Data-Driven Optimal Control:

An Inverse Optimization Model and Algorithm

Long, Youyuan

Supervisor: Peyman Mohajerin Esfahani

Daily Supervisors: Tolga Ok

Pedro Zattoni Scroccaro

TU Delft

Contents

- Introduction of Inverse Optimization (IO)
- Kernel Inverse Optimization Machine (KIOM)
- Sequential Selection Optimization (SSO)
- Conclusion and Future Study

TU Delft

Introduction of Inverse Optimization (IO)

Introduction of Inverse Optimization

In Inverse Optimization (IO), it is hypothesized that experts, when making decisions, implicitly engage in solving an optimization problem. If we can reconstruct this optimization problem using the decision data $\{(\hat{s}_i, \hat{u}_i)\}_{i=1}^N$ of the expert, then the behavior of the expert can be imitated.

TU Delft

Introduction of Inverse Optimization

In Inverse Optimization (IO), it is hypothesized that experts, when making decisions, implicitly engage in solving an optimization problem. If we can reconstruct this optimization problem using the decision data $\{(\hat{s}_i, \hat{u}_i)\}_{i=1}^N$ of the expert, then the behavior of the expert can be imitated.

The three crucial components in data-driven IO:

- Forward Optimization Problem/Model
- Loss Function
- Inverse Optimization Problem and Algorithm

Learning for Control: An Inverse Optimization Model [1]

[1] S. A. Akhtar, A. S. Kolarijani, and P. M. Esfahani (2021). "Learning for control: An inverse optimization approach". In: *IEEE Control Systems Letters* 6, pp. 187–192

Learning for Control: An Inverse Optimization Model [1]

- A quadratic forward model

$$f(s, u, \theta) := \begin{bmatrix} s \\ u \end{bmatrix}^T \theta \begin{bmatrix} s \\ u \end{bmatrix} \tag{1}$$

$$\begin{aligned} \mathbf{FOP}(s \mid \theta) := \min_{u} \ & f(s, u, \theta) \\ \text{s.t. } & M(s)u \leq W(s), \end{aligned} \tag{2}$$

where $s \in R^m$, $u \in R^n$, $M(s) \in R^{d \times m}$ and $W(s) \in R^d$.

[1] S. A. Akhtar, A. S. Kolarijani, and P. M. Esfahani (2021). “Learning for control: An inverse optimization approach”. In: *IEEE Control Systems Letters* 6, pp. 187–192

Learning for Control: An Inverse Optimization Model [1]

- A quadratic forward model

$$f(s,u,\theta) := \begin{bmatrix} s \\ u \end{bmatrix}^T \theta \begin{bmatrix} s \\ u \end{bmatrix} \tag{1}$$

$$\begin{aligned} \textbf{FOP}(s \mid \theta) := \min_u \; & f(s,u,\theta) \\ \text{s.t. } & M(s)u \leq W(s), \end{aligned} \tag{2}$$

where $s \in R^m$, $u \in R^n$, $M(s) \in R^{d \times m}$ and $W(s) \in R^d$.To ensure the convexity of the FOP, extra constraints should be imposed on the range of values for θ:

$$\begin{aligned} &\theta \in \Theta\,, \\ &where\ \Theta = \left\{ \theta = \begin{bmatrix} 0 & \theta_{su} \\ * & \theta_{uu} \end{bmatrix} \middle| \theta_{uu} \succeq I_n \right\}. \end{aligned} \tag{3}$$

TUDelft

[1] S. A. Akhtar, A. S. Kolarijani, and P. M. Esfahani (2021). "Learning for control: An inverse optimization approach". In: *IEEE Control Systems Letters* 6, pp. 187–192

- Suboptimality loss function

- Suboptimality loss function
  Let $\{(\hat{s}_i, \hat{u}_i)\}_{i=1}^{N}$ be the dataset[2] and the suboptimality loss is defined as:

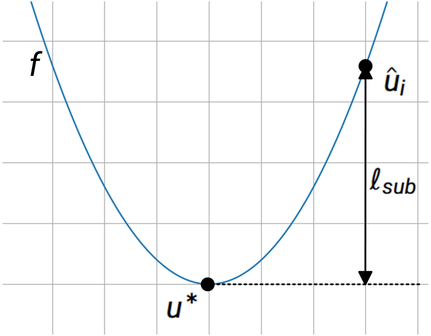
$$\begin{aligned} \ell_{sub}(\hat{s}_i, \hat{u}_i) &:= f(\hat{s}_i, \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} f(\hat{s}_i, u, \theta) \\ \textit{where}, \ U(\hat{s}_i) &= \{u \in R^n \mid M(\hat{s}_i)u \leq W(\hat{s}_i)\}. \end{aligned} \tag{4}$$

TU Delft

[2] The dataset is assumed to be consistent with its feasible set, i.e., $M(\hat{s}_i)\hat{u}_i \leq W(\hat{s}_i)$.

- Suboptimality loss function
  Let $\{(\hat{s}_i, \hat{u}_i)\}_{i=1}^{N}$ be the dataset[2] and the suboptimality loss is defined as:

$$\begin{aligned} \ell_{sub}(\hat{s}_i, \hat{u}_i) &:= f(\hat{s}_i, \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} f(\hat{s}_i, u, \theta) \\ \textit{where}, \ U(\hat{s}_i) &= \{u \in R^n \mid M(\hat{s}_i)u \leq W(\hat{s}_i)\}. \end{aligned} \tag{4}$$



[2] The dataset is assumed to be consistent with its feasible set, i.e., $M(\hat{s}_i)\hat{u}_i \leq W(\hat{s}_i)$.

- Suboptimality loss function
  Let $\{(\hat{s}_i, \hat{u}_i)\}_{i=1}^{N}$ be the dataset[2] and the suboptimality loss is defined as:

$$\ell_{sub}(\hat{s}_i, \hat{u}_i) := f(\hat{s}_i, \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} f(\hat{s}_i, u, \theta)$$
$$where, \ U(\hat{s}_i) = \{u \in R^n \mid M(\hat{s}_i)u \leq W(\hat{s}_i)\}. \tag{4}$$

- Inverse optimization problem

TU Delft

[2] The dataset is assumed to be consistent with its feasible set, i.e., $M(\hat{s}_i)\hat{u}_i \leq W(\hat{s}_i)$.

- Suboptimality loss function
  Let $\{(\hat{s}_i, \hat{u}_i)\}_{i=1}^N$ be the dataset[2] and the suboptimality loss is defined as:

$$\ell_{sub}(\hat{s}_i, \hat{u}_i) := f(\hat{s}_i, \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} f(\hat{s}_i, u, \theta)$$
$$\text{where, } U(\hat{s}_i) = \{u \in R^n \mid M(\hat{s}_i)u \leq W(\hat{s}_i)\}. \tag{4}$$

- Inverse optimization problem

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left\{ f(\hat{s}_i, \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} f(\hat{s}_i, u, \theta) \right\} \tag{5}$$
$$\text{s.t. } \theta \in \Theta \text{ in (3).}$$

[2] The dataset is assumed to be consistent with its feasible set, i.e., $M(\hat{s}_i)\hat{u}_i \leq W(\hat{s}_i)$.

- Suboptimality loss function
  Let $\{(\hat{s}_i, \hat{u}_i)\}_{i=1}^{N}$ be the dataset[2] and the suboptimality loss is defined as:

$$\begin{aligned} \ell_{sub}(\hat{s}_i, \hat{u}_i) &:= f(\hat{s}_i, \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} f(\hat{s}_i, u, \theta) \\ \textit{where},\ U(\hat{s}_i) &= \{u \in R^n \mid M(\hat{s}_i)u \leq W(\hat{s}_i)\}. \end{aligned} \tag{4}$$

- Inverse optimization problem

$$\begin{aligned} &\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left\{ f(\hat{s}_i, \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} f(\hat{s}_i, u, \theta) \right\} \\ &\text{s.t. } \theta \in \Theta \text{ in (3)}. \end{aligned} \tag{5}$$

The inverse optimization problem (5) is convex as the objective function is a pointwise maximum of infinitely many linear functions and the constraint is a Linear Matrix Inequality (LMI).

[2] The dataset is assumed to be consistent with its feasible set, i.e., $M(\hat{s}_i)\hat{u}_i \leq W(\hat{s}_i)$.

Theorem 1 (LMI Reformulation [3])

For the quadratic hypothesis function (1) *and the constraints* (3) *on* θ*, the inverse optimization problem* (5) *is equivalent to*

$$
\begin{aligned}
\min_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} \quad & \frac{1}{N}\sum_{i=1}^{N}\left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\hat{s}_i^T\theta_{su}\hat{u}_i + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\
\text{s.t.} \quad & \theta_{uu} \succeq I_n,\ \lambda_i \in R_+^d,\ \gamma_i \in R, \quad \forall i \le N \\
& \begin{bmatrix} \theta_{uu} & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^T\hat{s}_i \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \le N,
\end{aligned}
\tag{6}
$$

where λ_i *is the Lagrange multiplier and* γ_i *is the slack variable.*

TUDelft

[3] S. A. Akhtar, A. S. Kolarijani, and P. M. Esfahani (2021). "Learning for control: An inverse optimization approach". In: *IEEE Control Systems Letters* 6, pp. 187–192

Analysis of the model’s limitations

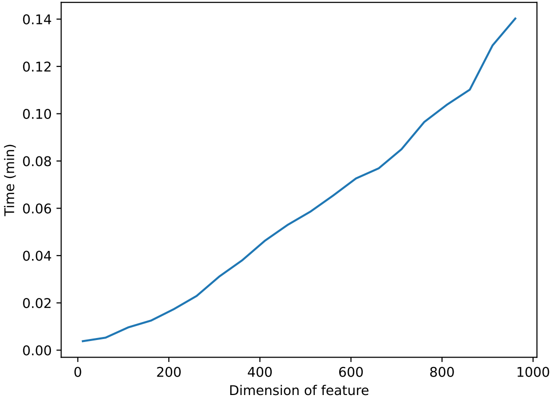
Analysis of the model's limitations

To enhance the model's capacity, a mapping function, $\phi(\cdot) : R^m \mapsto R^l$, is utilized to map the state s into a higher-dimensional feature space $\phi(s)$.

Analysis of the model's limitations

To enhance the model's capacity, a mapping function, $\phi(\cdot) : R^m \longmapsto R^l$, is utilized to map the state s into a higher-dimensional feature space $\phi(s)$.

- Increasing computational burden
- Challenges of feature engineering



Kernel Inverse Optimization Machine

(KIOM)

Theoretical derivation

Theoretical derivation

- Modified IO problem

$$\min_{\theta} k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left\{ f(\phi(\hat{s}_i), \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} f(\phi(\hat{s}_i), u, \theta) \right\} \tag{7}$$

s.t. $\theta \in \Theta$ in (3).

TU Delft

Theoretical derivation

- Modified IO problem

$$\min_{\theta} k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left\{f(\phi(\hat{s}_i), \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} f(\phi(\hat{s}_i), u, \theta)\right\} \tag{7}$$
$$\text{s.t. } \theta \in \Theta \text{ in (3).}$$

- Modified LMI reformulation[4]

$$\begin{aligned}
\min_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} \quad & k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 \\
& + \frac{1}{N}\sum_{i=1}^{N}\left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\phi(\hat{s}_i)^T\theta_{su}\hat{u}_i + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\
\text{s.t.} \quad & \theta_{uu} \succeq I_n,\ \lambda_i \in R_+^d,\ \gamma_i \in R, \quad \forall i \le N \\
& \begin{bmatrix} \theta_{uu} & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \le N.
\end{aligned} \tag{8}$$

[4]The derivation of (8) follows the same procedure as that of Theorem 1.

Theorem 2 (Kernel Inverse Optimization Machine)

For the quadratic hypothesis function (1) *and the constraints* (3) *on* θ, *the modified inverse optimization problem* (8) *can be reformulated as*

$$
\begin{aligned}
\min_{P,\Lambda_i,\Gamma_i} \quad & \frac{1}{4k}\left\|\left(\sum_{i=1}^{N}\frac{\hat{u}_i\hat{u}_i^T}{N}-\Lambda_i\right)-P\right\|_F^2 \\
& +\frac{1}{k}\begin{bmatrix} m_1 & m_2 & \dots & m_N \end{bmatrix}(K\otimes I_n)\begin{bmatrix} m_1 & m_2 & \dots & m_N \end{bmatrix}^T-\mathrm{Tr}(P) \\
\text{s.t.} \quad & P\succeq 0,\ \frac{W(\hat{s}_i)}{N}-2M(\hat{s}_i)\Gamma_i\geq 0, \quad \forall i\leq N \\
& \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix}\succeq 0, \quad \forall i\leq N,
\end{aligned}
\tag{9}
$$

where $m_i := \frac{\hat{u}_i^T}{N}-2{\Gamma_i}^T$ *for* $i\in\{1,\dots,N\}$. *K is the Gram matrix with respect to* $\hat{s}_1,\dots,\hat{s}_N$, *so* $K\in R^{N\times N}$ *and* $K_{ij}=\kappa(\hat{s}_i,\hat{s}_j)=\phi(\hat{s}_i)^T\phi(\hat{s}_j)$ *which is the inner product of the augmented states, and decision variables* $P,\Lambda_i\in R^{n\times n}$, $\Gamma_i\in R^n$ *for* $i\in\{1,\dots,N\}$. *The expressions of matrix* θ_{uu} *and* θ_{su} *are*

$$
\theta_{uu}=-\frac{\left(\sum_{i=1}^{N}\frac{\hat{u}_i\hat{u}_i^T}{N}-\Lambda_i\right)-P}{2k} \qquad \theta_{su}=-\frac{\sum_{i=1}^{N}\phi(\hat{s}_i)\left(\frac{\hat{u}_i^T}{N}-2{\Gamma_i}^T\right)}{k},
$$

where the weight θ_{su} *is a linear combination of the augmented states.*

Proof

Proof

Modified IO problem (Primal problem):

$$\begin{array}{cl}\min\limits_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} & k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^N \left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\phi(\hat{s}_i)^T\theta_{su}\hat{u}_i + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\ \text{s.t.} & \theta_{uu} \succeq I_n, \; --- P \\ & \lambda_i \in R_+^d, \quad \forall i \leq N --- \; \tilde{\lambda}_i \\ & \begin{bmatrix} \theta_{uu} & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \leq N --- \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \alpha_i \end{bmatrix}.\end{array}$$

Proof

Modified IO problem (Primal problem):

$$\begin{array}{rl} \min\limits_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} & k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^N \left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\phi(\hat{s}_i)^T\theta_{su}\hat{u}_i + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\ \text{s.t.} & \theta_{uu} \succeq I_n, \quad --- \; P \\ & \lambda_i \in R_+^d, \quad \forall i \le N \; --- \; \tilde{\lambda}_i \\ & \begin{bmatrix} \theta_{uu} & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \le N \; --- \; \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \alpha_i \end{bmatrix}. \end{array}$$

Lagrangian function:

$$L(\theta_{uu},\theta_{su},\lambda_i,\gamma_i,P,\tilde{\lambda}_i,\Lambda_i,\Gamma_i,\alpha_i) = k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^N\Big(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i$$

$$+W^\top\lambda_i\Big) - \mathrm{Tr}\left(P(\theta_{uu}-I_n)\right) + \sum_{i=1}^N \tilde{\lambda}_i^T(-\lambda_i) + \sum_{i=1}^N -\mathrm{Tr}\left(\begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \alpha_i \end{bmatrix}\begin{bmatrix} \theta_{uu} & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix}\right).$$

Proof

Modified IO problem (Primal problem):

$$\begin{array}{ll}\min\limits_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} & k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^N \left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\phi(\hat{s}_i)^T\theta_{su}\hat{u}_i + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top \lambda_i\right) \\ \text{s.t.} & \theta_{uu} \succeq I_n, \; ---P \\ & \lambda_i \in R_+^d, \quad \forall i \leq N --- \;\tilde{\lambda}_i \\ & \begin{bmatrix} \theta_{uu} & M(\hat{s}_i)^\top \lambda_i + 2\theta_{su}^\top \phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \leq N --- \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \alpha_i \end{bmatrix}.\end{array}$$

Lagrangian function:

$$L(\theta_{uu}, \theta_{su}, \lambda_i, \gamma_i, P, \tilde{\lambda}_i, \Lambda_i, \Gamma_i, \alpha_i) = k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^N \Big(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i$$

$$+W^\top\lambda_i\Big) - \mathrm{Tr}\left(P(\theta_{uu} - I_n)\right) + \sum_{i=1}^N \tilde{\lambda}_i^{\,T}(-\lambda_i) + \sum_{i=1}^N -\mathrm{Tr}\left(\begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \alpha_i \end{bmatrix}\begin{bmatrix} \theta_{uu} & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix}\right).$$

Lagrangian dual problem:

$$\begin{array}{ll}\max\limits_{P,\tilde{\lambda}_i,\Lambda_i,\Gamma_i,\alpha_i} & \inf\limits_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} L\left(\theta_{uu}, \theta_{su}, \lambda_i, \gamma_i, P, \tilde{\lambda}_i, \Lambda_i, \Gamma_i, \alpha_i\right) \\ \text{s.t.} & P \succeq 0, \; \tilde{\lambda}_i \in R_+^d, \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \alpha_i \end{bmatrix} \succeq 0, \quad \forall i \leq N.\end{array}$$

Proof

Modified IO problem (Primal problem):

$$\begin{aligned}
\min_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} \quad & k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^N \left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\phi(\hat{s}_i)^T\theta_{su}\hat{u}_i + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\
\text{s.t.} \quad & \theta_{uu} \succeq I_n, \quad --- \; P \\
& \lambda_i \in R_+^d, \quad \forall i \le N \; --- \; \tilde{\lambda}_i \\
& \begin{bmatrix} \theta_{uu} & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \le N \; --- \; \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \alpha_i \end{bmatrix}.
\end{aligned}$$

Lagrangian function:

$$L(\theta_{uu},\theta_{su},\lambda_i,\gamma_i,P,\tilde{\lambda}_i,\Lambda_i,\Gamma_i,\alpha_i) = k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^N\left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i\right.$$
$$\left. + W^\top\lambda_i\right) - \mathrm{Tr}\left(P(\theta_{uu} - I_n)\right) + \sum_{i=1}^N \tilde{\lambda}_i^{\,T}(-\lambda_i) + \sum_{i=1}^N -\mathrm{Tr}\left(\begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \alpha_i \end{bmatrix}\begin{bmatrix} \theta_{uu} & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix}\right).$$

Lagrangian dual problem:

$$\begin{aligned}
\max_{P,\tilde{\lambda}_i,\Lambda_i,\Gamma_i,\alpha_i} \quad & \inf_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} L\left(\theta_{uu},\theta_{su},\lambda_i,\gamma_i,P,\tilde{\lambda}_i,\Lambda_i,\Gamma_i,\alpha_i\right) \\
\text{s.t.} \quad & P \succeq 0, \; \tilde{\lambda}_i \in R_+^d, \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \alpha_i \end{bmatrix} \succeq 0, \quad \forall i \le N.
\end{aligned}$$

$$\frac{\partial L}{\partial \theta_{uu}}, \frac{\partial L}{\partial \theta_{su}}, \frac{\partial L}{\partial \lambda_i}, \frac{\partial L}{\partial \gamma_i} = 0$$

Proof

$$L(\theta_{uu}, \theta_{su}, \lambda_i, \gamma_i, P, \tilde{\lambda}_i, \Lambda_i, \Gamma_i, \alpha_i) = k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\Big(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i$$

$$+W^\top\lambda_i\Big) - \operatorname{Tr}\left(P(\theta_{uu} - I_n)\right) + \sum_{i=1}^{N}\tilde{\lambda}_i^{\,T}(-\lambda_i) + \sum_{i=1}^{N} -\operatorname{Tr}\left(\begin{bmatrix}\Lambda_i & \Gamma_i \\ * & \alpha_i\end{bmatrix}\begin{bmatrix}\theta_{uu} & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i\end{bmatrix}\right).$$

Proof

$$L(\theta_{uu}, \theta_{su}, \lambda_i, \gamma_i, P, \tilde{\lambda}_i, \Lambda_i, \Gamma_i, \alpha_i) = k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\Big(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i$$

$$+W^\top\lambda_i\Big) - \operatorname{Tr}\left(P(\theta_{uu} - I_n)\right) + \sum_{i=1}^{N}\tilde{\lambda}_i^{\,T}(-\lambda_i) + \sum_{i=1}^{N} -\operatorname{Tr}\left(\begin{bmatrix}\Lambda_i & \Gamma_i \\ * & \alpha_i\end{bmatrix}\begin{bmatrix}\theta_{uu} & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i\end{bmatrix}\right).$$

$$\frac{\partial L}{\partial \theta_{uu}} = 2k\theta_{uu} + \frac{1}{N}\sum_{i=1}^{N}\hat{u}_i\hat{u}_i^T - P + \sum_{i=1}^{N} -\Lambda_i = 0 \Rightarrow \theta_{uu} = -\frac{(\sum_{i=1}^{N}\frac{\hat{u}_i\hat{u}_i^T}{N} - \Lambda_i) - P}{2k}$$

$$\frac{\partial L}{\partial \theta_{su}} = 2k\theta_{su} + 2\sum_{i=1}^{N}\phi(\hat{s}_i)(\frac{\hat{u}_i^T}{N} - 2\Gamma_i^T) = 0 \Rightarrow \theta_{su} = -\frac{\sum_{i=1}^{N}\phi(\hat{s}_i)(\frac{\hat{u}_i^T}{N} - 2\Gamma_i^T)}{k}$$

$$\frac{\partial L}{\partial \lambda_i} = \frac{W}{N} - \tilde{\lambda}_i - 2M\Gamma_i = 0 \Rightarrow \tilde{\lambda}_i = \frac{W}{N} - 2M\Gamma_i$$

$$\frac{\partial L}{\partial \gamma_i} = \frac{1}{4N} - \alpha_i = 0 \Rightarrow \alpha_i = \frac{1}{4N}$$

Proof

$$L(\theta_{uu}, \theta_{su}, \lambda_i, \gamma_i, P, \tilde{\lambda}_i, \Lambda_i, \Gamma_i, \alpha_i) = k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\Big(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i$$
$$+W^\top\lambda_i\Big) - \mathrm{Tr}\left(P(\theta_{uu} - I_n)\right) + \sum_{i=1}^{N}\tilde{\lambda}_i^{\,T}(-\lambda_i) + \sum_{i=1}^{N} -\mathrm{Tr}\left(\begin{bmatrix}\Lambda_i & \Gamma_i \\ * & \alpha_i\end{bmatrix}\begin{bmatrix}\theta_{uu} & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i\end{bmatrix}\right).$$

$$\frac{\partial L}{\partial \theta_{uu}} = 2k\theta_{uu} + \frac{1}{N}\sum_{i=1}^{N}\hat{u}_i\hat{u}_i^T - P + \sum_{i=1}^{N} -\Lambda_i = 0 \Rightarrow \theta_{uu} = -\frac{(\sum_{i=1}^{N}\frac{\hat{u}_i\hat{u}_i^T}{N} - \Lambda_i) - P}{2k}$$

$$\frac{\partial L}{\partial \theta_{su}} = 2k\theta_{su} + 2\sum_{i=1}^{N}\phi(\hat{s}_i)(\frac{\hat{u}_i^T}{N} - 2\Gamma_i^T) = 0 \Rightarrow \theta_{su} = -\frac{\sum_{i=1}^{N}\phi(\hat{s}_i)(\frac{\hat{u}_i^T}{N} - 2\Gamma_i^T)}{k}$$

$$\frac{\partial L}{\partial \lambda_i} = \frac{W}{N} - \tilde{\lambda}_i - 2M\Gamma_i = 0 \Rightarrow \tilde{\lambda}_i = \frac{W}{N} - 2M\Gamma_i$$

$$\frac{\partial L}{\partial \gamma_i} = \frac{1}{4N} - \alpha_i = 0 \Rightarrow \alpha_i = \frac{1}{4N}$$

By substituting the expressions for θ_{uu}, θ_{su}, $\tilde{\lambda}_i$ and α_i into the Lagrange dual problem and simplifying it, we obtain Theorem 2. □

Theorem 2 (Kernel Inverse Optimization Machine)

For the quadratic hypothesis function (1) and the constraints (3) on θ, the modified inverse optimization problem (8) can be reformulated as

$$\begin{aligned}\min_{P,\Lambda_i,\Gamma_i} \quad & \frac{1}{4k}\left\|\left(\sum_{i=1}^{N}\frac{\hat{u}_i\hat{u}_i^T}{N}-\Lambda_i\right)-P\right\|_F^2 \\ & +\frac{1}{k}\begin{bmatrix} m_1 & m_2 & \dots & m_N\end{bmatrix}(K\otimes I_n)\begin{bmatrix} m_1 & m_2 & \dots & m_N\end{bmatrix}^T-\mathrm{Tr}(P) \\ \text{s.t.} \quad & P\succeq 0,\ \frac{W(\hat{s}_i)}{N}-2M(\hat{s}_i)\Gamma_i\geq 0, \quad \forall i\leq N \\ & \begin{bmatrix}\Lambda_i & \Gamma_i \\ * & \frac{1}{4N}\end{bmatrix}\succeq 0, \quad \forall i\leq N,\end{aligned} \tag{9}$$

where $m_i := \frac{\hat{u}_i^T}{N}-2{\Gamma_i}^T$ for $i\in\{1,\dots,N\}$. K is the Gram matrix with respect to $\hat{s}_1,\dots,\hat{s}_N$, so $K\in R^{N\times N}$ and $K_{ij}=\kappa(\hat{s}_i,\hat{s}_j)=\phi(\hat{s}_i)^T\phi(\hat{s}_j)$ which is the inner product of the augmented states, and decision variables $P,\Lambda_i\in R^{n\times n}$, $\Gamma_i\in R^n$ for $i\in\{1,\dots,N\}$. The expressions of matrix θ_{uu} and θ_{su} are

$$\theta_{uu}=-\frac{\left(\sum_{i=1}^{N}\frac{\hat{u}_i\hat{u}_i^T}{N}-\Lambda_i\right)-P}{2k} \qquad \theta_{su}=-\frac{\sum_{i=1}^{N}\phi(\hat{s}_i)\left(\frac{\hat{u}_i^T}{N}-2{\Gamma_i}^T\right)}{k},$$

where the weight θ_{su} is a linear combination of the augmented states.

KIOM for imitation learning

KIOM for imitation learning

After solving the KIOM problem (9), we can obtain the optimal variables P^*,Λ_i^*,Γ_i^* and then use these optimal variables to recover θ_{uu}^* and θ_{su}^*:

$$\theta_{uu} = -\frac{\left(\sum_{i=1}^{N} \frac{\hat{u}_i \hat{u}_i^T}{N} - \Lambda_i\right) - P}{2k} \quad \theta_{su} = -\frac{\sum_{i=1}^{N} \phi(\hat{s}_i) \left(\frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T\right)}{k}.$$

TU Delft

KIOM for imitation learning

After solving the KIOM problem (9), we can obtain the optimal variables $P^*, \Lambda_i^*, \Gamma_i^*$ and then use these optimal variables to recover θ_{uu}^* and θ_{su}^*:

$$\theta_{uu} = -\frac{\left(\sum_{i=1}^{N} \frac{\hat{u}_i \hat{u}_i^T}{N} - \Lambda_i\right) - P}{2k} \quad \theta_{su} = -\frac{\sum_{i=1}^{N} \phi(\hat{s}_i)\left(\frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T\right)}{k}.$$

Then, when confronted with a new state s_{new}, solving following FOP (10) and execute the optimal action u^* allows the agent to imitate the behavior of the expert:

$$\min_{M(s_{new})u \leq W(s_{new})} u^T \theta_{uu}^* u + 2\phi^T(s_{new})\theta_{su}^* u. \tag{10}$$

KIOM for imitation learning

After solving the KIOM problem (9), we can obtain the optimal variables $P^*, \Lambda_i^*, \Gamma_i^*$ and then use these optimal variables to recover θ_{uu}^* and θ_{su}^*:

$$\theta_{uu} = -\frac{\left(\sum_{i=1}^{N} \frac{\hat{u}_i \hat{u}_i^T}{N} - \Lambda_i\right) - P}{2k} \quad \theta_{su} = -\frac{\sum_{i=1}^{N} \phi(\hat{s}_i) \left(\frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T\right)}{k}.$$

Then, when confronted with a new state s_{new}, solving following FOP (10) and execute the optimal action u^* allows the agent to imitate the behavior of the expert:

$$\min_{M(s_{new})u \leq W(s_{new})} u^T \theta_{uu}^* u + 2\phi^T(s_{new}) \theta_{su}^* u. \tag{10}$$

Explicitly calculating θ_{su}^* may be infeasible. However, that is unnecessary. We only need to compute $\phi^T(s_{new})\theta_{su}^* = -\frac{1}{k}\sum_{i=1}^{N} \kappa(s_{new}, \hat{s}_i)(\frac{\hat{u}_i}{N} - 2\Gamma_i^*)^T$.

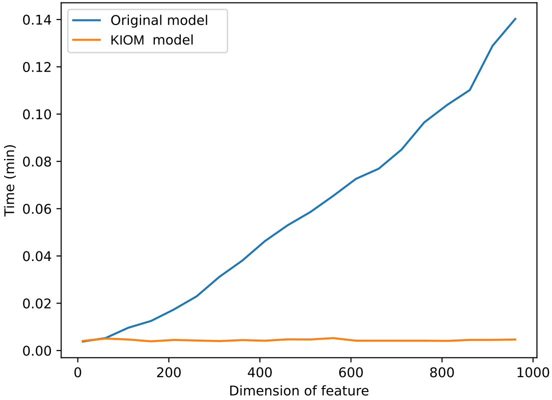
Does the KIOM model address the two deficiencies mentioned before?

Does the KIOM model address the two deficiencies mentioned before?

- Increasing computational burden
- Challenges of feature engineering

Does the KIOM model address the two deficiencies mentioned before?

- Increasing computational burden
- Challenges of feature engineering



Variant 1: A simpler version

Variant 1: A simpler version

In KIOM model, θ_{uu} is an $n \times n$ decision variable. However, in many experimental tests, we observed that the ultimately solved θ_{uu} is an identity matrix, which means the trained agent applies the same penalty coefficient to each dimension of the action.

Variant 1: A simpler version

In KIOM model, θ_{uu} is an $n \times n$ decision variable. However, in many experimental tests, we observed that the ultimately solved θ_{uu} is an identity matrix, which means the trained agent applies the same penalty coefficient to each dimension of the action.

Reason:

Reward Settings: same penalty on each dimension of the action $\xrightarrow{\text{RL Algorithm}}$ Expert: uniformly reduce the control effort for each dimension $\xrightarrow{\text{Sample}}$ $\{(\hat{s}_i, \hat{u}_i)\}_{i=1}^{N}$ $\xrightarrow{\text{IL: KIOM}}$ Agent ($\theta_{uu} = I_n$): Imitate the expert!

TU Delft

Variant 1: A simpler version

In KIOM model, θ_{uu} is an $n \times n$ decision variable. However, in many experimental tests, we observed that the ultimately solved θ_{uu} is an identity matrix, which means the trained agent applies the same penalty coefficient to each dimension of the action.

Reward Settings: same penalty on each dimension of the action $\xrightarrow{\text{RL Algorithm}}$ Expert: uniformly reduce the control effort for each dimension $\xrightarrow{\text{Sample}}$ $\{(\hat{s}_i, \hat{u}_i)\}_{i=1}^{N}$ $\xrightarrow{\text{IL: KIOM}}$ Agent ($\theta_{uu} = I_n$): Imitate the expert!

How to incorporate the prior knowledge of θ_{uu} into KIOM model?

Corollary 1 (A simpler model)

With the assumption $\theta_{uu} = I_n$, the KIOM model (9) can be simplified to

$$\begin{aligned}
\min_{\Lambda_i, \Gamma_i} \quad & \begin{bmatrix} m_1 & m_2 & \dots & m_N \end{bmatrix} \frac{K \otimes I}{k} \begin{bmatrix} m_1 & m_2 & \dots & m_N \end{bmatrix}^T + \sum_{i=1}^{N} \mathrm{Tr}(\Lambda_i) \\
\text{s.t.} \quad & \frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i \geq 0, \quad \forall i \leq N \\
& \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \succeq 0, \quad \forall i \leq N,
\end{aligned} \tag{11}$$

where $m_i = \frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T$ for $i \in \{1, \dots, N\}$, and K is the Gram matrix with respect to $\hat{s}_1, \dots, \hat{s}_N$, so $K \in R^{N \times N}$ and $K_{ij} = \kappa(\hat{s}_i, \hat{s}_j) = \phi(\hat{s}_i)^T \phi(\hat{s}_j)$. The expressions of matrix θ_{su} is

$$\theta_{su} = -\frac{\sum_{i=1}^{N} \phi(\hat{s}_i) \left(\frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T \right)}{k}.$$

Variant 2: An extension with ASL [5]

[5] P. Zattoni Scroccaro, B. Atasoy, and P. Mohajerin Esfahani (2023). “Learning in Inverse Optimization: Incenter Cost, Augmented Suboptimality Loss, and Algorithms”. In: *arXiv e-prints*, arXiv–2305

Variant 2: An extension with ASL [5]

Considering the quadratic objective function (1), the Augmented Suboptimality Loss (ALS) is defined as

$$\ell_{ASL}(\phi(\hat{s}_i), \hat{u}_i) = f(\phi(\hat{s}_i), \hat{u}_i, \theta) - \min_{u \in U(\hat{s}_i)} \{f(\phi(\hat{s}_i), u, \theta) - \|\hat{u}_i - u\|_\infty\}, \tag{12}$$

where $U(\hat{s}_i)$ is defined in (4). Note that in ASL, an additional penalty term, $\|\hat{u}_i - u\|_\infty$, for action divergence has been introduced.

[5] P. Zattoni Scroccaro, B. Atasoy, and P. Mohajerin Esfahani (2023). “Learning in Inverse Optimization: Incenter Cost, Augmented Suboptimality Loss, and Algorithms”. In: *arXiv e-prints*, arXiv–2305

Corollary 2 (An extension with ASL)

With ASL (12), the extended KIOM model is

$$
\begin{aligned}
\min_{\lambda_{ij},\Lambda_{ij},\Gamma_{ij}} \quad & \frac{1}{4k}\mathrm{Tr}\left(\sum_{i=1}^{N}\sum_{j=1}^{2n}(\lambda_{ij}\hat{u}_i\hat{u}_i^T-\Lambda_{ij})\sum_{i=1}^{N}\sum_{j=1}^{2n}(\lambda_{ij}\hat{u}_i\hat{u}_i^T-\Lambda_{ij})\right) \\
& +\frac{1}{2k}\mathrm{Tr}\left(\sum_{i=1}^{N}\sum_{j=1}^{2n}\sum_{k=1}^{N}\sum_{l=1}^{2n}\kappa(\hat{s}_i,\hat{s}_k)(\lambda_{ij}\hat{u}_i-2\Gamma_{ij})(\lambda_{kl}\hat{u}_k-2\Gamma_{kl})^T\right) \\
& -\sum_{i=1}^{N}\sum_{j=1}^{2n}\lambda_{ij}\hat{y}_j^T\hat{u}_i+\sum_{i=1}^{N}\sum_{j=1}^{2n}2\Gamma_{ij}{}^T\hat{y}_j \\
\text{s.t.} \quad & \lambda_{ij}W(\hat{s}_i)-2M(\hat{s}_i)\Gamma_{ij}\geq 0,\quad \frac{1}{N}-\sum_{j=1}^{2n}\lambda_{ij}\geq 0,\quad \lambda_{ij}\in R_+ \qquad \forall(i,j)\in[N]\times[2n] \\
& \begin{bmatrix}\Lambda_{ij} & \Gamma_{ij}\\ * & \lambda_{ij}/4\end{bmatrix}\succeq 0 \qquad \forall(i,j)\in[N]\times[2n],
\end{aligned}
$$

where $\kappa(\hat{s}_i,\hat{s}_j)=\phi(\hat{s}_i)^T\phi(\hat{s}_j)$, *and* $\Lambda_{ij}\in R^{n\times n}$, $\Gamma_{ij}\in R^n$, *and* $\lambda_{ij}\in R$ *are the decision variables. The matrices* θ_{uu} *and* θ_{su} *can be written as*

$$
\theta_{uu}=-\frac{\sum_{i=1}^{N}\sum_{j=1}^{2n}(\lambda_{ij}\hat{u}_i\hat{u}_i^T-\Lambda_{ij})}{2k},\quad \theta_{su}=-\frac{\sum_{i=1}^{N}\sum_{j=1}^{2n}\phi(\hat{s}_i)(\lambda_{ij}\hat{u}_i^T-2\Gamma_{ij}{}^T)}{k}.
$$

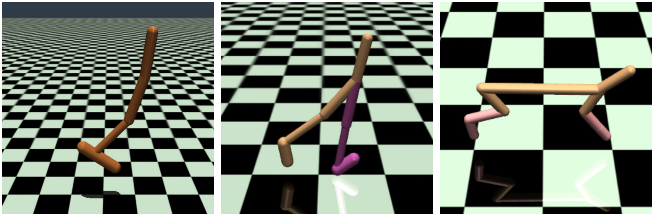
Numerical Experiments

Experimental environment [6]

[6] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis (Mar. 2023). *Gymnasium*. DOI: 10.5281/zenodo.8127026

Numerical Experiments

Experimental environment [6]



(a) Hopper (b)Walker2d (c)HalfCheetah

Task	Action Dim	State Dim
Hopper	3	11
Walker2d	6	17
HalfCheetah	6	17

TU Delft

[6] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis (Mar. 2023). *Gymnasium*. DOI: 10.5281/zenodo.8127026

Numerical Experiments

Dataset [7]

D4RL aims to provide standardized and diverse datasets that researchers can use to benchmark and evaluate their algorithms. We tested the algorithm on each task using both **Expert** and **Medium** datasets, resulting in a total of six datasets.

- **Expert:** Expert has 1M samples from a policy trained to completion with Soft Actor-Critic.
- **Medium:** Medium has 1M samples derived from a policy that is trained to achieve approximately 1/3 the performance of the expert.

[7] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine (2020). *D4RL: Datasets for Deep Data-Driven Reinforcement Learning*. arXiv: 2004.07219 [cs.LG]

Numerical Experiments

Solver[8]

SCS
SPLITTING CONIC SOLVER

- Open-source
- Designed to tackle large-scale convex cone problems (linear/Quadratic/SOCP)
- Known for its speed and scalability
- Parallel processing

[8] B. O'Donoghue (Aug. 2021). "Operator Splitting for a Homogeneous Embedding of the Linear Complementarity Problem". In: *SIAM Journal on Optimization* 31 (3), pp. 1999–2023

Numerical Experiments

Results

TUDelft

[9] S. Fujimoto and S. S. Gu (2021). “A minimalist approach to offline reinforcement learning”. In: *Advances in neural information processing systems* 34, pp. 20132–20145

[10] A. Kumar, A. Zhou, G. Tucker, and S. Levine (2020). “Conservative q-learning for offline reinforcement learning”. In: *Advances in Neural Information Processing Systems* 33, pp. 1179–1191

Numerical Experiments

Results

Task	KIOM	BC(TD3+BC)[9]	BC(CQL)[10]	Teacher agent
Hopper-expert	**109.9**(5k)	**111.5**	**109.0**	**108.5**
Hopper-medium	**50.2**(5k)	30.0	29.0	44.3
Walker2d-expert	108.5(10k)	56.0	**125.7**	107.1
Walker2d-medium	**74.6**(5k)	11.4	6.6	62.1
Halfcheetah-expert	84.4(10k)	105.2	**107.0**	88.1
Halfcheetah-medium	**39.0**(5k)	36.6	36.1	**40.7**

[9] S. Fujimoto and S. S. Gu (2021). "A minimalist approach to offline reinforcement learning". In: *Advances in neural information processing systems* 34, pp. 20132–20145

[10] A. Kumar, A. Zhou, G. Tucker, and S. Levine (2020). "Conservative q-learning for offline reinforcement learning". In: *Advances in Neural Information Processing Systems* 33, pp. 1179–1191

Sequential Selection Optimization (SSO)

Background of SSO[11]

Background of SSO[11]

- Problem description

Data size	Memory (RAM)
5K	16GB
10K	64GB
15K	128GB
20K	256GB

Table: The relationship between the dataset size and the required memory on the Halfcheetah-expert task.

TU Delft

Background of SSO[11]

- Problem description

Data size	Memory (RAM)
5K	16GB
10K	64GB
15K	128GB
20K	256GB

Table: The relationship between the dataset size and the required memory on the Halfcheetah-expert task.

- Idea
  SSO breaks the large SDP problem (11) into a series of smaller SDP problems and use the optimal solution of these small problem to recover the solution of original problem (11). This feature empowers SSO to efficiently address larger-scale optimization problems.

[11]The algorithm is inspired by the course "Networked and Distributed Control Systems", DCSC, TU Delft and algorithm "Sequential Minimal Optimization", John Platt.

Block Coordinate Descent

Block Coordinate Descent

We focus on the problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in X, \end{array}$$

where $f : R^n \mapsto R$ is a differentiable convex function and X is a Cartesian product of closed convex sets

$$X = X_1 \times X_2 \times \cdots \times X_m,$$

where X_i is a subset of R^{n_i}.

Block Coordinate Descent

We focus on the problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in X, \end{array}$$

where $f : R^n \mapsto R$ is a differentiable convex function and X is a Cartesian product of closed convex sets

$$X = X_1 \times X_2 \times \cdots \times X_m,$$

where X_i is a subset of R^{n_i}.The vector x is partitioned as

$$x = (x^1, x^2, \cdots, x^m),$$

where each x^i is a "block component" of x that is constrained to be in X_i.

Block Coordinate Descent

We focus on the problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in X, \end{array}$$

where $f : R^n \mapsto R$ is a differentiable convex function and X is a Cartesian product of closed convex sets

$$X = X_1 \times X_2 \times \cdots \times X_m,$$

where X_i is a subset of R^{n_i}.The vector x is partitioned as

$$x = (x^1, x^2, \cdots, x^m),$$

where each x^i is a "block component" of x that is constrained to be in X_i.The Block Coordinate Descent (BCD) method is defined as follow: given the current iterate $x_k = (x_k^1, x_k^2, \cdots, x_k^m)$, we generate the next iterate $x_{k+1} = (x_{k+1}^1, x_{k+1}^2, \cdots, x_{k+1}^m)$, according to

$$x_{k+1}^i \in \arg\min_{\xi \in X_i} f\left(x_{k+1}^1, \ldots, x_{k+1}^{i-1}, \xi, x_k^{i+1}, \ldots, x_k^m\right), \quad i = 1, \ldots, m; \tag{13}$$

where we assume that the preceding minimization has at least one optimal solution.

Block Coordinate Descent

We focus on the problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in X, \end{array}$$

where $f : R^n \mapsto R$ is a differentiable convex function and X is a Cartesian product of closed convex sets

$$X = X_1 \times X_2 \times \cdots \times X_m,$$

where X_i is a subset of R^{n_i}.The vector x is partitioned as

$$x = (x^1, x^2, \cdots, x^m),$$

where each x^i is a "block component" of x that is constrained to be in X_i.The Block Coordinate Descent (BCD) method is defined as follow: given the current iterate $x_k = (x_k^1, x_k^2, \cdots, x_k^m)$, we generate the next iterate $x_{k+1} = (x_{k+1}^1, x_{k+1}^2, \cdots, x_{k+1}^m)$, according to

$$x_{k+1}^i \in \arg\min_{\xi \in X_i} f\left(x_{k+1}^1, \ldots, x_{k+1}^{i-1}, \xi, x_k^{i+1}, \ldots, x_k^m\right), \quad i = 1, \ldots, m; \tag{13}$$

where we assume that the preceding minimization has at least one optimal solution.

Proximal BCD: $x_{k+1}^i \in \arg\min_{\xi \in X_i} \left\{ f\left(x_{k+1}^1, \ldots, x_{k+1}^{i-1}, \xi, x_k^{i+1}, \ldots, x_k^m\right) + \frac{1}{2c} \left\| \xi - x_k^i \right\|^2 \right\}.$

Block Coordiante Descent for KIOM

Block Coordiante Descent for KIOM

The problem focused on

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in X, \quad \textit{where} \quad X = X_1 \times X_2 \times \cdots \times X_m. \end{aligned}$$

Block Coordiante Descent for KIOM

The problem focused on

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in X, \quad \textit{where} \quad X = X_1 \times X_2 \times \cdots \times X_m. \end{aligned}$$

Simple version of KIOM

$$\begin{aligned} \min_{\Lambda_i, \Gamma_i} \quad & \begin{bmatrix} m_1 & m_2 & \dots & m_N \end{bmatrix} \frac{K \otimes I}{k} \begin{bmatrix} m_1 & m_2 & \dots & m_N \end{bmatrix}^T + \sum_{i=1}^{N} \mathrm{Tr}(\Lambda_i) \\ \text{s.t.} \quad & \frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i \geq 0, \; m_i = \frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T, \quad \forall i \leq N \\ & \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \succeq 0, \quad \forall i \leq N, \end{aligned}$$

Block Coordiante Descent for KIOM

The problem focused on

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in X, \quad \textit{where} \quad X = X_1 \times X_2 \times \cdots \times X_m. \end{array}$$

Simple version of KIOM

$$\begin{array}{ll} \min\limits_{\Lambda_i, \Gamma_i} & \begin{bmatrix} m_1 & m_2 & \dots & m_N \end{bmatrix} \frac{K \otimes I}{k} \begin{bmatrix} m_1 & m_2 & \dots & m_N \end{bmatrix}^T + \sum_{i=1}^N \mathrm{Tr}(\Lambda_i) \\ \text{s.t.} & \frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i \geq 0, \; m_i = \frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T, \quad \forall i \leq N \\ & \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \succeq 0, \quad \forall i \leq N, \end{array}$$

Definition of coordinate

$$\textit{coordinate\_i} := \{\Lambda_i, \Gamma_i\}.$$

TUDelft

Heuristics for choosing which coordinates to optimize

Heuristics for choosing which coordinates to optimize

Given the current values of $\{\Lambda_i, \Gamma_i\}_{i=1}^N$ (not necessarily optimal), we choose p coordinates that satisfy the condition[12]

$$\frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i > 0 \tag{14}$$

and have the largest KKT violators which is defined as

$$KKT\_violator(i) = \left| \mathrm{Tr}\left(\begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \begin{bmatrix} I_n & 2\theta_{su}^\top \phi(\hat{s}_i) \\ * & \|2\theta_{su}^\top \phi(\hat{s}_i)\|_2^2 \end{bmatrix} \right) \right|. \tag{15}$$

[12] The experiment found that over 80% of the coordinates in these tasks satisfy condition (14).

Heuristics for choosing which coordinates to optimize

Given the current values of $\{\Lambda_i, \Gamma_i\}_{i=1}^{N}$ (not necessarily optimal), we choose p coordinates that satisfy the condition[12]

$$\frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i > 0 \tag{14}$$

and have the largest KKT violators which is defined as

$$KKT\_violator(i) = \left| \mathrm{Tr}\left(\begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \begin{bmatrix} I_n & 2\theta_{su}^{\top}\phi(\hat{s}_i) \\ * & \|2\theta_{su}^{\top}\phi(\hat{s}_i)\|_2^2 \end{bmatrix} \right) \right|. \tag{15}$$

The reason is because when the decision variables $\{\Lambda_i, \Gamma_i\}_{i=1,\ldots,N}$ reach their optimum, based on KKT conditions, all the coordinates that satisfy (14) have KKT violators (15) of 0.

TUDelft

[12]The experiment found that over 80% of the coordinates in these tasks satisfy condition (14).

KKT Conditions

Stationarity:

$$\theta_{su} = -\frac{1}{k}\sum_{i=1}^{N}\phi(\hat{s}_i)\left(\frac{\hat{u}_i^T}{N} - 2\Gamma_i^T\right) \tag{16a}$$

$$\tilde{\lambda}_i = \frac{W}{N} - 2M\Gamma_i, \ \forall i \leq N \tag{16b}$$

Complementary slackness:

$$\tilde{\lambda}_i{}^T\lambda_i = 0, \ \forall i \leq N \tag{16c}$$

$$\mathrm{Tr}\left(\begin{bmatrix}\Lambda_i & \Gamma_i \\ * & \frac{1}{4N}\end{bmatrix}\begin{bmatrix}I_n & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i\end{bmatrix}\right) = 0, \ \forall i \leq N \tag{16d}$$

Primal feasibility:

$$\lambda_i \in R_+^d, \ \forall i \leq N \tag{16e}$$

Proof

First, based on current values of $\{\Lambda_i, \Gamma_i\}_{i=1}^N$, we choose coordinate i such that

$$\frac{W}{N} - 2M\Gamma_i > 0. \tag{17}$$

Based on KKT condition (16b), we have

$$\tilde{\lambda}_i > 0. \tag{18}$$

Then, based on conditions (16c) and (16e), one can obtain

$$\lambda_i = 0. \tag{19}$$

Substituting the result (19) into condition (16d) yields

$$\mathrm{Tr}\left(\begin{bmatrix}\Lambda_i & \Gamma_i \\ * & \frac{1}{4N}\end{bmatrix}\begin{bmatrix}I_n & 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i\end{bmatrix}\right) = 0, \tag{20}$$

where γ_i is the decision variable of the primal problem (21).

Note: For ease of notation, we omit writing the dependency of the matrices M and W on $\hat{s}_i$.

Proof

Then we derive the expression of γ_i based on the primal problem (21)

$$
\begin{aligned}
\min_{\theta_{su},\gamma_i,\lambda_i} \quad & k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\
\text{s.t.} \quad & \lambda_i \in R_+^d,\ \gamma_i \in R, \quad \forall i \le N \\
& \begin{bmatrix} I_n & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \le N.
\end{aligned} \tag{21}
$$

Proof

Then we derive the expression of γ_i based on the primal problem (21)

$$\begin{array}{ll}\min\limits_{\theta_{su},\gamma_i,\lambda_i} & k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i + W\left(\hat{s}_i\right)^\top\lambda_i\right) \\ \text{s.t.} & \lambda_i \in R_+^d,\ \gamma_i \in R, \quad \forall i \le N \\ & \begin{bmatrix} I_n & M\left(\hat{s}_i\right)^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \le N. \end{array} \tag{21}$$

By utilizing the Schur complement, we can prove

$$\begin{bmatrix} I_n & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0 \Leftrightarrow \gamma_i \ge \|M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i)\|_2^2.$$

Proof

Then we derive the expression of γ_i based on the primal problem (21)

$$\begin{array}{ll} \min\limits_{\theta_{su},\gamma_i,\lambda_i} & k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\ \text{s.t.} & \lambda_i \in R_+^d,\ \gamma_i \in R, \quad \forall i \leq N \\ & \begin{bmatrix} I_n & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \leq N. \end{array} \tag{21}$$

By utilizing the Schur complement, we can prove

$$\begin{bmatrix} I_n & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0 \Leftrightarrow \gamma_i \geq \|M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i)\|_2^2.$$

Therefore, problem (21) can be equivalently expressed as

$$\begin{array}{ll} \min\limits_{\theta_{su},\gamma_i,\lambda_i} & k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\ \text{s.t.} & \lambda_i \in R_+^d,\ \gamma_i \in R,\ \gamma_i \geq \|M^\top\lambda_i + 2\theta_{su}^\top\hat{s}_i\|_2^2, \quad \forall i \leq N, \end{array} \tag{22}$$

Proof

Then we derive the expression of γ_i based on the primal problem (21)

$$\begin{aligned} \min_{\theta_{su},\gamma_i,\lambda_i} \quad & k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\ \text{s.t.} \quad & \lambda_i \in R_+^d,\ \gamma_i \in R, \quad \forall i \le N \\ & \begin{bmatrix} I_n & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \le N. \end{aligned} \tag{21}$$

By utilizing the Schur complement, we can prove

$$\begin{bmatrix} I_n & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0 \Leftrightarrow \gamma_i \ge \|M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i)\|_2^2.$$

Therefore, problem (21) can be equivalently expressed as

$$\begin{aligned} \min_{\theta_{su},\gamma_i,\lambda_i} \quad & k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\ \text{s.t.} \quad & \lambda_i \in R_+^d,\ \gamma_i \in R,\ \gamma_i \ge \|M^\top\lambda_i + 2\theta_{su}^\top\hat{s}_i\|_2^2, \quad \forall i \le N, \end{aligned} \tag{22}$$

When γ_i attain its optimal values, the equality in the last constraint should hold: $\gamma_i = \|M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i)\|_F^2$. Note that $\lambda_i = 0$ (19), then the expression of γ_i is

$$\gamma_i = \|2\theta_{su}^\top\phi(\hat{s}_i)\|_2^2. \tag{23}$$

Proof

Then we derive the expression of γ_i based on the primal problem (21)

$$\begin{aligned} \min_{\theta_{su},\gamma_i,\lambda_i} \quad & k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\ \text{s.t.} \quad & \lambda_i \in R_+^d,\ \ \gamma_i \in R, \quad \forall i \leq N \\ & \begin{bmatrix} I_n & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \leq N. \end{aligned} \tag{21}$$

By utilizing the Schur complement, we can prove

$$\begin{bmatrix} I_n & M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0 \Leftrightarrow \gamma_i \geq \|M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i)\|_2^2.$$

Therefore, problem (21) can be equivalently expressed as

$$\begin{aligned} \min_{\theta_{su},\gamma_i,\lambda_i} \quad & k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(2\hat{u}_i^T\theta_{su}^T\phi(\hat{s}_i) + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\ \text{s.t.} \quad & \lambda_i \in R_+^d,\ \ \gamma_i \in R,\ \ \gamma_i \geq \|M^\top\lambda_i + 2\theta_{su}^\top\hat{s}_i\|_2^2, \quad \forall i \leq N, \end{aligned} \tag{22}$$

When γ_i attain its optimal values, the equality in the last constraint should hold: $\gamma_i = \|M^\top\lambda_i + 2\theta_{su}^\top\phi(\hat{s}_i)\|_F^2$. Note that $\lambda_i = 0$ (19), then the expression of γ_i is

$$\gamma_i = \|2\theta_{su}^\top\phi(\hat{s}_i)\|_2^2. \tag{23}$$

Substituting (23) into (20), we obtain $\mathrm{Tr}\left(\begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix}\begin{bmatrix} I_n & 2\theta_{su}^\top\phi(\hat{s}_i) \\ * & \|2\theta_{su}^\top\phi(\hat{s}_i)\|_2^2 \end{bmatrix}\right) = 0.$ □

Warm-up trick for improved initial guess

Warm-up trick for improved initial guess

First, we divide the N examples from the dataset into n datasets of p examples $S_1, \ldots, S_n$.

Warm-up trick for improved initial guess

First, we divide the N examples from the dataset into n datasets of p examples $S_1, \ldots, S_n$. Directly solving the KIOM problem (11) involving N data points is unfeasible. However, we can rapidly solve a modified KIOM subproblem (29) restricted to only p data points:

$$
\begin{aligned}
\mathit{subproblem}(l) := \min_{\Lambda_i, \Gamma_i} \quad & \frac{1}{k} \sum_{i \in S_l} \sum_{j \in S_l} \kappa_{ij} \left(\frac{\hat{u}_i^T}{N} - 2\Gamma_i^T \right) \left(\frac{\hat{u}_j}{N} - 2\Gamma_j \right) + \sum_{i \in S_l} \mathrm{Tr}(\Lambda_i) \\
\text{s.t.} \quad & \frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i \geq 0, \quad \forall i \in S_l \\
& \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \succeq 0, \quad \forall i \in S_l.
\end{aligned}
$$

Warm-up trick for improved initial guess

First, we divide the N examples from the dataset into n datasets of p examples $S_1, \ldots, S_n$. Directly solving the KIOM problem (11) involving N data points is unfeasible. However, we can rapidly solve a modified KIOM subproblem (29) restricted to only p data points:

$$
\begin{aligned}
subproblem(l) := \min_{\Lambda_i, \Gamma_i} \quad & \frac{1}{k} \sum_{i \in S_l} \sum_{j \in S_l} \kappa_{ij} \left(\frac{\hat{u}_i^T}{N} - 2\Gamma_i^T \right) \left(\frac{\hat{u}_j}{N} - 2\Gamma_j \right) + \sum_{i \in S_l} \mathrm{Tr}(\Lambda_i) \\
\text{s.t.} \quad & \frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i \geq 0, \quad \forall i \in S_l \\
& \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \succeq 0, \quad \forall i \in S_l.
\end{aligned}
$$

By solving n instances of such problems ($l = 1, \ldots, n$), we systematically traverse all N data points. Ultimately, we concatenate the optimal solutions of these n subproblems to form an initial estimate for the KIOM problem (11).

SSO Algorithm

SSO Algorithm

Algorithm 1 SSO

1: Initialize variable: $\{\Lambda_i, \Gamma_i\}_{i=1,...,N} \leftarrow \text{WarmUp}(\{\hat{s}_i, \hat{a}_i\}_{i=1,...N})$

2: **for** iteration = 1 to T **do**

3: $\quad \{\Lambda_{a_i}, \Gamma_{a_i}\}_{i=1,...,p} \leftarrow \text{HeuristicSelection}(\{\hat{s}_i, \hat{a}_i\}_{i=1,...N}, \{\Lambda_i, \Gamma_i\}_{i=1,...,N})$

4: $\quad \text{Update}(\{\Lambda_{a_i}, \Gamma_{a_i}\}_{i=1,...,p})$ {Update selected coordinates}

5: **end for**

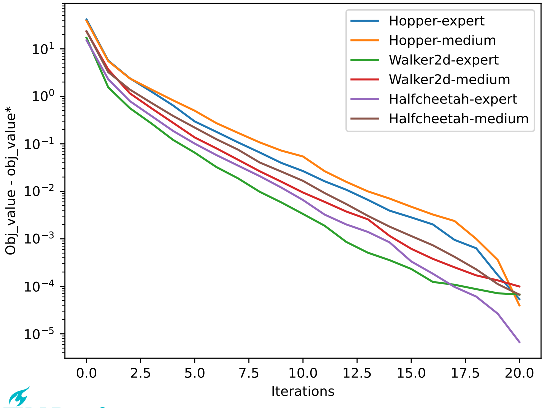
Numerical Experiments

Performance evaluation[13]

[13] All experimental parameters are held constant and in the SSO algorithm, $\frac{N}{2}$ coordinates are updated at each iteration, meaning that each subproblem encompasses half of the dataset.

Numerical Experiments

Performance evaluation[13]



Task	SCS		SSO	
	Obj Value	**Score**	**Obj Value**	**Score**
Hopper-expert	185.219	109.9	185.220	110.2
Hopper-medium	218.761	50.2	218.761	51.8
Walker2d-expert	140.121	108.5	140.121	109.2
Walker2d-medium	151.117	74.6	151.117	74.9
Halfcheetah-expert	165.041	84.4	165.041	83.8
Halfcheetah-medium	188.184	39.0	188.184	39.7

[13] All experimental parameters are held constant and in the SSO algorithm, $\frac{N}{2}$ coordinates are updated at each iteration, meaning that each subproblem encompasses half of the dataset.

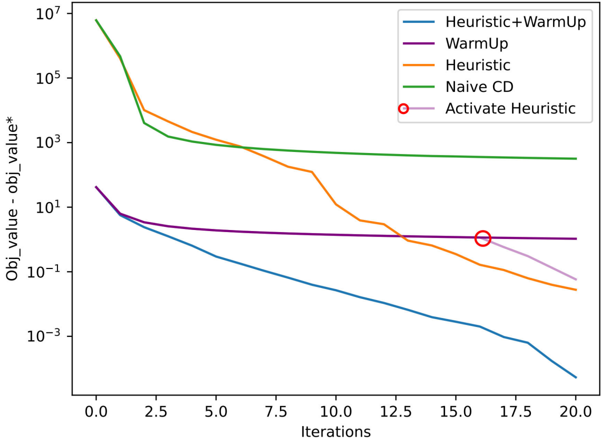
Numerical Experiments

Ablation studies[14]

[14] Tested on hopper-expert task. All experimental parameters are held constant and in the SSO algorithm, $N/2$ coordinates are updated at each iteration.

Numerical Experiments

Ablation studies[14]



[14] Tested on hopper-expert task. All experimental parameters are held constant and in the SSO algorithm, $N/2$ coordinates are updated at each iteration.

Conclusion and Future Study

Conclusion

- Introduction of Inverse Optimization
  - Idea of IO
  - Learning for control: A IO model
- Kernel Inverse Optimization Machine
  - Theoretical derivation
  - Two varients
  - Numerical experiments
- Sequential Selection Optimization
  - Theoretical derivation
  - Numerical experiments

Future study

In this thesis, we demonstrate the convergence guarantee when employing the proximal block coordinate descent algorithm to optimize the simplified version of the KIOM model. However, we do not analyze the convergence of the SSO algorithm, as this requires additional consideration of KKT conditions, which would make the analysis relatively challenging. Therefore, we list this as a direction for future research.

Thank you for your attention

Long, Youyuan

TU Delft