

# 最適化システム導入プロジェクト -土木構造物保守計画支援システム-

数理計画部(Optimization Division)

研究員 岩永二郎

[iwanaga@msi.co.jp](mailto:iwanaga@msi.co.jp)

# Point

---

- ❑ プロジェクトがどのように進められるのか？
- ❑ 通常システム構築と何が違うのか？
- ❑ 数理システムが提供できる数理計画の技術

# 土木構造物保守計画支援システム

## 2009年数理システムユーザカンファレンス

- 「土木の長期修繕費の平準化をめざした対策時期の割当て問題」

株式会社 NEWJEC 安野貴人

## 2010年オペレーションズ・リサーチ学会

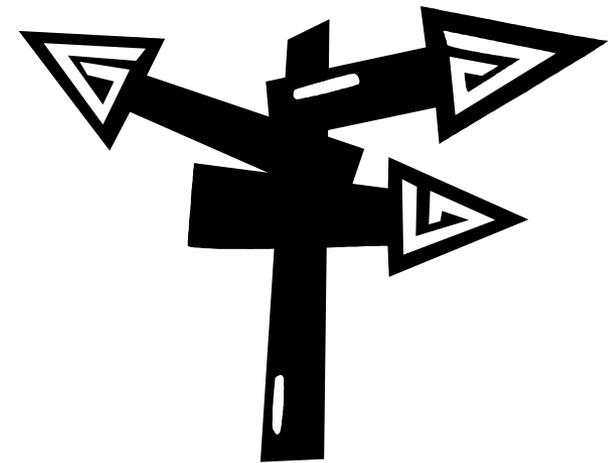
- 「ラグランジュ緩和を用いた土木構造物の長期修繕計画における予算平準化問題の汎用解法」

株式会社 NEWJEC 安野貴人

株式会社 数理システム 田辺隆人, 岩永二郎

# 本プロジェクトの狙い

- 最適化システムのプロトタイプを作成
  - 数理計画によるアプローチの有用性の証明
  - 今後の拡張課題・方向性の調査
  - モデリング・スキームの構築を最重要課題とし  
インターフェースは簡潔でOK



# インターフェース

---

## 実行方法

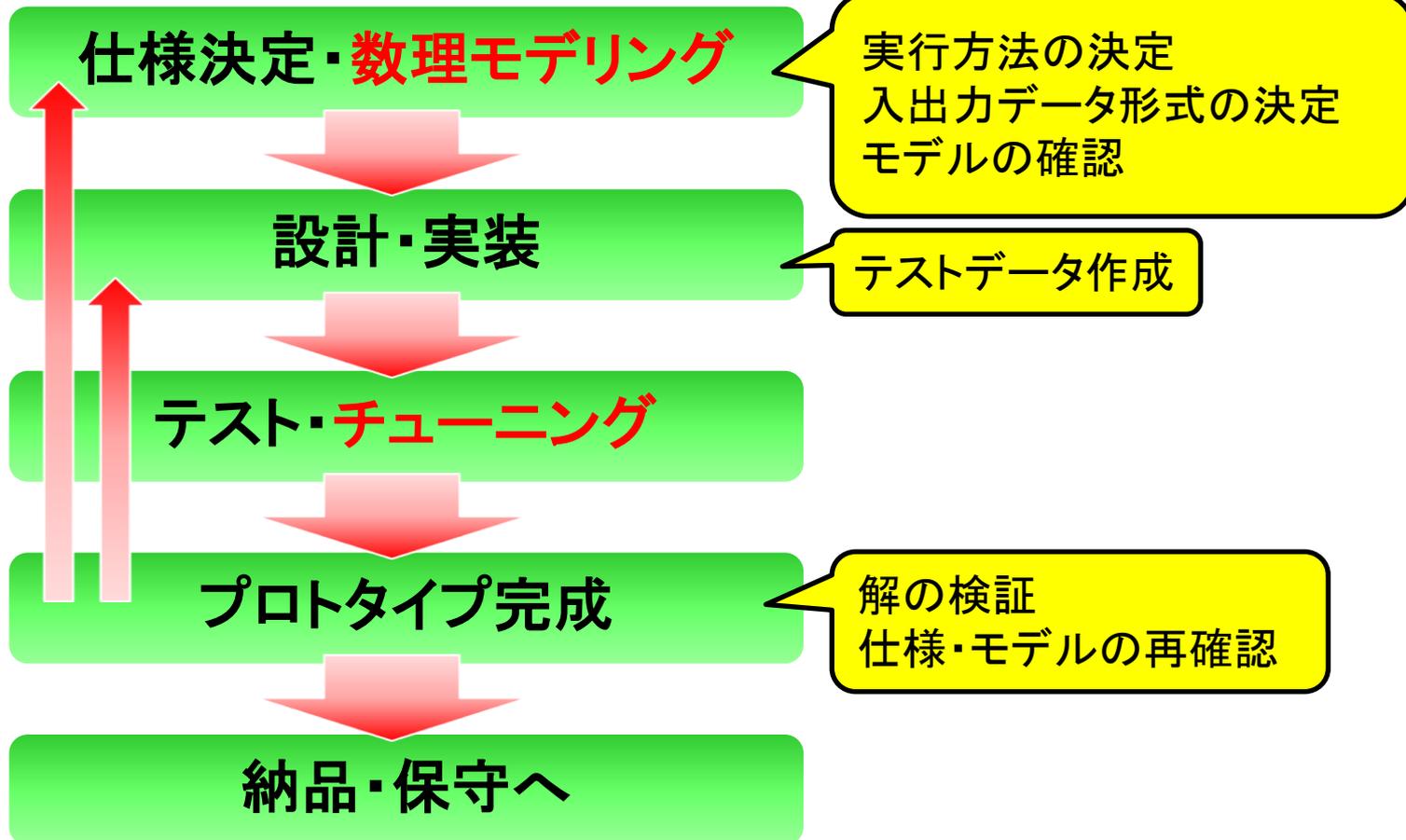
- コマンドプロンプト  
アイコンのダブルクリックによる実行も可

## 入出力データの形式

- csv形式ファイル/dat形式ファイル

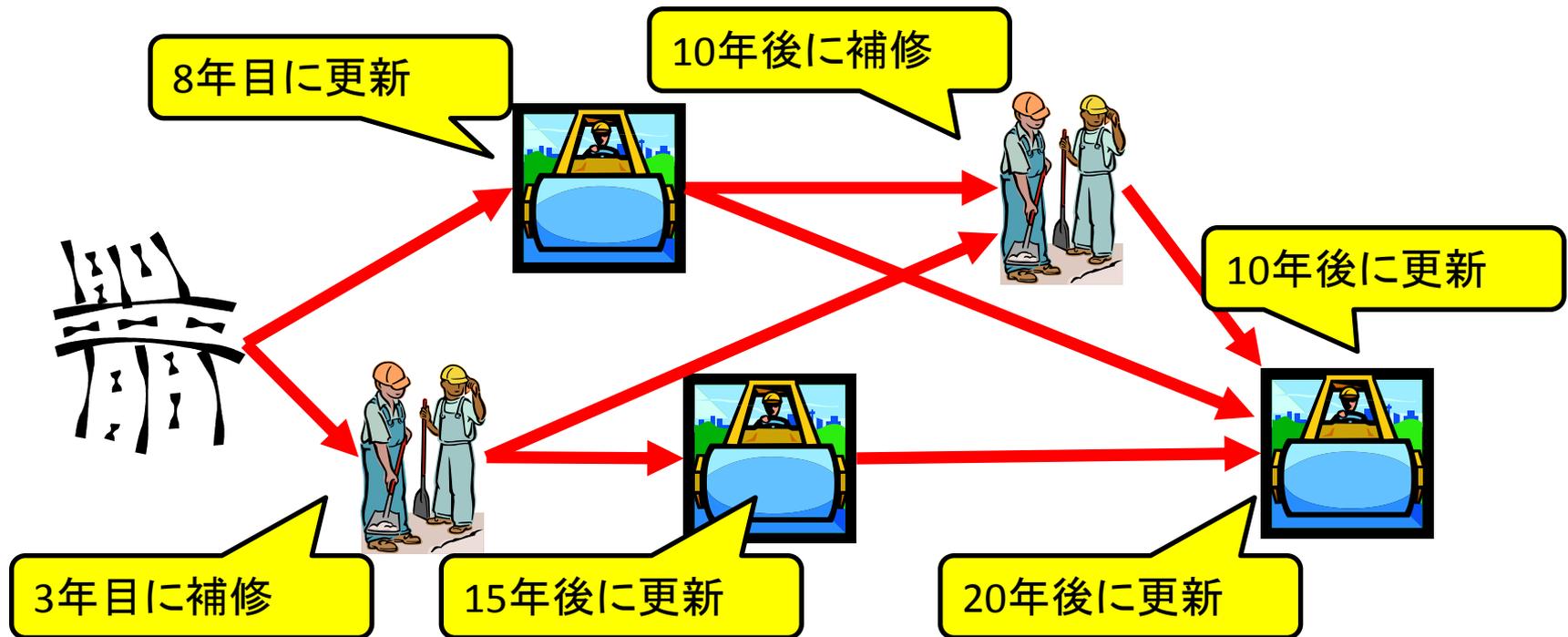
# 開発のスケジュール

## 2～3ヶ月の小規模プロジェクト



# 保守計画の策定

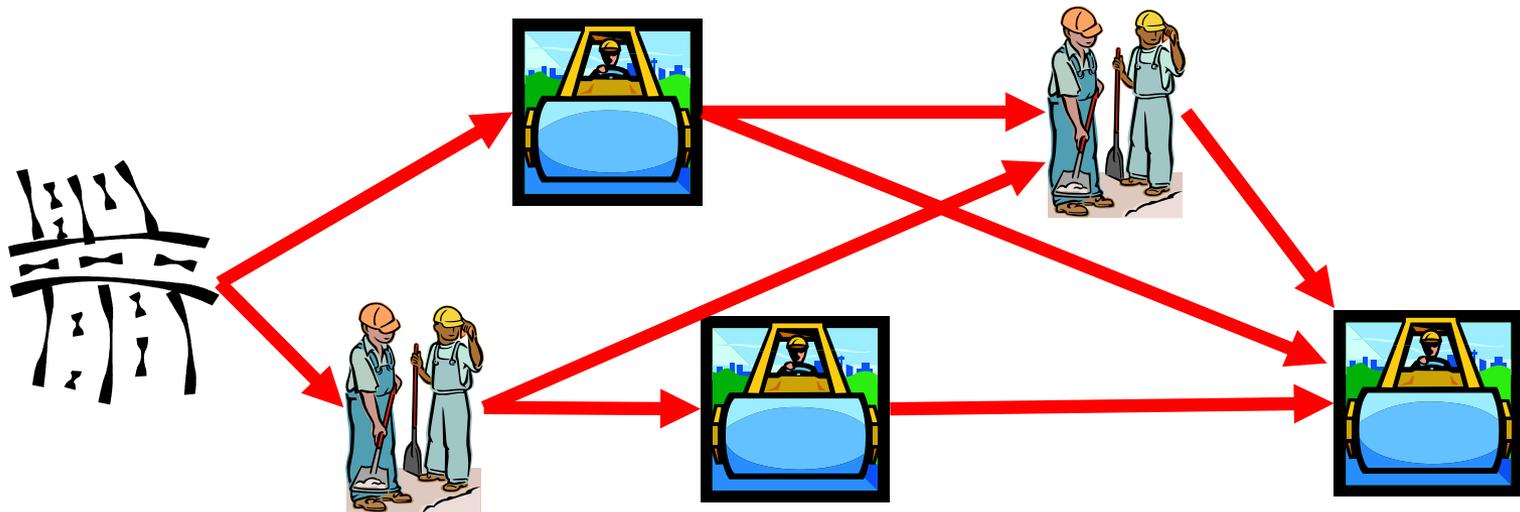
修繕計画の策定:「何年目にどの修繕をするか」の決定



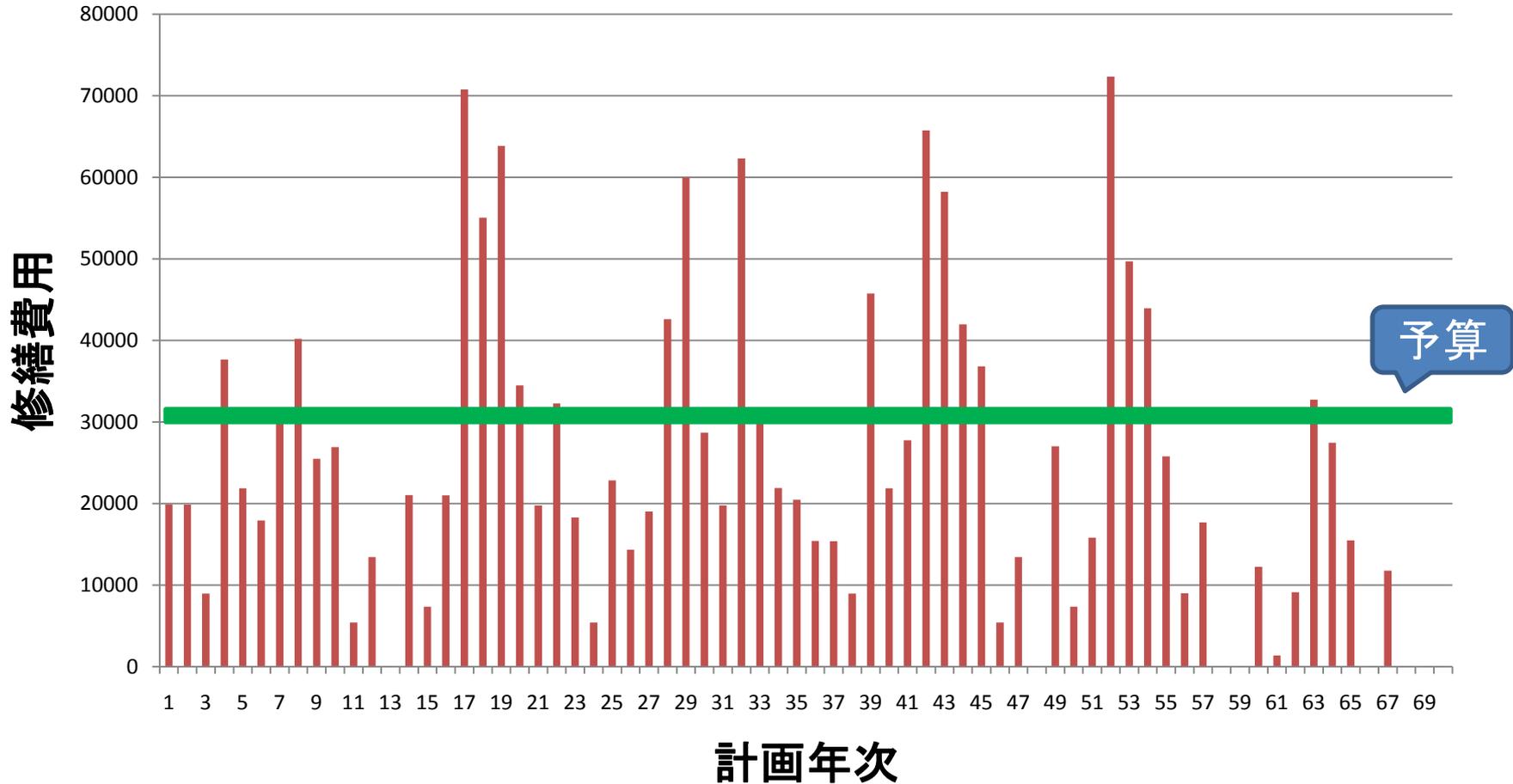
--- 修繕のストーリーは複数の分岐点がある ---

# 背景

- 以前は老朽化した道路から優先的に修繕
  - ➡ 昨今の**予算削減**により計画が実施できない
- 高度成長期の土木構造物が**一斉に老朽化**
  - ➡ 人間による計画作成が不可能な規模(数千オーダー)



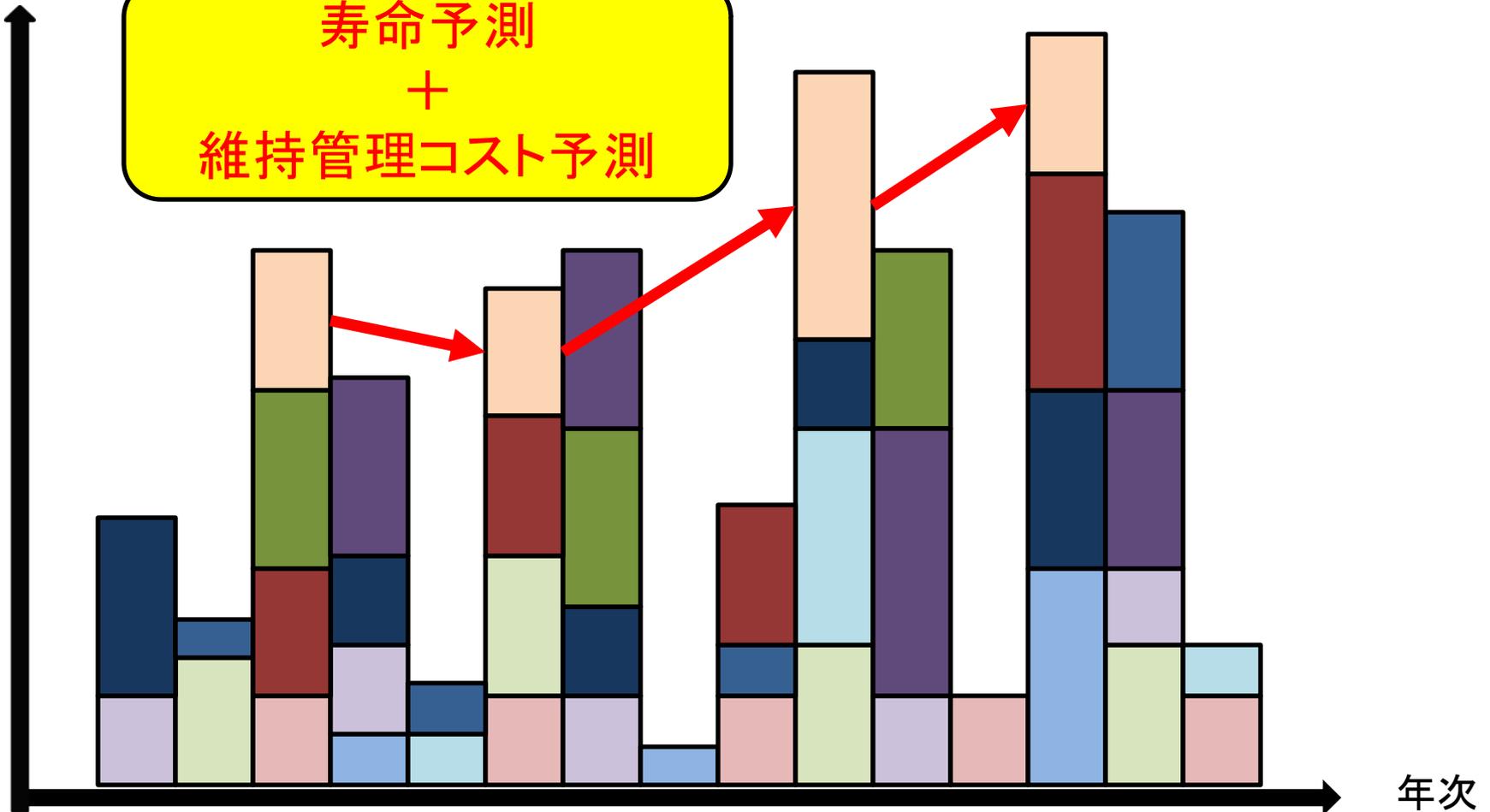
# 必要となる修繕費用のグラフ



# 計画年次と修繕費用

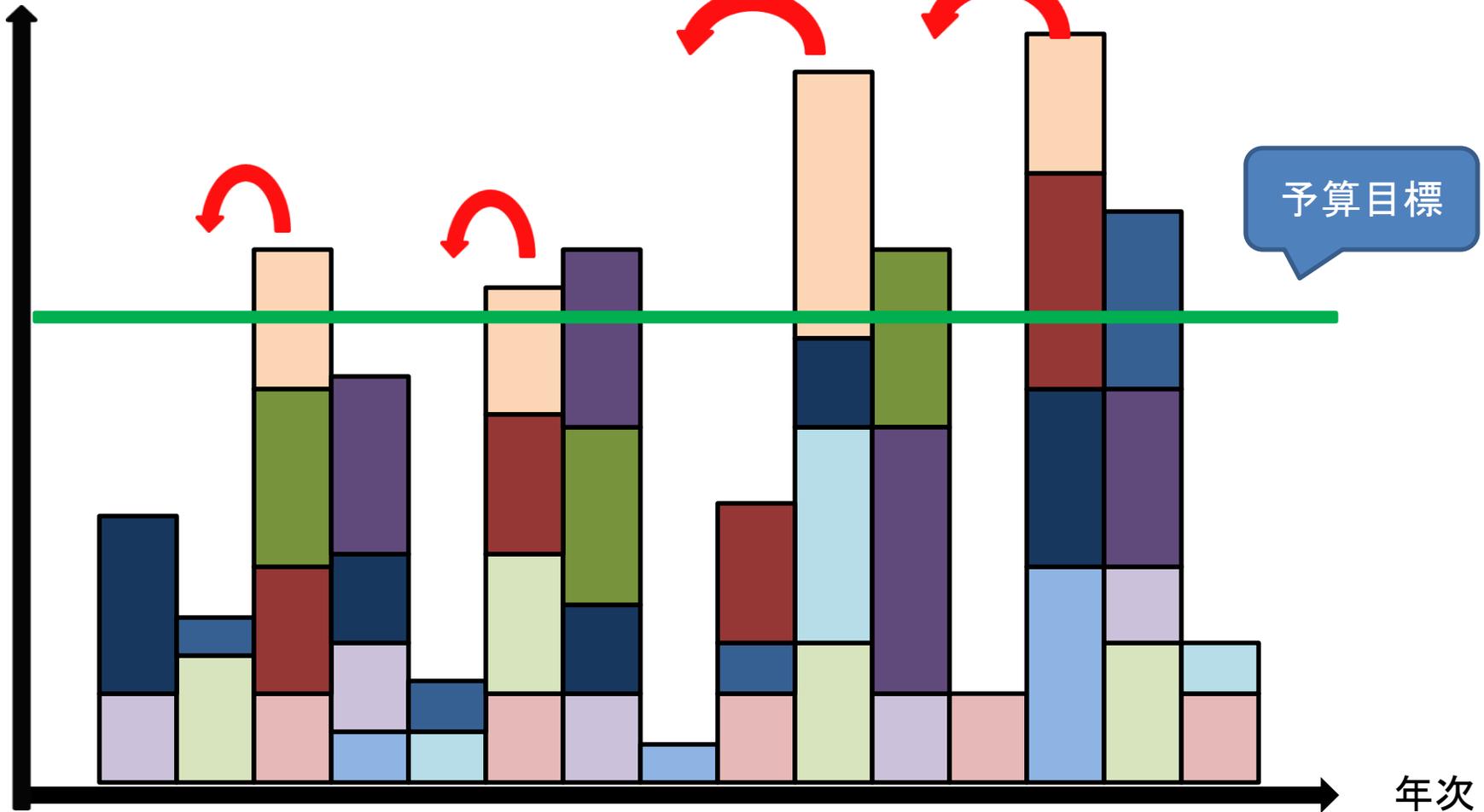
修繕費用

寿命予測  
+  
維持管理コスト予測



# 計画年次と修繕費用

修繕費用

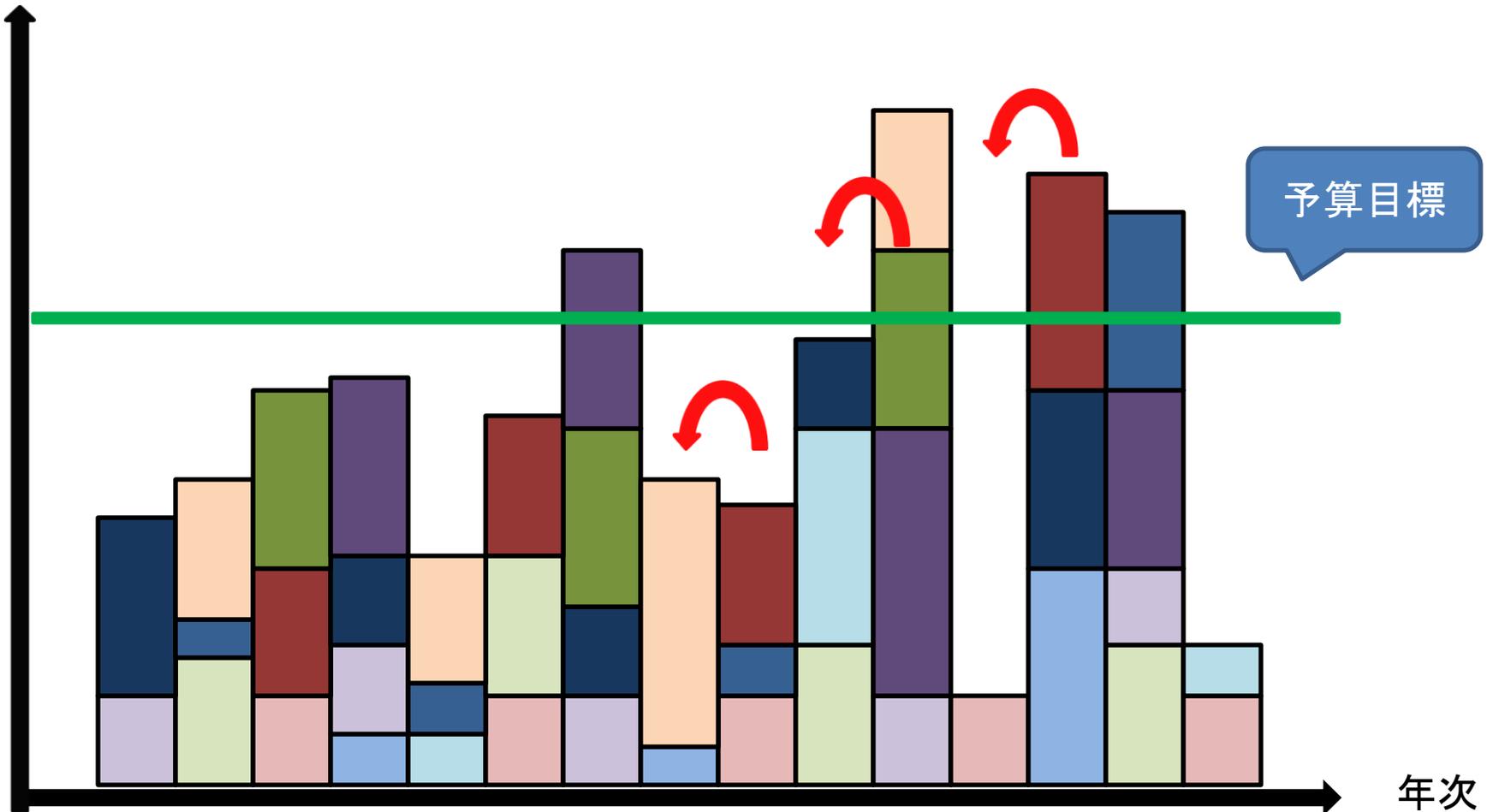


予算目標

年次

# 計画年次と修繕費用

修繕費用

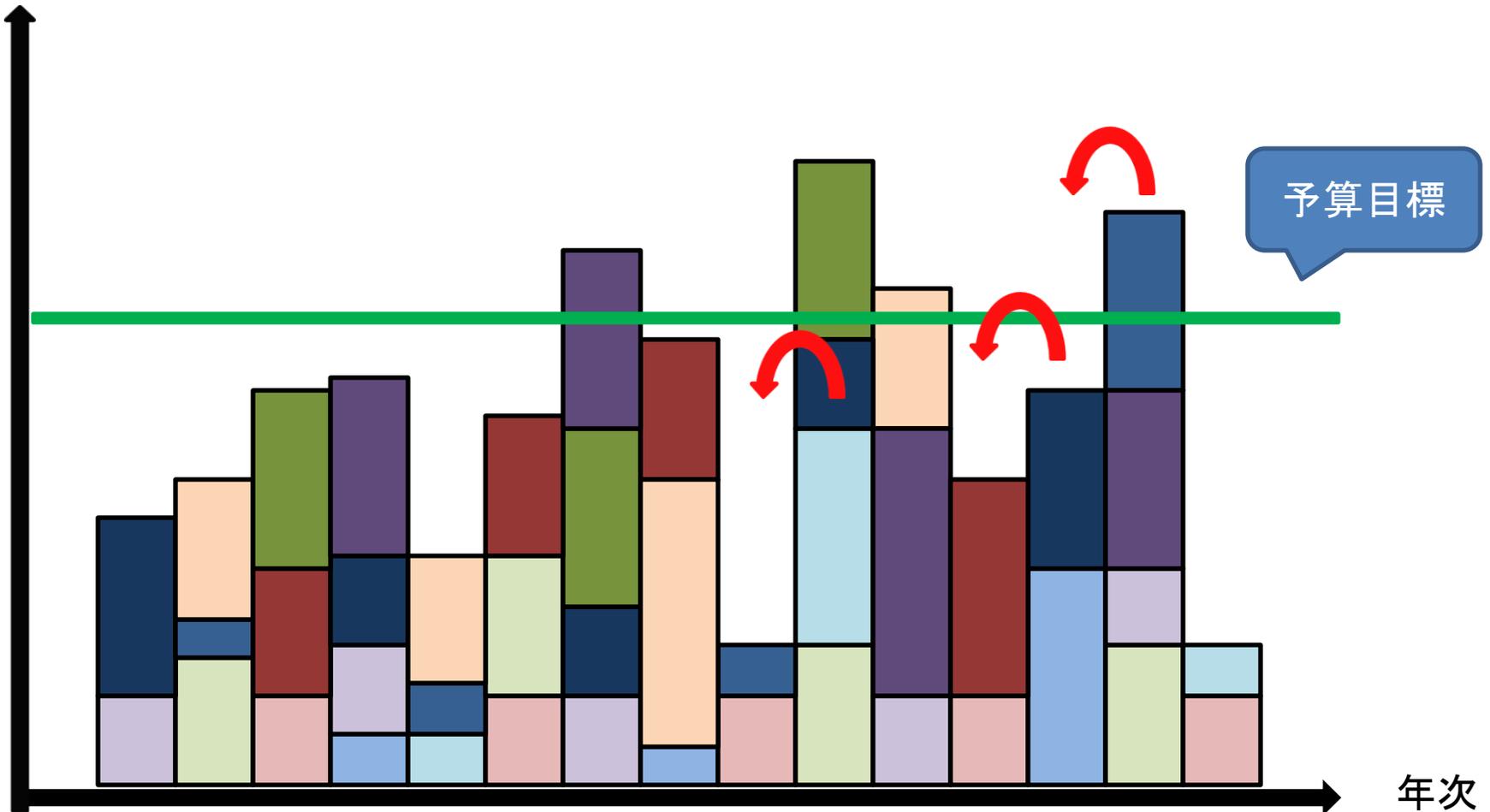


予算目標

年次

# 計画年次と修繕費用

修繕費用

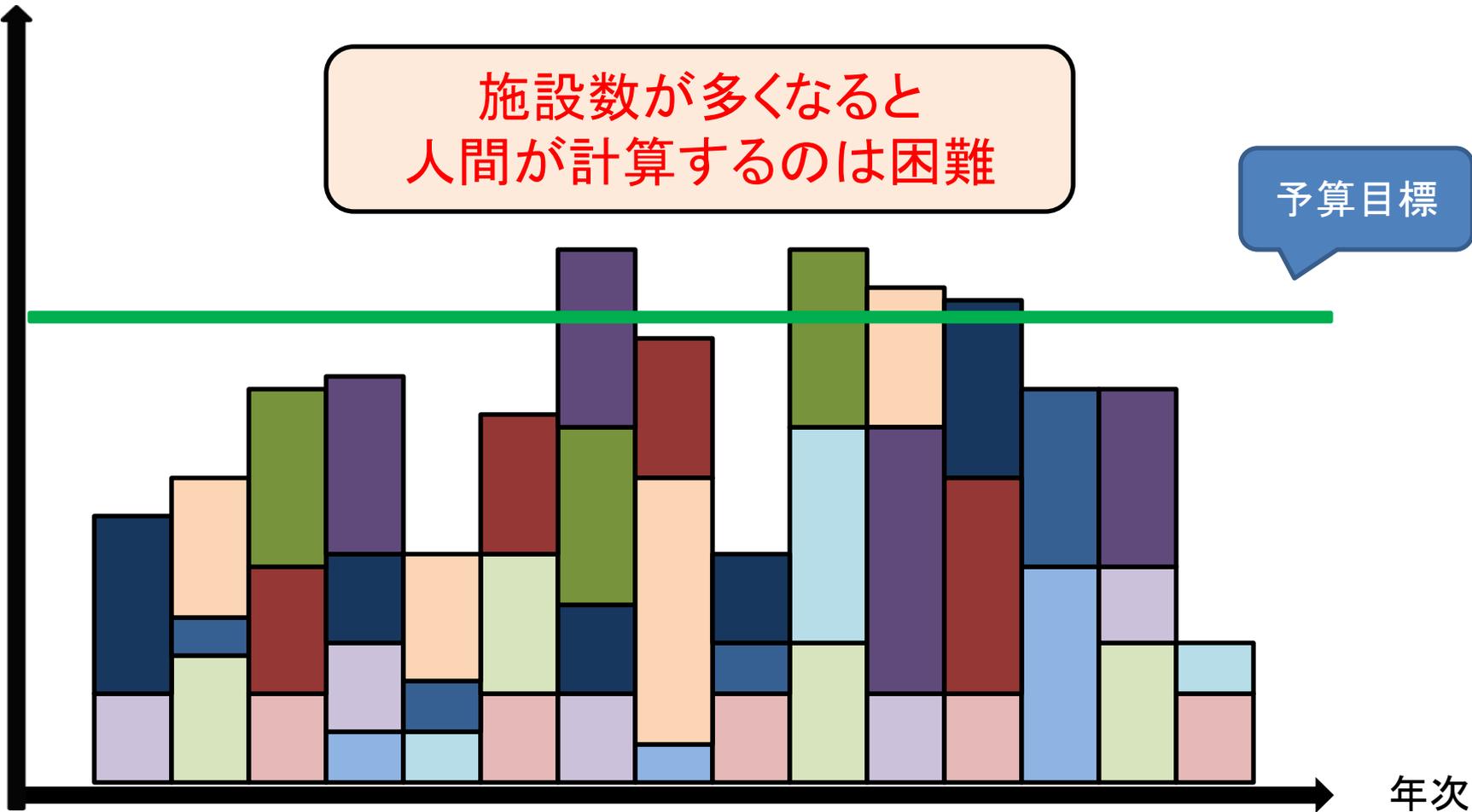


# 計画年次と修繕費用

修繕費用

施設数が多くなると  
人間が計算するのは困難

予算目標



# モデリング ---STEP1---

- 決定対象: ①各施設が何年目に何の修繕をするかを決定
- 制約式: ②各施設は現実的に可能な修繕計画を選択する
- 最適化の指標: ③予算違反量の最小化

- ①は修繕計画というパターンに置き換えることができる
- ②は各施設について修繕計画のパターンを全列挙できる

ex.) 青梅街道の向こう70年分の修繕計画の可能性を全列挙



# 基本的なアプローチ

各施設は**修繕パターン**を持っている



道路

修繕パターン1

3年目更新

17年目更新

修繕パターン2

4年目更新

18年目更新

修繕パターン3

2年目更新

17年目更新

修繕パターン1

8年目補修 40年目更新

修繕パターン2

7年目補修 38年目更新



トンネル

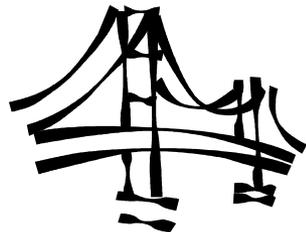
修繕パターン1

15年目補修 35年目補修

修繕パターン2

13年目補修 32年目補修

...



橋梁



# 基本的なアプローチ

各施設は修繕パターンの1つを選択



道路

修繕パターン3



全体の保守計画の策定

修繕パターン1



トンネル

修繕パターン1



橋梁

# モデリング ---STEP2---

■ **集合**:  $U$  施設の集合

$I_u$  施設  $u \in U$  の修繕パターンの集合

$T$  計画年次の集合

■ **変数**: 施設  $u \in U$  がパターン  $i \in I_u$  を選択する(1)か否(0)か

$$x_{ui} \in \{0, 1\} \quad (u \in U, i \in I_u)$$

■ **制約式**: 各施設は修繕パターンの1つを選択

$$\sum_{i \in I_u} x_{ui} = 1 \quad \forall u \in U$$

■ **目的関数**: 予算違反量の最小化

$$\min \sum_t \max(0, \sum_{u,i} x_{ui} \cdot cost_{tui} - budget_t)$$

## モデリング ---STEP2---

集合

変数

制約

目的

```
model.smp * - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
[Icons]
0 10 20 30 40 50
1 Set U,T,I;
2 |
3 Element u(set=U);
4 Element t(set=T);
5 Element i(set=I);
6 |
7 Parameter cost(index=(t,u,i));
8 Parameter budget(index=t);
9 |
10 IntegerVariable x(index=(u,i), type=binary);
11 |
12 sum(x[u,i], i) == 1, u<U;
13 |
14 softConstraint(1);
15 sum(x[u,i]*cost[t,u,i], (u,i)) <= budget[t], t<T;
16 |
17 solve();
18 |
19 |
20 |
```

(0)か

## モデリング ---STEP2---

集合

変数

制約

目的

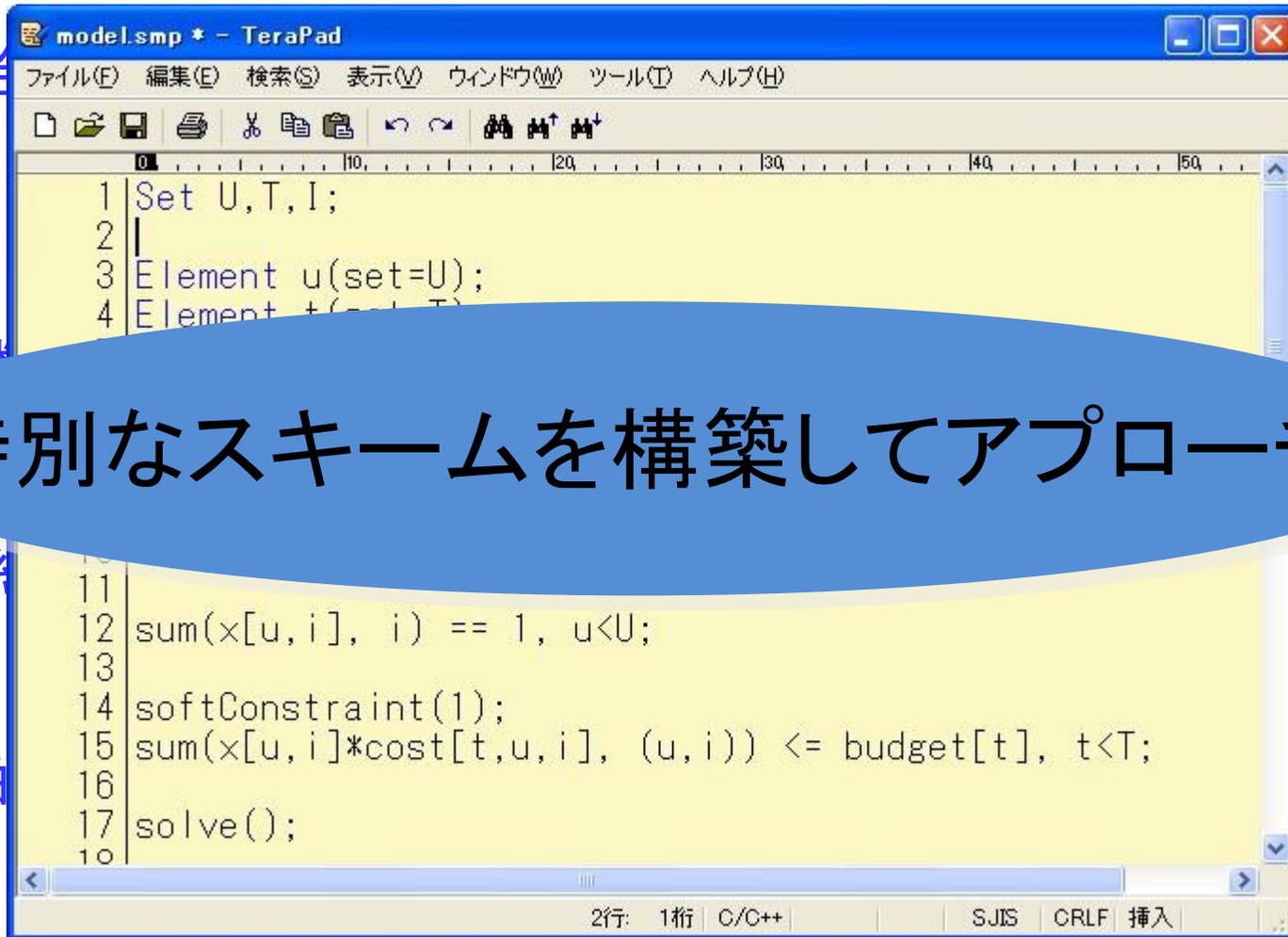
```
model.smp * - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
|
|
1 Set U,T,I;
2 |
3 Element u(set=U);
4 Element t(set=T);
5 Element i(set=I);
6 |
7 |
8 |
9 |
10 IntegerVariable x(index=(u,i), type=binary);
11 |
12 sum(x[u,i], i) == 1, u<U;
13 |
14 softConstraint(1);
15 sum(x[u,i]*cost[t,u,i], (u,i)) <= budget[t], t<T;
16 |
17 solve();
18 |
19 |
20 |
```

(0)か

大規模なのでこのままでは解けない!!

## モデリング ---STEP2---

集合



```
model.smp * - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
1 Set U,T,I;
2
3 Element u(set=U);
4 Element t(set=T);
5
6
7
8
9
10
11
12 sum(x[u,i], i) == 1, u<U;
13
14 softConstraint(1);
15 sum(x[u,i]*cost[t,u,i], (u,i)) <= budget[t], t<T;
16
17 solve();
18
```

変数

(0)か

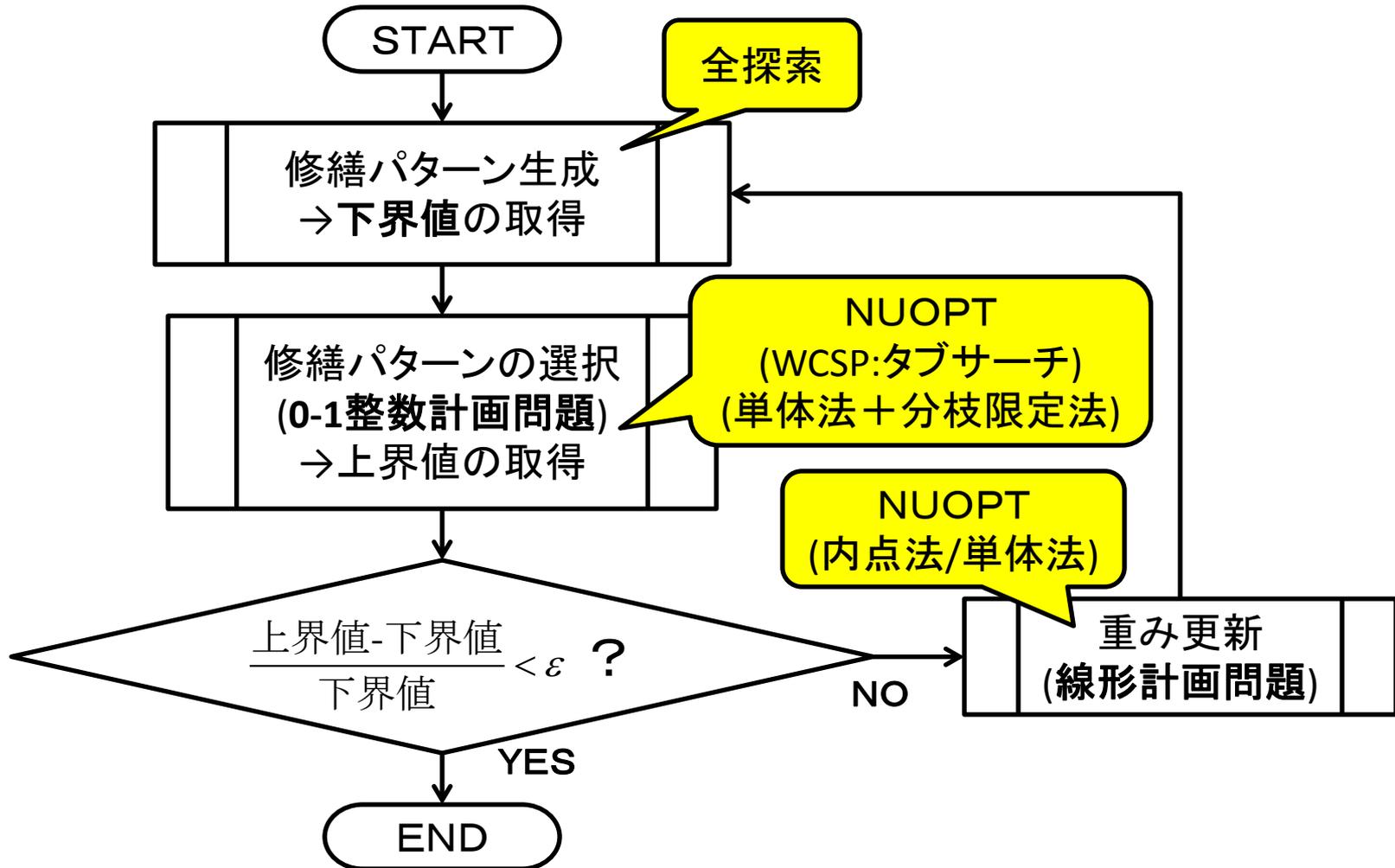
特別なスキームを構築してアプローチ

制約

目的

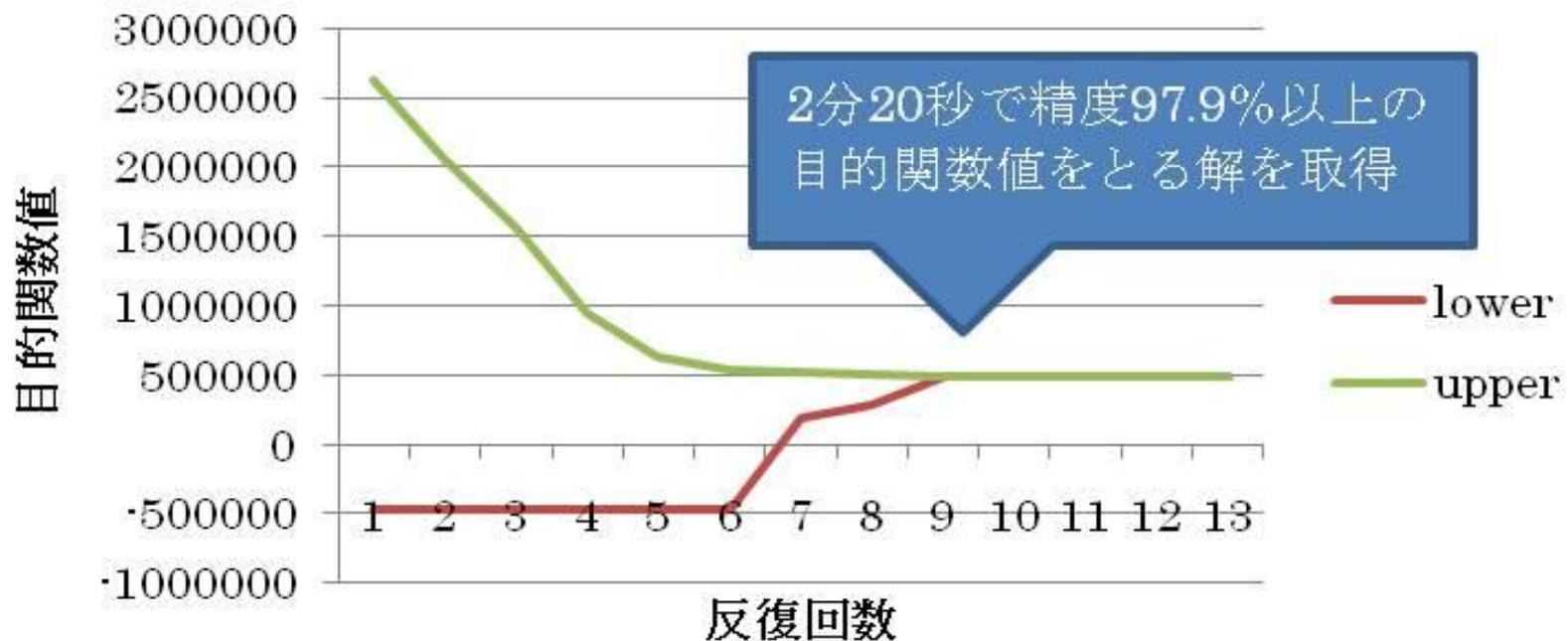
# 列生成法 + ラグランジュ緩和

# 解法のフローチャート

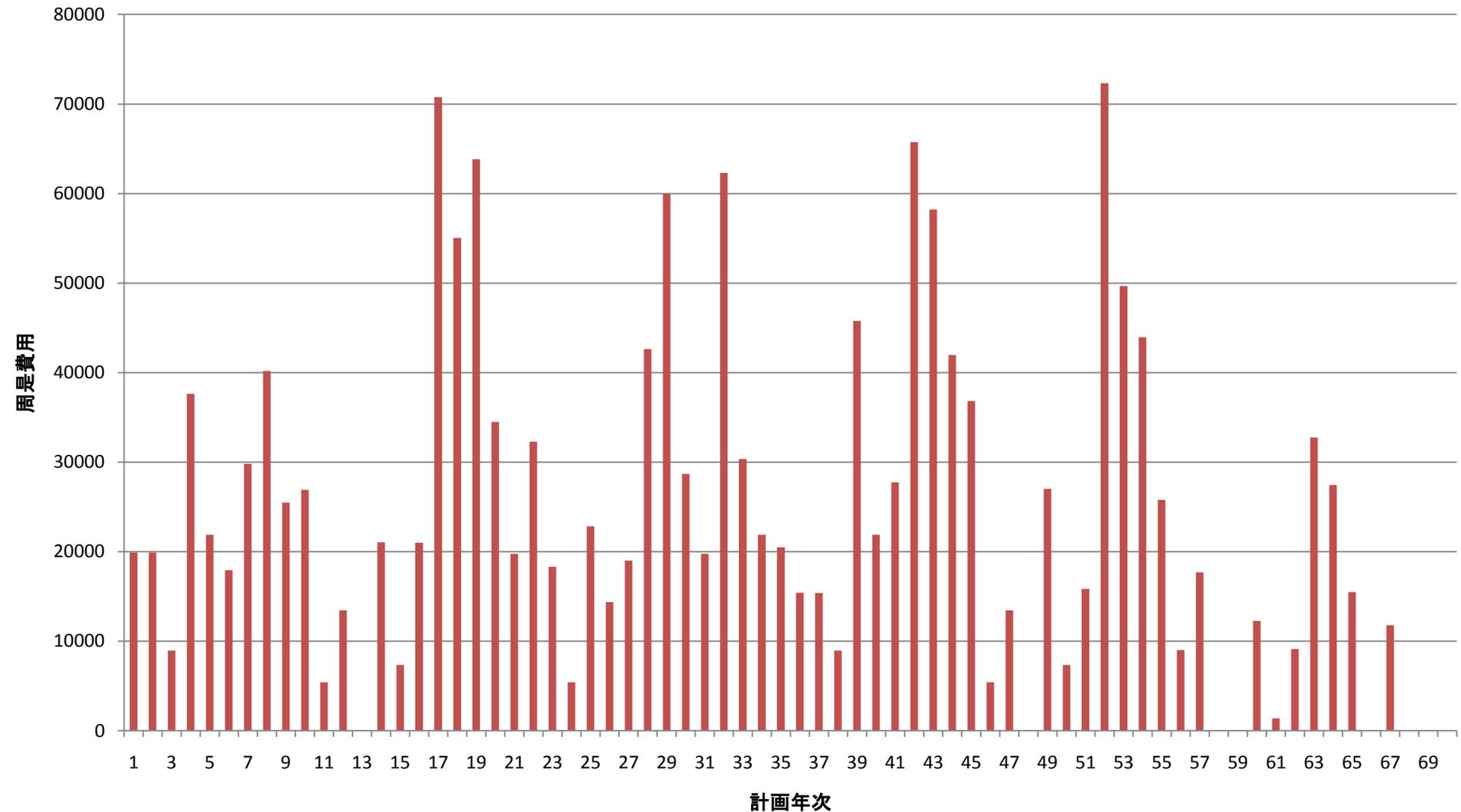


# 収束の様子

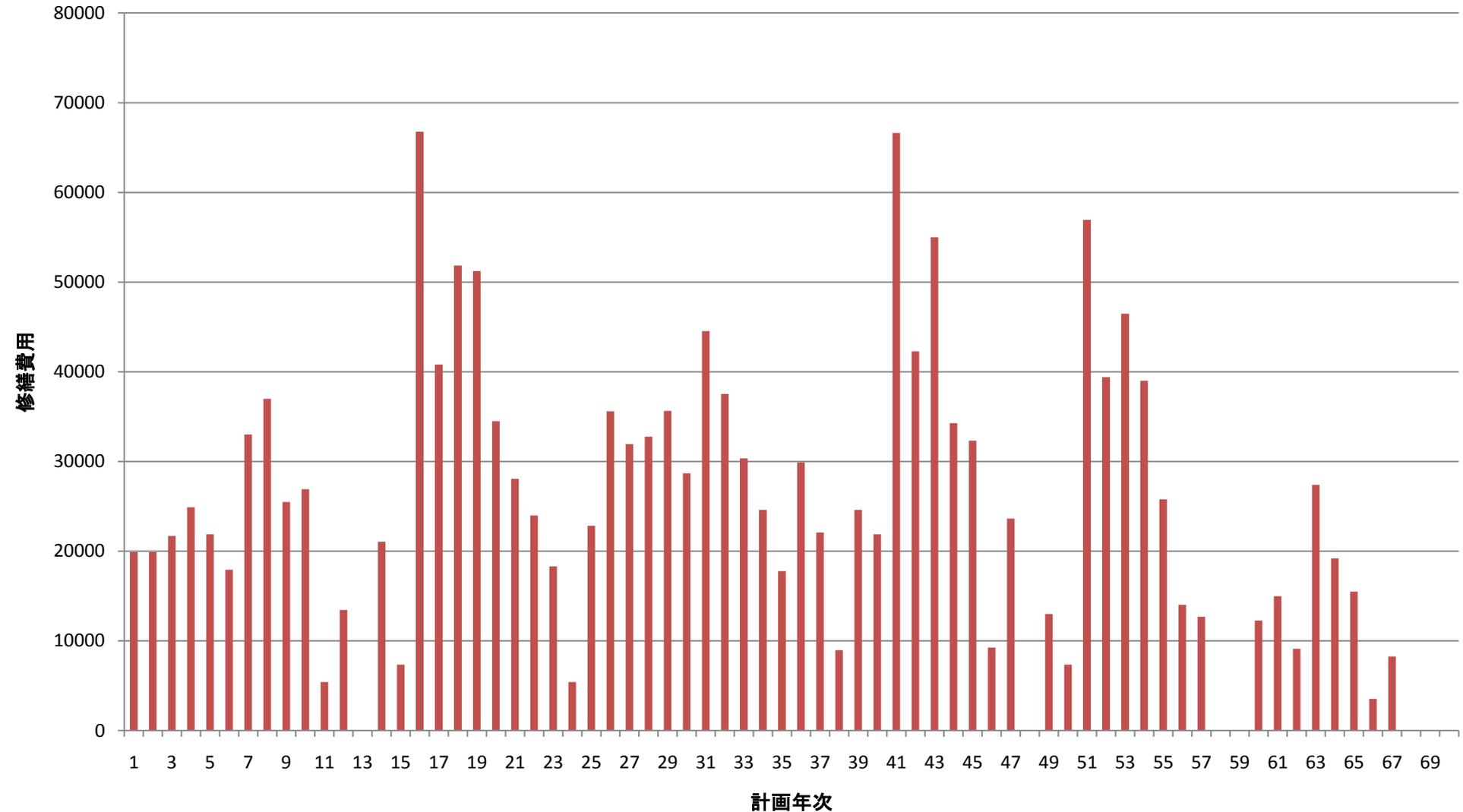
## 反復回数と目的関数値の 収束の様子



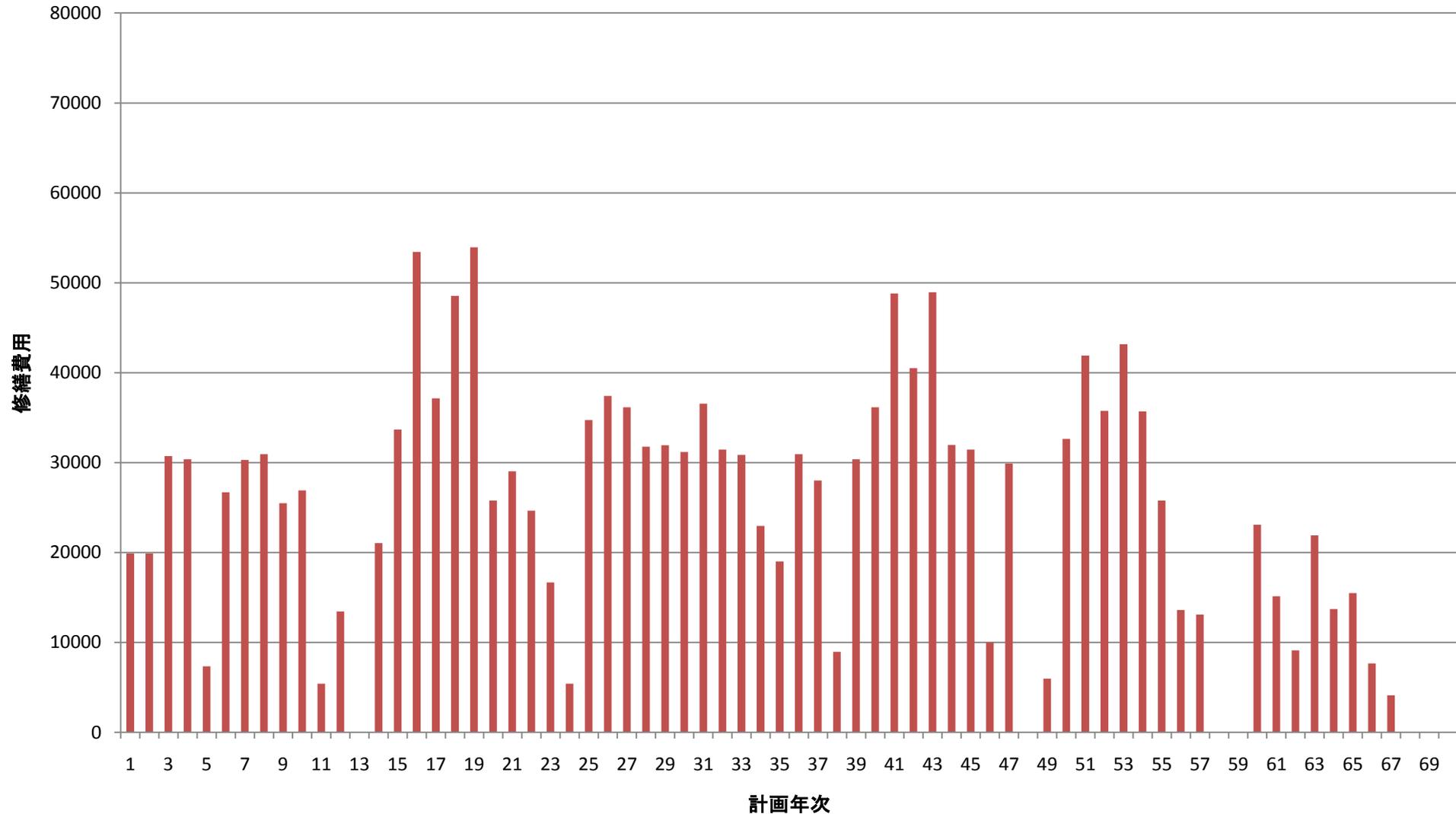
## 計画年次と修繕費用 (反復回数1)



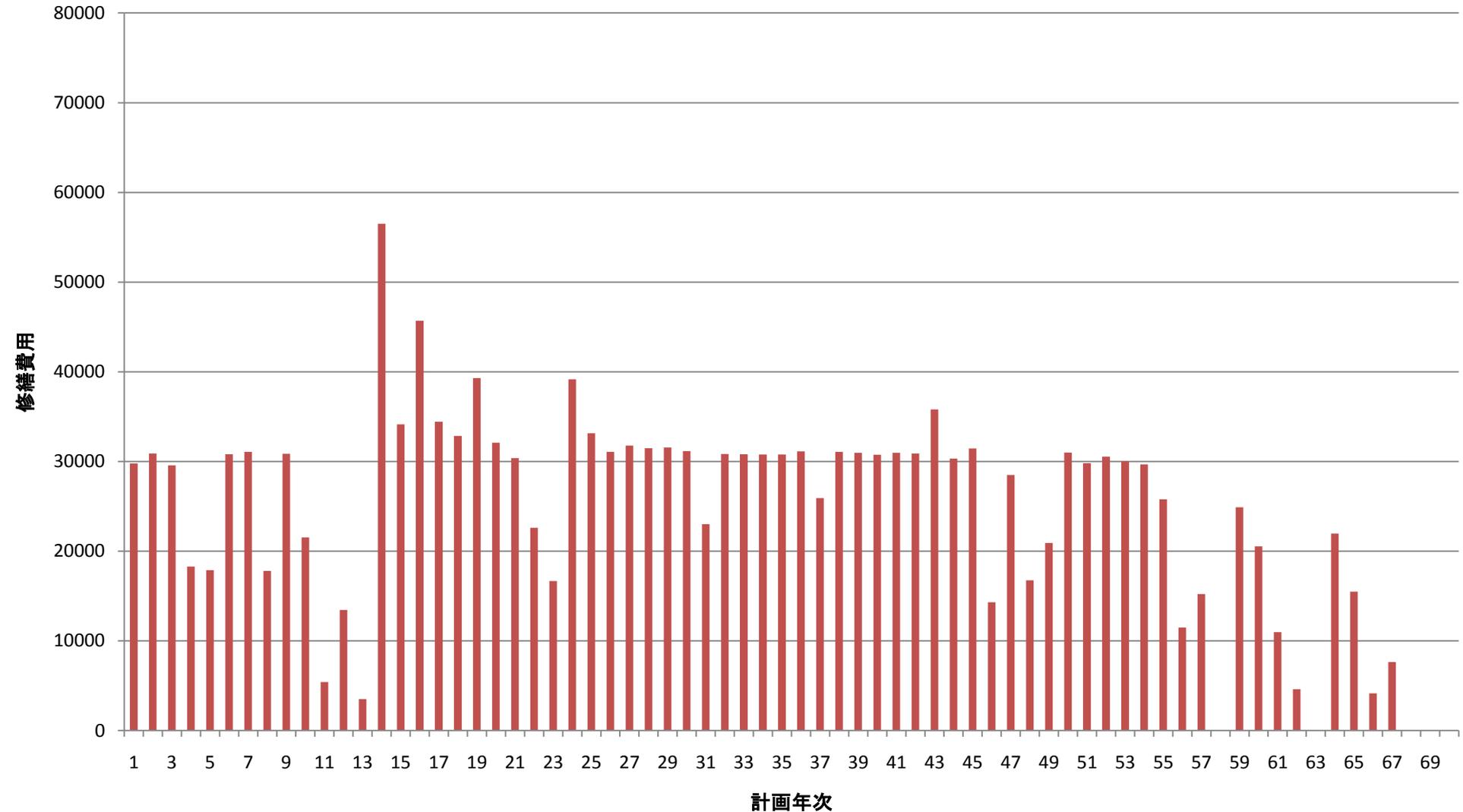
## 計画年次と修繕費用 (反復回数2)



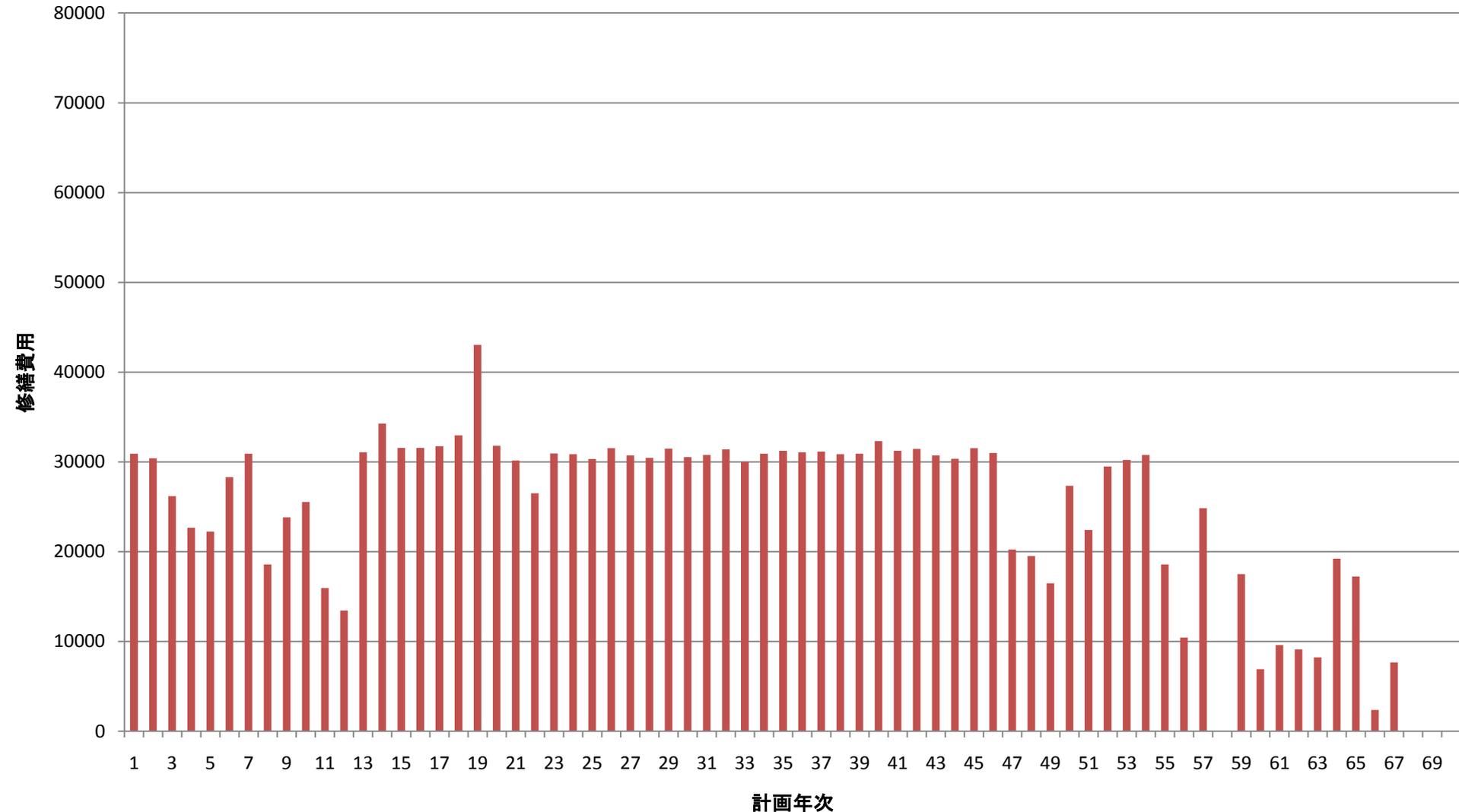
## 計画年次と修繕費用 (反復回数3)



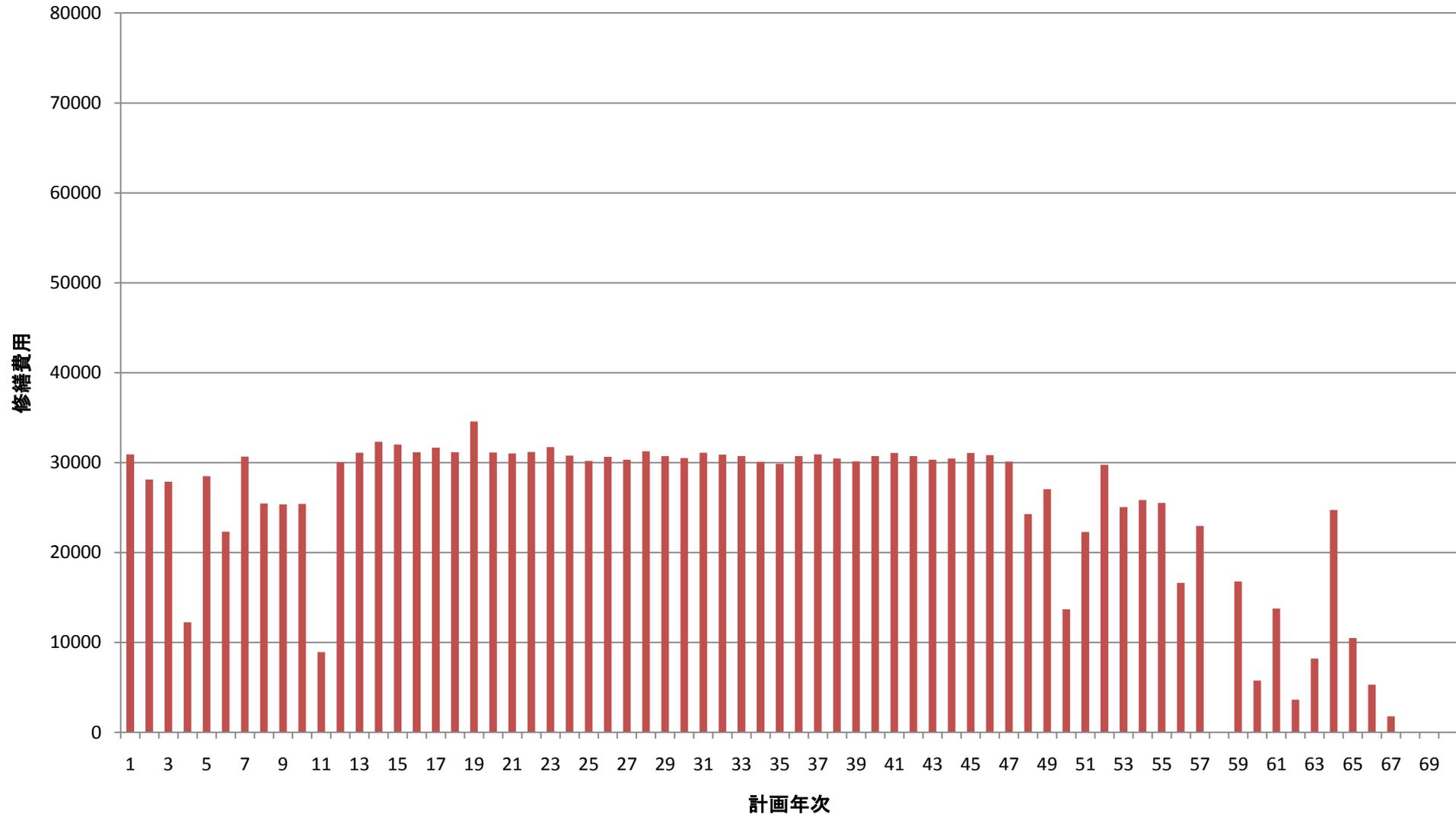
## 計画年次と修繕費用 (反復回数4)



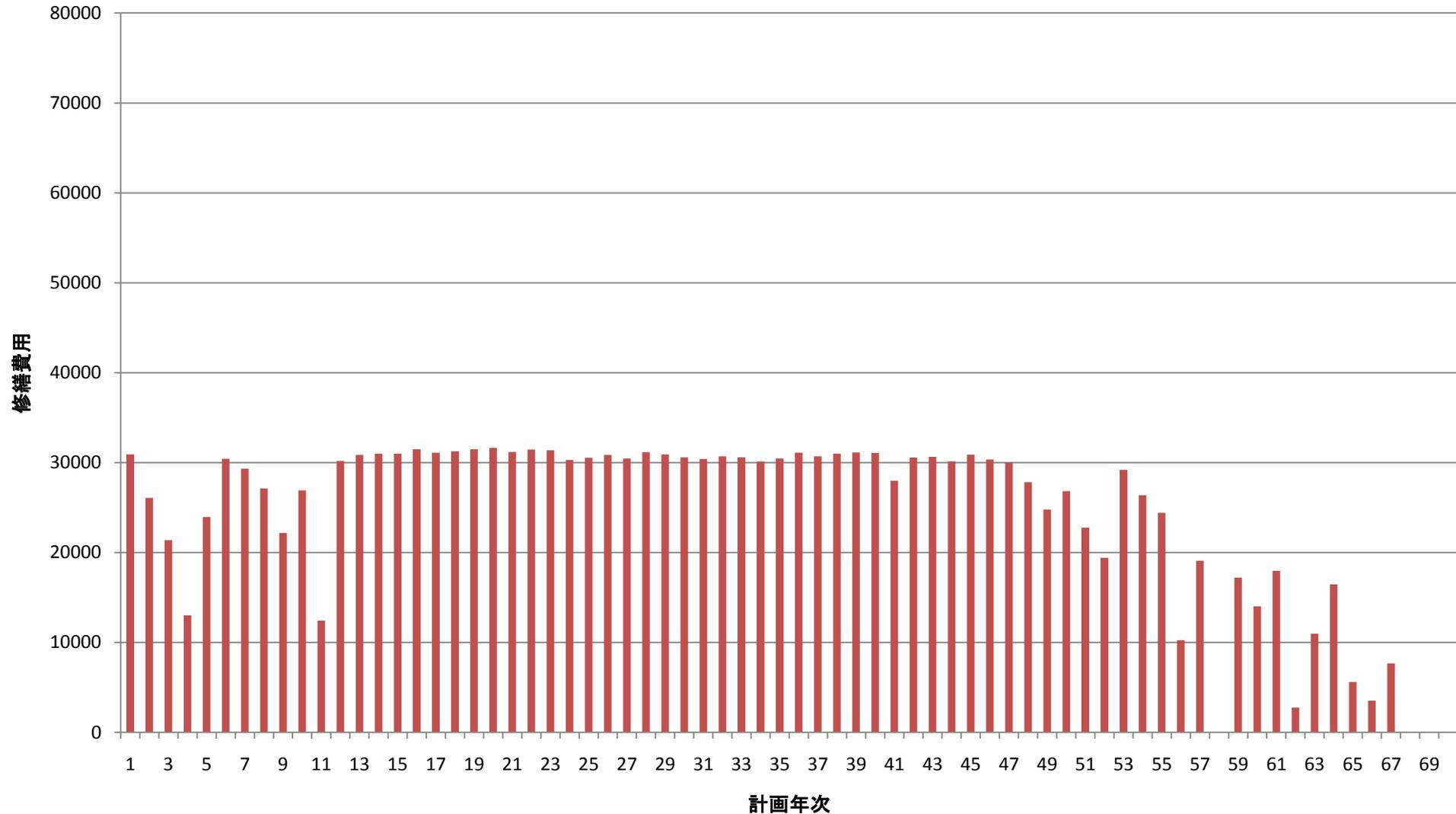
## 計画年次と修繕費用 (反復回数5)



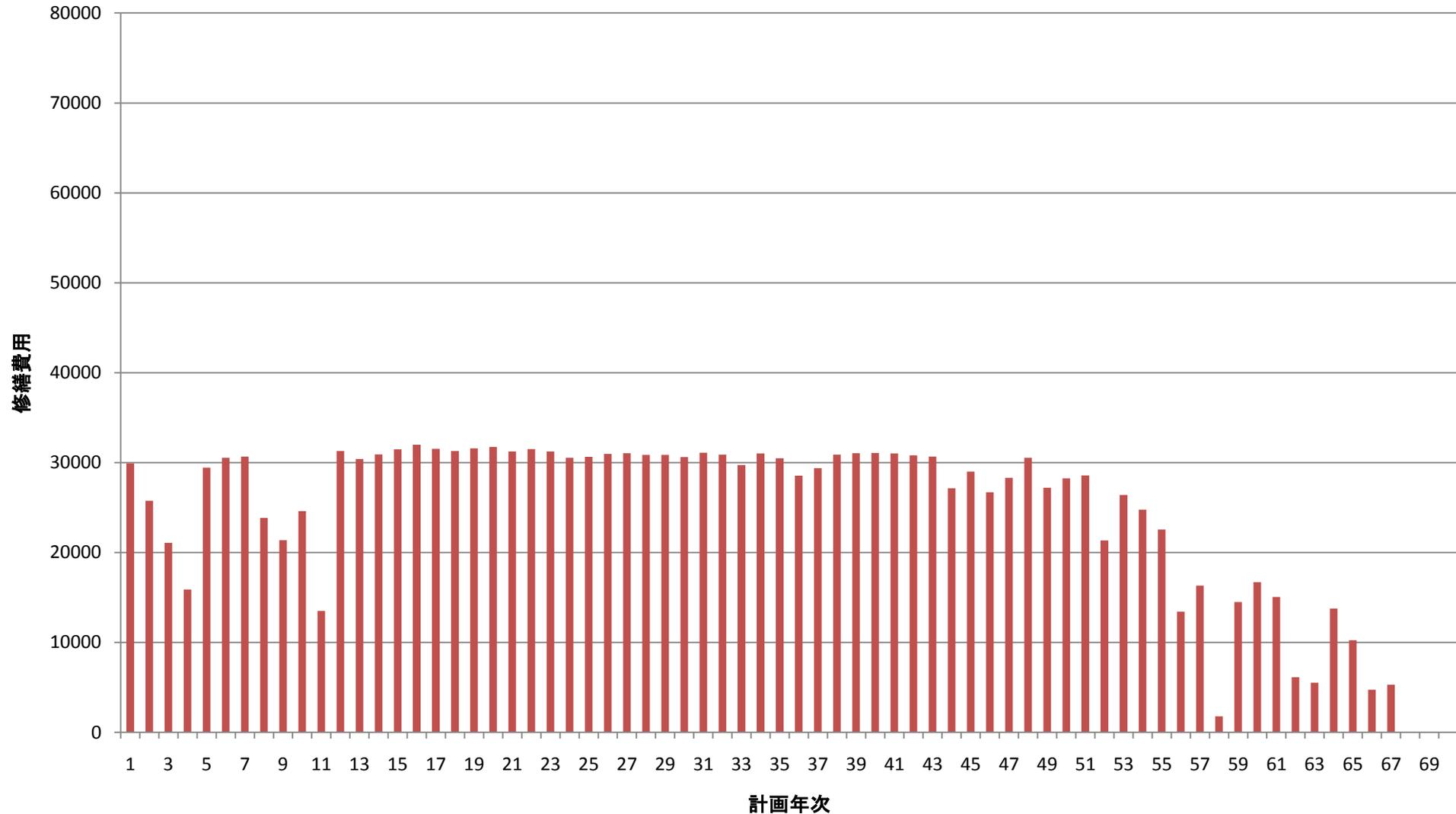
## 計画年次と修繕費用 (反復回数6)



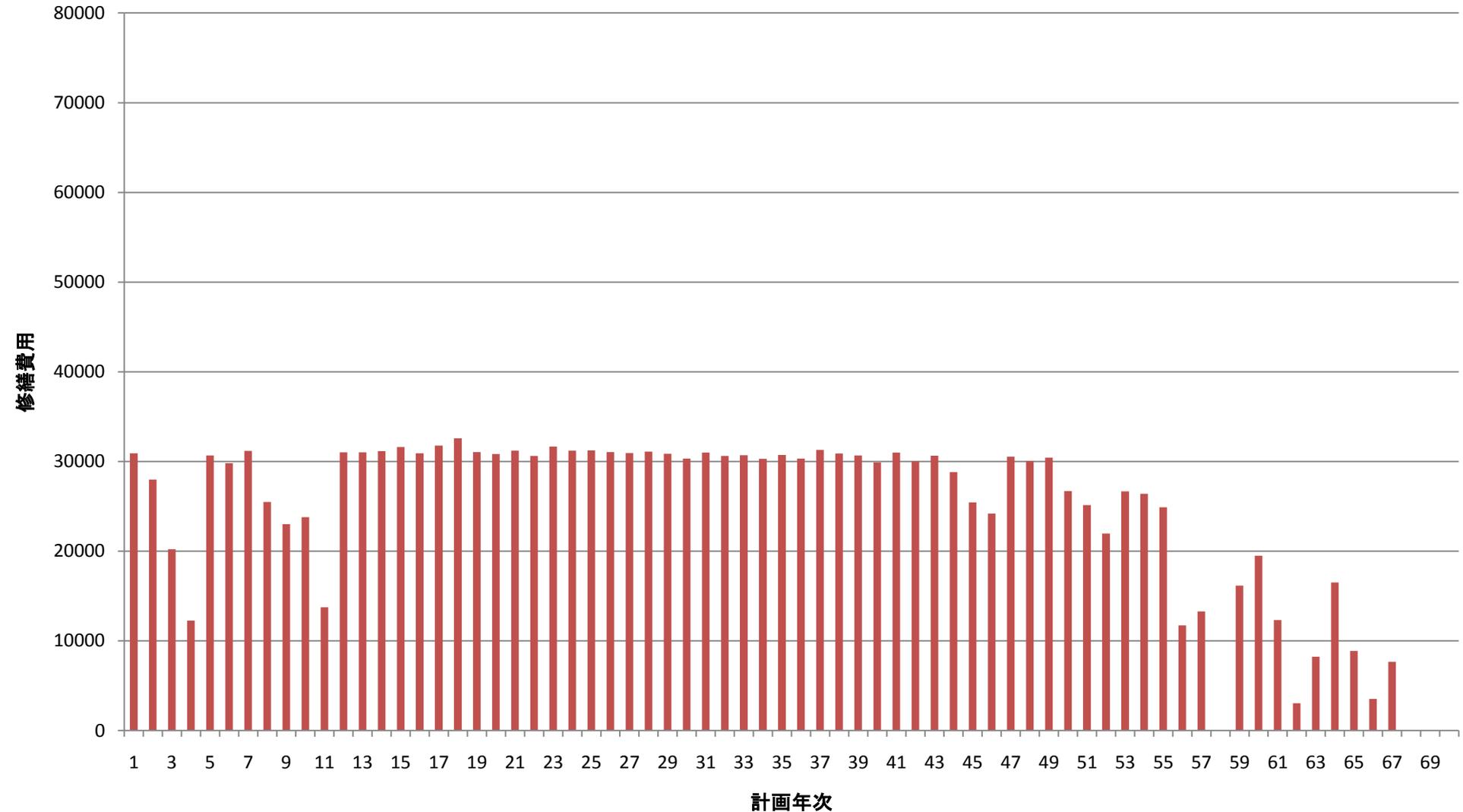
## 計画年次と修繕費用 (反復回数7)



## 計画年次と修繕費用 (反復回数8)



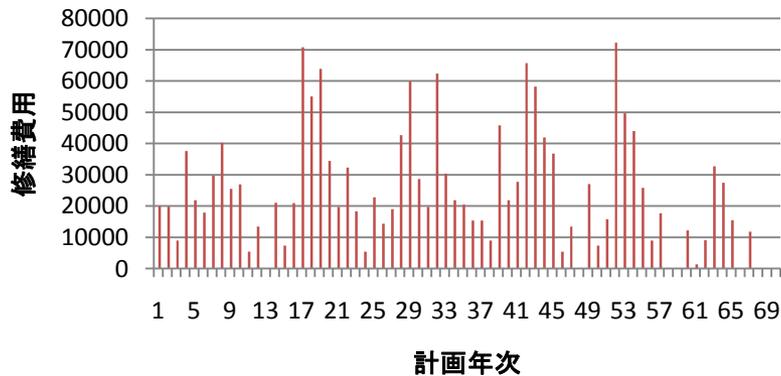
## 計画年次と修繕費用 (反復回数9)



# 最適化計算結果

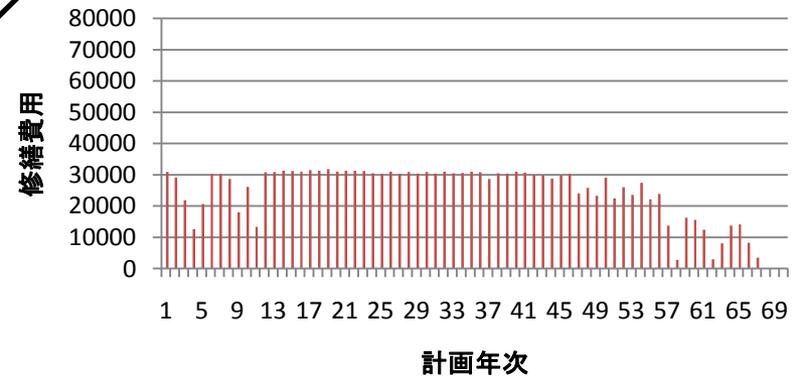
## -施設数200のケース-

計画年次と修繕費用  
(前倒し修繕無し)

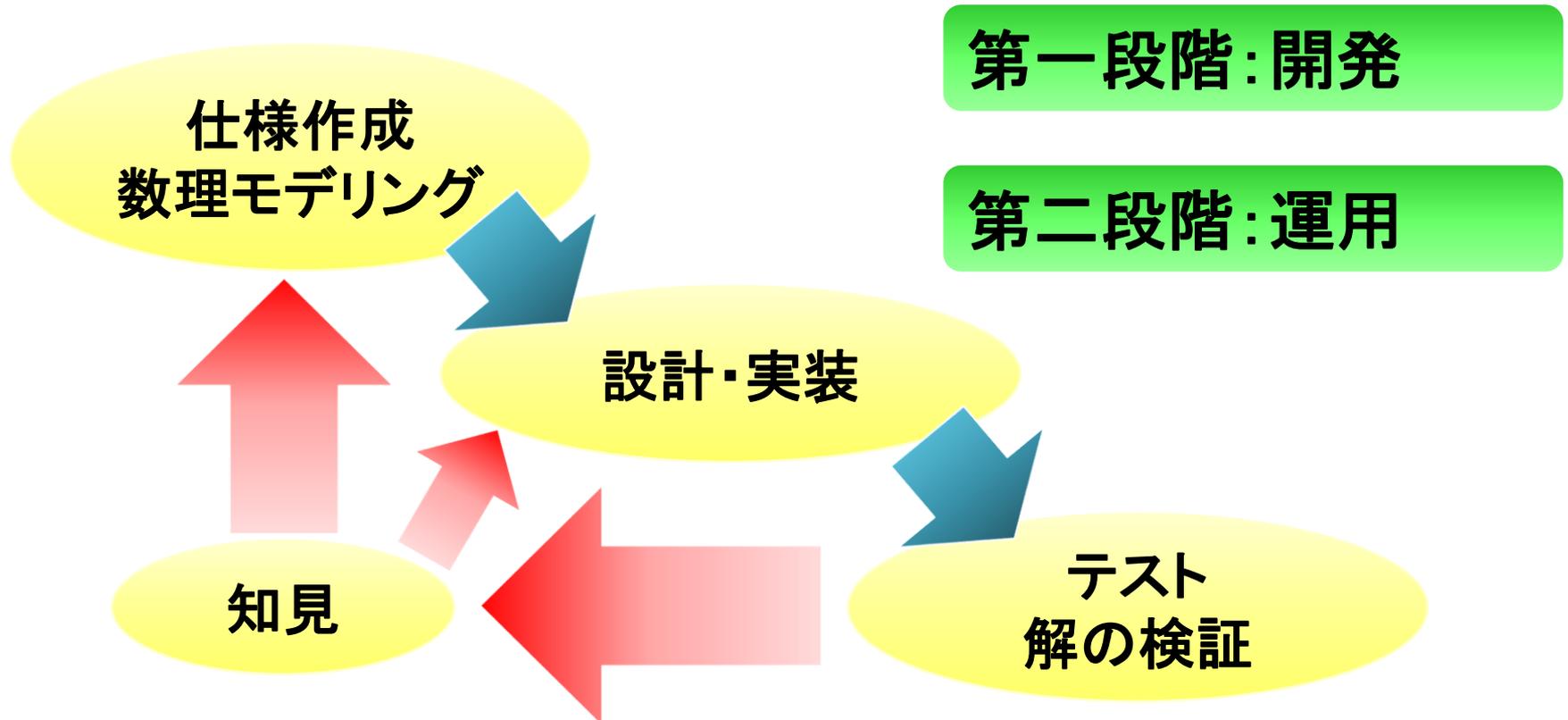


予算平準化

計画年次と修繕費用  
(3年以内まで前倒し修繕可)



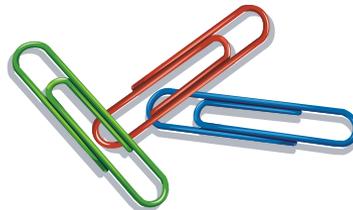
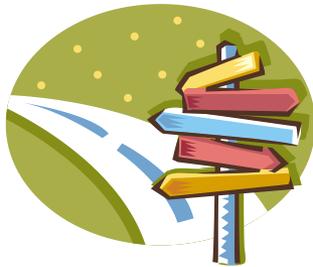
## システムの成長過程



# 最適化プロジェクトに向けて

問題の設定

インターフェースの選択



# 問題の設定-STEP1-

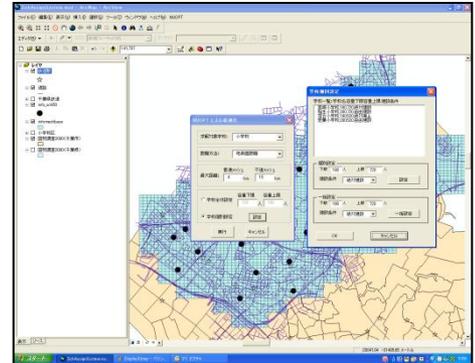
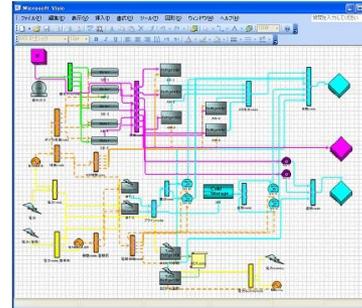
- 決定対象を明確にする
- 最適化の指標を定める
  
- 制約条件の洗い出し
  - ヒアリング
  - 制約条件の優先順位
- 💡 手元にあるデータは何か？  
計画は何の情報をもとに決定されるべきか？



# インターフェースの選択-STEP2-

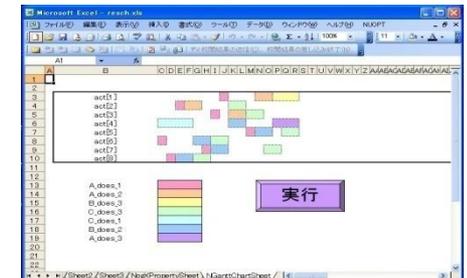
## 実行方法

- コマンドプロンプト
- エクセル
- その他インターフェース

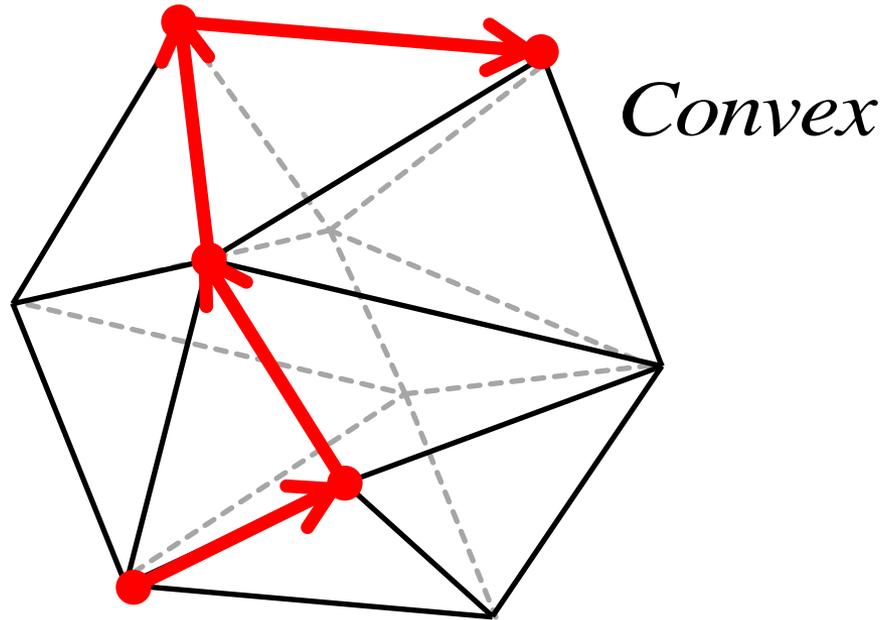


## 入出力データの形式

- csvファイル/datファイル
- Excelワークシート/グラフシート



# 数理計画の技術を是非ご利用ください



$$\min f(x) \quad s.t. \quad Ax \leq b \quad \& \quad x \in \mathbb{R}^n$$

# 最適化システム導入プロジェクト -土木構造物保守計画支援システム-

数理計画部(Optimization Division)

研究員 岩永二郎

[iwanaga@msi.co.jp](mailto:iwanaga@msi.co.jp)