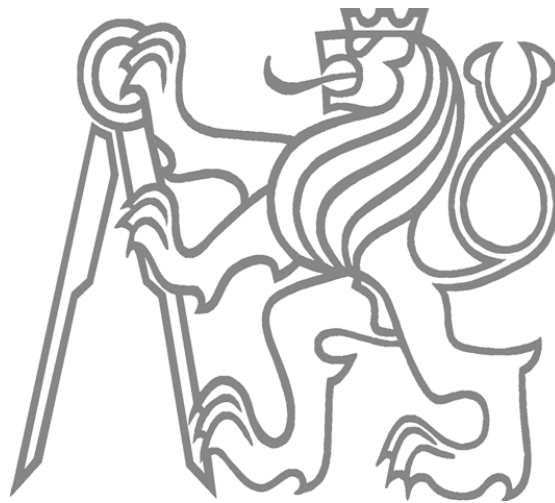


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
KATEDRA ŘÍDÍCÍ TECHNIKY



Diplomová práce

**Nástroj pro plánování plnění
kontejnerů**

autor

Bc. Josef Mrázik

vedoucí práce

Ing. Libor Waszniowski Ph.D.

květen 2008

Prohlášení o autorství

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne _____

podpis

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu diplomové práce, kterým byl Ing. Libor Waszniowski Ph. D., za cenné rady, metodické vedení a čas, který věnoval mé práci.

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra řídicí techniky

Abstract

The paper presents an algorithm for solving 3D-Bin Packing problem with specific constraints. The task is to place maximum amount of box-shaped items into a single bounded container, while the sequence of items has not to be changed. The task is NP-Hard and very hard to solve in praxis. Deterministic algorithm based od Branch and bound method is presented. The computational experiments were done to prove the functionality.

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra řídicí techniky

Abstrakt

Práce je věnována návrhu nástroje, sloužícího k řešení úlohy rozmístění co nejvyššího počtu předmětů tvaru kvádrů do kontejneru v zadaném pořadí. Tato úloha je NP-obtížná a v praxi tedy velmi těžko řešitelná. Práce prezentuje deterministický algoritmus plnění jednoho kontejneru založený na metodě větví a mezí, který byl navržen, implementován a na praktických výpočetních experimentech byla demonstrována jeho činnost.

Obsah

Prohlášení o autorství	i
Poděkování	ii
Abstract in English language	iii
Abstrakt v Českém jazyce	iv
Oficiální zadání	v
Seznam obrázků	viii
Seznam tabulek	ix
1 Úvod do problematiky	1
1.1 Účel nástroje a jeho praktické využití	1
1.2 Typ úlohy, složitost a možnosti řešení	2
1.2.1 Typ úlohy	2
1.2.2 Algoritmická složitost	3
1.2.3 Metody řešení	3
2 Definice řešeného problému a metody řešení	6
2.1 Formulace úlohy	7
2.2 Vlastnosti rozmisťovaných předmětů	7
2.3 Způsob plnění kontejneru	8
3 Algoritmus řešení	9
3.1 Přístup k řešení	9
3.2 Zvolená metoda řešení	11
3.3 Struktura algoritmu	12
3.4 Navržený způsob generování pozic k umístění předmětů	13
3.4.1 Generování nových pozic	14
3.4.2 Modifikace existujících pozic	17
3.5 Způsob vyhodnocení korektnosti pozice předmětu	20
3.5.1 Vzájemná pozice předmětů	20
3.5.2 Stabilita nákladu	21
3.6 Kritéria výběru kandidátů pro rozvoj	22

3.6.1	Kritéria založená na rozměrech	23
3.6.2	Kritérium zaplněného prostoru	23
3.6.3	Kritérium volného místa	25
3.7	Omezování stromu řešení	25
3.7.1	Systém rezervace prostoru	26
3.7.2	Podmínky odstranění uzlu	29
3.8	Postupy rozvoje stromu řešení	30
3.8.1	Úplné prohledání	31
3.8.2	Prohledávání po částech	31
3.8.3	Rychlé prohledávání	35
4	Implementace nástroje	36
4.1	Vstupy nástroje	36
4.2	Struktura nástroje a hlavní funkce	37
4.3	Hlavní nastavitelné parametry a jejich význam	39
5	Výpočetní experimenty	42
5.1	Velikost stavového prostoru	42
5.2	Význam rezervace prostoru	43
5.3	Vliv jednotlivých parametrů na řešení	45
6	Závěr	47
A	Obsah přiloženého CD	48
	Literatura	49

Seznam obrázků

2.1	Konvence v pojmenovávání prostorového uspořádání	6
2.2	Přímé umístění předmětů	8
3.1	Myšlenka definovaných pozic umístění	10
3.2	Nedokonalost původního algoritmu	11
3.3	Srovnání funkce algoritmů nalezení pozic pro umístění	12
3.4	Roviny vzniku nových pozic pro umístění	14
3.5	Předměty tvořící obálku v řezu	15
3.6	Získání nových pozic	16
3.7	Modifikace stávajících pozic	18
3.8	Odebírání znehodnocených pozic	19
3.9	Kontrola stability pozice	21
3.10	Nestabilní pozice neodhalené algoritmem	22
3.11	Hranice pro kritérium využitého objemu	24
3.12	Výpočet ztraceného místa	26
3.13	Rezervace objemu podle kritérií	28
3.14	Postup stromem po částech	33
4.1	Struktura implementovaných funkcí	38
5.1	Velikost stromu řešení	43

Seznam tabulek

5.1	Velikost stromu řešení	42
5.2	Význam systému rezervace	44
5.3	Úskalí rezervace	44
5.4	Vliv parametrů	46

Kapitola 1

Úvod do problematiky

Práce je věnována řešení jedné z mnoha formulací úlohy, řešící problém rozmístění zadané množiny předmětů ve třírozměrném prostoru při splnění předem známých omezujících podmínek. V úvodní kapitole je uveden praktický význam řešení úloh tohoto typu a proveden přehled jejich existujících variant a metod řešení.

1.1 Účel nástroje a jeho praktické využití

Současný celosvětový trend převádět výrobu do míst s levnější pracovní silou, společně s integrací jednotlivých státních trhů ve větší celky, vede mimo jiné k obrovskému nárůstu objemu zboží, které je nutné přepravit do místa jeho spotřeby či funkce. Díky tomu je ze strany výrobců vyvíjen obrovský tlak na snížení přepravních nákladů, při zachování kvality a rychlosti přepravy.

Jedním z kroků, které vedou k vyšší efektivitě přepravy přímo a bez ovlivnění její kvality, je ten ve své podstatě nejjednodušší - co nedůslednější využití přepravního prostoru, který je k dispozici. Hlavní výhodou tohoto přístupu je fakt, že vhodnější rozmístění nákladu v přepravním prostoru nepřináší žádné nežádoucí důsledky, ani nezvyšuje nároky na přepravní techniku, ale naopak alespoň částečně řeší i další aspekty přepravy. Jako ilustrační příklad lze uvést zvýšení stability nákladu během přepravy. Právě splnění požadavku na co nejefektivnější rozmístění nákladu v přepravním kontejneru je úkolem nástroje, který je výsledkem této práce.

1.2 Typ úlohy, složitost a možnosti řešení

1.2.1 Typ úlohy

Jak bylo naznačeno v úvodu kapitoly, existuje mnoho modifikací úlohy řešící rozmístění předmětů ve třírozměrném prostoru. Mezi nejčastěji řešené varianty s přímým vztahem k praxi patří:

- *Knapsack loading* - v této formulaci je úkolem vybrat ze vstupní množiny předmětů takovou podmnožinu, jejímž rozmístěním bude obsazena největší část vnitřního objemu kontejneru a takové rozmístění nalézt.
- *Container loading* - je formulací úlohy, kdy všechny předměty mají být rozmístěny do jednoho kontejneru, který má ovšem nekonečně velký jeden z rozměrů. Cílem úlohy je pak nalézt rozmístění předmětů minimalizující právě tento uvolněný rozměr nákladu.
- *Bin packing* - úkolem je rozmístit celou vstupní množinu předmětů do kontejnerů definovaných rozměrů (může být použito více kontejnerů různých velikostí). Cílem je přitom aby počet kontejnerů nutných pro umístění všech předmětů byl co nejnižší.

Výčet typů úloh není samozřejmě úplný a i uvedené varianty lze dále dělit podle konkrétních vlastností nákladu (např. offline — znám celý náklad předem vs. online — známe vždy jen následující předmět k umístění), omezujících podmínek (možnost vs. nemožnost měnit pořadí umísťovaných předmětů) a podobně. Podrobný přehled existujících variant je uveden v [1].

Úloha řešená v rámci této práce je jednou z modifikací úloh typu Bin packing a proto její vlastnosti budou rozebrány podrobněji. Ve své podstatě jde o rozšíření tzv. problému batohu (*Knapsack problem*) do třírozměrného prostoru. Problém batohu je definován jako rozhodovací úloha, kdy úkolem je vybrat z množiny kladných celých čísel (představujících hodnotu jednotlivých předmětů) takovou podmnožinu, jejíž součet bude roven nebo co nejbližší definované konstantě (představující kapacitu batohu). Již tato základní úloha je algoritmicky velmi těžko řešitelná a jejím rozšířením je složitost ještě umocněna.

1.2.2 Algoritmická složitost

Je obecně známo, že tzv. problém batohu patří do skupiny úloh, označovaných jako NP-úplné (*NP-Complete*). Rozšířením problému do 3D prostoru, spolu se změnou typu úlohy z rozhodovací na optimalizační kdy nám nejde jen o volbu zda ano či ne, ale i o vhodné umístění, složitost problému ještě roste a prostorová úloha typu *Bin Packing* se dostává do třídy složitosti NP-obtížné (*NP-hard*).

Úlohy typu NP-complete (NP-hard) se vyznačují nepolynomiální závislostí velikosti stavového prostoru řešení na velikosti řešeného problému, společně s možností ověřit správnost řešení v polynomiálním čase. Právě nepolynomiální závislost velikosti stavového prostoru na velikosti řešeného problému je důvodem, proč úlohy tohoto typu patří k algoritmicky nejobtížněji řešitelným problémům.

Složitost řešení úloh tohoto typu lze ilustrovat na příkladu hledání cesty skrze bludiště. Nalezení cesty je velmi složitá úloha s obrovským stavovým prostorem řešení (pokud uvažujeme pouze křižovatky typu rozdvojení, je složitost 2^n , kde n je počet rozdvojení), zatímco ověření správnosti nalezeného řešení je lineárně závislé na počtu křižovatek jimiž cesta prochází. Pro $n=50$ bychom v tomto jednoduchém případě dostali stavový prostor o velikosti přibližně $1,126 \cdot 10^{12}$. Při předpokladu, že dokážeme otestovat 1000 prvků za sekundu by to znamenalo, že stavový prostor prohledáme za přibližně 35 700 let. To je jistě neporovnatelné s dobou nutnou pro ověření řešení, která by byla 0,05 sekundy.

Podrobný popis vlastností NP-úplných a NP-obtížných přesahuje rozsah této práce. Více informací o problematice výpočetní složitosti lze nalézt v [2].

1.2.3 Metody řešení

Optimální algoritmy

Deterministické algoritmy pro nalezení optimálního řešení, jako například metoda hrubé síly procházející všechny možnosti, dosahují řešení úloh typu NP pouze v nepolynomiálním čase. Doba potřebná k řešení je přitom, jak bylo ilustrováno v předchozí podkapitole, již pro nevelký rozsah problému nepřijatelná a tyto algoritmy jsou tedy v obecném případě nevyužitelné.

V polynomiálním čase, lze nalézt optimální řešení NP úloh pouze s užitím nedeterministického algoritmu. Nedeterministický algoritmus lze v krátkosti popsat jako postup, který na cestě k výsledku volí vždy nejlepší možný směr, bez možnosti, ale i nutnosti, tuto volbu jakýmkoliv způsobem zdůvodnit. Není příliš velkým překvapením, že takovýto postup dosud nebyl v praxi zrealizován a existuje pouze v teoretické rovině.

Z výše uvedeného tedy vyplývá, že striktní požadavek na nalezení optimálního řešení by byl nerealistický. Při řešení jsme nuceni využít přibližných algoritmů a v řadě případů se spokojit s nalezením řešení suboptimálního.

Přibližné algoritmy

Přibližné algoritmy pro řešení úloh námi uvažovaného typu lze rozdělit na dvě základní skupiny, zásadně se lišící ve svém přístupu.

První skupinou jsou deterministické postupy, vycházející z analýzy struktury problému, nebo ze zkušeností a znalostí získaných o problému z jiných zdrojů. Mezi nejběžnější patří:

- Heuristická pravidla pro výběr pozice k umístění. Tato pravidla jsou nejjednodušší z přístupů k řešení. Poskytují řešení ve velmi krátkém čase a jsou proto použitelná i pro problémy velkého rozsahu, ovšem jejich výsledky jsou mnohdy neuspokojivé. Množství používaných pravidel a přístupů k řešení je obrovské, mezi jinými lze uvést nejjednodušší *First fit*, *Best fit*.
- Metoda větví a mezí (*Branch and bound*) založená na postupném rozvoji stromu řešení ve dvou opakujících se hlavních fázích. První (fáze větvení stromu řešení) je rozvoj uzlů stromu řešení, které mají nejvyšší hodnotu kritériální funkce a tedy nejvyšší šanci stát se základem pro nalezení nejlepšího možného řešení, druhou (fáze prořezávání stromu řešení) je pak odstranění uzlů, jejichž rozvojem již není možné stávající nejlepší dosažené řešení překonat. Způsoby výběru uzlů k rozvinutí i obebrání jsou závislé na konkrétní implementaci.

- Postupy speciálně navržené pro konkrétní formulace úlohy jako například *Strip packing* pro úlohu Container Loading. Tento postup je založen na rozdělení vnitřního prostoru na vhodně široké pásy, které jsou následně obsazovány předměty odpovídajících rozměrů. Přístup tohoto typu je popsán v [3]

Druhou skupinu tvoří postupy, které můžeme shrnout pod pojem *Softcomputing*¹. Tyto nedeterministické postupy se snaží pro řešení využít analogie s přírodními procesy. Mezi nejznámější patří:

- Genetické algoritmy - simulují princip evoluce druhů. Počáteční populace (nejčastěji náhodně vygenerovaných) jedinců (řešení úlohy) je na základě kritériální funkce podrobena selekci. Vybraní jedinci-rodice, pak tvoří základ pro vznik nové populace, která je tvořena jedinci vzniklými kombinací částí rodičů a s určitou pravděpodobností podrobena náhodným změnám (realizace křížení a mutace).
- Simulované žíhání - simuluje princip kalení kovů. Ve svém principu jde o lokální prohledávání s proměnlivou velikostí okolí a využitím pravděpodobnostních modelů (jako další pro rozvoj může být přijato i řešení s horší hodnotou kritériální funkce).

Tyto algoritmy mají dobré výsledky, ovšem také řadu úskalí. Prvním je fakt, že pracují s pravděpodobnostními modely a opakovatelnost výsledků je tedy nízká. Z dalších pak lze uvést problémy při definici stavové reprezentace a vztahu sousednosti jednotlivých řešení, které jsou pro dobrou funkci algoritmů tohoto typu zásadní. Problematika těchto algoritmů je detailně zpracována v [4].

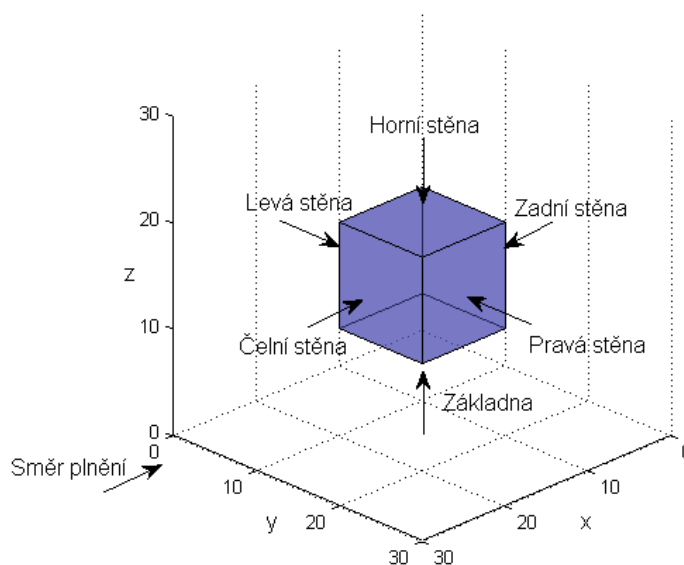
¹Softcomputing je přístup, který se snaží na principech tolerance, nepřesností, nejistot a zobecnění k dosažení zvládnutelnosti, robustnosti a nízké výpočetní ceny řešení zejména těch úloh, jejichž exaktní řešení je problematické. Jakýmsi vzorem pro tyto metody je činnost lidské mysli.

Kapitola 2

Definice řešeného problému a metody řešení

Jak bylo naznačeno v úvodní kapitole, existuje úloha rozmístění předmětů v tří-rozměrném prostoru v mnoha modifikacích, které se mohou od sebe výrazně lišit. Tato kapitola je proto věnována popisu parametrů řešeného problému, užitých zjednodušení a požadavků.

V textu je často nutné popisovat prostorové uspořádání nákladu. V celé práci bude nadále využito pojmenování směrů a souřadnic, jaké je uvedeno na obrázku 2.1.



OBRÁZEK 2.1: Názvosloví použité v práci při popisu prostorových vlastností nákladu a kontejneru.

2.1 Formulace úlohy

Úloha řešená v rámci této práce spočívá v nalezení takového rozmístění části z množiny předmětů do vnitřního prostoru jednoho uvažovaného přepravního kontejneru, aby procentuální využití objemu tohoto kontejneru bylo co nejvyšší. Rozměry kontejneru přitom nejsou fixní, nýbrž definovány jako jeden ze vstupů úlohy. Velmi důležitým aspektem řešené formulace je poněkud netradiční požadavek na zachování pořadí předmětů ze vstupní množiny. Předměty tedy musí být rozmísťovány v předem určeném pořadí, které nelze ovlivnit, a žádný nesmí být vynechán, což odpovídá úlohám typu online rozmísťování, ale při rozmísťování lze využít znalosti budoucí sekvence příchozích předmětů, což odpovídá úlohám typu offline rozmísťování. Řešená úloha je tedy jaksi na rozhraní obou typů.

Pro potřeby algoritmu můžeme tedy při uvažování výše uvedeného omezení úlohu formulovat jako nalezení takového korektního rozmístění, které obsahuje největší počet předmětů ze zadané vstupní sekvence tak, aby žádný z nich v žádném ze tří rozměrů nepřesahoval maximální povolený rozměr. Rozměry všech předmětů vstupní množiny, stejně jako všechny jejich další relevantní vlastnosti jsou k dispozici již na začátku řešení úlohy.

Vlastnostem předmětů a dalším omezením na způsob rozmísťování jsou věnovány samostatné podkapitoly.

2.2 Vlastnosti rozmísťovaných předmětů

Velmi důležitým zjednodušením obecné úlohy je předpoklad, že rozmísťované předměty mají tvar kvádrů. Tento předpoklad je využíván v naprosté většině prací zabývajících se touto problematikou (předpoklad konvexnosti tvaru snad bez výjimky), neboť výrazně zjednodušuje vyhodnocení vzájemné polohy předmětů a jen velmi málo zmenšuje okruh v praxi řešených problémů, kdy téměř vše je přepravováno v obalech právě tvaru kvádrů.

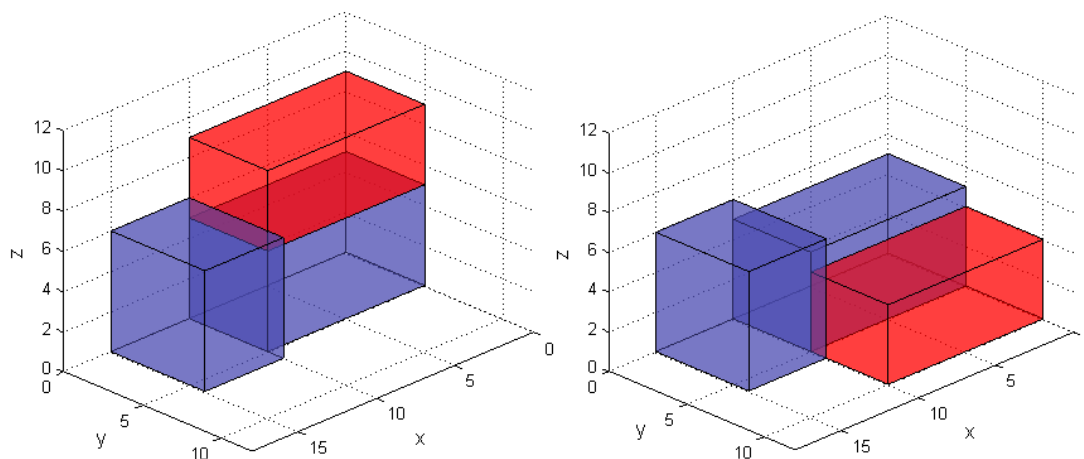
Dalším bodem zjednodušení je zanedbání hmotnosti jednotlivých předmětů. Hmotnost není uvažována jako omezující podmínka ve smyslu omezení maximální povolené hmotnosti nákladu a jedinou uvažovanou fyzikální vlastností předmětů jsou jejich rozměry. Při vyhodnocování stability předmětu je uvažováno homogenní

rozložení hmotnosti, což umožňuje stabilitu určit pouze z rozměrů a pozice vůči zbytku nákladu.

Velmi důležitým předpokladem řešení je možnost provádět rotace rozmisťovaných předmětů. Nástroj pracuje s rotacemi ve všech třech osách, ovšem pro každý předmět jsou povolené rotace definovány samostatně.

2.3 Způsob plnění kontejneru

Plnění kontejneru je uvažováno ze strany, nikoliv shora. Z požadavku na zachování pořadí tedy plyne omezení na způsob plnění kontejneru, neboť je třeba zajistit nejen aby na zvolené pozici byl pro konkrétní předmět prostor, ale také aby byla pozice při stávajícím stadiu řešení přímo přístupná. Pozice je přitom považována za přímo přístupnou, pokud na ní lze předmět umístit bez nutnosti jej zasouvat jiným než přímým směrem za již umístěné předměty. Není povoleno ani nechat předmět „zapadnout“ za jiný již umístěný. Omezení jsou ilustrována na obrázku 2.2.



OBRÁZEK 2.2: Ilustrace nepovolených způsobů umístění. Vlevo „zapadnutí“ za již umístěný předmět, vpravo zasunutí do strany.

Toto omezení ve svém důsledku zajišťuje realizovatelnost finálního rozmístění s pomocí mechanizace (např. vysokozdvizný vozík).

Kapitola 3

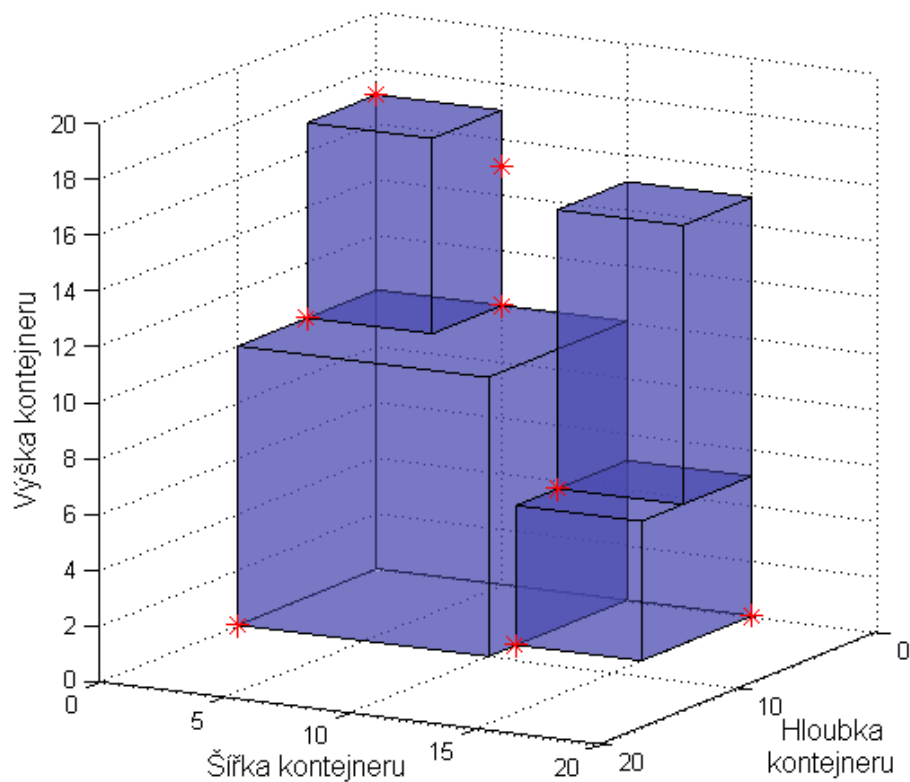
Algoritmus řešení

Tato kapitola je věnována popisu navrženého algoritmu. Nejprve je nastíněn ideový základ algoritmu. V dalších podkapitolách je pak představen nově navržený algoritmus a detailně popsány postupy užívané pro rozvoj a omezování stromu řešení. Nakonec jsou popsány užívané modifikace postupu a jejich vhodnost pro řešení v závislosti na velikosti konkrétní instance.

3.1 Přístup k řešení

Prostudováním dostupných materiálů bylo zjištěno, že úloha v požadované formulaci dosud nebyla řešena a není tedy možné přímo navázat na již existující řešení. Lze ovšem využít poznatků prezentovaných v dostupné literatuře a navázat na ně alespoň v ideové rovině.

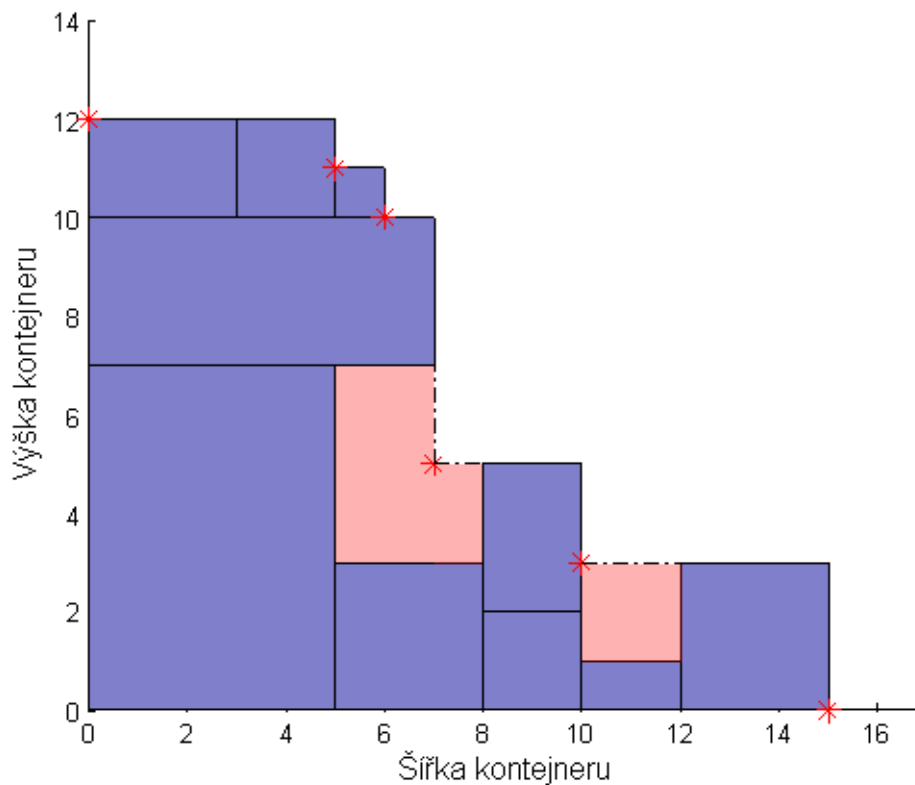
Jako ideový základ při návrhu vlastního řešení posloužila práce *The Three - Dimensional Bin Packing Problem*, jejímiž autory jsou Silvano Martello, David Pisinger a Daniele Vigo [5]. Jejich práce představuje exaktní algoritmus založený na metodě větví a mezí, řešící úlohu typu Bin Packing ve třírozměrném prostoru. V jejich formulaci je, narozdíl od naší formulace definované v kapitole 2, tedy naráz plněno několik kontejnerů a pořadí umísťovaných předmětů není pevně dáno. Nové předměty jsou přitom umísťovány pouze do předem určených pozic, vypočtených na základě prostorového uspořádání již umístěného nákladu. Právě tato myšlenka, ilustrovaná na obrázku 3.1, se stala základem námi navrženého nástroje, i když samotný postup musel být, kvůli odlišné formulaci úlohy, velmi zásadně upraven.



OBRÁZEK 3.1: Princip přesně definovaných pozic pro umístění předmětů

Původní algoritmus, detailně popsáný v [5] pracuje na principu řezů nákladem v rovinách definovaných již rozmístěnými předměty. Pro každé částečné řešení, je celá množina pozic pro umístění určována znovu, přičemž asymptotická složitost tohoto postupu je n^2 , kde n je počet již rozmístěných předmětů. Navíc jimi prezentovaný princip v rovině řezu zcela zanedbává prostor, který je z jednoho ze směrů zastíněn nákladem, jak je ilustrováno na obrázku 3.2.

Tento postup, obstojně fungující v případě možnosti řadit předměty a využít současně více kontejnerů, znamená v případě naší formulace obrovskou ztrátu potenciálně využitelného prostoru a musel být tedy modifikován. Hlavní změnou je schopnost námi navrženého systému využít, způsobem dokumentovaným v části 3.2, smysluplné pozice i v prostorech původním algoritmem zanedbávaných. Díky iterativnímu přístupu, kdy je, místo celkového výpočtu pozic pro každý uzel řešení, využita znalost množiny pozic přímého předchůdce aktuálního uzlu je asymptotická složitost výpočtu pozic s pomocí námi navrhovaného postupu $n \cdot \log n$. Odvození složitosti je provedeno v části 3.4.

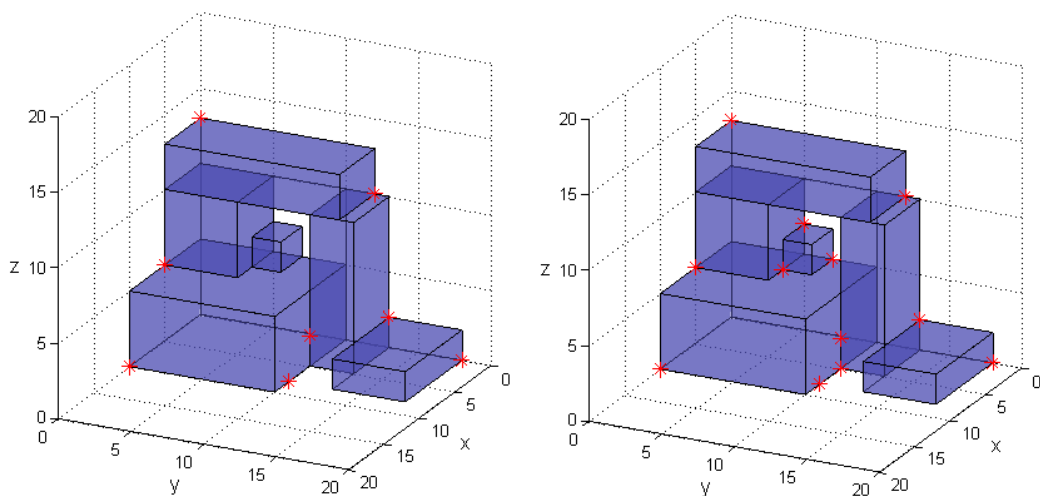


OBRÁZEK 3.2: Princip nalezení bodů pro umístění v řezu nákladem, používaný v [5], odkud byla ilustrace volně převzata. Červené plochy značí ztracené místo.

Pro ilustraci rozdílu ve funkci jsou na obrázku 3.3 uvedeny pozice určené na stejně rozmístěném nákladu původním a nově navrženým algoritmem.

3.2 Zvolená metoda řešení

Z mnoha možných přístupů k řešení byla zvolena metoda větví a mezí. Důvodem této volby je fakt, že tento přístup je velmi universální a zároveň lze modifikací jednotlivých částí postupu dosáhnout velmi dobrých výsledků. Nástroj navržený v rámci této práce přitom, podle rozsahu řešeného problému, nebo explicitního příkazu uživatele, používá jinou modifikaci této metody ve smyslu způsobu procházení stromu řešení a principů jeho prořezávání.



OBRÁZEK 3.3: Srovnání funkce původního (vlevo) a námi prezentovaného algoritmu nalezení pozic pro umístění předmětů.

3.3 Struktura algoritmu

Jak bylo uvedeno úvodu kapitoly, byla pro řešení zvolena metoda větví a mezí. Tento fakt do značné míry předurčuje základní strukturu algoritmu, která je následující:

- Inicializace stromu řešení - v této fázi jsou definovány parametry prohledávání stromu řešení a je vytvořena jeho instance.
- Rezervace místa - v této fázi je rezervováno místo pro předměty, jejichž umístění by mohlo být z nějakého důvodu problematické. Tento krok je podrobně vysvětlen v části 3.7.1.
- Přípravný krok - spočívá v nalezení nějakého řešení nejrychlejším možným způsobem. Nalezené rozmístění je pak využito při vlastním výpočtu pro omezení stromu řešení, popsáném v části 3.7.
- Vlastní výpočet - je hlavní částí a je tvořen cyklickým vykonáváním fází
 - Rozvoj stromu - je proveden rozvoj vybraných kandidátů a nové uzly přidány do stromu
 - Prořezání stromu - ze stromu jsou odstraněny uzly, jejichž rozvoj nemá při řešení smysl

- Ukončení prohledávání a zobrazení výsledků - po nalezení uspokojivého řešení, nebo vyčerpání stromu řešení je výpočet ukončen a výsledky jsou graficky interpretovány

3.4 Navržený způsob generování pozic k umístění předmětů

Jak bylo naznačeno v úvodní kapitole, je celý algoritmus založen na zjednodušujícím principu umísťování předmětů do přesně určených pozic, jehož myšlenka byla převzata z [5]. Právě způsob, jakým jsou tyto pozice určovány, je jádrem celého algoritmu, protože prakticky definuje podobu stromu řešení, a proto musel být původní postup velmi silně modifikován.

Při hledání vhodného postupu generování pozic, na které budou předměty umísťovány, musíme přitom nalézt kompromis mezi následujícími dvěma protichůdnými požadavky:

1. Pozice musí vznikat v dostatečném počtu, aby mohl být objem kontejneru dostatečným způsobem využit.
2. Pozic musí vznikat co možná nejméně, aby rozsáhlost stromu řešení zbytečně nerostla.

První požadavek je pochopitelný. Pokud bychom nebyli omezeni výpočetním výkonem, bylo by jistě nejlepší vygenerovat plnou síť bodů pokrývajících v požadované hustotě celý vnitřní prostor kontejneru a jednoduše vyzkoušet vhodnost každého jednoho z nich pro každý umísťovaný předmět. Pak bychom jistě našli optimální řešení. Výpočetní výkon však vždy bude omezený a tak je třeba hledat kompromisní řešení.

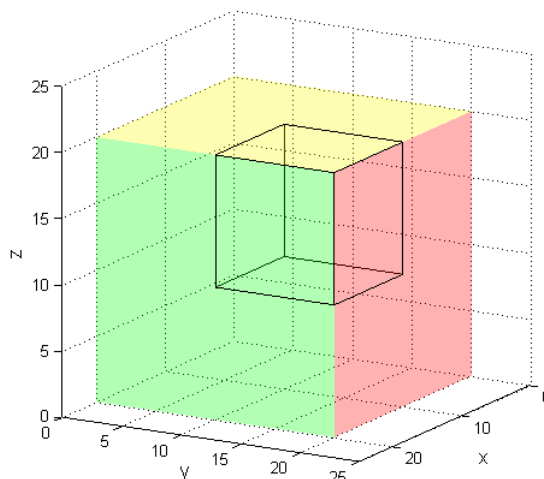
Druhý požadavek vychází právě z vědomí, že výpočetní výkon je omezený a je tedy nutné strom řešení efektivním způsobem omezovat tím, že budeme uvažovat jen takové pozice, které s vysokou pravděpodobností položí základ kvalitním řešením.

V následujících podkapitolách je posán postup generování pozic pro umístění, založený na vytváření imaginárních řezů prostorem rozmístěného nákladu. Postup

vychází z myšlenky, že nejnadějnější pozice pro umístění dalších předmětů vznikají v průsečících rovin definovaných již umístěnými předměty nebo stěnami kontejneru. Přitom je užit princip postupné úpravy množiny pozic s pomocí dvou operací-rozšíření o nově vzniklé pozice a odstranění dále nevyužitelných pozic.

3.4.1 Generování nových pozic

Jak bylo již výše naznačeno, je systém generování bodů založen na základní myšlence, že nové pozice pro umístění předmětů vznikají vždy jen v definovaných rovinách, přesněji řečeno v jejich částech. Tyto oblasti jsou definovány posledním umístěným předmětem, respektive třemi z jeho stěn, jak je ilustrováno na obrázku 3.4.

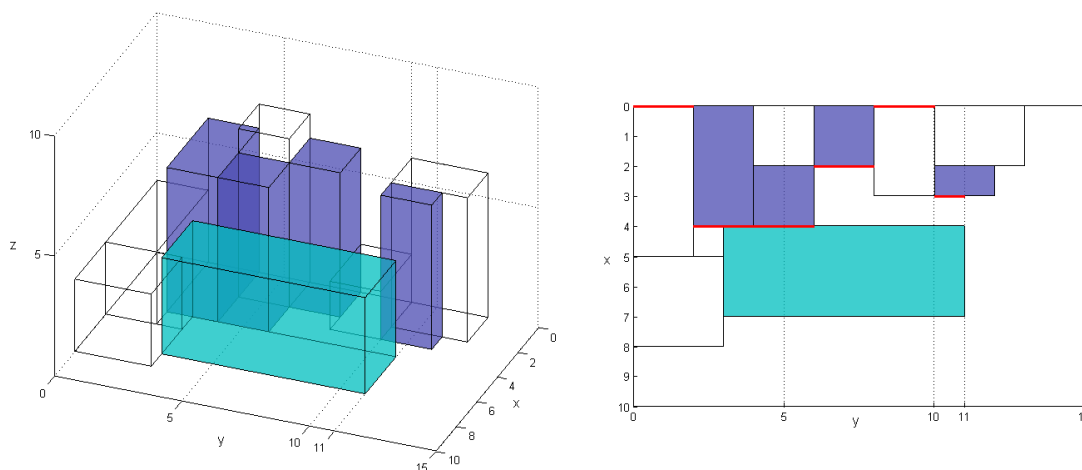


OBRÁZEK 3.4: Ilustrace oblastí, v nichž mohou vznikat nové pozice pro umístění.

Postup, jakým jsou získány nové pozice, je pro všechny tři roviny analogický a bude tedy podrobně popsán jen pro rovinu definovanou horní stěnou nově přidaného předmětu. Odlišnosti u zbylých dvou rovin jsou jen formální, vyplývající z jejich jiné prostorové orientace, a k plnému osvětlení použitého principu není jejich výčet potřebný.

Určení předmětů podílejících se na vzniku nových pozic

Prvním krokem při definici nových pozic je určení dalších předmětů, které se na jejich tvorbě budou podílet. Budou to ty předměty a stěny kontejneru, které protínají rovinu horní stěny nově umístěného předmětu a jejich čelní stěny jsou přímo viditelné ze směru plnění kontejneru. Ze znalosti těchto předmětů pak lze snadno sestavit obálku řezu nákladu v rovině horní stěny nově umístěného předmětu. Pro lepší představu je uveden ilustrační obrázek 3.5.



OBRÁZEK 3.5: Předměty tvořící obálku nákladu v řezu definovaném horní stěnou nového předmětu (světle modrý) jsou označeny modře. Předměty nepodílející se na obálce jsou bez výplně. Vpravo uvedena stejná situace v půdorysu s naznačenými relevantními úseky obálky.

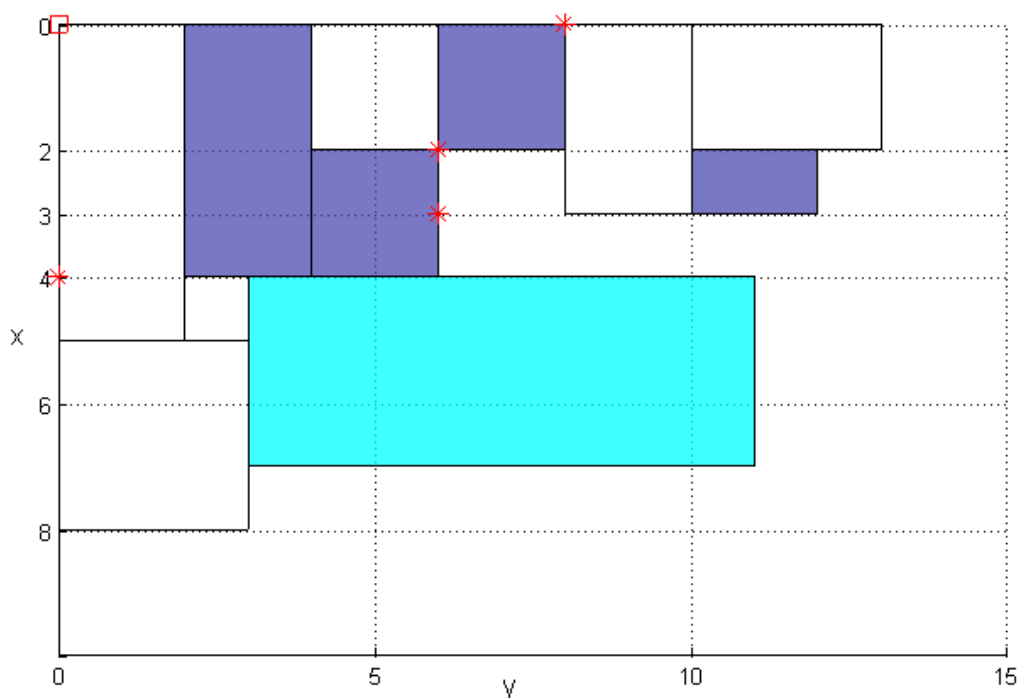
Algoritmus určení nových pozic

Znalost obálky nákladu v definované rovině nám poskytuje dostatek informací k určení poloh nových pozic. Pro tento účel byl navržen následující postup. První pozice je vytvořena v místě minimální y-ové souřadnice obálky. Následně jsou v cyklu v pořadí stoupající y-ové souřadnice procházeny úsečky tvořící obálku rovnoběžně s osou y a prováděno:

1. Pokud první z nich má vyšší x-ovou souřadnici (je dál od zadní stěny kontejneru) než druhá, je pozice pro umístění vytvořena v začátku druhé úsečky (v místě její minimální y-ové souřadnice).

2. Pokud první z nich má nižší x -ovou souřadnici (je blíže k zadní stěně kontejneru) než druhá, je pozice pro umístění vytvořena na prvním průsečíku polopřímky, vzniklé prodloužením první úsečky k levé stěně kontejneru, a obálky nákladu. V případě, že takový průsečík neexistuje, vzniká nová pozice na průsečíku polopřímky a levé stěny kontejneru.
3. Pokud y -ová souřadnice zlomových bodů právě překročila y -ovou souřadnici levé hrany nově umístěného předmětu, jsou z pozic vypočtených do této doby odebrány všechny, které mají nižší x -ovou souřadnici než první z aktuálně testovaných úseček.

Po otestování všech tří podmínek pro danou dvojici úseček pokračuje postup další dvojicí (aktuálně druhá úsečka se stává v nové dvojici první). Funkce je na příkladu ilustrována na obrázku 3.6.



OBRÁZEK 3.6: Ilustrace postupu algoritmu při generování pozic ze znalosti obálky. Bod označený čtvercem byl odstraněn v důsledku pravidla 3.

Důvod existence pravidel 1 a 2 je dostatečně zřejmý z ilustračního obrázku. S jejich pomocí jsou vytvářeny pozice na hranici stávající obálky umožňující umístění dalších předmětů na ten aktuálně umístěný. Pravidlo 3 již nemusí být na první pohled tak průhledné. S jeho pomocí jsou odstraněny pozice, z nichž není možné

využít aktuálně umístěný předmět jako oporu, což znemožňuje jejich praktické využití. Jak je z popisu patrné, skládá se tato část algoritmu z následujících na sebe navazujících operací:

1. Vyhledání předmětů tvořících obálku - asymptotická složitost maximálně n , kde n je celkový počet již rozmístěných předmětů
2. Setřídění vybraných předmětů podle y -ové souřadnice - asymptotická složitost setřídění je v nejhorším případě $n \cdot \log n$
3. Jeden průchod seznamem úseček - asymptotická složitost n

Vzhledem k tomu, že každá z operací je provedena pouze jednou je celková asymptotická složitost procesu vygenerování nových pozic $n \cdot \log n$.

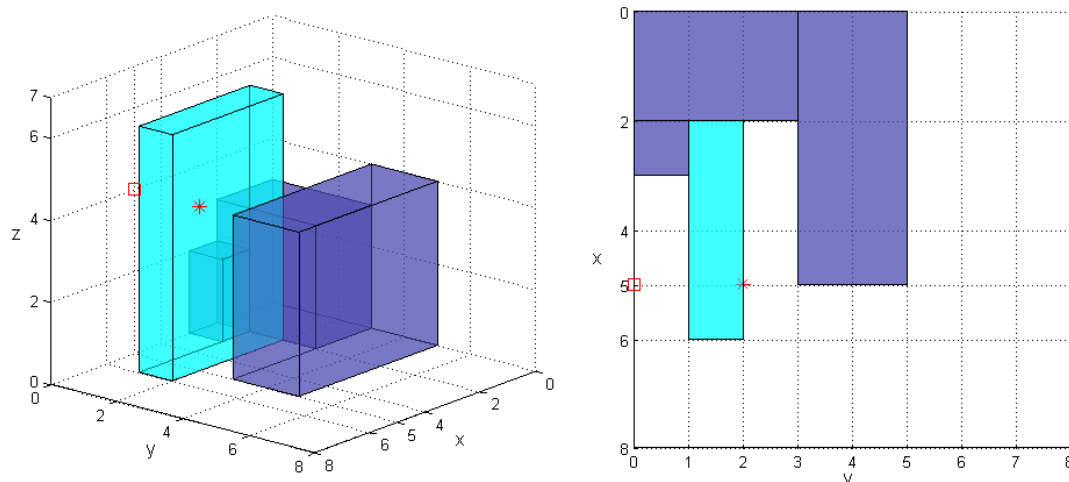
3.4.2 Modifikace existujících pozic

Umístění dalšího předmětu do nákladního prostoru má za následek nejenom vytvoření nových pozic pro umístění, ale také ovlivňuje některé z pozic vypočtených během předchozího řešení úlohy. Přitom existují v zásadě dvě možnosti podrobně rozebrané v následujících částech.

Úprava polohy existujících pozic

Z výše popsaného způsobu generování nových pozic je patrné, že tyto pozice mohou vznikat i mimo půdorys předmětu, jehož umístění bylo důvodem k jejich vytvoření. Představme si nyní situaci, kdy další předmět je umístěn mezi pozici a jí odpovídající předmět, jak je ilustrováno na obrázku 3.7.

V takovém případě je třeba její polohu modifikovat, respektive vytvořit novou pozici jak je na obrázku 3.7 naznačeno. Tento problém alespoň částečně řeší následující postup. Po umístění nového předmětu jsou ke všem předmětům z oblastí vlevo a pod novým předmětem vytvořeny jejich modifikované podoby. Polohy takto nově vzniklých pozic se od původních liší vždy pouze v jedné ze souřadnic (y , respektive z), přičemž nové pozice leží na stěně nejnověji umístěného předmětu. Všechny ostatní vlastnosti pozice včetně příslušnosti k původnímu předmětu jsou převzaty



OBRÁZEK 3.7: Princip modifikace stávajících pozic. Pozice označená hvězdičkou vznikla modifikací pozice označené čtvercem. Nově přidaný předmět je světle modrý.

od původní pozice. Původní pozice, které v důsledku umístění ztrácí smysl jsou odstraněny v následující části postupu.

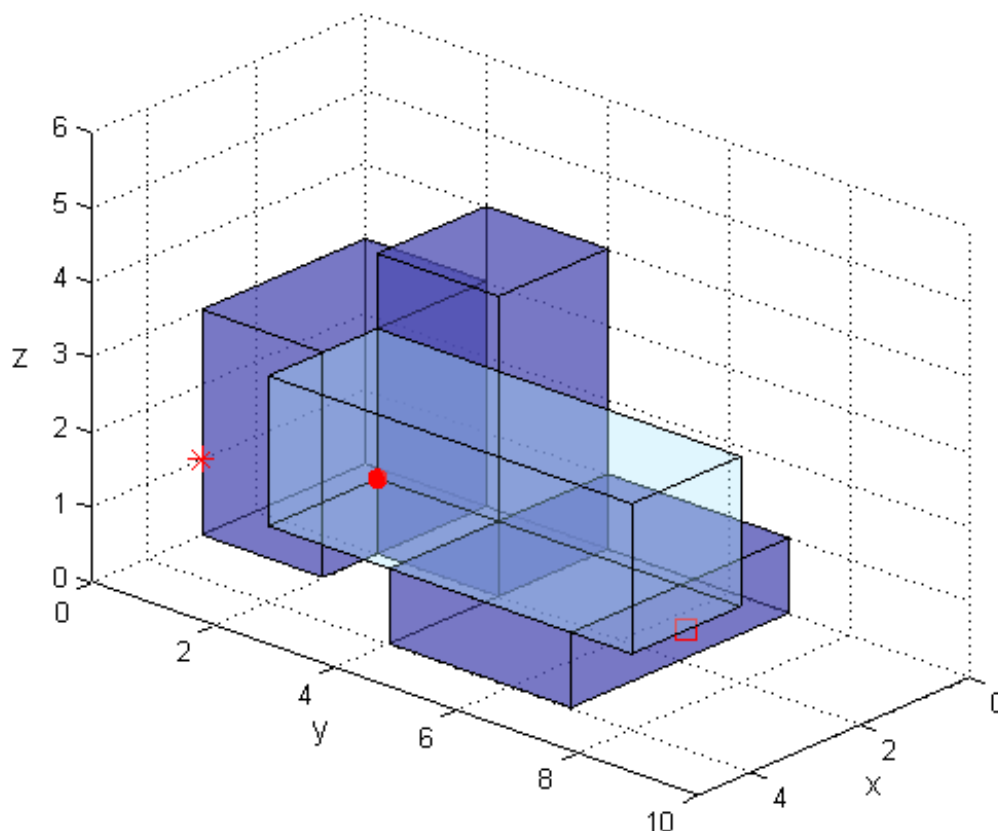
Na tomto místě je třeba poznamenat, že proces modifikace pozic znamená určitou výpočetní zátěž (asymptotická složitost jejího provedení je obdobně jako u generování $k \cdot \log k$, kde k je počet pozic pro umístění v daném uzlu stromu řešení), přičemž jeho přínos pro řešení může být poměrně nízký. Z tohoto důvodu lze (nastavením odpovídajícího parametru) tento krok při řešení úlohy vynechat.

Odstranění dále nevyužitelných pozic

V tomto kroku jsou odebrány ty pozice, které byly umístěním předmětu zcela znehodnoceny a již nebude možné je využít. Za důvod k prohlášení pozice za znehodnocenou je přitom považováno splnění jedné z následujících podmínek:

- Pozice leží na levé, zadní nebo spodní stěně nově umístěného předmětu. V takovém případě jistě nelze o budoucím využití pozice uvažovat ani v teoretické rovině.
- Y-ová nebo z-ová souřadnice pozice je vyšší než maximální odpovídající souřadnice jí příslušejícího předmětu (po jehož umístění byla vygenerována). V takovém případě by bylo další využití pozice teoreticky možné, ovšem neperspektivní z hlediska kvality řešení.

Ilustrace je uvedena na obrázku 3.8.



OBRÁZEK 3.8: Ilustrace principu odeírání znehodnocených pozic. Pozice označená červeným bodem byla po umístění světlejšího předmětu odebrána v důsledku pravidla 1, pozice označená čtvercem v důsledku pravidla 2.

Odstranění dále nevyužitelných pozic je provedeno po jednom průchodu seznamem všech pozic uzlu a jeho asymptotická složitost je tedy k , kde k je počet pozic pro umístění v daném uzlu stromu řešení.

Algoritmus určení pozic pro umístění nových předmětů jako celek je zřetězením dílčích kroků a jeho celková složitost je tedy $n \cdot \log n + k \cdot \log k$ (n -počet předmětů, k -počet pozic k umístění), což je při četnosti provádění této operace v porovnání s algoritmem prezentovaným v [5] zlepšení.

3.5 Způsob vyhodnocení korektnosti pozice předmětu

Vyhodnocení přípustnosti pozice pro umístění předmětu daných rozměrů je složeno z několika dílčích kroků. Z nich některé je třeba provést detailně, neboť jejich nesplnění se zásadně neslučuje s korektností řešení, jiné je naopak třeba vyhodnocovat pouze orientačně, neboť detailní vyhodnocení by znamenalo neúměrný nárůst celkové výpočetní doby.

Způsob kontroly korektnosti pozice pro umístění předmětu, který byl pro řešení navržen a je posán v této podkapitole, vyžaduje, díky použitým zjednodušením při vyhodnocování stability, pro svou úplnost pouze jeden průchod seznamem všech již umístěných předmětů. Jeho výpočetní náročnost je tedy lineárně závislá na velikosti umístěného nákladu, což je s ohledem na vysokou četnost jeho provádění velmi výhodné.

3.5.1 Vzájemná pozice předmětů

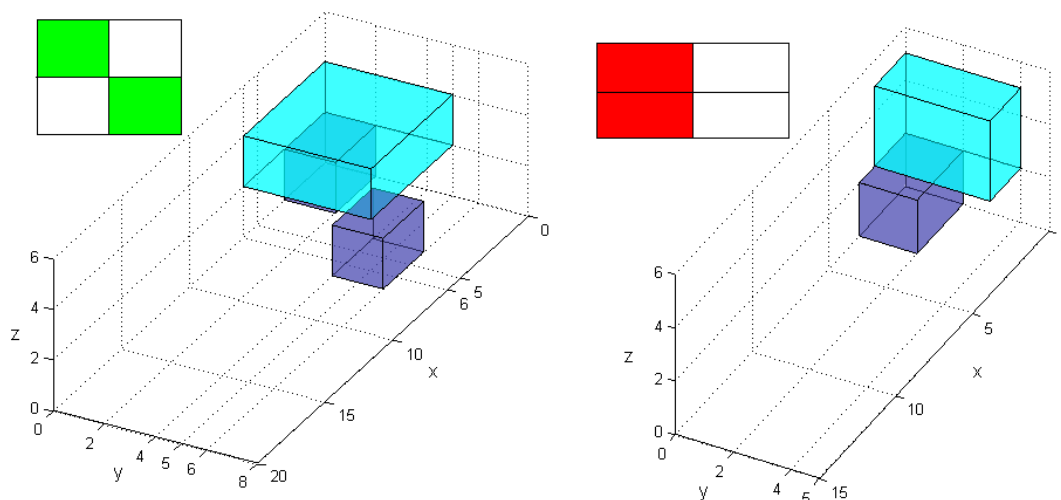
Vyhodnocení vzájemné pozice předmětů a prevence jejich kolize je základním požadavkem úlohy a musí být provedeno detailně. Algoritmus vyhodnocuje vždy pro poslední umístěný předmět, zda nekoliduje s již umístěnými předměty a také zda v některém z rozměrů nepřesahuje hranice kontejneru.

Pouhé ověření překrývání předmětů a přesahu maximální velikosti však není dostačující. Pořadí předmětů je pevně dáno a nemůžeme si tedy dovolit umisťovat předměty do volného prostoru a prostor pod nimi vyplnit až následně dalšími předměty. Dalším požadavkem na korektnost pozice je tedy nutně také kontrola skutečnosti, zda stávající obsah kontejneru poskytuje předmětu na dané pozici alespoň minimální podporu. Existence podpory je, stejně jako vyloučení kolizí, také nutnou podmínkou pro přijetí pozice jako vhodné pro umístění konkrétního předmětu a je řešena v následující části.

3.5.2 Stabilita nákladu

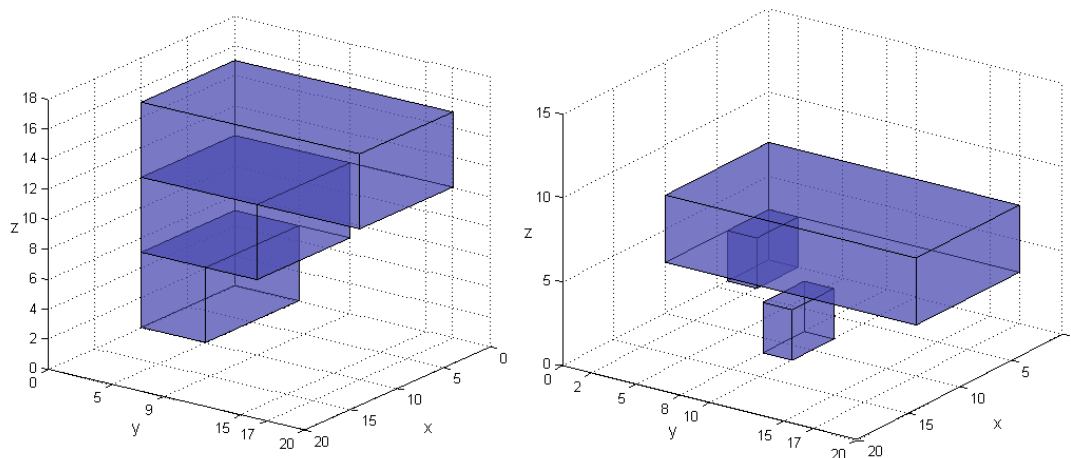
Požadavek na podporu předmětu na pozici v době umístění, je možné a také vhodné, rozšířit o test stability této pozice. Provedení důkladného vyšetření stability pozice by bylo výpočetně velmi náročné, což by při obrovské četnosti této operace během řešení úlohy ve svém důsledku vedlo k neúměrnému nárůstu výpočetní doby. Proto byl zvolen pouze přibližný postup určení stability, který lze s výhodou provést v rámci kontroly překrývání předmětů.

Tento postup je založen na kontrole opory předmětu v prostoru jeho základny a to následujícím způsobem. Základna předmětu je rozdělena na čtyři části, představující čtvrtiny její plochy. Následně je vyhodnocována přítomnost opory v jednotlivých čtvrtinách a pozice je prohlášena za stabilní, pouze pokud je opora přítomna v obou částech na jedné nebo druhé diagonále. Hranice mezi jednotlivými uvažovanými částmi přítom není počítána ani do jedné z nich. Ke kladnému vyhodnocení stability tedy mimo jiné nestačí, pokud je předmět umístěn na jiném například levou polovinou plochy své základny, jak je demonstrováno na obrázku 3.9.



OBRÁZEK 3.9: Ilustrace vyhodnocení stability s uvedeným umístěním podpory předmětu v kontrolovaných prostorech. Vlevo vyhovující, vpravo nevyhovující umístění světlejšího předmětu.

Je zřejmé, že vyhodnocení stability tímto způsobem, tedy pouze z bezprostředního okolí základny umísťovaného předmětu není úplné a neodhalí tedy všechny nestabilní stavy. Příklady situací, kdy k takovému selhání ve vyhodnocení dochází, jsou uvedeny na obrázku 3.10.



OBRÁZEK 3.10: Příklady nestabilních pozic neodhalených navrženým postupem.

Z praktických experimentů, dokumentovaných v části 5 však vyplývá, že tyto situace nastávají jen zřídka a nejsou pro praktické provedení výsledného rozmístění zásadním problémem. Možnost provést kontrolu stability v rámci kontroly překrývání, a tedy velmi rychle, tyto nedostatky jistě vyvažuje.

3.6 Kritéria výběru kandidátů pro rozvoj

Jelikož strom řešení je ve většině případů tak rozsáhlý, že jej nelze prozkoumat celý, je nutné nějakým způsobem poměřovat již dosažená řešení a podle výsledků jejich srovnání volit nejnadějnější směry dalšího postupu. Pro posouzení kvality řešení, potřebujeme definovat konkrétní kritéria. Přitom je zjevné, že kvalita těchto kritérií, nebo lépe řečeno jejich vhodnost pro konkrétní problém je zásadním předpokladem pro dosažení uspokojivých výsledků v přijatelném časovém rámci.

Pro potřeby dokumentovaného nástroje je k dispozici celá řada kritérií, jež lze rozdělit do tří skupin podle hodnocené vlastnosti. Tyto skupiny jsou podrobně rozebrány v jednotlivých částech této kapitoly. Nástroj přitom pracuje ve svých jednotlivých částech s těmi kritérii, která zvolí uživatel. V každé části lze navíc kritéria společně kombinovat a zvoleným pořadím jejich aplikace definovat jejich váhy.

3.6.1 Kritéria založená na rozměrech

Spočívají v hodnocení uzlů vývojového stromu podle prostorových vlastností nákladu nebo jeho částí. Pro všechna tato kritéria platí, že s rostoucí hodnotou kritéria klesá kvalita řešení. Velkou výhodou těchto kritérií je především rychlost jejich vyhodnocení a slouží většinou jako kritéria pomocná. Algoritmus využívá následující kritéria:

1. Maximální hodnota x-ové souřadnice roviny čelní stěny předmětů z nákladu.
2. Maximální hodnota y-ové souřadnice roviny čelní stěny předmětů z nákladu.
3. Součet x-ových souřadnic čelních stěn všech předmětů v nákladu. Toto kritérium zohledňuje polohu všech předmětů a zvýhodňuje rozmístění s předměty nejbližší u zadní stěny kontejneru.
4. Součet x-ových souřadnic čelních a y-ových souřadnic pravých bočních stěn všech předmětů v nákladu. Je rozšířením kritéria 3, přičemž jsou preferovány uzly reprezentující rozmístění předmětů co nejbližší k levé hraně zadní stěny kontejneru.
5. Součet x-ových souřadnic čelních, y-ových souřadnic pravých bočních a z-ových souřadnic horních stěn všech předmětů v nákladu. Je rozšířením kritéria 3, přičemž jsou preferovány uzly reprezentující rozmístění předmětů co nejbližší k levému zadnímu vrcholu podstavy kontejneru .

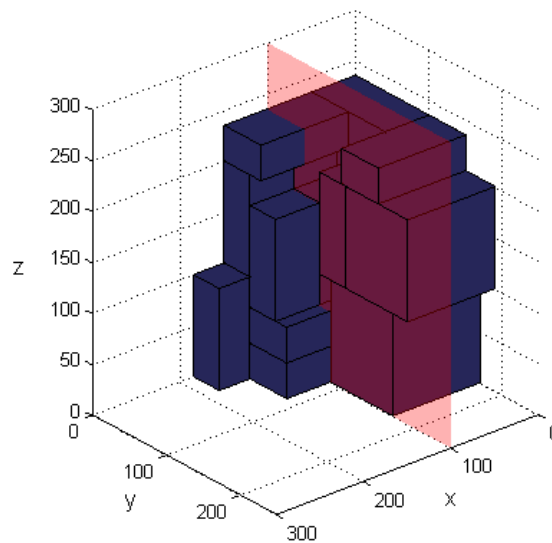
Kritéria jsou uvedena od těch nejvolnějších po ty nejkonkrétnější a to ve smyslu množství uzlů, pro něž pravděpodobně bude jedna hodnota kritéria společná.

3.6.2 Kritérium zaplněného prostoru

Toto kritérium je založeno na objemovém využití vnitřního prostoru kontejneru. Kritériem je procentuální využití definované části vnitřního prostoru nákladem, a přirozeně jsou za kvalitnější považovány uzly s vyšším procentním údajem.

Určení části pro výpočet kritéria zaplnění

Je logické, že část, ze které bude kritérium zaplnění počítáno se musí s postupným zaplňováním kontejneru nákladem přizpůsobovat. Toho bylo dosaženo na základě koncepce posouvající se imaginární hranice, dělící kontejner napříč na část pro níž je kritérium počítáno (část bližší zadní stěně kontejneru) a na část do kritéria nezahrnutou. Pro lepší představu je tato situace uvedena na obrázku 3.11. Tato



OBRÁZEK 3.11: Pro výpočet je relevantní pouze prostor za naznačenou rovinou.

hranice je pak v cyklech posouvána v závislosti na postupu plnění a to na základě splnění následujících podmínek:

- Zaplnění aktuální části pro výpočet je pro všechny kontrolované uzly vyšší než konkrétní mez, definovaná jako jeden z parametrů úlohy.
- Od posledního posunutí hranice byl umístěn potřebný počet předmětů, který je definován uživatelem na začátku řešení úlohy.

První z podmínek zaručuje posouvání hranice společně s hranicí umístovaného nákladu a to tak těsně, jak vysoká, respektive nízká, je hodnota nastaveného parametru. Druhá podmínka pak vytváří prostor pro shromáždění dostatečného počtu řešení ke vhodnému výběru kandidátů a brání následování směrů, které by mohly mít oproti ostatním lepší hodnotu kritéria pouze krátkodobě na úkor řešení

kvalitních z celkového pohledu. Zafixování pozice hranice totiž způsobuje rovnost kritéria pro vyšší počet řešení, v důsledku čehož jsou všechna kritériem přijata a krátkodobý pokles kritéria není důvodem k vyloučení uzlu.

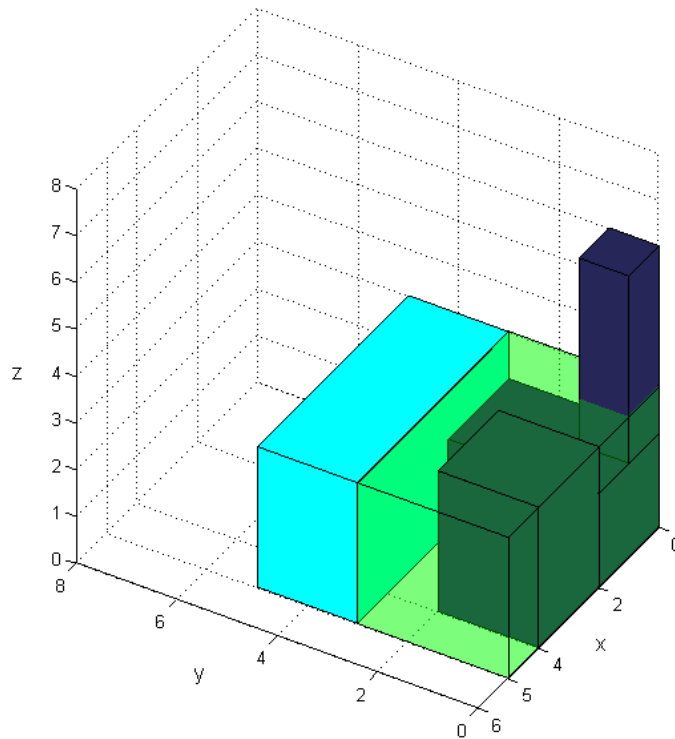
3.6.3 Kritérium volného místa

Je posledním a dá se říci hlavním z užitých přístupů k hodnocení kvality částečných řešení. Je založen na „hladovém“ přístupu založeném na předpokladu, že pokud budeme vždy umisťovat předměty tak, abychom minimalizovali ztracený objem přepravního kontejneru, bude v konečném řešení ztracený objem také nejmenší a řešení tedy nejlepší. V praxi tento zjednodušený předpoklad zřejmě do důsledku neplatí, ovšem může být velmi dobrým vodítkem.

Volné místo je určeno jako rozdíl objemu kontejneru a součtu objemu umístěných předmětů s údajem o místě ztraceném v důsledku nedokonalosti dosavadního plnění. Objem kontejneru i již rozmístěných předmětů je přitom přímo znám, ale ztracené místo je třeba definovat a určit. Detailní postup, zaručující dokonalé určení takového prostoru, by byl výpočetně neúměrně náročný a proto je třeba užít určitého zjednodušení. Použitým postupem je iterativní výpočet ztraceného místa. Vždy po přidání předmětu je vypočten objem volného prostoru za jeho zadní stěnou, který není obsazen ostatními předměty a ten je přičten k celkovému údaji o ztraceném místě. Tímto způsobem nevystávají žádné problémy s případným vícenásobným započítáváním volného místa, které by jinak hrozilo a výpočet je navíc relativně rychlý. Ilustrace je uvedena na obrázku 3.12.

3.7 Omezování stromu řešení

Omezování stromu řešení má za úkol uzavírat neperspektivní větve stromu řešení, což ve svém důsledku vede k jeho rozvoji požadovaným směrem a tedy k rychlejšímu dosažení výsledků. Základní metodou omezování stromu řešení je odebrání neperspektivních uzlů. Přitom je třeba si uvědomit, že odstraněním jednoho uzlu ze stromu řešení omezíme strom také o všechny jeho potomky. Z toho plyne, že čím blíže kořenu se odstraněný uzel nachází, tím významnějšího omezení stromu dosáhneme. Proto je snaha odhalovat a odstraňovat neperspektivní uzly co nejdříve



OBRÁZEK 3.12: Výpočet ztraceného objemu. Ten je vypočten jako rozdíl objemu zelené oblasti a objemu předmětů ležících uvnitř ní. Nově umístěný předmět je uveden světlou barvou.

je to možné. Odstraňování uzlů však není jediným použitým mechanismem omezování stromu řešení. V části 3.7.1 je popsán navržený systém rezervace objemu kontejneru, který se také podílí na omezování stromu řešení.

3.7.1 Systém rezervace prostoru

Požadavek na dodržení pořadí rozmisťovaných předmětů, přibližující úlohu k úlohám online rozmisťování, sice sám o sobě výrazně omezuje strom řešení, ovšem přináší s sebou také určité komplikace. Představme si modelovou situaci, kdy sekvence předmětů obsahuje na začátku předměty malých rozměrů a na konci předmět (předměty) nepoměrně větší. Pokud bychom v takovém případě rozhodovali o umístění předmětů pouze na základě aktuálního obsahu kontejneru a rozmisťovali je z hlediska využitého objemu optimálně, dostali bychom se snadno do situace, kdy v kontejneru sice je dostatečný objem pro umístění velkých předmětů, ale nemá odpovídající tvar a proto tyto rozměrnější předměty nelze umístit. Návrat ve stromu řešení k uzlům, jejichž potomci by umístění umožňovali, by byl

velmi problematický a časově náročný a je tedy třeba se této situace pokusit již předem vyvarovat. K tomuto účelu byl navržen způsob rezervace místa pro tyto nadměrné předměty, který využívá znalosti celé sekvence předmětů a nedovolí rozvoj stromu řešení do směrů vylučujících pozdější umístění předmětů pro něž bylo místo rezervováno.

Určení předmětů, pro něž má být místo rezervováno

Předměty, pro něž má být rezervováno místo v kontejneru, jsou ze vstupní sekvence vybírány na základě následujících tří kritérií:

1. Explicitní definice uživatelem - uživatel jako jeden ze vstupních parametrů definuje předměty, pro něž si přeje rezervovat místo.
2. Poměr velikosti předmětu ku velikosti kontejneru je v nějakém rozměru větší, než uživatelem definovaná konstanta. Například výška předmětu je větší než polovina výšky kontejneru a podobně.
3. Poměr objemu předmětu ku objemu kontejneru je vyšší, než uživatelem definovaná konstanta.

Tato kritéria jsou aplikována současně. Všechna jsou však plně využita pouze za předpokladu, že objem předmětů, pro něž má být rezervováno místo, je menší než polovina objemu kontejneru. V opačném případě jsou ze seznamu pro rezervaci postupně po jednom odebírány předměty splňující kritéria 3 a následně 2 a to až do doby, kdy je omezení na celkový objem splněno, nebo v seznamu pro rezervaci nezůstávají pouze explicitně definované předměty. Ty nejsou z listu odebrány ani v případě, že omezení není splněno.

Na tomto místě je nutné dodat, že zvláště u rozsáhlejších instancí úlohy je volba předmětů pro rezervaci místa velmi významným činitelem určujícím kvalitu dosaženého řešení a kritériím jejich volby by při řešení úlohy měla být věnována značná pozornost.

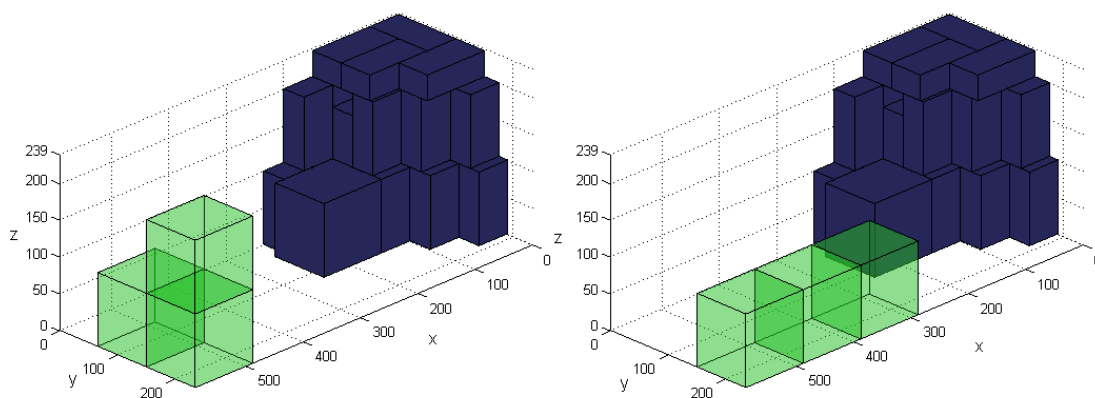
Určení tvaru rezervovaného místa

Po určení množiny předmětů, pro které má být rezervováno místo, zbývá určit jaký tvar toto místo zaujme v kontejneru. Ve své podstatě jde opět o rozmisťovací

problém, a lze tedy s výhodou použít stejný přístup jako při řešení hlavní úlohy, pouze s tím rozdílem, že při umísťování postupujeme z protějšího rohu základny kontejneru. Přitom je ale nutné brát v potaz následující:

- Předměty musí být umísťovány v opačném pořadí, než v jakém budou následně umísťovány v rámci hlavní úlohy. Dodržení tohoto požadavku zaručí praktickou využitelnost nalezeného rozmístění v rámci hlavní úlohy, protože v opačném případě by umístění prvního z rezervovaných předmětů mohlo způsobit nepřístupnost pozic pro další.
- Nalezení rezervovaného místa nesmí být moc časově náročné. Tento požadavek je splněn použitím nejjednoduššího rozmísťovacího postupu, popsáno v části 3.8.3 a je podpořen tím, že počet předmětů, pro něž má být místo rezervováno, je řádově nižší než celkový počet předmětů tvořících náklad.
- Rezervované místo musí být co možná nejkompaktnější a tak situované, aby co nejméně limitovalo umístění ostatních předmětů. Toho je dosaženo již výše zmíněnou konstrukcí z protějšího rohu (v porovnání s postupem řešení hlavní úlohy) a výběrem takového řešení, jehož celková šířka či hloubka (podle preference uživatele) je nejnižší.

Pro lepší představu je příklad rezervace prostoru v obou variantách uveden na obrázku 3.13.



OBRÁZEK 3.13: Ilustrace dvou způsobů rezervace prostoru, lišící se preferencí jedné z prostorových souřadnic rezervovaného prostoru (uveden zeleně).

Využití rezervovaného místa pro omezení stromu

Na první pohled nemusí být zřejmé, jakým způsobem se popisovaný rezervační systém podílí na omezování stromu řešení. Jeho, v řadě případů zásadní, vliv na kvalitu řešení tkví ve výrazném omezení slepých větví stromu ze znalosti budoucí sekvence předmětů. Uzly jsou tedy odstraněny daleko dříve, než kdybychom čekali až do situace, kdy se jejich dalším rozvojem větev skutečně uzavře. Toho je dosaženo zahrnutím rezervovaného místa do vyhodnocení kritéria kolize předmětů tvořících náklad. Tímto způsobem je znemožněn rozvoj ve směrech, které by v budoucnu znamenaly nemožnost umístit rezervované předměty a tudíž pokračovat ve zlepšování řešení.

Je přirozené, že rezervovaný prostor musí být v průběhu řešení upravován, respektive zmenšován o objem již umístěných předmětů, pro které bylo místo rezervováno. To je realizováno odebráním rezervovaného objemu pro konkrétní předmět v okamžiku, kdy je tento na řadě v řešení hlavní úlohy. Tím je zaručeno, že pro každý rezervovaný předmět bude vždy existovat alespoň jedna vyhovující pozice k umístění (ta, která pro něj byla rezervována).

Na závěr je třeba uvést, že rezervované místo slouží jen jako jakási pojistka proti rozvoji slepých větví, ale není striktním předpisem pro umístění odpovídajících předmětů. Jinými slovy, rezervované místo vždy musí zůstat volné až do chvíle, kdy má být umístěn jemu odpovídající předmět, ale nijak neovlivňuje, kam bude tento předmět skutečně umístěn. Pokud tedy existuje vhodnější místo, je toto využito a původně rezervované místo zůstává volné pro využití dalšími předměty.

3.7.2 Podmínky odstranění uzlu

Stanovení podmínek po jejichž splnění je uzel prohlášen za neperspektivní je velmi důležitým aspektem činnosti algoritmů založených na metodě větví a mezí. Přitom existují v zásadě dva přístupy.

- Konzervativní - uzly jsou odstraněny až poté, kdy není ani teoretická šance na to, že by jejich rozvojem mohlo být vylepšeno nejlepší stávající řešení. Tento přístup vylučuje možnost, že větev stromu jeví se jako neperspektivní bude odstraněna, ačkoliv by se jejím dalším rozvojem později ukázalo, že vede k vylepšení řešení.

- Agresivní - uzly jsou odstraněny, pokud jsou podle zvolených kritérií označeny za neperspektivní, i když stále může existovat teoretická možnost, že jejich rozvojem bude stávající nejlepší řešení vylepšeno. Tento přístup má tu výhodu, že v daleko větší míře omezuje rozsah stromu řešení a tedy rychlost algoritmu, ovšem přináší s sebou riziko, že při špatně definovaných kritériích může být kvalita řešení v důsledku odstranění velkého počtu uzlů výrazně snížena.

V navrženém algoritmu je používán konzervativní přístup, kdy je uzel odstraněn pouze při splnění jedné z následujících podmínek znemožňujících jeho další využití.

- Již neexistuje volná pozice dostatečně vzdálená od hranic kontejneru na to, aby na ni mohl být umístěn další předmět v pořadí. Tento bod je výrazně zjednodušen, neboť detailní vyhodnocení existence pozice, kam je možné předmět daných rozměrů umístit by vyžadoval uvažování již umístěného nákladu, což by bylo výpočetně velmi náročné. Toto zjednodušení ovšem nemůže vést k odebrání žádného ani teoreticky nadějného uzlu a vede pouze k nedokonalosti v prořezání stromu řešení.
- Objem přepravního prostoru, který není obsazen, ale je pro další umístování předmětů nevyužitelný (výpočet objemu tohoto prostoru byl popsán v části 3.6.2), je tak velký, že neumožňuje dosažení lepšího než již dříve dosaženého výsledku.

Výše popsané principy odstranění uzlů jsou základem, použitým za každých okolností. V různých variantách přístupu k procházení stromu řešení, jsou pak doplněny o další, agresivnější, které jsou popsány v rámci popisu jednotlivých přístupů v části 3.8.

3.8 Postupy rozvoje stromu řešení

Navržený nástroj pracuje se třemi odlišnými přístupy řešení úlohy. Jednotlivé varianty jsou vždy modifikací metody větví a mezí, všechny jsou založeny na principech popisovaných v předchozích částech práce, a liší se zejména agresivitou při prořezávání stromu řešení a také systémem jeho rozvoje. Jednotlivé modifikace jsou popsány v následujících podkapitolách.

3.8.1 Úplné prohledání

Je nejpodrobnějším přístupem, vyžadujícím dlouhý výpočetní čas, ovšem zaručujícím úplné prohledání stromu řešení a tedy nalezení nejlepšího možného výsledku.

Strom řešení je v tomto případě rozvíjen do šířky. Rozsáhlost stromu řešení neumožňuje provádět rozvoj do šířky po celých úrovních, a vždy je tedy rozvinut určitý počet uzlů (implicitně 250). Zbývající uzly jsou uloženy a rozvinuty později. Jediným způsobem omezování stromu řešení, který je v tomto případě použit je konzervativní prořezávání v podobě uvedené v části 3.7.2, což zaručuje prozkoumání všech korektních uzlů stromu řešení. Podmínkami ukončujícími řešení jsou:

- Strom řešení již nemá žádný uzel.
- Vypršel určený maximální čas pro řešení.
- Byly rozmístěny všechny předměty vstupní sekvence.

Je evidentní, že takovýto postup je nesmírně výpočetně náročný a ačkoliv zajišťuje nalezení nejlepšího možného řešení, kterého je navržený nástroj schopen dosáhnout, je v praxi použitelný jen velmi omezeně. Jeho použitelnost je omezena na problémy nejmenšího rozsahu (maximální rozsah problému přibližně 10 až 15 předmětů). Jeho význam je tak především demonstrativní, a umožňuje ilustrovat obrovský nárůst velikosti prostoru řešení v reakci na nevelké zvýšení rozsahu problému.

3.8.2 Prohledávání po částech

Tento postup je navržen pro řešení instancí středního rozsahu (několik desítek až stovky předmětů) a je určitým kompromisem mezi postupy z bodů 3.8.1 a 3.8.3. Struktura tohoto postupu je poněkud složitější a bude popsána ve dvou samostatných částech.

Princip postupu stromem řešení

Jak naznačuje název postupu, jedná se o přístup, kdy je strom řešení procházen po částech, kterými jsou úseky hloubky stromu. Rozdělením stromu řešení na úseky

definované maximální hloubkou prohledávání, využíváme myšlenky principu „rozděl a panuj“, kdy původní úlohu dělíme na podúlohy menšího rozsahu, které jsou mnohem snáze řešitelné. V pseudokódu, lze strukturu postupu zapsat následovně:

```
nalezení orientačního řešení
while (strom řešení není prázdný)
  for (i=1:počet částí)
    while (nejsou splněny podmínky ukončení prohledávání části)
      rozvoj řešení v rámci aktuální části
    end
    výběr vhodných kandidátů na kořeny podstromu pro další část
  end
end
```

Myšlenka postupu je tedy následující. Nejprve s pomocí rychlého algoritmu popsaného v části 3.8.3 nalezneme jedno řešení hlavní úlohy. Toto řešení následně slouží pro určení parametrů pro odebírání uzlů ze stromu řešení, dokud není hlavním postupem dosaženo řešení lepšího.

Po provedení tohoto přípravného kroku rozvíjíme strom řešení znovu od začátku ale nyní po částech, vždy pouze do určité hloubky. K další části přecházíme až v momentě, kdy dosáhneme uspokojivého řešení v rámci dané části (přesnému postupu rozvoje v rámci jedné části a podmínkám jeho ukončení je věnován samostatný odstavec). Po nalezení takového řešení vybereme podle zvoleného kritéria nevelký počet uzlů, jež jsou řešením podúlohy, a ty prohlásíme za kořeny stromu řešení následujícího podproblému. Tento postup vede k rozvoji stromu, který je ve velmi zjednodušeném případě uveden na obrázku 3.14. Tímto způsobem jsme schopni postupovat stromem řešení směrem k jeho listům neporovnatelně rychleji, než při úplném prohledávání, neboť prohledávaný stavový prostor je výrazně omezen, ovšem zachováváme poměrně vysokou pravděpodobnost nalezení nejlepší cesty, neboť prohledaná část stavového prostoru (byť neporovnatelně menší ve srovnání s jeho celkovou velikostí) není nutně situována v jedné části stromu řešení a může se rozprostřít po celé jeho šířce.

Na tomto místě je třeba uvést, že rychlost prohledávání společně s podrobností je závislá na velikosti částí, na které je úloha rozdělena a podrobnosti jejich prohledávání. Oba tyto parametry jsou v navrženém nástroji volitelné a jejich nastavení a tedy preference kvality řešení nebo rychlosti je ponecháno na uživateli.


```
    if (podstrom není prázdný)
        výběr kandidátů pro další rozvoj a uložení ostatních
    end
end
nahrání uzlů z nevyšší úrovně stromu řešení části
end
```

V rámci každé části postupujeme tedy následujícím způsobem. Z uzlů tvořících kořeny stromu vybereme určený počet uzlů, které jsou rozvinuty a zbytek je uložen. Z potomků je opět vybrán pouze určitý počet nejlepších, kteří jsou dále rozvinuti a zbytek je opět uložen pro pozdější využití. Strom řešení je tedy vždy tvořen jen uzly ze stejné úrovně (reprezentující rozmístění stejného počtu předmětů) a jeho šířka je limitována což podporuje rychlost postupu směrem k umístění dalších předmětů. Tímto způsobem pokračujeme, dokud není dosaženo hranice podúlohy, nebo dokud nevyčerpáme strom řešení.

Pokud je dosaženo hranice podúlohy, jsou uzly na této hranici uloženy jako řešení podúlohy, aby mohly být využity pro sestavení kořenů stromu řešení v následující části a strom řešení aktuální části je vymazán.

Pokud je strom řešení části prázdný, je obnoven z uzlů, které byly uloženy v průběhu řešení podúlohy. Přitom jsou za základ nového stromu vybrány vždy uzly, které byly uloženy nejdříve, tedy ty reprezentující rozmístění nejmenšího počtu předmětů. Tímto způsobem je vytvořen tlak na prohledání co nejširší oblasti stavového prostoru podúlohy.

Řešení podúlohy je ukončeno, pokud je splněna jedna z následujících podmínek:

1. Byl nalezen dostatečný počet uzlů, představujících úplné řešení podproblému. Implicitně je tento počet nastaven na 7500.
2. Počet obnovení stromu řešení podúlohy (nahrání uložených uzlů) přesáhl definovanou mez. Ta je implicitně nastavena na hloubku podproblému.
3. Byly rozmístěny všechny předměty vstupní sekvence.

Prarametry první dvojice ukončovacích podmínek mohou být uživatelem změněny, čímž je umožněno zvýšit nebo snížit důraz na důkladnost prohledání prostoru řešení podproblému.

3.8.3 Rychlé prohledávání

Jak sám název napovídá, jde o nejrychlejší variantu postupu stromem řešení. S rychlostí jde ovšem ruku v ruce také velmi agresivní způsob prořezávání, způsobující že výsledky tohoto postupu jsou předem velmi těžko odhadnutelné a silně závislé na vhodnosti zvoleného kritéria.

Rychlého postupu patří stromu řešení, reprezentujícími počet rozmístěných předmětů, je dosaženo rozvojem stromu striktně do hloubky a jeho velmi agresivním omezováním. Agresivita omezování stromu tkví v tom, že v implicitním nastavení jsou po rozvinutí uzlu do stromu řešení zařazeni pouze potomci, mající nejlepší hodnotu kritéria a ostatní jsou zahozeni. Agresivitu omezování stromu lze ovšem poněkud zmírnit parametrem minimálního počtu potomků k zachování. Podmínkami ukončujícími řešení jsou:

- Strom řešení již nemá žádný uzel.
- Vypršel určený maximální čas pro řešení.
- Od posledního vylepšení dosaženého výsledku byl rozvinut určený počet uzlů.
- Byly rozmístěny všechny předměty vstupní sekvence.

Je zřejmé, že tento „hladový“ přístup je sice rychlý, ovšem protože prochází jen velmi úzký úsek stromu řešení, dosahuje v řadě případů špatných výsledků. Jeho využití je tedy vhodné zejména pro řešení úloh velkého rozsahu čítající řádově několik stovek předmětů, nebo jako orientačního výpočtu či doplňku k jinému postupu.

Kapitola 4

Implementace nástroje

Nástroj byl implementován s pomocí prostředí Matlab společnosti MathWorks ve verzi R2007b. S ohledem na vlastnosti prostředí Matlab, byla nejčastěji použita maticová forma reprezentace proměnných. Jejich celkový výčet a konkrétní vlastnosti jsou detailně popsány v dokumentaci kódu na přiloženém CD. Z hlediska uživatelského prostředí je nástroj koncipován jako konzolová aplikace, ačkoliv poskytuje několik funkcí pro grafickou interpretaci výsledků a postupu řešení. V této kapitole bude osvětlena struktura nástroje z programátorského hlediska a ve stručnosti popsány hlavní nastavitelné parametry řešení.

4.1 Vstupy nástroje

Nástroj má dva vstupy. Prvním je sekvence předmětů, definovaná maticí. Přitom každý sloupec této matice má deset řádků a zcela definuje všechny vlastnosti jednoho předmětu. Jednotlivé řádky mají následující význam:

1. X-ová souřadnice umístění předmětu v daném uzlu řešení. Ve vstupní matici je nulová.
2. Y-ová souřadnice umístění předmětu v daném uzlu řešení. Ve vstupní matici je nulová.
3. Z-ová souřadnice umístění předmětu v daném uzlu řešení. Ve vstupní matici je nulová.

4. Velikost předmětu v rozměru souřadnice x (hloubka).
5. Velikost předmětu v rozměru souřadnice y (šířka).
6. Velikost předmětu v rozměru souřadnice z (výška).
7. Společně s osmým řádkem definuje povolené rotace předmětu. Detailně je popsáno v dokumentaci.
8. Společně se sedmým řádkem definuje povolené rotace předmětu. Detailně je popsáno v dokumentaci.
9. Pozice pro uložení pomocných informací. Je popsána v dokumentaci.
10. Pořadí předmětu v sekvenci

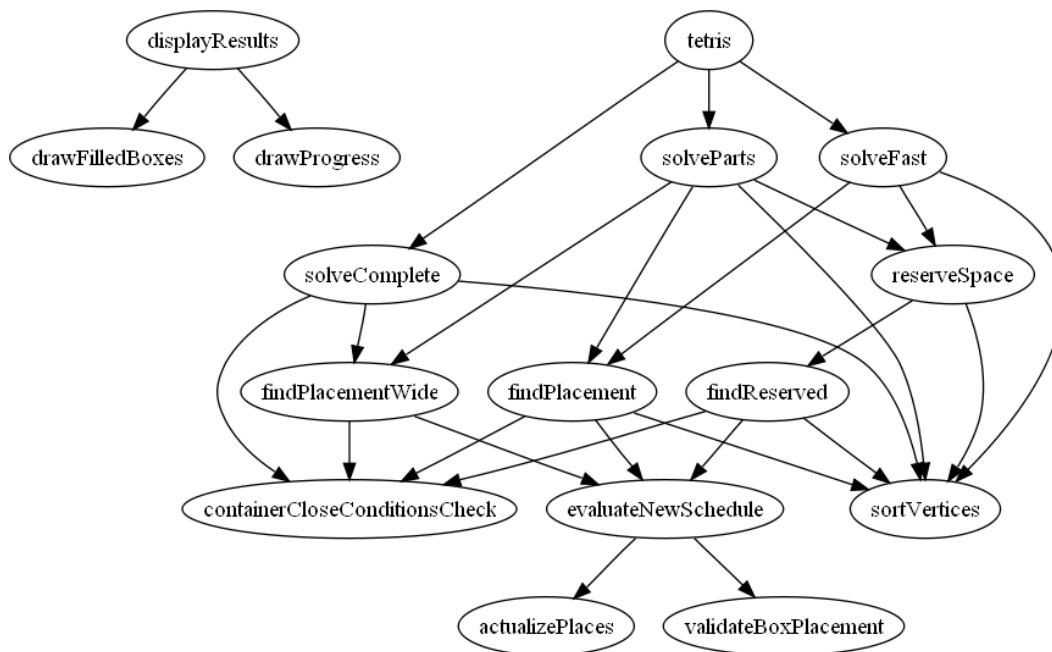
Druhým vstupem jsou rozměry kontejneru, definované jako vektor [hloubka × šířka × výška].

4.2 Struktura nástroje a hlavní funkce

S ohledem na to, že se jedná o první a do značné míry experimentální podobu nástroje, má v implementační rovině podobu skupiny volně svázaných samostatných funkcí. Tato struktura je vhodná pro svou přehlednost a snadnou možnost úpravy činnosti nástroje změnou, nebo úplným nahrazením jednotlivých funkcí bez nutnosti úpravy ostatních částí. Struktura hlavních funkcí a jejich vazeb je naznačena na obrázku 4.1. Z ilustračního obrázku je patrné, že centrálním bodem celého nástroje je skript `tetris`. Ten vytváří základní strukturu algoritmu, jsou zde definovány vstupy, nastaveny parametry řešení a následně volány odpovídající funkce. Vlastnosti jednotlivých funkcí jsou uvedeny v dokumentaci na CD.

Vstupní sekvence předmětů je nahrávána z podadresáře `boxes`, kam je třeba ji předem uložit pod názvem `boxes` ve formě `.mat` souboru. Je možné zadat několik vstupních souborů najednou a tyto jsou pak v příslušném pořadí postupně řešeny bez nutnosti dalších zásahů.

Při řešení je z důvodu obrovského rozsahu stromu řešení nutné ukládat jeho části, které aktuálně nejsou používány, na pevný disk. K tomuto účelu slouží podadresář



OBRÁZEK 4.1: Vztahy závislosti mezi hlavními funkcemi. Pro přehlednost byly vynechány méně důležité funkce. Komplettní struktura je uvedena v dokumentaci.

levels, který je v případě prohledávání po částech dále dělen na adresáře odpovídající úsekům hloubky stromu. Při každém novém spuštění procesu řešení je obsah tohoto adresáře smazán.

Výsledky řešení úlohy jsou koncentrovány do proměnné *result* a uloženy do podadresáře *results* jako *.mat* soubor s názvem ve formě *ResultNázevSouboruDefinujícíchPredmety.mat*. Ukládány jsou následující informace:

- Nejlepší dosažená řešení
- Čas potřebný k získání dosaženého řešení
- Výsledek přípravného kroku a jeho výpočetní čas
- Počet uzlů, které byly během řešení vytvořeny
- Údaj o rezervovaném místě pro potřeby rekonstrukce postupu řešení

4.3 Hlavní nastavitelné parametry a jejich význam

Jak je patrné z obsahu kapitoly 3, poskytuje nástroj možnost nastavení celé řady parametrů, s jejichž pomocí lze stanovit priority co se týče rychlosti nalezení řešení, důkladnosti prohledání stavového prostoru a použitých kritérií. Všechny tyto parametry jsou, jak bylo již výše uvedeno, nastavovány v hlavním skriptu nazvaném *tetris*. Následuje seznam těchto parametrů a jejich význam.

- *CONTAINER_SIZE* - Definuje rozměry kontejneru v cm. Implicitně je nastavena hodnota $[589\ 235\ 239]$ odpovídající kontejneru podle normy ISO20.
- *BOX_PREFERENCE* - parametr určující jakým rotacím bude dáвана přednost při rozvoji v případě rovnosti kritériální funkce. Možné hodnoty jsou *minX*, *minY*, kdy je preferována rotace s nižším x-ovým, respektive y-ovým, rozměrem předmětu. Implicitně je nastavena hodnota *minY*.
- *PLACES_PREFERENCE* - parametr určující jaké pozice budou preferovány pro umístění dalšího předmětu. Možné hodnoty *x*, *y*, *xy*, *xyz* znamenají po řadě preferenci pozic s nižší x-ovou souřadnicí, y-ovu souřadnicí, součtem obou souřadnic a součtem všech tří prostorových souřadnic. Implicitně je nastavena hodnota *xy*.
- *RESERVE_PREFERENCE* - parametr určující preferované prostorové uspořádání rezervovaného místa. Možné hodnoty *minX*, *minY* znamenají po řadě uspořádání k čelní a pravé stěně kontejneru. Implicitní hodnota je *minX*.
- *MOVESLICE_FILLING* - určuje hodnotu procentního objemového zaplnění části kontejneru, při jejímž dosažení všemi nově rozvinutými uzly je posunuta hranice části pro výpočet plnění. Implicitní hodnota je *80*.
- *MODIFY_PLACES* - definuje operaci modifikace stávajících pozic pro umístění po přidání nového předmětu. Hodnoty *move*, *remove*. V případě hodnoty *move* je proveden krok modifikace stávajících pozic, v druhém případě je provedeno pouze odstranění pozic dále nevyužitelných.
- *MAX_INPUT_DEPTH* - určuje omezení na maximální hloubku, do které lze vkládat předměty přes ty již umístěné. Hodnoty jsou v rozsahu 1 až hloubka kontejneru. Implicitní hodnotou je hloubka kontejneru, tedy žádné omezení.

- *stepResultLength* - určuje počet uzlů, které jsou současně rozvinuty v jednom kroku prohledávání do šířky a stávají se základem nové úrovně stromu. Implicitní hodnota je 5.
- *parts* - definuje počet částí, na které bude při prohledávání hloubka stromu rozdělena (viz část 3.6.2). Implicitní hodnota je 2.
- *returns* - určuje, kolik návratů na začátek bude v každé části při prohledávání provedeno. Implicitní hodnota je 1.
- *treeEvaluationsMethod* - slouží k explicitní definici metody, která má být k řešení použita. Hodnoty 0, 1, 2, 3 značí v pořadí volbu podle velikosti problému, úplné prohledání, prohledání po částech a rychlé prohledání.
- *definedPositions* - slouží k explicitní definici předmětů, pro které má být rezervován prostor. Implicitně je seznam prázdný.
- *reserveConditions* - definuje podmínky po jejichž splnění je pro předmět rezervováno místo. První složka určuje podmínku na velikost, druhá na objem předmětu. Podrobně je popsáno v dokumentaci.
- *maxTime* - určuje maximální dobu výpočtu. po jejím uplynutí je výpočet ukončen. Implicitní hodnota je 1800 sekund.
- *boxesToChangeSlice* - definuje nejnižší počet předmětů, které je třeba umístit mezi dvěma změnami velikosti části pro níž je určováno objemové zaplnění jako kritérium kvality. Implicitní hodnota je 5.
- *saveFullTree* - má význam požadavku na ukládání celého vývojového stromu na pevný disk. Hodnoty 0, 1 v pořadí znamenají: ukládat pouze první úroveň v každé části; ukládat celý strom. Ukládání celého stromu je velmi časově i paměťově náročné a proto je implicitně nastavena hodnota 0
- *criteriaDepth* - definuje typ kritériální funkce při rozvoji stromu řešení do hloubky. Hodnoty viz dokumentace. Implicitní nastavení je *'minSpaceLost'*, *'minXYSum'*.
- *criteriaWide* - definuje typ kritériální funkce při rozvoji stromu řešení do hloubky. Hodnoty viz dokumentace. Implicitní nastavení je *'filledRatio'*, *'yes'*, *'minXSum'*, *'no'*.

- *criteriaEndPart* - definuje typ kritériální funkce, která je aplikována při výběru kandidátů pro další rozvoj při vyhodnocení výsledků řešení části stromu. Hodnoty viz dokumentace. Implicitní nastavení je *'availableSpace'*, *'yes'*, *'minXSum'*, *'yes'*.
- *acceptableFilling* - definuje hranici objemového plnění, při jejímž dosažení v přípravném kroku je uživatel dotázán, zda ve výpočtu pokračovat. Implicitně je nastaveno *95*.

Kapitola 5

Výpočetní experimenty

V této kapitole jsou uvedeny experimenty dokumentující výsledky implementovaného nástroje při řešení praktických problémů a je diskutován vliv hlavních parametrů na kvalitu a rychlost řešení.

Všechny experimenty byly provedeny na osobním počítači s procesorem Intel Centrino™ 1.7GHz s operační pamětí velikosti 512MB.

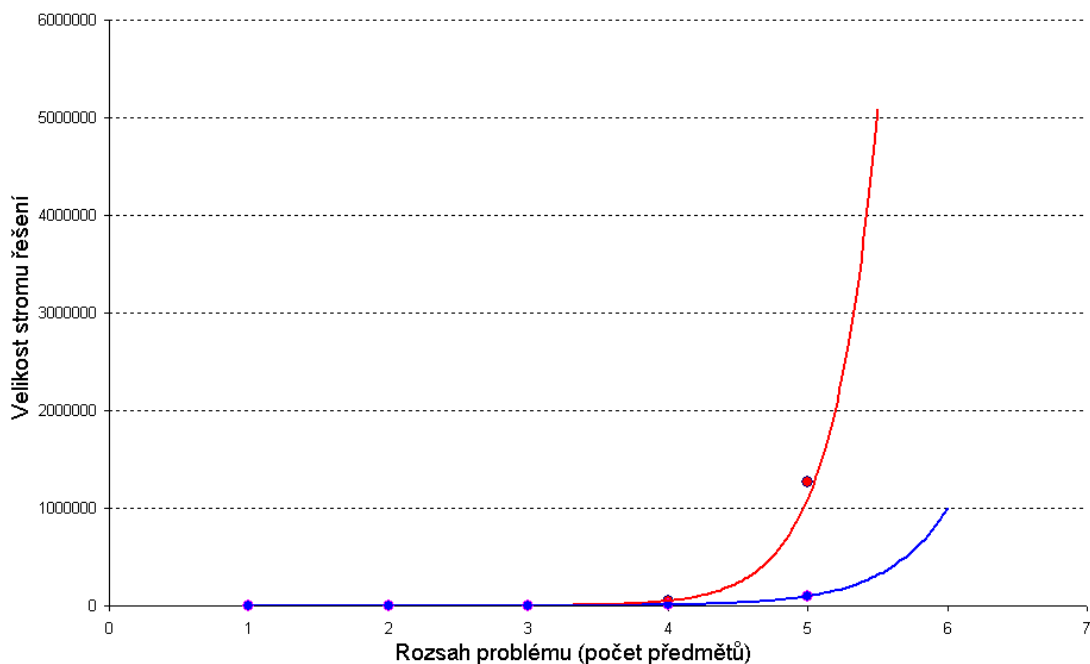
5.1 Velikost stavového prostoru

Cílem této části je demonstrovat obrovský nárůst velikosti stavového prostoru v důsledku nevelkého zvýšení rozsahu problému. Závislost velikosti stromu řešení je uvedena v tabulce 5.1. Hodnoty byly získány hrubou silou, rozvojem kompletního stromu řešení pro úlohu rozmístění předmětů velikosti [50cm×40cm×24cm] (sekvence předmětů uložena na přiloženém CD pod názvem *banany.mat*) do kontejneru velikosti [589cm×235cm×239cm] (odpovídá rozměrům standardního kontejneru podle normy ISO20). Zahrnuta jsou pouze korektní řešení, včetně částečných.

Rozsah problému (počet předmětů)	Počet uzlů stromu řešení
1	6
2	100
3	2 061
4	47 325
5	1 273 913

TABULKA 5.1: Závislost velikosti stromu řešení na rozsahu problému

Z tabulky je patrné, že stavový prostor roste skutečně velmi strmě. Pro rozvoj celého stromu problému rozsahu pěti předmětů, trval výpočet již několik desítek minut a řešení praktických úloh hrubou silou lze tedy vyloučit. Ještě názorněji je charakter závislosti zřejmý z obrázku 5.1.



OBRÁZEK 5.1: Závislost velikosti stromu řešení na velikosti problému. Pro názornost je uvedeno srovnání s funkcí 10^n (uvedena modře) a naznačen trend vývoje.

Z grafu je již jasně patrné, že závislost velikosti stromu řešení na velikosti problému je skutečně exponenciální a tedy nepolynomiální. Tento fakt je v souladu se zařazením úlohy do třídy složitosti NP a je hlavní příčinou obtížnosti řešení úlohy.

5.2 Význam rezervace prostoru

Cílem této části je ukázat význam systému rezervace prostoru pro volbu vhodného směru postupu stromem řešení. Následně jsou nastíněny také problémy související se špatnou volbou podmínek rezervace.

Pro demonstraci významu rezervačního systému byla využita sekvence předmětů s následujícími vlastnostmi. Sekvence je tvořena předměty náhodných rozměrů omezených shora čtvrtinou odpovídajícího rozměru kontejneru. Mezi tyto předměty

je pak do náhodně vybraných pozic vloženo šest předmětů „nadměrných“. Tato sekvence je uložena na příloženém CD pod názvem *setReservace.mat*. Výsledky rozmístění při použití rezervačního systému a bez něj jsou uvedeny v tabulce 5.2. Pro oba případy byly, kromě rezervace, užity stejné parametry řešení.

	S rezervací	Bez rezervace
Předmětů v sekvenci	403	403
Rozmístěných předmětů	398	299
Procentuální využití objemu	98.4	74.07
Čas pro řešení	46.13 sekund	>1800 sekund

TABULKA 5.2: Srovnání výsledků nástroje v závislosti na použití systému rezervace.

Z tabulky je patrný výrazný rozdíl v kvalitě dosaženého řešení, který použití rezervačního systému přináší. Tento rozdíl je způsoben tím, že předmět nacházející se v sekvenci na třísté pozici je v x-ovém rozměru stejně velký jako kontejner. Pokud tedy není místo pro tento předmět rezervováno, není možné jej později umístit. Strom řešení je přitom tak rozsáhlý, že se v dohledné době standardním postupem prohledávání k vhodnému směru nedostaneme, protože z hlediska zvolených kritérií se velmi dlouho jeví jako neperspektivní.

Přínos systému rezervace je evidentní, ovšem je třeba si uvědomit, že jeho dobrá funkce je velmi těsně svázána s výběrem předmětů, pro které má být místo rezervováno. Přitom nejde jen o jejich počet, ale také o jejich pozici ve vstupní sekvenci. To je demonstrováno v tabulce 5.3, kde je uveden výsledek pro sekvenci vzniklou setříděním sekvence užitá v tabulce 5.2 podle maximální velikosti hrany předmětu a podle objemu předmětu.

Střídění	Rozmístěných předmětů	Využití objemu [%]
Rozměr předmětu (vzestupně)	372	75.4
Objem předmětu (vzestupně)	337	59.23

TABULKA 5.3: Výsledky nástroje při nevhodně definovaných předmětech pro rezervaci prostoru.

Jak je patrné z tabulky, je kvalita těchto řešení zřetelně nižší. Je to způsobeno tím, že předměty pro rezervaci jsou koncentrovány všechny na úplném konci sekvence a prostor, který je pro ně rezervován omezuje umísťování předmětů v sekvenci dřívějších. V případě řazení podle objemu se pak přidává ještě nevhodné rozmístění rezervovaného místa.

Volba předmětů pro rezervaci prostoru je tedy poměrně zásadním aspektem řešení, ale její určení přímo z charakteru nákladu nebylo z důvodu složitosti do nástroje zahrnuto. Volba předmětů pro rezervaci tedy zůstává na uživateli, který ji provádí pomocí podmínek na velikost a objem předmětů a explicitní definicí pozic předmětů.

5.3 Vliv jednotlivých parametrů na řešení

Jak bylo uvedeno v kapitole 4, umožňuje nástroj uživateli ovlivnit proces výpočtu definicí celé řady parametrů. Zde bude na demonstračním příkladu předveden jejich vliv na kvalitu řešení a čas jeho dosažení.

Jako demonstrační sekvence byla zvolena posloupnost uložená na přiloženém CD po názvem *dvaTypy.mat*. Ta obsahuje náhodně seřazené předměty rozměrů [100cm×50cm×35cm] a [85cm×100cm×100cm]. Poměr zastoupení je přitom 9:1 a velikost kontejneru [589cm×235cm×239cm]. Následuje tabulka 5.4 s uvedenými dosaženými výsledky pro různá nastavení. Neuvedené parametry mají vždy hodnoty uvedené v kapitole 4 jako výchozí.

Z tabulky je patrné, že nejvýznamnějším parametrem z hlediska kvality řešení je bezpochyby zvolené kritérium ohodnocování částečných řešení, případně jejich kombinace. Ostatní parametry mají z hlediska kvality řešení mnohem menší význam a slouží vesměs k drobnému vylepšení celkového řešení.

Velmi závažným parametrem, a to zejména z hlediska výpočetního času je trojice [stepResultLength, parts, returns]. S pomocí těchto parametrů uživatel určuje podrobnost prohledávání v jednotlivých částech stromu. Tento parametr má zásadní vliv na dobu řešení, jak je z tabulky jasně patrné. V případě vhodně zvoleného kritéria nevede podrobnější prohledávání částí k dramatickému zlepšení výsledku, ovšem v případě nižší vhodnosti kritéria zajistí alespoň uspokojivý výsledek.

Na závěr této části je nutné dodat, že vhodnost hodnot jednotlivých parametrů, včetně kritériální funkce, je silně závislá na charakteru nákladu a nelze tedy určit jediné ideální nastavení. Výchozí nastavení však bylo provedeno tak, aby poskytovalo uspokojivé výsledky pro co nejširší okruh úloh.

Parametr <i>hodnota</i>	Rozmístěných předmětů	Využití objemu [%]	Doba řešení [s]
PLACES_PREFERENCE			
<i>xy</i>	126	91.14	119.8
<i>x</i>	126	91.14	91.3
<i>y</i>	123	89.55	110
BOXES_PREFERENCE			
<i>minX</i>	128	92.2	95.8
<i>minY</i>	126	91.14	90.3
MOVESLICE_FILLING			
<i>25%</i>	127	91.67	120.3
<i>50%</i>	127	91.67	121.6
<i>75%</i>	126	91.14	90
<i>95%</i>	125	90.61	123.9
[stepResultLength,parts,returns]			
<i>[5,1,1]</i>	126	91.14	88.7
<i>[25,10,10]</i>	128	92.2	1337.1
criteriaWide			
<i>minXSum</i>	127	91.67	116.9
<i>minXYSum</i>	94	70.13	79.2
<i>minSpaceLost</i>	96	73.76	75.2
<i>filledRatio</i>	121	88.49	82.8
<i>filledRatio, minXSum</i>	126	91.14	90.3
<i>filledRatio, minXYSum</i>	96	73.76	92.4
<i>minSpaceLost minXSum</i>	127	91.67	117.2

TABULKA 5.4: Výsledky nástroje pro různé nastavení parametrů řešení.

Kapitola 6

Závěr

V rámci práce byl navržen nový exaktní algoritmus pro řešení specifické formulace úlohy typu *Bin Packing* ve třírozměrném prostoru. Po prostudování literatury byla jako základ algoritmu zvolena metoda větví a mezí. Prezentovaný algoritmus ideově vychází z existujícího postupu umisťování nových předmětů do pozic určených z prostorového uspořádání stávajícího nákladu v kontejneru. Postup generování těchto pozic, který je základem algoritmu, byl však nově navržen tak, aby bylo možné využít i volný prostor, který byl původním postupem neuvažován. Asymptotická složitost procesu generování pozic pro umístění předmětů je přitom oproti n^2 u původního algoritmu snížena na $n \cdot \log n$.

Navržený algoritmus řešení byl implementován v prostředí Matlab a funkčnost výsledného nástroje byla dokumentována na praktických příkladech řešení instancí o rozsahu několika stovek předmětů.

Příloha A

Obsah příloženého CD

Obsah příloženého CD je následující:

- Adresář *documentation* - obsahuje podrobnou dokumentaci nástroje, koncipovanou jako vzájemně provázané webovské stránky. Hlavním rozcestníkem je soubor *index.html*.
- Adresář *code* - obsahuje soubory implementovaného nástroje. Jeho obsah je následující:
 - Adresář *boxes* - obsahuje sekvence předmětů použité při výpočetních experimentech uvedených kapitole 5.
 - Adresář *results* - slouží k uložení výsledků řešení.
 - Adresář *levels* - slouží k uložení právě nepoužívaných částí stromu řešení během výpočtu.
 - Adresář *functions* - obsahuje funkce nástroje.
 - Skript *tetris*, který je spouštěcím skriptem nástroje
- Adresář *text* - obsahuje textové soubory práce. Soubory jsou:
 - Soubor *Text.pdf* - je elektronickou verzí textu diplomové práce
 - Soubor *Zadani.pdf* - elektronická verze originálního zadání práce
 - Soubor *Prohlaseni.pdf* - elektronická verze podepsaného prohlášení o autorství

Literatura

- [1] Paolo Toth Silvano Martello. *Knapsack Problems*. John Wiley and Sons Inc; Revised edition, 1990. URL <http://www.or.deis.unibo.it/knapsack.html>.
- [2] M. R. Garey Michael R. Garey, David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman Company, 1979.
- [3] David Pisinger. Heuristics for the container loading problem. *European Journal of Operational Research*, 141(2):382–392, September 2002.
- [4] Antonio D. Nola, editor. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*. Springer.
- [5] Daniele Vigo Silvano Martello, David Pisinger. The three-dimensional bin packing problem. *Operation Research*, 48(2):256–267, March 2000.