Epäsileän optimoinnin DCA-algoritmi lineaarisille komplementaarisille rajoitteille

Veera Maila

Pro gradu -tutkielma
Huhtikuu 2024

Tarkastajat:
Prof. Marko Mäkelä
Dos. Yury Nikulin

MATEMATIIKAN JA TILASTOTIETEEN LAITOS

TURUN YLIOPISTO
Matematiikan ja tilastotieteen laitos

Veera Maila: Epäsileän optimoinnin DCA-algoritmi lineaarisille komplementaarisille rajoitteille
Pro gradu -tutkielma, 31 s.
Sovellettu matematiikka
Huhtikuu 2024

Tässä tutkielmassa käsitellään DCLCC-optimointitehtävien ratkaisemista DCA-algoritmeilla. DCA-algoritmi käsittelee DC-tehtäviä (Difference of two Convex functions) eli tehtäviä, joissa kohdefunktio on kahden konveksin funktion erotus. DCLCC-tehtävät ovat alaluokka DC-tehtäville, joissa on lineaariset komplementaariset rajoitteet. Työ alkaa pohjatiedoilla konveksisuudesta ja optimoinnista, jonka jälkeen esitellään DC-tehtävät ja DCA-algoritmit.

Tutkielman keskiössä on DCLCC-optimointitehtävä ja se, kuinka neljällä erilaisella sakkofunktiolla voidaan formuloida DCLCC-optimointitehtävä DC-tehtäväksi. Uudelleen formulointi halutaan tehdä siksi, että DC-optimointi on tehokas tapa lähestyä epäkonveksia optimointia. Optimointitehtävien muodostamisen jälkeen työssä käydään läpi algoritmien soveltamista QPLCC- ja EiCP-tehtäville. QPLCC-tehtävät ovat kvadraattisia tehtäviä lineaarisilla komplementaarisilla rajoitteilla. EiCP-tehtävät ovat puolestaan epäsymmetrisiä ja komplementaarisia ominaisarvotehtäviä. Tutkielman lopussa esitellään numeerisia tuloksia, joista ilmenee algoritmien tehokkuus aiempiin käytössä oleviin algoritmeihin verrattuna.

Asiasanat: matemaattinen optimointi, DC-funktiot, DCA-algoritmi, DCLCC-optimointitehtävä, sakkofunktiot.

Sisällys

1 Johdanto

Matemaattisessa optimoinnissa tavoitteena on löytää paras mahdollinen ratkaisu annetuissa olosuhteissa, joita kutsutaan rajoitteiksi. Optimointitehtävät esitetään yleisesti minimointitehtävinä, jonka vuoksi konveksit funktiot ovat mielekkäitä. Epäsileys puolestaan tarkoittaa, että funktio ei ole jatkuvasti differentioituva kaikkialla. Epäsileässä optimoinnissa (Nonsmooth Optimization) tätä funktiota arvioidaankin usein aligradienttien avulla muodostetuilla linearisoinneilla. Aligradientit ovat yleistyksiä gradienteista ja alidifferentiaali on joukko, joka muodostuu aligradienteista. Alidifferentiaalia ja aligradientteja käytetään silloin, kun funktiolla ei ole jatkuvia gradientteja. Tällaisia tilanteita tulee vastaan juuri epäsileässä optimoinnissa. [1]

Kahdesti differentioituva funktio voidaan aina esittää kahden konveksin funktion erotuksena eli niin sanottuna DC-funktiona. DCA-algoritmi on kehitetty DC-funktioiden optimointiin ja siinä arvioidaan jälkimmäistä DC-komponenttia linearisoinnilla, jolloin kohdefunktio on muotoa "konveksi miinus lineaarinen".

DCLCC eli DC-tehtävä lineaarisilla komplementaarisilla rajoitteilla on tutkielman keskeisin optimointitehtävä. Kyseinen tehtävä formuloidaan DC-tehtäväksi sakkofunktioiden avulla. DC-tehtävän muodostamisen jälkeen esitellään, kuinka muodostetut DCA-algoritmit voidaan muokata QPLCC- ja EiCP-tehtäviin sopiviksi. QPLCC-tehtävä tarkoittaa kvadraattista tehtävää lineaarisilla komplementaarisilla rajoitteilla ja EiCP-tehtävä on epäsymmetrinen ja komplementaarinen ominaisarvotehtävä. Tutkielman loppuun on listattu numeeriset tulokset. Tämä työ perustuu erityisesti artikkeliin [8].

Tutkielman rakenne on seuraava: Luku 2 koostuu konveksisuuden ja optimoinnin pohjatiedoista. Luku 3 käsittelee DC-funktioita ja DCA-algoritmeja. Lisäksi luvussa esitellään yleinen DCA-algoritmi. Luvussa 4 esitellään DCLCC-optimointitehtävä ja sakkofunktiot, kuten myös optimointitehtävän formulointi DC-tehtäväksi eri sakkofunktioiden avulla. Luku 5 esittää käsiteltyjen optimointimenetelmien soveltamisen QPLCC- ja EiCP-tehtäville. Luvussa 6 on puolestaan listattu numeerisia tuloksia, joista ilmenee käsiteltyjen algoritmien tehokkuus.

2 Pohjatietoa

Aluksi esitellään tutkielmassa käytettyjä merkintöjä sekä konveksisuuteen ja optimointiin liittyviä määritelmiä ja lauseita. Luvun tulokset perustuvat lähteisiin [1] ja [3].

Merkintöjä:

$\mathbb{R}^n$	n-ulotteinen Euklidinen avaruus
$\boldsymbol{x} \to \boldsymbol{y}$	$\boldsymbol{x}$ lähestyy arvoa $\boldsymbol{y}$
$t \downarrow 0$	$t \to 0_+$
sup	*supremum* eli pienin yläraja
inf	*infimum* eli suurin alaraja
max	*maksimi*
min	*minimi*
$\arg\min f(\boldsymbol{x})$	piste, jossa funktio f saa minimiarvonsa
$f(\boldsymbol{x})$	funktion arvo pisteessä $\boldsymbol{x}$
$\nabla^2 f(\boldsymbol{x})$	funktion f *Hessen matriisi* pisteessä $\boldsymbol{x}$
$\nabla f(\boldsymbol{x})$	funktion f *gradientti* pisteessä $\boldsymbol{x}$
$f'(\boldsymbol{x}, \boldsymbol{d})$	funktion f *suuntaderivaatta* pisteessä $\boldsymbol{x}$ suuntaan $\boldsymbol{d}$
$\partial_C f(\boldsymbol{x})$	funktion f *alidifferentiaali* pisteessä $\boldsymbol{x}$
$\partial f(\boldsymbol{x})$	konveksin funktion f alidifferentiaali pisteessä $\boldsymbol{x}$
$\frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x})$	osittaisderivaatta pisteen $\boldsymbol{x}$ suhteen
$K_S(\boldsymbol{x})$	joukon S kontingenttikartio (contingent cone) pisteessä $\boldsymbol{x}$
$T_S(\boldsymbol{x})$	joukon S tangenttikartio pisteessä $\boldsymbol{x}$
$N_S(\boldsymbol{x})$	joukon S normaalikartio pisteessä $\boldsymbol{x}$

Kaikki vektorit $\boldsymbol{x} \in \mathbb{R}^n$ ovat pystyvektoreita ja vastaavasti niiden transpoosit $\boldsymbol{x}^T$ ovat vaakavektoreita. Sisätuloa merkitään $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ ja Euklidisen avaruuden $\mathbb{R}^n$ normia merkitään $\|\boldsymbol{x}\|$. Näin ollen

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{n} x_i y_i \quad \text{ja} \quad \|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle},$$

missä $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ ja $x_i, y_i \in \mathbb{R}$ ovat vektoreiden komponentteja.

Neliömatriisi on matriisi, jolla on yhtä monta vaaka- ja pystyriviä, $n \times n$ matriisi on n-kertainen neliömatriisi. Neliömatriisi $A \in \mathbb{R}^{n \times n}$ on symmetrinen, jos $A = A^T$ eli, jos $(A)_{ij} = (A)_{ji}$ kaikilla $i, j \in \{1, \ldots, n\}$ ja $(A)_{ij}$ on matriisin A alkio vaakarivillä i ja pystyrivillä j. Matriisin A *transpoosi* on $A^T \in \mathbb{R}^{n \times n}$.

Neliömatriisi $A \in \mathbb{R}^{n \times n}$ on *positiividefiniitti*, jos

$$\langle \boldsymbol{x}, A\boldsymbol{x} \rangle > 0, \forall \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq \boldsymbol{0}$$

ja *negatiividefiniitti*, jos

$$\langle \boldsymbol{x}, A\boldsymbol{x} \rangle < 0, \forall\, \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq \mathbf{0}.$$

Vastaavasti matriisi $A \in \mathbb{R}^{n \times n}$ on *positiivisemidefiniitti*, jos

$$\langle \boldsymbol{x}, A\boldsymbol{x} \rangle \geq 0, \forall\, \boldsymbol{x} \in \mathbb{R}^n$$

ja *negatiivisemidefiniitti*, jos

$$\langle \boldsymbol{x}, A\boldsymbol{x} \rangle \leq 0, \forall\, \boldsymbol{x} \in \mathbb{R}^n.$$

Jos matriisi ei ole kumpaakaan edellisistä, se on *indefiniitti.*

Ominaisarvo matriisille $A \in \mathbb{R}^{n \times n}$ on skalaari λ, jos

$$A\boldsymbol{x} = \lambda \boldsymbol{x}$$

jollekin nollasta eroavalle vektorille $\boldsymbol{x} \in \mathbb{R}^n$. Vektoria $\boldsymbol{x}$ kutsutaan ominaisarvoa λ vastaavaksi *ominaisvektoriksi.*

Avointa ja suljettua palloa, jonka keskipiste on $\boldsymbol{x} \in \mathbb{R}^n$ ja säde on $r > 0$, merkitään $B(\boldsymbol{x}, r)$(avoin) ja $\overline{B}(\boldsymbol{x}, r)$(suljettu) eli

$$B(\boldsymbol{x}, r) = \{\boldsymbol{y} \in \mathbb{R}^n \mid \|\boldsymbol{y} - \boldsymbol{x}\| < r\} \quad \text{ja} \quad \overline{B}(\boldsymbol{x}, r) = \{\boldsymbol{y} \in \mathbb{R}^n \mid \|\boldsymbol{y} - \boldsymbol{x}\| \leq r\}.$$

Merkintä $[\boldsymbol{x}, \boldsymbol{y}]$ tarkoittaa suljettua viivasegmenttiä pisteiden $\boldsymbol{x}$ ja $\boldsymbol{y}$ välissä siten, että päätepisteet $\boldsymbol{x}$ ja $\boldsymbol{y}$ kuuluvat kyseiseen joukkoon eli

$$[\boldsymbol{x}, \boldsymbol{y}] = \{\boldsymbol{z} \in \mathbb{R}^n \mid \boldsymbol{z} = \lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}, \quad \text{kun } 0 \leq \lambda \leq 1\}.$$

Vastaavasti merkintä $(\boldsymbol{x}, \boldsymbol{y})$ tarkoittaa avointa viivasegmenttiä, jossa päätepisteet $\boldsymbol{x}$ ja $\boldsymbol{y}$ eivät kuulu mukaan joukkoon.

Merkintä $d(\boldsymbol{x}, S)$ tarkoittaa pisteen $\boldsymbol{x} \in \mathbb{R}^n$ etäisyyttä epätyhjästä joukosta $S \subseteq \mathbb{R}^n$. Sen määritelmä on

$$d(\boldsymbol{x}, S) = \inf\{\|\boldsymbol{x} - \boldsymbol{s}\| \,|\boldsymbol{s} \in S\}.$$

2.1 Konveksisuus

Käydään läpi muutamia perustuloksia konveksisuudesta asioiden käsittelyn helpottamiseksi. Perusmääritelmän lisäksi kerrataan myös konveksi kartio ja normaali kartio, sekä konveksit funktiot.

Määritelmä 1. Olkoon S joukon $\mathbb{R}^n$ osajoukko ($S \subseteq \mathbb{R}^n$). Joukko S on *konveksi*, jos

$$\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in S$$

kaikilla $\boldsymbol{x}, \boldsymbol{y} \in S$ ja $\lambda \in [0, 1]$.

Geometrisesti tämä tarkoittaa, että joukko on konveksi, jos suljettu viivasegmentti $[\boldsymbol{x}, \boldsymbol{y}]$ kuuluu joukkoon S kokonaan riippumatta siitä, missä päätepisteet $\boldsymbol{x}$ ja $\boldsymbol{y}$ ovat joukon S sisällä.

Määritelmä 2. *(Konveksi kartio)* Joukko $C \subseteq \mathbb{R}^n$ on *kartio*, jos $\lambda \boldsymbol{x} \in C$ kaikilla $\boldsymbol{x} \in C$ ja $\lambda \geq 0$. Lisäksi, jos C on konveksi, niin sitä kutsutaan *konveksiksi kartioksi.*

Lause 1. *Joukko $C \subseteq \mathbb{R}^n$ on konveksi kartio, jos ja vain jos*

$$\lambda \boldsymbol{x} + \mu \boldsymbol{y} \in C \text{ kaikilla } \boldsymbol{x}, \boldsymbol{y} \in C \text{ ja } \lambda, \mu \geq 0.$$

Todistus. Jälkimmäisestä ominaisuudesta selvästikin seuraa, että C on konveksi kartio. Oletetaan, että joukko C on konveksi kartio ja $\boldsymbol{x}, \boldsymbol{y} \in C$ ja $\lambda, \mu \geq 0$. Koska joukko C on kartio pätee $\lambda \boldsymbol{x} \in C$ ja $\mu \boldsymbol{y} \in C$. Lisäksi koska C on konveksi saadaan

$$\frac{1}{2}\lambda \boldsymbol{x} + (1 - \frac{1}{2})\mu \boldsymbol{y} \in C$$

ja käyttämällä kartion ominaisuutta saadaan

$$\lambda \boldsymbol{x} + \mu \boldsymbol{y} = 2(\frac{1}{2}\lambda \boldsymbol{x} + (1 - \frac{1}{2})\mu \boldsymbol{y}) \in C.$$

□

Määritelmä 3. *(Tangenttikartio)* Epätyhjän joukon S *tangenttikartio* pisteessä $\boldsymbol{x} \in S$ on joukko

$$T_S(\boldsymbol{x}) = \{\boldsymbol{d} \in \mathbb{R}^n \mid \forall t_i \downarrow 0 \text{ ja } \boldsymbol{x}_i \to \boldsymbol{x} \text{ kun } \boldsymbol{x}_i \in S, \exists \boldsymbol{d}_i \to \boldsymbol{d} \text{ siten että } \boldsymbol{x}_i + t_i \boldsymbol{d}_i \in S\}.$$

Määritelmä 4. *(Normaalikartio)* Epätyhjän joukon S *normaalikartio* pisteessä $\boldsymbol{x} \in S$ on joukko

$$N_S(\boldsymbol{x}) := \{\boldsymbol{z} \in \mathbb{R}^n \mid \langle \boldsymbol{z}, \boldsymbol{d} \rangle \leq 0, \forall\, \boldsymbol{d} \in T_S(\boldsymbol{x})\}.$$

Joukon $N_S(\boldsymbol{x})$ alkioita kutsutaan *normaalivektoreiksi.*

Seuraavaksi käydään läpi konveksit funktiot.

Määritelmä 5. Funktio $f : \mathbb{R}^n \to \mathbb{R}$ on *konveksi*, jos

$$f(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1 - \lambda) f(\boldsymbol{y}),$$

kun pisteet $\boldsymbol{x}$ ja $\boldsymbol{y}$ kuuluvat joukkoon $\mathbb{R}^n$ ja $\lambda \in [0, 1]$. Jos epäyhtälössä yllä on aito epäsuuruus kaikilla $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ siten, että $\boldsymbol{x} \neq \boldsymbol{y}$ ja $\lambda \in (0, 1)$, niin funktio f on *aidosti konveksi.* Jos on olemassa $\alpha > 0$ jolla $f(\boldsymbol{x}) - \alpha \|\boldsymbol{x}\|^2$ on konveksi, niin funktio f on *vahvasti konveksi.* Funktio $f : \mathbb{R}^n \to \mathbb{R}$ on *konkaavi*, jos $-f$ on konveksi.

Seuraava lause esittää toisen määritelmän konveksille funktiolle.

Lause 2. *(Jensenin epäyhtälö) Funktio $f : \mathbb{R}^n \to \mathbb{R}$ on konveksi, jos ja vain jos*

$$f\left(\sum_{i=1}^{m} \lambda_i \boldsymbol{x}_i\right) \leq \sum_{i=1}^{m} \lambda_i f(\boldsymbol{x}_i),$$

aina kun $\boldsymbol{x}_i \in \mathbb{R}^n, \lambda_i \in [0,1]$ kaikilla $i = 1, \ldots, m$ ja $\sum_{i=1}^{m} \lambda_i = 1$.

Todistus. Seuraa induktiolla konveksin funktion määritelmästä. □

Funktio f on *lokaalisti Lipschitz-jatkuva*, jos kaikissa pisteissä $\boldsymbol{x} \in \mathbb{R}^n$ on olemassa Lipschitz vakio $L > 0$ ja skalaari $\varepsilon > 0$ joilla

$$|f(\boldsymbol{y}) - f(\boldsymbol{z})| \leq L\|\boldsymbol{y} - \boldsymbol{z}\| \quad \forall\, \boldsymbol{y}, \boldsymbol{z} \in B(\boldsymbol{x}, \varepsilon).$$

Konveksi funktio on aina lokaalisti Lipschitz-jatkuva [1].

Suuntaderivaatta antaa arvion funktion arvojen muutoksesta, minkä vuoksi sitä voidaan käyttää löytämään niitä suuntia, joihin funktion arvo kasvaa, pienenee tai pysyy samana. Funktion $f : \mathbb{R}^n \to \mathbb{R}$ *suuntaderivaatta* pisteessä $\boldsymbol{x} \in \mathbb{R}^n$ suuntaan $\boldsymbol{d} \in \mathbb{R}^n$ määritellään

$$f'(\boldsymbol{x}, \boldsymbol{d}) = \lim_{t \downarrow 0} \frac{f(\boldsymbol{x} + t\boldsymbol{d}) - f(\boldsymbol{x})}{t}.$$

Gradientin yleistystä kutsutaan aligradientiksi [3]. Se on tärkeä työkalu optimoinnissa silloin, kun jatkuvia gradientteja ei funktiolla ole.

Määritelmä 6. *Alidifferentiaali* konveksille funktiolle $f : \mathbb{R}^n \to \mathbb{R}$ pisteessä $\boldsymbol{x} \in \mathbb{R}^n$ on joukko

$$\partial f(\boldsymbol{x}) = \big\{\boldsymbol{\xi} \in \mathbb{R}^n \mid f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{\xi}, \boldsymbol{y} - \boldsymbol{x} \rangle \text{ kaikilla } \boldsymbol{y} \in \mathbb{R}^n\big\}.$$

Jokainen vektori $\boldsymbol{\xi} \in \partial f(\boldsymbol{x})$ on funktion f *aligradientti* pisteessä $\boldsymbol{x}$.

Seuraavan lauseen mukaan konveksi funktio voidaan esittää sen aligradienttien avulla.

Lause 3. *Jos funktio $f : \mathbb{R}^n \to \mathbb{R}$ on konveksi, niin kaikille pisteille $\boldsymbol{y} \in \mathbb{R}^n$*

$$f(\boldsymbol{y}) = \max\{f(\boldsymbol{x}) + \langle \boldsymbol{\xi}, \boldsymbol{y} - \boldsymbol{x} \rangle \mid \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{\xi} \in \partial f(\boldsymbol{x})\}.$$

Todistus. Oletetaan, että $\boldsymbol{y} \in \mathbb{R}^n$ mielivaltainen piste ja $\boldsymbol{\zeta} \in \partial f(\boldsymbol{y})$. Olkoon

$$S := \{f(\boldsymbol{x}) + \langle \boldsymbol{\xi}, (\boldsymbol{y} - \boldsymbol{x}) \rangle \mid \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{\xi} \in \partial f(\boldsymbol{x})\}.$$

Konveksin funktion alidifferentiaalin määritelmän mukaan

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{\xi}, (\boldsymbol{y} - \boldsymbol{x}) \rangle \text{ kaikilla } \boldsymbol{x} \in \mathbb{R}^n \text{ ja } \boldsymbol{\xi} \in \partial f(\boldsymbol{x})$$

vihjaten, että joukko S on ylhäältä rajoitettu ja

$$\sup S \leq f(\boldsymbol{y}).$$

Toisaalta pätee

$$f(\boldsymbol{y}) = f(\boldsymbol{y}) + \langle \boldsymbol{\zeta}, (\boldsymbol{y} - \boldsymbol{y}) \rangle \in S,$$

mikä merkitsee, että $f(\boldsymbol{y}) \leq \sup S$. Täten

$$f(\boldsymbol{y}) = \max\{f(\boldsymbol{x}) + \langle \boldsymbol{\xi}, (\boldsymbol{y} - \boldsymbol{x}) \rangle \mid \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{\xi} \in \partial f(\boldsymbol{x})\}.$$

□

Määritelmä 7. *(Clarken yleistetty suuntaderivaatta)* Olkoon funktio $f : \mathbb{R}^n \to \mathbb{R}$ lokaalisti Lipschitz-jatkuva pisteessä $\boldsymbol{x} \in \mathbb{R}^n$. Funktion f *yleistetty suuntaderivaatta* pisteessä $\boldsymbol{x}$ suuntaan $\boldsymbol{d} \in \mathbb{R}^n$ määritellään

$$f^o(\boldsymbol{x}, \boldsymbol{d}) = \limsup_{\boldsymbol{y} \to \boldsymbol{x},\, t \downarrow 0} \frac{f(\boldsymbol{y} + t\boldsymbol{d}) - f(\boldsymbol{y})}{t}.$$

Määritelmä 8. *(Clarken alidifferentiaali)* Olkoon funktio $f : \mathbb{R}^n \to \mathbb{R}$ lokaalisti Lipschitz-jatkuva pisteessä $\boldsymbol{x} \in \mathbb{R}^n$. Silloin funktion f *yleistetty alidifferentiaali* tai *Clarken alidifferentiaali* pisteessä $\boldsymbol{x}$ on joukko $\partial_C f(\boldsymbol{x})$ vektoreita $\boldsymbol{\xi} \in \mathbb{R}^n$ siten, että

$$\partial_C f(\boldsymbol{x}) = \{\boldsymbol{\xi} \in \mathbb{R}^n \mid f^o(\boldsymbol{x}, \boldsymbol{d}) \geq \langle \boldsymbol{\xi}, \boldsymbol{d} \rangle, \forall\, \boldsymbol{d} \in \mathbb{R}^n\}.$$

Jokainen vektori $\boldsymbol{\xi} \in \partial_C f(\boldsymbol{x})$ on funktion f *yleistetty aligradientti* tai *Clarken aligradientti* pisteessä $\boldsymbol{x}$.

Lause 4. *Jos funktio $f : \mathbb{R}^n \to \mathbb{R}$ on konveksi, niin*

(i) $f'(\boldsymbol{x}, \boldsymbol{d}) = f^o(\boldsymbol{x}, \boldsymbol{d}), \quad \forall\, \boldsymbol{d} \in \mathbb{R}^n$ *ja*

(ii) $\partial f(\boldsymbol{x}) = \partial_C f(\boldsymbol{x})$.

Todistus. Todistettu lähteessä [1]. □

Aligradientit ovat yleistyksiä normaalista gradientista.

Lause 5. *Jos funktio f on jatkuvasti differentioituva pisteessä $\boldsymbol{x}$, niin*

$$\partial_C f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}.$$

Todistus. Todistettu lähteessä [1]. □

2.2 Optimoinnista

Tässä työssä tarkasteltava optimointitehtävä on muotoa

$$\begin{cases} \min & f(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in S, \end{cases} \tag{1}$$

missä kohdefunktio $f : \mathbb{R}^n \to \mathbb{R}$ on lokaalisti Lipschitz-jatkuva ja sallittu joukko $S \subseteq \mathbb{R}^n$ on epätyhjä. *Minimize* (lyhyesti min) tarkoittaa kohdefunktion minimointia ja *subject to* (lyhyesti s.t.) merkitsee rajoitteita. Jos funktio f on konveksi ja S on konveksi joukko, niin tehtävää (1) kutsutaan konveksiksi. Jos $S = \mathbb{R}^n$, niin tehtävää (1) kutsutaan rajoitteettomaksi.

Määritelmä 9. Piste $\boldsymbol{x}^* \in S$ on tehtävän (1) *globaali optimi (tai minimi)*, jos se toteuttaa

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in S.$$

Määritelmä 10. Piste $\boldsymbol{x}^* \in S$ on tehtävän (1) *lokaali optimi (tai minimi)*, jos on olemassa sellainen $\delta > 0$, että

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in S \cap B(\boldsymbol{x}^*, \delta).$$

Optimointi on helpointa konvekseille ja lineaarisille funktioille. Käytännössä useat reaalimaailman mallinnustehtävät eivät ole kumpaakaan [3]. Kun kohfefunktio on jatkuvasti differentioituva, sitä kutsutaan *sileäksi.* Puolestaan, kun tehtävä ei ole sileä, kutsutaan sitä epäsileäksi. Kyseessä on tällöin siis *epäsileän optimoinnin* tehtävä.

Rajoitteettomassa sileässä optimoinnissa optimaalisuusehtona on se, että optimissa gradientin arvo on nolla. Kuitenkaan se, että gradientti on nolla, ei takaa optimaalisuutta, ellei funktio ole konveksi. Mikäli kaikki gradientin nollakohdat löydetään, on optimi välttämättä oltava näiden joukossa, jos se ylipäätään on olemassa. Lisäksi optimi on pisteistä se, jolla kohdefunktion arvo on pienin.
Nämä tulokset voidaan yleistää myös rajoitteiseen epäsileään tapaukseen.

Lause 6. *Olkoon $\boldsymbol{x}^*$ tehtävän (1) lokaali optimi, missä funktio $f : \mathbb{R}^n \to \mathbb{R}$ lokaalisti Lipschitz-jatkuva pisteessä $\boldsymbol{x}^* \in S \neq \emptyset$. Tällöin*

$$\mathbf{0} \in \partial_C f(\boldsymbol{x}^*) + N_S(\boldsymbol{x}^*).$$

Todistus. Todistettu lähteessä [1]. □

Lause 7. *Jos optimointitehtävä (1) on konveksi, niin $\boldsymbol{x}^* \in S$ on tehtävän (1) globaali optimi, jos ja vain jos*

$$\mathbf{0} \in \partial f(\boldsymbol{x}^*) + N_S(\boldsymbol{x}^*).$$

Todistus. Todistettu lähteessä [1]. □

Jos tehtävä on rajoitteeton eli $S = \mathbb{R}^n$, niin silloin $N_S(\boldsymbol{x}) = \mathbb{R}^n$ ja lauseiden 6 ja 7 optimaalisuusehdot redusoituvat muotoon $\mathbf{0} \in \partial_C f(\boldsymbol{x}^*)$ ja $\mathbf{0} \in \partial f(\boldsymbol{x}^*)$.

Määritelmä 11. Pistettä $\boldsymbol{x} \in \mathbb{R}^n$ jolle pätee $\mathbf{0} \in \partial_C f(\boldsymbol{x})$ kutsutaan funktion f *stationääriseksi pisteeksi.*

Seuraavaksi tarkastellaan kohdefunktion laskeutumissuuntaa. Konveksille funktiolle laskeutumissuunta osoittaa aina kohti globaalia minimiä.

Määritelmä 12. Suunta $\boldsymbol{d} \in \mathbb{R}^n$ on funktion $f : \mathbb{R}^n \to \mathbb{R}$ *laskeutumissuunta* pisteessä $\boldsymbol{x} \in \mathbb{R}^n$, jos on olemassa sellainen $\varepsilon > 0$, että kaikille $t \in (0, \varepsilon]$

$$f(\boldsymbol{x} + t\boldsymbol{d}) < f(\boldsymbol{x}).$$

Lause 8. *Olkoon funktio $f : \mathbb{R}^n \to \mathbb{R}$ lokaalisti Lipschitz-jatkuva pisteessä $\boldsymbol{x} \in \mathbb{R}^n$. Suunta $\boldsymbol{d} \in \mathbb{R}^n$ on funktion f laskeutumissuunta pisteessä $\boldsymbol{x}$, jos*

$$\langle \boldsymbol{\xi}, \boldsymbol{d} \rangle < 0 \quad \forall \quad \boldsymbol{\xi} \in \partial_C f(\boldsymbol{x}) \qquad \textit{tai} \qquad f^o(\boldsymbol{x}, \boldsymbol{d}) < 0.$$

Todistus. Todistettu lähteessä [1]. □

Lauseen 8 tuloksen perusteella voidaan generoida laskeutumissuunta epäsileälle funktiolle. Tulos edellyttää kuitenkin koko alidiferentiaalin tuntemista, mikä valitettavasti käytännössä on usein liian suuri vaatimus [3]. Näin ollen perusoletuksena epäsileässä optimoinnissa onkin, että koko alidifferentiaalin sijaan jokaisessa pisteessä on saatavilla yksi mielivaltainen aligradientti.

Kimppumenetelmät puolestaan arvioivat kohdefunktion alidifferentiaalia joukolla edellisillä iteraatiokierroksilla kerätyillä aligradienteilla. Saatujen tietojen avulla kohdefunktiosta luodaan malli, jonka avulla voidaan määrittää alkuperäisen funktion laskeutumissuunta. Vaikka kimppumenetelmien konvergointinopeudet eivät myöskään ole kovinkaan hyviä, ovat ne yleisesti ottaen tehokkaampia menetelmiä kuin aligradienttimetodit. [3].

Esitellään vielä Slaterin rajoite-edelletys, joka on keskeinen käsite jatkossa. Nimittäin välttämättömien KKT-optimaalisuusehtojen oletuksena on Slaterin rajoite-edellytyksen toteutuminen [1].

Määritelmä 13. Funktio g toteuttaa *Slaterin rajoite-edellytyksen*, jos on olemassa $\boldsymbol{x} \in S$ siten, että $g(\boldsymbol{x}) < 0$.

3 DC-funktioista

DC-funktiot *(Difference of Convex)* muodostavat erittäin suuren osan epäkonvekseista funktioista, sillä kaikki ainakin kahdesti differentioituvat funktiot voidaan kirjoittaa kahden konveksin funktion erotuksena [3]. Tällä valinnalla voidaan arvioida erikseen konveksia ja konkaavia osaa.

3.1 DC-funktiot

DC-funktio on kahden konveksin funktion erotus. DC-funktio on muotoa

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) - h(\boldsymbol{x}),$$

missä g ja h ovat konvekseja funktioita. Tässä on hyvä huomata, että kahden konveksin funktion erotus on yleensä epäkonveksi. Tämän takia perinteiset konveksit optimointimenetelmät eivät sovellu suoraan DC-funktioille.

Lause 9. *Seuraavat DC-tehtävän muotoilut ovat ekvivalentteja:*

(i) $\sup\{f(\boldsymbol{x}) : \boldsymbol{x} \in C\}$, *missä funktio f ja joukko C ovat konvekseja,*

(ii) $\inf\{g(\boldsymbol{x}) - h(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^n\}$, *missä funktiot g ja h ovat konvekseja,*

(iii) $\inf\{g(\boldsymbol{x}) - h(\boldsymbol{x}) : \boldsymbol{x} \in C, f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}) \leq 0\}$, *missä funktiot g, h, f_1 ja f_2 sekä joukko C ovat konvekseja.*

Todistus. Todistettu lähteessä [10]. □

DC-tehtävät käsittelevät minimointitehtävää funktiolle f, joka on konveksien funktioiden erotus koko määrittelyjoukossa X. Yleisesti ottaen, niin kutsuttu *standardi DC-tehtävä* on muotoa

$$\alpha = \inf\{f(\boldsymbol{x}) := g(\boldsymbol{x}) - h(\boldsymbol{x}) : \boldsymbol{x} \in X\}, \tag{2}$$

missä g ja h ovat konvekseja funktioita. DC-funktiolla f on DC-komponentit g ja h. Erotus $g - h$ on DC-hajotelma funktiosta f. [6]

Määritelmä 14. *Polyhedraali DC-tehtävä* on DC-tehtävä, jossa ainakin toinen funktioista f tai h on konveksi ja polyhedraalinen. [5]

Lause 10. *Olkoot funktiot $g, h : \mathbb{R}^n \to \mathbb{R}$ konvekseja. Jos piste $\boldsymbol{x}^* \in \mathbb{R}^n$ on funktion $f = g - h$ lokaali minimi, niin seuraavat ehdot pätevät*

$$\begin{aligned} \partial h(\boldsymbol{x}^*) &\subset \partial g(\boldsymbol{x}^*), \\ \mathbf{0} &\in \partial_C f(\boldsymbol{x}^*) \textit{ ja} \\ \partial g(\boldsymbol{x}^*) \cap \partial h(\boldsymbol{x}^*) &\neq \emptyset. \end{aligned}$$

Toisin sanoen mikäli ratkaisu on olemassa, on näiden kolmen ehdon myös toteuduttava [3].

Piste $\boldsymbol{x}^*$ on primäärisen DC-tehtävän (2) (eli funktion $f = g - h$) *kriittinen piste*, jos $\partial g(\boldsymbol{x}^*) \cap \partial h(\boldsymbol{x}^*) \neq \emptyset$. Sitä kutsutaan primäärisen DC-tehtävän (2) *vahvasti kriittiseksi pisteeksi*, jos $\emptyset \neq \partial h(\boldsymbol{x}^*) \subset \partial g(\boldsymbol{x}^*)$ [6].

3.2 DCA-algoritmi

Alkuperäinen DCA-algoritmin pääidea nojautuu DC-tehtävän kohdefunktion rakenteeseen $f = g - h$ sekä kriittisten ja vahvasti kriittisten pisteiden tarkasteluun. Jokaisella iteraatiolla k DCA-algoritmi arvioi komponenttia $h(\boldsymbol{x})$ sen affiinisella approksimaatiolla. Matemaattisesti voidaan siis ilmaista $h(\boldsymbol{x}^k) + \langle \boldsymbol{y}^k, \boldsymbol{x} - \boldsymbol{x}^k \rangle$, missä $\boldsymbol{y}^k \in \partial h(\boldsymbol{x}^k)$ [6]. Tämän jälkeen saatu konveksi funktio minimoidaan. Toisin sanoen linearisoidaan funktio h, jolloin kohdefunktio on muotoa "konveksi miinus lineaarinen".

Tavallinen DCA-algoritmi standardin DC-tehtävän (2) ratkaisemiseen

Alustus Valitse aloituspiste $\boldsymbol{x}^0 \in \mathbb{R}^n$, aseta $k := 0$.
Toista

Askel 1: Laske $\boldsymbol{y}^k \in \partial h(\boldsymbol{x}^k)$.

Askel 2: Laske $\boldsymbol{x}^{k+1} \in \operatorname{argmin}\{g(\boldsymbol{x}) - h(\boldsymbol{x}^k) - \langle \boldsymbol{y}^k, \boldsymbol{x} - \boldsymbol{x}^k \rangle : \boldsymbol{x} \in \mathbb{R}^n\}$.

Askel 3: Aseta $k := k + 1$.

Kunnes $\{\boldsymbol{x}^k\}$ konvergoi.

DCA-algoritmin konvergointi kriittiseen pisteeseen standardille DC-tehtävälle on todettu lähteen [12] lauseessa 3.

3.3 Yleinen DCA

Niin sanotun *yleisen DC-tehtävän* muoto on

$$\min\{g(\boldsymbol{x}) - h(\boldsymbol{x}) : \boldsymbol{x} \in C, g_i(\boldsymbol{x}) - h_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, m\}, \tag{3}$$

missä C on epätyhjä suljettu konveksi joukko euklidisessa avaruudessa $\mathbb{R}^n$ ja g, h, g_i, h_i, kun $i = 1, \ldots, m$, ovat konvekseja funktioita. Tämän luokan DC-tehtävät ovat kaikkein yleisimpiä DC-tehtävien muotoja. Lisäksi ne ovat vaikeampia kuin standardit DC-tehtävät, koska rajoitteet ovat epäkonvekseja. Optimointitehtävän (3) ratkaisemiseksi on esitetty pari vaihtoehtoa lähteessä [8]. Ensimmäinen tavoista hyödyntää sakkofunktiota, jolloin rajoitteinen tehtävä (3) muuntuu standardin DC-tehtävän (2) muotoon. Tehtävä ratkaistaan DCA-algoritmista kehitetyllä versiolla, jossa päivitetään sakkoparametria. Toinen tapa käsitellä DC-tehtävän rajoitteita on käyttää vastaavaa ideaa kuin standardin tehtävän (2) DCA-algoritmissa. Tällöin linearisoidaan DC-tehtävän rajoitteiden jälkimmäiset DC-komponentit, jotta voidaan luoda arvio sallitusta joukosta. Tulokseksi saatu konveksi aliongelma on jokaisella iteraatiokierroksella muotoa

$$\min\{g(\boldsymbol{x}) - \langle \boldsymbol{y}^k, \boldsymbol{x}\rangle : \boldsymbol{x} \in C, g_i(\boldsymbol{x}) - h_i(\boldsymbol{x}^k) - \langle \boldsymbol{y}_i^k, \boldsymbol{x} - \boldsymbol{x}^k\rangle \leq 0, \forall\, i\}, \tag{4}$$

missä $\boldsymbol{y}^k \in \partial h(\boldsymbol{x}^k), \boldsymbol{y}_i^k \in \partial h_i(\boldsymbol{x}^k)$ kaikilla $i = 1, \ldots, m$. Usein tälläinen konveksi approksimaatio ei kuitenkaan ole riittävän hyvä, ja se voi tehdä konveksin aliongelman (4) ratkaisemisen mahdottomaksi. Tällöin aliongelmiin voidaan kokeilla relaksointia. Tällöin tehtävän (4) sijaan ratkaistaankin ongelmaa

$$\begin{aligned} \min\{g(\boldsymbol{x}) - \langle \boldsymbol{y}^k, \boldsymbol{x}\rangle + t_k s : \boldsymbol{x} \in C, s \geq 0, \\ g_i(\boldsymbol{x}) - h_i(\boldsymbol{x}^k) - \langle \boldsymbol{y}_i^k, \boldsymbol{x} - \boldsymbol{x}^k\rangle \leq s, i = 1, \ldots, m\}, \end{aligned} \tag{5}$$

missä $t_k \geq 0$ on sakkoparametri. Selvästi (5) on konveksi optimointitehtävä, joka on aina mahdollista ratkaista. Lisäksi Slaterin rajoite-edellytys täyttyy optimointitehtävän (5) rajoitteille, jolloin Karush-Kuhn-Tuckerin (KKT) optimaalisuusehto pätee ratkaisulle $(\boldsymbol{x}^{k+1}, s^{k+1})$ tehtävässä (5) [8]. Optimointitehtävän (3) DCA-algoritmi on seuraavanlainen.

DCA-algoritmi yleisen DC-tehtävän (3) ratkaisemiseen

Alustus: Valitse aloituspiste $\boldsymbol{x}^0 \in C; \delta_1, \delta_2 > 0$ ja aloitussakkoparametri $t_0 > 0$. Aseta $k := 0$.

Askel 1: Laske $\boldsymbol{y}^k \in \partial h(\boldsymbol{x}^k)$, $\boldsymbol{y}_i^k \in \partial h_i(\boldsymbol{x}^k)$, missä $i = 1, \ldots, m$.

Askel 2: Laske konveksin tehtävän (5) ratkaisu $(\boldsymbol{x}^{k+1}, s^{k+1})$ ja vastaavat Lagrangen kertoimet λ_i^{k+1} kaikille $i = 1, \ldots, m$.

Askel 3: *(Pysäytystesti)* jos $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k$ ja $s^{k+1} = 0$ niin lopeta, muuten siirry askeleeseen 4.

Askel 4: *(Sakkoparametrin päivitys)* laske

$$r_k = \min\Big(\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|^{-1}, \sum_{i=1}^{m} |\lambda_i^{k+1}| + \delta_1\Big)$$

ja aseta

$$t_{k+1} = \begin{cases} t_k & , \text{jos } t_k \geq r_k, \\ t_k + \delta_2 & , \text{jos } t_k < r_k. \end{cases}$$

Askel 5: Aseta $k := k + 1$ ja siirry askeleeseen 1.

Yllä olevan algoritmin globaali konvergenssi kriittiseen pisteeseen on todistettu lähteessä [5]. Sakkoparametrin päivitys r_k on valittu lähteen [8] mukaan.

4 DCLCC ja sakkofunktiot

Seuraavaksi esitellään DCLCC-optimointitehtävä, joka formuloidaan DC-tehtäväksi sakkofunktioiden avulla.

4.1 DCLCC-optimointitehtävä ja stationäärisyyden käsitteet

DCLCC (DC lineaarisilla komplementaarisilla rajoitteilla) on optimointitehtävä, joka on muotoa

$$\begin{array}{ll} \min & f(\boldsymbol{x}, \boldsymbol{y}) = g(\boldsymbol{x}, \boldsymbol{y}) - h(\boldsymbol{x}, \boldsymbol{y}) \\ \text{s.t.} & A\boldsymbol{x} + B\boldsymbol{y} + \boldsymbol{a} \leq \boldsymbol{0}, \\ & N\boldsymbol{x} + M\boldsymbol{y} + \boldsymbol{q} = \boldsymbol{w}, \\ & \boldsymbol{y} \geq \boldsymbol{0}, \quad \boldsymbol{w} \geq \boldsymbol{0}, \quad \langle \boldsymbol{y}, \boldsymbol{w} \rangle \geq 0, \end{array} \tag{6}$$

missä funktiot $g, h : \mathbb{R}^{n+m} \to \mathbb{R}$ ovat konvekseja ja jatkuvasti differentioituvia, parametrit $\boldsymbol{a} \in \mathbb{R}^p, \boldsymbol{q} \in \mathbb{R}^m, A \in \mathbb{R}^{p\times n}, B \in \mathbb{R}^{p\times m}, N \in \mathbb{R}^{m\times n}$ ja $M \in \mathbb{R}^{m\times m}$. Lineaariset komplementaariset rajoitteet rajaavat tehtävän sallittut ratkaisut, eli mitkä ratkaisut ovat käytännössä mahdollisia. Optimointitehtävän muotoilu seuraa lähteestä [8].

Selkeyden vuoksi käydään läpi joitain DCLCC-tehtävän stationäärisiä konsepteja. Olkoon $\mathcal{F}$ optimointitehtävän (6) sallittujen pisteiden joukko. Kaikille $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*) \in \mathcal{F}$, määritellään indeksijoukot:

$$\begin{aligned} I_y(\boldsymbol{y}^*, \boldsymbol{w}^*) &= \{i : \boldsymbol{y}_i^* < \boldsymbol{w}_i^*\}, \\ I_w(\boldsymbol{y}^*, \boldsymbol{w}^*) &= \{i : \boldsymbol{y}_i^* > \boldsymbol{w}_i^*\}, \\ I_0(\boldsymbol{y}^*, \boldsymbol{w}^*) &= \{i : \boldsymbol{y}_i^* = \boldsymbol{w}_i^* = 0\}. \end{aligned}$$

Määritelmä 15. Piste $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*) \in \mathcal{F}$ on:

(a) *heikosti stationäärinen*, jos on olemassa kertoimet $\boldsymbol{\beta} \in \mathbb{R}^p$ ja $\boldsymbol{v}^y, \boldsymbol{v}^w \in \mathbb{R}^m$ siten, että

$$\begin{aligned}
&\nabla_x f(\boldsymbol{x}^*, \boldsymbol{y}^*) + A^T \boldsymbol{\beta} - N^T \boldsymbol{v}^w = \boldsymbol{0}, \\
&\nabla_y f(\boldsymbol{x}^*, \boldsymbol{y}^*) + B^T \boldsymbol{\beta} - M^T \boldsymbol{v}^w - \boldsymbol{v}^y = \boldsymbol{0}, \\
&\boldsymbol{\beta} \geq \boldsymbol{0}, \langle \boldsymbol{\beta}, A\boldsymbol{x}^* + B\boldsymbol{y}^* + \boldsymbol{a} \rangle = 0, \\
&v_i^w = 0, i \in I_y(\boldsymbol{y}^*, \boldsymbol{w}^*), \\
&v_i^y = 0, i \in I_w(\boldsymbol{y}^*, \boldsymbol{w}^*),
\end{aligned}$$

(b) *vahvasti stationäärinen*, jos se on heikosti stationäärinen ja

$$v_i^y \geq 0, v_i^w \geq 0 \quad \forall\, i \in I_0(\boldsymbol{y}^*, \boldsymbol{w}^*),$$

(c) *B-stationäärinen*, jos

$$\langle \nabla_x f(\boldsymbol{x}^*, \boldsymbol{y}^*), d\boldsymbol{x} \rangle + \langle \nabla_y f(\boldsymbol{x}^*, \boldsymbol{y}^*), d\boldsymbol{y} \rangle \geq 0, \quad \forall\, (d\boldsymbol{x}, d\boldsymbol{y}, d\boldsymbol{w}) \in T_{\mathcal{F}}(\boldsymbol{z}^*),$$

missä $T_{\mathcal{F}}(\boldsymbol{z}^*)$ on optimointitehtävän sallitun joukon $\mathcal{F}$ tangenttikartio pisteessä $\boldsymbol{z}^*$.

Lisätietoa stationäärityyksistä löytty muun muassa lähteistä [8] ja [13].

Huomautus 1.

(i) Jos $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on vahvasti stationäärinen piste optimointitehtävässä (6), niin se on myös B-stationäärinen piste samassa tehtävässä [13].

(ii) Jos f on konveksi ja $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on B-stationäärinen piste optimointitehtävässä (6), niin $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on optimointitehtävän (6) lokaali optimi [9].

Seuraavaksi esitellään DCA-algoritmi, joka sopii DCLPP-tehtävälle.

4.2 DCLCC-tehtävän formulointi DC-tehtäväksi sakkofunktioiden avulla

Oletetaan koko ajan, että optimointitehtävän (6) lineaaristen rajoitteiden määrittelemä joukko on epätyhjä. Määritellään funktio $\psi : \mathbb{R}^2 \to \mathbb{R}$ seuraavasti $\psi \in \{\psi^{\min}, \psi^{FB}\}$ ja

$$\begin{aligned}
&\psi^{\min}(a, b) = \min(a, b) \text{ (minimi funktio)} \\
&\psi^{FB}(a, b) = a + b - \sqrt{a^2 + b^2} \text{ (Fischer-Burmeister funktio).}
\end{aligned}$$

Voidaan todistaa [8], että ψ toteuttaa seuraavat ominaisuudet:

(i) ψ on konkaavi ja jatkuva joukossa $\mathbb{R}^2_+$;

(ii) $\psi(a, b) \geq 0, \forall\, a, b \geq 0$;

(iii) $\psi(a,b) = 0 \Leftrightarrow ab = 0, a \geq 0, b \geq 0.$

Täten komplementaarinen rajoite $\langle \boldsymbol{y}, \boldsymbol{w} \rangle = 0$ tehtävässä (6) voidaan korvata

$$\text{joko } \sum_{i=1}^{m} \psi(y_i, w_i) = 0 \text{ tai } \max_{i=1\ldots,m} \psi(y_i, w_i) = 0.$$

Olkoon nyt p lyhenne yhdestä neljästä sakkofunktiosta $p_i : \mathbb{R}^{2m} \to \mathbb{R}, i = 1, \ldots, 4$, jotka ovat

$$p_1(\boldsymbol{y}, \boldsymbol{w}) := \sum_{i=1}^{m} \psi^{\min}(y_i, w_i) = \sum_{i=1}^{m} \min(y_i, w_i), \tag{7}$$

$$p_2(\boldsymbol{y}, \boldsymbol{w}) := \sum_{i=1}^{m} \psi^{FB}(y_i, w_i) = \sum_{i=1}^{m} (y_i + w_i - \sqrt{(y_i)^2 + (w_i)^2}), \tag{8}$$

$$p_3(\boldsymbol{y}, \boldsymbol{w}) := \max_{i=1,\ldots,m} \psi^{\min}(y_i, w_i) = \max_{i=1,\ldots,m} \min(y_i, w_i), \tag{9}$$

$$p_4(\boldsymbol{y}, \boldsymbol{w}) := \max_{i=1,\ldots,m} \psi^{FB}(y_i, w_i) = \max_{i=1,\ldots,m} (y_i + w_i - \sqrt{(y_i)^2 + (w_i)^2}). \tag{10}$$

Olkoon joukko $\mathcal{C}$ määritelty optimointitehtävän (6) lineaaristen rajoitteiden mukaan, eli

$$\mathcal{C} = \Big\{ (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) \in \mathbb{R}^{n+2m} : A\boldsymbol{x} + B\boldsymbol{y} + \boldsymbol{a} \leq \boldsymbol{0}, N\boldsymbol{x} + M\boldsymbol{y} + \boldsymbol{q} = \boldsymbol{w}, \boldsymbol{y} \geq \boldsymbol{0}, \boldsymbol{w} \geq \boldsymbol{0} \Big\}.$$

Siten optimointitehtävän (6) sallittujen pisteiden joukko $\mathcal{F}$ on

$$\mathcal{F} = \{ (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C} : p(\boldsymbol{y}, \boldsymbol{w}) \leq 0 \},$$

ja optimointitehtävä (6) voidaan kirjoittaa muotoon

$$\min\{ f(\boldsymbol{x}, \boldsymbol{y}) : (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{F} \}. \tag{11}$$

Tarkastellaan sakkotehtävää

$$\min\{ f(\boldsymbol{x}, \boldsymbol{y}) + tp(\boldsymbol{y}, \boldsymbol{w}) : (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C} \}. \tag{12}$$

Seuraava lause osoittaa, että kun joukko $\mathcal{C}$ on rajoitettu niin käytetty sakkofunktio on eksakti eli sakkotehtävä (12) on ekvivalentti tehtävän (11) kanssa, kun t on tarpeeksi suuri.

Lause 11. *Olkoon $\mathcal{C}$ rajoitettu ja optimointitehtävän (6) sallittujen pisteiden joukko epätyhjä. Täten on olemassa positiivinen luku t_0 siten, että kaikilla $t > t_0$ optimointitehtävät (11) ja (12) ovat ekvivalentteja niin, että niillä on sama optimiarvo ja optimaalinen ratkaisu.*

Todistus. Todistus seuraa lähteessä [8] esitettyä todistusta. Ensin todistetaan, että on olemassa $t > 0$ jolle

$$d((\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}), \mathcal{F}) \leq tp(\boldsymbol{y}, \boldsymbol{w}), \quad \forall\, (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}.$$

Lauseen 5 mukaan lähteessä [7], on olemassa $t_1 > 0$ jolla

$$d((\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}),\mathcal{F}) \leq t_1 p_1(\boldsymbol{y},\boldsymbol{w}), \quad \forall\,(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}. \tag{13}$$

Toisaalta,

$$p_1(\boldsymbol{y},\boldsymbol{w}) \leq \frac{2+\sqrt{2}}{2} p_2(\boldsymbol{y},\boldsymbol{w}), \quad \forall\,\boldsymbol{y},\boldsymbol{w} \geq 0, \tag{14}$$

sillä jos $0 < a \leq b$, niin

$$a+b-\sqrt{a^2+b^2} = \frac{2ab}{a+b+\sqrt{a^2+b^2}} = \frac{2a}{\frac{a}{b}+1+\sqrt{(\frac{a}{b})^2+1}} \geq \frac{2a}{2+\sqrt{2}}.$$

Samoin mikäli $0 < b \leq a$ niin

$$a+b-\sqrt{a^2+b^2} \geq \frac{2b}{2+\sqrt{2}}.$$

Näistä seuraa epäyhtälö

$$a+b-\sqrt{a^2+b^2} \geq \frac{2}{2+\sqrt{2}} \min(a,b), \quad \forall\,a,b \geq 0.$$

Täten epäyhtälö (14) pätee.

Lisäksi on helppo nähdä, että

$$p_1(\boldsymbol{y},\boldsymbol{w}) \leq m p_3(\boldsymbol{y},\boldsymbol{w}), \quad p_2(\boldsymbol{y},\boldsymbol{w}) \leq m p_4(\boldsymbol{y},\boldsymbol{w}), \quad \forall\,\boldsymbol{y},\boldsymbol{w} \geq 0. \tag{15}$$

Epäyhtälöistä (13)- (15) seuraa

$$d((\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}),\mathcal{F}) \leq t p(\boldsymbol{y},\boldsymbol{w}), \quad \forall\,(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C} \text{ jolla } t = \frac{2+\sqrt{2}}{2} m t_1.$$

Selvästi, jos $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}$, niin $p(\boldsymbol{y},\boldsymbol{w}) \geq 0$ ja funktio f on Lipschitz-jatkuva joukossa $\mathcal{C}$ (koska f on jatkuvasti differoituva ja joukko $\mathcal{C}$ on kompakti). Seurauksena lähteen [7] väitteen 4 mukaan on olemassa positiivinen luku t_0, jolle kaikilla $t \geq t_0$ tehtävät (11) ja (12) ovat ekvivalentit. □

Samantyylinen tulos lauseen 11 sakkofunktiolle p_1 on esitetty lähteessä [9] lauseessa 2.4.3 affiineille tasapainopisterajoitteisille matemaattisille optimointitehtäville. Kahdessa seuraavassa aliluvussa esitellään DCA-algoritmi tehtävälle (12) käyttäen sakkofunktioita (7)–(10).

4.3 Standardeja DCA-tehtäviä sakkofunktioiden noudattaessa L^1-metriikkaa

Kun $p(\boldsymbol{y},\boldsymbol{w}) = p_1(\boldsymbol{y},\boldsymbol{w})$, tehtävä (12) kirjoitetaan standardina DC-tehtävänä:

$$\min\{F_t^1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) := G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) - H_t^1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathbb{R}^{n+2m}\}, \tag{16}$$

missä

$$G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) := g(\boldsymbol{x},\boldsymbol{y}) + \chi_{\mathcal{C}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}),\ H_t^1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) := h(\boldsymbol{x},\boldsymbol{y}) - t\sum_{i=1}^{m}\psi^{\min}(y_i,w_i).$$

Merkintä $\chi_{\mathcal{C}}$ tarkoittaa joukon C *indikaattorifunktiota*, joka määritellään $\chi_{\mathcal{C}}(\boldsymbol{x}) = 0$, jos $\boldsymbol{x} \in C$, ja $+\infty$ muuten.
Kun $p(\boldsymbol{y},\boldsymbol{w}) = p_2(\boldsymbol{y},\boldsymbol{w})$, niin tehtävä (12) voidaan myös ilmaista standardina DC-tehtävänä:

$$\min\{F_t^2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) := G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) - H_t^2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathbb{R}^{n+2m}\}, \tag{17}$$

missä

$$H_t^2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) := h(\boldsymbol{x},\boldsymbol{y}) - t\sum_{i=1}^{m}\psi^{FB}(y_i,w_i).$$

Standardin DC-tehtävän ratkaisuun käytettävän tavallisen DCA-algoritmin mukaan jokainen DCA-algoritmin iteraatiokierros tehtävään (16) (vastaavasti tehtävään (17)) edellyttää aligradientin

$$(\overline{\boldsymbol{x}}^k,\overline{\boldsymbol{y}}^k,\overline{\boldsymbol{w}}^k) \in \partial H_t^j(\boldsymbol{x}^k,\boldsymbol{y}^k,\boldsymbol{w}^k),\ \text{kun } j = 1,2,$$

laskemista, jolloin ratkaistavana on konveksi optimointitehtävä

$$\min\{g(\boldsymbol{x},\boldsymbol{y}) - \langle\overline{\boldsymbol{x}}^k,\boldsymbol{x}\rangle - \langle\overline{\boldsymbol{y}}^k,\boldsymbol{y}\rangle - \langle\overline{\boldsymbol{w}}^k,\boldsymbol{w}\rangle : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}\}. \tag{18}$$

Näin ollen kaksi ensimmäistä DCA-algoritmia tehtävien (16) ja (17) ratkaisemiseen eroavat ainoastaan ensimmäisessä askeleessa.
Aligradientti $(\overline{\boldsymbol{x}},\overline{\boldsymbol{y}},\overline{\boldsymbol{w}}) \in \partial H_t^1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w})$ voidaan laskea seuraavasti:

$$\overline{\boldsymbol{x}} = \nabla_x h(\boldsymbol{x},\boldsymbol{y}), \quad \overline{\boldsymbol{y}} = \nabla_y h(\boldsymbol{x},\boldsymbol{y}) + t\widehat{\boldsymbol{y}}, \quad \overline{\boldsymbol{w}} = t\widehat{\boldsymbol{w}}, \tag{19}$$

missä $(\widehat{y}_i,\widehat{w}_i) \in \partial(-\psi^{\min})(y_i,w_i)$, $i \in \{1,\ldots,m\}$. Koska

$$-\psi^{\min}(y_i,w_i) = \max(-y_i,-w_i),$$

voidaan aligradientti $(\widehat{y}_i,\widehat{w}_i) \in \partial(-\psi^{\min})(y_i,w_i)$ valita seuraavasti

$$\widehat{y}_i = \begin{cases} -1 & \text{, jos } y_i < w_i, \\ 0 & \text{, jos } y_i \geq w_i, \end{cases} \qquad \widehat{w}_i = \begin{cases} 0 & \text{, jos } y_i < w_i, \\ -1 & \text{, jos } y_i \geq w_i. \end{cases} \tag{20}$$

Aligradientti $(\overline{\boldsymbol{x}},\overline{\boldsymbol{y}},\overline{\boldsymbol{w}}) \in \partial H_t^2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w})$ voidaan myös laskea kaavan (19) avulla, missä $(\widehat{y}_i,\widehat{w}_i) \in \partial(-\psi^{FB})(y_i,w_i)$.
Funktion ψ^{FB} määritelmän mukaan

$$-\psi^{FB}(y_i,w_i) = \sqrt{(y_i)^2 + (w_i)^2} - y_i - w_i,$$

minkä takia aligradientti $(\widehat{y}_i,\widehat{w}_i) \in \partial(-\psi^{FB})(y_i,w_i)$ voidaan valita seuraavasti

$$\widehat{y_i} = \begin{cases} \frac{y_i}{\sqrt{(y_i)^2+(w_i)^2}} - 1 & \text{, jos } (y_i, w_i) \neq (0,0), \\ -1 & \text{, jos } y_i = w_i = 0, \end{cases} \tag{21}$$

$$\widehat{w_i} = \begin{cases} \frac{w_i}{\sqrt{(y_i)^2+(w_i)^2}} - 1 & \text{, jos } (y_i, w_i) \neq (0,0), \\ -1 & \text{, jos } y_i = w_i = 0. \end{cases} \tag{22}$$

Seuraavaksi esitetään tehtäviin (16) ja (17) sovellettavat DCA-algoritmit.

DCA1-algoritmi: DCA-algoritmi standardin tehtävän (16) ratkaisemiseen

Alustus: Valitse piste $\boldsymbol{z}^0 := (\boldsymbol{x}^0, \boldsymbol{y}^0, \boldsymbol{w}^0) \in \mathbb{R}^{n+2m}$, sakkoparametri $t > 0$ ja tarpeeksi pieni positiivinen luku ε. Aseta $k := 0$.

Askel 1: Laske $(\overline{\boldsymbol{x}}^k, \overline{\boldsymbol{y}}^k, \overline{\boldsymbol{w}}^k) \in \partial H_t^1(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{w}^k)$ käyttäen kaavoja (19) ja (20).

Askel 2: Laske tehtävän (18) ratkaisu $\boldsymbol{z}^{k+1} := (\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$.

Askel 3: Jos

$$\|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\| \leq \varepsilon(\|\boldsymbol{z}^k\| + 1) \text{ tai } |F_t^1(\boldsymbol{z}^{k+1}) - F_t^1(\boldsymbol{z}^k)| \leq \varepsilon(|F_t^1(\boldsymbol{z}^k)| + 1),$$

niin lopeta. Muuten aseta $k := k + 1$ ja siirry askeleeseen 1.

Seuraavaksi esiteltävä DCA2-algoritmi on samantyylinen DCA1-algoritmin kanssa. Ainut ero on, että Askel 1 on korvattu Askeleella 1a alla ja funktio F_t^1 Askeleessa 3 on korvattu funktiolla F_t^2.

DCA2-algoritmi: DCA-algoritmi standardin tehtävän (17) ratkaisemiseen

Askel 1a: Laske $(\overline{\boldsymbol{x}}^k, \overline{\boldsymbol{y}}^k, \overline{\boldsymbol{w}}^k) \in \partial H_t^2(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{w}^k)$ käyttäen kaavoja (19), (21) ja (22).

Huomautus 2. Hyvän sakkoparametrin t valitseminen on vaikeaa, minkä takia käytännössä sakkoparametrit päivitetään seuraavasti.

DCA1-algoritmi päivitetyllä sakkoparametrilla

Alustus: Valitse piste $\boldsymbol{z}^0 := (\boldsymbol{x}^0, \boldsymbol{y}^0, \boldsymbol{w}^0) \in \mathbb{R}^{n+2m}$, sakkoparametri $t_0 > 0$, parametri $\delta > 1$, yläraja $t_{\max}$ ja tarpeeksi pienet positiiviset luvut $\varepsilon_1, \varepsilon_2$. Aseta $k := 0$.

Askel 1: Laske $(\overline{\boldsymbol{x}}^k, \overline{\boldsymbol{y}}^k, \overline{\boldsymbol{w}}^k) \in \partial H_{t_k}^1(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{w}^k)$ käyttäen kaavoja (19) ja (20).

Askel 2: Laske tehtävän (18) ratkaisu $\boldsymbol{z}^{k+1} := (\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$.

Askel 3: Laske $v_k = p_1(\boldsymbol{y}_{k+1}, \boldsymbol{w}_{k+1})$. Jos

$$\|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\| \leq \varepsilon_1(\|\boldsymbol{z}^k\| + 1) \text{ tai } |F_{t_k}^1(\boldsymbol{z}^{k+1}) - F_{t_k}^1(\boldsymbol{z}^k)| \leq \varepsilon_1(|F_{t_k}^1(\boldsymbol{z}^k)| + 1)$$

ja $v_k \leq \varepsilon_2$, niin lopeta.

Askel 4: Päivitä sakkoparametri.

$$\text{Aseta } t_{k+1} = \begin{cases} t_k & \text{, jos } v_k \le \varepsilon_2, \\ \min(\delta_{t_k}, t_{\max}) & \text{, jos } v_k > \varepsilon_2. \end{cases}$$

Askel 5: Aseta $k := k + 1$ ja siirry askeleeseen 1.

Tämän algoritmin pysäytysehto takaa, että sakkotermin mahdollinen virhe ei ylitä arvoa ε_2.
DCA2-algoritmin tapa päivittää sakkoparametrit toimii samoin kuin DCA1-algoritmin sakkoparametrien päivitys.

Seuraava lause antaa osviittaa DCA1- ja DCA2-algoritmien konvergointiominaisuuksista.

Lause 12. *Seuraavat pätevät:*

(i) DCA1-algoritmi (vast. DCA2) muodostaa joukossa $\mathcal{C}$ lukujonon $\{\boldsymbol{z}^k = (\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{w}^k)\}$, jolla $\{F_t^1(\boldsymbol{z}^k)\}$ (vast. $\{F_t^2(\boldsymbol{z}^k)\}$) on laskeva.

(ii) Jos tehtävän (16) (vast. (17)) optimiarvo on äärellinen ja lukujono $\{\boldsymbol{z}^k\}$ on rajoitettu, niin jokainen lukujonon rajapiste $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on tehtävän (16) (vast. (17)) kriittinen piste.

(iii) Lisäksi, jos $p_1(\boldsymbol{y}^*, \boldsymbol{w}^*) = 0$ (vast. $p_2(\boldsymbol{y}^*, \boldsymbol{w}^*) = 0$, niin $\boldsymbol{z}^*$ on heikosti stationäärinen piste DCLCC-tehtävälle (6). Jos vielä $y_i^* \neq w_i^*$ kaikilla $i \in \{1, 2, \dots, m\}$, niin $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on tehtävän (6) vahvasti stationäärinen piste.

Todistus. Kohdat (i) ja (ii) seuraavat DCA-algoritmien konvergointiominaisuuksista, (katso lause 3 lähteessä [12]). Kohdan (iii) todistus on DCA1- ja DCA2-algoritmeille samanlainen, joten näytetään vain DCA1-algoritmin todistus.
Selvästi $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*) \in \mathcal{C}$. Koska $p_1(\boldsymbol{y}^*, \boldsymbol{w}^*) = 0, (\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on sallittu DCLCC-tehtävälle (6).
Kohdan (ii) mukaan $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on tehtävän (16) kriittinen piste ja siten on olemassa $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}}) \in \partial H_t^1(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$, missä

$$(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}}) \in \partial G(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*).$$

Tästä seuraa, että

$$(\nabla_x h(\boldsymbol{x}^*, \boldsymbol{y}^*), \nabla_y h(\boldsymbol{x}^*, \boldsymbol{y}^*) + t\widehat{\boldsymbol{y}}, t\widehat{\boldsymbol{w}}) \in (\nabla_x g(\boldsymbol{x}^*, \boldsymbol{y}^*), \nabla_y g(\boldsymbol{x}^*, \boldsymbol{y}^*), \boldsymbol{0}) + N_{\mathcal{C}}(\boldsymbol{z}^*),$$

missä $(\widehat{y_i}, \widehat{w_i}) \in \partial(-\psi^{\min})(y_i^*, w_i^*), N_{\mathcal{C}}(\boldsymbol{z}^*)$ on joukon $\mathcal{C}$ normaalikartio eli

$$N_{\mathcal{C}}(\boldsymbol{z}^*) = \partial \chi_{\mathcal{C}}(\boldsymbol{z}^*) = \{\boldsymbol{u} \in \mathbb{R}^{n+2m} : \langle \boldsymbol{u}, \boldsymbol{z} - \boldsymbol{z}^* \rangle \le 0, \forall\, \boldsymbol{z} \in \mathcal{C}\}.$$

Täten

$$\boldsymbol{0} \in (\nabla_x g(\boldsymbol{x}^*, \boldsymbol{y}^*), \nabla_y g(\boldsymbol{x}^*, \boldsymbol{y}^*), \boldsymbol{0}) - (\nabla_x h(\boldsymbol{x}^*, \boldsymbol{y}^*), \nabla_y h(\boldsymbol{x}^*, \boldsymbol{y}^*) + t\widehat{\boldsymbol{y}}, t\widehat{\boldsymbol{w}}) + N_{\mathcal{C}}(\boldsymbol{z}^*).$$

Tästä seuraa, että $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on optimiratkaisu tehtävälle

$$\begin{aligned}\min\{g(\boldsymbol{x}, \boldsymbol{y}) - \langle \nabla_x h(\boldsymbol{x}^*, \boldsymbol{y}^*), \boldsymbol{x}\rangle - \langle \nabla_y h(\boldsymbol{x}^*, \boldsymbol{y}^*) + t\widehat{\boldsymbol{y}}, \boldsymbol{y}\rangle - \langle t\widehat{\boldsymbol{w}}, \boldsymbol{w}\rangle :\\ A\boldsymbol{x} + B\boldsymbol{y} + \mathbf{a} \le \mathbf{0}, \quad N\boldsymbol{x} + M\boldsymbol{y} + \mathbf{q} = \boldsymbol{w}, \quad \boldsymbol{y} \ge \mathbf{0}, \boldsymbol{w} \ge \mathbf{0}\}.\end{aligned}$$

Tällöin on olemassa sopivasti valitut kertoimet $\boldsymbol{\beta}, \overline{\boldsymbol{v}}^y, \overline{\boldsymbol{v}}^w$, joilla

$$\nabla_x f(\boldsymbol{x}^*, \boldsymbol{y}^*) + A^T\boldsymbol{\beta} - N^T(t\widehat{\boldsymbol{w}} + \overline{\boldsymbol{v}}^w) = \mathbf{0}, \tag{23}$$

$$\nabla_y f(\boldsymbol{x}^*, \boldsymbol{y}^*) + B^T\boldsymbol{\beta} - M^T(t\widehat{\boldsymbol{w}} + \overline{\boldsymbol{v}}^w) - t\widehat{\boldsymbol{y}} - \overline{\boldsymbol{v}}^y = \mathbf{0}, \tag{24}$$

$$\boldsymbol{\beta} \ge \mathbf{0}, \langle \boldsymbol{\beta}, A\boldsymbol{x}^* + B\boldsymbol{y}^* + \mathbf{a}\rangle = 0, \tag{25}$$

$$\overline{v}_i^y \ge 0, \quad \overline{v}_i^y y_i^* = 0; \quad \overline{v}_i^w \ge 0, \quad \overline{v}_i^w w_i^* = 0, \quad i = 1, \dots, m. \tag{26}$$

Asetetaan $\boldsymbol{v}^y = \overline{\boldsymbol{v}}^y + t\widehat{\boldsymbol{y}}, \boldsymbol{v}^w = \overline{\boldsymbol{v}}^w + t\widehat{\boldsymbol{w}}$.
Kaikille $i \in I_w(\boldsymbol{y}^*, \boldsymbol{w}^*)$ pätee $y_i^* > w_i^* = 0$. Täten $\widehat{y}_i = 0$ ja $\overline{v}_i^y = 0$. Samoin $\widehat{w}_i = 0$ ja $\overline{v}_i^w = 0$, kaikilla $i \in I_y(\boldsymbol{y}^*, \boldsymbol{w}^*)$.
Tällöin

$$\nabla_x f(\boldsymbol{x}^*, \boldsymbol{y}^*) + A^T\boldsymbol{\beta} - N^T\boldsymbol{v}^w = \mathbf{0}, \tag{27}$$

$$\nabla_y f(\boldsymbol{x}^*, \boldsymbol{y}^*) + B^T\boldsymbol{\beta} - M^T\boldsymbol{v}^w - \boldsymbol{v}^y = \mathbf{0}, \tag{28}$$

$$v_i^y = 0,\ i \in I_w(\boldsymbol{y}^*, \boldsymbol{w}^*); \quad v_i^w = 0,\ i \in I_y(\boldsymbol{y}^*, \boldsymbol{w}^*). \tag{29}$$

Kohdista (25) ja (27)-(29) seuraa, että $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on DCLCC-tehtävän (6) heikosti stationäärinen piste.
Jos $y_i^* \neq w_i^*$ kaikilla $i \in \{1, 2, \dots, m\}$, niin $I_0(\boldsymbol{y}^*, \boldsymbol{w}^*) = \emptyset$. Tästä seuraa, että $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on optimointitehtävän (6) vahvasti stationäärinen piste. □

Kun f on konveksi, niin olkoon $g = f, h = 0$, jolloin H_t^1 on konveksi ja polyhedraalinen. Täten (16) on polyhedraali DC-tehtävä. DCA-algoritmin äärellisen konvergenssin, joka on todettu lähteessä [8], sekä lauseen 12 seurauksina saadaan polyhedraalisille DCA-algoritmeille seuraavat tulokset.

Seuraus 1. *Olkoon f jatkuvasti differentioituva konveksi funktio ja tehtävän (16) optimiarvo äärellinen. Tällöin seuraavat väittämät pätevät.*

(i) DCA1-algoritmi muodostaa lukujonon $\left\{\boldsymbol{z}^k = (\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{w}^k)\right\}$, jossa on äärellisen monta joukon $\mathcal{C}$ eri alkiota siten, että lukujono $\{F_t^1(\boldsymbol{z}^k)\}$ on laskeva. Täten DCA1-algoritmi päättyy äärellisen monen iteraatiokerran jälkeen.

(ii) Oletetaan, että DCA1-algoritmi päättyy pisteessä $\boldsymbol{z}^{k+1} = (\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$. Silloin $\boldsymbol{z}^{k+1}$ on tehtävän (16) kriittinen piste.

(iii) Lisäksi, jos $p_1(\boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1}) = 0$, niin $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$ on DCLCC-tehtävän (6) heikosti stationäärinen piste. Jos lisäksi $y_i^{k+1} \neq w_i^{k+1}$, kaikilla $i \in \{1, \dots, m\}$, niin $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$ on tehtävän (6) lokaali optimi.

4.4 Sakkofunktiot, jotka noudattavat L^∞-metriikkaa

Kun $p(\boldsymbol{y},\boldsymbol{w}) = p_3(\boldsymbol{y},\boldsymbol{w})$, niin tehtävä (12) voidaan myös kirjoittaa standardin DC-tehtävän muodossa. Tällöin DC-tehtävän hajotelma $f(\boldsymbol{x},\boldsymbol{y}) + tp_3(\boldsymbol{y},\boldsymbol{w})$ voidaan valita kuten lähteessä [12] seuraavasti

$$f(\boldsymbol{x},\boldsymbol{y}) + tp_3(\boldsymbol{y},\boldsymbol{w}) = G_t(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) - H_t(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}),$$

missä

$$\begin{aligned} G_t(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) &= g(\boldsymbol{x},\boldsymbol{y}) + t\max_i \left\{ -\max_{j\neq i} \psi^{\min}(y_j, w_j) \right\}, \\ H_t(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) &= h(\boldsymbol{x},\boldsymbol{y}) - t\max_i \psi^{\min}(y_i, w_i). \end{aligned}$$

Kuitenkin tällainen hajotelma johtaa vaikeisiin aliongelmiin, koska tällöin aliongelmien kohdefunktioissa on maksimissaan m epäsileää funktiota. Tämän vaikeuden ylitsepääsemiseksi muunnetaan tehtävä (12) ekvivalenttiin muotoon

$$\min\{f(\boldsymbol{x},\boldsymbol{y}) + ts : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}, \psi^{\min}(y_i, w_i) \leq s, i = 1,\ldots,m;\ s \geq 0\}.$$

Tehtävä voidaan edelleen muokata seuraavaan yleisen DC-tehtävän muotoon:

$$\begin{aligned} \min\{F_t^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) :=& G_i^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) - H^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}, s \geq 0, \\ & - s - [-\psi^{\min}(y_i, w_i)] \leq 0, i = 1,\ldots,m\}, \end{aligned} \tag{30}$$

missä $G_t^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) := g(\boldsymbol{x},\boldsymbol{y}) + ts, H^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) := h(\boldsymbol{x},\boldsymbol{y})$.

Luvussa 3.2 esiteltiin tavallinen DCA-algoritmi. Sitä soveltamalla yleisille DC-tehtäville voidaan muodostaa DCA-algoritmi, joka ratkaisee tehtävän (30). Tällöin jokaisella iteraatiolla k lasketaan

$$(\overline{y}_i^k, \overline{w}_i^k) \in \partial(-\psi^{\min})(y_i^k, w_i^k), \quad i = 1,\ldots,m,$$

jonka jälkeen ratkaistaan konveksi aliongelma

$$\begin{aligned} \min\{g(\boldsymbol{x},\boldsymbol{y}) + ts - \langle \nabla_x h(\boldsymbol{x}^k,\boldsymbol{y}^k), \boldsymbol{x}\rangle - \langle \nabla_y h(\boldsymbol{x}^k,\boldsymbol{y}^k), \boldsymbol{y}\rangle : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}, s \geq 0, \\ -s + \psi^{\min}(y_i^k, w_i^k) - \overline{y}_i^k(y_i - y_i^k) - \overline{w}_i^k(w_i - w_i^k) \leq 0, i = 1,\ldots,m\}. \end{aligned} \tag{31}$$

Seuraavaksi esitellään yleinen DCA-algoritmi, jota on sovellettu tehtävään (30).

DCA3-algoritmi: DCA-algoritmi yleisen tehtävän (30) ratkaisemiseen

Alustus: Valitse piste $X^1 := (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{w}^1,s^1) \in \mathbb{R}^{n+2m+1}$, sakkoparametri $t > 0$ ja tarpeeksi pieni positiivinen luku ε. Aseta $k := 1$.

Askel 1: Laske $(\overline{y}_i^k, \overline{w}_i^k) \in \partial(-\psi^{\min})(y_i^k, w_i^k)$, kaikilla $i = 1,\ldots,m$ käyttäen kaavaa (20).

Askel 2: Ratkaise tehtävä (31), jolloin saadaan $X^{k+1} := (\boldsymbol{x}^{k+1},\boldsymbol{y}^{k+1},\boldsymbol{w}^{k+1},s^{k+1})$.

Askel 3: Jos

$$\|X^{k+1} - X^k\| \le \varepsilon(\|X^k\| + 1) \text{ tai } |F_t^3(X^{k+1}) - F_t^3(X^k)| \le \varepsilon(|F_t^3(X^k)| + 1),$$

niin lopeta. Muulloin aseta $k := k + 1$ ja siirry askeleeseen 1.

Seuraavaksi esitellään algoritmi sille, miten DCA3-algoritmissa voidaan päivittää sakkoparametrit. Tämä seuraa luvun 3.3 algortimia.

DCA3-algoritmi päivitetyillä sakkoparametreilla

Alustus: Valitse piste $X^1 := (\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{w}^1, s^1) \in \mathbb{R}^{n+2m+1}$, sakkoparametri $t_1 > 0$, parametrit $\delta > 1, \delta_1 > 0$, yläraja $t_{\max}$ sekä tarpeeksi pienet positiiviset luvut ε_1 ja ε_2. Aseta $k := 1$.

Askel 1: Laske $(\overline{y}_i^k, \overline{w}_i^k) \in \partial(-\psi^{\min})(y_i^k, w_i^k)$, kaikilla $i = 1, \ldots, m$, käyttäen kaavaa (20).

Askel 2: Ratkaise tehtävä (31), jolloin saadaan $X^{k+1} := (\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1}, s^{k+1})$ ja Lagrangen kertoimet λ_i^{k+1} kaikille $i = 1, \ldots, m$.

Askel 3: Jos

$$\|X^{k+1} - X^k\| \le \varepsilon_1(\|X^k\| + 1) \text{ tai } |F_{t_k}^3(X^{k+1}) - F_{t_k}^3(X^k)| \le \varepsilon_1(|F_{t_k}^3(X^k)| + 1)$$

ja $s^{k+1} \le \varepsilon_2$, niin lopeta.

Askel 4: Päivitä sakkoparametri laskemalla

$$r_k = \min\left(\|X^{k+1} - X^k\|^{-1}, \sum_{i=1}^{m} |\lambda_i^{k+1}| + \delta_1 \right),$$

ja aseta

$$t_{k+1} = \begin{cases} t_k & \text{, jos } t_k \ge r_k, \\ \min(\delta_{t_k}, t_{\max}) & \text{, jos } t_k < r_k. \end{cases}$$

Askel 5: Aseta $k := k + 1$ ja siirry askeleeseen 1.

Vastaavasti kuin $p(\boldsymbol{y}, \boldsymbol{w}) = p_4(\boldsymbol{y}, \boldsymbol{w})$, voidaan tehtävä (12) kirjoittaa yleisen DC-tehtävän muotoon

$$\begin{aligned} \min\{F_t^4(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}, s) =& G_t^3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}, s) - H^3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}, s) : (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}, s \ge 0, \\ & - s - [-\psi^{FB}(y_i, w_i)] \le 0, i = 1, \ldots, m\}. \end{aligned} \tag{32}$$

Seuraava algoritmi esittää sen, miten DCA-algoritmia voidaan soveltaa tehtävään (32). Algoritmi on vastaavanlainen DCA3-algoritmin kanssa, mutta askeleet 1 ja 2 on korvattu askeleilla 1a ja 2a.

DCA4-algoritmi: DCA-algoritmi yleisen tehtävän (32) ratkaisemiseen

Askel 1a: Laske $(\overline{y}_i^k, \overline{w}_i^k) \in \partial(-\psi^{FB})(y_i^k, w_i^k)$, missä $i = 1, \ldots, m$ käyttäen kaavoja (21) ja (22).

Askel 2a: Laske konveksin tehtävän

$$\min\{g(\boldsymbol{x}, \boldsymbol{y}) + ts - \langle \nabla_x h(\boldsymbol{x}^k, \boldsymbol{y}^k), \boldsymbol{x}\rangle - \langle \nabla_y h(\boldsymbol{x}^k, \boldsymbol{y}^k), \boldsymbol{y}\rangle : (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}, s \geq 0,$$
$$- s + \psi^{FB}(y_i^k, w_i^k) - \overline{y}_i^k(y_i - y_i^k) - \overline{w}_i^k(w_i - w_i^k) \leq 0, i = 1, \ldots, m\}$$

optimiratkaisu $X^{k+1} := (\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1}, s^{k+1})$ ja Lagrangen kertoimet λ_i^{k+1} kaikille $i = 1, \ldots, m$.

Huomautus 3. Käytännössä sakkoparametrit päivitetään DCA4-algoritmissa kuten DCA3-algoritmissa.

DCA3- ja DCA4-algoritmien konvergenssiominaisuudet tulevat ilmi seuraavassa lauseessa.

Lause 13. *Oletetaan, että joko g tai h ovat vahvasti konveksi ja tehtävän (30) (vast. (32)) optimiratkaisut ovat äärellisiä. Olkoon $\{X^k = (\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{w}^k, s^k)\}$ DCA3-algoritmin (vast. DCA4) muodostama lukujono. Tällöin seuraavat pätevät:*

(i) $\{F_t^3(X^k)\}$ on laskeva.

(ii) Jos lukujono $\{\boldsymbol{x}^k\}$ on rajoitettu ja $X^* = (\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*, s^*)$ on lukujonon $\{X^k\}$ rajapiste, jolla $s^* = 0$, niin $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on DCLCC-tehtävän (6) heikosti stationäärinen piste. Jos lisäksi $y_i^* \neq w_i^*$ kaikilla $i \in \{1, 2, \ldots, m\}$, niin $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{w}^*)$ on tehtävän (6) vahvasti stationäärinen piste.

Todistus. Lause on todistettu lähteessä [8]. □

Huomautus 4. Mikäli g ja h eivät kumpikaan ole vahvasti konvekseja, niin funktio g voidaan korvata funktiolla $g + \frac{1}{2}\|\cdot\|^2$ ja funktio h funktiolla $h + \frac{1}{2}\|\cdot\|^2$. Tällöin saadaan vahvasti konveksit funktiot.

Kun f on konveksi, $g = f, h = 0$, niin DCA3-algoritmin äärellinen konvergoituvuus voidaan osoittaa seuraavalla tuloksella.

Seuraus 2. *Oletetaan, että f on jatkuvasti differentioituva konveksi funktio ja tehtävän (30) optimiarvo on äärellinen. Tällöin seuraavat pätevät:*

(i) DCA3-algoritmi muodostaa lukujonon $\left\{X^k = (\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{w}^k, s^k)\right\}$, jossa on äärellisen monta eri alkiota siten, että lukujono $\{F_t^3(X^k)\}$ on laskeva. Täten DCA3-algoritmi päättyy äärellisen monen iteraatiokerran jälkeen.

(ii) Oletetaan että DCA3-algoritmi päättyy pisteessä $X^{k+1} = (\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1}, s^{k+1})$ toteuttaen ehdon $s^{k+1} = 0$. Silloin $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$ on DCLCC-tehtävän (6) heikosti stationäärinen piste. Lisäksi, jos $y_i^{k+1} \neq w_i^{k+1}$ kaikilla $i \in \{1, \ldots, m\}$, niin $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$ on tehtävän (6) lokaali optimi.

Todistus.

(i) Koska funktio f on konveksi, aliongelma (31) saadaan muotoon

$$\begin{aligned} &\min\{f(\boldsymbol{x},\boldsymbol{y}) + ts : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}, s \geq 0, \\ &- s + \psi^{\min}(y_i^k, w_i^k) - \overline{y}_i^k(y_i - y_i^k) - \overline{w}_i^k(w_i - w_i^k) \leq 0, i = 1, \ldots, m\}. \end{aligned} \tag{33}$$

Asetetaan $I_k = \{i \in \{1, \ldots, m\} : y_i^k < w_i^k\}$ ja huomataan, että

$$\psi^{\min}(y_i^k, w_i^k) + \overline{y}_i^k y_i^k + \overline{w}_i^k w_i^k = 0.$$

Tällöin tehtävä (33) voidaan kirjoittaa muodossa

$$\min\{f(\boldsymbol{x},\boldsymbol{y}) + ts : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}; y_i \leq s, i \in I_k; w_i \leq s, i \notin I_k; s \geq 0\}. \tag{34}$$

Selvästi $\{F_t^3(X^k)\}$ on laskeva, koska X^{k+1} on tehtävän (33) optimiratkaisu, kun X^k on tehtävän (33) sallittu piste.
Jokaiselle I_k tehtävän (34) ratkaisu on X^{k+1}. Koska alijoukkojen I_k määrä kaikille $\{1, \ldots, m\}$ on äärellinen, lukujonolla $\{X^k\}$ on vain äärellisen monta eri alkiota. Tämä johtaa siihen, että äärellisen monen iteraatiokerran jälkeen $F_t^3(X^k) = F_t^3(X^{k+1})$. Siis kohta (i) on todistettu.

(ii) on seuraus lauseesta 13 ja huomautuksesta 1.

□

Vertailtaessa näitä neljää DCA-algoritmia voidaan tehdä muutamia arvioita niiden suorituskyvystä. Standardit DCA-algoritmit tehtävälle (12) ovat samanlaisia ja niillä on sama ensimmäinen DC-komponentti. Toisaalta toinen DC-komponentti on joko $\psi^{\min}$ tai ψ^{FB}. Yleiset DC-tehtävän formuloinnit tehtävälle (12) ovat myös samanlaisia, koska näillä tehtävillä on samat kohdefunktion DC-hajotelmat sekä samat ensimmäiset DC-komponentit DC-tehtävän rajoitteissa. Täten suoritusajan pitäisi olla nopeampi DCA1-algoritmilla (vast. DCA3) kuin DCA2-algoritmilla (vast. DCA4). Tämä johtuu siitä, että aligradientin $-\psi^{\min}$ laskeminen vie vähemmän aikaa kuin aligradientin $-\psi^{FB}$. Lisäksi DCA1-algoritmin (vast. DCA2) voidaan arvioida olevan nopeampi kuin DCA3-algoritmin (vast. DCA4). Tämä johtuu siitä, että yleisen DCA-algoritmin jokaisessa aliongelmassa on $m+1$ kappaletta enemmän rajoitteita kuin standardin DCA-algoritmin aliongelmassa. Tällöin kohdefunktiot saadaan samanlaisiksi (summa $g(\boldsymbol{x},\boldsymbol{y})$ + lineaarinen funktio). Nämä havainnot on varmistettu myös laskennallisissa tarkasteluissa [8].

5 Soveltaminen

Tässä luvussa käsitellään sitä, miten esiteltyjä algoritmeja voidaan käyttää QPLCC-(*Quadratic Problem with Linear Complementarity Constraints*) ja EiCP-tehtävien (*asymmetric Eigenvalue Complementarity Problem*) ratkaisemiseen.

5.1 QPLCC-tehtävä

Tarkastellaan erityistapausta DCLCC-tehtävästä (6), missä funktio f on muotoa

$$f(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\langle(\boldsymbol{x}, \boldsymbol{y}), P(\boldsymbol{x}, \boldsymbol{y})\rangle + \langle(\boldsymbol{c}, \boldsymbol{d}), (\boldsymbol{x}, \boldsymbol{y})\rangle$$

oleva kvadraattinen funktio.
Tässä matriisi $P \in \mathbb{R}^{(n+m)\times(n+m)}$ on symmetrinen, $\boldsymbol{c} \in \mathbb{R}^n$ ja $\boldsymbol{d} \in \mathbb{R}^m$. Tarkastellaan DC-hajotelmaa $f(\boldsymbol{x}, \boldsymbol{y}) = g(\boldsymbol{x}, \boldsymbol{y}) - h(\boldsymbol{x}, \boldsymbol{y})$, missä

$$g(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\langle(\boldsymbol{x}, \boldsymbol{y}), (P + \rho I)\rangle(\boldsymbol{x}, \boldsymbol{y}) + \langle(\boldsymbol{c}, \boldsymbol{d}), (\boldsymbol{x}, \boldsymbol{y})\rangle, \quad h(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\rho(\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2).$$

Tässä $\rho = 0$, jos P on positiivisemidefiniitti, muulloin $-\lambda_{\min}(P) + 0.001$ ($\lambda_{\min}(P)$ merkitsee matriisin P pienintä ominaisarvoa). DC-kaavat sakkotehtävään (12) neljän vastaavan sakkofunktion osalta on esitetty taulukossa 1.
Lisäksi jokaisen iteraatiokierroksen k kaksi askelta kustakin DCA-algoritmista on esitetty alla.

QP-DCA1

1. Laske $(\overline{\boldsymbol{x}}^k, \overline{\boldsymbol{y}}^k, \overline{\boldsymbol{w}}^k) \in \partial H_t^1(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{w}^k)$ kaavoilla:
$$\overline{\boldsymbol{x}}^k = \rho\boldsymbol{x}^k, \overline{\boldsymbol{y}}^k + t\widehat{\boldsymbol{y}^k}, \overline{\boldsymbol{w}}^k = t\widehat{\boldsymbol{w}^k}, \tag{35}$$
missä $\widehat{y^k}_i$ ja $\widehat{w^k}_i$ lasketaan käyttäen kaavaa (20).

2. Laske kvadraattisen tehtävän (QPLC) ratkaisu $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$ lineaarisilla rajoitteella:
$$\begin{aligned} &\min \frac{1}{2}\langle(\boldsymbol{x}, \boldsymbol{y}), (P + \rho I)\rangle(\boldsymbol{x}, \boldsymbol{y}) + \langle(c - \overline{\boldsymbol{x}}^k, d - \overline{\boldsymbol{y}}^k), (\boldsymbol{x}, \boldsymbol{y})\rangle - \langle(\overline{\boldsymbol{w}}^k), \boldsymbol{w}\rangle \\ &s.t.\ (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}. \end{aligned} \tag{36}$$

QP-DCA2

1. Laske $(\overline{\boldsymbol{x}}^k, \overline{\boldsymbol{y}}^k, \overline{\boldsymbol{w}}^k) \in \partial H_t^2(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{w}^k)$ kaavan (35) mukaan, missä $\widehat{\boldsymbol{y}^k}_i$ ja $\widehat{\boldsymbol{w}^k}_i$ lasketaan kaavoilla (21) ja (22).

2. Laske $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$ ratkaisemalla tehtävä (36).

QP-DCA3

1. Laske $(\overline{y}_i^k, \overline{w}_i^k) \in \partial(-\psi^{\min})(y_i^k, w_i^k)$ kaikilla $i = 1, \ldots, m$ käyttäen kaavaa (20).

2. Laske $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1}, s^{k+1})$ ratkaisemalla tehtävä:
$$\begin{aligned} &\min \frac{1}{2}\langle(\boldsymbol{x}, \boldsymbol{y}), (P + \rho I)\rangle(\boldsymbol{x}, \boldsymbol{y}) + ts + \langle(c - \rho\boldsymbol{x}^k, d - \rho\boldsymbol{y}^k), (\boldsymbol{x}, \boldsymbol{y})\rangle \\ &s.t.\ (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}, \quad s \geq 0, \\ &\psi^{\min}(y_i^k, w_i^k) - \overline{y}_i^k(y_i - y_i^k) - \overline{w}_i^k(w_i - w_i^k) - s \leq 0, \quad i = 1, \ldots, m. \end{aligned} \tag{37}$$

QP-DCA4

1. Laske $(\overline{y}_i^k, \overline{w}_i^k) \in \partial(-\psi^{FB})(y_i^k, w_i^k)$ kaikilla $i = 1, \ldots, m$ käyttäen kaavoja (21) ja (22).

2. Laske $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1}, s^{k+1})$ ratkaisemalla tehtävä:

$$\begin{aligned} &\min \frac{1}{2}\langle(\boldsymbol{x},\boldsymbol{y}),(P+\rho I)\rangle(\boldsymbol{x},\boldsymbol{y}) + ts + \langle(c-\rho\boldsymbol{x}^k, d-\rho\boldsymbol{y}^k),(\boldsymbol{x},\boldsymbol{y})\rangle \\ &\quad s.t.\ (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}, \quad s \geq 0, \\ &\psi^{FB}(y_i^k, w_i^k) - \overline{y}_i^k(y_i - y_i^k) - \overline{w}_i^k(w_i - w_i^k) - s \leq 0, \quad i = 1,\ldots,m. \end{aligned} \tag{38}$$

Taulukko 1: DC-formuloinnit QPLCC-tehtävälle

Sakko-funktiot	**DC kaavat**
$p_1(\boldsymbol{y},\boldsymbol{w})$	$\min\{F_t^1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) = G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) - H_t^1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathbb{R}^{n+2m}\}$, $G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) = \frac{1}{2}\langle(\boldsymbol{x},\boldsymbol{y}),(P+\rho I)\rangle(\boldsymbol{x},\boldsymbol{y}) + \langle(\boldsymbol{c},\boldsymbol{d}),(\boldsymbol{x},\boldsymbol{y})\rangle + \chi_{\mathcal{C}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w})$, $H_t^1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) = \frac{1}{2}\rho(\lVert\boldsymbol{x}\rVert^2 + \lVert\boldsymbol{y}\rVert^2) - t\sum_{i=1}^m \psi^{\min}(y_i, w_i)$
$p_2(\boldsymbol{y},\boldsymbol{w})$	$\min\{F_t^2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) = G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) - H_t^2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) : (\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathbb{R}^{n+2m}\}$, $H_t^2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) = \frac{1}{2}\rho(\lVert\boldsymbol{x}\rVert^2 + \lVert\boldsymbol{y}\rVert^2) - t\sum_{i=1}^m \psi^{FB}(y_i, w_i)$
$p_3(\boldsymbol{y},\boldsymbol{w})$	$\min\{F_t^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) = G_t^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) - H^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s)$: $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}; -s - [-\psi^{\min}(y_i,w_i)] \leq 0, i = 1,\ldots,m; s \geq 0\}$, $G_t^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) = \frac{1}{2}\langle(\boldsymbol{x},\boldsymbol{y}),(P+\rho I)\rangle(\boldsymbol{x},\boldsymbol{y}) + \langle(\boldsymbol{c},\boldsymbol{d}),(\boldsymbol{x},\boldsymbol{y})\rangle + ts$, $H^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) = \frac{1}{2}\rho(\lVert\boldsymbol{x}\rVert^2 + \lVert\boldsymbol{y}\rVert^2)$
$p_4(\boldsymbol{y},\boldsymbol{w})$	$\min\{F_t^4(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) = G_t^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s) - H^3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w},s)$: $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}; -s - [-\psi^{FB}(y_i,w_i)] \leq 0, i = 1,\ldots,m; s \geq 0\}$,

5.2 EiCP-tehtävä

Olkoon A-matriisi n-kertainen neliömatriisi ja B-matriisi $n \times n$-kokoinen symmetrinen positiividefiniitti matriisi. CP-tehtävässä (*Complementarity Problem*) tavoitteena on löytää reaaliluku $\lambda > 0$ ja vektori $\boldsymbol{y} \in \mathbb{R}^n \backslash \{\boldsymbol{0}\}$, jolla pätee

$$\boldsymbol{w} = (\lambda B - A)\boldsymbol{y}, \quad \boldsymbol{w} \geq \boldsymbol{0}, \quad \boldsymbol{y} \geq \boldsymbol{0}, \quad \langle\boldsymbol{y},\boldsymbol{w}\rangle = 0.$$

Mille tahansa EiCP-tehtävän ratkaisulle $(\lambda, \boldsymbol{y})$, λ on matriisin (A, B) komplementaarinen ominaisarvo ja $\boldsymbol{y}$ on vastaava komplementaarinen ominaisvektori. Kuten lähteessä [4] mainitaan, matriisin A luokka on erittäin oleellisessa osassa EiCP-tehtävän ratkaisemisessa. Kun matriisi A on symmetrinen, on EiCP-tehtävä samanlainen tehtävä kuin sellainen tehtävä, jossa simplexillä etsitään Rayleightin funktion stationääristä pistettä [4]. Kuitenkaan tämä ei päde silloin, kun matriisi A on epäsymmetrinen. Tässä sovelluksessa oletetaan, että matriisi A on epäsymmetrinen. Olkoot

$$v = \frac{1}{\lambda} \text{ ja } \boldsymbol{u} = v\boldsymbol{y}.$$

Tällöin EiCP-tehtävä voidaan muuttaa *matemaattiseksi optimointitehtäväksi lineaarisilla komplementaarisilla rajoitteilla* eli MPLCC-tehtäväksi

$$\begin{aligned} \min_{\boldsymbol{u},v,\boldsymbol{y},\boldsymbol{w}} \{ & f(\boldsymbol{u}, v, \boldsymbol{y}) := \|\boldsymbol{u} - v\boldsymbol{y}\|^2 : \boldsymbol{w} = B\boldsymbol{y} - A\boldsymbol{u}, \langle \boldsymbol{e}, \boldsymbol{y} \rangle = 1, \\ & l_b \leq v \leq u_b, \boldsymbol{u} \geq \boldsymbol{0}; \boldsymbol{y} \geq \boldsymbol{0}, \boldsymbol{w} \geq \boldsymbol{0}, \langle \boldsymbol{y}, \boldsymbol{w} \rangle = 0 \}, \end{aligned} \tag{39}$$

missä $\boldsymbol{e}$ merkitsee yksikkövektoria, $l_b = \frac{\lambda_{\min}(B)}{\lambda_{\max}(D)}, D = \frac{1}{2}(A + A^T), u_b$ on lineaarisen tehtävän

$$\max_{\boldsymbol{u},v,y} \{ v : B\boldsymbol{y} - A\boldsymbol{u} \geq \boldsymbol{0}, \langle \boldsymbol{e}, \boldsymbol{y} \rangle = 1, \langle \boldsymbol{e}, \boldsymbol{u} \rangle = v, \boldsymbol{u} \geq \boldsymbol{0}, \boldsymbol{y} \geq \boldsymbol{0}, v \geq 0 \}$$

optimiarvo.
Mikäli maksimointitehtävä on rajoittamaton, voidaan u_b olettaa tarpeeksi suureksi positiiviseksi luvuksi. Tällöin $(\lambda, \boldsymbol{y})$ on EiCP-tehtävän ratkaisu, jos ja vain jos $(\frac{\boldsymbol{y}}{\lambda}, \frac{1}{\lambda}, \boldsymbol{y}, \boldsymbol{w})$ on MPLCC-tehtävän (39) ratkaisu kohdefunktion arvolla nolla.
Olkoot $\rho_1 = \max\{4u_b, 2u_b + 2\}$ ja $\rho_2 = \max\{2u_b^2 + 4u_b, 4u_b + 2\}$. Lähteen [11] perusteella valitaan seuraava DC-hajotelma funktiolle f:

$$f(\boldsymbol{u}, v, \boldsymbol{y}) = g(\boldsymbol{u}, v, \boldsymbol{y}) - h(\boldsymbol{u}, v, \boldsymbol{y}),$$

missä

$$\begin{aligned} g(\boldsymbol{u}, v, \boldsymbol{y}) &= \left(\frac{\rho_1}{2} + 1\right)\|\boldsymbol{u}\|^2 + \frac{\rho_1 + \rho_2}{2} v^2 + \frac{\rho_1 + \rho_2}{2}\|\boldsymbol{y}\|^2, \\ h(\boldsymbol{u}, v, \boldsymbol{y}) &= g(\boldsymbol{u}, v, \boldsymbol{y}) - f(\boldsymbol{u}, v, \boldsymbol{y}). \end{aligned}$$

Täten tehtävä (39) on samassa muodossa kuin tehtävä (6), missä $\boldsymbol{x} = (\boldsymbol{u}, v)$ ja lineaariset rajoitteet ovat

$$\mathcal{C} = \left\{ (\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) \in \mathbb{R}^{3n+1} : \substack{\boldsymbol{w} = B\boldsymbol{y} - A\boldsymbol{u},\ \langle \boldsymbol{e}, \boldsymbol{y} \rangle = 1,\ l_b \leq v \leq u_b \\ \boldsymbol{u} \geq \boldsymbol{0},\ \boldsymbol{y} \geq \boldsymbol{0},\ \boldsymbol{w} \geq \boldsymbol{0}} \right\}.$$

DC-funktiot sakkotehtävien (12) vastaaville sakkofunktioille on koottu taulukkoon 2.
Jokaisen iteraatiokierroksen k kaksi askelta vastaavasta DCA-algoritmista kuvaillaan seuraavaksi.

Ei-DCA1

1. Laske $(\overline{\boldsymbol{u}}^k, \overline{v}^k, \overline{\boldsymbol{y}}^k, \overline{\boldsymbol{w}}^k) \in \partial H_t^1(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k, \boldsymbol{w}^k)$ käyttämällä kaavoja

$$\overline{\boldsymbol{u}}^k = \nabla_u h(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k), \quad \overline{v}^k = \frac{\partial}{\partial v} h(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k), \tag{40}$$

$$\overline{\boldsymbol{y}}^k = \nabla_y h(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k) + t\widehat{\boldsymbol{y}^k}, \quad \overline{\boldsymbol{w}}^k = t\widehat{\boldsymbol{w}^k}, \tag{41}$$

missä

$$\nabla_u h(\boldsymbol{u}, v, \boldsymbol{y}) = \rho_1 \boldsymbol{u} + 2v\boldsymbol{y}, \tag{42}$$

$$\frac{\partial}{\partial v} h(\boldsymbol{u}, v, \boldsymbol{y}) = (\rho_1 + \rho_2)v + 2\langle \boldsymbol{y}, \boldsymbol{u}\rangle - 2v\|\boldsymbol{y}\|^2 \tag{43}$$

$$\nabla_y h(\boldsymbol{u}, v, \boldsymbol{y}) = (\rho_1 + \rho_2)\boldsymbol{y} + 2v\boldsymbol{u} - 2v^2\boldsymbol{y}. \tag{44}$$

Tässä $\widehat{y^k}_i$ ja $\widehat{w^k}_i$ lasketaan kaavalla (20).

2. Laske konveksin kvadraattisen tehtävän

$$\begin{aligned} &\min g(\boldsymbol{u}, v, \boldsymbol{y}) - \langle(\overline{\boldsymbol{u}}^k), \boldsymbol{u}\rangle - \overline{v}^k v - \langle(\overline{\boldsymbol{y}}^k), \boldsymbol{y}\rangle - \langle(\overline{\boldsymbol{w}}^k), \boldsymbol{w}\rangle \\ &s.t. \quad (\boldsymbol{u}^k, v^k, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}. \end{aligned} \tag{45}$$

optimiratkaisu $(\boldsymbol{u}^{k+1}, v^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$.

Ei-DCA2

1. Laske $(\overline{\boldsymbol{u}}^k, \overline{v}^k, \overline{\boldsymbol{y}}^k, \overline{\boldsymbol{w}}^k) \in \partial H_t^2(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k, \boldsymbol{w}^k)$ kaavojen (40) –(44) mukaan, missä $\widehat{y^k}_i$ ja $\widehat{w^k}_i$ lasketaan käyttäen kaavoja (21) ja (22).
2. Laske $(\boldsymbol{u}^{k+1}, v^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1})$ ratkaisemalla minimointitehtävä (45).

Ei-DCA3

1. Laske $(\overline{y}_i^k, \overline{w}_i^k) \in \partial(-\psi^{\min})(y_i^k, w_i^k)$ kaikilla $i = 1, \ldots, m$ käyttäen kaavaa (20).
2. Laske konveksin kvadraattisen tehtävän

$$\begin{gathered} \min g(\boldsymbol{u}, v, \boldsymbol{y}) + ts - \langle \nabla_u h(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k), \boldsymbol{u}\rangle - \frac{\partial}{\partial v} h(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k) v - \langle \nabla_y h(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k), \boldsymbol{y}\rangle \\ s.t \quad (\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}, \quad s \geq 0 \\ \psi^{\min}(y_i^k, w_i^k) - \overline{y}_i^k(y_i - y_i^k) - \overline{w}_i^k(w_i - w_i^k) - s \leq 0, \quad i = 1, \ldots, m \end{gathered}$$

optimiratkaisu $(\boldsymbol{u}^{k+1}, v^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1}, s^{k+1})$.

Ei-DCA4

1. Laske $(\overline{y}_i^k, \overline{w}_i^k) \in \partial(-\psi^{FB})(y_i^k, w_i^k)$ kaikilla $i = 1, \ldots, m$ käyttäen kaavoja (21) ja (22).
2. Laske konveksin kvadraattisen tehtävän

$$\begin{gathered} \min g(\boldsymbol{u}, v, \boldsymbol{y}) + ts - \langle \nabla_u h(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k), \boldsymbol{u}\rangle - \frac{\partial}{\partial v} h(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k) v - \langle \nabla_y h(\boldsymbol{u}^k, v^k, \boldsymbol{y}^k), \boldsymbol{y}\rangle \\ s.t \quad (\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}, \quad s \geq 0 \\ \psi^{FB}(y_i^k, w_i^k) - \overline{y}_i^k(y_i - y_i^k) - \overline{w}_i^k(w_i - w_i^k) - s \leq 0, \quad i = 1, \ldots, m \end{gathered} \tag{46}$$

optimiratkaisu $(\boldsymbol{u}^{k+1}, v^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{w}^{k+1}, s^{k+1})$.

Taulukko 2: DC-formuloinnit EiCP-tehtävälle

Sakko-funktiot	**DC kaavat**
$p_1(\boldsymbol{y}, \boldsymbol{w})$	$\min\{F_t^1(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) = G(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) - H_t^1(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) : (\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) \in \mathbb{R}^{3n+1}\}$, $G(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) = g(\boldsymbol{u}, v, \boldsymbol{y}) + \chi_{\mathcal{C}}(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w})$, $H_t^1(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) = g(\boldsymbol{u}, v, \boldsymbol{y}) - f(\boldsymbol{u}, v, \boldsymbol{y}) - t\sum_{i=1}^m \psi^{\min}(y_i, w_i)$
$p_2(\boldsymbol{y}, \boldsymbol{w})$	$\min\{F_t^2(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) = G(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) - H_t^2(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) : (\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) \in \mathbb{R}^{3n+1}\}$, $H_t^2(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) = g(\boldsymbol{u}, v, \boldsymbol{y}) - f(\boldsymbol{u}, v, \boldsymbol{y}) - t\sum_{i=1}^m \psi^{FB}(y_i, w_i)$
$p_3(\boldsymbol{y}, \boldsymbol{w})$	$\min\{F_t^3(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}, s) = G_t^3(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}, s) - H^3(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}, s) :$ $(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}; -s - [-\psi^{\min}(y_i, w_i)] \leq 0, i = 1, \ldots, m; s \geq 0\}$, $G_t^3(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}, s) = g(\boldsymbol{u}, v, \boldsymbol{y}) + ts, H^3(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}, s) = g(\boldsymbol{u}, v, \boldsymbol{y}) - f(\boldsymbol{u}, v, \boldsymbol{y})$
$p_4(\boldsymbol{y}, \boldsymbol{w})$	$\min\{F_t^4(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}, s) = G_t^3(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}, s) - H^3(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}, s) :$ $(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}; -s - [-\psi^{FB}(y_i, w_i)] \leq 0, i = 1, \ldots, m; s \geq 0\}$,

6 Numeeriset tulokset ja vertailua

Lähdeartikkelissa [8] esitellyt neljä algoritmia on testattu kahdella testitehtäväsarjalla, QPLCC-tehtäviä testattiin tasapaino (equilibrium) rajoitteisilla tehtävillä, eli MPEC-tehtävillä, AMPL-ohjelmistolla (24 tehtävää) ja EiCP-tehtäviä testattiin Matriisi Marketin datalla (18 tehtävää). DCA-algoritmien tehokkuuden vertailuun on käytetty optimointiohjelmaa KNITRO.

6.1 Numeeriset tulokset

Tarkat numeeriset tulokset löytyvät artikkelista [8], tässä tulokset pähkinänkuoressa:

1. QPLCC-tehtävän tulokset

   (a) Kohdefunktion arvon kannalta:

   - Yleisesti DCA-algoritmit (QP-DCA3 ja QP-DCA4) ovat parempia kuin KNITRO-ohjelma, kun taas standardi DCA-algoritmit (QP-DCA1 ja QP-DCA2) ovat vertailtavissa KNITRO-ohjelman kanssa.
   - Kaikki algoritmit antavat saman tuloksen 4/24 tehtävässä. QP-DCA4-algoritmi voittaa KNITRO-ohjelman 11/24 tehtävässä ja on samantasoinen viidessä tehtävässä. Kahdeksalla jäljelle jääneessä tehtävässä QP-DCA4-algoritmi ja KNITRO-ohjelma antavat saman tuloksen.

QP-DCA3-algoritmi voittaa myös KNITRO-ohjelman 11/24 tehtävässä mutta tuottaa hieman huonomman tuloksen kolmessa tehtävässä. QP-DCA1-algoritmi (vast. QP-DCA2) on KNITRO-ohjelmaa parempi 8/24 (vast. 9/24) tehtävässä ja huonompi 7/24 tehtävässä.

- Huomautettakoon, että QP-DCA3- ja QP-DCA4-algoritmit löytävät erittäin hyvän ratkaisun yhteen tehtävään (kohdefunktion arvo on -683.0330 verrattuna KNITRO-ohjelman löytämään arvoon -230.0386). Kun taas QP-DCA1- ja QP-DCA2-algoritmit eivät löytäneet sallittua ratkaisua tälle 3600 sekunnin jälkeen.
- Esitellyistä algoritmeista yleiset DCA-algoritmit ovat samantasoisia kuin KNITRO, kun taas standardeissa DCA-algoritmeissa QP-DCA2-algoritmi tuottaa paremmat ratkaisut tehtävissä, joissa on keskitasoinen määrä komplementaarisia rajoitteita.

(b) Suoritusajan perusteella:

- QP-DCA1-algoritmi on kaikkein tehokkain kaikista DCA-algoritmeista, se on nopeampi kuin kaikki muut DCA-algoritmit 16/24 tehtävässä. Keskimäärin se on 1.33 kertaa nopeampi kuin QP-DCA2-algoritmi ja noin 2.2 kertaa nopeampi kuin QP-DCA3- ja QP-DCA4-algoritmit.
- QP-DCA1-algoritmi (vast. QP-DCA3) on nopeampi kuin QP-DCA2-algoritmi (vast. QP-DCA4) 17/24 tehtävässä ja QP-DCA1-algoritmi (vast. QP-DCA2) on nopeampi kuin QP-DCA3-algoritmi (vast. QP-DCA4) 21/24 tehtävässä (vast. 18/24 tehtävässä).
- DCA-algoritmien suoritusaika on melko vakaa, se vaihtelee 0.01 ja 15.03 sekunnin välillä kun taas KNITRO-ohjelman suoritusaika NEOS-serverillä vaihtelee 0.005 ja 102.148 sekunnin väillä.

2. EiCP-tehtävän tulokset

(a) Kohdefunktion arvon kannalta:

- Ei-DCA3-algoritmi on paras. Ei-DCA3-algoritmi voittaa KNITRO-ohjelman kaikissa tapauksissa ja jää globaalista optimiratkaisusta (39) vaille vähemmän kuin 10^{-6} (vast. 10^{-5}) 15/18 (vast. 17/18) tehtävässä.
- Tämän jälkeen tulee Ei-DCA4-algoritmi, se voittaa KNITRO-ohjelman 17/18 tehtävässä (missä se antaa globaalin optimiratkaisun EiCP-tehtävälle pienemmällä tarkkuudella kuin 10^{-5}).
- Ei-DCA1- ja Ei-DCA2-algoritmit voittavat KNITRO-ohjelman 16/18 tehtävässä. Ei-DCA1-algoritmi (vast. Ei-DCA2) tuottaa globaalin ratkaisun EiCP-tehtäviin pienemmällä tarkkuudella kuin 10^{-5} 15/18 (vast, 16/18) tehtävässä ja epäonnistuu (toisin sanoen ei löydä sallittua ratkaisua 3600 sekunnin jälkeen) kahdessa tehtävässä (vast. yhdessä tehtävässä).
- KNITRO-ohjelman ratkaisuilla on huonommat tarkkuudet (suurempi kuin 10^{-4}) 7/18 EiCP-tehtävässä ja se löytää epäsallitun ratkaisun kahteen tehtävään.

- Yleinen DCA-algoritmi tuottaa oikein hyvän ratkaisun yhteen tehtävään (kohdefunktion arvo on pienempi kuin 1.1×10^{-10}), kun taas muut algoritmit epäonnistuivat kyseisessä tehtävässä.
- Standardi (vast. yleisessä) DCA-algoritmissa, Ei-DCA1-algoritmi on parempi kuin Ei-DCA2-algoritmi 10/18 tehtävässä ja huonompi kuin Ei-DCA2-algoritmi kahdessa tehtävässä (vast. Ei-DCA3-algoritmi voittaa Ei-DCA4-algoritmin 14/18 tehtävässä ja on vertailtavissa lopuissa tehtävissä).

(b) Suoritusajan perusteella:

- Suurista dimensioista huolimatta, DCA-algoritmit suoriutuvat silti sangen nopeasti (vähemmän kuin 12 sekuntia).
- Keskimäärin Ei-DCA1-algoritmi on nopein ja se on myös nopeampi kuin kaikki muut DCA-algoritmit 8/18 tehtävässä.

6.2 EIDCA3- ja DCA-NPL-algoritmien vertailua

Vertaillaan Ei-DCA3-algoritmia DCA-algoritmilla (paras näistä DCA versioista) lähteen [11] NLP1-tehtäviin, jotka ovat muotoa

$$\min\{F(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) = \|\boldsymbol{u} - v\boldsymbol{y}\|^2 + \langle \boldsymbol{y}, \boldsymbol{w}\rangle : (\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}\}. \tag{47}$$

Kaksi DC-hajotelmaa funktiosta F on esitelty lähteessä [11]. Vastaavia DCA-algoritmeja testattiin aikaisempaan 18 tehtävään ja seuraava DC-hajotelman mukainen DCA algoritmi huomattiin olevan parempi

$$\begin{aligned} F(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) &= G(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) - H(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}), \\ G(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) &= \big(\frac{\rho_1}{2} + 1\big)\|\boldsymbol{u}\|^2 + \frac{\rho_1 + \rho_2}{2} v^2 + \frac{\rho_1 + \rho_2}{2}\|\boldsymbol{y}\|^2 + \frac{\|\boldsymbol{y} + \boldsymbol{w}\|}{4}, \\ H(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) &= G(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}) - F(\boldsymbol{u}, v, \boldsymbol{y}, \boldsymbol{w}), \end{aligned}$$

missä ρ_1 ja ρ_2 on annettu osiossa 5.2. Täten verrataan Ei-DCA3-algoritmia DCA-algoritmilla, jolla on tällainen hajotelma.
Keskitytään kohdefunktion arvoon $\|\boldsymbol{u} - v\boldsymbol{y}\|^2 + \langle \boldsymbol{y}, \boldsymbol{w}\rangle$, joka saadaan kahdella vastaavalla DCA-algoritmilla ja näiden suoritusaikaan. Huomataan, että $(\lambda, \boldsymbol{y})$ on ratkaisu EiCP-tehtävälle, jos ja vain jos $(\boldsymbol{y}/\lambda, 1/\lambda, \boldsymbol{y}, \boldsymbol{w})$ on ratkaisu tehtävään (47) kohdefunktio arvolla nolla. Näin ollen, kohdefunktion arvo nolla merkitsee, että $(1/v, \boldsymbol{y})$ on optimiratkaisu EiCP-tehtäville. DCA-NLP-tehtäville valitaan tarkkuusrajan olevan 10^{-6} parempien kohdefunktion arvojen saamiseksi. Ei-DCA3-algoritmin lopetusehto on sama kuin DCA-NLP-tehtävissä. Laskennalliset tulokset löytyvät lähteestä [8]. Tuloksista voidaan havainnoida, että:

1. Kohdefunktion arvon kannalta, Ei-DCA3-algoritmi on hieman parempi kuin DCA-NLP suurimmassa osassa tehtävistä, nimenomaan 12/18 tehtävistä. Ei-DCA3-algoritmi (vast. DCA-NLP) tuottaa ratkaisun EiCP-tehtäville pienemmällä tarkkuudella kuin 10^{-6} 15/18 tehtävissä (vast. 12/18 tehtävissä).

2. Suoritusajan perusteella, Ei-DCA3-algoritmi on nopeampi kuin DCA-NLP 12/18 tehtävissä. Keskimäärin Ei-DCA3-algoritmi on 1.62 kertaa nopeampi kuin DCA-NLP.

7 Yhteenveto

Tässä työssä on keskitytty DC-funktioihin ja DCA-algoritmeihin sekä erityisesti siihen, kuinka DCLCC-optimointitehtävä voidaan ratkaista DCA-algoritmin avulla. Tutkielman alussa on käyty läpi matemaattisen optimoinnin perustietoja. Tämän jälkeen on käyty läpi DC-funtiot ja DCA-algoritmit, keskittymällä nimenomaan yleiseen ja standardiin DCA-algoritmiin. DCLCC-optimointitehtävän esittelyn jälkeen on luotu neljä sakkofunktiota, jotka perustuivat min/Fisher-Burmeisterin funktioihin. Tällöin sakottamalla komplementaaristen rajoitteiden rikkoontumista on voitu formuloida DCLCC-tehtävä standardiksi DC-tehtäväksi. Vastaavalla tavalla on myös esitelty neljä DCA-algoritmia, jotka soveltuvat nimenomaan DCLCC-tehtävien ratkaisemiseen.

Yleisesti ottaen, standardit DC-tehtävät ovat helpompia ratkaista kuin yleiset DC-tehtävät. Kuitenkin tässä työssä on huomattu asian olevan päinvastoin: kaksi neljästä yleisen DC-tehtävän muotoon formuloidusta DC-tehtävästä ovat sopivampia DCA-algoritmien formulointiin. Numeerisissa tuloksissa kaksi vastaavaa yleistä DCA-algoritmia ovat olleet tehokkaampia ratkaisemaan täydentävien rajoitteiden tehtävää kuin kaksi standardia DCA-algoritmia. Esitettyjen algoritmien ja lähestymistapojen tehokkuus ilmenee erityisesti numeerisissa tuloksissa, jotka on esitetty työn lopussa. Tuloksista ilmenee käsiteltyjen algoritmien ylivoimaisuus KNITRO-ohjelmaan ja vanhaan DCA-algoritmiin verrattuna EiCP-tehtävissä.

Viitteet

[1] Bagirov, A., Karmitsa, N., Mäkelä, M. M.: *Introduction to Nonsmooth Optimization: Theory, Practice and Software.* Springer Verlag, Cham, 2014.

[2] de Oliveira, W.: *The ABC of DC Programming.* Set-Valued and Variational Analysis, 28, 679-706, 2020.

[3] Joki, K.: *Bundle methods in nonsmooth DC optimization.* Doctoral Thesis, University of Turku, 2018.

[4] Júdice, J.J., Sherali, H.D., Ribeiro, I.M.: *The eigenvalue complementarity problem.* Computational Optimization and Applications, 37, 139-156, 2007.

[5] Le Thi, H.A., Huynh, V.N., Pham Dinh, T.: *DC Programming and DCA for general DC Programs.* Advanced Computational Methods for Knowledge Engineering, 15−35, 2014.

[6] Le Thi, H.A., Pham Dinh, T.: *DC programming and DCA: thirty years of developments.* Mathematical Programming, Ser. B, 169(1), 5–68, 2018.

[7] Le Thi, H.A., Pham Dinh, T., Huynh, V.N.: *Exact penalty and error bounds in DC programming.* Journal of Global Optimization, 52, 509-535, 2012.

[8] Le Thi, H.A., Tam Nguyen, T.M., Pham Dinh, T.: *On solving difference of convex functions programs with linear complementarity constraints.* Computational Optimization and Applications, 86, 163-197, 2023.

[9] Luo, Z.Q., Pang, J.S., Raplp, D.: *Mathematical Programs with Equilibrium Constraints.* Cambridge University Press, Cambridge, 1996.

[10] MIT OpenCourseWare: *DC Programming: The Optimization Method You Never Knew You Had To Know.* MIT OpenCourseWare, 15.097 Prediction: Machine Learning and Statistics, 2012.

[11] Niu, Y.S., Pham Dinh, T., Le Thi, H.A., Júdice, J.J.: *Efficient DC programming approaches for the asymmetric eigenvalue complementarity problem.* Optimization Methods and Software, 28(4), 812–829, 2013.

[12] Pham Dinh, T., Le Thi, H.A.: *Convex analysis approach to D.C. programming: theory, algorithms and applications.* Acta Mathematica Vietnamica, 22(1), 289-355, 1997.

[13] Scheel, H., Scholtes, S.: *Mathematical programs with complementarity constraints: stationary, optimality, and sensitivity.* Mathematics of Operations Research, 25(1), 1–22, 2000.