

Univerzita Hradec Králové
Fakulta informatiky a managementu

Diplomová práce

2006

Dušan Saiko

Univerzita Hradec Králové
Fakulta informatiky a managementu

Optimalizace konkrétního problému kontejnerové dopravní logistiky pomocí aplikace využívající genetický algoritmus

Diplomová práce

Autor:	Dušan Saiko
Studijní obor:	Informační management
Vedoucí práce:	Ing. Karel Mls

Hradec Králové

2006

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval (pod vedením vedoucího diplomové práce) samostatně a uvedl jsem všechny použité prameny a literaturu.

V Hradci Králové dne

.....

Anotace

Optimalizace konkrétního problému kontejnerové dopravní logistiky pomocí aplikace využívající genetický algoritmus.

Práce je zaměřena na optimalizaci konkrétního problému firmy Denios s.r.o. s distribucí kontejnerů výrobků do zahraničních centrálních skladů.

V teoretické části práce je nastíněna problematika dopravní logistiky a prozkoumány softwarové nástroje, které jsou při řešení logistických úkolů používány. Je zde popsán konkrétní problém distribuce kontejnerů, jeho současné používané řešení a možnosti použití existujícího software. Výsledkem teoretické úvahy je návrh aplikace na optimalizaci problému s využitím genetického algoritmu, úvod do genetických algoritmů, jejich výhody a nevýhody a jejich obhajoba pro zvolenou problematiku logistiky.

Praktická část práce implementuje aplikaci s prvky umělé inteligence, která ve svém teoretickém výsledku zlepšuje roční distribuční náklady firmy Denios s.r.o. z 5 529 000 Kč na 4 830 300 Kč, s úsporou téměř 700 000 Kč.

Pro svou charakteristiku optimalizace běžného dopravního problému najde aplikace v budoucnu praktické využití nejen ve firmě Denios s.r.o.

Klíčová slova: optimalizace, logistika, kontejnerová doprava, genetický algoritmus

Annotation

Optimization of container shipment with application using genetic algorithm.

The document is targeted to optimize the problem of Denios s.r.o company with distribution and shipment of their containers to the foreign central warehouse.

In the theoretical part of the work, the background of logistics is explained, and current software marked is research for existing application solutions. After the description of the exact problem of container shipment, the possibilities of using existing software are described. The theoretical part ends in suggesting implementing optimizing application, which uses genetic algorithm. Genetic algorithms are introduced and advocated for the use in the logistic area.

The practical part implements the application with the UI characteristics, which results in theoretical cutting the yearly distribution costs of Denios s.r.o. from 5 529 000 Kč down to 4 830 300 Kč, with savings of 700 000 Kč.

For the global characteristic of common problem of shipment optimization, the application will find its usage not only in the Denios s.r.o company.

Keywords: optimization, shipment, container, logistics, genetic algorithm

Poděkování

Děkuji tímto svému okolí za dlouhodobou podporu, bez níž by nešlo skloubit náročné zaměstnání a studium. Děkuji všem za potřebnou míru tolerance a trpělivosti při řešení chaotických situací všedního života. Především děkuji:

Ing. Karlu Mlsovi za vedení diplomové práce, podporu v realizaci zábavného a tvůrčího projektu, pomoc v rozhodování, trpělivost a lidský přístup.

Ing. Milanu Ruskovi za jeho dlouhodobé kamarádství, nápady a konkrétní problém optimalizace dopravy kontejnerů.

Firmě **Denios s.r.o.** za poskytnutí jména, reálných podkladů a dat k provádění výpočtů.

Zaměstnavateli, společnosti **mgm technology partners** za dobré pracovní prostředí, studijní úlevy a obecnou podporu ve studiu.

Fakultě Informatiky a managementu UHK za podporu kreativity studentů a zpravidla tvůrčí a přínosné složení předmětů.

A v neposlední řadě ženě a rodičům za jejich dlouhodobou podporu v životě i ve studiu.

Obsah

I.Úvod do dopravní logistiky.....	1
Vymezení pojmu logistika, její vývoj.....	1
Význam logistiky v ekonomice.....	2
Logistika v éře informatiky.....	3
Logistika ve vzájemné souvislosti s dopravou.....	6
Doprava a s ní související náklady.....	7
Vliv logistiky na systém dopravy.....	8
Vzájemné působení logistiky a skladového systému.....	9
II.Popis konkrétního problému firmy Denios s.r.o.....	10
Představení firmy.....	10
Popis problému.....	13
V současnosti používané řešení.....	15
III.Existující logistický software.....	17
Horry – ekonomický a logistický software.....	18
abas Business Software - ERP.....	19
Ostatní software.....	20
Výsledek rešerše.....	21
IV.Příležitost využití genetického algoritmu.....	22
Definice genetických algoritmů.....	22
Základní pojmy.....	24
Výhody genetického algoritmu.....	25
Nevýhody genetických algoritmů.....	26
Problém obchodního cestujícího.....	26
Hlavní problémy genetického algoritmu.....	27
Příklady nasazení.....	31
V.Existující využití genetických algoritmů v logistice.....	33
VI.Zadání aplikace.....	35
VII.Uživatelské rozhraní aplikace.....	37
VIII.Implementace genetického algoritmu.....	41
Chromozom.....	41
Umístění kontejneru.....	41
Vytvoření počáteční generace.....	42
Genetický algoritmus.....	43
Kriteriální funkce.....	44
Optimalizace.....	46
Parametry výpočtu.....	46
Výsledky výpočtů.....	48
IX.Další vybrané problémy implementace.....	52
Grafický posun – respektování hranic obdélníků.....	53
Knihovna objektů.....	58
Ukládání a nahrávání dokumentů.....	59
X.Příloha 1: Struktura přiloženého CD.....	60
XI.Příloha 2: Seznam obrázků a tabulek.....	61
XII.Příloha 3: Seznam odkazů a použité literatury.....	62

I. Úvod do dopravní logistiky

[1.Logistika] Záměrem logistiky je zabezpečit a organizačně zvládnout takovou techniku a technologie, které zajistí zrychlení, zpružnění a celkové zefektivnění ve výrobně - obchodních procesech podniku.

Vymezení pojmu logistika, její vývoj

Naučný slovník z let 1929-1932^[1.Logistika]:

Ve starověku až do roku 1600 praktické počítání s číslicemi, na rozdíl od aritmetiky, vědecké nauky o číslech.

Naučný slovník z roku 1966^[3.Slovník]:

Logistika

1. je idealistický formalistický směr v soudobé teorii logiky.
2. je matematická logika - někdy nazývaná symbolickou logikou.
3. vojenský termín, který v prvopočátku zastřešoval vojenská zásobování armád a jeho organizační složku založenou na moderních matematických metodách.

Největšího rozšíření našel pojem logistika v historii ve vojenství. Byzantský císař Leontos VI. (886-911) charakterizoval logistiku takto: *Předmětem logistiky je mužstvo zaplatit, příslušně vyzbrojit a vybavit ochranou i municí, včas a důsledně se postarat o jeho potřeby a každou akci v polním tažení příslušně připravit, tzn. vypočítat prostor a čas, správně ohodnotit terén z hlediska pohybu vojska, i možnosti protivníkovy odporu a tyto funkce zvládnout z hlediska pohybu vojsk i v případě nutnosti jejich rozdělení.*

V této větě je specifikována náplň logistiky, která musí zvládnout pohyby lidí a pohyby materiálu tak, aby se příslušný objekt nacházel na příslušném místě v potřebném čase.

Hospodářská logistika je disciplína, která se zabývá systémovým řešením, koordinací a synchronizací řetězců hmotných a nehmotných operací, vznikajících jako důsledek dělby práce a spojených s výrobou a oběhem určité finální produkce. Je zaměřena na uspokojení potřeby zákazníka jako na konečný efekt, kterého se snaží dosáhnout s co největší pružností a hospodárností. Pernica P., Praha 1994

Některé z dalších moderních definic logistiky jsou následující:

Systém tvorby, řízení, regulace a vlastního průběhu materiálového toku, energií, informací a přemísťování osob. JHDE, G.B.: Logistik. Stuttgart 1972

Souhrn činností, kterými se utvářejí, řídí a kontrolují všechny pohybové a skladovací pochody. Souhrou těchto činností mají být efektivně překlenuty prostor a čas.

PFOHL, H. CH.: Logistik systeme Betriebswirtschaftliche Grundlagen.

Berlin, Springer 1985

Je samozřejmé, že se logistika, tak jako každá vědní disciplína, vyvíjí. Nejprve sloužila k dosažení úspěchů ve válce, posléze je používána také v ekonomice. K tomuto spojení došlo v šedesátých letech, kdy zahrnovala takové pojmy, jako jsou optimalizační postupy při tvorbě, řízení a regulaci procesů vně i uvnitř podniku.

Význam logistiky v ekonomice

Vývoj logistiky se dá charakterizovat následujícími pěti fázemi:

Prvopočátek - termín vojenské logistiky:

Hlavním účelem byla optimalizace pohybu a zásob v armádě.

Optimalizace v oblasti distribuce zboží:

Logistika slouží ke zkvalitnění podmínek prodeje, k včasnému a cílenému dodání zboží zákazníkovi.

Uplatnění hlediska produkce:

V této fázi byli výrobci nuceni maximálně se přizpůsobit zákazníkovi a podmínkám zakázkové výroby. Docházelo ke sladování zpracování zakázek, výroby a distribuce, vyrábí se dle skutečných potřeb zákazníka.

Systém skladování:

Potřeby, představy o zboží jsou již jasné. Nyní se dostává na řadu plánování provozu s cílem výroby bez zásob, skladů - tzv. princip **just-in-time**. Tato myšlenka je už v jiné teoretické dimenzi, nazýváme ji maximálním logistickým managementem. Optimalizuje se čas mezi výrobou a spotřebou.

Komplexní systém logistiky:

Plánování toku materiálu a zboží se děje uplatněním nároků a podmínek dodavatelů, zákazníků a nositelů služeb. Logistika se stává nadpodnikovým kooperativním řídicím systémem toku materiálů, zboží a informací.

V budoucnu se očekává, že logistika zasáhne i do jiných oblastí podnikání. Celkově dále poroste význam logistiky také ve středně velkých podnicích. V budoucnosti bude čím dál tím menší šance uspět v konkurenčním boji bez vzájemného komplexního sladění všech procesů, týkajících se konečného výrobku.

Logistika v éře informatiky

Počínaje 21. stoletím je možné hovořit o éře informatiky. Využití logistiky v hospodářské praxi bude muset splňovat následující atributy^[4.Logistický management].

1. logistika se bude muset stát součástí globální strategie – časově podmíněné rozmísťování zdrojů jako zboží, lidí, kapacit a informací – působí ve smyslu substituce hmotných procesů za informační procesy
2. logistické služby budou muset být jedním z klíčových nástrojů k dosažení vyšší konkurenceschopnosti podniku při optimálních logistických nákladech
3. logistický systém bude muset být integrovaný, propojující všechny dodavatele se všemi distribučními a obchodními články až po konečného zákazníka. S ohledem na konkrétní finální výrobek či zakázku, zpětné toky odpadů a obalů k recyklaci či likvidaci, propojující výrobu s vývojem a s tvorbou koncepcí.

4. logistika bude muset být založena na:

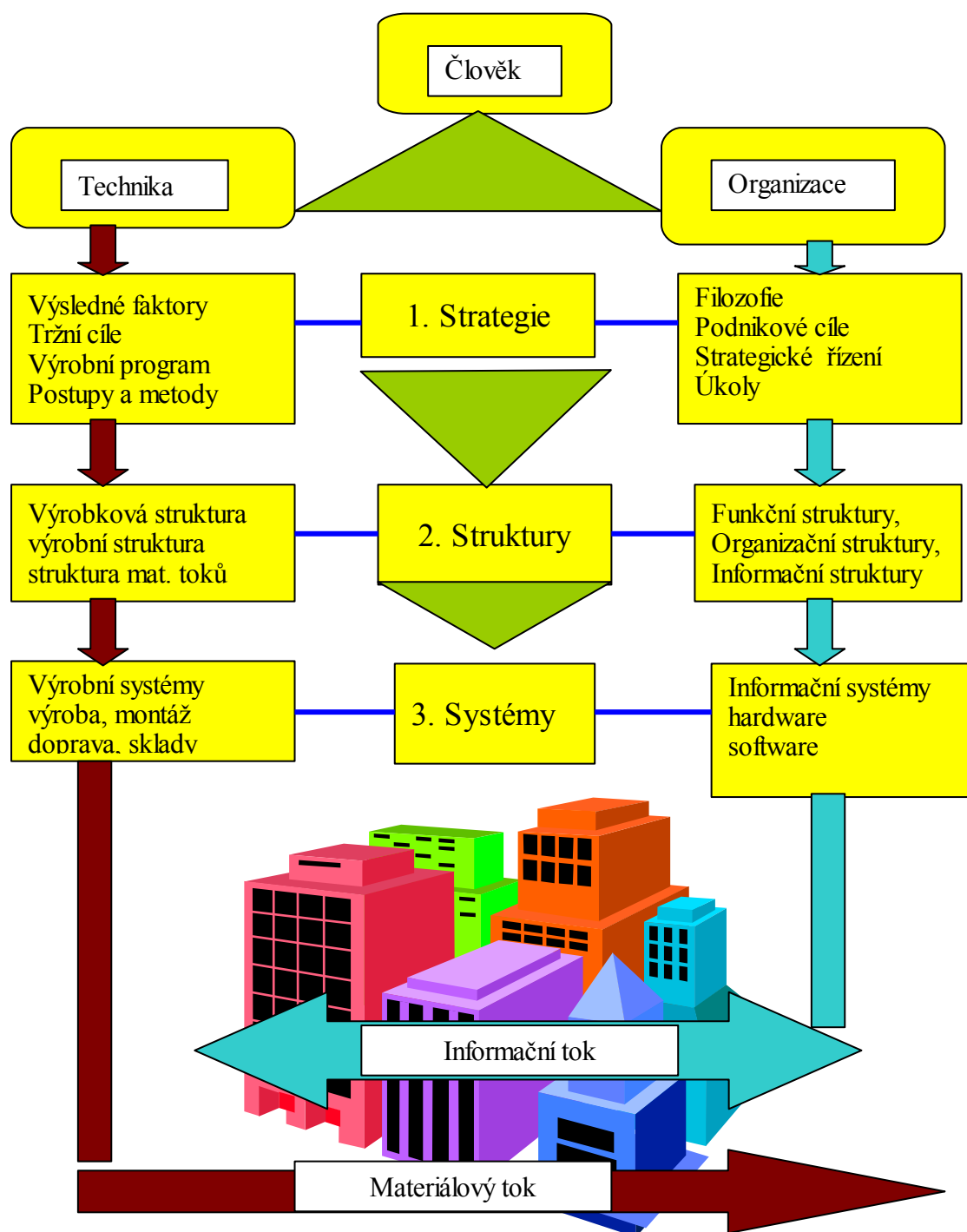
- prognózování, strategickém řízení, projektovém řízení, řízení procesů, informačních technologiích
- službách zákazníkům, distribuci, tvorbě zásob, inventarizaci zásob, opatřování a správě, nákupu a zásobování, exportu a importu
- plánování a operativním řízení výroby včetně stanovení výrobního sortimentu
- řízení materiálového hospodářství, balení, přepravě, skladování, manipulaci a recyklaci
- kvalitě

5. logističtí specialisté budou muset zvládnout:

- v oblasti profesního know-how: znalosti z oblasti logistické strategie, logistických systémů včetně dopravních systémů a distribučních center, plánování a řízení výroby, logistického přístupu ke globálnímu podnikání, opatřování a distribuci, distribuce v podmínkách evropského a celosvětového trhu
- v oblasti metodologického know-how: navrhování logistických systémů, výběr a oceňování prognostických modelů, logistických informačních systémů, lidských zdrojů, financí a controllingu, operačního výzkumu, modelování a simulace.

Vzhledem k celkové komplexnosti a komplikovanosti logistiky a navazujících oborů vzniká příležitost až nutnost k vývoji a využití moderních informačních systémů, které mají za úkol zautomatizovat a optimalizovat logistické procesy.

Následující ilustrace představuje informační a materiální toky v komplexním logistickém systému průmyslového podniku (zdroj ilustrace: logistika_prednasky_1.doc, www.seminarky.cz)



Ilustrace 1: Logistické toky v komplexním průmyslovém podniku

Logistika ve vzájemné souvislosti s dopravou

[2.Ekonomika dopravy] Logistikou ve skutečnosti rozumíme využití systémového myšlení pro všechny procesy v závodě z hlediska dopravy - pro dopravu, překládku, skladování, balení a distribuci. Co však vyžaduje současnost, je co nejkomplexnější přístup k problematice.

Jeho cílem je vytvoření optimálního řešení, které nám nabídne několik možných kombinací úsporných variant. Tyto možnosti musí obsahovat takovou kombinaci procesů a jejich nástrojů, aby byla v konečném pojetí zajištěna optimalizace dopravního procesu.

Obory problematiky logistiky:

- doprava
- překládka
- skladování
- balení
- distribuce

Nejjednodušším cílem optimalizace logistiky je minimalizace nákladů v jednotlivých dílčích oborech, například minimalizace dopravních nákladů, skladovacích nákladů a nákladů na balení. Vzájemná interakce jen do určité míry ovlivní návaznost jednotlivých fází.

Při odborném pohledu z hlediska komplexní logistiky je cílem nalézt vhodné sladění všech pěti fází, přičemž výsledkem je opět optimalizace, ale již zcela v jiné rovině. Velmi důležitá je návaznost těchto pěti procesů - vzájemná interakce je maximální. Z tohoto teoretického modelu vychází zcela jasný a velmi důležitý závěr: Odborník, který tuto problematiku spojí s logistikou, může navrhnout mnohdy i varianty řešení problémů, které si vyžádají vyšší logistické náklady. Na druhé straně však musí prokázat, že vzniknou vyšší výnosy.

Doprava a s ní související náklady

[2.Ekonomika dopravy] Mnohdy až 80 % všech logistických funkcí v rámci subjektu trhu připadá na organizaci pohybu materiálu, surovin, polotovarů a konečného produktu - výrobku. Doprava se tak stává integrující součástí jednotného logistického systému.

Lze ji charakterizovat těmito procesy :

- přeprava surovin, pomocných, provozních látek a polotovarů k místu výroby
- přeprava na pracovišti během výrobního procesu
- přeprava polotovarů mezi různými pracovišti vlastního podniku
- přeprava hotových výrobků pro obchod
- přeprava ze skladu do obchodu nebo přímo ke konečnému spotřebiteli nebo uživateli
- přeprava z obchodního podniku ke konečnému spotřebiteli nebo uživateli

V současném vývoji dopravy lze pozorovat následující změny:

- se vzrůstající technikou, s rozvojem vědecko – technického pokroku stoupá i složitost vyráběných výrobků. Tento fakt klade vyšší a vyšší nároky na kvalitu přepravy. Poměr přepravy surovin, materiálů vzhledem k přepravě polotovarů, či hotových výrobků markantně klesl.
- se vzrůstající dělbou práce roste i počet článků či fází v přepravě
- systémy pro přepravu se díky vědecko-technickému pokroku stávají méně problematickými
- přepravní náklady v posledních dvou desetiletích relativně klesly

Vliv logistiky na systém dopravy

[2.Ekonomika dopravy] Skutečnost propojení a vlivu logistiky a systému dopravy se v současnosti stala tzv. logistickým axiomem. Dříve se samozřejmě také odborníci snažili o koordinaci všech činností a procesů souvisejících s pohybem materiálu a zboží. Mnohdy však zamýšlený cíl nenalezl naplnění, protože chyběla taková technika, která by zajišťovala optimalizaci zmíněného procesu. Můžeme tedy říci, že dopravní procesy jsou ve velké míře ovlivněny a podmíněny vyšší logistickou kvalitou. Tato kvalita spočívá v koordinaci přepravy a systému vývoje zakázek, skladovacími a překládkovými systémy, balicími systémy a systémy přepravních pomocných prostředků, informačními systémy atd. Je jen otázkou, do jaké míry logistický systém přepravy zboží změnil samotnou podstatu a formu dopravních systémů.

K její změně také přispěly tyto faktory :

- charakter zboží
- marketing
- alokace podniku
- marketingová strategie, filozofie a instrumentarium podniku

Tyto faktory tlačí na moderní dopravní systémy a staví je do situace, kdy mezi jejich nezbytností musí patřit:

- časová adaptabilita, připravenost
- alternativní možnosti využití
- jednotnost všech fází pohybu zboží - následkem je snazší průběh přepravního procesu
- maximální flexibilita týkající se kapacity a specializace
- cena související s celkovými náklady na přepravu
- spolehlivost
- respektování životního prostředí

Vzájemné působení logistiky a skladového systému

Jednou z funkcí podnikového logistického systému je snížení nákladů při skladování a překládce zboží. Existují dvě možnosti, jak tento cíl splnit:

1. Výroba a distribuce bez zásob

Při materiálním zásobování výroby a při rozdělování zboží se vylučuje jakékoli skladování materiálu, popř. zboží přímými dodávkami, popř. přímou expedicí /Just-in-time/. Upouští se také od shromažďování do větších ložných jednotek a dopravuje se v množstvích, která jsou zpracována v jedné směně nebo mohou být spotřebována v relativně krátkých časových objednacích či dodacích lhůtách. Přes relativně vysoké náklady na častější přepravu je tento způsob v konečném posouzení efektivní. Musíme si tudíž uvědomit, že celkové náklady mohou být omezeny, protože se tím ušetří vyšší náklady na skladování a překládku.

2. Centrální skladový systém

Tento systém se prosazuje stále více tam, kde se musí předpokládat rozdělování zboží z více výrobních stanovišť nebo je naopak třeba dodávat na mnoho přijímacích stanic. Tento systém probíhá tím způsobem, že ze všech výrobních míst, kde se odehrávala výroba, se přepraví výrobky do centrálního skladu. Zde se sestavují dle požadovaných představ - objednávek. To často znamená úsporu minimálně dvou skladových stupňů. Úspory skladovacích nákladů jsou vyšší než přírůstek nákladů dopravy, takže celkové náklady v distribučním systému poklesly.

II. Popis konkrétního problému firmy Denios s.r.o.

Představení firmy

[www.denios.cz]

Založení

Nynější DENIOS AG byl založen roku **1986** jako "P&D Systemtechnik" v Porta Westfalica. Nosnou ideou podniku je ochrana životního prostředí, výroba produktů a know-how sloužících k zajištění čistoty půdy a vody.

1987 byl podnik přemístěn na své nynější sídlo do Bad Oeynhausenu a již v roce **1988** otevřel první prodejní pobočku ve Stuttgartu.

Expanze do Severní Ameriky

1990 byl založen P&D Systemtechnik Inc. Louisville, KY/ USA.

Expanze v Evropě

V letech **1992-2000** následovala výstavba dceřiných společností v Itálii, Francii, Nizozemí, Rakousku, České republice a rovněž v Belgii a Velké Británii. K tomu byla otevřena pobočka v Lipsku. Od roku **2001** kooperace se Španělskem.

Změna názvu a loga

1999 se firma P&D Systemtechnik GmbH přejmenovala na DENIOS AG.

Vstup na trh ve Skandinávii

2001 rozšiřuje DENIOS AG své aktivity založením DENIOS AB ve Švédsku (Malmö). Od října **2001** se nachází sídlo firmy v Jönköpingu.

Kvalita a bezpečnost

Všechny produkty podléhají přísným požadavkům na kvalitu , bezpečnost a permanentním kontrolám TÜV dříve, než obdrží značku kvality nebo stavební povolení, a tak mohou být vyráběny v sériích. Spolupracovníci firmy DENIOS jsou rovněž zainteresováni. Speciálně v oblasti ochrany životního prostředí vytvořilo vedení podniku stimulaci pro všechny zaměstnance. V roce **2000** byla poprvé propůjčena čestná cena jakosti a životního prostředí jako prémie za nejlepší nápad.

Produkce firmy

Společnost se zabývá podporou skladových systémů k ochraně životního prostředí, výrobků pro skladování nebezpečných látek (zvláště záchytných van), tepelných komor a dalších výrobků.

Česká pobočka

Česká pobočka Denios s.r.o sídlí ve Strakonících, městě s dlouhodobou strojírenskou historií (ČZ Strakonice zaměstnávalo ve své největší slávě až 9 000 zaměstnanců), a zaměstnává v současné době přibližně 50 zaměstnanců.

Pobočka se zabývá dovozem a výrobou hliníkových van a jiných výrobků dle zakázky mateřské německé firmy. Hotové výrobky se kamiónovou dopravou rozváží zpět do centrálních skladů v Německu, Francii a Itálii.

Následující strana obsahuje úvodní list z produktového katalogu.



Ochrana prostředí /

Skladování nebezpečných látek

Vybavení výroby /

Manipulační systémy



Bezpečnost práce /

Osobní ochranné prostředky



Obsah

1	Záchytné vany z oceli, nerezové oceli a plastu	od strany 5
2	Regály na nebezpečné látky a sudové regály	od strany 46
3	Skříně na nebezpečné látky a sudové skříně	od strany 58
4	Skladové systémy pro skladování na volném prostranství	od strany 64
5	Skladování plynových lahví	od strany 100
1	Manipulační a transportní systémy	od strany 107
2	Čistící systémy	od strany 125
3	Bezpečnostní nádoby	od strany 130
4	Vytápěné sklady	od strany 133
5	Míchací zařízení a čerpadla	od strany 136
6	Příslušenství	od strany 143
1	DENSORB-sorbenty a sady pro nouzové případy	od strany 152
2	Těsnící rohože a bariéry	od strany 166
3	Vybavení pro nouzové případy a prostředky první pomoci	od strany 169
4	Ochranné pracovní pomůcky	od strany 176
5	Bezpečnostní vybavení	od strany 178

Popis problému

Rozvoz produktů

Firma vyrábí produkty, které následně vyváží do skladů mateřské společnosti, a to konkrétně do Německa - Bad Oeynhausenu, Francie – Nassandres, a Itálie - Isola del Cantone (GE).

S nákladem je vždy vypraven kamión s jedním návěsem, a to s přibližně následující frekvencí cest:

- do Německa 2 až 3 kamiony týdně, cena cesty 26 000 Kč
- do Francie 1 kamion týdně, cena cesty 37 500 Kč
- do Itálie 1 kamion měsíčně, cena cesty 30 500 Kč

Rozměry typického návěsu jsou 13450 mm x 2450 mm, celková ložná plocha 32,95 m², nosnost 24 tun.

Při nakládání se sledují následující charakteristiky:

- celková hmotnost nákladu nesmí překročit maximální nosnost návěsu, rozdělení hmotnosti nákladu musí být rovnoměrné v úložném prostoru – jak na přední a zadní nápravu, tak na levou a pravou stranu návěsu, aby nedocházelo k naklánění kamionu na jednu stranu
- rozměry kontejneru s výrobkem jsou různé, v některých kontejnerech je již zabaleno více stejných výrobků
- výrobky jsou již baleny v kontejnerech v přibližné výšce odpovídající výšce úložného prostoru návěsu, případně se ví o konkrétních výrobcích, na které je možno ještě nějaký menší produkt přiložit
- současné využití ložné plochy je v průměru 80% - 26,36 m²
- současné využití hmotnostní kapacity návěsu je přibližně 50 až 75 % - 11 až 18 tun.

- organizaci nákladu kamionu má na starosti mistr výroby, kterému organizace jedné dodávky trvá přibližně 2 až 4 hodiny. Hodinová sazba na mistra výroby je přibližně 450 Kč/hodinu.

Odvoz výrobků se provádí z množiny hotových produktů, kterých je ale více, než je možno odvést, tedy ve skladu je více výrobků než lze odvést.

Některé výrobky musí být v zásilce zahrnuty, z ostatních lze vybírat.

Zpáteční cestou kamionu se není třeba z tohoto pohledu zabývat, pokud se kamion vrací zpět, veze obvykle těžké strojírenské suroviny či polotovary (plechy), které zdaleka nevyčerpají objem nákladního prostoru při dosažení maximální nosnosti kamionu.

Definice problému:

Problém, na který se v implementační části diplomové práce chci soustředit, jsem si vybral jako názorný příklad chybějícího využití možností informačních technologií.

Jde o systematickou, opakovanou, ale manuální a netvůrčí činnost, která pravidelně a zbytečně dlouho zaměstnává odbornou pracovní sílu firmy, a tedy neumožňuje její efektivní a tvůrčí využití.

Mistr výroby organizuje dle aktuálních požadavků náklad do kamionu a sleduje výše uvedené charakteristiky. Jeho snahou je co nejefektivnější organizace zboží nákladu, tedy co nejvíce převezeného zboží v jednom kamionu, respektive co nejméně vyslaných kamiónů. Dle zadání je cena dopravy zboží měsíčně přibližně 450 000, je vypraveno 15 kamionů, což zabere odborné kvalifikované síle až 45 hodin organizačního času, v odhadované ceně 20 250 Kč.

Možnosti zvyšování efektivity využití nákladového prostoru jsou vzhledem ke kontejnerové charakteristice přepravy omezené.

Hlavním problémem je časová náročnost mechanické úlohy, která je komplikovaná častou změnou požadavků a tedy přeorganizováním již připraveného nákladu.

V současnosti používané řešení

Společnost Denios s.r.o. postupně vyvíjela svůj přístup k organizaci nákladu kamionu. Nejprve docházelo k intuitivnímu, ale neformálnímu způsobu organizace kontejnerů.

K zvýšené efektivitě došlo již pouhou jednoduchou, ale časově náročnou simulací rozložení kontejnerů pomocí rozstříhaných obdélníkových papírů.

Vedení společnosti se také pokoušelo najít některý použitelný a finančně nenáročný software, bohužel se nepodařilo najít žádnou specializovanou ani podobnou aplikaci s automatickým rozmisťováním předmětů.

V současné době je tedy používána jistá forma počítačové podoby posouvání obdélníkových papírů, nyní již ale v elektronické podobě, s možností popisů, ukládání a tisknutí dokumentů. Inteligentní rozmístění nákladu a veškerá kontrolní činnost je ale povinností lidské obsluhy programu.

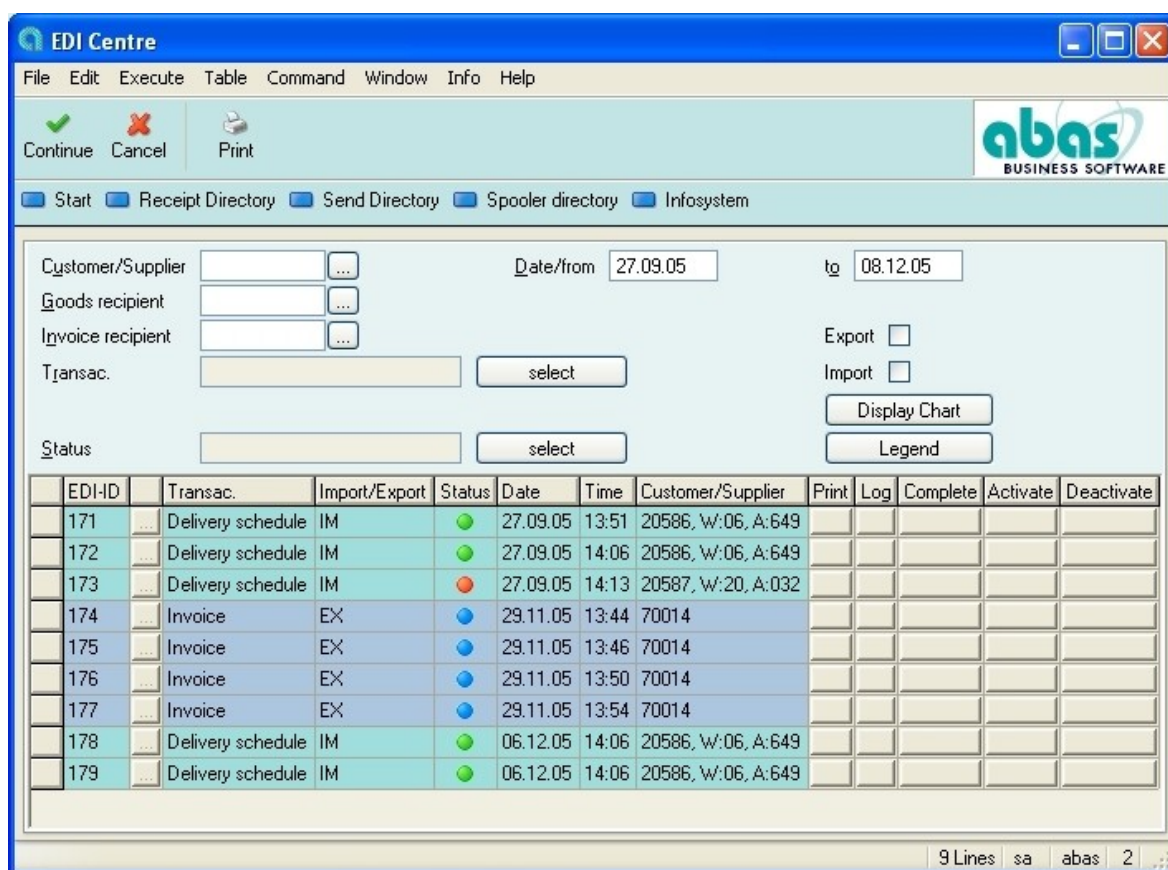
Používaná aplikace Room Arranger je určena na jednoduché bytové návrhy, nákladní prostor tvoří jedna místnost, kontejnery jsou reprezentovány v podobě obdélníkového nábytku uloženého v knihovně objektů.

Současný stav je prověřený praxí, nicméně není optimální, a přitom lze jednoduchou „vtipnou“ aplikací v kombinaci se zkušeností obsluhy a jejími konkrétními poznatky z praxe dosáhnout výrazného zlepšení, zjednodušení a ušetření práce i nákladů.

III.Existující logistický software

Logistika, jako nedílná část ekonomických systémů pro celofiremní řízení, je již samozřejmě mnohokrát zpracována v rámci existujících informačních řešení.

Cílem této kapitoly je zmapovat logistickou funkčnost největších a nejznámějších aplikačních řešení, uvést funkčnost některých z nich a pokusit se najít program, který by umožňoval řešit specifický problém organizace distribuce kontejnerů v nákladovém prostoru kamionu.



Ilustrace 4: Logistický modul ekonomického softwaru ABAS

Horry – ekonomický a logistický software

[www.horry.cz]

Tento software je typickou ukázkou komplexního ekonomicko – logistického řešení fungování průmyslového podniku.

„Hlavní činností společnosti je implementace a vývoj informačního systému pro celkové řízení firmy. V současné době využívá náš systém cca 1100 stálých zákazníků, kteří se dělí na uživatele DOSovské verze a našeho nového systému HorryWin. Při realizaci zakázky jsme schopni poskytnout související služby jako například realizaci a správu počítačové sítě, dodávku hardware či poskytování poradenských služeb.“

Nabízené komponenty systému HorryWin:

Obchod a sklady: Patří mezi stěžejní součásti operativní evidence projektu HorryWin. Umožňuje komplexně řešit problematiku firmy v oblasti obchodu počínaje uzavřením smlouvy s obchodním partnerem, přes evidenci objednávek dodavatelských i odběratelských, rozvozových linek, dále kompletní evidenci skladů až po odběratelskou fakturaci.

Účetnictví a ekonomika: Představuje komplexní nabídku účetní a finanční evidence firmy. Vzhledem k možnosti širokého pojetí problematiky zpracování ekonomických informací je velmi vhodným produktem pro středně velké a velké organizace účtující v soustavě podvojného účetnictví.

Mzdy a personalistika: Je operativní agenda projektu HORRY, která umožňuje komplexně řešit problematiku firmy v oblasti lidských zdrojů. Agendu tvoří dva na sobě nezávislé moduly, jejichž pomocí lze evidovat pracovní síly a zpracovávat mzdy.

Majetek: Je operativní součástí projektu HORRY. Umožňuje komplexně řešit problematiku vaší firmy v oblasti dlouhodobého majetku a drobného dlouhodobého majetku.

Doprava a expedice: Patří mezi základní součásti operativní evidence projektu HorryWin. Umožňuje komplexně řešit problematiku firmy v oblasti dopravy počínaje přesnou

a podrobnou evidencí jízd vozidel osobní a nákladní dopravy, přes evidenci různých nákladů na jednotlivé stroje, technické údržby a prohlídky až po zpracování podkladů pro mzdy a stravné. **Nákladní doprava** umožňuje vést podrobnou evidenci dokladů o provozu vozidla nákladní dopravy a mechanizované operace. Uvedené doklady obsahují přesný popis prováděných prací (instrukce nebo příkaz k jízdě) s ohledem na časové, fakturační a provozní podmínky. Je v nich dispečerem určena osádka (řidič, závozníci), motorové vozidlo a datum, kdy se má jízda uskutečnit podle uvedených dispozic. Do již vydaného dokladu potom řidič průběžně zaznamenává jednotlivé části jízdy a provedené úkony údají jako jsou ujeté kilometry (celkové a s nákladem), doby jízdy a různých prostojů, co vezl za náklad a jaké množství v tunách a kudy jel včetně všech případných mimořádností během jízdy. Řidič po skončení jízdy před odevzdáním dokladu k dalšímu zpracování doplní stav tachometru po příjezdu, počáteční a koncové stavy, čerpání a spotřebu pohonných hmot.

abas Business Software - ERP

www.abas.de

Komplexní ekonomicko organizační software obsahující následující výčet funkcí.

Nákup: požadavky, příkazy, potvrzení, faktury, stanovování cen

Celkové sledování nákupní transakce od požadavku, k nákupnímu příkazu, potvrzení o nákupu a ověření faktury. Možnosti sledování cen zboží a generování nákupních příkazů při výhodné ceně zboží. Každý nákup je propojen s finančním účetnictvím.

Prodej: nabídky, příkazy, odchozí zboží, doprava, fakturace

Prodejní proces je zaznamenáván od vytvoření nabídky, k příkazu k prodeji, odeslání zboží a jeho fakturaci. Funkce nabízejí historii nabídky, statistiky pro obrát, obchodní marži, cenové trendy zboží a další. Použitím systému se zmenšují skladové zásoby, zvyšuje se profitabilita zboží.

Sklady: skladové hospodářství, velikosti dodávek, správa sériových čísel, alokace materiálu, trvanlivosti materiálů.

Plánování: plánování požadavků, velikosti zásob, kapacity.

Úkolem plánování je zajistit dostatečné zdroje a kapacity pro požadovanou produkci výrobků, kontrolu a odhad dodacích lhůt.

Plánování a kontrola produkce: návrhy produkce, pořadí prací, plánování kapacit, využití zdrojů, správa objednávek množiny produktů, kontrola produkce a její kvality.

Rozpočet: předběžný rozpočet, celkový rozpočet, porovnání celkového výsledku produkce s možností prohlížení po produktech či objednávkách.

Další funkce:

Finanční účetnictví

Nákladové účetnictví

Ostatní software

Dále byly prostudovány funkčnosti (nejen) následujících logistických informačních systémů:

<http://www.truckmaster.com> – TruckMaster Logistic System - Software pro řízení přepravy

<http://www.mjc2.com/> - Software pro distribuční plánování, strategickou logistiku a plánování výroby

<http://www.logility.com/> - Supply Chain Management Software Solutions for Inventory management, Forecasting and Demand Planning.

<http://www.vema.cz/> - dodavatel ucelených řešení pro řízení lidských zdrojů, ekonomiky a logistiky.

<http://www.tecsys.com/> - Transportation Management Software

<http://www.vanderlande.com> - Vision - The Logistic Software Suite
Logistický software propojený s informačním systémem SAP.

<http://www.progress.com/> - Distribution & Logistics Solutions

Výsledek rešerše

Existuje obrovské množství logistického softwaru, stále především od mezinárodních společností, které čerpají výhody z velkého nadnárodního trhu a dlouhodobější provázanosti firem s informačními systémy.

Pokusil jsem se zmapovat některé existující produkty, zajisté jsem nevyčerpal množinu všech možných softwarových řešení, nicméně jsem došel k následujícímu zobecněnému závěru:

Většina logistických řešení je zaměřena zobecněně a komplexně na celkovou ekonomickou oblast logistiky, spravování databáze zakázek, faktur, vyúčtování, případně v návaznosti na správu lidských zdrojů a mzdové systémy.

Další velká kategorie programů se zabývá skladovým hospodářstvím a minimalizací skladových zásob.

Některé speciální aplikace jsou určeny či obsahují moduly pro sledování pohybu zakázky, plánování cesty dopravy, organizace doručení zásilky zákazníkovi.

Nenalezl jsem ale žádné řešení pro faktickou konkrétní organizaci nákladového prostoru. Buď jde o problém speciální, nebo přehlížený a řešený triviálním a rutinním intuitivním postupem. Ve většině případech je také zakázka pevně daná, bez možnosti její optimalizace. Pokud by ale existoval software, buď ve formě samostatného programu, nebo modulu do existujících softwarových balíčků, určitě by se našlo hodně firem, které by ho dokázaly využít a pro něž by byl přínosem. I při pevně zadané dopravní zakázce by bylo dobré mít možnost zdokumentovat organizaci nákladního prostoru a vědět statistiky využití nákladové plochy či rozložení hmotnosti nákladu.

IV. Příležitost využití genetického algoritmu

Pokud hledáme možné řešení zadaného problému, budeme se zabývat prvky umělé inteligence, která by nahradila manuální kombinatorickou činnost organizace nákladového prostoru.

Vhodným výrazovým prostředkem pro řešení takové úlohy jsou genetické algoritmy, které umožní při znalostech požadavků na rozložení nákladu určit optimálně jeho organizaci, a to včetně aspektů rovnoměrného rozložení hmotností a dalších kritérií.

Zautomatizování procesu přinese dvě výrazná pozitivní zlepšení.

Využití ložné plochy návěsu se maximalizuje, je možné, že se do nákladového prostoru vejde více zboží než při manuální organizaci.

Druhým viditelným efektem bude zrychlení organizačního procesu a eliminace časových ztrát při reorganizaci nákladu podle změny požadavků.

Definice genetických algoritmů

[5.GA.U1] Genetické algoritmy patří do třídy evolučních algoritmů, které mimo ně zahrnují také evoluční programování, evoluční strategii a genetické programování. Jsou to vyhledávací algoritmy založené na mechanismu přirozeného výběru a principech genetiky. Jejich velkou výhodou je poměrná jednoduchost. Ideovým vzorem pro genetické algoritmy byly principy vývoje, které se uplatňují v přírodě. Zde existují populace jednotlivých živočišných druhů složených z jedinců různých vlastností. Tyto vlastnosti jsou prvotně zakódovány v jejich genech, které tvoří větší celky, chromozomy. Při křížení vznikají noví jedinci, kteří mají zpravidla náhodně část genů od jednoho rodiče a část genů od rodiče druhého. Přitom ve zvlášť výjimečném případě může dojít k náhodné změně některého genu v chromozomu, tzv. mutaci, která může být pro další vývoj druhu příznivá nebo ne. Podle svých vlastností má každý z potomků větší nebo menší schopnost obstát v přirozeném výběru a vytvořit další generaci. Proces výběru se stále opakuje a v jeho průběhu se zlepšují genetické vlastnosti daného druhu. Tak probíhala celá evoluce v přírodě.

V teorii umělé inteligence je genetický algoritmus procesem postupného vylepšování populace jedinců opakovanou aplikací genetických operátorů, který vede k evoluci takových jedinců, kteří lépe vyhovují stanoveným podmínkám než jedinci, ze kterých vznikli. Proces konverguje k situaci, kdy je populace tvořena jen těmi nejlepšími jedinci. Hlavním principem je kopírování a vyměňování řetězců - chromozomů. Chromozom má pevnou délku, jednotlivé pozice tvoří geny. Geny reprezentuje často binární 0 nebo 1, ale obecně mohou mít libovolnou hodnotu. Množina chromozomů tvoří populaci. Každý chromozom v populaci má definovanou hodnotící funkci, nazývanou fitness funkce, která charakterizuje vhodnost chromozomu.

Genetický algoritmus definuje operátory křížení, mutaci a reprodukci. Operátor křížení znamená vytváření nových jedinců podle následujícího pravidla: Z populace vybereme náhodné páry a náhodně zvolíme pozici k . Do prvního potomka zkopírujeme geny 1 až k prvního rodiče a geny $k + 1$ až n druhého rodiče, kde n je počet genů, a do druhého potomka kopírujeme geny opačně.

Při aplikaci operátoru mutace vybereme s malou pravděpodobností náhodný gen v náhodném chromozomu a změníme jeho hodnotu z 0 na 1 nebo naopak. Operátor reprodukce kopíruje chromozomy do nové populace. Chromozomy s vyšší fitness hodnotou jsou do nové populace kopírovány s vyšší pravděpodobností. Pravděpodobnost kopírování je dána vzorcem $p_i = f_i / \text{Suma}(f)$, kde

- p_i = pravděpodobnost reprodukce i -tého chromozomu
- f_i = hodnota fitness funkce i -tého chromozomu
- $\text{Suma}(f)$ = součet všech hodnot fitness funkce v populaci

Někdy se používá tzv. elitářský mechanismus: Určité procento nejlepších jedinců je do nové generace reprodukováno vždy.

V praxi se pomocí genetických algoritmů řeší úlohy optimalizace, využívají se k vyhledávání nejlepší topologie, v technologii a výrobě a v průmyslové automatizaci a jako alternativní metody učení neuronových sítí. Na rozdíl od gradientních metod, které reprezentují hledání lokálního minima nebo maxima pomocí jednoho zpřesňujícího

se řešení, představují genetické algoritmy jiný přístup, který používá populaci prozatímních řešení, jež paralelně procházejí parametrický prostor a navzájem se ovlivňují a modifikují pomocí genetických operátorů. Tím se dosahuje toho, že populace jedinců najde správné řešení rychleji, než kdyby se prohledával prostor izolovaně.

Základní pojmy

Při popisování principů použitých v genetickém programování se setkáváme s následujícími pojmy, které je vhodné si uvést definicí:

Algoritmus - je popis teoretického řešení problému, kterým lze vyřešit určitý typ úlohy

Evoluce, evoluční proces – pozvolný průběžný vývoj, který se řídí postupným zdokonalováním či adaptací na konkrétní prostředí. V biologickém prostředí je evoluce založena na cyklické výměně generací, které si předávají, obměňují a zdokonalují své vlastnosti.

Populace – rozsáhlejší množina prvků, v našem případě budeme mluvit o populaci jako o množině různých algoritmů (jednotlivců), kteří mají schopnost množit se, definovat svou znalost řešení algoritmu a předávat ji dál.

Gen – charakteristická znalost (rys) jedince, určující jeho vlastnosti, chování a dosažený výsledek při řešení zpracovávaného problému.

Chromozom- též genom, soubor genů, které unikátně identifikují organismus a definují jeho vlastnosti.

Potomek – jedinec vytvořený odvozením z jednoho nebo více jedinců populace. Potomek přejímá některé rysy svých rodičů, kombinuje je a doplňuje vlastními poznatky a zkušenostmi.

Heuristika – ^[6.GA.Heuristika] je postup, který nedává přesné řešení daného problému, ani nezaručuje nalezení tohoto řešení v krátkém čase. Ve většině případech dává dostatečně přesné řešení rychle, ale obecně takové tvrzení nelze dokázat. Heuristická metoda není metodou přísného důkazu. Obecná teorie heuristické metody dosud neexistuje. Její rozpracování je společným úkolem pro různé vědecké disciplíny, jako je např. dialektická

logika, matematika, psychologie atd. Heuristický algoritmus obvykle v relativně krátké době nalezne dostatečně přesné řešení, neexistují ale žádné záruky, že to tak musí být vždy. Obvykle lze připravit takovou množinu vstupních údajů, se kterými se algoritmus nedokáže buď vůbec poradit, nebo se čas nutný k výpočtu prudce zvýší, nebo jsou výsledky zcela nepoužitelné. V praktickém životě je ale výskyt takových vstupních údajů téměř vyloučen, proto se heuristické algoritmy používají pro řešení úloh velmi často.

Mutace jedince – náhodná změna v genu jedince, pomáhá evoluci populace.

Kriteriální funkce – funkce, která vyjadřuje schopnost chromozomu řešit zadanou úlohu. Též fitness funkce.

Lokální extrém – [7.GA.Fce] Lokální extrém je bod, ve kterém je funkční hodnota kriteriální funkce vyšší (lokální maximum) či nižší (lokální minimum) než funkční hodnota v libovolném bodě nějakého okolí tohoto bodu. Pokud je funkční hodnota v tomto bodě menší, resp. větší než funkční hodnota v libovolném jiném bodě celého definičního oboru (množiny všech možných výsledků), označuje se tento bod jako globální minimum, respektive maximum.

Výhody genetického algoritmu

Genetické algoritmy patří do skupiny algoritmů umělé inteligence. Jedním z klíčových cílů využití informatiky je, aby počítač uměl vyřešit úlohu takovým způsobem, aniž bychom podrobně popsali postup řešení.

Genetický algoritmus vytváří obecný postup automatického nalezení řešení úloh se složitou abstrakcí, u kterých neumíme důsledně popsat postup řešení úlohy, ale víme, čeho chceme dosáhnout, umíme popsat podmínky a pravidla při hledání řešení a můžeme mezi sebou porovnávat jednotlivé dosažené výsledky.

Genetické algoritmy nám dále umožňují efektivně procházet stavový prostor možných řešení pro kombinatorické problémy, které nelze pro svůj rozsah technicky řešit analytickými postupy.

Nevýhody genetických algoritmů

Hlavní nevýhodou genetického algoritmu je nemožnost přesného ohodnocení dosaženého výsledku. U problémů s možnou grafickou reprezentací, jako hledání cesty v grafu, můžeme ze zobrazeného grafického výsledku logicky usoudit, zda dosažené řešení odpovídá představám o výsledku. U problému ryze abstraktního charakteru nevíme, jestli je výsledek již dobrým řešením, nebo zcela ještě nevhodným výstupem. Výpočet se může zastavit v lokálním extrému a nedosáhnout tak námi požadované kvality výstupu. Tato neurčitost dosaženého výsledku ovlivňuje množinu úloh, pro kterou lze genetické algoritmy použít.

Genetické algoritmy sice urychlují výpočty složitých úloh, při práci s velkou populací a mnoha cykly tvorby generací je ale genetický algoritmus také náročný na výpočetní sílu použitého počítačového vybavení.

Problém obchodního cestujícího

Principy genetického algoritmu budou uvedeny na konkrétním příkladu řešení úlohy obchodního cestujícího (Traveling Salesman Problem – TSP).

Obchodní cestující je jedinec, který musí objet určité množství měst, různě od sebe geograficky vzdálených, a vrátit se do svého počátečního města. Úspěšnost obchodního cestujícího se dá měřit jako doba, za kterou všechna města navštíví. Aby doba průjezdu všemi městy byla co nejmenší, musí obchodní cestující projet městy v takovém pořadí, aby celková ujetá vzdálenost mezi městy byla co nejmenší, tedy cesta co nejkratší nejrychlejší. Pořadí měst, kterými cestující projíždí, můžeme chápat jako jeho individuální znalost (genom), která nám umožňuje ohodnotit jedince – lepší je ten jedinec, který zná takové pořadí mezi městy, které ústí v kratší cestu, než znají ostatní jedinci populace.

Tato principiálně jednoduchá úloha se v tradičním analytickém přístupu nalezení výsledku nejkratší možné cesty dá přepsat do soustavy rovnic, jejichž množství roste exponenciálně pro počet jednotlivých měst. Pro takové analytické řešení je pro velké množství měst potřeba nepřiměřeného výpočetního výkonu.

Problém rychlého a kvalitního nalezení optimálního řešení pro TSP je předmětem obecného zájmu komunity lidí zabývajících se rozvojem a optimalizací algoritmů, TSP se pro své jednoduché zadání problému používá i jako ukázkový příklad využití metod umělé inteligence. Na internetu jsou problému věnovány rozsáhlé zdroje, například ^[9.GA.TSP] uvádí kompletní dokumentaci k pozadí řešení problému včetně použitých algoritmů, externích odkazů a ukázek řešení, například propojení 25 000 reálných švédských měst.

Při řešení problému TSP genetickým algoritmem představuje jedince populace objekt s genetickou informací (genem, znalostí), která určuje pořadí měst.

Populace obsahuje z počátku jedince s náhodně inicializovanou znalostí pořadí procházení měst.

Tito jedinci jsou vyhodnoceni podle své znalosti, cestující s nejhorší trasou mezi městy jsou z populace vyloučeni, ta je doplněna potomky vybraných jedinců s nejlepšími vlastnostmi. Jako celkový výsledek genetického algoritmu se bere nejkratší vzdálenost, kterou dosáhla současná populace jedinců.

Výpočet populace bývá obvykle kombinován heuristickými metodami, komplexní popis a implementace takového řešení je obsažen ve zdroji ^[10.GA.TSP].

Cyklus evoluce se opakuje do té doby, dokud algoritmus nedosáhne optimálního výsledku podle předem zadaných kritérií, nebo po určitý počet generací již nezlepšuje populace svůj dosažený výsledek.

Hlavní problémy genetického algoritmu

Definování genomu

Prvním problémem při řešení úlohy z pohledu genetických algoritmů je určit hledanou charakteristiku problému a umět ji **zakódovat do datové struktury, která umožňuje vytvořit genom jedince genetické populace**. V případě řešení problému obchodního cestujícího je genomem pořadí měst, kterými cestující projíždí. Jak ale definovat gen například pro šachovou hru?

Definování kritériální funkce

Pro vyhodnocení úspěšnosti jedince v populaci potřebujeme umět ocenit jeho chromozom absolutní hodnotou výsledku nebo alespoň umět porovnat dva jedince populace a určit, který z nich představuje lepší řešení problému.

Vyhodnocovací funkce je použita k rozřídění populace podle úspěšnosti jedinců, úspěšní jedinci jsou preferováni v tvorbě potomků.

U problému obchodního cestujícího může být ohodnocovací funkcí celková vzdálenost dosažená navštívením všech měst v daném pořadí. Ohodnocující kritériální funkce může mít ale i složitější podobu, například kvůli průběhu vývoje populace můžeme zvolit funkci sčítající druhé mocniny vzdáleností mezi městy a snažit se tak potlačit případy, v kterých většina měst je propojena krátkou cestou, ale existují v nich ojedinělé dlouhé spojnice.

Pro komplexní úlohy může ohodnocující funkce obsahovat komplexní logiku porovnávání dvou jedinců populace, aniž by uměla číselně vyjádřit kvalitu jednotlivého objektu.

Výběr vhodného genetického algoritmu

Genetický algoritmus upravuje chování populace, její růst, třídění jedinců populace a náhodné mutování genů.

Dále může implementovat další charakteristiky inspirované biologickou společností, jako stárnutí populace se změnou schopnosti reprodukce, sexuální chování objektů, či pro zmenšení rizik lokálních extrémů může implementovat existenci oddělených populací a postupnou migraci a globalizaci v jednu výslednou populaci.

Genetický algoritmus také řeší chování celkového programu z hlediska správy objektů populace, rychlosti algoritmu, paměťových nároků, případné paralelní zpracování úlohy.

Vhodně nastavené parametry

Velmi citlivým problémem jsou správně nastavené výchozí parametry výpočtového systému, jako velikost populace, intenzita náhodných mutací.

Čistě genetický algoritmus bude pravděpodobně potřebovat větší základnu populace, než algoritmus kombinovaný s heuristickými znalostmi.

Inicializace výchozí populace

Oproti přírodě, kde dle Darwinovy teorie vše se vyvinulo od prapůvodní základní buňky, vytváříme systém, který již začíná pracovat na definovaném základě a s existující populací.

Na správně inicializované populaci, například dostatečně náhodné, závisí to, zda se výpočet od začátku nebude ubírat špatným směrem do blízkého lokálního extrému.

Algoritmus tvorby potomků

Nejsložitější otázkou úlohy řešené genetickým algoritmem je algoritmus, jak tvořit z jednoho či více rodičů potomky tak, aby se v genu potomka objevily dobré charakteristiky obou jeho rodičů. Volba správného dědičného algoritmu určuje, jak rychle a jakým „správným“ směrem se bude populace měnit.

Nejjednodušší tvorba potomků je odvozením a náhodnou změnou genu od jednoho rodiče. Taková dědičnost je ale založena jen na náhodných změnách, neprojevuje se v ní přenos a kombinace charakteristik rodičů, výsledky použití nejsou příliš dobré, ale implementace je jednoduchá a přímočará.

Biologický svět zná jednoho nebo dva rodiče. Teoreticky bychom mohli přemýšlet i o více rodičích jednoho nového jedince, problém ale je, jak zkombinovat dva či více genů do genu nového, aby smysl a funkčnost genu zůstala zachována a zároveň se do genu promítly, pokud možno nejlepší charakteristiky předků. Tímto problémem se zabývá například práce na odkazu ^[8.GA.Ex]

Implementace náhodných změn

Náhoda a náhodné procesy jsou všudypřítomné, ovlivňují vývoj společnosti (populace) kladným i negativním způsobem. Biologické náhodné procesy jsou představovány například fyzikálními jevy prostředí, jeho radiací, nemocemi, genetickou vzdáleností či blízkostí rodičů, nehodami, objevy atd..

Náhodné procesy pomáhají překonávat lokální extrém, mohou pomoci vychýlit rovnováhu výpočtu do neobvyklých poloh, které mohou vést k výhodnějším řešením.

Příliš radikální změny nebo jejich velká intenzita by ale naučný genetický proces mohly posunout do roviny náhody a potlačit tak význam učení a optimalizace již dosaženého výsledku.

Využití heuristiky

Použití heuristických metod, které optimalizují průběh výpočtu na základě znalosti jeho charakteristiky, mohou radikálně zlepšit dosažený čas i výsledek.

Použitím heuristických znalostí vytváříme částečně hybridní systém spojení genetického a analytického programování a ztrácíme tím obecnost průběhu genetického řešení, do kterého již svou analytickou „náповědou“ zasahujeme.

Výkonný hardware

Genetické výpočty založené na velké populační bázi potřebují jak větší množství operační paměti, tak (paralelní) výpočetní sílu, která provádí genetické operace nad jednotlivými generacemi populace, která může mít miliony jedinců.

Vhodným výkonným prostředkem zpracování genetických úloh je paralelní zpracování včetně rozdělení práce a zpracování na síťově distribuovaných clusterech počítačů.

V případě zdroje ^[9.GA.TSP] bylo pro výpočet problému obchodního cestujícího mezi 25 000 městy použito 96 dvouprocesorových pracovních stanic.

Zajímavou podporu pro vývoj a testování genetických algoritmů může nabídnout především univerzitní prostředí.

Hardware

Zatímco klasické počítače jsou pořád ještě postaveny na schématech, které vznikly na začátku druhé poloviny dvacátého století, budoucnost přinese převratné změny týkající se celkové koncepce výpočetních jednotek.

Pro genetické programování již dnes existují hardwarová řešení, která svou konstrukcí umožňují vývojové změny ve vlastních obvodech, samoučení a zdokonalování. Více například v dokumentech konference o evolučním hardwaru, pořádané Nasa ^[18.GA.HW].

Příklady nasazení

Základní prvky algoritmů umělé inteligence jsou dnes již obsaženy v mnoha praktických a využívaných úlohách. V následující sekci uvádím odkazy a krátký popis na praktické možnosti využití genetických algoritmů v teoretické i praktické formě v různých kategoriích.

Teoretické problémy s praktickými aplikacemi

- **Traveling Salesman Problem**

Hledání nejkratší cesty pro navštívení všech prvků množiny bodů.

- **Problém batohu**

Jak co nejefektivněji využít omezený prostor batohu. Z různě oceněných věcí máme vybrat do batohu takovou množinu, aby cena věcí v batohu byla maximální.

- **Problém N královen**

Úkolem je rozmístit na šachovnici o rozměrech $N \times N$ polí N královen tak, aby se navzájem neohrožovaly.

Optimalizace a plánování

Navrhování rozvrhů

Zkombinování a vyhledání nejlepšího rozvrhu dle kritérií výuky, tříd a učitelů. Obdobně vytvoření pracovního rozvrhu pro stroj dle podmínek daných jednotlivými úkoly, dostupností materiálu a pracovníků, prioritami úkolů.

Rozvržení materiálu na užitečnou plochu^[GE.7]

Pro vyřezávací či vysekávací stroje rozmístit součástky na plochu způsobem, aby byl z výroby co nejmenší odpad materiálu. Úloha počítá s parametrem tloušťky řezu a uchycení desky pracovními nástroji.

Návrh organizace počítačové sítě

Reorganizace v pružném výrobním systému^[11.GA.Reconfiguring]

Genetický systém pružně reagující na změnu požadavků trhu a možností výrobní firmy. Systém vytváří rozvrhy práce a výroby pro dynamicky měnící se pracovní scénáře.

Elektromagnetická optimalizace pro návrh antén^[13.GA.Antena]

Biologie

Učící se genetická aplikace pro predikci bílkovinné struktury^[14.GA.Protein]

Matematika

Maximalizace výstupu funkce o více proměnných^[15.GA.Maximum]

Obecný genetický algoritmus, který kombinuje, analyzuje a upravuje vstupní hodnoty funkce tak, aby bylo dosaženo maximální hodnoty funkce. Tato optimalizace hodnot má sama o sobě řadu dalších využití v různorodých systémech.

Design

Evoluční výpočty pomocí genetických algoritmů^[16.GA.Design]

Výpočty pohybu, tvaru, grafický design a obrazy tvořené genetickými algoritmy.

Genetic Algorithms and Engineering Design^[17.GA..Design2]

Finance

Návrh a optimalizace akciového portfolia

Při návrhu akciového portfolia se vychází z různých kritérií volby jistoty, rizika a požadovaného výnosu.

V.Existující využití genetických algoritmů v logistice

Z rešerše existujících prací zabývajících se logistikou ve spojitosti s genetickými algoritmy je vidět úspěšná aplikace technologie na logistickou problematiku.

Uvádím výběr prací s odkazy a přeloženou anotací práce.

Aplikace genetického algoritmu pro návrh evropské logistické sítě automobilového průmyslu^[19.GA.Logistic]

Práce se zabývá Evropskou dopravní legislativou s ohledem na požadavky obnovy dosluhujících vozidel a představuje komplexní model dopravního řetězce zachycující velké množství kritérií. Dále jsou sepsány požadavky pro optimalizaci modelu, navrženo a obhajováno řešení ideální dopravní sítě.

Genetický algoritmus s fuzzy rozhodováním určený k optimalizaci priorit logistických systémů^[20.GA.Logistic2]

Dokument popisuje problém optimalizace rozvrhu logistických činností s použitím genetického algoritmu a funkcí s fuzzy rozhodováním, jehož úkolem je organizace činností v takovém sledu, aby bylo možno k určitém čase vyřídit více objednávek se současným ohledem na jejich priority. Výsledky simulace ukazují, že navrhovaný genetický algoritmus je vhodný pro řešení tohoto typu problémů.

Hybridní algoritmus pro řešení distribučního problému^[21.GA.Logistic3]

Hybridní genetický algoritmus pro řešení distribučního problému skandinávských producentů dřevin navrhuje rozvrhy dodávek a cest lodí pro snížení distribučních nákladů. Řešení je rozděleno na dvě úrovně, jedna optimalizuje dopravní toky pro daný rozvrh dodávek, druhá optimalizuje samotný rozvrh dodávek a vybírá ten nejvhodnější. Simulace je postavena na reálných datech a porovnává výsledek výpočtů s faktickými náklady.

Organizace různorodého nákladu na palety s využitím genetického prohledávání^[22.GA.Logistic4]

Organizace různorodého nákladu na palety vyžaduje ve většině případů lidský dohled a rozhodování. Systém umožňuje s dostatečnou rychlostí vytvořit optimální plán a automatizovat proces pro nakládací vozidla. Aplikace je úspěšně nasazena v existujícím velkokapacitním automatickém skladu.

Výběr cesty ve veřejné dopravě s využitím genetického algoritmu^[23.GA.Logistic5]

Většina dopravně zaměřených aplikací je určena pro komerční soukromý trh vozidel. Existující aplikace zabývající se veřejnou dopravou naráží na problém, kdy lidé, kteří používají veřejnou dopravu, mohou změnit cestu nejen ve smyslu stejného dopravního prostředku, ale například zaměnit autobus za podzemní dráhu. Systém by s touto možností měl počítat a zahrnout aspekty přestupu mezi jednotlivými způsoby dopravy. Tato studie navrhuje rámec pro stavbu veřejných dopravních systémů, které vytváří optimální dopravní cesty v prostředí více dopravních možností.

VI.Zadání aplikace

Aplikace je realizována od své funkční kostry směrem k detailům a rozšiřování uživatelského rozhraní, práce s vytvořenými dokumenty, tiskem a reportováním.

Hlavní smysl aplikace je zautomatizovat a zrychlit manuální neodbornou mechanickou práci. Aplikace bude umět následující činnosti.

Popis kontejneru

- Kontejnery, s kterými se v aplikaci pracuje, mají následující vlastnosti: název kontejneru, identifikační číslo systému SAP, rozměry, hmotnost.
- Používají se pouze dva rozměry nákladu, výška je ignorována, kontejnery jsou připraveny tak, aby se vešly do nákladního prostoru kamionu, nedá se na ně do další vrstvy nic naložit kvůli rozměrům nebo jejich zaoblenému vrchnímu tvaru, případně jsou známe jednotlivé výjimky kontejnerů, které lze kombinovat. Kontejner lze otáčet o 90°.

Pracovní plocha

- Pracovní plocha představuje místo pro umístění kontejnerů, které jsou k dispozici ve skladu, a jejich organizaci do nákladního prostoru. Schematicky je zde zobrazen půdorys návěsu kamionu, na plochu lze pokládat kontejnery z knihovny a pracovat s nimi podobně jako s nábytkem v používané aplikaci na návrh bytu. Kontejnery lze po ploše přesouvat, jde vybrat více kontejnerů, mazat je z plochy, manuálně ukládat a organizovat do znázorněného návěsu.
- Přesouvání kontejnerů neumožňuje překrývání objektů – jsou respektovány vzájemné hrany kontejnerů a návěsu.
- Kontejnery a návěs jsou zobrazeny v reálném poměru.
- Na kontejneru na pracovní ploše se zobrazuje jen jeho název, jsou k dispozici informace o detailech jednotlivých kontejnerů.

- Pracovní plocha umožňuje zobrazit seznam kontejnerů ve skladu (tzn. na ploše), dále aktuální stav využití návěsu, tj. využitá ložná plocha návěsu, celkové zatížení a hmotnostní poměry ve vertikálním a horizontálním směru.
- Rozměry a celkovou nosnost návěsu lze konfigurovat.
- Na pracovní ploše lze provést automatický výpočet optimalizace kontejnerů. Výpočet optimálně zorganizuje disponibilní kontejnery do nákladového prostoru, je možné vybrat kontejnery, které v nákladu musí být, lze respektovat kontejnery předem vložené na návěs.
- Na pracovní ploše budou zobrazeny informace o dokumentu: datum vytvoření dokumentu, jeho název a popis, autor dokumentu.

Kriteria výpočtu

- Při výpočtu nesmí být překročena maximální nosnost návěsu, sleduje se optimální rovnoměrné zatížení pravé a levé, přední a zadní strany návěsu.
- V případě, že kamion nelze zcela zaplnit disponibilními kontejnery, či v případě, že kontejnery by v každé kombinaci při plném zaplnění návěsu přesahovaly jeho nosnost, výpočet navrhne řešení s neúplně využitou kapacitou nákladové plochy.

Dokumenty

- Je možno uložit a nahrát dokument pracovní plochy.
- Formáty ukládaných dokumentů budou v přenositelné podobě, ve formě komprimovaných XML souborů.

Tisk

- Tisk dokumentu umožní vytisknout současný stav pracovní plochy, zobrazí statistiky využití ložné plochy návěsu, seznam naložených kontejnerů včetně pozice a SAP identifikací, seznam všech dostupných kontejnerů ve skladu. Tisková funkcionalita bude přidána do budoucích verzích programu.

VII.Uživatelské rozhraní aplikace

V uživatelském rozhraní je kladen důraz na jeho nekomplikovanost, jednoduchost a intuitivnost používání.

Multiplatformní aplikace

Aplikace je vytvořena ve formě multiplatformního grafického programu, který je nezávislý na uživatelském operačním systému. K chodu aplikace není třeba připojení na internet. Program není rozdělen na grafickou a výkonnou funkčnost, která by se dala volat odděleně, například z příkazového řádku. K chodu programu je zapotřebí nainstalované prostředí Javy.

Spuštění

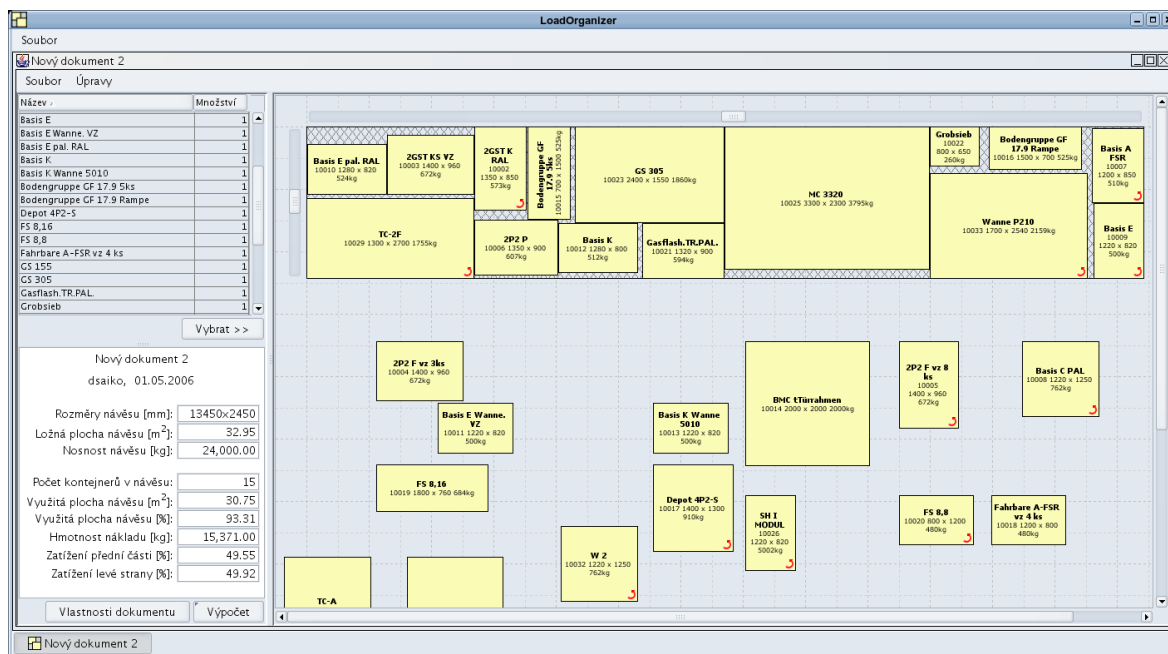
Program nepotřebuje instalaci, disk využívá pouze k ukládání a nahrávání knihovny a dokumentů.

Program lze spustit z dodaného programového JAR archivu **load-organizer.jar**, později bude přístupný na internetu ke spuštění přes technologii Java WebStart (doména www.saiko.cz není nyní z technických důvodů funkční). Spuštění z internetu obstará automatickou aktualizaci programu, případně automatické doinstalování chybějících Java knihoven.

K běhu programu je zapotřebí prostředí Java verze 1.5.

Vzhled aplikace

Program běží v grafickém okně, do kterého je možno otevřít více dokumentů. Při startu aplikace lze volit z nativního - systémového vzhledu, nebo ze vzhledu prostředí Java knihoven.



Ilustrace 5: Snímek obrazovky aplikace LoadOrganizer

Okno aplikace je rozděleno na dvě části: informační část s panelem knihovny a panelem statistických informací o aktuálním nákladu kontejnerů a pracovní plochu s kontejnery, na které je znázorněn prostor návěsu a posuvné lišty, graficky znázorňující rozložení hmotnosti návěsu.

Z knihovny jsou přidávány kontejnery na pracovní plochu, kde jdou posouvat myší, vybírat do skupin, umisťovat do návěsu, mazat z plochy. Program automaticky hlídá hrany kontejnerů a návěsu, objekty se nepřekrývají, kontejner může být uvnitř návěsu, nesmí však protínat jeho hranu. Po každém pohybu kontejneru jsou přepočítány hodnoty informačního panelu.

Kontejner může být z označen za povinný, pokud je v jeho okolí dostatek místa, lze jej otočit.

Z nabídky v hlavním menu lze nastavit vlastnosti prostředí a výpočtu, z nabídky v menu dokumentu lze nastavit vlastnosti dokumentu a spustit výpočet.

Hlavní menu aplikace

- **Soubor**

- **Nový**

- Vytvoří nový dokument, při vytváření dokumentu lze specifikovat vlastnosti dokumentu (název a autor dokumentu, velikost návěsu ...)

- **Otevřít**

- Otevře uložený dokument

- **Nastavení**

- Nastavení prostředí (barvy, formát čísel) a výpočtu.

- **Konec**

- Ukončení programu

Menu otevřeného dokumentu

- **Soubor**

- **Uložit**

- Uloží otevřený dokument do souboru.

- **Vlastnosti dokumentu**

- Zobrazí vlastnosti dokumentu (název a autor dokumentu, velikost návěsu), velikost návěsu však lze měnit jen při otvírání nového dokumentu.

- **Zavřít**

- Zavře otevřený dokument.

- **Úpravy**

- **Povinný kontejner**

- Označí vybrané kontejnery jako povinné. Povinný kontejner má ve svém grafickém zobrazení v levém horním rohu červený vykřičník.

- **Otočit**

- Otočí vybrané kontejnery, menu je aktivní pouze, když je v okolí kontejneru prostor pro jeho otočení. Otočený kontejner má v ve svém grafickém zobrazení pravé dolní části červený symbol rotace.

- **Smazat**

- Smazání vybraných kontejnerů z plochy.

- **Vybrat vše**

- Vybere všechny kontejnery na pracovní ploše.

VIII. Implementace genetického algoritmu

Aplikace je rozdělena na grafickou část a logiku genetického algoritmu.

Chromozom

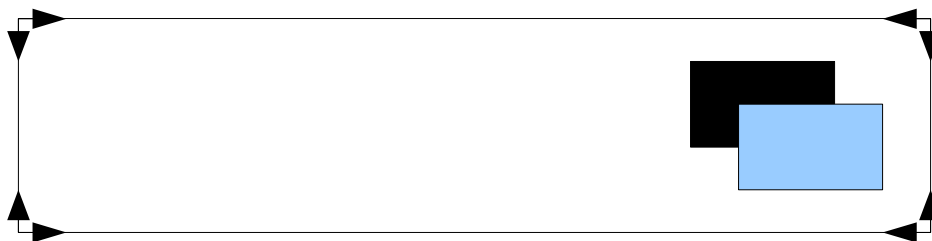
Populace genetického algoritmu je tvořena generací chromozomů, které obsahují seznam kontejnerů s pozicí jejich umístěním v návěsu.

Chromozom automaticky počítá své statistiky, kterými jsou počet umístěných kontejnerů, celková využitá plocha návěsu, rozložení hmotností. Ze zjištěných statistik vrací hodnotu své kritériální funkce.

Umístění kontejneru

Umístění kontejneru do návěsu znamená přidání validního záznamu pozice kontejneru do existujícího chromozomu.

Při přidávání kontejneru do návěsu je vybrán náhodný směr postupu hledání volného místa pro pozici kontejneru. Je možné začít z každého rohu návěsu a začít postupovat buď ve směru osy x nebo y.



Ilustrace 6: Osm možností směru postupu při hledání volného místa v návěsu

V případě, že při pokusu vložit do chromozomu na konkrétní pozici nový kontejner je již pozice obsazena existujícím objektem, jsou vráceny obdélníkové hranice tohoto kolizního objektu.

V inicializovaném směru hledání volného místa se nový objekt posune přes známé rozměry kolizního objektu a testuje chromozom na nové volné místo.

V implementaci algoritmu se do kontejneru přidávají objekty tak, že se náhodně vybírají z množiny dostupných kontejnerů a náhodně se jim nastavuje příznak otočení kontejneru. Nejprve se ale do chromozomu rozmístí všechny kontejnery, v aplikaci označené jako povinné.

Vytvoření počáteční generace

Počáteční generace chromozomů se vytváří na zcela náhodné bázi s tím, že do návěsu jsou nejprve umístěny kontejnery povinné.

Při inicializaci náhodné populace výše popsaným algoritmem umístování kontejnerů možno získat využití plochy přes 85%, což je samo o sobě lepší než firmou Denios udávaná současná hodnota využití plochy 80%.

Genetický algoritmus

Princip genetického algoritmu je obsažen v následující ukázce zdrojového kódu. Existující generace je seříděna podle kvality jednotlivých chromozomů a je zjištěna hodnota kritériální funkce nejlepšího jedince populace. Pokud se již definovaný počet generací tato hodnota nezměnila, je algoritmus ukončen, jinak je vytvořena nová populace chromozomů.

```
while(true) {
    //sort population and get the best chromosome
    sortPopulation(population);
    Chromosome cl=population.get(0);

    double bestResult=cl.getUsedArea();
    double fitness=cl.getFitnessValue();

    boolean stable=true;

    if(fitness!=prevFitness) {
        stable=false;
        prevFitness=fitness;
    }

    if(!stable) { stableSituationCount=0; } else { stableSituationCount++; }

    sysout(progressMessage());

    if(stableSituationCount>=preferences.getEngineFinishStableCount()) break;

    //create new population
    createNewGeneration(generation,population);
    generation++;

    //check for user stop request
    if(stopRequests.size()>0) break;
}
```

Vytvoření nové generace probíhá vytvořením mutací a potomků z generace současné a jejich přidáním do populace. Populace je seříděna podle hodnoty kritériální funkce a počet jedinců je snížen na požadovanou hodnotu velikosti populace, nejméně silní jedinci jsou z populace vypuštěni.

```
private void createNewGeneration(int generation, List<Chromosome> population)
throws Exception
{
    //mate some of the individuals
    ...
    //optimize some of the individuals
    ...
    //randomize some of the individuals
    ...
    //if the population has grown up, cut the losers to remain on the stable
    // population size
    while(population.size()>populationSize) {
        population.remove(population.size()-1);
    }
}
```

Kriteriální funkce

Jednoduchou, ale kouzelnou formulí genetického algoritmu je kriteriální funkce ohodnocení kvality chromozomu (fitness value). Populace chromozomů se vyvíjí na základě této funkce, její změnou lze dosahovat zajímavých modifikací vývoje populace.

Výsledná kriteriální funkce upřednostňuje co nejvíce využitou přepravní plochu, se současným ohledem na rozložení distribuce hmotnosti co nejvíce na přední část návěsu a zároveň s rovnoměrně rozloženou hmotností mezi pravou a levou stranu návěsu. Vzhledem k měřítkům jednotlivých veličin (plocha je v jednotkách milionů milimetrech čtverečních, rozložení hmotnosti v procentech) se vyhodnocovací algoritmus chová tak, že upřednostňuje využití plochy návěsu, při stejném využití plochy (při stejné množině kontejnerů v návěsu) dává přednost lépe rozložené hmotnosti.

```
public double getFitnessValue() {  
    if(totalWeight==0) return 0;  
    return usedArea+(weightFront/totalWeight)*100+  
        (1-Math.abs(0.5-weightLeft/totalWeight))*100;  
}
```

Změna kriteriální funkce je zajímavá svými efekty na celkový vývoj populace a bude demonstrována následujícími snímky výsledků výpočtu dosažených na stejné množině kontejnerů.

Výsledná forma kriteriální funkce, optimální kombinace všech sledovaných hodnot, náklad je umístěn do přední části kontejneru a hmotnosti jsou stranově distribuovány

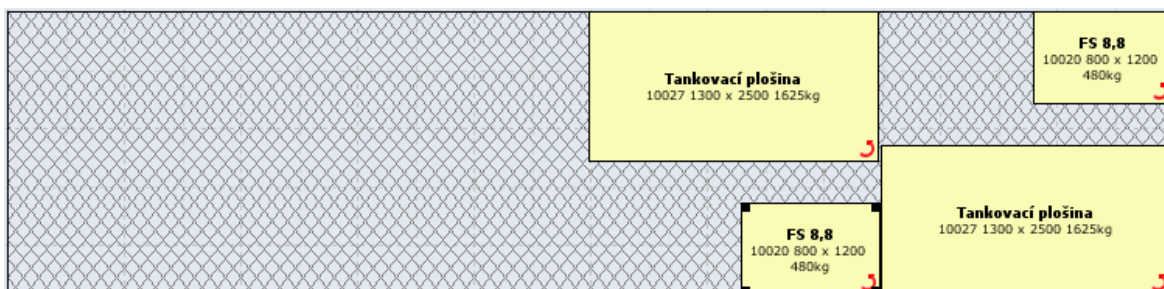
```
fitnessValue =  
    usedArea+(weightFront/totalWeight)*100+  
    (1-Math.abs(0.5-weightLeft/totalWeight))*100;
```



Ilustrace 7: Působení výsledné kriteriální funkce

kriteriální funkce pro zatížení zadní části návěsu:

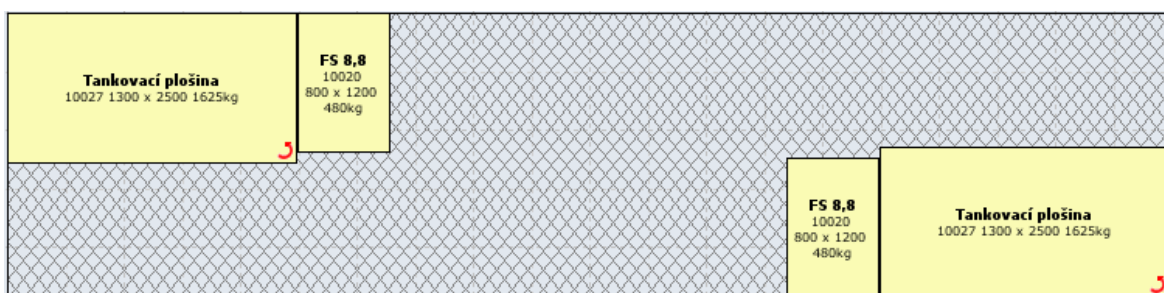
```
fitnessValue =
    usedArea+(1-weightFront/totalWeight)*100+
    (1-Math.abs(0.5-weightLeft/totalWeight))*100;
```



Ilustrace 8: Působení kriteriální funkce pro zatížení zadní části návěsu

kriteriální funkce pro rozložení hmotnosti jak na strany, tak podélně:

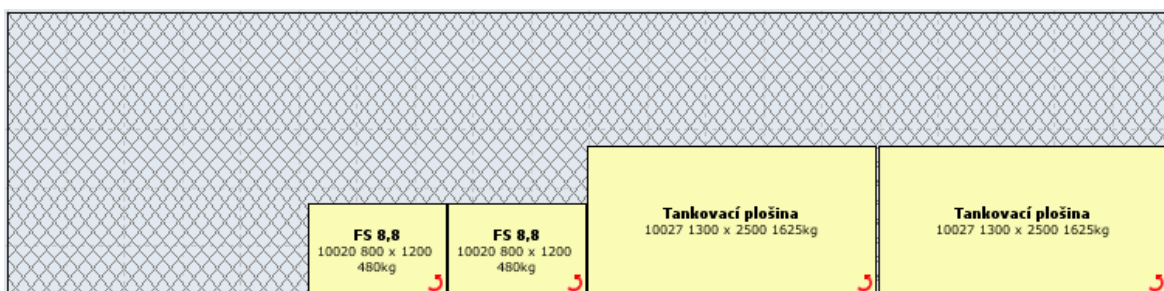
```
fitnessValue =
    usedArea+(1-Math.abs(0.5-weightFront/totalWeight))*100+
    (1-Math.abs(0.5-weightLeft/totalWeight))*100;
```



Ilustrace 9: Obměna kriteriální funkce pro rozložení hmotnosti

Kriteriální funkce pro zatížení levé části návěsu:

```
fitnessValue = weightLeft;
```



Ilustrace 10: Obměna kriteriální funkce pro zatížení levé části návěsu

Optimalizace

Kromě svých základních charakteristik si chromozom udržuje informaci o využití podélní a svislé první a druhé poloviny kontejneru.

Optimalizace chromozomu spočívá ve vybrání jeho nejlepší části a doplnění zbytku buď původními kontejnery v jiném pořadí, novými kontejnery, nebo při vytváření potomků od dvou rodičů doplněním kontejnerů z lepší části druhého rodiče. Optimalizace je kombinována s náhodnými změnami, při nichž je náhodný počet kontejnerů z chromozomu odebrán a chromozom doplněn novou množinou náhodných objektů.

Parametry výpočtu

Základní parametry výpočtu jsou:

- Velikost počáteční populace kombinovaná s procentuálním růstem populace.
Algoritmus dává pro reálné zadání úlohy dobré výsledky již při pěti chromozomech v základní populaci.
- Postačující stabilita výsledku
- Hloubka optimalizace
Při optimalizaci je v jednom cyklu možno optimalizovat i již nově získané optimalizované chromozomy (při hloubce optimalizace větší než 1), dochází k okamžitému pokusu o optimalizaci optimalizovaných výsledků, operace je časově náročná.
- Posun kontejneru
Posun kontejneru pro testování volného místa na návěsu. Jak bylo popsáno u rozmíst'ování kontejnerů, posun po hlavní směrové ose se řídí hranicemi kolizních objektů a nepřičítá se zde žádná konstanta. Při posunu po vedlejší ose se ale využívá pevná délka kroku. Tato hodnota je v milimetrech, rozměry kontejnerů jsou udávány s přesností na deset milimetrů. Posun kontejneru krokem o velikosti 15 cm je stále velice dostačující, v hlavním směru posouvání objektů se tato mezera neprojeví – objekty jsou řazeny přímo těsně ke svým vzájemným hranám.

- Počet cyklů při generování náhodnosti

Při generování náhodné počáteční populace a při doplňování chromozomů o nové náhodné objekty je dostupná množina kontejnerů procházena v určitém počtu cyklů, který lze nastavit.

Nastavení

Nastavení výpočtu

Velikost počáteční populace: 10

Růst populace [%]: 0

Dostačující stabilita výsledku: 15

Hloubka optimalizace: 1

Posun kontejneru [mm]: 150

Počet cyklů při generování náhodnosti: 2

Nastavení grafického prostředí

Nový dokumentu při startu aplikace: ☒

Automatická maximalizace nového okna: ☒

Bodů na metr: 75

Formát celých čísel: ###0

Formát desetinných čísel: ###0.00

Formát datumu: dd.MM.yyyy

Barva kontejneru: #fabbb4

Barva popisu kontejneru: #000000

OK Storno

Ilustrace 11: Dialog nastavení aplikace

Výsledky výpočtů

Převzetí reálných dat definice objektů, které jsou k dispozici pro vyslání kamionu, je uloženo do knihovny objektů, která je přiložena k programu.

Při tomto konkrétním případě odvozu zboží bylo v reálu dosaženo využití ložné plochy přibližně 80%. Při dodržení reálných rozměrů návěsu a požadovaných kritérií vrací algoritmus stabilně a v rozumně krátké době výsledky převyšující hodnotu 90% využití ložné plochy. Tato úspora znamená ušetření jednoho kamionu na deset skutečně vyslaných v současné době.

Při porovnání výsledků se zadáním problému, kdy jsou vyslány následující kamiony,

- do Německa 2 až 3 kamiony týdně (10 měsíčně), cena cesty 26 000 Kč
- do Francie 1 kamion týdně (4 měsíčně), cena cesty 37 500 Kč
- do Itálie 1 kamion měsíčně, cena cesty 30 500 Kč

a při organizační práci trvající přibližně 3 hodiny na jeden kamion (s hodinovou sazbou 450 Kč) jsou roční náklady na odeslání zboží v průměru rovny

$$((10 * 26000 + 4 * 37500 + 30500) + 15 * 3 * 450) * 12 = 5\,529\,000$$

Při předpokládaném ušetření desetiny potřebných kamionů a při zrychlení potřebné doby práce na jednu polovinu dostáváme výsledek nových ročních nákladů

$$((10 * 26000 + 4 * 37500 + 30500) * \frac{9}{10} + 15 * \frac{9}{10} * 3 * \frac{1}{2} * 450) * 12 = 4\,830\,300$$

s celkovou roční úsporou 698 700 Kč.

Tabulka výsledků

Následuje tabulka výsledků opakovaného spouštění úlohy na disponibilní množině jednoho kusu všech kontejnerů z knihovny, při standardních rozměrech návěsu 13450mm x 2450mm.

Časy výpočtu odpovídají spouštění programu na notebooku Compaq nx7010, Centrino 1.5Ghz, s operačním systémem Linux OpenSuse 10.0.

Algoritmus byl spuštěn s výchozími hodnotami svého nastavení, velikost populace je 10 jedinců.

	Čas výpočtu	Využití plochy	Počet generací	Hmotnost na přední část	Hmotnost na levou část
1	01:56 min	92.62 %	89	39.34 %	44.63 %
2	02:10 min	93.19 %	83	49.63 %	48.43 %
3	02:25 min	93.31 %	119	49.55 %	49.92 %
4	01:14 min	93.04 %	52	50.28 %	49.32 %
5	01:38 min	91.76 %	77	50.09 %	49.81 %

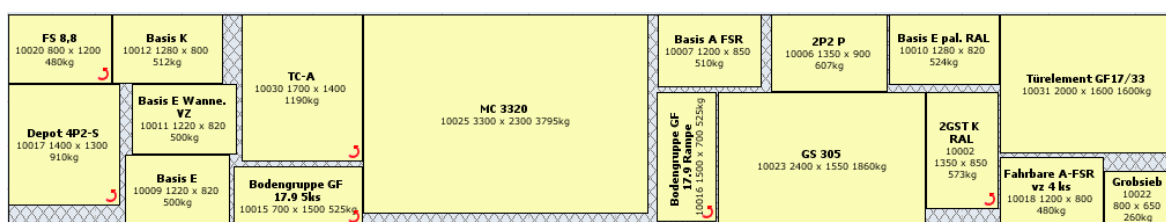
Ilustrace 12: Tabulka měřených výsledků výpočtu

Poznámka k zatížení přední a levé strany: pokud je hmotnost jednotky plochy kontejnerů přibližně stejná, je rozložení hmotnosti při souvislém zaplnění plochy vždy přibližně 50%.

Nejlepší celkový výsledek dosažený při testování programu byl přes 94% využití plochy návěsu, výsledek dosažitelný manuální organizací kontejnerů se pohybuje kolem 92%, ale za notně delší čas.

Ukázka výsledku 93.19% využití plochy návěsu:

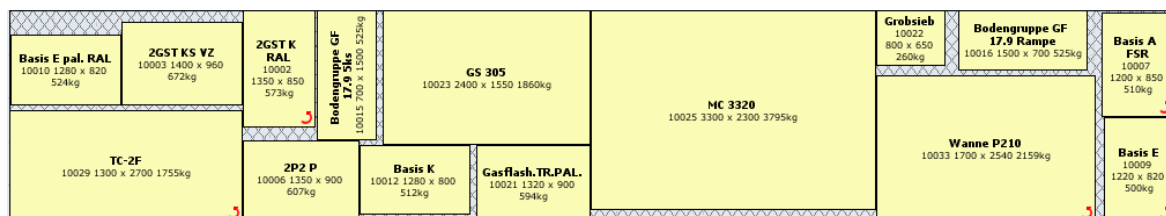
Nový dokument 2	
dsaiko, 01.05.2006	
Rozměry návěsu [mm]:	13450x2450
Ložná plocha návěsu [m ²]:	32.95
Nosnost návěsu [kg]:	24,000.00
Počet kontejnerů v návěsu:	17
Využitá plocha návěsu [m ²]:	30.71
Využitá plocha návěsu [%]:	93.19
Hmotnost nákladu [kg]:	15,351.00
Zatížení přední části [%]:	49.63
Zatížení levé strany [%]:	48.43



Ilustrace 13: Grafická podoba dosaženého výsledku využití plochy 93,19%

Ukázka výsledku 93.31% využití plochy návěsu:

Nový dokument 2	
dsaiko, 01.05.2006	
Rozměry návěsu [mm]:	13450x2450
Ložná plocha návěsu [m ²]:	32.95
Nosnost návěsu [kg]:	24,000.00
Počet kontejnerů v návěsu:	15
Využitá plocha návěsu [m ²]:	30.75
Využitá plocha návěsu [%]:	93.31
Hmotnost nákladu [kg]:	15,371.00
Zatížení přední části [%]:	49.55
Zatížení levé strany [%]:	49.92



Ilustrace 14: Grafická podoba dosaženého výsledku využití plochy 93,31%

Výhody programu oproti současnému řešení

Program umožňuje rychlejší a efektivnější práci s organizací kontejnerů. Výpočet urychluje organizaci kontejnerů a zlepšuje dosažitelné využití ložné plochy návěsu, čímž při dané velikosti transportu dochází k teoretické finanční úspoře 700.000 Kč za rok.

Program je ale použitelný i pro manuální úpravy nákladu kamionu, hlídá hrany návěsu i ostatních kontejnerů, automaticky jsou přepočítávány statistiky využití návěsu a rozložení hmotnosti.

IX. Další vybrané problémy implementace

Aplikace je implementována v jazyce Java s využitím knihoven Sun Java 1.5, na pracovní stanici s nainstalovaným operačním systémem Linux, ve vývojovém prostředí Eclipse IDE.

Grafické knihovny

Pro vytvoření aplikace v grafickém prostředí bylo možno vybírat mezi dvěma základními technologiemi Javy – Swing a SWT. Swing umožňuje více detailní práci s grafickými objekty, lze v něm naprogramovat jakoukoliv grafickou komponentu, program si vnitřně veškeré komponenty vykresluje. Oproti tomu SWT od firmy IBM umožňuje rychlejší vývoj i běh výsledné aplikace, je navázáno přímo na komponenty uživatelského operačního systému, neumožňuje mít takovou (grafickou) kontrolu nad jednotlivými ovládacími prvky aplikace jako Swing, ani možnost libovolné změny grafických komponent. SWT také není 100% nezávislé na platformě operačního systému, přinejmenším z grafického hlediska. I přes částečně komplexní překombinovanost knihoven Swing, hlavně z důvodu své dlouhodobé zkušenosti s touto technologií, je aplikace postavena právě na těchto knihovnách. Více informací o srovnání Swing a SWT lze nalézt např. na ^[23.Java.SwingSWT].

Databáze, formát souborů

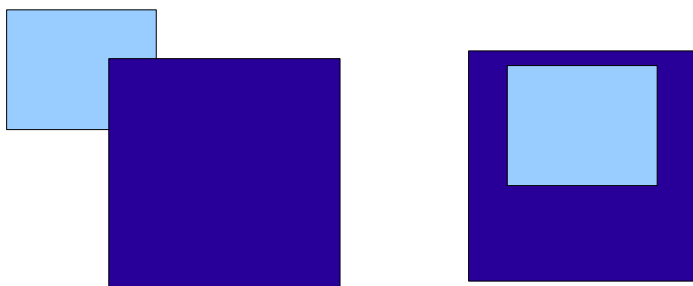
Při rozmyšlení o možnostech využití databáze se ukázalo, že funkčnost aplikace využívá úložiště dat minimálně, pouze ve formě jedné tabulky objektů knihovny. Také vzhledem k případné instalaci databáze a formátu datových souborů a jejich přenositelnosti či využitelnosti z jiných aplikací, je jako úložný formát pro knihovnu a dokumenty použit textový XML soubor ve formátování UTF-8, komprimovaný technologií GZip.

Grafický posun – respektování hranic obdélníků

Prvním zajímavým implementačním problémem je respektování hranic objektů při jejich grafickém posunu. Tato úloha je teoreticky zajímavá již pro řešení problém obdélníkových boxů, její rozpracování pro polygonální útvary by mohlo být námětem pro samostatnou práci.

Zjištění kolize objektů

Dva obdélníkové objekty kolidují tehdy, pokud se protínají jejich hrany, nebo pokud je jeden obdélník pohlcen v druhém.



Ilustrace 15: Kolize dvou obdélníků

Třída `java.awt.geom.Rectangle2D` obsahuje následující implementaci funkce, která zjišťuje konflikt dvou obdélníků:

```
/**
 * Tests if the interior of this <code>Rectangle2D</code>
 * intersects the interior of a specified set of rectangular
 * coordinates.
 * @param x,&nbsp;y the coordinates of the upper left corner
 * of the specified set of rectangular coordinates
 * @param w the width of the specified set of rectangular
 * coordinates
 * @param h the height of the specified set of rectangular
 * coordinates
 * @return <code>true</code> if this <code>Rectangle2D</code>
 * intersects the interior of a specified set of rectangular
 * coordinates; <code>false</code> otherwise.
 */
public boolean intersects(double x, double y, double w, double h) {
    if (isEmpty() || w <= 0 || h <= 0) {
        return false;
    }
    double x0 = getX();
    double y0 = getY();
    return (x + w > x0 &&
            y + h > y0 &&
            x < x0 + getWidth() &&
            y < y0 + getHeight());
}
```

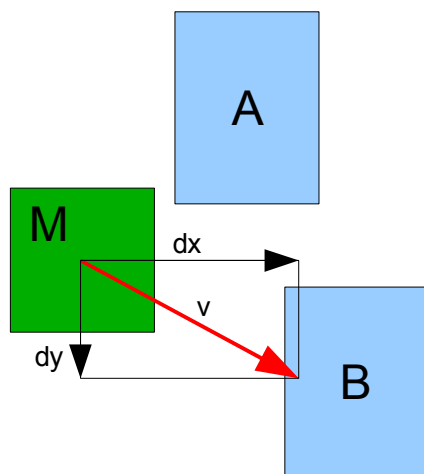
Jak se má chovat pohyb objektu při kolizi ?

Pokud je objekt posouván v určitém směru a v jeho cestě stojí kolmá překážka, pohyb objektu se zastaví. Pokud ale pohyb není na překážku kolmý, záleží na fyzikálních vlastnostech povrchu objektů, jestli dojde k zastavení pohybu, nebo, jestli se pohybová síla rozložená na vektory osy x a y převede do pohybu v jiném směru, v případě obdélníku ve směru osy, na jejímž směrovém vektoru neleží překážka.

Při pohybu objektu myši máme k dispozici přímo osově vektory pohybu – dx a dy , o které chceme objekt posunout.

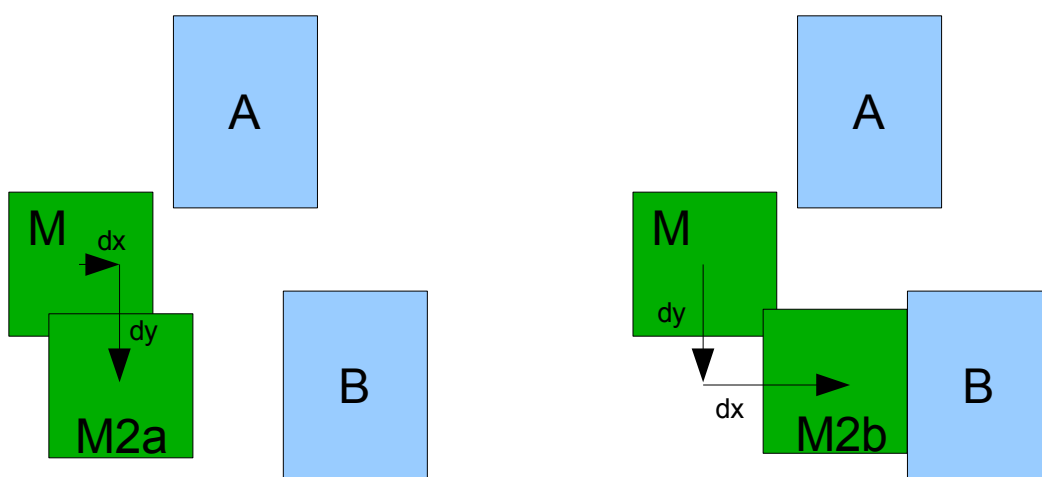
Kolize může nastat na jedné ze čtyř hran obdélníku, v extrémním případě na rozích objektů. Pokud dojde ke kolizi na horizontální hraně rovnoběžné s osou x , pohyb ve směru x se zastaví, pozice objektu je nastavena těsně ke stěně kolizního objektu. Při pohybu směrem nahoru je to spodní hrana druhého objektu, při pohybu směrem dolů je to jeho horní hrana. Při kolizi na stranách rovnoběžných s osou y dochází k obměněnému případu.

Při kolizi s více objekty je nejednoznačné, jakým způsobem postupovat při zjištění maximální polohy objektu ve směru směrového vektoru. Pokud na obrázku budeme pohybovat objektem M ve směru vektoru v , známe rozložený směrový vektor do složek dx a dy .



Ilustrace 16: Požadavek posunu objektu při vícenásobné kolizi

Pokud při zjišťování maximální polohy objektu budeme nejprve pohybovat objektem ve směru osy x a pak ve směru osy y , dostaneme objekt M2a. Při pohybu nejprve ve směru osy y dostaneme výslednou polohu objektu M2b. Pokud bychom pohybovali objektem přímo po směrovém vektoru, objekty M a A by kolidovaly na střetu svých hran.



Ilustrace 17: Možné výsledky sčítání jednotlivých složek pohybu

Nicméně při jednotkovém pohybu, kdy vektor v je dostatečně malý, se uvedený rozpor ve výpočtu výsledné pozice neprojeví. Jednotkový vektor pohybu nikdy nebude vzhledem k velikosti kontejneru tak veliký, jako na obrázku.

Pohyb a kolize objektů jsou dále komplikovány možností výběru více objektů a jejich společného pohybu po ploše.

Dále je na ploše umístěný objekt návěsu kamionu, na který nelze libovolně pokládat kontejnery. Nutná podmínka je, že pokud se objekt kontejneru a návěsu překrývají, musí být kontejner celý uvnitř návěsu – nemůže přesahovat jeho hrany.

V aplikaci je implementován algoritmus, který při pohybu objektu zjišťuje kolize s jinými kontejnery a návěsem, v případě kolize nastavuje maximální možnou pozici ve směru vektoru pohybu. Algoritmus posouvá všechny vybrané objekty v jednotkových krocích po směru osy x a následně osy y. Směr pohybu po ose se uchovává v proměnné signx a signy, určující znaménko přírůstku pohybu ve směru osy. Relativní velikost pohybu dx a dy jsou pak dvě kladná čísla, větší nebo rovna nule.

```
/**
 * Check if the rectangles from the list do conflict with existing
 * workplace components, excluding the checks for the selectedComponents
 * Also checks the x and y borders of the workplace as well
 * as the borders of the truck panel
 * @param bounds
 * @param selectedComponents - can be null
 * @return t
 */
private boolean isConflict(
    List<Rectangle> bounds, List<ContainerComponent> selectedComponents)
{
    for(Rectangle r: bounds) {
        if(r.x<0) return true;
        if(r.y<0) return true;

        for (ContainerComponent c2 : workplaceComponents) {
            if (selectedComponents != null &&
                !notIn(c2,selectedComponents))
                continue;

            Rectangle r2 = c2.getBounds();
            if (r2.intersects(r)) {
                return true;
            }
        }

        //check the borders of truck
        Rectangle truckBounds=truck.getBounds();
        if(truckBounds.intersects(r) && !truckBounds.contains(r)) {
            return true;
        }
    }
    return false;
}

/**
 * checks if the component can be moved to requested location.
 *
 * @param sourceComponent - component, which is dragged
 * @param requestedX - requested x position
 * @param requestedY - requested y position
 */
void moveSelectedComponents(
    ContainerComponent sourceComponent, int requestedX, int requestedY
) {
    Rectangle orgRect = sourceComponent.getBounds();
    Rectangle rNew = new Rectangle(requestedX, requestedY, orgRect.width,
        orgRect.height);

    // dx is positive vector of motion in the x direction
    int dx = rNew.x - orgRect.x;
    // dy is positive vector of motion in the y direction
    int dy = rNew.y - orgRect.y;
}
```



```

// signs for motion vectors
int signx = 1; if (dx < 0) { signx = -1;    dx = -dx; }
int signy = 1; if (dy < 0) { signy = -1;    dy = -dy; }

// does this move conflicts with other components ?
// this check speeds up the algorithm by skipping the next steps,
// but this check next is also necessary - sometimes there is no path
// between the two places
boolean conflict = false;
if(conflict==false) {
    for (ContainerComponent selectedComponent : selectedComponents) {
        Rectangle bounds = selectedComponent.getBounds();
        if (isConflict(selectedComponent, bounds.x + signx * dx,
            bounds.y + signy * dy,selectedComponents)) {
            conflict = true;
            break;
        }
    }
}

int dxM=dx;
int dyM=dy;

// if conflict, find the maximum dx and dy how we can move
if(conflict) {
    dxM=0;
    dyM=0;

    List<Rectangle> rects=new ArrayList<Rectangle>();
    for (ContainerComponent selectedComponent : selectedComponents) {
        rects.add(selectedComponent.getBounds());
    }

    //move all components in the steps in x direction and check validity
    for(int i=1; i<=dx; i++) {
        for(Rectangle r : rects) {
            r.x+=signx;
        }
        if(!isConflict(rects,selectedComponents)) {
            dxM=i;
        }
    }

    //restore the rectangles
    rects.clear();
    for (ContainerComponent selectedComponent : selectedComponents) {
        Rectangle r=selectedComponent.getBounds();
        //use newly computed X
        r.x = r.x + signx * dxM;
        rects.add(r);
    }

    //move all components in the steps in x direction and check validity
    for(int i=1; i<=dy; i++) {
        for(Rectangle r : rects) {
            r.y+=signy;
        }
        if(!isConflict(rects,selectedComponents)) {
            dyM=i;
        }
    }
}

// the result position is defined with the size of vector dx and dy
// does this move conflicts with other components ?
for (ContainerComponent selectedComponent : selectedComponents) {
    orgRect = selectedComponent.getBounds();
    requestedX = orgRect.x + signx * dxM;
    requestedY = orgRect.y + signy * dyM;
    selectedComponent.setLocation(requestedX, requestedY);
}
}

```

Knihovna objektů

Kontejnery, které jsou pro aplikaci dostupné, jsou definovány v souboru library.xml, který se nachází v aktuálním adresáři aplikace load-organizer.jar.

V knihovně přiložené na doprovodném CD jsou definovány reálné objekty používané firmou Denios s.r.o.

Knihovna definuje následující formát dat:

```
<?xml version="1.0" encoding="utf-8"?>

<library>

<container>
  <sap>10001</sap>
  <size1>1284</size1>
  <size2>759</size2>
  <weight>487</weight>
  <name>Wanne W1</name>
</container>

<container>
  <sap>10002</sap>
  <size1>1350</size1>
  <size2>850</size2>
  <weight>573</weight>
  <name>2GST K RAL</name>
</container>

...

<container>
  <sap>10031</sap>
  <size1>2000</size1>
  <size2>1600</size2>
  <weight>1600</weight>
  <name>Türelement GF17/33</name>
</container>

</library>
```

Ukládání a nahrávání dokumentů

Dokumenty jsou ukládány do souborů s příponou LOD – Load Organizer Document. Soubor je xml dokumentem komprimovaným technologií GZip.

V uloženém dokumentu jsou následující sekce informací:

- nastavení dokumentu – autor, název, datum, rozměry návěsu

- statistiky dokumentu – využitá plocha návěsu, rozložení hmotností.

Statistiky uložené v dokumentu jsou jen informačního charakteru, při nahrávání souboru se ignorují, aktuální stav je přepočítán.

- seznam kontejnerů na ploše a jejich umístění.

Při nahrávání dokumentu je nejprve zkontrolována knihovna, jestli obsahuje všechny kontejnery v dokumentu použité. Pokud ne, knihovna je doplněna. Všechny rozměry v dokumentu jsou v milimetrech, pozice kontejnerů je relativní k pozici návěsu.

Příklad dekomprimovaného obsah souboru .LOD:

```
<?xml version="1.0" encoding="utf-8"?>
<load-organized-document version="1.0">
  <settings>
    <name>Vzorový dokument</name>
    <author>Dušan Saiko</author>
    <date>2006/05/01</date>
    <truck-width>13450</truck-width>
    <truck-height>2450</truck-height>
    <truck-tonnage>24000</truck-tonnage>
  </settings>
  <statistics>
    <total-containers>14</total-containers>
    <used-area>31.056456</used-area>
    <used-area-percentage>94.24613003565739</used-area-percentage>
    <total-weight>20028</total-weight>
    <load-percentage-front>60.9496704613541</load-percentage-front>
    <load-percentage-left>42.6802476532854</load-percentage-left>
  </statistics>
  <workplace>
    <container>
      <sap>10002</sap>
      <size1>1350</size1>
      <size2>850</size2>
      <weight>573</weight>
      <name>2GST K RAL</name>
      <rotated>true</rotated>
      <mandatory>false</mandatory>
      <x>11333.0</x>
      <y>933.0</y>
    </container>
    ...
  </workplace>
</load-organized-document>
```

X.Příloha 1: Struktura přiloženého CD

Přiložené CD obsahuje následující strukturu dat:

- /
 - ukh-saiko-diplomova-prace-2006.odt**
soubor ve formátu OpenOffice Document
 - ukh-saiko-diplomova-prace-2006.pdf**
 - ukh-saiko-diplomova-prace-2006.txt**
textový soubor bez formátování
 - load-organizer.bat**
spouštěcí skript aplikace se systémovým vzhledem
 - load-organizer-javalookandfeel.bat**
spouštěcí skript aplikace se vzhledem Javy
 - load-organizer.sh**
 - load-organizer-javalookandfeel.sh**
- /bin
 - Vzorový dokument lod**
ukázkový uložený dokument
 - library.xml**
reálná knihovna objektů (pro spuštění programu musí být v cestě)
 - load-organizer.jar**
programový archiv, lze spustit příkazem **java -jar load-organizer.jar**
- /src
 - Zdrojové kódy programu**
- /jre
 - Instalace Java JRE 1.5_06 pro OS Windows a Linux.**

XI.Příloha 2: Seznam obrázků a tabulek

Ilustrace 1: Logistické toky v komplexním průmyslovém podniku.....	5
Ilustrace 2: Úvodní list produktového katalogu Denios s.r.o.....	12
Ilustrace 3: Room Arranger jako jednoduchý systém pro organizaci nákladu kamionu.....	16
Ilustrace 4: Logistický modul ekonomického softwaru ABAS.....	17
Ilustrace 5: Snímek obrazovky aplikace LoadOrganizer.....	38
Ilustrace 6: Osm možností směru postupu při hledání volného místa v návěsu.....	41
Ilustrace 7: Působení výsledné kriteriální funkce.....	44
Ilustrace 8: Působení kriteriální funkce pro zatížení zadní části návěsu.....	45
Ilustrace 9: Obměna kriteriální funkce pro rozložení hmotnosti.....	45
Ilustrace 10: Obměna kriteriální funkce pro zatížení levé části návěsu.....	45
Ilustrace 11: Dialog nastavení aplikace.....	47
Ilustrace 12: Tabulka měřených výsledků výpočtu.....	49
Ilustrace 13: Grafická podoba dosaženého výsledku využití plochy 93,19%.....	50
Ilustrace 14: Grafická podoba dosaženého výsledku využití plochy 93,31%.....	50
Ilustrace 15: Kolize dvou obdélníků.....	53
Ilustrace 16: Požadavek posunu objektu při vícenásobné kolizi.....	54
Ilustrace 17: Možné výsledky sčítání jednotlivých složek pohybu.....	55

XII.Příloha 3: Seznam odkazů a použité literatury

[1.Logistika]

LOGISTIKA – teorie a praxe
Josef Sixta, Václav Mačát
CP Books, a.s. 2005

[2.Ekonomika dopravy]

Ekonomika dopravy.
Gürtlich,G.H. - Demand,E.
Nakladatelská společnost BaBtext, Praha 1993.

[3.Slovník]

Stručný filozofický slovník
Praha, Svoboda 1966

[4.Logistický management]

Logistický management,
Pernica, P.
Praha: Radix 1998

[5.GA.UI]

Umělá Inteligence - genetické algoritmy a jejich využití v praxi
RNDr. Jaroslav Teda, Ph.D.
URL: <http://www.programujte.com/view.php?cislocclanku=2005072601>

[6.GA.Heuristika]

Heuristika
URL: <http://cs.wikipedia.org/wiki/Heuristika>

[7.GA.Fce]

Průběh funkce
URL: http://cs.wikipedia.org/wiki/Pr%C5%AFb%C4%9Bh_funkce

[8.GA.Ex]

Adaptive Genetic Algorithms with Multiple Subpopulations and Multiple Parents
B. Thijssen
March 1997
URL: <http://www.liacs.nl/MScThesis/bthijssse.97.ps.gz>

[9.GA.TSP]

Traveling Salesman Problem
Georgia Institute of Technology
URL: <http://www.tsp.gatech.edu/>

[10.GA.TSP]

Traveling Salesman Problem - Java Genetic Algorithm Solution

Dušan Saiko

URL: <http://www.saiko.cz/ai/tsp.html>

[11.GA.Reconfiguring]

Re-configuration in Agile Manufacturing Systems

Department of Mechanical Engineering

Indian Institute of Technology

URL: http://www.ici.ro/ici/revista/sic2000_4/art01.htm

[12.GA.Circuits]

Physical Design of VLSI Circuits and the Application of Genetic Algorithms

CiteSeer.IST

URL: <http://citeseer.ist.psu.edu/lienig97physical.html>

[13.GA.Antena]

A Parallel Electromagnetic Genetic-Algorithm Optimization (EGO) Application for Patch Antenna Design

IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, VOL. 52, NO. 9, SEPTEMBER 2004

URL: http://www.ee.ucla.edu/faculty/papers/yrs_trans-ant-prop_sept04.pdf

[14.GA.Protein]

Learning genetic algorithm application to the protein structure prediction problem

Symposium on Applied Computing

URL: <http://portal.acm.org/citation.cfm?id=372202.372357>

[15.GA.Maximum]

Genetic Algorithms

Searching a maximum of a function

ITC-Leipzig/ Ralf Moros

URL: http://techni.tachemie.uni-leipzig.de/jsga/applications/fktmax/fktmax_gashort.html

[16.GA.Design]

Evolutionary Computation and its application to art and design

Craig Reynolds

URL: <http://www.red3d.com/cwr/evolve.html>

[17.GA.Design2]

Genetic Algorithms and Engineering Design

Mitsuo Gen, Runwei Cheng

URL: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471127418.html>

[18.GA.HW]

2004 NASA/DoD Conference on Evolvable Hardware

URL: <http://ehw.jpl.nasa.gov/events/nasaeh04/>

[19.GA.Logistic]

Application of Genetic Algorithms for the Design of Large-Scale Reverse Logistic Networks in Europe's Automotive Industry
University of Technology Aachen 2005
URL: <http://doi.ieeecomputersociety.org/10.1109/HICSS.2004.1265246>

[20.GA.Logistic2]

Evolved genetic algorithms with fuzzy aggregation applied to priorities in logistic systems
Silva, C.A. Sousa, J.M. Runkler, T. Sa da Costa, J.M.G.
Dept. of Mech. Eng., Tech. Univ. Lisbon, Portugal; 2005
URL:
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=1248777&isnumber=27958

[21.GA.Logistic3]

A Hybrid Algorithm for Distribution Problems
David Bredström, Linköping University, 2005
URL: <http://doi.ieeecomputersociety.org/10.1109/MIS.2005.59>

[22.GA.Logistic4]

A palletize-planning system for multiple kinds of loads using GA search and traditional search
S. Koide, Ishikawajima-Harima Heavy Ind. Co. Ltd., Japan
URL: <http://doi.ieeecomputersociety.org/10.1109/IROS.1995.525933>

[23.GA.Logistic5]

Route Selection in Public Transport Network Using GA (Genetic Algorithm)
Univ. of Seoul, Dept. of Geoinformatics, Jeonnong, Dongdaemun
URL: http://www10.giscafe.com/link/display_detail.php?link_id=13138

[24.Java.SwingSWT]

Swing and SWT: A Tale of Two Java GUI Libraries
Mauro Marinilli
URL: http://www.developer.com/java/other/article.php/10936_2179061_1