

双対性を利用した アルゴリズム

京都大学大学院情報学研究科

山下信雄

概要

1. 双対な関係のアルゴリズム

2. 双対問題を解くアルゴリズム

Sinkhorn algorithm

Trust-region subproblem, 3次正則化ニュートン法

3. 双対性を利用した分解アルゴリズム・並列アルゴリズム

並列アルゴリズム

Benders分解法

双対な関係のアルゴリズム

主問題を解くアルゴリズムが
双対問題で見ると別のアルゴリズムとなっていることがある.

例：①乗数法 $\Leftrightarrow$ 近接点法

②交互方向乗数法(ADMM) $\Leftrightarrow$ Douglas Rashfordアルゴリズム

一方の収束性を使えば、他方の収束性を示せることもある.

双対な関係のアルゴリズム①

乗数法と近接点法

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & h_i(x) = 0 \\ & g_j(x) \leq 0\end{array}$$

拡張ラグランジュ関数: $L_c(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \left\{ \lambda_i h_i(x) + \frac{c}{2} h_i(x)^2 \right\} + \frac{1}{2c} \sum_{j=1}^r \left\{ \max\{0, \mu_j + c g_j(x)\}^2 - \mu_j^2 \right\}$

乗数法

$$x^{k+1} \in \arg \min_x L_c(x, \lambda^k, \mu^k)$$

$$\lambda_i^{k+1} = \lambda_i^k + c h_i(x^{k+1}), i = 1, \dots, m$$

$$\mu_j^{k+1} = \max\{0, \mu_j^k + \alpha g_j(x^{k+1})\}, i = 1, \dots, r$$

部分問題の最適性の条件

$$0 = \nabla_x L_c(x^{k+1}, \lambda^k, \mu^k)$$

$$= \nabla f(x^{k+1}) + \sum_{i=1}^m \nabla h_i(x^{k+1}) (\lambda_i^k + c h_i(x^{k+1})) + \sum_{j=1}^r \nabla g_j(x^{k+1}) \max\{0, \mu_j^k + c g_j(x^{k+1})\}$$

$$= \nabla f(x^{k+1}) + \sum_{i=1}^m \nabla h_i(x^{k+1}) \lambda_i^{k+1} + \sum_{j=1}^r \nabla g_j(x^k) \mu_j^{k+1}$$

$$= \nabla_x L(x^{k+1}, \lambda^{k+1}, \mu^{k+1})$$

双対問題に対する近接点法

$$(\bar{\lambda}^{k+1}, \bar{\mu}^{k+1}) = \arg \min_{\substack{\lambda \in R^m \\ \mu \geq 0}} \left\{ -\omega(\lambda, \mu) + \frac{1}{2c} \|\lambda - \bar{\lambda}^k\|^2 + \frac{1}{2c} \|\mu - \bar{\mu}^k\|^2 \right\}$$

部分問題のKKT条件

$$-\eta + \frac{1}{c}(\bar{\lambda}^{k+1} - \bar{\lambda}^k) = 0, \quad \eta \in \partial_{\lambda} \omega(\bar{\lambda}^{k+1}, \bar{\mu}^{k+1})$$

$$-\xi + \frac{1}{c}(\bar{\mu}^{k+1} - \bar{\mu}^k) - \gamma = 0, \quad \xi \in \partial_{\mu} \omega(\bar{\lambda}^{k+1}, \bar{\mu}^{k+1})$$

$$\gamma \geq 0, \bar{\mu}^{k+1} \geq 0, \gamma^{\top} \bar{\mu}^{k+1} = 0$$

$$\bar{\mu}^{k+1} - c\gamma = \bar{\mu}^k + c\xi, \quad \gamma \geq 0, \bar{\mu}^{k+1} \geq 0, \gamma^{\top} \bar{\mu}^{k+1} = 0$$

$\Leftrightarrow$

$$\bar{\mu}_j^{k+1} = \max\{0, \bar{\mu}_j^k + c\xi_j\}$$

劣勾配の計算： $\bar{x}^{k+1} \in \arg \min L(x, \bar{\lambda}^{k+1}, \bar{\mu}^{k+1})$ ，つまり $\nabla L(\bar{x}^{k+1}, \bar{\lambda}^{k+1}, \bar{\mu}^{k+1}) = 0$ としたとき

$$h(\bar{x}^{k+1}) \in \partial_{\lambda} \omega(\bar{\lambda}^{k+1}, \bar{\mu}^{k+1})$$

$$g(\bar{x}^{k+1}) \in \partial_{\mu} \omega(\bar{\lambda}^{k+1}, \bar{\mu}^{k+1})$$

乗数法と近接点法

乗数法

$$\nabla L(x^{k+1}, \lambda^{k+1}, \mu^{k+1}) = 0,$$

$$\lambda_i^{k+1} = \lambda_i^k + ch_i(x^{k+1}), i = 1, \dots, m,$$

$$\mu_j^{k+1} = \max\{0, \mu_j^k + cg_j(x^{k+1})\}, i = 1, \dots, r$$



近接点法

$$\bar{\lambda}^{k+1} = \bar{\lambda}^k + \alpha\eta, \quad \eta \in \partial_{\lambda}\omega(\bar{\lambda}^{k+1}, \bar{\mu}^{k+1})$$

$$\bar{\mu}^{k+1} = \max\{0, \bar{\mu}^k + \alpha\xi\}, \quad \xi \in \partial_{\mu}\omega(\bar{\lambda}^{k+1}, \bar{\mu}^{k+1})$$

劣勾配(の一つ)

$$\nabla L(\bar{x}^{k+1}, \bar{\lambda}^{k+1}, \bar{\mu}^{k+1}) = 0$$

$$h(\bar{x}^{k+1}) \in \partial_{\lambda}\omega(\bar{\lambda}^{k+1}, \bar{\mu}^{k+1})$$

$$g(\bar{x}^{k+1}) \in \partial_{\mu}\omega(\bar{\lambda}^{k+1}, \bar{\mu}^{k+1})$$

補足①

- 拡張ラグランジュ関数による双対問題の劣勾配法とみることもできる.

目的関数： $\omega_c(\lambda, \mu) = \inf_x L_c(x, \lambda, \mu)$

劣勾配： $h(x^{k+1}) \in \partial_\mu \omega_c(\lambda^k, \mu^k), \quad \frac{1}{c} \max\{0, \mu^k + cg(x^{k+1})\} - \frac{1}{c} \mu^k \in \partial_\lambda \omega_c(\lambda^k, \mu^k)$

ステップ幅を c とすると

$$\lambda^{k+1} = \lambda^k + ch(x^{k+1})$$

$$\mu^{k+1} = \mu^k + c \left(\frac{1}{c} \max\{0, \mu^k + cg(x^{k+1})\} - \frac{1}{c} \mu^k \right) = \max\{0, \mu^k + cg(x^{k+1})\}$$

- 乗数法の(双対で見た場合の)加速化法は既に提案されている.
もちろん新しい加速化法ができれば、乗数法も加速化できる.

補足② composite optimizationに対しては

解きたい問題

$$\begin{aligned} \min \quad & f(x) + g(y) \\ \text{s.t.} \quad & Ax + By = b \end{aligned}$$

双対問題

$$\max -b^\top \lambda - f^*(A^\top \lambda) - g^*(B^\top \lambda)$$

乗数法 $(x^{k+1}, y^{k+1}) \in \arg \min \left\{ f(x) + g(y) + \langle \lambda^k, Ax - y \rangle + \frac{c}{2} \|Ax - y\|^2 \right\}$

$$\lambda^{k+1} = \lambda^k + c(Ax^{k+1} - y^{k+1})$$

近接点法 $\lambda^{k+1} \in \arg \min \left\{ b^\top \lambda + f^*(A^\top \lambda) + g^*(B^\top \lambda) + \frac{1}{2c} \|\lambda - \lambda^k\|^2 \right\}$

部分問題では f と g を同時にして扱う必要がある。
これらを別々に扱うのがADMMとDouglas-Rachford Algorithmである。

双対な関係のアルゴリズム②

ADMMとDouglas-Rachford algorithm (DRA)

ECE236C - Optimization Methods for Large-Scale Systems

Prof. L. Vandenberghe, UCLA

<https://www.seas.ucla.edu/~vandenbe/ee236c.html>

Douglas-Rachford algorithm (DRA) は二つの極大写像の和の写像の零点を求めるアルゴリズムである。

凸最適化においては二つの凸関数の和の最小化を行う。

$$\min_x p(x) + q(x) \quad \Leftrightarrow \quad 0 \in \partial p(x) + \partial q(x)$$

ただし p と q は凸関数である。

近接点法とDRA

凸関数 f に対して近接オペレータ (proximal operator) を以下のように定義する

$$\text{Prox}_f(x) = \arg \min_x \left\{ f(y) + \frac{1}{2} \|y - x\|^2 \right\}$$

前頁の問題 $\min_x p(x) + q(x)$ に対しては

近接点法 : $x^{k+1} = \text{Prox}_{p+q}(x^k)$

別々の近接オペレータ

DRA : $x^{k+1} = x^k + \text{Prox}_q(2\text{Prox}_p(x^k) - x^k) - \text{Prox}_p(x^k)$

どちらも非拡大写像の不動点アルゴリズムである.

2021年夏季学校(飯塚先生)

DR algorithmの別表現

$$x^{k+1} = x^k + \text{Prox}_q(2\text{Prox}_p(x^k) - x^k) - \text{Prox}_p(x^k)$$



中間変数 λ^k の導入

$$\textcircled{1} \lambda^{k+1} = \text{Prox}_p(x^k), \quad \textcircled{2} x^{k+1} = x^k + \text{Prox}_q(2\lambda^{k+1} - x^k) - \lambda^{k+1}$$



順番を変える

$$\textcircled{1} x^{k+1} = x^k + \text{Prox}_q(2\lambda^k - x^k) - \lambda^k, \quad \textcircled{2} \lambda^{k+1} = \text{Prox}_p(x^{k+1}),$$



中間変数 u^k の導入

$$\textcircled{1} u^{k+1} = \text{Prox}_q(2y^k - x^k), \quad \textcircled{2} x^{k+1} = x^k + u^{k+1} - y^k, \quad \textcircled{3} y^{k+1} = \text{Prox}_p(x^k + u^{k+1} - y^k)$$



変数の変換: $w^k = y^k - x^k$ ($x^k = y^k - w^k$)

$$\textcircled{1} u^{k+1} = \text{Prox}_q(\lambda^k + w^k), \quad \textcircled{2} \lambda^{k+1} = \text{Prox}_p(u^{k+1} - w^k), \quad \textcircled{3} w^{k+1} = w^k + \lambda^{k+1} - u^{k+1}$$

Composite optimizationの双対問題に対するDRA

解きたい問題

$$\min f(x) + g(y)$$

$$\text{s.t. } Ax + By = b$$

双対問題

$$\max -b^\top \lambda - f^*(A^\top \lambda) - g^*(B^\top \lambda)$$

ここで,

$$p(\lambda) = c(b^\top \lambda + f^*(A^\top \lambda)), \quad q(\lambda) = c g^*(B^\top \lambda)$$

として, DRAを適用する. ただし c は正の定数である.

$$\text{DRA} \quad \textcircled{1} u^{k+1} = \text{Prox}_q(\lambda^k + w^k), \textcircled{2} \lambda^{k+1} = \text{Prox}_p(u^{k+1} - w^k), \textcircled{3} w^{k+1} = w^k + \lambda^{k+1} - u^{k+1}$$

$$\textcircled{1} u^{k+1} = \text{Prox}_q(\lambda^k + w^k) = \arg \min_u \left\{ (b^\top u + f^*(A^\top u)) + \frac{1}{2c} \|u - (\lambda^k + w^k)\|^2 \right\}$$

これは次の問題の乗数法に対応した近接点法とみなせる.

$$\min q(x)$$

$$\text{s.t. } Ax = b$$

$$\text{よって } x^{k+1} \in \arg \min \left\{ f(x) + \langle y^k + w^k, Ax - b \rangle + \frac{t}{2} \|Ax - b\|^2 \right\}$$

$$u^{k+1} = y^k + w^k + t(Ax^{k+1} - b)$$

$$\text{同様に } \textcircled{2} \lambda^{k+1} = \text{Prox}_p(u^{k+1} - w^k) = \arg \min \left\{ g^*(B^\top y) + \frac{1}{2c} \|y - (u^{k+1} - w^k)\|^2 \right\} \text{ から}$$

$$y^{k+1} \in \arg \min \left\{ g(y) + \langle u^{k+1} - w^k, By \rangle + \frac{t}{2} \|By\|^2 \right\}$$

$$\lambda^{k+1} = u^{k+1} - w^k + tBy^{k+1}$$

$$\text{最後に } \textcircled{3} w^{k+1} = w^k + \lambda^{k+1} - u^{k+1} = w^k + u^{k+1} - w^k + cBy^{k+1} - u^{k+1} = cBy^{k+1}$$

整理すると

③より $w^k = tBy^k$ とみなせる. これを①, ②に代入すると

$$\textcircled{1} \quad x^{k+1} \in \arg \min_x \left\{ f(x) + g(y^k) + \langle \lambda^k, Ax + By^k - b \rangle + \frac{c}{2} \|Ax + By^k - b\|^2 \right\}$$

$$u^{k+1} = \lambda^k + c(Ax^{k+1} + By^k + b)$$

目的関数の定数項は
影響を与えない

$$\textcircled{2} \quad y^{k+1} \in \arg \min_y \left\{ g(y) + \langle u^{k+1} - w^k, By \rangle + \frac{c}{2} \|By\|^2 \right\}$$

$$= \arg \min_y \left\{ g(y) + \langle \lambda^k + c(Ax^{k+1} - b), By \rangle + \frac{c}{2} \|By\|^2 \right\}$$

$$= \arg \min_y \left\{ f(x^{k+1}) + g(y) + \langle \lambda^k, Ax^{k+1} + By - b \rangle + \frac{t}{2} \|Ax^{k+1} + By - b\|^2 \right\}$$

$$\lambda^{k+1} = \lambda^k + c(Ax^{k+1} + By^{k+1} - b)$$

ADMM

拡張ラグランジュ関数

$$L_c(x, y, \lambda) = f(x) + g(y) + \langle \lambda, Ax + By - b \rangle + \frac{c}{2} \|Ax + By - b\|^2$$

さきほどの双対問題に対するDRAは

$$x^{k+1} \in \arg \min_x L_c(x, y^k, \lambda^k)$$

$$y^{k+1} \in \arg \min_y L_c(x^{k+1}, y, \lambda^k)$$

$$\lambda^{k+1} = \lambda^k + c(Ax^{k+1} + By^{k+1} - b)$$



主問題に対するADMM

注意

双対問題に対するアルゴリズムの大域的収束性から
主問題に対するアルゴリズムの大域的収束性を示すことができる.

しかし, 双対問題に対するアルゴリズムの収束が速いからといって,
主問題に対するアルゴリズムの収束が速くなるとは限らない.

拡張ラグランジュ関数のパラメータ c で主変数、双対変数の収束の速さを
調整できる.

概要

1. 双対な関係のアルゴリズム

2. 双対問題を解くアルゴリズム

Sinkhorn algorithm

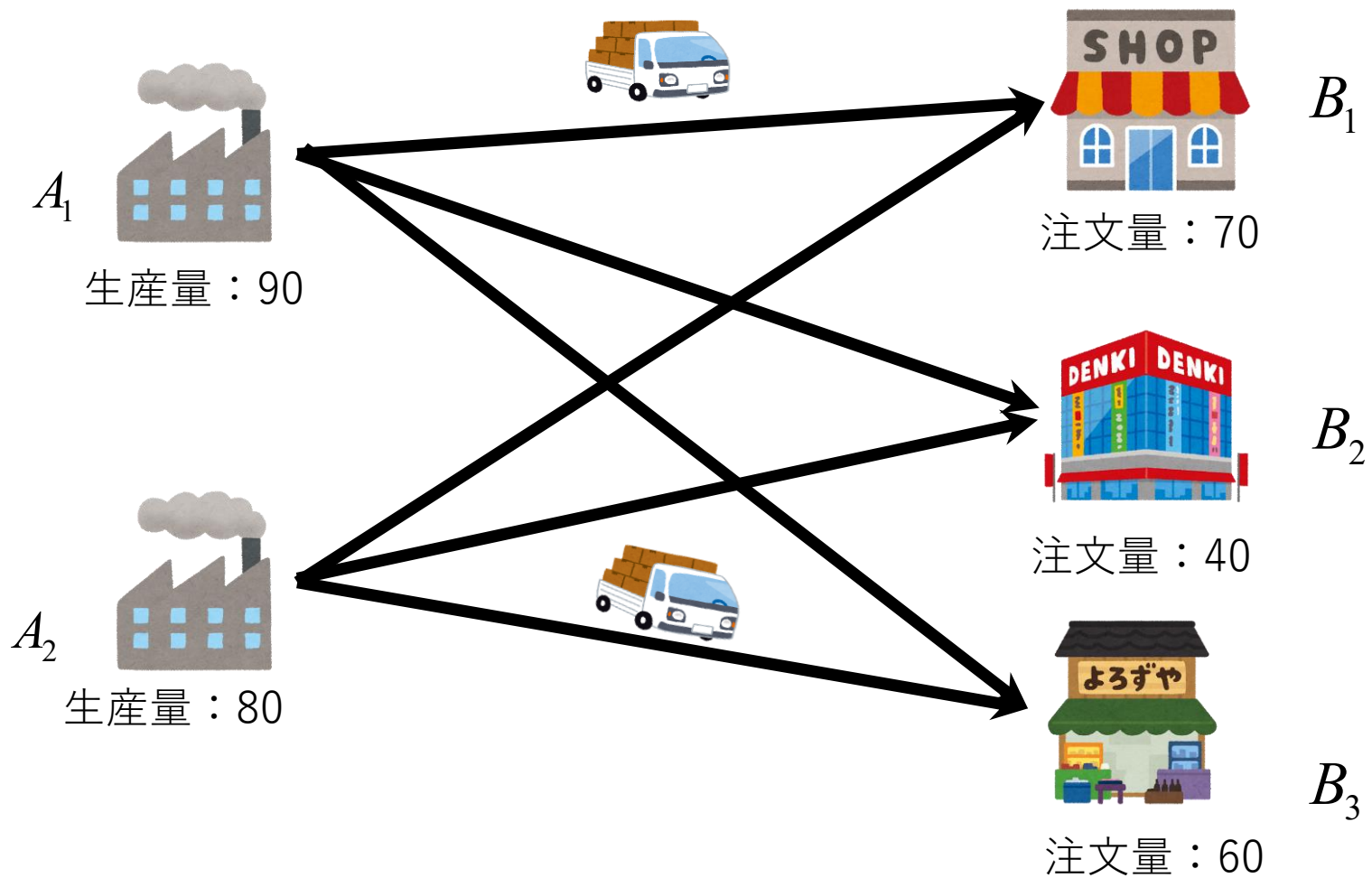
Trust-region subproblem, 3 次正則化ニュートン法

3. 双対性を利用した分解アルゴリズム・並列アルゴリズム

並列アルゴリズム

Benders分解法

輸送問題



輸送問題の定式化

工場 i から店舗 j への輸送コスト(単位量あたり) : c_{ij}

工場 i から店舗 j への輸送量(決定変数) : x_{ij}

目的関数:

$$\sum_i \sum_j c_{ij} x_{ij}$$

制約条件 :

工場 i の出荷量 p_i に関する制約条件: $\sum_j x_{ij} = p_i$

店舗 j の入荷量 q_j に関する制約条件: $\sum_i x_{ij} = q_j$

輸送問題

$$\min \sum_i \sum_j c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_j x_{ij} = p_i \quad \forall i$$

$$\sum_i x_{ij} = q_j \quad \forall j$$

$$x_{ij} \geq 0 \quad \forall i, j$$

輸送計画の画像処理への応用： Wasserstein距離(の離散版)

距離の公理

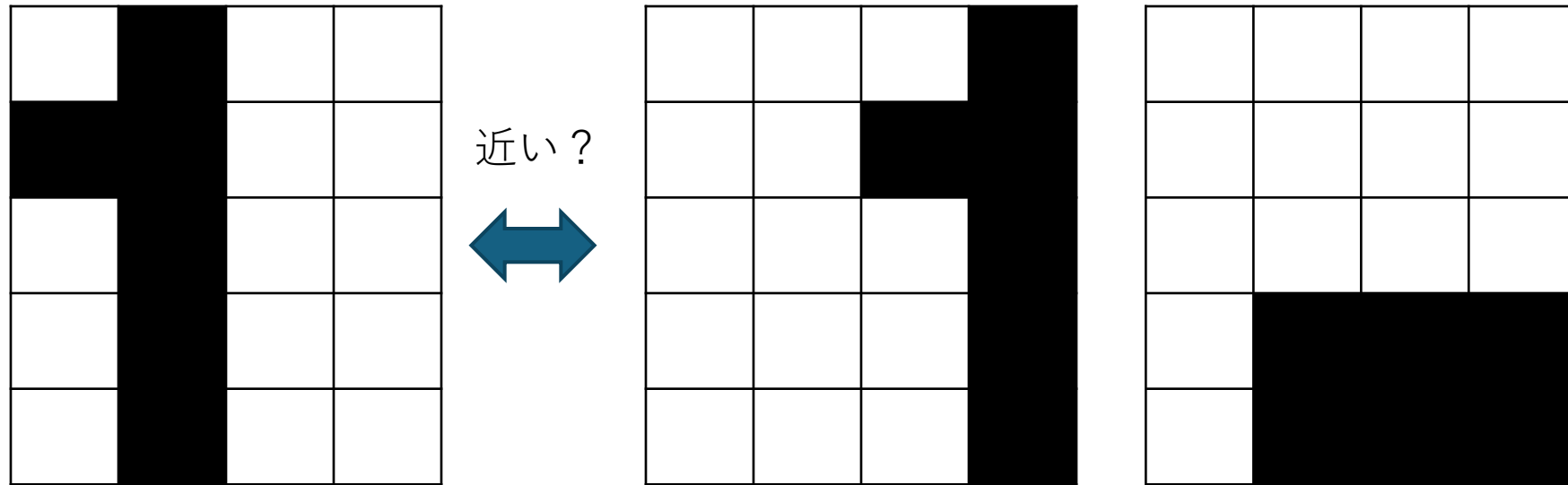
- (i) $d(x, y) = 0 \Leftrightarrow x = y$
- (ii) $d(x, y) = d(y, x)$
- (iii) $d(x, y) + d(y, z) \geq d(x, z)$

距離の例： $x, y \in R^n$

ユークリッド距離： $d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$

マンハッタン距離： $d(x, y) = \sum_{i=1}^n |x_i - y_i|$

画像間の近さはどう測る？

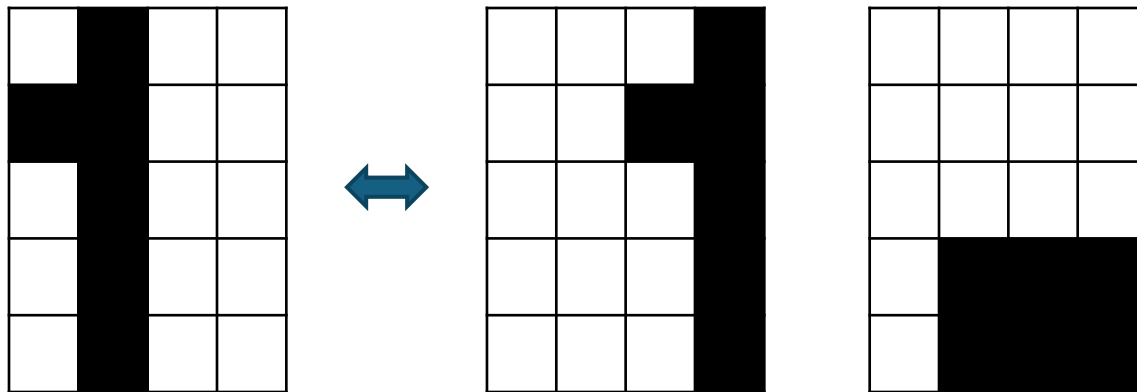


画像をベクトル表現し，近さをユークリッド距離やマンハッタン距離で測ると，全体にずれたときに大きな値をとってしまう．（右の方が近くなる）

Wasserstein距離

画像Xから画素を運んで画像Yにすることを考える．距離をコストとする．
この画素の**輸送問題の最適値をWasserstein距離**という．

輸送コストをマンハッタン距離とした場合



Wasserstein距離： 12
マンハッタン距離： 12

17
8

Sinkhorn algorithm

輸送問題に正則化項(negative entropy)を付けた問題：

$$\begin{aligned} \min \quad & \sum_i \sum_j \{ c_{ij} x_{ij} + \epsilon x_{ij} \ln x_{ij} \} \\ \text{s.t.} \quad & \sum_j x_{ij} = p_i \quad \forall i, \quad \sum_i x_{ij} = q_j \quad \forall j \\ & x_{ij} \geq 0 \quad \forall i, j \end{aligned}$$

の解法の一つ.

双対問題に対する交互方向法.

二つの等式制約をもつ凸最適化問題

$$\min f(x)$$

$$\text{s.t. } Ax = a$$

$$Bx = b$$

$$x \in S$$

ラグランジュ関数

$$L(x, \alpha, \beta) = f(x) + \alpha^\top (Ax - a) + \beta^\top (Bx - b)$$

双対問題と交互方向法

双対問題

$$\max \omega(\alpha, \beta) := \inf_{x \in S} L(x, \alpha, \beta)$$

交互方向法

$$\alpha^{k+1} \in \arg \max_{\alpha} \omega(\alpha, \beta^k)$$

$$\beta^{k+1} \in \arg \max_{\beta} \omega(\alpha^{k+1}, \beta)$$

交互方向法の部分問題って何？

$$\alpha^{k+1} \in \arg \max_{\alpha} \omega(\alpha, \beta^k)$$

強双対性が成り立つとき,

$$\max_{\alpha} \inf_{x \in S} L(x, \alpha, \beta^k) = \inf_{x \in S} \max_{\alpha} L(x, \alpha, \beta^k)$$

よって、次の問題のKKT点のラグランジュ乗数が α^{k+1} となる.

$$\min f(x) + (\beta^k)^\top (Bx - b)$$

$$\text{s.t. } Ax = a$$

$$x \in S$$

Sinkhorn algorithm

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} + \varepsilon \sum_{i=1}^n \sum_{j=1}^n x_{ij} \ln x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} = p_i, i = 1, \dots, n \\ & \sum_{i=1}^n x_{ij} = q_j, j = 1, \dots, n \\ & x_{ij} \geq 0, i, j = 1, \dots, n \end{aligned}$$

交互方向法

$$\begin{aligned} \alpha^{k+1} &= \arg \max_{\alpha} \omega(\alpha, \beta^k) \\ \beta^{k+1} &= \arg \max_{\beta} \omega(\alpha^{k+1}, \beta) \end{aligned}$$

$$\begin{aligned} u_i &= \exp\left(\frac{\alpha_i}{\varepsilon}\right) \\ v_j &= \exp\left(\frac{\beta_j}{\varepsilon}\right) \end{aligned}$$

Sinkhorn Algorithm

$$K_{ij} = \exp\left(-\frac{c_{ij}}{\varepsilon}\right) \text{ とする.}$$

$$u_i^{k+1} = \frac{p_i}{\sum_{j=1}^n K_{ij} v_j^k}, i = 1, \dots, n$$

$$v_j^{k+1} = \frac{q_j}{\sum_{i=1}^n K_{ij} u_i^{k+1}}, j = 1, \dots, n$$

なぜこのアプローチが人気なのか？

- Wasserstein 距離の近似 $d_\varepsilon(p, q)$ が求まる.
- **GPUが使える.**
- Wasserstein 距離の近似 $d_\varepsilon(p, q)$ の微分が計算できる.

$$\alpha^* = \nabla_p d_\varepsilon(p, q)$$

$$\beta^* = \nabla_q d_\varepsilon(p, q)$$

⇒ 画像を生成する機械学習に使える.

概要

1. 双対な関係のアルゴリズム

2. 双対問題を解くアルゴリズム

Sinkhorn algorithm

Trust-region subproblem, 3 次正則化ニュートン法

3. 双対性を利用した分解アルゴリズム・並列アルゴリズム

並列アルゴリズム

Benders分解法

信頼領域法と 3 次正則化ニュートン法

制約なし最適化問題 $\min f(x)$ を考える. $g^k = \nabla f(x^k)$, $H_k = \nabla^2 f(x^k)$ とする.

信頼領域法(trust region method)

$$d^k = \arg \min_d \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d \mid \|d\| \leq \Delta_k \right\}$$

$$x^{k+1} = x^k + d^k$$

部分問題は非凸
解ける??

3 次正則化ニュートン法(Cubic regularized Newton method)

$$d^k = \arg \min_d \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} \|d\|^3 \right\}$$

$$x^{k+1} = x^k + d^k$$

Trust region subproblem (TRS) $\min_d \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d \mid \|d\| \leq \Delta_k \right\}$

TRSは2次制約付き2次最適化問題. ラグランジュ双対はSDPで書ける.
強双対性が成り立つ. (S-lemmaなどで証明)

d^k がTRSの最適解であることの必要十分条件は以下のものである.

(1) KKT条件をみたす.

$$g^k + (H_k + \mu I) d^k = 0, \|d^k\| \leq \Delta_k, \mu \geq 0, \mu(\|d^k\| - \Delta_k) = 0$$

(2) 次の不等式を満たす. (ラグランジュ双対の制約条件)

$$H_k + \mu I \succeq 0$$

この条件より **TRSは1変数 μ の非線形方程式**として解くことができる.

3 次正則化ニュートン法 $\min_d \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} \|d\|^3 \right\}$

Nesterov, Y., Polyak, B. Cubic regularization of Newton method and its global performance. *Math. Program.* 108, 177–205 (2006).

d^k が部分問題の最適解であることの必要十分条件は以下のものである.

(i) 最適性の必要条件をみtas.

$$g^k + (H_k + \mu I)d^* = 0, \quad \mu = \frac{M}{2} \|d^*\|$$

(ii) 次の不等式を満たす.

$$H_k + \mu I \succeq 0$$

この条件より 1 変数 μ の非線形方程式として解くことができる.

証明

$$\min_d \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} \|d\|^3 \right\}$$

まず問題を等価に変換する.

$$\begin{aligned} \min_{d,t} \quad & \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} t^{\frac{3}{2}} \\ \text{s.t.} \quad & \|d\|^2 \leq t \end{aligned}$$

必要性：省略.

2024年夏季学校(丸茂先生)演習問題 5

十分性：1変数の双対問題を導出して，その強双対性から証明.

$$\begin{aligned}
\min_{\|d\|^2 \leq t} \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} t^{\frac{3}{2}} \right\} &= \min_{t, d} \sup_{\mu \geq 0} \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} t^{\frac{3}{2}} + \frac{\mu}{2} (\|d\|^2 - t) \right\} \\
&\geq \sup_{\mu \geq 0} \min_{t, d} \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} t^{\frac{3}{2}} + \frac{\mu}{2} (\|d\|^2 - t) \right\} \\
&\geq \sup_{\substack{\mu \geq 0 \\ H + \mu I \succeq 0}} \min_{t, d} \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} t^{\frac{3}{2}} + \frac{\mu}{2} (\|d\|^2 - t) \right\}
\end{aligned}$$

よって、次の問題は**弱双対性**が成り立つ双対問題となる.

$$\begin{aligned}
\max_{\mu} \quad & \omega(\mu) := \min_{t, d} \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} t^{\frac{3}{2}} + \frac{\mu}{2} (\|d\|^2 - t) \right\} \\
\text{s.t} \quad & \mu \geq 0 \\
& H + \mu I \succeq 0
\end{aligned}$$

条件(i), (ii)が成り立つときに強双対性が成り立つことを示す.

$$\mu = \frac{M}{2} \|d^*\| \quad \text{とすると, 条件(ii)より } \mu \geq 0, H_k + \mu I \succeq 0$$

つまり, 双対問題の実行可能解. さらに

$$\begin{aligned} \omega(\mu) &= \min_{d,t} \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{M}{6} t^{\frac{3}{2}} + \frac{\mu}{2} (\|d\|^2 - t) \right\} \\ &= \min_d \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{\mu}{2} \|d\|^2 \right\} + \min_t \left\{ \frac{M}{6} t^{\frac{3}{2}} - \frac{\mu t}{2} \right\} \end{aligned}$$

$$\text{条件(i),(ii)より, } \min_d \left\{ \langle g^k, d \rangle + \frac{1}{2} d^\top H_k d + \frac{\mu}{2} \|d\|^2 \right\} = \langle g^k, d^k \rangle + \frac{1}{2} (d^k)^\top H_k d^k + \frac{\mu}{2} \|d^k\|^2$$

$$\text{さらに } \min_t \left\{ \frac{M}{6} t^{\frac{3}{2}} - \frac{\mu t}{2} \right\} \text{ は } t = \left(\frac{2\mu}{M} \right)^2 = \|d^k\|^2 \text{ のとき最小値 } \frac{M}{6} \|d^k\|^2 - \frac{\mu}{2} \|d^k\|^2$$

$$\text{よって, } \omega(\mu) = \langle g^k, d^k \rangle + \frac{1}{2} (d^k)^\top H_k d^k + \frac{\mu}{2} \|d^k\|^2 + \frac{M}{6} \|d^k\|^2 - \frac{\mu}{2} \|d^k\|^2 = \langle g^k, d^k \rangle + \frac{1}{2} (d^k)^\top H_k d^k + \frac{M}{6} \|d^k\|^2$$

おまけ：SVMの解法

SVMも(カーネルトリックを使った)双対問題を解く.

このとき, 交互方向法を一般化した座標降下法を使うことが多い.

SVMにおいては, サポートベクターでないデータに対応するラグランジュ乗数 α_i は最適解において0または C になる.

つまり双対問題の決定変数のほとんどは0になる.

SVMに対する座標降下法はそのような性質を活用している.

概要

1. 双対な関係のアルゴリズム

2. 双対問題を解くアルゴリズム

Sinkhorn algorithm

Trust-region subproblem, 3次正則化ニュートン法

3. 双対性を利用した分解アルゴリズム・並列アルゴリズム

並列アルゴリズム

Benders分解法（時間があれば）

連続最適化における並列計算

パターン① 典型的な数値計算に並列計算機を利用

最適化アルゴリズムの中に現れる数値計算，特に行列計算で利用GPUなど，使いやすい環境がそろっている．

例： 行列とベクトルの積 $\Rightarrow$ 共役勾配法(線形方程式の解法)

パターン② 双対性を利用して問題を小さい部分問題に分解して解く

複数の計算機で解く．計算機ごとに全情報(データ)を持つ必要はない．

応用例： 分散制御，連合学習(federated learning)

分解方法：制約条件で繋がっている問題

$$\begin{aligned} \min \quad & \sum_{k=1}^K f_k(x_k) \\ \text{s.t.} \quad & x_k \in X_k, k=1, \dots, K \\ & \sum_{k=1}^K A_k x_k = b \end{aligned}$$

応用例：電力の発電計画

ラグランジュ緩和(双対問題の目的関数)

$$\begin{aligned} \min \quad & -b^\top \lambda + \sum_{k=1}^K (f_k(x_k) + \lambda^\top A_k x_k) \\ \text{s.t.} \quad & x_k \in X_k, k=1, \dots, K \end{aligned}$$



$$\begin{aligned} P_k(\lambda): \min \quad & f_k(x_k) + \lambda^\top A_k x_k \\ \text{s.t.} \quad & x_k \in X_k \end{aligned}$$

劣勾配法で双対問題を解く

ステップ 1 : $P_k(\lambda^k), k = 1, \dots, K$ を並列で解く.

ステップ 2 : $P_k(\lambda^k), k = 1, \dots, K$ の最適解から,
双対問題の目的関数の劣勾配を構成

ステップ 3 : λ^k を更新してステップ 1 へ.

λ^k は等式制約の潜在価格.

劣勾配法で調整することで等式制約をみたすようにする.

注意 : 劣勾配法は遅い. 拡張ラグランジュ関数だと分解できない.

分解方法：変数で繋がっているとき

$$\begin{array}{ll} \min & \sum_{k=1}^K f_k(x) \\ \text{s.t.} & Ax = b \end{array}$$

応用：連合学習(Federated Learning)

ADMMを使うとすると..

$$\begin{array}{ll} \min & \sum_{k=1}^K f_k(x_k) \\ \text{s.t.} & x_k = y, k=1, \dots, K \\ & Ay = b \end{array}$$

拡張ラグランジュ関数

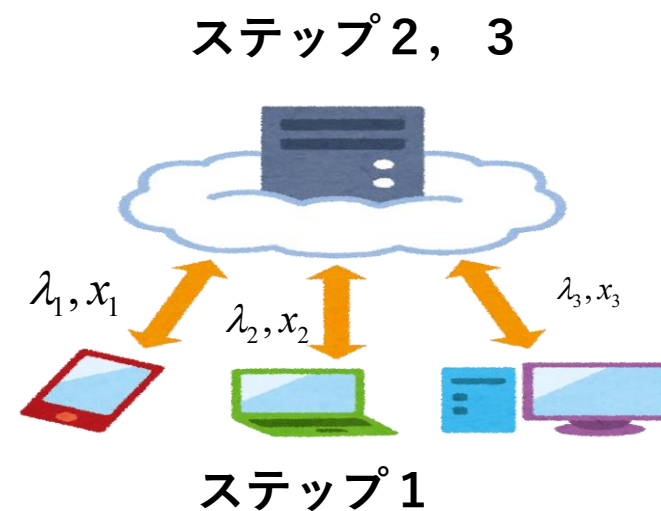
$$L_c(x_1, \dots, x_K, y, \lambda_1, \lambda_2, \dots, \lambda_K, \lambda_y) = \sum_{k=1}^K \left(f_k(x_k) + \langle \lambda_k, x_k - y \rangle + \frac{c}{2} \|x_k - y\|^2 \right) + \langle \lambda_y, Ay - b \rangle + \frac{c}{2} \|Ay - b\|^2$$

ADMM

ステップ 1 :
$$x_k^{\ell+1} = \operatorname{argmin} \left\{ f_k(x_k) + \langle \lambda_k^\ell, x_k - y \rangle + \frac{c}{2} \|x_k - y^\ell\|^2 \right\}, \quad k = 1, \dots, K$$

ステップ 2 :
$$y^{\ell+1} = \operatorname{argmin} \left\{ \sum_{k=1}^K \left(-\langle \lambda_k^\ell, y \rangle + \frac{c}{2} \|y - x_k^{\ell+1}\|^2 \right) + \langle \lambda_y^\ell, Ay - b \rangle + \frac{c}{2} \|Ay - b\|^2 \right\}$$

ステップ 3 :
$$\lambda_k^{\ell+1} = \lambda_k^\ell + c(x_k^{\ell+1} - y^{\ell+1}), \quad k = 1, \dots, K$$
$$\lambda_y^{\ell+1} = \lambda_y^\ell + c(Ay^{\ell+1} - b)$$



他にもいろいろな解き方がある。泥臭い

概要

1. 双対な関係のアルゴリズム

2. 双対問題を解くアルゴリズム

Sinkhorn algorithm

Trust-region subproblem, 3 次正則化ニュートン法

3. 双対性を利用した分解アルゴリズム・並列アルゴリズム

並列アルゴリズム

Benders分解法（時間があれば）

いまさらBenders分解法？

- 現実の問題では**複数の最適化問題の融合問題**が多い。
例：0-1整数最適化問題+SDP
- 大域的最適解が求まることが保証されている。
そこそこよい解が早い段階で求まる。
- 個々の最適化問題に対して効率的にとけるソルバーが充実してきている。
⇒ **実装が簡単**

実用的なテクニック．卒論，修論にお勧め

Benders分解法の概要

考える問題

$$\begin{aligned} \min & f(x) + h(x, y) \\ \text{s.t. } & x \in S \\ & y \in \Omega(x) \end{aligned}$$



等価な問題（マスター問題）

$$\begin{aligned} \min & f(x) + \theta(x) \\ \text{s.t. } & x \in S \end{aligned}$$

ただし, θ は最適値関数 $\theta(x) = \min_{y \in \Omega(x)} h(x, y)$

Benders分解法

1. x^k を固定した子問題 $\theta(x^k) = \min_{y \in \Omega(x^k)} h(x^k, y)$ を解く.
2. $\theta(x^k)$ の関数値, 微分を求め, マスター問題(等価な問題)の**近似問題(部分問題)**を構成しその最適解 x^{k+1} を求める

Benders分解法 = 切除平面法

最適値関数 $\theta(x)$ が**凸関数のとき**  大事！！

$$\theta(x) \geq \theta(x^k) + \langle \eta^k, x - x^k \rangle, \eta^k \in \partial\theta(x^k)$$

よって、 k 回反復までの $\{(\theta(x^i), \eta^i)\}$ を用いれば、
 $\theta(x)$ の下界を与える関数 $\theta^k(x)$ が得られる。

$$\theta(x) \geq \theta^k(x) := \max_{1 \leq i \leq k} \left\{ \theta(x^i) + \langle \eta^i, x - x^i \rangle \right\}$$

マスター問題の近似問題(部分問題)

部分問題

$$\begin{array}{ll}\min & f(x) + \theta^k(x) \\ \text{s.t.} & x \in S\end{array}$$



$$\begin{array}{ll}\min & f(x) + t \\ \text{s.t.} & t \geq \theta(x^i) + \langle \eta^i, x - x^i \rangle \quad \forall i \\ & x \in S\end{array}$$

ベンダーズカットという

- f が線形関数, $S = Z^n$ であれば, 部分問題は混合整数最適化問題
⇒ GurobiなどのMIPソルバーで解ける.
- $\theta(x) = \min_{y \in \Omega(x)} h(x, y)$ が凸最適化問題であれば, 凸最適化ソルバーで解ける.

うまく分解できれば, 既存ソルバーの組合せで解ける.

Benders分解法が実行できるための仮定：

$\min_{y \in \Omega(x)} h(x, y)$ が凸最適化となり, $\theta(x)$ が凸関数となる.

この仮定が成り立つための十分条件：

- $h(x, y)$ は jointly convex.
- $\Omega(x)$ は jointly convex な関数 $g_j(x, y), j = 1, \dots, m$ を用いて

$$\Omega(x) = \{y \mid g_j(x, y) \leq 0, j = 1, \dots, r\}$$

と表される.

これ以降のスライドではこの十分条件を仮定する.

最適値関数 $\theta(x) = \min_{y \in \Omega(x)} h(x, y)$ が計算できないときは？

マスター問題の近似問題では、 $\Omega(x)$ の実行可能性を考慮していない。
そのため、 $\Omega(x^k) = \emptyset$ となることがある。
このとき、 $\theta(x^k)$ を計算できない。

そのようなときは、マスター問題の近似問題にベンダーズカットを加えるのではなく、実行可能になるような制約条件を加える。

そのような制約条件を **feasibility cut** という。

feasibility cut

$$\phi(x) = \min_y \max_i \{g_i(x, y)\} \text{ とする.}$$

- $\Omega(x) \neq \emptyset \Leftrightarrow \phi(x) \leq 0$
- ϕ は凸関数.

実行不可能なとき, つまり $\phi(x^k) > 0$ のときは次のfeasibility cutを加える.

$$0 \geq \phi(x^k) + \langle \xi^k, x - x^k \rangle, \quad \xi^k \in \partial \phi(x^k)$$

上界値と下界値

上界値

$\min_{y \in \Omega(x^k)} h(x^k, y)$ が実行可能な反復のとき、つまり $\phi(x^k) \leq 0$ のとき、
 (y^{k+1}, x^k) は元の問題の実行可能解である。ただし、 $y^{k+1} \in \arg \min_{y \in \Omega(x^k)} h(x^k, y)$ である。
そのため、 $f(x^k) + h(x^k, y^{k+1})$ は元の問題の上界値を与える。

下界値

$$\theta(x) \geq \theta^k(x) = \max_i \left\{ \theta(x^i) + \langle \eta^i, x - x^i \rangle \right\}$$

が成り立つため、マスター問題の近似問題の最適値は元の問題の下界値を与える。

Benders分解法

Step 0 : 下界値 $\text{lb} = -\infty$, 上界値 $\text{ub} = +\infty$, ベンダーズカットの集合 $B^0 = \emptyset$
feasibility cutの集合 $F^0 = \emptyset$, $x^0 \in R^n, k = 0$.

Step 1 : $\Omega(x^k) \neq \emptyset$ であれば, $(y^{k+1}, \theta(x^k), \eta^k)$ を求め, $B^{k+1} = B^k \cup \{k\}, F^{k+1} = F^k$
とし, $\text{ub} = \min\{\text{ub}, f(x^k) + h(x^k, y^{k+1})\}$ とする. そうでなければ
 $(\phi(x^k), \xi^k)$ を求め, $B^{k+1} = B^k, F^{k+1} = F^k \cup \{k\}$ とする.

Step 2 : 次の問題の最適解を (x^{k+1}, t^*) を求め, $\text{lb} = \max\{\text{lb}, f(x^{k+1}) + t^*\}$ とする.

$$\min f(x) + t$$

$$\text{s.t. } t \geq \theta(x^i) + \langle \eta^i, x - x^i \rangle \quad \forall i \in B^{k+1}$$

$$0 \geq \phi(x^j) + \langle \xi^j, x - x^j \rangle \quad \forall j \in F^{k+1}$$

$$x \in S$$

Step 3 : 上界値 ub と下界値 lb の差が小さければ終了.

Benders分解法の収束性

Flippo, O.E. and Rinnoy Kan, A.H.G. Decomposition in general mathematical programming. Mathematical Programming, Vol. 60, 361–382, 1993.

以下の二つの条件を満たすとき、Benders分解法で生成される点列は大域的最適解に収束する。

- $h, g_j, j = 1, \dots, m$ は jointly convex.
- S が有限集合(整数計画)になるか,
KKT点のラグランジュ乗数が有界(連続最適化)
 S が有限集合のときは、有限回の反復で終了する。

Benders分解法の応用例

スパース最適化

Bertsimas, Dimitris and Cory-Wright, Ryan and Pauphilet, Jean, A Unified Approach to Mixed-Integer Optimization Problems With Logical Constraints, SIAM Journal on Optimization, vol. 31, pp.2340-2367, 2021.

MPEC(均衡制約つき最適化問題)

Hu, Jing and Mitchell, John E. and Pang, Jong-Shi and Bennett, Kristin P. and Kunapuli, Gautam, On the Global Solution of Linear Programs with Linear Complementarity Constraints, SIAM Journal on Optimization, vol. 19, pp. 445-471, 2008.

多項式最適化

Kleniati, P.M., Parpas, P. & Rustem, B. Partitioning procedure for polynomial optimization. *J Glob Optim.*, Vol. 48, 549–567 (2010).