Proper Circular Arc Graph における最小安全支配集合問題

下西希実　(指導教員：長尾篤樹)

1　はじめに

警備員の配置場所等のモデル化として最小安全支配集合問題が考えられている。この問題は NP 完全ではあるが、Proper Interval Graph に関しては線形時間アルゴリズムが存在することが知られている。[1] 本研究では Proper Circular Arc Graph においても最小安全支配集合が多項式時間で解けることを証明する。

2　安全支配集合問題

頂点数 n の無向グラフ $G=(V,E)$ について、集合 $D \subseteq V$ は、各 $u \in V \setminus D$ がある $v \in D$ に隣接するとき、D は G の支配集合である。Proper Circular Arc Graph では $O(n)$ で求められるとわかっている。[2]

G の支配集合 $S \subseteq V$ は各 $u \in V \setminus S$ について、隣接する $v \in S$ との交代集合 $(S \setminus \{v\}) \cup \{u\}$ が再び G の支配集合となるとき、S は G の安全支配集合である。このとき v は u を守るという。[3]

2.1　既存研究

Proper Interval Graph とは、区間グラフのうち、ある区間の真部分集合になるような区間が存在しない区間グラフである。頂点数 n の Proper Interval Graph の左端の頂点を v_1 とし、右向きに順に $v_1, v_2, ..., v_n$ とする。v_i の隣接頂点を $N[v_i]$ と表す。全ての i について $N[v_i]$ が連続、つまり、ある i_1, i_2 に対して $N[v_i] = \{v_t | i_1 \leq t \leq i_2\}$ となるとき、グラフ G の頂点 $(v_1, ..., v_n)$ は Consecutive Ordering を持つという。またこのとき、隣接頂点 $N[v_i]$ のうち最大のものを $maxN(v_i)$ と表す。グラフ G が consecutive ordering を持つとき、グラフ G は proper interval graph である。[4] これを利用し、Proper Interval Graph における最小安全支配集合問題は $O(|V(G)|)$ で求められることが証明されている。[1]

3　Proper Circular Arc Graph における安全支配集合問題

3.1　Proper Circular Arc Graph

Proper Circular Arc Graph とは区間が円弧上になった Proper Interval Graph である。区間を頂点とみなし、ある区間に対し、その始点を含む区間を左側の区間 (頂点) と呼び、その終点を含む区間を右側の区間 (頂点) と呼ぶ。この定義の下で、ある頂点を v_1 と決めたなら、各頂点は時計周りを右向きと見なして順に $v_1, v_2, ..., v_n$ と呼ぶことができ、consecutive ordering となる。これにより、$maxN(v_i)$ は v_i の隣接頂点のうち、最も右側にあるものとする。さらに、$i > j$ に対し、$N[v_i]$ が v_j を v_i の右側に含むとき、その v_j を二周目の v_j と呼ぶ。安全支配集合 S の要素を $\{x_1, ..., x_k\}$, x_i が守る左端の頂点を y_i, 右端の頂点を t_i とする。頂点の位置関係を $\leq$ で表現し、$v_i \leq v_j$ は $i \leq j$ を表す。さらに、v_i^+ を v_i の右側にある頂点で一番左側にあるものとする。

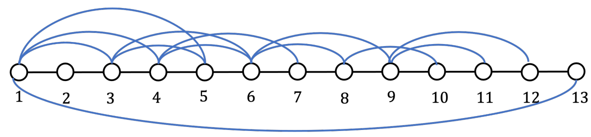

図 1　Proper Circular Arc Graph の例

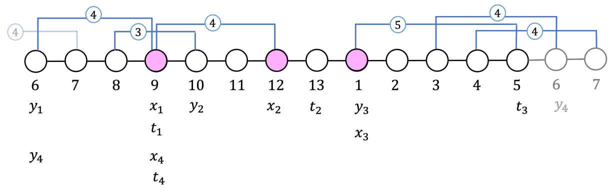

図 2　頂点番号 6 を v_1 としてアルゴリズム 1 を適用したもの。y_4 は二周目の頂点番号 6 となっている。

3.2　提案アルゴリズム

本節では、Proper Circular Arc Graph における安全支配集合問題を解くアルゴリズムを提案する。付録にある本アルゴリズムは Proper Circular Arc Graph のある頂点を v_1 と指定したのちに本問題を解く。以下、本アルゴリズムの正当性を証明する。

Lemma 1. $y_1 \leq y_k$ ならば $S = \{x_i, ..., x_{k-1}\}$ は G 全体を守る。(y_k のみ二周目のものとする。)

Proof. 全ての $i = 1, 2, ..., k$ について $x_i \leq t_i \leq maxN(x_i)$ であり、$y_i \in N[x_i]$ である。すなわち、x_i は y_i から t_i の全ての頂点に隣接する。また、$y_i = (t_{i-1})^+$ であるので、$y_1 \leq y_k$ であるとき、t_{k-1} が y_1 の左隣またはそれより右側となるので G の全ての頂点は S の頂点に隣接する。 □

Theorem 2. アルゴリズム *1* で得られる集合 S は安全支配集合である。

Proof. Lemma 1 より X が G 全体を守るためには $y_1 \leq y_k$ であれば良い。$x_1 \leq x_k$ ならば $y_1 \leq y_k$ であることを示す。(x_k のみ二周目のものとする。)

(1) $x_k = maxN(y_k)$ の場合
$x_1 = maxN(y_1)$ で $x_1 \leq x_k$ つまり $maxN(y_1) \leq maxN(y_k)$ G は Proper Circular Arc Graph なので $y_1 \leq y_k$

(2) $x_k = maxN(maxN(y_{k-1})^+)$ の場合
$maxN(y_{k-1}) < t_{k-1}$ よって $maxN(y_{k-1}^+) < y_k$
$y_k < y_1$ とすると $x_k = maxN(maxN(y_{k-1})^+) < maxN(y_1)$ となり、$x_k \leq x_1$ であることに矛盾。よって仮定が否定され、$y_1 \leq y_k$ □

Lemma 3. 頂点 u を開始頂点としてアルゴリズム *1* を適用した際に得られる *SDS* S_u と、頂点 v を開始頂点としてアルゴリズム *1* を適用した際に得られる *SDS* S_v を考える。このとき、S_u における x_1 と S_v における x_1 とが同じ頂点を指す場合、$S_u = S_v$ である。

Proof. $x_1 = t_1 = maxN(y_1)$ より、y_1 から t_1 はクリーク。$y_2 = t_1^+$ で、$x_2 = maxN(y_2) = maxN(x_1^+)$.

x_1 から x_2 がクリークかどうかは y_1 に依らない。クリークならば $t_2 = maxN(x_2)$, クリークでないならば $t_2 = maxN(maxN(x_1)^+)$ よって、x_1 が決まれば、y_1 の位置に関わらず、y_2, x_2, t_2 が一意に定まり、$i > 2$ でも $i-1$ の頂点のみで y_i, x_i, t_i が定まる。$x_1 \leq x_k$ が成り立てば停止するので、$S_u = S_v = \{x_1, ..., x_{k-1}\}$ 全て同じ要素となる。したがって、S_u と S_v で x_1 が同じ頂点を指す場合、$S_u = S_v$ □

Lemma 4. *Proper Circular Arc Graph* における *SDS* S_u に対し、ある頂点 $u \in S_u$ が G に含まれるどの極大なクリークにも右端の頂点ではないとする。この時、u を含む一番左に存在する極大なクリークを C_u とし、その右端の頂点を v とすると、$S_u \cup \{v\} \setminus \{u\}$ もまた、*SDS* である。

Proof. u と隣接する左端の頂点はクリーク C_u の左端の頂点である。右端の頂点は $maxN(u)$ である。u が守る頂点の範囲は最大で C_u の左端の頂点から $maxN(u)$ である。v もクリーク C_u 内の頂点なので、v と隣接する左端の頂点は変わらない。$u < v$ より $maxN(u) \leq maxN(v)$ なので v と隣接する右端の頂点は u と隣接する右端の頂点よりも手前には存在しない。したがって、C_u の右端の頂点 v は u と入れ替えても守ることのできる頂点の範囲は減ることはない。よって $(S_u \cup \{v\}) \setminus \{u\}$ もまた、G における SDS である。 □

G における全ての頂点が何かしらの極大なクリークの右端の頂点となっている任意の最小安全支配集合 $S_u = \{s_1, ..., s_l\}$ に対し、$x_1 = s_1$ となる条件からアルゴリズム 1 を実行して得られる安全支配集合を $S = \{x_1, ..., x_k\}$ とする。この時、$v_1 = y_1, v_n^+ = v_1$ とする。このとき、アルゴリズムより $s_i \leq x_i$ が言え、さらに以下の補題が成り立つ。

Lemma 5. $i \geq 2$ について、$1 \leq j \leq i-1$ に対して、s_j は y_{j+1} を守らない。

Proof. $j \in \{2, 3, ..., i-1\}$ に対して、s_j が y_{j+1} を守ると仮定する。つまり、s_j が y_{j+1} に隣接し、$S_u^{'} = (S_u \setminus \{s_j\}) \cup \{y_{j+1}\}$ が支配集合であると仮定する。s_j は y_{j+1} に隣接するため $y_{j+1} \leq maxN(s_j)$ であり、さらに仮定より $s_j \leq x_j \leq t_j < y_{j+1}$ である。以下、アルゴリズムの挙動により場合分けをする。
(1) $t_j = maxN(x_j)$ の場合
$y_{j+1} = t_j^+ = maxN(x_j)^+$ である。s_j が y_{j+1} を守るとすると、$y_{j+1} \leq maxN(s_j) \leq maxN(x_j) = t_j$ となり矛盾。
(2) $t_j = maxN(maxN(x_{j-1})^+)$ の場合
$w = maxN(x_{j-1})^+$ とする。$y_{j+1} = t_j^+ = maxN(w)^+$ であるので、w は y_{j+1} と隣接しない。また、$w > maxN(x_{j-1}) \geq maxN(s_{j-1})$ なので、s_{j-1} は w と隣接しない。よって、$S_u^{'} = (S_u \setminus \{s_i\}) \cup \{y_{j+1}\})$ の中に w と隣接する点はない。よって矛盾。 □

Theorem 6. X は G の最小安全支配集合である。

Proof. $k = l$ を示す。つまり、アルゴリズム 1 の while 文が $i = l+1$ の時に停止することを示す。これを示すために、$x_1 \leq x_{l+1}$ であることを示す。まず、$v_n \leq maxN(y_l)$ かつ $v_n \leq t_l$ であることを示す。Lemma 5 より、s_{l-1} は y_l を守らないので、y_l は s_l に守られている。よって、$(S_u \setminus \{s_l\}) \cup \{y_l\}$ は支配集合。y_l は x_l に隣接し、x_l は v_n を守っているので、$v_n \leq' maxN(y_l)$ $s_l \leq x_l \leq v_n \leq maxN(y_l)$ より s_l は v_n に隣接するので、x_l は v_n と隣接。つまり、$v_n \leq maxN(x_l)$ 以下、アルゴリズムの挙動により場合分けをする。
(1) $x_l = maxN(y_l)$ の場合
$v_n \leq maxN(x_l) = t_l$ となるので $k = l$ である。
(2) $x_l = maxN(maxN(y_{l-1})^+)$ の場合
S_u は安全支配集合なので、s_l は v_n を守る。よって、$S_l = (S_u \setminus \{s_l\}) \cup \{v_n\}$ は支配集合。$s^{'} = maxN(s_{l-1})^+$ とする。S_l は支配集合で、$s^{'}$ は s_{l-1} と隣接しないので、$s^{'}$ は v_n と隣接する。つまり、$v_n \leq maxN(s^{'})$ したがって、$t_l = maxN(maxN(x_{l-1})^+) \geq maxN(maxN(s_{l-1})^+) \geq maxN(s^{'}) \geq v_n$
以上より、$v_n \leq maxN(y_l)$ かつ $v_n \leq t_l$ が成り立つ。$y_{l+1} = t_l^+ \geq v_n^+ = v_1$, $x_{l+1} = maxN(y_{l+1}) \geq maxN(v_1) = x_1$ となり、$x_1 \leq x_{l+1}$ が成り立ち、$i = l+1$ の時、アルゴリズム 1 の while 文が終了するため、$k = l$ である。 □

Theorem 7. *Proper Circular Arc Graph* G に対し、$O(|V(G)|^2)$ 時間で最小安全支配集合を出力するアルゴリズムが存在する。

Proof. アルゴリズム 1 は $O(|V(G)|)$ で実行可能である。また、x_1 となる頂点の候補から開始頂点を限定したとき、最大でも $|V(G)|$ 回実行すれば良いので $O(|V(G)|^2)$ で求めることができる。

□

4 まとめと今後の課題

本研究では Proper Circular Arc Graph における最小安全支配集合問題が多項式時間で解けることを示した。今後の課題について、計算量を線形時間に減らせないか、関連問題である Total Dominating Set 問題等も Proper Circular Arc Graph に限定することで多項式時間で解けるかということが考えられる。

参考文献

[1] Toru Araki and Hiroka Miyazaki. Secure domination in proper interval graphs. *Discrete Applied Mathematics*, Vol. 247, pp. 70–76, 2018.

[2] Wen-Lian Hsu and Kuo-Hui Tsai. Linear time algorithms on circular-arc graphs. *Information Processing Letters*, Vol. 40, No. 3, pp. 123–129, 1991.

[3] E.J. Cockayne, P.J.P. Grobler, W.R. Grundlingh, J. Munganga, and J.H. Van Vuuren. Protection of a graph. *Utilitas Mathematica*, Vol. 67, p. 19 – 32, 2005.

[4] Peter J. Looges and Stephan Olariu. Optimal greedy algorithms for indifference graphs. *Computers & Mathematics with Applications*, Vol. 25, No. 7, pp. 15–25, 1993.

付録

Algorithm 1 提案アルゴリズム

Input: Proper Circular Arc Graph G, consecutive ordering for $V(G)$

Output: 最小安全支配集合 S

$maxN(v_1), ..., maxN(v_n)$ を計算
$y_1 \leftarrow v_1,\ x_1 \leftarrow maxN(y_1),\ t_1 \leftarrow maxN(y_1)$
$S \leftarrow \{x_1\},\ i \leftarrow 1$
while x_i が二周目でない **do**
  $i \leftarrow i+1, y_i \leftarrow t_{i-1}^{+}$
  if $maxN(y_{i-1}) < t_{i-1}$ **then**
    $x_i = maxN(maxN(y_{i-1})^{+})$
  else
    $x_i = maxN(y_i)$
  end if
  if $maxN(x_{i-1}) < x_i$ **then**
    $t_i = maxN(maxN(x_{i-1})^{+})$
  else
    $t_i = maxN(x_i)$
  end if
  $S \leftarrow S \cup \{x_i\}$
end while
return $S \setminus \{x_i\}$