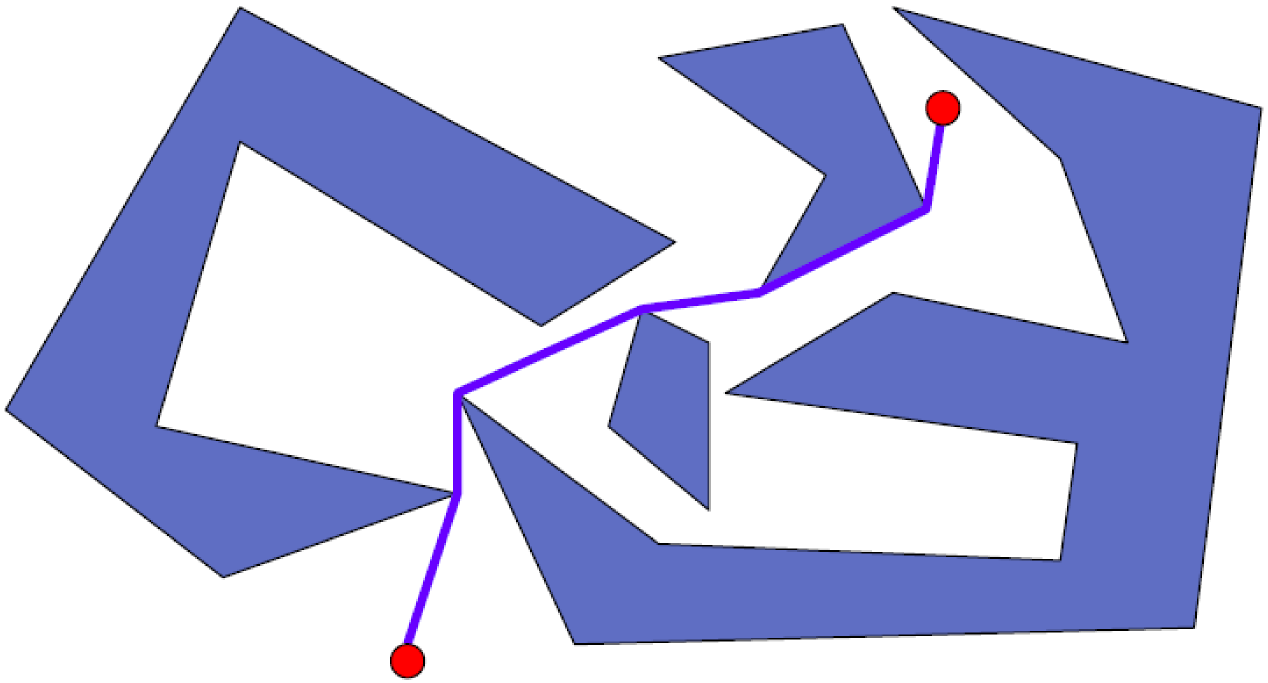
I482F: 実践的アルゴリズム特論
5　幾何的な最短経路問題

上原隆平
北陸先端科学技術大学院大学
uehara@jaist.ac.jp

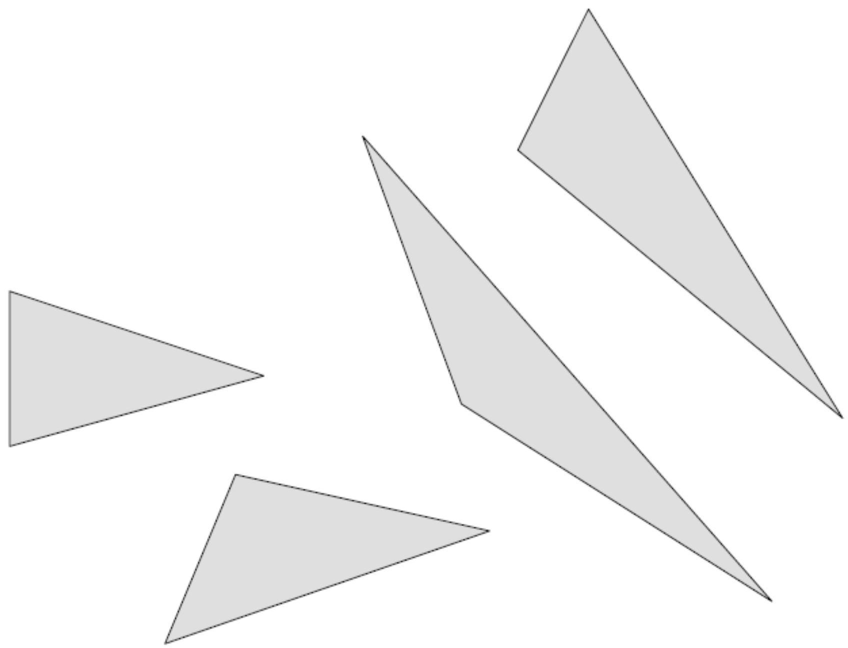
幾何的な最短経路問題

平面上に多数の多角形状の障害物が与えられているとき，
任意に指定した２点を結ぶ最短経路を求めよ.



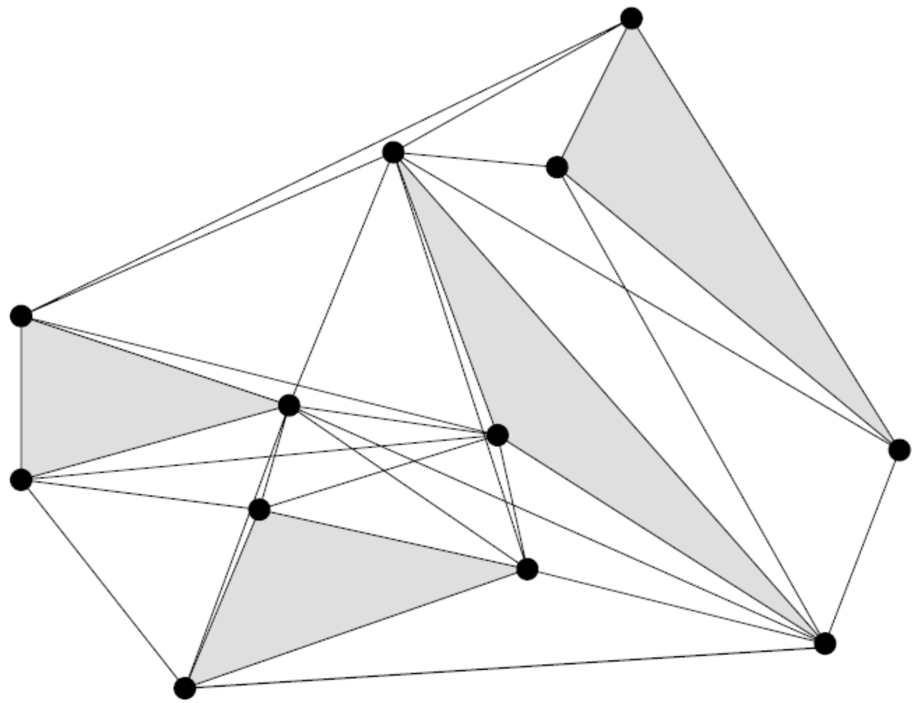
可視グラフの定義

平面上に多角形の障害物が多数与えられたとき，障害物の頂点をグラフの頂点とし，２頂点を結ぶ線分がどの障害物とも交差しないときに，対応する2頂点を辺で結ぶことによって定まるグラフを可視グラフという.



可視グラフの定義

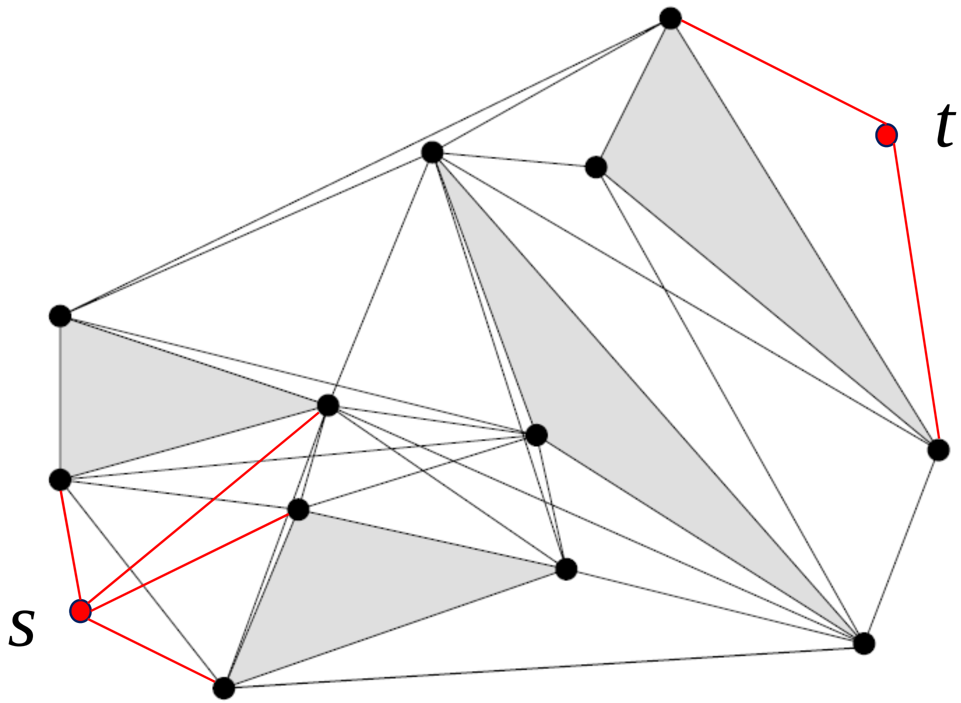
平面上に多角形の障害物が多数与えられたとき，障害物の頂点をグラフの頂点とし，２頂点を結ぶ線分がどの障害物とも交差しないときに，対応する2頂点を辺で結ぶことによって定まるグラフを可視グラフという.



可視グラフの定義

平面上に多角形の障害物が多数与えられたとき，障害物の頂点をグラフの頂点とし，2頂点を結ぶ線分がどの障害物とも交差しないときに，対応する2頂点を辺で結ぶことによって定まるグラフを可視グラフという.

始点sと
目的地tに
対応する
頂点を加える

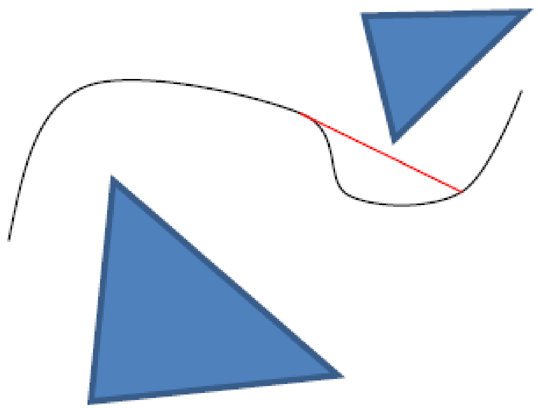


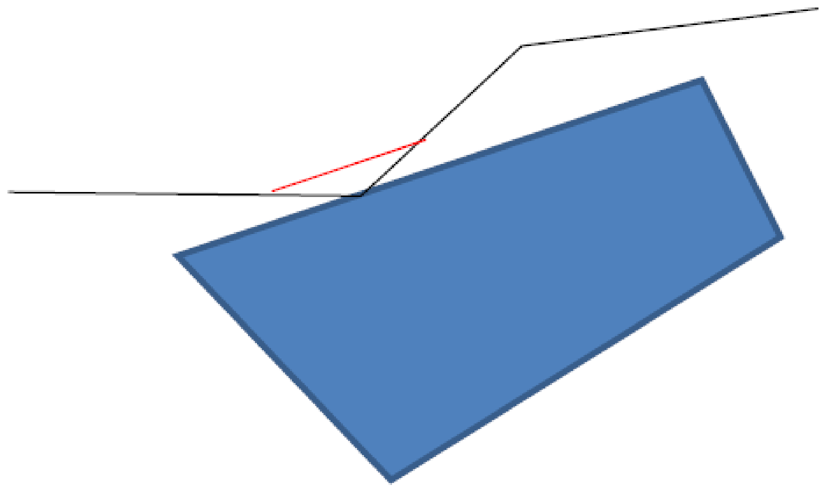
補題　障害物を避けた平面上の2点を結ぶ最短経路を求めるには、可視グラフ上の最短経路を求めればよい。

[証明] 障害物を避けた最短経路が、障害物の頂点でしか曲がらないことを示せばよい。

最短経路が障害物の頂点以外のところで曲がったと仮定して、矛盾を示す。

障害物から離れたところで曲がると、常に近道が存在する。また、辺の途中で折れ曲がっても近傍で近道ができる。よってどちらも最短路であるという仮定に矛盾する。





2点間の幾何的な最短経路を求めるアルゴリズム

入力:　多角形の障害物、始点 s、目的地 t

1. 多角形の障害物に対する可視グラフを構成する.
2. 可視グラフに始点と目的地点に対応する頂点を挿入.
3. 追加した2頂点からの可視辺を挿入.
4. 構成された重み付きグラフ上で始点から目的地点までの最短経路を求める.
5. グラフの最短経路を平面上の経路に変換して出力.

アルゴリズムの解析のポイント:

- 可視グラフを構成するアルゴリズムは？その計算時間は？
- 可視グラフ上で最短経路を求めるアルゴリズムの計算時間は？
- 全体の計算時間は？

可視グラフを構成するアルゴリズム

素朴な方法:

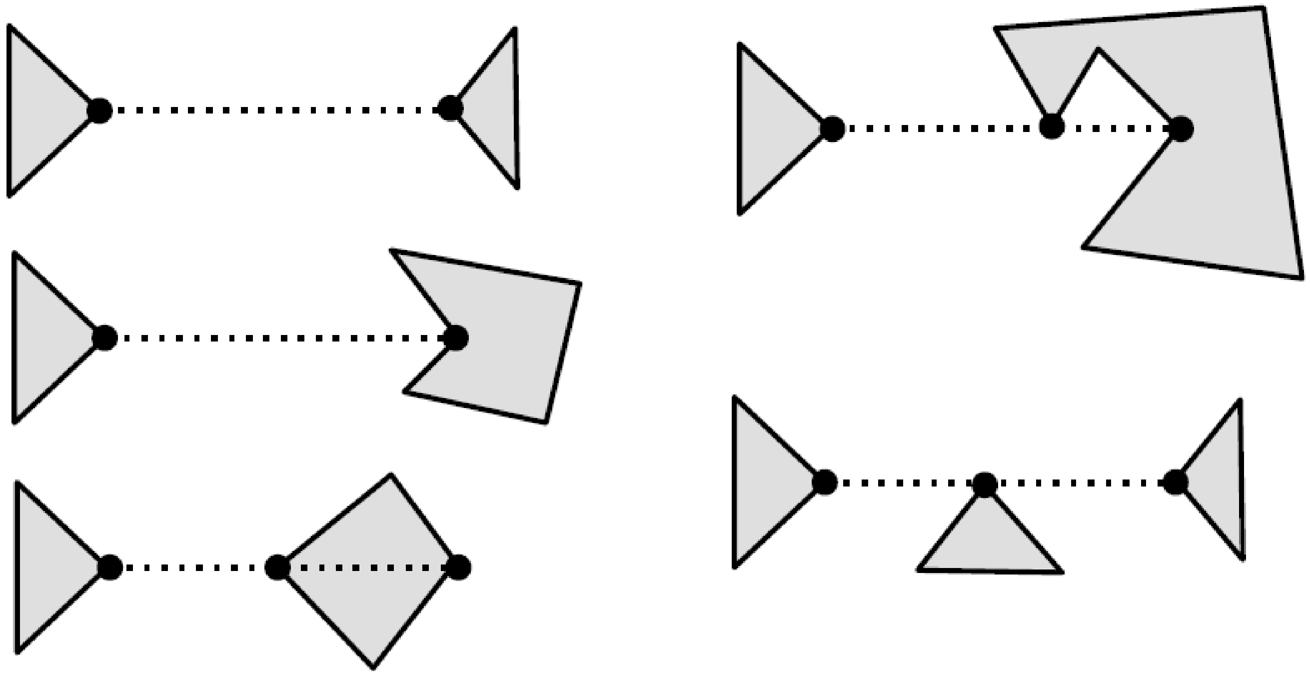
各頂点対について、それらの頂点を結ぶ線分が障害物の内部と交差するかどうかを判定

障害物の頂点数を n とすると、障害物の辺数も O(n)。したがって1本の線分が障害物の内部と交差するかどうかは O(n) 時間で判定できる。

注意:基本的には障害物の辺との交差の有無で可視かどうかを判定できるが、微妙な場合もある。

可視グラフを構成するアルゴリズム

注意：基本的には障害物の辺との交差の有無で可視かどうかを判定できるが、微妙な場合もある。



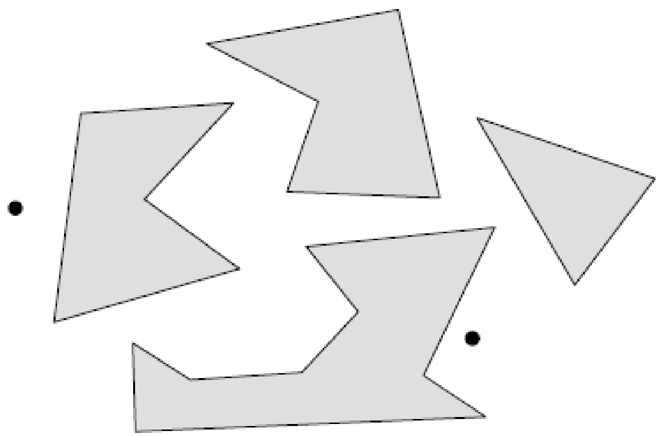
2頂点を結ぶ線分が途中に別の線分を横切ったり，頂点を含んだりすれば，その2頂点は互いに可視ではないとする．つまり，2頂点に接続する合計4辺以外とは接触も交差もしないことが可視辺を定めルール．頂点対は$O(n^2)$個なので，この判定は全体で$O(n^3)$時間でできる．

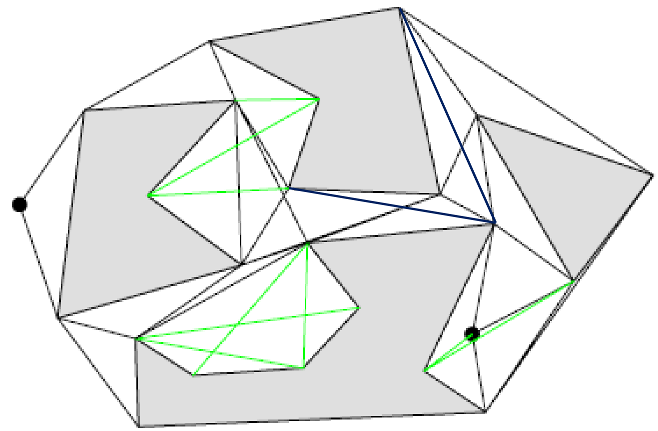
(やや余談)現実的な側面からの簡単化/高速化

実際の場面では、すべての頂点対に対して可視辺を生成する必要はない。

1. 最短経路が折れ曲がる可能性があるのは、凸頂点だけ（Why?）

∴障害物の凸点と始点と目的地点だけをグラフの頂点とすれば十分。



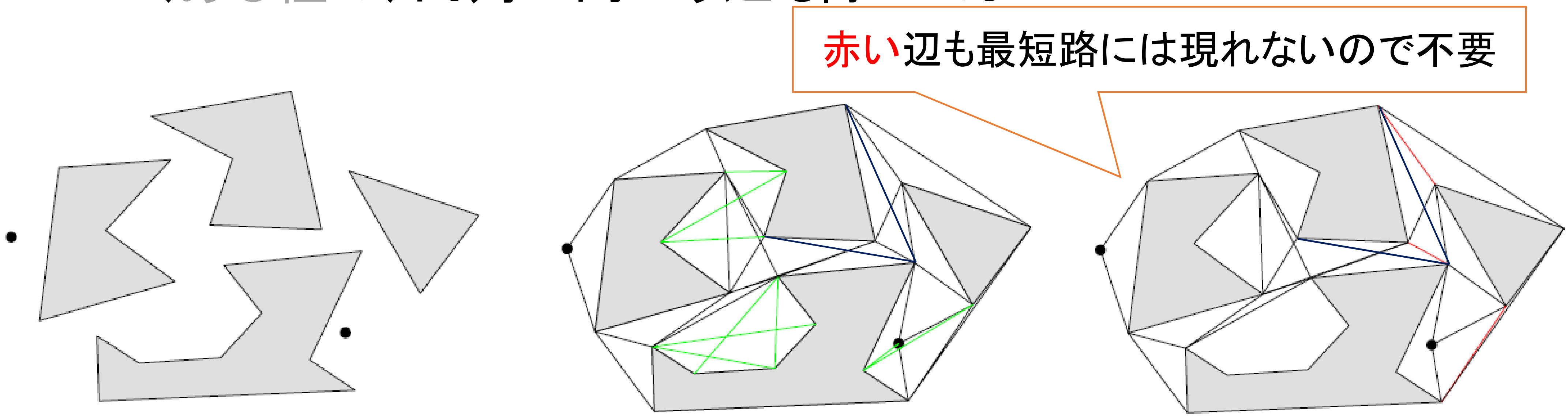


緑の辺は可視辺であるが、最短路には現れないので、作る必要がない。

(やや余談)現実的な側面からの簡単化/高速化

実際の場面では、すべての頂点対に対して可視辺を生成する必要はない。

1. 最短経路が折れ曲がる可能性があるのは、凸頂点だけ
2. (ある種の)内角に向かう辺も除いてよい



素朴な可視グラフ構成アルゴリズムの実行時間解析

素朴な方法:

```
1.  For each 凸頂点 u do
2.    for each 凸頂点 v do {
3.      uv を結ぶ線分を L とする
4.      cross = False
5.      for 障害物の各辺 e do {
6.        if e の端点は u でも v でもない and e はLと交差か接触
7.          then cross = True としてループから脱出
8.      }
9.      if cross = False then グラフに辺(u,v)を挿入
10.   }
11. }
```

O(n)

O(n)

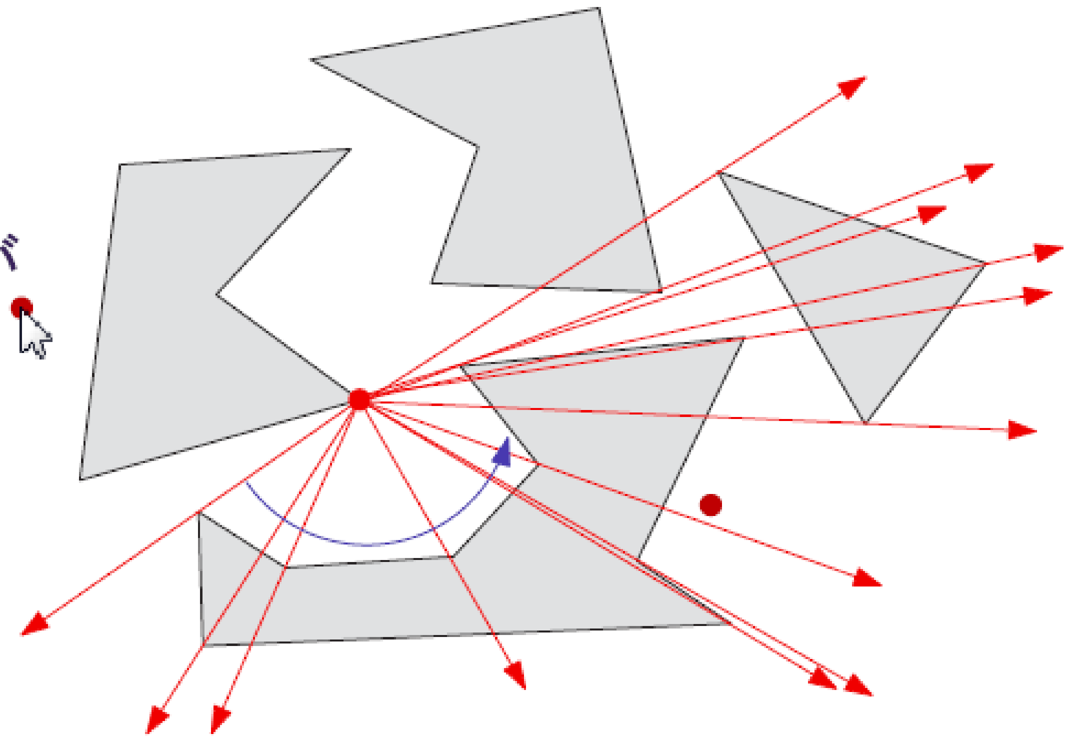
O(n)

計算時間は
明らかに O(n^3)。
改善できるか？

平面走査法に基づく計算時間の短縮

各凸頂点 u に関して次の操作を行う.
他のすべての頂点 v を u に関する偏角でソートし,
uからそれらの頂点を通る半直線を偏角順に調べる.

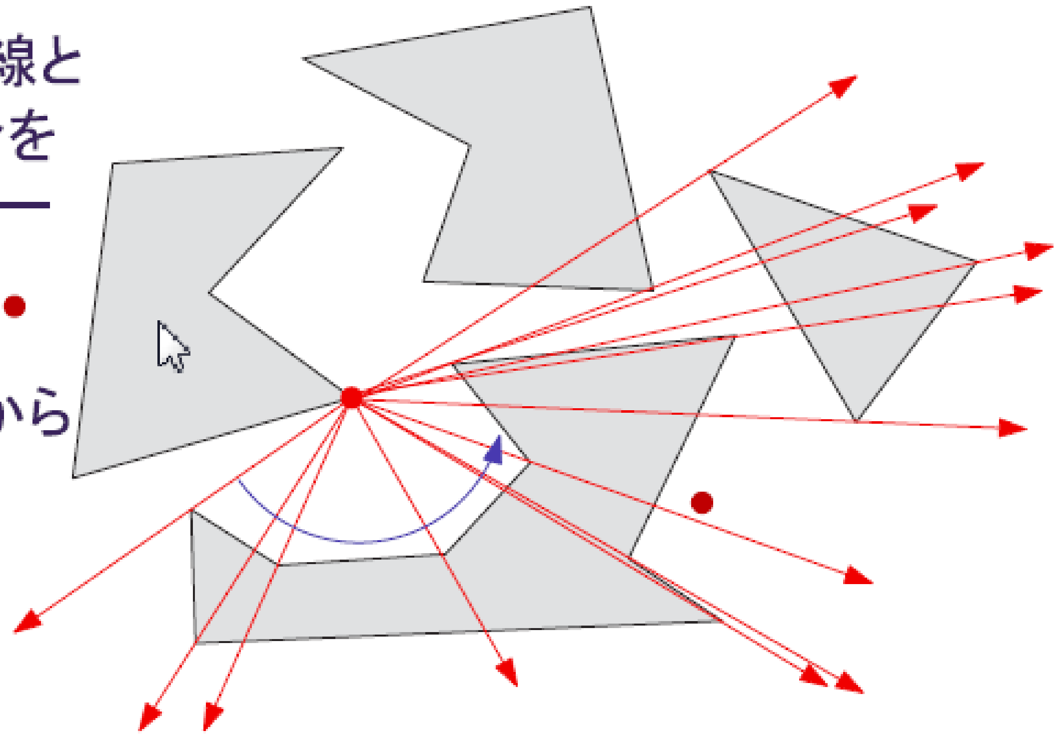
どの頂点でも
半直線が最初に
交差する辺だけが
頂点uから見える.



平面走査法に基づく計算時間の短縮

各凸頂点 u に関して次の操作を行う.
　他のすべての頂点 v を u に関する偏角でソートし,
　uからそれらの頂点を通る半直線を偏角順に調べる.

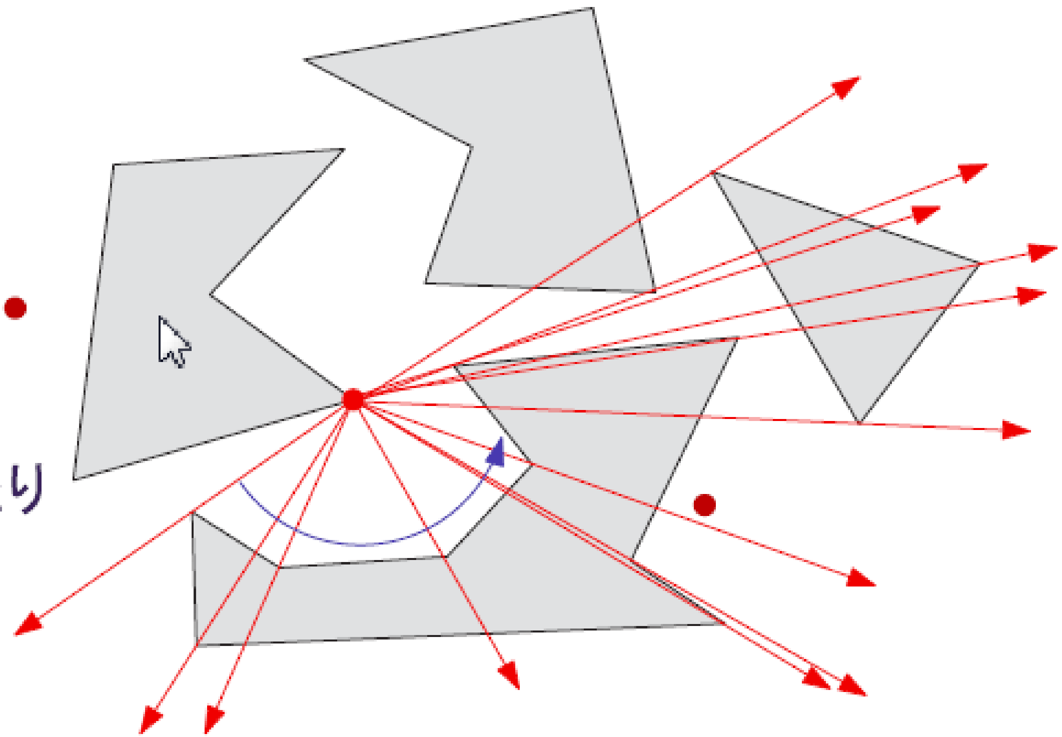
頂点uからの半直線と交差する辺の集合を優先順位付きキューで管理する.
新たに挿入された辺の端点 v が u から最も近いところにあれば, (u, v)を辺として挿入する.



平面走査法に基づく計算時間の短縮

各凸頂点 u に関して次の操作を行う.
　他のすべての頂点 v を u に関する偏角でソートし,
　uからそれらの頂点を通る半直線を偏角順に調べる.

頂点のソートは
O(n log n)時間.
各点での処理も
良いデータ構造を
使うと，毎回
O(log n)時間.
よって，1頂点あたり
O(n log n)時間.
全体では,
O(n^2log n)時間.



幾何的最短経路問題に対する理論的な研究成果

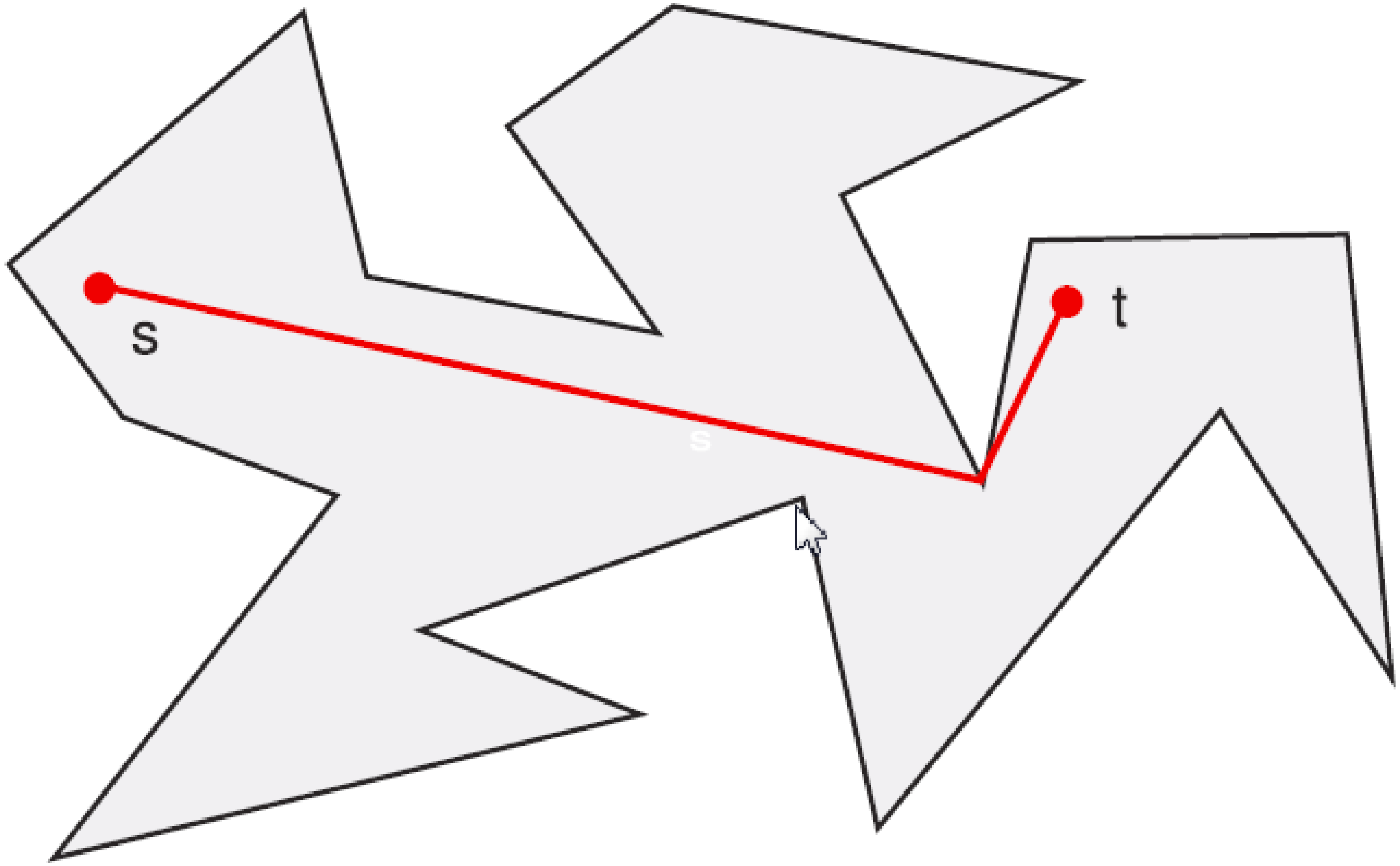
T. Asano, T. Asano, L.J. Guibas, J. Hershberger, and H. Imai: "Visibility of disjoint polygons, " Algorithmica, 1:49-63, 1986.

$O(n^2)$時間のアルゴリズム.

J. Hershberger and S. Suri: "An optimal algorithm for Euclidean shortest paths in the plane," SIAM J. Computing, 28:2215-2256, 1997.

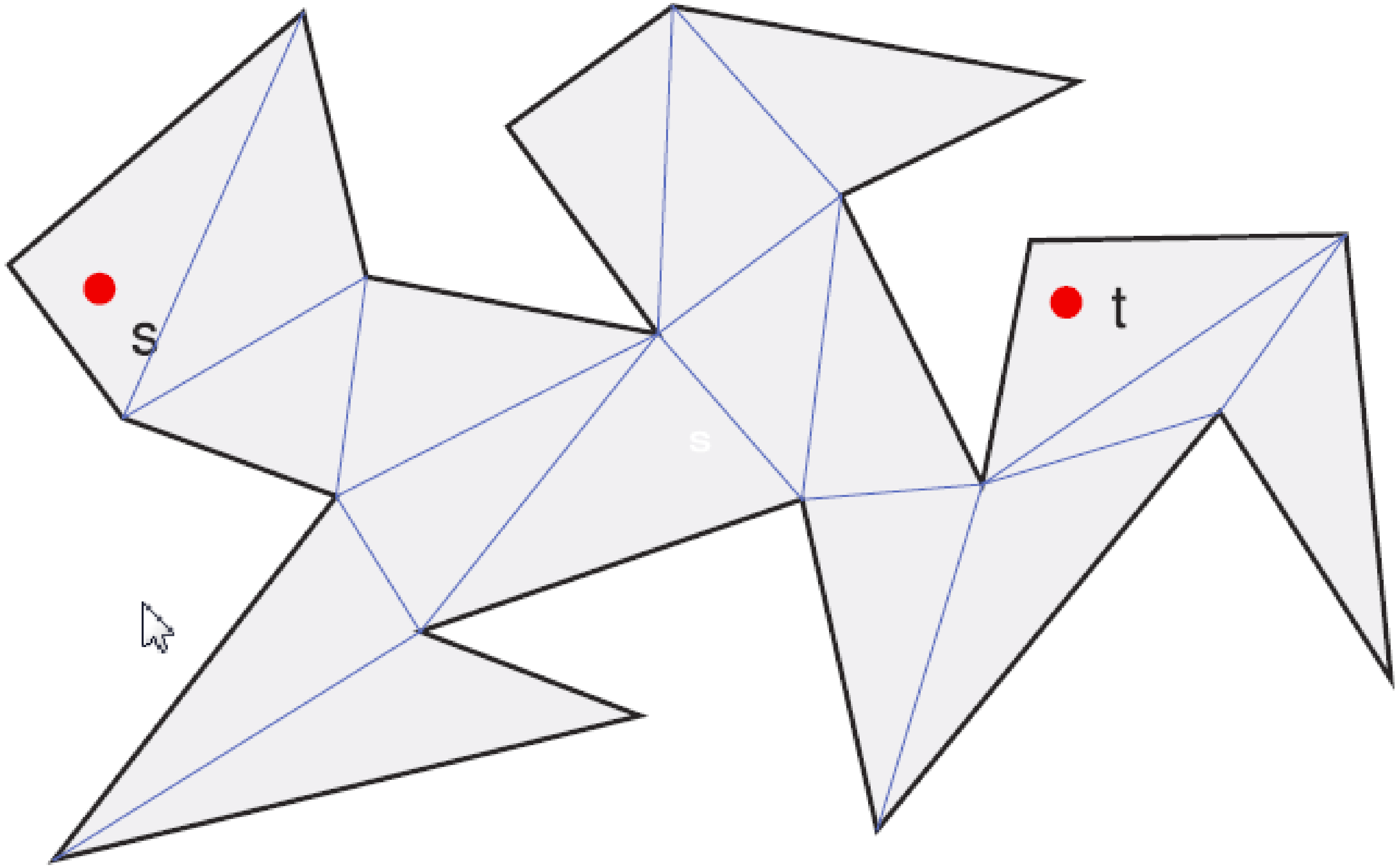
$O(n \log n)$時間のアルゴリズム.
これが理論的に最適.

特殊な場合：
単純な多角形の内部での最短経路



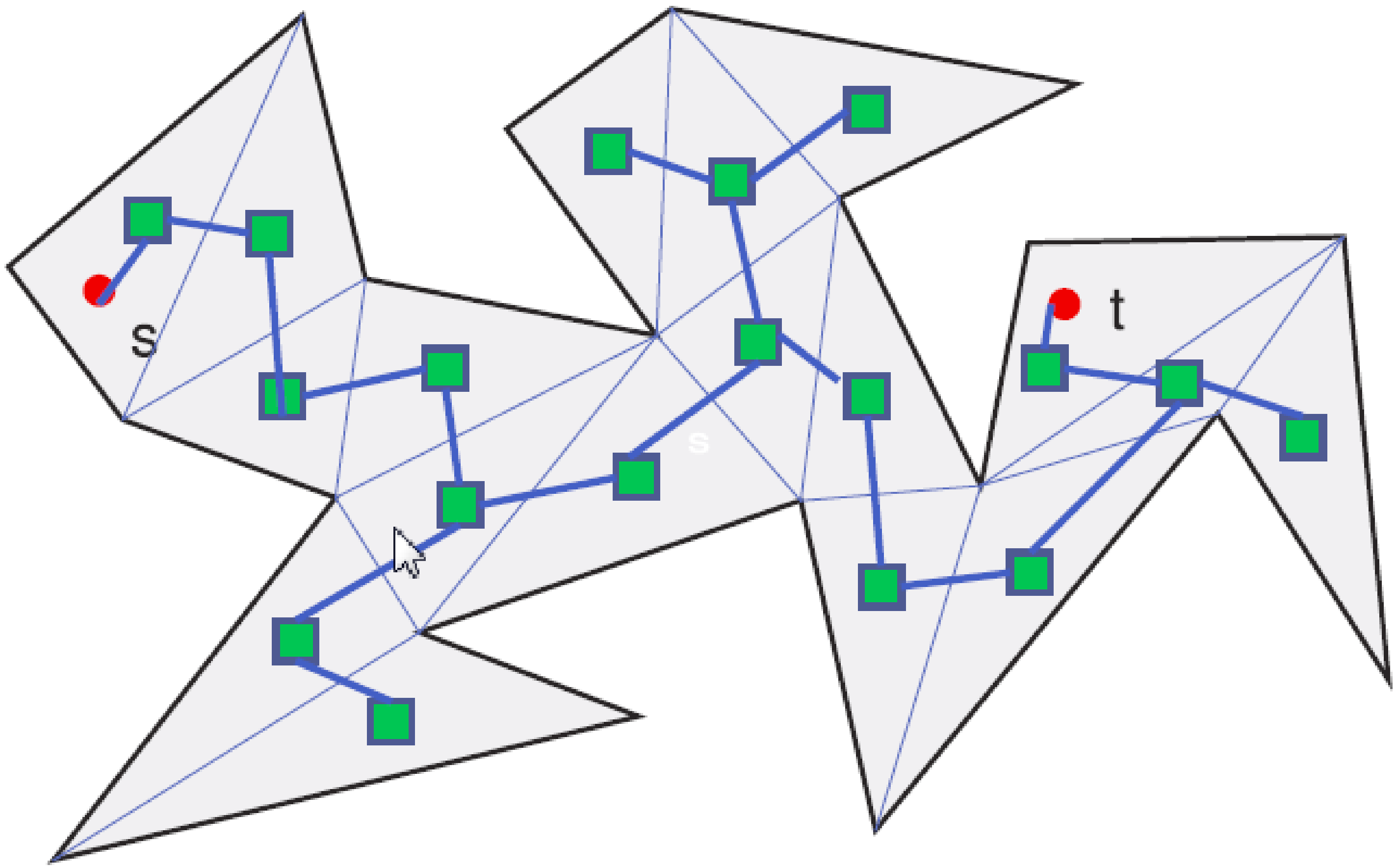
単純な多角形の内部の２点を結ぶ最短経路

特殊な場合：
単純な多角形の内部での最短経路



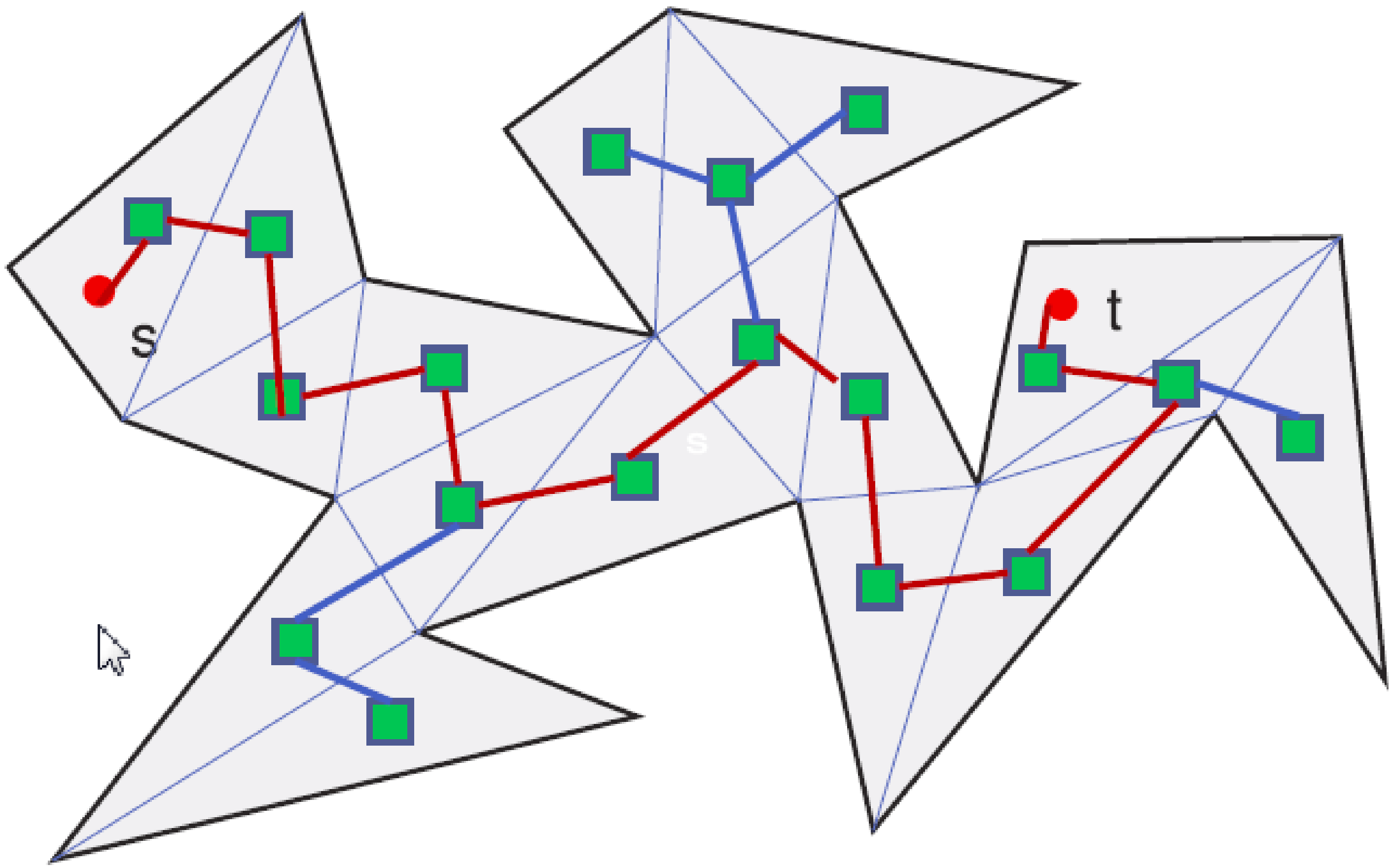
単純な多角形の内部を三角形に分割

特殊な場合:
単純な多角形の内部での最短経路



三角形の隣接関係を表すグラフを構成➔ 木となる.

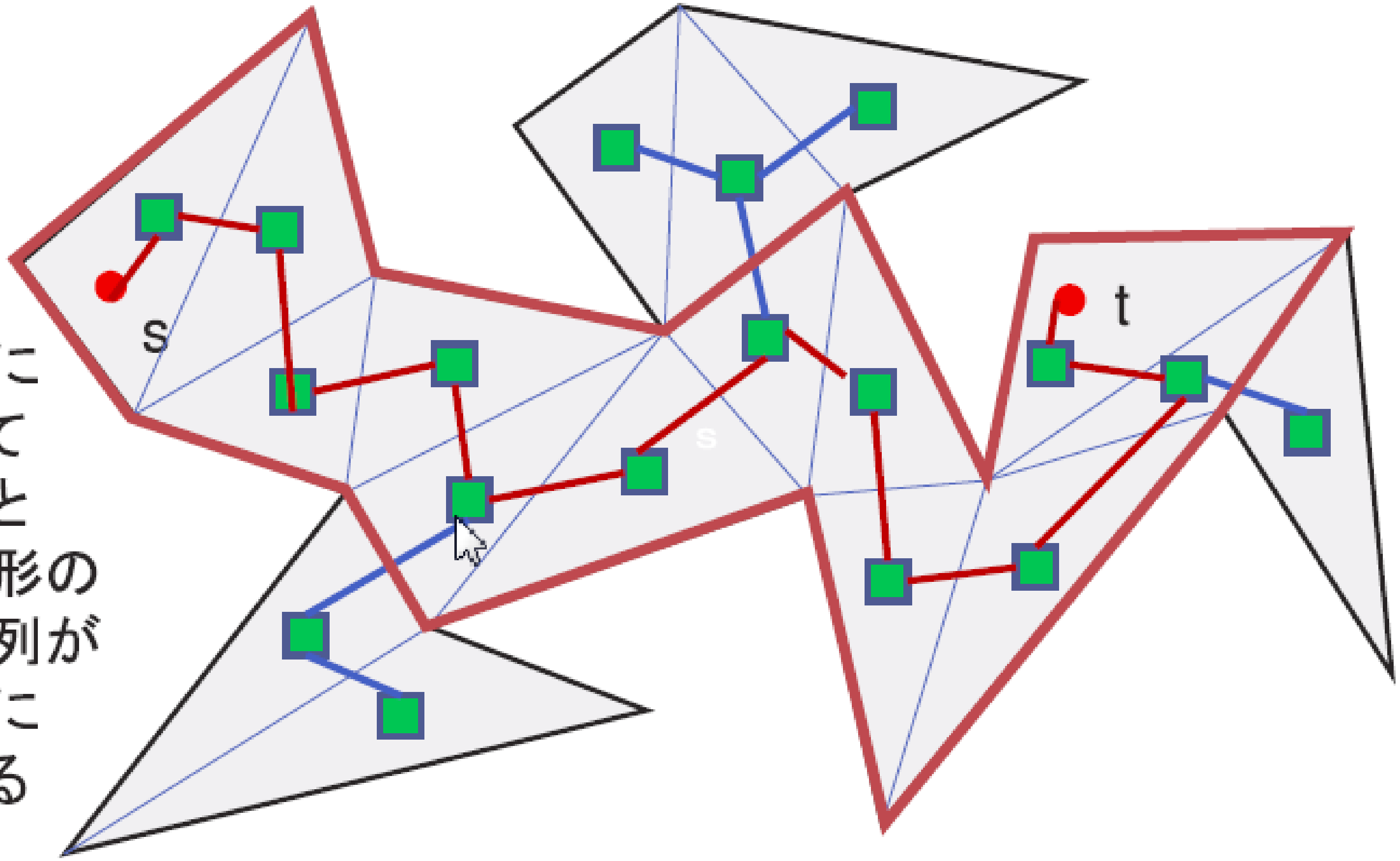
特殊な場合：
単純な多角形の内部での最短経路



木の上では2点間の経路はユニークに定まる

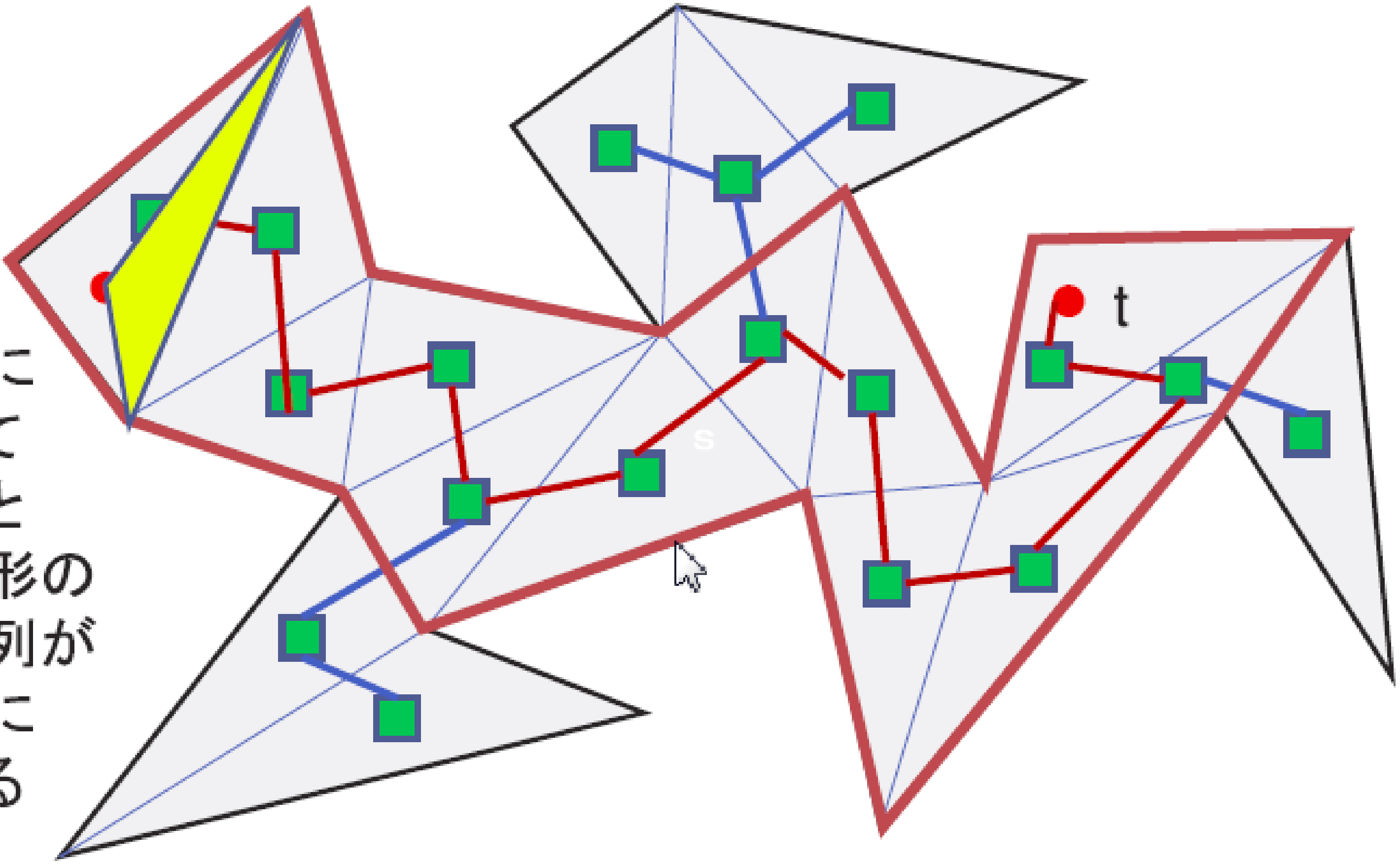
特殊な場合：
単純な多角形の内部での最短経路

経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

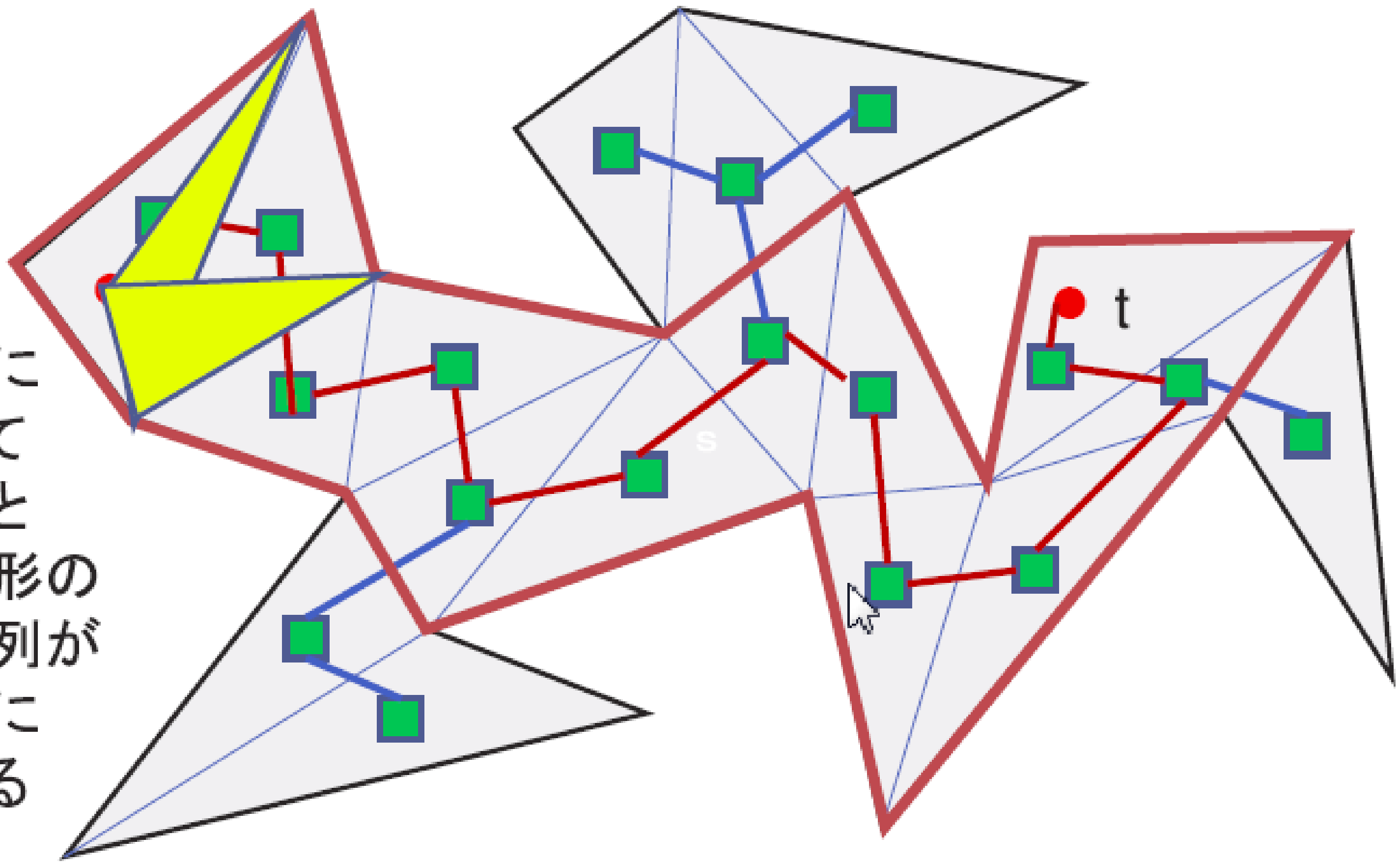


経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる

木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

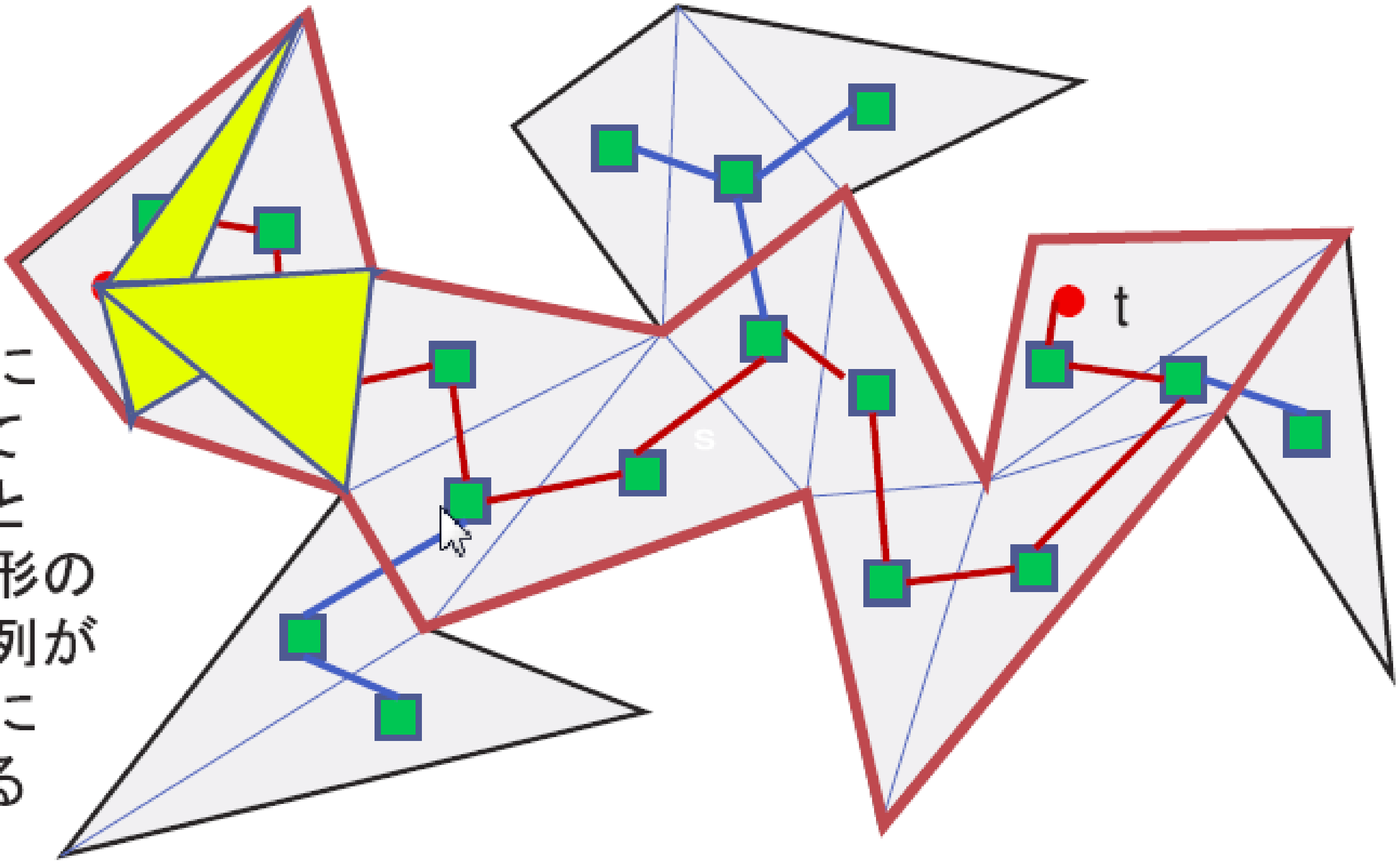
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

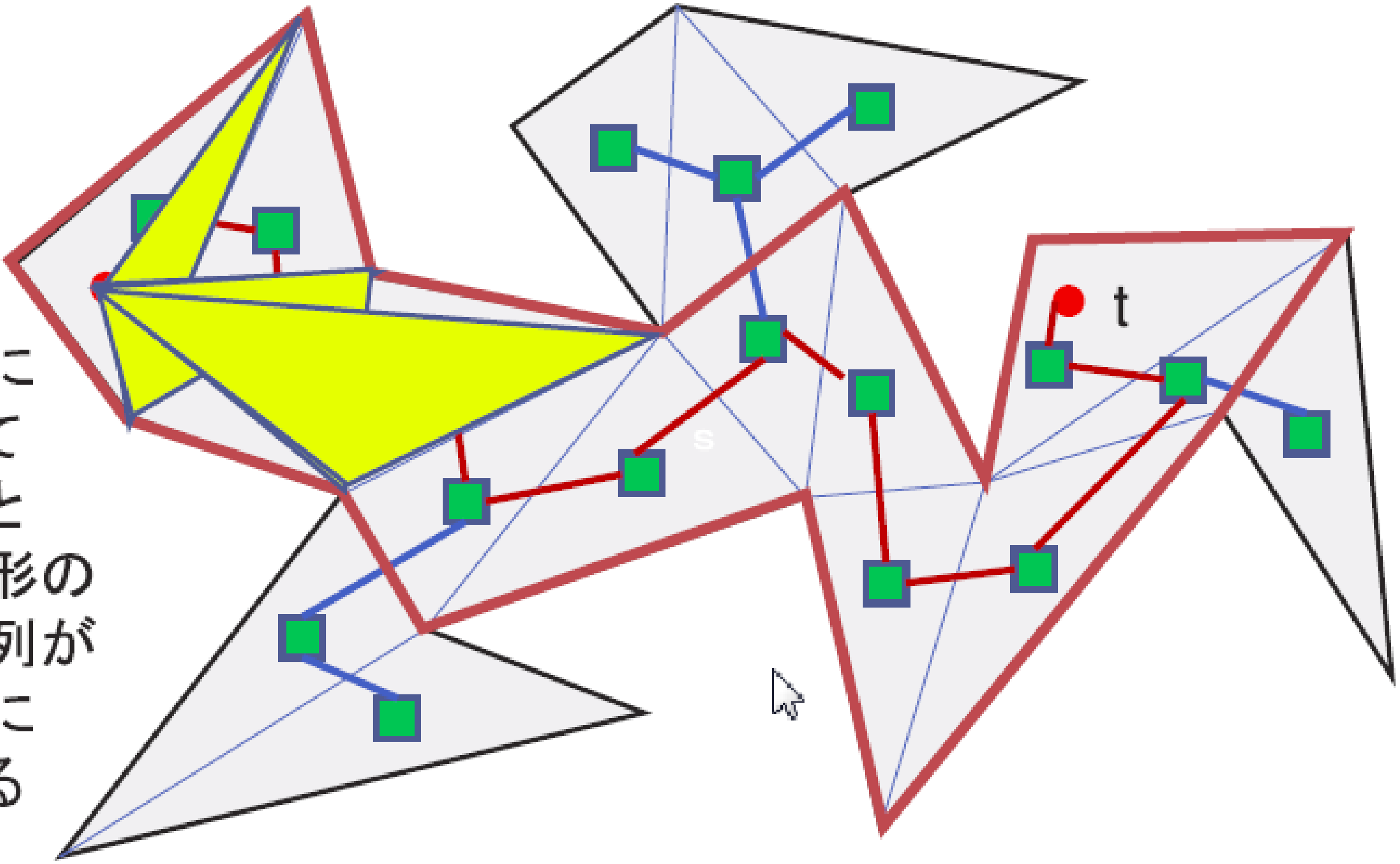
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

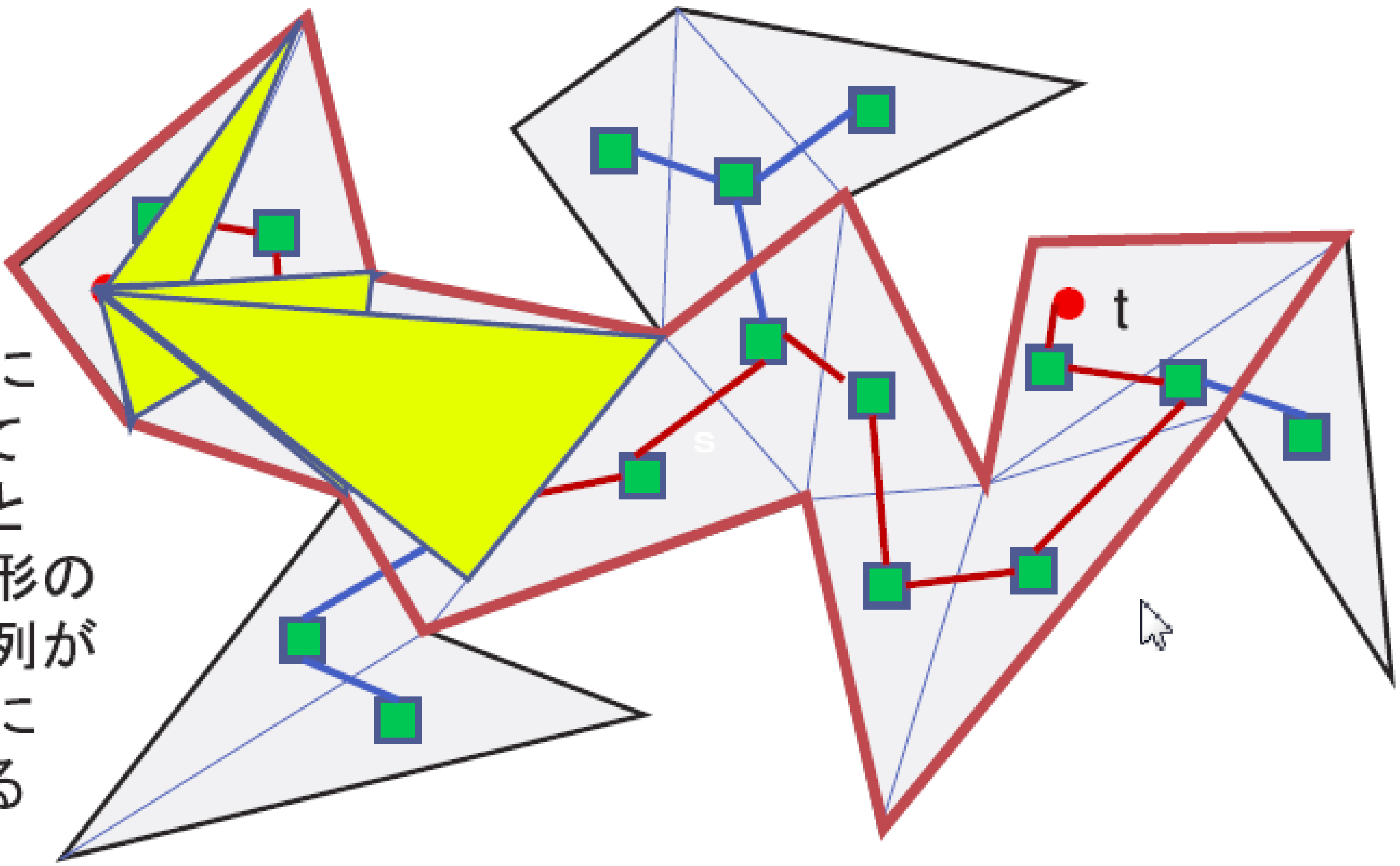
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

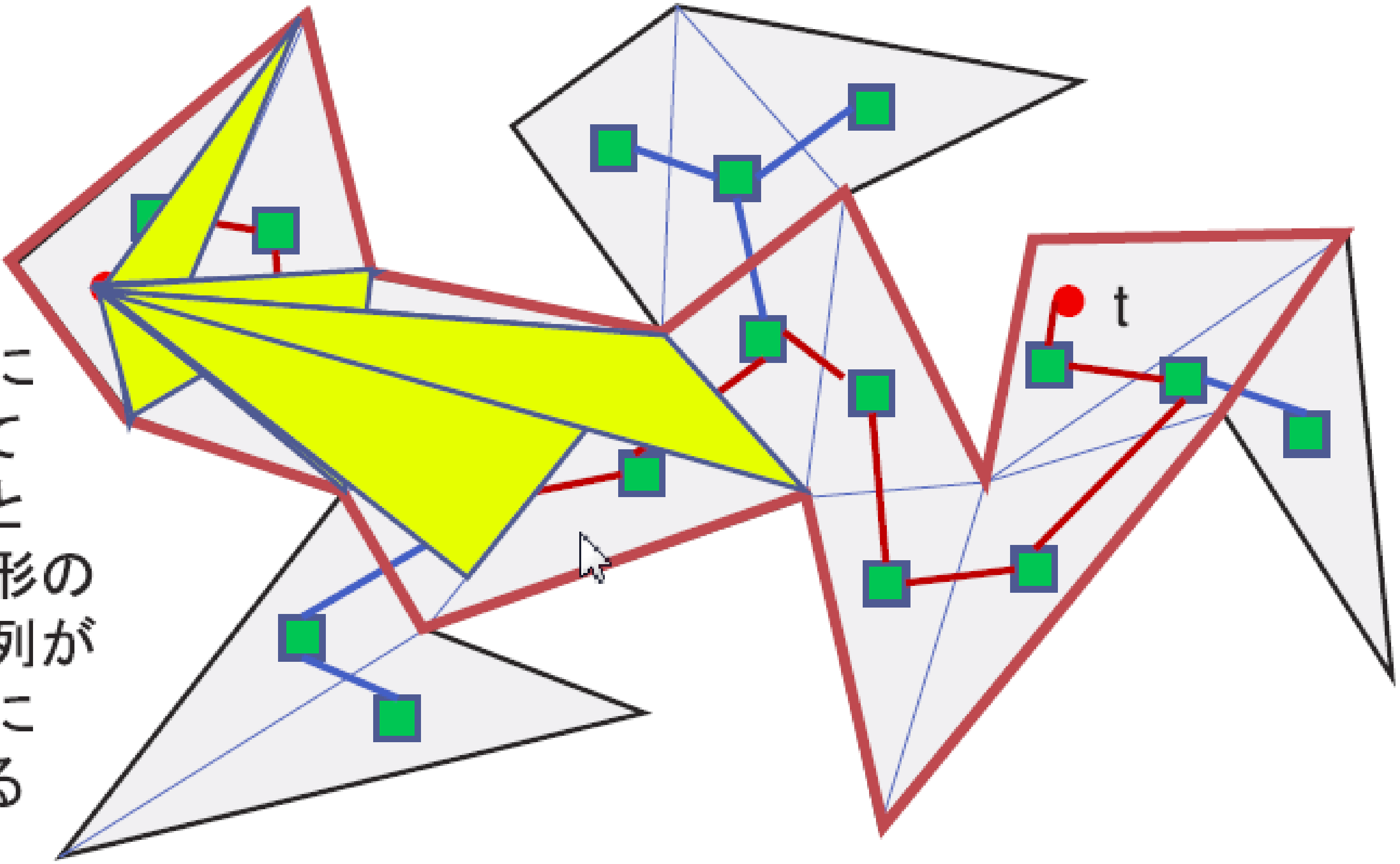
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

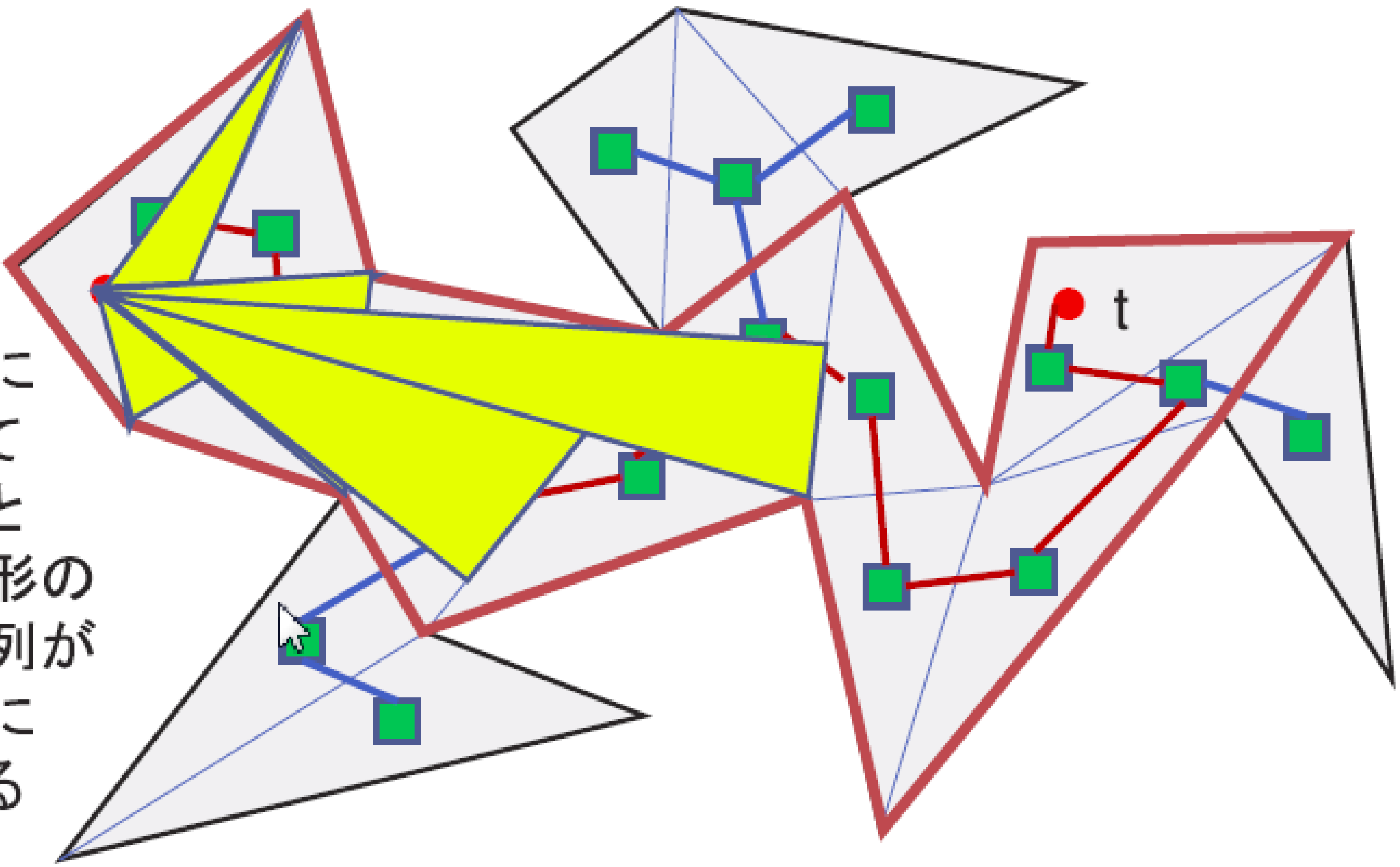
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

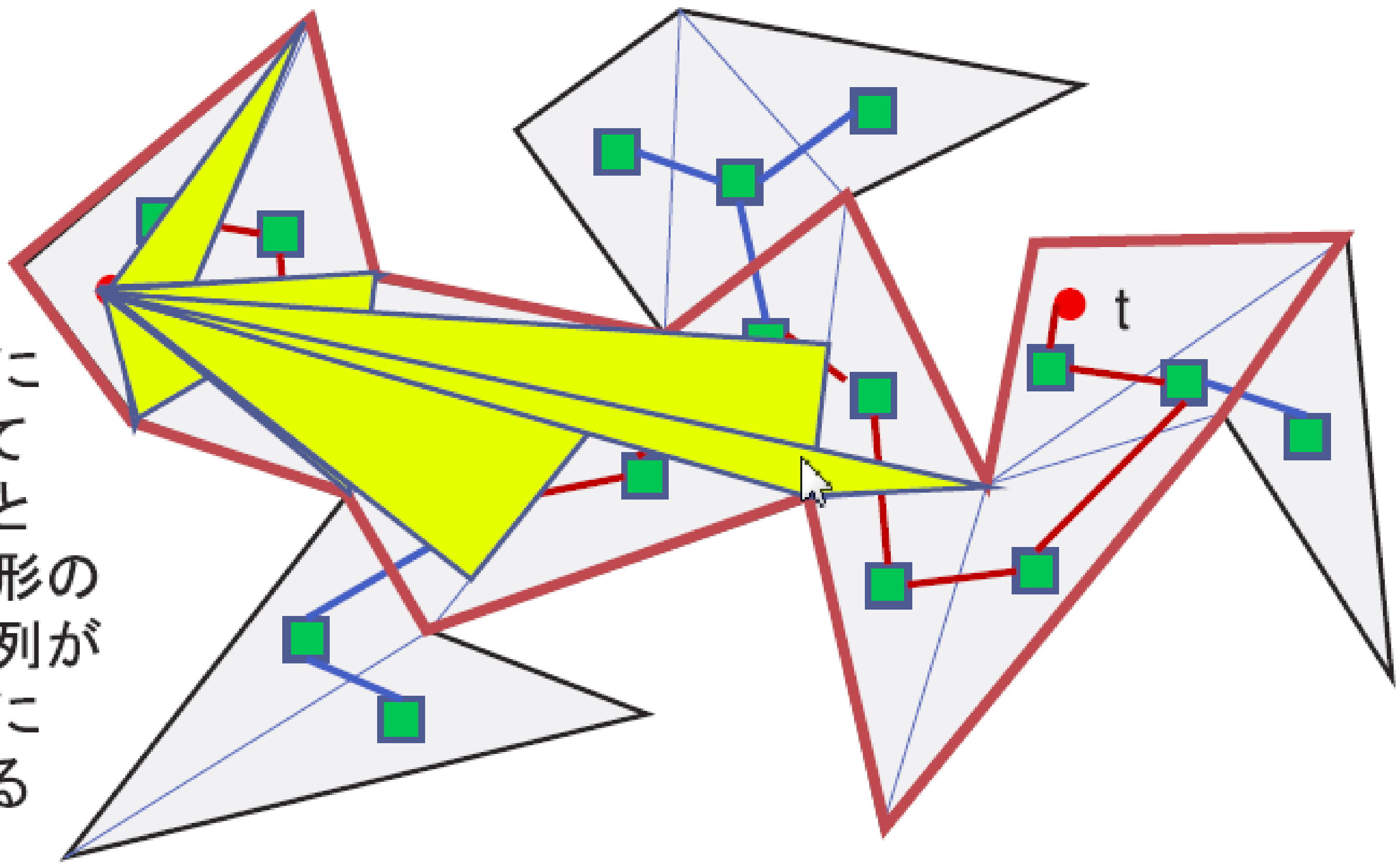
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

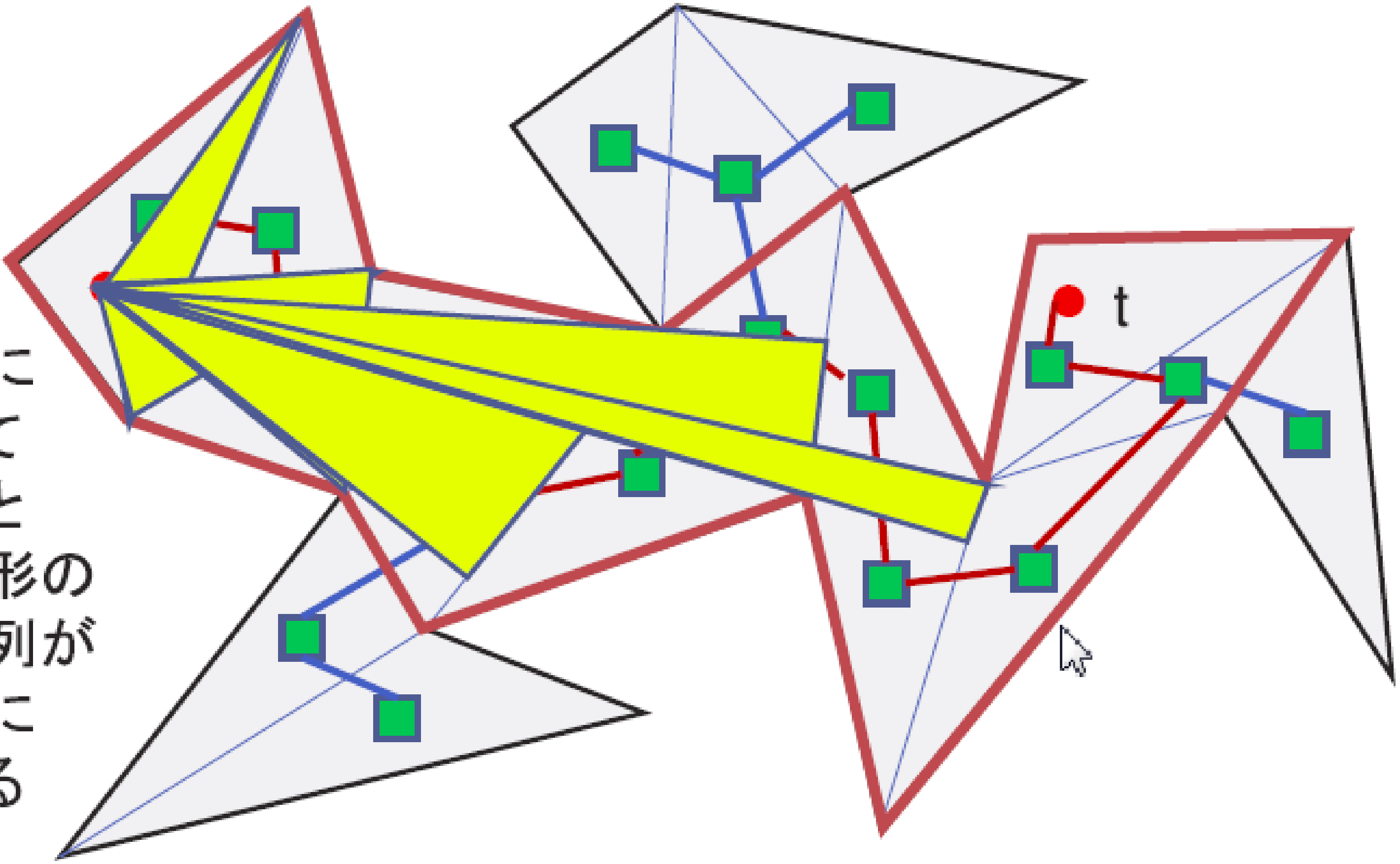
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

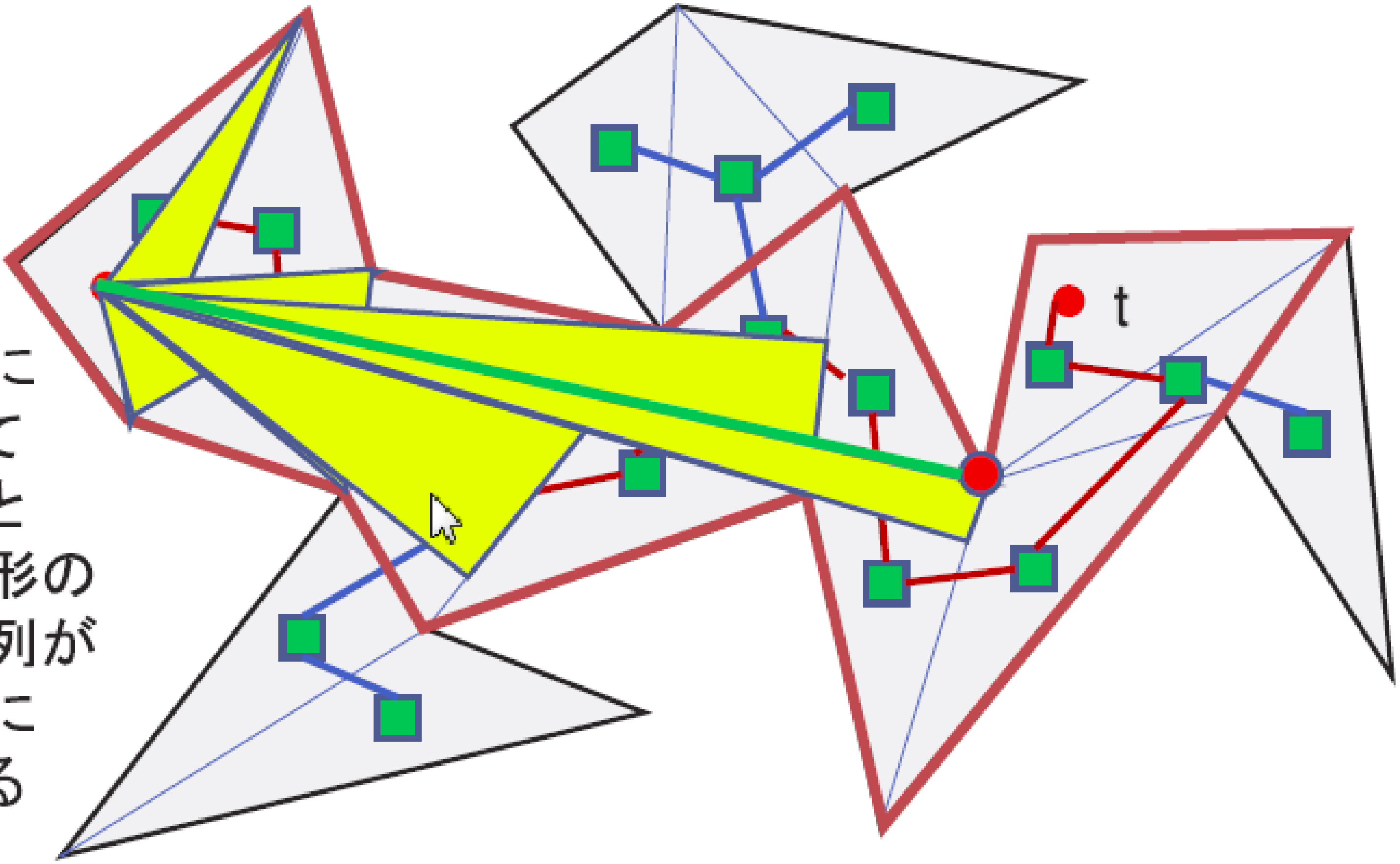
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

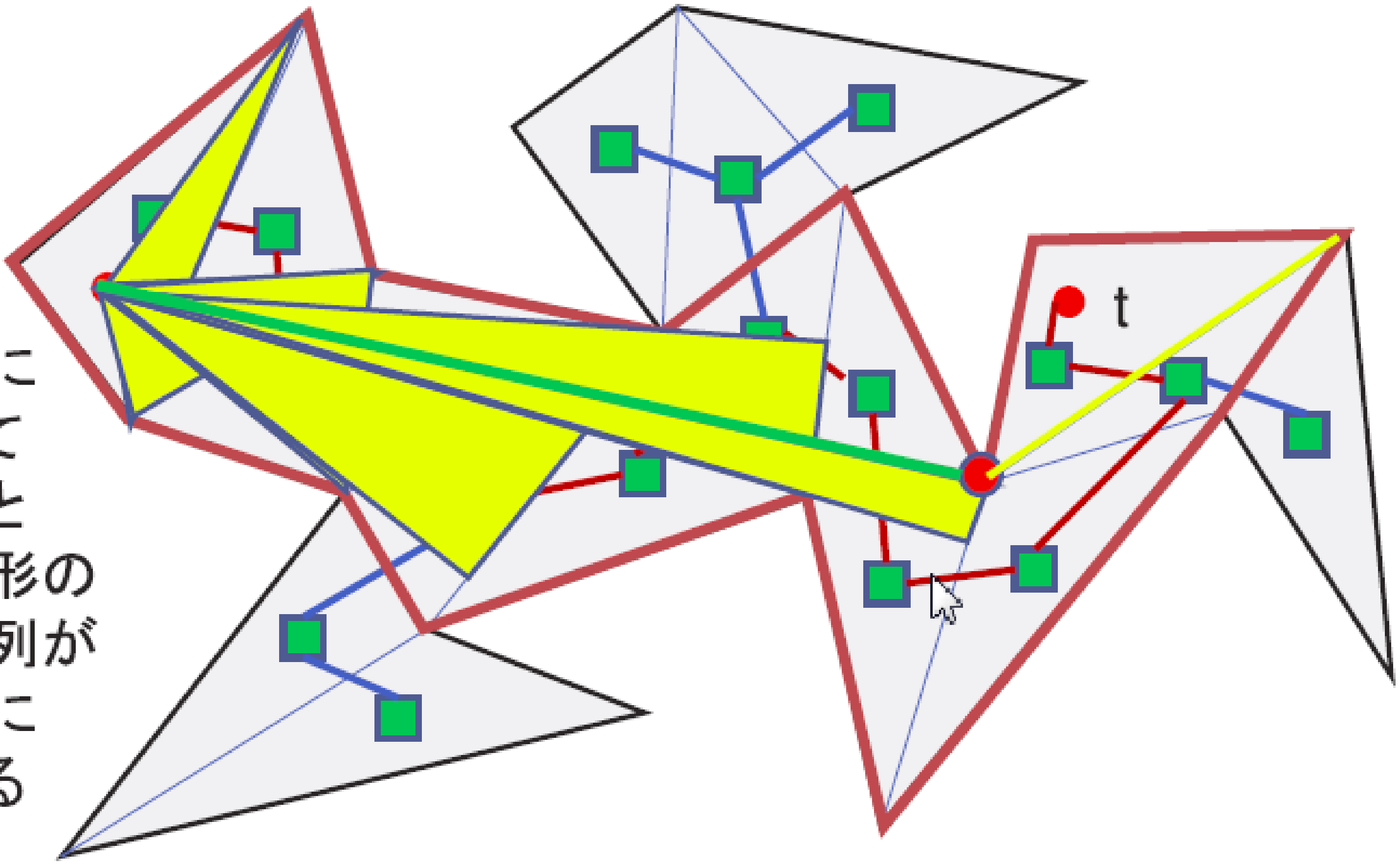
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

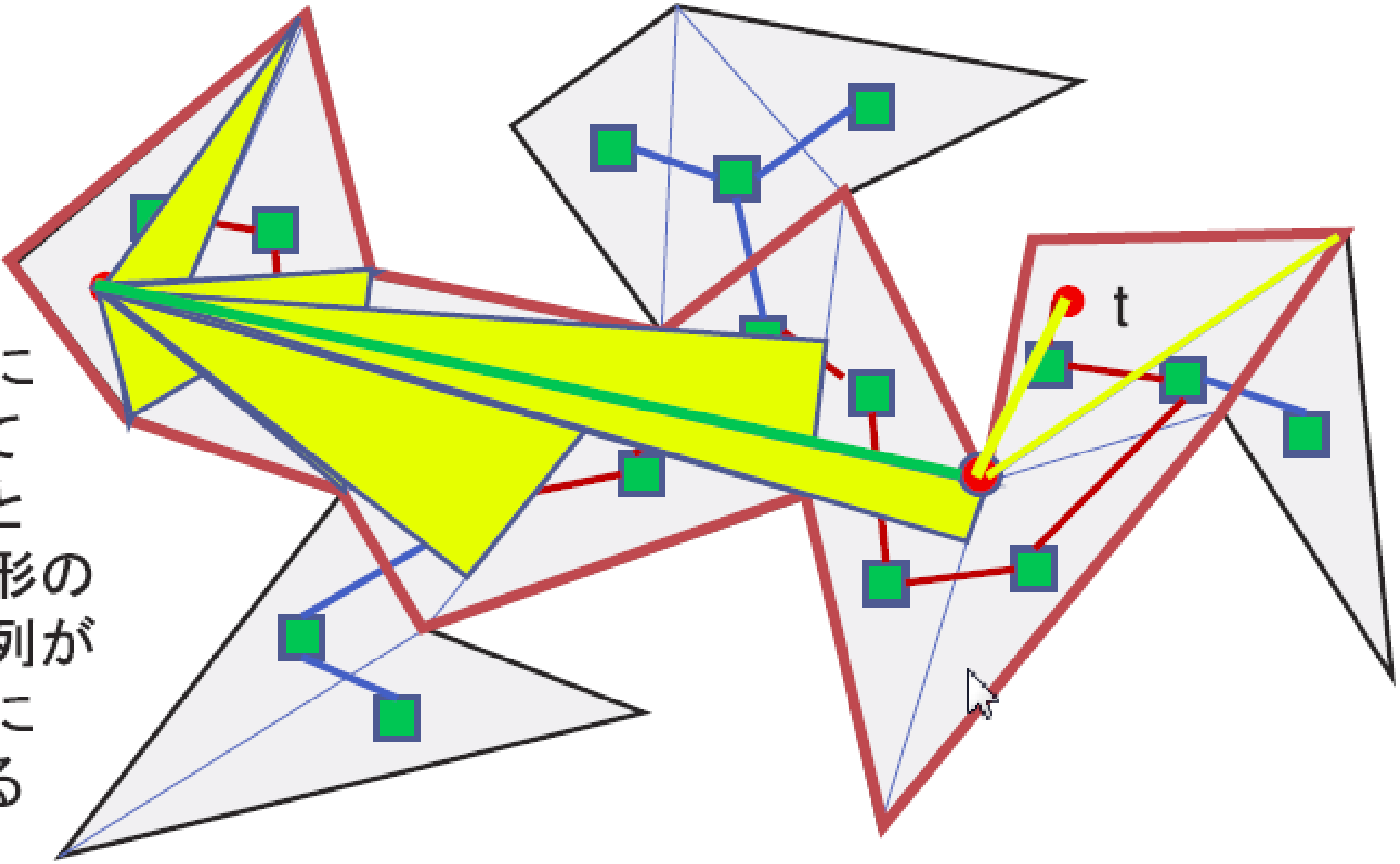
経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

特殊な場合：
単純な多角形の内部での最短経路

経路に
沿って
進むと
三角形の
辺の列が
自然に
定まる



木上での２点間の経路に対応する部分多角形

アルゴリズムのまとめ

入力：単純な多角形Pと，Pの内部の任意の2点s, t

1. 多角形Pの内部を三角形に分割．
2. 三角形の隣接関係を表すグラフを構成．木となることに注意．
3. s, tをそれらを含む三角形の頂点と辺で結ぶ．
4. このようにして作られた木の上でs-t間の経路を求める．
5. この経路に相当する三角形の列，さらには経路が通過する
   三角形の辺の列を求める．
6. 始点sから最初の辺に対する視角を求める．
7. 上記の辺の列に従って，次々と辺に対する視覚を求める．
8. 視覚が0になれば，その頂点vを先の出発点uと結ぶ辺(u, v)
   を出力し，出発点をvに置き換えてさらに先に進む．
9. 目的地に到達すれば終了．

アルゴリズムの計算時間

このアルゴリズムでは，最初にO(n log n)時間で多角形を三角形分割した後，O(n)個の辺に対して視角をO(1)時間で求めることを繰り返す．

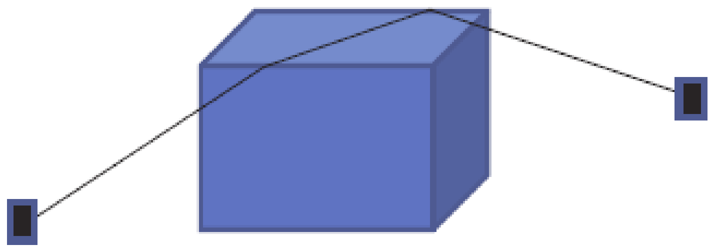
よって，総計算時間はO(n log n)時間．

実は，B. Chazelleによる難解なアルゴリズムを用いると線形時間で三角形分割が可能．よって，アルゴリズム全体もO(n)時間で実行可能である．

幾何的最短経路問題の一般化

3次元空間への拡張

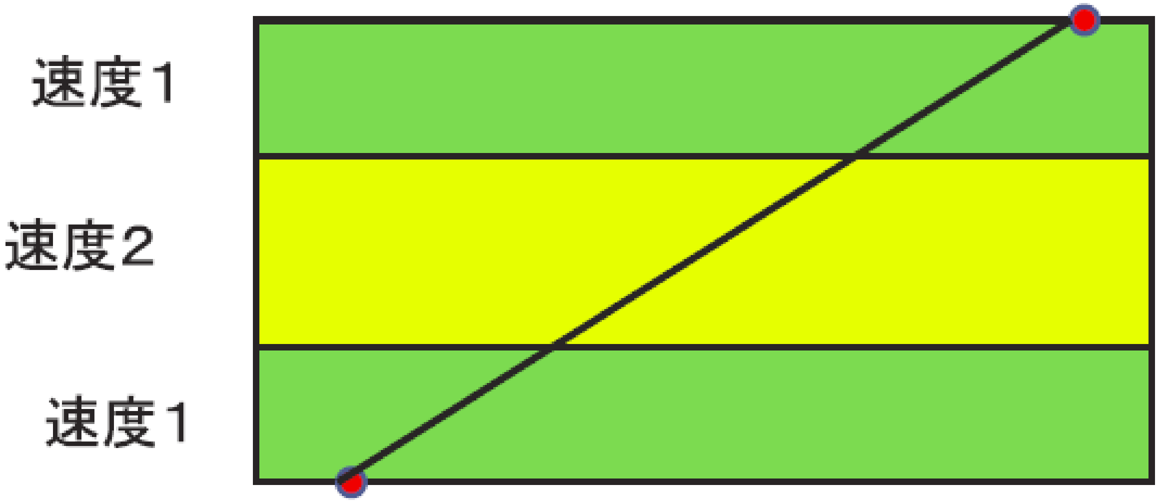
3次元空間における最短経路問題はどの程度難しいか.
障害物は多面体の集合.
最短経路問題はNP-困難であることが証明されている.
（多面体の辺上の任意の点で折れ曲がることがあるから）
ただし, 1個の凸多面体の表面上での最短経路問題は
多項式時間で解ける（展開図の利用）.



幾何的最短経路問題の一般化

領域によって速度が異なる場合への拡張

２次元平面で領域によって速度が異なる場合に2点間の（時間に関する）最短経路を求める問題は難しいか？

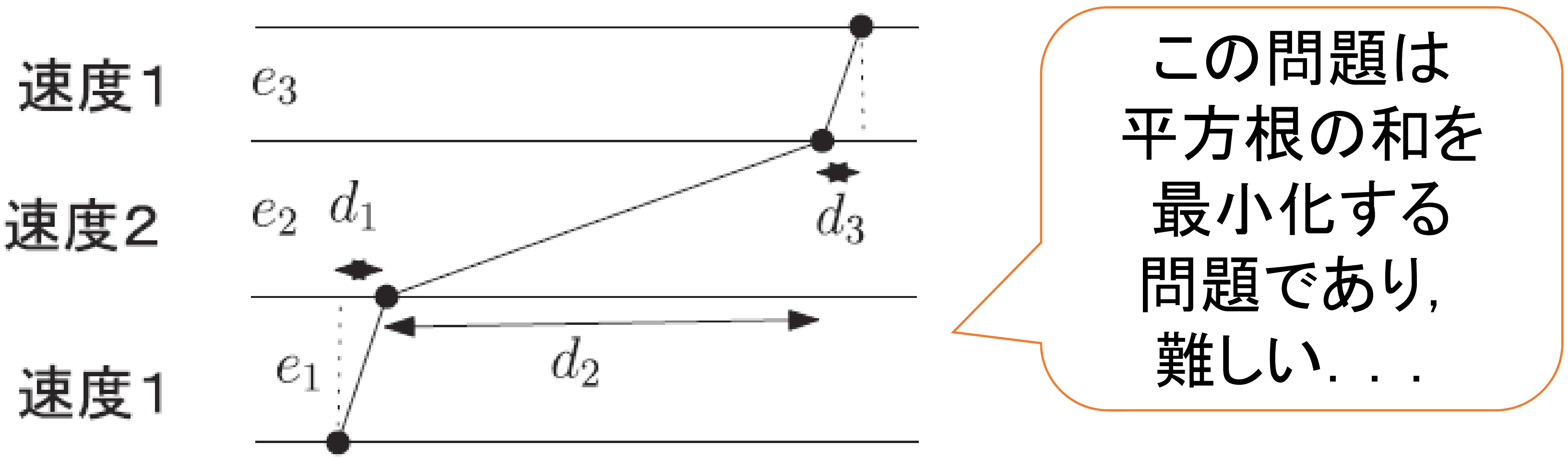


速度一定の場合なら直線
速度が異なれば？

幾何的最短経路問題の一般化

領域によって速度が異なる場合への拡張

2次元平面で領域によって速度が異なる場合に2点間の(時間に関する)最短経路を求める問題は難しいか？



$$\sqrt{d_1^2 + e_1^2} + \sqrt{d_2^2 + e_2^2}/2 + \sqrt{d_3^2 + e_3^2} \longrightarrow \min$$