

Descrizione dettagliata dell'algoritmo minover

Autore

February 22, 2025

1 Introduzione

L'algoritmo **minover** è un metodo iterativo per trovare un punto all'interno di un politopo definito da un insieme di vincoli lineari. L'algoritmo cerca il vincolo più violato e sposta la soluzione corrente in direzione per soddisfarlo, minimizzando lo scostamento rispetto ai limiti imposti.

Questo documento fornisce una descrizione dettagliata dell'algoritmo, analizzando ogni variabile e il suo ruolo nel processo.

2 Descrizione dell'algoritmo

L'algoritmo **minover** segue questi passaggi:

1. Identificare il vincolo più violato, ovvero quello per cui la soluzione corrente è più lontana dai limiti imposti.
2. Calcolare un fattore di correzione per riportare la soluzione entro i limiti.
3. Applicare l'aggiornamento alla soluzione corrente.
4. Ripetere il processo finché tutti i vincoli sono soddisfatti.

3 Definizione delle variabili

Le seguenti variabili sono utilizzate nell'implementazione:

- **ok** ($\in \{0, 1\}$): Flag che indica se tutti i vincoli sono soddisfatti ($ok = 1$) o meno ($ok = 0$).
- **alpha** (α): Fattore di aggiornamento utilizzato per correggere la soluzione.
- **xmin**: Il valore minimo tra le violazioni dei vincoli.
- **min**: Indice del vincolo più violato.
- **sign** (± 1): Indica se la correzione avviene verso l'alto o verso il basso.
- **flux**: Vettore delle variabili della soluzione corrente.

- **matrice**: Lista che memorizza gli indici delle variabili associate a ciascun vincolo.
- **matrice2**: Lista dei coefficienti dei vincoli.
- **boundmin**, **boundmax**: Limiti inferiore e superiore per ciascun vincolo.
- **norm**: Norma della direzione di aggiornamento.

4 Passaggi dettagliati

L'algoritmo esegue un ciclo finché non trova una soluzione valida:

1. Inizialmente, imposta $xmin$ ad un valore molto grande per identificare il vincolo più violato.
2. Per ogni vincolo i , calcola il valore di x dato dalla combinazione lineare:

$$x = \sum_j \text{flux}[\text{matrice}[i][j]] \cdot \text{matrice2}[i][j] \quad (1)$$

3. Calcola le distanze dai limiti inferiori e superiori:

$$x_1 = x - \text{boundmin}[i], \quad x_2 = \text{boundmax}[i] - x \quad (2)$$

4. Se x_1 o x_2 sono i minimi rilevati, aggiorna $xmin$, min e $sign$.
5. Se $xmin > 0$, termina l'algoritmo con $ok = 1$.
6. Altrimenti, calcola la norma della direzione di aggiornamento:

$$\text{norm} = \sum_j \text{matrice2}[\text{min}][j]^2 \quad (3)$$

7. Aggiorna la soluzione corrente con:

$$\alpha = -1.8 \frac{xmin}{\text{norm}}, \quad \text{flux}[\text{matrice}[\text{min}][j]] + = \text{sign} \cdot \alpha \cdot \text{matrice2}[\text{min}][j] \quad (4)$$

8. Continua finché tutti i vincoli sono soddisfatti.

5 Conclusioni

L'algoritmo **minover** è un metodo efficace per trovare un punto interno a un politopo rispettando i vincoli. L'approccio iterativo consente di correggere iterativamente le violazioni, rendendolo un metodo robusto per problemi di ottimizzazione con vincoli lineari.