

# Multiagentní hledání cest (s protivníkem)

Marika Ivanová

Katedra teoretické informatiky a matematické logiky (KTIML)

MFF UK

ivanova@ktiml.mff.cuni.cz

October 22, 2020



Dobrý den, vítejte na virtuálním dni otevřených dveří na matfyzu. V tomto videu budeme diskutovat téma Multiagentního hledání cest s protivníkem, což je jedna z oblastí umělé inteligence. Nejdříve si povíme něco o Multiagentního hledání cest a poté ukážeme, jak se tento problém rozšíří o element protivníka.

# Multiagentní hledání cest

## Popis problému

- Skupina agentů (robotů) rozmístěných v daném prostředí
- Každý agent má definovanou počáteční a cílovou pozici
- **Úkol:** Přemístit agenty na cílové pozice aniž by došlo ke kolizi

## Účelová funkce:

- Minimalizovat čas dojetí posledního agenta (makespan)
- Minimalizovat součet počtu přesunů agentů (total distance)
- Minimalizovat součet časů jednotlivých agentů (total arrival time)

2020-10-22

Inf UFG

└ Multiagentní hledání cest

└ Multiagentní hledání cest

Multiagentní hledání cest

Popis problému

- Skupina agentů (robotů) rozmístěných v daném prostředí
- Každý agent má definovanou počáteční a cílovou pozici
- **Úkol:** Přemístit agenty na cílové pozice aniž by došlo ke kolizi

Účelová funkce:

- Minimalizovat čas dojetí posledního agenta (makespan)
- Minimalizovat součet počtu přesunů agentů (total distance)
- Minimalizovat součet časů jednotlivých agentů (total arrival time)

Uvažujme skupinu agentů, například robotů rozmístěných v nějakém prostředí s překážkami. Každý agent má zadefinovanou počáteční a cílovou pozici. Naším úkolem je navigovat agenty do jejich jednotlivých cílových pozic, aniž by do sebe nebo do překážek po cestě narázeli.

Nejen že se snažíme dosáhnout cílových pozic, ale rádi bychom to udělali co nejlépe. Existuje několik kritérií pro posouzení kvality řešení.

Nejtypičtěji se používá takzvaný minimum makespan, neboli snažíme se o co nejkratší čas dojetí posledního agenta do svého cíle. Dále je možnost minimalizovat počet přesunů, tedy snažíme se, aby agenti celkově urazili co nejmenší vzdálenost. Nebo lze také minimalizovat celkový čas, který agenti stráví na cestě.

# Motivace

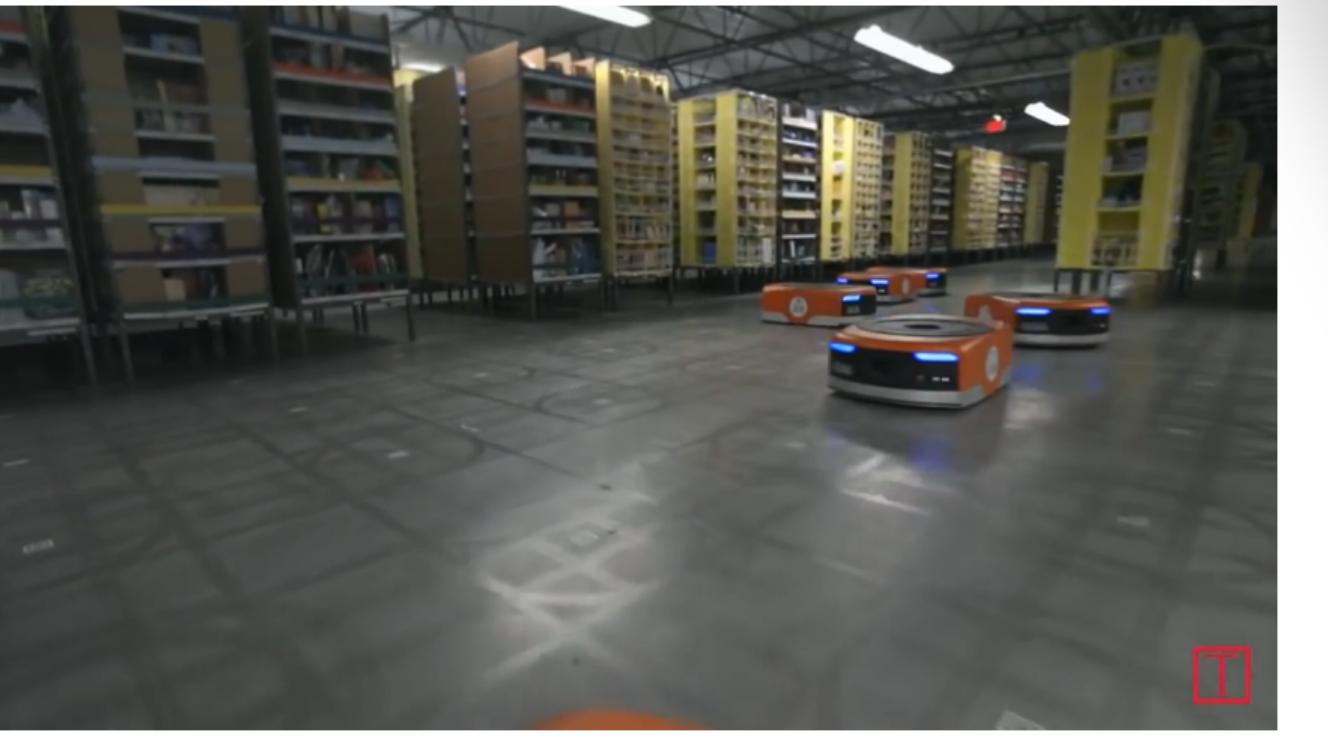
- Přemíšťování zboží ve skladech
- Manipulace s lodními kontejnery
- Navigování skupiny autonomních vozidel



- Přemíšťování zboží ve skladech
- Manipulace s lodními kontejnery
- Navigování skupiny autonomních vozidel



Tento problém je motivován řadou praktických aplikací. V poslední době se ve velkých skladech nasazují roboti, kteří musejí přemisťovat zboží či celé regály z místa na místo. Další aplikací v logistice je přemíšťování lodních kontejnerů. Agenti jsou v tomto případě kontejnery. Je zřejmé, že přemíšťování takových velkých objektů je vhodné provést co nejfektivněji. Dalším poměrně žhavým tématem je ovládání autonomních vozidel, zejména v dopravních špičkách.



3/13

Inf UFG

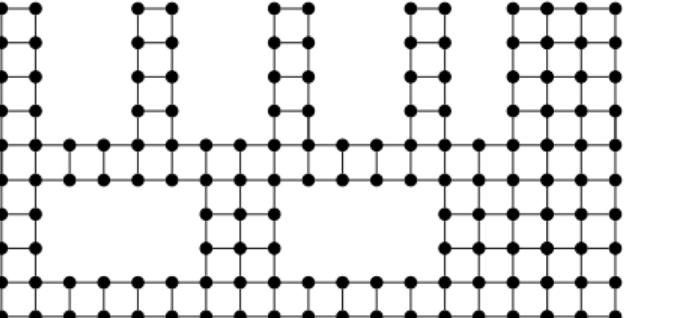
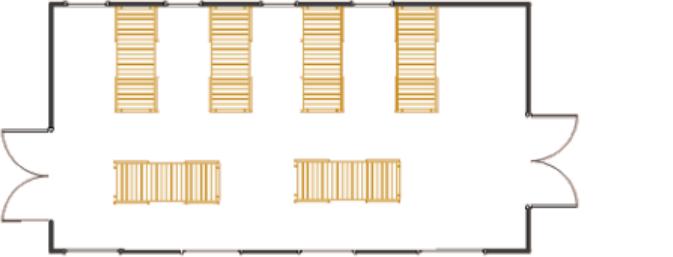
└ Multiagentní hledání cest

2020-10-22



Zde je krátká video ukazující jak se kiwa roboti pohybují ve skladu v amazonu. Vidíme zde skupinu robotů, kteří se potřebují dopravit na určité místo a nesmí kolidovat.

# Abstrakce problému

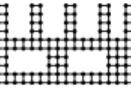


- Matematická formulace nezbytná pro teoretické studium i počítačové zpracování
- Prostředí modelujeme pomocí grafu
  - Graf - množina vrcholů a hran
  - vrcholy - možná umístění agentů
  - hrany spojují sousední umístění
- Definice počátečních a cílových pozic

Abychom mohli problém studovat ať už teoreticky, či provádět simulace, je nezbytné jej formálně zadefinovat. Prostředí, ve kterém jsou agenti rozmištěni modelujeme pomocí grafu. Graf je matematická struktura sestávající z množiny vrcholů a hran. Vrcholy jsou body a hrany jsou spojnice mezi nimi. Vrcholy reprezentují možná umístění agentů, hrany pak spojují přilehlá umístění. Zde na horním obrázku vidíme příklad plánu nějakého skladu. Pro každou pozici, na které může robot stát máme vrchol. Předpokládáme, že se roboti můžou pohybovat ve čtyřech směrech, a tedy nám vzniknem mřížkový graf jako na spodním obrázku.

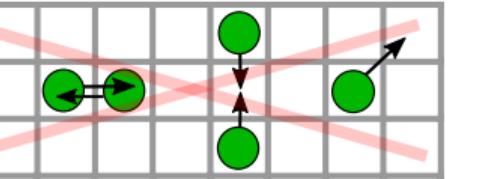
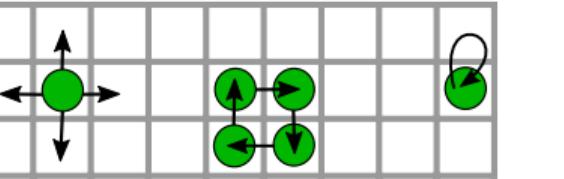


- Matematická formulace nezbytná pro teoretické studium i počítačové zpracování
- Prostředí modelujeme pomocí grafu
  - Graf - množina vrcholů a hran
  - vrcholy - možná umístění agentů
  - hrany spojují sousední umístění
- Definice počátečních a cílových pozic



# Pravidla pohybu

- Každý agent umístěn v jednom vrcholu
- Čas rozdělen na diskrétní kroky
- V každém kroku se agent pohybuje podél hrany nebo stojí na místě
- Agent se může posunout do vrcholu opouštěného jiným agentem ve stejném kroku
- 2 agenti si nesmí vyměnit pozice v jednom kroku

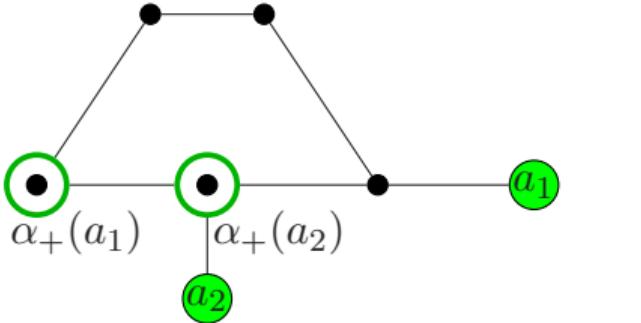


- Každý agent umístěn v jednom vrcholu
- Čas rozdělen na diskrétní kroky
- V každém kroku se agent pohybuje podél hrany nebo stojí na místě
- Agent se může posunout do vrcholu opouštěného jiným agentem ve stejném kroku
- 2 agenti si nesmí vyměnit pozice v jednom kroku



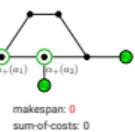
Dále je třeba zadefinovat pravidla, podle kterých se agenti pohybují. Každý agent je umístěn v právě jednom vrcholu. Spojitý čas rozdělujeme na diskrétní kroky, například vteřiny. V jednom časovém kroku se agent může přesunout na sousední vrchol podél hrany, nebo také může zůstat stát na místě. Rovněž je povolen přesun agenta na vrchol, který je současně opouštěn jiným agentem- jako vagony vlaku. Výjimkou k tomuto pravidlu je případ, kdy by si 2 agenti pouze vyměnili pozice - to povoleno není.

# Multiagentní hledání cest



makespan: 0

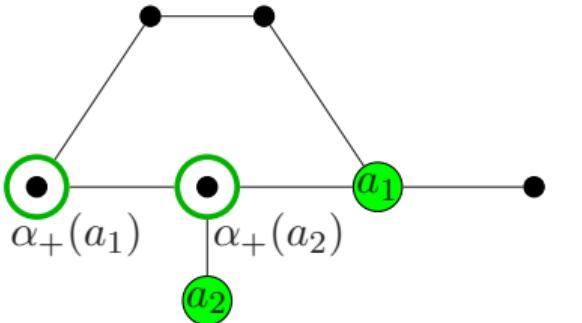
sum-of-costs: 0



Nyní si ukážeme příklad velmi jednoduché instance se dvěma agenty. Zakroužkované vrcholy představují čísla agentů. Podíváme se, jaké je optimální řešení pro nejmenší makespan, tedy nejkratší čas dojetí posledního agenta. Ačkoli  $a_2$  stojí hned vedle svého cíle, v prvním kroku se do  $n_{2j}$  neposune. Místo toho  $A_1$  specha po nejkratší cestě ke svému cíli.  $A_2$  tedy jakoby dává přednost agentovi  $a_1$ . Když  $A_1$  provádí poslední krok, může se i  $A_2$  přemístit do svého cíle. Toto řešení má makespan 3, zatímco celkový čas jednotlivých agentů je 6.

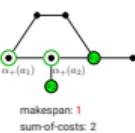
Nyní uvidíme řešení, které má sice horší makespan, ale lepší součet časů jednotlivých agentů. V tomto řešení  $A_2$  okamžitě dorazí do cíle, zatímco  $A_1$  jej musí objet po delší cestě. Toto řešení má součet časů 5, což je optimální. Naopak makespan je 4, což lze udělat lépe, jak jsme viděli na předchozím případě.

# Multiagentní hledání cest



makespan: 1

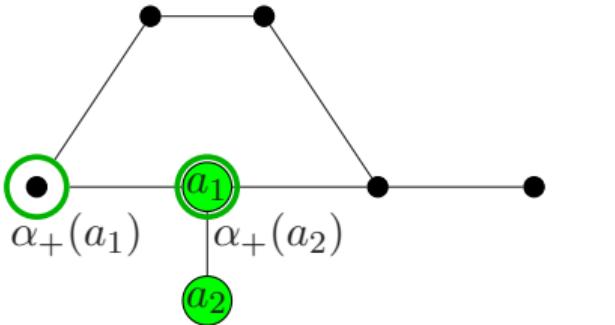
sum-of-costs: 2



Nyní si ukážeme příklad velmi jednoduché instance se dvěma agenty. Zakroužkované vrcholy představují čísla agentů. Podíváme se, jaké je optimální řešení pro nejmenší makespan, tedy nejkratší čas dojetí posledního agenta. Ačkoli a2 stojí hned vedle svého cíle, v prvním kroku se do n2j neposune. Místo toho A1 specha po nejkratší cestě ke svému cíli. A2 tedy jakoby dává přednost agentovi a1. Když A1 provádí poslední krok, může se i A2 přemístit do svého cíle. Toto řešení má makespan 3, zatímco celkový čas jednotlivých agentů je 6.

Nyní uvidíme řešení, které má sice horší makespan, ale lepší součet časů jednotlivých agentů. V tomto řešení A2 okamžitě dorazí do cíle, zatímco A1 jej musí objet po delší cestě. Toto řešení má součet časů 5, což je optimální. Naopak makespan je 4, což lze udělat lépe, jak jsme viděli na předchozím případě.

# Multiagentní hledání cest



makespan: 2

sum-of-costs: 4

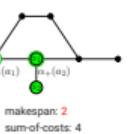
Inf UFG

└ Multiagentní hledání cest

└ Multiagentní hledání cest

2020-10-22

Multiagentní hledání cest



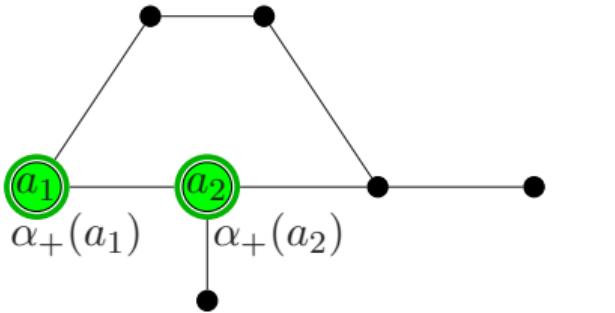
makespan: 2

sum-of-costs: 4

Nyní si ukážeme příklad velmi jednoduché instance se dvěma agenty. Zakroužkované vrcholy představují čísla agentů. Podíváme se, jaké je optimální řešení pro nejmenší makespan, tedy nejkratší čas dojetí posledního agenta. Ačkoli  $a_2$  stojí hned vedle svého cíle, v prvním kroku se do  $n_{2j}$  neposune. Místo toho  $A_1$  specha po nejkratší cestě ke svému cíli.  $A_2$  tedy jakoby dává přednost agentovi  $a_1$ . Když  $A_1$  provádí poslední krok, může se i  $A_2$  přemístit do svého cíle. Toto řešení má makespan 3, zatímco celkový čas jednotlivých agentů je 6.

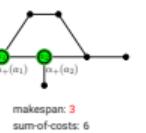
Nyní uvidíme řešení, které má sice horší makespan, ale lepší součet časů jednotlivých agentů. V tomto řešení  $A_2$  okamžitě dorazí do cíle, zatímco  $A_1$  jej musí objet po delší cestě. Toto řešení má součet časů 5, což je optimální. Naopak makespan je 4, což lze udělat lépe, jak jsme viděli na předchozím případě.

# Multiagentní hledání cest



makespan: 3

sum-of-costs: 6



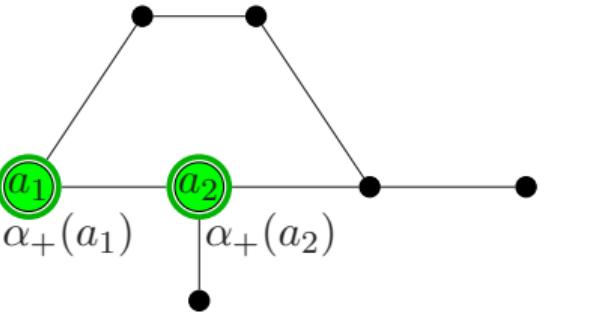
makespan: 3

sum-of-costs: 6

Nyní si ukážeme příklad velmi jednoduché instance se dvěma agenty. Zakroužkované vrcholy představují čísla agentů. Podíváme se, jaké je optimální řešení pro nejmenší makespan, tedy nejkratší čas dojetí posledního agenta. Ačkoli  $a_2$  stojí hned vedle svého cíle, v prvním kroku se do  $n_{2j}$  neposune. Místo toho  $A_1$  specha po nejkratší cestě ke svému cíli.  $A_2$  tedy jakoby dává přednost agentovi  $a_1$ . Když  $A_1$  provádí poslední krok, může se i  $A_2$  přemístit do svého cíle. Toto řešení má makespan 3, zatímco celkový čas jednotlivých agentů je 6.

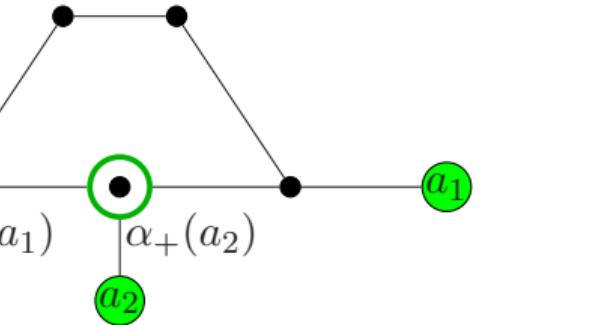
Nyní uvidíme řešení, které má sice horší makespan, ale lepší součet časů jednotlivých agentů. V tomto řešení  $A_2$  okamžitě dorazí do cíle, zatímco  $A_1$  jej musí objet po delší cestě. Toto řešení má součet časů 5, což je optimální. Naopak makespan je 4, což lze udělat lépe, jak jsme viděli na předchozím případě.

# Multiagentní hledání cest



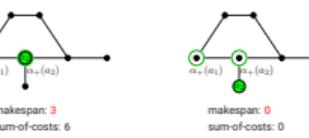
makespan: 3

sum-of-costs: 6



makespan: 0

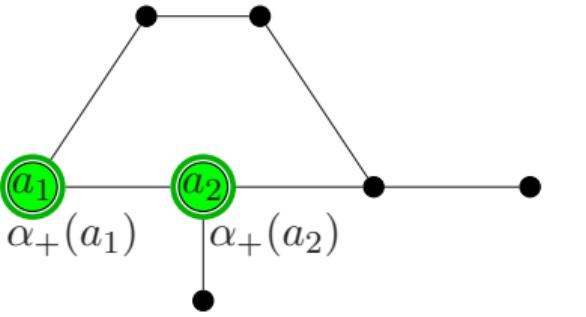
sum-of-costs: 0



Nyní si ukážeme příklad velmi jednoduché instance se dvěma agenty. Zakroužkované vrcholy představují čísla agentů. Podíváme se, jaké je optimální řešení pro nejmenší makespan, tedy nejkratší čas dojetí posledního agenta. Ačkoli a2 stojí hned vedle svého cíle, v prvním kroku se do n2 j neposune. Místo toho A1 specha po nejkratší cestě ke svému cíli. A2 tedy jakoby dává přednost agentovi a1. Když A1 provádí poslední krok, může se i A2 přemístit do svého cíle. Toto řešení má makespan 3, zatímco celkový čas jednotlivých agentů je 6.

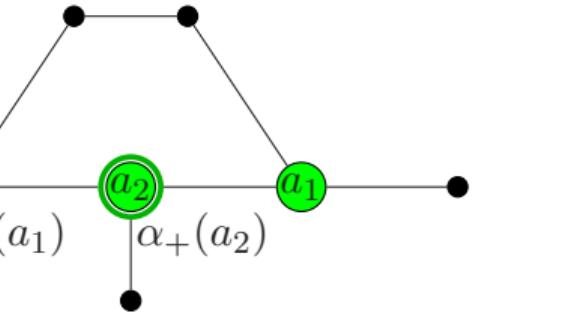
Nyní uvidíme řešení, které má sice horší makespan, ale lepší součet časů jednotlivých agentů. V tomto řešení A2 okamžitě dorazí do cíle, zatímco A1 jej musí objet po delší cestě. Toto řešení má součet časů 5, což je optimální. Naopak makespan je 4, což lze udělat lépe, jak jsme viděli na předchozím případě.

# Multiagentní hledání cest



makespan: 3

sum-of-costs: 6



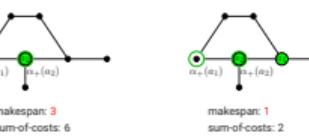
makespan: 1

sum-of-costs: 2

2020-10-22  
Inf UFG

└ Multiagentní hledání cest  
    └ Multiagentní hledání cest

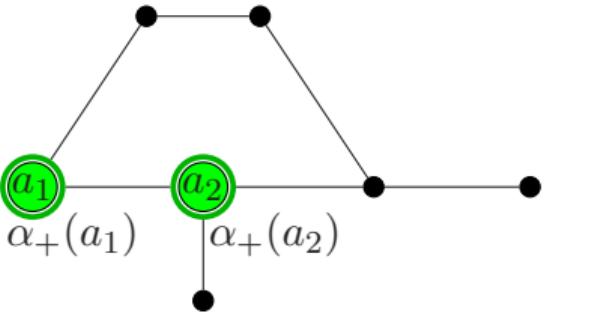
Multiagentní hledání cest



Nyní si ukážeme příklad velmi jednoduché instance se dvěma agenty. Zakroužkované vrcholy představují čísla agentů. Podíváme se, jaké je optimální řešení pro nejmenší makespan, tedy nejkratší čas dojetí posledního agenta. Ačkoli a2 stojí hned vedle svého cíle, v prvním kroku se do n2j neposune. Místo toho A1 specha po nejkratší cestě ke svému cíli. A2 tedy jakoby dává přednost agentovi a1. Když A1 provádí poslední krok, může se i A2 přemístit do svého cíle. Toto řešení má makespan 3, zatímco celkový čas jednotlivých agentů je 6.

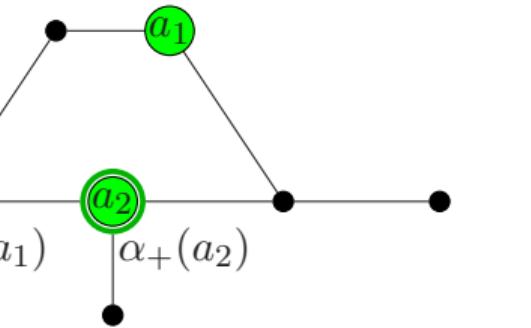
Nyní uvidíme řešení, které má sice horší makespan, ale lepší součet časů jednotlivých agentů. V tomto řešení A2 okamžitě dorazí do cíle, zatímco A1 jej musí objet po delší cestě. Toto řešení má součet časů 5, což je optimální. Naopak makespan je 4, což lze udělat lépe, jak jsme viděli na předchozím případě.

# Multiagentní hledání cest



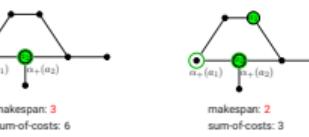
makespan: 3

sum-of-costs: 6



makespan: 2

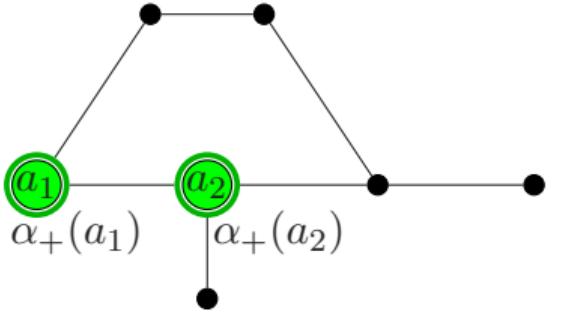
sum-of-costs: 3



Nyní si ukážeme příklad velmi jednoduché instance se dvěma agenty. Zakroužkované vrcholy představují čísla agentů. Podíváme se, jaké je optimální řešení pro nejmenší makespan, tedy nejkratší čas dojetí posledního agenta. Ačkoli a2 stojí hned vedle svého cíle, v prvním kroku se do n2j neposune. Místo toho A1 specha po nejkratší cestě ke svému cíli. A2 tedy jakoby dává přednost agentovi a1. Když A1 provádí poslední krok, může se i A2 přemístit do svého cíle. Toto řešení má makespan 3, zatímco celkový čas jednotlivých agentů je 6.

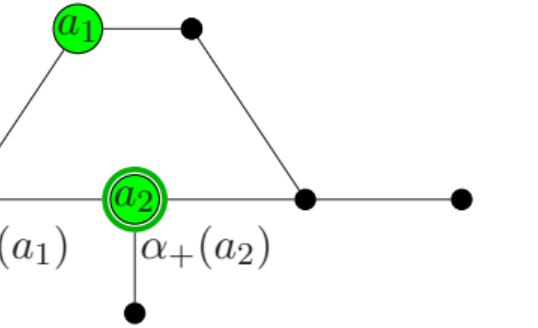
Nyní uvidíme řešení, které má sice horší makespan, ale lepší součet časů jednotlivých agentů. V tomto řešení A2 okamžitě dorazí do cíle, zatímco A1 jej musí objet po delší cestě. Toto řešení má součet časů 5, což je optimální. Naopak makespan je 4, což lze udělat lépe, jak jsme viděli na předchozím případě.

# Multiagentní hledání cest



makespan: 3

sum-of-costs: 6



makespan: 3

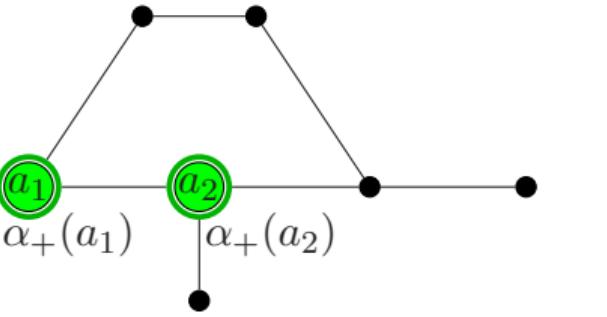
sum-of-costs: 4



Nyní si ukážeme příklad velmi jednoduché instance se dvěma agenty. Zakroužkované vrcholy představují čísla agentů. Podíváme se, jaké je optimální řešení pro nejmenší makespan, tedy nejkratší čas dojetí posledního agenta. Ačkoli a2 stojí hned vedle svého cíle, v prvním kroku se do n2j neposune. Místo toho A1 specha po nejkratší cestě ke svému cíli. A2 tedy jakoby dává přednost agentovi a1. Když A1 provádí poslední krok, může se i A2 přemístit do svého cíle. Toto řešení má makespan 3, zatímco celkový čas jednotlivých agentů je 6.

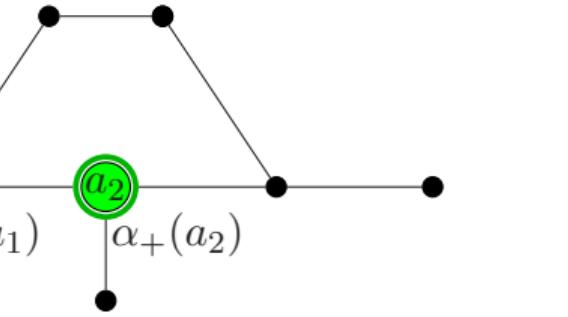
Nyní uvidíme řešení, které má sice horší makespan, ale lepší součet časů jednotlivých agentů. V tomto řešení A2 okamžitě dorazí do cíle, zatímco A1 jej musí objet po delší cestě. Toto řešení má součet časů 5, což je optimální. Naopak makespan je 4, což lze udělat lépe, jak jsme viděli na předchozím případě.

# Multiagentní hledání cest



makespan: 3

sum-of-costs: 6

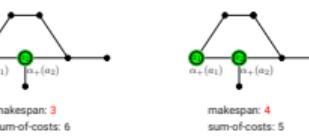


makespan: 4

sum-of-costs: 5

Nyní si ukážeme příklad velmi jednoduché instance se dvěma agenty. Zakroužkované vrcholy představují čísla agentů. Podíváme se, jaké je optimální řešení pro nejmenší makespan, tedy nejkratší čas dojetí posledního agenta. Ačkoli a2 stojí hned vedle svého cíle, v prvním kroku se do n2j neposune. Místo toho A1 specha po nejkratší cestě ke svému cíli. A2 tedy jakoby dává přednost agentovi a1. Když A1 provádí poslední krok, může se i A2 přemístit do svého cíle. Toto řešení má makespan 3, zatímco celkový čas jednotlivých agentů je 6.

Nyní uvidíme řešení, které má sice horší makespan, ale lepší součet časů jednotlivých agentů. V tomto řešení A2 okamžitě dorazí do cíle, zatímco A1 jej musí objet po delší cestě. Toto řešení má součet časů 5, což je optimální. Naopak makespan je 4, což lze udělat lépe, jak jsme viděli na předchozím případě.



# Řešící metody

## 1. Centralizované

## 2. Decentralizované

2020-10-22  
Inf UFG  
└ Multiagentní hledání cest  
    └ Řešící metody

Řešící metody

1. Centralizované

2. Decentralizované

Řešící metody lze rozdělit na 2 základní skupiny - Centralizované a decentralizované. U centralizovaných metod je celá skupina agentů považována za jednu entitu. Cesty jsou tedy hledány současně pro všechny agenty. U decentralizovaných metod jsou cesty pro agenty hledány postupně. Výsledné řešení u centralizovaných metod bývá typicky lepší než u decentralizovaných metod, ovšem centralizované metody bývají často výpočetně náročnější.

# Řešící metody

## 1. Centralizované

- Agenti považováni za jednu entitu
- Cesty se hledají simultánně

## 2. Decentralizované

2020-10-22

Inf UFG  
└ Multiagentní hledání cest  
    └ Řešící metody

Řešící metody

1. Centralizované
  - Agenti považováni za jednu entitu
  - Cesty se hledají simultánně

2. Decentralizované

Řešící metody lze rozdělit na 2 základní skupiny - Centralizované a decentralizované. U centralizovaných metod je celá skupina agentů považována za jednu entitu. Cesty jsou tedy hledány současně pro všechny agenty. U decentralizovaných metod jsou cesty pro agenty hledány postupně. Výsledné řešení u centralizovaných metod bývá typicky lepší než u decentralizovaných metod, ovšem centralizované metody bývají často výpočetně náročnější.

# Řešící metody

## 1. Centralizované

- Agenti považováni za jednu entitu
- Cesty se hledají simultánně

## 2. Decentralizované

- Agenti zpracovávání jednotlivě

Inf UFG

└ Multiagentní hledání cest

└ Řešící metody

2020-10-22

Řešící metody

1. Centralizované
  - Agenti považováni za jednu entitu
  - Cesty se hledají simultánně

2. Decentralizované
  - Agenti zpracovávání jednotlivě

Řešící metody lze rozdělit na 2 základní skupiny - Centralizované a decentralizované. U centralizovaných metod je celá skupina agentů považována za jednu entitu. Cesty jsou tedy hledány současně pro všechny agenty. U decentralizovaných metod jsou cesty pro agenty hledány postupně. Výsledné řešení u centralizovaných metod bývá typicky lepší než u decentralizovaných metod, ovšem centralizované metody bývají často výpočetně náročnější.

# Řešící metody

## 1. Centralizované

- Agenti považováni za jednu entitu
- Cesty se hledají simultánně

## 2. Decentralizované

- Agenti zpracovávání jednotlivě

Centralizované metody dávají typicky lepší řešení, ale jsou výpočetně náročnější

1. Centralizované
- Agenti považováni za jednu entitu
  - Cesty se hledají simultánně

2. Decentralizované
- Agenti zpracovávání jednotlivě
- Centralizované metody dávají typicky lepší řešení, ale jsou výpočetně náročnější

# Multiagentní hledání cest s protivníkem (AMAPF)

- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů



Zobecnění problému představuje zavedení elementu protivníka do multiagentního hledání cest. Agenti jsou rozděleni do dvou týmů nazývaných ÚTOČNÍCI a OBRÁNCI. Úkolem útočníka je stejný jako jsme dosud viděli, tedy přemístit agenty na cílové pozice aniž by kolidovali. Obránci se naopak snaží uchránit tyto cíle před útočníky. Problém se tedy stává hrou dvou hráčů, kdy se jednotlivé týmy střídají na tahu.

- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů



# Multiagentní hledání cest s protivníkem (AMAPF)

- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů
  - Útočníci *A*
  - Obránci *D*



8/13

2020-10-22

Inf UFG

└ Multiagentní hledání cest

    └ Multiagentní hledání cest s protivníkem  
(AMAPF)

Multiagentní hledání cest  
s protivníkem (AMAPF)

- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů
  - Útočníci *A*
  - Obránci *D*



Zobecnění problému představuje zavedení elementu protivníka do multiagentního hledání cest. Agenti jsou rozděleni do dvou týmů nazývaných ÚTOČNÍCI a OBRÁNCI. Úkolem útočníka je stejný jako jsme dosud viděli, tedy přemístit agenty na cílové pozice aniž by kolidovali. Obránci se naopak snaží uchránit tyto cíle před útočníky. Problém se tedy stává hrou dvou hráčů, kdy se jednotlivé týmy střídají na tahu.

# Multiagentní hledání cest s protivníkem (AMAPF)

- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů
  - Útočníci *A*
  - Obránci *D*
- Útočníci se snaží dosáhnout cílových pozic
- Obránci v tom brání



- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů
  - Útočníci *A*
  - Obránci *D*
- Útočníci se snaží dosáhnout cílových pozic
- Obránci v tom brání



Zobecnění problému představuje zavedení elementu protivníka do multiagentního hledání cest. Agenti jsou rozděleni do dvou týmů nazývaných ÚTOČNÍCI a OBRÁNCI. Úkolem útočníka je stejný jako jsme dosud viděli, tedy přemístit agenty na cílové pozice aniž by kolidovali. Obránci se naopak snaží uchránit tyto cíle před útočníky. Problém se tedy stává hrou dvou hráčů, kdy se jednotlivé týmy střídají na tahu.

# Multiagentní hledání cest s protivníkem (AMAPF)

- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů
  - Útočníci *A*
  - Obránci *D*
- Útočníci se snaží dosáhnout cílových pozic
- Obránci v tom brání
- Hra dvou hráčů
- Týmy se střídají na tahu v každém kroku



- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů
  - Útočníci *A*
  - Obránci *D*
- Útočníci se snaží dosáhnout cílových pozic
- Obránci v tom brání
- Hra dvou hráčů
- Týmy se střídají na tahu v každém kroku



Zobecnění problému představuje zavedení elementu protivníka do multiagentního hledání cest. Agenti jsou rozděleni do dvou týmů nazývaných ÚTOČNÍCI a OBRÁNCI. Úkolem útočníka je stejný jako jsme dosud viděli, tedy přemístit agenty na cílové pozice aniž by kolidovali. Obránci se naopak snaží uchránit tyto cíle před útočníky. Problém se tedy stává hrou dvou hráčů, kdy se jednotlivé týmy střídají na tahu.

# Multiagentní hledání cest s protivníkem (AMAPF)

- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů
  - Útočníci *A*
  - Obránci *D*
- Útočníci se snaží dosáhnout cílových pozic
- Obránci v tom brání
- Hra dvou hráčů
- Týmy se střídají na tahu v každém kroku
- **Motivace:** vojenské a bezpečnostní simulace, herní průmysl



- Zobecnění MAPF
- Agenti rozděleni do dvou soupeřících týmů
  - Útočníci *A*
  - Obránci *D*
- Útočníci se snaží dosáhnout cílových pozic
- Obránci v tom brání
- Hra dvou hráčů
- Týmy se střídají na tahu v každém kroku
- Motivace: vojenské a bezpečnostní simulace, herní průmysl



# Multiagentní hledání cest s protivníkem (AMAPF)

**Výhra útočníka:** Všichni agenti z týmu  $A$  dosáhnou svých cílů

**Problém:** nalézt vyhrávající strategii pro tým  $A$



Výhra týmu útočníků nastane v případě, že se všem agentům podaří dojet na své cíle. Naším úkolem je nalézt vítěznou strategii pro útočníky. Vyhrávající strategie znamená, že v každém kroku, kdy jsou útočníci na tahu, zahrají takový tah, že obránci nemohou zabránit vítězství útočníka ať hrají jakoli. Teoretické studium tohoto problému ukazuje, že už jen rozhodnutí, zda nějaká vyhrávající strategie pro útočníka existuje je velmi těžké, konkrétně PSPACE- těžké. V poslední době se ale zdá, že je to ještě těžší, neboli EXPTIME těžké. Co tyto pojmy přesně znamenají si řekneme až na matfyzu.

# Multiagentní hledání cest s protivníkem (AMAPF)

**Výhra útočníka:** Všichni agenti z týmu  $A$  dosáhnou svých cílů

**Problém:** nalézt vyhrávající strategii pro tým  $A$

## Vyhrávající strategie

V každém kroku, kdy je tým  $A$  na tahu, zahrát takový tah, že  $A$  vyhraje bez ohledu na odpověď soupeře  $D$



Výhra týmu útočníků nastane v případě, že se všem agentům podaří dojet na své cíle. Naším úkolem je nalézt vítěznou strategii pro útočníky. Vyhrávající strategie znamená, že v každém kroku, kdy jsou útočníci na tahu, zahrají takový tah, že obránci nemohou zabránit vítězství útočníka ať hrají jakoli. Teoretické studium tohoto problému ukazuje, že už jen rozhodnutí, zda nějaká vyhrávající strategie pro útočníka existuje je velmi těžké, konkrétně PSPACE- těžké. V poslední době se ale zdá, že je to ještě těžší, neboli EXPTIME těžké. Co tyto pojmy přesně znamenají si řekneme až na matfyzu.

# Multiagentní hledání cest s protivníkem (AMAPF)

**Výhra útočníka:** Všichni agenti z týmu  $A$  dosáhnou svých cílů

**Problém:** nalézt vyhrávající strategii pro tým  $A$

## Vyhrávající strategie

V každém kroku, kdy je tým  $A$  na tahu, zahrát takový tah, že  $A$  vyhraje bez ohledu na odpověď soupeře  $D$

- Rozhodnutí, zda existuje vyhrávající strategie je velmi těžké  
(PSPACE těžké)



2020-10-22

Inf UFG

└ Multiagentní hledání cest

    └ Multiagentní hledání cest s protivníkem  
(AMAPF)

Multiagentní hledání cest  
s protivníkem (AMAPF)

Výhra útočníka: Všichni agenti z týmu  $A$  dosáhnou svých cílů  
Problém: nalézt vyhrávající strategii pro tým  $A$

**Vyhrávající strategie**  
V každém kroku, kdy je tým  $A$  na tahu, zahrát takový tah, že  $A$  vyhraje bez ohledu na odpověď soupeře  $D$

- Rozhodnutí, zda existuje vyhrávající strategie je velmi těžké  
(PSPACE těžké)

# Multiagentní hledání cest s protivníkem (AMAPF)

**Výhra útočníka:** Všichni agenti z týmu  $A$  dosáhnou svých cílů

**Problém:** nalézt vyhrávající strategii pro tým  $A$

## Vyhrávající strategie

V každém kroku, kdy je tým  $A$  na tahu, zahrát takový tah, že  $A$  vyhraje bez ohledu na odpověď soupeře  $D$

- Rozhodnutí, zda existuje vyhrávající strategie je velmi těžké (**PSPACE těžké**)
- Zdá se ale, že je to ještě těžší (**EXPTIME těžké**)



2020-10-22

Inf UFG

└ Multiagentní hledání cest

    └ Multiagentní hledání cest s protivníkem  
(AMAPF)

Multiagentní hledání cest  
s protivníkem (AMAPF)

Výhra útočníka: Všichni agenti z týmu  $A$  dosáhnou svých cílů  
Problém: nalézt vyhrávající strategii pro tým  $A$

**Vyhrávající strategie**  
V každém kroku, kdy je tým  $A$  na tahu, zahrát takový tah, že  $A$  vyhraje bez ohledu na odpověď soupeře  $D$

- Rozhodnutí, zda existuje vyhrávající strategie je velmi těžké (**PSPACE těžké**)
- Zdá se ale, že je to ještě těžší (**EXPTIME těžké**)

Výhra týmu útočníků nastane v případě, že se všem agentům podaří dojet na své cíle. Naším úkolem je nalézt vítěznou strategii pro útočníky. Vyhrávající strategie znamená, že v každém kroku, kdy jsou útočníci na tahu, zahrají takový tah, že obránci nemohou zabránit vítězství útočníka ať hrají jakoli. Teoretické studium tohoto problému ukazuje, že už jen rozhodnutí, zda nějaká vyhrávající strategie pro útočníka existuje je velmi těžké, konkrétně PSPACE- těžké. V poslední době se ale zdá, že je to ještě těžší, neboli EXPTIME těžké. Co tyto pojmy přesně znamenají si řekneme až na matfyzu.

# Multiagentní hledání cest s protivníkem (AMAPF)

**Výhra útočníka:** Všichni agenti z týmu  $A$  dosáhnou svých cílů

**Problém:** nalézt vyhrávající strategii pro tým  $A$

## Vyhrávající strategie

V každém kroku, kdy je tým  $A$  na tahu, zahrát takový tah, že  $A$  vyhraje bez ohledu na odpověď soupeře  $D$

- Rozhodnutí, zda existuje vyhrávající strategie je velmi těžké (**PSPACE těžké**)
- Zdá se ale, že je to ještě těžší (**EXPTIME těžké**)
- Na MFF se dozvíte, co to znamená :)



2020-10-22

Inf UFG

└ Multiagentní hledání cest

    └ Multiagentní hledání cest s protivníkem  
(AMAPF)

Multiagentní hledání cest  
s protivníkem (AMAPF)

Výhra útočníka: Všichni agenti z týmu  $A$  dosáhnou svých cílů  
Problém: nalézt vyhrávající strategii pro tým  $A$

Vyhrávající strategie  
V každém kroku, kdy je tým  $A$  na tahu, zahrát takový tah, že  $A$  vyhraje bez ohledu na odpověď soupeře  $D$

- Rozhodnutí, zda existuje vyhrávající strategie je velmi těžké (**PSPACE těžké**)
- Zdá se ale, že je to ještě těžší (**EXPTIME těžké**)
- Na MFF se dozvíte, co to znamená :)

Výhra týmu útočníků nastane v případě, že se všem agentům podaří dojet na své cíle. Naším úkolem je nalézt vítěznou strategii pro útočníky. Vyhrávající strategie znamená, že v každém kroku, kdy jsou útočníci na tahu, zahrají takový tah, že obránci nemohou zabránit vítězství útočníka ať hrají jakoli. Teoretické studium tohoto problému ukazuje, že už jen rozhodnutí, zda nějaká vyhrávající strategie pro útočníka existuje je velmi těžké, konkrétně PSPACE- těžké. V poslední době se ale zdá, že je to ještě těžší, neboli EXPTIME těžké. Co tyto pojmy přesně znamenají si řekneme až na matfyzu.

# Ukázka pohybu agentů v AMAPF instanci



Nyní se podíváme na krátkou vizualizaci pohybu agentů v AMAPF. Tři světle zelení agenti 4, 5, a 6 nahoře mají tři tmavě zelené cíle dole. Agent č. 3 má cíl ve zbývajícím tmavě zeleném vrcholu vpravo nahoře. Podívejme se, jaké posuny musí agenti udělat, aby červení útočníci jim nezabránili v dosažení cílů.

# Řešící metody

- V typické instanci se nepodaří všem útočníkům dojet do cíle
- Snažíme se tedy o obsazení co největšího počtu cílů

## Hladové

Provede se tah jevící se v danou chvíli nejlépe. Posuzováno dle různých kritérií.

## Herní

Klasické (alpha-beta) i méně známé algoritmy (Monte Carlo tree search) používané ve hrách dvou hráčů jako šachy nebo go.

## MAPF

Rozšíření metod navržených pro MAPF o element protivníka.

Inf UFG

└ Multiagentní hledání cest

└ Řešící metody

2020-10-22

## Řešící metody

- V typické instanci se nepodaří všem útočníkům dojet do cíle
- Snažíme se tedy o obsazení co největšího počtu cílů

Hladové

Herní

MAPF

Provede se tah jevící se v danou chvíli nejlépe. Posuzováno dle různých kritérií.

Klasické (alpha-beta) i méně známé algoritmy (Monte Carlo tree search) používané ve hrách dvou hráčů jako šachy nebo go.

Rozšíření metod navržených pro MAPF o element protivníka.

Ve velké většině případů se nepodaří úplně všem agentům útočníkům dojet do cíle. Snažíme se tedy o to, aby co největší počet útočníků dojel do cíle. Navrhujeme algoritmy, které dostanou na vstupu nějakou instanci problému a vždy když je útočník na tahu zvolí, jaký krok mají agenti provést. Zatím jsme zkoumali 3 různé typy algoritmů. Nejjednodušší hladové metody zahrájí vždy takový tah, který vede do nejlepší pozice. To může být často hodně krátkozraké. Naopak herní algoritmy jako minimax, alpha beta a podobné se snaží dívat dál dopředu a zahrát takový tah, který vede k nelepšímu možnému výsledku za určitý časový úsek. pro Multiagentní hledání cest bez protivníka bylo navrženo mnoho řešících algoritmů. Je tedy možné se z nich inspirovat a upravit je tak, aby v sobě měly element protivníka zahrnutý.

# Budoucí výzkum

MAPF je již mnoho let intenzivně studován

Naopak AMAPF je spíše v začátcích

Možné směry dalšího výzkumu:

- Výpočetní složitost speciálních variant
- Podrobnější srovnání řešících algoritmů
- Návrh nových řešících technik
- Strategie pro tým obránců
- Studium modifikací problému (např. ne všichni agenti mají cíle)

Inf UFG

└ Multiagentní hledání cest

└ Budoucí výzkum

2020-10-22

Budoucí výzkum

MAPF je již mnoho let intenzivně studován  
Naopak AMAPF je spíše v začátcích

Možné směry dalšího výzkumu:

- Výpočetní složitost speciálních variant
- Podrobnější srovnání řešících algoritmů
- Návrh nových řešících technik
- Strategie pro tým obránců
- Studium modifikací problému (např. ne všichni agenti mají cíle)

Multiagentní hledání cest s protivníkem představuje téma s velmi dobrým výzkumným potenciálem. Lze jej studovat teoreticky i prakticky, a v obou směrech lze očekávat zajímavé výsledky, které bude možno publikovat na mezinárodních konferencích o umělé inteligenci a příbuzných tématech.

# Zdroje

1. Adversarial Cooperative Path-Finding: Complexity and Algorithms, 2014  
IEEE 26th International Conference on Tools with Artificial Intelligence
2. Adversarial Cooperative Path-Finding: A First View, AAAI (Late-Breaking Developments) 2013
3. [www.koupy.net/graphrec.php](http://www.koupy.net/graphrec.php)
4. [www.therobotreport.com](http://www.therobotreport.com)
5. [www.smartcitiesworld.net](http://www.smartcitiesworld.net)

2020-10-22

Inf UFG  
└ Multiagentní hledání cest  
└ Zdroje

Zdroje

1. Adversarial Cooperative Path-Finding: Complexity and Algorithms, 2014  
IEEE 26th International Conference on Tools with Artificial Intelligence
2. Adversarial Cooperative Path-Finding: A First View, AAAI (Late-Breaking Developments) 2013
3. [www.koupy.net/graphrec.php](http://www.koupy.net/graphrec.php)
4. [www.therobotreport.com](http://www.therobotreport.com)
5. [www.smartcitiesworld.net](http://www.smartcitiesworld.net)

Tímto bych tedy ráda prezentaci zakončila. Děkuji za shlédnutí a doufám, že již brzy bude situace příznivější a bude možné se sekat osobně.