Aalto University
School of Science
and Technology

Trimmitysongelman LP-relaksaation ratkaiseminen sarakkeita generoivalla algoritmilla ja brute-force-menetelmällä (aihe-esittely)

Vesa Husgafvel
17.09.2012

Ohjaaja: DI Mirko Ruokokoski
Valvoja: Prof. Harri Ehtamo

Tausta

- Trimmityksellä tarkoitetaan (tässä yhteydessä) kustannustehokasta tuotannonsuunnittelua, jossa tyydytetään kysyntä minimoimalla käytetyn raaka-aineen määrä.
- Esimerkki: miten sahata vaneria, niin että syntyvien hukkapalojen määrä olisi mahdollisimman pieni.
- Ongelma voidaan mallintaa lineaarisena kokonaislukutehtävänä.
- Vanerinsahaus on esimerkki 2D-tehtävästä.
- Tässä työssä tarkasteltava tehtävä on yksiulotteinen.

Trimmitysongelma (1/3)

- Paperitehdas valmistaa emorullaa, jonka pituus on $W \in \mathbb{Z}_+$.
- Asiakasyritys i, $i = 1, \ldots, m$, haluaa ostaa b_i kappaletta rullia, joiden pituus on $w_i \leq W$, $w_i \in \mathbb{Z}_+$.
- Pienempiä rullia saadaan viipaloimalla emorullia erilaisilla leikkausmuoteilla.
- Kutakin leikkausmuottia $j = 1, \ldots, n$, vastaa sarakevektori $\mathbf{A}_j$, siten että sarakkeen alkio $a_{ij} \in \mathbb{Z}_+$ ilmaisee, kuinka monta kertaa rulla w_i esiintyy kyseisessä leikkausmuotissa.
- Luonnollisesti leikkausmuotissa olevien pienten rullien yhteispituus ei saa ylittää emorullan pituutta, mistä saadaan rajoitusehdot

$$\sum_{i=1}^{m} a_{ij} w_i \leq W, \qquad j = 1, 2, \ldots, n.$$

Trimmitysongelma (2/3)

- Kun paperitehdas haluaa minimoida valmistettujen emorullien lukumäärän tyydyttäen samalla asiakasyritysten kysynnät, saadaan tehtävän formulaatioksi

$$
\begin{aligned}
Z_{IP} \quad = \quad & \min \quad && \sum_{j=1}^{n} x_j \\
& \text{s.e.} && \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \qquad i = 1, 2, \ldots, m, \\
& && x_j \in \mathbb{Z}_+, \qquad\quad j = 1, 2, \ldots, n,
\end{aligned}
$$

missä x_j on leikkausmuotin j mukaan leikattujen emorullien lukumäärä.

Trimmitysongelma (3/3)

- Yksinkertaisuuden vuoksi siirrytään tarkastelemaan tehtävän LP-relaksaatiota

$$
\begin{aligned}
Z_{LP} \quad = \quad & \min \quad && \sum_{j=1}^{n} x_j \\
& \text{s.e.} && \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \qquad i = 1, 2, \ldots, m, \\
& && x_j \geq 0, \qquad\qquad j = 1, 2, \ldots, n.
\end{aligned}
$$

- Ongelmana kuitenkin on, että rajoitusmatriisi **A** on tuntematon, eikä tehtävää siksi voida ratkaista suoraan esimerkiksi Simplex-algoritmilla.
- Esitetään seuraavaksi kaksi ratkaisumenetelmää:

Brute-force-menetelmä

- Muodostetaan rajoitusmatriisi **A** kokonaisuudessaan etsimällä kaikki mahdolliset sarakevektorit $\mathbf{A}_j = [a_{1j}, a_{2j}, \ldots, a_{mj}]' \in \mathbb{Z}_+^m$, jotka toteuttavat ehdon

$$\sum_{i=1}^{m} a_{ij} w_i \leq W, \qquad j = 1, 2, \ldots, n.$$

- Geometrisesti tilanne vastaa annetun monitahokkaan sisältämien kokonaislukupisteiden määritystä.
- Menetelmässä joudutaan käymään läpi hyvin suuri määrä pisteitä, mistä johtuen käytetään brute-force-nimeä.
- Kun rajoitusmatriisi on selvitetty, on LP-tehtävä ratkaistavissa Simplexillä.

Sarakkeita generoiva algoritmi

- Tarkoitettu lineaarisen ohjelmoinnin tehtäville.
- Algoritmi alustetaan poimimalla $k \leq n$ kappaletta rajoitusmatriisin $\mathbf{A}$ sarakkeista ja muodostetaan näistä uusi rajoitusmatriisi $\tilde{\mathbf{A}}_k$.
- Ratkaistaan LP-tehtävä rajoitusmatriisin $\tilde{\mathbf{A}}_k$ suhteen.
- Tarkastelemalla ratkaisun $\tilde{\mathbf{x}}_k$ redusoituja kustannuksia, saadaan muodostettua uusi sarake, joka lisätään rajoitusmatriisiin $\tilde{\mathbf{A}}_k$ ($k \hookrightarrow k+1$).
- Ratkaistaan LP-tehtävä rajoitusmatriisin $\tilde{\mathbf{A}}_{k+1}$ suhteen.
- Lopulta löydetään alkuperäisen LP-tehtävän optimiratkaisu jollakin arvolla $k = q$, $q \leq n$.

Työn tavoite

- Ratkaistaan trimmitysongelman LP-relaksaatio sarakkeita generoivalla algoritmilla ja brute-force-menetelmällä.
- Brute-force-menetelmä toteutetaan MATLABilla ja CPLEXillä, sarakkeita generoiva algoritmi pelkällä CPLEXillä.
- Vertaillaan numeerisesti menetelmien laskenta-aikoja erilaisissa instansseissa.

Aikataulu

- Kesä 2011: aiheeseen tutustuminen sekä ratkaisumenetelmien koodaus
- Kesäkuu 2012: kandidaatintyön kirjoitus ja menetelmien numeerinen testaus
- Heinäkuu 2012: kandidaatintyö valmis
- 17.09.2012: aiheen esittely
- 19.11.2012: työn tulosten esittely