



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

Lina Želvytė

3D OBJEKTŲ PAKAVIMO METODAI IR PROGRAMINĖ ĮRANGA

Magistro darbas

Darbo vadovas

doc. dr. Dalius Rubliauskas

KAUNAS, 2004

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

TVIRTINU

Katedros vedėjas

doc. dr. Dalius Rubliauskas

2004-05-

3D OBJEKTŲ PAKAVIMO METODAI IR PROGRAMINĖ ĮRANGA

Informatikos mokslo magistro baigiamasis darbas

Kalbos konsultantė

Lietuvių kalbos katedros lektorė

dr. J. Mikelionienė

2004 05

Vadovas

doc. dr. Dalius Rubliauskas

2004 05

Recenzentas

doc. dr. E. Bareiša

2004 05

Atliko

IFM–8/3 gr. stud.

L. Želvytė

2004-05-25

KAUNAS, 2004

SUMMARY

The essence of 3D object packing problem is to load a set of distinct boxes with given dimensions in containers to maximize volume utilization. The goal of this research is to support with algorithmic techniques the design of package layouts that meet the functional relationships between the parts and match market needs including cost, safety, comfort, as well as economic and environmental aspects.

An analysis of 3D object packing algorithms, created by foreign authors, and commercial three-dimensional pallet packing software packages was performed in this work; also methods' advantages and disadvantages were pointed out. A model of three-dimensional rectangular object packing into containers of the same shape was formed, and the software solving 3D object packing problems was created according to this model. The tests proved that the suggested method is effective and beneficial.

TURINYS

IVADAS	5
2. ANALITINĖ DALIS	7
2.1. Trimačio pakavimo metodų analizė	7
2.2. Komerčinės 3D objektų pakavimo programos.....	13
2. PROJEKTINĖ DALIS	17
2.1. Reikalavimų dokumentas.....	17
2.2. 3D objektų pakavimo sistemos modeliai.....	20
2.3. Programinės įrangos projektavimas	22
2.3.1. Programinės įrangos bendra charakteristika	22
2.3.2. Duomenų struktūra	23
2.3.3. Projektuojamos sistemos algoritminė diagrama.....	26
2.3.4. Iteracijų paruošimas ir vykdymas.....	28
2.3.5. Išvesties duomenų atvaizdavimas.....	34
2.4. 3D objektų pakavimo programos modelio specifikacija	35
2.5. Rizikos įvertinimo ir mažinimo planas.....	38
3. VARTOTOJO DOKUMENTACIJA	40
3.1. Programinės įrangos paskirtis ir galimybės	40
3.2. Naudojimosi programa instrukcija	41
3.3. Programinės įrangos įdiegimo dokumentacija.....	43
4. BANDOMASIS EKSPLOATACINIS TYRIMAS	43
5. PRODUKTO KOKYBĖS ĮVERTINIMAS	46
6. PROGRAMINĖS ĮRANGOS TESTAVIMAS	46
IŠVADOS	50
LITERATŪRA	51
PRIEDAI	53

ĮVADAS

Su trijų dimensijų (3D) pakavimo problemomis susiduriama įvairiose pramonės srityse, kuriose yra būtinybė kuo geriau supakuoti tam tikrus objektus projektuojant juos į erdvę. Dažnai pakuojant 3D objektus yra optimaliai neišnaudojama sutalpinimo erdvė, eikvojamos papildomos laiko ir pinigų sąnaudos, dėl veiklos produktyvumo stokos gaunami lūkesčius netenkinantys darbo rezultatai.

Pakuojamiems 3D objektams keliami ir papildomi reikalavimai:

- ribojamas objektų, dedamų ant kitų viršaus, svoris;
- kai kurių objektų neleidžiama vartyti;
- reikalaujama, kad būtų pakuojamo konteinerio (autofurgono) ratų ašių svorio balansas;
- gali reikėti į konteinerį patalpinti tam tikrą kai kurios rūšies objektų kiekį;
- kartais supakuotus objektus reikalaujama iškrauti skirtingose vietose, todėl jie pakuojami tam tikra eilės tvarka, kad būtų galima juos išimti iš konteinerio reikiamu laiku.

Dažniausiai su pakavimo problemomis susiduriama stiklo, medžio ar kitų medžiagų pjaustymo, krovinių pervežimo, veiklos planavimo srityse. Norint, kad minėtose srityse būtų efektyviai sumažinti atliekų kiekiai, transportavimo kaštai ir efektyviau panaudoti turimi resursai, yra **aktualu** nagrinėti 3D objektų pakavimo metodus, tirti jų ypatumus, pritaikymo galimybes ir principus.

Šiame darbe bus nagrinėjamas stačiakampio gretasienio formos objektų (dėžių) pakavimo į tokios pačios formos konteinerį uždavinys.

Tyrimo **tikslas** yra sudaryti 3D objektų pakavimo algoritmą, sukuriant techninį pakavimo projektavimo planą, kuris užpildytų kuo didesnę konteinerio erdvės tūrį, patenkindamas funkcinius ryšius tarp dalių rinkos sąlygomis, įskaitant kainą, saugumą, komfortą, įvertinant ekonominius ir aplinkos aspektus.

Tikslui pasiekti keliami tokie **uždaviniai**:

- Išanalizuoti trimačio pakavimo metodus ir programinę įrangą.
- Atlikti duomenų analizę, parinkti duomenų struktūras.
- Parinkti programavimo priemones.
- Sudaryti programų sistemos projektą.
- Testuoti programą, bandyti jos praktinį veikimą.
- Paruošti dokumentaciją.

- Įdiegti programą.

Metodai: mokslinės literatūros analizė, programavimas.

2. ANALITINĖ DALIS

2.1. Trimačio pakavimo metodų analizė

Sprendžiant pakavimo problemas buvo atlikta daugybė tyrimų ypač vienos arba dviejų dimensijų atvejams. Šiuo metu didelis dėmesys yra skiriamas 3D objektų pakavimo problemas sprendžiančių efektyvių modelių kūrimui ir vystymui. Darbe analizuojami įvairių autorių sukurti algoritmai, kuriuos apibendrinus išskiriami jų veikimo esminiai privalumai ir trūkumai. Kadangi ne visi autoriai atskleidžia savo sukurtų algoritmų specifika, todėl nedarant eksperimentinių bandymų ir vertinant jų sukurtas procedūras tik pagal teorinę analizę, privalumai ir trūkumai išskiriami pagal jų sukurtos programos atliekamas galimybes arba algoritme vykdomas arba nevykdomas procedūras.

George ir Robinson (1980) yra pirmieji anglų mokslininkai pradėję tyrinėti objektų pakavimo problemas. Autoriai tiksliai įvardijo konteinerių pakrovimo problemą, smulkiai aprašydami įvairaus kiekio ir skirtingų dydžių dėžėmis konteinerio pakrovimo algoritmą. Jie panaudojo pakankamai sudėtingą sienų rentimo būdą, kuriame konteinerio dalys yra pilnai užpildomos atsižvelgiant į pločio ir aukščio matus. *Algoritmo idėja* – padalinti konteinerį į kelis lygiagrečius sienoms sluoksnius ir juos užpildyti atskirai. Toks būdas daugeliu atvejų garantuoja, kad taip supakuoto krovinio didesnė dalis bus stabili. George ir Robinson sukurtas algoritmas pakuoja iš 20 dėžių susidedantį krovinį. Pakuojant skirtingų dydžių dėžes ir naudojant skirtingas jų padėtis, gali būti gaunami skirtingi pakavimo variantai. Kai naudojant to pačio tipo dėžę nepavyksta užpildyti sieną, sienos viršuje ir dešinėje generuojami tarpai, juos užpildant tarpų pildymo procedūra. Metodas visada stengiasi plokštumą išlaikyti pakuojamo paviršiaus viršuje. Procedūra stengiasi neišskirti panašaus tipo dėžių, apibrėžiant dėžių tipus: 'atidaryta' ir 'uždaryta'. Taip pat procedūra skaičiuoja dėžių padėties apribojimus. Atidaryto tipo dėžės – tai būtini dėžės tipai, kuriais visada pradedamas pakavimas, tuo tarpu uždaryto tipo dėžės – nebūtini pakavimui dėžių tipai.

George ir Robinson sukurto algoritmo privalumai:

- gali supakuoti įvairaus kiekio ir skirtingų dydžių dėžes;
- naudojama tarpų pildymo procedūra;
- procedūra skaičiuoja dėžių padėties apribojimus;
- procedūra garantuoja supakuoto krovinio didesnės dalies stabilumą;
- nurodo dvi alternatyvias atvirų dėžių tipo pasirinkimo taisykles.

Trūkumai:

- *sukurtas algoritmas efektyviai (per trumpą laiką ir pan.) gali supakuoti ne daugiau kaip iš 20 dėžių susidedantį krovinį;*
- *nepateikia tikslių detalių, kurių pasirinkus atvirų dėžių tipo taisyklę bus gaunami geresni rezultatai.*

Bischoff ir Dowsland (1982) taip pat nagrinėjo konteinerio pildymo procedūrą, pagrįstą sluoksnių formavimo skersai pločio principu. Tačiau yra du pagrindiniai skirtumai tarp Bischoff ir Dowsland ir George ir Robinson naudojamų procedūrų:

- 1) kiekvienas sluoksnis yra sudarytas tik iš vieno tipo dėžių (pagal dimensijas, galimas orientacijas ir pan.);
- 2) dėžių išdėliojimas sluoksnio viduje yra apibrėžiamas dvimate pakavimo procedūra, kuri siekia maksimizuoti panaudojimo skerspjūvio plotą (pvz., stačiakampio sudaryto iš konteinerio pločio ir aukščio).

Ši dvimatė konteinerio pakavimo procedūra yra euristinė ir buvo pasiūlyta kaip vienas iš būdų sudaryti efektyvią dėžių krovimo į konteinerį modelio schemą.

Bischoff ir Dowsland pasiūlytoje procedūroje nėra įvertintas ir nagrinėjamas tuštumų užpildymas sluoksniuose, todėl sluoksniai yra formuojami tvarka, kuri neturi įtakos pakavimo efektyvumui. Kriterijus naudojamas sluoksnio dimensijos gyliui nustatyti vis dėlto yra lemiamai reikšmingas. Kiekviena dėžė iš trijų pusių nagrinėjama paeiliui kaip potencialus sluoksnio gylio matas. Esant šiai fiksuotai dimensijai, maksimalus dėžių skaičius determinuojamas dviejų dimensijų pakavimo šablono būdu. Tarkim, konteinerio plotis ir aukštis yra žymimi W ir H , dėžė žymima trim matmenimis - w , l ir h . Taigi, w traktuojant kaip sluoksnio gylį, randamas skaičius, kuris ir parodo, kiek stačiakampių $l \times h$ telpa į stačiakampį $W \times H$. Jeigu gaunamas didesnis skaičius nei pakrautų tam tikro tipo dėžių, pilnas sluoksnis negali būti suformuotas, ir svarbus gylio matas dėl tos priežasties yra atmetamas tolesnėje procedūros eigoje. Jeigu yra daugiau negu vienas sluoksnio užbaigimui reikalingas galimas lankstus gylio nustatymo būdas, turi būti pasirenkamas vienas iš jų. Vienas iš pasirinkimų nulemiantis kriterijų yra santykinis skersinio pjūvio užpildymas (arba tūrio, arba ploto atžvilgiu). Kitas racionalus būdas - bandyti tinkamai maksimizuoti dėžių skaičių, kurios talpinamos pilnai užpildytuose sluoksniuose. Tai gali būti naudinga pasirenkant tokią dėžės vietą, kuri paliktų mažiausiai nesupakuotų dėžių ir būtų galima sudaryti kuo daugiau užpildytų sluoksnių. Tam tikroje procedūros stadijoje tikslas bus pasiektas kaip visos dėžės arba yra supakuojamos į užbaigtus sluoksnius arba, kas labiau tikėtina, nebus galima toliau

tęsti bet kokio tipo dėžių formavimo į užbaigtus sluoksnius procedūros. Tokiu atveju likusios dėžės yra pakuojamos naudojant George ir Robinson pasiūlytą būdą.

Bischoff ir Dowsland sukurto algoritmo privalumai:

- *sluoksniui užpildyti naudoja tik vienodo tipo (pagal dimensijas, galimas orientacijas ir pan.) dėžes, siekiant gauti lygų pakavimo paviršių;*
- *dėžes sluoksnyje išdėlioja, naudojant dvimačio pakavimo procedūrą.*

Trūkumai arba ko procedūra nedaro:

- *procedūroje nenagrinėjamas tuštumų užpildymas sluoksniuose;*
- *procedūros užbaigimui, kai nebus galima supakuoti bet kokio tipo dėžių į užbaigtus sluoksnius, siūlytina naudoti George ir Robinson pasiūlytą būdą.*

Daugeliu atvejų gali būti surasti keli skirtingi tos pačios euristicos variantai. George ir Robinson tiksliai nurodo dvi alternatyvias atvirų dėžių tipo pasirinkimo taisykles. Autorių darbuose aptarinėjami bandymų rezultatai gaunami šiek tiek geresni naudojant antrąją taisyklę (tvarkančią atviro tipo dėžes tuo pačiu principu kaip ir uždaro tipo). Tačiau George ir Robinson nepateikia pasiūlymų, kuris kriterijus turėtų būti naudojamas efektyvesniems pakavimų rezultatams pasiekti. Uždaro tipo dėžės, atsižvelgiant į skirtingus prioritetus, taip pat yra testuojamos. Prieinama išvados, - „kad pradinė priorityvi struktūra dažniausiai duoda geriausius rezultatus“ (George; Robinson: 1980: 154). Tačiau nėra pateikiamos tikslios detalės, kiek pateikus kitos taisyklės geriausią sprendimą gauti rezultatai geresni ir kiek blogesni.

Sienų rentimo būdas yra kaip natūralus 3D objektų pakavimo problemos supaprastinimas. Jis formuoja svarbų algoritmo komponentą, kurį nagrinėjo Bischoff ir Mariott (1990) ir kuris buvo perimtas kitų autorių. Liu ir Chen (1981) pateikė sienos pildymo algoritmą, kuriame nagrinėjo skirtingus dėžių vietos nustatymo būdus, kuriuos galima panaudoti konteinerio pločio utilizacijai maksimizuoti. Turint asignuotą sienos bazę panašus būdas gali būti taikomas ir konteinerio aukščiui. Gehring, Menschner ir Meyer (1990) taip pat pristatė euristicą, skirtą neidentiškų gaminių pakavimui konteineryje. Jie, kaip George ir Robinson, panaudojo dalių pakavimo skersai konteinerio pilno pločio ir aukščio idėją. Autoriai panaudojo tvarkymo metodą, pagrįstą talpos mažinimu ir pirmojo bloko patalpinimu į sluoksnį, kuris apibrėžiamas dėžėmis. Jie vysto pakavimo būdą, kuriuo pakavimo procesas iš pradžių vykdomas skersai konteinerio grindinio, vėliau nukreipiant aukštyn. Pirmiausia siekiama pateikti ką nors, kas sumažintų įspraudimo tarpą skersai konteinerio pločio. Gehring,

Menschner ir Meyer teigia, jog geri sprendimai yra pasiekiami, bet autoriai savo dviejų problemų taikymui pateikia tik rezultatus. Jie per daug užtikrinti, jog išlaiko plokščią priekinio pakavimo sieną, bet skirtingai nuo George ir Robinson, jie sukliudo dėžėms būti išdėstytomis plačiai abiejuose gretimuose sluoksniuose. Jie tvirtina, jog šis būdas užtikrina, kad krovinių sluoksniai gali būti kilnojami taip, kad įgautų atitinkamą perskirstymo svorį, bet jis neabejotinai pradės atskirais atvejais mažinti talpumo panaudojimą. Kitas aspektas rodo, jog nesudaromos sąlygos dėžių plačiam išdėstymui sluoksniais, todėl nukenčia krovinių stabilumas. Pakavimai, kai dėžės išsidėstę plačiai tarp sluoksnių, sudaro stabilesnius krovinius.

Gehring, Menschner ir Meyer sukurto algoritmo privalumai:

- *krovinių sluoksniai gali būti kilnojami taip, kad įgautų atitinkamą perskirstymo svorį;*
- *dėžėms neleidžiama išsidėstyti plačiai abiejuose gretimuose sluoksniuose.*

Trūkumai:

- *krovinių sluoksnių kilnojimas įgaunant atitinkamą perskirstymo svorį, sumažins konteinerio talpumo panaudojimą;*
- *sluoksniuose stengiantis dėžes išdėstyti kuo glausčiau, neužtikrinamas krovinių stabilumas.*

Han, Knott ir Egbelu (1989) teigia, jog sienų idėja neturi būti apribota vertikaliomis konteinerio kraštinėmis. Autoriai apibūdina algoritmą, kuriame konteineris (didžioji prizmė) pakuojamas identiškais dėžėmis (mažiausiomis prizmėmis). Apibūdintas algoritmas yra projektuotas atskiram dėžės tipui, kuris yra pastovus dydžio ir formos atžvilgiu, neįvertinant praktinių apribojimų. Han, Knott ir Egbelu siūlo pateikti L-formos modulių pakavimus su pradiniais modulių vertinimais, apimančiais visą konteinerio bazę ir vieną iš konteinerio sienų. Išdėstymas „L“ viduje yra determinuojamas dinaminio programavimo, kuris maksimizuoja kraštų panaudojimą. Vienas iš įdomesnių sienų užpildymo būdų yra pagal vieną iš šešių konteinerio plokštumų. Autorių aprašomame pavyzdyje talpinama viena dėžė mažiau, negu gaunama kraunant dėžes dviejų skirtingų sienų išdėstymo būdais ant konteinerio grindinio. Han, Knott ir Egbelu pateikto metodo trūkumas yra L modulio perimetro utilizacijos maksimizavimas. Autoriai nepateikia įrodymų, kodėl turi būti pasirinktas būtent L-formos modelio metodas.

Han, Knott ir Egbelu sukurto algoritmo privalumai:

- *maksimizuojamas pakuojamo konteinerio kraštų panaudojimas.*

Trūkumai:

- *neįvertinami pakuojamoms dėžėms praktiniai apribojimai;*
- *naudojant L modulį maksimaliai išnaudojamas perimetras;*
- *nepateikiami įrodymai, kodėl naudoti L-formos modulį.*

Mohanty, Mathur ir Ivancic (1994) pasiūlė pritaikyti daugiamačią „kuprinės“ problemos metodą trijų dimensijų pakavimo problemoms spręsti, pripildant dėžėmis skirtingus konteinerius. Jų tikslas buvo konteineriuose maksimaliai išnaudoti erdvę arba konteinerių turinių dydį. Autoriai pateikė stulpelių generavimo procedūrą, kuri euristiškai naudoja „godų“ metodą. Šis metodas vienu metu generuoja stulpelius, neatsižvelgdamas į apribojimus, išskyrus persidengimą ir konteinerių dimensijas. Autorių siūlomas metodas nėra stabilus ir jame problematiška atrinkti skirtingus pakavimo elementus.

Mohanty, Mathur ir Ivancic sukurto algoritmo privalumai:

- *siekiami maksimaliai išnaudoti konteinerių erdvę.*

Trūkumai:

- *naudojant „godų“ metodą neatsižvelgiama į apribojimus išskyrus persidengimą ir konteinerių dimensijas;*
- *metodas nėra stabilus;*
- *problematiška atrinkti skirtingus pakavimo elementus.*

Chen, Lee ir Shen (1995) pristatė sveikų skaičių 0-1 linijinio programavimo modelį (angl. *zero-one mixed integer linear programming model*), skirtą konteinerio trijų dimensijų pakrovimo problemai spręsti. Uždavinio tikslas – turint daug įvairaus dydžio konteinerių, supakuoti duotas dėžes į juos taip, kad transportavimo kaštai būtų mažiausi. Modelis nagrinėja dėžės vietos nustatymo, sudėtinio dėžės dydžio, sudėtinio konteinerio dydžio, dėžių laisvos vietos perdengimo ir erdvės utilizavimo problemas. Buvo atkreipiamas dėmesys į tokias konteinerio pakrovimo specifines problemas, kaip vieno konteinerio išrinkimas iš keleto variantų, svorio balanso ir skirtingų konteinerių ilgių. Šioms situacijoms buvo numatyta bendro modulio modifikacija. Modulio pagrindimą aiškina remiantis siauro pavyzdžio problemomis (tik su 6 dėžėmis). Tolimesniuose tyrimuose gali būti pasiūlyta į modelius įtraukti papildomus apribojimus konteinerio pakrovimo problemai spręsti: pakavimo šablono stabilumas, kiekvienos dėžės tipo integralumas, svorio apribojimai. Naudojant autorių pateiktą analitinį modelį neįmanoma realiai išspręsti problemos, kol kintamųjų ir apribojimų skaičius išauga dvigubai kaip ir dėžių skaičius. (Jis išauga $2n^2$, kur n yra dėžių skaičius).

Išvadose Chen, Lee ir Shen teigia, jog didesnės apimties konteinerio pakrovimo problemoms spręsti reikia efektyvesnio būdo.

Chen, Lee ir Shen sukurto algoritmo privalumai:

- dėžės pakuojamos parenkant vieną konteinerį iš keleto galimų variantų;
- atsižvelgiama į svorio balansą, skirtingus konteinerių ilgius;
- yra galimybė į modelį įtraukti papildomus apribojimus.

Trūkumai:

- modulio pagrindimas aiškinamas remiantis siauro pavyzdžio problemomis (tik su 6 dėžėmis);
- netinka didesnės apimties konteinerio pakrovimo problemoms spręsti.

Bischoff, Janetz ir Ratcliff (1995) vystydami trijų dimensijų euristinį požiūrį, išbandė kelias savo algoritmo modifikacijas. Viena algoritmo variante jie leido tą patį sluoksnį formuoti iš ne daugiau kaip dviejų tipų dėžių, kitame – iš ne daugiau kaip dviejų tipų dėžių su vienodais aukščiais, trečiame – iš vieno tipo dėžių. Jie išbandė visus savo algoritmo variantus pakuodami 1400 skirtingų problemų. Efektyviausias pasirodė pirmasis variantas, tačiau didėjant skirtingų dėžių kiekiui sprendinio kokybė mažėja.

Bischoff, Janetz ir Ratcliff sukurto algoritmo privalumai:

- algoritmas modifikuojamas, surandant efektyviausią spendimo variantą;
- pasižymi dideliu pakavimo stabilumu.

Trūkumai:

- didėjant skirtingų dėžių kiekiui sprendinio kokybė mažėja.

Martello, Pisinger ir Vigo (2000) vystė „šakos ir ribos“ algoritmą, padedantį išspręsti trijų-dimensijų dėžių pakavimo problemas. Uždavinio tikslas – turint daug vienodo dydžio konteinerių, supakuoti duotas dėžes į kiek galima mažesni jų skaičių. Jų sprendimas nėra tiksliai trijų dimensijų. Pagal autorių sukurtą algoritmą, pirmiausia sukuriama dėžių sluoksniai, turintys plotį W , aukštį H ir skirtingus gylius. Tuomet sluoksniai sujungiami į trijų-dimensijų dėžes. Į pakavimo algoritmą nebuvo įtraukti skirtingi svarbūs dėžių pakavimui apribojimai ir nebuvo leistinas daiktų kaitaliojimas. Nepaisant tokių sąlygų, autorių pateiktas 3D objektų pakavimo būdas pasiekė gerų rezultatų.

Martello, Pisinger ir Vigo sukurto algoritmo privalumai:

- dėžės pakuojamos į kuo mažesni vienodų konteinerių skaičių.

Trūkumai:

- pakuojant dėžes, neleistinas jų kaitaliojimas;

- *neatsižvelgiama į svarbius skirtingus dėžių pakavimo apribojimus.*

Faina (2000) nagrinėdamas 3D objektų pakavimo problemas nenaudoja konteinerio dalijimo į sluoksnius euristicos ir pateikia modelį, suvedantį trimačio pakavimo sprendinius į skaičiąją aibę. Modelis leidžia nagrinėti problemas, kai leistini visi galimi dėžių vartymai, o vienintelis taikomas apribojimas – dėžės turi tilpti į konteinerį. Pakavimo uždaviniui spręsti autorius naudoja atkaitinimo modeliavimo metodą. Pakuojant 32 dėžes, autorius sugebėjo parinkti aušinimo tvarkaraščius, duodančius geriausius rezultatus, tačiau, kai dėžių skaičius viršydavo 64, skaičiavimo laikas tapdavo nebepriimtinas.

Faina sukurto algoritmo privalumai:

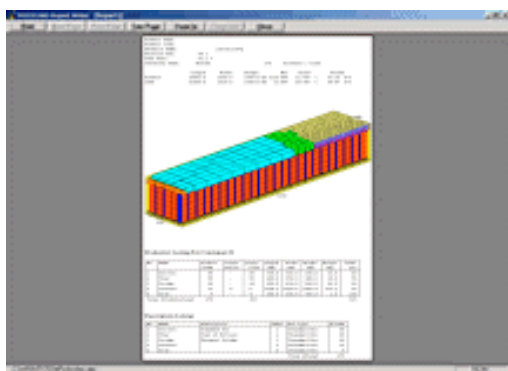
- *galimi visi dėžių vartymai, taikomas tik vienas apribojimas.*

Trūkumai:

- *didėjant dėžių skaičiui rezultatas blogėja;*
- *didėjant dėžių skaičiui skaičiavimo laikas ilgėja ir tampa nebepriimtinas.*

2.2. Komercinės 3D objektų pakavimo programos

Daugelis 3D objektų pakavimo komercinių programų gali būti naudojamos krovinio pakrovimui planuoti, naudojant bet kokius stačiakampius konteinerius, sunkvežimius, priekabas, apkalas (baldams transportuoti), drezinas, kilnojamas platformas ir kt., taip pat tinka bet kokio kitokio trijų matavimų erdvės pakavimo uždaviniams spręsti.



1 pav. *Truckfill* ekranvaizdis

Viena iš kampanijų **CAPE Systems Inc.** siūlo keletą skirtingų komercinių produktų. Viena iš jų vadinamasis „Sunkvežimio pildymas“ (angl. *Truckfill*) (1 pav.). Šios firmos svetainėje rašoma, kad naudojama vėliausia optimizavimo technika ir programa gali greitai pateikti sunkvežimio ar konteinerio užpildymo geriausią sprendinio variantą, remiantis praktinėmis pakavimo taisyklėmis. Firmos svetainėje nėra jokios informacijos

apie programoje naudojamą algoritmą, taip pat nėra galimybės parsisiųsti programos bandomosios versijos.

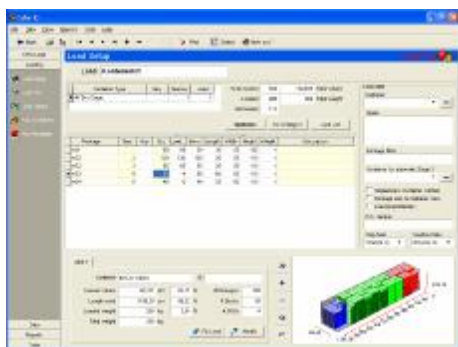
Programos privalumai ir galimybės:

- *Programa vaizdiškai parodo, kaip pakrauti konteinerius.*
- *Suteikia galimybę su klientais bendrauti tiesiogiai internetu.*
- *Teikia kroviniams pakrovimo prioritetus.*

- Programa gali valdyti sunkvežiminio ar konteinerio pakrovimo procesą: planuoti, kurti, koreguoti, spausdinti ir palaikyti pakrovimo planus.
- Programa gali perskaičiuoti pakrovimo skaičiavimus, krovinių svorį, dimensijas ir bendro krovinio gravitacijos centrą po kiekvieno krovimo plane padaromo pakeitimo.

Trūkumai:

- Sudėtinga sistema.
- Didelė kaina.



Kampanija **MagicLogic Optimization Inc.** sukūrė programą Qube-IQ 3.0 (2 pav.), kurios bandomąją versiją galima nemokamai parsisiųsti iš jų interneto svetainės (<http://www.magiclogic.com>). Apie programoje naudojamą algoritmą autoriai neužsimena.

2 pav. Qube-IQ 3.0 ekranvaizdis

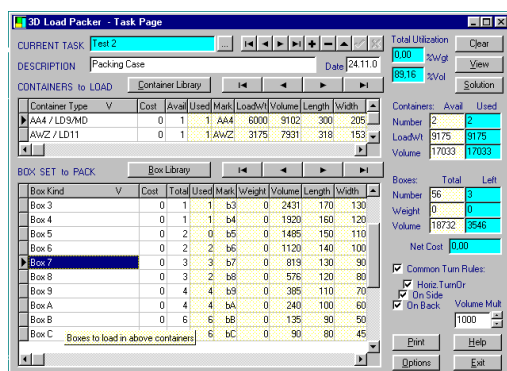
Programos privalumai ir galimybės:

- programa leidžia pakuoti į stačiakampio gretasienio formos konteinerius, į konteinerius, kurių grindys arba stogas yra neplokšti;
- galima pakuoti duotus objektus į kelis skirtingo dydžio konteinerius;
- pakuojami objektai gali būti dėžės (stačiakampiai gretasieniai), cilindro arba sofos (L raidės) formos;
- pakuojamiems objektams galima nurodyti ant kokių plokštumų juos galima statyti, kiek ant jų galima dėti kitų objektų, atsižvelgiant į jų svorį;
- galima objektams priskirti keletą specialių atributų, pavyzdžiui, jei plokščia tai krauti viršuje (taikytina didelėms plokščioms dėžėms, kurios nuo ant jų užkrauto svorio įlinksta);
- programai galima nurodyti norimą gravitacijos centrą;
- programa automatiškai palaiko reikiamą ašių svorio limitą.

Trūkumai:

- Sudėtinga sistema.
- Nespecializuota, universali.
- Sudėtinga vartotojo sąsaja.

- *Suranda gerą, bet ne geriausią sprendinį.*



3 pav. 3D Load Packer ekranvaizdis didesnes stačiakampes talpas (konteinerius).

Programos privalumai ir galimybės:

- *į programą galima įtraukti papildomus apribojimus: bendrus krovinio svorio ir sunkvežimio ašinio svorio limitus;*
- *gali pilnai kontroliuoti leistiną dėžių išsikišimą;*
- *gali įvardinti kiekvienos dėžės, talpos ir daikto kainą, pagal ką apskaičiuojamos bendros sąnaudos;*
- *gali supakuoti įvairių dydžių talpas tuo pačiu metu, atsižvelgiant į bendrą visų dėžių komplekto numatytą ir leistiną užimamą tūrį, nurodant jį kiekvienai dėžei atskirai arba visoms kartu;*
- *sukrovimo struktūra pateikiama lentelėmis ir grafiškai – spalvotais trimačiais izometriniais vaizdais, kuriuos galima vartyti, stebėti palaipsnių krovimą bei atspausdinti bet kurį vaizdą, kaip ir bet kurią kitą atskaitą.*

Trūkumai:

- *Nėra galimybės krovinius pakuoti nustatyta eilės tvarka.*

Apibendrinant nagrinėtus autorių sukurtus algoritmus, galima išskirti šiuos esminius jų privalumus:

- *gali supakuoti įvairaus kiekio ir skirtingų dydžių dėžes;*
- *galimi visi dėžių vartymai;*
- *naudojamos tarpų pildymo procedūros;*
- *skaičiuojami dėžių padėties apribojimai, arba leidžiama juos papildomai įtraukti į modelį;*
- *siekiami maksimaliai išnaudoti konteinerių tūrį;*

- *atsižvelgiama į svorio balansą, skirtingus konteinerių ilgius;*
- *sluoksnyje dėžės išdėliojamos, naudojant dvimačio pakavimo procedūrą.*

Apibendrinant ir įvertinant 3D objektų pakavimo algoritmus, galima išskirti šiuos esminius jų veikimo trūkumus:

- *didėjant skirtingų dėžių kiekiui sprendinio kokybė mažėja;*
- *nenagrinėjamas tuštumų užpildymas sluoksniuose;*
- *neužtikrinamas krovinio stabilumas;*
- *neatsižvelgiama į svarbius skirtingus dėžių pakavimo apribojimus;*
- *didėjant dėžių skaičiui skaičiavimo laikas ilgėja ir tampa nebepriimtinas.*

Išanalizavus 3D objektų pakavimo programas, apibendrinant galima išskirti šiuos sistemų privalumus:

- *pakuojant atsižvelgiama į papildomus apribojimus: bendrus krovinio ir sunkvežimio ašinio svorio limitus, teikia kroviniams pakrovimo prioritetus, neleidžia kai kurių objektų vartyti, riboja, dedamą ant kitų objektų, svorį;*
- *programos vaizdiškai parodo, kaip pakrauti konteinerius;*
- *programos gali valdyti konteinerio pakrovimo procesą: planuoti, kurti, koreguoti, spausdinti ir palaikyti pakrovimo planus;*
- *kai kurios programos universalesnės ir pakuoja ne tik stačiakampes dėžes, bet ir cilindro arba sofos (L raidės) formos objektus.*

Išanalizavus 3D objektų pakavimo programas, apibendrinant galima išskirti šiuos sukurtų sistemų trūkumus:

- *Sudėtinga sistema.*
- *Didelė kaina.*
- *Suranda gerą, bet ne geriausią sprendinį.*
- *Paini vartotojo sąsaja.*

Apibendrinant nagrinėtų algoritmų ir programinės įrangos privalumus ir trūkumus išskeliame sau uždavinį sukurti specializuotą – skirtą naudoti tik stačiakampėms dėžėms pakrauti į konteinerį, lengvai naudojamą ir greitai pateikiančią pakankamai gerą konteinerio pakuojamos erdvės užpildymo rezultata, programinę įrangą ir ją realizuoti konkrečios įmonės veikloje.

2. PROJEKGINĖ DALIS

2.1. Reikalavimų dokumentas

Vartotojai. Potencialiausi programinio projekto vartotojai yra transportavimo kompanijų, gabenančių krovinius, supakuotus stačiakampėse dėžėse, darbuotojai. Taip pat daugelis įmonių, kurioms yra aktualu, kuo geriau kraunant stačiakampius objektus išnaudoti pakuojamą erdvę.

Programine įranga gali naudotis kiekvienas, net ir mažai darbo su kompiuteriu įgūdžių turintis firmos darbuotojas. Tačiau darbas bus efektyvesnis, jei vartotojas turės bent minimalias su Microsoft Windows operacine sistema darbo žinias ir įgūdžius.

Vartotojams būtinos lietuvių kalbos žinios, kadangi programa yra skirta lietuviškai kalbančių vartotojų grupei, todėl visi paaiškinimai ir rezultatai pateikiami būtent šia kalba.

Projekto tikslas

- Sukurti nesunkiai įsisavinamą, nebrangią programinę įrangą, leidžiančią vartotojui gauti pakankamai gerą rezultatą pakuojant stačiakampius objektus trimatėje erdvėje.
- Sukurti mažai darbo patirties su kompiuteriu turintiems vartotojams patogią ir lengvai suprantamą sistemą.
- Sukurti universalią 3D stačiakampių objektų pakavimo į konteinerį programinę įrangą, kurią būtų lengva pritaikyti atitinkamiems vartotojams pagal jų poreikius, papildomai į programinį paketą įtraukiant tam tikrus funkcionalumo reikalavimus ar pakavimo apribojimus.
- Sudaryti greitai ir vaizdžiai pateikiančią rezultatų programinę įrangą, kuri leidžia vartotojui pateikti programos apdorojimo duomenis Microsoft Windows ir Microsoft Office dažniausiai naudojamų standartinio paketo programų *.txt formato faile.

Pagrindinės produkto funkcijos yra:

- Pagal duotas konteinerio ir dėžės dimensijas pateikti geriausią stačiakampių dėžių supakavimo konteineryje sprendimo būdą.
- Pateikti vartotojui geriausią supakuotų dėžių sprendinį skaitmenine ir vizualine išraiška su rezultatų (ataskaitos) spausdinimo galimybe.

Funkciniai vartotojo reikalavimai. Produktas tiesiogiai sąveikauja su vartotoju, kadangi visų dėžių bei konteinerių dimensijų duomenys nuskaitomi iš vartotojo apibrėžto failo.

Vartotojas turi suvesti įėjimo reikšmes: konteinerio dimensijas, dėžės matmenis ir vienodo tipo dėžių skaičių. Programa išveda geriausią dėžių supakavimo sprendinį paskaičiuodama jį pagal užduotas įėjimo reikšmes. Sprendinio duomenys turi būtų išvedami kuo greičiau.

Duomenų įvedimas ir pateikiami rezultatai turi būti aiškūs ir suprantami.

Programinė įranga yra skirta jos vartotojui, todėl norint išsiaiškinti vartotojo poreikius, reikėtų išskelti keletą klausimų:

- Ko iš kuriamos programos nori vartotojas?

Vartotojas tikisi, kad programa užims mažai vietos kompiuterio atmintyje ir bus paprastai įdiegiama. Vartotojas nori įvedimo duomenis suvesti nepriklausomai nuo programos: surašyti juos atskirame faile. Vartotojas tikisi, kad programa pateiks tikslų ir aiškų stačiakampių dėžių pakavimo į konteinerį rezultatą, pateikiant kiekvienos supakuotos dėžės pakavimo koordinatas ir padėtis, taip pat likusių nesupakuotų dėžių sąrašą. Vartotojas tikisi, kad programa pateiks dėžių supakuotų į konteinerį grafinį vaizdą, iš kurio bus galima aiškiai matyti, kurioje konteinerio vietoje yra padėta tam tikra dėžė.

- Kaip įsivaizduojamas programinės įrangos veikimo procesas?

§ Sukuriamas duomenų įvesties failas.

§ Vykdoma programa, kuri sugeneruoja rezultatus.

§ Išvedamas rezultatas - ataskaita.

§ Parodomas supakuotų dėžių konteineryje grafinis vaizdas.

- Kokie yra vartotojo keliami uždaviniai 3D objektų pakavimo sistemai?

§ Leisti pasirinkti bet kokius konteinerio matmenis.

§ Leisti pasirinkti skirtingus dėžių matmenis.

§ Leisti patiems nepriklausomai nuo programinės įrangos sukurti įvesties failus.

§ Leisti koreguoti faile netinkamai įvestas konteinerio arba dėžės dimensijas.

§ Leisti perskaičiuoti gautą rezultatą, įvesties faile pakeitus konteinerio arba dėžės dimensijas.

§ Greitai gauti rezultatus.

§ Ataskaitose pateikti tik reikalingiausią informaciją.

§ Grafinis vaizdas turi aiškiai parodyti, kurioje konteinerio vietoje yra supakuota tam tikra dėžė.

§ Gautus rezultatus (ataskaitas, vaizdus) sistema turi leisti atspausdinti.

Nefunkciniai vartotojo reikalavimai.

Reikalavimai patikimumui ir kokybei:

- Sukurtas modelis, realizuotas C++ programine kalba, turi realiai funkcionuoti ir išvesti teisingus rezultatus.
- Programa turi veikti taip, kaip vartotojas tikisi ir turi atitikti aprašymo reikalavimus.
- Turi būti suteikta galimybė pasirinkti galimas įėjimo reikšmes.
- Rezultatai privalo būti gaunami prie bet kurių pasirinktų reikšmių.
- Turi būti vengiama kritinių situacijų.
- Turi būti patogi ir neerzinanti vartotojo sąsaja.
- Dokumentacija turi būti išsami, ir padėti naudotis vartotojui neturinčiam įgūdžių šioje srityje.

Programinės bei aparatūrinės įrangos reikalavimai:

- Aparatūrinė įranga:
IBM PC.
- Programinė įranga:
 - § Programos greitis priklausys nuo turimos techninės įrangos.
 - § Planuojamas programos dydis: apie 3 Mb.
 - § OS: Microsoft Windows bet kuri terpė (viena iš labiausiai paplitusių operacinių sistemų vietinėse rinkose.)
 - § Programinei įrangai kurti turi būti naudojamos objektinio projektavimo ir programavimo technologijos.

Sistemos architektūra. Projektuojant siekiama sukurti kiek galima lankstesnę sistemą, kurią būtų galima reikalui esant nesunkiai išplėsti papildant naujomis funkcionalumo ar vizualizacijos galimybėmis. Siekiant anksčiau paminėtų tikslų didelis dėmesys bus skiriamas tam, kad kuriama sistema būtų patikima, paprasta naudotis vartotojams.

Programinės įrangos sukūrimui ir testavimui bus vartojami standartiniai IBM šeimos kompiuteriai. Turi būti naudojamos objektinio programavimo technologijos, kurios bus realizuotos C++ programavimo kalba.

Reikalavimai vartotojo sąsajai. Vartotojo grafinė sąsaja padeda verifikavimui, galimų klaidų tikrinimui ir prezentaciniam geriausio sprendinio pateikimui. Sugeneravus rezultatą vartotojo grafinėje sąsajoje turėtų būti parodomas aiškus supakuotų dėžių vaizdas. Programa turi leisti vartotojui valdyti grafinį vaizdą klaviatūros mygtukų pagalba tam, kad būtų matoma kuo aiškesnė supakuotų dėžių vieta konteineryje:

- galimybė išjungti arba įjungti konteinerio kraštinių rodymą;
- sukroti dėžes apie X ir Y ašį;
- matyti atskirai kiekvienos supakuotos dėžės vietą ir padėtį konteineryje.

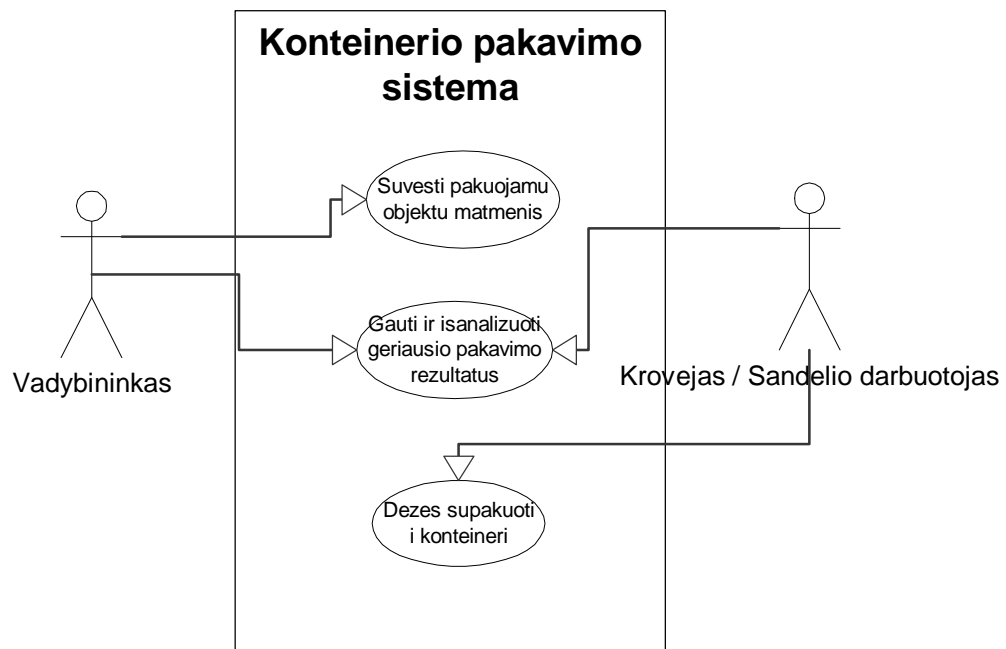
Programa turi pritaikyti visas dimensijos reikšmes ir atitinkamas koordinates taip, kad šios tilptų į ekraną.

Reikalavimai programinės įrangos patikimumui, saugumui, mobilumui

- Programa turi teisingai veikti su visomis įmanomomis įėjimo reikšmėmis.
- Turi būti išvengta kritinių situacijų.
- Duomenų įvedimo ir išvedimo procedūros turi būti aiškios ir paprastai naudojamos.
- Pateikti pakavimo rezultatai turi būti tikslūs.
- Programa turėtų veikti su naujausiomis Windows versijomis: Windows 2000, Windows 98, Windows 98 Second Edition, Windows ME, Windows XP.

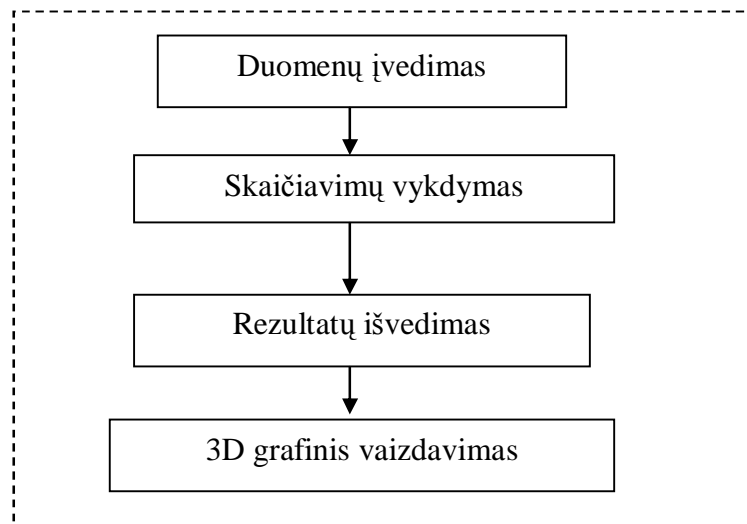
2.2. 3D objektų pakavimo sistemos modeliai

Sistemos panaudojimo atvejų modelyje (4 pav.) veiklos dalyvis „vadybininkas“ dalyvauja tokiuose veiksmuose: pakuojamų objektų duomenų suvedimas, geriausio pakavimo rezultato gavimas ir analizavimas. „Krovėjas / sandėlio darbuotojas“ taip pat analizuoja programos gautus geriausio pakavimo rezultatus ir pagal juos dėžes sukrauna į konteinerį.



4 pav. *Konteinerio geriausio pakavimo sistemos panaudojimo atvejų modelis.*

Abstrakčią kuriamos sistemos struktūrą apibrėšime sistemos struktūros modeliu (5 pav.).



5 pav. *Sistemos struktūros modelis.*

- Duomenų įvedimas – tai vartotojo apibrėžto failo sukūrimas, kuriame suvedami konteinerio ir dėžių matmenys. Sistema naudodama šiuos duomenis, ieško geriausio dėžių pakavimo į konteinerį sprendinio.
- Skaičiavimų vykdymas – tai vartotojo įvestų duomenų generavimas ir geriausio dėžių į konteinerį pakavimo sprendinio suradimas.
- Rezultatų išvedimas – tai skaičiavimo rezultatų pateikimas ataskaitoje.
- Grafinis vaizdavimas – tai skaičiavimo rezultatų pateikimas grafiniame pavidale. Pateikiamas dėžių supakuotų į konteinerį grafinis vaizdas.

2.3. Programinės įrangos projektavimas

2.3.1. Programinės įrangos bendra charakteristika

3D objektų geriausio pakavimo problemoms spręsti sukurtas modelis, kuris realizuotas C++ programine kalba. Atlikus suderinimą, verifikavimą bei ratifikavimą, buvo sukurtas programinis paketas, kuris atlieka tam tikro skaičiaus dėžių pakavimą į duotus konteinerius procesą. Sukurtas modelis į duotą konteinerį supakuoja tiek dėžių kiek tik įmanoma sutalpinti, pasirenkant atitinkamas iš jų eilės dėžes. Sukurto modelio pagalba taip pat galima atlikti stačiakampio tipo dėžių pakavimo operaciją skirtingomis padėtimis. Taip pat atliekamas pakavimas priklausomai nuo skirtingos konteinerio padėties. Kitaip tariant, jis sudaro sienas ar sluoksnius išilgai kurios nors vienos iš šešių duotų konteinerio plokštumų, jeigu viso trys konteinerio dimensijos nėra vienodos.

Kiekvienos iteracijos metu modelis sudaro vis naują pakavimo veiksmą. Šis būdas neriboja skirtingų dėžių skaičiaus kiekviename sluoksnyje. Sluoksnyje galima supakuoti bet koki skaičių skirtingų dėžių, jeigu jų išorinis paviršius tinkamas užpildyti likusiems tušties tarpams sluoksnyje.

Mano sukurtas modelis yra unikalus tuo, kad naudoja abu būdus: sluoksnių pildymo ir sienos statymo. Programa naudoja ir keletą kitų dėžių efektyvaus pakavimo metodų. Vienas iš jų yra posluoksnių pakavimas paskutiniame pildomajame sluoksnyje, kur yra likę nors kiek tuščios, neužpildytos erdvės. Tai vadinama sluoksnių - sluoksnyje pakavimo būdu. Sluoksnių pildymo technika yra tokia efektyvi, kad net po keleto iteracijų, modelis gali pateikti sprendinį, kuris 92-100 proc. užpildo pakuojamą talpą. Sukurtas projektas yra ypatingai efektyvus, kai reikia supakuoti didelį kiekį ir skirtingų tipų dėžes. Ši programa per trumpą laiką suranda geriausią dėžių pakavimo į konteinerį sprendinį.

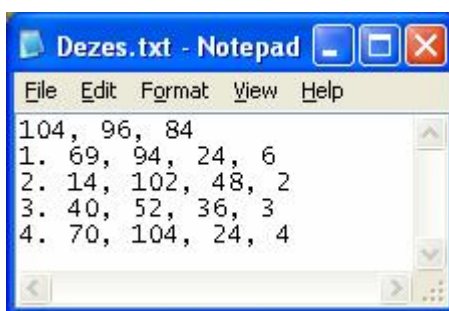
Galima išskirti šiuos sukurtos 3D objektų pakavimo sistemos naudojamus metodus ir jų privalumus:

- gali supakuoti įvairaus kiekio ir skirtingų dydžių dėžes;
- galimi visi dėžių vartymai;
- siekiama maksimaliai išnaudoti konteinerio talpą;
- naudojama tarpų pildymo procedūra;
- dėžės pakuojamos sluoksniais;
- sluoksnyje dėžės išdėliojamos, naudojant dvimačio pakavimo procedūrą;
- naudojamas posluoksnių pakavimo metodas;
- siekiama gauti kuo tikslesnius duomenis;

- *siekiama sumažinti sprendinio gavimo laiką;*
- *suteikiama galimybė papildomai dėžių padėties apribojimus ir kitus papildomus reikalavimus įtraukti į sistemą;*

2.3.2. Duomenų struktūra

Pradiniai duomenys ir jų įvedimas. Visų dėžių bei konteinerių dimensijų duomenys nuskaitomi iš vartotojo apibrėžto failo. Vartotojo apibrėžtas failas turi būti pateikiamas Microsoft Windows ir Microsoft Office standartinio paketo programų *.txt formato faile. Sukurtas failas privalo būti tame pačiame disko aplanke, kur ir yra vykdoma programa. Įvesti duomenis į failą vartotojas privalo apibrėžtu formatu (6 pav.):



6 pav. Įvedamų duomenų failo pavyzdys.

Šiame faile pirmoje eilutėje nurodomi trys skaičiai, kurie nusako konteinerio x, y, z dimensijas (arba konteinerio ilgis, aukštis ir plotis). Išdėstymo tvarka taip pat yra svarbi, kadangi pabaigoje geriausio sprendinio išvesties duomenų formatas konvertuojamas į konteinerio įvesto įvesties faile padėtį. Visos sekančios eilutės pateikia informaciją apie dėžes. Pirmas skaičius kiekvienoje eilutėje nusako dėžės žymę ir neturi poveikio sprendimo rezultatui, kol programa neatsižvelgia į šias reikšmes. Antras, trečias ir ketvirtasis skaičiai yra atitinkamai kiekvieno tipo dėžės x, y ir z matmenys. Kadangi programa išbando visas galimas kiekvienos dėžės padėtis, todėl eiliškumo tvarka dėžės matmenų nėra svarbi. Penkta reikšmė duomenų eilutėje nurodo to pačio tipo dėžių skaičių. Nors kableliai tarp nurodomų reikšmių nėra privalomi, tačiau bent vieno tarpo simbolis turėtų skirti skaičius.

Rezervuotas atminties masyvas atributams saugoti. Rezervuotas atminties masyvas atributams saugoti naudojamas, norint išvengti didžiulių atminties kiekių, kurie kaupiasi saugojant kiekvieną sprendinį, gautą vykdant sprendimo procesą. Kiekvienos iteracijos metu gautas sprendinys saugo tikrai tris parametrus. Tie atributai yra: konteinerio

padėtis, pradinio sluoksnio storumo reikšmė, pilnutinis talpos išnaudojimas. Tuomet geriausio sprendinio atributai yra naudojami rekonstruoti surastą geriausią sprendinį.

Duomenų struktūra. Sukurtame projekte yra svarbu, kad duomenys būtų gaunami kuo greičiau. Tuo tikslu naudojami du skirtingi masyvai bei dvejojo kreipimosi sąrašas. Pirmasis yra dėžių sąrašo masyvas (DEZSARAS []), kuriame yra visų dėžių matmenų duomenys, supakuotų į konteinerį dėžių koordinatės bei kiti reikalingi duomenys. Šio masyvo kiekviename įrašė yra viso dvylika laukų (1 lentelė):

1 lentelė. Dėžių sąrašo masyvo įrašų laukai.

Elementas	Vardas	Apibūdinimas
1.	Pakst	Pakavimo padėtis (0: nepakuotas; 1: supakuotas)
2.	Sk	Skaičius dėžių, kurios turi vienodas dimensijas
3.	Dim1	Ilgis vienos iš trijų dimensijų
4.	Dim2	Ilgis kitos dimensijos
5.	Dim3	Ilgis kitos dimensijos
6.	Kox	X koordinatė pakuojamos dėžės vietos
7.	Koy	Y koordinatė pakuojamos dėžės vietos
8.	Koz	Z koordinatė pakuojamos dėžės vietos
9.	Pakx	X supakuotos dėžės padėties dimensija
10.	Paky	Y supakuotos dėžės padėties dimensija
11.	Pakz	Z supakuotos dėžės padėties dimensija
12.	Talpa	Dėžės talpa (Dim1*Dim2*Dim3)

Taip pat yra saugomi kiekvienos dėžės tūrio duomenys, todėl modeliui nereikia atlikti skaičiavimų kiekvieną kartą, kai prireikia tokių duomenų. Laukai 6-11 yra nereikšmingi, jeigu Pakst reikšmė yra nulis, tačiau jie suteikia informaciją apie pakavimą, jei jau kartą dėžė buvo supakuota ir Pakst turi reikšmę 1. Kiekviena dėžė turi įrašą dėžių sąrašo masyve (DEZSARAS[]).

Kitas yra sluoksnių masyvas (SLUOKS []). Šiame masyve saugoma visų dėžės dimensijų skirtingi ilgiai. Kiekviena sluoksnio dimensijos reikšmė (Sluoksdim) rodo skirtingo sluoksnio storio reikšmę su kuria kiekviena iteracija gali pradėti pakavimą. Prieš pradėdant iteracijas, visi skirtingi ilgiai visų dėžės dimensijų kartu su įvertinimo reikšmėmis saugomi šiame masyve. Masyvo kiekvienas įrašas sudarytas iš dviejų skirtingų duomenų laukų (2 lentelė):

2 lentelė. Sluoksnio masyvo įrašų laukai.

Elementas	Vardas	Apibūdinimas
1.	Sluoksdim	Dimensijos reikšmė
2.	Sluoksvor	Svorio reikšmės nustatymas atitinkamai Sluoksdim reikšmei

Dvejopo kreipimosi sąrašas saugo sukonstruoto sluoksnio topologinę padėtį. X ir Z koordinatės laikomos kiekvieno tarpo dešiniu kampu. Programa atsižvelgia į šiuos tuščius tarpus ir bando užpildyti juos dėžėmis kaip tik tuo metu bandydama laikytis sluoksnio krašto (briaunos). Kiekvienas įrašas dvejopo kreipimosi sąrašė turi šiuos duomenų laukus (3 lentelė):

3 lentelė. Dvejopo kreipimosi sąrašo duomenų laukai.

Elementas	Vardas	Apibūdinimas
1.	*pri	Rodyklė, kuri saugo ankstesnio įrašo adresą
2.	Kumx	Nusako X koordinatę tarpo dešinio kampo
3.	Kumz	Nusako Z koordinatę tarpo dešinio kampo
4.	*sek	Rodyklė, kuri nurodo sekančio įrašo adresą

Atliekant kiekviena iteraciją šis dvejopo kreipimosi sąrašas gali turėti tikrai vieną įrašą, arba gali turėti šimtus įrašų remiantis dėžės dydžiais iš jų supakuotos aibės. Dažniausiai nauji įrašai yra įtraukiami, kai nenaudingi įrašai yra pašalinami. Taigi dvejopo kreipimosi sąrašas yra naudojamas dinamiškai ir efektyviai tvarkyti šiuos poreikius.

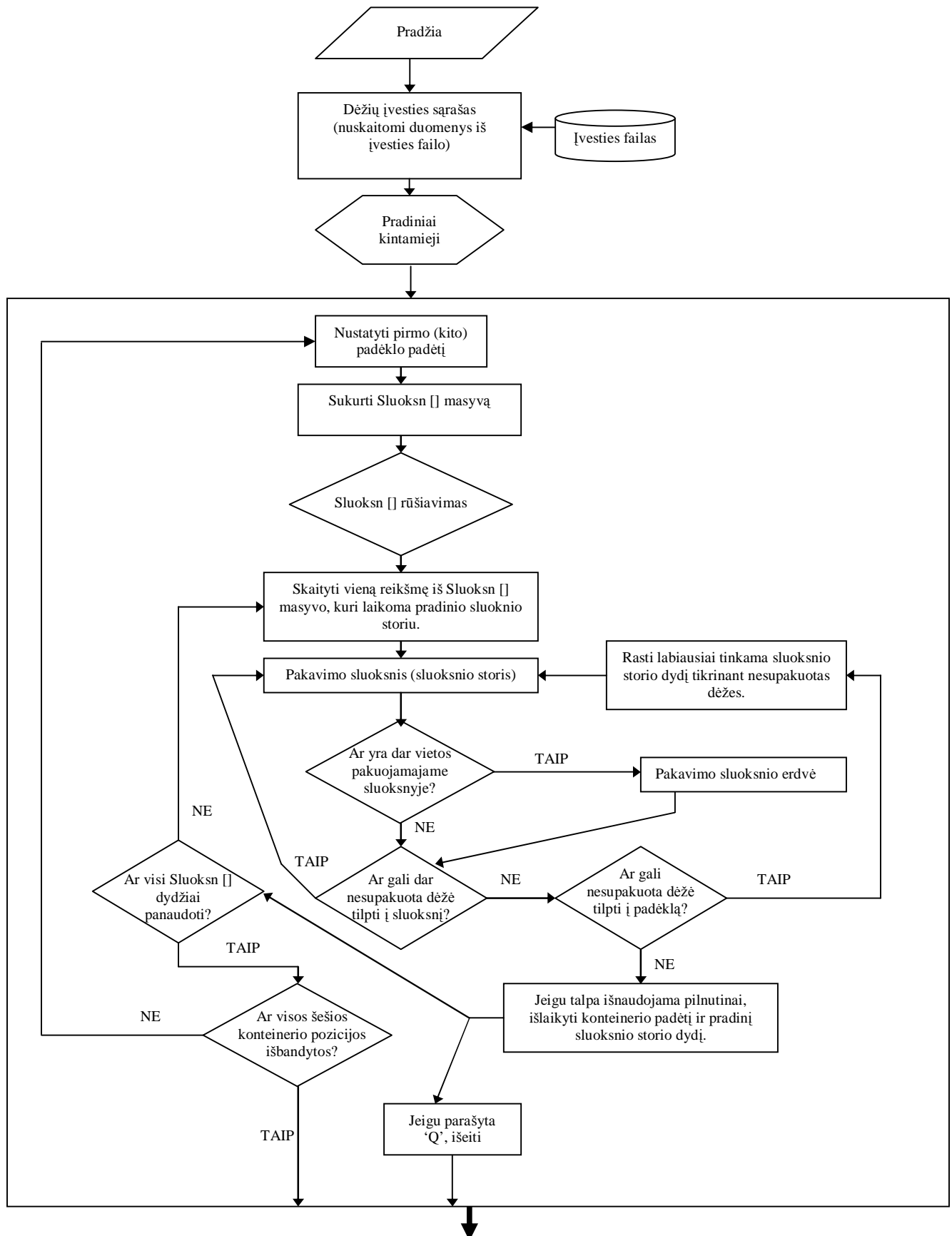
Skaitmeninės ribos. Šio modelio apribojimai buvo nustatyti atsižvelgiant į vidutinę kompiuterio atminties talpą ir pakavimo realias problemas (4 lentelė).

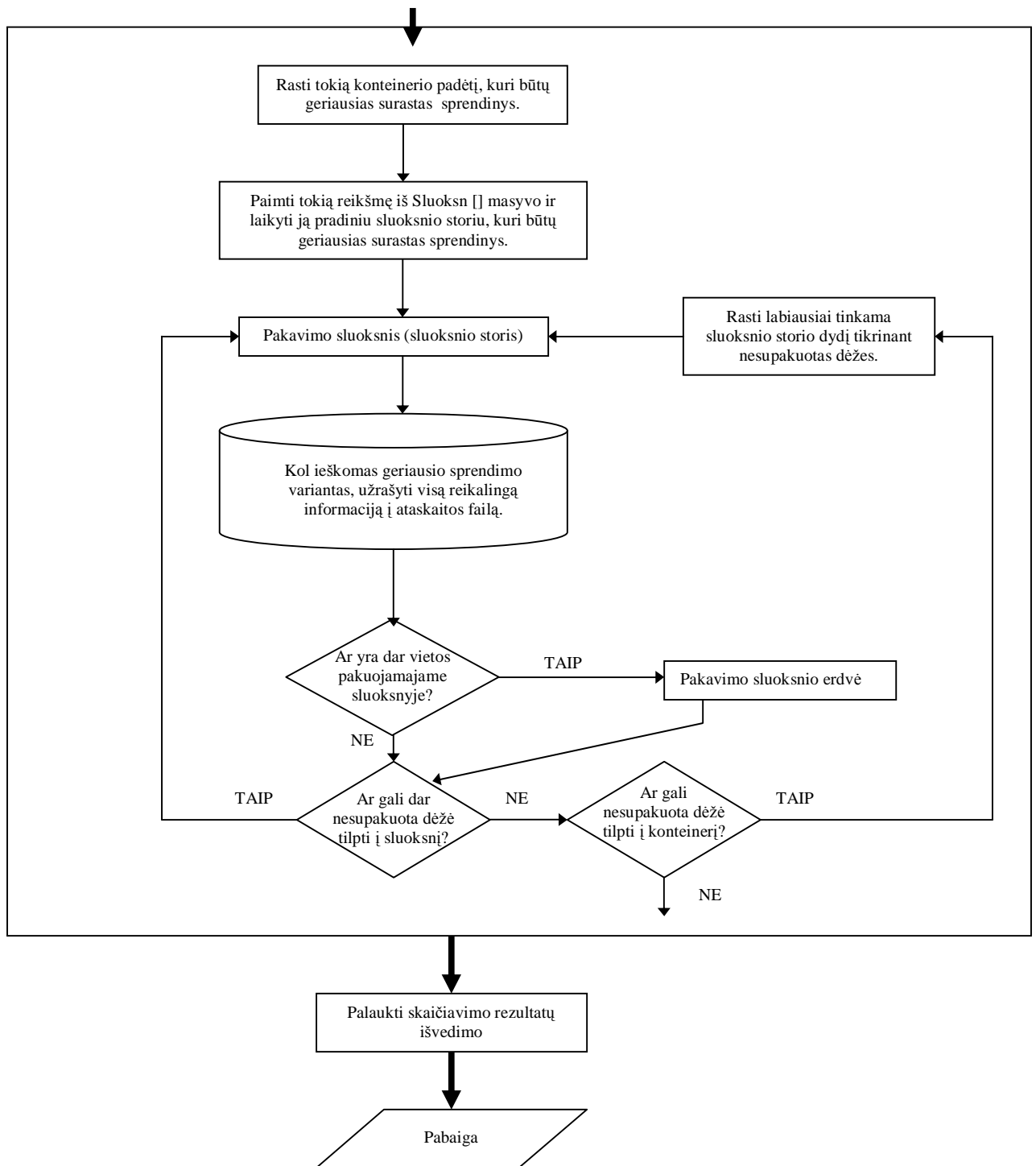
4 lentelė. Skaitmeninės ribos.

Maksimalus dėžių skaičius jų rinkinyje	5000
Bendras skirtingų matavimo reikšmių skaičius	1000
Maksimalus dimensijos ilgis bet kurio konteinerio arba dėžės	36
Visos matavimų reikšmės turi būti sveikųjų skaičių tipo	

2.3.3. Projektuojamos sistemos algoritminė diagrama

Algoritminė diagrama rodo kaip vyksta iteracijų procesas (7 pav.):





7 pav. 3D objektų pakavimo projektuojamos sistemos algoritminė diagrama

2.3.4. Iteracijų paruošimas ir vykdymas

Iteracijų paruošimas. DEZSARAS [] masyvo sukūrimas (5 lentelė):

5 lentelė. DEZSARAS [] masyvo sukūrimo pavyzdys.

Kintamiesiems priskiriamos reikšmės: X=104; Y=96; Z=84;												
Dezsaras [x]=(Pakst	Sk	Dim1	Dim2	Dim3	Kox	Koy	Koz	Pakx	Paky	Pakz	Talpa
Dezsaras [1]=(0,	4,	70,	104,	24,	0,	0,	0,	0,	0,	0,	174720
Dezsaras [2]=(0,	4,	70,	104,	24,	0,	0,	0,	0,	0,	0,	174720
Dezsaras [3]=(0,	4,	70,	104,	24,	0,	0,	0,	0,	0,	0,	174720
Dezsaras [4]=(0,	4,	70,	104,	24,	0,	0,	0,	0,	0,	0,	174720
Dezsaras [5]=(0,	2,	14,	104,	48,	0,	0,	0,	0,	0,	0,	69888
Dezsaras [6]=(0,	2,	14,	104,	48,	0,	0,	0,	0,	0,	0,	69888
Dezsaras [7]=(0,	3,	40,	52,	36,	0,	0,	0,	0,	0,	0,	74880
Dezsaras [8]=(0,	3,	40,	52,	36,	0,	0,	0,	0,	0,	0,	74880
Dezsaras [9]=(0,	3,	40,	52,	36,	0,	0,	0,	0,	0,	0,	74880

Pasibaigus duomenų įvesties procesui modelis sukuria sluoksnio storio masyvą SLUOKS []. Šis masyvas talpina kiekvieną unikalią dėžių dimensiją mažesnę negu esamos konteinerio padėties y dimensiją. SLUOKS [] masyvas sukuriamas kiekvieno konteinerio padėčiai. Kiekviena įvedama reikšmė gali būti ir sluoksnio storio reikšmė, skirta atlikti iteracijas su esama konteinerio padėtimi pradendant pakavimą.

Jeigu analizuojamo sluoksnio dimensijos reikšmė būtų išrinkta sluoksnio storio dydžiu, ji parodytų, kaip visų kitų dėžių matmenys yra arti sluoksnio aukščio matmens. Sukurtas modelis suskaičiuoja šias analizuojamas reikšmes sekančiai:

1. Išrenka dėžę ir vieną iš jos dimensijų.
2. Nagrinėjamos ankstesnės sluoksnio dimensijos reikšmės masyve.
3. Jeigu ilgis yra skirtingas ir mažesnis už konteinerio esamos padėties y dimensiją, tada ilgis išsaugomas sluoksnio dimensijos masyve kaip naujas elementas.
4. Taip nagrinėjama kiekviena dėžė išrenkant jos dimensiją artimiausia sluoksnio dimensijos reikšmei; ir išsaugoma absoliuti reikšmė, kuri yra skirtumas tarp tos dimensijos ir sluoksnio dimensijos reikšmės.
5. Sluoksnio dimensijos reikšmė su mažiausia *Sluoksvor* (svorio reikšmės nustatymas atitinkamai *Sluoksdim* reikšmei) reikšme yra labiausiai tinkamas sluoksnio storio dydis; Šis dydis turėtų būti tinkamiausias sluoksnio aukštis.

Pvz.: SLUOKS [] masyvas, kai konteinerio matmenys yra tokie: X=104, Y=96, Z=84.

SLUOKS[X] = (Sluoksdim, Sluoksvor)

Abs(70-70)+Abs(70-70)+Abs(70-70)+Abs(70-48)+Abs(70-48)+Abs(70-52)+Abs(70-52)+ Abs(70-52)=98;

SLUOKS[1]=(70,98).

Abs(24-24)+Abs(24-24)+Abs(24-24)+Abs(24-14)+Abs(12-14)+Abs(24-36)+Abs(24-36)+ Abs(24-36)=56;

SLUOKS[2]=(24,56).

Abs(14-24)+Abs(14-24)+Abs(14-24)+Abs(14-24)+Abs(14-14)+Abs(14-36)+Abs(14-36)+Abs(14-36)=106;

SLUOKS[3]=(14,106).

Abs(48-70)+Abs(48-70)+Abs(48-70)+Abs(48-70)+Abs(48-48)+Abs(48-40)+Abs(48-40)+Abs(48-40)=100;

SLUOKS[4]=(48,100).

Abs(40-24)+Abs(40-24)+Abs(40-24)+Abs(40-24)+Abs(40-48)+Abs(40-48)+Abs(40-40)+Abs(40-40)=80;

SLUOKS[5]=(40,80).

Abs(52-70)+Abs(52-70)+Abs(52-70)+Abs(52-70)+Abs(52-48)+Abs(52-48)+Abs(52-52)+Abs(52-52)=80;

SLUOKS[6]=(52,80).

Abs(36-24)+Abs(36-24)+Abs(36-24)+Abs(36-24)+Abs(36-48)+Abs(36-48)+Abs(36-36)+Abs(36-36)=72;

SLUOKS[7]=(36,72).

Yra 8 skirtingos dimensijos reikšmės. Kadangi dimensijos reikšmė 104 yra didesnė už konteinerio esamos padėties y dimensiją 96, todėl negalima nustatyti šios dimensijos kaip galimo sluoksnio storio dydžio. Turėdami paruoštą SLUOKS[] masyvą, reikėtų jo duomenis surikiuoti didėjančia tvarka atsižvelgiant į sluoksnio svorio dimensijos reikšmes (sluoksvor):

SLUOKS[X] = (Sluoksdim, Sluoksvor)

SLUOKS[1]=(24,56),

SLUOKS[2]=(36,72),

SLUOKS[3]=(52,80),

SLUOKS[4]=(40,80),

SLUOKS[5]=(70,98),

$SLUOKS[6]=(48,100)$,

$SLUOKS[7]=(14,106)$.

Mažiausia sluoksnio svorio reikšmė galėtų būti tinkamiausias sluoksnio storio dydis, todėl turėdami šį surūšiuotą sąrašą, galima pradėti pakavimą nuo perspektyviausio sluoksnio storio dydžio. Jei pakuojamas didelis kiekis skirtingų dėžių tipų, ypatingai svarbus faktorius - sprendimo laiko sumažinimas.

Iteracijų vykdymas. Iteracijos glaudžiai sujungia šešias galimas konteinerio padėtis. Vykstant iteracijoms supakuojamos visos konteinerio padėtys. Kiekviena unikali konteinerio padėtis yra traktuojama kaip konteinerio pakavimas.

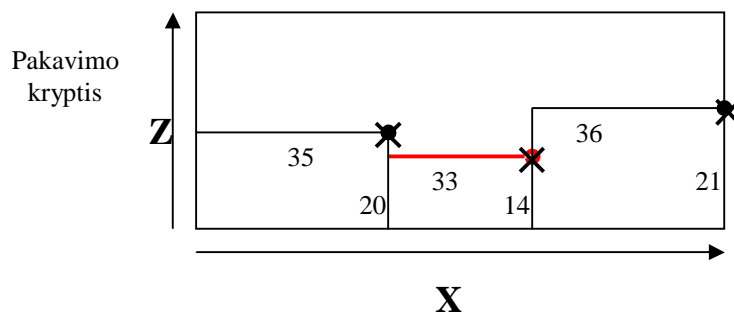
- Jeigu konteinerio trys dimensijos yra lygios (1 unikali dimensija), tai jis turi tik vieną padėtį.
- Jeigu konteineris turi 2 unikalias dimensijas, tai jis turi dvi padėtis.
- Jeigu konteineris turi 3 unikalias dimensijas, tai jis turi 6 padėtis.

Kiekvienos iteracijos metu, kiekviena konteinerio padėtis yra pakuojama vieną kartą kiekvienam elementui $SLUOKS []$ masyve. Kiekviena iteracija pradeda pakuoti naudodama pirminį sluoksnio storį, kuris yra $Sluoksdim$ reikšmė paimta iš $SLUOKS []$ masyvo. Taigi, jeigu yra 7 skirtingų dimensijų reikšmės $SLUOKS []$ masyve ir konteineris turi 3 unikalias dimensijas, programa potencialiai suskaičiuoja $6*7=42$ iteracijas. Vadinasi, programos sprendimo laikas priklauso nuo skirtingų dimensijų reikšmių ir bendro pakuojamų dėžių skaičiaus. Skirtingų tipų dėžių skaičius neturi jokios tiesioginės įtakos sprendimo laikui. Kol programos sluoksnių pakavimo būdas yra efektyvus, yra galimybė ypač per trumpą laiką efektyviai išnaudoti konteinerio talpą.

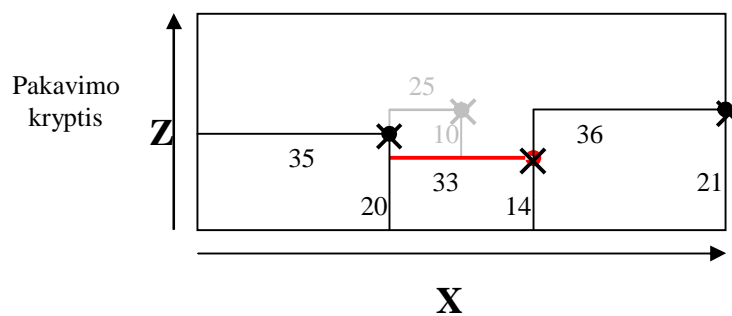
Kiekviena iteracija prasideda, kai konteineris yra tuščias, visos dėžės yra laisvos, konteinerio padėtis ir sluoksnio storis yra imami iš $SLUOKS []$ masyvo. Toliau einančios iteracijos pakeičia pradinio sluoksnio storį arba konteinerio padėtį. Geriausio pakavimo suradimo parametrai, grindžiami jau supakuota talpa, yra išsaugoti kaip einamasis pakavimo sprendinys.

Programa atlikdama sluoksnio pakavimą ir sienos statymą taip pat sprendžia ir dviejų dimensijų pakavimo problemas, pakuodama išilgai x ir z ašių (8 pav.). Kiekvieno dešinio kampo koordinatės (pažymėta kryžiuoku) yra laikomos dvejojo kreipimosi sąrašė. Kai dėžės yra supakuotos, koordinatė duomenys pasikeičia. Dvejojo kreipimosi sąrašas palengvina koordinatė pakeitimą. Šis būdas padeda išvengti sluoksnių ir konteinerio kraštų persidengimo.

Kiekvienas pakavimo veiksmas prasideda surandant mažiausią z reikšmę iš koordinacių duomenų sąrašo (mažiausia surasta reikšmė yra 14). Tuo pačiu šiame sąraše surandamas neužpildytas sluoksnio tarpas (pažymėtas raudona linija). Tikrinamos visos galimos dėžės, kol surandama tinkama dėžė užpildyti tarpą. Tinkama dėžė tai dėžė, kuri užpildydama tarpą atitinka esamo sluoksnio storį arba viršija esamą sluoksnio storį mažiausiu galimu dydžiu (pavyzdyje surastos tinkamos dėžės koordinatės: $x=25$, $z=10$). Jeigu nerandama tinkamos dėžės užpildyti tarpą, tarpas ignoruojamas. Tikrinama kiekviena dėžė visose padėtyse.



*PRI	KUMX	KUMZ	*SEK
0	35	20	
35+33	= 68	14	
68+36	= 104	21	0



*PRI	KUMX	KUMZ	*SEK
0	35	20	
35+25	= 60	14+10=24	Naujas įrašas
	68	14	
	104	21	0

8 pav. Sluoksnių pakavimas (dvejopo kreipimosi sąrašo duomenų laukai paašškinti 3 lentelėje).

3D objektų pakavimo programos naudojamos pagrindinės funkcijos aprašytos 6 lentelėje:

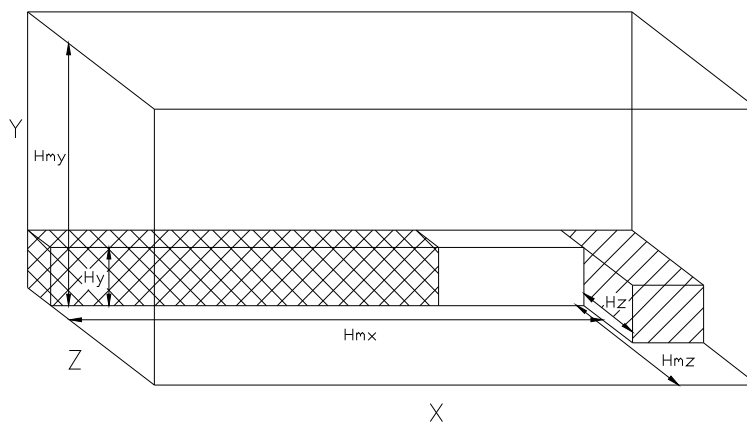
6 lentelė. Programos funkcijų aprašas.

Funkcija	Tikslas
<i>Paksluoks</i>	Atnaujinti kreipimosi sąrašą ir masyvą DEZSARAS [], kai dėžė supakuota.
<i>Rastimazz</i>	Nustatyti tarpą turintį mažiausią z reikšmę esamajame sluoksnyje.
<i>Rastidez</i>	Surasti dėžę, kuri geriausiai tinka esamam tarpui.
<i>Analyzdez</i>	Naudojant <i>Rastidez</i> analizuoti dėžės dimensijas.
<i>Tikrnitsurad</i>	Nustatyti, kurią dėžę supakuoti.
<i>Vykditerac</i>	Vykdyti iteracijas iškviečiant reikiamas funkcijas.
<i>Atask</i>	Daryti kopijas geriausių pakavimo duomenų.
<i>Isvestidezсарas</i>	Įrašyti pakavimo informaciją į failą.
<i>Grafinpakduom</i>	Surašyti pakavimo tvarką vizualizuojančiai programai.

Rastidez funkcija analizuoja nesupakuotas dėžes naudodama *Analyzdez* funkciją (9 pav.). Kiekvienai nesupakuotų dėžių skirtingai padėčiai *Analyzdez* parametrai yra tokie (7 lentelė):

7 lentelė. *Analyzdez* funkcijos parametrai kiekvienai nesupakuotų dėžių skirtingai padėčiai.

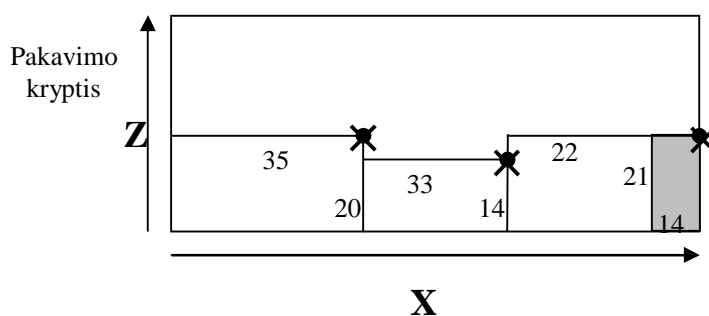
Hmx:	maksimali galima x dimensija esamam tarpui užpildyti.
Hy:	esamo sluoksnio storio reikšmė.
Hmy:	maksimali galima y dimensija esamam tarpui užpildyti.
H _z :	z dimensija esamam tarpui užpildyti.
Hmz:	maksimali galima z dimensija esamam tarpui užpildyti.
Dim1:	tinkamos dėžės padėties x dimensija.
Dim2:	tinkamos dėžės padėties y dimensija.
Dim3:	tinkamos dėžės padėties z dimensija.



9 pav. Analizdez funkcijos parametrai

Analizdez pirmumo tvarka ieško dėžės, kurios y dimensija yra artimiausia H_y , bet nedidesnė už H_{my} , ir kurios x dimensija artimiausia, bet nedidesnė už H_{mx} , ir z dimensija artimiausia H_z , bet nedidesnė už H_{mz} . Tai reiškia, jog ši funkcija atsižvelgia į y dimensiją, kai tarp dėžių turinčių tą pačią y dimensiją, tai žiūrima į x dimensiją. Kai tarp dėžių, turinčių tą pačią y ir x dimensiją, tai žiūrima į z dimensiją. Taip yra suskaičiuojami skirtumai tarp tarpų ir kiekvienos dėžės dimensijų, ir parenkama dėžė turinti mažiausiai skirtumų, kaip viena iš geriausiai tinkamų dėžių. Taip pat surandama antra dėžė su didesne y dimensija nei esamas sluoksniu storis, bet artimiausia esamam sluoksniu storiui. Tinkamos ir reikiamo storio (y reikšmės) dėžės supakuojamos.

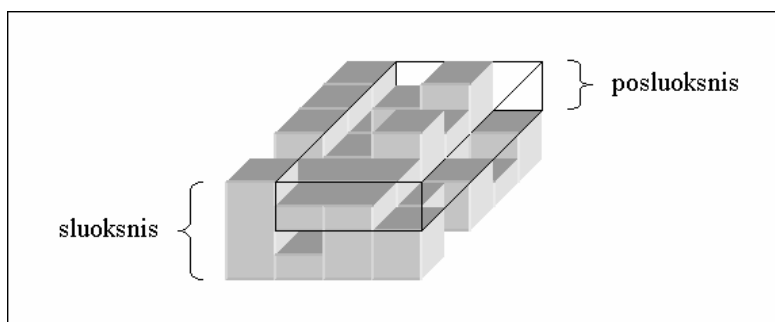
Jeigu nėra tinkamų dėžių esamam tarpui, nagrinėjamas tarpas lieka nesupakuotas (10 pav. pažymėtas pilka spalva).



*PRI	KUMX	KUMZ	*SEK
0	35	20	
35+33	= 68	14	
68+22	=90	21	Pašalinta
90+14	= 104	0	0

10 pav. Nesupakuoto tarpo palikimas (dvejopo kreipimosi sąrašo duomenų laukai paaiškinti 3 lentelėje).

Jeigu reikalinga, esamo sluoksnio storis padidinamas iki aukščiausios dėžės y dimensijos, kurią suranda *Rastidez* funkcija. Kai padidinamas esamo sluoksnio storis, nuo pradžios iki galo esamo pakuojamo sluoksnio bendras padidėjimas yra išsaugojamas kaip sluoksnio sluoksnyje kintamasis. Jeigu sluoksnio sluoksnyje kintamasis yra didesnis nei 0, prieš baigiant pakuoti esamą sluoksnį vyksta kitas pakavimas sluoksnio viduje dėl šios sluoksnio sluoksnyje reikšmės (11 pav.).



11 pav. Posluoksnių pakavimas

Posluoksnio pakavimas reiškia, kad programa gali utilizuoti daugiau konteinerio erdvės ir gauti daug geresnius rezultatus.

Kol `SLUOKS[]` masyvo įrašai yra naudojami pradėti kiekvienos iteracijos pirmą sluoksnį, tolesni sluoksniai reikalauja skaičiavimų determinuojančių gerą sluoksnio storio reikšmę. Šie skaičiavimai kopijuoja šias charakteristikas `SLUOKS[]` masyvo įrašams, bet naudoja tik dar nesupakuotom dėžėm. Geriausia sluoksnio reikšmė naudojama kaip sluoksnio storis tolesniems pakavimo sluoksniams. Sluoksniavimas tęsiasi kol konteineris negali įtalpinti tolimesnių sluoksnių. Konteineris tuomet laikomas užpildytu.

Kiekvienas pakavimas pagrįstas supakuotų dėžių ir nesupakuotų dėžių talpa, siekiant gauti konteinerio utilizavimo matus ir supakuotų dėžių talpos procentinį dydį. Kiekvienas pakavimas taip pat turi jungtinę konteinerio padėtį ir `SLUOKS[]` masyvo rodiklį. Geriausio pakavimo rezultato radimas yra galutinis modelio sprendinys.

Prieš gaunant geriausio sprendinio parametrus, iškviečiama funkcija *Atask*. Funkcija *Atask* iš naujo vykdo pakavimą naudodama geriausio surasto sprendinio parametrus. *Isvestidezsaras* funkcija generuoja ataskaitos failą, *Grafinpakduom* funkcija - programos vizualizavimo failą. Nesupakuotų dėžių informacija pateikiama ataskaitos failo pabaigoje.

2.3.5. Išvesties duomenų atvaizdavimas

Sukurta programa rezultatų duomenis išveda dviem būdais:

- ataskaitos failą,

- grafinį vaizdą.

Programa ataskaitą išveda atskirame faile, kurio pavadinimas “.out”. Ataskaitos failas yra sprendinio skaitmeninė santrauka - pakavimo koordinatės ir padėtyš kiekvienos supakuotos dėžės. Likusių nesupakuotų dėžių sąrašas yra pridamas pabaigoje ataskaitos failo. Konteinerio dimensijos kartu su pakavimo koordinatėmis ir kiekvienos supakuotos dėžės padėtyš yra išvedamos vizualiniame faile. Programa šį failą naudoja grafinėje sąsajoje sprendinio vizualizavimui.

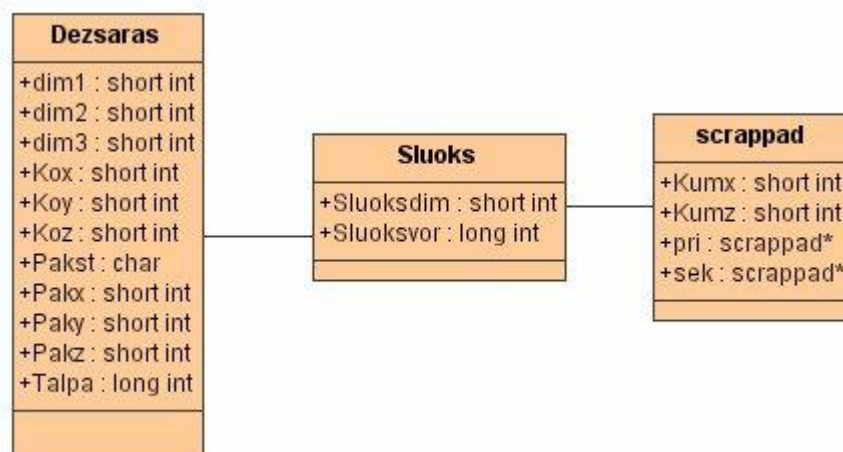
Grafinėi sąsajai išvesti naudojami du programiniai paketai DirectX 9.0 Software Development Kit (SDK) ir TrueVision3D 6.2. Šie paketai – tai trimačių vaizdų apdirbimo bibliotekos (funkcijų ir metodų rinkinys, generuojantis 3D vaizdus ir aplinką), kurios naudojamos grafiniam vaizdui projektuoti. Kadangi šis darbas yra mokomojo – tiriamojo pobūdžio, todėl darbe naudojamos nekomercinės programinės bibliotekos. Naudojant TrueVision3D 6.2 bibliotekų paketą, nubraižomas firminis logotipas, kuris yra matomas programos grafinės sąsajos lange.

Minėtų programų paketai yra reikalingi norint programą kompiliuoti arba toliau tobulinti, įtraukiant papildomas funkcijas ir reikalavimus.

Supakuotų dėžių konteineryje grafinis vaizdas gali būti valdomas kompiuterio klaviatūros mygtukų pagalba: galima išjungti arba įjungti konteinerio kraštinių rodyimą, sukroti dėžių vaizdą apie X ir Y ašį, balta spalva atskirai rodyti kiekvieną supakuotą dėžę.

2.4. 3D objektų pakavimo programos modelio specifikacija

Duomenų struktūrą galima atvaizduoti duomenų struktūros klasės diagrama (12 pav.)



12 pav. Duomenų struktūrų klasės

Aprašymas:

Masyvas **Dezsaras**[]

Pakst - pakavimo padėtis (0: nepakuotas; 1: supakuotas).

Sk - skaičius dėžių, kurios turi vienodas dimensijas.

Dim1 - ilgis vienos iš trijų dimensijų.

Dim2 - ilgis kitos dimensijos.

Dim3 - ilgis kitos dimensijos.

Kox - X koordinatė pakuojamos dėžės vietos.

Koy - Y koordinatė pakuojamos dėžės vietos.

Koz - Z koordinatė pakuojamos dėžės vietos.

Pakx - X supakuotos dėžės padėties dimensija.

Paky - Y supakuotos dėžės padėties dimensija.

Pakz - Z supakuotos dėžės padėties dimensija.

Talpa - dėžės talpa ($Dim1 * Dim2 * Dim3$).

Masyvas **Sluoks**[]

Sluoksdim - dimensijos reikšmė.

Sluoksvor - svorio reikšmės nustatymas atitinkamai Sluoksdim reikšmei.

Dvejopo kreipimosi sąrašas **scrappad**

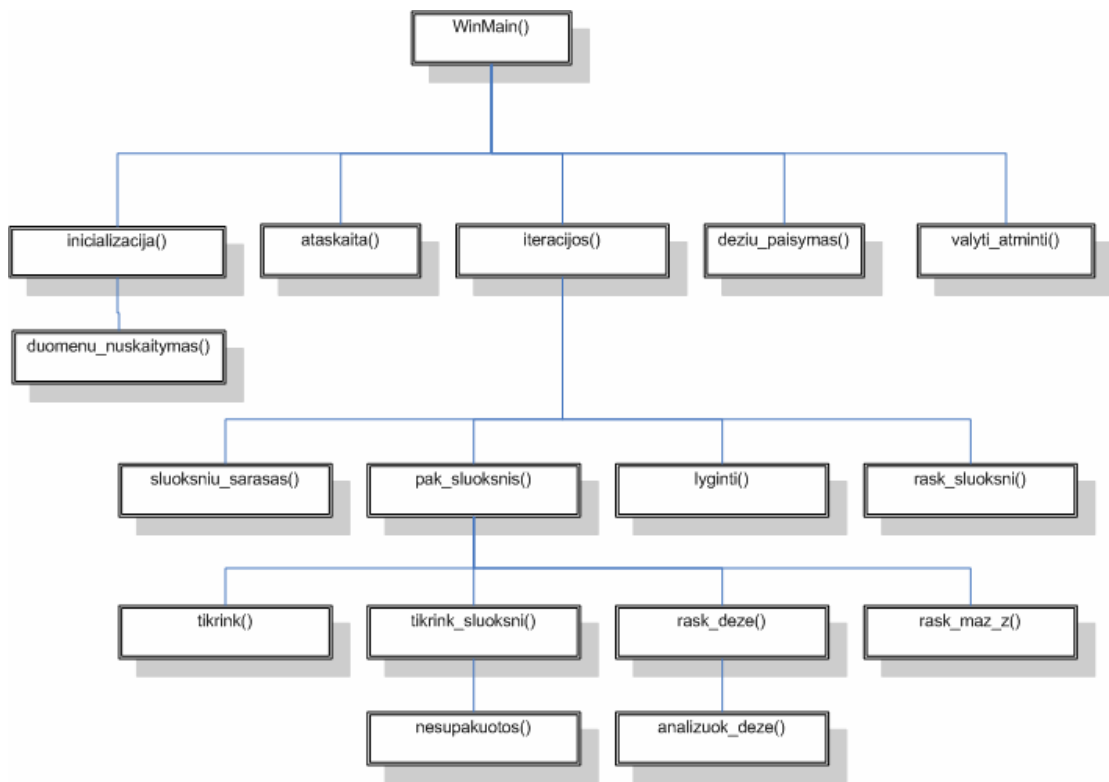
Kumx - nusako X koordinatę tarpo dešinio kampo.

Kumz - nusako Z koordinatę tarpo dešinio kampo.

***pri** - rodyklė, kuri saugo ankstesnio įrašo adresą.

***sek** - rodyklė, kuri nurodo sekančio įrašo adresą.

Programos veikimo hierarchinė diagrama (13 pav.):



13 pav. Programos veikimo hierarchinė diagrama

Aprašymas:

WinMain – pagrindinė programa.

Inicializacija – dėžių koordinacių nuskaitymas iš duomenų failo „matmenys.txt“.

Duomenu_nuskaitymas - konteinerio bei dėžių matmenų nuskaitymas iš duomenų failo.

Ataskaita - naudojant surastus parametrus, pakuojama pagal geriausią surastą sprendinį ir perduodama ataskaitos failui.

Iteracijos - atliekamos skaičiavimų iteracijos.

Deziu_paisymas - vaizdo perpiešimas.

Valyti_atminti - grafikos elementų išvalymas iš atminties.

Sluoksniu_sarasas - sudaromas visų įmanomų sluoksnių sąrašas pagal duotus aukščius.

pak_sluoksnis - supakuojamos visos dėžės bei sutvarkomi sluoksnių įrašai.

lyginti - pagalbinė funkcija rūšiavimui.

rask_sluoksnis - suranda patį tinkamiausią sluoksnio aukštį, žiūrint į likusias nesupakuotas dėžes bei įmanomus sluoksnių aukščius.

Tikrink - tikrinamos atrinktos dėžės ir sluoksnio padėty.

tikrink_sluoksnis - supakavus kiekvieną dėžę, patikrinama 100% pakavimo sąlyga.

rask_deze - surandama pati tinkamiausia dėžė pagal visas įmanomas 6 orientacijas turimai tuščiai ertmei.

rask_maz_z - surandama pirmoji pakavimo vieta.

Nesupakuotos - ataskaitos duomenys suvedami į rezultatų failą.

analizuok_deze - analizuojamos visos nesupakuotos dėžės, taikant į turimą tuščią ertmę.

2.5. Rizikos įvertinimo ir mažinimo planas

Projekto rizikos:

Ar galimi reikalavimų pasikeitimai?

Kuriant programą bus laikomasi reikalavimų dokumento nurodymų. Reikalavimai gali keistis tik tuo atveju, jeigu programos kūrimo proceso metu bus pasiūlyta patobulinti keletą programos detalių arba vartotojui atsirastų poreikis įtraukti į programą tam tikrus papildomus reikalavimus.

Su vartotoju susijusios rizikos:

Ar gali vartotojas atsisakyti produkto?

Vartotojas gali atsisakyti produkto. Bet projektas yra kuriamas ir moksliniais tikslais.

Sprendimas: Kuriant 3D objektų pakavimo programą siekiama sukurti kiek galima lankstesnę sistemą, kurią būtų galima reikalui esant nesunkiai išplėsti papildant naujomis funkcionalumo ar vizualizacijos galimybėmis. Todėl sukurtą programos universalų paketą galima lengvai pritaikyti pagal kiekvieno vartotojo poreikius.

3D objektų pakavimo programa yra aktuali daugeliui transportavimo kompanijų, gabenančių krovinius, supakuotus stačiakampėse dėžėse, todėl yra mažai tikimybės, kad vartotojai produkto atsisakys.

Proceso rizika:

Ar gali atsirasti kiti, papildomi, darbai?

Gali atsirasti papildomi darbai, kurie apsunkintų savalaikį projekto užbaigimą.

Sprendimas: Paaiškinti vartotojui apie galimą projekto uždelsimą, uždelsimo priežastis ir galimas pasekmes. Derėtis su vartotoju, pabrėžiant, kad projekto kūrimo uždelsimas gali duoti tik teigiamus rezultatus, nes per ilgesnį laiką bus sukurtas efektyvesnis ir tobulesnis produktas.

Priešingu atveju, vartotojui nesutikus dėl projekto laiko uždelsimo, dirbant teikti prioritetus svarbiausioms projekto dalims, jog vartotojui atrodytų, kad praktiškai visas funkcionalumas egzistuoja ir projektas pristatytas laiku. Pilnas produkto funkcionalumas įdiegiamas vėliau.

Techninės rizikos:

Ar gali sugesti kompiuteriai?

Kompiuteriai gali sugesti. Nors šių dienų kompiuterinė technika ganėtinai patikima, tačiau tokia tikimybė išlieka. Sugedus kietajam diskui galima prarasti visą informaciją. Tai įtakotų projekto kūrimo eigą.

Sprendimas: Kuriamos programos projekto kopija periodiškai yra įrašoma į kompaktinį diską CD-R, taip pat į kitų kompiuterių kietuosius diskus. Tokiu atveju prarastos informacijos kiekis būtų nedidelis.

Darbo priemonių rizikos:

Ar yra tikimybė, kad kokia nors projektui reikalinga priemonė nebus galima pasinaudoti?

Tikimybė yra labai maža, kadangi šiuo metu visos priemonės yra turimos, o atsiradus poreikiui naujoms priemonėms, jas būtų nesunku gauti universitete.

Darbuotojų komandos dydžio ir patirties rizikos:

Ar komanda pakankamo dydžio?

Komanda pradinei projekto daliai atlikti yra pakankamo dydžio. Nors ją realiai sudaro vienas narys, tačiau jo aplinkoje yra nemažai kolegų, kurie gali suteikti profesionalią pagalbą. Plečiantis projektui komandoje turėtų būti daugiau narių.

Ar komandos nariai pakankamai kompetentingi?

Komandos narys yra pakankamai kompetentingas projekto vykdymui. Jis turi pakankamai teorinių ir praktinių žinių 3D objektų pakavimo problemoms spręsti.

Rizikas įvertiname 8 lentelėje:

8 lentelė. Rizikos vertinimo lentelė.

Rizika	Tikimybė (%)	Įtaka
Kompiuterių gedimas	20	3
Pavėluotas pristatymas	30	1
Reikalavimų pasikeitimai	5	3
Kompetencijos trūkumas	10	3
Kietojo disko gedimas	20	2
Vartotojo atmetimas	30	3

Papildomi darbai	50	1
------------------	----	---

Įtakos vertinimo skalė:

- 1 – Tragiška
- 2 – Labai blogai
- 3 – Pakenčiama
- 4 – Neturi įtakos

3. VARTOTOJO DOKUMENTACIJA

3.1. Programinės įrangos paskirtis ir galimybės

3D objektų pakavimo programinė įranga skirta stačiakampio gretasienio formos objektų (dėžių) pakavimo į tokios pačios formos konteinerį uždaviniui spręsti. Programinė įranga pagal duotas konteinerio ir dėžės dimensijas suskaičiuoja ir pateikia geriausią stačiakampių dėžių pakavimo konteineryje sprendimo būdą. Programa skirta naudotis įmonėms, kurioms yra aktualu, kuo geriau kraunant stačiakampius objektus išnaudoti pakuojamą erdvę. Kitaip tariant, kuo daugiau stačiakampėse dėžėse supakuotų gaminių sukrauti į konteinerį maksimaliai išnaudojant jo plotą.

Programinės įrangos paketas užima nedaug vietos kompiuterio atmintyje ir yra paprastai įdiegiamas. Vartotojas pats paruošia programai įvedimo duomenis (konteinerių ir dėžių dimensijas), kuriuos programa apdoroja ir išveda dėžių supakuotų į konteinerį rezultatą – ataskaitą ir grafinį vaizdą. Naudojant šią programą galima efektyviau išnaudoti sutalpinimo erdvę, taupyti laiko ir pinigų sąnaudas, gauti lūkesčius tenkinančius darbo rezultatus.

Programinės įrangos galimybės:

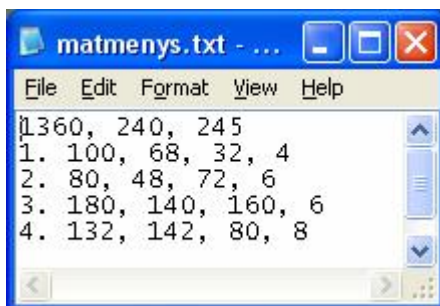
- Programa parodo, kaip reikėtų efektyviai išnaudoti konteinerio erdvę į jį pakuojant stačiakampes dėžes.
- Programa gali supakuoti stačiakampio tipo dėžes skirtingomis orientacijomis.
- Programa yra ypatingai efektyvi, kai reikia supakuoti didelį kiekį ir skirtingų tipų dėžes.
- Programa per trumpą laiko tarpą suranda geriausią dėžių pakavimo į konteinerį sprendinį.

- Darbo rezultatai pateikiami dviem būdais: ataskaita ir supakuotų dėžių konteineryje grafinis vaizdas.

3.2. Naudojimosi programa instrukcija

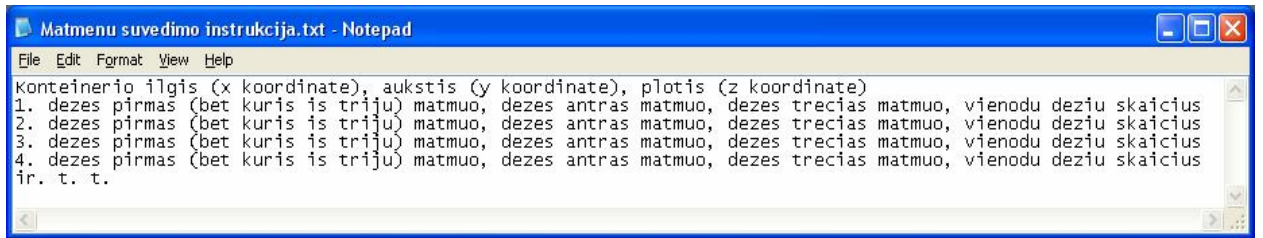
Vartotojas pradėdamas naudotis 3D objektų pakavimo programa pirmiausia turi paruošti įvedimo duomenis programai. Konteinerio ir dėžių dimensijų duomenys turi būti pateikti atskirame faile.

- Informaciją vartotojas turi suvesti į failą pavadinimu „matmenys.txt“. Šį failą galima sukurti naudojantis Microsoft Windows ir Microsoft Office standartinio paketo programomis ir daugelį kitų sukuriančių *.txt formato failą. Galima naudoti programas: Notepad, WordPad, Microsoft Word ir kt.
- Įvesti duomenis į failą vartotojas privalo apibrėžtu formatu (14 pav.):



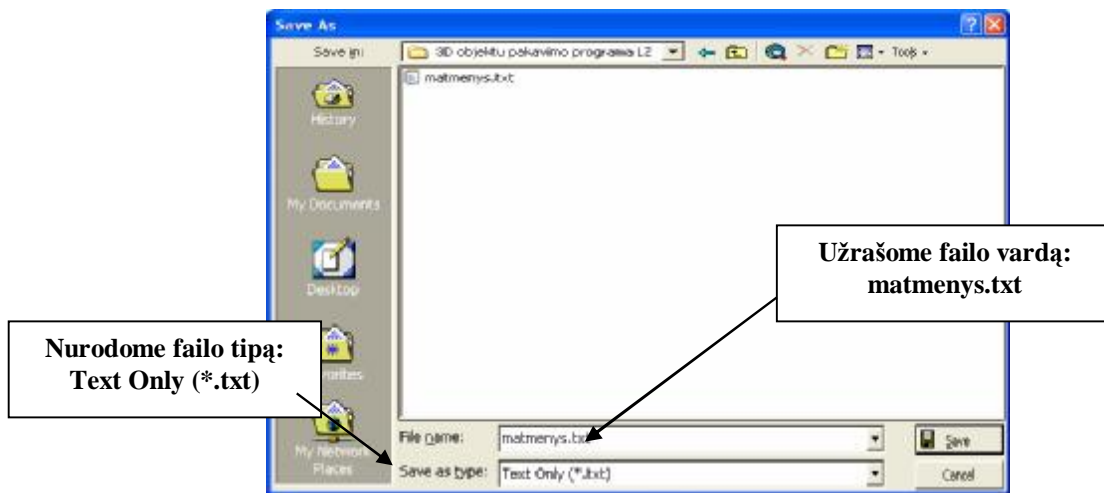
14 pav. Įvedamų duomenų failo pavyzdys.

Pirmoje eilutėje nurodomi trys skaičiai, kurie nusako konteinerio x, y, z dimensijas. Konteinerio matmenų išdėstymo tvarka yra labai svarbi, todėl pirmasis dydis turi atitikti konteinerio ilgį, antrasis – aukštį, o trečiasis – plotį. Visos sekančios eilutės pateikia informaciją apie dėžes. Pirmas skaičius kiekvienoje eilutėje nusako dėžės žymę. Antras, trečias ir ketvirtasis skaičiai yra atitinkamai kiekvieno tipo dėžės x, y ir z matmenys. Eiliškumo tvarka dėžės matmenų nėra svarbi. Penkta reikšmė duomenų eilutėje nurodo to pačio tipo dėžių skaičių. Nors kableliai tarp nurodomų reikšmių nėra privalomi, tačiau bent vieno tarpo simbolis turėtų skirti skaičius. Kokiais matų vienetais bus įvesti duomenys į failą, tokiais pačiais matų vienetais bus pateikiamas rezultatas. Įvedimo duomenų aiškinamasis pavyzdys (15 pav.):



15 pav. Įvedimo duomenų aiškinamasis pavyzdys.

- Suvedus duomenis į failą atliekami sekantys veiksmai: meniu pasirinkimo grafoje spaudžiama „File, Save As“ ir atsidariusiame lange „Save As“ (16 pav.) užrašome failo vardą ir nurodome failo tipą:



16 pav. Įvesties failo išsaugojimo procedūra.

- Sukurtą failą reikia įkelti į tą patį disko aplanką, kuriame yra vykdoma programa.
- Programa paleidžiama paspaudus failą „pakuotojas.exe“.
- Programa suskaičiuoja vartotojo suvestus duomenis ir pateikia rezultatus, sukurdamą tame pačiame disko aplanke, kur yra vykdoma programa, ataskaitos failą pavadinimu „.out“.
- Dėžių supakuotų konteineryje grafinis vaizdas pateikiamas iš kart programai suskaičiavus rezultatus. Jį galima valdyti kompiuterio klaviatūros mygtukais:

- § mygtukas „Delete“ - išjungti arba įjungti konteinerio rodymą;
- § kursoriaus mygtukai aukštyn, žemyn - sukioja dėžes apie Y ašį;
- § kursoriaus mygtukai į kairę ir dešinę - sukioja dėžes apie X ašį;
- § mygtukai „1“ ir „2“ – parodo atskirai kiekvieną supakuotą dėžę, išryškinant ją balta spalva. Mygtukas „1“ keičia dėžes iš kairės pusės, mygtukas „2“ – iš dešinės;

§ Mygtukas „Esc“ – darbo pabaiga (užverčiamas programos langas).

3.3. Programinės įrangos įdiegimo dokumentacija

Programinė įranga įdiegiama sekančiais etapais:

1. Programinė įranga iš CD kompaktinio disko įrašoma į kompiuterį. Programinį paketą sudaro:
 - § 3 failai su plėtiniu „*.dll“ (3D objektų vaizdavimui naudojamos bibliotekos);
 - § failas pavadinimu „instal.bat“ (priregistruoja bibliotekas);
 - § failas pavadinimu „pakuotojas.exe“ vykdo programą.
2. Į programos aplanką reikia įkelti sukurtą įvedimo duomenų failą „matmenys.txt”.
3. Aktyvuoti failą „instal.bat“, kuris priregistruoja bibliotekas.
4. Aktyvuoti programą, paspaudžiant failą „pakuotojas.exe“.

4. BANDOMASIS EKSPLOATACINIS TYRIMAS

Programa buvo testuojama konkrečiose įmonėse, kurioms yra aktualios 3D stačiakampių objektų pakavimo į konteinerius problemos. Šioms įmonėms buvo pristatytas sukurtas produktas, ir pagal įmonės vadovų pageidavimą, programinė įranga buvo pristatyta jų įmonėms darbuotojams.

UAB „Link Interiors Baltic“ perka baldus iš Lietuvos gamintojų ir eksportuoja juos į Angliją, Japoniją, Švediją ir kitas užsienio šalis. Šiai įmonei yra ypač svarbu kuo maksimaliau išnaudoti autofurgono talpą, kraunant į jį baldus, supakuotus stačiakampės formos dėžėse.

Sukurta programinė įranga buvo testuojama pagal konkrečius šios įmonės darbuotojų pateiktus duomenis (9 lentelė).

9 lentelė. UAB „Link Interiors Baltic“ pakuojamų krovinių duomenys

Autofurgono matmenys (ilgis x aukštis x plotis)	1360x240x245		
Pakuojamų baldų matmenys ir jų kiekis			
Eilės nr.	Baldų pavadinimai	Matmenys	Kiekis
1.	Lempos staliukai	55x55x12	4
2.		65x65x13	4
3.	Kavos staliukas	125x65x14	4
4.	Valgomieji stalai	130x80x12	8
5.		155x90x13	8
6.		185x90x13	14
7.		205x95x13	10
8.	Kėdės (supakuotos po 2)	46x70x110	10
9.		46x60x100	10
10.		115x40x125	8

11.	Knygų lentynos	80x45x125	8
12.		110x40x160	12
13.	TV staliukas	102x65x50	6
14.	Indaujos	175x50x89	8
15.		120x50x89	6
16.	Serviravimo staliukai	125x50x90	6
17.		145x50x90	8
18.	Gėlių stovėlis	35x45x90	8
19.	Krėslas	70x70x140	12
20.	Spinta 2 durų	125x65x200	6
21.	Stalo prailgintojai	90x80x8	2
22.		90x90x8	2
23.	Suolai	130x40x12	2
24.		155x40x12	2
25.		185x40x12	1

Pagal turimus duomenis, įmonės darbuotojai patys sukūrė įvedimo duomenų failą (1 priedas). Pagal jų pateiktus duomenis programa sugeneravo rezultatus ir išvedė juos ataskaitoje (2 priedas) ir parodė grafinį supakuotų dėžių vaizdą (17 pav.).

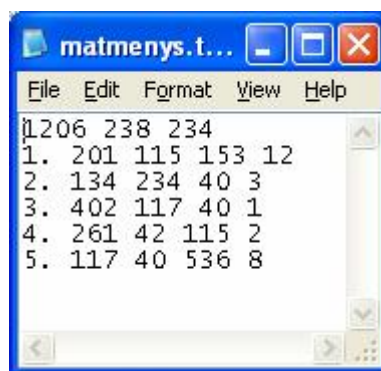


17 pav. UAB „Link Interiors Baltic“ supakuoto autofurgono grafinis dėžių vaizdas (dėžių padėtis pasukta x ašies kryptimi ir pažymėta viena supakuota dėžė balta spalva; konteinerio vaizdas išjungtas).

Įmonės darbuotojai teigia, kad iš 169 dėžių sandėlio darbuotojai į konteinerį supakuoja tik 131 dėžę. Programa supakuoja 157 dėžes ir užpildo 87 proc. konteinerio talpos. Vadinas, programos testavimas įrodo, jog sukurta programa suranda geriausią supakuotų dėžių į konteinerį sprendinį.

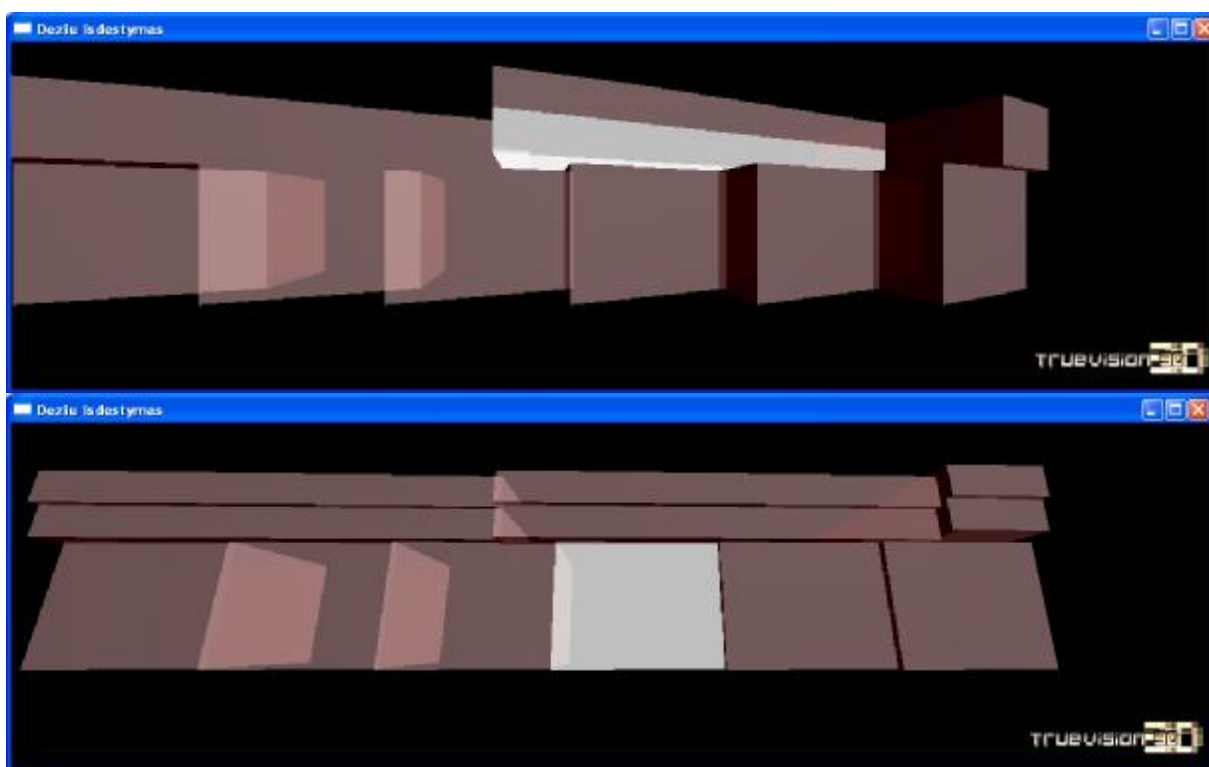
Programos testavimas buvo atliktas su įmonės UAB „Evelkas“ pateiktomis konteinerio ir dėžių dimensijomis. Ši firma importuoja prekes iš JAV jūriniais konteineriais. Naudojamo jūrinio konteinerio matmenys 1206 x 238 x 234 (ilgis x aukštis x plotis). Kroviniai – tai automobilių eksploatacinės medžiagos, įrankiai stiklų keitimui, automobilių stiklų gumos. Visos prekės supakuotos į stačiakampes dėžes.

Firmos darbuotojai naudodamiesi vartotojo dokumentacija sukūrė įvesties duomenų failą (18 pav.):



18 pav. UAB „Evelkas“ darbuotojų sukurtas įvesties duomenų failas

Pagal pateiktus duomenis, programa sugeneravo rezultatus ir išvedė juos ataskaitoje (3 priedas) ir parodė grafinį supakuotų dėžių vaizdą (19 pav.).



19 pav. UAB „Evelkas“ supakuoto jūrinio konteinerio grafinis dėžių vaizdas (viršutiniame paveiksle dėžių padėtis pasukta x ašies kryptimi, apatiniame - dėžių padėtis pasukta y ašies kryptimi, pažymėta po vieną supakuotą dėžę balta spalva; konteinerio vaizdas išjungtas).

Programa iš pateiktų 26 dėžių supakavo 22 dėžes ir užpildė 96 proc. konteinerio talpos. Liko nesupakuotos tik 4 dėžės. Tai rodo, jog sukurta programa gali pakankamai efektyviai išspręsti 3D stačiakampių dėžių pakavimo uždavinius.

5. PRODUKTO KOKYBĖS ĮVERTINIMAS

Sukurtos programinės įrangos kokybė vertinama pagal tokius kriterijus (10 lentelė):

10 lentelė. Programinės įrangos projektavimo kriterijai.

Kriterijai	Vertinimas prioritetais
Naudojamumas	3
Saugumas	3
Efektyvumas	4
Teisingumas	5
Patikimumas	4
Palaikomumas	5
Testuojamumas	2
Lankstumas	3
Suprantamumas	4
Pakartotinis panaudojamumas	4
Pernešamumas	2
Įsiliejamumas	3

Kokybė vertinama skalėje nuo 1 (žemiausia kategorija) – 5 (aukščiausia kategorija).

6. PROGRAMINĖS ĮRANGOS TESTAVIMAS

Sukurta programinė įranga buvo testuojama pagal šias testavimo metodikas:

- Testavimui naudosime struktūrinį testavimą, dar vadinamą „baltos dėžės“ testavimu, kadangi žinoma programos struktūra ir jos veikimas.
- Pavienių objektų testavimą.

- Integruotos sistemos testavimą.

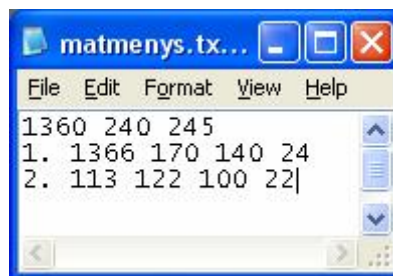
Vartotojo sąsajos testavimas verifikuoja vartotojo sąveiką su programine įranga. Vartotojo sąsajos testavimas reikalingas tam, kad užtikrinti, jog vartotojo sąsaja suteikia vartotojui galimybę pasiekti testuojamo objekto funkcijas. Taip pat šis testavimas užtikrina, jog objektai vartotojo sąsajos funkcijoje veikia kaip tikimasi ir atitinka bendrus ar sukurtus standartus.

Vartotojo funkcionalumo strategija identifikuoja klaidas, susijusias su teisingu programos funkcijų atlikimu, t. y. ar programa atlieka funkcijas teisingai, pagal reikalavimus. Šio testo tikslas yra verifikuoti tinkamą duomenų priėmimą, apdorojimą ir išvedimą bei tinkamą verslo taisyklių įdiegimą. Šio tipo testavimas yra paremtas juodos dėžės principu.

Testavimas buvo vykdomas sekančiais etapais:

- § Suprojektuoti testavimo atvejai.
- § Paruošti testavimo duomenys.
- § Paleidžiama programa su testavimo duomenimis.
- § Palyginami rezultatai su testavimo atvejais.

Sukurta programinė įranga buvo testuojama paruošiant klaidingus įvedimo duomenis. Įvesties faile buvo suvestos dėžių dimensijos didesnės už konteinerio dimensijas (20 pav.).



20 pav. Testiniai duomenys

Programa atlikus skaičiavimus, dėžės, kurių dimensijos didesnės už konteinerio, nesupakavo, o pateikė nesupakuotų dėžių sąrašą (21 pav.). Lyginant rezultatus su testavimo atveju, programoje klaidų neaptikta.

```

*** ATASKAITA ***

Bendras atliktu iteracijų skaičius      : 18
Geriausias sprendimas rastas           : Iteracijų: 1 variantas: 1
Bendras dežių skaičius                 : 46
Supakuotų dežių skaičius               : 22
Bendra dežių talpa                     : 810588672
Konteinerio talpa                      : 79968000
Geriausias sprendinys išnaudoja talpa  : 30329200 iš 79968000
Procentinis konteinerio talpos išnaudojimas : 37.926670 %
Procentine supakuotų dežių talpa       : 3.741627 %
Konteinerio esama padėtis              : X= 1360; Y= 240; Z= 245
-----
Nr: PAKST  DIMEN-1  DIMEN-2  DIMEN-3  KO-X    KO-Y    KO-Z    PAKX    PAKY    PAKZ
-----
25  1      113      122      100      0       0       0       113     122     100
26  1      113      122      100     113      0       0       113     122     100
27  1      113      122      100     226      0       0       113     122     100
28  1      113      122      100     339      0       0       113     122     100
29  1      113      122      100     452      0       0       113     122     100
30  1      113      122      100     565      0       0       113     122     100
31  1      113      122      100     678      0       0       113     122     100
32  1      113      122      100     791      0       0       113     122     100
33  1      113      122      100     904      0       0       113     122     100
34  1      113      122      100    1017      0       0       113     122     100
35  1      113      122      100    1130      0       0       113     122     100
36  1      113      122      100    1243      0       0       113     122     100
37  1      113      122      100      0       0       100     113     122     100
38  1      113      122      100     113      0       100     113     122     100
39  1      113      122      100     226      0       100     113     122     100
40  1      113      122      100     339      0       100     113     122     100
41  1      113      122      100     452      0       100     113     122     100
42  1      113      122      100     565      0       100     113     122     100
43  1      113      122      100     678      0       100     113     122     100
44  1      113      122      100     791      0       100     113     122     100
45  1      113      122      100     904      0       100     113     122     100
46  1      113      122      100    1017      0       100     113     122     100

*** Nesupakuotų dežių sąrašas ***
1 1366 170 140
2 1366 170 140
3 1366 170 140
4 1366 170 140
5 1366 170 140
6 1366 170 140
7 1366 170 140
8 1366 170 140
9 1366 170 140
10 1366 170 140
11 1366 170 140
12 1366 170 140
13 1366 170 140
14 1366 170 140
15 1366 170 140
16 1366 170 140
17 1366 170 140
18 1366 170 140
19 1366 170 140

```

21 pav. Testavimo duomenų ataskaita

Testuojant programą buvo pateikti atsitiktiniai įvesties failo duomenys (22 pav).

```

matmenys...
File Edit Format View Help
1206 238 234
1. 119 300 100 4
2. 80 120 90 6
3. 180 140 160 6
4. 132 142 80 8
5. 145 95 120 6

```

22 pav. Atsitiktiniai parinkti įvesties failo duomenys.

Programa su testavimo duomenimis nepasileido. Programiniame kode surasta klaida. Grafiniame vaizde pirma supakuota dėžė pažymima balta spalva. Nustatyta, kad pirma sąraše esanti dėžė buvo įtraukta į nesupakuotų dėžių sąrašą, o programa žymėdama dėžes, neatsižvelgė į supakuotas ir nesupakuotas dėžes.

Programos testavimas buvo atliktas sėkmingai, visos surasto klaidos ištaisytos.

IŠVADOS

1. Atlikta užsienio autorių sukurtų 3D objektų pakavimo algoritmų ir komercinių programinių paketų analizė.
2. Pagal atliktą analizę, nustatyti trijų dimensijų objektų pakavimo metodų privalumai ir trūkumai.
3. Atsižvelgiant į analizės rezultatus, sudarytas 3D stačiakampių objektų pakavimo į tokios pačios formos konteinerius modelis.
4. Pagal modelį sukurta programinė įranga trijų dimensijų stačiakampių objektų pakavimo uždaviniams spręsti.
5. Atlikti tyrimai įrodė, jog pasiūlytas 3D stačiakampių objektų pakavimo metodas yra efektyvus ir naudingas.

LITERATŪRA

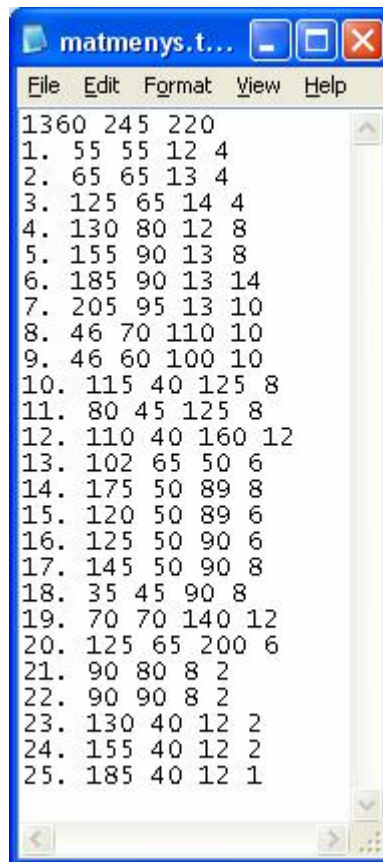
1. Astrokettle Algorithms, Inc. 3D Load Packer software [interaktyvus]. [žiūrėta 2004-03-14]. Prieiga per Internetą: <http://www.astrokettle.com/pr3dlp.html>
2. Bischoff, E.; Dowsland, W. B. An application of the micro to product design and distribution. *Journal of the Operational Research Society*, 1982. Nr. 33 (3), p. 271-281.
3. Bischoff, E. E.; Janetz, F.; Ratcliff, M. S. W. Loading pallets with non-identical items, *European Journal of Operational Research*, 1995. Nr. 84, p. 681-692.
4. Bischoff, E. E.; Marriott, M. D. A comparative evaluation of heuristics for Container loading. *European Journal of Operational Research*, 1990. Nr. 44 (2), p. 267-276.
5. CAPE Systems, Inc. Truckfill packaging design and pallet loading software [interaktyvus]. [žiūrėta 2003-11-12]. Prieiga per Internetą: <http://www.capesystems.com>
6. Chen, C. S.; Lee, S. M.; Shen, Q. S. An analytical model for container loading problem. *European Journal of Operational Research*, 1995. Nr. 80, p. 68-76.
7. DirectX 9.0 Software Development Kit (SDK) with the DirectX 9.0b Runtime [interaktyvus]. [žiūrėta 2004-04-19]. Prieiga per Internetą: <http://www.microsoft.com/downloads/details.aspx?FamilyID=1d97f320-9dfd-4e7a-b947-3a037ccf84af&DisplayLang=en>
8. Faina L. A global optimization algorithm for the three-dimensional packing problem, *European Journal of Operational Research*, 2000. Nr. 126, p. 340-354.
9. Gehring, H.; Menschner, K.; Meyer, M. A computer-based heuristic for packing pooled shipment containers. *European Journal of Operational Research*, 1990. Nr. 44 (2), p. 277-289.
10. George, J. A.; Robinson, D. F. A heuristic for packing boxes into a container. *Computers and Operational Research*, 1980. Nr. 7, p. 147-156.
11. Han, C. P.; Knott, K.; Egbelu, P. J. A heuristic approach to the three-dimensional cargo-loading problem. *International Journal of Production Research*, 1989. Nr. 27 (5), p. 757-774.
12. Liu, N. C.; Chen, L. C. A new algorithm for container loading. *Compsac 81-5th International Computer Software and Applications Conference Papers (Chicago: IEEE)*, 1981. p. 292-299.

13. Magic Logic Optimization, Inc. Cube-IQ Load Optimization Software [interaktyvus]. [žiūrėta 2003-11-12]. Prieiga per Internetą: <<http://www.magiclogic.com/cube-iq.cfm>>
14. Martello, S.; Pisinger, D.; Vigo, D. The three-dimensional bin packing problem. operations research, 2000. Nr. 2, p. 256-267.
15. Mohanty, B. B.; Mathur, K.; Ivancic, N. J. Value considerations in three-dimensional packing – A heuristic procedure using the fractional knapsack problem. European Journal of Operational Research 1994. Nr. 74, p. 143-151.
16. TrueVision3D 6.2 Software [interaktyvus]. [žiūrėta 2004-04-19]. Prieiga per Internetą: <<http://www.truevision3d.com/download.php?unsecure=2&file=setup.exe>>

PRIEDAI

1 PRIEDAS

Įmonės UAB „Link Interiors Baltic“ įvesties duomenų failas:



```
matmenys.t...
File Edit Format View Help
1360 245 220
1. 55 55 12 4
2. 65 65 13 4
3. 125 65 14 4
4. 130 80 12 8
5. 155 90 13 8
6. 185 90 13 14
7. 205 95 13 10
8. 46 70 110 10
9. 46 60 100 10
10. 115 40 125 8
11. 80 45 125 8
12. 110 40 160 12
13. 102 65 50 6
14. 175 50 89 8
15. 120 50 89 6
16. 125 50 90 6
17. 145 50 90 8
18. 35 45 90 8
19. 70 70 140 12
20. 125 65 200 6
21. 90 80 8 2
22. 90 90 8 2
23. 130 40 12 2
24. 155 40 12 2
25. 185 40 12 1
```