KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Artūras Ražas

Dėžių pakavimo trimatėje erdvėje algoritmas ir jo taikymas logistikos uždaviniams spręsti

Magistro darbas

Darbo vadovas
doc. dr. Rimantas Butleris

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Artūras Ražas

Dėžių pakavimo trimatėje erdvėje algoritmas ir jo taikymas logistikos uždaviniams spręsti

Magistro darbas

Kalbos konsultantė
Lietuvių k. katedros doc.
J. Mikelionienė
2006-05

Vadovas
doc. dr. Rimantas Butleris
2006-05

Recenzentas
doc. dr. Stasys Maciulevičius
2006-05

Atliko
IFM-0/2 gr. stud.
Artūras Ražas
2006-05-15

Kaunas, 2006

Turinys

Summary

Whether you are in industrial manufacturing striving to optimize your supply chain, international or domestic carrier committed to lower your operational costs, retailer dedicated to run your distribution network more efficiently, or anywhere else where the words "cargo", "freight", "shipment" are in your business language, automated load planning and optimization will significantly improve your business process.

While the concept of computerized simulation of 3D load building is not new, with some companies offering software solutions for "virtual" loading, no one has a single product that is capable to handle complex variety of *business rules* and *constraints* of the modern transportation industry.

We are analyzing the problems related with 3D box loading and then to analyze existing 3D load building systems. After accumulating all analyzes data we are having enough experience to create our own algorithm. And we did it! We create 3D box loading algorithm witch is fast and accurate and 3D box loading system which is using this new algorithm.

To ensure that we succeeded we compare this new system with existing 3D box loading software to make sure that it’s really useful for solving logistic problems.

After testing we are sure that we made a big job and we have created system, which can compete with the logistic IT leaders made solutions.

1 Įvadas

Pastarieji metai Lietuvos vežėjų įmonėms buvo sunkūs dėl brangstančio kuro ir didėjančios konkurencijos. Atlikus preliminarią logistikos kompanijų apklausą apie tai ar jos naudoja kokią nors programinę įrangą, padedančia taupiau sukrauti prekes į krovininį automobilį, paaiškėjo, kad Lietuvos vežėjai netik, neturi pakankamai informacijos apie galimus sprendimus ir netiki, kad jiems tokių sistemų naudojimas padėtų taupyti resursus.

Nuspręsta išanalizuoti galimybę sukurti informacinę sistemą, kuri padėtų pervežimo kompanijoms optimaliau išnaudoti krovininius automobilius ir padidinti pervežamų prekių kiekį, taip sumažinant įmonės išlaidas.

Dėžių pakavimo trimatėje erdvėje problema yra žinoma jau seniai, tačiau tik neseniai kompiuteriai pasiekė tokią skaičiavimo galią, kuri leidžia kurti algoritmus, kurių duodami rezultatai padėtų geriau pakrauti automobilį nei žmogus.

Dėžių pakavimo trimatėje erdvėje (*3D bin packing*) uždavinys formuluojamas taip: turime aibę dėžių ir jas reikia sukrauti į mažiausią skaičių konteinerių. Sprendžiant šį uždavinį, atsiranda įvairių apribojimų: visų pirma, kraunant kai kurių dėžių negalima vartyti; antra, krauti dėžes viena ant kitos galima tik tada, jei nebus sugniuždyta apatinė dėžė; trečia, kraunant dėžes į krovininį automobilį, ašių apkrova negali viršyti nustatytos maksimalios leistinos ašių apkrovos. Be to, algoritmas turi maksimaliai pakrauti konteinerį, o skaičiavimai turi trukti apibrėžtą ir suskaičiuojamą laiko tarpą.

Šiame darbe kuriamas dėžių pakavimo trimatėje erdvėje algoritmas ir prekių pakavimo trimatėje erdvėje sistema „TURIS“, kuri naudos šį algoritmą optimizuoti prekių krovimą į krovininius automobilius. Naudodamosi šia informacine sistema vežėjų kompanijos galės optimaliai išnaudoti savo resursus, o tai, kaip žinia, leis sutaupyti finansinių išteklių ir mažinti pervežimo kaštus. Sukurtas algoritmas turi dirbti greitai ir efektyviai. Maksimalus dėžių skaičius gali būti 20 000, o esant tokiam dėžių skaičiui skaičiavimai turi trukti ne ilgiau kaip 5 minutes.

Kad vartotojui būtų patogiau krauti dėžes į krovininį automobilį, gauti rezultatai bus atvaizduojami grafiškai trimatėje erdvėje. Vaizdą bus galima pasukti, atitolinti ir priartinti. Vartotojas ekrane galės matyti detalų planą, kaip dėžės turi būti sudedamos į krovininį automobilį.

2 Dėžių pakavimo trimatėje erdvėje sistemos analizė

Pasaulyje yra sukurta gausybė įvairios paskirties programinės įrangos, todėl naivu tikėtis, kad nėra egzistuojančių sistemų panašiems uždaviniams spręsti. Atlikus paiešką internete rastos šios sistemos (kurios turi nemokamą bandomąją versiją):

- „CargoWiz“ (detaliau nagrinėjama 2.1.1 skyriuje),
- „3D Load Packer” (detaliau nagrinėjama 2.1.2 skyriuje),
- „LoadPlanner” (detaliau nagrinėjama 2.1.3 skyriuje),
- „Cube-IQ” (detaliau nagrinėjama 2.1.4 skyriuje).

Bus atliktas šių sistemų įvertinimas ir analizė.

Atlikus egzistuojančių problemų tyrimą bus identifikuojamos problemos, kurias būtina įvertinti kuriant dėžių pakavimo trimatėje erdvėje optimizavimo sistemą ir algoritmą.

Pirmoje šio darbo dalyje (po egzistuojančių sistemų analizės), pateikiama problemų, kurios iškyla kraunant prekes į krovininį automobilį, analizė:

- dėžių pakavimo optimizavimo trimatėje erdvėje algoritmas (2.2 skyrius),
- dėžių vartymas ir dėžių vartymo ribojimas erdvėje (2.3 skyrius),
- dėžių gniuždymo jėgos įvertinimas (2.4 skyrius),
- krovininio automobilio maksimalios ašių apkrovos skaičiavimas (2.5 skyrius),
- dėžių krovimo į krovininį automobilį grafinis atvaizdavimas (2.6 skyrius).

Iškilusios problemos aprašomos tokiais aspektais:

- Suformuojama projektavimo metu iškilusi projektavimo ar programavimo inžinerijos problema.
- Pateikiama galimų sprendimų apžvalga.
- Pagrindžiamas ir įvertinamas priimtas bei realizuotas sprendimas.

2.1 Egzistuojančios dėžių pakavimo trimatėje erdvėje sistemos

Pasinaudojus internetine paieška pavyko rasti šias egzistuojančias sistemas:

- „CargoWiz“,
- „3D Load Packer”,
- „LoadPlanner,
- „Cube-IQ”.

Panagrinėsime šias sistemas detaliau, atkreipdami dėmesį į tokius programos parametrus:

- vartotojo sąsają,
- funkcionalumą,
- algoritmo parametrus,
- rezultatų pateikimą,
- grafinis rezultatų atvaizdavimą,
- dėžių grupavimą.

2.1.1 „CargoWiz“ – krovininių automobilių ir konteinerių krovimo sistema

Nagrinėjama šios programos bandomoji versija, kuri buvo parsisiųsta iš CargoWiz internetinės svetainės adresu

http://www.softtruck.com/Container_Loading_Software_download.htm [14].

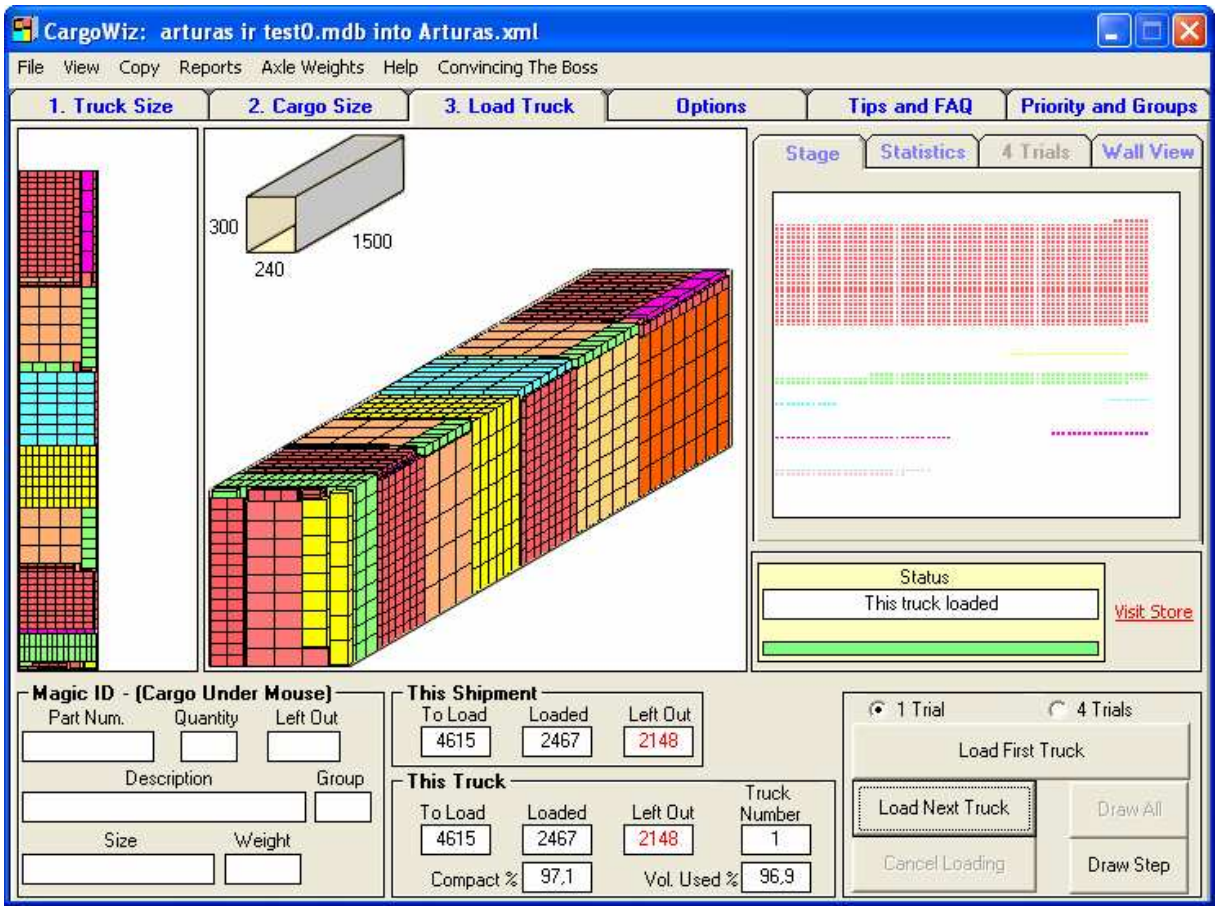
Vartotojo sąsaja: Programos vartotojo sąsaja draugiška ir aiški, lengvai suprantama.

Funkcionalumas: Funkciniu atžvilgiu programa labai panaši į mūsų projektuojamą. Informacijoje apie sistemą nėra pateikta informacijos apie naudojamą algoritmą ir jo efektyvumą. Bandomoji versija bus naudojama ištirti sistemos algoritmo efektyvumą (4 skyrius). Pirmasis žingsnis: krovininio automobilio tipo pasirinkimo. Antrasis žingsnis: pasirenkame kokias prekes krausime. Prekės turi tokius parametrus: plotį, ilgį, aukštį, kiekį kiek daugiausiai prekių galima statyti į aukštį, svorį, leidžiamas orientacijas, pasirinkimą „tik dugnu žemyn“. Pasirinkus krovininį automobilį ir prekes galima pereiti prie trečiojo žingsnio: skaičiavimų (skaičiavimų langas pateikiamas 1 pav.). Programa leidžia pasirinkti kiek kartų bus ieškoma optimalaus sprendinio. Galima ieškoti 1 kartą (1 Trial) arba 4 kartus (4 Trial).

Atlikus skaičiavimus galima patikrinti ašių apkrovą, ar neviršija maksimalios leistinos, galima peržiūrėti pažingsniui kaip sukrautos prekės, galima atsispausdinti ataskaitas, kuriose parašyta kaip krauti prekes. Informacija pateikiama suprantamai, bet ne visada patogiai.

Algoritmo parametrai: Pradedant darbą su programa reikia nustatyti programos parametrus. Parametrai susiję su vykdomu optimizavimo algoritmu: ar naudoti svorio

ribojimą kraunant dėžes ir skaičiuoti 1 ar 4 iteracijomis. Skaičiuojant 4 iteracijomis skaičiavimo laikas pailgėja daugiau nei 4 kartus, tačiau rezultatai gaunami tikslesni.



1 pav. „CargoWiz“ programos skaičiavimų langas

Grafiniam rezultatų atvaizdavimas: trūksta funkcionalumo (negalima pasukti, priartinti ar padidinti). Nėra atskiro rezultatų grafinio atvaizdavimo lango!

Vienas iš didesnių sistemos trūkumų yra tai, kad dėžės mažai grupuojamos, todėl trūksta nuoseklios krovimo logikos. Krovinio krovėjui tai apsunkina darbą, nes reikia ne tik krauti dėžes į krovininį automobilį, bet dar ir žaisti dėlionę.

Bendras sistemos įvertinimas:

Vartotojo sąsajos patogumas: 9.

Naudingos informacijos pateikimas: 8.

Grafinis rezultatų atvaizdavimas: 7.

Dėžių grupavimas: 8.

Bendras įvertinimas (be algoritmo efektyvumo): **8,00**.

Algoritmo efektyvumas: įvertinimas pateikiamas prie sistemų tyrimo 4 skyriuje.

2.1.2 „3D Load Packer“ – dėžių pakavimo sistema

Sistemos analizei atlikti bandomoji versija parsisiųsta iš internetinės svetainės: http://www.astrokettle.com/downld.html .

3D Load Packer (3DLP) yra unikali dėžių pakavimo sistema [3]. Pakrovimams optimizuoti naudoja savo sukurtą optimizavimo algoritmą. Kraunami objektai gali būti tik stačiakampės dėžės.

Programos vartotojo sąsaja atrodo senamadiškai (2 pav.). Lyginant su prieš tai aprašytu CargoWiz produktu, šio produkto vartotojo sąsaja mažiau draugiška, dizainas primityvesnis.

Funkcionalumas apsiriboja pagrindinėmis funkcijomis. Programą sudaro 5 langai ():

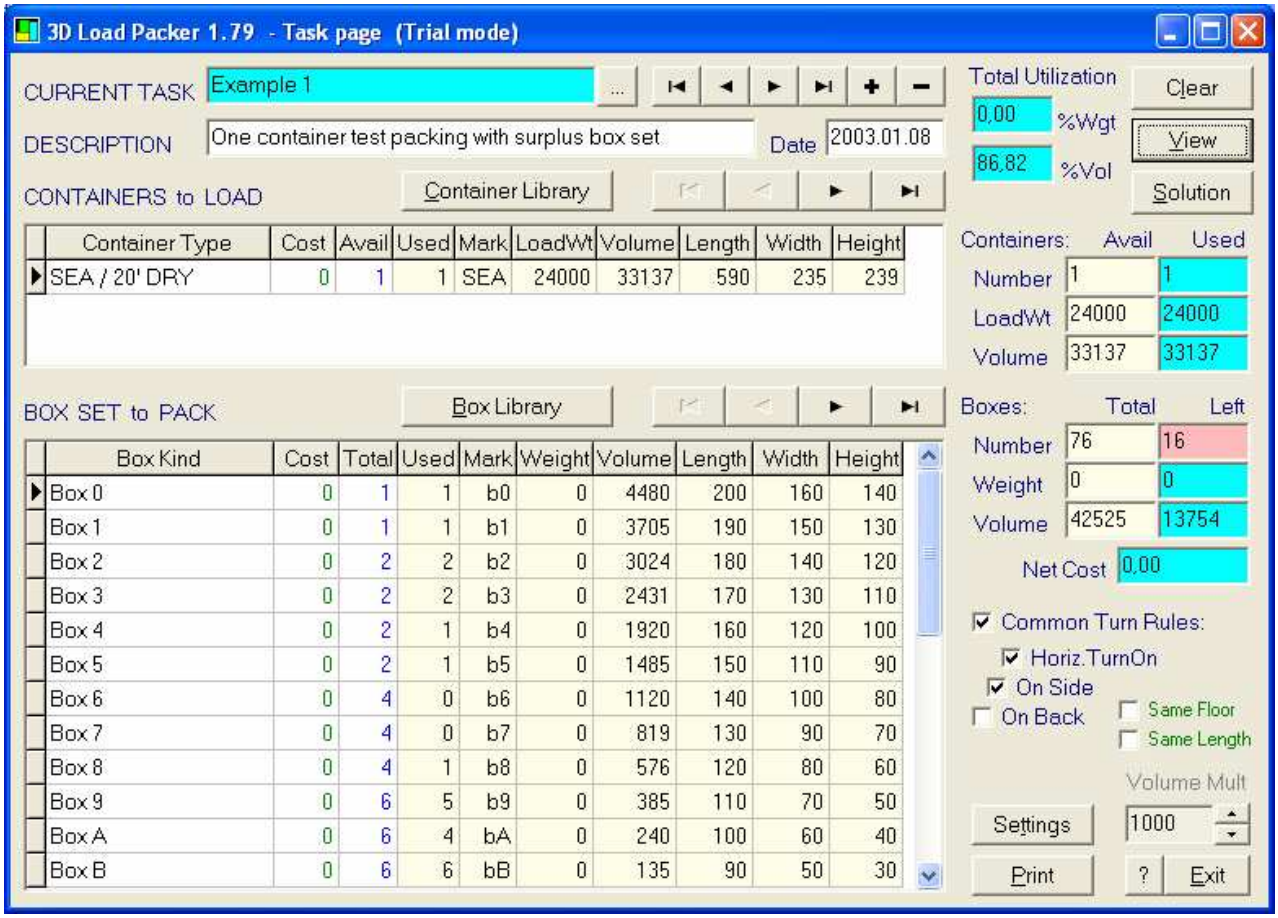
Pagrindinis langas, kuriame formuojamos užduotys ir skaičiuojami rezultatai.

Konteinerių langas, kuriame įvedama ir redaguojama informacija apie konteinerius.

Dėžių langas, kuriame įvedama ir redaguojama informacija apie dėžes.

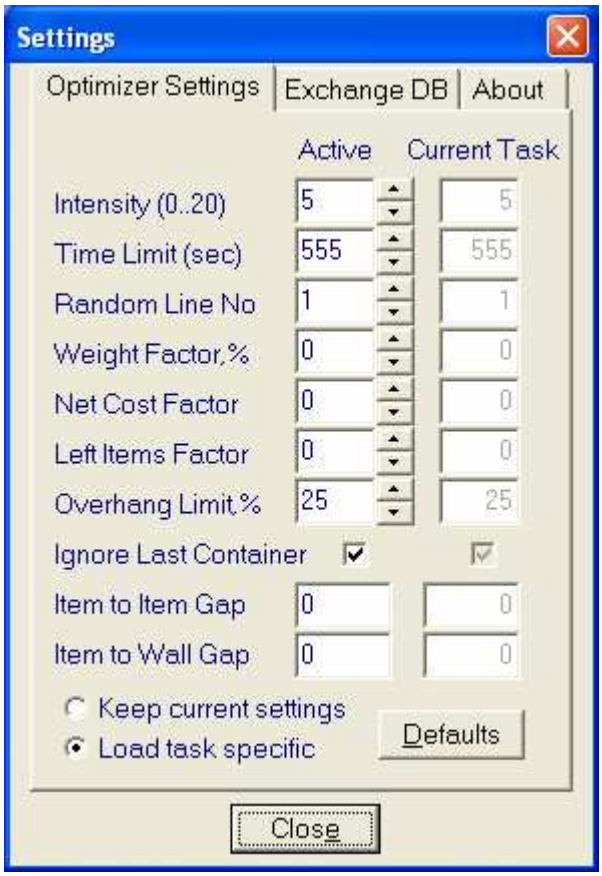
Parametrų langas, kuriame nustatomi pagrindiniai parametrai. Su algoritmu susijusių parametrų yra vos keli.

Rezultatų atvaizdavimo langas, kuriame trimatėje erdvėje atvaizduojami grafiniai rezultatai.



2 pav. „3D Load Packer“ pagrindinis langas

Algoritmo parametrai: Sistema turi keletą naudingų parametrų, kurie leidžia suderinti algoritmo veikimą pagal vartotojo poreikius (3 pav.).



3 pav. „3D Load Packer“ sistemos algoritmo parametrai

Grafinis rezultatų atvaizdavimas. Grafiniai rezultatai atvaizduojami trimatėje erdvėje, grafinis variklis leidžia sukioti vaizdą į visas puses. Grafinio atvaizdavimo trūkumas yra tai, kad dėžės nėra grupuojamos, o grafinis variklis leidžia dėti po vieną dėžę, kas nėra patogu vartotojui.

Bendras sistemos įvertinimas:

Vartotojo sąsajos patogumas: 6.

Naudingos informacijos pateikimas: 8.

Grafinis rezultatų atvaizdavimas: 8.

Dėžių grupavimas: 6.

Bendras įvertinimas (be algoritmo efektyvumo): **7,00**.

Algoritmo efektyvumas: įvertinimas pateikiamas prie sistemų tyrimo 4 skyriuje.

2.1.3 „LoadPlanner“ – prekių pakavimo ir optimizavimo sistema

Analizuojama internetinė sistemos versija, ją galima rasti šiuo adresu: http://loadplanner.com/.

LoadPlanner – tai prekių pakavimo ir optimizavimo sprendimas pažangiam logistikos valdymui. LoadPlanner širdis yra modernus prekių pakavimo trimatėje erdvėje algoritmas, kuris buvo kuriamas ir tobulinamas daugybe metų padedant pirmaujančioms logistikos kompanijoms. Šį produktą išskirtiniu daro modernus taisyklių valdymo rinkinys [4]:

- Veiklos objektų (prekių, užsakymų, konteinerių) klasifikavimas į patogią kategorijų sistemą.
- Aukšto lygio veiklos taisyklių ir apribojimų formulavimas ir jų pritaikymas pasirinktoms kategorijos.
- Veiklos taisyklių ir apribojimų naudojimas prekių pakavimo optimizavimo procese.
- Gebėjimas spręsti kelių lygių prekių pakavimo uždaviniams (pakavimas – krovimas ant palečių – krovimas į konteinerį).
- Interaktyvus ir patogus rezultatų atvaizdavimas.

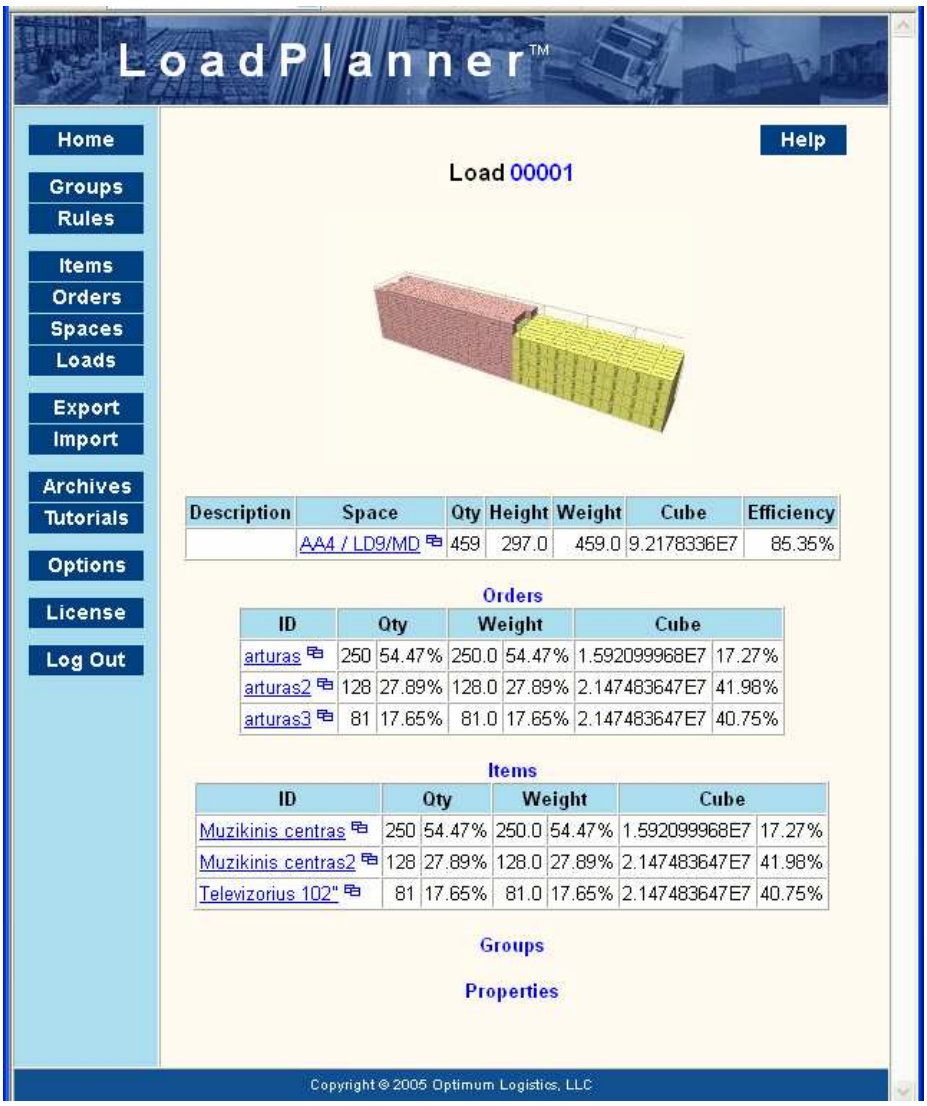
Šia sistema galima naudotis internetinėje versijoje, kliento-serverio architektūros versijoje arba paprasčiausioje kliento versijoje.

Vartotojas susipažinti su LoadPlanner programa gali nemokamoje internetinėje versijoje, tad tęsdami panašių produktų įvertinimą atliksime šios sistemos internetinės versijos analizę.

Vartotojo sąsaja patogi, tačiau ne visada lengvai suprantama.

Funkcionalumas: Pabandykime sukurti testinę užduoti ir susimuliuoti rezultatus. Pirmiausiai apsirašome dėžes, kurias krausime į automobilį. Sistemoje dėžėms aprašyti yra 4 parametrai: ilgis, plotis, aukštis ir svoris. Prekes priskyrus prie užsakymų galima nurodyti leistinas dėžių orientacijas. Sukūrus prekes kitas žingsnis yra sukurti erdvę, į kurią krausime prekes. Be pagrindinių parametrų konteineriui galima nurodyti orientaciją, parametrus, grupes ir pakrovimus. Prieš paleidžiant skaičiuoti algoritmą dar reikėtų nustatyti taisykles, kurių yra net 32. Veiklos taisyklių detaliau nebus nagrinėjamos, tai bus atlikta tyrimo dalyje.

Paleidus skaičiuoti algoritmą po kelių sekundžių gaunami rezultatai (4 pav.).



4 pav. „LoadPlanner“ rezultatų langas

Algoritmo parametrai: Algoritmas turi net 32 veiklos taisykles. Šitokia veiklos taisyklių gausa yra patogumas vartotojui, tačiau neaišku kaip jos paveikia algoritmo vykdymo laiką.

Grafinis rezultatų atvaizdavimas: Paspaudus ant paveiksliuko atsidaro rezultatų grafinio atvaizdavimo langas, kuriame patogiai atvaizduojami rezultatai.

Bendras sistemos įvertinimas:

Vartotojo sąsajos patogumas: 7.

Naudingos informacijos pateikimas: 8.

Grafinis rezultatų atvaizdavimas: 8.

Dėžių grupavimas: 7.

Bendras įvertinimas (be algoritmo efektyvumo):**7,5**.

Algoritmo efektyvumas: įvertinimas pateikiamas prie sistemų tyrimo 4 skyriuje.

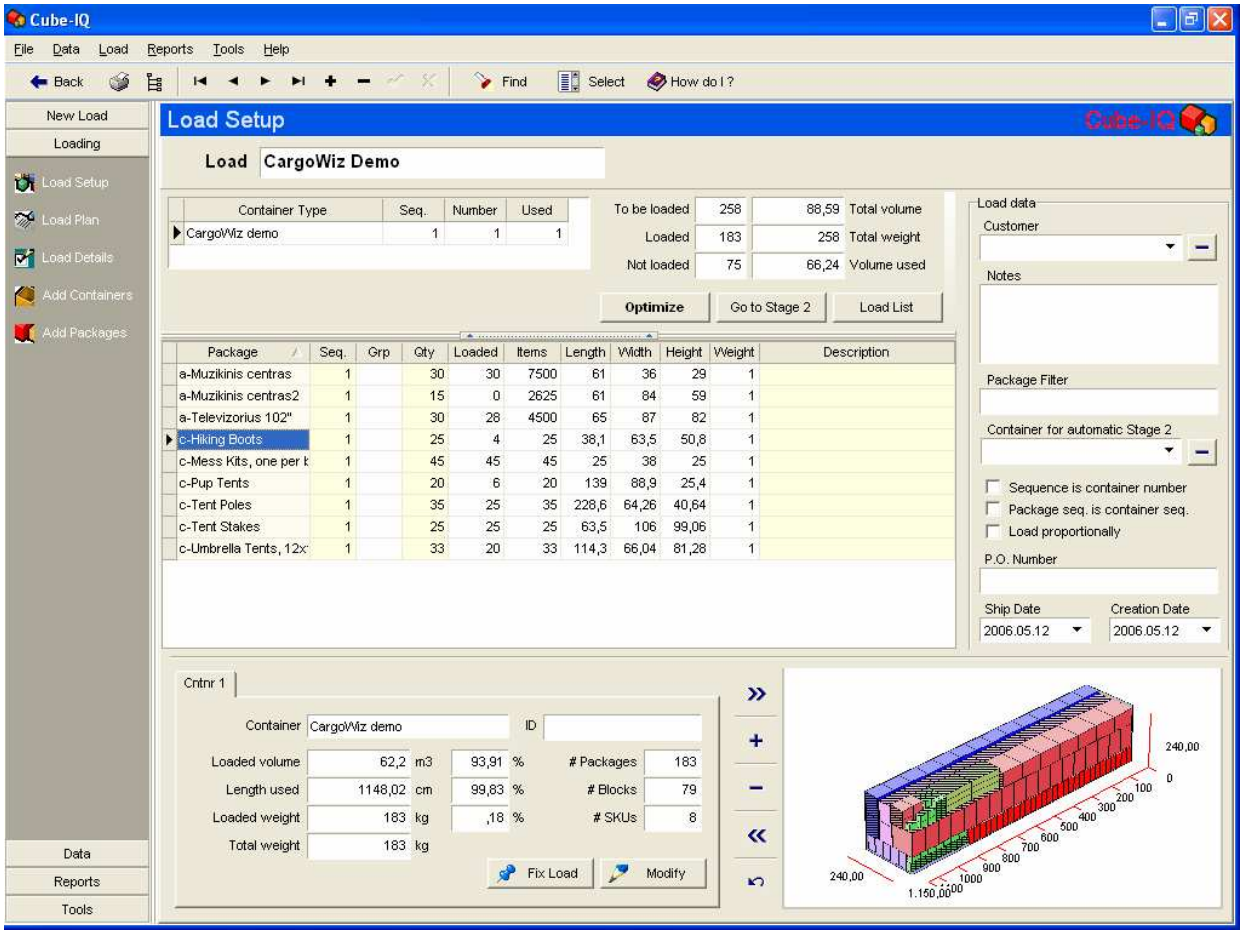
2.1.4 „Cube-IQ“ – dėžių krovimo optimizavimo sistema

Sistemos analizei atlikti bandomoji versija parsisiųsta iš internetinės svetainės: http://www.magiclogic.com/downloads.html .

Ši sistema rinkoje pasirodė visai neseniai, tik šiais metais. Iš visų analizuotų prekių pakavimo sistemų ši sistema turi daugiausiai pranašumų [6]. Atlikus standartinę sistemos apžvalgos analizę jos surinkti bendro įvertinimo balai aukščiausi.

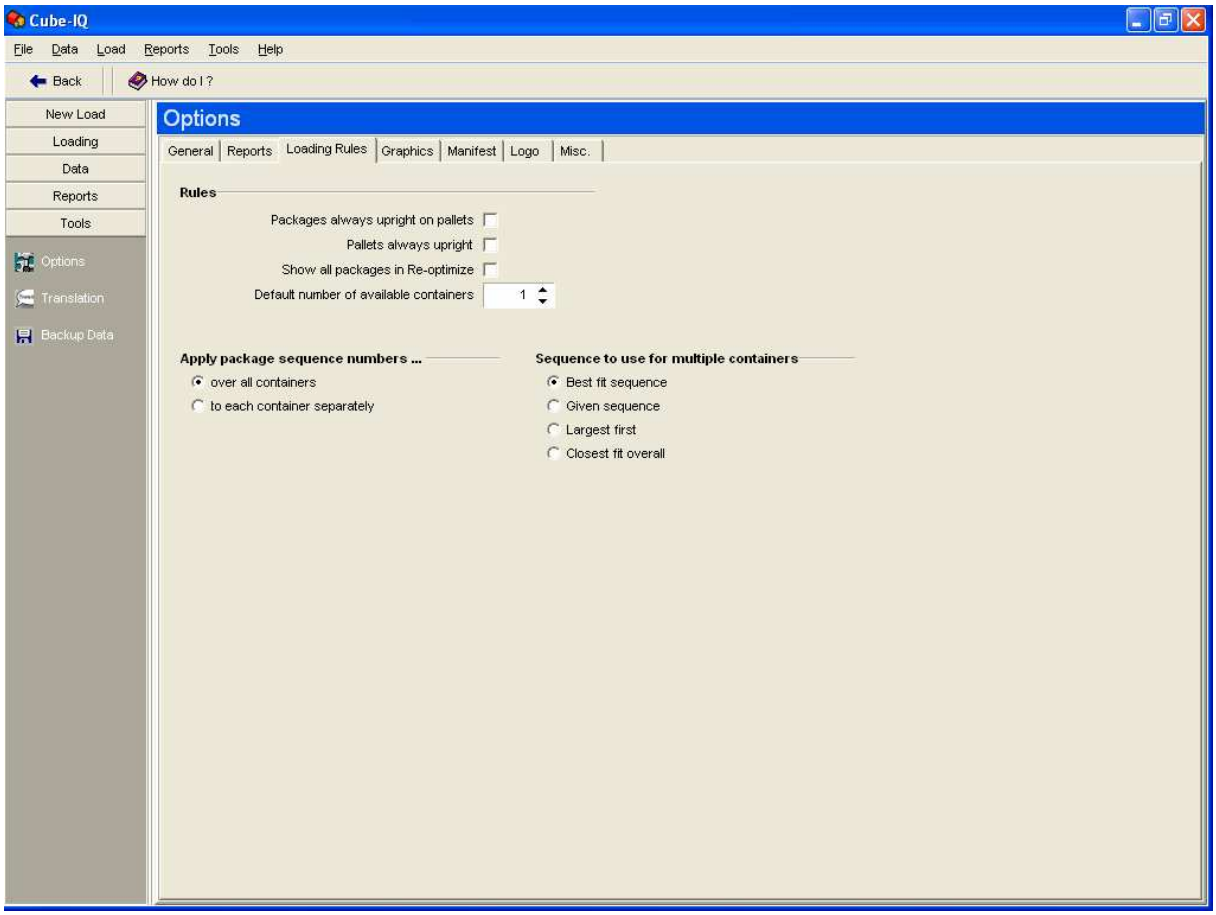
Vartotojo sąsaja aiški ir patogi, lengvai intuityviai suprantama.

Funkcionalumas: Sistema turi savo duomenų bazę, kuri saugoma Access duomenų faile, duomenys lengvai importuojami/eksportuojami. Pasirinkus, kurias dėžes krauti į kokį konteinerį, ir paspaudus optimizavimo vykdymo mygtuką, įvykdomi skaičiavimai ir pateikiama pagrindinė informacija apie prekių krovimo optimizavimą (5 pav.). Norint detaliau peržiūrėti gautus rezultatus reikia pasirinkti tam skirtas ataskaitas, kurių yra nemažai ir kurios yra labai patogios.



5 pav. „Cube-IQ“ krovimo langas

Algoritmo parametrai: Sistema turi tik keletą naudingų parametrų, kurie leidžia suderinti algoritmo veikimą pagal vartotojo poreikius (6 pav.).



6 pav. Cube-IQ parametrų langas

Grafiniai rezultatai pateikiami įvairiais pjūviais vaizdą galima sukioti į įvairias puses, galima simuliuoti dėžių krovimą po vieną. Grafinis variklis gerai apgalvotas ir puikiai realizuotas, didesnių trukumų nepastebėta.

Bendras sistemos įvertinimas:

Vartotojo sąsajos patogumas: 10.

Naudingos informacijos pateikimas: 9.

Grafinis rezultatų atvaizdavimas: 10.

Dėžių grupavimas: 9.

Bendras įvertinimas (be algoritmo efektyvumo): **9,5**.

Algoritmo efektyvumas: įvertinimas pateikiamas prie sistemų tyrimo 4 skyriuje.

2.1.5 Bendras sistemų įvertinimas

1 lentelėje pateikiamas bendras visų sistemų įvertinimas.

1 lentelė. Bendras sistemų įvertinimas

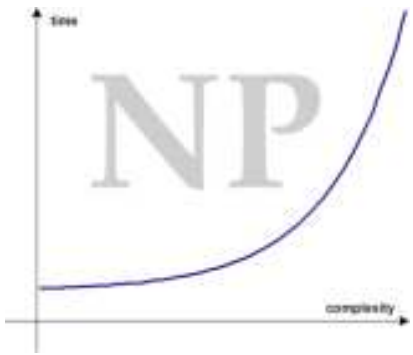
Sistema / Kriterijus	„CargoWiz“	„3D Load Packer“	„LoadPlanner“	„Cube-IQ“
Vartotojo sąsajos patogumas	9	6	7	10
Naudingos informacijos pateikimas	8	8	8	9
Grafinis rezultatų atvaizdavimas	7	8	8	10
Dėžių grupavimas	8	6	7	9
Bendras įvertinimas (be algoritmo efektyvumo)	**8**	**7**	**7,5**	**9,5**

Matome, kad stipriai pralenkusi konkurentus balų skaičiais pirmą vietą užėmė „Cube-IQ“ sistema. Ji neabejotinai yra rinkos lyderė. Ši sistema pasižymi puikia sąsaja su vartotoju, puikiu grafiniu varikliu, tačiau truputi trūksta algoritmo parametrų, su kuriai būtų galima priderinti algoritmą prie vartotojo poreikių, bei dėžės galėtų būti labiau grupuojamos, kad vartotojui būtų lengviau jas krauti.

Antrojoje vietoje liko „CargoWiz“. Ši sistema visais kriterijais atsiliko nuo „Cube-IQ“, o silpniausia šios sistemos vieta yra grafinis rezultatų atvaizdavimas.

2.2 Algoritmas dėžių pakavimo trimatėje erdvėje uždaviniui spręsti

Dėžių pakavimo trimatėje erdvėje uždavinys, matematikoje dar žinomas kaip Knapsack problema [11], yra vienas sudėtingiausių pilno perrinkimo uždavinių. 1972 metais R. Karp įrodė, kad tai yra NP-pilnumo problema (7 pav.), viena pačių sudėtingiausių pilno perrinkimo problemų. Trimatėje erdvėje šis uždavinys tampa dar labiau komplikuotas, tačiau teoriškai, naudojant neapibrėžtą skaičiavimo galią turinčius kompiuterius, jis gali būti išspręstas per polinominį laiko tarpą.



7 pav. NP grafikas

Deja, visi žinomi algoritmai, kurie sprendžia NP-sudėtingumo uždavinius, panašius į Knapsack uždavinius, turi eksponentinio sudėtingumo laiko priklausomybę, kuri yra ekvivalentiška pilno perrinkimo laikui. Tai reiškia, kad naudojant tokį algoritmą, netgi mažo krovinio (apie 50 dėžių) pakrovimo optimizavimas, atliekant skaičiavimus su superkompiuteriu, užtruktų ne vienerius metus.

2.2.1 Algoritmų tipai

Kuriamas naujas algoritmas bus labai naudingas verslo procesuose, o verslas reikalauja labai greitų sprendimų, todėl reikalingas algoritmas, kuris sugebėtų išspręsti šį uždavinį per labai trumpą ir apibrėžtą laiko tarpą.

Egzistuoja kelių tipų algoritmai, tinkami šiam uždaviniui spręsti:

Euristiniai algoritmai

Uždaviniui spręsti skirtas algoritmas turi per įrodomai trumpą skaičiavimo laiką rasti įrodomai gerą optimalų sprendinį. Euristinis algoritmas pasižymi viena iš šių dviejų savybių arba abejomis kartu. Pavyzdžiui, jis dažniausiai randa optimalų sprendinį, tačiau negalima įrodyti, kad jis taip veiks su visais duomenų rinkiniais; arba jis visada skaičiavimus atlieka labai greitai, bet nėra įrodoma, kad taip bus esant bet kokiems pradinių duomenų rinkiniams. Dažniausiai egzistuoja tokie pradiniai duomenys, su kuriais euristinis algoritmas grąžins labai blogus rezultatus arba skaičiavimai truks labai ilgai, tačiau šie atvejai dėl savo specifiškumo gali niekada neįvykti praktikoje. Panaudojus euristinį algoritmą mūsų uždavinio sprendimui galima tikėtis per keletą minučių pasiekti vidutiniškai 80-90% erdvės pakrovimo tankį su keliasdešimt tūkstančių dėžių. [12]

Genetiniai algoritmai

Genetiniai algoritmai kitaip dar yra vadinami apytikrio skaičiavimo algoritmai. Genetiniai algoritmai yra tam tikra evoliucinių algoritmų klasė, kurie naudojasi evoliucinės biologijos principais, tokiais kaip paveldėjimas, mutacija, natūrali selekcija ir rekombinacija. Pasirenkami keli pradiniai sprendimai, kurie skaičiavimo eigoje evoliucionuoja, todėl sprendimo pabaigoje lieka tik geriausi sprendimo variantai. [2]

Mišrūs algoritmai

Tai algoritmai pasižymintys tiek euristinių, tiek genetinių algoritmų savybėmis. [1]

2.2.2 Parametrai, kuriuos reikia įvertinti kuriant algoritmą

Dėžių pakavimo trimatėje erdvėje algoritmas, ieškodamas optimalaus pakrovimo varianto, turėtų įvertinti tokius apribojimus:

- **Dėžių vartymas erdvėje** (detaliau nagrinėjama 2.3 skyriuje)

Asmuo, kraudamas dėžes į krovininį automobilį, analizuoja, kaip yra naudingiau dėti dėžę, kokia dėžes padėtis būtų geriausia automobilyje, ar tikslinga ją paversti prieš kraunant. Taigi norint dėžes į krovininį automobilį sudėlioti optimaliai, reikia, kad algoritmas jas vartytų ir ieškotų, kokia orientacija jas įdėti, kad būtų gauti optimalūs rezultatai.

- **Dėžių vartymo ribojimas erdvėje** (detaliau nagrinėjama 2.3 skyriuje)

Reikia atsižvelgti, jog kai kurių prekių negalima apversti, arba negalima paversti ant kurio nors šono (pvz., šaldytuvo, orkaitės), todėl algoritmas turi įvertinti ir dėžių vartymo ribojimus erdvėje, kad prekės nebūtų pažeistos ar visiškai sugadintos.

- **Dėžių gniuždymo jėgos įvertinimas** (detaliau nagrinėjama 2.4 skyriuje)

Dar vienas ribojimas, į kurį reikia atsižvelgti projektuojant algoritmą, yra dėžių gniuždymo jėga. Pavyzdžiui, jeigu ant mažos, netvirtos dėžės (pvz., lygintuvo) uždedama daug sunkių dėžių arba viena didelė labai sunki dėžė (pvz., šaldytuvas), tai tikėtina, kad prekės pakuotė bus pažeista ir prekė bus sugadinta.

- **Krovininio automobilio ašių apkrovos skaičiavimas** (detaliau nagrinėjama 2.5 skyriuje)

Labai svarbus krovininių automobilių parametras yra maksimali ašių apkrova. Kuriant algoritmą, reikia atsižvelgti, kad ašies apkrova neviršytų leidžiamos maksimalios krovininio automobilio ašies apkrovos.

2.2.3 Egzistuojantys algoritmai

Didžioji dalis informacijos apie tokių uždavinių sprendimą yra mokama arba publikuojama mokamuose žurnaluose. Pavyko rasti šiuos nemokamus algoritmus, skirtus trimačiui dėžių pakavimo uždaviniui spręsti:

- „An exact algorithm for the Three-dimensional Bin-packing Problem" [7].
- „TSpack: A Unified Tabu Search Code for Multi-Dimensional Bin Packing Problems" [15].
- „Guided local search for the three-dimensional bin packing problem" [9].

Šie visi trys algoritmai yra skirti teoriniam dėžių pakavimo uždavinio variantui spręsti. Juose nėra orientacijos ribojimo erdvėje, gniuždymo jėgos įvertinimo bei ašies apkrovos skaičiavimo. Dar vienas trūkumas yra tai, jog šie algoritmai sprendžia tik klasikinį uždavinio variantą, kuris skiriasi nuo šiame darbe nagrinėjamo uždavinio varianto. Klasikinio uždavinio tikslas yra turimų dėžių aibę sukrauti į kuo mažesnį konteinerių skaičių. Šiame darbe nagrinėjamas uždavinio tikslas - kuo daugiau dėžių sukrauti į vieną turimą konteinerį. Apribojus klasikinį uždavinio variantą sąlyga, jog turimas tik vienas konteineris, gaunamas šiame darbe nagrinėjamo uždavinio variantas. Taigi formuluojamas uždavinys yra klasikinio uždavinio paprastesnis variantas. Sprendžiant šį variantą ir atsiveria perspektyvos sukurti algoritmą, kuris duos geresnių rezultatų, nes bus paprastesnis, taip pat turės anksčiau minėtus vartymo erdvėje, gniuždymo ir kitus apribojimus.

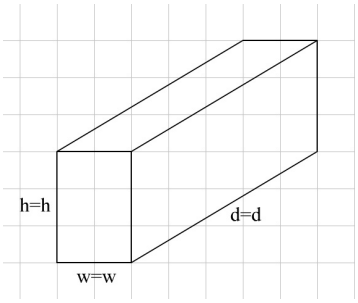
2.2.4 Algoritmo pasirinkimas

Išnagrinėjus euristinių ir genetinių algoritmų privalumus bei trūkumus ir išanalizavus rastus klasikinio dėžių pakavimo trimatėje erdvėje uždavinio sprendimo algoritmų principus (2.2.3 skyrius „Egzistuojantys sprendimai"), nuspręsta, kad bus kurtas greitas ir tikslus euristinis optimizavimo algoritmas. Šis algoritmas turės įvertinti realiame pasaulyje egzistuojančius apribojimus (rasti algoritmai pavyzdiniai į šiuos apribojimus neatsižvelgia), kurie detaliau analizuojami **2.3, 2.4, 2.5 skyriuose**.

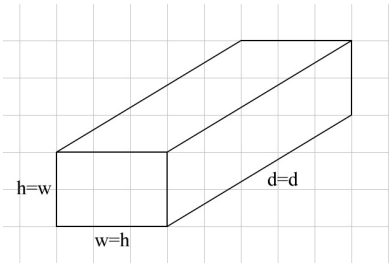
2.3 Dėžių vartymas ir dėžių vartymo ribojimas erdvėje

Dauguma dėžių pakavimo trimatėje erdvėje algoritmų (kuriuos pavyko rasti) ieškodami optimalaus dėžių išdėstymo erdvėje, dėžių erdvėje nevarto ir jas deda tokia orientacija, kokia pateikiami dėžės matmenys, dėl to gaunami blogesni rezultatai.

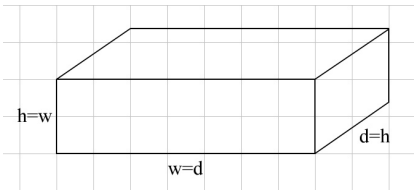
Geometriniu požiūriu, dėžė – tai trimatis geometrinis kūnas, turintis 6 stačiakampes sienas. Paguldžius dėžę ant kiekvienos sienos, ją galima orientuoti 4 kryptimis, todėl egzistuoja 24 skirtingos orientacijos. Tačiau, kuriant algoritmą, yra svarbios tik tos orientacijos, kurios turi skirtumą geometriniu požiūriu. Todėl iš tikrųjų yra tik 6 geometriškai skirtingos orientacijos erdvėje, į kurias reikia atsižvelgti:



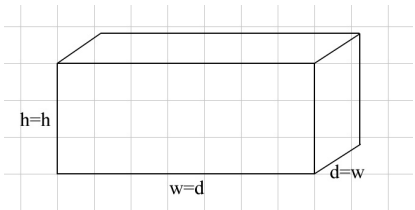
8 pav. Orientacija 1.



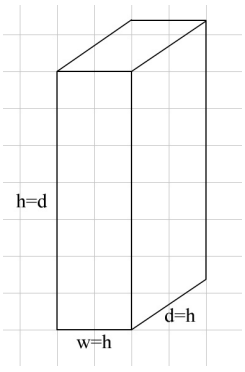
9 pav. Orientacija 2.



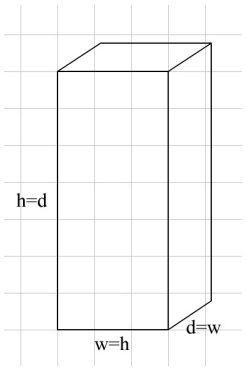
10 pav. Orientacija 3.



11 pav. Orientacija 4.



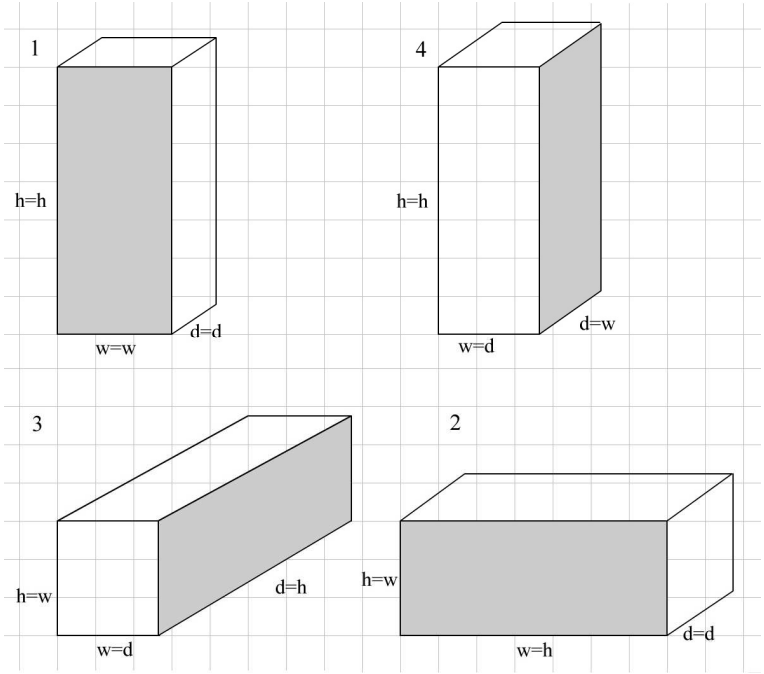
12 pav. Orientacija 5.



13 pav. Orientacija 6.

čia: h (*height*) – aukštis;
w (*width*) – plotis;
d (*deep*) – gylis arba ilgis.

Kuriant algoritmą reikia įvesti ne tik dėžių vartymo erdvėje funkciją, bet ir galimybę riboti dėžių vartymą erdvėje, kadangi ne visas prekes dedant į krovininį automobilį galima vartyti bet kuria kryptimi. Pavyzdžiui, šaldytuvą galima statyti tik ant 2 sienų, ant dugno ir ant vieno šono (iš keturių). Todėl statant šaldytuvą, jį galima pasukti tik 4 skirtingomis kryptimis (2 orientacijos ant dugno ir 2 ant šono, žr. 14 pav.):



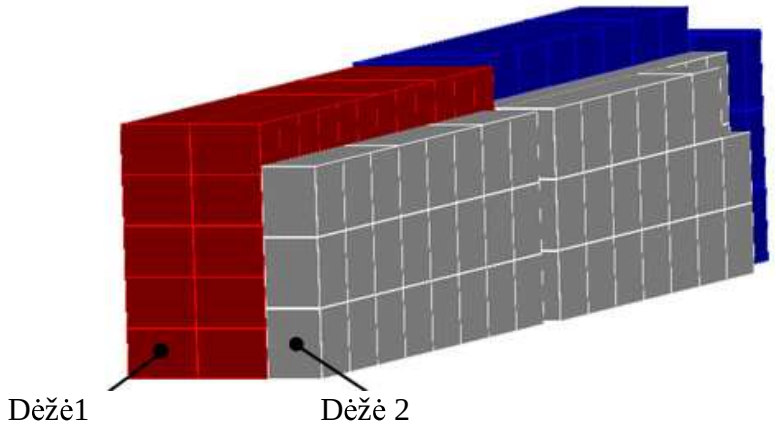
14 pav. Šaldytuvo statymo kryptys.

Kad šį apribojimą būtų lengviau įvertinti ir realizuoti, tai galima pažiūrėti į jį iš kitos pusės: ne kaip į tam tikrų orientacijų draudimą, o kaip į tam tikrų orientacijų leidimą. Tokiu atveju ieškoma orientacijų, kuriomis būtų galima dėti dėžę, o ne tų, kuriomis negalima. Jei dėžė neturi apribojimų, tai ją galima vartyti visomis kryptimis, todėl galimos orientacijos yra: 1, 2, 3, 4, 5, 6. Jeigu, pvz. turimas šaldytuvas, o jį galima guldyti tik ant dugno ir ant vieno šono, vadinasi, jį galima sukinėti ant dugno, t.y. galimos orientacijos yra 1, 4, ir ant šono – orientacijos 2, 3 (14 pav.).

2.4 Dėžių gniuždymo jėgos įvertinimas

Dar viena problema, su kuria tenka susidurti, pritaikant algoritmą realiems uždaviniams spręsti, yra gniuždymo jėga. Prekių pakuotės dažniausiai būna iš kartono, todėl uždėjus ant jų didelį svorį pakuotė arba pati prekė gali būti pažeista. Žmogus logiškai mąstydamas sugeba įvertinti, kad dėti dideles dėžes ant mažų yra pavojinga, kadangi didelės dėžės būna sunkios ir gali suspausti mažas dėžutes. Kompiuteris neturi tokio loginio mąstymo, dėl to jam reikia aprašyti taisykles, pagal kurias jis galėtų „suprasti“, kada negalima dėti dėžės ant kitos, kad nebūtų suspausta apatinė apatinės.

Taisyklė yra pakankamai paprasta. Kiekviena dėžė turi parametrą, kuris reiškia jėgą į cm^2, kurią gali atlaikyti pakuotė. Algoritmas, kraudamas dėžes, skaičiuoja, kokia jėga yra veikiama apatinė dėžė, ir, jei ši jėga pasidaro didesnė nei nurodyta dėžės maksimalios apkrovos parametre, tada viršutinės dėžės krauti nebegalima. Kadangi krovimas vyksta sluoksniais, todėl reikia paskaičiuoti sluoksnio apatinės dėžės apkrovą (15 pav. Dėžė 1 ir Dėžė 2). Jeigu apkrova neviršija leistinos, dėžių sluoksnį galima krauti, jeigu viršija, tada reikia ieškoti geresnio sprendimo.



15 pav. Dėžių gniuždymas

2.5 Krovininio automobilio ašių apkrovos skaičiavimas

Siekiant užtikrinti krovinius vežančio automobilio stabilumą ir kad nebūtų viršyta leistina maksimali ašių apkrova, prekės turi būti tinkamai paskirstytos automobilio priekaboje. Neteisingai paskirsčius krovinį gali būti viršyta leistina maksimali priekinės arba galinės ašies apkrova, net jeigu ir bendra į automobilį pakrauta masė neviršija į automobilį leistinos pakrauti maksimalios masės.

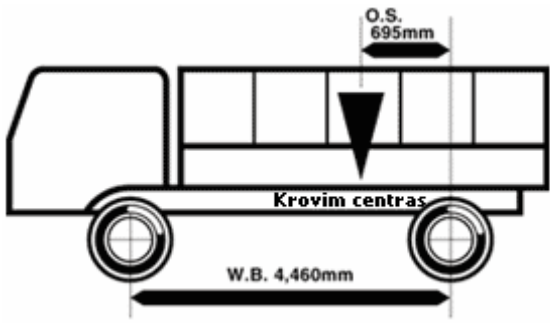
Kai krovinio masės centas sutampa su galine ašimi, tai krovinio paskirstymas yra lygus nuliui (16 pav.), todėl priekinės ašies apkrova yra labai maža, o galinės ašies apkrova labai didelė. Vairuoti tokį krovininį automobilį yra labai sudėtinga ir pavojinga. Kadangi visas kovinio svoris yra sukoncentruotas ant galinės ašies, tai maksimalus svoris, kurį galime pakrauti į krovininį automobilį yra lygus galinės ašies maksimaliai leistinai apkrovai. Rezultatas: krovininis automobilis neoptimaliai pakrautas, o jo valdymas neįmanomas.

16 pav. Automobilio krovinio masės centras sutampa su galine ašimi

Tokie rezultatai netenkintų, nei vienos pervežimo įmonės, todėl vienas iš reikalavimų yra skaičiuoti ašių apkrovas ir stengtis, kad algoritmas optimaliai paskirstytu krovinį maksimaliai išnaudodamas automobilio tūrį, ir neviršytų maksimalios ašių apkrovos.

Ašių apkrovai skaičiuoti bus naudojamas svirties metodas. Metodui paaiškinti naudosime Fk615-H serijos ratų bazės (17 pav.) krovininį automobilį.

Kai kraunamas tos pačios rūšies krovinys, krovimo centras yra to krovinio centras. Atstumas nuo krovimo centro iki galinės ašies vadinamas galine atsvara (žymima O.S.). Pavyzdyje jis lygus 695mm.



17 pav. Automobilio krovinio masės centras nesutampa su galine ašimi

Pagal svirties principą priekinės ašies(Pf) ir galinės ašies (Pr) apkrovą galima apskaičiuoti naudojantis šiomis formulėmis [13]:

$$Pf = P \times \frac{O.S.}{W.B.} \qquad \textbf{Formulė (1)}$$

$$Pr = P \times \frac{W.B. - O.S.}{W.B.} \qquad \textbf{Formulė (2)}$$

P – viso automobilio apkrova.

Pf – priekinės ašies apkrova.

Pr – galinės ašies apkrova.

2.6 Dėžių krovimo į krovininį automobilį grafinis vaizdavimas ekrane

Šiame skyrelyje bus nagrinėjama, kaip grafiškai pateikti rezultatus sandėlininkui, kad jis pagal juos galėtų sėkmingai krauti dėžes į krovininį automobilį.

Sandėlininkas turi matyti vaizdą iš visų pusių, turi matyti siūlomą dėžių krovimo tvarką, turi skirti, kurioje vietoje yra tam tikra dėžė, dėžės turi būti sugrupuotos, kad darbuotojui nereikėtų viso automobilio krauti po vieną dėžę .

Kad sandėlininkui rezultatai būtų pateikti grafiškai patogiu formatu, reikalingas grafinis variklis. Reikia sukurti grafinį variklį, kuris rezultatus atvaizduotų trimatėje erdvėje, sugebėtų pasukti, paversti, priartinti ir patolinti vaizdą. Skirtingų tipų dėžės turėtų būti lengvai atskiriamos ir ant jų turėtų matytis dėžės identifikatorius.

Kurti grafinį variklį galima su šiomis aplinkomis:

- Windows' DirectX ;
- OpenGL;
- Surrender;
- SciTech's MGL and VBE/AF (nemokama);
- Simple Direct Media (atviro kodo);

Labiausiai paplitusios ir didžiausiu funkcionalumu pasižymi šios dvi 3D grafinės sąsajos kūrimo priemonės: Microsoft DirectX ir OpenGL.

2.6.1 Window’s DirectX aplinka

Microsoft DirectX yra kompanijos Microsoft sukurta grupė funkcijų, skirtų atvaizduoti taikomąsias programas, kurios turi daug realaus laiko 3D grafinių vaizdų, filmukų, muzikos ir erdvinio garso efektų. Microsoft DirectX aplinka skirta kompiuteriams su Windows šeimos operacinėmis sistemomis, o platforma reikalinga vykdyti šias funkcijas yra integruota į pačią Windows OS.

DirectX programų kūrėjams leidžia naudotis API, kuris užtikrina prieigą prie 3D grafinių ir garso plokščių siūlomų naujovių. Šios API valdo žemiausio lygio funkcijas, kuriomis naudojantis galima valdyti grafinę atmintį ir vaizdo perdavimą. Žemiausio lygio funkcijos yra sugrupuoto į komponentus, kurie sudaro DirectX aplinką: Microsoft Direct3D®, Microsoft DirectDraw®, Microsoft DirectInput®, Microsoft DirectMusic®, Microsoft DirectPlay®, Microsoft DirectSound®, and Microsoft DirectShow®.

Direct3D versija DX6 pasižymi tokiomis galimybėmis:

- Lankstus kampo viršūnės geometrinis aprašymas.
- Kampų viršūnių saugojimas.
- Daugiasluoksnės tekstūros vaizdinis generavimas.

- Automatinis tekstūros valdymas.
- Perjungiamas gylio buferizavimas (naudojant z-buferius arba w-buferius).
- Taškinės aplinkos žymėjimas BUMPENVMAP atsispindintiems paviršiams ir vandens efektams.

Direct3D versija DirectX 7.0 pasižymi tokiomis galimybėmis:

- Aplinkos žymėjimas su kubinės aplinkos žemėlapiais.
- Geometrijos derinimas.
- Patobulintas automatinis tekstūros valdymas.
- Automatinis tekstūros koordinačių generavimas, tekstūros transformacijos, projektų tekstūros ir pasirenkamas plokštumų karpymas.
- 3D funkcijų biblioteka.

Ši aplinka puikiai tinka programuojant su C++, C, VB.NET programavimo kalbomis. [8].

2.6.2 OpenGL aplinka

OpenGL grafinių funkcijų aplinka yra nepriklausoma nuo techninės įrangos, programinės įrangos, ar nuo gamintojo grafinės API sąsajos. Pagrindinė prieiga prie grafinių funkcijų yra realizuota C, Fortran ir Ada programavimo kalbose.

OpenGL aplinka sukurta naudojantis kliento/serverio principu, kuris leidžia kliento taikomosioms programoms ir grafiniam serveriui, valdančiam kompiuterių grafines plokštes, būti tame pačiame kompiuteryje. Tinklas taikomosioms programoms yra permatomas.

OpenGL neturi tiesioginių funkcijų, leidžiančių valdyti pagrindines geometrines figūras, tokias kaip kubas ar sfera. Šios figūros turi būti kuriamos naudojantis pagrindinėmis funkcijomis.

Keletas naujovių, naudojamų OpenGL aplinkoje:

- geometrinės ir taškinė matricos baziniai elementai,
- RGBA arba spalvotas indeksavimas,
- vaizdavimo registras,
- apšviestumas ir šešėliai,
- paslėptų paviršių pašalinimas (gylio buferis),
- alfa perėjimas iš vienos spalvos į kitą (permatomumas),
- tekstūros žymėjimas,
- atmosferiniai efektai (rūkas, dūmai, migla),
- atsakomoji reakcija ir selekcija,

- šablonų plokštumos,
- akumuliatorinis buferis,
- gylio žymėjimas,
- judesio suliejimas.

OpenGL aplinkoje yra realizuotos šios trys bibliotekos:

- OpenGL paslaugų biblioteka (GLU), turinti keletą šablonų, kurie naudoja žemesnio lygio OpenGL komandas.
- OpenGL praplėtimas Windows šeimos sistemoms (GLX), kuris leidžia programuojant naudotis Windows langų šablonais.
- OpenGL programavimo gido pagalbinė biblioteka, kuri aprūpina šablonais, leidžiančiais kurti ir atidarinėti langus, valdyti X įvykius ir naudotis dažnai pasitaikančiomis kompleksinėmis geometrinėmis figūromis, tokiomis kaip kubai, sferos ar cilindrai [10].

1.6.3 DirectX ir OpenGL aplinkų palyginimas

Išanalizavus, abiejų aplinkų galimybes, lengvumą programuoti ir 1 priedo lentelėje pateikiamą OpenGL 1.2 ir DirectX 7, 8 versijų funkcijų palyginimą, nuspręsta naudoti DirectX funkcijas.

3 „TURIS“ dėžių pakavimo trimatėje erdvėje sistemos modelis

Skyriaus paskirtis – apibrėžti kuriamos dėžių pakavimo trimatėje erdvėje sistemos modelio architektūrą.

Šiame skyriuje pateikiamas išsamus kuriamos sistemos architektūros aprašymas, panaudojant įvairius skirtingus architektūros aprašymo būdus, siekiant išryškinti skirtingus sistemos aspektus. Bus aprašomi svarbiausi kuriamo produkto architektūros aspektai: panaudojimo atvejų (2.2 skyrius), statinis (2.3 skyrius), dinaminis (2.4 skyrius), išdėstymo (2.5 skyrius) ir duomenų vaizdai (2.6 skyrius). Kiekvienas iš išvardintų architektūros aspektų aprašomas atskirame skyriuje. Šiame skyriuje siekiama užfiksuoti ir aprašyti visus svarbiausius architektūros sprendimus, kurie bus panaudoti kuriant sistemą.

Detalesnė sistemos architektūra apibrėžiama naudojant šiuos skirtingus vaizdus:

- **Panaudojimo atvejų vaizdas** – pateikiami pagrindiniai sistemos panaudojimo atvejai. Naudojami modeliavimo elementai: panaudojimo atvejų diagrama.
- **Sistemos statinis vaizdas** – pateikiamas sistemos išskaidymas į paketus. Aprašomi bei detalizuojami pagrindiniai sistemos komponentai ir posistemiai. Naudojamas modeliavimo elementas: komponentų diagrama, klasių diagrama.
- **Sistemos dinaminis vaizdas** – detalizuojami ir aprašomi panaudojimo atvejai. Naudojamos modeliavimo priemonės: sekų, bendradarbiavimo, būsenų kaitos ir veiklos diagramos.
- **Išdėstymo vaizdas** – aprašoma techninės įrangos, kurioje sistema bus išdėstyta ir veiks, konfigūracija bei sistemos komponentai, išdėstyti atskiruose techninės įrangos mazguose. Naudojamos modeliavimo priemonės: komponentų ir sistemos išdėstymo diagramos.
- **Duomenų vaizdas** – pateikiamas duomenų bazės modelis.

3.1 Architektūrinių sprendimų pagrindimas

Dėžių pakavimo trimatėje erdvėje programa realizuota .NET C# programavimo kalba. C# programavimo kalba pasirinkta dėl šių priežasčių:

- **Patirtis** – turiu 2 metus programavimo su C# programavimo kalba patirties.
- **Pernešamumas** – sukurta programinė įranga .NET aplinkoje puikiai veikia visuose Windows šeimos platformose. .NET aplinka nereikalauja iš programinės įrangos kūrėjo papildomų pastangų, kad programa veiktų ant kitų Windows šeimos platformų. Reikalavimuose nėra nurodyta, kad sistema turi dirbti ne ant Windows šeimos platformos.
- **Objektiškumas** – C# yra objektiškai orientuota kalba, kurios bibliotekos išsiskiria geru objektiniu dizainu. Tai labai palengvina programavimą C# aplinkoje.
- **Populiarumas** – ši programavimo aplinka yra labai populiari, todėl yra labai daug informacijos, susijusios su įvairių problemų sprendimu. Taip pat galima rasti daugybę įvairių nemokamų ir labai patogių įrankių.

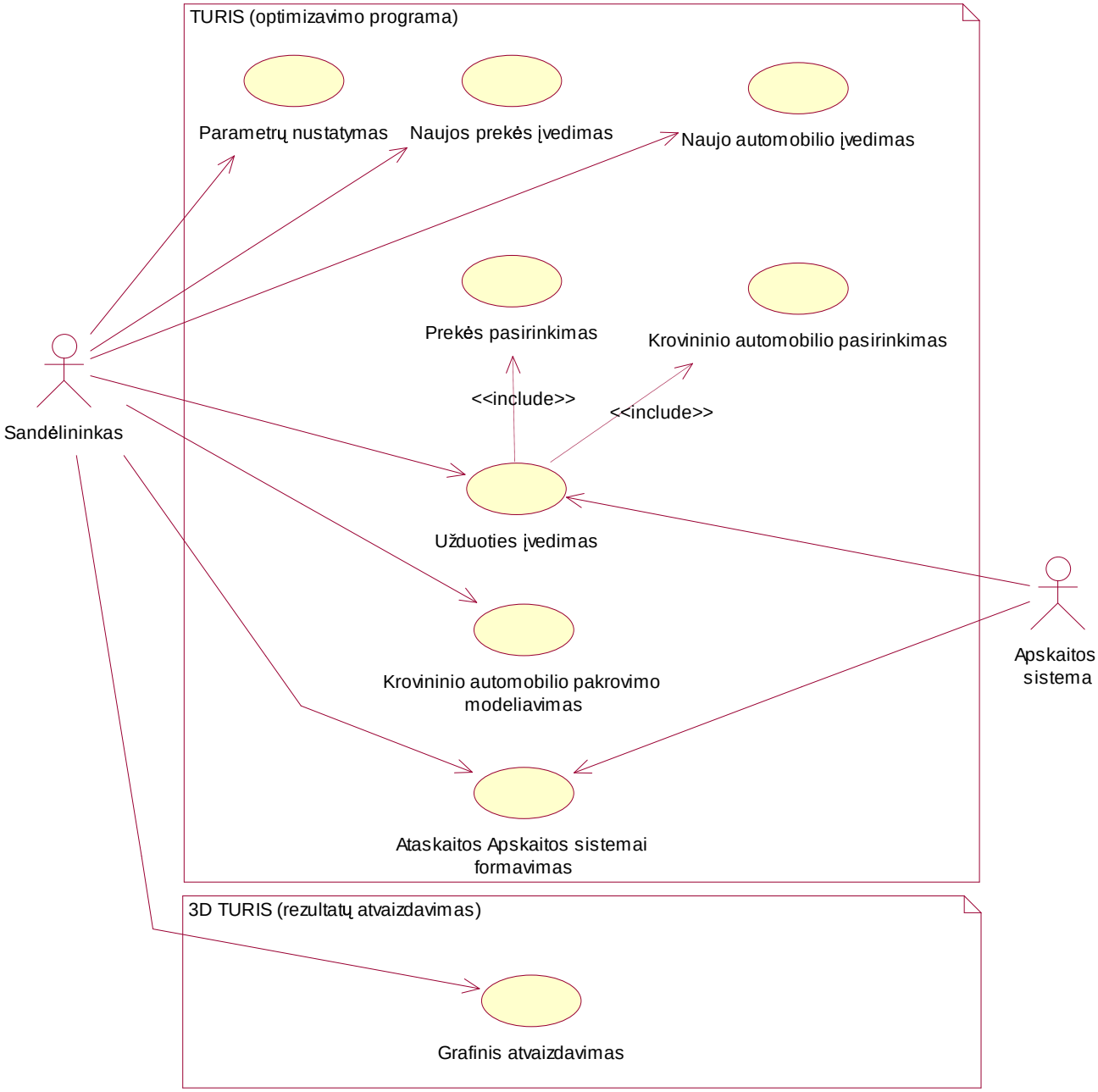
3D grafiniam varikliui kurti buvo pasirinkta DirectX funkcijų biblioteka, dėl šių priežasčių:

- **Suderinamumas** – C# labai gerai suderinama su šią funkcijų biblioteka.
- **Funkcionalumas** – ši funkcijų biblioteka turi daug patogių funkcijų.
- **Populiarumas** – dėl šios aplinkos populiarumo, internete galima rasti daugybę mokomųjų straipsnių ir informacijos apie įvairių problemų sprendimą.
- **Nepriklausomumas** – programuojant nereikia rūpintis kokia vaizdo plokštė yra instaliuota į kompiuterį. Kompiuteryje, kuriame leidžiama programa, turi būti suinstaliuoto DirectX tvarkyklės, kurios ir rūpinasi technine dalimi.

Duomenims apie užduotis, prekes, krovininius automobilius ir rezultatus saugoti buvo pasirinkta MS SQL DBVS, nes užsakovas naudoja šią DBVS informacijai apie prekes saugoti.

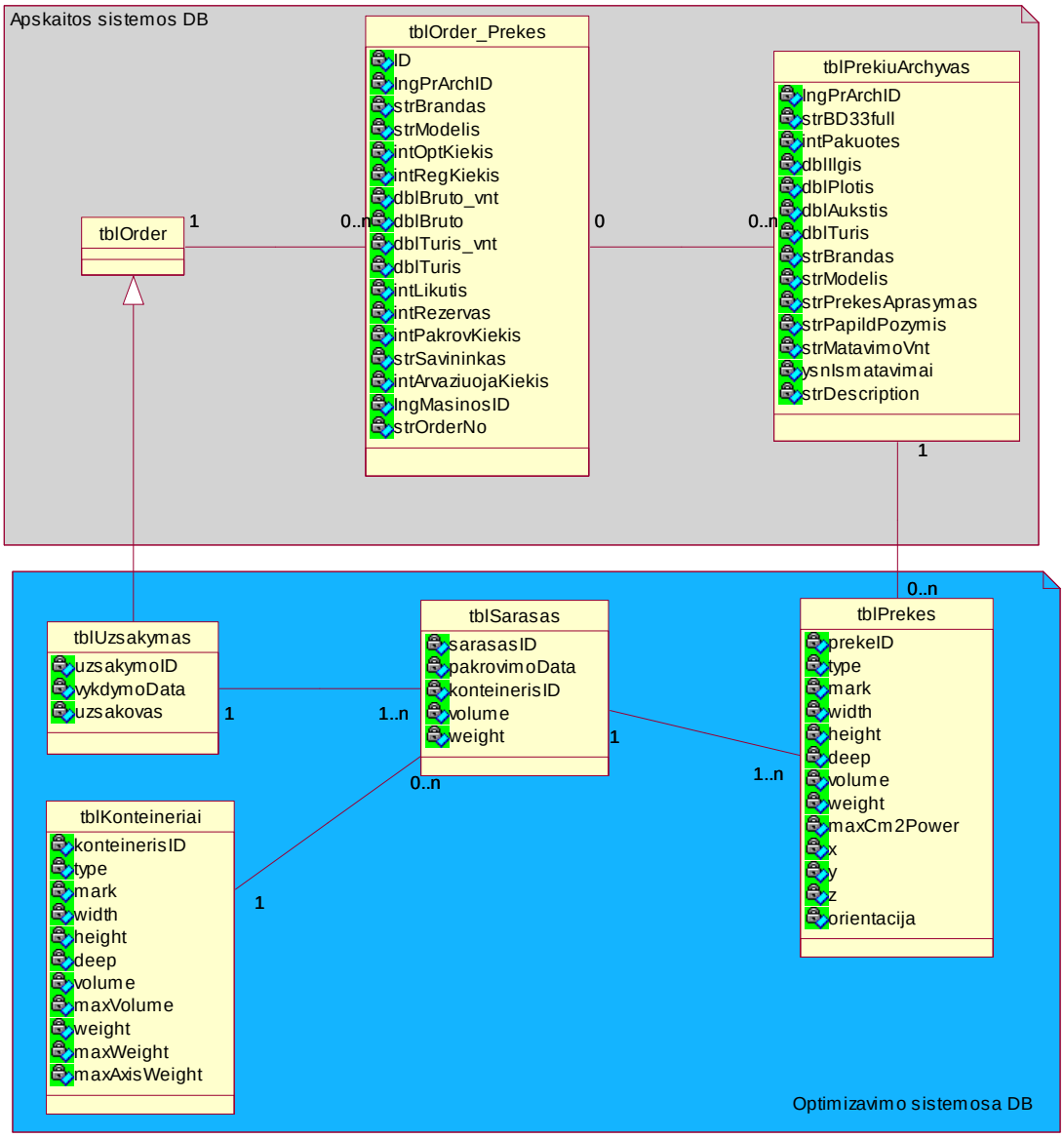
3.2 „TURIS“ sistemos panaudojimo atvejų vaizdas

Ribas tarp sistemos ir vartotojo nusako panaudojimo atvejų diagrama (18 pav.). Panaudojimo atvejų diagrama sudaroma įvertinant kiekvieną išskirtą veiklos įvykį ir kuriamos sistemos indėlį šio įvykio atžvilgiu.



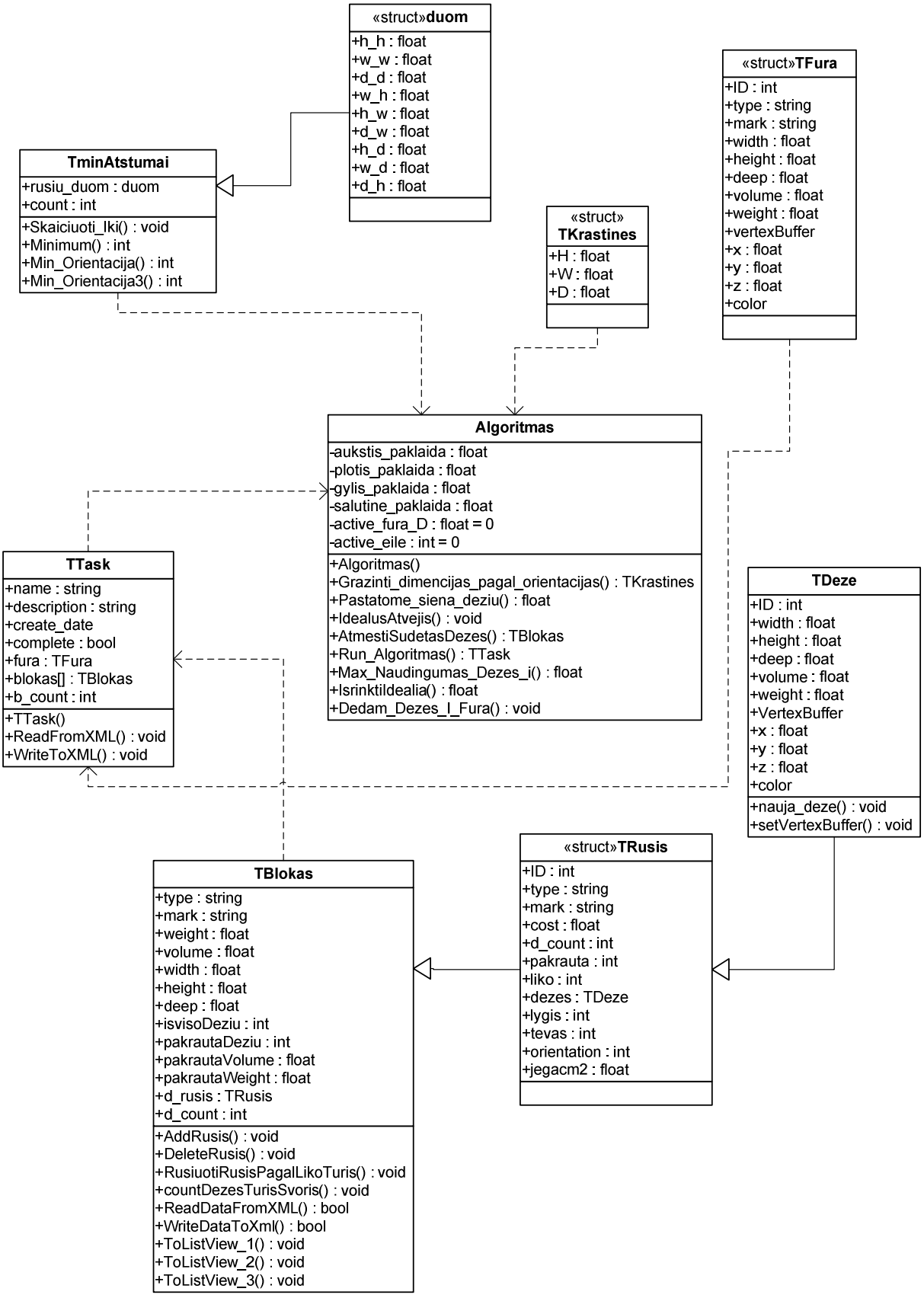
18 pav. Panaudojimo atvejų diagrama

3.3 „TURIS“ sistemos duomenų vaizdas



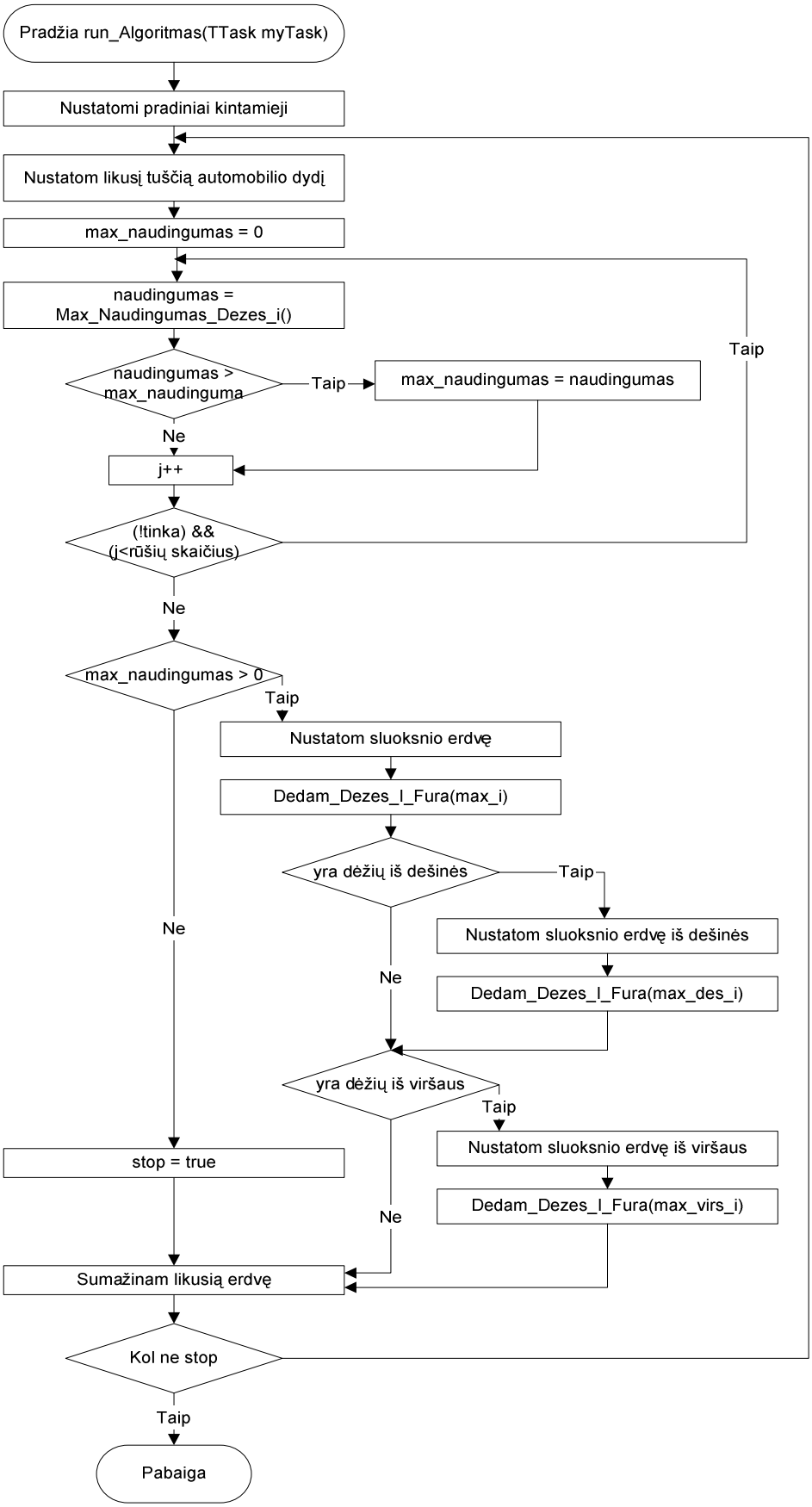
19 pav. Duomenų bazės modelis

3.4 Prekių pakavimo trimatėje erdvėje algoritmo klasių diagramos



20 pav. Algoritmo klasių diagramos

3.5 Prekių pakavimo trimatėje erdvėje algoritmas

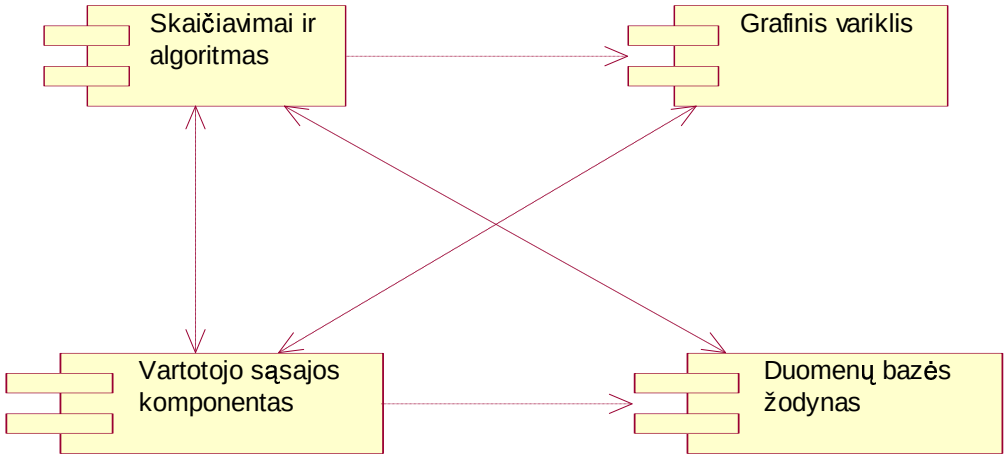


21 pav. Prekių pakavimo trimatėje erdvėje algoritmas

3.6 „TURIS“ sistemos statinis vaizdas

3.6.1 „TURIS“ sistemos komponentai

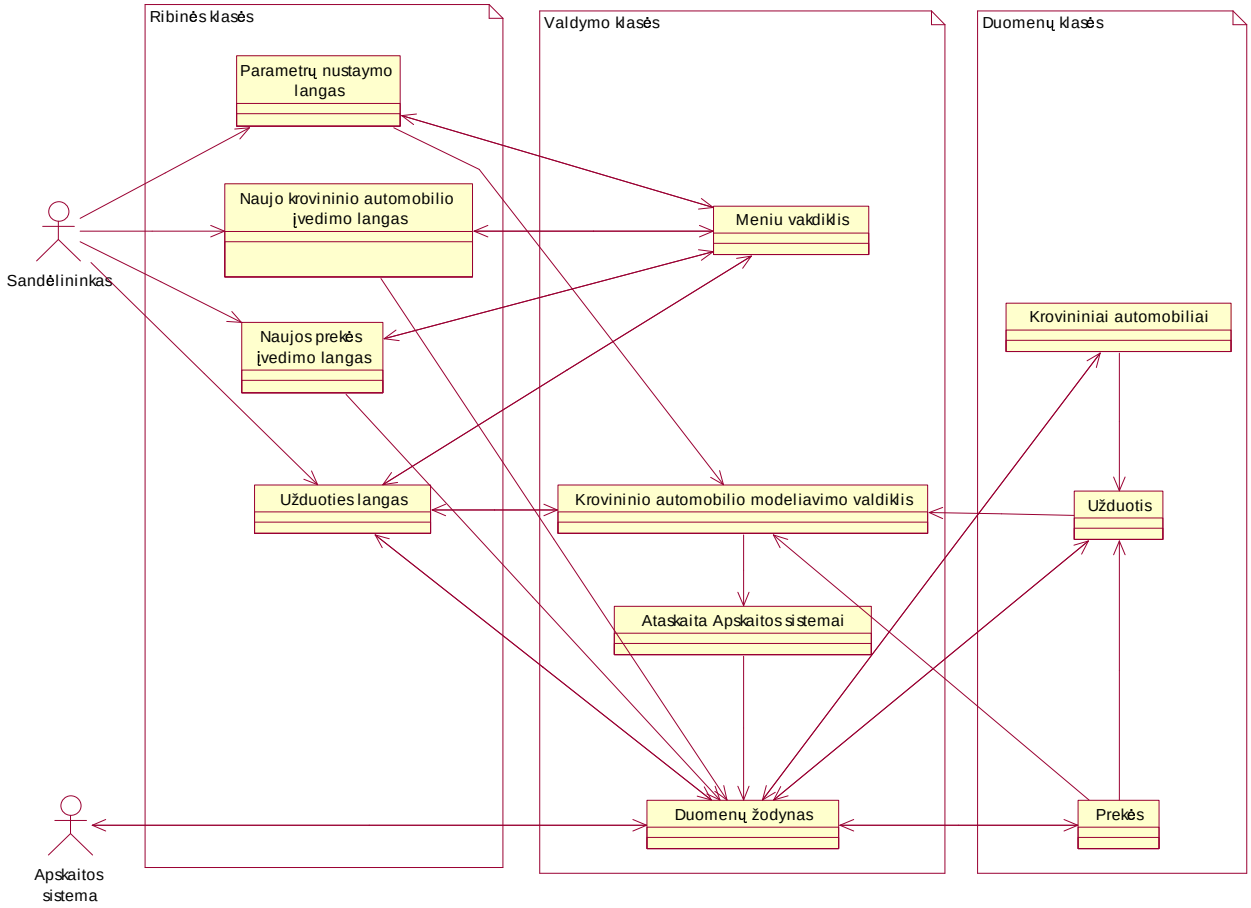
22 pav. pateikiamas sistemos išskaidymas į komponentus.



22 pav. Sistemos komponentų diagrama

3.6.2 Komponentų detalizavimas

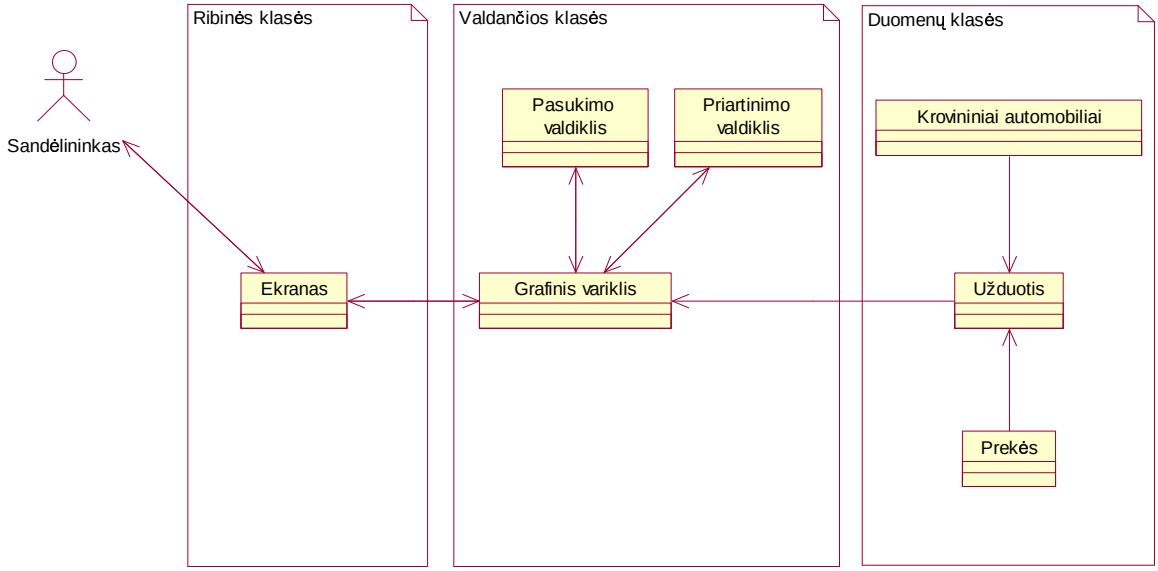
3.6.2.1 Komponentas „Skaičiavimai ir algoritmas“



23 pav. „Skaičiavimai ir algoritmai“ paketo klasių diagrama

Skaičiavimų ir algoritmo komponentas iš paduotų pradinių duomenų skaičiuoja rezultatus, kaip krauti prekes į krovininį automobilį. Šis komponentas pavaizduotas 23 pav.

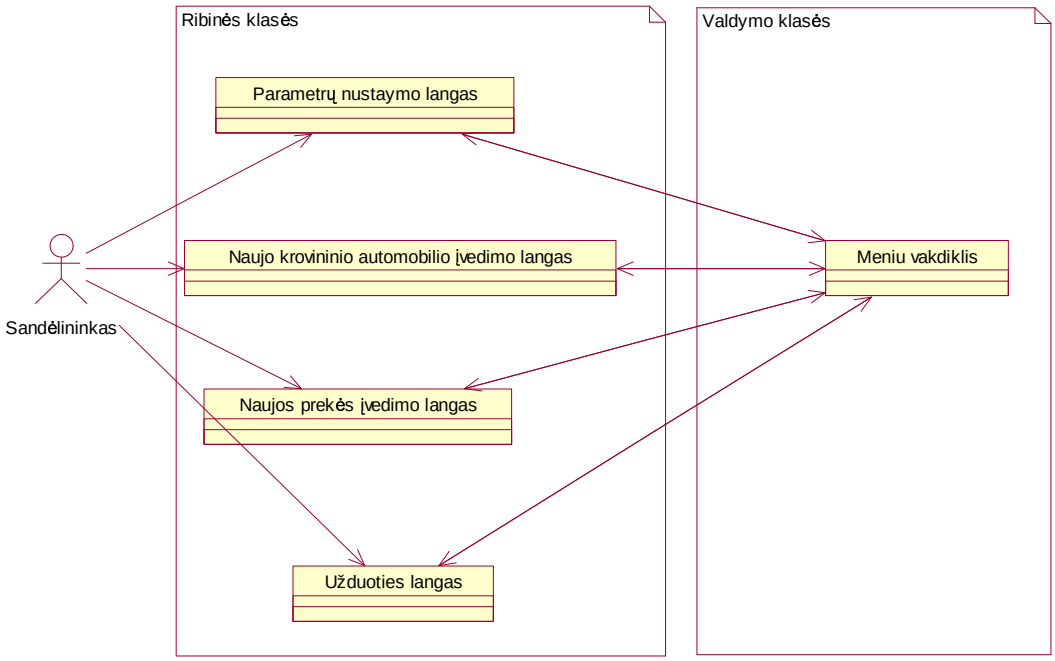
3.6.2.2 Komponentas „Grafinis variklis“



24 pav. „Grafinis variklis“ komponento klasių diagrama

Komponento paskirtis pateikti sandėlininkui rezultatus grafiškai, kad sandėlininkas galėtų lengvai ir aiškiai krauti krovininį automobilį. Vaizdas pavaizduojamas trimatėje erdvėje, yra galimybė vaizdą pasukti, paversti, priartinti, atitolinti. Šias funkcijas valdo grafinis variklis. Grafinis variklis apdoroja rezultatus ir juos atvaizduos ekrane. Grafinio variklio klasių diagrama pateikiama 24 pav.

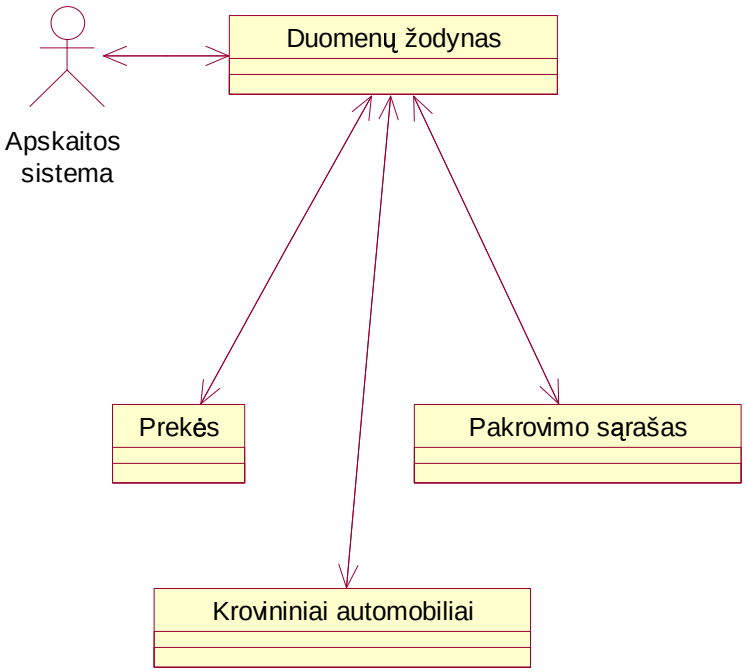
3.6.2.3 Komponentas „Vartotojo sąsaja“



25 pav. „Vartotojo sąsaja“ paketo klasių diagrama

Vartotojo sąsajos komponentas yra svarbus vartotojo ir sistemos bendravimo komponentas, kurio pagalba vartotojas gali atlikti programos funkcijas ir peržiūrėti rezultatus. Vartotojo sąsajos komponentas pavaizduotas 25 pav.

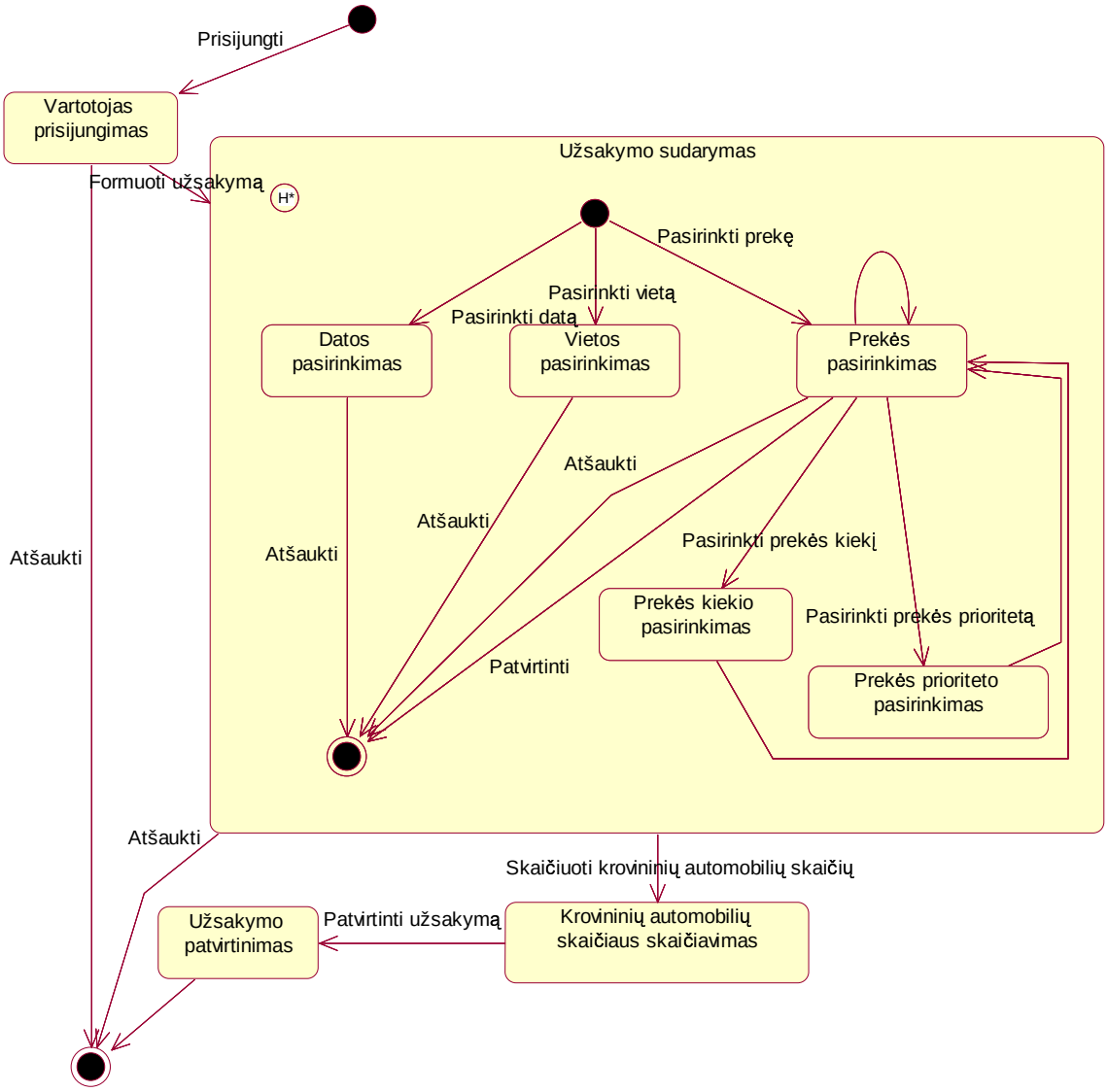
3.6.2.4 Komponentas „Duomenų bazės žodynas“



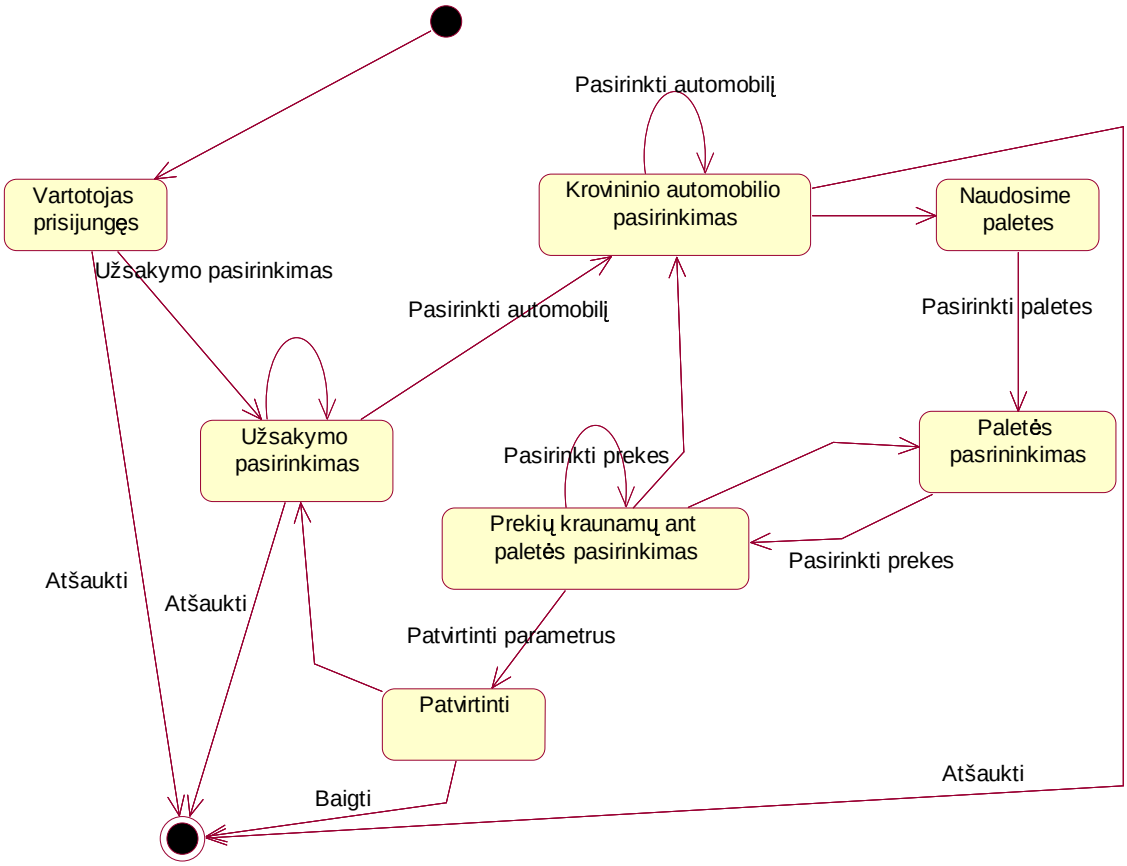
26 pav. „Duomenų bazės žodynas“ paketo klasių diagrama

Kadangi duomenų bazės struktūra nėra statinė ir laikui bėgant keičiasi, svarbu nepririšti programos prie statinės duomenų bazės struktūros. Kad lengviau būtų pritaikyti pasikeitusią DB struktūrą, naudojamas duomenų žodyno komponentas (kaip pavyzdys būtų PRO/5® naudojamas duomenų žodynas). Duomenų žodyno komponento klasių diagrama pavaizduota 26 pav.

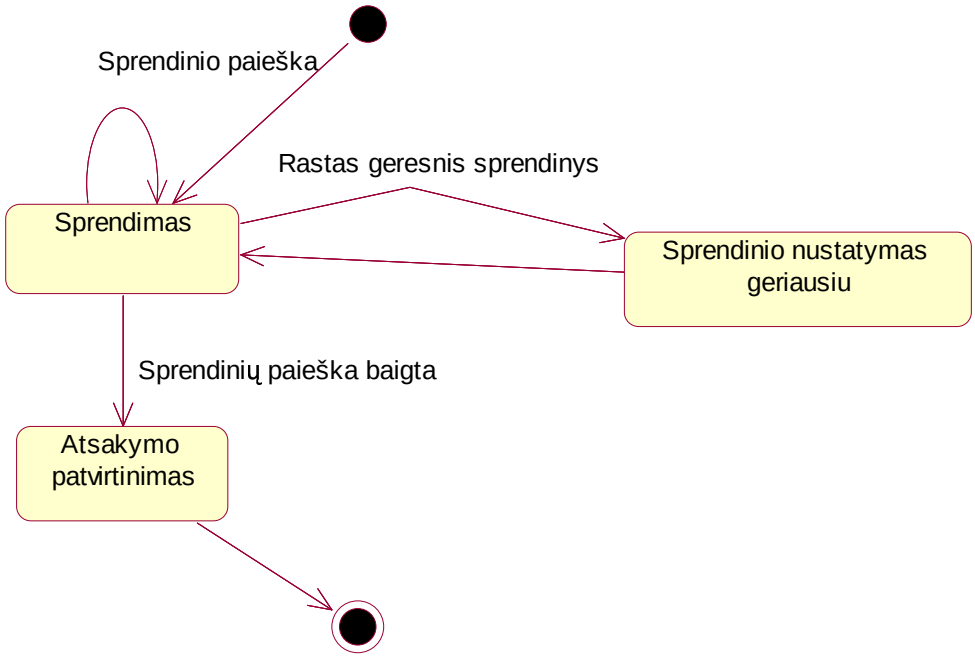
3.7 „TURIS“ sistemos būsenų diagramos



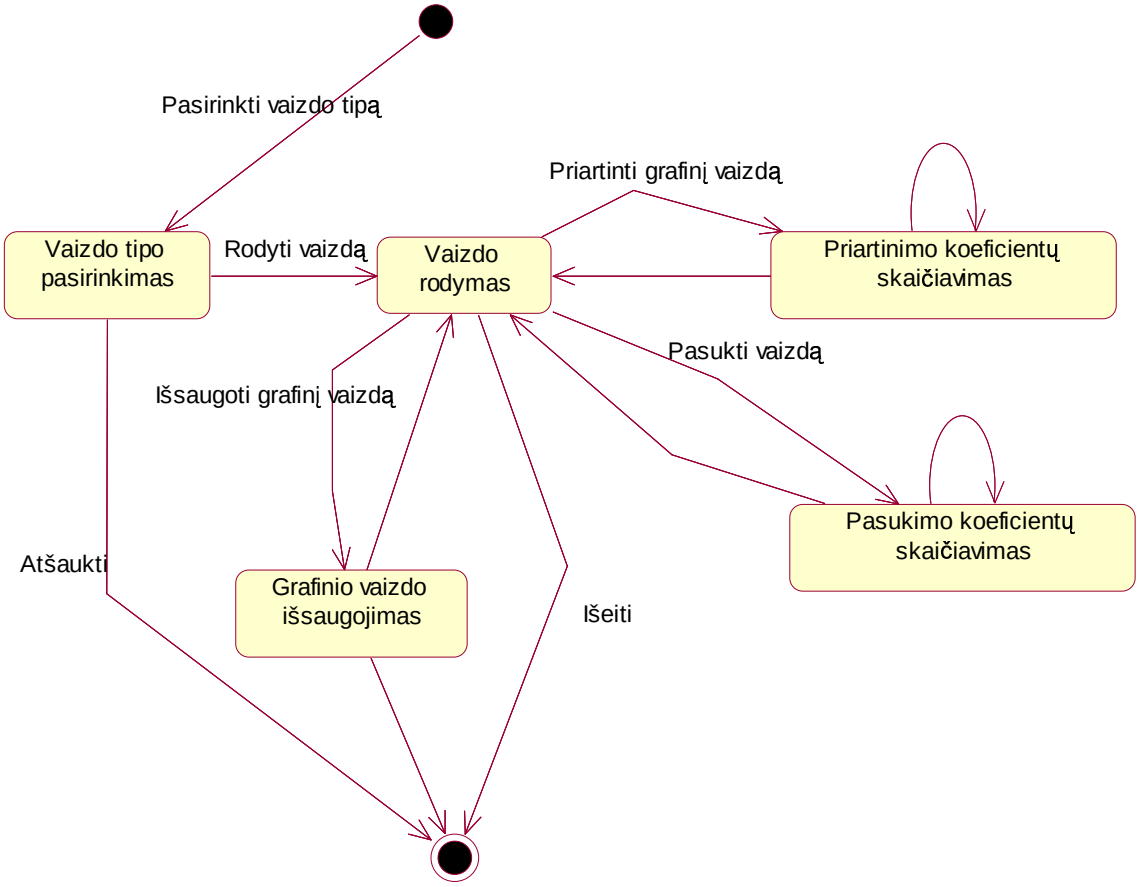
27 pav. „Prekių įvedimo“ būsenų diagrama



28 pav. „Parametrų nustatymo“ būsenų diagrama



29 pav. „Krovininio automobilio pakrovimo modeliavimo“ būsenų diagrama

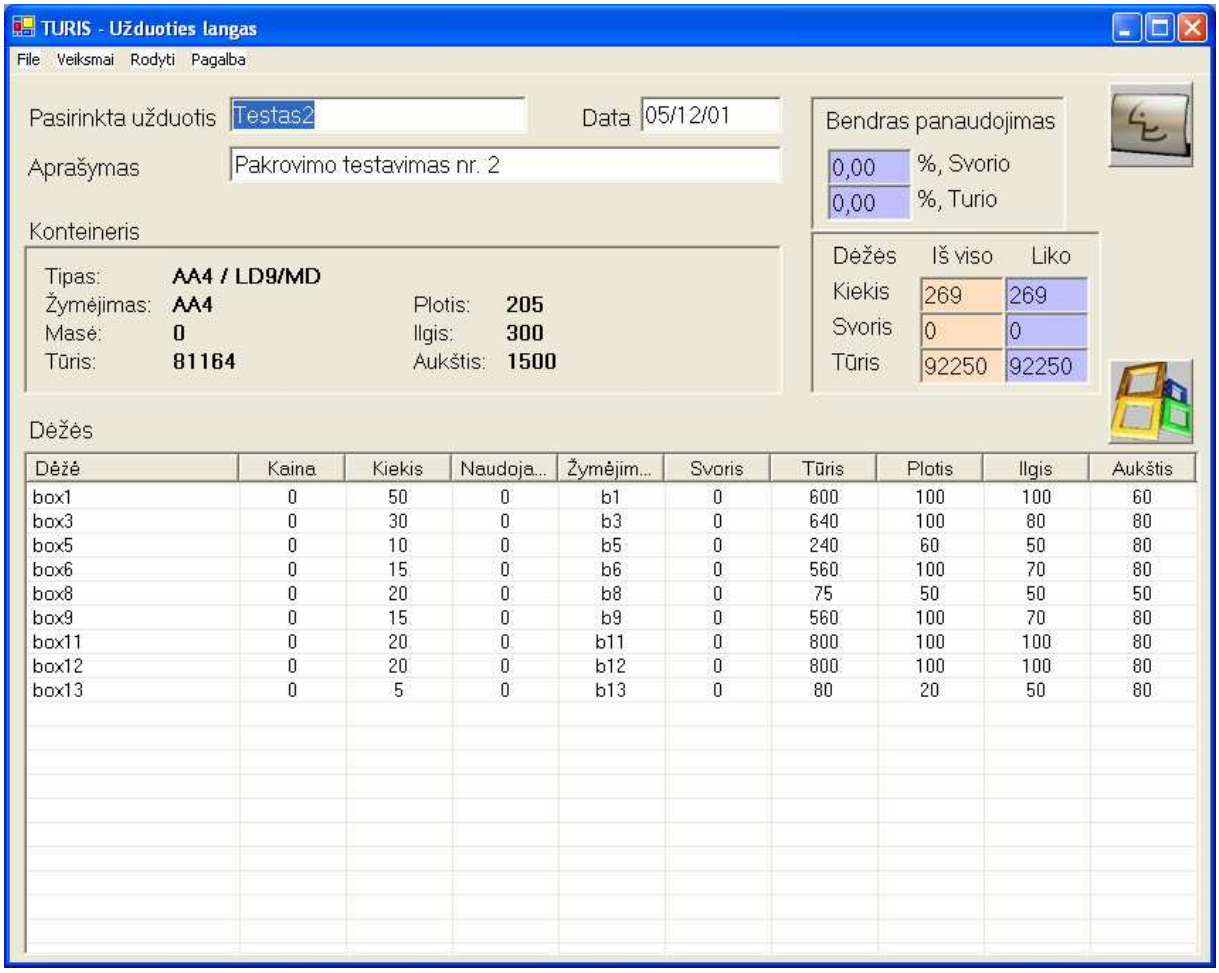


30 pav. „Grafinis atvaizdavimas“ būsenų diagrama

3.8 „TURIS“ sistemos langai

3.8.1 Užduoties langas

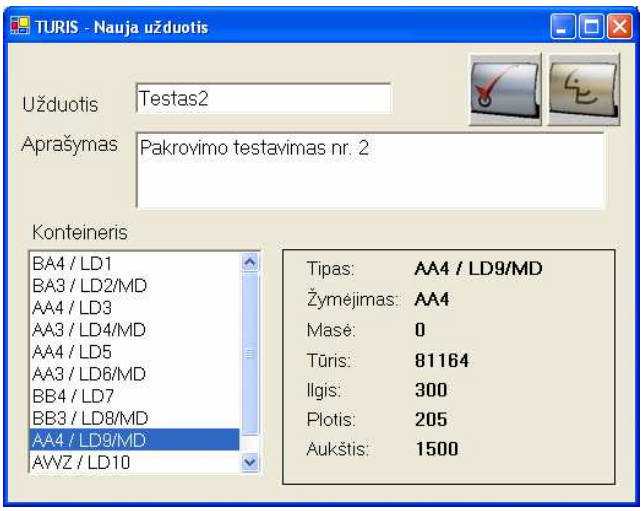
Užduoties langas (31 pav.) skirtas kurti naujoms užduotims arba jau sukurtoms redaguoti ir vykdyti. Užduotis – tai uždavinys, kai turime aibę dėžių ir šias dėžes norime sukrauti į krovininį automobilį.



31 pav. Užduoties langas

3.8.2 Naujos užduoties kūrimo langas

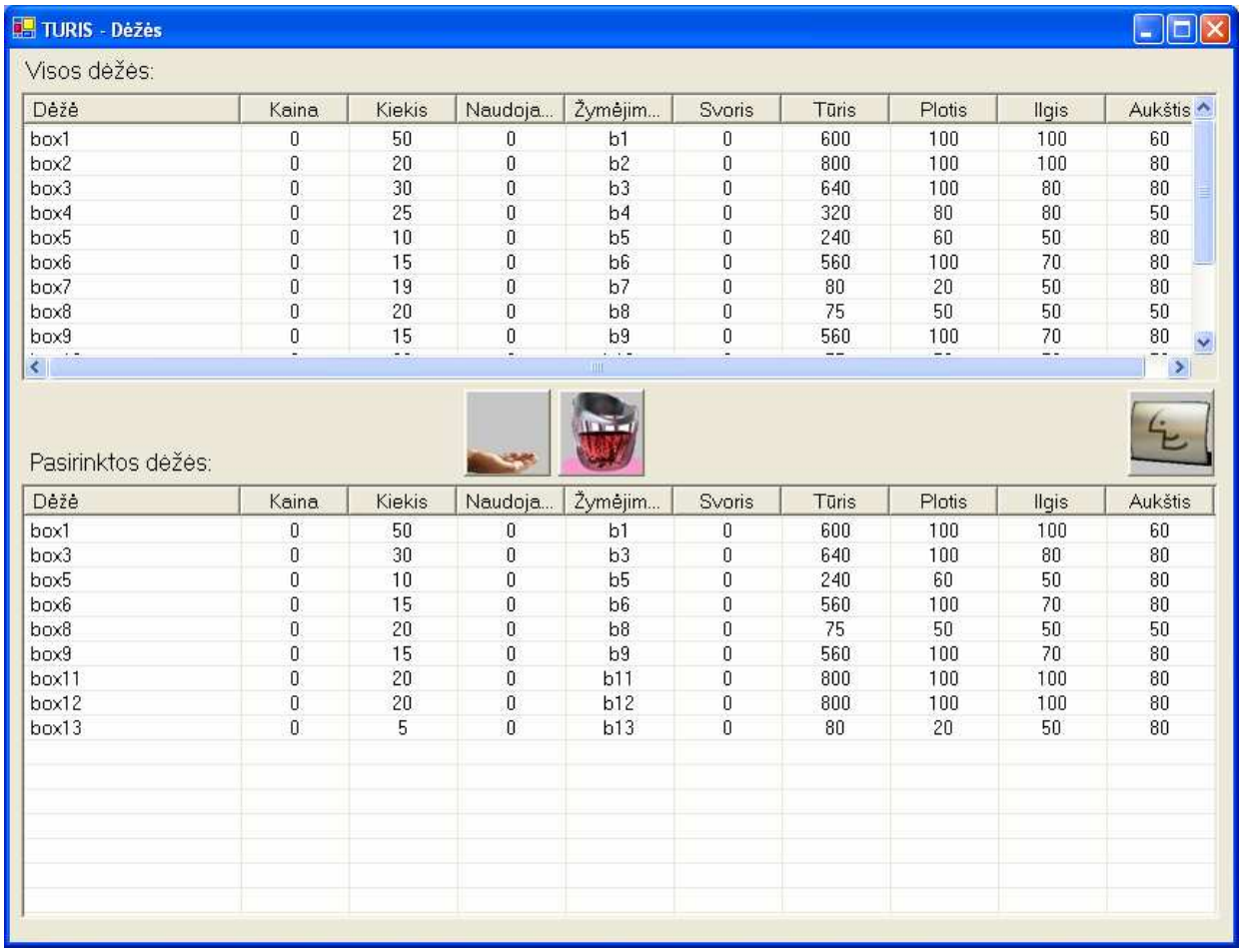
„Nauja užduotis“ (32 pav.) langas skirtas naują sukurti užduotį.



32 pav. Užduoties langas

3.8.3 Dėžių pasirinkimo langas

Langas „Dėžės“ (33 pav.) skirtas pasirinkti dėžėms, kurios bus kraunamos į konteinerį.



33 pav. Dėžių pasirinkimo langas

3.8.4 Optimizavimo parametrų nustatymo langas

Sukūrus naują arba užkrovus egzistuojančią užduotį, prieš atliekant pakrovimo modeliavimą, reikia nustatyti optimizavimo parametrus. Tai atliekama parametrų nustatymo lange (34 pav.). Optimizavimo parametrai reikalingi nurodyti kaip algoritmas turėtų skaičiuoti rezultatus.

34 pav. Optimizavimo parametrų kortelė

3.8.5 Skaičiavimų progreso langas

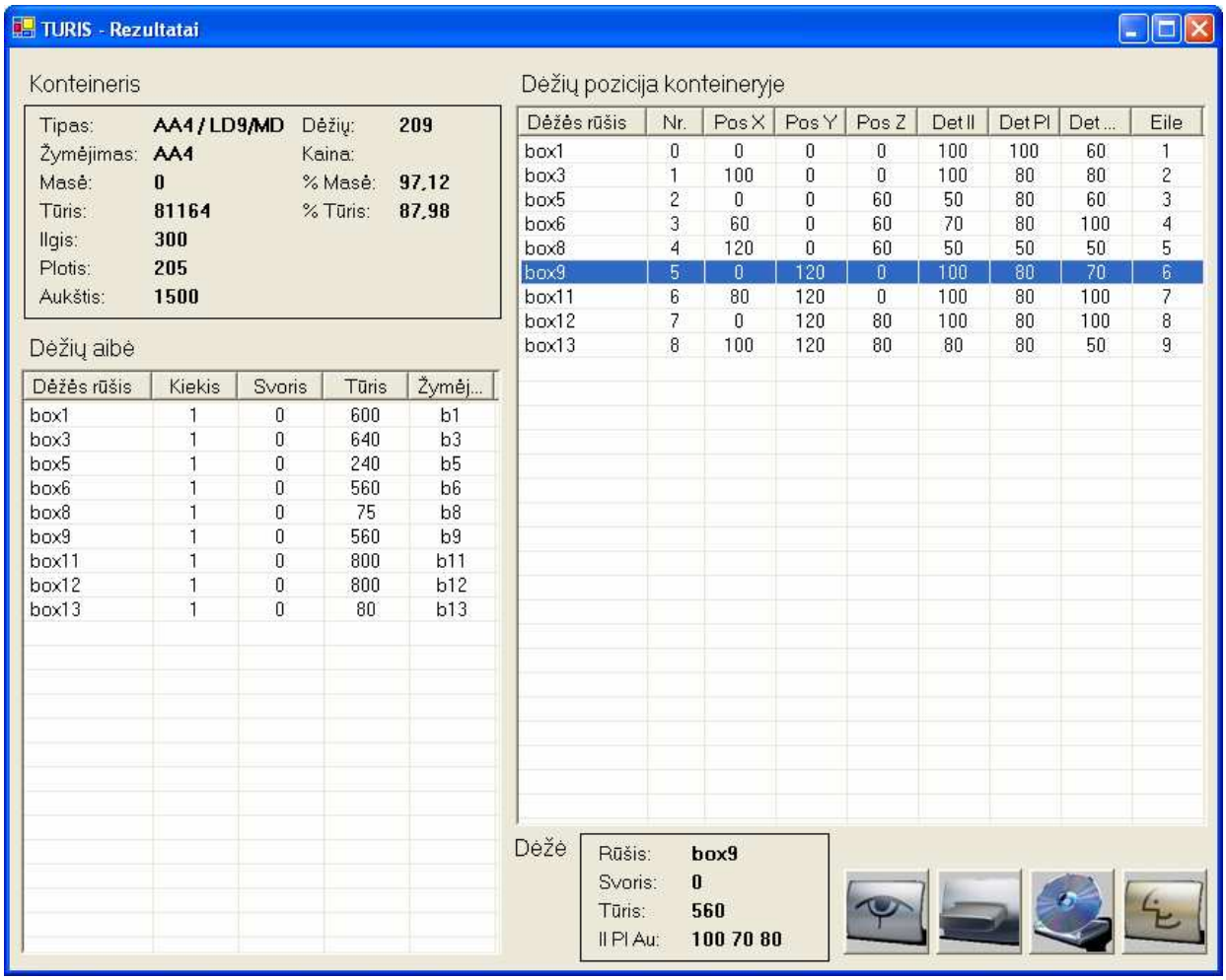
Suvedus visą informaciją apie užduotį ir nustačius parametrus galima pradėti vykdyti dėžių pakrovimo modeliavimo algoritmą. Algoritmo vykdomų skaičiavimų progresas matomas skaičiavimų lange (35 pav.).

35 pav. Skaičiavimų pažangos langas

Pasibaigus modeliavimui arba pasibaigus laikui, skirtam skaičiavimams, rezultatus galėsite peržiūrėti **Rezultatų lange** (36 pav.).

3.8.6 Rezultatų langas

Atlikus pakrovimo modeliavimą gaunami rezultatai atvaizduojami **Rezultatų** lange (36 pav.). Šiam lange pateikiama informacija kokias prekes ir kaip krauti.

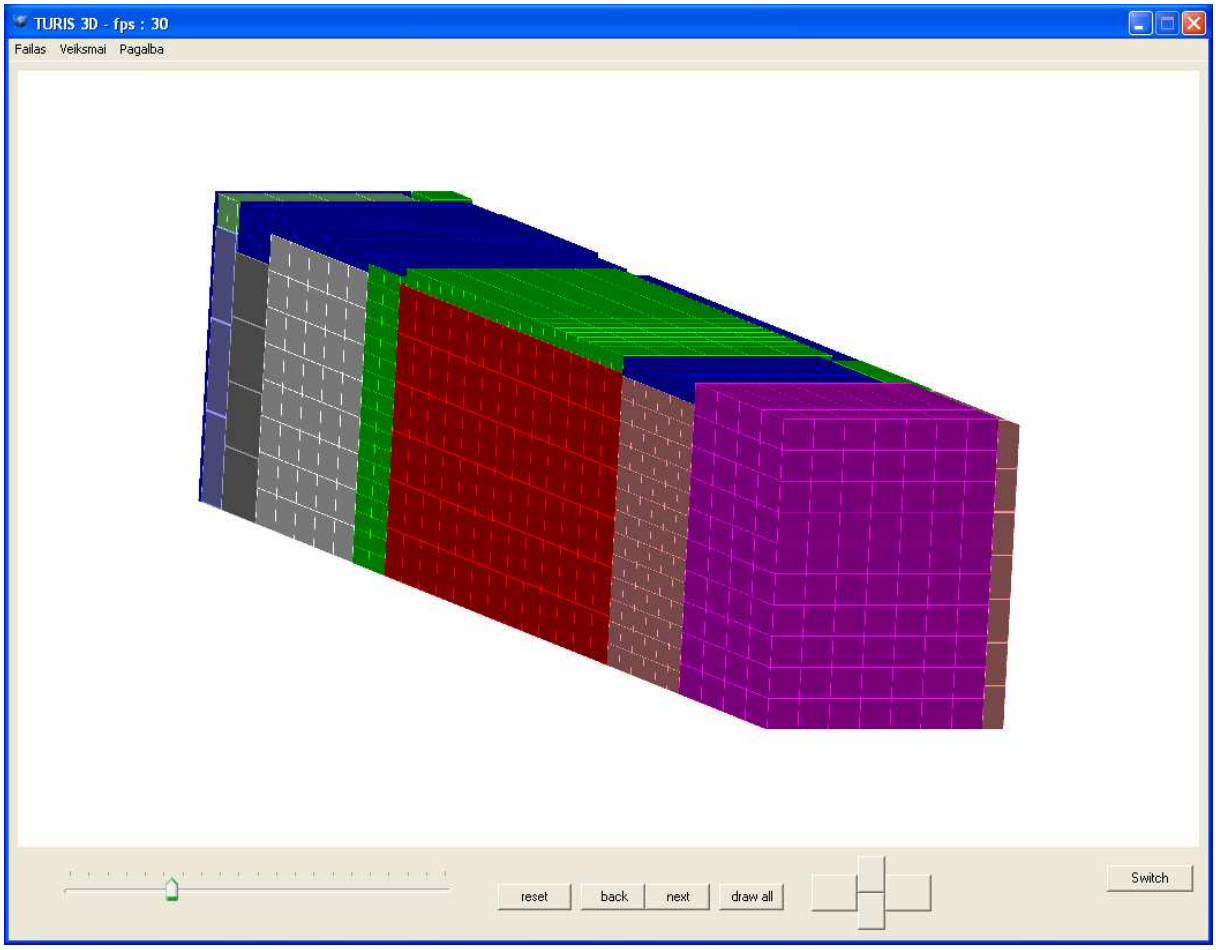


36 pav. Rezultatų langas

Iš šio lango paspaudus ant mygtuko, ant kurio pavaizduota akis, galima iškviesti rezultatų grafinio atvaizdavimo trimatėje erdvėje langą (37 pav.).

3.8.7 Rezultatų grafinio atvaizdavimo trimatėje erdvėje langas

Rezultatų grafinio atvaizdavimo trimatėje erdvėje langas (37 pav.) – tai vartotojui patogi priemonė grafiškai atvaizduoti sumodeliuotus dėžių krovimo į konteinerį duomenis. Šiame lange grafiškai (3D erdvėje) pateikiami užduoties modeliavimo rezultatai.



37 pav. Rezultatų grafinio atvaizdavimo trimatėje erdvėje langas

4 Prekių pakavimo trimatėje erdvėje sistemų tyrimas

4.1 Tyrimo tikslas

Išanalizavus panašias sistemas ir problemas, susijusias su dėžių pakavimu trimatėje erdvėje, buvo sukurtas algoritmas, kuris trimatėje erdvėje modeliuoja optimalų dėžių pakrovimą į krovininį automobilį, įvertindamas pagrindinius apribojimus, kurie aprašyti analizės dalyje (2 skyrius). Bei buvo sukurta dėžių pakavimo trimatėje erdvėje sistema „TURIS“, kuri naudoja šį algoritmą. Tyrimo dalyje atliekamas šios sistemos palyginimas su kitomis egzistuojančiomis sistemomis ir įvertinama sukurto algoritmo kokybė.

4.2 Tyrimo scenarijus

Eksperimentas bus atliekamas su keturiomis sistemomis, kurios modeliuoja dėžių krovimą į krovininį automobilį trimatėje erdvėje, ir šiame projekte sukurta „TURIS“ sistema.

Kiekvienai programai bus paduodami testiniai duomenys ir su jais atliekami skaičiavimai. Bus fiksuojamas algoritmo vykdymo laikas, kiek dėžių pavyko pakrauti į krovininį automobilį, kokiu procentu pavyko užpildyti automobilio erdvę.

Bus atliktas gautų rezultatų apibendrinimas.

4.3 Tyrimo objektai ir kriterijai

4.3.1 Tiriamos sistemos

Tyrime naudojamos šios dėžių pakavimo trimatėje erdvėje sistemos:

„CargoWiz“ (detaliau aprašyta 2.6.1 skyriuje).

„3D Load Packer” (detaliau aprašyta 2.6.1 skyriuje).

„LoadPlanner” (detaliau aprašyta 2.6.1 skyriuje).

„Cube-IQ” (detaliau aprašyta 2.6.1 skyriuje).

„TURIS” – šiame darbe sukurta sistema.

4.3.2 Testiniai atvejai

Prieš tai išvardintoms sistemoms bus paduodami 2 skirtingų kategorijų testiniai duomenys.

Pirmoji kategorija (A) charakterizuoja kaip sistemos algoritmo greitis ir efektyvumas priklauso nuo kraunamų dėžių kiekio. Dėžių rūšių skaičius pastovus ir nesikeičia. Pirmajai testinių atvejų kategorijai priklauso 3 tipų testiniai atvejai:

Testinis atvejis A1. Bendras dėžių kiekis lygus 625.

2 lentelė. A1 testinio atvejo testiniai duomenys

Pavadinimas	Kiekis	Plotis, cm	Ilgis, cm	Aukštis, cm	Svoris, kg	Orientacijos
Muzikinis centras	400	36	61	29	8	Visos
Muzikinis centras 2	125	84	61	59	10	Visos
Televizorius 102"	100	87	65	82	17	Visos

Testinis atvejis A2. Bendras dėžių kiekis lygus 1875.

3 lentelė. A2 testinio atvejo testiniai duomenys

Pavadinimas	Kiekis	Plotis, cm	Ilgis, cm	Aukštis, cm	Svoris, kg	Orientacijos
Muzikinis centras	1200	36	61	29	8	Visos
Muzikinis centras 2	300	84	61	59	10	Visos
Televizorius 102"	375	87	65	82	17	Visos

Testinis atvejis A3. Bendras dėžių kiekis lygus 7250.

4 lentelė. A3 testinio atvejo testiniai duomenys

Pavadinimas	Kiekis	Plotis, cm	Ilgis, cm	Aukštis, cm	Svoris, kg	Orientacijos
Muzikinis centras	6000	36	61	29	8	Visos
Muzikinis centras 2	750	84	61	59	10	Visos
Televizorius 102"	500	87	65	82	17	Visos

Antroji kategorija (R) charakterizuoja kaip sistemos algoritmo greitis ir efektyvumas priklauso nuo kraunamų dėžių rūšių skaičiaus, šiuo atvejų bendras dėžių kiekis pastovus. Antrajai testinių atvejų kategorijai priklauso, kaip ir pirmajai, 3 tipų testiniai atvejai:

Testinis atvejis R1. Testinius duomenis sudaro 10 skirtingų rūšių dėžės. Bendras dėžių kiekis lygus 12 300.

5 lentelė. R1 testinio atvejo testiniai duomenys

Pavadinimas	Kiekis	Plotis, cm	Ilgis, cm	Aukštis, cm	Svoris, kg	Orientacijos
Garų surinktuvas	250	64	56	17	1	Visos
Keptuve Tefal	300	31	58	30	1	Visos
Mikseriai	500	42	31,5	21,5	1	Visos
Muzikinis centras	1250	36	61	29	1	Visos
Plauku džiovintuvai	1000	46	31	25	1	Visos
Radijo imtuvas	5000	28	19	7	1	Visos
Skalbimo mašinos	750	65	53	89	1	Visos
Svarstyklės	1250	30	42	17	1	Visos
Televizorius 45"	1000	44	45	47	1	Visos
Virtuvinis kombainas	1000	47	51	35	1	Visos

Testinis atvejis R2. Antrąjį testinį variantą sudaro testiniai duomenys, kuriuose yra 30 rūšių dėžių, tačiau skirtingų tik 10, kurios pasikartoja 3 kartus. Bendras dėžių kiekis toks pats kaip ir pirmame testiniame atvejyje. Rūšies dėžių kiekis padalintas iš trijų ir paskirstytas, taip virtualiai sukuriant skirtingas dėžių rūšis.

6 lentelė. R2 testinio atvejo testiniai duomenys

Pavadinimas	Kiekis	Plotis, cm	Ilgis, cm	Aukštis, cm	Svoris, kg	Orientacijos
Garų surinktuvas	85	64	56	17	1	Visos
Keptuve Tefal	100	31	58	30	1	Visos
Mikseriai	165	42	31,5	21,5	1	Visos
Muzikinis centras	400	36	61	29	1	Visos
Plauku džiovintuvai	350	46	31	25	1	Visos
Radijo imtuvas	1650	28	19	7	1	Visos
Skalbimo mašinos	250	65	53	89	1	Visos
Svarstyklės	400	30	42	17	1	Visos
Televizorius 45"	350	44	45	47	1	Visos

Pavadinimas	Kiekis	Plotis, cm	Ilgis, cm	Aukštis, cm	Svoris, kg	Orientacijos
Garų surinktuvas	85	64	56	17	1	Visos
Keptuve Tefal	100	31	58	30	1	Visos
Virtuvinis kombainas	350	47	51	35	1	Visos

X

3

Testinis atvejis R3.

Trečiąjį testinį variantą sudaro testiniai duomenys, kuriuose yra 100 rūšių dėžių, tačiau skirtingų tik 10, kurios pasikartoja 10 kartų. Bendras dėžių kiekis toks pats kaip ir pirmame testiniame atvejyje. Rūšies dėžių kiekis padalintas iš dešimt ir paskirstytas, taip virtualiai sukuriant skirtingas dėžių rūšis.

7 lentelė. R3 testinio atvejo testiniai duomenys

Pavadinimas	Kiekis	Plotis, cm	Ilgis, cm	Aukštis, cm	Svoris, kg	Orientacijos
Garų surinktuvas	25	64	56	17	1	Visos
Keptuve Tefal	30	31	58	30	1	Visos
Mikseriai	50	42	31,5	21,5	1	Visos
Muzikinis centras	125	36	61	29	1	Visos
Plauku džiovintuvai	100	46	31	25	1	Visos
Radijo imtuvas	500	28	19	7	1	Visos
Skalbimo mašinos	75	65	53	89	1	Visos
Svarstyklės	125	30	42	17	1	Visos
Televizorius 45"	100	44	45	47	1	Visos
Virtuvinis kombainas	100	47	51	35	1	Visos

X

10

4.3.3 Tyrimui naudojama techninė aparatūra

Testavimui atlikti naudojamas kompiuteris, kurio parametrai pateikiami 7 lentelėje.

8 lentelė. Kompiuterio parametrai, su kuriuo atliekami testavimai

Procesorius	Pentium IV 1,73 GHz;
RAM	512 MB;
HDD	80 GB;
Operacinė sistema	Microsoft Windows XP
Monitorius	15,4 colių;
Vaizdo plokštė	128 MB;

4.3.4 Vertinimo kriterijai:

Tyrimo metu vertinami šie algoritmo efektyvumo kriterijai:

- Algoritmo **skaičiavimo laikas** – kiek laiko užtruko skaičiavimai. Jei algoritmas skaičiuoja iteracijomis ir jas parodo ekrane, tai skliausteliuose prie laiko užrašomas iteracijų skaičius. „CargoWiz“ algoritmas vykdo skaičiavimus vieną iteraciją arba keturias, todėl pateikiami du šio algoritmo parametrai. Pirmas vienos iteracijos, o antras (skliausteliuose) – keturių iteracijų.
- Kiek **pakrauta dėžių** – pakrautų ir nepakrautų dėžių į krovininį automobilį skaičius.
- Automobilio **pakrovimo tankis** – procentas, kiek pavyko pakrauti krovininio automobilio tūrio.

4.4 Tyrimo eiga

Pagal sukurtus testinius atvejus sukurti kiekvienai sistemai pradiniai duomenys. Sistemoms, kurios nerodo skaičiavimo trukmės, skaičiavimo laikas matuojamas su kompiuteriniu laikrodžiu. Kiekvienai sistemai paduodami testiniai duomenys, o gauti rezultatai surašomi į lenteles.

4.4.1 Tyrimo rezultatai su testiniais duomenimis A1

9 lentelė. Testavimo rezultatai su A1 testiniais duomenimis

Programa	Skaičiavimo laikas	Pakrauta dėžių	Nepakrauta dėžių	Pakrovimo tankis
„CargoWiz“	3 s. (12 s.)	535 (528)	90 (97)	92,1% (91,7%)
„3D Load Packer”	125 s.	590	30	89,38 %
“LoadPlanner”	62 s.	605	20	93,22 %
“Cube-IQ”	24 s. (1995 iter.)	548	77	95,5%
“TURIS”	16 ms.	608	17	94,21%

Pastabos: „3D Load Packer“ skaičiavimų laikas viršija 2 minutes, kai tuo tarpu kiti algoritmai skaičiuoja žymiai greičiau, todėl toliau ši sistema nebus testuojama.

4.4.2 Tyrimo rezultatai su testiniais duomenimis A2

10 lentelė. Testavimo rezultatai su A2 testiniais duomenimis

Programa	Skaičiavimo laikas	Pakrauta dėžių	Nepakrauta dėžių	Pakrovimo tankis
„CargoWiz“	8 s. (41 s.)	1375 (1387)	500 (488)	96,2% (96,5%)
„3D Load Packer”	–	–	–	–
“LoadPlanner”	379 s.	1703	172	94,29%
“Cube-IQ”	44 s. (1565 iter.)	1446	429	97,31%
“TURIS”	24 ms.	1689	186	96,86%

Pastabos: „LoadPlanner“ skaičiavimų laikas viršija 5 minutes, todėl toliau ši sistema nebus testuojama.

4.4.3 Tyrimo rezultatai su testiniais duomenimis A3

11 lentelė. Testavimo rezultatai su A3 testiniais duomenimis

Programa	Skaičiavimo laikas	Pakrauta dėžių	Nepakrauta dėžių	Pakrovimo tankis
„CargoWiz“	68 s. (248 s.)	6782 (5269)	468 (1981)	92,4% (95,3%)
„3D Load Packer”	–	–	–	–
“LoadPlanner”	–	–	–	–
“Cube-IQ”	41 s. (1730 iter.)	6751	499	97,31%
“TURIS”	31 ms.	6885	365	96,86%

4.4.4 Tyrimo rezultatai su testiniais duomenimis R1

12 lentelė. Testavimo rezultatai su R1 testiniais duomenimis

Programa	Skaičiavimo laikas	Pakrauta dėžių	Nepakrauta dėžių	Pakrovimo tankis
„CargoWiz“	188 s.	5335	6965	98%
„3D Load Packer”	–	–	–	–
“LoadPlanner”	–	–	–	–
“Cube-IQ”	99 s. (105 iter.)	5592	6708	98,99%
“TURIS”	125 ms.	11041	1259	97,97%

4.4.5 Tyrimo rezultatai su testiniais duomenimis R2

13 lentelė. Testavimo rezultatai su R2 testiniais duomenimis

Programa	Skaičiavimo laikas	Pakrauta dėžių	Nepakrauta dėžių	Pakrovimo tankis
„CargoWiz“	280 s.	5785	6515	97,9%
„3D Load Packer”	–	–	–	–
“LoadPlanner”	–	–	–	–
“Cube-IQ”	100 s. (41 iter.)	6199	6101	98,82%
“TURIS”	734 ms.	9722	2578	98,58%

4.4.6 Tyrimo rezultatai su testiniais duomenimis R3

14 lentelė. Testavimo rezultatai su R3 testiniais duomenimis

Programa	Skaičiavimo laikas	Pakrauta dėžių	Nepakrauta dėžių	Pakrovimo tankis
„CargoWiz“	318 s.	5752	6548	98,3%
„3D Load Packer”	–	–	–	–
“LoadPlanner”	–	–	–	–
“Cube-IQ”	103 s. (50 iter.)	6758	5542	98,73%
“TURIS”	13 s.	8249	4051	98,41%

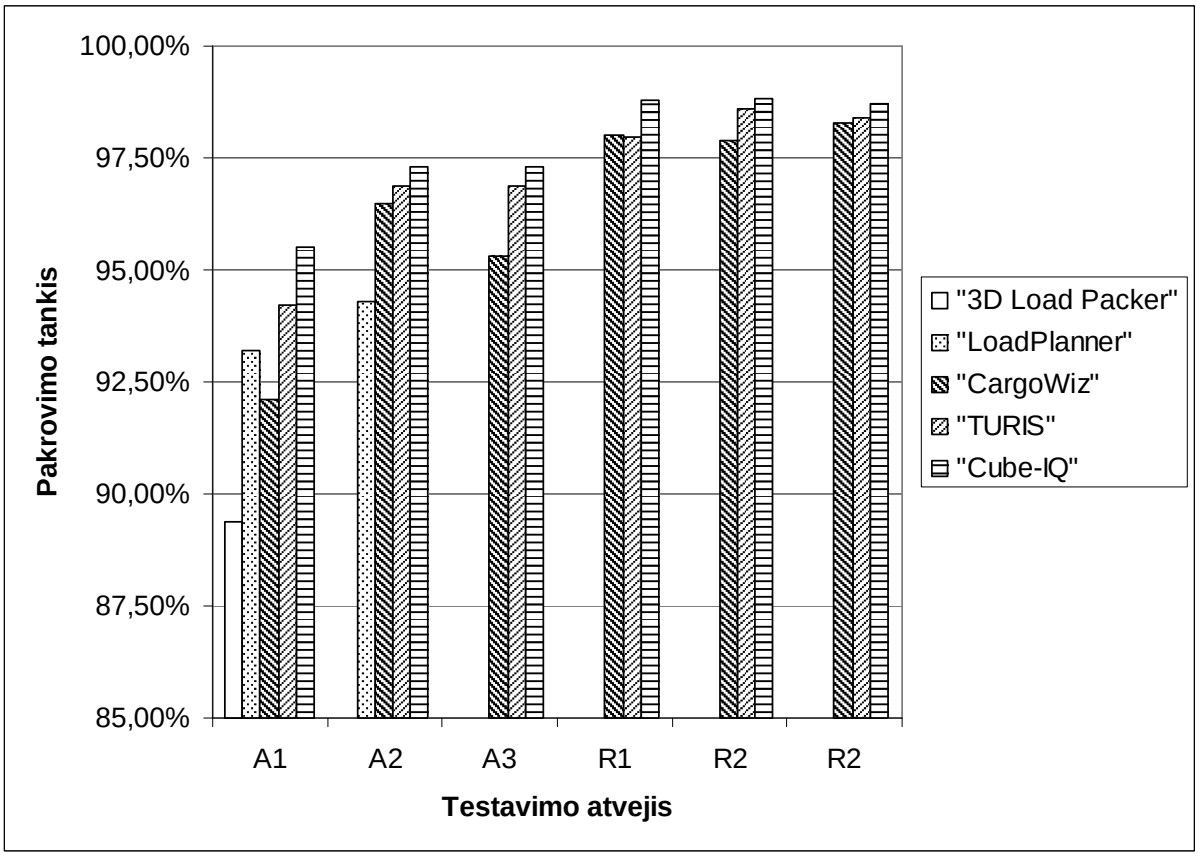
4.5 Tyrimo apibendrinimas

Sistemų testavimo rezultatai atlikti su visais testiniais atvejais pateikiami 15 lentelėje. Kai dėžių skaičius viršija 1 000, „3D Load Packer“ ir „LoadPlanner“ parodė labai prastus rezultatus, todėl su kitais testiniais variantais sistemos nebetestuotos.

15 lentelė. Dėžių pakavimo trimatėje erdvėje sistemų algoritmų efektyvumai

Programa Testinis atvejis	„CargoWiz“	„3D Load Packer”	“LoadPlanner”	“Cube-IQ”	“TURIS”
A1	92,1%	89,38%	93,22%	95,5%	94,21%
A2	96,5%	–	94,29%	97,31%	96,86%
A3	95,3%	–	–	97,31%	96,86%
R1	98%	–	–	98,8%	97,97%
R2	97,9%	–	–	98,82%	98,58%
R2	98,3%	–	–	98,73%	98,41%

37 paveiksle testavimo duomenys pateikiami grafiškai.



38 pav. Dėžių pakavimo trimatėje erdvėje sistemų algoritmų efektyvumai

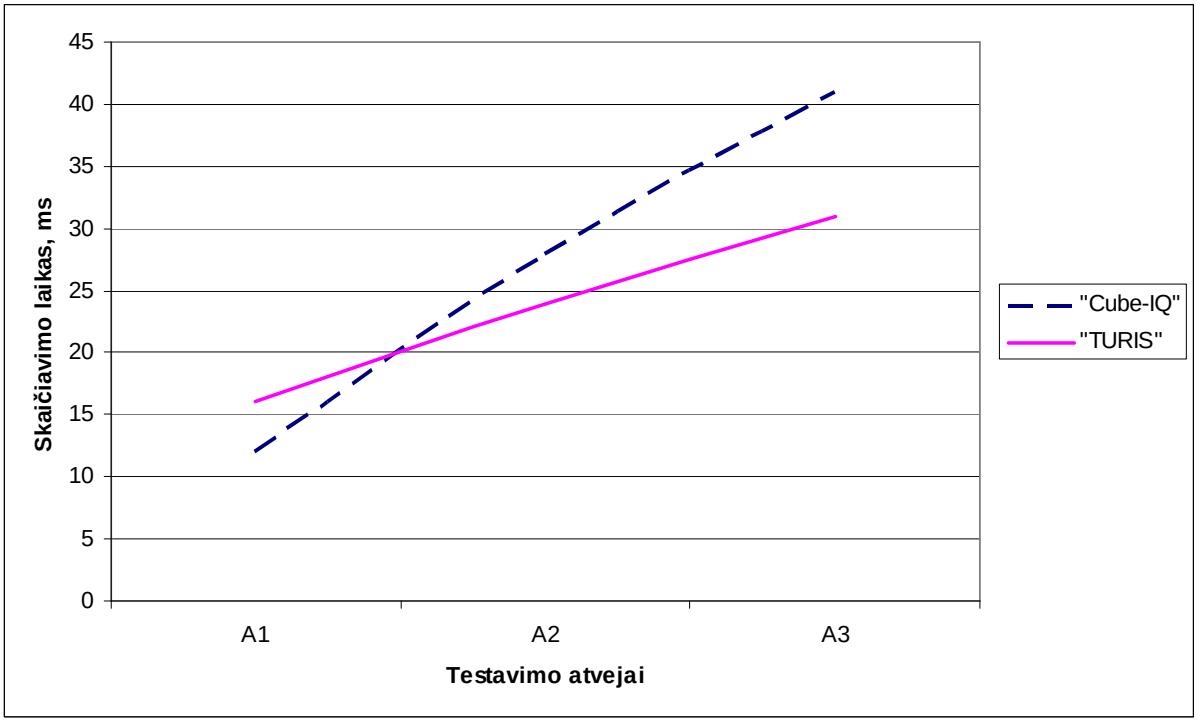
16 lentelėje pateikiami sistemų algoritmų skaičiavimų laikai – kiek laiko buvo atliekami skaičiavimai su kiekvienu testiniu atveju.

16 lentelė. Dėžių pakavimo trimatėje erdvėje sistemų algoritmų skaičiavimo greičiai

Programa Testinis atvejis	„CargoWiz“	„3D Load Packer”	“LoadPlanner”	“Cube-IQ”	“TURIS”
A1	3 s.	125 s.	62 s.	12 ms./iter.	16 ms.
A2	8 s.	–	379 s.	28 ms./iter.	24 ms.
A3	68 s	–	–	41 ms./iter.	31 ms.
R1	188 s.	–	–	943 ms./iter.	125 ms.
R2	280 s.	–	–	2,44 s./iter.	734 ms.
R3	318 s.	–	–	2,01 s./iter.	13 s.

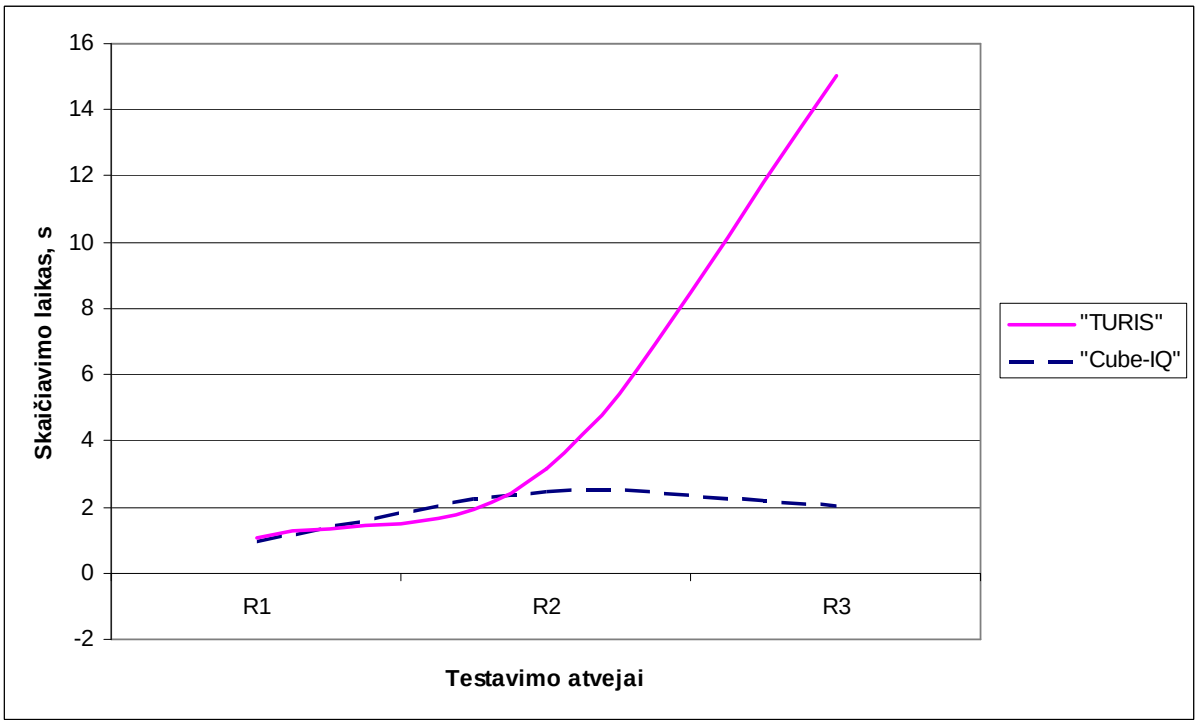
„3D Load Packer” ir “LoadPlanner” labai lėtai dirba su duomenimis, kuriuose daugiau nei 1 000 dėžių. „CargoWiz“ programa atliko skaičiavimus su visais testiniais atvejais, tačiau skaičiavimai, lyginant su „Cube-IQ“ ir „TURIS“, skiriasi 100 kartų.

38 paveiksle palyginamas sukurtos programos „TURIS“ algoritmo skaičiavimo laikas su pagrindinės konkurentės – sistemos „Cube-IQ“ algoritmo vykdymo laiku (testavimo atvejai A1, A2 ir A3).



39 pav. Dėžių pakavimo trimatėje erdvėje sistemų algoritmų skaičiavimo greičiai

39 paveiksle palyginami sistemų „TURIS“ ir „Cube-IQ“ algoritmų skaičiavimo laikai su testiniais atvejais R1, R2, R3.



40 pav. Dėžių pakavimo trimatėje erdvėje sistemų algoritmų skaičiavimo greičiai

5 Išvados

- Išanalizavus visas sistemas, pastebėtas vienas iš trūkumų būdingas visoms sistemoms: prastas prekių grupavimas. Labai svarbu grupuoti tos pačios rūšies prekes, kad sistemos vartotojui būtu lengviau krauti krovininį automobilį.
- Ištyrus visas sistemas nustatyta, kad per didelis taisyklių, skirtų optimizavimo algoritmui, įvedimas labai sulėtina algoritmo skaičiavimus ir sumažina optimizavimo efektyvumą. Kuriant taisykles būtina įvertinti kiek šios taisyklės yra aktualios ir kaip jos paveiks algoritmo greitį ir efektyvumą. Jei taisyklės yra labai svarbios, tačiau tik mažai vartotojų grupei, tai galbūt geresnis sprendimas būtų sukurti atskirą algoritmo modifikaciją, kad ji nepaveiktu likusių vartotojų grupių.
- Identifikavus pagrindines problemas susijusias su prekių pakavimu trimatėje erdvėje, atlikus jų analizę ir pasirinkus sprendimo būdą, sukurtas euristinis algoritmas, kuris iš tyrinėtų sistemų algoritmų greičiausiai atlieka skaičiavimus ir duoda gerus optimizavimo rezultatus. Tačiau konkuruojant su egzistuojančiais produktais reikėtu padidinti optimizavimo efektyvumą. Pavyzdžiui, „Cube-IQ“ naudojamas mišraus tipo algoritmas yra lėtesnis, tačiau truputį tikslesnis. Modifikavus sukurtą euristinį algoritmą į mišrų (genetinį-euristinį) galima tikėtis pagerinti algoritmo optimizavimo efektyvumą.
- Panaudojus sukurtą algoritmą sukurta informacinė sistema „TURIS“, kuri 20 000 dėžių į krovininį automobilį pakrauna greičiau nei per 5 minutes, o pakraunamų prekių tūris užima 88,9% - 100% krovininio automobilio tūrio. Sistemoje naudojamas algoritmas įvertina visus pagrindinius prekių krovimo apribojimus, kurie buvo aprašyti 2 skyriuje.
- Atlikus bandymus su sukurta sistema „TURIS“ ir egzistuojančiomis sistemomis, „TURIS“ užėmė antrą vietą, pirmoje vietoje liko „Cube-IQ“ sprendimas. Tyrimas parodė, kad „Cube-IQ“ sukurtas algoritmas duoda geresnius rezultatus, likusių sistemų algoritmai duoda blogesnius rezultatus. Tai yra labai geras rezultatas, įvertinant, kad „Cube-IQ“ pasirodė rinkoje tik šiais metais, o ją kūrusi komanda neabejotinai turėjo žymiai daugiau resursų, be to ja kūrusi kompanija „Magic Logic“ patenka į geriausių pasaulio logistikos IT kompanijų 100.
- Modifikuojant algoritmą, kad pagal gautus sprendinius būtų atliekami nauji skaičiavimai, kurių pagalba dėžės būtų labiau sugrupuojamos (genetinė sprendinio paieška), galima padidinti prekių grupavimą lygį. Taip palengvinant vartotojui prekių krovimą į krovininį automobilį.

- Ateityje algoritmą galima būtu modifikuoti įvedant sąlygą, kad prekės gali būti ne tik stačiakampės dėžės, bet ir cilindrai, birios medžiagos. Algoritmą galima modifikuoti, kad jis dūžtančias ir degias prekes krautu krovininio automobilio saugioje vietoje, kur jos nesusidaužytu. Logistikos problemų yra labai daug, todėl algoritmo tobulinimui ribų nėra.

Terminų ir santraukų žodynas

3D – trimatė erdvė.

API (*application programming interfaces*) – taikomųjų programų programavimo sąsaja.

DirectX – Microsoft Windows API sąsaja, leidžianti programų kūrėjams tiesiogiai kreiptis per kompiuterio periferinius įrenginius į žemo lygio funkcijas.

SDK (*Software Development Kit*) – taikomųjų programų programavimo įrankių paketas.

OpenGL – tai grafinis SDK paketas skirtas kurti 3D žaidimus.

3DLP (*3D Load Packer*) – 3D optimizavimo programinė įranga.

OS – operacinė sistema.

GLU (*OpenGL Utility Library*) - OpenGL paslaugų biblioteka.

GLX (*The OpenGL Extension to the X Window System*) – OpenGL praplėtimas X Window sistemai.

UI (*User Interface*) – vartotojo sąsaja.

DB – duomenų bazė.

Literatūra

1. A Hybrid Genetic Algorithm for Packing in 3D with Deepest Bottom Left with Fill Method [interaktyvus]. [Žiūrėta 2006 m. balandžio 5 d.]. Prieiga per internetą: <http://www.springerlink.com/index/AWHRDD1YU82EK38B.pdf>.

2. Approaches To 3D Free-Form Cutting And Packing Problems [interaktyvus]. [Žiūrėta 2006 m. balandžio 5 d.]. Prieiga per internetą: <http://citeseer.ist.psu.edu/context/992285/91537>.

3. Astrokettle Algorithms [interaktyvus]. [Žiūrėta 2006 m. Kovo 22 d.]. Prieiga per internetą: <http://www.astrokettle.com/>.

4. Comprehensive Load Planning And Optimization Solution For Advanced Logistics Management [interaktyvus]. [Žiūrėta 2006 m. kovo 27 d.]. Prieiga per internetą: <http://loadplanner.com/>.

5. CORNER Rich's. Direct3D vs. OpenGL: A Comparison [interaktyvus]. [Žiūrėta 2006 m. vasario 15 d.]. Prieiga per internetą: <http://roxen.xmission.com/~legalize/d3d-vs-opengl.html>.

6. Cube-IQ Load Optimizacion Software [interaktyvus]. [Žiūrėta 2006 m. balandžio 25 d.]. Prieiga per internetą: <http://www.magiclogic.com/cube-iq.cfm>.

7. David Pisinger's optimization codes [interaktyvus]. [Žiūrėta 2006 m. balandžio 5 d.]. Prieiga per internetą: <http://www.diku.dk/~pisinger/codes.html>.

8. *DirectX Technology Overview* [interaktyvus]. [Žiūrėta 2005 m. vasario 25 d.]. Prieiga per internetą: <http://www.microsoft.com/windows/directx/>.

9. Faroe O., Pisinger D., Zachariasen M. *Guided local search for the three-dimensional bin packing problem*. [interaktyvus]. 1999. [Žiūrėta 2006 m. kovo 25 d.] Prieigai per internetą: <ftp://v6.kobra.ktu.lt>

10. LIPCHAK Benj. *Overview of OpenGL* [interaktyvus]. [Žiūrėta 2006 m. vasario 25 d.]. Prieiga per internetą: <http://www.cs.wpi.edu/~matt/courses/cs563/talks/OpenGL_Presentation/OpenGL_Presentation.html>.

11. New Heuristics for Three-Dimensional Packing Problems [interaktyvus]. [Žiūrėta 2006 m. vasario 25 d.]. Prieiga per internetą: <http://www.crt.umontreal.ca/scrojopt2006/en/program/session.php?id=145>.

12. PACKER3D. *An optimal cargo packing algorithm* [interaktyvus]. [Žiūrėta 2006 m. vasario 25 d.]. Prieiga per internetą: <http://www.packer3d.com/modules/p3dcontent/?id=about_algorithm>.

13. Payload Capacity and Weight Distribution [interaktyvus]. [Žiūrėta 2006 m. vasario 20 d.]. Prieiga per internetą: <http://www.mitsubishi-fuso.com/en/technology/qanda/02_02.html?a2>.

14. Truck and Container Loading Software by softtruck [interaktyvus]. [Žiūrėta 2006 m. kovo 15 d.]. Prieiga per internetą: <http://www.softtruck.com/Container_Loading_Software_download.htm>.

15. TSpack: a unified Tabu Search code for Muti-Dimensional Bin Packing Problems [interaktyvus]. [Žiūrėta 2006 m. kovo 25 d.]. Prieiga per internetą: <http://www.or.deis.unibo.it/research_pages/ORcodes/TSpack.html>.

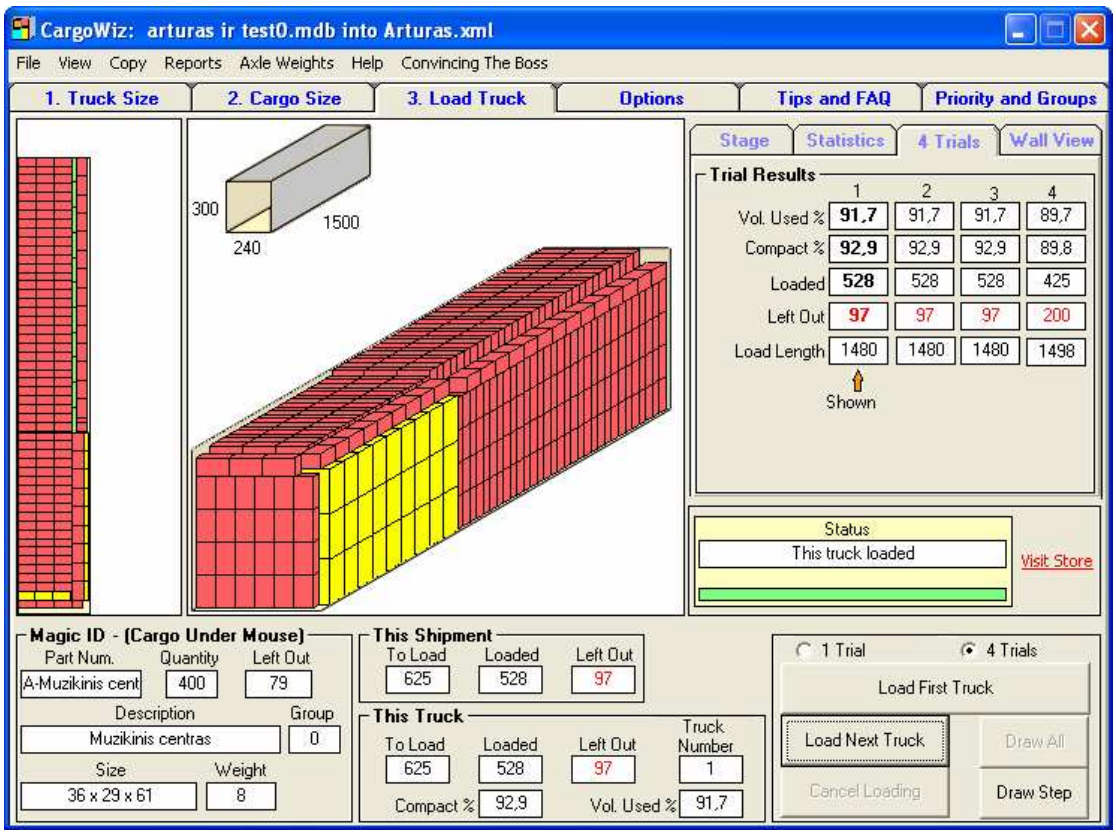
PRIEDAI

Priedas1. DirectX ir OpenGL aplinkų palyginimas

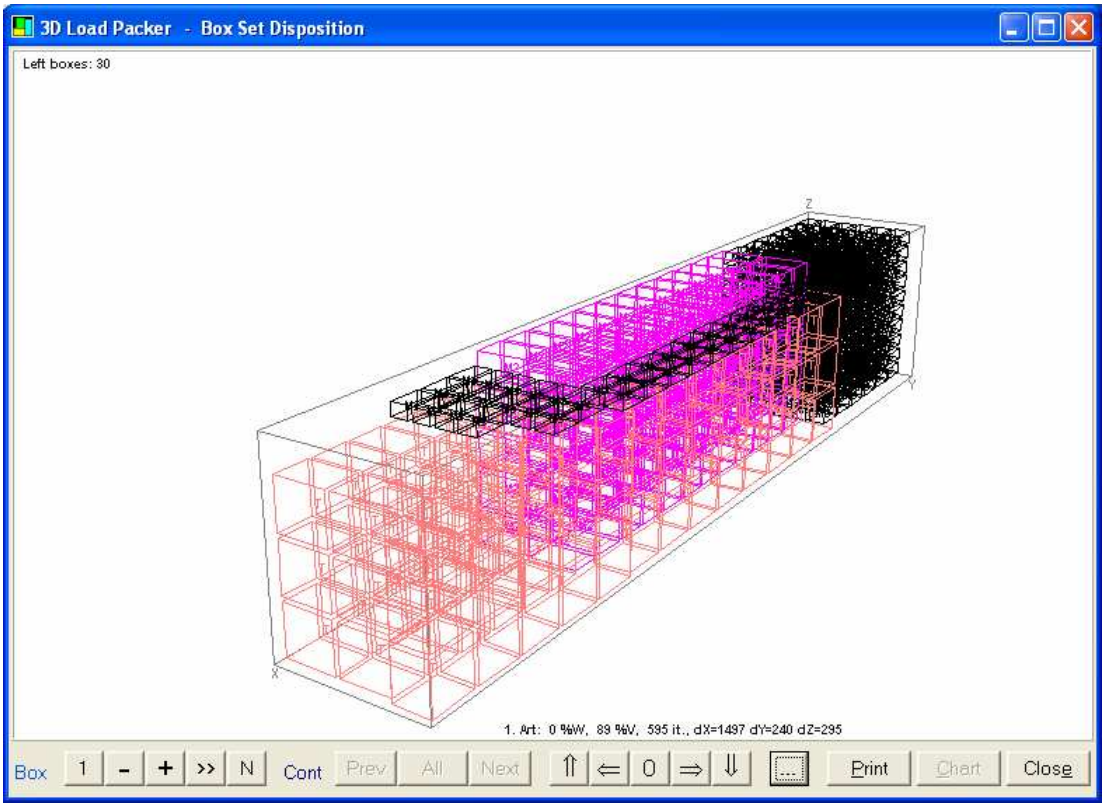
17 lentelė. DirectX ir OpenGL aplinkų palyginimas [5]

Naujovės	OpenGL 1.2	Direct3D 7	Direct3D 8
Systema			
Palaikoma operacinė sistema	Windows (9x, NT, 2000), MacOS, BeOS, *nix, kitos	Windows (9x, 2000, CE)	Windows (9x, 2000)
Modeliavimas			
Fixed-Function Vertex Blending	Ne	Taip	Taip
Programmable Vertex Blending	Ne	Ne	Taip
Parametric Curve Primitives	Taip	Ne	Taip
Parametric Surface Primitives	Taip	Ne	Taip
Hierarchical Display Lists	Taip	Ne	Ne
Vaizdo perpiešimas			
Two-sided Lighting	Taip	Ne	Ne
Point Size Rendering Attributes	Taip	Ne	Taip
Line Width Rendering Attributes	Taip	Ne	Ne
Programmable Pixel Shading	Ne	Ne	Taip
Triadic Texture Blending Operations	Ne	Ne	Taip
Cube Environment Mapping	Ne	Taip	Taip
Volume Textures	Taip	Ne	Taip
Multitexture Cascade	Ne	Taip	Taip
Texture Temporary Result Register	Ne	Ne	Taip
Mirror Texture Addressing	Ne	Taip	Taip
Texture "Wrapping"	Ne	Taip	Taip
Range-Based Fog	Ne	Taip	Taip
Bump Mapping	Ne	Taip	Taip
Modulate 2X Texture Blend	Ne	Taip	Taip
Modulate 4X Texture Blend	Ne	Taip	Taip
Add Signed Texture Blend	Ne	Taip	Taip
Kadrų buferis			
Hardware Independent Z Buffer Access	Taip	Ne	Ne
Full-Screen Antialiasing	Taip	Taip	Taip
Motion Blur	Taip	Ne	Taip
Depth of Field	Taip	Ne	Taip
Accumulation Buffers	Taip	Ne	Ne
Įvairialypiškumas			
Picking Support	Taip	Ne	Ne
Multiple Monitor Support	Ne	Taip	Taip
Stereo Rendering	Taip	Taip	Ne

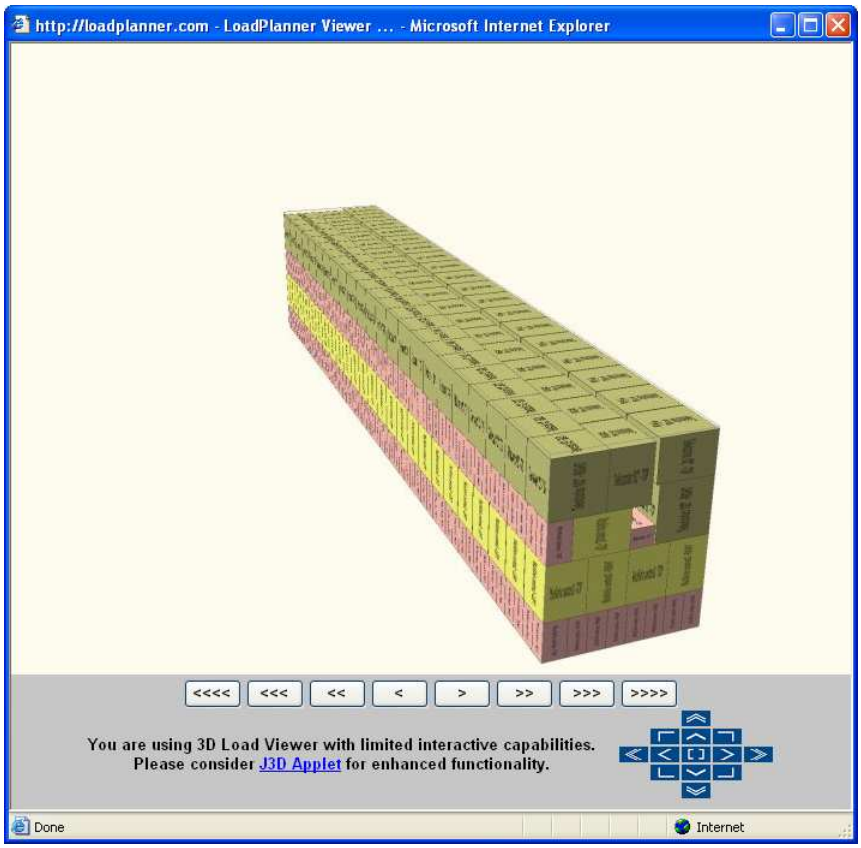
Priedas 2. Testavimo rezultatai su testiniu atveju A1



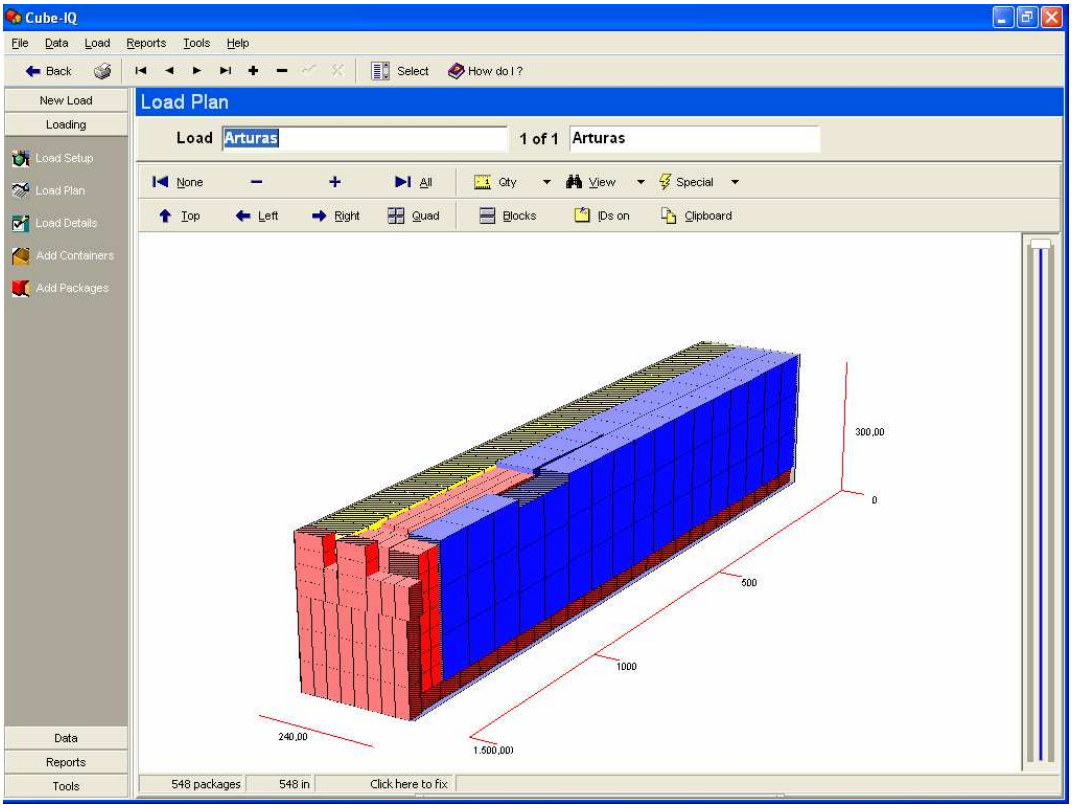
41 pav. „CargoWiz“ testavimo rezultatai su testiniu atveju A1



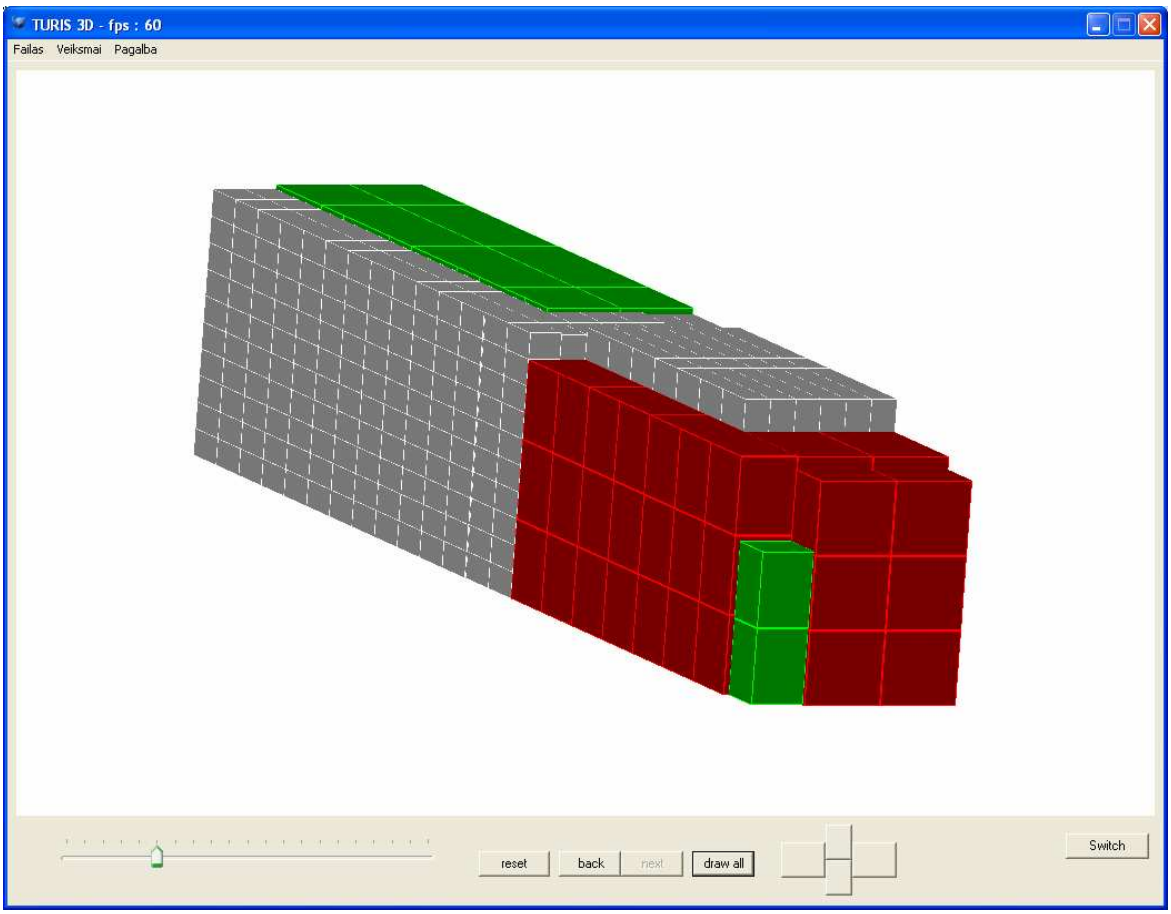
42 pav. „3D Load Packer“ testavimo rezultatai su testiniu atveju A1



43 pav. „LoadPlanner“ testavimo rezultatai su testiniu atveju A1

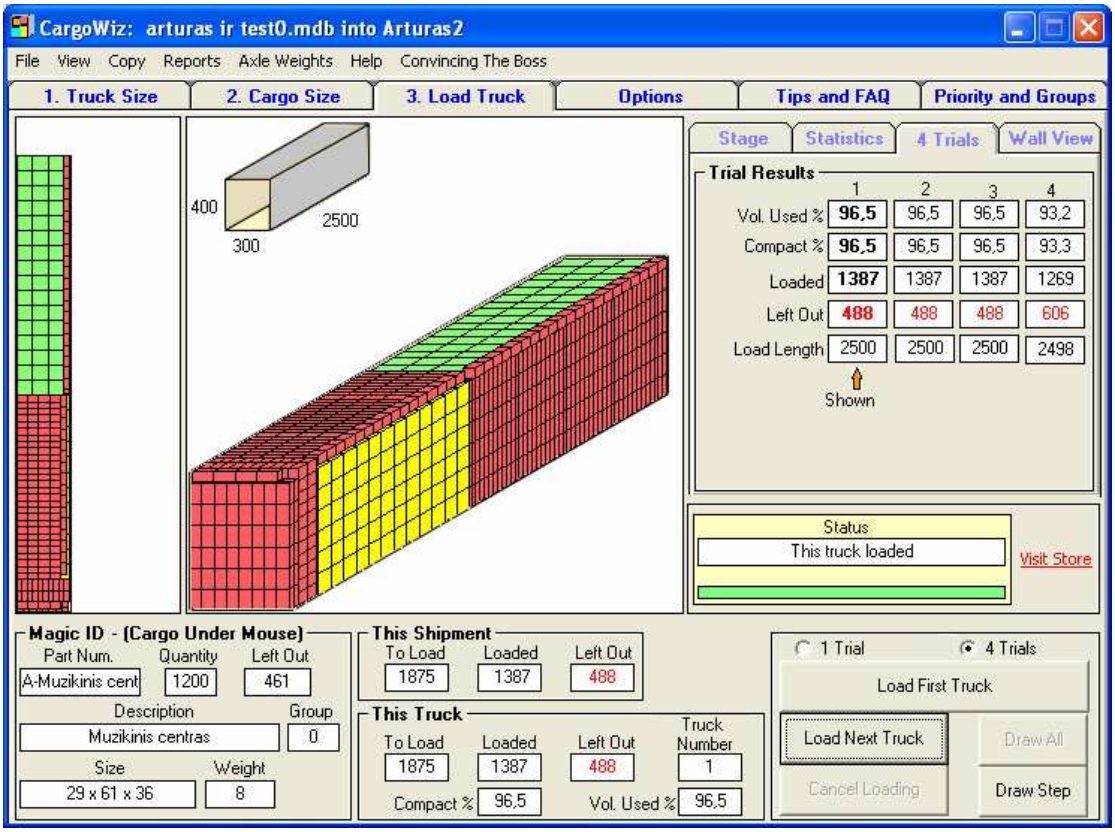


44 pav. „Cube-IQ“ testavimo rezultatai su testiniu atveju A1

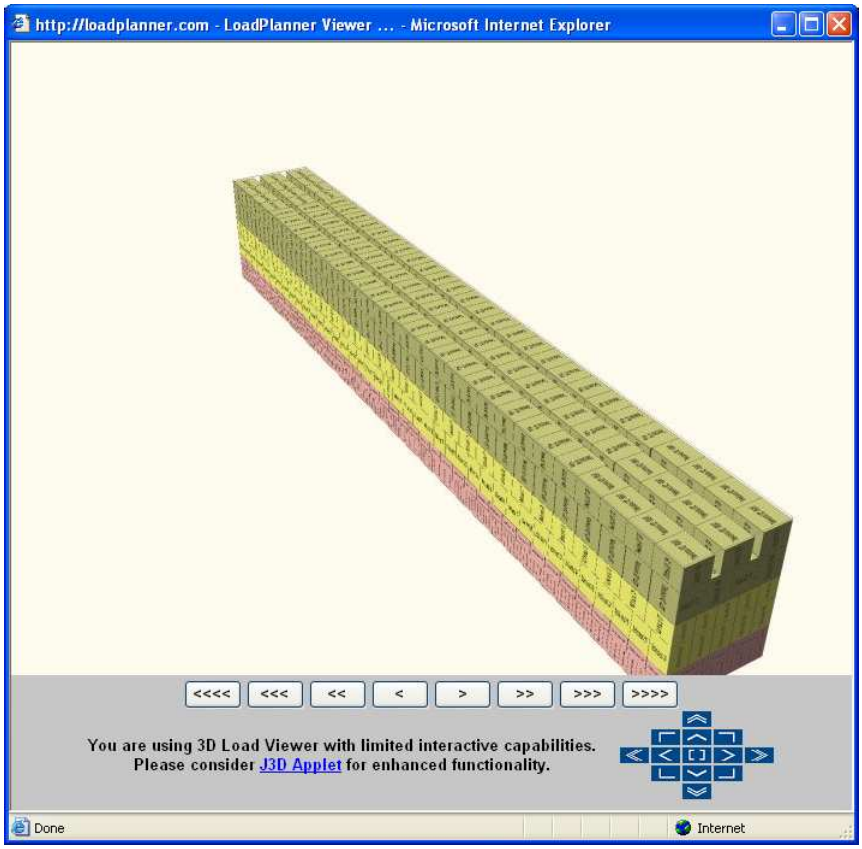


45 pav. „TURIS“ testavimo rezultatai su testiniu atveju A1

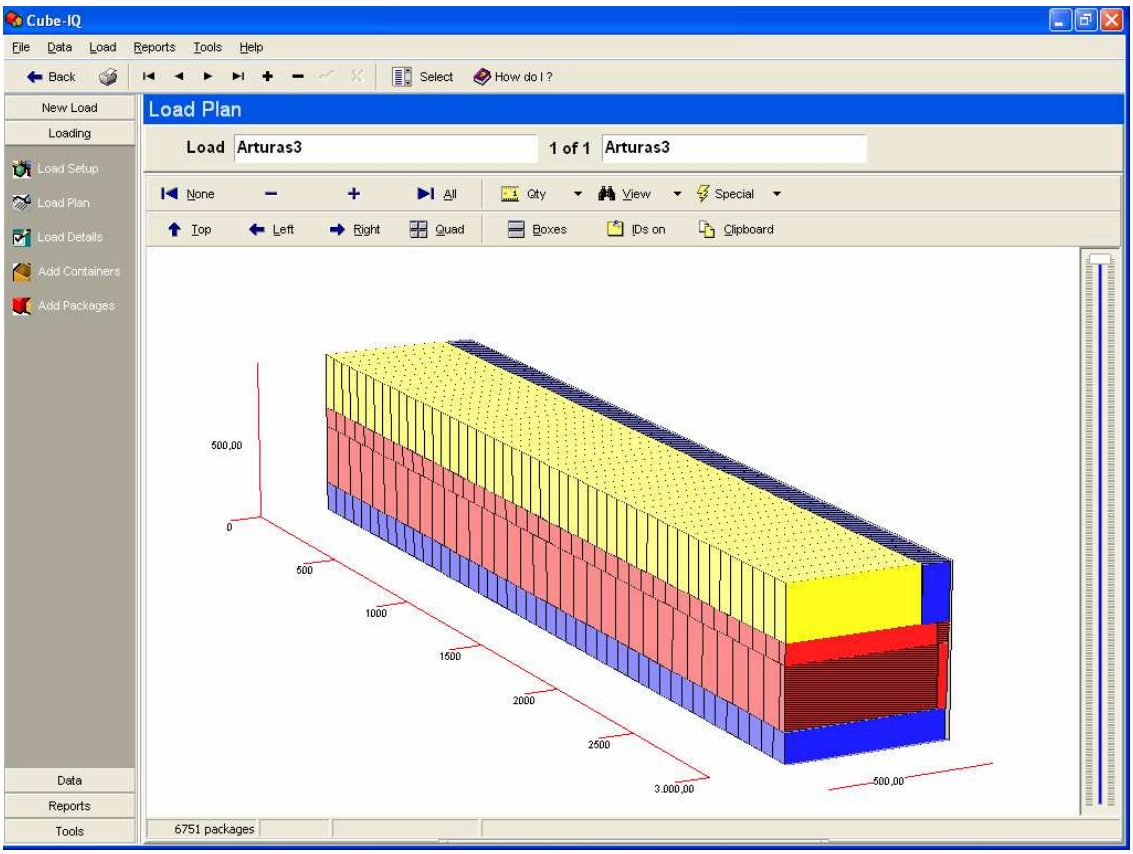
Priedas 3. Testavimo rezultatai su testiniu atveju A2



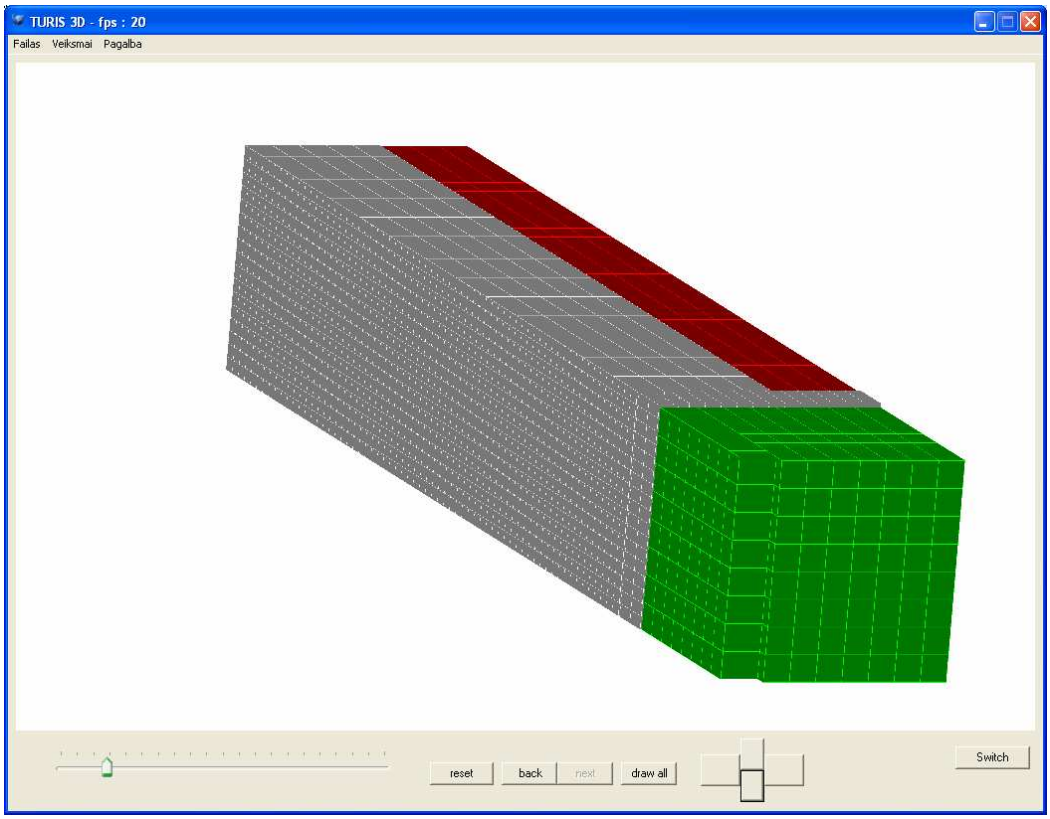
46 pav. „CargoWiz“ testavimo rezultatai su testiniu atveju A2



47 pav. „LoadPlanner“ testavimo rezultatai su testiniu atveju A2

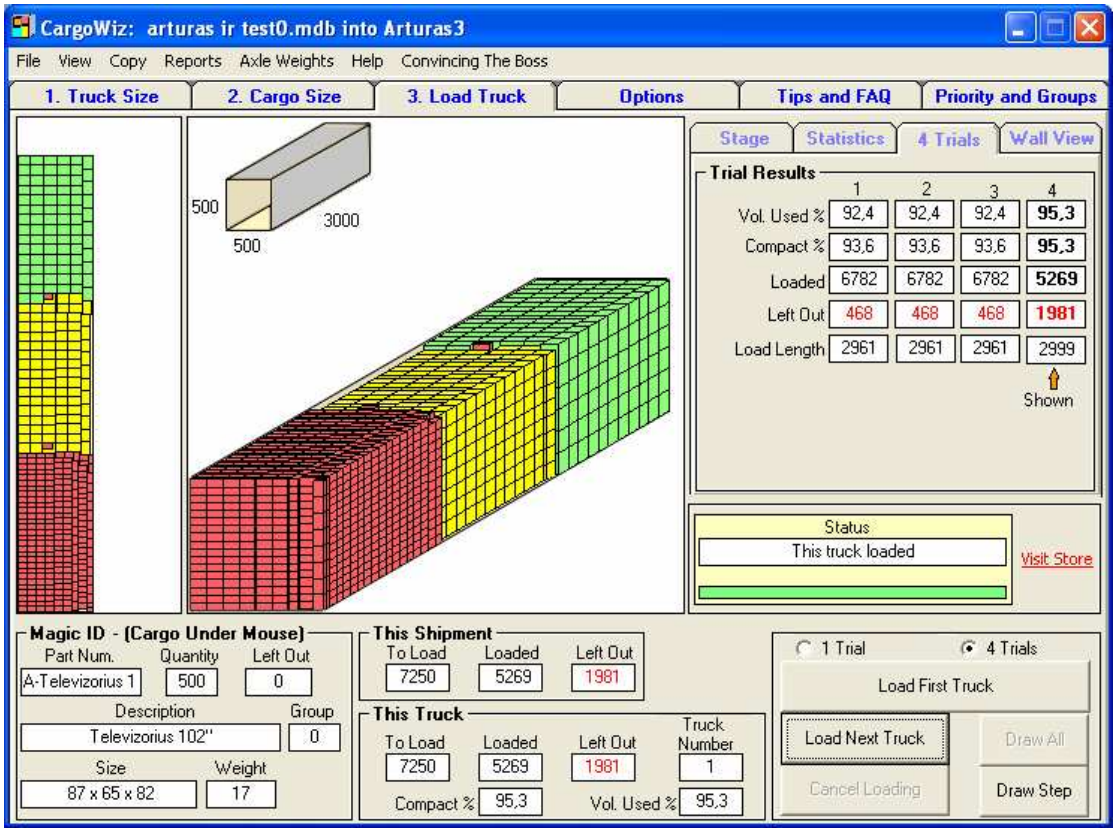


48 pav. „Cube-IQ“ testavimo rezultatai su testiniu atveju A2

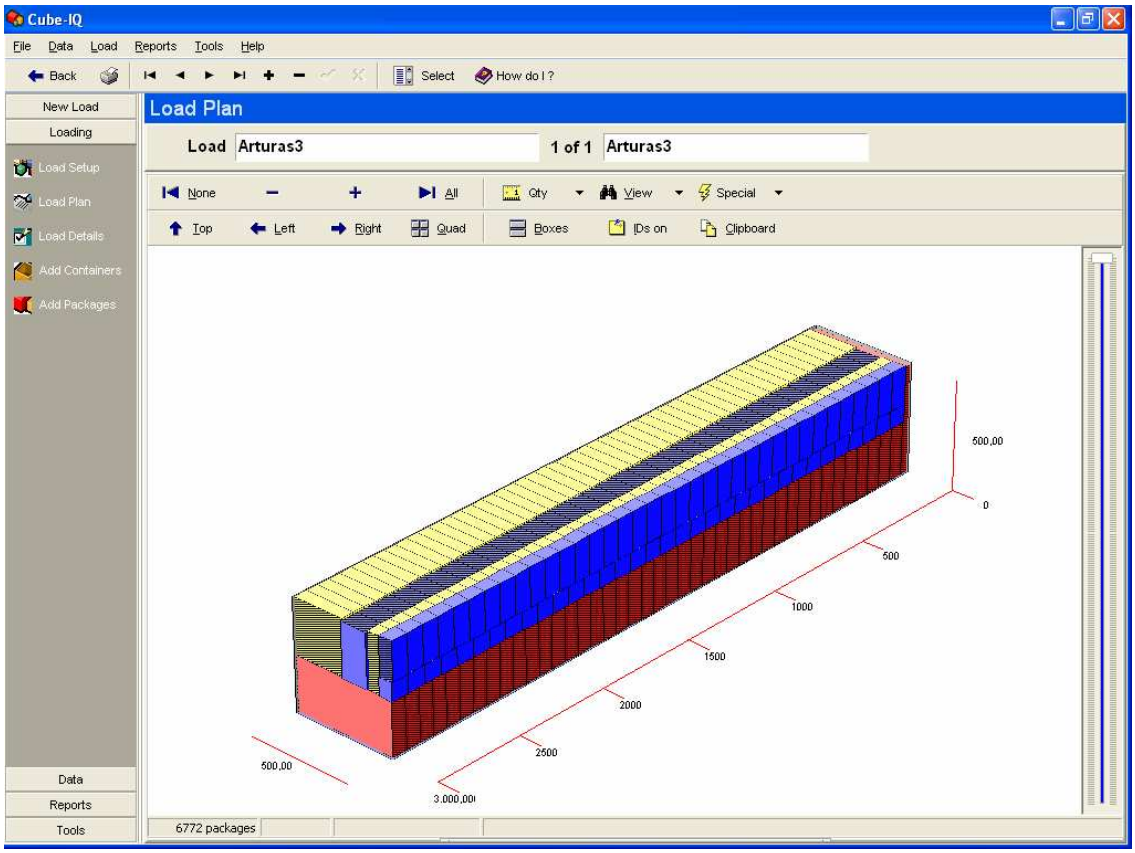


49 pav. „TURIS“ testavimo rezultatai su testiniu atveju A2

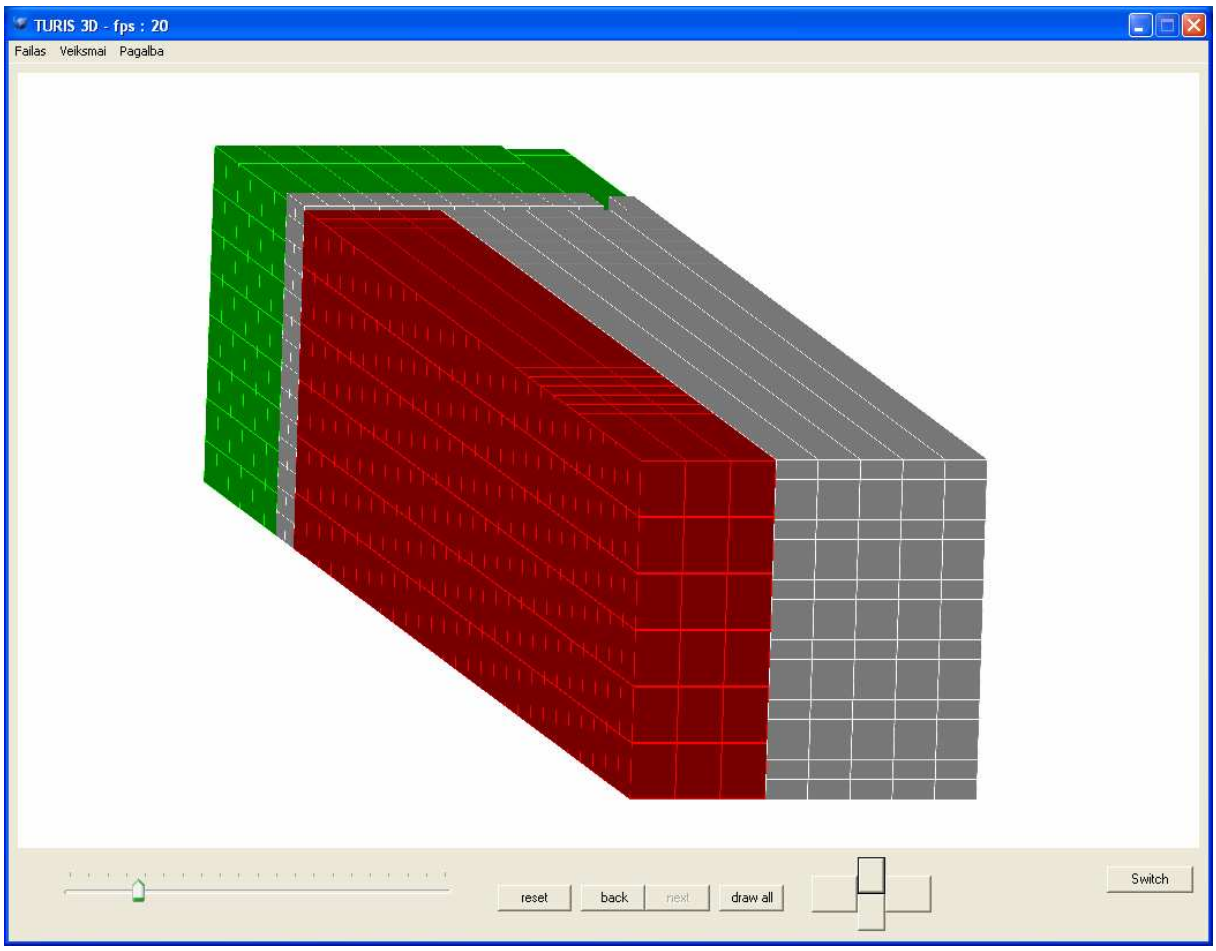
Priedas 5. Testavimo rezultatai su testiniu atveju A3



50 pav. „CargoWiz“ testavimo rezultatai su testiniu atveju A3

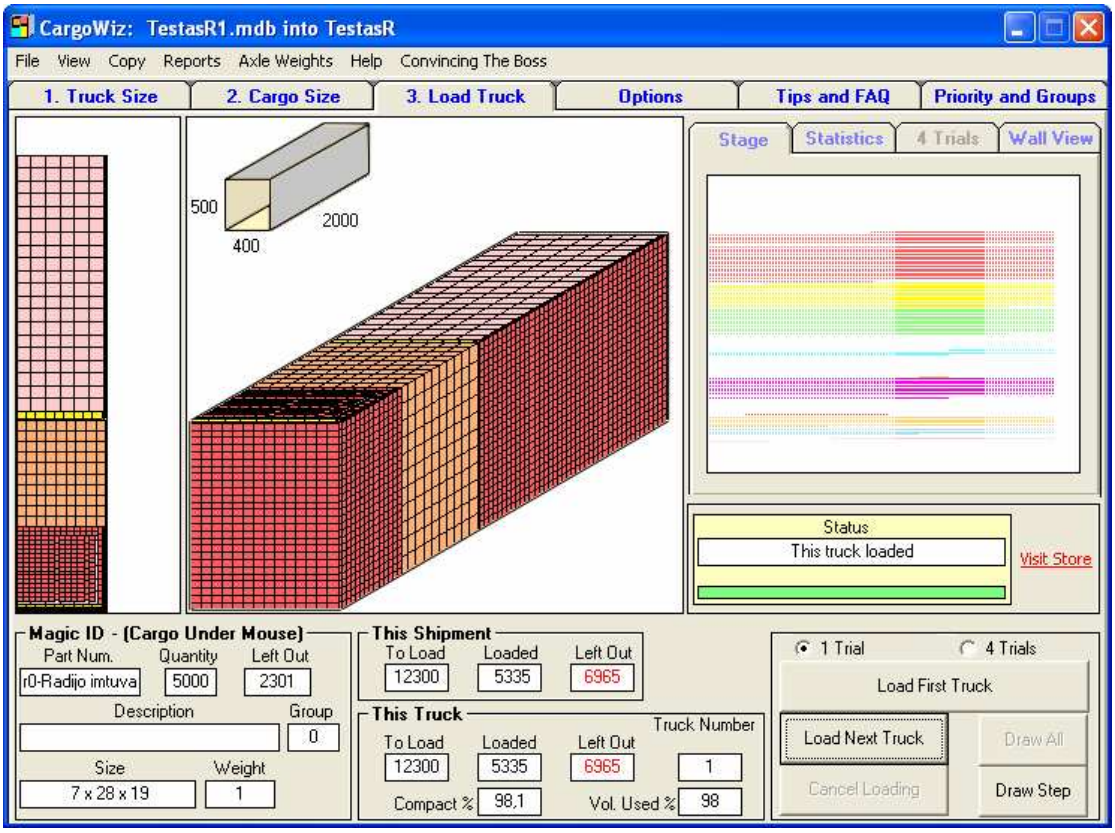


51 pav. „Cube-IQ“ testavimo rezultatai su testiniu atveju A3

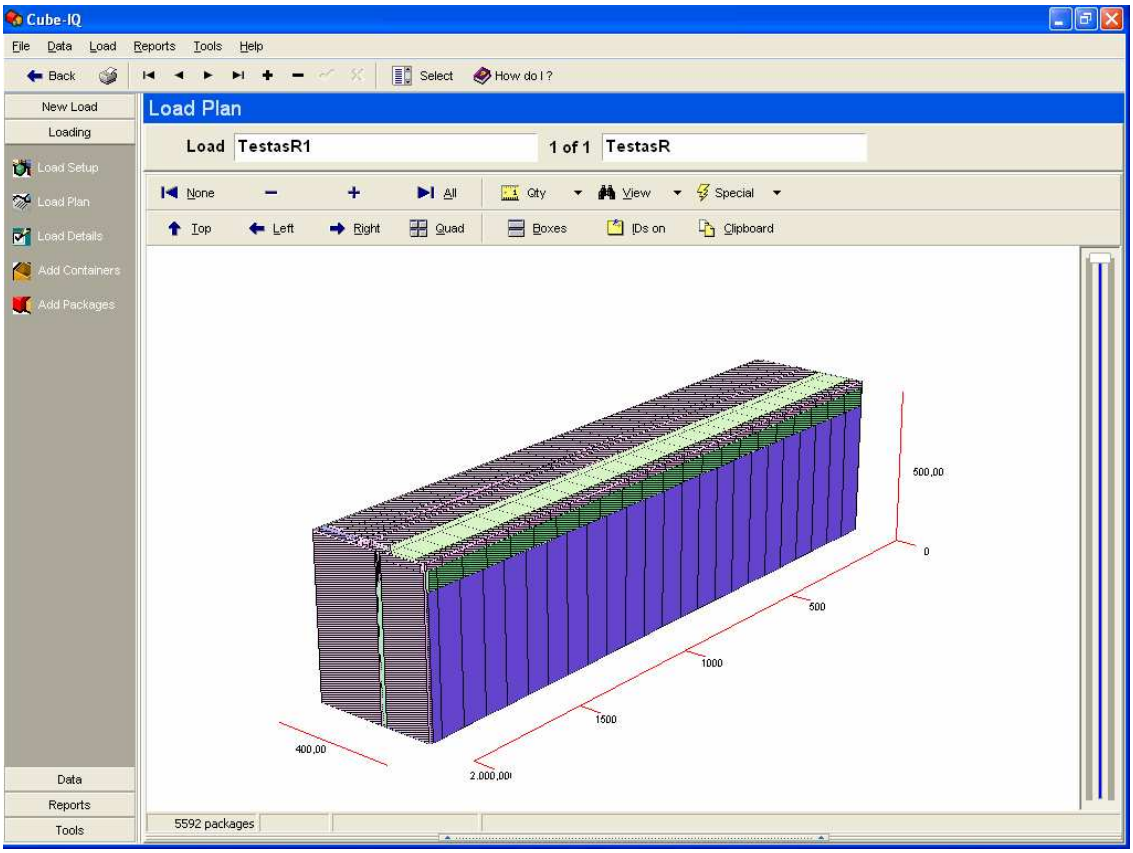


52 pav. „TURIS“ testavimo rezultatai su testiniu atveju A3

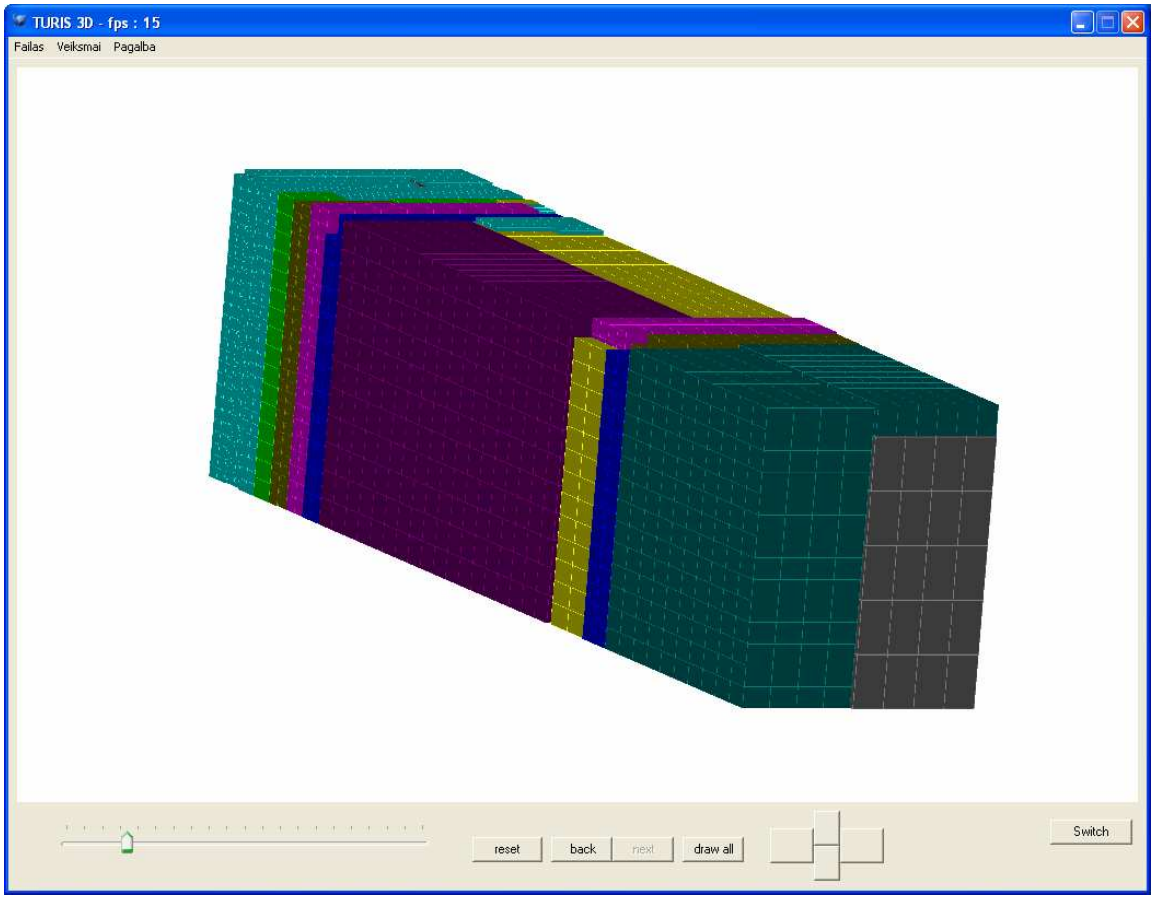
Priedas 6. Testavimo rezultatai su testiniu atveju R1



53 pav. „CargoWiz“ testavimo rezultatai su testiniu atveju R1

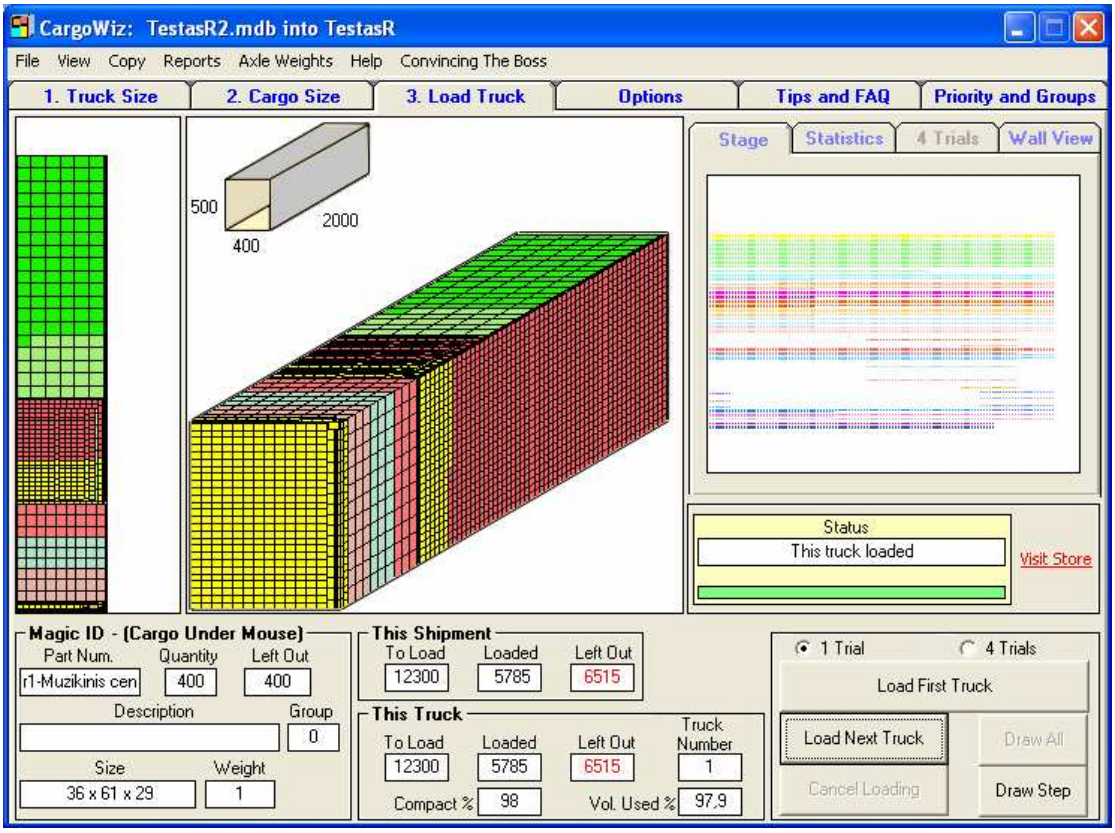


54 pav. „Cube-IQ“ testavimo rezultatai su testiniu atveju R1

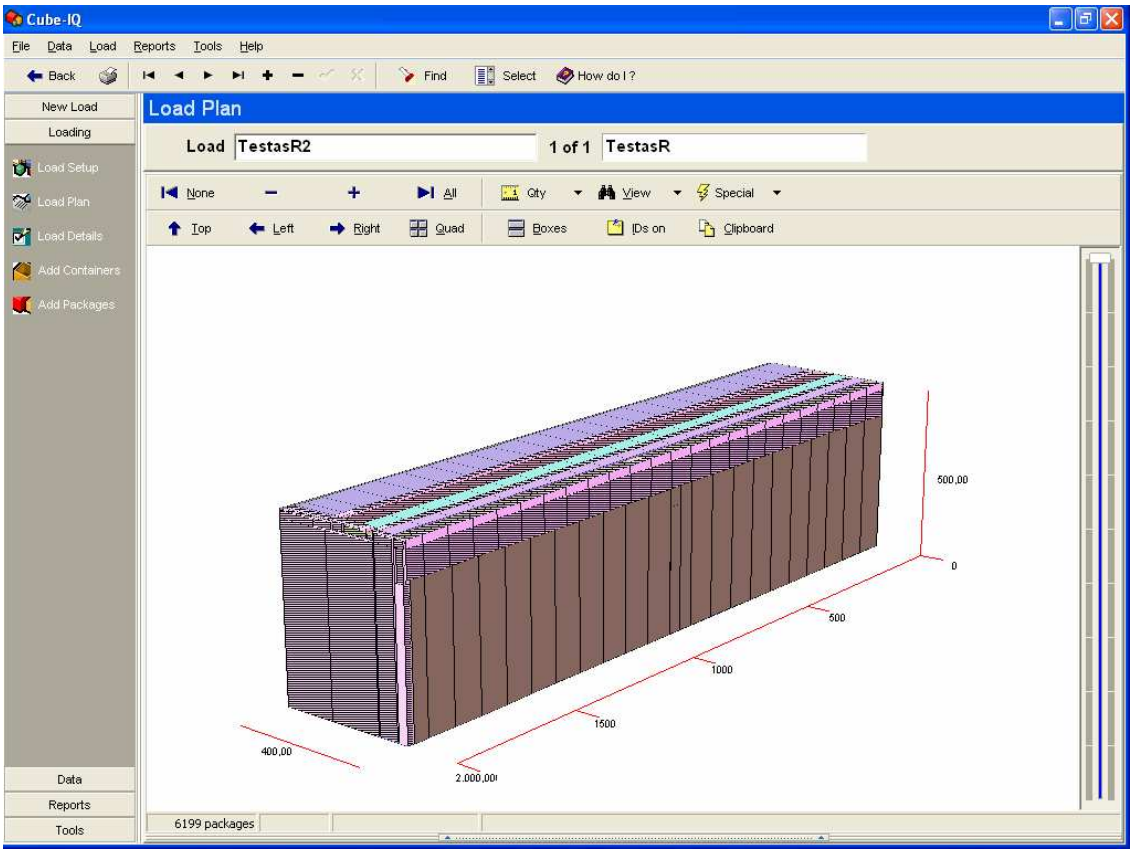


55 pav. „TURIS“ testavimo rezultatai su testiniu atveju R1

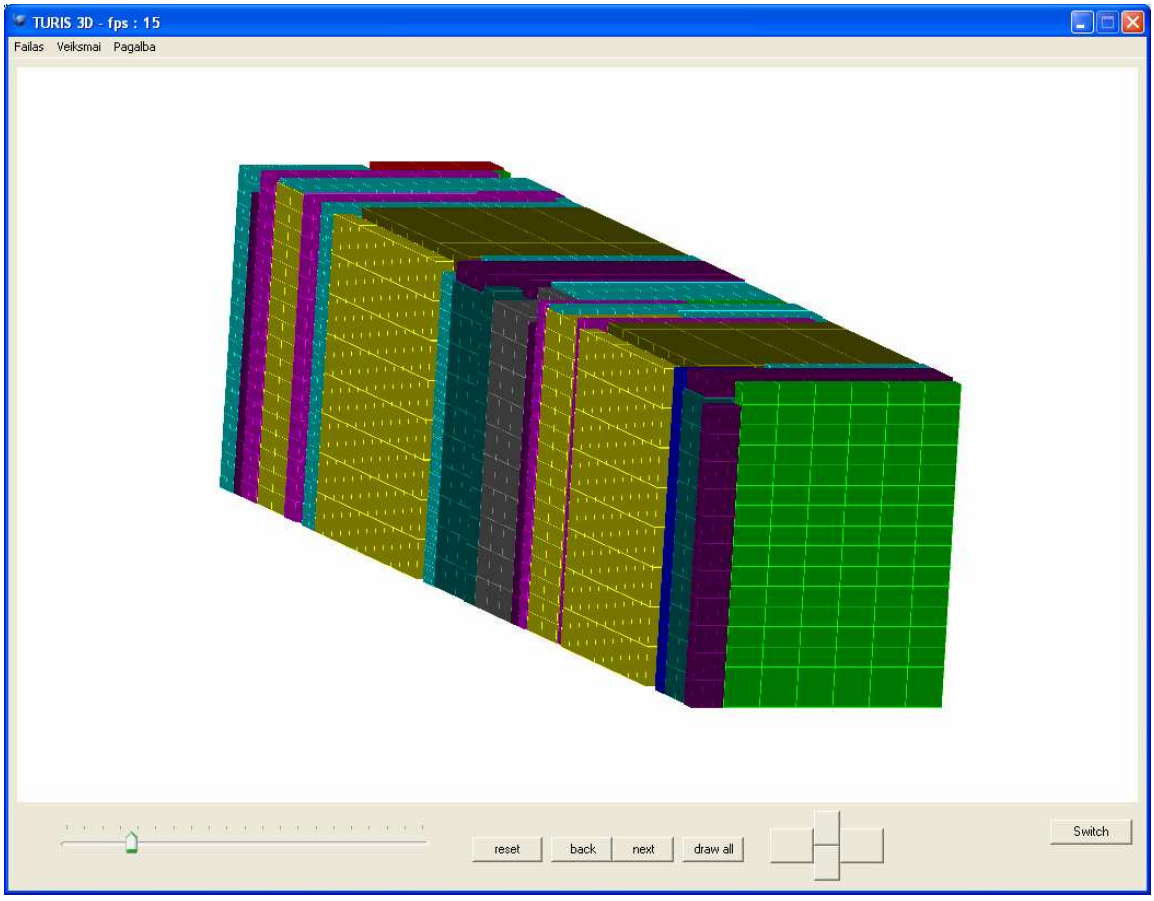
Priedas 7. Testavimo rezultatai su testiniu atveju R2



56 pav. „CargoWiz“ testavimo rezultatai su testiniu atveju R2

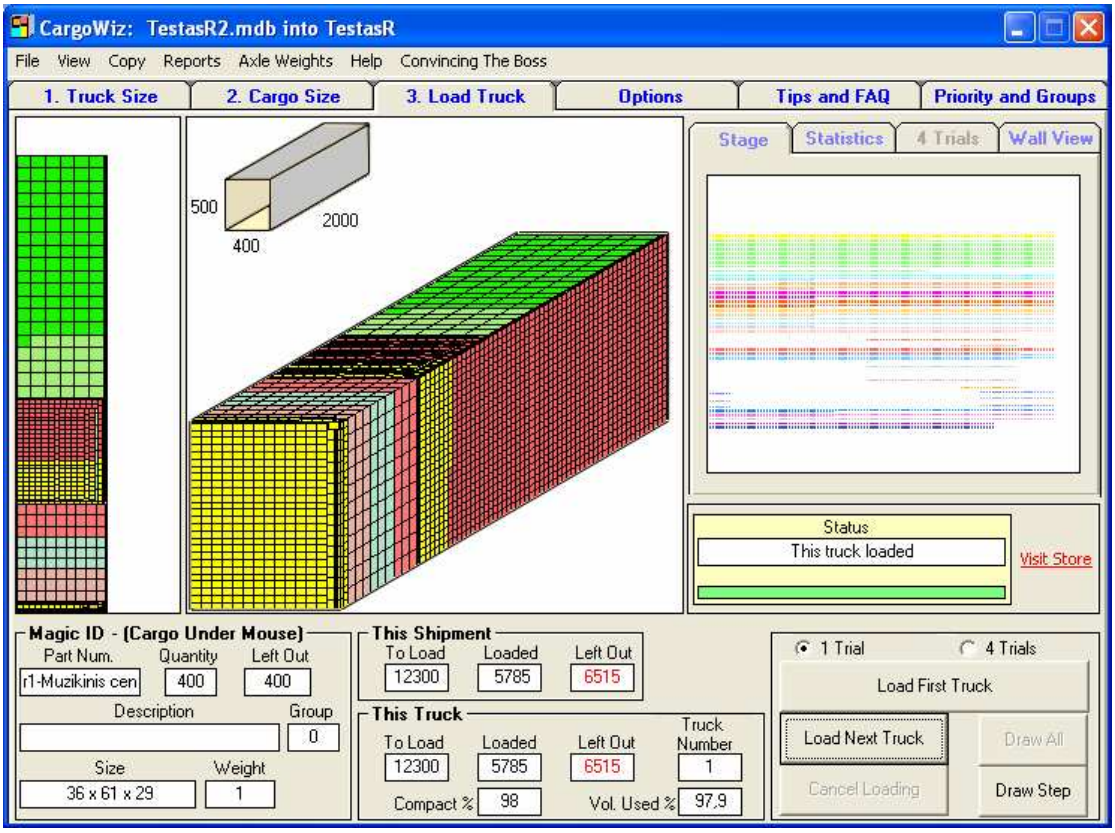


57 pav. „Cube-IQ“ testavimo rezultatai su testiniu atveju R2

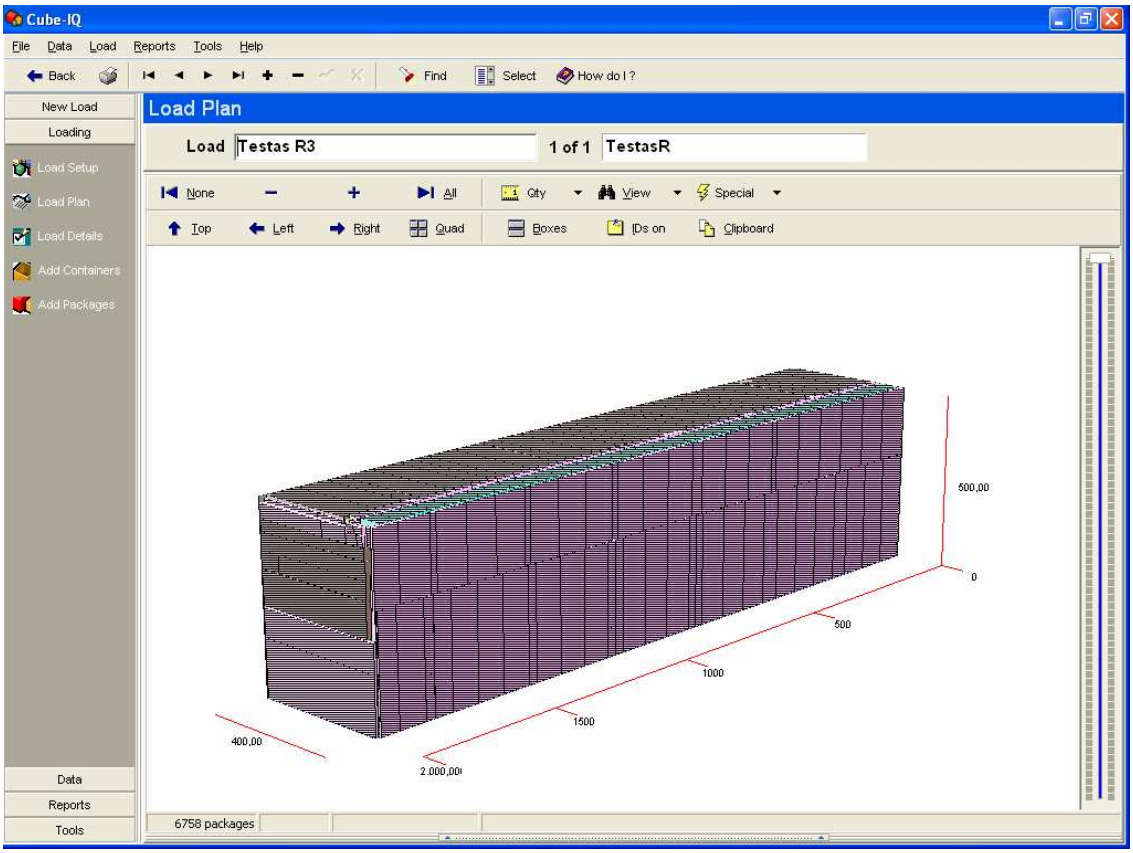


58 pav. „TURIS“ testavimo rezultatai su testiniu atveju R2

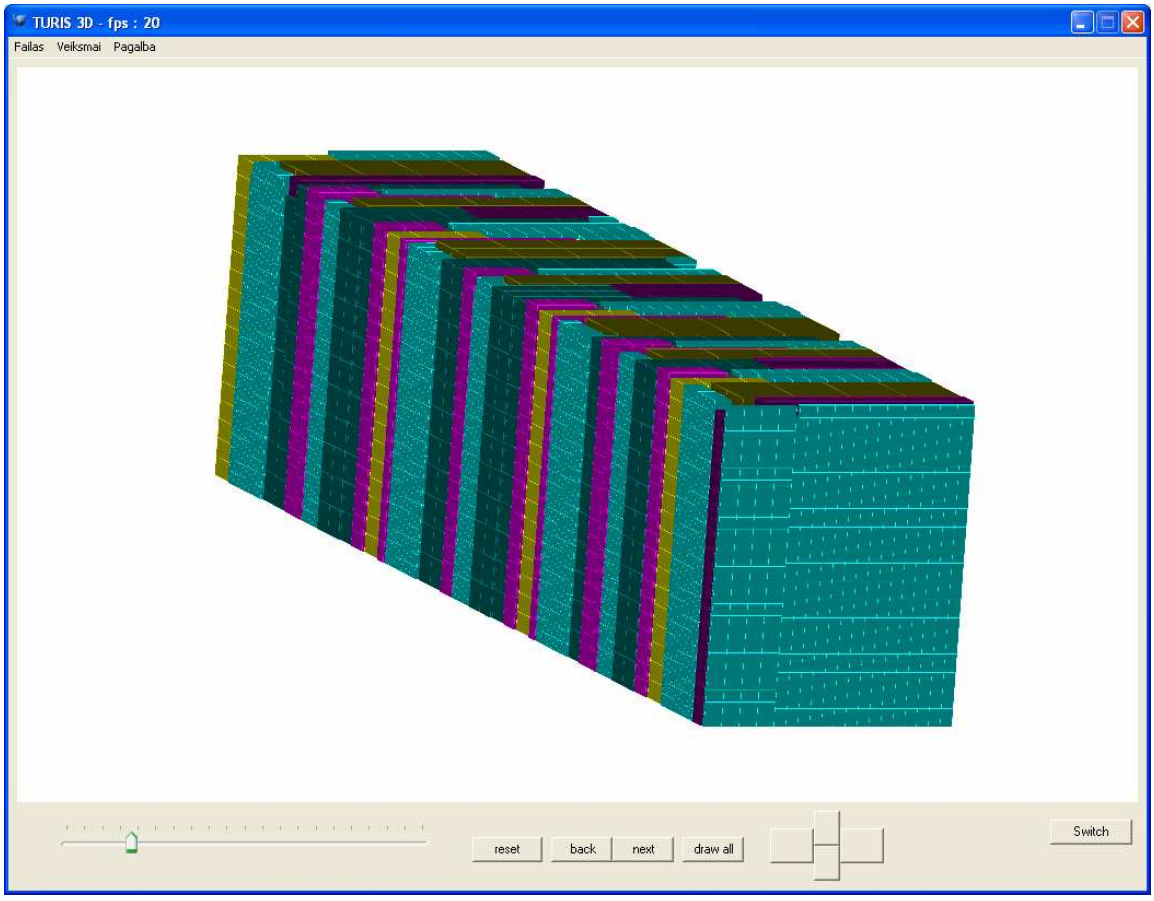
Priedas 8. Testavimo rezultatai su testiniu atveju R3



59 pav. „CargoWiz“ testavimo rezultatai su testiniu atveju R3



60 pav. „Cube-IQ“ testavimo rezultatai su testiniu atveju R3



61 pav. „TURIS“ testavimo rezultatai su testiniu atveju R3