Periodic Latency Problem with Profits

東京大学 総合文化研究科 広域科学専攻 広域システム科学系 河村研究室 修士 1 年
能城 秀彬

2016 年 7 月 27 日

1 Introduction

1.1 問題設定

辺の長さが与えられた無向グラフ $G = (V, E)$ の上を 1 人または複数の巡査が一定の速さで動きながら各点を警備する．各点 $v_i \in V$ には周期 q_i と利得 p_i が与えられている．ある点を警備できているとは，どの連続した 2 回の訪問も間隔がその点の周期以内であることを言う．警備している点から得られる利得の合計を最大化するのが目的である．

今回は大きく分けて以下の 3 つの問題を考える．

- PLPP(Periodic Latency Problem with Profit)：巡査が 1 人のとき，警備できる点部分集合を選び，得られる利得の合計を最大化する問題．
- MPLPP(Multiple server PLPP)：巡査が複数人のときの PLPP.
- PLP(Periodic Latency Problem)：全点を警備できる最小巡査数を求める問題．この場合は利得は無関係となる．

巡査は 2 人以上同時に 1 つの点に存在してもよいし，1 つの点は複数人の巡査により警備されてもよいとする．[2]

現実的な設定としては，巡査は点を訪問するときに一定時間そこにとどまる必要があるとするものが考えられるが（以降この時間を service time と呼ぶ），後で説明するようにグラフを変形することにより service time は 0 である問題に帰着できる．

1.2 記号の定義

- $G = (V, E)$：無向グラフ．
  - $V = \{v_1, v_2, \ldots, v_n\}$
    - p_i : v_i を警備することにより得られる利得．$p_i \leq 0$ の点は最初に除外してよいので $p_i > 0$ であるとする．
    - q_i : 周期．どの連続した 2 回の点 v_i の訪問も間隔が周期 q_i 以内であるときこの点 v_i を警備していると定義する．$q_i \geq 0$.
    - w_i : v_i を訪問したときにそこにとどまる必要がある時間 (service time)．$w_i \geq 0$.

- $E \subseteq V \times V$. 辺 $(v_i, v_j) \in E$ を e_{ij} とも書く．
  - $d(e)$: 辺 $e \in E$ の重み（長さ）．$e = (v_i, v_j)$ に対し $d(v_i, v_j)$ や $d_{i,j}$ のようにも表す．
- servers : $S = \{s_1, \ldots, s_m\}$
  - 速度 1 以下で辺上を動く．
  - $V_{s_i} \subseteq V$: 巡査 s_i が警備する点の集合．
  - $V_S \subseteq V$: $\bigcup_{s_i \in S} V_{s_i}$. 巡査 $s_i \in S$ が警備する点の集合の和集合．

1.3 トポロジー

今回はグラフ $G = (V, E)$ のトポロジーとして

- Line : $E = \{(v_i, v_{i+1}) \mid 1 \leq i < n\}$
- Circle : $E = \{(v_i, v_{i+1}) \mid 1 \leq i < n\} \cup \{(v_n, v_1)\}$
- Star : $E = \{(v_i, 0) \mid 1 \leq i \leq n\}$ 端点の一方が原点であるような Tree.
- StarC : $E = \{(x_i, 0) \mid 1 \leq i \leq n\}$, Star で辺の長さ $d_{i,0}$ が全て等しい場合．
- Tree : n 点からなる木．
- General : 一般のグラフ．

を考えることにする．Star は NP 困難性を示す際に便利なため導入している．StarC は Star と違い簡単に解ける場合があるため特別に考えることにする．

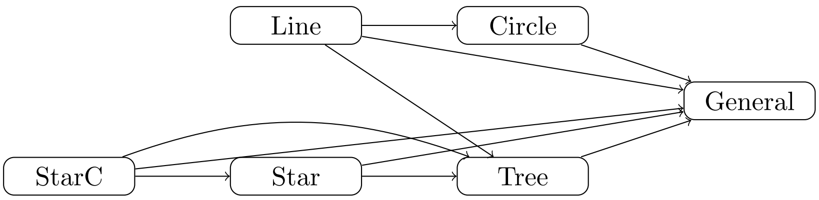
2 The Results

2.1 PLPP, MPLPP, PLP の関係

PLPP は明らかに MPLPP の特殊ケース ($m = 1$) であるので MPLPP へ帰着可能である．PLP は，巡査数は点と同数の n 人あれば自明に警備できることから，MPLPP を $m = 1$ から n までの場合について順に解き，初めて全点を巡回できたとき（最大化した利得が $\sum_{i=1}^{n} p_i$ に等しいとき）の m により求められるので，PLP は MPLPP へ多項式時間帰着可能である．

2.2 トポロジー間の関係

Line, Star は Tree の特殊な場合であるので Tree へ帰着可能である．Line は Circle に帰着可能である（Circle で $d(v_n, v_1)$ を十分大きくすればこの辺は使う必要が無いので Line の解になる）．今回考えるトポロジーについて帰着できる関係を矢印で示すと図 2.2 のようになる．



以下に現在確かめられている計算複雑性を示す．太字の部分を証明することにより他の部分を決めることが

できる．?の部分はまだ確定していないところである．

表 1　PLPP

	Line	Circle	StarC	Star	Tree	General
$q_i = Q$, $p_i = 1$	P	P	P	P	**P**	**NP-hard**
$q_i = Q$, arbitrary p_i	P	P	**P**	**NP-hard**	NP-hard	NP-hard
arbitrary q_i, $p_i = 1$	P	P	?	**NP-hard**	NP-hard	NP-hard
arbitrary q_i, arbitrary p_i	**P**	**P**	?	NP-hard	NP-hard	NP-hard

表 2　PLP

	Line	Circle	StarC	Star	Tree	General
$q_i = Q$	P	?	P	**NP-hard**	NP-hard	NP-hard
arbitrary q_i	?	?	?	NP-hard	NP-hard	NP-hard

表 3　MPLPP

	Line	Circle	StarC	Star	Tree	General
$q_i = Q$, $p_i = 1$	P	?	P	NP-hard	NP-hard	NP-hard
$q_i = Q$, arbitrary p_i	**P**	?	**P**	NP-hard	NP-hard	NP-hard
arbitrary q_i, $p_i = 1$	?	?	?	NP-hard	NP-hard	NP-hard
arbitrary q_i, arbitrary p_i	?	?	?	NP-hard	NP-hard	NP-hard

以下ではまず Service Time 無しの場合で考える．Service Time ありの場合は後述する．

3　Service Time 無し，巡査が 1 人の場合 (PLPP)

3.1　PLPP on Line

Line の場合は実直線上に全ての点を置くことができる．よって，以降トポロジーが Line の場合については x 軸上に点を置き，それぞれの座標を $x_1, x_2, \ldots, x_n$ とする．また，添え字は $x_1 \leq x_2 \leq \cdots \leq x_n$ となっているとしておく．

定理 3.1. PLPP on Line は多項式時間で解くことができる．

補題 3.2. PLPP on Line に対するある実行可能解 A があったとき，A で巡査が警備している最も左の（x 座標の小さい）点 v_l と最も右の点 v_r について，区間 $[x_l, x_r]$ を往復する動きにより A と同等以上の利得を得ることができる．

Proof. A で巡査 s が警備している点の集合を V_s と書く．$x_i < x_j$ であるような任意の $v_i, v_j \in V_s$ について，v_i を訪問した後 v_j を訪問し再び v_i に戻るまでの時間について，v_j と v_i の間の移動時間の最小値は片道 $x_j - x_i$ なので，点 v_i を q_i 以内に訪問できていることから，行き帰りの時間について，

$$2(x_j - x_i) \leq q_i \tag{1}$$

が成り立つ．同様に点 v_j についても同様に

$$2(x_j - x_i) \leq q_j \tag{2}$$

となる．$x_i < x_j$ であるような $v_i, v_j \in V_s$ は任意に選んでいたので

$$^{\forall}v_i, v_j \in V_s.\bigl(2(x_j - x_i) \leq q_i \text{ and } 2(x_j - x_i) \leq q_j\bigr) \tag{3}$$

x_i を基準に考えると

$$^{\forall}v_i \in V_s.\bigl(^{\forall}v_j \in V_s.\bigl(2(x_j - x_i) \leq q_i\bigr)\bigr) \tag{4}$$

すなわち

$$^{\forall}v_i \in V_s.\bigl(\max_{v_j \in V_s} 2(x_j - x_i) \leq q_i\bigr) \tag{5}$$

巡査が警備する点のうち左端と右端の点のいずれかが x_i から最も遠い点となるので

$$^{\forall}v_i \in V_s.\bigl(\max(2(x_i - x_l), 2(x_r - x_i)) \leq q_i\bigr) \tag{6}$$

ここで，v_r と v_l を往復する動きを考えると，これは式 (6) を満たすので V_s に含まれる点全てを警備することができる． □

以上により, PLPP on Line では 2 点間を往復する動きのみ考えれば良いことが分かったので，${}_nC_2$ 通りの両端を選び，その区間に含まれる点 x_i について $\max(2(x_i - x_l), 2(x_r - x_i)) \leq q_i$ を満たすものを警備するときの利得の合計を比較することで $O(n^3)$ で最適解を得ることができる． □(3.1).

これについては Coene ら [2] が $O(n^2)$ で最適解を得るアルゴリズムを与えている．

3.2 PLPP on Circle

定理 3.3. PLPP on Circle は多項式時間で解くことができる．

Circle の場合は Line の場合のアルゴリズムを利用して以下のように解くことができる．

補題 3.4. PLPP on Circle の最適解は，n 本の辺から 1 本を選び切り開いてできる n 通りの Line での解と，巡査が Circle 上を周回し続けるときの解を比較することで得られる．

Proof. 円周の長さ $d_{1,2} + d_{2,3} + \cdots + d_{n-1,n} + d_{n,1}$ を L とおく．

ある実行可能解 A があったとき，A で警備しているある点 $v_i \in V_s$ について，v_i から長さ $L/2$ の位置（円周上でちょうど反対の位置）$\overline{v_i}$ を巡査が一度も踏まない場合，その点を含む辺を切り開いた Line の解により A を改善することができる．どの点 $v_i \in V_s$ についても，円周上でその反対側の位置 $\overline{v_i}$ まで行く場合，最後に v_i を出発して $\overline{v_i}$ を訪れ，再び v_i に戻ってくるまでの時間が q_i 以下となるはずである．この時間は明らかに L 以上となるので，V_s に含まれる全ての点を周期 L で訪問できる周回運動により少なくとも A 以上の利得を得ることができる． □

以上により，最適解は n 通りの Line の結果と，周回運動により警備できる点の利得の合計を比較することにより得られるので，PLPP on Circle は多項式時間で解くことができる． □(3.3).

これについても Coene ら [2] が $O(n^2)$ で最適解を得るアルゴリズムを与えている．

3.3 PLPP on Star

Star は全ての辺の端点の一方が原点 0 であるグラフである．Star の場合は v_i と原点の間の辺の長さを $d_{i,0}$ と書く代わりに d_i と書くことにする．

PLPP on Star で $p_i = 1$, $q_i = Q$ の場合は簡単で，周期 Q を満たす範囲で原点からの辺の長さ d_i が短い点を順に選んでいけば良い．

一方，p_i, q_i のいずれかが任意の場合は NP 困難となる．

定理 3.5. PLPP on Star with arbitrary p_i は NP 困難である．

Proof. ナップザック問題からの帰着による．

ナップザック問題とは，入力として容量 B の容器と，品物の集合 ： $I = \{1, 2, \ldots, n\}$ が与えられ，各品物 i のサイズが a_i, 利得が w_i であるとき，I の部分集合 I' であって $\sum_{i \in I'} a_i \leq B$ を満たしながら利得の合計 $\sum_{i \in I'} w_i = W$ となるものが存在するかを判定する問題である．

この問題は PLPP on Star with arbitrary p_i に帰着できる．各品物 i に対して点 x_i を用意し，各辺の長さを $d_i = a_i/2$, 周期を $q_i = B$, 利得を $p_i = w_i$ とする．

すると，ナップザック問題の解 I' が与えられたとき，I' に対応する点の集合 V' について

$$\sum_{x_i \in V'} 2d_i = \sum_{i \in I'} a_i \leq B \tag{7}$$

より巡査は c' の点を全て警備することができ，利得の合計は W となる．逆に，PLPP on Star の解の点集合 V' が与えられたとき，利得の合計を W にするような巡回が存在するとき，V' に対応する品物集合 I' は

$$\sum_{i \in I'} a_i = \sum_{x_i \in V'} 2d_i \leq B \tag{8}$$

を満たすので，ナップザック問題の解とすることができる．以上によりナップザック問題を帰着できたので PLPP on Star with arbitrary p_i は NP 困難となる． □

定理 3.6. PLPP on Star with arbitrary q_i は NP 困難である．

Proof. 3-partition 問題からの帰着による．

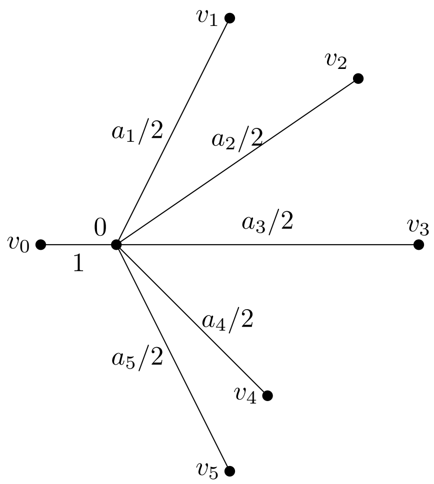
3-partition 問題とは，入力として，容量 B の容器が k 個与えられ，サイズ a_i が $B/4 < a_i < B/2$ を満たす品物が $3k$ 個与えられるとき，k 個の容器に $3k$ 個の品物を詰めることができるかを判定する問題である．$B/4 < a_i < B/2$ という制約から品物は 3 個まで容器に詰めることができるが，3 個ずつ詰められた場合のみ $3k$ 個の品物を k 個の容器に詰めることができる．

この問題は PLPP on Star with arbitrary q_i に帰着できる．点を $3k+1$ 個用意し，図 1 のように 1 つの点 v_0 は周期 $q_z = B + 2$, $d_{z0} = 1$, その他の点 v_i は周期 $q_i = k(B+2)$, $d_{i0} = a_i/2$ というようにする．

すると，3-partition が可能なときは，PLPP on Star で v_0 からスタートし，各容器に詰める 3 個ずつの品物に対応する点を 3 つ訪問して v_0 に戻るという巡回をすることにより，各点の周期を満たしながら全点を警備することができる．

逆に，PLPP on Star で全点を警備できるとき，巡査は v_0 を訪問した後 3 つの異なる点を選んで訪問し v_0 に戻ってくるということを k 回繰り返している必要がある．なぜならば，v_0 の周期は $B+2$ なので原点まで

図 1　3-Partition の PLPP への帰着



の行き帰り分の 2 を引くと点までの道のりを B まで走ることができるが，$B/4 < a_i$ という条件から高々 3 つの点しか訪問することはできず，逆に 2 つ以下しか訪問せずに v_0 に戻るとすると全点を 1 周するまでに k 回より多く v_0 に戻る必要があり，各点の周期 $k(B+2)$ を満たすことができないからである．よって，全点を 1 周できたときには 3 個ずつの点 k 組に分割できたことになるので 3-partition の解となる．

以上により 3-partition 問題を帰着できたので，PLPP on Star with arbitrary q_i は NP 困難となる．　□

3.4　PLPP on Star with $d_i = d$ (StarC)

以上で表 1 の PLPP on Star の列は決めることができたが，Star については d_i が任意ではなく一定とした場合も考えてみる．この場合は周期・利得のうちいずれかが定数ならば多項式時間で解くことができることが言える．

- $q_i = Q$ のときは全ての点の訪問のコストが等しいので，p_i の大きいものから選ぶことで最適解が得られる．p_i をソートして上位から $\left\lfloor \frac{Q}{2d} \right\rfloor$ 個の点を選ぶだけなので，ヒープソートを用いれば $O\left(n + \left\lfloor \frac{Q}{2d} \right\rfloor \log n\right)$ の時間で解くことができる．
- $p_i = 1$ のときは q_i の大きいものから選ぶことで，警備できる点の数を最大化できる．なぜならば，警備している点の集合 V_s について，$x_i \in V_s$ と $x_j \in V \setminus V_s$ であって $q_i < q_j$ となる組が存在すれば，辺の長さが全て同じなので x_i を訪問する時間で代わりに x_j を訪問することができ，$q_i < q_j$ であるからそれ以外の時間は元の巡査の動きをそのまま用いることができるためである．こうして得られる警備する点の集合 $V_s \cup \{x_j\} \setminus \{x_i\}$ の利得は元の利得以上となる．
  しかし，q_i の大きいものからいくつ選べるかは自明ではなく，現時点では未解決．
- p_i も q_i も定数でない場合も未解決．

↓ TODO ↓

開始時刻指定付き，q_i ちょうどで訪問，$p_i = 1$ という問題を考えるとこれは最大独立点集合問題からの帰着により NP 困難であることが示される．

「グラフ $G' = (V', E')$ にサイズ M 以上の独立点集合があるか」という問題を「PLPP on StarC, 開始時刻指定付き，q_i ちょうどで訪問，$p_i = 1$ ，で利得 M 以上の警邏が存在するか」に帰着する．

G' 上の独立点集合を考えるとき，$(u_i, u_j) \in E'$ である 2 点 $u_i, u_j \in V'$ は両方選ぶことはできないが，これを，PLPP on StarC で訪問されなければならない時刻が重複する時刻があるような 2 つの点 $v_i, v_j \in V$ を両方警備することはできないことにより表現する．

$(u_i, u_j) \in E'$ に対しては少なくとも 1 回 v_i, v_j の両方で同時刻に訪問しなければならないような時刻 t があるようにし，$(u_i, u_j) \notin E'$ に対しては 1 回もそのような時刻が現れないようにしたい．

各点の指定された訪問時刻の開始時刻を r_i とするとある 2 点 v_i, v_j について

$$^{\forall}k, l \in \mathbb{Z}. \, (q_i k + r_i \neq q_j l + r_j)$$

と同値な条件は，ユークリッド互除法により，最大公約数を $gcd(\,\cdot\,,\,\cdot\,)$ として

$$\Leftrightarrow {}^{\forall}k, l \in \mathbb{Z}. \, (q_i k - q_j l \neq r_j - r_i) \Leftrightarrow {}^{\forall}n \in \mathbb{Z}. \, (r_j - r_i \neq gcd(q_i, q_j) \cdot n)$$

と表される．同様に，

$$^{\exists}k, l \in \mathbb{Z}. \, (q_i k + r_i = q_j l + r_j) \Leftrightarrow {}^{\exists}n \in \mathbb{Z}. \, (r_j - r_i = gcd(q_i, q_j) \cdot n)$$

このとき，$^{\forall}n \in \mathbb{Z}. \, (r_j - r_i \neq gcd(q_i, q_j) \cdot n)$ である 2 点 v_i, v_j については q_i, q_j が互いに素でないことが必要になる．

↑ TODO ↑

3.5 PLPP on Tree

PLPP on Star は，PLPP on Tree の特殊な場合なので，p_i, q_i のいずれかが定数でない場合は NP 困難となる．よって，$p_i = 1$, $q_i = Q$ の場合のみを考えるが，これは Orienteering Problem と呼ばれる問題と同値であり多項式時間で解くことができることが知られている [1].

3.6 PLPP on General

一般のグラフについても $p_i = 1$, $q_i = Q$ のみを考えるが，これは NP 困難となる．

定理 3.7. PLPP on arbitrary topology with $p_i = 1$, $q_i = Q$ は NP 困難である．

Proof. ハミルトン閉路問題からの帰着による．

ハミルトン閉路問題とは，入力としてグラフ $G = (V, E)$ が与えられたときに，全点をちょうど 1 回訪れる巡回が存在するかを判定する問題である．

これは PLPP の問題に帰着できる．まずグラフ G を元に完全グラフ $G' = (V, E')$ で，E' の各辺 e_{ij} についてその長さが

$$d_{ij} = \begin{cases} 1 & \text{if } e_{ij} \in E \\ 2 & \text{if } e_{ij} \notin E \end{cases} \tag{9}$$

であるものを考え，V の点に点を配置し $p_i = 1, q_i = |V|$ とする．

すると，PLPP により利得 $|V|$ が得られるかを判定する問題は，長さ 1 の辺のみを動いて全点を警備できるかの判定に相当するので，ハミルトン閉路問題と同値となる．

以上によりハミルトン閉路問題を帰着できたので，PLPP on arbitrary topology with $p_i = 1, q_i = Q$ は NP 困難である． □

さらに d をユークリッド距離である場合に制限しても巡回セールスマン問題からの帰着により NP 困難であることが示される．

4 Service Time 無し，巡査が複数人の場合 (PLP, MPLPP)

巡査が複数人の場合，どの巡査も同じ速さ 1 以下としているため，巡査がすれ違うような動きはお互いの動きを交換し引き返すような動き方にすることができる．これにより特に Line の場合，巡査は初期配置の x 軸上の順番を保って動くとしてよいことが言える．ある実行可能解 A についてこのような変換を行ったものを A^* と表すことにする．

巡査が複数人の場合，Star 以上に複雑なトポロジーが全て NP 困難であることを PLP on Star で示したのち，残る Line, Circle の場合について調べていく．

4.1 PLP on Star

定理 4.1. PLP on Star は NP 困難である．

Proof. Partition からの帰着による．

Partition は，入力としてある非負の値の集合 $X = \{a_1, \ldots, a_n\}$ が与えられたとき，これを

$$X_1 \cap X_2 = \emptyset, \quad X_1 \cup X_2 = X, \quad \sum_{a_i \in X_1} a_i = \sum_{a_i \in X_2} a_i =: k$$

というように分割できるかを判定する問題である．

これは PLP on Star に帰着できる．先ほどの Partition の入力に対し，サイズ n の点の集合 $V = \{v_1, \ldots, v_n\}$ を考え，各 v_i について $d_i = a_i/2, q_i = k$ というようにする．この場合の PLP を解き，巡査数が 2 人で済むならばそれぞれが周期 k をちょうど満たしながら独立な点集合を巡回していることになるので，Partition は Yes となり，巡査 s_1, s_2 の警備する集合 V_{s_1}, V_{s_2} がそれぞれ X の分割 X_1, X_2 となる．巡査数が 3 人以上となるならば Partition は No となる．

以上により Partition 問題を帰着できたので PLP on Star は NP 困難である． □

Star の複数人の巡査による警邏は NP 困難であることが示されたが，StarC の場合，$^\forall q_i = Q$ の場合は多項式時間で最適解が得られることを示すことができる．

4.2 MPLPP on StarC, $q_i = Q$

この場合，点の訪問のコストが全て等しいことから，利得の大きいものから選べばよいことは明らかであるが，ちょうど $\left\lfloor \frac{mQ}{2d} \right\rfloor$ 点を警備する警邏を以下のように構成できる．

まず，利得の大きいものから $\left\lfloor \dfrac{mQ}{2d} \right\rfloor (=: l)$ 点を求め，$v'_1, v'_2, \ldots, v'_l$ とする $(p'_1 \geq p'_2 \geq \cdots \geq p'_l)$.

最初に巡査 s_1 が時刻 t_0 に原点を出発して $v'_1, v'_2, \ldots, v'_l$ を順番に速さ 1 で動きながら訪問していく．巡査 s_i は時刻 $t_0 + (i-1)Q$ に原点を出発し，s_1 より $(i-1)Q$ だけ遅れて同じ動きをする．各巡査が原点を出発して $v'_1, \ldots, v'_l$ を訪問し再び原点に戻るまでの時間は $2dl$ であり，$2dl = 2d \left\lfloor \dfrac{mQ}{2d} \right\rfloor \leq mQ$ であるから，時刻 $t_0 + (i-1)Q$ に原点を出発した巡査 s_i は時刻 $t_0 + (i-1)Q + mQ$ までには v'_l まで訪問して原点に戻っており，時刻 $t_0 + (i-1)Q + mQ$ に原点を出発して再び同じ動きを繰り返すことができる．

このような動きを繰り返すことで，どの点 v'_i についてもちょうど Q おきに巡査が訪問しているため，$v_1, \ldots, v'_l$ を警備できている．

一方，m 人の巡査が警備できる点は高々 $\left\lfloor \dfrac{mQ}{2d} \right\rfloor$ 点である．

任意の長さ Q の時間 $[t_0, t_0 + Q]$ を切り取って考える．点集合 $V_S \subseteq V$ の点を全て警備しているとき，V_S の全ての点は少なくとも 1 回はこの時間に訪問されなければならない．すると，各点 $v_i \in V_S$ に対して $[t_0, t_0 + Q]$ のうち少なくとも $2d$ の時間は v_i に対応する辺 e_i に巡査は存在しなければならない $(\cdots\star)$ ことを示すことができる．すると， m 人の巡査で V_S の点を訪問するとき巡査が使える時間は明らかに合計 mQ であるので，$2d \cdot |V_S| \leq mQ$ である必要があり，これにより $|V_S| \leq \left\lfloor \dfrac{mQ}{2d} \right\rfloor$ が言える．それでは $\star$ を示す．

任意の点 $v_i \in V_S$ について，時間 $[t_0, t_0 + Q]$ に少なくとも 1 回 v_i はいずれかの巡査により訪問される．

- ある巡査が時間 $[t_0 + d, t_0 + Q - d]$ に v_i を 1 度でも訪問しているとき（いずれかの巡査が v_i に存在する時刻が $[t_0 + d, t_0 + Q - d]$ の中にあるとき），少なくともその前後 $2d$ の時間はその巡査は辺 e_i 上に存在しなければならず，またこの時間は $[t_0, t_0 + Q]$ に含まれている．
- v_i が $[t_0 + d, t_0 + Q - d]$ に 1 度も訪問されないときは，$[t_0, t_0 + d)$ または $(t_0 + Q - d, t_0 + Q]$ に少なくとも 1 度訪問される．$[t_0, t_0 + d)$ に訪問される場合，点 v_i に巡査が存在した最後の時刻を $t_e \in [t_0, t_0 + d)$ とすると，$t_e + Q$ までに再び v_i は訪問されるはずである．$[t_e, t_0 + Q - d]$ に再び訪問されることはないときを考えているので，再度訪問される時刻は $[t_0 + Q - d, t_e + Q]$ に含まれることになる．すると，巡査は $[t_0, t_e + d] \cup [t_e + Q - d, t_0 + Q]$ の間は辺 e_i に存在することになるので，$[t_0, t_0 + Q]$ の間に合計で少なくとも $(t_e + d - t_0) + (t_0 + Q - (t_e + Q - d)) = 2d$ の時間巡査は辺 e_i に存在することになる．$(t_0 + Q - d, t_0 + Q]$ の場合も対称なので同様に考えればよい．

以上により $\star$ が示された．

以上から $\left\lfloor \dfrac{mQ}{2d} \right\rfloor$ 点の警邏が最適解であることが分かる．

次に Line の場合を考える．

4.3 MPLPP on Line, $q_i = Q$

定理 4.2. MPLPP on Line with $q_i = Q$ は多項式時間で解くことができる．

まず次の補題を示す．

補題 4.3. ある巡査 s 1 人のみにより警備されている点 x_i が存在するとき，s が動ける範囲は $|x - x_i| \leq q_i/2$ となる．

Proof of (4.3). もし s が $|x_{out} - x_i| > q_i/2$ であるような位置 x_{out} に行くことがあるとすると，s が最後に

x_i を出発してからの時間と，次に x_i に戻るまでの時間を合わせた時間 t は

$$t > 2|x_{out} - x_i| > q_i \tag{10}$$

となるため，x_i は s 以外の巡査により訪問されていないので x_i を警備できていないことになり矛盾する． □

補題 4.4. MPLPP on Line, $q_i = Q$ に対するある実行可能解 A について，各巡査がそれぞれ独立で両端に点がいるような区間を往復する実行可能解であって A 以上の利得を得られるようなものが存在する．

Proof of (4.4). A で警備する点の集合を $V_S = \{v'_1, v'_2, \dots, v'_k\} \quad (x'_1 \le x'_2 \le \dots \le x'_k)$ とする．A は巡査の初期順序を保つ動き A^* に変換してあるとし，それらを x 軸上の左側から $s_1, s_2, \dots, s_m$ とする．さらに，全ての巡査は区間 $[x'_1, x'_k]$ を動くように変換できる（$x < x'_1$ を動くものはその時間 x'_1 で待機するようにしても損をしないため．$x'_k < x$ を動くものも同様）．また，全点の周期が等しいため，同じ点に存在する複数の点はそれらの合計の利得をもつ新たな 1 つの点にまとめることができる．よって以下では $x'_1 < x'_2 < \dots < x'_k$ となっているとして進める．

まず x'_1 に注目する．いま，巡査順序を保つ A^* に変換しており，巡査は x'_1 より左には進まないように変換しているので，x'_1 に巡査 $s_i \ne s_1$ が訪れるときは必ず s_1 が x'_1 に存在することになる．よって，s_1 以外の巡査は x'_1 を訪れずに $x'_1 < x$ を動くようにしてよい（さらに $x'_2 \le x$ としても同じ利得が得られることが分かる）．

補題 (4.3) により s_1 の可動範囲は $[x'_1, x'_1 + Q/2]$ となるが，s_1 はこの区間を往復することによりこの区間に存在する全ての点を警備できる（PLPP on Line での証明より）ので，この区間 $[x'_1, x'_1 + Q/2]$ まで入ってくる s_1 以外の巡査はその時間 $x'_1 + Q/2$ で待機するようにしてよい．より正確には，V_S の点のうち x_a を超えない最も左にある点の位置を $lm(x_a) := \min_{\{x_i \in V_S \mid x_a \le x_i\}} x_i$, x_b を超えない最も右にある点の位置を $rm(x_a) := \min_{\{x_i \in V_S \mid x_i \le x_r\}} x_i$, と書くことにすると，$s_1$ は区間 $[x'_1, rm(x'_1 + Q/2)]$ を往復すればよく，$s_2, \dots, s_m$ のうち $[x'_1, x'_1 + Q/2]$ の内部を動いていた巡査は他の巡査の動く範囲に交わらない適当な位置に待機させることで取り除き，$[x'_1, x'_1 + Q/2]$ に入ってくる巡査は $lm(x'_1 + Q/2)$ でその時間待機するようにしてよい．

このような変換の後，$lm(x'_1 + Q/2)$ を警備している最も添え字の若い巡査を s_j とすると，$s_j, s_{j+1}, \dots, s_m$ と区間 $[lm(x'_1 + Q/2), x'_k]$ に存在する点について，s_1 のときと同様の変換をさらに行うことができる．これを最後まで繰り返すと，全ての巡査がそれぞれ両端に点が存在するような独立な区間を往復（または停止）するような動きに変換できる．この変換で A で警備していた点は全て警備できているので A 以上の利得が得られている． □

以上により，各巡査の動きとしては区間 $[x_i, rm(x_i + Q/2)]$ を往復するもののみを考えれば良いため，n 個の区間 $[x_i, rm(x_i + Q/2)] \quad (i = 1, 2, \dots, n)$ から m 個の重複のない区間を選ぶことで最適解が得られる．

PLP ならば全ての点を警備する必要があるので，これらの区間を左から重複のないように貪欲に選んでいけば巡査の動きを決定でき，最小巡査数が得られる．

MPLPP の場合は n 個の区間 $I_i := [x_i, rm(x_i + Q/2)]$ についてその区間に含まれる点から得られる利得の合計 P_i を求めておけば，重み付き区間の選択問題で区間数 $\le m$ の制限付きの場合を解くことになる．

まず $x_1, \dots, x_n$ はソートされていなければソートしておく ($O(n \log n)$).

前半の P_i の求め方だが，これは $x_1, \dots, x_n$ と $(x_1 + Q/2), \dots, (x_n + Q/2)$ を 1 つの長さ $2n$ の配列に併合し ($O(n)$), この配列を先頭から 1 つずつ見ていって，

- x_i なら $p \leftarrow p + p_i$,
- $x_i + Q/2$ なら $P_i \leftarrow p$; $p \leftarrow p - p_i$

という操作をしていくことで計算できる．p は今見ている長さ $Q/2$ の区間に含まれる点の利得の合計を表す一時変数で，0 で初期化する．

後半の，重複のない m 個の区間を選び重みの和を最大化する問題は，以下の漸化式 (11) に従う動的計画法で $m \times n$ の表を左上から計算することにより $O(mn)$ で最適な区間を選択できる．$h(j)$ は，区間 I_j より左にある交わらない区間のうち最も右にあるものの添え字を返すもので，$1 \leq j \leq n$ について $O(n)$ の前計算で得られる．$OPT(i,j)$ は，区間 $I_1, \ldots, I_j$ から 最大 i 個の区間を選ぶときの重みの合計の最大値を表す．$OPT(m,n)$ が求めたい利得の最大値となる．選ばれた区間も表をトレースバックすることにより再計算できる．

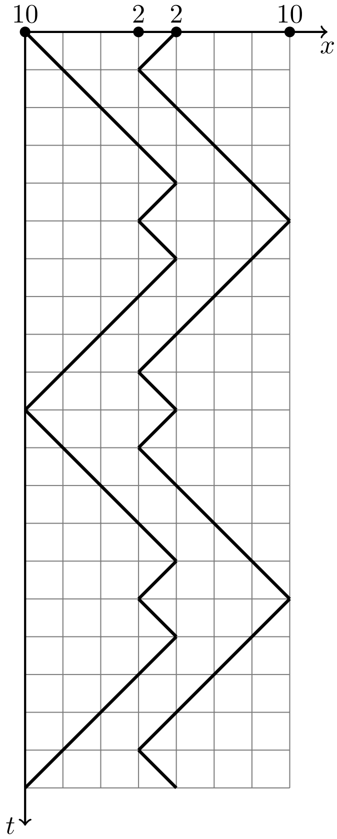
$$OPT(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ \max\Big(0, OPT(i,j-1), P_j + OPT(h(j), i-1)\Big) & \text{if } i \neq 0 \end{cases} \tag{11}$$

このアルゴリズムの計算量は全体で $O(n \log n + nm)$ である． $\square$(4.2).

4.4 PLP on Line with arbitrary q_i

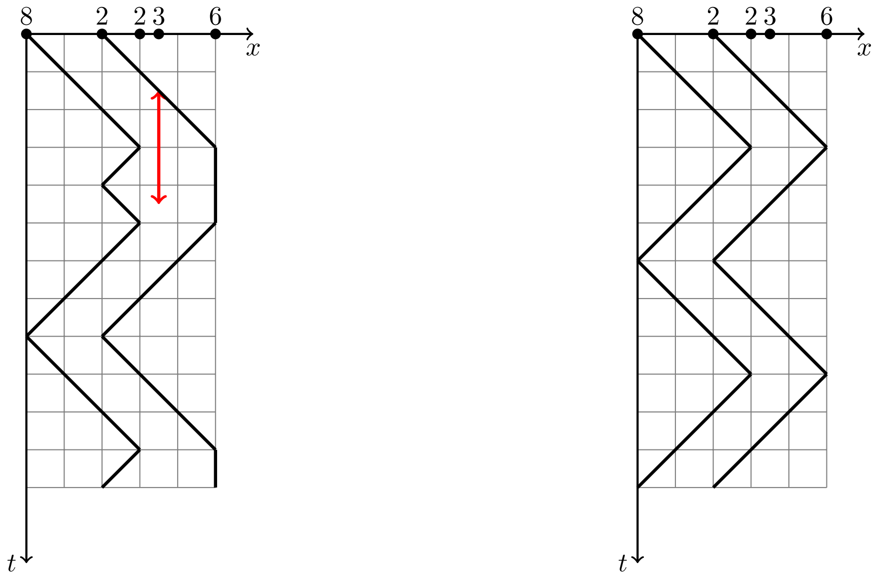
次に Line で周期が任意の場合を考える．点の周期が全て Q であるときは，巡査は各々の区間を独立に往復すればよかったが，周期が任意の場合，動く範囲に交わりがあり往復でもない動きをするのが最適であるような例が存在する．図 2 は巡査の軌跡（各時刻 t における巡査の位置 x を表す関数 $f : t \mapsto x$ のグラフ）を横軸を点の位置 x, 縦軸を時刻として t-x 平面に書いたものであるが，この例では，左から $q_1 = 10$, $q_2 = 2$, $q_3 = 2$, $q_4 = 10$ である 4 つの点が存在するとき，図のような動き方をしなければ 3 人目の巡査が必要になる．

図 2　複数人の巡査が複雑な動きをする例



そこでまず，この例から考えられる予想として，各巡査の動きは，最も左を動く s_1 から順に「なるべく右に手伝いに行く」動き方（後で数学的に定義する）により貪欲決定してよいかが気になるが，実はこれも成り立たない例が存在する．図 3 のように，左から $q_1 = 8$, $q_2 = 2$, $q_3 = 2$, $q_4 = 3$, $q_5 = 6$ である 5 つの点が存在するとき，左図のように s_1 がなるべく右に行くような動きをしてしまうと x_2, x_3, x_5 の周期を満たす動きが左図のようなものになり，x_4 が周期 3 以内に訪問されなくなってしまうので 3 人目の巡査が必要になるが，右図のようにあえて周期 6 で x_1 に戻るような動き方をすれば 2 人の巡査で 5 つの点を警備できる．

図 3　「なるべく右に手伝いに行く」戦略が失敗する例



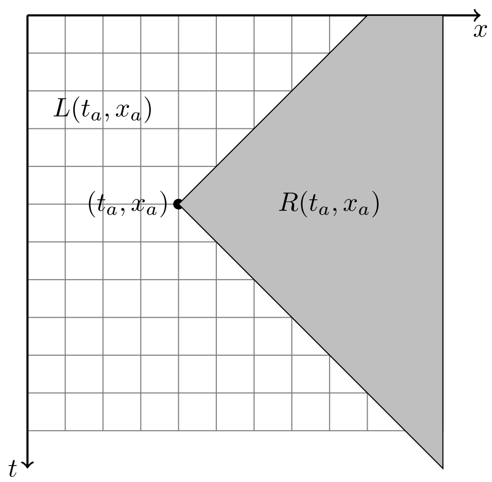
各点の周期 q_i は，q_i 以内に訪問されていれば良いという条件だったが，それにより，図 3 の例ではあえて周期 8 の点を周期 6 で訪問する方が良いために左側から巡査の動きを決定できないという難しさが生じてしまった．そこでまず，訪問しなければならない時刻に関して最も自由度の少ない，t-x 平面に訪問しなければならない位置と時刻を表す点が全て与えられているときを考えてみることにする．この設定では，左側から巡査を割り当て，なるべく右に行く動き方をさせることにより最適解が得られることを以下のように示すことができる．

定理 4.5. 各点 $x_i(i = 1, 2, \ldots, n)$ について，訪問すべき時刻の系列 $t_{i_1}, t_{i_2}, \ldots$ が指定されているとき，つまり，t-x 平面に巡査の動きを書いたときに巡査が踏むべき点の集合 $X = \bigcup_{i=1}^{n} \bigcup_{j} \{(t_{i_j}, x_i)\}$ が与えられているとき，s_1 から順に「なるべく右に行く動き方」で最適解（最小の巡査による訪問）を得られる．

定義 4.6. t-x 平面において，点 $a = (t_a, x_a)$ に対して

$$\begin{aligned} R(a) = R(t_a, x_a) &:= \{(t, x) \mid t < x - x_a + t_a \quad \text{and} \quad t > -x + x_a + t_a\} \\ L(a) = L(t_a, x_a) &:= \{(t, x) \mid t \geq x - x_a + t_a \quad \text{or} \quad t \leq -x + x_a + t_a\} \end{aligned}$$

と定義する．t-x 平面を全体として，$R(a)$ は $L(a)$ の補集合になる．



補題 4.7. 巡査 s の軌跡が t-x 平面上のある点 (t_a, x_a) を通るとき，s の軌跡は $L(t_a, x_a)$ に含まれる（$R(t_a, x_a)$ を通らない）.

Proof of (4.7). s の軌跡が (t_a, x_a) と $(t_b, x_b) \in R(t_a, x_a)$ を通るとする．$(t_1, x_1) \in R$ より

$$t_b < x_b - x_a + t_a \tag{12}$$

$$t_b > -x_b + x_a + t_a \tag{13}$$

が成り立つ.

(i) $t_b = t_a$ のときは，式 (12) から $x_b > x_a$ となり，s が同時に異なる 2 点に存在することはできないので矛盾.

(ii) $t_b < t_a$ のとき，巡査の速さが 1 以下であることから，s が $x = x_b$ から $x = x_a$ に移動するのには少なくとも $|x_b - x_a| = x_b - x_a$ 時間かかるので，時刻 t_b 以降で x_a に最初に到達する時刻 t_a は $t_a \geq t_b + x_b - x_a$ を満たすが，これは 式 (13) に矛盾する.

(iii) $t_b > t_a$ のときも (ii) と同様.

よって， s は $R(a, t_a)$ の点を通ることはできないので，s の軌跡は $L(a, t_a)$ に含まれる. □

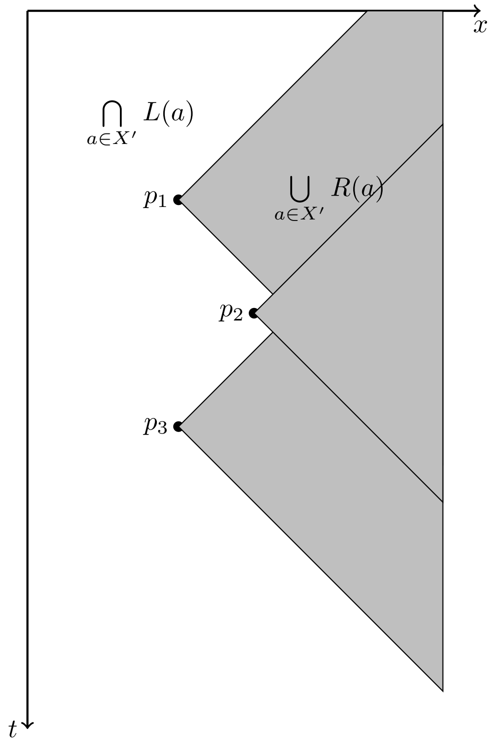
t-x 平面上の点の集合 X' が与えられ，何人かの巡査により X' の点全てを通る必要があるとする．$X' \neq \emptyset$ ならば，巡査 s を 1 人用意し，X' を担当する巡査の中で最も左を動くようにする．このとき，次の補題が成り立つ.

補題 4.8. s の軌跡は $\bigcap_{a \in X'} L(a)$ に含まれる.

Proof of (4.8). もし s の軌跡が $\bigcup_{a \in X'} R(a)$ の点 $b = (t_b, x_b)$ を通るとすると，ある $a = (t_a, x_a) \in X'$ が存在して $b \in R(a)$ であるが，このとき補題 (4.7) によると s の軌跡は a を通ることができない．すると，この a を通る巡査 s' が新たに必要になるが，時刻 t_a において s は $x \geq x_b - |t_b - t_a|$ の領域に存在するため，

- $t_b < t_a$ ならば，式 (13) から $x \geq x_b - |t_b - t_a| = x_b + t_b - t_a > x_a$.
- $t_b = t_a$ ならば，式 (12) または (13) から $x \geq x_b > x_a$.
- $t_b > t_a$ ならば，式 (12) から $x \geq x_b - |t_b - t_a| = x_b - t_b + t_a > x_a$.

より，時刻 t_b において s' は s より左に存在することになる．これは s を最も左側にある巡査にしたことに矛盾する．よって，最も左側の巡査 s の軌跡は $\bigcup_{a\in X'} R(a)$ の点を含まないので，$\bigcap_{a\in X'} L(a)$ に含まれることが言える． □



補題 4.9. t-x 平面上の点の集合 X' に対し，$\bigcap_{a\in X'} L(a)$ の（右の）境界線上にある点の集合を

$$B(X') := \left\{ b \in X' \;\middle|\; b \in \bigcap_{a\in X'} L(a) \right\}$$

と表す．X' を訪問する巡査のうち最も左にあるものを s とすると，s はその軌跡が $\bigcap_{a\in X'} L(a)$ の（右側の）境界と一致する動きが最適であり，これにより $B(X')$ の点全体を担当することができる（これを「なるべく右に行く動き」と定義する）．

Proof of (4.9). 補題 (4.7) より，s の軌跡は $\bigcap_{a\in X'} L(a)$ に含まれるので，s が通ることができる点全体は $B(X')$ の点の部分集合となる．逆に，s は $B(X')$ の点全てを通ることができる．なぜならば，$\bigcap_{a\in X'} L(a)$ の境界線は常に傾きが ± 1 であるため，$B(X')$ の点のうち時刻の最も早い点から出発して境界線上を動くことにより $B(X')$ の点全てを通ることができるためである．

また，$B(X')$ の点を全て通る動き方は最適である．なぜならば，$B(X')$ のうち通らない点がある場合，それらの点の集合を X'' と置くと，$X'' \cup (X' \setminus B(X'))$ の点全てを通る巡査達の動きはそのまま $X' \setminus B(X')$ 全体を通ることができるので，$B(X')$ のうち通らない点を残すことにより得をしていないからである． □

以上より，定理 (4.5) に戻ると，

$$X_i := \begin{cases} X & i=1 \\ X_{i-1} \setminus B(X_{i-1}) & i \geq 2 \end{cases}$$

として，X_i が空集合になるまで $i=1$ から順に $B(X_i)$ を巡査 s_i が担当するようにすれば，最適解が得られる．　□(4.5).

この定理により得られた結果はこの問題設定に対しては巡査を左からなるべく右に行く動き方で割り当てていけば良いということを数学的に示したものであるが，有限の手続きにより X 全体を通る巡査数を求めることができるとは限らない．なぜならば，X の点が時刻 t に関して周期的になっていない場合 X は t 方向に無限に長い領域で考える必要があるため X や $B(X)$ が無限集合となるためである．

X が有限集合の場合は $N := |X|$ として以下の手順により $\Theta(N \log N)$ で t-x 平面上の巡査達の軌跡を描くことができる．

まず，X の各点の元の t-x 座標系での表示を $45°$ 反時計回りに回転した u-y 座標系での表示に変換する．すなわち，各点 (t_i, x_i) を以下のように (u_i, y_i) に変換する．

$$\begin{pmatrix} u_i \\ y_i \end{pmatrix} = \begin{pmatrix} \cos(-45°) & -\sin(-45°) \\ \sin(-45°) & \cos(-45°) \end{pmatrix} \begin{pmatrix} t_i \\ x_i \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} t_i \\ x_i \end{pmatrix}$$

次にこの変換後の点 $a_1, \ldots, a_N$ を u の昇順，同着はさらに y の昇順でソートする．変換・ソート後の点を，$a'_1, a'_2, \ldots, a'_N$ とする $(u_1 \leq u_2 \leq \cdots \leq u_N)$．

以降，この u-y 平面上の点 $a'_1, a'_2, \ldots, a'_N$ に対して，y 軸に平行な scanline を $u = u_1$ の位置から右に動かしていくことを考える．

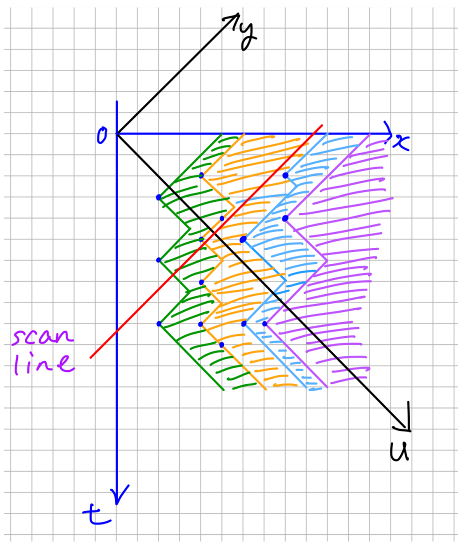


図 4　t-x 平面上に描いた巡査達の軌跡

初めに scanline 全体を色 c_0 で塗る．

scanline が a'_1 に重なったとき，scanline の $y > y_1$ の部分を色 c_1 で塗る．次に scanline が a'_2 に重なったとき，y_2 が scanline の色 c_0 の範囲（境界は含まれる）にあれば $y_2 \leq y \leq y_1$ と点 a'_2 を色 c_1 で塗る．y_2 が scanline の色 c_1 の範囲にあれば $y \geq y_2$ と点 a'_2 を色 c_2 で塗る．

このように，scanline を動かしていって新たに重なる点 a'_i が scanline の色 c_k の部分にあれば，y_i 以上の色 c_{k+1} で塗られた部分に行きつくまで（無ければ $y = \infty$ まで）色 c_{k+1} で塗ることを繰り返していく．このように点と重なるごとに更新されていく scanline の色で t-x 平面を塗り分けていくことで平面上のそれぞれの色 c_i の領域が $\bigcup_{a \in B(X_i)} R(a) \setminus \bigcup_{a \in B(X_{i+1})} R(a)$ を表すことになり，その境界線上を各巡査が動けばよいことが分かる．

scanline の色として説明した部分は，scanline の色に対応する区間を管理すればよく，平衡二分木を用いて，各点と重なる毎に $O(\log N)$ で検索・追加をすれば，合計 $O(N \log N)$ で平面の塗り分けが完了する．

scanline は各点を通る時点で各点を担当する巡査の番号と軌跡が決められるので，随時必要な情報を出力していくことができる．

これにより巡査の数を知ることができるのも明らかである．

5 Service Time 有りの場合

service time w_i が 0 でないときは，元のグラフの点の位置から長さ $w_i/2$ の枝を生やしてその末端に点を配置し service time 無しとした問題に帰着できる．すると，元の問題で全ての点が原点に集まっている場合 $(v_1 = v_2 = \cdots = v_n)$ でも帰着後のグラフが Star となるので，service time 無しの Star での結果から任意のトポロジーについて，PLPP では p_i, q_i いずれかが定数でない場合は NP 困難，PLP, MPLPP では $p_i = 1, q_i = Q$ でも NP 困難となる．

PLPP で $p_i = 1, q_i = Q$ の場合については，Service time 付き Line は Tree に帰着されるので多項式時間で解ける．Service time 付き Circle は未解決．StarC, Star は Star, Tree は Tree に帰着されるので多項式時間で解ける．

6 TODO

- PLP on Line, ちょうど q_i ごとの時刻は必ず訪問する場合.
- PLP/MPLPP on Circle with $q_i = Q$ (poly-time?)

参考文献

[1] Sofie Coene, Carlo Filippi, Frits CR Spieksma, and Elisa Stevanato. Balancing profits and costs on trees. *Networks*, Vol. 61, No. 3, pp. 200–211, 2013.

[2] Sofie Coene, Frits CR Spieksma, and Gerhard J Woeginger. Charlemagne's challenge: the periodic latency problem. *Operations research*, Vol. 59, No. 3, pp. 674–683, 2011.