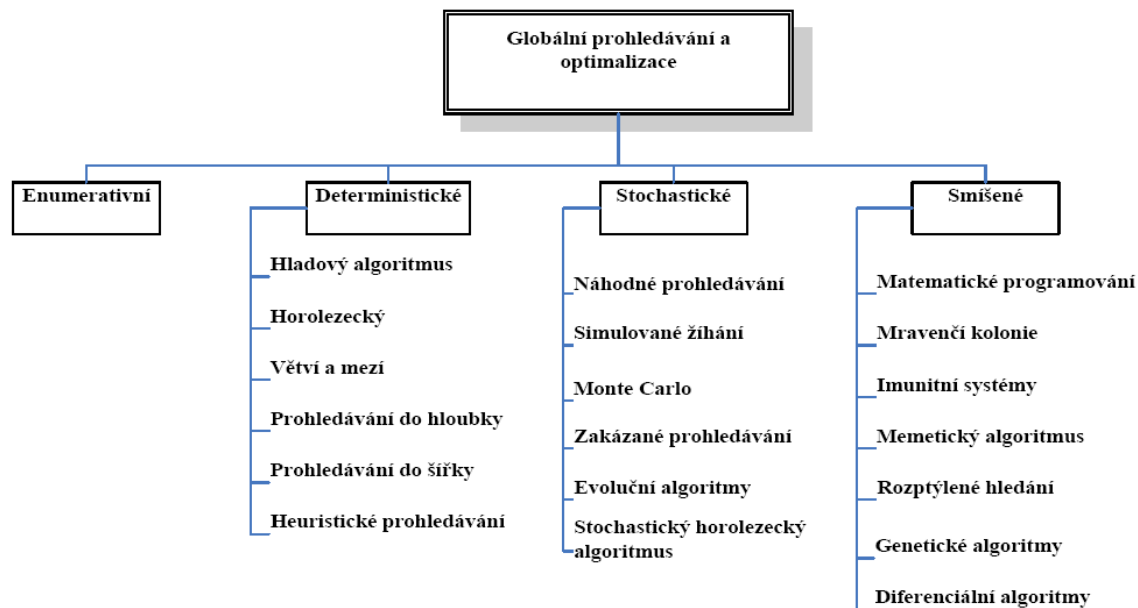
$$10! = 3628800 \approx 0,36 \cdot 10^7$$

$$20! = 2432902008176640000 \approx 0,24 \cdot 10^9$$

·

·

$$100! = 9332621544394415268169923885626670049071596826438162146859296389521759999322991560894146397615651828625369792082$$



Obr. 30. Rozdělení prohledávacích algoritmů – převzato z [8]

„Jednotlivé třídy algoritmů představují obecně způsob řešení konkrétního problému pomocí metod s různým stupněm efektivity a složitosti a lze je charakterizovat následujícím způsobem:

a) **enumerativní** – jedná se o výpočet všech možných kombinací argumentů daného problému. Tento přístup je vhodný pro problémy, u nichž jsou argumenty účelové funkce diskrétního charakteru a nabývají malého množství hodnot. Pokud by byl tento přístup použit obecně, zcela reálně by mohl potřebovat na úspěšné ukončení více času, než je doba existence vesmíru;

b) **deterministické** – tato skupina algoritmů je postavena pouze na rigorózních metodách klasické matematiky. Algoritmy tohoto charakteru obvykle vyžadují předběžné předpoklady, jež umožní této metodě podávat efektivní výsledky. Jsou to obvykle následující předpoklady:

- i. problém je konvexní;
- ii. prohledávaný prostor možných řešení je spojitý;
- iii. účelová funkce má pokud možno pouze jeden extrém (je unimodální);
- iv. problém je definován v analytickém tvaru;

c) **stochastické** – algoritmy tohoto typu jsou založeny na využití náhody. Jde v podstatě o čistě náhodné hledání hodnot argumentů účelové funkce s tím, že výsledkem je vždy to nejlepší řešení, které bylo nalezeno během celého náhodného hledání. Tyto typy algoritmů jsou většinou vhodné pro řešení úloh

menšího rozsahu, avšak existují i jisté modifikace těchto algoritmů, které lze použít i pro úlohy většího rozsahu;

d) **smíšené** – tato třída algoritmů představuje směs metod deterministických a stochastických, které ve vzájemné spolupráci dosahují překvapivě dobrých výsledků. Poměrně silnou skupinou těchto algoritmů jsou evoluční algoritmy. Tyto smíšené algoritmy:

- i. jsou robustní, což znamená, že nezávisle na počátečních podmínkách často naleznou kvalitní řešení;
- ii. jsou efektivní a výkonné, tzn., že jsou schopné nalézt kvalitní řešení během relativně malého počtu ohodnocení účelové funkce;
- iii. jsou odlišné od čistě stochastických metod, a to díky přítomnosti podmnožiny deterministických metod;
- iv. mají minimální nebo žádné požadavky na předběžné informace;
- v. jsou schopné pracovat s problémy typu „černá skříňka“, tzn., že nepotřebují ke své činnosti analytický popis problému;
- vi. má-li účelová funkce globální extrém ve více argumentech, tyto algoritmy jsou schopny nalézt více než jeden takovýto argument.

Po shrnutí předešlých informací lze konstatovat, že:

- a) enumerativní a deterministická optimalizace není vhodná na problémy, u nichž se prohledává rozlehlý prostor možných řešení;
- b) stochastické algoritmy pracují dobře na problémech, u nichž se prohledává úzký prostor možných řešení, avšak po určitých modifikacích daných algoritmů jsou vhodné i pro použití na větší rozsahy možných řešení;
- c) smíšená optimalizace je vhodná na problémy bez omezení velikosti jejich prostoru možných řešení.“ [8]

4.5 Některé vybrané optimalizační algoritmy užívané pro řešení TSP

V následujících řádcích budou uvedeny některé nejčastěji používané algoritmy v rámci optimalizace hledání nejkratší hamiltonovské cesty.

Jedním z nejzákladnějších a relativně nejjednodušších je algoritmus lokálního prohledávání nebo také jinak řečeno horolezecký algoritmus. Jedná se o gradientní metodu, jejíž výsledek bývá často využíván jako počáteční řešení, které je následně optimalizováno některou stochastickou heuristickou metodou. Stručné naznačení v pseudokódu.

Horolezecký algoritmus:

- 1) vyber start uzel;
- 2) last:=start;
- 3) **while** existuje nenavštívený uzel **do**
- 4) z last jdi do nejbližšího uzlu
- 5) označ uzel, ve kterém stojíš jako last;
- 6) označ uzel, ze kterého jsi šel jako navštívený;
- 7) **end**;
- 8) z last přejdi do start.

Výhodou tohoto algoritmu je relativní jednoduchost a rychlost. Nevýhodou je obecný problém gradientních metod a tím je náchylnost k uvíznutí v dílčích lokálních minimech, ze kterých je „násilně“ vymaněn, nicméně obvyklá cena je, že globální řešení není zdaleka optimální. V aplikaci na problematice řešené touto diplomovou prací by to v praxi znamenalo, že kdybychom měli obsloužit všechny pozice jedné regálové řady v pátém patře a k nim několik málo z přízemí, algoritmus by prošel přímkou páté patro a teprve poté by se přes celou délku řady vrátil do přízemí. Výsledek by byl dost vzdálen optimu. Pro odstranění tohoto handicapu bývá často užíváno tzv. metaheuristik, které algoritmu pomáhají najít z nalezených lokálních řešení v každé iteraci nejbližší globální => ne vždy je lokální minimum v součtu rovno globálnímu. Opět zpátky k demonstrovanému příkladu s regálovou řadou. Metaheuristika by v tomto případě zařídila, že by se přerušilo přecházení z uzlu na uzel v pátém patře a v nejbližším možném uzlu by se přeskočilo na uzel přízemí i když přechod k němu není v danou chvíli lokálním minimem.

Genetický algoritmus:

- 1) $t := 0$
- 2) Initialize $G(0)$ / inicializuj počáteční generaci
- 3) Evaluate $G(0)$ / proved' ohodnocení
- 4) **while** not Done **do** / dokud není splněna ukončovací podmínka
- 5) $t := t + 1$
- 6) Select $G(t)$ from $G(t-1)$ / proved' přirozený výběr
- 7) Crossover $G(t)$ / aplikuj křížení
- 8) Mutate $G(t)$ / aplikuj mutaci
- 9) Evaluate $G(t)$ / proved' ohodnocení
- 10) **End**.

„Genetické algoritmy pracují tím způsobem, že se nejprve vytvoří počáteční populace o velikosti p jedinců a pak se tato populace mění pomocí genetických operátorů tak dlouho, dokud není splněna nějaká podmínka ukončení. Počáteční populace se většinou získá náhodným generováním. Byly provedeny také pokusy nasadit do počáteční populace kvalitní řešení, získaná jinými heuristickými metodami, ale při tomto způsobu se ukázalo nebezpečí předčasné konvergence do nějakého ještě ne dost dobrého lokálního optima. Pokud jde o velikost p populace, je jasné, že příliš malá populace může zavinit špatné pokrytí prostoru řešení, zatímco velká populace zvyšuje výpočetní náročnost. Zkušenosti ukazují, že pro většinu problémů je dostačující

populace o velikosti 50 až 200 jedinců.“ [8] Hlavním problémem nasazení genetického algoritmu je nalezení přesného optima a v neposlední řadě poměrně velká výpočetní náročnost pro vysoké množství vyhodnocování cílové funkce.

Simulované žíhání:

```

1)  $x :=$  náhodně vygenerované řešení;
2)  $T := T_{\max}$ ;  $x^* := x$ ;  $k := 1$ ;
3) while ( $T > T_{\min}$ ) and ( $k > 0$ ) do
4)   begin  $t := 0$ ;  $k := 0$ ;
5)     while ( $t < t_{\max}$ ) and ( $k < k_{\max}$ ) do
6)       begin  $t := t + 1$ ;
7)          $x' := \text{transformace}(x)$ ;
8)         if  $f(x') \leq f(x)$  then  $Pr := 1$ 
9)           else  $Pr := \exp(- (f(x') - f(x))/T)$ ;
10)        if  $\text{random} < Pr$  then
11)          begin  $x := x'$ ;  $k := k + 1$ ;
12)            if  $f(x) < f(x^*)$  then  $x^* := x$ ;
13)          end;
14)        end;
15)      $T := \alpha \cdot T$ ;
16) end.
```

Pozn.: x^* je nejlepší nalezené řešení v průběhu běhu algoritmu;

T je koeficient teploty;

α je koeficient ochlazování

„Celočíselné proměnné t a k jsou počítadla pro vnější, resp. vnitřní while-cyklus. Proměnná t zaznamenává celkový počet pokusů algoritmu pro danou teplotu T , zatímco proměnná k zaznamenává počet úspěšných pokusů, které byly akceptovány Metropolisovým vzorcem. Pro volbu maximálních hodnot $t_{\max}$ a $k_{\max}$ neexistuje všeobecný předpis, $t_{\max}$ a $k_{\max}$ jsou obvykle ve vztahu k velikosti množiny sousedů, hodnota $k_{\max}$ je volena od několika set do několika tisíc a $t_{\max} = 10 * k_{\max}$. Volba jednotlivých parametrů metody je obvykle uskutečněna na základě zkušeností po mnoha experimentech. Podmínka ukončení algoritmu je splněna buď dosažením minimální teploty, nebo tzv. zmrazením krystalu, tj. neuskutečněním při dané teplotě ani jednoho úspěšného pokusu z celkového počtu $t_{\max}$ pokusů o „vykmitnutí z pevné pozice v krystalické mřížce“. Reálná proměnná random je náhodně generované číslo z intervalu (0,1). Proměnná x^* zaznamenává nejlepší řešení v průběhu provádění celého algoritmu. Ve všeobecnosti proměnná x nemusí po skončení SA obsahovat nejlepší řešení. Výkon algoritmu silně závisí na typu snižování teploty. Nikoliv překvapivě bylo navrženo mnoho druhů snižování teploty.“ [8]

Zakázané prohledávání:

```

1)  $x :=$  náhodně vygenerované řešení;
2)  $\text{time} := 0$ ;  $f_{\min} := \infty$ ;  $T := \emptyset$ ;
3) while  $\text{time} < \text{timemax}$  do
```

```

4) begin  $time := time + 1$ ;  $f_{loc-min} := \infty$ ;
5)       for  $t \in S$  do
6)       begin  $x' := tx$ ;
7)           if  $f(x') < f_{loc-min}$  and ( $t \notin T$  or  $f(x') < f_{min}$ ) then
8)               begin
9)                    $x^* := x'$ ;  $t^* := t$ ;  $f_{loc-min} := f(x')$ ;
10)              end;
11)       end;
12)       if  $f_{loc-min} < f_{min}$  then
13)           begin
14)                $f_{min} := f_{loc-min}$ ;  $x_{min} := x^*$ ;
15)           end;
16)    $x := x^*$ ;
17)   if  $|T| < k$  then  $T := T \cup \{t^{*-1}\}$ 
18)       else  $T := (T \cup \{t^{*-1}\}) / \{t^{\wedge}\}$ ;
19) end.

```

Metoda zakázaného prohledávání vychází z horolezeckého algoritmu, který modifikuje k odstranění zacyklení při uváznutí v lokálním minimu. „Hlavní myšlenka heuristiky je realizována pomocí zakázaného seznamu T , který dočasně obsahuje inverzní transformace k transformacím použitým v předcházejících iteracích. Zakázaný seznam transformací T je omezen svou maximální velikostí k_{max} . Zakázaný seznam T je obnovován v průběhu chodu celého algoritmu. Jestliže transformace t patří do zakázaného seznamu, pak se nemůže používat v lokální minimalizaci v rámci okolí aktuálního řešení. Při inicializaci algoritmu je zakázaný seznam prázdný, po každé iteraci se do zakázaného seznamu přidá transformace inverzní k transformaci, která poskytla nejlepší řešení v rámci sousedství v předcházejícím kroku. (např. se do zakázaného seznamu zapíše pořadová čísla vyměněných úloh). Zakázaný seznam musí být kratší, než je počet možných transformací v rámci sousedství. Po k_{max} iteracích zakázaný seznam už obsahuje k_{max} transformací a každé další přidání nové transformace je doprovázeno vyloučením momentálně nejstarší transformace $t^{\wedge}$ v tabu seznamu. Říkáme, že zakázaný seznam se cyklicky obnovuje.

Numerické zkušenosti s algoritmem zakázaného prohledávání ukazují, že velikost k_{max} zakázaného seznamu je důležitým parametrem ovlivňujícím možnost vymanit se z lokálních minim. Jestliže je parametr k_{max} příliš malý, pak se může vyskytnout zacyklení algoritmu, stejně jako u klasického horolezeckého algoritmu, ale například po více krocích. Naopak je-li parametr k_{max} příliš velký, potom s velkou pravděpodobností přeskočíme nadějná hluboká údolí funkce.“ [8]

Závěrem stručného přehledu nejčastěji užívaných algoritmů pro řešení problému obchodního cestujícího krátké zhodnocení. Neexistuje obecné pravidlo, které by prikazovalo použití některého z algoritmů na konkrétní úlohu. Pouze vhodným rozbořením řešené problematiky, vymezením reprezentace stavového prostoru a definicí účelové funkce lze společně se znalostí chování jednotlivých algoritmů vybrat vhodného kandidáta k implementaci. Je nasnadě, že nejvhodnější jsou vzájemné kombinace metod s využitím maxima znalostí o řešené úloze. Vztaženo na předmětnou problematiku této práce je to zejména znalost plošného i prostorového rozložení uzlů na mapě skladu a z toho plynoucí omezení pro prohledávání stavového prostoru. Definice

start bodu. Pravoúhlost prostoru i hledané trasy. Strategie postupu systémem s ohledem na množství obslužné techniky, metody uskladňování do pozic, konsolidace zboží dle klienta, šarže, PNC a postavení regálových řad. Nejzásadnějším kritériem je rychlost zpracování až řádově stovek uzlů, což je podmínka praktického nasazení implementované optimalizační metody. Všechny tyto specifické vlastnosti a požadavky mají vliv na volbu metody řešení optimalizace naší úlohy.

4.6 Minimální kostra grafu

Dle teorie grafu je kostra (faktor) souvislého grafu G jeho stromem (graf, který neobsahuje cyklus) spojujícím všechny uzly grafu G . Koster na obecném grafu je možno nalézt n^{n-2} , kde n je počet vrcholů.

Pakliže se hovoří o ohodnoceném grafu, kdy hodnoty hran reprezentují vzdálenosti mezi uzly, je minimální kostrou myšlena právě ta kostra, jejíž součet hran je mezi všemi kostrami na grafu minimální. Minimální kostra je tedy v tomto případě zároveň minimální spojnicí všech uzlů v grafu.

Pakliže v grafu existuje minimální kostra, která neobsahuje uzly stupně vyššího než dva, jedná se zároveň o hamiltonovskou cestu, tedy nejlevnější spojnicí všech uzlů grafu, kdy každý uzel je incidentní pouze s hranou vstupující nebo, případně zároveň, vystupující. Opačně řečeno je zřejmé, že hamiltonovská cesta je zároveň kostrou grafu. Jestliže je zároveň minimální kostrou grafu, jedná se o minimální hamiltonovskou cestu. „Odstraníme-li ze zadaného grafu hranu, která neleží na nejkratší hamiltonovské cestě, řešení úlohy (tj. nejkratší hamiltonovská cesta) se nezmění.“[4]

Algoritmus hledání nejlevnější hamiltonovské cesty

- 1) $w_{\min} := \infty$;
- 2) Najdeme minimální kosteru K daného grafu G .

(a) Tvoří-li tato kostra hamiltonovskou cestu (tj. všechny vrcholy v této kostře mají stupeň ≤ 2), pak tato kostra je nejkratší hamiltonovskou cestou (kdyby totiž existovala nějaká kratší hamiltonovská cesta, byla by zároveň kratší kostrou, což by byl spor).

(b) K obsahuje vrchol x tak, že $d_K(x) = k \geq 3$ (stupeň vyšší než 2), a tedy některá z hran z $H_K(x)$ neleží ve výsledné nejkratší hamiltonovské cestě. Sestrojíme zmenšené grafy $G_1, G_2, \dots, G_k$ takové, že $H(G_i) = H(G) - \{h_i \text{ vycházející z vrcholu } x\}$, $i = 1, \dots, k$ a zkusíme vyhledat minimální kostry zmenšených grafů.

i. G_i vynětím hrany přestal být graf souvislý, tj. neobsahuje již žádnou kosteru (není třeba dále zkoumat).

- ii. Minimální kostra grafu G_i tvoří hamiltonovskou cestu, délku (cenu) této cesty srovnáme s délkou dosud nejkratší hamiltonovské cesty a minimum si zaznamenáme

```

if  $w(K(G_i)) < w_{\min}$ 
then begin  $w_{\min} := w(K(G_i))$ 
    zaznamenání hamiltonovské cesty
end;

```

iii. Nejlevnější kostra grafu G_i *netvoří* hamiltonovskou cestu.

```

if  $w(K(G_i)) > w_{\min}$ 
then podúlohu dále nezkoumáme, protože vytržením hrany z původního
    grafu bez respektování optimálního pořadí vypouštění hran nemůže délka
    minimální kostry klesnout (kosta zmenšeného grafu je i kostrou v
    původním grafu), řešení podúlohy  $G_i$  tedy určitě nebude lepší než  $w_{\min}$ 
    (bude-li vůbec existovat)
else rekurzivně vyvoláme krok (b) pro  $G_i$ , tj. vyřazujeme hrany z  $G_i$  atd.

```

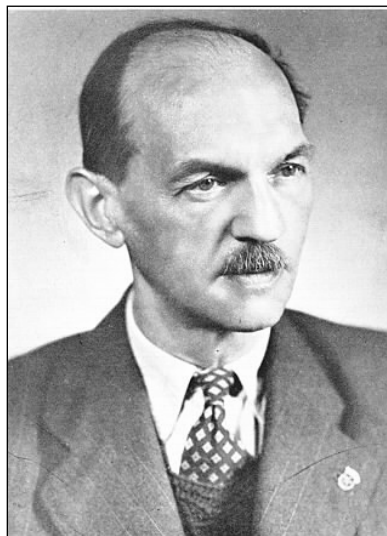
Algoritmus převzat z [4]

Popsaný algoritmus hledání nejlevnější hamiltonovské kostry se nazývá metodou větví a mezí. Jeho užitím lze nalézt optimální hamiltonovskou cestu, ovšem praktická použitelnost je omezená na problémy malého rozsahu, neboť vede na mnohonásobná výpočetní větvení v rámci odstraňování uzlů stupně vyššího než dva.

Algoritmy pro hledání nejlevnější kostry na souvislém grafu se zabývala řada matematiků, kdy mezi přední objevitele a průkopníky patří prof. Borůvka (1899–1995), který působil na Masarykově univerzitě v Brně, a prof. Jarník (1897–1970), který krátce pracoval na české technice v Brně.



Obr. 31. Otakar Borůvka [14]



Obr. 32. Vojtěch Jarník [14]

5. Praktické řešení aplikace

Praktické řešení popisuje výběr vývojového prostředí, konceptuální model aplikace, datové struktury, řešení I/O, interface uživatelského prostředí a rovněž podrobně vysvětluje některé vybrané speciálně navržené funkce a procedury. Závěrem statisticky shrnuje rozsah vytvořeného díla včetně předpokladů a podmínek provozování aplikace v praxi.

Než přejdeme k jednotlivým řešeným problematikám, nejdříve několik slov k pojmenování aplikace. Každý systém má právo na svou jedinečnou a jednoznačnou identifikaci, která se uživatelům lehce vryje do paměti a kterou poté nadále užívají při vzájemné komunikaci. Při výběru názvu řešeného problému byl brán zřetel na jeho funkci a účel. S ohledem na jednoduchost a výše uvedené byl zvolen název OPCES, což je akronym utvořený ze slov **o**ptimální **c**esta. Za názvem pokračuje číselná hodnota, která identifikuje verzi aplikace.

5.1 Vývojové prostředí

Jedním ze zásadních rozhodnutí před započatím práce nad aplikací je výběr vhodného vývojového prostředí. Programovací nástroj by měl splňovat pokud možno následující kritéria:

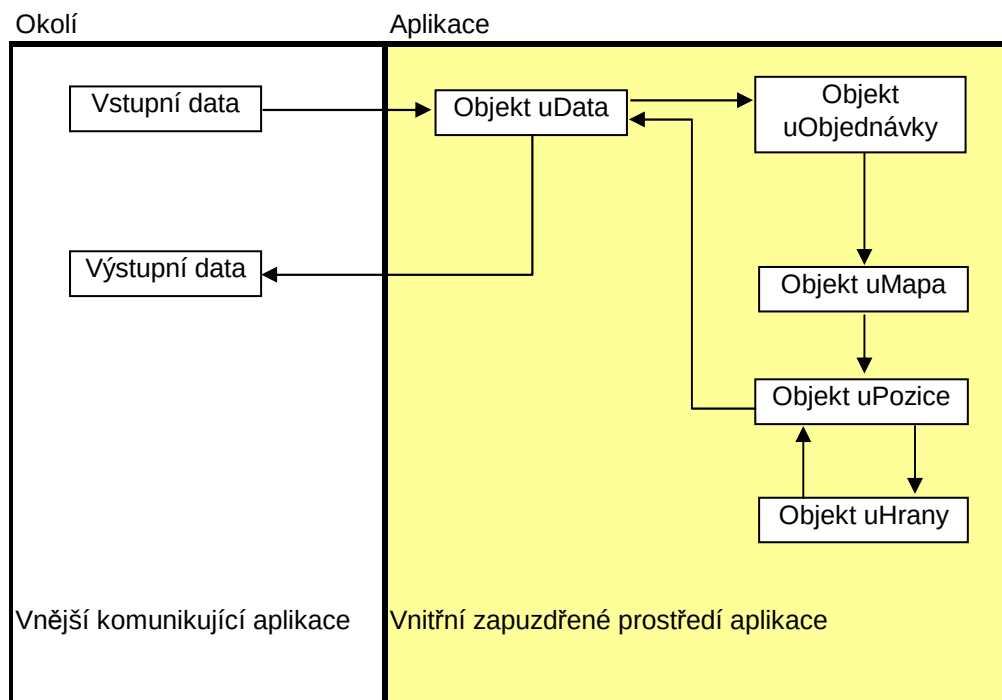
- objektově orientovaný vyšší programovací jazyk;
- vyspělé komponenty;
- pokud možno příjemně řešit vstup a výstup dat do nebo z aplikace;
- maximálně podporovat práci s dynamickými sadami záznamů;
- nebýt přísně typově vyhraněný;
- mít implementovány základními funkcemi pro práci s řetězci;
- mít implementovány běžné matematické a logické funkce;
- kvalitní kontrola chyb syntaxe kódu;
- vyspělé možnosti ladění programu, zejména pak sledování změn proměnných za běhu;
- praktické zkušenosti autora s vývojovým prostředím.

Z výše uvedených kritérií pro výběr vývojového prostředí byly závěrem vybrány dva kandidáti, a to zejména s přihlédnutím na zkušenosti autora s nimi. Prvním systémem je MS Access, ve kterém jsem řešil bakalářskou práci, a druhým je Delphi 7, ve kterém byla řešena řada semestrálních projektů. První systém má nespornou výhodu v uživatelsky příjemném řešení uložení dat, v podpoře výstupů a jejich výměně s ostatními aplikacemi. Jednoznačnou nevýhodou je, že se jedná pouze o interpreta, nikoliv o překladač. Program by tedy mohl běžet pouze na strojích vybavených systémem MS Access, a to navíc jen určité verze, neboť kompatibilita mezi verzemi nástrojů řady MS Office je všeobecně nedůvěryhodná, zejména pak v prostředí MS Access zkušenostmi ověřená.

Naproti tomu systém Delphi 7 poskytuje veškeré výhody moderního vývojového prostředí na základě vyššího programovacího jazyka Object Pascal a i když je nutné vynaložit vyšší úsilí do řešení některých problémů, například typu export/import dat z/do aplikace, padlo rozhodnutí o výběru vývojového prostředí na tento nástroj. Výsledkem praktické části diplomové práce tedy bude přeložený a zkompileovaný exe soubor, spustitelný v prostředí operačního systému MS Windows. Přirozenou otázkou je, proč nebylo zahrnuto více systémů do výběru, neboť dozajista mohou vykazovat vyšší vhodnost použití. Odpovědí je poslední kritérium z výše uvedeného výčtu.

5.2 Konceptuální model aplikace

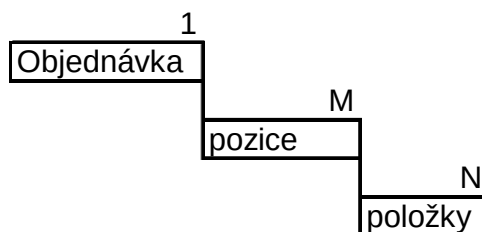
Myšlenkový model aplikace řeší koncept implementace datového toku od vstupu do aplikace po uchovávání dat v aplikaci přes práci se sadami záznamů uvnitř aplikace až po vyhodnocení a výstup sady dat z aplikace. Model informačního toku přes datové objekty je zobrazen na obr. 33.



Obr. 33. Datový model aplikace a rozhraní s okolím

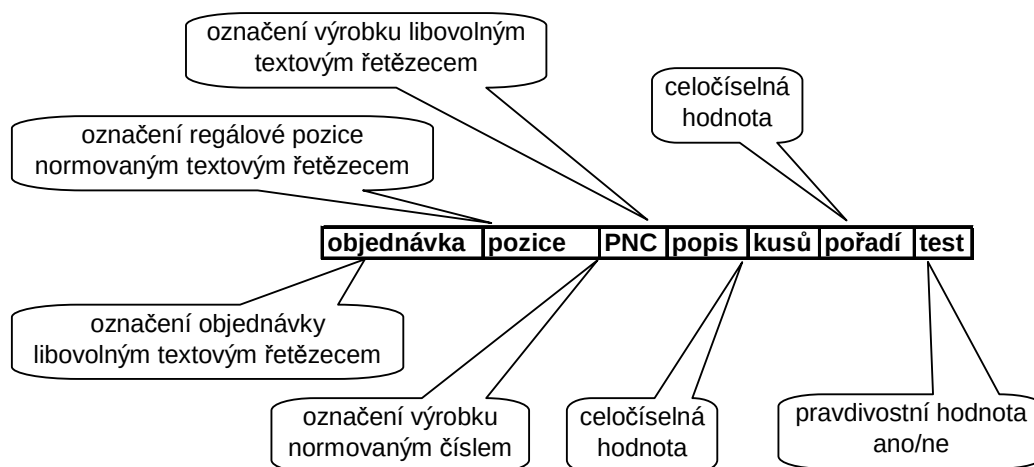
Popis jednotlivých objektů a jejich vazeb

1. Vstupní data jsou sada záznamů vyexportovaná z informačního systému skladového hospodářství. Data obsahují informace o objednávkách, pozicích, které je třeba navštívit, PNC kódu položky, popisu položky a počtu kusů, které je třeba vyskladnit z pozice. Jedná se tedy o skupinu objednávek rozpadajících se na skupinu pozic rozpadajících se na položky. Rozpad je naznačen na obr. 34.



Obr. 34. Struktura vstupních dat

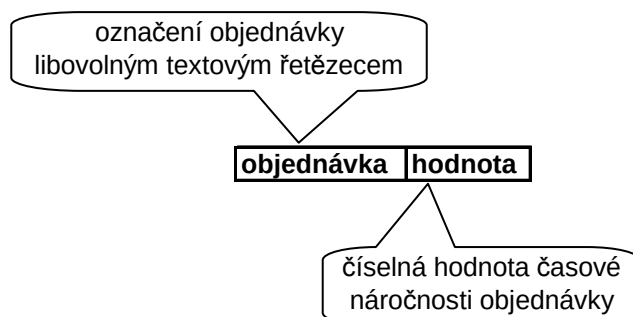
2. Objekt `uData` je zapouzdřením importovaných dat v aplikaci. S okolím komunikuje pomocí přístupových vlastností a metod. Struktura dat je rovnocenná se vstupními daty, jen přibýly následující příznaky: pořadí, testováno. Veškerá uložená data uvnitř objektu jsou typu `string` (textový řetězec). Příznak pořadí po vyhodnocení určuje permutaci zpracování pozic a příznak testováno slouží k označení objednávky v rámci zpracovávané dávky. Naznačení struktury dat je na obr. 35.

Obr. 35. Struktura a typ dat objektu `uData`

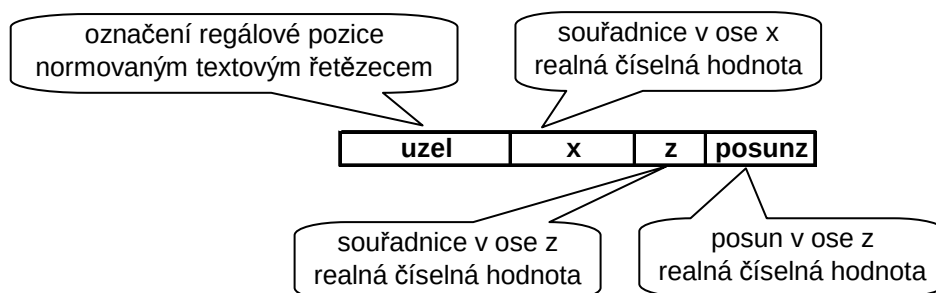
3. Objekt `uObjednávky` je zapouzdřením jedinečných identifikátorů objednávek z právě zpracovávané dávky. S okolím komunikuje pomocí přístupových vlastností a metod. Struktura dat je jednoduchá, jde o dvě hodnoty. První je identifikátor objednávky a druhá její hodnota. Hodnota vyjadřuje časovou náročnost na obsloužení právě jedné konkrétní objednávky. Veškerá data uvnitř jsou typu `string` (textový řetězec). Hodnoty identifikátoru jsou načítány z objektu `uData` a ohodnocení objednávky je dopočítáváno po vyhodnocení celkové délky trasy dané expediční dávkou. Objednávky z objektu `uObjednávky` mají příznak `true`

v objektu `uData` a nejsou tedy zpracovány, přenášeny do objektu `uPozice`. Veškerá data uvnitř objektu jsou typu `string` (textový řetězec). Naznačení struktury dat je na obr. 36.

4. Objekt `uMapa` je zapouzdřením informací o rozložení hranic regálových řad na ploše skladu vůči počátečnímu bodu. Zjednodušeně řečeno se jedná o mapu hraničních pozic v plošných souřadnicích vůči start pozici. S okolím komunikuje pomocí přístupových vlastností a metod. Struktura dat objektu `uMapa` je následující: uzel reprezentuje ve zkráceném tvaru adresu hraniční pozice (zkrácený tvar z důvodu vyhovění adresám všech pater regálu), x je souřadnice v ose x , z je souřadnice v ose z a posun $v z$ je vzdálenost posunu pozice příčného regálu na nejbližší ulici, což je užito v případě průchodu napříč řadami k téměř protějším příčnému regálu. Zde totiž nefunguje Manhattanská metrika (viz kapitola 3.1), protože $|z_1 - z_2| + |x_1 - x_2| \neq$ vzdálenosti (p_1, p_2) jelikož nelze projít napříč regálovou řadou, takže je nutný posun na nejbližší ulici. V tomto případě platí $(\text{posun } v z_1 + \text{posun } v z_2) + |x_1 - x_2| = \text{vzdálenost } (p_1, p_2)$. Veškerá data uvnitř objektu jsou typu `string` (textový řetězec). Naznačení struktury dat je na obr. 37.



Obr. 36. Struktura a typ dat objektu `uObjednavky`

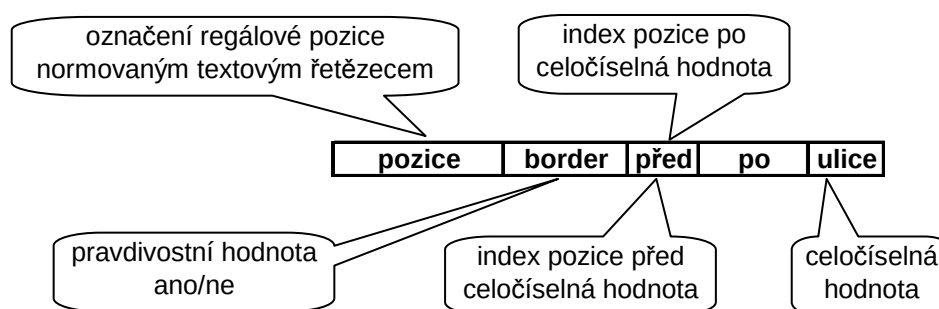


Obr. 37. Struktura a typ dat objektu `uMapa`

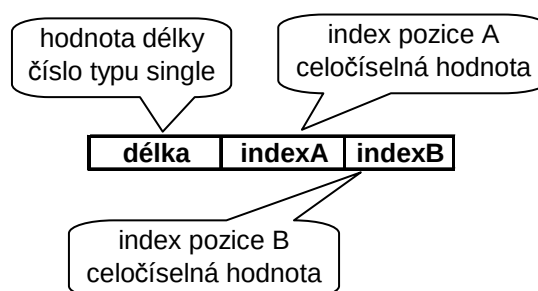
5. Objekt `uPozice` je zapouzdřením informací o pozicích, které je nutné navštívit při průchodu regálovým systémem. Tento datový objekt je základním kamenem pro výpočet optimální trasy. S okolím komunikuje pomocí přístupových vlastností a metod. Struktura dat objektu `uPozice` je

následující: pozice reprezentuje normovaný tvar adresy uzlu v regálovém systému, border je příznak příslušnosti uzlu k hraničním, před je index uzlu předcházejícího v permutaci, po je index uzlu následujícího v permutaci a ulice je číslo ulice, ve které se nachází příslušná pozice. Veškerá data uvnitř objektu jsou typu string (textový řetězec). Naznačení struktury dat je na obr. 38.

- Objekt uHrany je zapouzdřením informací o hranách spojujících uzly. Tento datový objekt slouží ke shromažďování údajů o hranách z aktuálně zkoumané oblasti, jejich ohodnocení a následné předání informací objektu uPozice. Struktura dat objektu uHrany je následující: délka je hodnotou vzdálenosti z pozice A do pozice B, index A je ukazatel na prvek A v uPozice, index B je ukazatel na prvek B v uPozice. Veškerá data uvnitř objektu jsou typu string (textový řetězec). Naznačení struktury dat je na obr. 39.



Obr. 38. Struktura a typ dat objektu uPozice



Obr. 39. Struktura a typ dat objektu uHrany

- Výstupní data jsou sada záznamů vyexportovaná z aplikace. Data obsahují informace o pozicích, které je třeba navštívit, PNC kódu položky, popisu položky a počtu kusů, které je třeba vyskladnit z pozice. Oproti vstupním datům neobsahují informaci o objednávkách, která není pro obsluhu expediční dávky důležitá. Krom tohoto je jediný rozdíl oproti vstupním datům v uspořádání posloupnosti pořadí, které odpovídá optimální cestě průchodu regálovým systémem.

Objekty byly naprogramovány jako vlastní třídy, jejichž vlastnosti a metody přístupu jsou strukturovaně popsány v tabulce na obr. 40. Pro praktické použití hraje velkou roli hodnota spotřebovaného času na jeden výpočet optimální trasy při střední hodnotě pozic 314. S ohledem na tuto skutečnost a výše zmíněné vlastnosti regálového systému je navržen následující metaheuristický postup výpočtu optimální trasy expediční dávky:

- 1) definuj regálové pozice expediční dávky do objektu uPozice;
- 2) nastav hraniční uzly pro vstup do obsazených regálových řad;
- 3) vykalkuluj průchod regálovým systémem dle stanovené strategie;
- 4) vykalkuluj průchod všemi potřebnými ulicemi dle zadané strategie;
- 5) spočti celkovou délku trasy.

Objekt	Metody a vlastnosti	Popis
uData	Create	Vytvoří objekt, alokuje paměť.
	Destroy	Zruší objekt, uvolní paměť.
	Adresa	Zapíše nový záznam nebo nalezne zadaný záznam.
	Velikost	Vrátí velikost objektu.
	opr, apr, ppr, tpr, kpr, pncpr, poppr	Vrátí hodnotu prvku z předaného indexu nebo zapíše hodnotu prvku na předaný index.
	Smaz	Smaže veškerá data z objektu.
	Serad	Seřadí vzestupně záznamy v objektu dle hodnoty prvku pořadí.
	TestUzlu	Vrátí true když pozice je v uData.
uObjednavky	Create	Vytvoří objekt, alokuje paměť.
	Destroy	Zruší objekt, uvolní paměť.
	Dej	Vrátí index objednávky předané parametrem.
	Pridej	Přidá jedinečnou objednávku a její hodnotu.
	Velikost	Vrátí velikost objektu.
	opr, upr	Vrátí hodnotu prvku z předaného indexu nebo zapíše hodnotu prvku na předaný index.
	Smaz	Smaže veškerá data z objektu.
	Serad	Seřadí sestupně záznamy v objektu dle hodnoty úspory.
uMapa	Create	Vytvoří objekt, alokuje paměť.
	Destroy	Zruší objekt, uvolní paměť.
	Dej	Vrátí index pozice předané parametrem.
	Pridej	Přidá jedinečnou hraniční adresu včetně hodnot.
	Velikost	Vrátí velikost objektu.
	upr, zpr, xpr, pzpr	Vrátí hodnotu prvku z předaného indexu.
uPozice	Create	Vytvoří objekt, alokuje paměť.

	Destroy	Zruší objekt, uvolní paměť.
	Pozice	Přidá jedinečnou pozici do objektu včetně hodnot nebo najde pozici v objektu dle předaných parametrů.
	Hranice	Přidá hraniční prvek do objektu.
	Velikost	Vrátí velikost objektu.
	VelikostP	Vrátí velikost permutace optimální cesty.
	ppr, bpr, PRpr, POpr, upr	Vrátí hodnotu prvku z předaného indexu nebo zapíše hodnotu prvku na předaný index.
	DejH	Vrátí pro číslo řady předané parametrem hranici horní a dolní (sloupec 1 nebo 29) plus patro hraničního bodu odvozeno od nejbližšího mimohraničního bodu. Když je result nula, tak řada není v dávce, když je result jedna, tak je hranice řady na sloupci 1, když je result dva, tak je hranice na sloupci 29, když je result tři, tak je hranice na sloupci 1 i 29, tedy ulice se musí projít od začátku do konce. Result je složen ze tří nebo pětiznakového řetězce. První znak vrací předešle popsané, druhé dva znaky vrací patro na hranici sloupce 1 (když je) a poslední dva vrací patro na hranici sloupce 29.
	DejI	Vrátí index pozice předané parametrem.
	TestJH	Vrátí true, když existuje hraniční uzel mimo vyšetřovanou cestu.
	TestChU	Vrátí true, když existuje uzel v ulici předané parametrem mimo vyšetřovanou cestu.
	Test10	Vrátí true, když existuje nenavštívená pozice v řadě 10 mimo vyšetřovanou cestu.
	NastavPermutaci	Nastaví permutaci indexů pozic dle optimální cesty.
	DejPer	Vrátí hodnotu permutace optimální cesty na zadaném indexu.
	Smaz	Smaže veškerá data z objektu.
uHrany	Create	Vytvoří objekt, alokuje paměť.
	Destroy	Zruší objekt, uvolní paměť.
	Zapis	Vloží hranu do objektu.
	Velikost	Vrátí velikost objektu.
	dpr, apr, bpr	Vrátí hodnotu prvku z předaného indexu.
	Serad	Seřadí vzestupně záznamy v objektu dle hodnoty délky hrany.

Obr. 40. Vlastnosti a metody objektů

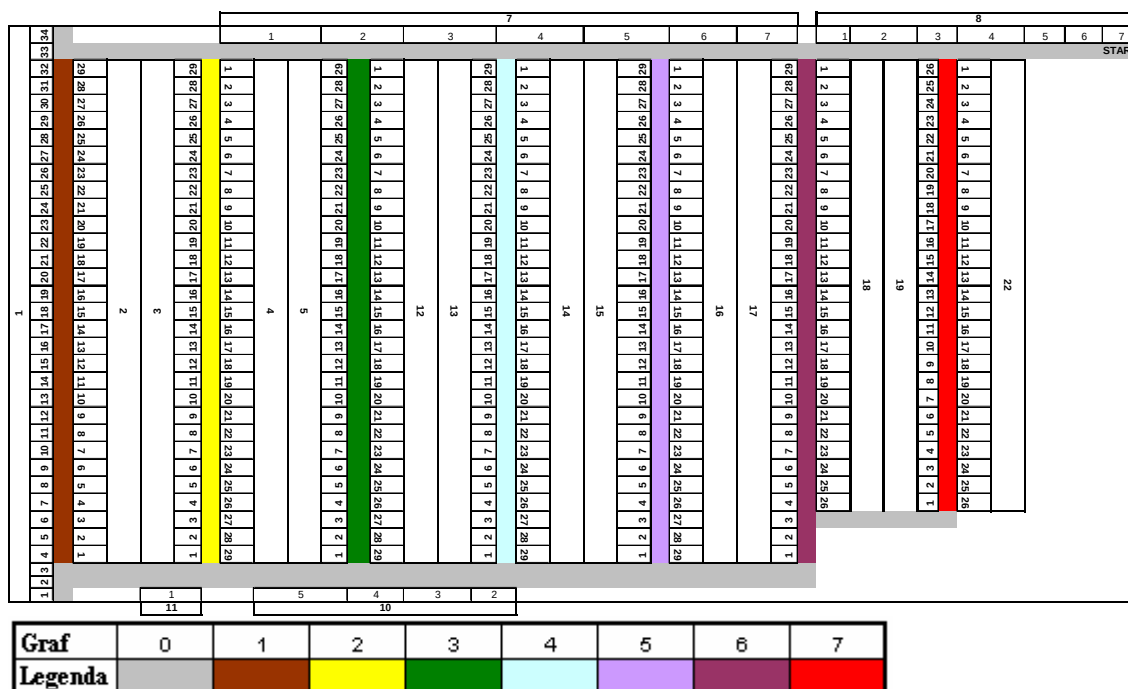
5.3 Grafy a jejich definice

Aby bylo možno efektivně pracovat se stavovým prostorem regálového systému, který se vyznačuje poměrně velkou nekompaktností, je třeba tento rozdělit na jednotlivé menší kompaktní celky, na kterých lze jednoduše uplatnit mechanismy prohledávání a

kalkulací uzlové metriky. Způsob rozdělení je naznačen na obr. 41. I přes toto rozdělení je nutné čelit nepravidelnostem, které do systému adresování a tvaru řad regálů zanesli různí tvůrci v rámci života jeho vývoje. Například již jednou zmíněný stejný směr číslování řady 1 a 2. Různé délky řady 1, 18, 19, 22. Nesymetrické příčné položení řady 7, 8, 10, 11. Těmito nepravidelnostmi se definice kalkulační nad systémem značně rozvětvila a zkomplikovala.

Příslušnost uzlů do jednotlivých podgrafů nad grafem regálového systému je možno zobrazit v aplikaci menu grafy => podmenu název grafu. Do datové mřížky jsou poté zobrazeny pouze uzly příslušného podgrafu, které je nutné navštívit. Tato funkcionality je zprovozněna čistě ze studijně demonstrativních důvodů. Pro praktické nasazení nemá větší význam.

Hraniční uzly mezi jednotlivými podgrafy tvoří zvlášť samostatný podgraf, na kterém je kalkulován průchod systémem. Ostatní podgrafy jsou definovány ulicemi, která je přirozeně tvořena dvojicí regálové řady. Hraniční uzly podgrafů ulice jsou zároveň součástí podgrafu hranice.



Obr. 41. Rozdělení regálového systému na jednotlivé podgrafy

5.4 Funkce ohodnocení hrany

Základním stavebním prvkem pro kalkulace vzdáleností mezi jednotlivými uzly jsou dvě hodnotící funkce. První funkce kalkuluje vzdálenosti mezi uzly na úrovni grafu 0, a to za pomoci definice mapy, která obsahuje souřadnice 97 hraničních pozic. Druhá hodnotící funkce kalkuluje vzdálenosti mezi prvky grafu 1 až 7, a to na základě znalostí základních měř viz tabulka na obr. 22. Informace o lokaci pozice je získána z adresy pozice. Komplikace s nepravidelným číslováním odstranila konverzní funkce a funkce ekvivalence. Konverzní funkce vrací hodnotu pozice dle koncového znaku

řetězce adresy a funkce ekvivalence vrací pozici protější pozice v jedné ulici. Pro představu, jak taková funkce vypadá a s čím si musí poradit, následuje názorná okomentovaná ukázka.

Definice funkce ohodnocení hrany grafu 1-7

```
function TForm1.HranaU(avrcholA:integer; avrcholB:integer):single;
var
  A, B:integer; // indexy pozic
  x, y, z:single; // souřadnice pohybu
  sA, sB:string; // adresy pozic
begin
  A:=avrcholA;      //index pozice posledního vrcholu cesty v uPozice
  B:=avrcholB;      //index pozice vrcholu mimo cesty v uPozice
  sA:=pseznam.ppr[A];
  sB:=pseznam.ppr[B];
  //posun v ose z
  if StrToInt(LeftStr(sA,2))<> StrToInt(LeftStr(sB,2))then z:= 3.2 else z:=0;

  //posun v ose x ve stejné řadě
  if StrToInt(LeftStr(sA,2))= StrToInt(LeftStr(sB,2)) then
    begin
      //když se jedná o stejný sloupec
      if (StrToInt(MidStr(sA,4,2)) - StrToInt(MidStr(sB,4,2)))=0 then
        x:=(Abs(KonverzePozice(RightStr(sA, 1))-KonverzePozice(RightStr(sB, 1))))*0.93
      else

        // když se pohybují v kladném směru 1-2-3
        if (StrToInt(MidStr(sA,4,2)) - StrToInt(MidStr(sB,4,2)))<0 then
          x:=((((Abs(StrToFloat(MidStr(sA,4,2)) - StrToFloat(MidStr(sB,4,2))))*2.79)+1.86)-
            //delka přes všechny sloupce a od ní se odečte začátek a konec
            ((Abs((1-KonverzePozice(RightStr(sA, 1))))*0.93)+((3-
              KonverzePozice(RightStr(sB, 1))))*0.93)))
          else

            // když se pohybují v záporném směru 3-2-1
            x:=((((Abs(StrToFloat(MidStr(sA,4,2)) - StrToFloat(MidStr(sB,4,2))))*2.79)+1.86)-
              //delka přes všechny sloupce a od ní se odečte začátek a konec
              (((3-KonverzePozice(RightStr(sA, 1))))*0.93)+(Abs((1-
                KonverzePozice(RightStr(sB, 1))))*0.93)));
        end
      else
        // posun v ose x v protější řadě
        begin
          if (StrToInt(MidStr(sA,4,2)) - StrToInt(LeftStr(Ekvivalent(B),2)))=0 then
            //když se jedná s ekvivalencí o stejný sloupec
            x:=(Abs(KonverzePozice(RightStr(sA, 1))-StrToFloat(RightStr(Ekvivalent(B),
              1))))*0.93
          else
            // když se pohybují v kladném směru 1-2-3
            if (StrToInt(MidStr(sA,4,2)) - StrToInt(LeftStr(Ekvivalent(B), 2)))<0 then
```

```

x:=(((Abs(StrToFloat(MidStr(sA,4,2)) - StrToFloat(LeftStr(Ekvivalent(B),2)))
*2.79)+1.86)- ((Abs((1-KonverzePozice(RightStr(sA, 1))))*0.93)+((3-StrToFloat
(RightStr(Ekvivalent(B), 1))))*0.93)))
// když se pohybují v záporném směru 3-2-1
x:=(((Abs(StrToFloat(MidStr(sA,4,2)) - StrToFloat(LeftStr(Ekvivalent(B),2)))
*2.79)+1.86)- (((3-KonverzePozice(RightStr(sA, 1))))*0.93)+(Abs((1-StrToFloat
(RightStr(Ekvivalent(B), 1))))*0.93)));
end;

//posun v ose y
y:= Abs(StrToFloat(MidStr(sA,7,2)) - StrToFloat(MidStr(sB,7,2)))*1.75; //posun v y

Result:= z + x + y;
end;

```

Stručné shrnutí práce výše uvedené funkce na výpočet délky hrany v grafu 1-7 je následující:

- 1) vypočti délku posunu v ose z (nastane pouze když jde o přechod přes ulici);
- 2) vypočti délku posunu v ose x, a to dle druhu pohybu:
 - a) jedná se o prvky ve stejné řadě i stejném sloupci;
 - b) jedná se o prvky ve stejné řadě, různém sloupci a kladném směru posunu;
 - c) jedná se o prvky ve stejné řadě, různém sloupci a záporném směru posunu;
 - d) jedná se o prvky v různé řadě, ale ekvivalencí náleží do stejného sloupce;
 - e) jedná se o prvky v různé řadě, ekvivalencí náleží do různého sloupce a posun je v kladném směru;
 - f) jedná se o prvky v různé řadě, ekvivalencí náleží do různého sloupce a posun je v záporném směru;
- 3) vypočti délku posunu v ose y;
- 4) sečti posuny v ose x, y, z.

Výsledné ohodnocení hrany je uloženo do objektu uHrany pro ohodnocení prioritou a další zpracování algoritmem optimálního průchodu grafem.

5.5 Strategie

Pro optimální průchod regálovým systémem byla navržena strategie, která zajišťuje specifické vlastnosti prohledávacího algoritmu, které vyplývají z podmínek pro konstrukci otevřeného hamiltonovského cyklu a praktických požadavků na obsluhu systému. Strategie jako taková se dělí na dvě dílčí části. První je podstrategie průchodu systémem přes graf 0 ze start bodu do koncového bodu. Strukturovaně by se dala popsat takto:

- výchozím bodem je vždy pozice 08-07-01a;
- koncovým bodem je vždy poslední zařazený uzel grafu 0 z objektu uPozice;
- pakliže nejsou v expediční dávce hraniční body grafů 1-7, které je třeba navštívit (graf obsahuje alespoň jeden nehraniční uzel, který je nutno obsloužit), přidej je, protože cesta přes ně přirozeně musí vést;
- obsluha regálových řad je ohodnocena prioritami;
- nejvyšší prioritu má příčný regál 8 a 7 (startovací strana);
- následuje priorita průchodu regálovou řadou, která je vyhodnocena z podmínky obsazení řady alespoň jedním uzlem na vzdálenější polovině oproti aktuálnímu hraničnímu uzlu, ze kterého je vyhodnocován postup;
- o stupeň níže je strukturovaná priorita obsluhy nejvzdálenějšího místa od startu až po řadu 10;
- poté přichází priorita řady 10;
- postupuje strukturovaná priorita nejvzdálenějšího bodu od startu;
- poslední je priorita nejkratší hrany;

Druhá podstrategie řídí průchod grafem 1-7. Průchod je započat vstupem do prvního uzlu grafu, který je zároveň hraniční, a ukončen výstupem z grafu uzlem, který je opět zároveň hraničním. Strukturovaně by se dala popsat takto:

- rozhodnutí o výhodnosti průchodu křižováním (obsluha obou stran ulice při postupu v jednom směru) nebo postupnou obsluhou jedné strany ve směru tam a druhé strany ulice ve směru zpět. Rozhodnutí je provedeno kontinuálně s předchozí podstrategií, neboť určuje polohu vstupu a výstupu z grafu ulice;
- při postupu křižováním priorita návratu pro prvek, který zaostal o více než jednu délku mezi dvěma sousedními uzly v ose x ;
- při postupu jednou stranou priorita návratu pro prvek, který zaostal ve směru postupu o více než dvojnásobek délky hrany mezi dvěma sousedními uzly v ose x ;
- priorita dokončení řady před přechodem na jinou;
- priorita zařazení všech požadovaných uzlů z grafu do cesty před jeho opuštěním;
- priorita nejlevnější hrany.

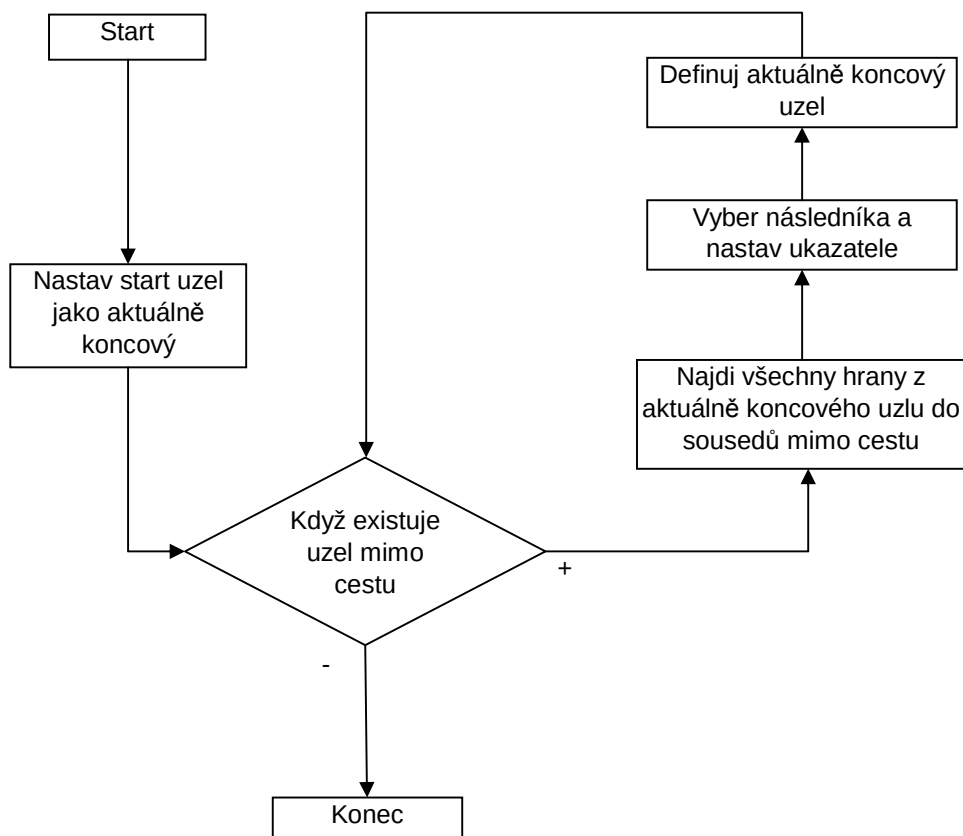
Hodnoty vzdáleností, kdy je již algoritmus nucen k návratu pro uzel vynechaný z cesty (priorita návratu pro prvek) nejpřímochařejším postupem k cílovému bodu, byly stanoveny odhadem a následně potvrzeny experimentálně. Jedná se o parametry, které lze v případě změny vlastností řešené úlohy modifikovat.

Svázáním těchto dvou podstrategií vznikne výsledná strategie, která je aplikována pomocí systému pravidel na prohledávacím algoritmu a která mu dodává potřebnou míru „intelligence“ (určuje směr prohledávání, postupu) důležitou k ochraně proti uvíznutím v lokálních minimech.

5.6 Prohledávací algoritmus

Prohledávací algoritmus je modifikací Jarníkova algoritmu hledání nejlevnější kostry na grafu. Zásadním odklonem od jeho principu je pevná definice start bodu, což zajistí ochranu proti rozvětvení a nasazení pravidel dle uvedené strategie k nastavení směru. Výsledná kostra je tedy hamiltonovskou cestou i když není zaručeno, že je minimální. Strukturovaný popis principu funkce algoritmu je následující:

- 1) aktuální uzel: = start uzel;
- 2) **while** not všechny uzly grafu v kostře **do**
- 3) najdi všechny hrany z aktuální do všech možných následníků;
- 4) dle strategie vyber následníka;
- 5) aktuálnímu nastav ukazatel na následníka;
- 6) následníku nastav ukazatel na aktuálního;
- 7) aktuální:= následník.
- 8) **end while**



Obr. 42. Blokové schéma prohledávacího algoritmu

5.7 Ohodnocení objednávek

Postup ohodnocení objednávek je procedura, která přiřazuje každé objednávce, s vlivem na optimální cestu, hodnotu vyjadřující v jednotkách minut čas, který by bylo možno uspořit vyřazením příslušné objednávky z expediční dávky.

Přesná metoda kalkulace ohodnocení objednávek je velmi výpočetně náročná a skládá se z výpočtu času na celou expediční dávku a následně znovu přepočet pro celou expediční dávku bez právě jedné dílčí objednávky, která je z dávky vyňata. Rozdíl obou vykalkulovaných časů je hodnota úspory vynecháním dílčí objednávky z expediční dávky. Takto postupně opakováno přes všechny objednávky až do ohodnocení všech objednávek.

Je nutné si uvědomit, že tento teoretický postup v každé iteraci vymezuje novou dávku vždy o jednu objednávku menší, což je samo o sobě zatěžující a navíc znovu a znovu kalkuluje optimální trasu. Časová náročnost takového postupu se může vyšplhat až do řádu desítek minut, což je pro praktické využití nepřijatelné.

Jak takovýto proces optimalizovat? Prvně je nutné si uvědomit, že ne každá objednávka má přímý vliv na výslednou optimální trasu, přesněji řečeno ne každá objednávka vynětím z expediční dávky změní počet navštívených uzlů. Toto je přirozeně dáno opakováním se pozic na jednotlivých objednávkách. Z toho plyne, že objednávky, které nemají vliv na počet navštívených pozic a tím na celkovou délku trasy, což je jediný proměnný parametr kalkulace náročnosti expediční dávky, není třeba vůbec zahrnovat do testu nové trasy. Vzhledem k tomu, že experimentálně bylo zjištěno, že se jedná cca o 20–40 % objednávek bez vlivu na změnu délky trasy, je to úleva výrazná, nicméně stále ne dostačující. Pro příklad uveďme, že maximální reprezentativní dávka se skládá ze 128 objednávek rozpadlých na 1 221 položkových záznamů s 535 jedinečnými pozicemi. Vyloučením objednávek bez efektu změny počtu navštívených pozic dojdeme k číslu 79, které mají přímý vliv. Oproštěno od testování je tedy 38 % objednávek. Přesto kalkuluje-li průměrný počítač optimální trasu na maximální dávce kolem 7 s, je čas spotřebovaný jen na rekalkulaci dávek 9,2 min. (nehledě na nastavování dávek). Závěrem tohoto krátkého rozboru je, že odstraněním objednávek bez vlivu na délku trasy dosáhneme úspory procesorového času, ovšem stále nedostačující pro praktické nasazení. Možnost záchrany situace předimenzováním hardwarových prostředků nevylučuji, ovšem zároveň neberu v potaz jako výlučné a nutné řešení. Jde pouze o sekundární podporu úspory výpočetního času.

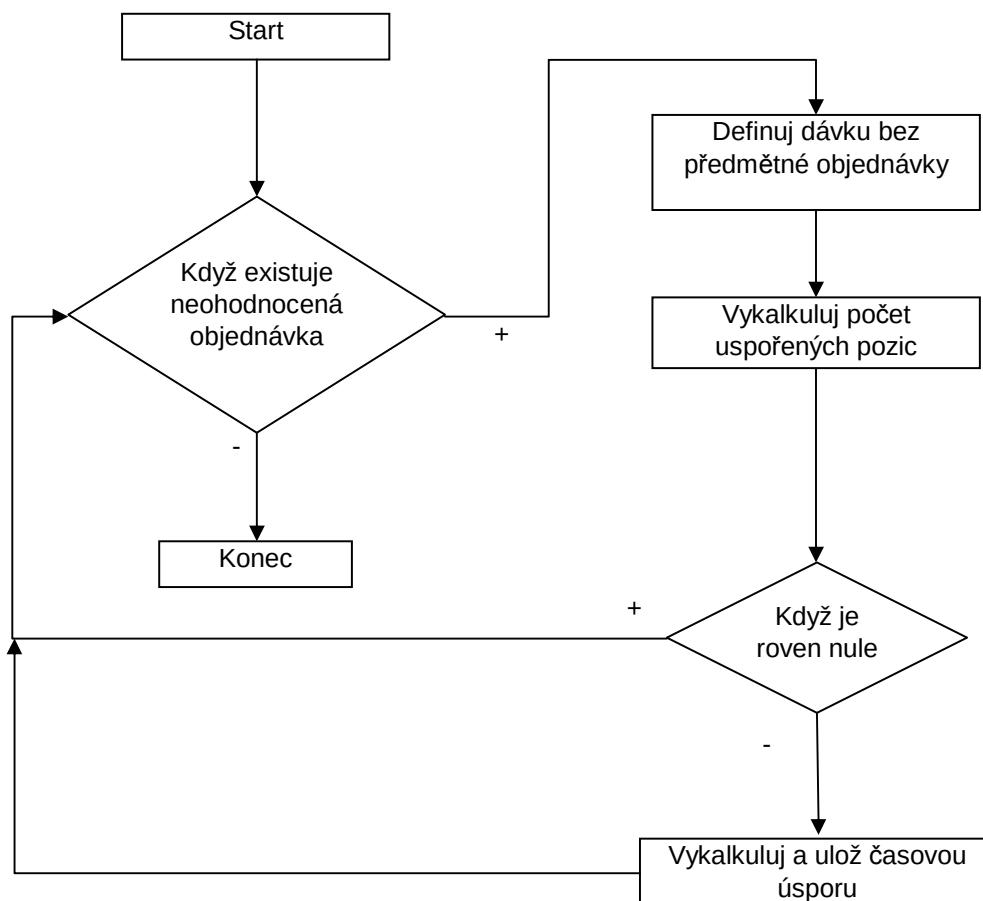
Aproximativní metoda určení časového ohodnocení každé jednotlivé dílčí objednávky z expediční dávky se jeví jako nezbytné řešení pro praktické nasazení. Je zřejmé, že budeme-li kalkuloval časovou úsporu objednávek přes reálnou úsporu počtu navštívených pozic vynětím dílčí objednávky z expediční dávky a k časovému údaji přijde podílem času celkové dávky na jednu pozici, tedy aritmetickým průměrem na jednu pozici celkové dávky, zcela se vyhneme rekalkulacím optimálních tras a tedy i enormní časové ztrátě na procesorovém čase. Metoda velmi úsporná, neboť zcela odstraňuje výpočet optimální trasy v každé iteraci, která představuje kalkulaci dílčí objednávky, nicméně má svoje teoretické úskalí. Riziko v nepřesnosti, které tímto aproximativním způsobem zjednodušené kalkulace podstupujeme, nastává teoreticky nejvíce v případě, kdy objednávka o jedné uspořené pozici ovlivňuje celkovou délku trasy mnohonásobně více než objednávky s velkou úsporou pozic. Je to obecná otázka

rozložení průměru přes objednávky, kdy objednávky s maximální a minimální úsporou budou nejvíce nepřesné.

Vzhledem k tomu, že ukládací specifikace do regálového systému konsolidují zboží po PNC kódech, šaržích, typech skladu, majiteli zboží a obsluha navíc dle typu balení, je pravděpodobnost vzniku popsaného rizikového stavu nepřesnosti zanedbatelné. Blokové schéma naznačující postup popsaného algoritmu je na obr. 43. Naproti tomu graf na obr. 44 představuje časovou úsporu zvoleného řešení.

	Exaktní metoda	Aproximativní metoda	Úspora
Největší dávka	862,69	37,96	96%
Střední dávka	179,74	7,58	96%
Nejmenší dávka	1,82	0,22	88%

Obr. 44. Porovnání časové úspory aproximativního řešení oproti exaktnímu



Obr. 43. Blokové schéma algoritmu ohodnocení objednávek

5.8 Vstupní a výstupní rozhraní

Již z podstaty řešené aplikace, jejíž model je naznačen na obr. 45 metodou tzv. černé skříňky, je na vstupu sada dat reprezentujících expediční dávku a pocházejících

jako výstup z informačního systému skladového hospodářství a na straně výstupu rovnocenná sada dat přetvořená v optimální permutaci pozic odbavení expediční dávky.



Obr. 45. Model datového toku vně aplikace

Ideálním řešením pro praktické použití by byla implementace systému OPCES do informačního systému skladového hospodářství jako samostatný modul, který by mohl pracovat na pozadí. Jelikož je OPCES vyvíjen jako samostatná aplikace, bylo nutno přistoupit k řešení vstupu a výstupu datových sad.

Vstup je vyřešen importní funkcí ze souboru aplikace Excel, kde musí být datová sada uložena v poli začínajícím buňkou A1. Import se spouští z menu aplikace Soubor podmenu Import z xls. Data jsou načtena do datové mřížky aplikace a validována kontrolní procedurou, která kontroluje, jestli odpovídají očekávanému složení, které musí odpovídat formátu objektu uData popsanému v kapitole 5.2 a zobrazenému na obr. 46. Vyhodnocení je oznámeno uživateli zprávou. Teprve po potvrzení jsou záznamy uloženy do objektu uData.

objednávka	pozice	PNC	popis	kusů
------------	--------	-----	-------	------

Obr. 46. Struktura importovaných dat

Výstup z aplikace je řešen exportem dat z datové mřížky aplikace do schránky MS Office. Tento export je řešen procedurou načítající strukturu odpovídající struktuře užívanou programem Excel. Tímto způsobem může uživatel vyexportovat veškerá data, která aplikace zobrazuje v datové mřížce včetně záhlaví.

Ve všech ostatních směrech pracuje aplikace obvyklým způsobem řízením vstupů přes klávesnici nebo myš.

5.9 Uživatelské rozhraní a obsluha

Systém OPCES je navržen tak, aby komunikoval s uživatelem pokud možno jednoznačně a jednoduše. Obsluha a informační obsah jsou pro intuitivní ovládání co nejprůhlednější a nejjednodušší. V případě, že si obsluha není jista, co který parametr na formuláři znamená, stačí na něj najet kurzorem a je zobrazena kontextová nápověda. Základním kamenem pro obsluhu je ovládací panel nabídky hlavního menu. Informace v podobě datových sad jsou obsluze zobrazovány v datové mřížce aplikace, která pružně reaguje na obsah svojí velikostí. Vypočtená hodnota délky optimální trasy, časové náročnosti expediční dávky a čas výpočtu jsou zobrazeny v záhlaví formuláře. Záhlaví zároveň obsahuje editační pole pro vstup hodnot k výpočtu celkové náročnosti dávky. Posledním nejmenovaným editačním prvkem záhlaví je pole pro vložení cesty

k souboru s importovanými daty. Datová mřížka je odblokována k editaci záznamů formou kopírování vkládání a mazání textových řetězců jednotlivých buněk, nicméně tyto změny nemají žádný vliv na zpracovávaná data uvnitř aplikace. Účel této možnosti spočívá v jejím využití před exportem záznamů z datové mřížky nebo pro export pouze jedné konkrétní hodnoty. Celkový pohled na uživatelské rozhraní aplikace OPCES nabízí obr. 47.

Panel nabídky hlavního menu obsahuje:

1. Soubor

- a) Import z xls >>>>>> spouští import dat;
- b) Kopíruj formát xls >>>>>> vloží datovou mřížku do schránky;
- c) Reset >>>>>> připraví aplikaci na novou dávku;
- d) Konec >>>>>> odchod z aplikace;

2. Úpravy

- e) Copy >>>>>> kopíruje jako text do schránky;
- f) Cut >>>>>> vyjme jako text do schránky;
- g) Paste >>>>>> vloží text ze schránky;

3. Grafy

- h) Mimo ulice >>>>>> zobrazí graf 0 do datové mřížky;
- i) Ulice 1 >>>>>> zobrazí graf 1 do datové mřížky;
- j) Ulice 2 >>>>>> zobrazí graf 2 do datové mřížky;
- k) Ulice 3 >>>>>> zobrazí graf 3 do datové mřížky;
- l) Ulice 4 >>>>>> zobrazí graf 4 do datové mřížky;
- m) Ulice 5 >>>>>> zobrazí graf 5 do datové mřížky;
- n) Ulice 6 >>>>>> zobrazí graf 6 do datové mřížky;
- o) Ulice 7 >>>>>> zobrazí graf 7 do datové mřížky;

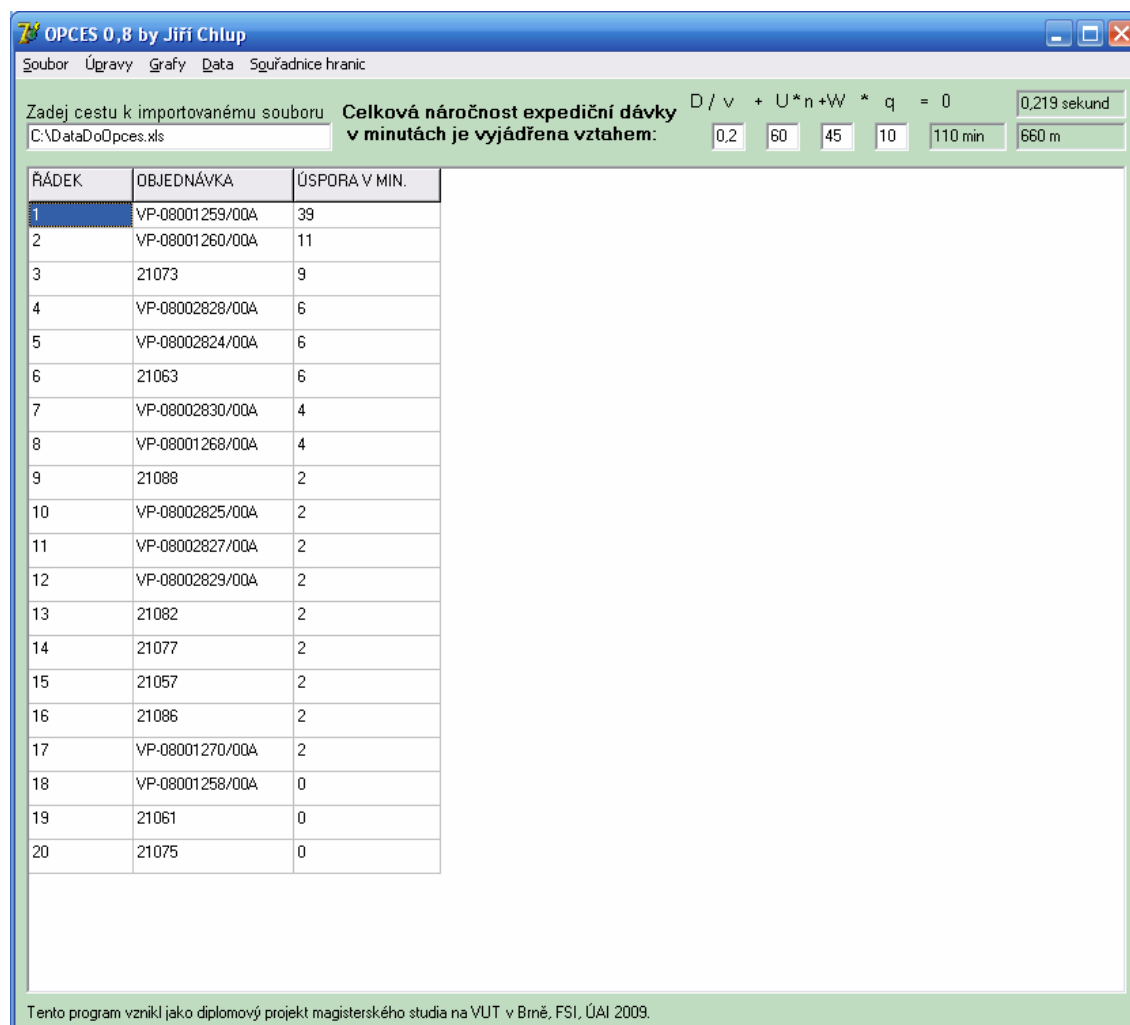
4. Data

- p) Objednávky >>> zobrazí data objednávek do datové mřížky podrobně;
- q) Pozice >>> zobrazí pozice k obsluze do datové mřížky;
- r) Optimální cesta >>> zobrazí permutaci optimální cesty do datové mřížky;
- s) Hodnota objednávek > zobrazí ohodnocení objednávek do datové mřížky;
- t) Data pro obsluhu >>> zobrazí data pro obsluhu do datové mřížky;

5. Souřadnice hranic >>> zobrazí data souřadnic hraničních bodů do datové mřížky;

Datová mřížka zobrazuje informační obsah v průběhu činnosti aplikace automaticky nebo vyvoláním z hlavního menu, a to v následující podobě.

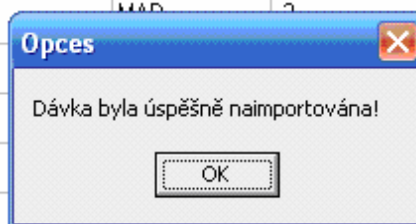
- 1) Ihned po importu, ještě před potvrzením jeho úspěšnosti, zobrazí datová mřížka surovou podobu importovaného datového pole, tak jak bylo nalezeno ve zdrojovém souboru xls. V případě nekorektnosti datové struktury je tedy možno najít případnou chybu v sadě zdrojových záznamů, jako například importování záhlaví, chybně seřazené entity atp. Náhled znázorňuje obr. 48.



Obr. 47. Vzhled uživatelského rozhraní aplikace

- 2) Následně po potvrzení úspěšného importu je zobrazen obsah datového objektu uData. Toto zobrazení se rovná tlačítku Objednávky v hlavním menu. Entity pořadí jsou nastaveny na nekonečno a test na false. Pakliže si přes menu po kalkulaci zobrazíme optimální trasy, bude pořadí nastaveno dle optimální permutace. Náhled je k dispozici na obr. 49.
- 3) Optimální trasa je zobrazena v datové mřížce vyvoláním výpočtu z hlavního menu tlačítko Optimální cesta. Zobrazena je permutace pozic v optimálním pořadí včetně start bodu a hraničních pozic, kterými trasa vede a jsou nutné k výpočtu optimálního průchodu systémem. Náhled je k dispozici na obr. 50.
- 4) Pozice, které musí být navštíveny, definují řešený graf expediční dávky, jsou zobrazeny do datové mřížky vyvoláním z hlavního menu tlačítkem Pozice. Ukazatele prvku před a po jsou nastaveny na nekonečno. Náhled je k dispozici na obr. 51.
- 5) Jednotlivé dílčí grafy, jejich uzly, jsou zobrazeny vyvoláním z hlavního menu. Ukázka je znázorněna na obr. 52.

21073	05-03-01a	9723	KKH	-900
21077	05-04-01a	776	KKH	-10
21077	05-04-01a	773	KKH	-10
21073	05-08-02b	12714	KKH	-120
VP-08001259/0	07-03-02c	34994	MAD	-3
VP-08001259/0	07-03-03b	42983	MAD	-2
VP-08001270/0	07-05-03b	37848	MAD	-24
VP-08001260/0	08-01-04a	38759	MAD	-2
VP-08001259/0	08-01-04b	41427	MAD	-1
VP-08001259/0	08-05-01b	42943	MAD	-2
21063	12-05-02b	12773		
21063	12-14-03b	12887		
21086	12-19-03a	12773		
21061	12-20-01a	2600521		
21075	12-20-01a	2600511	KKH	-2



Obr. 48. Surová importovaná data zobrazená v datové mřížce

ŘÁDEK	OBJEDNÁVKA	ADRESA	PNC	POPIS	KUSŮ	POŘADÍ	TEST
1	21073	05-03-01a	9723	KKH	-900	99999	0
2	21077	05-04-01a	776	KKH	-10	99999	0
3	21077	05-04-01a	773	KKH	-10	99999	0
4	21073	05-08-02b	12714	KKH	-120	99999	0
5	VP-08001259/00A	07-03-02c	34994	MAD	-3	99999	0
6	VP-08001259/00A	07-03-03b	42983	MAD	-2	99999	0
7	VP-08001270/00A	07-05-03b	37848	MAD	-24	99999	0
8	VP-08001260/00A	08-01-04a	38759	MAD	-2	99999	0
9	VP-08001259/00A	08-01-04b	41427	MAD	-1	99999	0
10	VP-08001259/00A	08-05-01b	42943	MAD	-2	99999	0
11	21063	12-05-02b	12773	KKH	-200	99999	0
12	21063	12-14-03b	12887	KKH	-100	99999	0
13	21086	12-19-03a	12773	KKH	-800	99999	0
14	21061	12-20-01a	2600521	KKH	-1	99999	0
15	21075	12-20-01a	2600511	KKH	-2	99999	0

Obr. 49. Zobrazení objektu uData následně po importu do datové mřížky

POŘADÍ	POZICE	HRANIČNÍ	ULICE	PŘED	PO
1	08-07-01c	1	0	68	8
2	08-05-01b	1	0	68	7
3	08-01-04b	1	0	8	6
4	08-01-04a	1	0	7	5
5	07-05-03b	1	0	6	4
6	07-03-03b	1	0	5	3
7	07-03-02c	1	0	4	53
8	12-01-02a	1	3	3	9
9	12-05-02b	0	3	53	10
10	12-14-03b	0	3	9	11
11	12-19-03a	0	3	10	12
12	12-20-01a	0	3	11	13
13	12-20-01b	0	3	12	2
14	05-08-02b	0	3	13	1
15	05-04-01a	0	3	2	0

Obr. 50. Zobrazení optimální cesty v datové mřížce

ŘÁDEK	POZICE	HRANIČNÍ	ULICE	PŘED	PO
1	05-03-01a	0	3	99999	99999
2	05-04-01a	0	3	99999	99999
3	05-08-02b	0	3	99999	99999
4	07-03-02c	1	0	99999	99999
5	07-03-03b	1	0	99999	99999
6	07-05-03b	1	0	99999	99999
7	08-01-04a	1	0	99999	99999
8	08-01-04b	1	0	99999	99999
9	08-05-01b	1	0	99999	99999
10	12-05-02b	0	3	99999	99999
11	12-14-03b	0	3	99999	99999
12	12-19-03a	0	3	99999	99999
13	12-20-01a	0	3	99999	99999
14	12-20-01b	0	3	99999	99999
15	12-26-01c	0	3	99999	99999

Obr. 51. Obsah objektu uPozice v datové mřížce

POZICE	HRANIČNÍ	ULICE
14-29-01a	0	4
14-26-01b	0	4
14-27-04a	0	4
14-24-04b	0	4
14-24-05b	0	4
13-07-01a	0	4
13-10-01c	0	4
14-19-01a	0	4
14-20-04a	0	4
13-11-04b	0	4
14-18-05b	0	4
14-17-05c	0	4
14-17-04c	0	4
14-17-04b	0	4
13-15-01c	0	4

Obr. 52. Zobrazení grafu 4 do datové mřížky

POZICE	SOUŘADNICE Z	SOUŘADNICE X	POSUN V Z
01-01-1	51,2	92,07	0
01-01-2	51,2	91,14	0
01-01-3	51,2	90,21	0
01-02-1	51,2	89,28	0
01-02-2	51,2	88,35	0
01-02-3	51,2	87,42	0
01-03-1	51,2	86,49	0
01-03-2	51,2	85,56	0
01-03-3	51,2	84,63	0
01-04-1	51,2	83,7	0
01-32-3	51,2	3,72	0
01-33-1	51,2	2,79	0
01-33-2	51,2	1,86	0
01-33-3	51,2	0,93	0
01-34-1	51,2	0	0

Obr. 53. Zobrazení souřadnic hraničních bodů do datové mřížky

ŘÁDEK	OBJEDNÁVKA	ÚSPORA V MIN.
1	VP-08001259/00A	39
2	VP-08001260/00A	11
3	21073	9
4	VP-08002828/00A	6
5	VP-08002824/00A	6
6	21063	6
7	VP-08002830/00A	4
8	VP-08001268/00A	4
9	21088	2
10	VP-08002825/00A	2
11	VP-08002827/00A	2
12	VP-08002829/00A	2
13	21082	2
14	21077	2
15	21057	2

Obr. 54. Zobrazení ohodnocení objednávek v datové mřížce

KÓD PNC	NÁZEV	UMÍSTĚNÍ	POČET	ODEBRÁNO
42943	MAD	08-05-01b	-2	
41427	MAD	08-01-04b	-1	
38759	MAD	08-01-04a	-2	
37848	MAD	07-05-03b	-24	
42983	MAD	07-03-03b	-2	
34994	MAD	07-03-02c	-3	
12773	KKH	12-05-02b	-200	
12887	KKH	12-14-03b	-100	
12773	KKH	12-19-03a	-800	
2600511	KKH	12-20-01a	-2	
2600521	KKH	12-20-01a	-1	
2744294	KKH	12-20-01b	-1	
12714	KKH	05-08-02b	-120	
773	KKH	05-04-01a	-10	
776	KKH	05-04-01a	-10	

Obr. 55. Zobrazení datové sady určené k exportu z aplikace

- 6) Souřadnice hraničních bodů jsou zobrazeny do datové mřížky vyvoláním z hlavního menu tlačítkem Souřadnice pozic. Ukázka je na obr. 53.
- 7) Časová úspora dílčích objednávek je zobrazena do datové mřížky vyvoláním výpočtu z hlavního menu tlačítkem Hodnota objednávek. Entita Úspora v min. je seřazena sestupně, tedy na prvním řádku je objednávka s největší úsporou času vyřazením z expediční dávky. Náhled je na obr. 54.
- 8) Výstupní data pro obsluhu manipulační techniky jsou zobrazena do datové mřížky vyvoláním z hlavního menu tlačítkem Data pro obsluhu. Jedná se o optimální permutaci pozic rozpadlých na jednotlivé PNC kódy, takže pozice se přirozeně opakují v případě výběru více položek z jedné pozice, což je patrné z ukázky znázorněné na obr. 55.

Aplikace OPCES od verze 0,9 je optimalizována proti opakování náročných výpočtů na stejné dávce. Znamená to tedy, že například opětovným vyvoláním tlačítka ohodnocení objednávek nedojde k opětovnému přepočtení, nýbrž k zobrazení prvně vykalkulovaných hodnot. Optimální cestu lze nad jednou importovanou expediční dávkou provést pouze jednou. Důvodem je neukládání výchozích vstupních dat expediční dávky (není důvod). Veškeré potřebné výstupní hodnoty zůstávají uloženy až do užití tlačítka Reset nebo ukončení aplikace. Konkrétně ohodnocení objednávek, celková délka optimální cesty, časová náročnost expediční dávky a výstupní data pro obsluhu.

5.10 Podmínky provozu

Aplikace OPCES je samostatným, přeloženým a zkompilevaným programem, který je možno provozovat na počítačích vybavených operačním systémem MS Windows. Pakliže má aplikace vykazovat funkčnost rovnocennou vyvinuté testovací verzi 0,9, je nutné ji provozovat za těchto minimálních podmínek, které zaručují funkčnost a výsledky v uspokojivých testovaných časech.

- operační systém MS Windows XP;
- počítač s procesorem o taktovací frekvenci 1,5 GHz;
- L1 cache 32 KB a L2 1 MB;
- FSB 64 bitů s pásmem 800 MB/s a taktem 100 MHz;
- operační paměť alespoň 512 MB;
- taktovací frekvence operační paměti 266 MHz.

Nezbytnou součástí vybavení na provozovaném počítači pro verzi 0,9 je nainstalovaný program MS Excel, pomocí kterého jsou aplikaci předávána zdrojová data. Provoz na počítačích s vyššími parametry, než jsou výše uvedené, povede přirozeně k vyšší výpočetnímu výkonu.

6. Testy

Neoddiskutovatelnou obhajobou všech až doposud uvedených rozborů, návrhů řešení, myšlenkových konceptů a fyzické implementace jsou výsledky konkrétních testů na reálných datech. Touto zásadní problematikou se zabývám v předposlední kapitole této diplomové práce, kterou lze právem považovat za jednu z nejdůležitějších.

Testy jsou provedeny na reprezentativním vzorku dat, který je definovaný v kapitole 3, a jsou rozděleny do čtyř základních okruhů. Získané hodnoty, jejich rozbor a vyhodnocení následuje posléze.

- 1) Porovnání vypočtené délky optimální trasy oproti délce trasy získané stávajícím řešením včetně z toho plynoucího ohodnocení časové náročnosti expediční dávky.
- 2) Test přesnosti použité metody aproximativního ohodnocení objednávek.
- 3) Test implementovaných algoritmů oproti ideálnímu řešení.
- 4) Test časové náročnosti výpočtů v aplikaci.

První okruh řeší jednoduše porovnatelné hodnoty, kdy jejich rozdíl jednoznačně prokazuje míru úspěchu implementovaného řešení. Měření je provedeno na třech reprezentativních vzorcích dat, a to na celkovém průchodu regálovým systémem. Navíc u nejnáročnější dávky také na průchodu nejvytíženější ulicí a nejvytíženější řadou.

Druhý okruh testuje přesnost zvolené aproximativní metody ohodnocení objednávek časovou náročností. Princip testu spočívá ve srovnání s exaktní metodou ohodnocení objednávek popsanou v kapitole 5.7.

Třetí okruh si bere za cíl porovnat schopnost algoritmu nalézt optimální trasu na ohraničené části řešeného prostoru regálového systému, kde je známo teoretické ideální řešení. Konkrétně na případě obsazení jedné kompletní regálové řady, kdy díky vlastnosti pravoúhlé síťové struktury regálových pozic je známa ideální hamiltonovská cesta, tedy optimální řešení průchodu (šachovnice).

Čtvrtý okruh testuje časovou náročnost výpočtu dvou základních úloh aplikace. Výpočet optimální trasy a výpočet ohodnocení objednávek. Obě výpočetně náročné úlohy jsou testovány na největší expediční dávce a pro srovnání na třech různých počítačích.

Každým ze čtyř testovaných okruhů se separátně zabývá jedna z následujících podkapitol, které zobrazují a interpretují naměřená data a zároveň diskutují zjištěné hodnoty.

6.1 Optimální trasa a náročnost expediční dávky

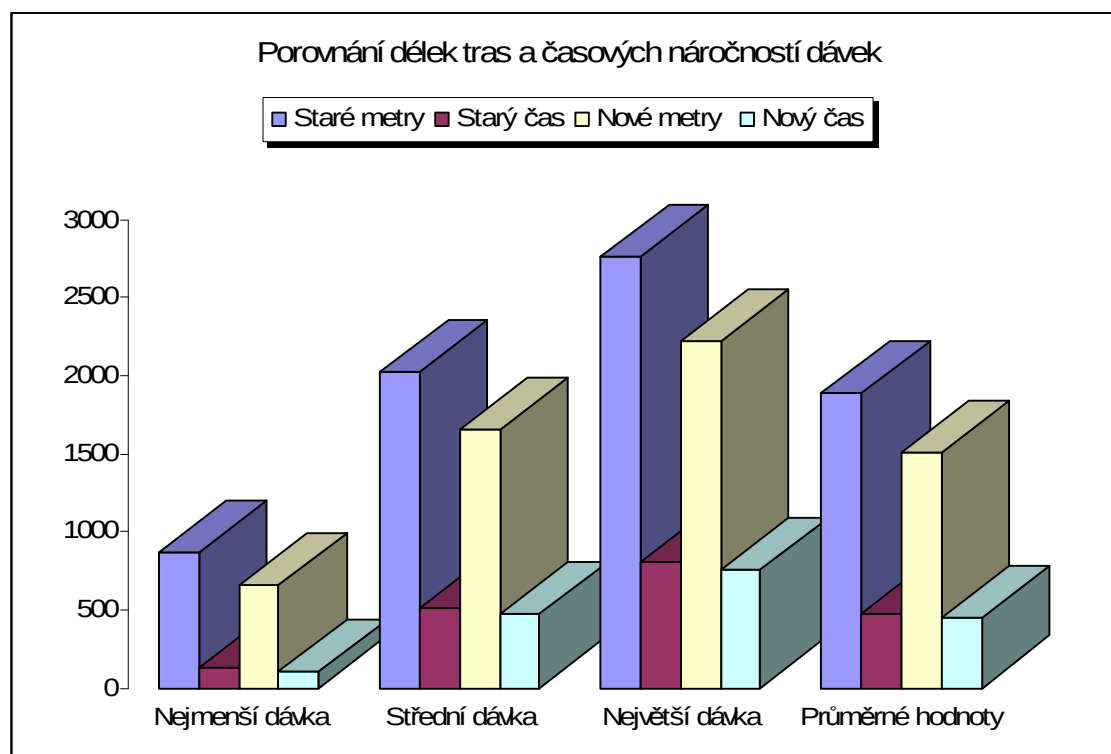
Data získaná měřeními optimálních tras a náročností expedičních dávek dle stávajícího systému a pomocí aplikace OPCES 0,9 jsou zobrazena v tabulce na obr. 56 a v grafu na obr. 57.

Úspora vzdálenosti celkové trasy průchodu regálovým systémem činí v průměru přes všechny tři testované dávky 374 metrů, což je téměř pětkrát délka regálové řady

v bloku. Poměrově to znamená něco přes 20 %, což považuji za výsledek velmi dobrý, který naplnil cíl této diplomové práce. Takovéto zkrácení celkové trasy průchodu regálovým systémem přináší časovou úsporu v průměru 31 minut, což by se mohlo zdát nepříliš velká hodnota, nicméně důležité je si uvědomit, že délka trasy nehraje v časové úspoře nejzásadnější roli, neboť zde je hlavním hráčem rychlost pojezdu z uzlu do uzlu, rychlost odbavení uzlu a rychlost výměny palety. Tyto parametry nejzásadněji ovlivňují časovou náročnost expediční dávky. Přesto průměrných 8,5 % časové úspory získané jen díky optimalizaci celkové délky trasy považuji za velký přínos.

		Nejmenší dávka	Střední dávka	Největší dávka	Průměrné hodnoty
Počet navštívených pozic		51	314	535	300
Stávající řešení	Celková délka D v metrech	869	2024	2760	1884
	Celková náročnost O v min.	127	506	805	480
OPCES 0,8	Celková délka D v metrech	660	1649	2222	1510
	Celková náročnost O v min.	110	475	760	448
Rozdíl	Úspora metrů	209	375	538	374
	Úspora metrů v procentech	24,1%	18,5%	19,5%	20,7%
	Úspora minut	17	31	45	31
	Úspora minut v procentech	13,7%	6,2%	5,6%	8,5%

Obr. 56. Výsledky testů okruhu jedna



Obr. 57. Grafické zobrazení výsledků testu okruhu jedna

Na obr. 58 jsou výsledky měření nejnáročnější dávky přes nejvytíženější ulici a řadu. Dalo by se spekulovat, že implementovaná metoda těží úsporu nejvíce z promyšleného průchodu přes regálové řady oproti nesystematičnosti prostého seřazení, ovšem právě měření přes ulici a řadu tuto domněnku vyvrací. Celková úspora na průchodu právě jednou ulicí, kdy bylo třeba projít přes 219 uzlů, činí 92 metrů, což je o 12 metrů více než délka celé ulice. Řekněme dobrá, je to v strategii průchodu, ale co výsledek na jedné řadě, kdy nelze šetřit křížováním z jedné strany ulice na druhou oproti původnímu řešení, kdy průchod ulicí byl fixně dán jednou řadou tam a druhou zpět. Na tuto otázku dává jednoznačnou odpověď poslední měření přes nejvytíženější řadu 14, kdy bylo třeba navštívit 115 uzlů jen v jedné řadě průchodem z jedné strany na druhou, jak naznačuje obrázek 58. Výsledek je vynikajících 52 uspořených metrů na celkové délce řady 80 metrů, což hovoří samo za sebe.

	Původní řešení v metrech	OPCES 0,9 v metrech	Úspora v metrech	Úspora v %
Ulice 4	869	777	92	11%
Rada 14	411	359	52	13%

Obr. 58. Výsledky měření optimální cesty přes nejvytíženější ulici 4 a řadu 14 největší expediční dávky

6.2 Přesnost aproximativního ohodnocení objednávek

Měření bylo provedeno na aplikaci vyhodnocených deseti nejnáročnějších objednávkách střední velikosti expediční dávky. Data získaná exaktním výpočtem ohodnocení objednávek oproti použitému aproximativnímu výpočtu v aplikaci OPCES 0,9 jsou zobrazena v tabulce na obr. 59 a graficky znázorněna na obr. 60.

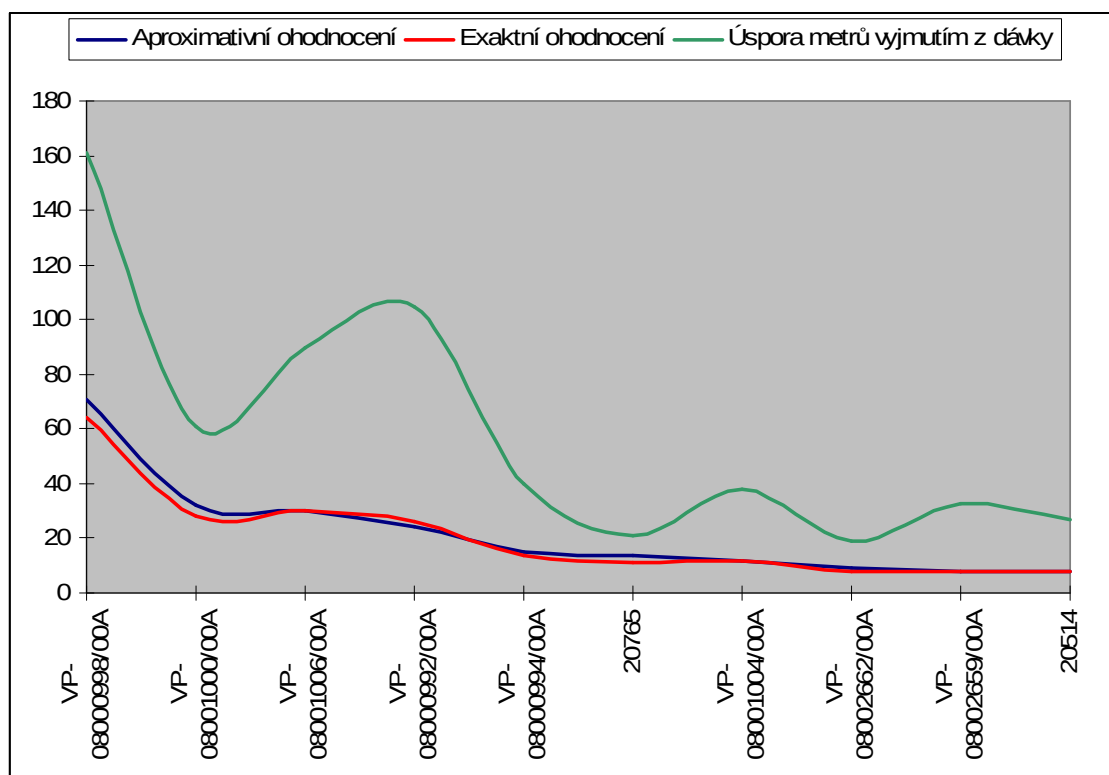
Pořadí	Objednávka	Aproximativní ohodnocení	Exaktní ohodnocení	Úspora metrů vyjmutím z dávky
1	VP-08000998/00A	71 min.	64 min.	161
2	VP-08001000/00A	32 min.	28 min.	61
3	VP-08001006/00A	30 min.	30 min.	90
4	VP-08000992/00A	24 min.	26 min.	105
5	VP-08000994/00A	15 min.	14 min.	40
6	20765	14 min.	11 min.	21
7	VP-08001004/00A	12 min.	12 min.	38
8	VP-08002662/00A	9 min.	8 min.	19
9	VP-08002659/00A	8 min.	8 min.	33
10	20514	8 min.	8 min.	27

Obr. 59. Výsledky testů okruhu dva

Výsledky měření přináší zdánlivě zvláštní chování systému, a sice, že objednávka A, jejímž vynětím z expediční dávky zkrátíme výslednou délku více než vynětím objednávky B, nepřinese větší časovou úsporu než objednávka B, což by se na první pohled nedalo očekávat. Konkrétním příkladem je objednávka číslo VP-08000992/00A, která celkovou délku trasy zkrátí o 105 metrů, ovšem celkový čas jen o 26 minut oproti objednávce číslo VP-08001000/00A, která zkrátí celkovou délku o pouhých 61 metrů, ale přinese časovou úsporu 28 minut. Jedná se o relativně nic neřešící dvě minuty, nicméně pointa je v ovlivnění času množstvím uspořených uzlů vynětím některé objednávky, neboť do výpočtu celkové časové náročnosti vstupuje parametr doby odbavení jednoho uzlu. Průměrná odchylka aproximativního řešení je pouhých 5 %, což je zcela zanedbatelné zkreslení oproti získané časové úspoře na práci procesoru.

Uvedené tvrzení dokladuje graf na obr. 60. Zde je vidět závislost aproximativního ohodnocení (modrá křivka) na celkové úspoře délky vynětím objednávky z expediční dávky (zelená křivka). Je evidentní, že aproximativní řešení reaguje na změnu délky jen velmi mírně a velmi dobře kopíruje exaktní řešení reprezentované červenou křivkou.

Závěrem je tedy nutné konstatovat, že s výtečnou přesností na jednotky minut je systémem OPCES 0,9 identifikována sestupná posloupnost nejnáročnějších objednávek expediční dávky.



Obr. 60. Grafické zobrazení výsledků testu okruhu dva

6.3 Test ideálního řešení

Znalosti ideálního řešení optimalizační úlohy hledání nejkratší trasy na pravidelné rektilineární mřížce při jejím plném obsazení (šachovnice) jsou často využívány pro testování úspěšnosti navržených optimalizačních algoritmů. Tomuto testu jsem se závěrem rozhodl podrobit i oba navržené algoritmy, které v systému OPCES 0,9 pracují. I když je již známo splnění cíle, tedy zkrácení cesty průchodu regálovým systémem expediční dávky oproti stávajícímu řešení, bylo by vhodné zjistit, jestli jsou v použitých optimalizačních algoritmech rezervy a pokud ano, tedy jaké.

Měření proběhlo na dvou regálových řadách. Prvně na plně obsazené řadě číslo osm, kterou obsluhuje algoritmus hraničních bodů, který je vybaven jinou strukturou prioritních pravidel oproti algoritmu průchodu blokových řad. Ten byl testován na plném obsazení regálové řady čtrnáct. Výsledky obou testů zobrazuje tabulka na obr. 61.

Algoritmus	Ideální řešení	OPCES 0,9	Současné řešení
Hranic a příčných řad	97 metrů	97 metrů	228 metrů
Bloků ulic	407 metrů	722 metrů	1127 metrů

Obr. 61. Výsledek testu okruhu tři

Naměřené hodnoty na první pohled zobrazují kde a jak velké se skrývají rezervy. Prioritní pravidla, která jsou nasazena na algoritmus řešící oblasti příčných regálů a hraničních bodů zapracovala natolik dobře, že donutila algoritmus najít ideální řešení průchodu nastraženou teoretickou šachovnicí. Je to opravdu znamenité zjištění a potvrzení toho, že zvolená strategie byla správná pro řešení předmětné problematiky této diplomové práce.

Naproti tomu byla odkryta silná rezerva v nastavení prioritních pravidel u algoritmu řešícího průchod ulicemi v bloku řad. Je vynikající, že i přes tento „neúspěch“ vykazuje systém vyšší efektivitu při tvorbě optimální trasy přes regálovou řadu než stávající řešení, což je také jednoznačně patrné z výsledků na obr. 61 a prakticky z příkladu viz test řady 14 v kapitole 6.1.

Výsledky testu ideálního řešení přináší vodítko, jak prioritní pravidla naladit tak, aby bylo dosaženo ideálního řešení, což je jeho největším přínosem. Zároveň odstraňuje pochybnosti o míře efektivity implementovaného řešení, a to oběma směry. Výsledek lze s čistým svědomím považovat za úspěch.

6.4 Časová náročnost výpočtů

Jedním z hlavních cílů diplomové práce bylo, aby výsledná aplikace byla prakticky použitelná v reálném provozu distribučního skladu. Řešení bylo od začátku úzce spjato s prioritou výpočetního času. Ověření, zda se podařilo dosáhnout zvoleného cíle, tedy akceptovatelného výpočetního času, za který lze považovat hodnotu kalkulace pod jednu minutu, sleduje právě poslední, třetí okruh testů.

Testy byly provedeny na nejnáročnější expediční dávce, a to na třech různých počítačích vždy v pětiminutovém odstupu třikrát za sebou. Při měření na počítačích nebyla spuštěna žádná jiná úloha. Stručný popis jednotlivých počítačů využitých k měření je následující.

1. Notebook Fujitsu Siemens Amilo Pro V 2035. Na tomto počítači byl celý systém naprogramován. Notebook disponuje následujícími základní parametry:

- operační systém MS Windows XP;
- Celeron M350 o taktovací frekvenci 1,5 GHz;
- operační paměť 1,2 GB;
- taktovací frekvence operační paměti 266 MHz.

2. Stolní počítač PC vlastnoručně složený z komponent a disponující následujícími parametry:

- operační systém MS Windows XP;
- Pentium 4 o taktovací frekvenci 3 GHz;
- operační paměť 512 MB;
- taktovací frekvence operační paměti 166 MHz.

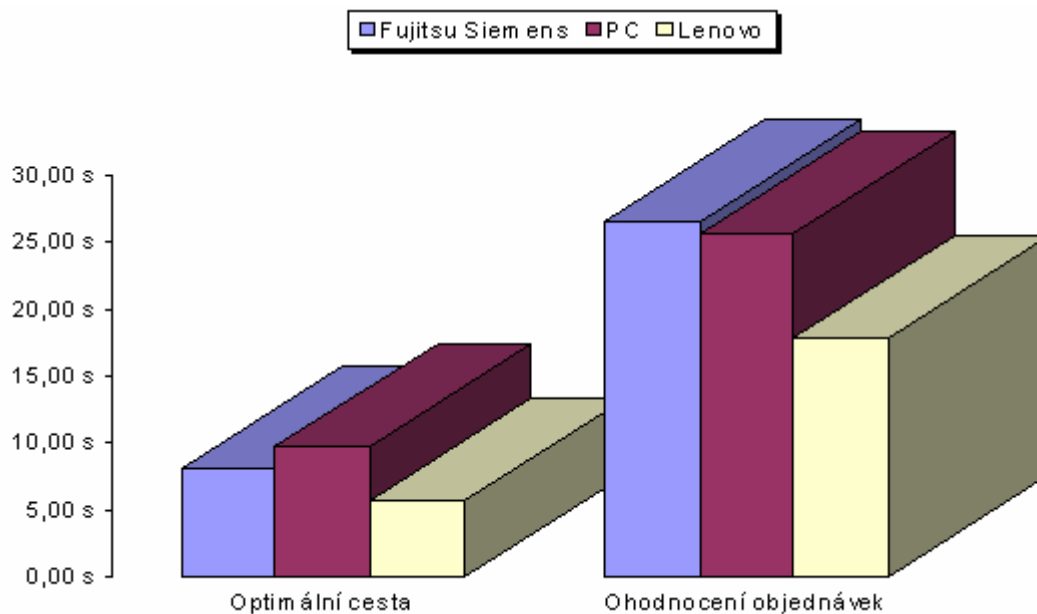
3. Notebook Lenovo R400 ThinkPad s následujícími parametry:

- operační systém MS Windows XP;
- Intel Core 2 Duo o taktovací frekvenci 2 GHz;
- operační paměť 1,92 GB;
- taktovací frekvence operační paměti 533 MHz.

	Počítač	Optimální cesta	Ohodnocení objednávek
Měření 1	Fujitsu Siemens	8,28 s	26,82 s
	PC	9,79 s	25,81 s
	Lenovo	5,75 s	17,81 s
Měření 2	Fujitsu Siemens	8,18 s	26,46 s
	PC	9,79 s	25,37 s
	Lenovo	5,71 s	17,81 s
Měření 3	Fujitsu Siemens	8,04 s	26,51 s
	PC	9,64 s	25,71 s
	Lenovo	5,70 s	17,93 s
Průměry	Fujitsu Siemens	8,17 s	26,60 s
	PC	9,74 s	25,63 s
	Lenovo	5,72 s	17,85 s

Obr. 62. Výsledky testů okruhu čtyři

Výsledky naměřených výpočetních časů jsou zobrazeny v tabulce na obr. 62 a graficky na obr. 63. Naměřené hodnoty prokazují, že i na starších počítačích a s hodnotami hardwarových prvků hluboce pod dnešní špičkou lze dosáhnout stanoveného cíle a tedy aplikace splňuje očekávání při nasazení v praktickém provozu. Při spouštění na moderním serveru lze počítat s ještě lepšími výsledky, tedy s výpočetním časem hluboce pod 30 s i při nejnáročnější expediční dávce.



Obr. 63. Grafické znázornění výsledků testů okruhu čtyři

7. Závěr

Tématem této diplomové práce je optimalizace expedice zboží z regálového systému logistického skladu. Problematika je řešena v konkrétním distribučním skladě a jeho regálovém systému. Hlavní cíle této diplomové práce jsou tři.

Prvně analyzovat stávající stav expedice objednávek z regálového systému, využití zdrojů a tvorbu plánu. Tímto tématem se relativně podrobně zabývají první tři kapitoly. Krátkým shrnutím výsledku provedené analýzy je uvedení do praktické problematiky expedice objednávek z regálového systému a specifikace přímé fokusace na konkrétní díl problematiky úlohy, který má být optimalizován. Tímto dílem je problematika tvorby optimální trasy průchodu regálovým systémem pro expediční dávku a metoda definice časové náročnosti jednotlivých dílčích objednávek a celkově expediční dávky.

Druhým cílem diplomové práce hned po specifikaci úzkého hrdla, které má být optimalizováno, je snaha o vytvoření takového nástroje, který by expedičním pracovníkům umožňoval efektivně vytvářet „jízdní řád“ pro obsluhu pozic v regálovém systému a zároveň dokázal ohodnotit celkovou náročnost expediční dávky v jednotkách času, tak aby odpovědní pracovníci dokázali pružně reagovat a optimalizovat tvorbu expedičních dávek, ze kterých díky metodě ohodnocování objednávek dokáží v případě časové tísně vyloučit vždy a jen právě tu objednávku, která přinese nejvyšší časovou úsporu při fyzické expedici z regálového systému. Zásadní motivací pro takovýto nástroj je vyvarování se přecenění velikosti expediční dávky a zabezpečení tak jejího zpracování v požadovaném limitním čase, a to i za cenu neprovedení expedice jen jednotek nejnáročněji ohodnocených objednávek namísto všech objednávek z celé dávky. Efektivní tvorbou „jízdního řádu“ je míněno vytvoření optimální trasy průchodu regálovým systémem, čímž je docíleno zkrácení vychystávajícího času a tedy zabezpečení maximalizace odbavených objednávek za limitní čas.

Třetím cílem je provedení a vyhodnocení srovnávacích testů. V případě optimalizace délky vychystávající trasy v regálovém systému bylo dosaženo průměrného zlepšení přes 20 %, což v reálu představuje řádově necelých 400 metrů na expediční dávku. Toto zkrácení přinese průměrnou úsporu časové náročnosti 8,5 %, což přibližně vychází mírně nad půl hodiny. Ačkoliv se může zdát, že ušetřená půlhodina je relativně krátký časový úsek, představuje to v případě dvou paralelně pracujících obslužných vozíků hodinu vychystávajícího času navíc, což je opravdu hodně. Test úspěšnosti zvoleného aproximativního ohodnocení časové náročnosti objednávek byl proveden ve srovnání s přesným ohodnocením, neboť stávající prakticky užívaná metoda expertního odhadu pracovníky skladu není pro testování relevantní. Aproximativní metoda byla zvolena s ohledem na výpočetní náročnost metody exaktní, jak je podrobně vysvětleno v kapitole 5.7. Výsledky zvolené metody jsou demonstrovány v kapitole Testy, okruh dva, podkapitola 6.2 a jsou nadmíru vyhovující. Předposlední provedený test si dal vyšší cíl než jen srovnání se současným stavem, a to porovnání kvality implementovaného řešení s ideálním řešením na ohraničeném teoretickém příkladu řešení optimální cesty na plně obsazené regálové řadě, což je obdoba úlohy průchodu šachovnice. Výsledek prokázal správnost zvoleného řešení optimalizační metody, i když zároveň ukázal, že je ještě třeba doladit algoritmus průchodu ulicemi v bloku řad. Bez tohoto testu by nebylo možno efektivně odhadnout ideální úspěšnost implementovaného řešení a najít tak místo, na které je třeba se dále

zaměřit. Posledním úkolem pro testování bylo zhodnotit časovou náročnost výpočetních úloh, tak aby bylo dosaženo výpočetní náročnosti pro jednu expediční dávku pod jednu minutu, což, jak testování prokázalo, je pro praktické případy splněno. Průměrná výpočetní doba na moderním notebooku je v součtu za obě nejnáročnější úlohy 23,57 s pro nejnáročnější expediční dávku. V případě provozu na moderním serverovém počítači se dá předpokládat snížení zhruba na polovinu.

Z řešení zadané problematiky vyplynulo, že další největší slabinou v systému expedice objednávek z regálového systému po optimalizaci trasy a ohodnocení časové náročnosti je reakční doba lidské obsluhy manipulační techniky, která hraje nejdůležitější roli v rychlosti pojezdu mezi odbavovanými uzly. Přírozeně je velmi obtížné bystře se zorientovat, kde se nachází další bod k odbavení a podniknout k němu rychlý a přesný přesun. Myslím, že řešení tohoto problému by mohlo být dalším podnětným a přínosným tématem pro samostatnou práci, stejně tak jako srovnání použité metody řešení s jinou užívanou metodou například GA na stejném typu úlohy.

Nyní několik slov ke koncepci implementovaného systému. Aplikace s názvem OPCES verze 0,9, jejíž název je odvozen z počátečních písmen slov optimální cesta, je naprogramována a přeložena jako samostatný program ve vývojovém prostředí Delphi 7. Zdrojová data importuje ze souboru aplikace MS Excel a výsledky dokáže exportovat přes schránku MS Office ve formátu oddělených buněk opět nejlépe do souboru MS Excel. Toto řešení je dostačující pro školní demonstraci, ale lze jej nasadit i prakticky. Nejedná se sice o nejelegantnější řešení pro praxi, nicméně je použitelné. Ideální praktické nasazení by znamenalo nejlépe implementaci do informačního systému skladového hospodářství, nebo alespoň přímé zprostředkování vzájemné komunikace pro výměnu dat. Toto ovšem může přinést až schválení nasazení systému vedením firmy a zadáním projektu implementace oddělení informačních technologií.

Ekonomické efekty z praktického nasazení systému jsou dvojího charakteru, a to přímé a nepřímé. Přímé mají za následek odstranění přesčasových hodin pracovníků obsluhy regálového systému zkrácením doby vychystávání. Nepřímé představují vyšší spokojenost klientů logistické společnosti, jimž bude plynuleji a hlavně rychleji proudit zboží k zákazníkům, kteří toto dozajista ocení zvýšenou poptávkou, což má vliv na obrátku zboží ve skladě a tedy na fakturaci za logistické služby. Spokojený klient zároveň hlásí dobré jméno svého logistického partnera ve světě byznysu, což je ta nejlepší možná reklama.

Závěrem několik statistických údajů k samotnému naprogramovanému systému. Zdroj v jazyce Object Pascal vývojového prostředí Delphi je rozdělen do devíti unitů o celkové obsahu 2 652 řádků kódu, což v porovnání s právě čteným textem diplomové práce, který má po toto místo 2 172 řádků svým rozsahem tento výrazně převyšuje. V programu je deklarováno a odladěno celkem 93 vlastních funkcí a procedur do, kterých nejsou počítány automaticky generované ovládací procedury komponent Delphi. Hodiny strávené fyzickým programováním nemám sečteny, nicméně řádově je odhaduji na 250.

Diplomová práce jako završení šestiletého úsilí o načerpání nových znalostí ze světa technických věd a osvojení si metod inženýrského přístupu k řešení složitých problémů praktického technického života je jistým měřítkem míry dosaženého. Věřím, že částečně touto diplomovou prací a poté dalším svým profesním životem budu dobře hájit a prokazovat dobré jméno brněnského Vysokého učení technického, zejména pak Fakulty strojního inženýrství.

Seznam použité literatury

- [1] POLIŠČUK, R.: Titulní strana závěrečné práce
- [2] POLIŠČUK, R.: Instrukce pro autory závěrečných prací, 2006,
http://autnt.fme.vutbr.cz/doc/SZZ2006_Instrukce.pdf
- [3] MICHALEWICZ, Z., FOGEL, B. D.: *How to solve it: modern heuristics*. Springer, 2004, 554 s. ISBN 3540224947
- [4] ŠEDA, M.: *Teorie grafů*. Sylabus VUT FSI Brno 2003.
- [5] OPLATKOVÁ, Z., OŠMERA, P., ŠEDA, M., VČELAŘ, F., ZELINKA, I.: *Evoluční výpočetní techniky - principy a aplikace*. BEN - technická literatura, 2008, 536 s., ISBN 80-7300-218-3
- [6] PÍSEK, S.: *DELPHI-začínáme programovat*. Grada Publishing a.s., 2002, 328 s., ISBN 80-247-0547-8
- [7] KVASNIČKA, V., POSPÍCHAL, J., TIŇO, P.: *Evoluční algoritmy*. STU Bratislava, 2000, 215 s., ISBN 80-227-1377-5.
- [8] PANUŠ, J.: *Evoluční algoritmy v optimalizačních problémech veřejné správy*. Pardubice, 2008. 130 s., Dizertační práce. Fakulta ekonomicko správní, Univerzita Pardubice.
- [9] ČERNÝ, J. < kuba@ matfyz.cz > *Základní grafové algoritmy*. [HTML dokument]. Matematicko-fyzikální fakulta, Univerzita Karlova v Praze [cit. 10. května 2009]. Dostupné z: <<http://kam.mff.cuni.cz/~kuba/ka/>>.
- [10] JAROŠ, J.: *Případová studie 1 – Problém obchodního cestujícího*. [HTML dokument]. Fakulta informačních technologií, VUT Brno, [cit. 13. května 2009]. Dostupné z: <<http://www.kapela-marshal.cz/O.pdf>>
- [11] COOK, W.: *Traveling Salesman problem* [HTML dokument]. 2008 [cit. 15. května 2009]. Dostupné na: < <http://www.tsp.gatech.edu/index.html> >
- [12] LUNER, P.: *Jemný úvod do genetických algoritmů*. [HTML dokument]. [cit. 16. května 2009]. Matematicko fyzikální fakulta, UK Praha. Dostupné na: <<http://cgg.mff.cuni.cz/~pepca/prg022/luner.html>>
- [13] BLAŽEK, J., HABIBBALA, H., PAVLIŠKA, V.: *Vybrané heuristiky pro globální optimalizaci a jejich implementace v Matlabu*. [online]. [cit. 16. května 2009]. Ostravská universita v Ostravě, Ostrava. Dostupné na: < <http://www.volny.cz/habiballa/publ/matlab05.pdf> >

- [14] FOLTA, J., LUKÁŠOVÁ, A. ŠIŠMA, P.: *Vojtěch Jarník*. [online]. [cit. 18. května 2009]. Masarykova universita Brno, Brno.
Dostupné na: < <http://www.math.muni.cz/math/biografie/> >
- [15] BARBUCHA, D., JEDRZEJOWICZ, P.: *An Agent-Based Approach to Vehicle Routing Problem*. International Journal of Applied Mathematics and Computer Science, vol. 4, no. 1, 2007, pp. 19–23.
- [16] GROSSENBACHER, S.: *Torry's Delphi Pages*. [online]. 2001.
Dostupné z: <<http://www.swissdelphicenter.ch/torry/tipresult.php>>
- [17] TROELS, J.: *Delphi stuff*. [online]. 2006.
Dostupné na: < <http://www.math.muni.cz/math/biografie/> >

Seznam obrázků

Číslo	Popis	Strana
1	Blokové schéma toku zboží a objednávek	3
2	Pohled mezi regálové řady	5
3	Pohled na regálové řady	5
4	Pohled na paletové pozice	6
5	Ukázka adresování pozic	6
6	Motorový nízkozdvihový vozík zvaný motorka	9
7	Vychystávací vozík	9
8	Graf počtu objednávek, položek a kusů na sledovaném vzorku dat za období 1.10.-28.11.2008 seskupeno po týdnech	10
9	Mapa regálového systému půdorys	11
10	Demonstrace adresování na části čelního pohledu regálové řady 3	12
11	Příklad konkrétní adresy regálové pozice z řady 3	13
12	Časový průběh procesů expedice	13
13	Část posloupnosti pozic stávající tiskové sestavy vychystávající trasy	15
14	Graf týdenního množství	16
15	Graf týdenních maxim	17
16	Tabulka sledovaných hodnot	17
17	Denní přehled množství pozic	18
18	Reprezentativní vzorek dat	18
19	Reprezentace 535 pozic přes regálové řady	19
20	Reprezentace 114 pozic řady 14 přes jednotlivé sloupce	19
21	Definice souřadného systému	22
22	Tabulka hran jednoznačně určujících vzdáleností v regálovém systému	22
23	Tabulka hodnot reprezentativního vzorku dat dle stávajícího způsobu zpracování expediční dávky	23
24	Půdorys trasy nejmenší dávky	23
25	Průchod regálovou řadou při plném obsazení metodou seřazení	24
26	Sir William Rowan Hamilton (1805 – 1865) [11]	27
27	Hamiltonova hra Icosian [11]	28
28	Tabulka vývoje řešení TSP dle velikosti instance [11]	28
29	Propojení 24 978 švédských měst [11]	29
30	Rozdělení prohledávacích algoritmů – převzato z [8]	31
31	Otakar Borůvka [14]	37
32	Vojtěch Jarník [14]	37
33	Datový model aplikace a rozhraní s okolím	39
34	Struktura vstupních dat	40

35	Struktura a typ dat objektu uData	40
36	Struktura a typ dat objektu uObjednávky	41
37	Struktura a typ dat objektu uMapa	41
38	Struktura a typ dat objektu uPozice	42
39	Struktura a typ dat objektu uHrany	42
40	Vlastnosti a metody objektů	44
41	Rozdělení regálového systému na jednotlivé podgrafy	45
42	Blokové schéma prohledávacího algoritmu	49
43	Blokové schéma algoritmu ohodnocení objednávek	51
44	Porovnání časové úspory aproximativního řešení oproti exaktnímu	51
45	Model datového toku vně aplikace	52
46	Struktura importovaných dat	52
47	Vzhled uživatelského rozhraní aplikace	54
48	Surová importovaná data zobrazená v datové mřížce	55
49	Zobrazení objektu uData následně po importu do datové mřížky	55
50	Zobrazení optimální cesty v datové mřížce	56
51	Obsah objektu uPozice v datové mřížce	56
52	Zobrazení grafu 4 do datové mřížky	57
53	Zobrazení souřadnic hraničních bodů do datové mřížky	57
54	Zobrazení ohodnocení objednávek v datové mřížce	58
55	Zobrazení datové sady určené k exportu z aplikace	58
56	Výsledky testů okruhu jedna	61
57	Grafické zobrazení výsledků testu okruhu jedna	61
58	Výsledky měření optimální cesty přes nejvytíženější ulici 4 a řadu 14 největší expediční dávky	62
59	Výsledky testů okruhu dva	62
60	Grafické zobrazení výsledků testu okruhu dva	63
61	Výsledek testů okruhu tři	64
62	Výsledky testů okruhu čtyři	65
63	Grafické znázornění výsledků testů okruhu čtyři	66