複数の巡査による
指定地点の警邏について

能城秀彬（東京大学）

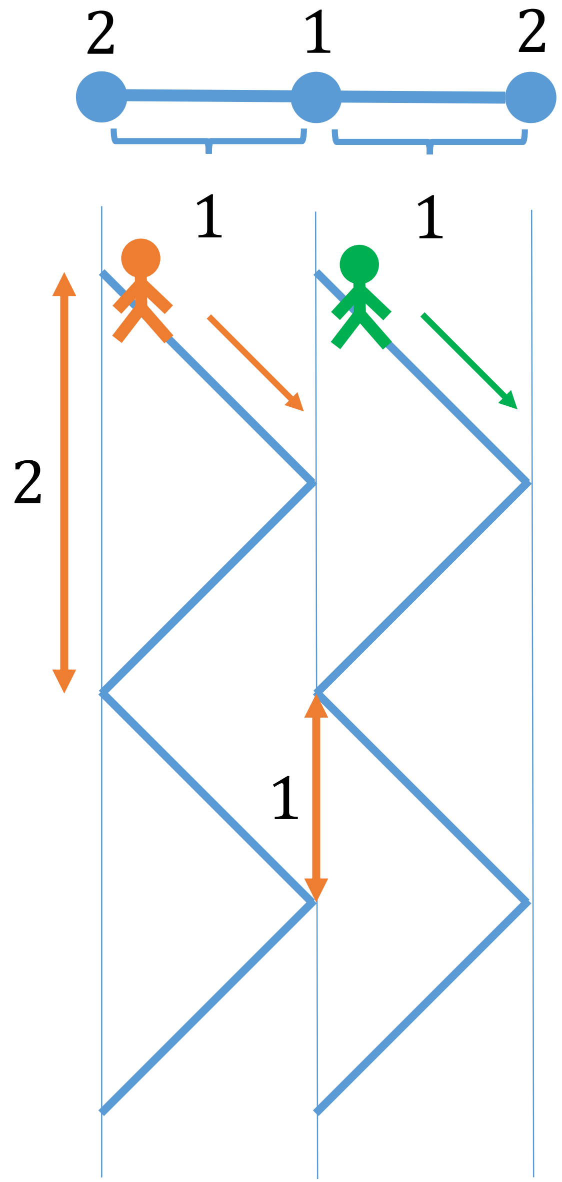
警邏（けいら）

- 警邏(patrolling)とは
  - 1人または複数の巡査により
  - 領域内のあらゆる場所を<u>十分な頻度</u>で訪問すること
- 警邏する領域の例
  - 二次元の領域
  - 線分や閉路などの全体
  - グラフの頂点

今回はグラフの頂点の警邏を考える

問題設定 – “放置可能時間”

- 頂点を警備するのに必要な
  訪問の頻度を定める
- 連続した2回の訪問時刻の差として
  許される最大値
- 訪問とは点で表される巡査が
  頂点を踏むこと
- 頂点を警備するには,
  放置可能時間を満たしながら
  訪問し続けなければならない
  （警備の定義）



問題設定

辺の長さ

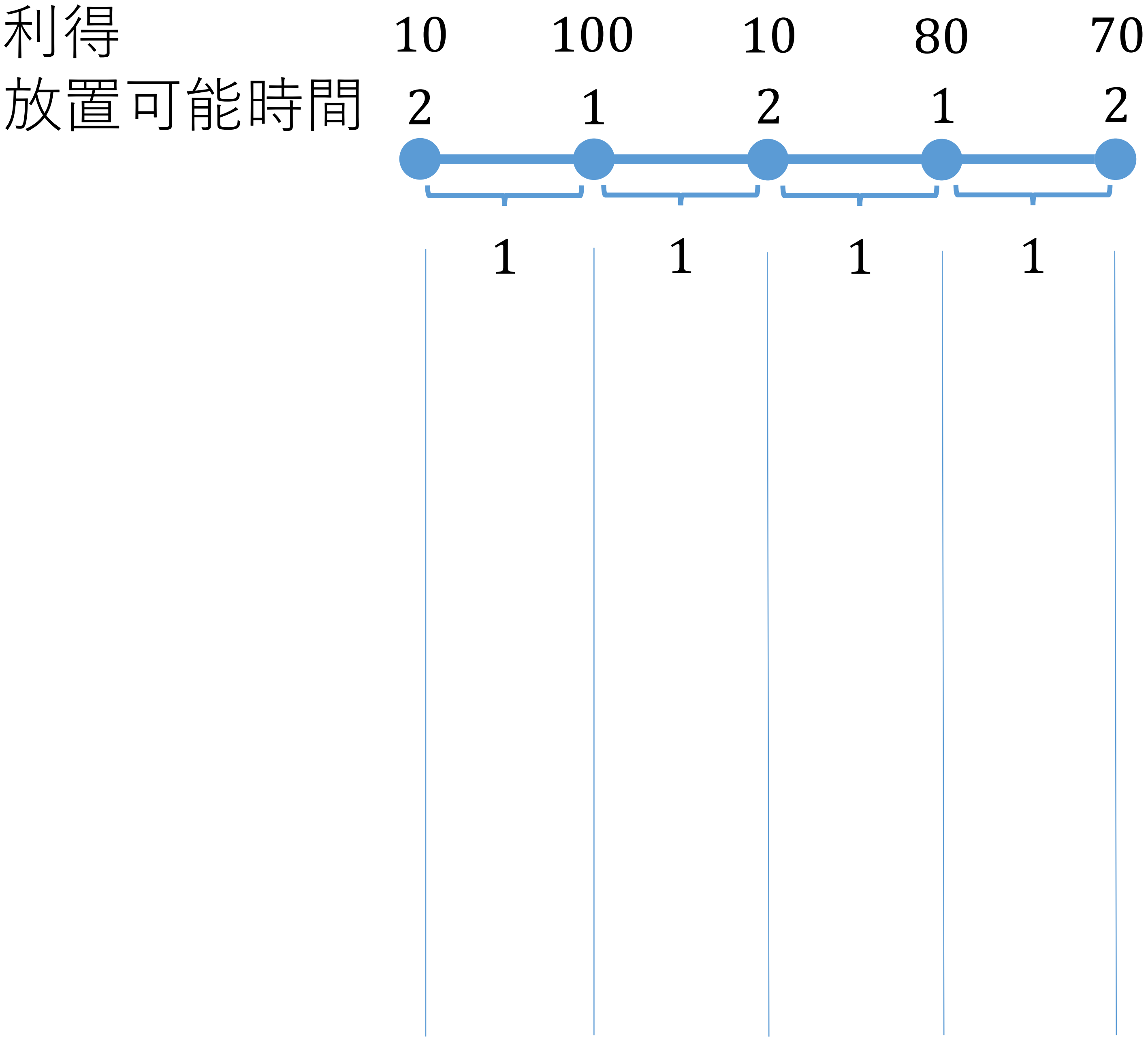
- 入力
  - 無向グラフ $G = (V, E, d)$ （警備する対象）
  - 各頂点の放置可能時間
  - 巡査の人数（どの巡査も速さ1以下で動く）
- 目的
  - DecisionPP： 全頂点を警備できるかどうかを判定
  - OptimizePP： 各頂点の利得も入力として与える．
    警備できる頂点部分集合のうち，
    利得の合計が最大のものを求める

DecisionPPの一般化

この2つの問題についてそれぞれ計算量を調べる

例

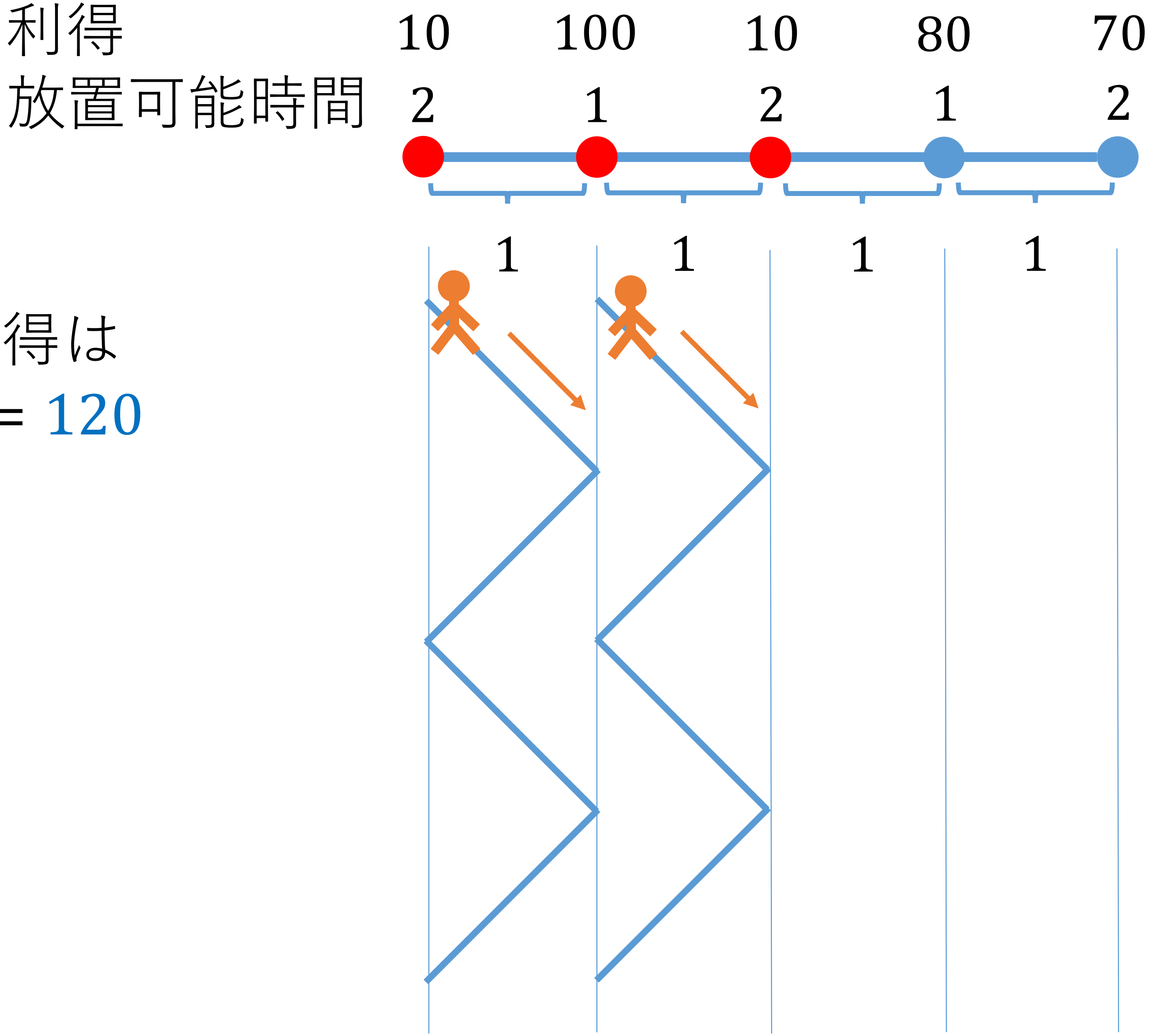
- 巡査が2人



例

- 巡査が2人
- 青の動きを選ぶと利得は

$$10 + 100 + 10 = 120$$

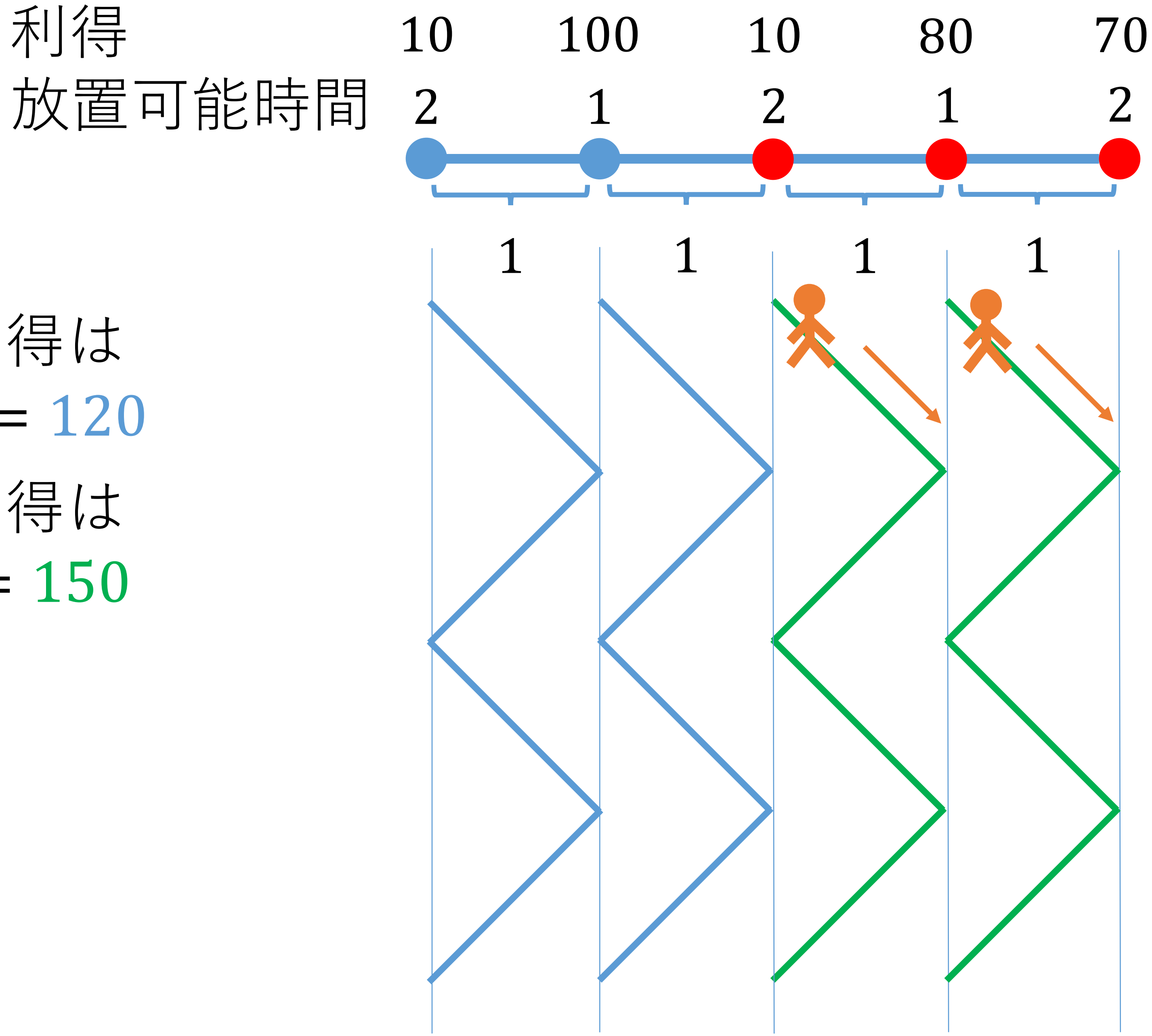


例

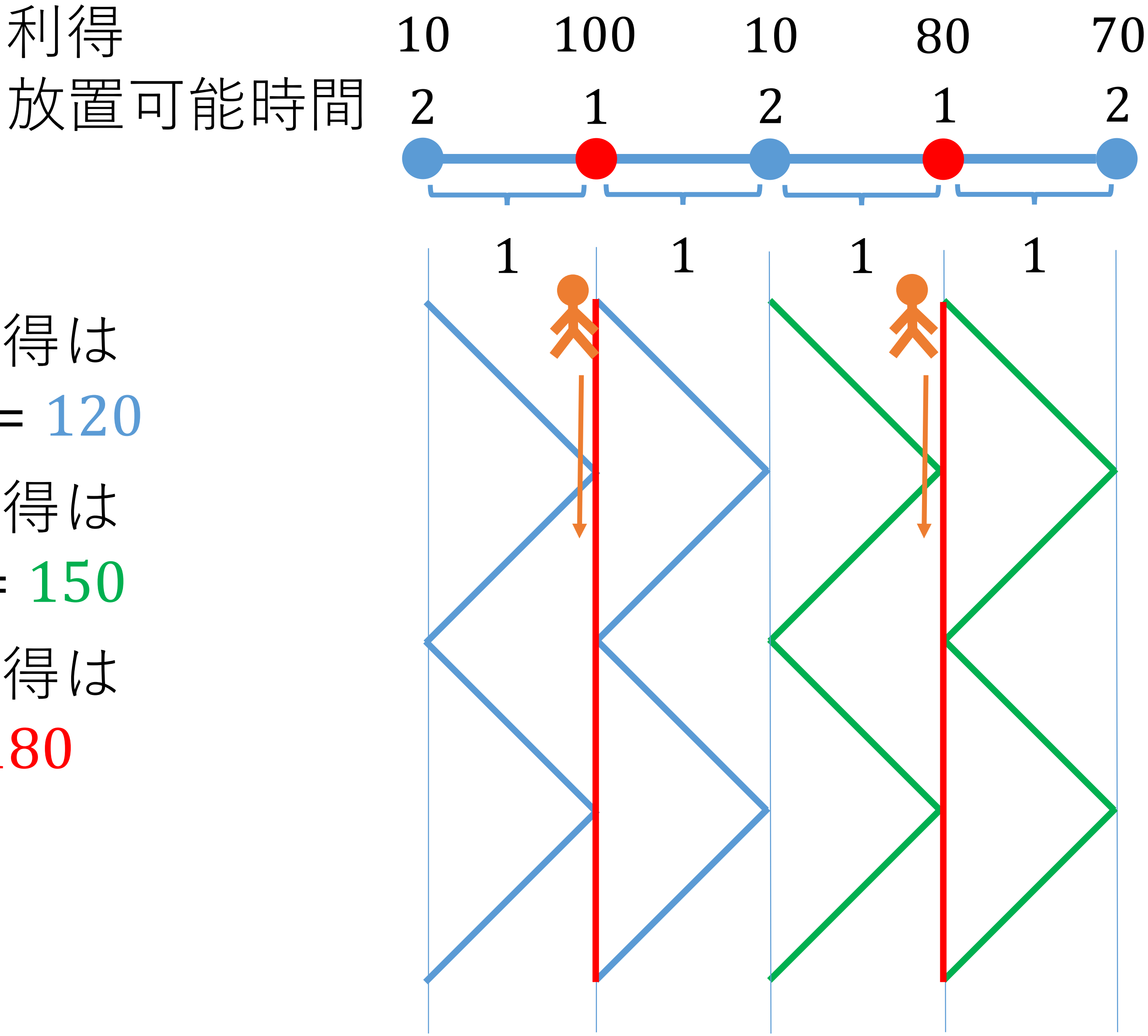
- 巡査が2人
- 青の動きを選ぶと利得は

  $10 + 100 + 10 = 120$
- 緑の動きを選ぶと利得は

  $10 + 80 + 70 = 150$



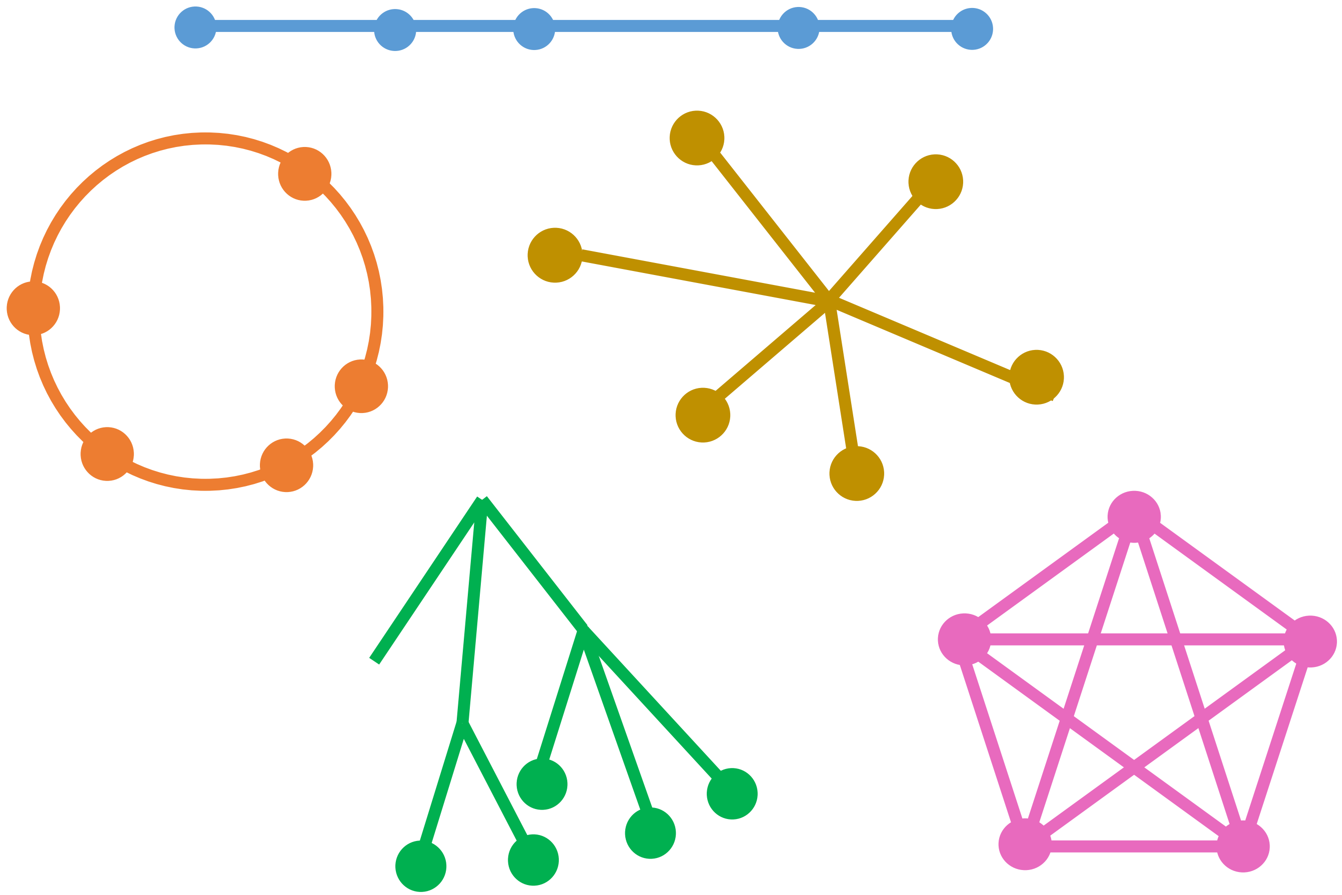
例



- 巡査が2人
- 青の動き を選ぶと利得は
  $10 + 100 + 10 = 120$
- 緑の動き を選ぶと利得は
  $10 + 80 + 70 = 150$
- 赤の動き を選ぶと利得は
  $100 + 80 = 180$

同じ問題設定の先行研究[1]

- Line（線分）
- Circle（閉路）
- Star（星）
- Tree（木）
- 完全グラフ



[1] : S. Coene, F.C.R. Spieksma, and G.J. Woeginger. (2011). Charlemagne's challenge: the periodic latency problem. *Operations Research*, 59(3), pp. 674–683.

同じ問題設定の先行研究[1]

- Line（線分）
- Circle（閉路）

巡査が1人の場合はOptimizePPに
多項式時間アルゴリズムあり

- Star（星）
- Tree（木）

特別な場合を除きNP困難

- 完全グラフ

→ NP困難

巡査が1人で,
全頂点の利得・放置可能時間が
全て等しいならばOptimizePPに
多項式時間アルゴリズムあり

辺の長さを十分大きくすれば
実質使えない辺を作れるので
一般のグラフを表せる

[1] : S. Coene, F.C.R. Spieksma, and G.J. Woeginger. (2011). Charlemagne's challenge: the periodic latency problem. *Operations Research*, 59(3), pp. 674–683.

同じ問題設定の先行研究[1]

- Line（線分）
- Circle（閉路）

巡査が複数の場合は未解決

- Star（星）
- Tree（木）

特別な場合を除きNP困難

より単純な図形ならどうか？

- 完全グラフ

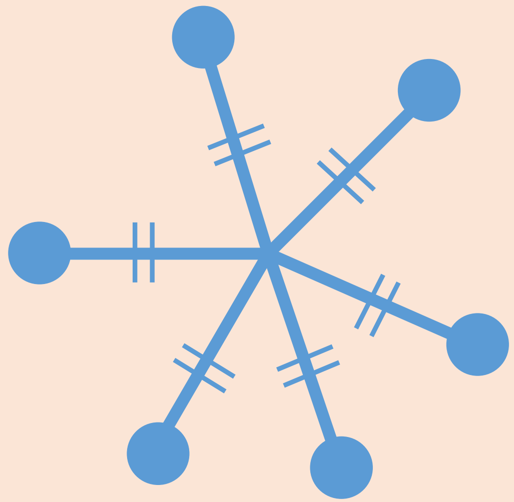
→ NP困難

[1] : S. Coene, F.C.R. Spieksma, and G.J. Woeginger. (2011). Charlemagne's challenge: the periodic latency problem. *Operations Research,* 59(3), pp. 674–683.

今回扱う図形

- Line
  巡査が複数の場合のみ調べる

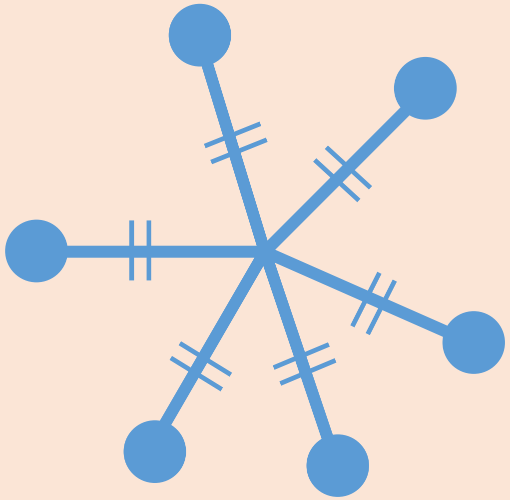
- UStar（星で辺の長さが全て等しい場合）



今回扱う図形

- Line
  巡査が複数の場合のみ調べる

- UStar（星で辺の長さが全て等しい場合）



Lineの場合の概要

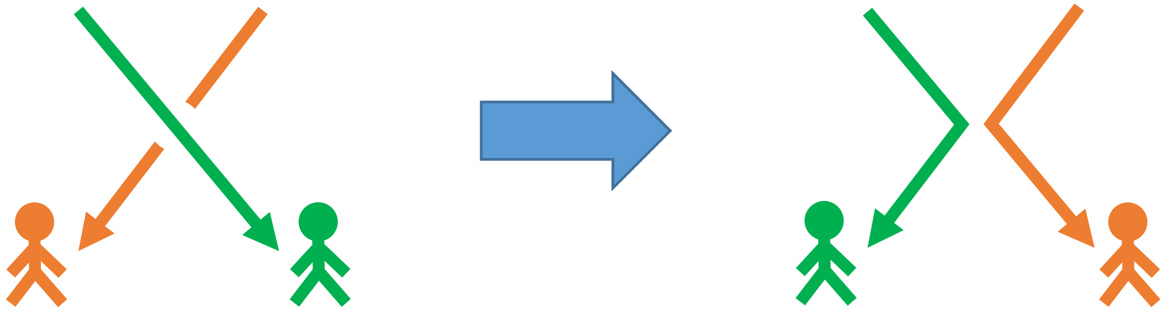
- 巡査が1人の場合（既知）
  → OptimizePPに多項式時間アルゴリズムあり
- 巡査が複数の場合（本研究）
  - 放置可能時間が全て同じであるとき
    → OptimizePPに多項式時間アルゴリズムあり
  - 放置可能時間が一般のとき → 未解決
    - 複雑な動きの例
    - 別の問題設定について

Lineの場合の概要

- 巡査が1人の場合（既知）
  → OptimizePPに多項式時間アルゴリズムあり
- 巡査が複数の場合（本研究）
  - 放置可能時間が全て同じであるとき
    → OptimizePPに多項式時間アルゴリズムあり
  - 放置可能時間が一般のとき → 未解決
    - 複雑な動きの例
    - 別の問題設定について

Line：巡査の位置関係について

- 巡査は線分上を右か左に動く（か停止）
- 巡査の能力は全員同じなので,
  すれ違う代わりに互いに引き返してもよい



→ 巡査は初期配置の順番を保って動くとしてよい

Line：巡査が複数，放置可能時間が全て同じとき

定理1

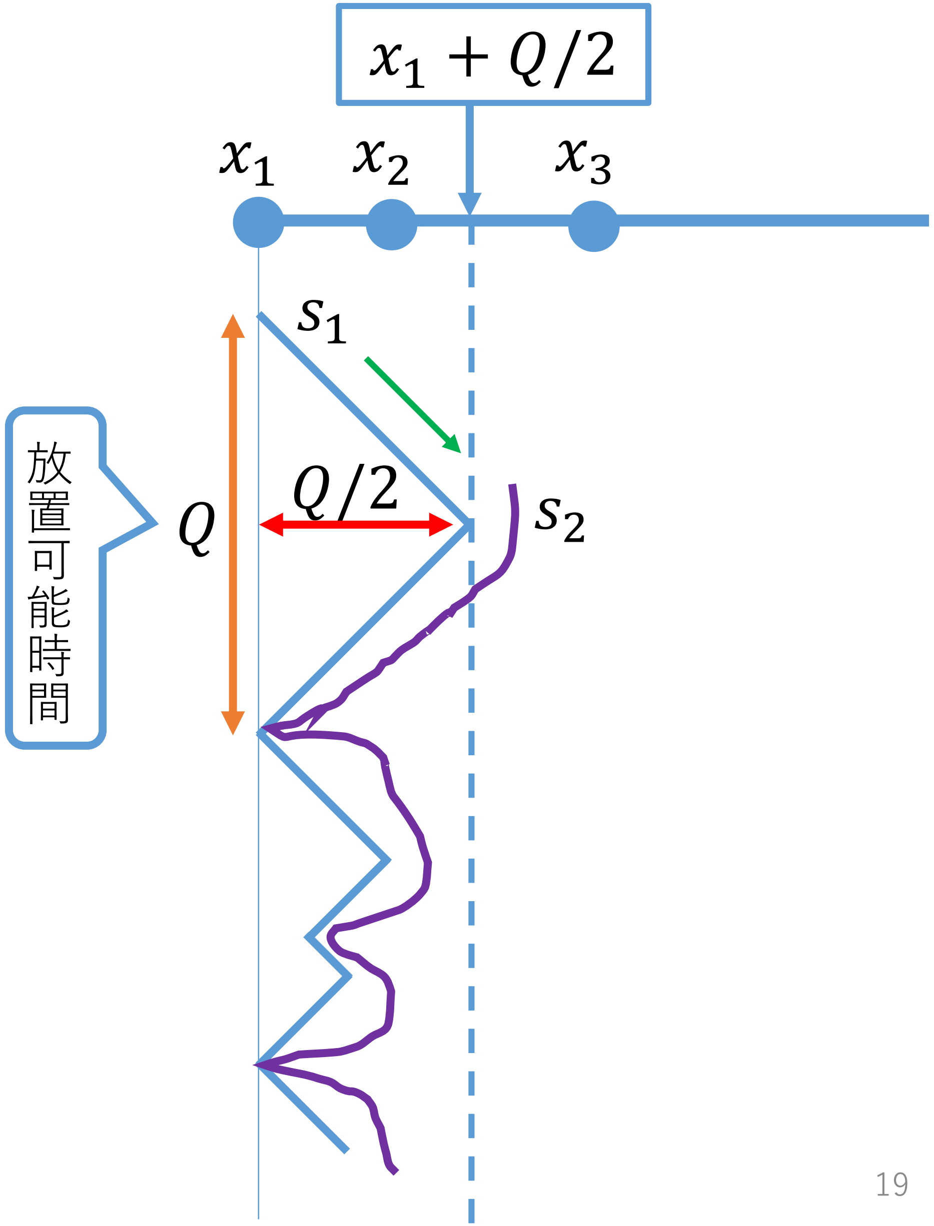
Lineで放置可能時間が全て等しい場合，巡査が複数でもOptimizePPに多項式時間アルゴリズムが存在する．

定理1の証明手順

1. 任意の実行可能解の変換
   頂点部分集合 $V_s \subset V$ を警備できる巡査の動き方が存在するならば，ある特別な動き方（後述）でも V_s を警備できることを示す
2. 特別な動き方のなかで最適解を求める
   このような動き方での最適解が存在するので
   それを探す多項式時間アルゴリズムを示す

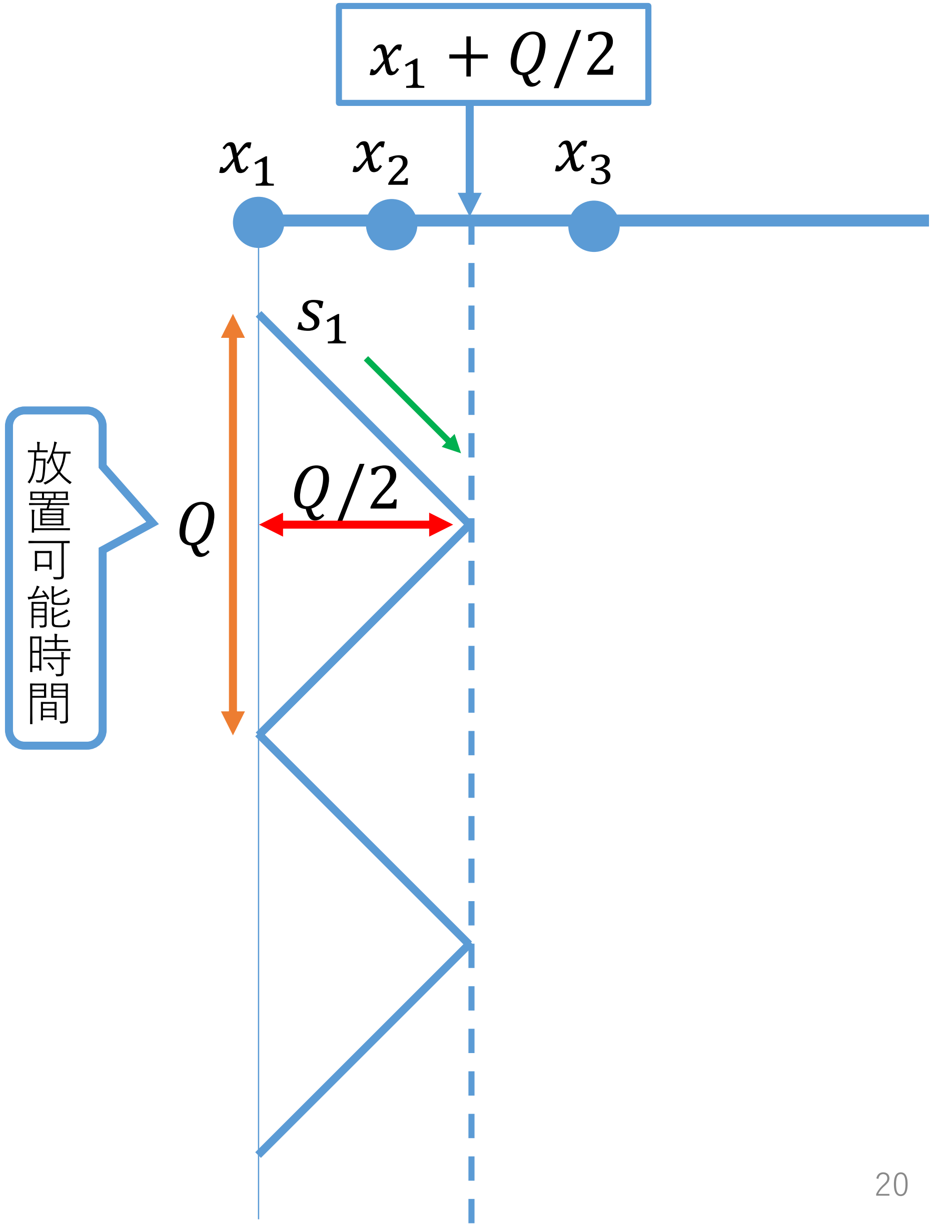
定理1の証明：step 1

- ある頂点部分集合 V_S を警備できる巡査の動きがあったとする
- V_S の点の座標を $x_1 \leq x_2 \leq \cdots \leq x_k$ とする
- 初期順序を保つとき，最も左の巡査 s_1 以外が x_1 にいるならば s_1 も x_1 にいる
  → x_1 はs_1 のみにより警備される



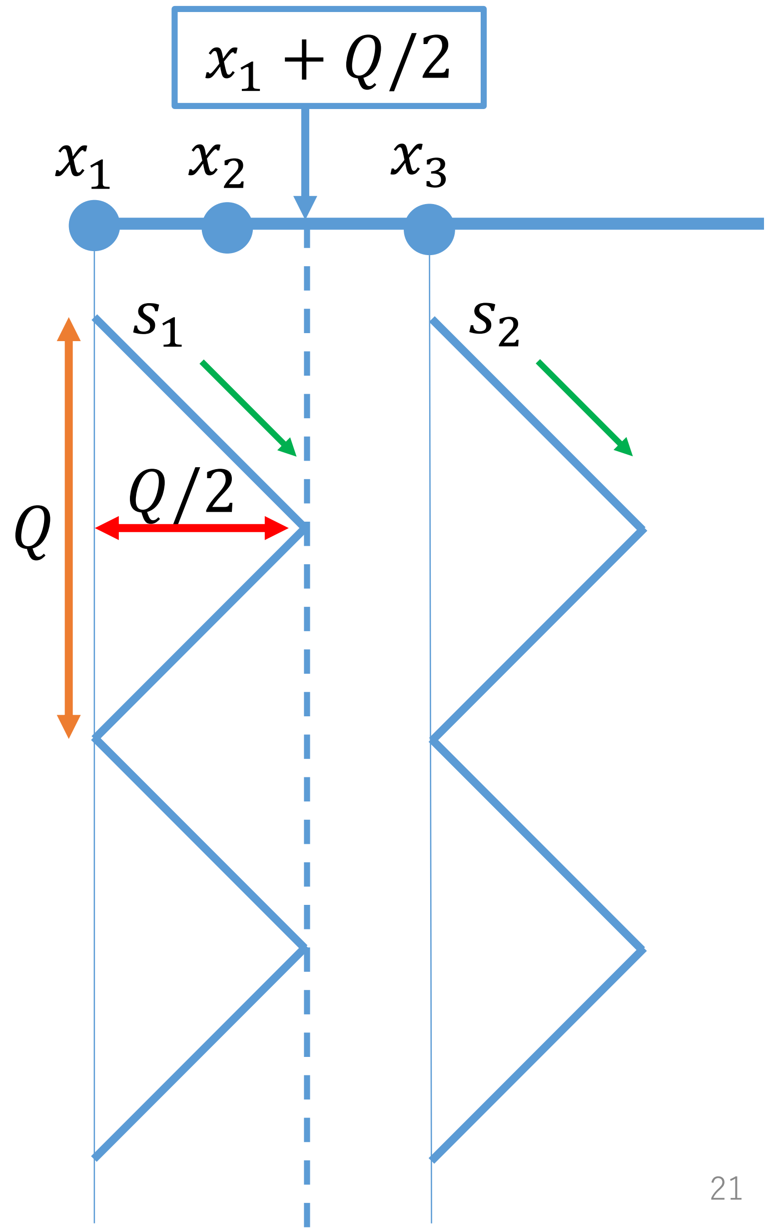
定理1の証明：step 1

- s_1 は $x_1 + Q/2$ までしか動けない
- s_1 は $[x_1, x_1 + Q/2]$ を往復すれば, この区間に含まれる全点を警備できる
- これ以上は警備できないのでこれが最適



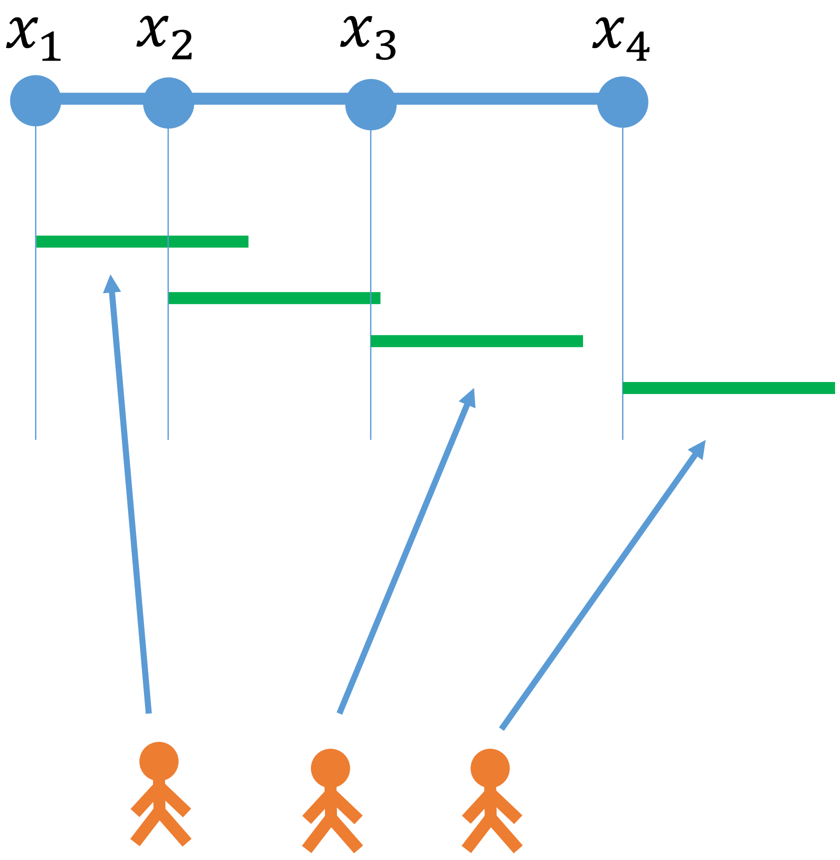
定理1の証明：step 1

- $x_1 + Q/2$ より右側は元々 s_1 以外の巡査達により警備可能
- 残りの点と巡査で同じ変換を繰り返す
- 変換後の動き
  $= m$ 人の巡査それぞれが，n 個の区間
  $$\left[x_1, x_1 + \frac{Q}{2}\right], \ldots, \left[x_n, x_n + \frac{Q}{2}\right]$$
  のうち互いに交わりのない m 個以下の区間を1つずつ担当し往復する動き



定理1の証明：step 2

- n個の利得付きの区間から，利得の合計が最大となる交わりの無い$m(< n)$個の区間を選べばよい
  - あとはそれらの区間を巡査が往復するだけ
- 動的計画法により$O(n \log n + nm)$ で計算できる（省略）



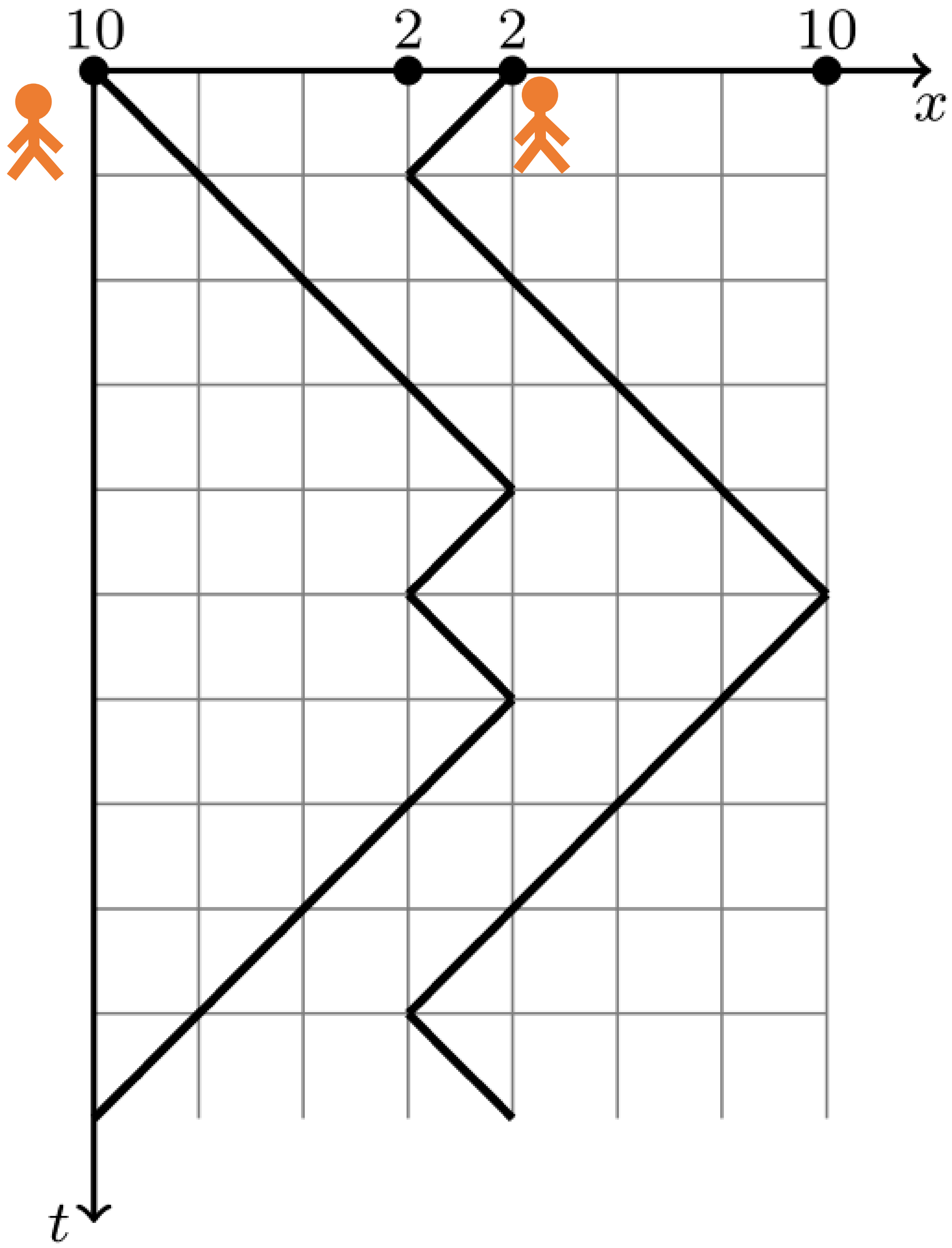
Lineの場合の概要

- 巡査が1人の場合（既知）
  → OptimizePPに多項式時間アルゴリズムあり
- 巡査が複数の場合（本研究）
  - 放置可能時間が全て同じであるとき
    → OptimizePPに多項式時間アルゴリズムあり
  - 放置可能時間が一般のとき → 未解決
    - 複雑な動きの例
    - 別の問題設定について

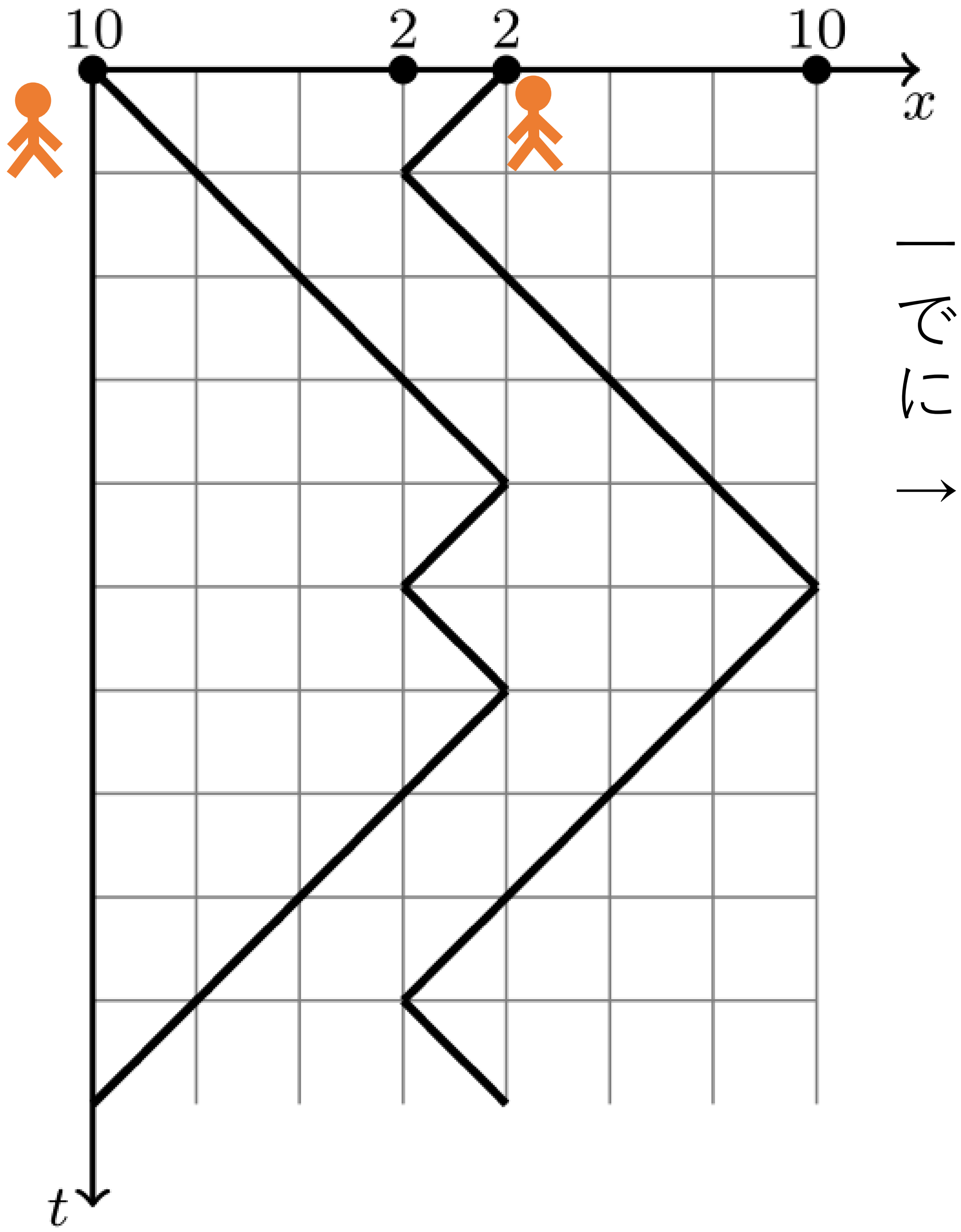
Line：巡査が複数，放置可能時間が一般のとき

- 放置可能時間が全て同じならば，互いに交わりのない区間を往復する単純な動きのみ考えればよかった
- 放置可能時間が一般の場合は，そうでない動きが最適となる例が存在

放置可能時間

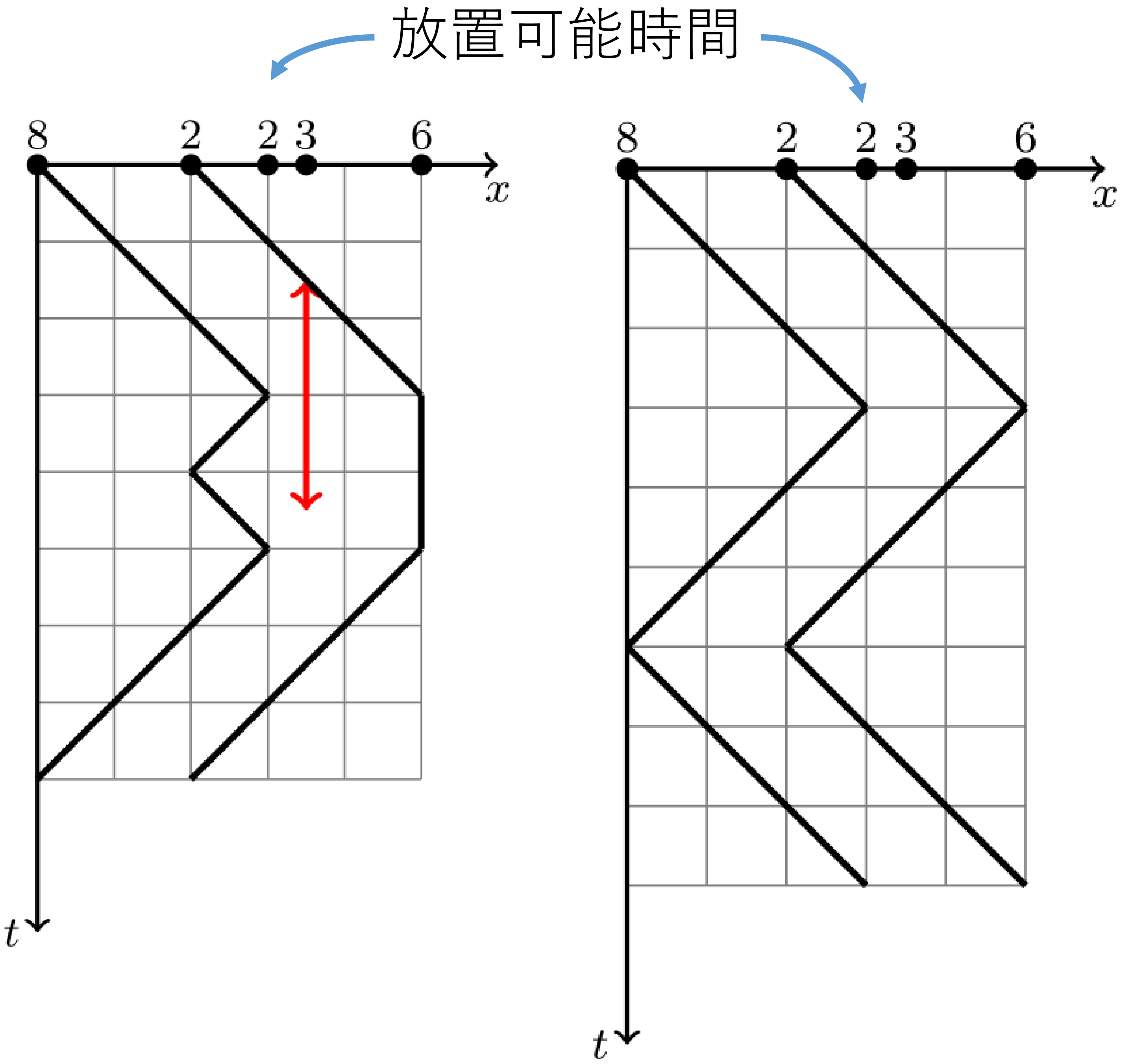


放置可能時間



一番左の巡査は
できるだけ右に手伝い
に行ってよい？

→ No.



一番左の巡査はできるだけ右に手伝いに行ってよい？

→ No.

あえて早めに戻ると
協力しやすくなることもある

→「あえて早めに戻る」が許されない問題設定にしたらどうか？

別の問題設定

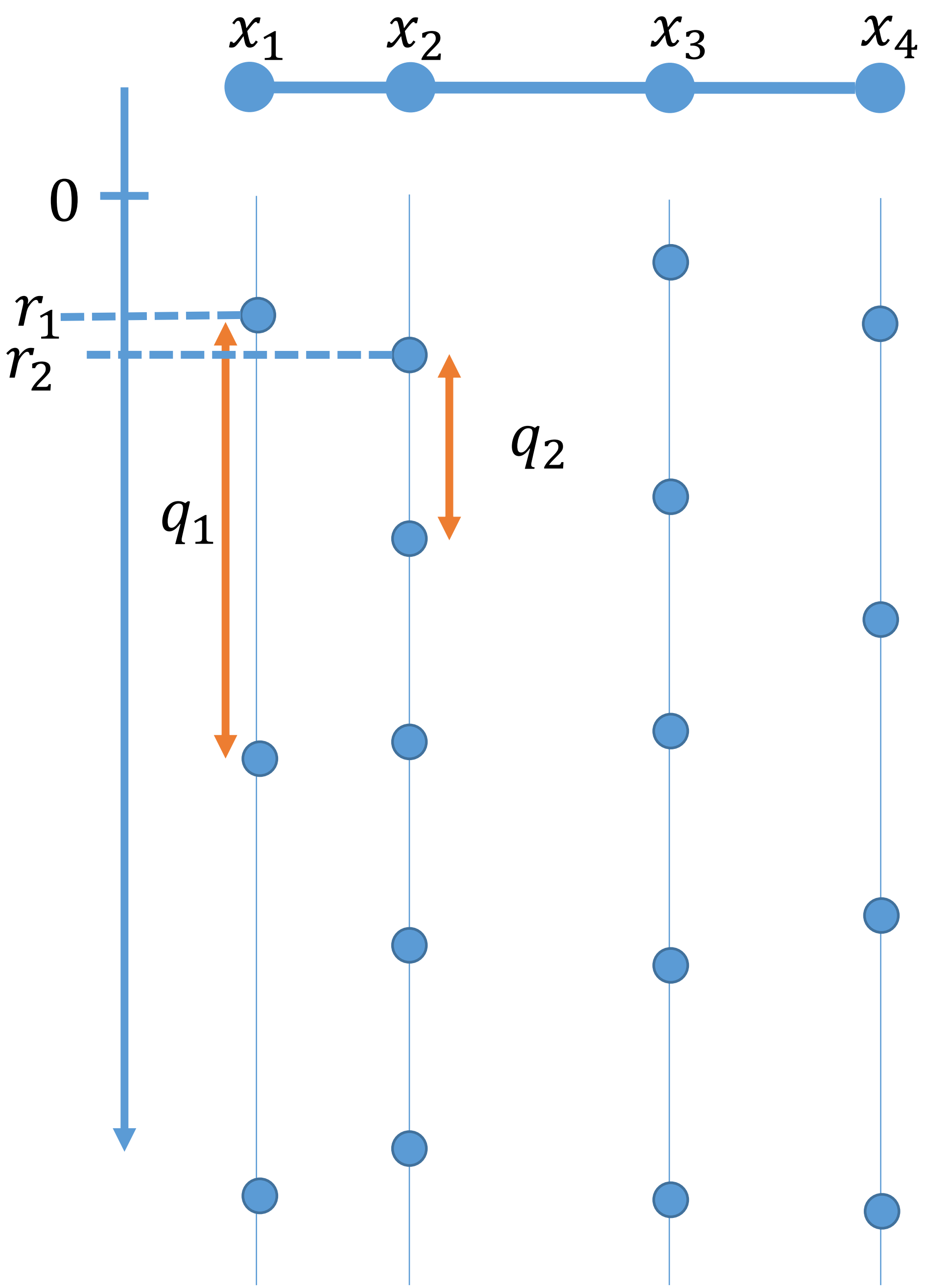
- 放置可能時間　… ある訪問後この時間<u>以内</u>にまた訪問

↓

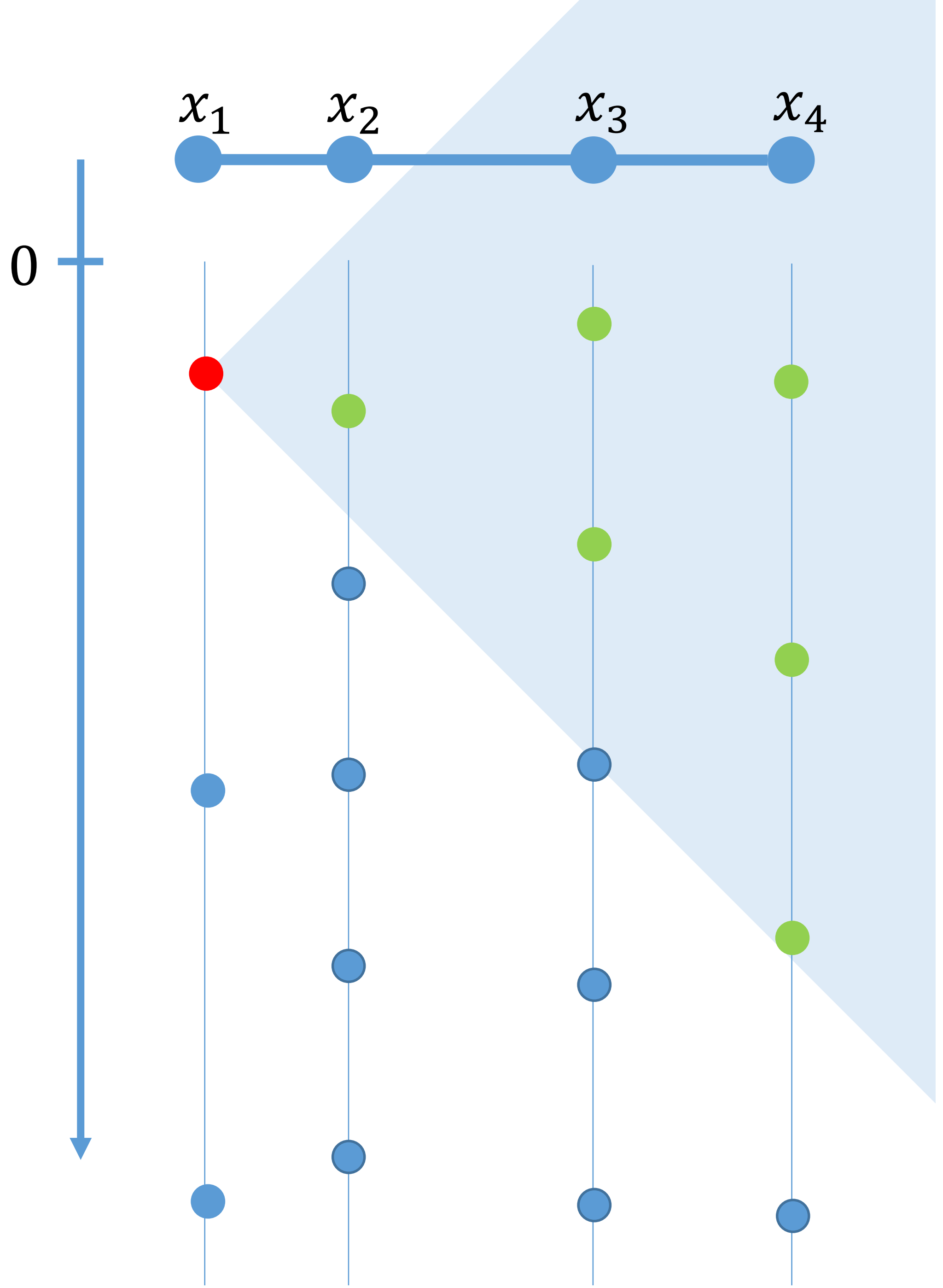
- 周期　… ある訪問後この時間<u>ちょうど</u>にまた訪問
  - 先ほどの例のように「あえて早めに戻る」ができないように

  - さらに最初の訪問時刻も指定
    - できるだけ右に手伝いに行く戦略が最適に
      - ただし全点の警備の場合のみ適用できるので使えるのは DecisionPP だけ

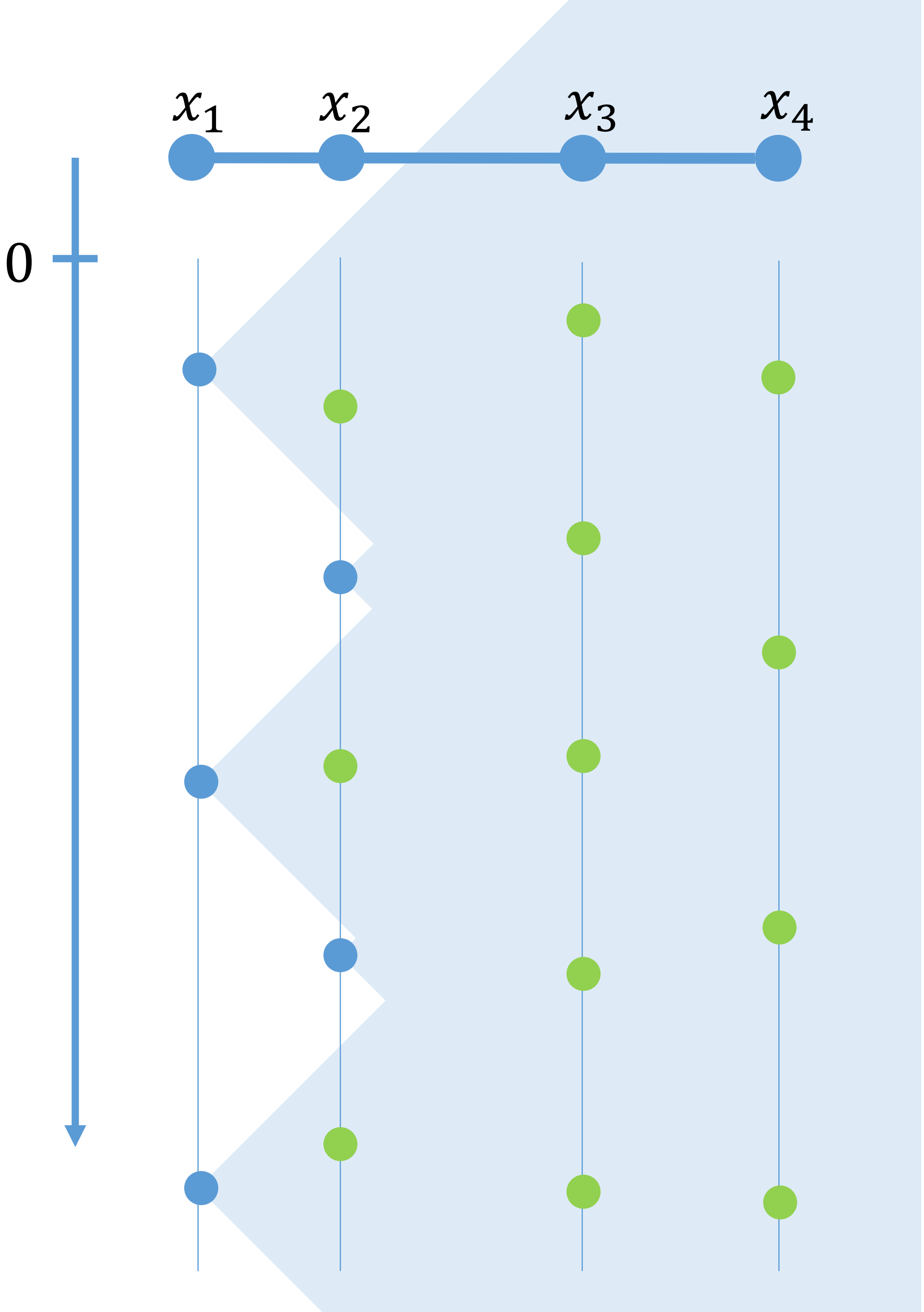
- $t-x$ 平面に訪問すべき時刻と位置の組 (t,x) を表す点が全て与えられる



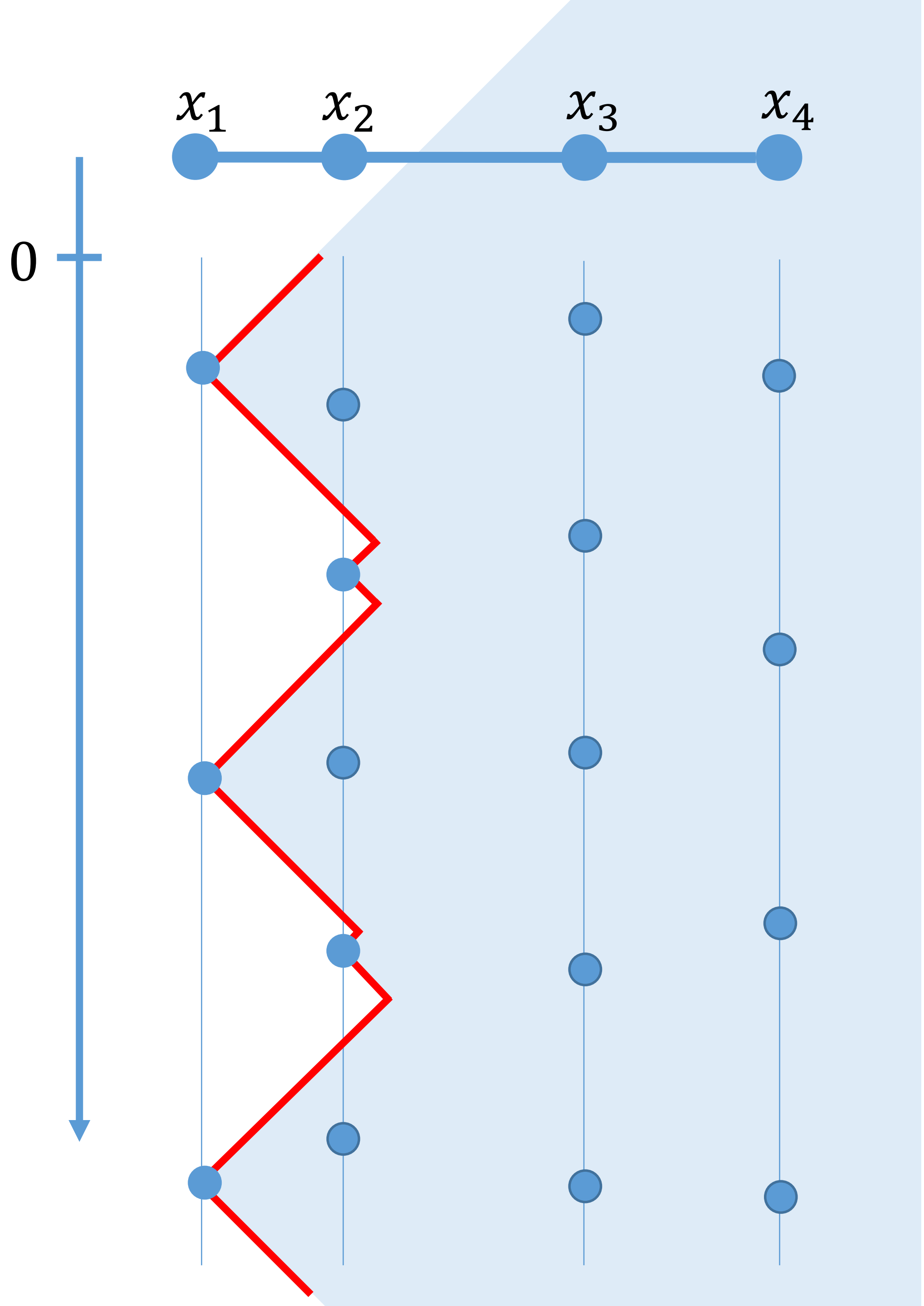
- 巡査 s_1 がある点を訪問する→その右に広がる直角三角形の領域（境界含まず）に含まれる点は訪問することができない



- 平面上の全ての点で
このような直角三角形領域の
和集合をとる
- この領域に含まれる点は
s_1が訪問できない
（∵ 訪問すると，
s_1 より左に別の巡査が必要
になり， s_1 が最も左でなく
なる）



- その境界を s_1 が動けばよい
  → 訪問できない点以外を
  全て訪問できているので
  これが最適
- できるだけ右に行く戦略が最適になっている
- あとは s_1 が訪問した点を除いてこれを繰り返せばよい
- 必要な巡査の数が分かる



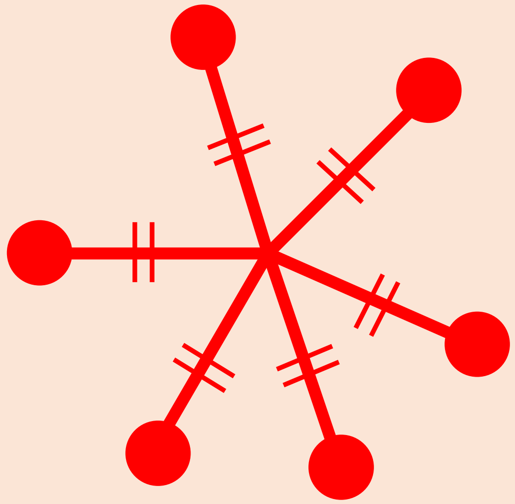
Lineの場合（巡査複数）のまとめ

- 放置可能時間が全て同じであるとき
  → 多項式時間アルゴリズムあり($O(n\log n + nm)$)
- 放置可能時間が一般のとき → 未解決
  - 複雑な動きが最適となる例が存在
  - 別の問題設定 …
    放置可能時間のかわりに周期と最初の訪問時刻を与える
    → DecisionPP ならばなるべく右に行く戦略が最適に
  - 最初の訪問時刻は与えられない場合は未解決
    （全点警備できるように最初の訪問時刻を設定できるか）

今回扱う図形

- Line
  巡査が複数の場合のみ調べる

- UStar（星で辺の長さが全て等しい場合）



UStarの場合の概要

- Star（既知）
  - 巡査が1人で利得・放置可能時間が全て等しい
    →多項式時間アルゴリズムが存在
  - それ以外（いずれかが一般の場合）→ NP困難
- UStar（本研究）
  - 放置可能時間が全て等しい→ 多項式時間アルゴリズム存在
  - 放置可能時間が一般の場合とき → 未解決
    - 別の問題設定を2つ考える

UStar：放置可能時間が全て同じとき

定理2

UStarで放置可能時間が全て等しい場合，巡査が複数でもOptimizePPに多項式時間アルゴリズムが存在する．

- UStarの枝の長さを d,
- 放置可能時間を Q,
- 巡査を m 人

とすると，利得の大きいものから $\left\lfloor \frac{mQ}{2d} \right\rfloor$ 個の頂点を警備できる

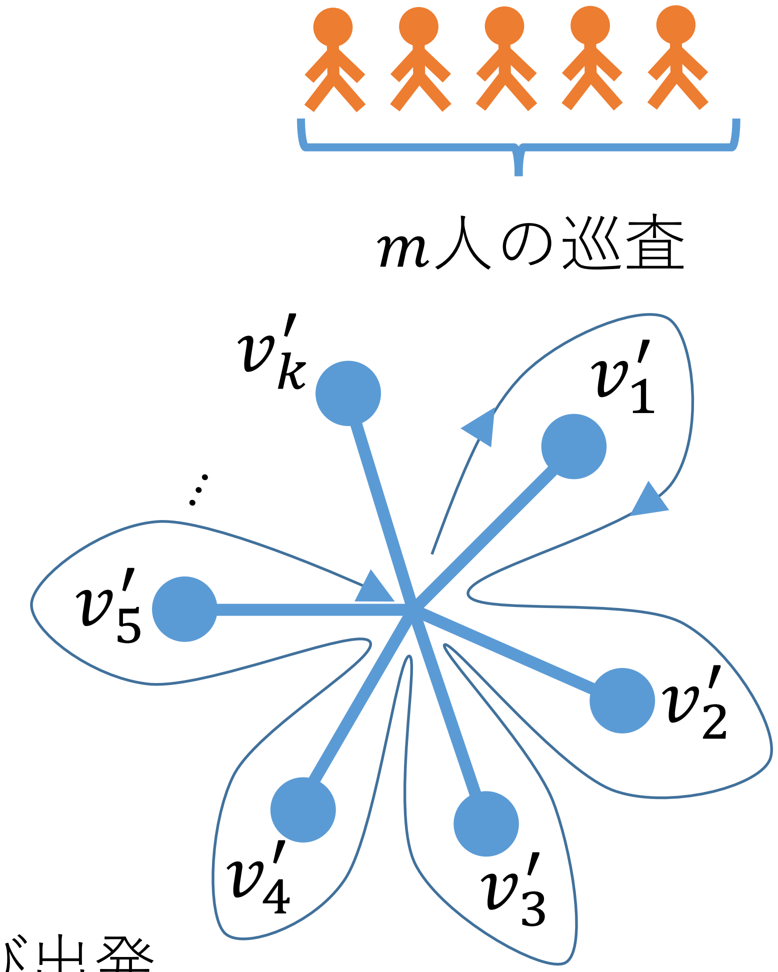
定理2の証明手順

1. $\left\lfloor \frac{mQ}{2d} \right\rfloor$個より多くの頂点は警備できないことを示す（省略）
2. $\left\lfloor \frac{mQ}{2d} \right\rfloor$個の頂点を警備できる巡査の動き方を示す（最適解）
   - UStarで放置可能時間が全て等しいので
     頂点を訪問するコストはどれも同じ
   - 利得の大きいものから$\left\lfloor \frac{mQ}{2d} \right\rfloor$個を選べばよい

定理2の証明：step 2

$\left\lfloor \frac{mQ}{2d} \right\rfloor$個の頂点を警備できる巡査の動き方

- まず巡査 s_1 が時刻0に出発し
  $v_1' \to v_2' \to \cdots \to v_k'$ を順番に
  速さ1で動きながら訪問
- 時間 Q ずつ遅れて $s_2, s_3, \ldots$ も同様に動く
  - 1周にかかる時間は $2d \cdot \left\lfloor \frac{mQ}{2d} \right\rfloor \leq mQ$
  - 最後の巡査 s_m が出発してから時間 Q 後には
    s_1 は中心に戻っているので，この時刻に再び出発
  - 全ての頂点は時間 Q ごとに訪問されるので警備できている



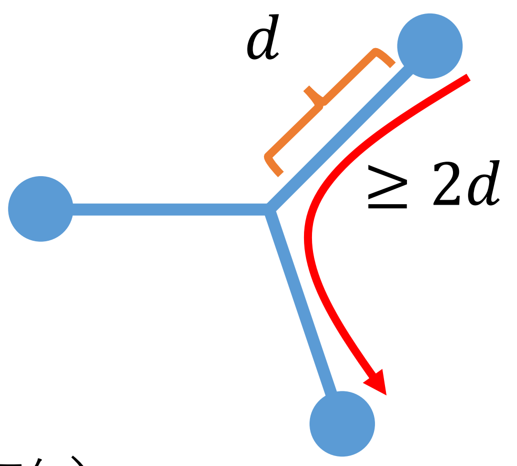
UStar：放置可能時間が一般のとき

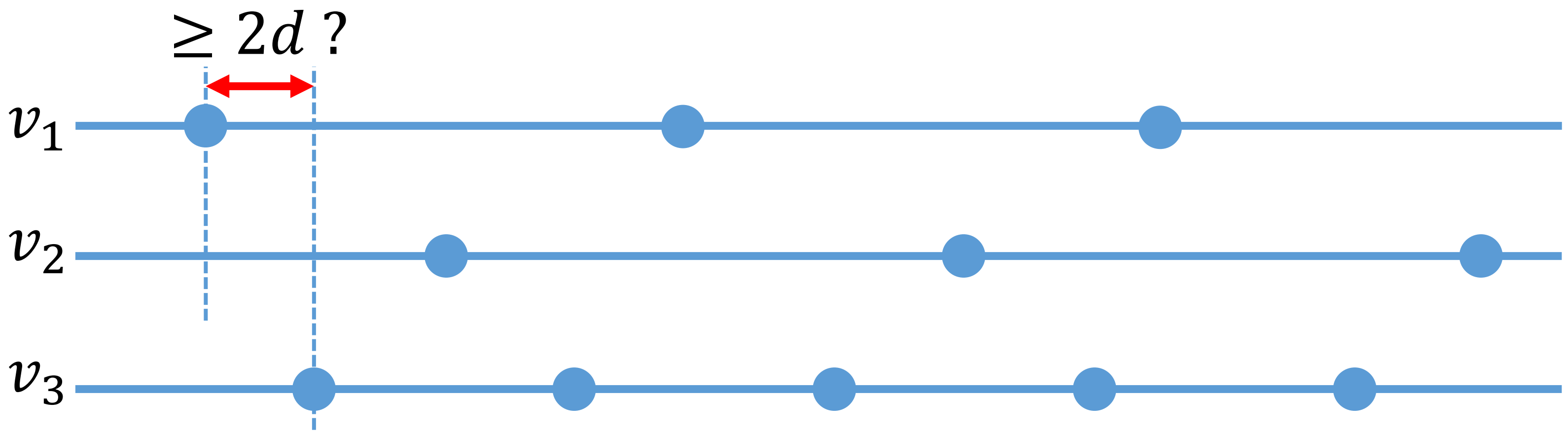
- 放置可能時間が全て等しい → 多項式時間アルゴリズムあり
- 放置可能時間が一般のとき → 未解決
  → ここでも，最初の訪問時刻と訪問の周期が与えられる問題を代わりに考えてみる
  - DecisionPP
    - 巡査が1人　→ 簡単に解ける
    - 巡査が複数 → 多項式時間でないアルゴリズムはある
  - OptimizePP
    - 巡査が1人で利得が全て等しくてもNP困難

UStar：最初の訪問時刻と訪問の周期

DecisionPP, 巡査1人

- 任意の異なる2頂点間の移動には
  最低 $2d$ の時間がかかる
- 訪問しなければならない時刻の間隔が
  $2d$ 以上になっているか調べればよい
- 周期の最大公約数を計算すればよい（詳細略）





UStar：最初の訪問時刻と訪問の周期

OptimizePP → NP困難（最大独立点集合問題から帰着）

- 最大独立点集合問題
  n点からなる無向グラフが与えられたときに
  独立点集合のうちサイズが最大のものを求める

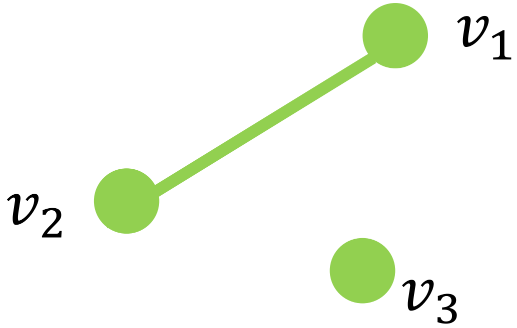
↓

- OptimizePP
  n点からなるUStar が与えられたときに
  警備できる頂点の数の最大値を求める

利得を全て1とすればよい

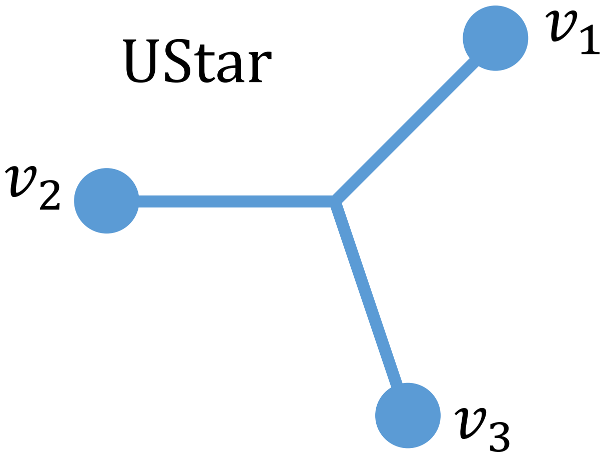
UStar：最初の訪問時刻と訪問の周期

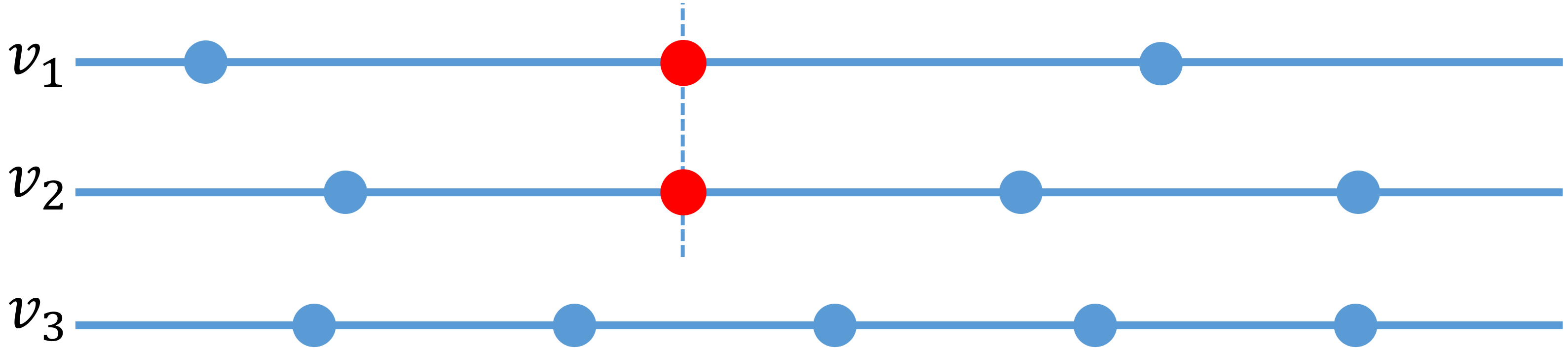
- 最大独立点集合問題
  「間に辺がある2頂点の両方は選べない」



こうなるように
最初の訪問時刻と周期を
設定すれば良い（詳細略）

- OptimizePP
  「2頂点の訪問しなければならない時刻が重複しているので両方は警備できない」





UStar

- 最初の訪問時刻と訪問の周期が与えられる問題
  - DecisionPP
    - 巡査が1人　→ 簡単に解ける
    - 巡査が複数 → 未解決
  - OptimizePP
    - 巡査が1人で利得が全て等しくてもNP困難
- 訪問の周期のみ与える問題
  - 「うまく最初の訪問時刻を設定すれば全点を警備できるか？」
  - DecisionPPで巡査が1人で利得が全て等しくてもNP困難
    (“Disjoint Residue Class Problem”[2]と同等)

[2] A. Kawamura and M. Soejima. Simple strategies versus optimal schedules in multi-agent patrolling. In International Conference on Algorithms and Complexity, pp.261–273. Springer, 2015.

まとめ

- LineもUStarも放置可能時間が全て同じならばOptimizePPを多項式時間で解くことができる.
- 放置可能時間が一般の場合は，Lineで巡査が複数の場合とUStarでは未解決

→ 放置可能時間を周期にして最初の訪問時刻を与える問題設定
  - UStar でDecisionPPだと巡査が1人なら多項式時間アルゴリズムあり
  - UStar でOptimizePPだと巡査が1人でもNP困難
  - いくつかの場合には多項式時間ではないアルゴリズムあり

→ 周期のみ与えられるとき
  - UStar はDecisionPPで巡査1人でもNP困難