

DC アルゴリズムを用いた
トラス構造のロバスト最適設計法

寒野 善博

東京大学

日本オペレーションズ・リサーチ学会 2018 年春季研究発表会

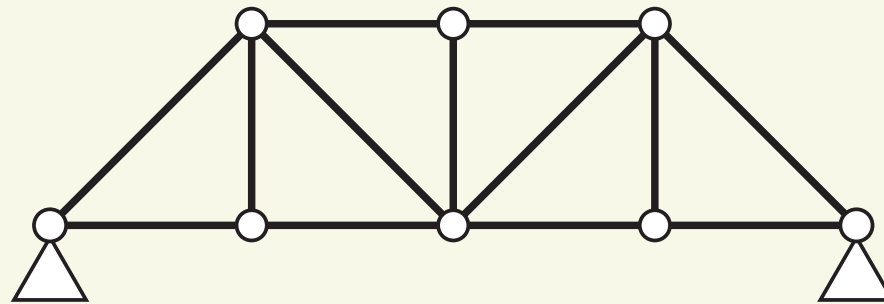
(March 15–16, 2018)

構造物のロバスト最適化

- 外力の不確実性を考慮
 - 連続体 [Cherkaev & Cherkaev '03, '08], [Guo, Bai, Zhang, Gao '09]
[de Gournay, Allaire, & Jouve '08], [Guo, Du, & Gao '11]
[Takezawa, Nii, Kitamura, & Kogiso '11], [Holmberg, Thore, & Klarbring '15]
 - トラス [Ben-Tal & Nemirovski '97], [Yonekura & K. '10], [K. & Guo '10]
- 方法論：
 - 外力がばらつく範囲（集合）を指定して,
 - コンプライアンスの最大値を最小化する.
 - コンプライアンス（= 外力仕事）は 柔性の指標

トラス (truss) とは

- 構造物の形式の一つ
- 直線状の部材が ピン節点で接合されている.

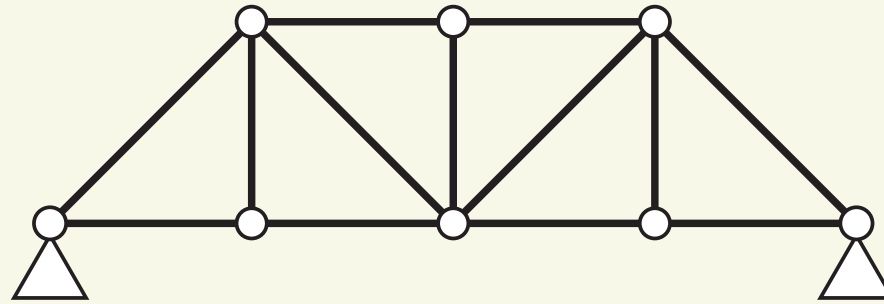


(Wikipedia より)

Y. Kanno (ORSJ '18-S)

トラス (truss) とは

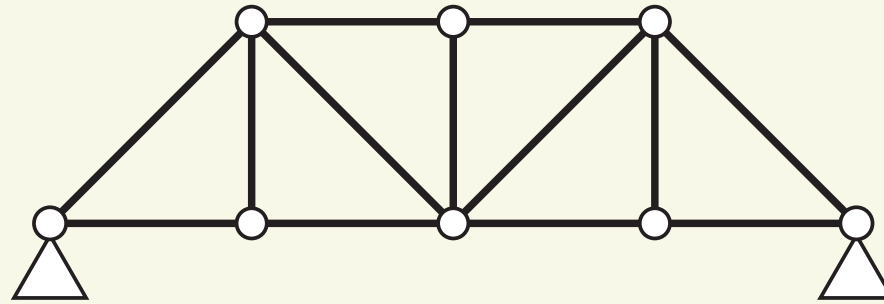
- 構造物の形式の一つ
- 直線状の部材が ピン節点で接合されている.



- グラフとみると ...
 - 部材 \simeq 枝
 - 節点 \simeq 頂点 (ただし 座標の情報をもつ)

トラス (truss) とは

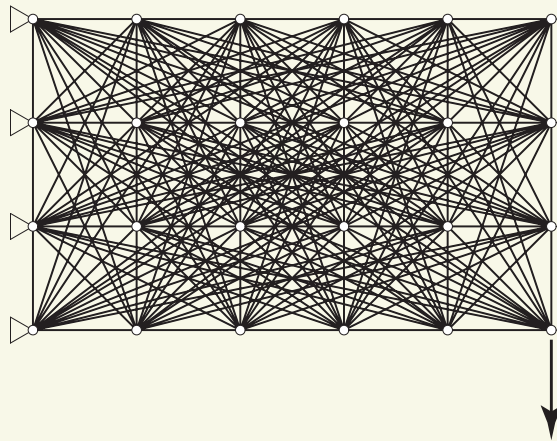
- 構造物の形式の一つ
- 直線状の部材がピン節点で接合されている.



- グラフとみると ...
 - 部材 \simeq 枝
 - 節点 \simeq 頂点 (ただし 座標の情報をもつ)
- 部材の断面積 (\simeq 枝の太さ) を最適化したい.
 - 太い枝には “たくさん力が流れる”.

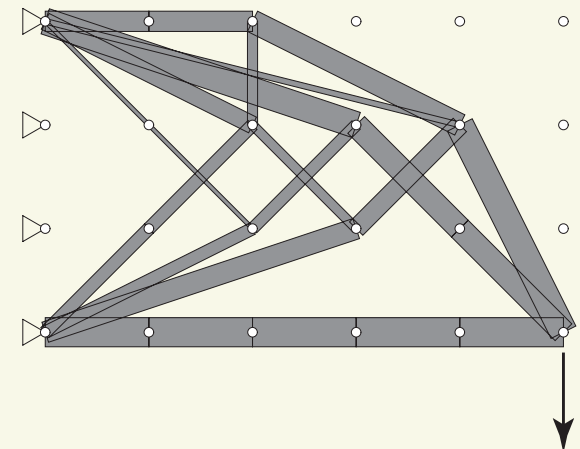
トラスの剛性最大化設計

- コンプライアンス (= 柔性の指標) の最小化
 - 設計変数は それぞれの部材の断面積
 - 制約は 部材の体積 (= 断面積 × 長さ) の総和の上限値
 - 断面積がゼロになった部材は 除去
 - ← 「トポロジー最適化」という.



初期設定

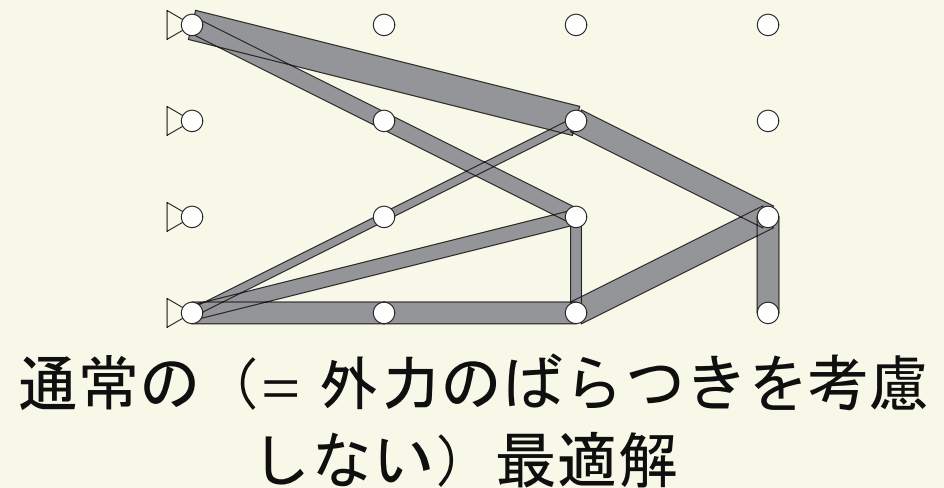
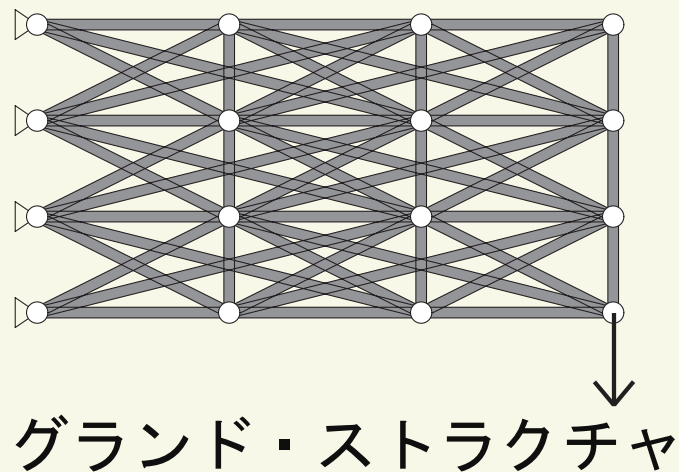
(グラウンド・ストラクチャという)



最適解

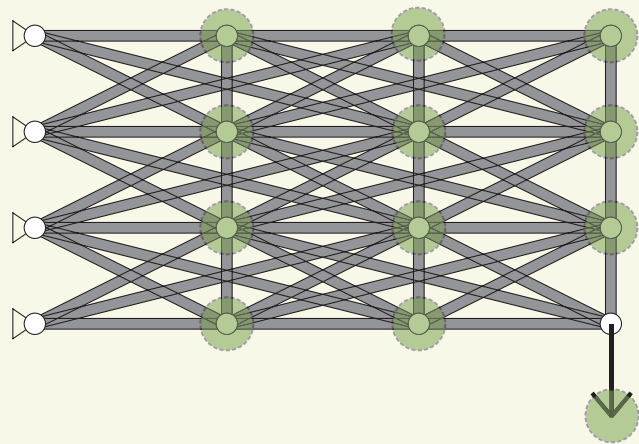
トラスのロバスト最適化（既往の手法）

- コンプライアンスの最悪値の最小化 (SDP) [Ben-Tal & Nemirovski '97]
 - 外力がばらつく範囲を（高次元の）楕円体として指定

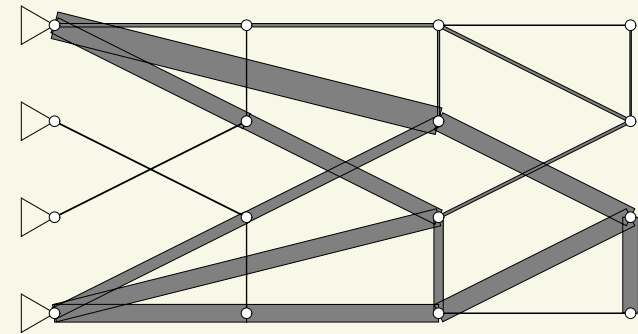


トラスのロバスト最適化（既往の手法）

- コンプライアンスの最悪値の最小化 (SDP) [Ben-Tal & Nemirovski '97]
 - 外力がばらつく範囲を（高次元の）楕円体として指定



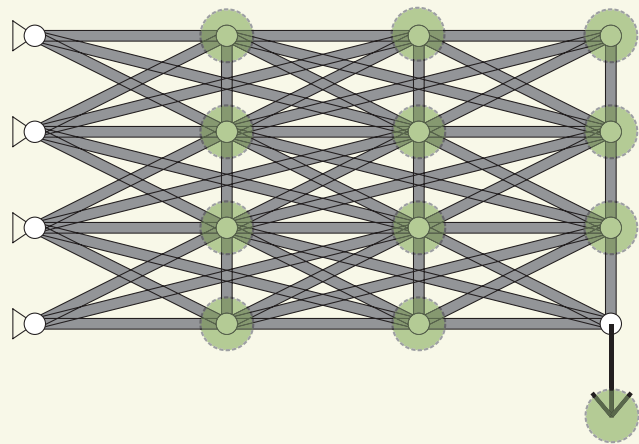
グランド・ストラクチャ



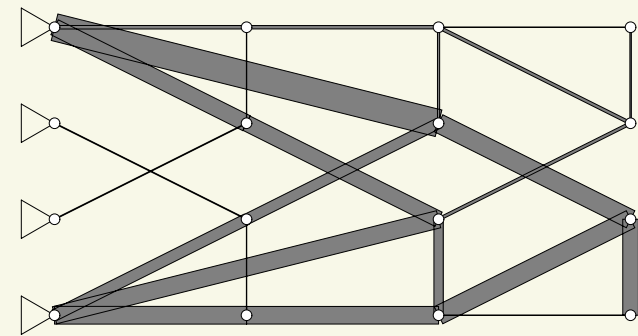
ロバスト最適化の解

トラスのロバスト最適化（既往の手法）

- コンプライアンスの最悪値の最小化 (SDP) [Ben-Tal & Nemirovski '97]
 - 外力がばらつく範囲を（高次元の）楕円体として指定



グランド・ストラクチャ

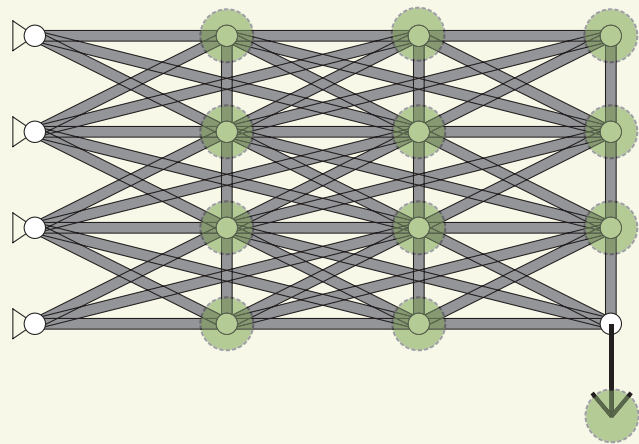


ロバスト最適化の解

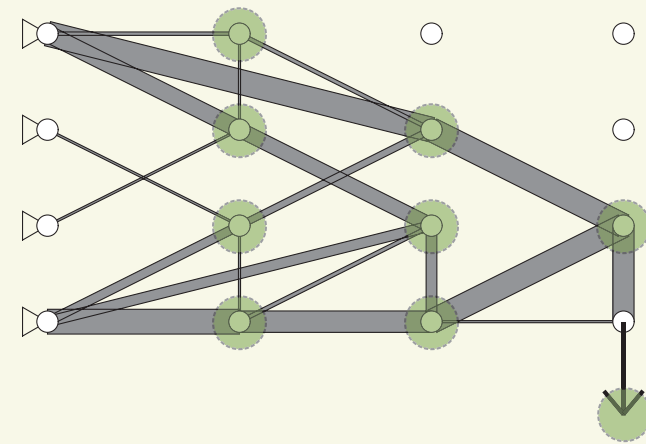
- 「楕円体」を固定するとすべての節点が残ってしまう。
 - 節点の消失の可能性を考慮できるモデルが必要

トラスのロバスト最適化（既往の手法）

- コンプライアンスの最悪値の最小化 (SDP) [Ben-Tal & Nemirovski '97]
 - 外力がばらつく範囲を（高次元の）楕円体として指定



グランド・ストラクチャ



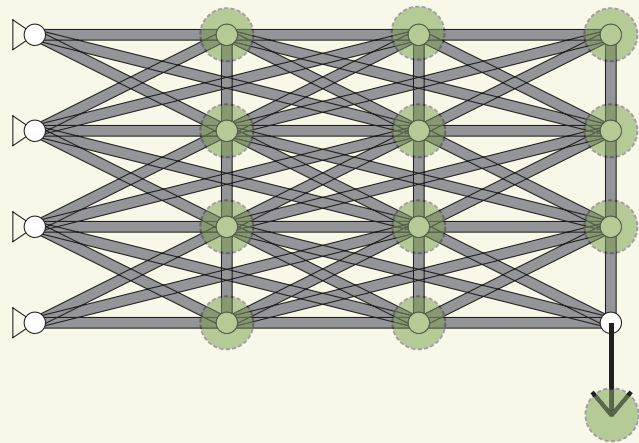
ロバスト最適化の解

- 「楕円体」を固定するとすべての節点が残ってしまう。
 - 節点の消失の可能性を考慮できるモデルが必要
- → 存在する節点のみに不確かな外力が作用するモデル

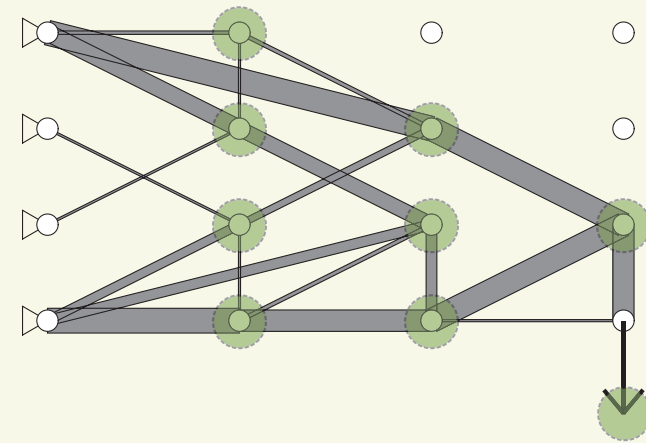
[Yonekura & K. '10], [K. & Guo '10]

トラスのロバスト最適化（既往の手法）

- コンプライアンスの最悪値の最小化 (SDP) [Ben-Tal & Nemirovski '97]
 - 外力がばらつく範囲を（高次元の）楕円体として指定



グランド・ストラクチャ



ロバスト最適化の解???

- 「楕円体」を固定するとすべての節点が残ってしまう。
 - 節点の消失の可能性を考慮できるモデルが必要
- → 存在する節点のみに不確かな外力が作用するモデル

[Yonekura & K. '10], [K. & Guo '10]

外力の不確実性モデル

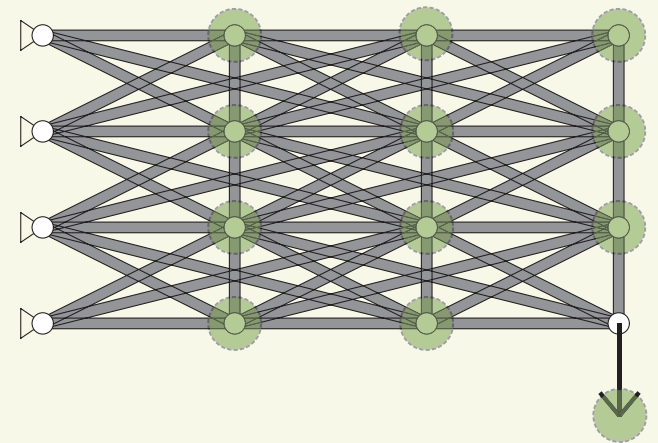
- 楕円体モデル :

[Ben-Tal & Nemirovski '97]

$$\bar{\mathcal{F}} = \{Qe \mid 1 \geq \|e\|\}$$

(Q : 定行列)

- → ロバスト最適化は半正定値計画 (SDP) に帰着
- 節点の消失が許容されない.



外力の不確実性モデル

- 楕円体モデル :

[Ben-Tal & Nemirovski '97]

$$\bar{\mathcal{F}} = \{Qe \mid 1 \geq \|e\|\} \quad (Q: \text{定行列})$$

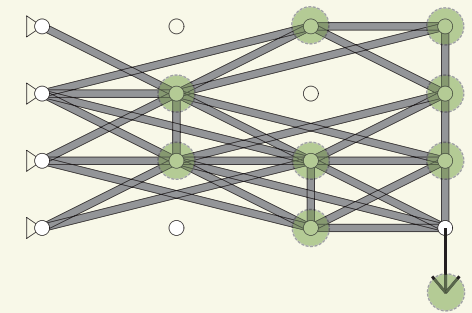
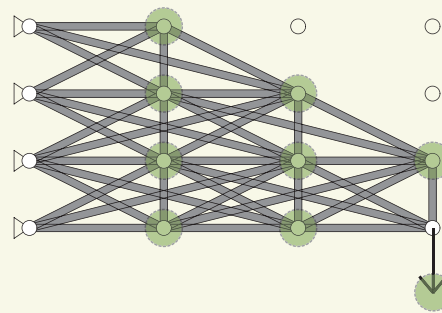
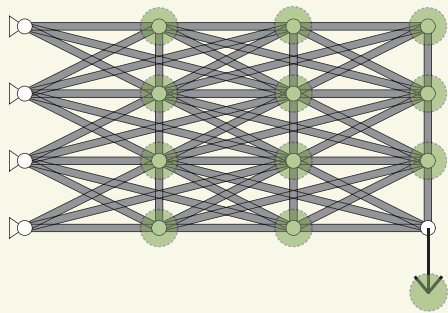
- → ロバスト最適化は半正定値計画 (SDP) に帰着

- 設計依存の楕円体モデル :

[Yonekura & K. '10]

$$\mathcal{F}(s) = \{\text{diag}(s)Qe \mid 1 \geq \|e\|\}$$

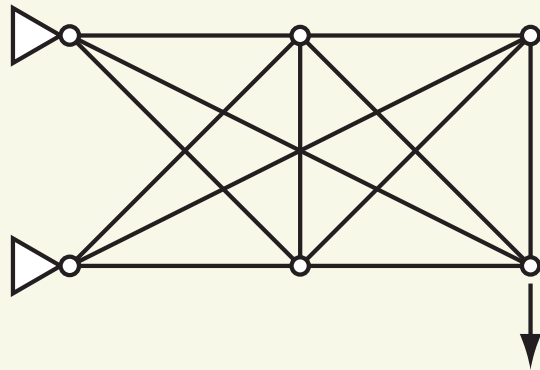
- $s_j = \begin{cases} 1 & \text{節点 } j \text{ が存在するとき} \\ 0 & \text{節点 } j \text{ が消失したとき} \end{cases}$



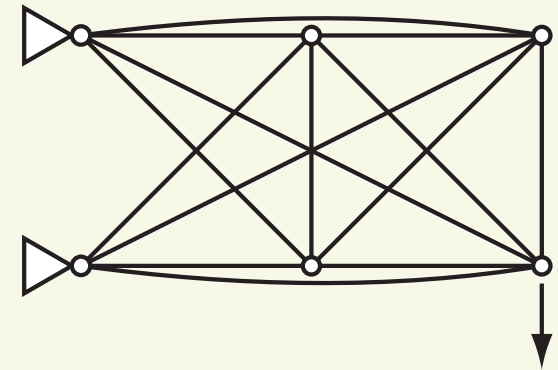
- → 整数制約つき SDP

グラウンド・ストラクチャ（初期設定）再考

- 重複部材について



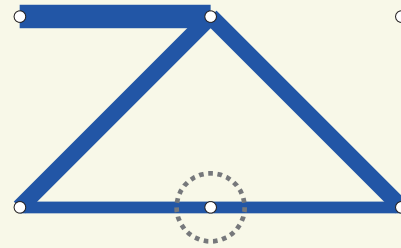
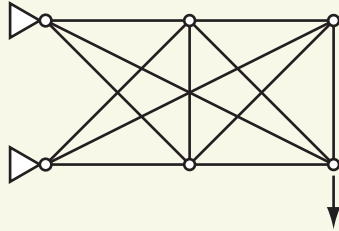
通常は こちらを用いる.



ロバスト最適化では
こちらを用いるべき
ではないか.

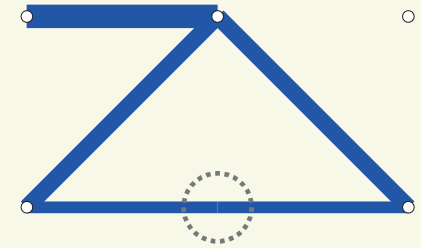
通常最適化 と ロバスト最適化

- 通常最適化



最適解

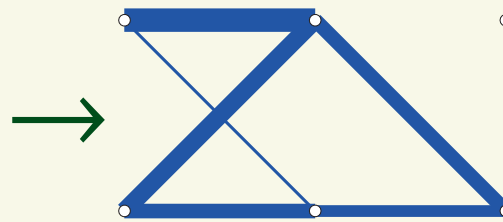
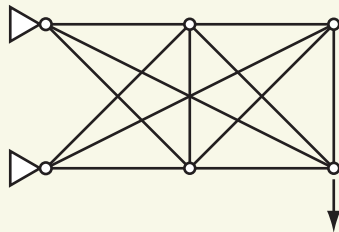
≡



ヒンジは消去可能

- ロバスト最適化

- 重複部材なしの GS (グラウンド・ストラクチャ) の場合

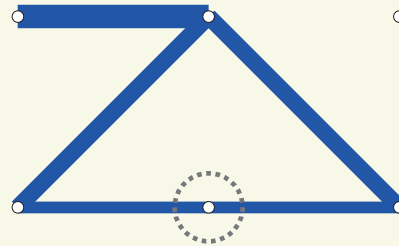
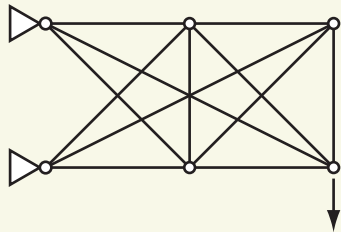


最適解

- ヒンジを安定化させる部材が追加されている.

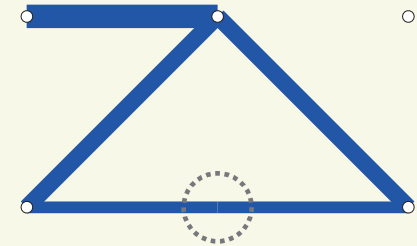
通常最適化とロバスト最適化

● 通常最適化



最適解

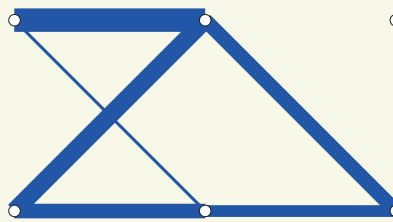
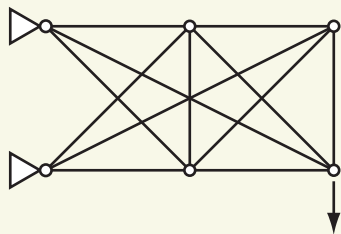
≡



ヒンジは消去可能

● ロバスト最適化

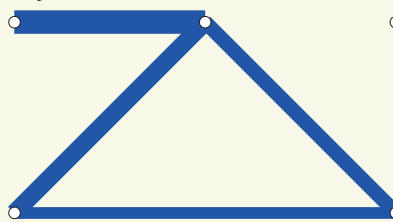
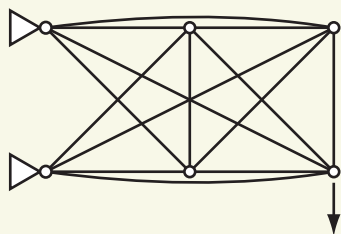
● 重複部材なしの GS (グランド・ストラクチャ) の場合



最適解???

- ヒンジを安定化させる部材が追加されている.

● 重複部材ありの GS の場合

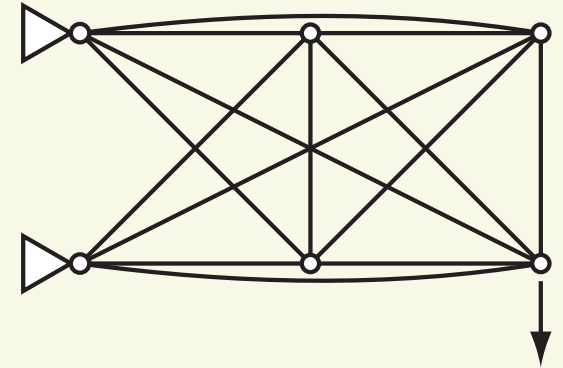


これが最適解

- 「長い単一部材」を選択すれば、より目的関数値が小さくなる.

新たな定式化の必要性

- 初期設定（グラウンド・ストラクチャ）は重複部材が必要であるが、
- 設計解（最適解）は重複部材を含んではならない。



新しい定式化 (1/2)

- 不確かな外力の乗数（大きさ） $s_j \in [0, 1]$:

$$s_j = \begin{cases} 1 & \text{節点 } j \text{ が存在するとき} \\ 0 & \text{節点 } j \text{ が消失したとき} \end{cases}$$

- 本来は, s_j は節点があつ自由度に対して定義する.

新しい定式化 (1/2)

- 不確かな外力の乗数（大きさ） $s_j \in [0, 1]$:

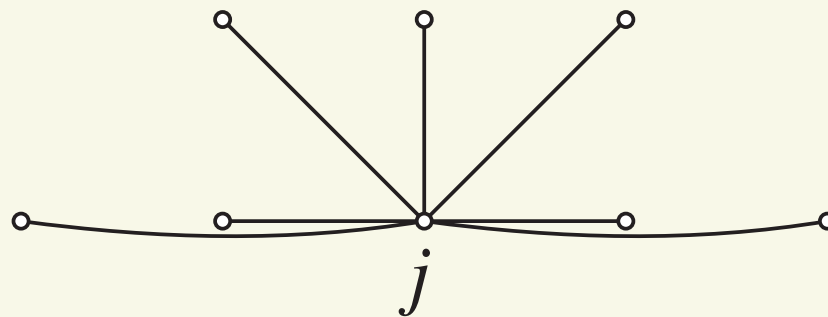
$$s_j = \begin{cases} 1 & \text{節点 } j \text{ が存在するとき} \\ 0 & \text{節点 } j \text{ が消失したとき} \end{cases}$$

- 節点 j に接続する部材の断面積の総和 :

$$n_j = \sum_{e \in N(j)} x_e$$

- x_e : 部材 e の断面積

- n_j :



- 相補性 :

$$(1 - s_j)n_j = 0$$

新しい定式化 (2/2)

- 不確かな外力の乗数（大きさ） $s_j \in [0, 1]$:

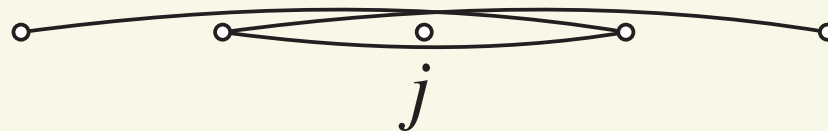
$$s_j = \begin{cases} 1 & \text{節点 } j \text{ が存在するとき} \\ 0 & \text{節点 } j \text{ が消失したとき} \end{cases}$$

- 節点 j をまたぐ部材の断面積の総和 :

$$a_j = \sum_{e \in A(j)} x_e$$

- x_e : 部材 e の断面積

- a_j :



- 相補性 :

$$s_j a_j = 0$$

さらに、実用的な制約として ...

- 細すぎる・太すぎる 部材は 存在してはならない.
- つまり, x_i を部材の断面積として,

$$x_i \in \{0\} \cup [\underline{x}, \bar{x}]$$

- \underline{x}, \bar{x} : 指定値
- 相補性条件を用いて記述できる :

$$\begin{aligned} \underline{x} - z_i &\leq x_i \leq \bar{x} \\ x_i z_i &= 0. \end{aligned}$$

- ただし, 妥当な不等式として,

$$0 \leq z_i \leq \underline{x}$$

ここまです要約すると ...

- 相補性制約：
 - 節点の有無と 不確かな外力の有無を 関連づける.
 - 重複する部材を避ける.
 - 細すぎる部材を避ける.
- 既往の半正定値計画としての定式化 [Ben-Tal & Nemirovski '97]
 - 不確かな外力が作用する節点は指定.
 - 重複する部材は予め除去（解空間が限定されている）.
 - 最適解は しばしば細すぎる部材を含む.

ここまです要約すると ...

- 相補性制約：
 - 節点の有無と不確かな外力の有無を関連づける.
 - 重複する部材を避ける.
 - 細すぎる部材を避ける.
- 既往の半正定値計画としての定式化 [Ben-Tal & Nemirovski '97]
 - 不確かな外力が作用する節点は指定.
 - 重複する部材は予め除去（解空間が限定されている）.
 - 最適解は しばしば細すぎる部材を含む.
- 新しいモデリング
 - 相補性制約つき半正定値計画

発見的解法

- 相補性制約つき半正定値計画 を解きたい.
- CCCP (concave-convex procedure)
 - 別名 :
 - convex-concave procedure
 - DCA (difference-of-convex algorithm) [Pham Dinh & Le Thi '97]
 - EM アルゴリズムなどとも 関連がある.
[Dempster, Laird, & Rubin '77]
- DC (difference-of-convex) 計画 に対する 発見的解法

トラスのロバスト最適設計に対する DC アルゴリズム

- 相補性制約つき半正定値計画 :

$$\begin{array}{ll} \text{Min.} & \mathbf{c}^\top \mathbf{x} \\ \text{s. t.} & (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \Omega, \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0}, \mathbf{y}^\top \mathbf{z} = 0. \end{array}$$

- Ω : 線形行列不等式制約

トラスのロバスト最適設計に対する DC アルゴリズム

- 相補性制約つき半正定値計画 :

$$\begin{aligned} \text{Min.} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s. t.} \quad & (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \Omega, \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0}, \mathbf{y}^\top \mathbf{z} = 0. \end{aligned}$$

- Ω : 線形行列不等式制約

- ペナルティ法 : [Jara-Moroni, Pang, & Wächter '16]

$$\begin{aligned} \text{Min.} \quad & \mathbf{c}^\top \mathbf{x} + \underbrace{\rho (\|\mathbf{y} + \mathbf{z}\|^2 - \|\mathbf{y} - \mathbf{z}\|^2)}_{4\mathbf{y}^\top \mathbf{z}} \\ \text{s. t.} \quad & (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \Omega, \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0}. \end{aligned}$$

- ρ : (十分大きな) ペナルティ・パラメーター

トラスのロバスト最適設計に対する DC アルゴリズム

- 相補性制約つき半正定値計画：

$$\begin{aligned} \text{Min.} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s. t.} \quad & (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \Omega, \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0}, \mathbf{y}^\top \mathbf{z} = 0. \end{aligned}$$

- Ω ：線形行列不等式制約

- ペナルティ法—DC program [Jara-Moroni, Pang, & Wächter '16]

$$\begin{aligned} \text{Min.} \quad & \underbrace{\mathbf{c}^\top \mathbf{x} + \rho \|\mathbf{y} + \mathbf{z}\|^2}_{\square} - \underbrace{\rho \|\mathbf{y} - \mathbf{z}\|^2}_{\square} \\ \text{s. t.} \quad & (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \Omega, \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0}. \end{aligned}$$

- ρ ：（十分大きな）ペナルティ・パラメーター
- DC = “difference of convex”

トラスのロバスト最適設計に対する DC アルゴリズム

- 相補性制約つき半正定値計画 :

$$\begin{aligned} \text{Min.} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s. t.} \quad & (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \Omega, \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0}, \mathbf{y}^\top \mathbf{z} = 0. \end{aligned}$$

- Ω : 線形行列不等式制約

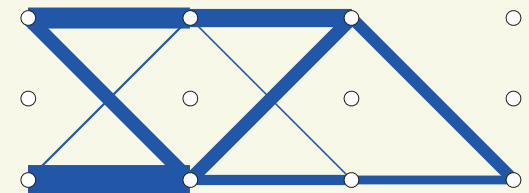
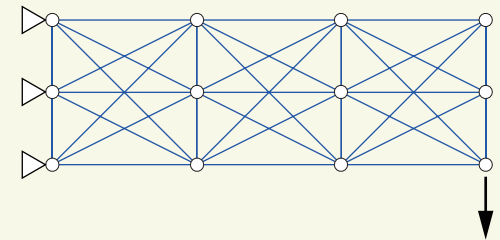
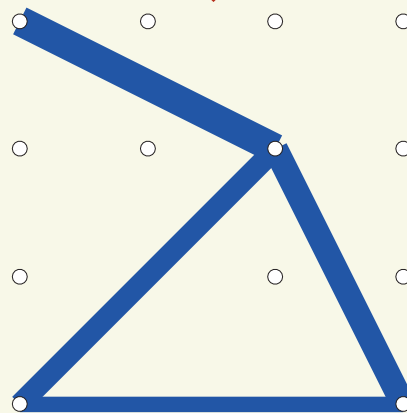
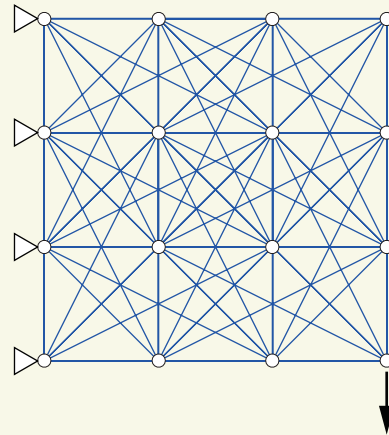
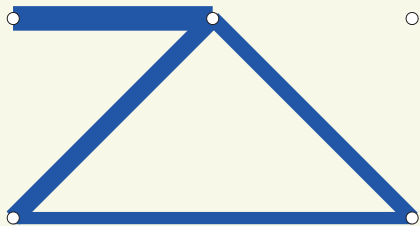
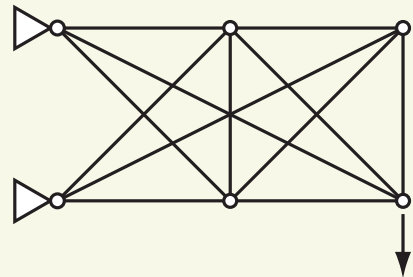
- DC アルゴリズム :

$$\begin{aligned} \text{Min.} \quad & \underbrace{\mathbf{c}^\top \mathbf{x} + \rho_k \|\mathbf{y} + \mathbf{z}\|^2}_{\text{凸}} - \underbrace{\rho_k \|\mathbf{y} - \mathbf{z}\|^2}_{(\mathbf{y}_k, \mathbf{z}_k) \text{ において線形近似}} \\ \text{s. t.} \quad & (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \Omega, \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0}. \end{aligned}$$

- 各反復では子問題として半正定値計画を解く.
- ρ_k は 徐々に大きくする.

数値実験 (1) : 小規模な問題例

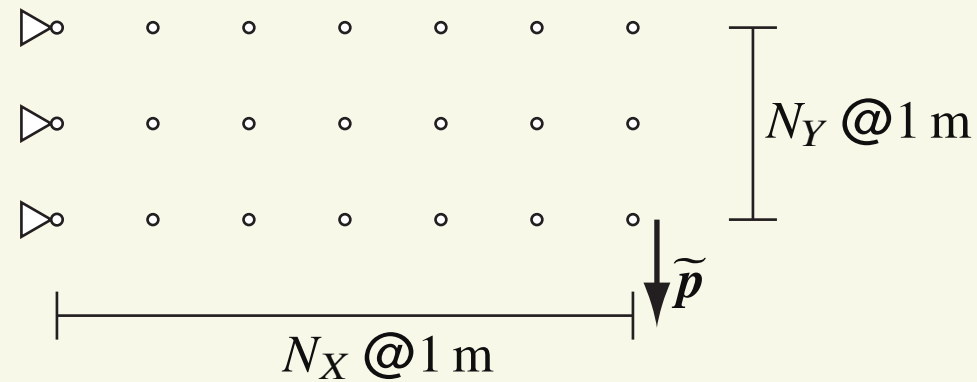
- 分枝限定法 (YALMIP) と比較
 - 提案手法は 大域解に収束



数値実験 (2)

- 問題設定

- 完全グラフから, 3 m よりも長い枝は除去

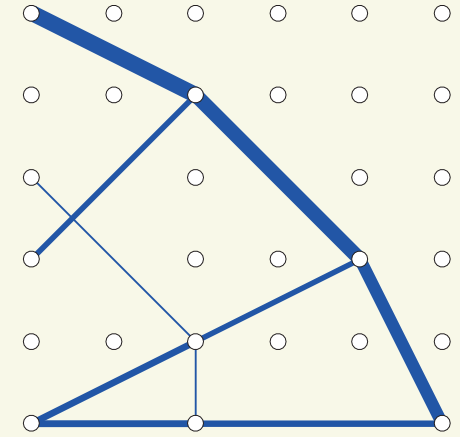
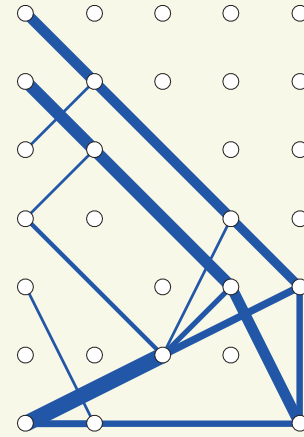
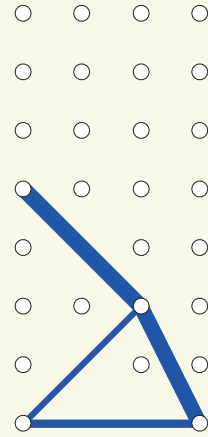


- 荷重の不確か性を考慮

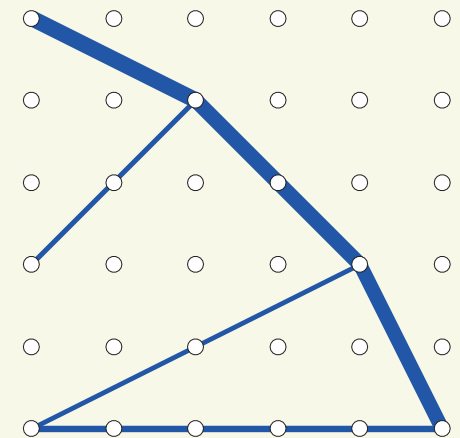
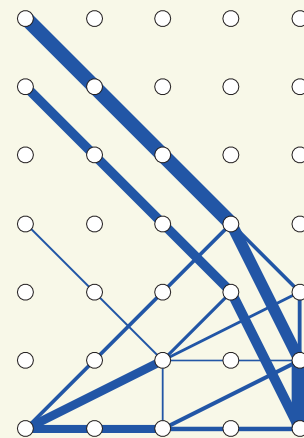
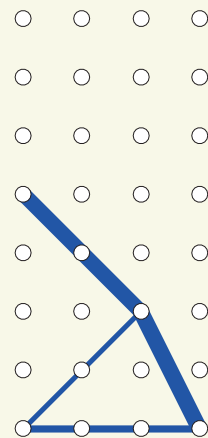
- \tilde{p} : 公称荷重

数値実験 (2)

- ロバスト最適化



- 不確実性なし

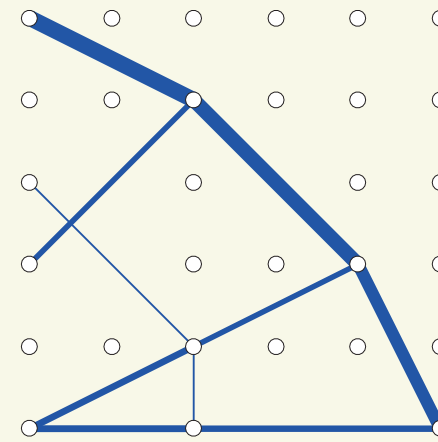
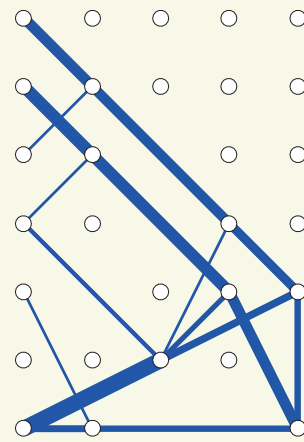
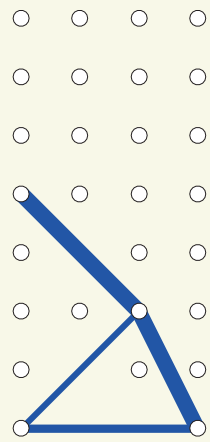


- ロバスト最適化の解では,

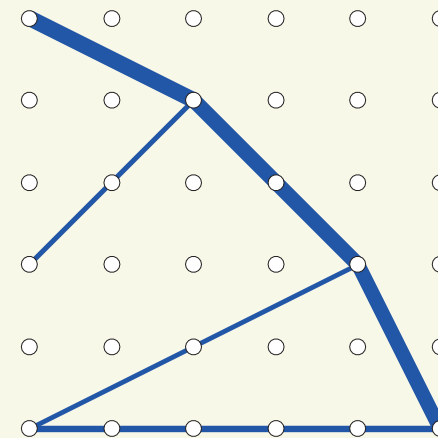
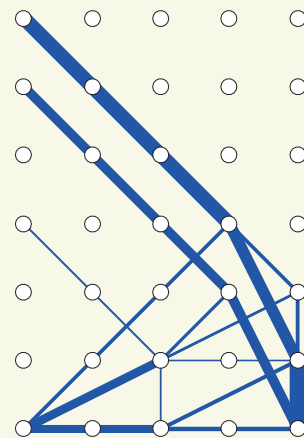
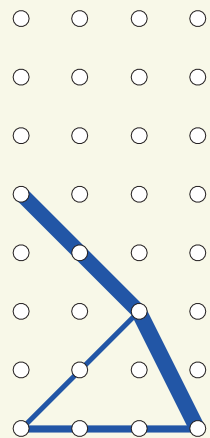
- 「鎖」がしばしば長い単一の部材に.
- 初期設定より長い部材は存在しない.

数値実験 (2)

- ロバスト最適化



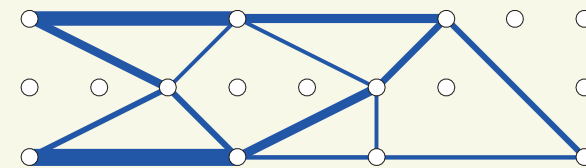
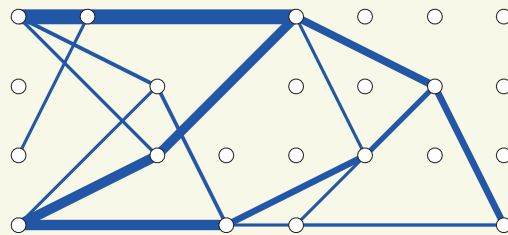
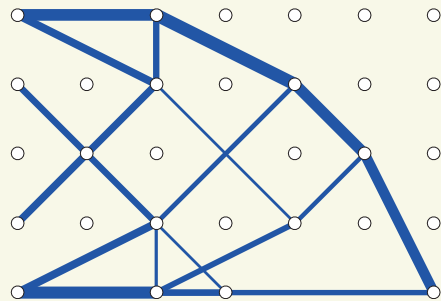
- 不確実性なし



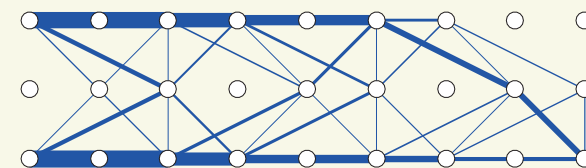
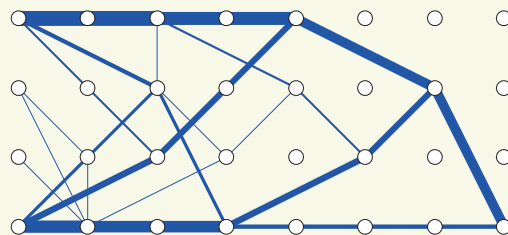
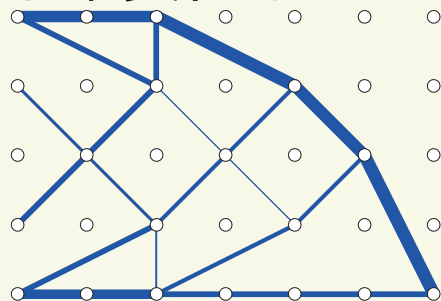
(N_X, N_Y)	#mbr.	#iter.	Time (s)
(3, 7)	250	9	46.4
(4, 6)	292	39	242.9
(5, 5)	306	35	249.6

数値実験 (2) : 他の例

- ロバスト最適化



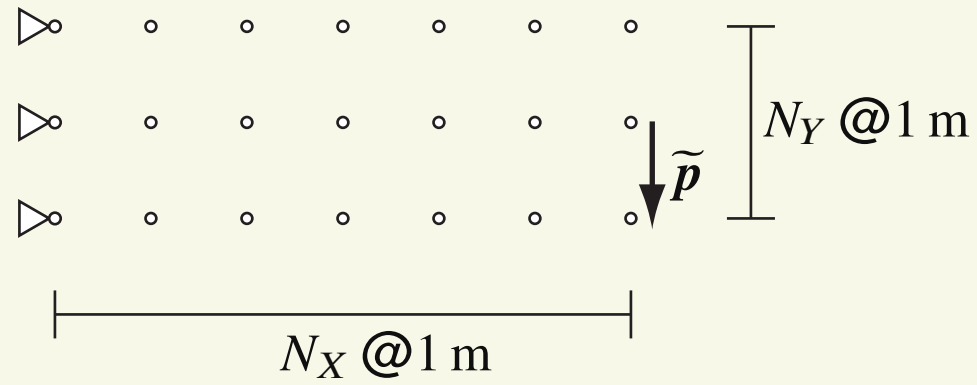
- 不確実性なし



(N_X, N_Y)	#mbr.	#iter.	Time (s)
(6, 4)	292	21	142.1
(7, 3)	250	40	193.9
(8, 2)	180	32	103.9

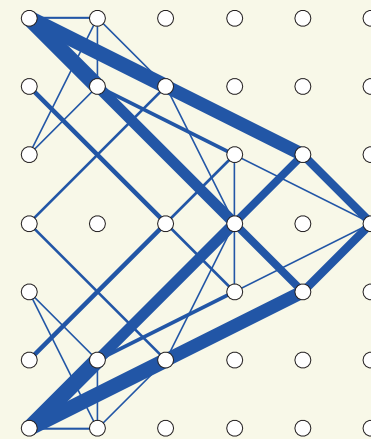
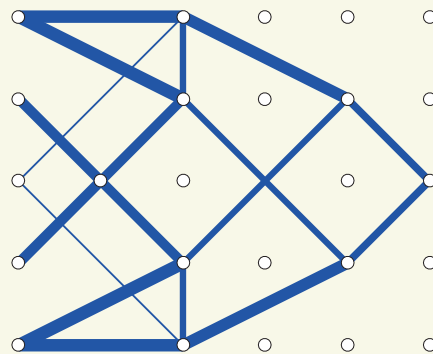
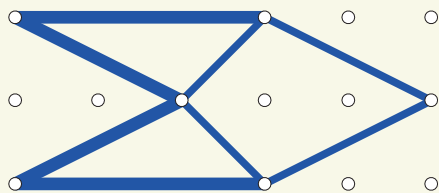
数值実験 (3)

- 問題設定

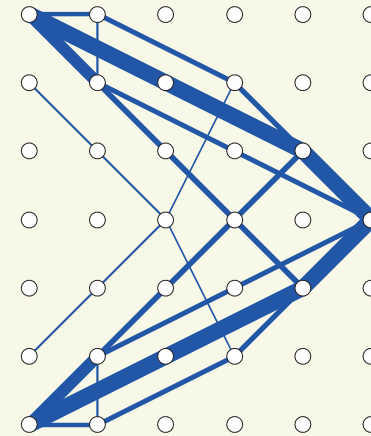
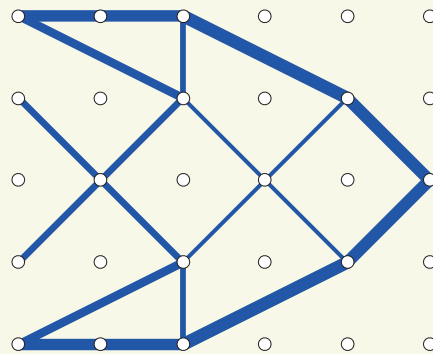
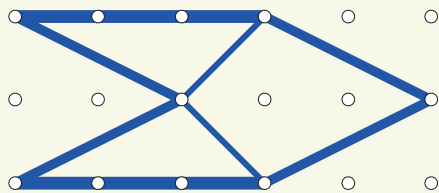


数値実験 (3)

- ロバスト最適化の解



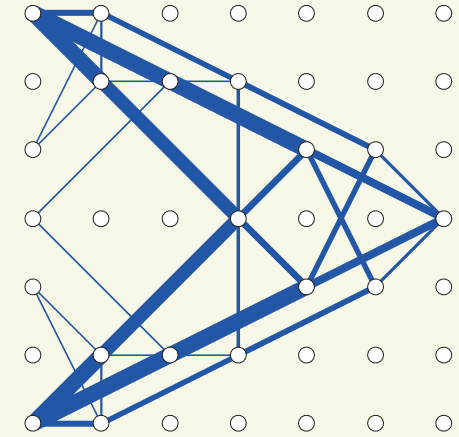
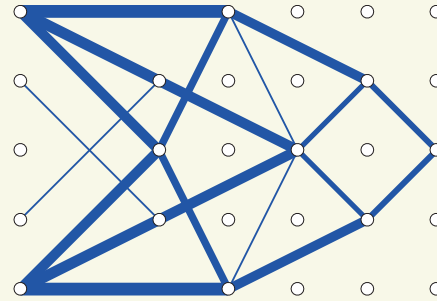
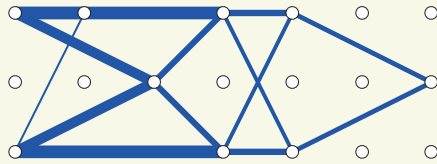
- 不確実性なしの最適解



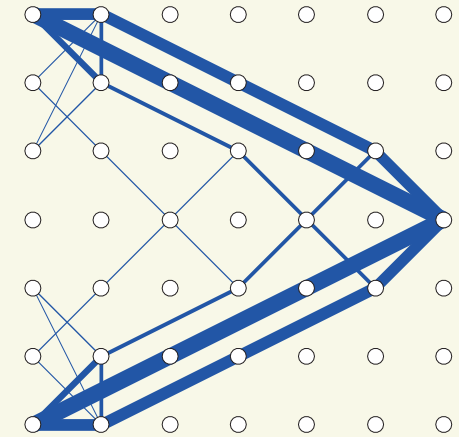
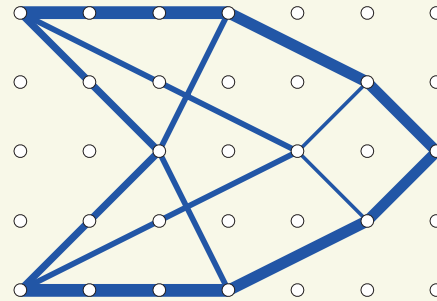
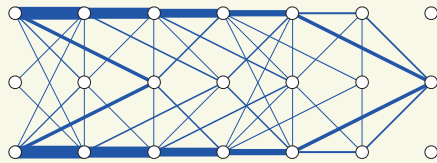
(N_X, N_Y)	#memb.	#iter.	Time (s)
(5, 2)	108	18	25.0
(5, 4)	240	16	62.1
(5, 6)	372	26	212.8

数値実験 (3) : 別の例

- ロバスト最適化の解



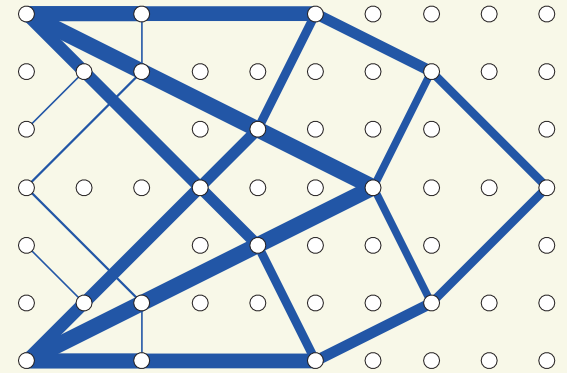
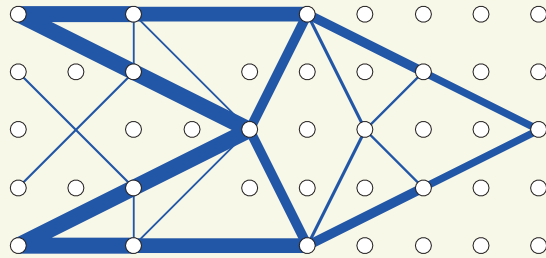
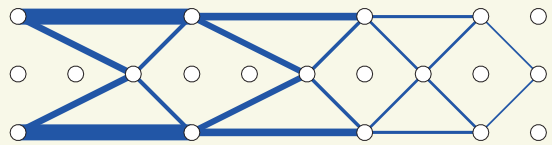
- 不確実性なしの最適解



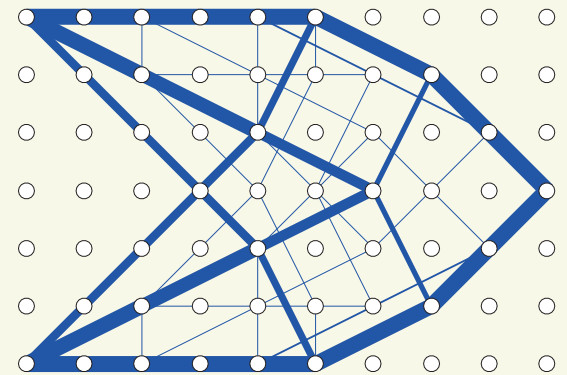
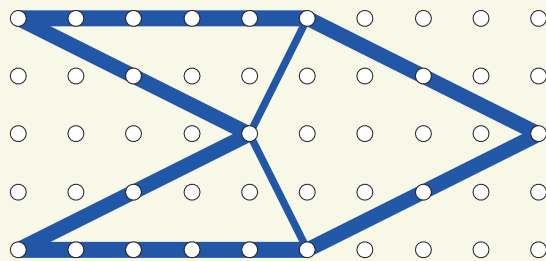
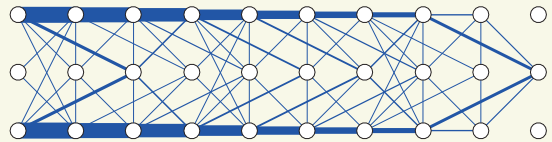
(N_X, N_Y)	#memb.	#iter.	Time (s)
(6, 2)	132	43	70.4
(6, 4)	292	19	113.9
(6, 6)	452	19	208.0

数値実験 (3) : さらに別の例

- ロバスト最適化の解



- 不確実性なしの最適解



(N_X, N_Y)	#memb.	#iter.	Time (s)
(9, 2)	294	28	100.7
(9, 4)	448	24	286.4
(9, 6)	692	35	863.0

まとめ

- トラスのロバスト最適設計（外力が不確実）
 - 頂点の有無 と 不確実な外力の有無
 - 重複する枝は許容しない。
 - 頂点の有無 と それをまたぐ枝の有無
 - 細い枝は許容しない.
- 定式化
 - 相補性制約つき半正定値計画
 - 等価な DC 計画
- 発見的解法
 - DC アルゴリズム
 - 各反復では 半正定値計画を解く.