

# 複数の巡査による指定地点の警邏について

河村 彰星<sup>1,a)</sup> 能城 秀彬<sup>1,b)</sup>

**概要:** 1人または複数の巡査が所与の領域を動き回り、その領域内のあらゆる場所を十分な頻度で訪問することで、これを守備、監督することを警邏 (patrolling) という。Coene らは、辺の長さの与えられた無向グラフと各頂点の持つ利得・放置可能時間、巡査の人数が与えられたとき、巡査がグラフ上を最高速度 1 で動き頂点を訪問することで警邏できる頂点集合のうち、利得の合計を最大化するものを求める警邏問題を考えた。ある頂点を警邏できているとは、どの連続した 2 回の訪問も間隔がその点の放置可能時間以内となるように訪問し続けられることと定義される。グラフが Line (1 つの線分上にすべての頂点があるグラフ) または Circle (Line の両端をつなぐ辺を足したグラフ) で巡査が 1 人の場合には多項式時間アルゴリズム、星、木、一般の場合については NP 困難性が示されている。本稿ではこの問題を取り上げ、多項式時間で解けるか否か未解決であった Line で巡査が複数の場合や、既に NP 困難性が示されている星よりも単純な、星で枝の長さがすべて等しい図形 (本稿では UStar と呼ぶことにする) について詳しく調べることにする。そのままの問題設定では解決できなかった部分については、放置可能時間の代わりに周期が与えられたときに、最初の訪問時刻からその周期ごとの時刻は必ず訪問しなければならないという問題、さらに最初の訪問時刻も与えられる問題も考える。

## Patrolling designated points on graphs

AKITOSHI KAWAMURA<sup>1,a)</sup> HIDEAKI NOSHIRO<sup>1,b)</sup>

### 1. はじめに

1人または複数の巡査が所与の領域を動き回り、その領域内のあらゆる場所を十分な頻度で訪問することで、これを守備、監督することを警邏 (patrolling) という [5]。

警邏に関する研究には様々な問題設定があり、例えば線分や閉路のような交わりの無い 1 次元的な領域のすべての点を警邏する塀の警邏 (Fence Patrolling) 問題 [2], [5] や、より一般的なグラフで辺全体ではなく頂点を警備する警邏問題を考えることもできる [4]。グラフと巡査が与えられて警邏可能かを判定する問題だけでなく、全体を警邏可能な最小の巡査数を求める問題、塀の警邏問題においてはなるべく長い塀を警邏する問題 [5] や全体の訪問の待ち時間の最大値を最小化する問題 [2]、頂点の警邏問題においては警邏できる頂点数を最大化したり、さらに頂点に利得を

設定しそれを最大化する問題、放置可能時間が頂点ごとに異なる場合 [4] も考えられている。

Coene ら [4] は、辺の長さの与えられた無向グラフと各頂点の持つ利得・放置可能時間、巡査の人数が与えられたとき、グラフ上を最高速度が 1 である 1 人または複数の巡査が動きながら、各頂点を訪問する間隔をその点の放置可能時間以内にしなければならない制約を満たしながら訪問し続けられる頂点から得られる利得の合計を最大化するような点部分集合を求める警邏問題を考えた。本稿ではこの問題を取り上げ、多項式時間で解けるか否か未解決であった形状について調べる。そのままの問題設定では解決できなかった部分については、放置可能時間の代わりに周期が与えられたときに、最初の訪問時刻からその周期ごとの時刻は必ず訪問しなければならないという問題、さらに最初の訪問時刻も指定される問題も考える。与えられた入力に対して単に全点を警邏できるかを判定する問題 DECISIONPP (Point Patrolling) と、利得の合計を最大化する問題 OPTIMIZEPP を考える。DECISIONPP は OPTIMIZEPP により解けるの

<sup>1</sup> 東京大学

<sup>a)</sup> kawamura@graco.c.u-tokyo.ac.jp

<sup>b)</sup> noshiro.pf@gmail.com

で OPTIMIZEPP は DECISIONPP 以上に難しいといえる。

他に、全点を警備できる最小の巡査数を求める問題も考えられるが、頂点が  $n$  点のとき、巡査が  $n$  人いれば各頂点に 1 人ずつ停止させることで全点を警備できることから、巡査数  $m = 1, \dots, n$  について DECISIONPP を解き全点を警備できた最初の  $m$  によって最小巡査数を求めることができ、逆に巡査数最小化を解きその巡査数と DECISIONPP の入力の巡査数の大小を比較することで DECISIONPP を判定できるので、巡査数最小化は多項式時間帰着の関係において DECISIONPP と同等である。

### 1.1 問題設定の詳細

頂点集合を  $V = \{v_1, \dots, v_n\}$ , 辺を  $E \subseteq V \times V$ , 辺の長さを  $d: E \rightarrow \mathbb{N}$  とする無向グラフ  $G = (V, E, d)$  の上を 1 人または複数人の点で表される巡査が速さ 1 以下で動きながら各頂点を訪問する。各頂点  $v_i$  には利得  $p_i$  と放置可能時間  $q_i$  が与えられている。ある頂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放置可能時間以内となるように訪問し続けられることと定義する。グラフ  $G$  と利得  $p_1, \dots, p_n$ , 放置可能時間  $q_1, \dots, q_n$  と巡査の人数  $m$  が与えられたときに、警備される点から得られる利得の合計を最大化する問題を OPTIMIZEPP と呼び、グラフ  $G$  と放置可能時間  $q_1, \dots, q_n$  と巡査の人数  $m$  が与えられたときに、 $G$  の全点を警備できるかを判定する問題を DECISIONPP と呼ぶことにする。DECISIONPP は全点を警備する問題なので利得は無関係になっている。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交代で訪問して警備してもよいとする。利得が 0 以下のものは警備する必要がないので最初に除外して利得はすべて正の整数であるとする。放置可能時間は非負整数とする。

## 2. 結果

### 2.1 既存の結果との関連と本稿の構成

本稿では Coene らと同じ問題設定 [4] を考える。Coene らは、グラフの形状として Line (線分), Circle (閉路), Star (星), Tree (木), 一般のグラフを扱っていた。Line は全ての頂点がある 1 つの線分上に存在するグラフ, Circle は Line の両端がつながり辺を足したグラフのことである。一般のグラフの場合、巡査が 1 人で利得・放置可能時間がすべて等しいという最も簡単な場合でも NP 困難であることが示されており、Star, Tree については、巡査が 1 人で利得・放置可能時間がすべて等しい場合のみには多項式時間アルゴリズムが存在するが、それ以外の巡査が複数である場合や利得・放置可能時間のいずれかが一般の場合はいずれも NP 困難であることが示されている。Line, Circle については巡査が 1 人の場合は多項式時間アルゴリズムが存在することが示されているが、巡査が複数の場合に対す

る多項式時間アルゴリズムは間違っていた。また、枝の長さがすべて等しい Star (本稿では UStar と呼ぶ) についても Coene ら [4] が巡査 1 人で利得がすべて等しい場合にのみ触れており、この場合は多項式時間アルゴリズムが存在すると述べているがこれも誤りであり、さらに巡査が複数の場合などについても扱っていなかった。以上のように、Coene らの論文 [4] で既に Star のような単純な図形でもほとんどの場合で NP 困難となっている一方、Line で巡査が複数の場合や UStar のような単純な図形については未解決の場合があるので本稿で調べることにした。

Coene ら [4] は、Line で巡査が複数の場合、すべての巡査はある区間を往復運動しており、そのうち任意の 2 つの区間は、互いに交わりの無い区間であるか、一方の区間がもう一方の区間に含まれるかのいずれかであるというような最適解が存在すると述べているが、これは誤りであり反例を 3.2 章で述べる。本稿では Line については巡査が複数の場合について調べ、まず、放置可能時間がすべて等しいならば多項式時間アルゴリズムが存在することを示した (定理 3.1)。放置可能時間が一般の場合については多項式時間アルゴリズムも NP 困難性も示せなかったので、放置可能時間ではなく周期と最初の訪問時刻が与えられ、最初の訪問時刻からその周期ごとの時刻は必ず訪問しなければならないという問題を考え、アルゴリズムを示した (定理 3.4)。

Star については上述のように多くの場合について NP 困難性が既に示されているが、辺の長さが異なることが証明においても重要な役割を果たしており問題の難しさの一つの要因であると考えられ、実際 UStar の場合は、放置可能時間がすべて等しい場合については巡査が複数でも多項式時間アルゴリズムを示すことができた (定理 4.1)。

以上をまとめた現在分かっている計算量を以下の表に示す。?の部分は未解決であることを表しており、括弧内は対応する定理番号を表す。太字が今回示した部分である。

表 1 OPTIMIZEPP の計算量一覧

放置可能時間	巡査数	Line	UStar	Star
すべて等しい	1	P	<b>P</b>	NP 困難 (*1)
すべて等しい	複数	<b>P (3.1)</b>	<b>P (4.1)</b>	NP 困難
一般の場合	1	P	? (*3)	NP 困難
一般の場合	複数	? (*2)	? (*4)	NP 困難

\*1 は利得がすべて等しいならば辺の短いものから移動時間の合計が  $Q$  を超えない範囲で選ばれよいため P となる [4]。

?のうち Line の \*2 の部分は放置可能時間ではなく周期として最初の訪問時刻も指定される問題ならば定理 3.4 の特殊な場合となり、この定理で述べるアルゴリズムにより解ける。UStar で放置可能時間ではなく周期として最初の訪問時刻も指定される問題ならば \*3 の部分に対応する巡

査が1人の場合は DECISIONPP ならば定理 4.2 のように P となり、一方 OPTIMIZEPP は巡査が1人でも定理 4.3 のように NP 困難となる (\*3,4 の部分に対応)。また、同じ設定で最初の訪問時刻は指定されない場合だと DECISIONPP でも定理 4.4 のように NP 困難となる (\*3,4 の部分に対応)。

### 3. Line

$G$  が Line の場合は実直線上にすべての点を置くことができるので、それぞれの座標を  $x_1, x_2, \dots, x_n$  とする。また、添え字は  $x_1 \leq x_2 \leq \dots \leq x_n$  としておく。

巡査が1人の場合については Coene ら [3] により多項式時間アルゴリズムが示されているので、巡査が複数の場合を考える。

巡査達の能力は皆同じなので、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行うことができる。これにより、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。

#### 3.1 放置可能時間がすべて等しい場合

まず放置可能時間がすべて等しい場合を考える。この場合、同じ位置に複数の点が存在するとき、そのどれか1つを警備できるならばすべて警備できるので、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。よってここでは  $x_1 < x_2 < \dots < x_n$  となっているとして進める。

**定理 3.1.**  $G$  が Line で放置可能時間がすべて等しい場合、巡査が複数で OPTIMIZEPP でも多項式時間アルゴリズムが存在する。

初めに次の補題を示す。

**補題 3.2.** ある1人の巡査  $s$  のみによって警備されている点  $x_i$  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。

**証明.** 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  $x_i$  を出発してから時間  $t$ 、次に  $x_i$  に戻るまでの時間を合わせた時間  $t$  は  $t > 2|x_{out} - x_i| > q_i$  となり、 $x_i$  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 3.3.** 巡査が複数、 $G$  が Line、 $q_i = Q$  の警邏問題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あ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。

**証明.**  $A$  で警備する点の集合を  $V_S = \{v'_1, v'_2, \dots, v'_k\}$  ( $x'_1 < x'_2 < \dots < x'_k$ ) とする。まず  $A$  を巡査の初期順

序を保つ動き  $A^*$  に変換し、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、 $x < x'_1$  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 ( $x'_k < x$  も同様)、すべての巡査は区間  $[x'_1, x'_k]$  を動くように変換できる。

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ため、 $x'_1$  に  $s_1$  以外の巡査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  $x'_2 \leq x$  を動くようにしてよい。

補題 3.2 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すべての点を警備できるので、この区間  $[x'_1, x'_1 + Q/2]$  に含まれる点を  $s_1$  以外の巡査が警備する必要はない。 $V_S$  の点のうち  $x$  より左にあり最も  $x$  に近い点の位置を  $l(x)$  より右にあり最も  $x$  に近い点の位置を  $r(x)$  と書くことにすると、 $s_1$  は区間  $[x'_1, l(x'_1 + Q/2)]$  を往復すればよく、 $s_2, \dots, s_m$  のうち  $[x'_1, r(x'_1 + Q/2))$  を動いていた巡査は  $r(x'_1 + Q/2)$  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、 $r(x'_1 + Q/2)$  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  $[r(x'_1 + Q/2), x'_k]$  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、すべ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1点で停止）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すべ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  $[x_i, l(x_i + Q/2)]$  を往復するもののみを考えれば良いため、 $n$  個の区間  $[x_i, l(x_i + Q/2)]$  ( $i = 1, 2, \dots, n$ ) から得られる利得の大きい順に  $m$  個の重複のない区間を選ぶことで最適解が得られる。 $n$  個の区間  $I_i := [x_i, l(x_i + Q/2)]$  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  $P_i$  を求めておき、重み付き区間の選択問題で選べる区間数が  $m$  個以下という制限付きの場合を解けばよい。

まず  $x_1, \dots, x_n$  はソートしておく。前半の  $P_i$  の求め方だが、 $x_1, \dots, x_n$  を  $x$  軸上に並べ、幅  $Q/2$  の区間を左から動かしていきながら点  $x_i$  がその区間を出入りするごとにこの区間に含まれる点の利得の合計を更新していくことで計算できる。これは、 $x_1, \dots, x_n$  と  $(x_1 + Q/2), \dots, (x_n + Q/2)$  を先頭から見て小さい方から1つずつ取り出していき、 $x_i$  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。 $p$  は今見ている幅  $Q/2$  の区間に含まれる点の利得の合計を表す一時変数で、最初に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1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  $I_1, \dots, I_j$  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, 巡査は互いに独立にそれらの区間を往復すればよい.

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

この算法の計算量は最初のソートも含めて全体で  $O(n \log n + nm)$  である. これで定理 3.1 が示された.

### 3.2 放置可能時間が一般の場合

次に, 放置可能時間が一般の場合を考える. こちらは計算量を決定できなかったが, この場合の難しさを示唆するいくつかの例を示す.

点の放置可能時間がすべて  $Q$  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, 放置可能時間が任意の場合, 動く範囲に交わりがあり往復でもない動きが最適となる次のような例が存在する. 図 1 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  $f: t \mapsto x$  を  $t$ - $x$  平面に描いたときのグラフ) 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全点を警邏するのに 3 人目の巡査が必要になる.

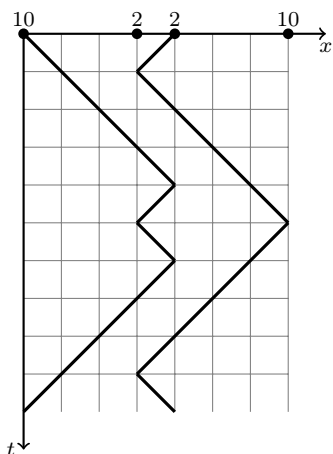


図 1 複数人の巡査が複雑に動く例

この例で最も左を動く  $s_1$  は「可能な限り右に手伝いに行

行く」動き方 (後で数学的に定義する) をしているのに, 各巡査の動きは最も左を動く  $s_1$  から順に「可能な限り右に手伝いに行く」動き方により貪欲決定してよいのではないかという予想を立てたが, これも成り立たない次のような例が存在する. 図 2 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  $s_1$  が可能な限り右に行くような動きをしてしまうと  $v_2, v_3, v_5$  の放置可能時間を満たす動きが左図のようなものになり ( $s_1$  の動きを図のように決めると  $v_2$  と  $v_5$  の訪問のために  $s_2$  の動きも図のように決まってしまう),  $v_4$  が放置可能時間 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放置可能時間 6 で  $v_1$  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.

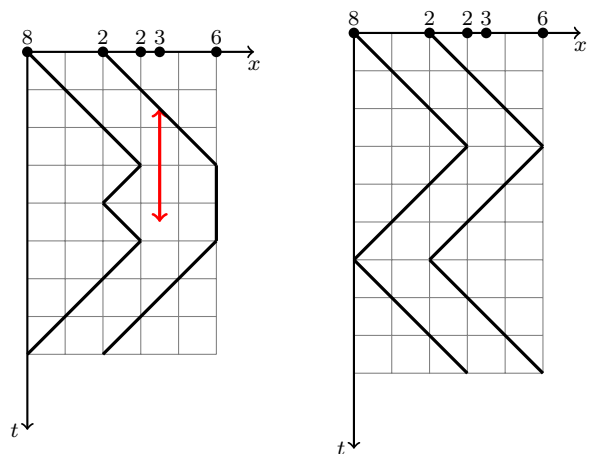


図 2 「可能な限り右に手伝いに行く」戦略が失敗する例

#### 3.2.1 最初の訪問時刻指定, 周期ちょうど毎に訪問

各頂点の放置可能時間とは, 連続する 2 回の訪問の間隔の上限であったが, それにより, 図 2 の例のようにあえて放置可能時間 8 の点を時間 6 毎に訪問する方が良いという, 左端から巡査の動きを貪欲に決定できないという難しさが生じてしまった. 放置可能時間の代わりに周期と最初の訪問時刻が与えられたときに, その最初の訪問時刻からその周期ごとの時刻は必ず訪問しなければならないという問題を考えてみる. すると, この設定では, 左側から巡査を割り当て, 「可能な限り右に行く」動き方をさせることにより全点の警備を考える DECISIONPP では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. 実際にはより一般的に, 時刻  $t$  と位置  $x$  について  $t$ - $x$  平面に訪問しなければならない位置と時刻を表す点がすべて与えられている問題についての次の定理により示される.

**定理 3.4.**  $G$  が Line で, 各頂点  $x_i$  に対し訪問しなければならない時刻  $t_{i_1}, t_{i_2}, \dots, t_{i_{N_i}}$  が有限個指定されているとき, 巡査が複数でも,  $X = \bigcup_{i=1}^n \bigcup_{k=1}^{N_i} \{(t_{i_k}, x_i)\}$  として  $O(|X| \log |X|)$  で DECISIONPP を解くことができる.

まず次の補題を示す.

**補題 3.5.** 各頂点  $x_i$  について, 訪問すべき時刻の列が指定されているとき,  $s_1, s_2, \dots$  を順に「可能な限り右に行く動き方」により動きを決定することで DECISIONPP の最適解を得られる.

**定義 3.6.**  $t$ - $x$  平面において, 点  $a = (t_a, x_a)$  に対して

$$R(a) := \{(t, x) \mid -x + x_a + t_a < t < x - x_a + t_a\}$$

$$L(a) := \{(t, x) \mid (t, x) \notin R(a)\}$$

と定義する.  $R(a)$  を  $R(t_a, x_a)$  のようにも書くことにする.

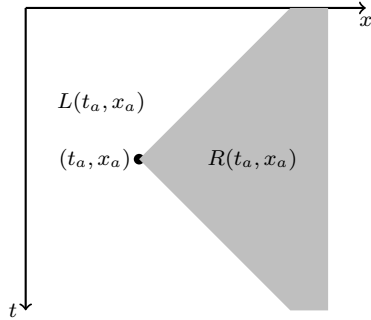


図 3  $L(t_a, x_a)$  と  $R(t_a, x_a)$  の定義

**補題 3.7.** 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).

**証明.**  $s$  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.  $(t_b, x_b) \in R$  より

$$-x_b + x_a + t_a < t_b < x_b - x_a + t_a \quad (2)$$

が成り立つ.  $t_b = t_a$  のときは, 式 2 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.  $t_b < t_a$  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 2 に矛盾する.  $t_b > t_a$  のときも同様.

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

$t$ - $x$  平面上の点の集合  $X'$  が与えられ, 何人かの巡査により  $X'$  の点すべてを通る必要があるとする.  $X' \neq \emptyset$  ならば, 巡査  $s_1$  を 1 人用意し,  $X'$  を担当する巡査の中で最も左を動くものとする. このとき, 次の補題が成り立つ.

**補題 3.8.**  $s_1$  の軌跡は  $\bigcap_{a \in X'} L(a)$  に含まれる.

**証明.** もし  $s_1$  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 3.7 によると  $s_1$  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  $s_2$  が新たに必要になるが, 時刻  $t_a$  において  $s_1$  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 2 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 2 から  $x \geq x_b > x_a$ ,  $t_b > t_a$  ならば, 式 2 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  $s_2$  は  $s_1$  より左に存在することになる. これは  $s_1$  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  $s_1$  の軌跡は  $\bigcup_{a \in X'} R(a)$  の点を含まないで,  $\bigcap_{a \in X'} L(a)$  に含まれることができる.  $\square$

**補題 3.9.**  $t$ - $x$  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

$$B(X') := \left\{ b \in X' \mid b \in \bigcap_{a \in X'} L(a) \right\} = X' \cap \bigcap_{a \in X'} L(a)$$

と表す.  $X'$  を訪問する巡査のうち最も左にあるものを  $s_1$  とすると,  $s_1$  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「可能な限り右に行く動き」と定義する).

**証明.** 補題 3.7 より,  $s_1$  の軌跡は  $\bigcap_{a \in X'} L(a)$  に含まれるので,  $s_1$  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,  $s_1$  は  $B(X')$  の点すべ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  $\pm 1$  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  $B(X')$  の点すべてを通ることができるためである.

また,  $B(X')$  の点をすべ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,  $X' \setminus B(X') \subseteq X' \cup X'' \cup (X' \setminus B(X'))$  より  $X'' \cup (X' \setminus B(X'))$  の点すべてを通る巡査達の動きにより  $X' \setminus B(X')$  全体を通ることができるので,  $B(X')$  をすべて  $s_1$  が通ることで損をしないからである.  $\square$

以上より, 定理 3.5 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. これで定理 3.5 が示された.

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可能な限り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,  $X$  が無限集合のときは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． $X$  が時刻方向に周期的になっている場合などのように有限集合で考えてよい場合は  $N := |X|$  以下の手順により  $O(N \log N)$  で  $t$ - $x$  平面上の巡査達の軌跡を描くことができる．

まず， $X$  の各頂点の元の  $t$ - $x$  座標系での表示を  $45^\circ$  反時計回りに回転した  $u$ - $y$  座標系での表示に変換する．

次にこの変換後の  $N$  点を  $u$  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  $a_1, \dots, a_N$  とする ( $u_1 \leq u_2 \leq \dots \leq u_N$ ) ．

以降，この  $u$ - $y$  平面上の  $N$  点に対して， $y$  軸に平行な scanline を  $u = u_1$  から  $u = u_N$  に向かって右に動かしていくことを考える．

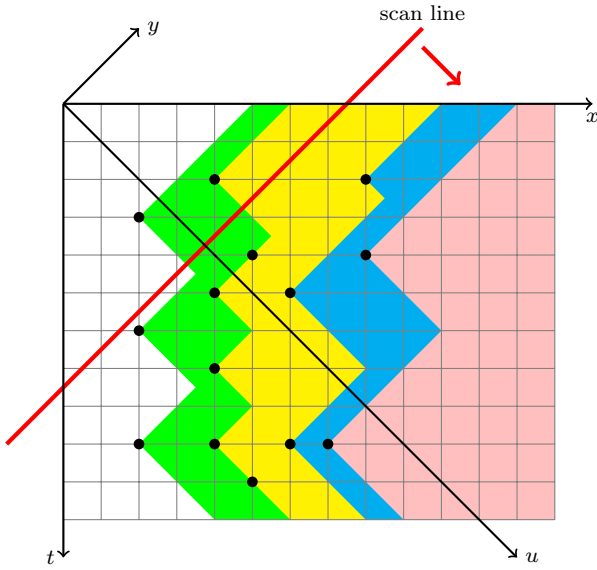


図 4  $t$ - $x$  平面上に描いた巡査達の軌跡

初めに scanline 全体を色  $c_0$  で塗る．

scanline が点  $a_1$  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 を動かしていったって新たに重なる点  $a_i$  が scanline の色  $c_k$  の部分にあれば， $y_i$  から色  $c_{k+1}$  で塗られた部分まで（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， $\bigcup_{a \in B(X_i)} L(a)$  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．ソートと合わ

せた全体の計算量は  $O(N \log N)$  となる．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．これにより必要な巡査の数を知ることができるのも明らかである．

系 3.10.  $G$  が Line, 巡査が 1 人または複数で，最初の訪問時刻が指定されており，以降周期ちょうど毎の時刻に訪問しなければならないとき， $N := \sum_{i=1}^n \frac{lcm(q_1, \dots, q_n)}{q_i}$  として  $O(N \log N)$  で必要な巡査の数とその軌跡を計算できる．

これは  $lcm(q_1, \dots, q_n)$  時間周期で繰り返しとなるので明らかである．

## 4. UStar

Star では巡査 1 人で利得と放置可能時間がすべて等しい場合を除き DECISIONPP は NP 困難であったが，UStar の場合は放置可能時間がすべて等しければ巡査が複数で OPTIMIZEPP でも多項式時間アルゴリズムが存在する．

### 4.1 放置可能時間がすべて等しい場合

定理 4.1.  $G$  が UStar で放置可能時間がすべて等しい場合，巡査が複数で OPTIMIZEPP でも多項式時間アルゴリズムが存在する．

証明．この場合，点の訪問のコストがすべて等しいことから，利得の大きいものから選ば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  $\left\lfloor \frac{mQ}{2d} \right\rfloor$  ( $=: l$ ) 点を求め， $v'_1, v'_2, \dots, v'_l$  とする ( $p'_1 \geq p'_2 \geq \dots \geq p'_l$ ) ．

最初に巡査  $s_1$  が時刻  $t_0$  に原点を出発して  $v'_1, v'_2, \dots, v'_l$  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  $v'_1, \dots, v'_l$  を訪問し再び原点に戻るまでの時間は  $2dl$  であり， $2dl = 2d \left\lfloor \frac{mQ}{2d} \right\rfloor \leq mQ$  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， $v_1, \dots, v'_l$  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  $\left\lfloor \frac{mQ}{2d} \right\rfloor$  点であることが示せる．任意の長さ  $Q$  の時間  $[t_0, t_0 + Q]$  を切り取って考える．警備している頂点集合を  $V_S$  とすると， $V_S$  のすべての頂点は少なくとも 1 回はこの時間に訪問されなければならない．すると，各頂点  $v_i \in V_S$  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．なぜならば，(1) ある巡査が時間  $[t_0 + d, t_0 + Q - d]$  に  $v_i$  を 1 度以上訪問している場合，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ない，またこの時間は  $[t_0, t_0 + Q]$  に含まれており，(2) 時間  $[t_0 + d, t_0 + Q - d]$  に  $v_i$  が 1 度も訪問されていない場合， $[t_0, t_0 + d]$  か  $(t_0 + Q - d, t_0 + Q]$  の少なくとも一方で訪問されるはずだが， $[t_0, t_0 + d]$  に訪問されるならば  $v_i$  にいた最後の時刻を  $t_e$  として  $t_e + Q$  までにもう一度は訪問されるはずであり，その時刻は  $[t_0 + d, t_0 + Q - d]$  に含まれない場合を考えているので，少なくとも  $[t_0, t_e + d]$  と  $[t_e + Q - d, t_0 + Q]$  の時間はどれかの巡査が  $e_i$  に存在しており，この時間は合わせて  $2d$  となり，いずれの場合も  $2d$  の時間は  $v_i$  の訪問のために使っている． $(t_0 + Q - d, t_0 + Q]$  の場合も同様である．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  $|V_S| \leq \left\lfloor \frac{mQ}{2d} \right\rfloor$  が言える．

以上から  $\left\lfloor \frac{mQ}{2d} \right\rfloor$  点の警邏が最適解であることが分かり，ヒープソートなどを用いれば  $O\left(n + \left\lfloor \frac{mQ}{2d} \right\rfloor \log n\right)$  の時間で解くことができる．これで定理 4.1 が示された．  $\square$

## 4.2 放置可能時間が一般の場合

UStar で放置可能時間がすべて等しい場合には多項式時間アルゴリズムが示せたが，放置可能時間が一般の場合は計算量を決定できなかった．そこで，ここでも Line のときのように，最初の訪問時刻からその周期ごとの時刻は必ず訪問しなければならないという問題をここでも考えてみる．

### 4.2.1 最初の訪問時刻指定，周期ちょうど毎に訪問

**定理 4.2.**  $G$  が UStar で巡査が 1 人，最初の訪問時刻と周期が与えられたときに，最初の訪問時刻からその周期ごとの時刻は必ず訪問しなければならないという制約の場合，周期は異なっても DECISIONPP には多項式時間アルゴリズムが存在する．

**証明.** 各頂点  $v_i$  を訪問しなければならない時刻は  $q_i k + r_i$  ( $k \in \mathbb{Z}$ ) として与えられる．全  $n$  点に対するこれらの和集合を  $T := \bigcup_{i=1}^n \{q_i k + r_i \mid k \in \mathbb{Z}\}$  とおくと，DECISIONPP が Yes である (= 全点を警備できる) ことと，連続した訪問しなければならない時刻の差が移動時間  $2d$  以上であること，すなわち  $T$  に含まれるどの 2 つの整数も差が  $2d$  以上であることが同値であることが分かる．これは任意の 2 点  $v_i, v_j \in V$  と任意の整数  $k, l$  に対して  $|(q_i k + r_i) - (q_j l + r_j)| \geq 2d$  となり，これは任意の整数  $n$  に対して  $|(r_i - r_j) + \gcd(q_i, q_j)n| \geq 2d$  と同値であり，左辺の最小値を考えると  $|r_i - r_j|$  を  $\gcd(q_i, q_j)$  で割った余り

$a$  と  $\gcd(q_i, q_j) - a$  のうち小さい方が  $2d$  以上かをチェックすればよい．この計算は定数時間であり  $nC_2$  通りこれを試せばよいので全体で多項式時間となる．  $\square$

DECISIONPP で巡査を複数とすると， $T$  に差が  $2d$  未満の整数が含まれていても複数の巡査によりそれぞれ訪問できる場合が生じる． $m$  人の巡査がいるので，任意の  $t \in T$  に対し  $[t, t + 2d)$  の時間に含まれる  $T$  の時刻が  $m$  個以下であれば全点を訪問でき， $m + 1$  以上となる  $t$  が存在すれば  $m$  人の巡査により訪問できない点が 1 つ以上生じる．これは  $T$  に含まれる整数の分布の情報を知る必要があるが，効率的なアルゴリズムは見つけれなかった．

DECISIONPP では巡査が 1 人ならば多項式時間アルゴリズムが存在したが，一方 OPTIMIZEPP は巡査が 1 人でも (複数人でも) NP 困難となる．

**定理 4.3.**  $G$  が UStar で，最初の訪問時刻と周期が与えられたときに，最初の訪問時刻からその周期ごとの時刻は必ず訪問しなければならないという制約の場合，巡査が 1 人で利得がすべて等しい場合でも，OPTIMIZEPP は NP 困難である．

**証明.** 最大独立点集合問題からの帰着による．

最大独立点集合問題は，無向グラフ  $G' = (V', E')$  と自然数  $M$  が与えられたときに独立点集合でサイズが最大のもを求める問題で NP 完全であることが知られている．この問題において，2 点間に辺の存在する  $v_i, v_j \in V'$  を両方選ぶことはできないということを，OPTIMIZEPP において 2 点のどちらか一方しか警備できないという状況を作ることで表現する．

まず簡単のため  $d = 1/2$  とする．これにより巡査がちょうど速さ 1 で動くすると各頂点の訪問にかかる時間は 1 となり，UStar なのですべての整数の時刻にどれか 1 点を訪問できる．その上で，周期  $q_i$ ，最初の訪問時刻  $r_i$  は整数， $p_i = 1$  とする．OPTIMIZEPP の UStar の点集合  $V$  は  $V'$  のサイズと同じとし，以降同じ添え字で対応させる．点  $v_i \in V$  を警備するために訪問しなければならない時刻の列は  $q_i k + r_i$  ( $k \in \mathbb{N}$ ) で表される．すると， $v_i$  と  $v_j$  の両方を警備できる必要十分条件は  $q_i k + r_i = q_j l + r_j$ ，移項すると  $r_i - r_j = q_j l - q_i k$  となる自然数  $k, l$  が存在しないこととなるが，これは  $r_i - r_j = \gcd(q_i, q_j)n$  となる整数  $n$  が存在しないことと同値である．よって， $v_i$  と  $v_j$  の両方を警備できる必要十分条件は  $r_i \not\equiv r_j \pmod{\gcd(q_i, q_j)}$  となる．

ここで， $nC_2$  個の相異なる素数  $p_{ij}$  ( $1 \leq i < j \leq n$ ) を用意し，各頂点の周期を  $q_i = p_{1i} p_{2i} \cdots p_{(i-1)i} p_{i(i+1)} \cdots p_{i,n}$  とすると， $\gcd(q_i, q_j) = p_{ij}$  ( $i < j$  のとき) となり，先ほどの条件は  $r_i \not\equiv r_j \pmod{p_{ij}}$  となる．

$G'$  において  $(v'_i, v'_j) \in E'$  ならば  $r_i \equiv r_j \equiv 0 \pmod{p_{ij}}$ ， $(v'_i, v'_j) \notin E'$  ならば  $r_i \equiv 0, r_j \equiv 1 \pmod{p_{ij}}$  と定めると，

各  $r_k$  に対して相異なる  $n-1$  個の素数で割ったときの余りが与えられるので、中国剰余定理からそのような  $r_k$  がその  $n-1$  個の素数の積  $q_k$  を法として一意に存在することが言え、これにより、 $(v'_i, v'_j) \in E'$  ならば  $r_i \equiv r_j \pmod{p_{ij}}$  ( $v'_i, v'_j \notin E'$  ならば  $r_i \not\equiv r_j \pmod{p_{ij}}$ ) を満たすように各  $r_k$  を定めることができる。

最後に、 $nC_2$  個の相異なる素数を用意する計算量も確かめる必要がある。  $k$  番目に小さい素数を  $P_k$  と書くと、 $k \geq 6$  のときは  $P_k < k(\ln k + \ln \ln k)$  であることが知られているため [6],  $k(\ln k + \ln \ln k)$  までの自然数を順に素数かどうか判定していくことで  $k$  個以上の素数を得ることができる。ある数が素数であるかどうかを判定する多項式時間アルゴリズムは存在するので [1],  $nC_2$  個の素数の列挙は  $n$  の多項式時間でできる。

以上の手順で  $q_k$  と  $r_k$  を設定することにより、 $G'$  においてサイズ最大の独立点集合を求める問題を、巡査 1 人、 $G$  が UStar, 最初の訪問時刻が指定され周期ちょうど毎に訪問しなければならない制約での OPTIMIZEPP に帰着できた。以上より定理 4.3 は示された。  $\square$

#### 4.2.2 周期ちょうど毎に訪問

今、最初の訪問時刻と周期が指定された場合を考えだが、周期のみが指定されている場合は DECISIONPP でも NP 困難となる。

**定理 4.4.**  $G$  が UStar で、周期のみ与えられたときに、最初の訪問時刻からその周期ごとの時刻は必ず訪問しなければならないという制約の場合、巡査が 1 人で DECISIONPP でも NP 困難である。

**証明.** Disjoint Residue Class Problem からの帰着による。

ある整数のペアの集合  $\{(m_1, r_1), \dots, (m_n, r_n)\}$  が Disjoint Residue Class であるとは、任意の整数  $x$  に対して  $x \equiv r_i \pmod{m_i}$  となるような  $i$  が高々 1 つ存在することと定義する。Disjoint Residue Class Problem とは整数の組  $(m_1, \dots, m_n)$  が与えられたときに、 $\{(m_1, r_1), \dots, (m_n, r_n)\}$  が Disjoint Residue Class となるような組  $(r_1, \dots, r_n)$  が存在するかを判定する問題であり、これは NP 困難であることが示されている [7]。

Disjoint Residue Class Problem は、巡査 1 人、 $G$  が UStar,  $p_i = 1$  で周期ごとの時刻は必ず訪問しなければならないという制約での DECISIONPP に多項式時間帰着できる。Disjoint Residue Class Problem の入力  $(m_1, \dots, m_n)$  のとき、UStar で  $V = \{v_1, \dots, v_n\}$ ,  $q_i = m_i$ ,  $d = 1/2$  とする。  $d = 1/2$  より整数の時刻にいずれかの点を訪問できるようにする。各頂点  $v_i$  の最初の訪問時刻を  $r_i$  とすると、この点を警備するために訪問しなければならない時刻の列は  $q_i k + r_i (k \in \mathbb{N})$  で与えられるが、全点を警備するためには任意の 2 点  $v_i, v_j \in V$ , 任意の整数  $k, l$  について

$q_i k + r_i \neq q_j l + r_j$  である必要がある。Disjoint Residue Class Problem の解  $(r_1, \dots, r_n)$  が存在するならば、これにより任意の時刻  $t \in \mathbb{Z}$  に対して  $t \equiv r_i \pmod{q_i}$ , すなわち  $t = r_i + q_i k$  となる  $k \in \mathbb{Z}$  が存在するような  $i$  は高々 1 つであり、任意の  $k, l \in \mathbb{Z}$  に対して  $q_i k + r_i \neq q_j l + r_j$  が成り立つので、巡査は全点を警備できる。よって、Disjoint Residue Class Problem が Yes ならばこの DECISIONPP も Yes となる。

逆に、DECISIONPP の解が存在するとき、全点を警備できるのでその警邏において各頂点  $v_i$  を最初に訪問する時刻を  $r_i$  とすると、任意の  $v_i, v_j \in V$ ,  $k, l \in \mathbb{Z}$  に対し

$$q_i k + r_i \neq q_j l + r_j \quad (3)$$

が成り立つ。すると、任意の時刻  $t \in \mathbb{Z}$  に対して  $t \equiv r_i \pmod{q_i}$  となるような  $i$  が 2 つ存在するとすると、それを  $i, j$  として  $t \equiv r_i \pmod{q_i}$ ,  $t \equiv r_j \pmod{q_j}$  すなわち、ある整数  $k, l$  が存在して  $t = q_i k + r_i$ ,  $t = q_j l + r_j$  となり、この  $k, l$  によって  $q_i k + r_i = q_j l + r_j$  となり、式 3 に矛盾する。よって、任意の整数  $t$  に対して  $t \equiv r_i \pmod{q_i}$  を満たす  $i$  は高々 1 つであるような  $r_i$  が与えられたので、DECISIONPP が Yes ならば Disjoint Residue Class Problem も Yes となる。

以上より定理 4.4 は示された。  $\square$

**謝辞** 本研究は旭硝子財団および科研費新学術領域「計算限界解明」の助成を受けて行われた。

#### 参考文献

- [1] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. *Annals of Mathematics*, 160(2), pp. 781–793, 2004.
- [2] K. Chen, A. Dumitrescu, and A. Ghosh. On fence patrolling by mobile agents. In *CCCG*, 2013.
- [3] S. Coene, C. Filippi, F.C.R. Spieksma, and E. Stevanato. Balancing profits and costs on trees. *Networks*, 61(3), pp. 200–211, 2013.
- [4] S. Coene, F.C.R. Spieksma, and G.J. Woeginger. Charlemagne's challenge: the periodic latency problem. *Operations research*, 59(3), pp. 674–683, 2011.
- [5] J. Czyzowicz, L. Gąsieniec, A. Kosowski, and E. Kranakis. Boundary patrolling by mobile agents with distinct maximal speeds. In *European Symposium on Algorithms*, pp. 701–712. Springer, 2011.
- [6] P. Dusart. The  $k$ th prime is greater than  $k(\ln k + \ln \ln k - 1)$  for  $k \geq 2$ . *Mathematics of Computation*, 68, pp. 411–415, 1999.
- [7] 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.