OI-OPPA. European Social Fund
Prague & EU: We invest in your future.

Výpočetní geometrie

Plánování pohybu robota
(Robot Motion Planning)

Ondřej Kopřiva

Obsah

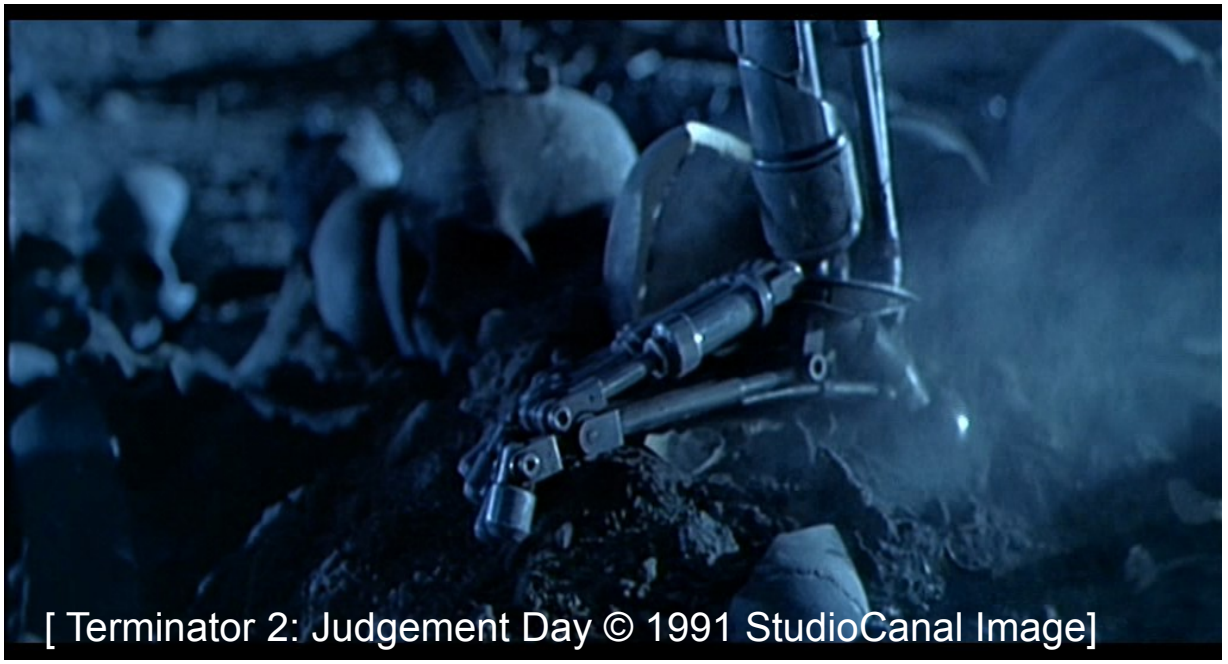
- Trocha teorie
- Algoritmus
- Ukázka
- Otázky

O co jde?

[Terminator 2: Judgement Day © 1991 StudioCanal Image]

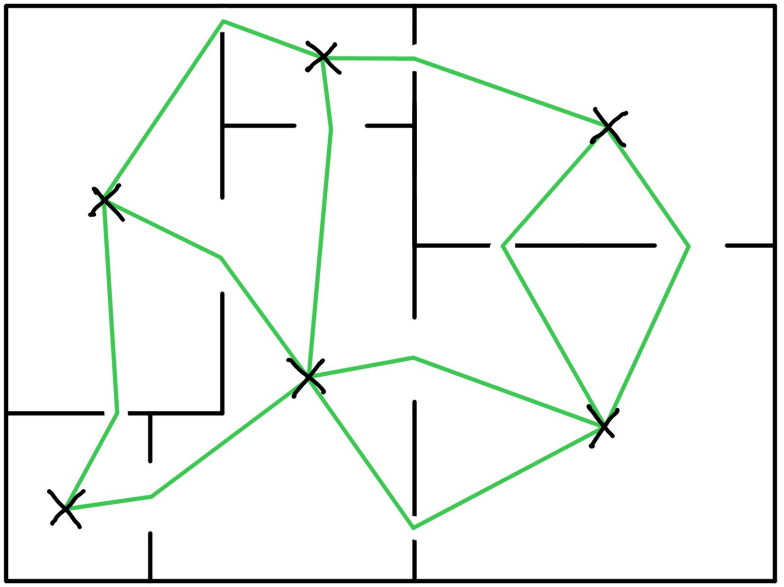
O co jde?

- Robot má nějakě informace o okolním prostředí získané z vlastních senzorů nebo předem zadané.
- Cílem je vytvořit si mapu prostředí, která by umožnovala rychlé nalezení vhodné cesty k jeho cíli.



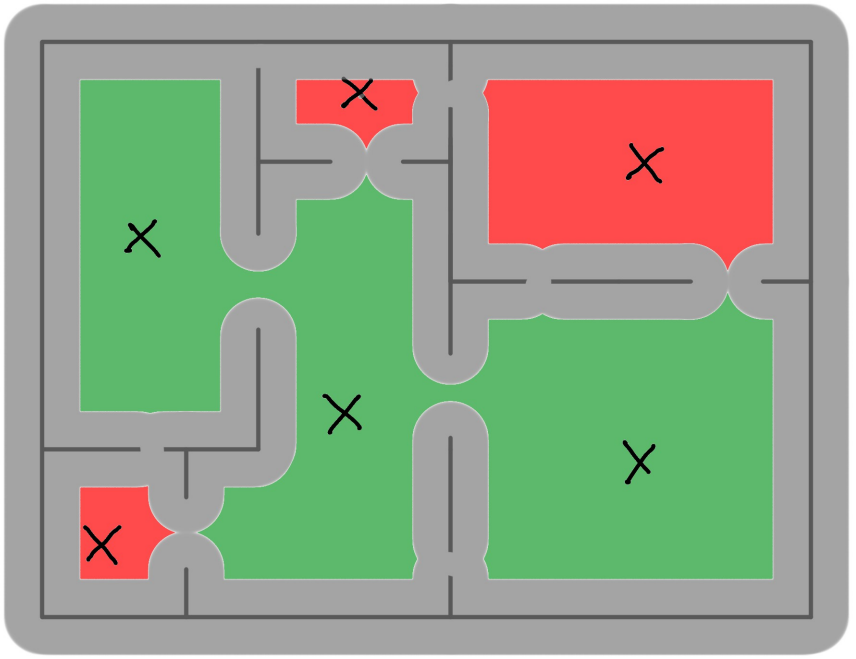
Zjednodušení

- Pro názornost zavedeme některá zjednodušení.
- Robot se pohybuje pouze v rovině nebo v prostoru, který lze jako rovinu interpretovat.
- Pro roboty, kteří nelétají to není příliš zásadní omezení.



Bodový robot

- Dále budeme uvažovat bezrozměrného robota.
- Opět to není zásadní problém, protože naši mapu prostředí lze snadno upravit tak abychom mohli s robotem ve tvaru polygonu pracovat jako s bezrozměrným. (Minkowského suma)



Pohyb

- Robot se pohybuje pouze posunem v libovolném směru a nebere se v úvahu jeho otočení.

Algoritmus

Vstup: Množina vzájemně se nepřekrývajicích polygonů S.

Výstup: Lichoběžníková mapa prostoru volného pro pohyb a graf obsahujicí možné cesty.

1. Nechť je E je množina hran polygonů v S.

2. Vypočítej lichoběžníkovou mapu. (viz. přednášky)

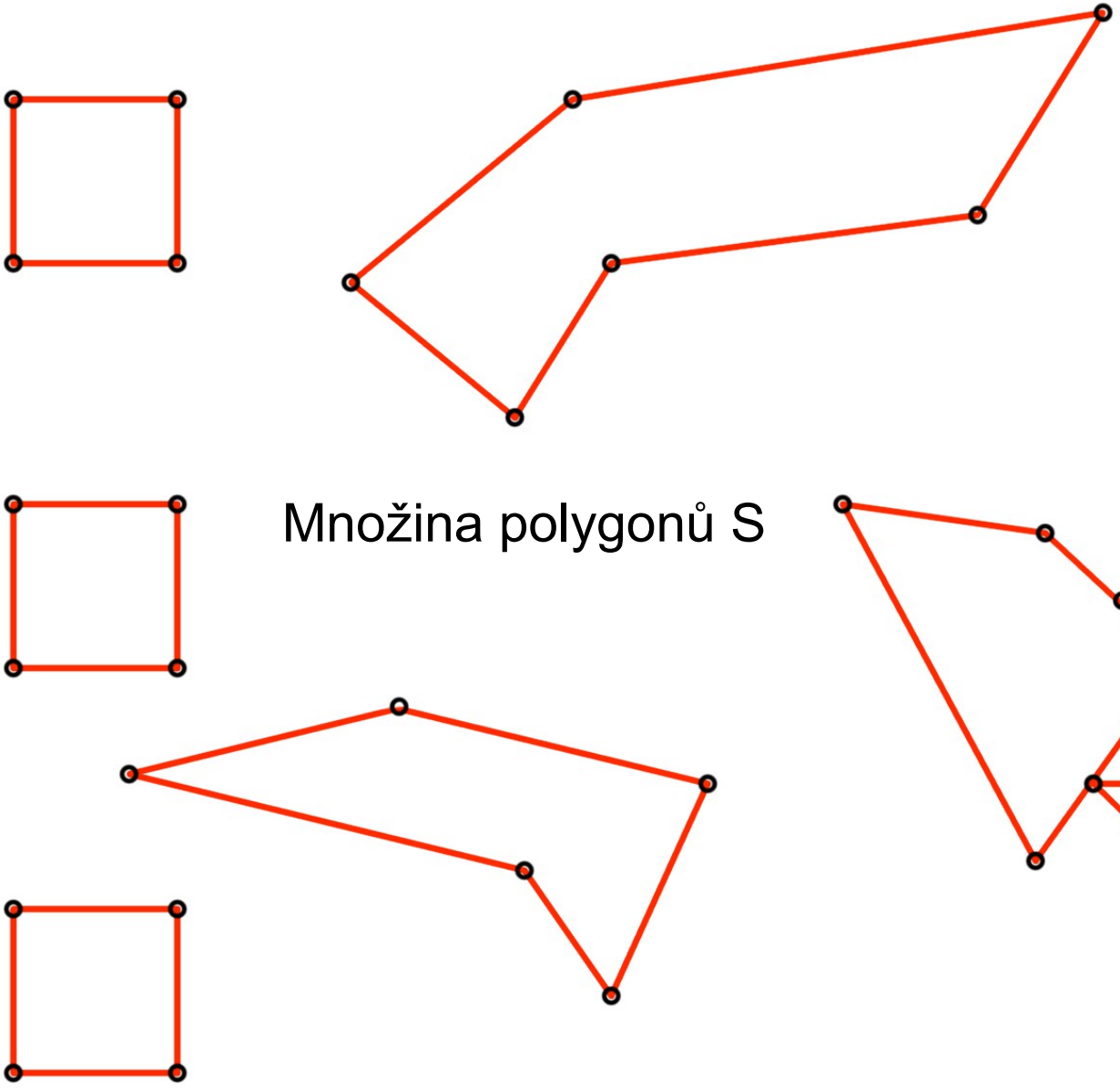
3. Odstraň lichoběžníky, nacházejicí se uvnitř polygonů z S.

4. Ze zbývajicích lichoběžníků sestav graf tak, že na každé hraně, která odděluje 2 lichoběžníky a ve středu každého lichoběžníku vznikne vrchol grafu a všechny vrcholy grafu na okrajích lichoběžníku jsou spojeny hranou s vrcholem ve středu lichoběžníku.

Algoritmus, pokračování

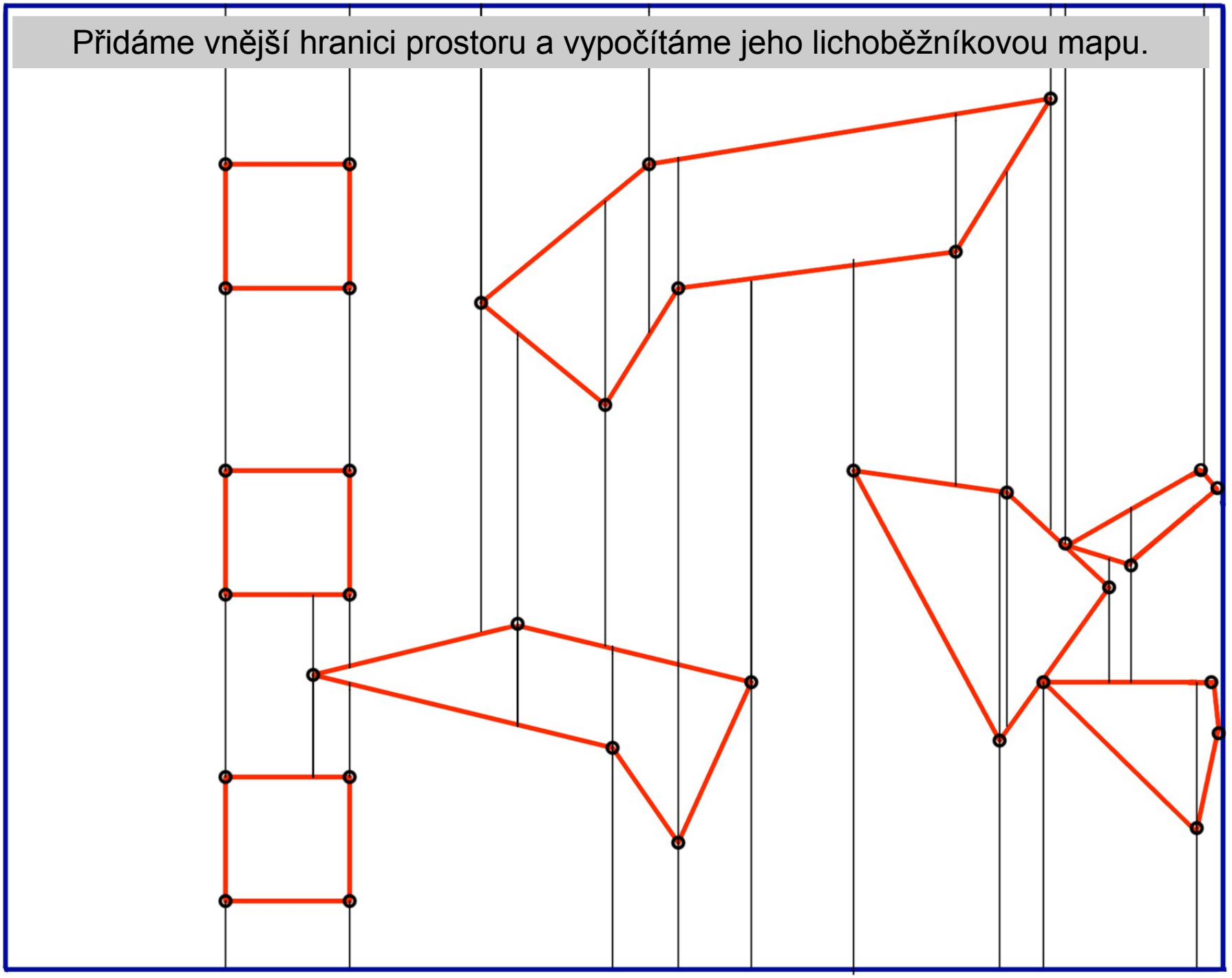
- Algoritmus na výpočet lichoběžníkové mapy zároveň vytvoří vyhledávací strom, který slouží k nalezení, ve kterém lichoběžníku se nachází start a cíl cesty.
- Po nalezení odpovídajicích lichoběžníků už stačí pouze najít prohledáváním do šířky cestu mezi startem a cílem.
- Složitost vytvoření mapy je O(n log n) a prohledávání pomocí BFS O(n) kde n je počet hran polygonů.

Ukázka algoritmu – mapa světa

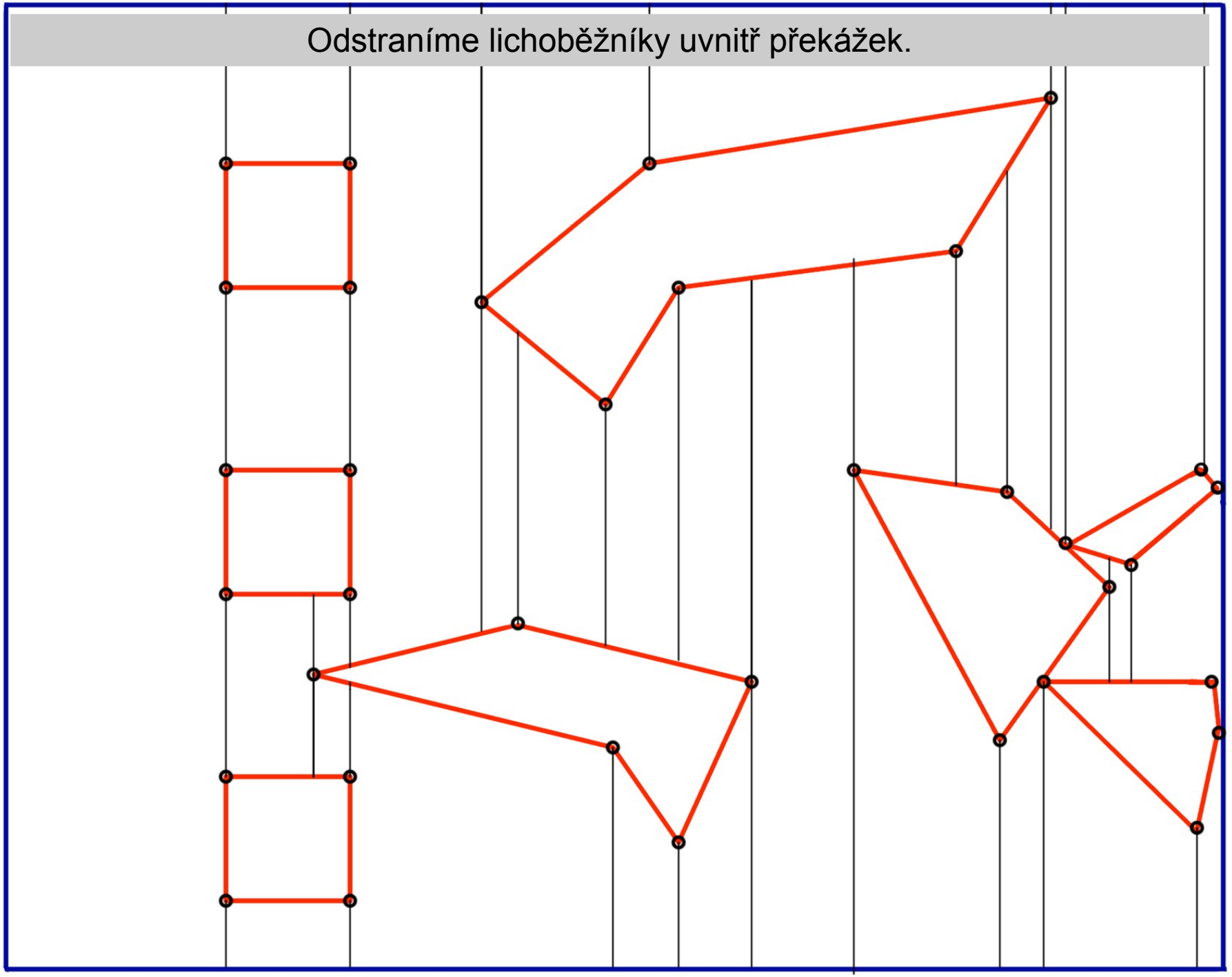


Množina polygonů S

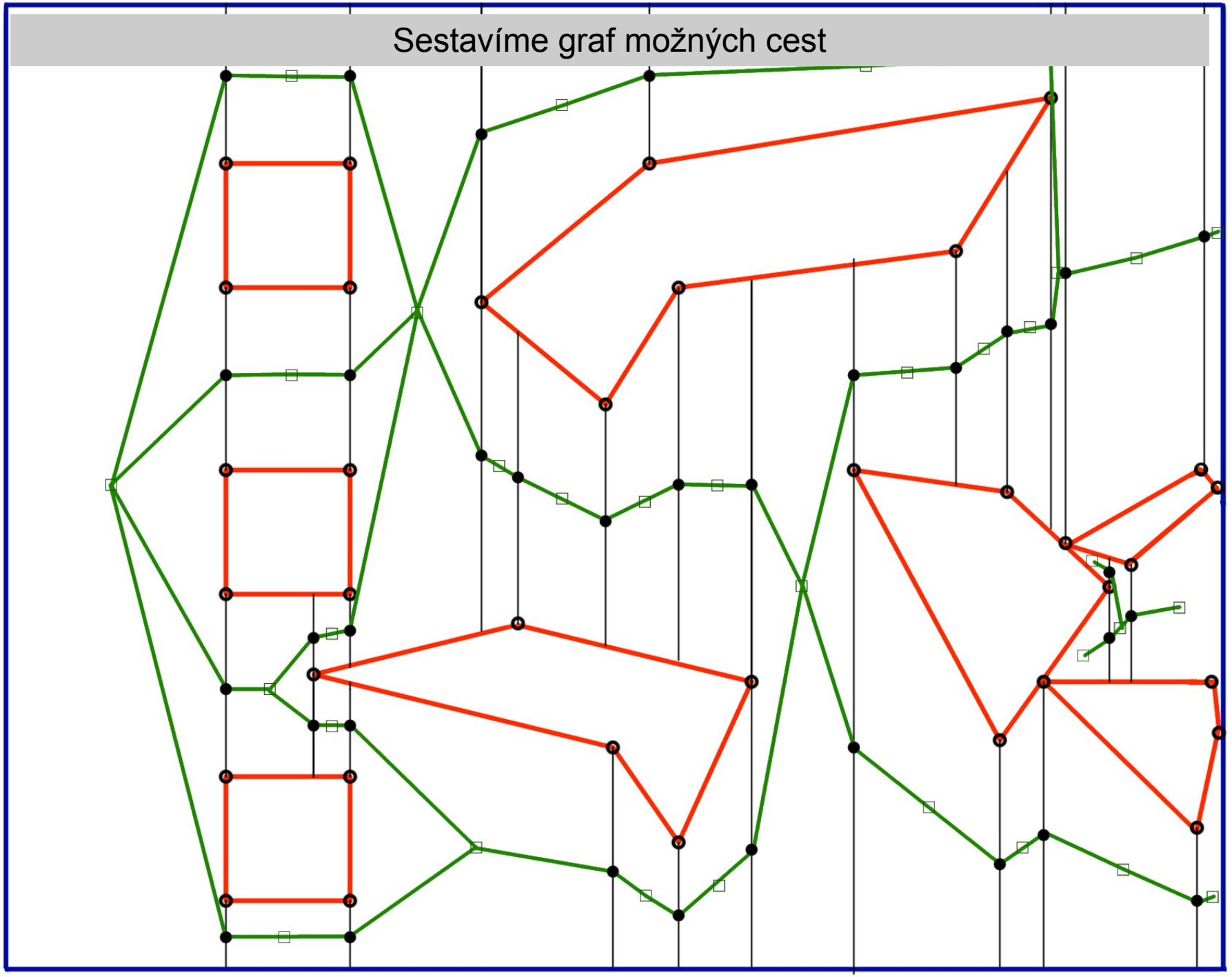
Přidáme vnější hranici prostoru a vypočítáme jeho lichoběžníkovou mapu.

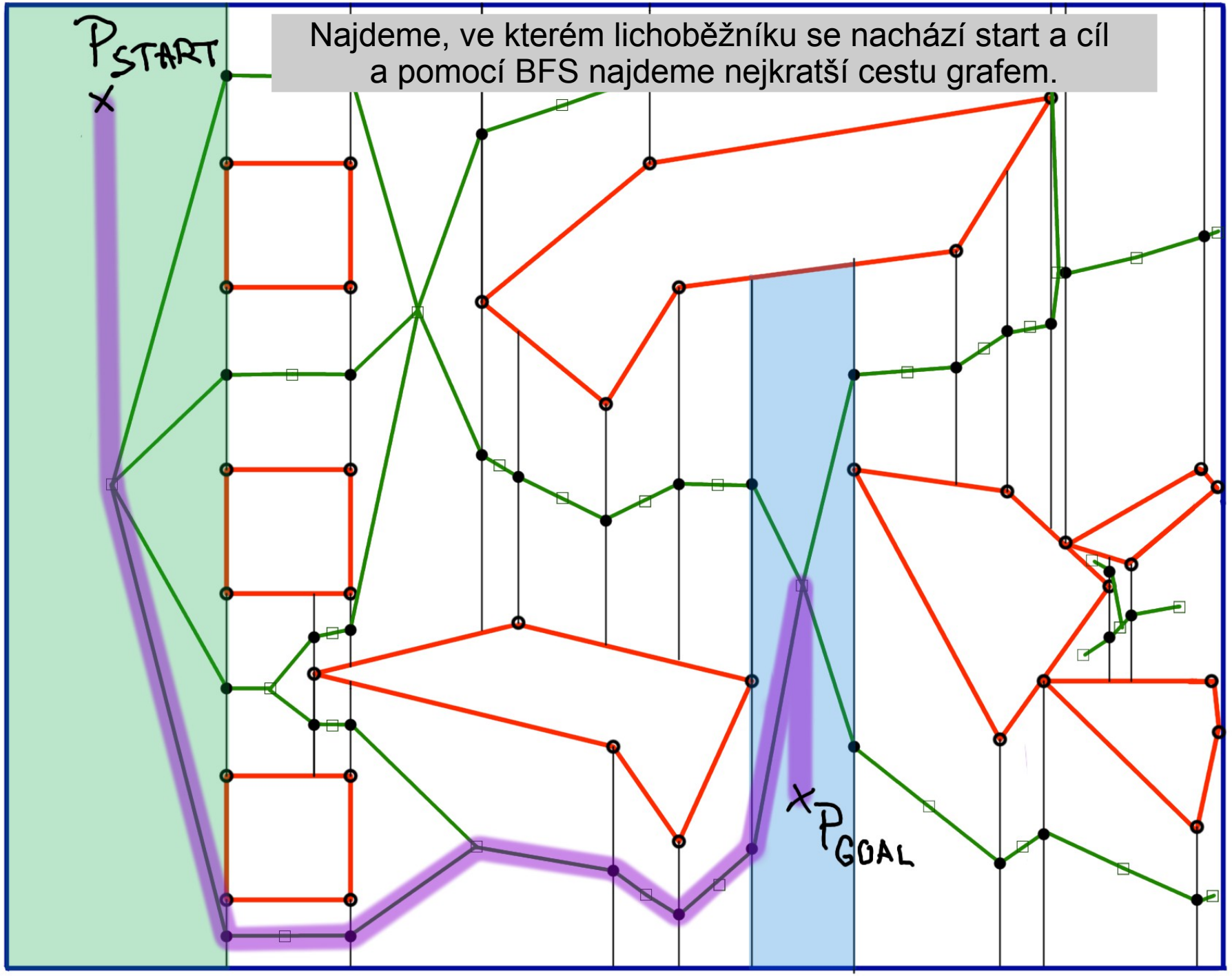


Odstraníme lichoběžníky uvnitř překážek.



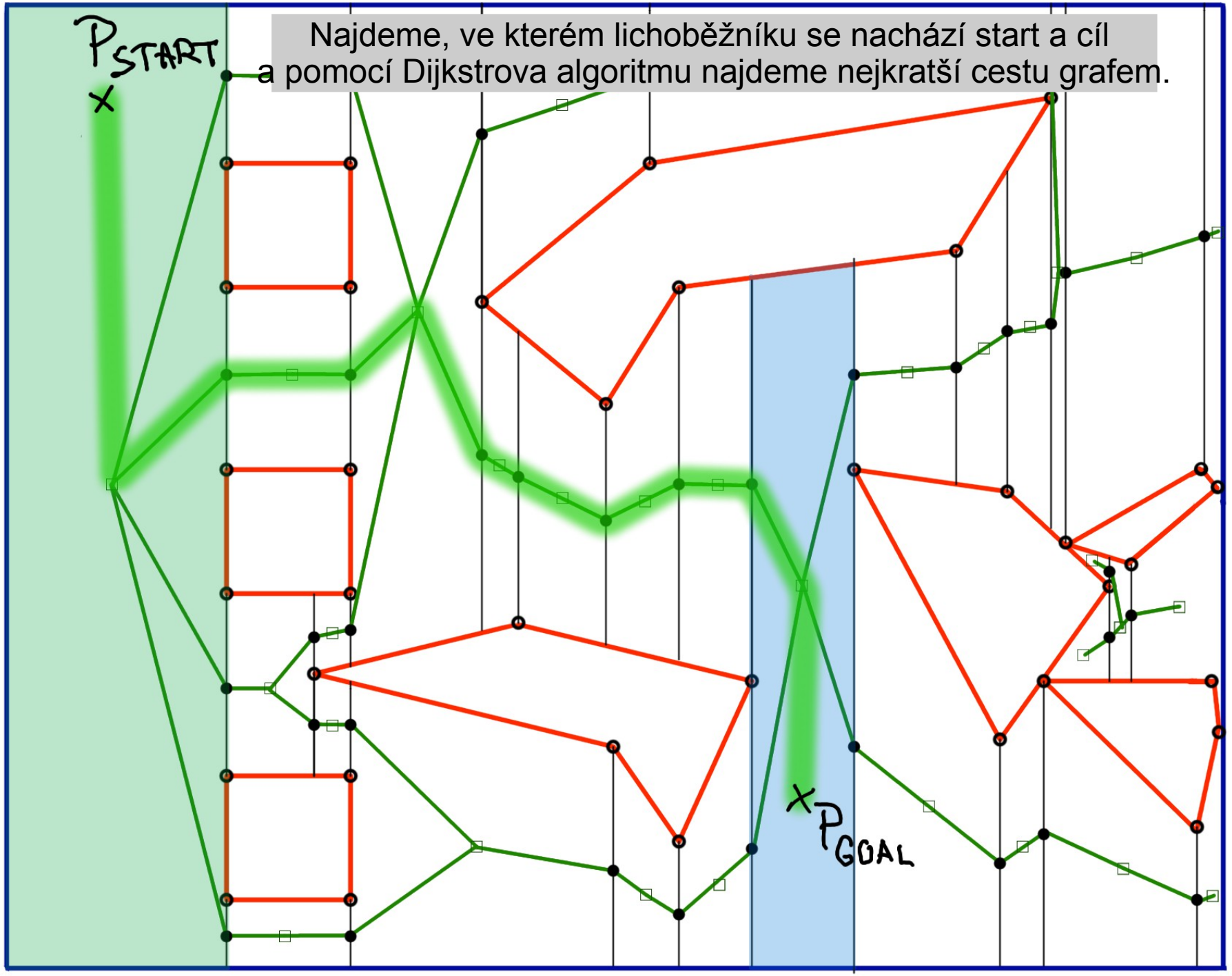
Sestavíme graf možných cest




Najdeme, ve kterém lichoběžníku se nachází start a cíl
a pomocí BFS najdeme nejkratší cestu grafem.
PSTART
PGOAL

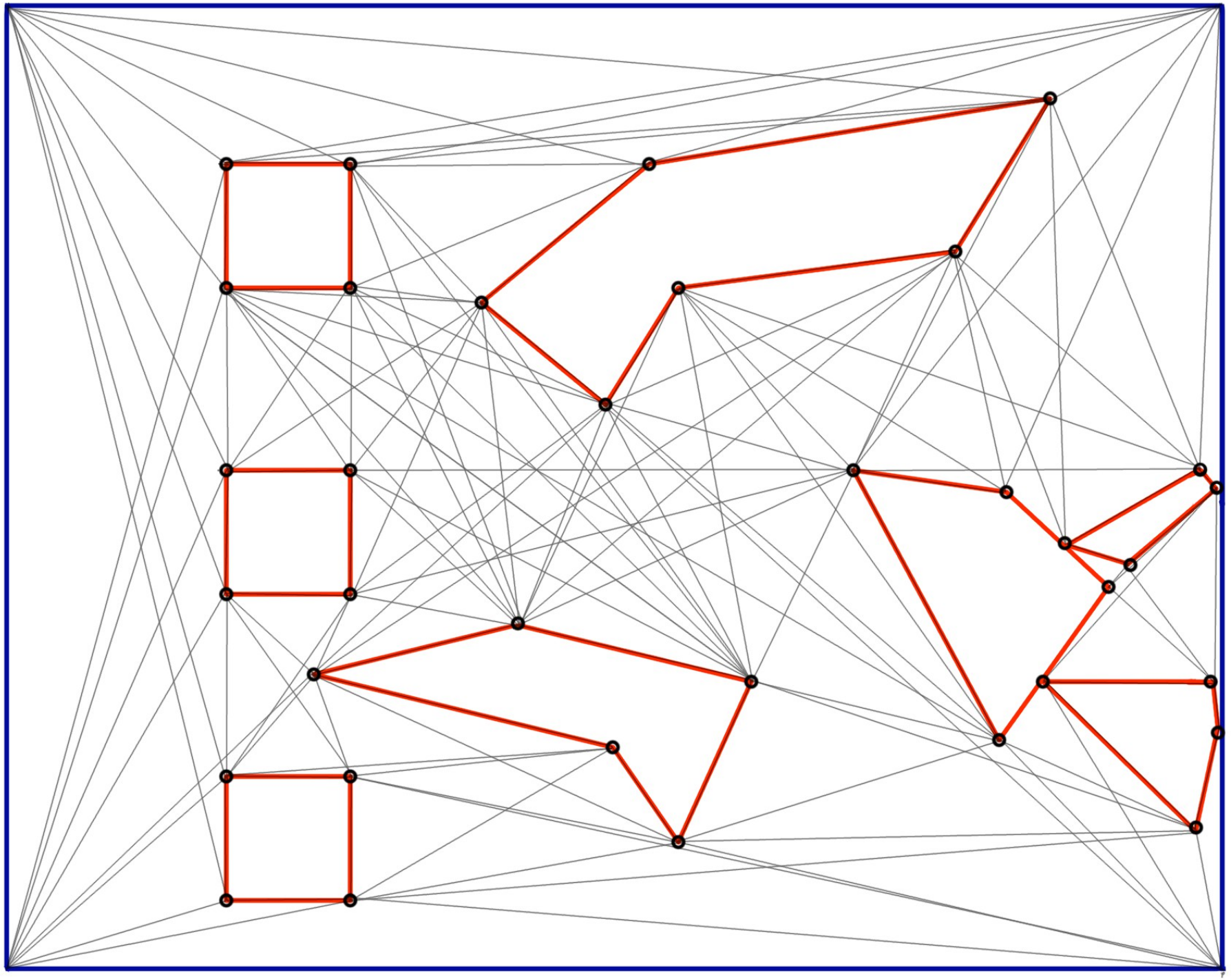
Kvalita cesty

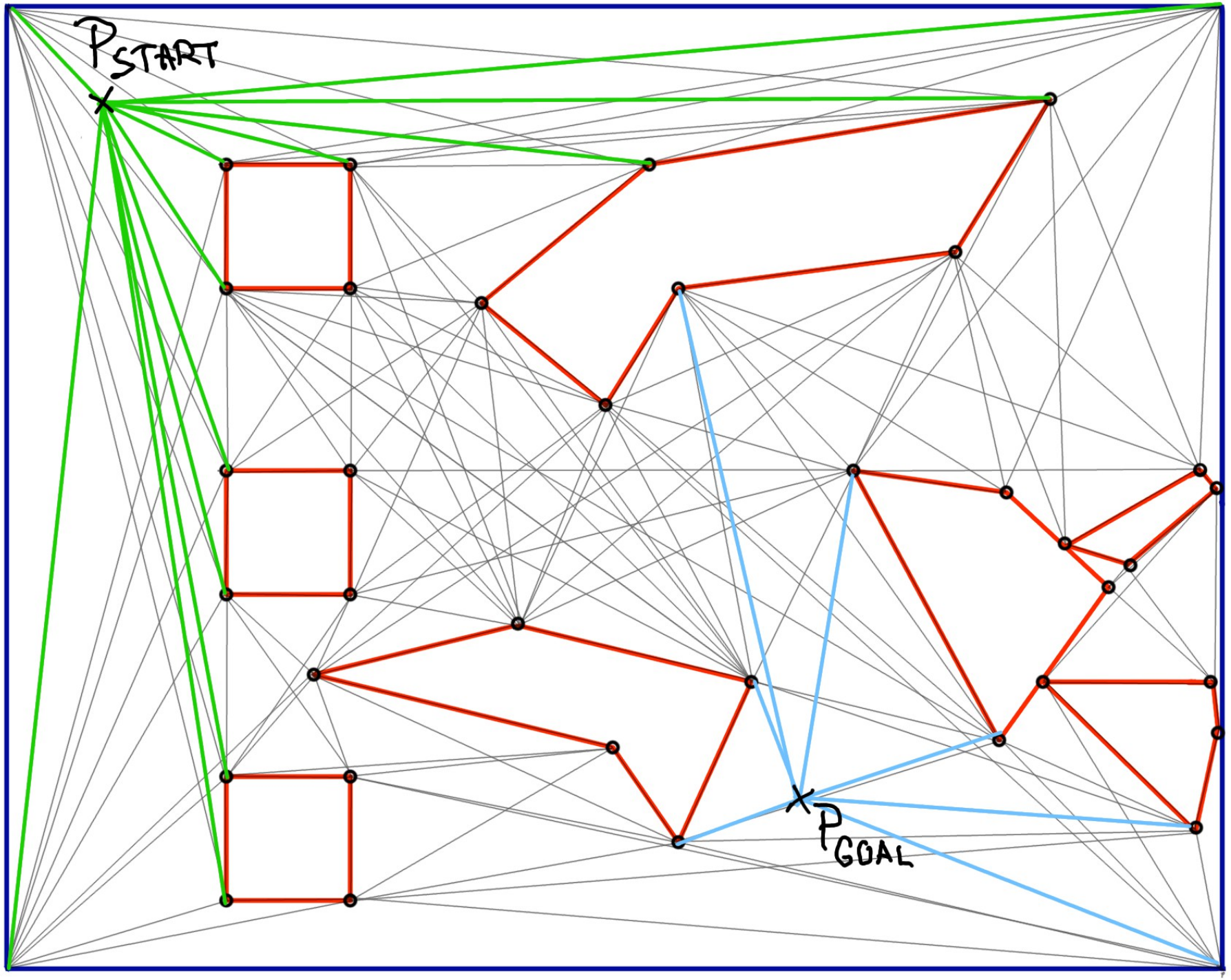
- Algoritmus sice garantuje nalezení cesty pokud nějaká existuje, ale neřeší optimalitu cesty.
- Nalezenou cestu lze vylepšit prohledáváním např. Dijkstrovým algoritmem na úkor rychlosti. Složitost vyhledávání je pak $O(|E| + n \log n)$
- Ale kvalita cesty je stále omezována grafem.

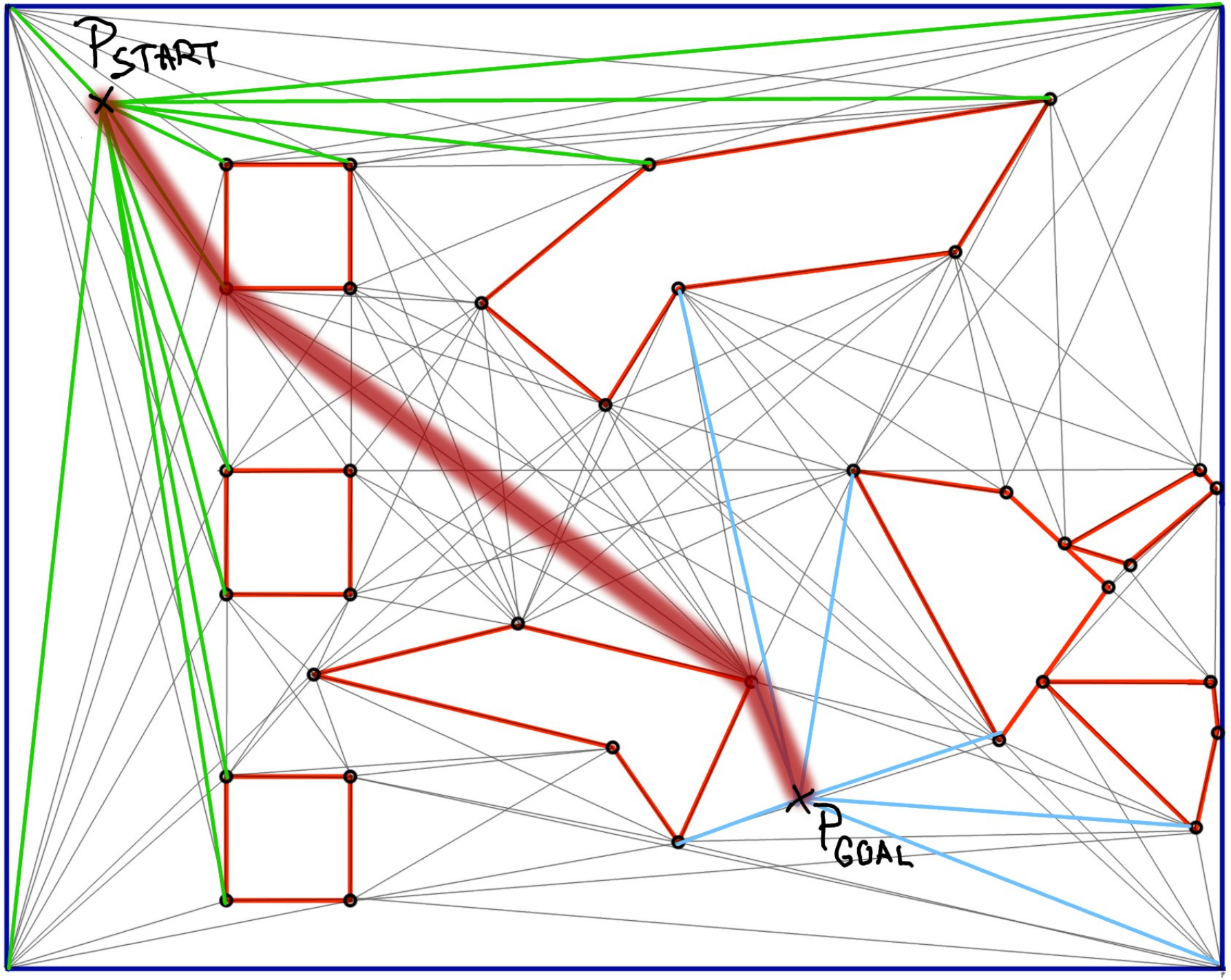

Najdeme, ve kterém lichoběžníku se nachází start a cíl
a pomocí Dijkstrova algoritmu najdeme nejkratší cestu grafem.
PSTART
PGOAL

Nalezení optimální cesty

- Jenom pro zajímavost.
- Pomocí grafu viditelnosti.
- Složitost $O(n^2 \log n)$.




p_{START}
p_{GOAL}


p_{START}
p_{GOAL}

Děkuji za pozornost

Otázky?

Zdroje

- Berg, M. de, Cheong, O., Kreveld, M. van, Overmars, M.: Computational Geometry. Algorithms and Applications, Springer-Verlag, Berlin, 3rd ed., 2008. ISBN: 978-3-540-77973-5

OI-OPPA. European Social Fund
Prague & EU: We invest in your future.