Määrittelydokumentti

Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit (alkukesä)

Sami Korhonen
014021868
sami.korhonen@helsinki.fi

Tietojenkäsittelytieteen laitos
Helsingin yliopisto
2. kesäkuuta 2014

Työn aihe

Ongelma ja tavoite

Rahtifirma NopsaToimitus haluaa optimoida konttikuljetuksissa käytettävän tilan. Oletettavasti kontin tila on käytetty mahdollisimman tehokkaasti kun kontin lattian pinta-alaa on käytetty mahdollisimman vähän ja kontin, tai täytetyn lattia-alueen tilavuudessa on mahdollisimman vähän tyhjää.

Tavoitteena on kehittää mahdollisimman nopea ja pätevä algoritmi, joka selvittää sopivan tavan pakata annetut laatikot annettuun konttiin. Ongelmaan ei ole toistaiseksi löydetty ratkaisua, joka antaisi parhaan mahdollisen ratkaisun kaikissa tilanteissa siedettävässä ajassa, eikä sitä luultavasti tule löytymään tämän harjoitustyön tuloksenakaan. Ongelman luonteesta johtuen algoritmin tarkoitus on löytää ongelmaan sopiva ratkaisu tietyllä tarkkuudella sopivassa ajassa. Tämä tarkkuus voidaan määritellä tyhjän tilan avulla.

Määritelmiä

Yksiköt ja koordinaatisto

Työssä käytetään yksikköinä kokonaisluvuiksi pyöristettyjä senttimetrejä, sillä todellisuudessa laatikoiden mitat eivät ole esimerkiksi puristuvuuden vuoksi tämän tarkempia.

Koska kontti ja laatikot ovat suorakulmaisia särmiöitä, voidaan käyttä kolmiulotteista karteesista koordinaatistoa. Tässä työssä koordinaatiston akseleita merkitään kirjaimin x, y ja z.

Laatikot

Laatikoiden asettamiseen liittyy muutamia sääntöjä:

1. Laatikon tulee olla kokonaan kontin sisäpuolella
2. Laatikot eivät saa olla limittäin toistensa kanssa

3. Laatikon tulee olla kontin seinien suuntaisesti
4. Laatikko tulee olla tuettu koko pohjaltaan

Pakkaaminen

Toistaiseksi työssä ei kiinnitetä huomiota seuraaviin seikkoihin:

1. Samaa tavaraa sisältävät laatikot tulisi asettaa vierekkäin
2. Rahdin tulisi olla mahdollisimman helppo pakata ja purkaa

Ratkaisu

Tietueet ja tietorakenteet

Palkki

Palkki koostuu yhdestä tai useammasta laatikosta ja se on muodoltaan suorakulmainen särmiö. Kaikista jäljellä olevista laatikoista koostuvat palkit kasataan listaan. Tätä varten tarvittaessa toteuttaa sopiva tietorakenne.

Pakkaussuunnitelma

Pakkaussuunnitelma on lista asetetuista palkeista. Jokaista setettua pakettia varten talletetaan koordinaatiston piste, johon palkin referenssikulma asetetaan. Listauksen suunnittelu on vielä kehittämättä.

Tilapalkki

Kontissa vapaana oleva eli tyhjä tila koostuu suorakulmaisista särmiöistä, joita kutsun tässä työssä tilapalkeiksi.

Tila

Tämä pitää tietoa pakkauksen tämänhetkisestä tilasta esimerkiksi sisältäen pakkaussuunnitelman, käytetyn tilavuuden, listan vapaista laatikoista sekä pinon vapaista tilapalkeista.

Algoritmit ja funktiot

Pääalgoritmi

Kontin pakkaaminen perustuu seuraavaan algoritmiin:

```
function pakkaaKontti(kontti, laatikot)
    paras = new PakkausSuunnitelma
    haunVaativuus = 1 // määrittelee haun tarkkuuden iteraatiossa
    // tehdään uusia pakkaussuunnitelmia kunnes aikaraja umpeutuu
    while (aika < aikaraja)
        tila = Tila.initialisoi() // aloitetaan tyhjästä kontista
        // tilapalkkiPino:ssa on tilapalkit, eli kontin vapaana olevat tilat
        while (tila.tilapalkkiPino != empty)
             tilapalkki = tila.tilapalkkiPino.pop()
            palkki = haeParasPalkki(tila, tilapalkki, haunVaativuus)
            lisaaPalkki(tila, palkki)
        endwhile
        if (parempi(tila.pakkausSuunnitelma,paras))
            paras = tila.pakkausSuunnitelma
        // update search effort for next iteration
        haunVaativuus *= 2
    endwhile
return paras
```

Tässä generoidaan useita, toinen toistaan parempia pakkaussuunnitelmia, kunnes aikaraja umpeutuu. Kutsun nyt iteraatioksi yhden pakkaussuunnitelman luomista. HaunVaativuus on parametri, jolla määritetään kuinka hyvin soveltuva palkki halutaan löytää tiettyyn vapaaseen tilaan. Tätä kasvatetaan iteraatioiden välillä, jolloin saadaan aina entistä parempi pakkaussuunnitelma. Iteraation aluksi initialisoidaan tila, jonka tilapalkkiPino sisältää koko kontin kokoisen vapaan tilapalkin. Tämä tilapalkki jaetaan palkkeja lisätessä pienemmiksi tilapalkeiksi, joita tulee aina kolme kappaletta lisää kun asetetaan uusi palkki pakkaussuunnitelmaan. Mikäli tilapalkkiin löytyy siihen mahtuva palkki, se lisätään pakkaussuunnitelmaan.

haeParasPalkki(tila, tilapalkki, haunTarkkuus)

Tämä funktio hakee haunTarkkuuden tarkkuudella sopivan laatikon tilapalkkiin Kun haunTarkkuus on pieni, tulee palkin löytyä hyvin nopeasti. Toteutan tämän haun siksi hakupuuna, josta erään version ovat tehneet Fanslau Tobias ja Bortfeldt Andreas [1].

lisaaPalkki(tila, tilapalkki, palkki)

Mikäli palkki on null, ei ole löytynyt tilaan sopivaa palkkia, jolloin tilapalkki poistetaan. Muutoin palkki lisätään pakkaussuunnitelmaan, tilapalkki poistetaan, ja tilapalkkiPinoon asetetaan kolme uutta tilapalkkia.

Syötteet ja käyttäminen

Syötteet

Ohjelmalle syötetään kontin mitat alustavasti komentorivipohjaisella käyttöliittymällä, mutta ohjelmaan voidaan toteuttaa myöhemmin graafinen käyttöliittymä. Laatikoiden koot ja lukumäärät voidaan syöttää käyttöliittymän kautta tai lukemalla tiedot tiedostosta.

Käyttäminen

Ohjelman käyttö tapahtuu yksinkertaisen komentorivipohjaisen tai graafisen käyttöliittymän avulla. Lisäksi ohjelma esittää mahdollisesti tuotetun lastausjärjestelmän kuvina ja/tai kenties 3D-graafisesti.

Aika- ja tilavaativuudet

Aikavaativuus

Algoritmin käyttämä aika riippuu hyvin pitkälti sille määritellystä aikarajasta, eikä konttiin pakattavien laatikoiden määrästä. Ilman aikarajaa algoritmi

laajentaisi hakuaan kunnes jokin muistialue käy liian pieneksi. Koska ongelmaan ei löydetä varsinaisesti oikeaa ratkaisua, on jokseenkin mielivaltaista tutkia aikavaativuutta perinteisin tavoin. Enkä näe tavotteellisten aikavaativuuksien määrittelemistä mielekkäänä. Aikavaativuuksia voisi tutkia sen mukaan, kuinka kauan on tarvittu aikaa tietyllä tarkkuudella optimaalisen pakkaussuunnitelman löytymiseen tietyllä laatikkomäärällä.

Viitteet

[1] Tobias Fanslau, Andreas Bortfeldt, 2008
A Tree Search Algorithm for Solving the Container Loading Problem
http://www.fernuni-hagen.de/wirtschaftswissenschaft/download/beitraege/db426.pdf
viitattu 18. toukokuuta 2014