

# RESOLVING SPATIAL-TIME CONFLICTS IN A SET OF ANY-ANGLE OR ANGLE- CONSTRAINED GRID PATHS



**Konstantin Yakovlev, PhD**

Federal Research Center  
“Computer Science and Control”  
of Russian Academy of Sciences



**Anton Andreychuk**

Peoples’ Friendship  
University of Russia  
(master student)

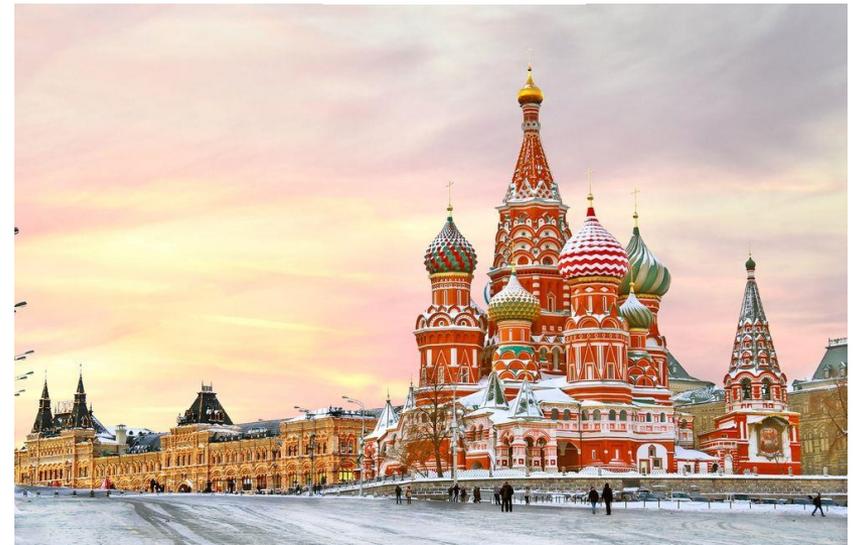


This work was partially supported by Russian Foundation for Basic Research  
research project No. 15-37-20893.



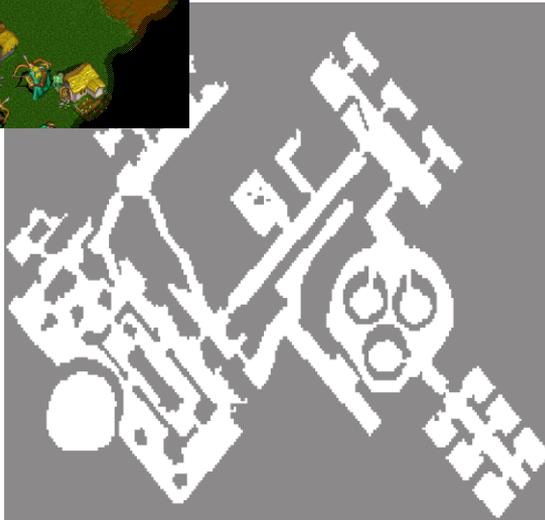
RUSSIAN  
FOUNDATION  
FOR BASIC  
RESEARCH

# Authors' origin



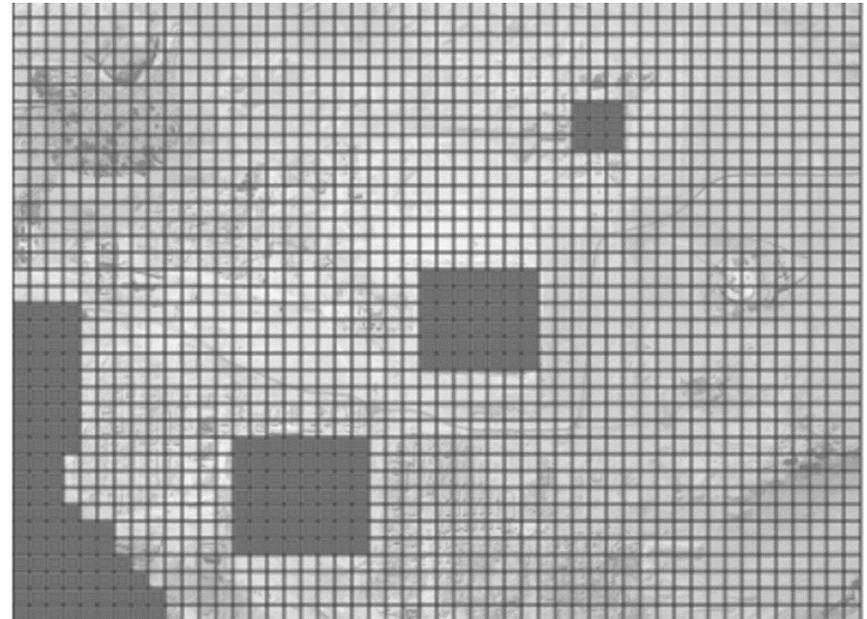
# Motivation (1/4) :: Scenario

- Virtual agents acting in virtual worlds (video games)
- Real-world agents acting in real world (UAVs performing flight in urban environment)

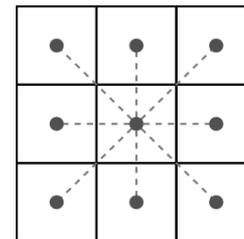


# Motivation (2/4) :: Graph model

Square grids - simple, informative and **easy-to-construct** spatial graph model



Agents locations –  
**centers** of grid cells



# Motivation (3/4) :: Any-angle and Angle-Constrained paths

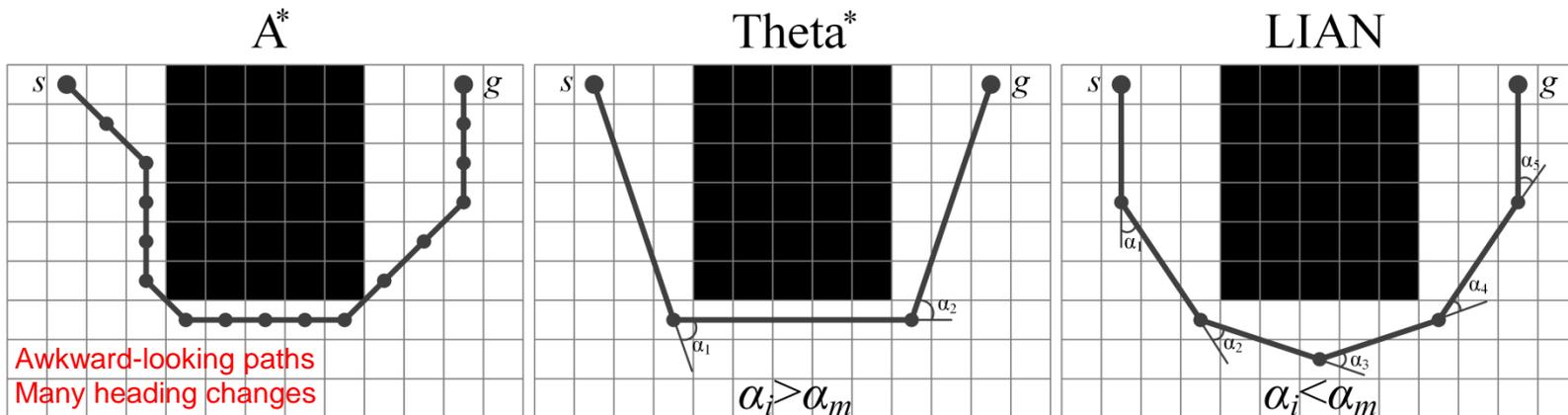
- Agent is allowed to move into **arbitrary** direction
  - As long as the endpoint of the move is tied to distinct grid element

**Any-angle** path is a sequence of adjacent traversable **sections** – line segments connecting the centers of the distinct grid cells

- Theta\*, Anya\*, etc.

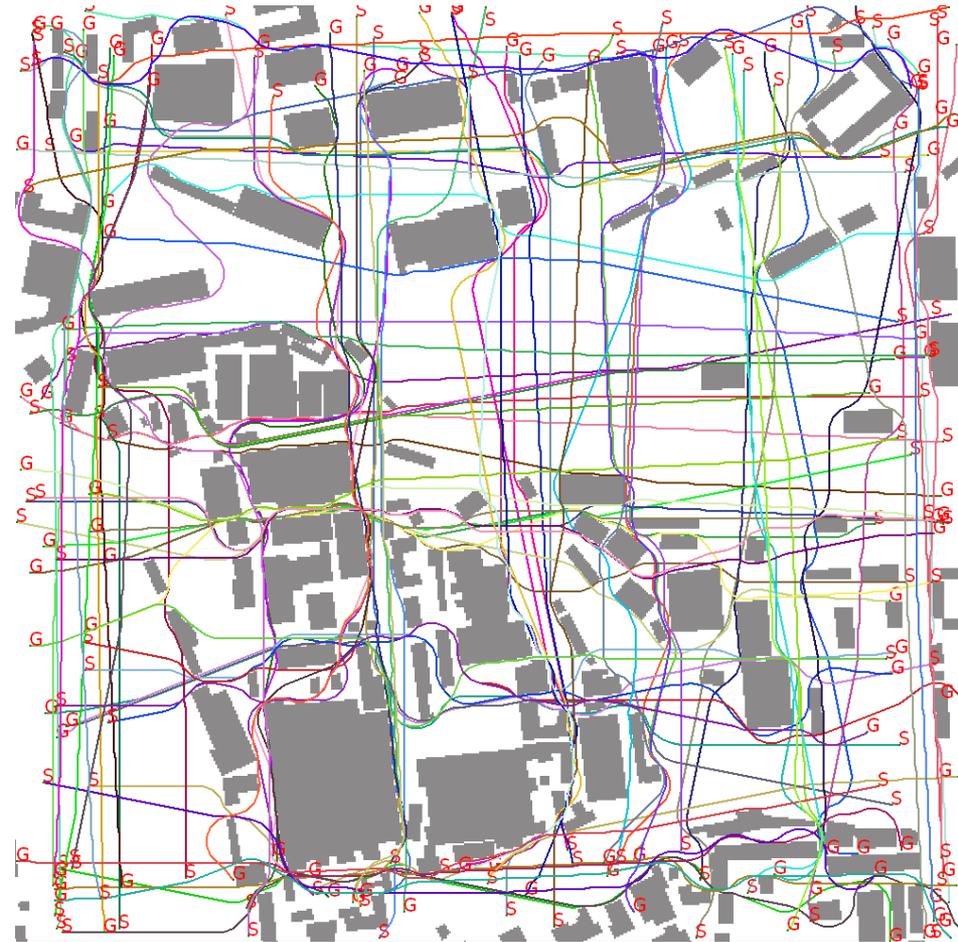
**Angle-constrained** path is a sequence of adjacent traversable section **and** angle between consecutive sections doesn't exceed predefined threshold

- ARC-Theta\*, LIAN



# Motivation (4/4) :: Summary

Square grids  
+  
Multiple agents  
+  
Any-angle  
(angle-constrained)  
paths  
=  
Not a well studied problem  
so far

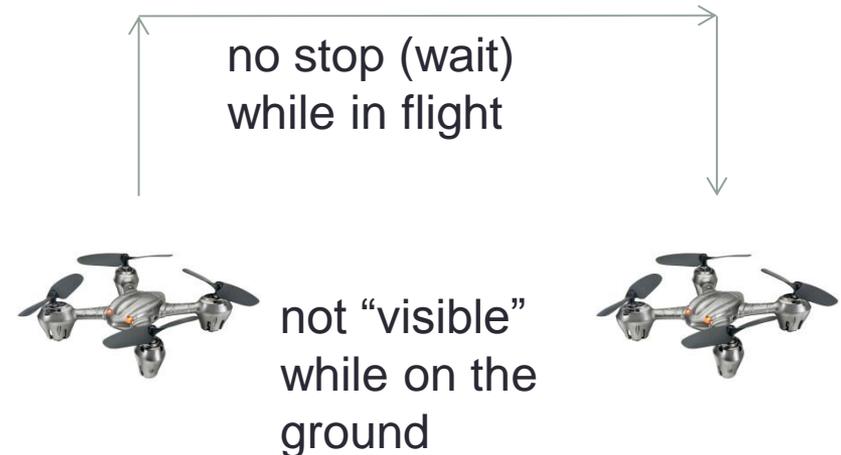




# Targeted scenario



- Agents = UAVs performing nap-of-the earth flight in urban environment
  - Constant speed
  - Constant flight height
  - *Not allowed to stop during the flight*
- Each agent
  - Initially is on the ground
  - Takes off (*possibly with some delay*)
  - Performs the flight to the destination
  - Lands (*thus going off the radar*)
- Cost objective
  - Sum Of Costs
- **100+ agents** on 500x500 grids
  - Finding optimal MAPF solution (with CBS) can be very burdensome. In **[Boyarski et al., 2015b]** Success Rate of **50%** is reported for **80 agents** (250x250 grids with large open spaces, 5min cutoff )
  - **sub-optimal solutions at low computational costs** are desirable



**[Boyarski et al., 2015b]** E. Boyarski, A. Felner, R. Stern, G. Sharon, O. Betzalel, D. Tolpin, and E. Shimony. ICBS: The improved conflict-based search algorithm for multi-agent pathfinding. In Eighth Annual Symposium on Combinatorial Search, 2015.

# Approach

- Decoupled approach:
  - Independent path finding, e.g. each path is planned by each agent individually
    - Any-angle paths by Theta\* planner [Nash et al., 2007]
    - AC-paths by LIAN planner [Yakovlev et al., 2015]
  - **Centralized conflicts resolution**
    - **Notion of the conflict is formalized**
    - **Conflict resolving algorithm is proposed**
    - Can be considered as greedy version of CBS
    - Not studied theoretically so far
    - Effective in practical tasks

**Contribution**

[Nash et al., 2007] Alex Nash, Kenny Daniel, Sven Koenig, and Ariel Felner. Theta\*: Any-angle path planning on grids. In Proceedings of the National Conference on Artificial Intelligence, volume 22, page 1177, Menlo Park, Calif, 2007. AAAI Press.

[Yakovlev et al., 2015] Konstantin Yakovlev, Egor Baskin, and Ivan Hramoin. Grid-based angle-constrained path planning. In Proceedings of The 38th Annual German Conference on Artificial Intelligence (KI2015), pages 208–221. Springer International Publishing, 2015.

# Definitions (1/2) :: Solutions

**Individual solution (p-solution)** =  $\langle \pi^{(i)}, t^{(i)} \rangle$

$\pi^{(i)}$  – path (ac-path) for the  $i$ -th agent

$\pi^{(i)} = \{ e_1^{(i)}, \dots, e_j^{(i)}, \dots, e_v^{(i)} \}$ ,  $e_j^{(i)}$  – section

$len(\pi^{(i)}) = len(e_1^{(i)}) + \dots + len(e_v^{(i)})$  – path's length

$t^{(i)}$  – time offset

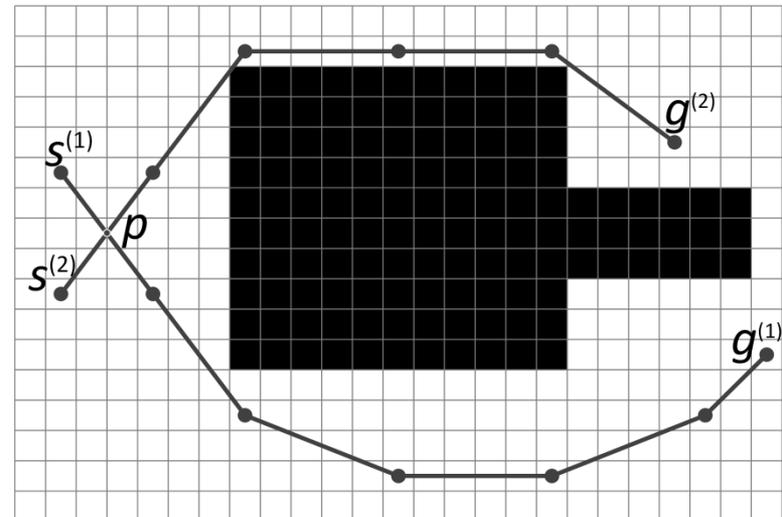
in case agent's speed is one metric unit per one time unit, time offset is measured in the *same units* as path length

$cost(PPS^{(i)}) = t^{(i)} + len(\pi^{(i)})$

## MAPF Potential Solution

$PS = \{ PPS^{(1)}, \dots, PPS^{(i)}, \dots, PPS^{(n)} \}$

$cost(PS) = \sum cost(PPS^{(i)})$



## Definitions (2/2) :: Conflicts

Consider point on a section:  $p \in e_j^{(i)}$  ( $e_j^{(i)} \in \pi^{(i)}$ ,  $\pi^{(i)} \in PPS^{(i)}$ )

**g-value** of that point:

Start point of the section  
 $\downarrow$

$$g(p, e_j^{(i)}, PPS^{(i)}) = t^{(i)} + \text{len}(e_1^{(i)}) + \dots + \text{len}(e_{j-1}^{(i)}) + \text{dist}(\text{sp}(e_j^{(i)}), p)$$

Consider sections  $e_i \in PPS^{(1)}$  and  $e_j \in PPS^{(2)}$

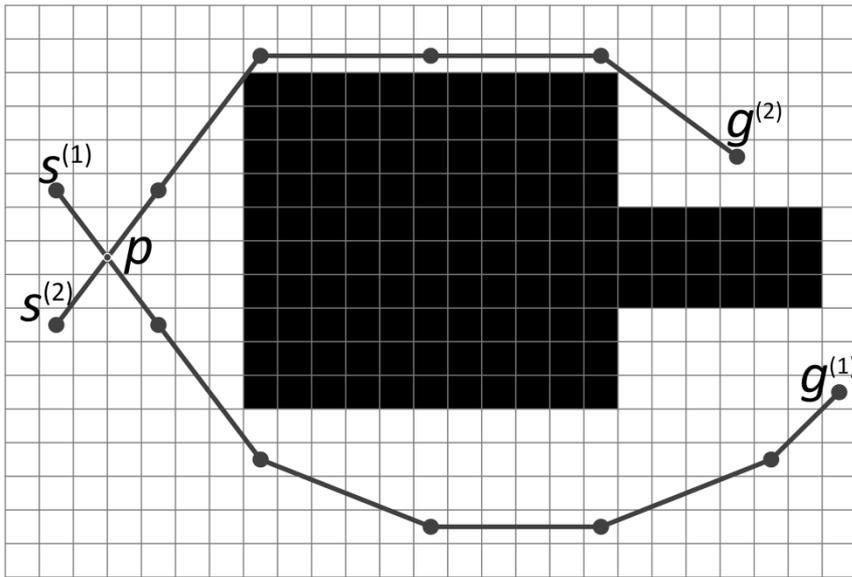
**Section conflict:**  $(e_i, e_j) \in \text{SCON}$  if

$$\exists p: p \in e_i, p \in e_j, \mathbf{g(p, e_i, PPS^{(1)}) = g(p, e_j, PPS^{(2)})}$$

**p-solution conflict:**  $(PPS^{(1)}, PPS^{(2)}) \in \text{CON}$  if

$$\exists e_i \in \pi^{(1)}, e_j \in \pi^{(2)}: (e_i, e_j) \in \text{SCON}$$

# Conflicts :: Example



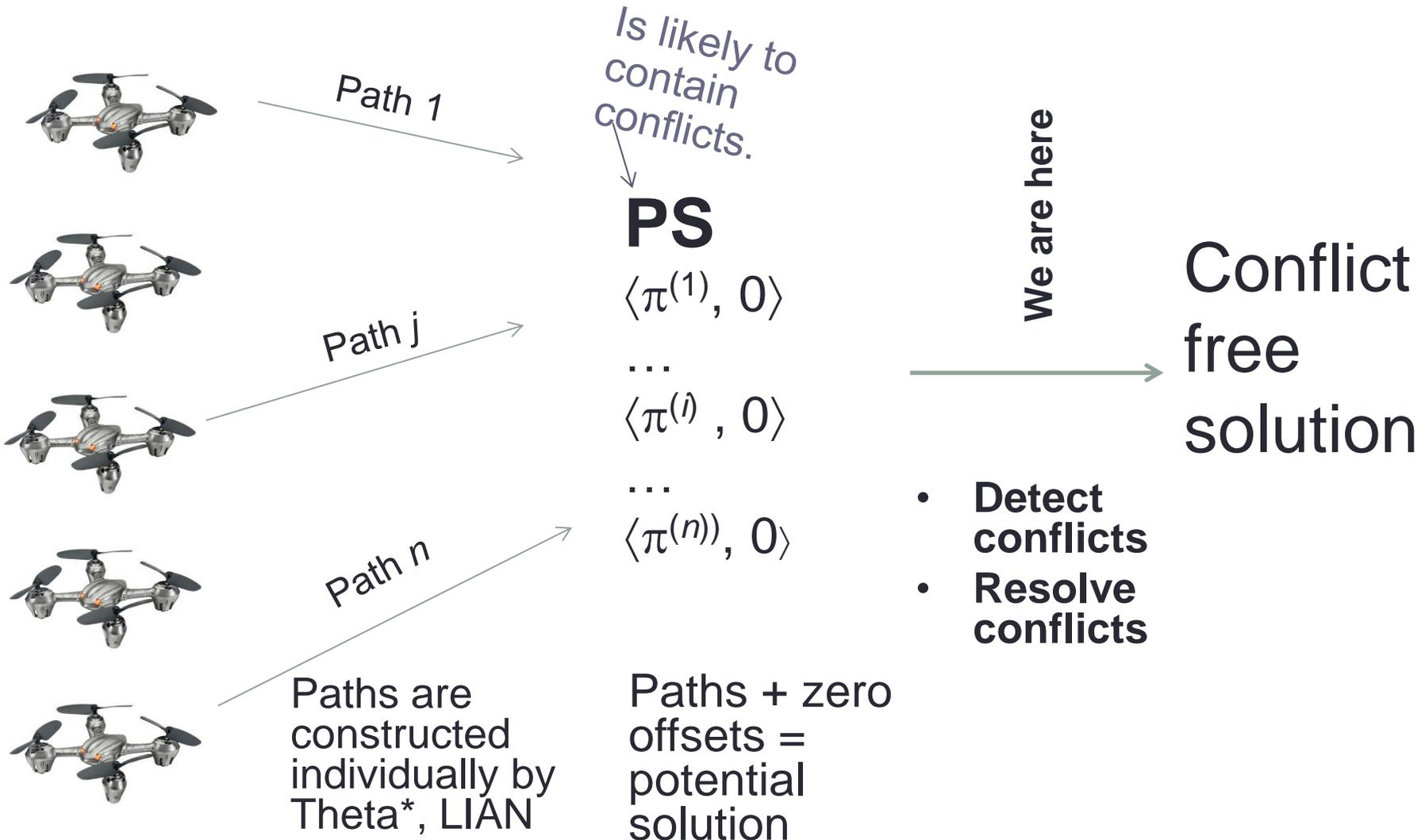
- If  $t^{(1)} = t^{(2)}$  then  $PPS^{(1)}$  and  $PPS^{(2)}$  are in a **conflict** state
- If  $t^{(1)} \neq t^{(2)}$  then  $PPS^{(1)}$  and  $PPS^{(2)}$  do not have conflicts

## Goal

Eliminate all the conflicts and produce a **conflict-free solution PS**

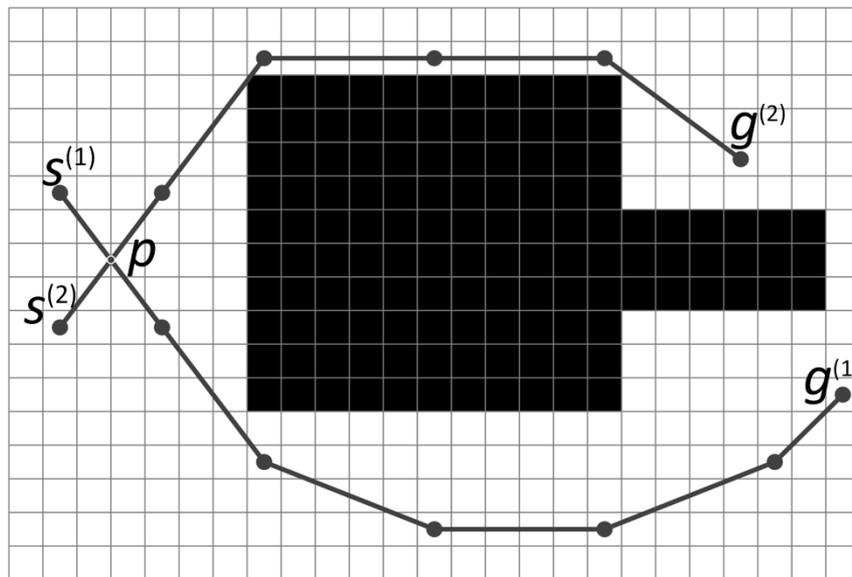
$PS$  is conflict free if  $\forall i, j=1..n (PPS^{(i)}, PPS^{(j)}) \notin \text{CON}$

# MAPF Framework



# Just before we proceed: $\Delta$ -paths

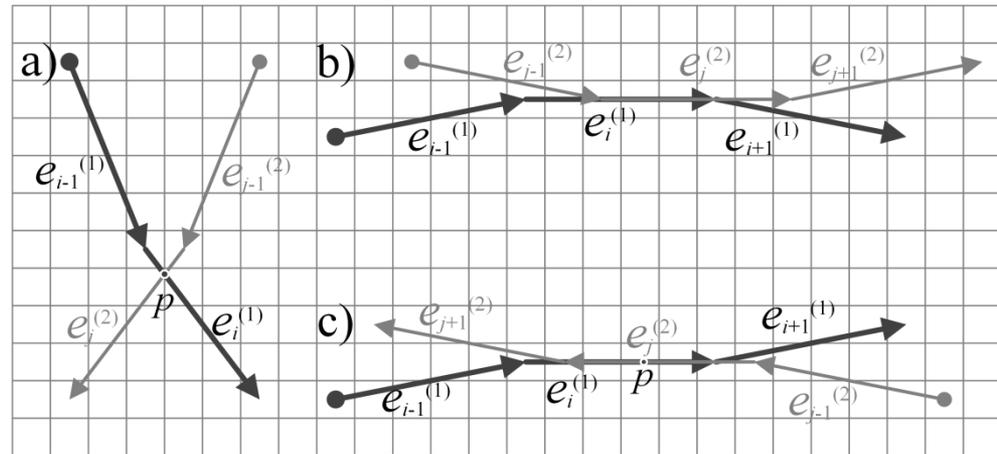
- Consider fixed integer  $\Delta$
- **$\Delta$ -section** is such a section  $e$  that  $\text{round}(\text{len}(e)) = \Delta$
- **$\Delta$ -path** is a path composed of  $\Delta$ -sections with the possible exception of the final section



- (1) angle-constrained  $\Delta$ -path,  $\Delta=5$
  - (2) any-angle  $\Delta$ -path,  $\Delta=5$
- 
- (1) LIAN searches for  $\Delta$ -paths by default
  - (2) Theta\* can be slightly modified to search for  $\Delta$ -paths

# Conflict detection :: Conflict types

- a) Intersection
- b) Pursuit
- c) Head-on



## N.B.

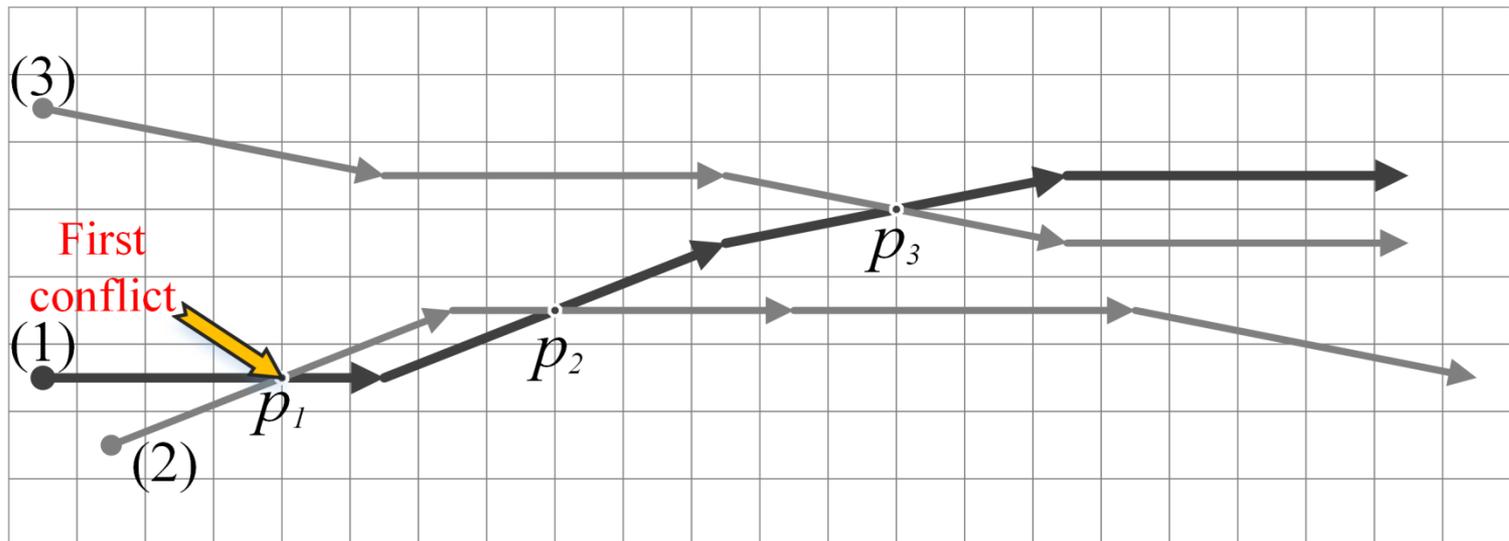
- 1) if  $dist(ep(e_i^{(1)}), ep(e_j^{(2)})) > len(e_i^{(1)}) + len(e_j^{(2)}) \Rightarrow$  conflict can not exist.  
agents are too far away from each other spatial-wise
- 2) if  $g(sp(e_i^{(1)}), PPS^{(1)}) \leq g(ep(e_j^{(2)}), PPS^{(2)})$  and  $g(sp(e_j^{(2)}), PPS^{(2)}) \leq g(ep(e_i^{(1)}), PPS^{(1)}) \Rightarrow$  conflict can not exist.  
agents are too far away from each other time-wise

# Conflict detection :: Algorithm

- **if** conflict can exist
  - **if** sections are **not collinear** // only intersection conflict is to be checked
    - check for **intersection** conflict  
*calculate g-values of the intersection point and check if they are equal*
  - **else** // section are collinear, either pursuit or head-on conflicts should be checked
    - **if** sections point in the **same direction**
      - check for **pursuit** conflict  
*calculate g-values of any point from the overlap set and check if they are equal*
    - **else** // sections point in the opposite directions
      - check for **head-on** conflict  
*if  $g(sp(e_i^{(1)}), PPS^{(1)}) + dist(sp(e_i^{(1)}), ep(e_j^{(2)})) \leq g(ep(e_j^{(2)}), PPS^{(2)})$  **and**  
 $g(sp(e_j^{(2)}), PPS^{(2)}) + dist(sp(e_j^{(2)}), ep(e_i^{(1)})) \leq g(ep(e_i^{(1)}), PPS^{(1)})$*   
**conflict exists**

# FindFirstConflict function

- Input:            p-solution  $PPS^{(i)} \in PS$   
                      set of p-solutions  $PS' \subseteq PS$
- Output:          **first** conflict (pair of sections)  
                      // in case no conflict exists the output is  $\emptyset$



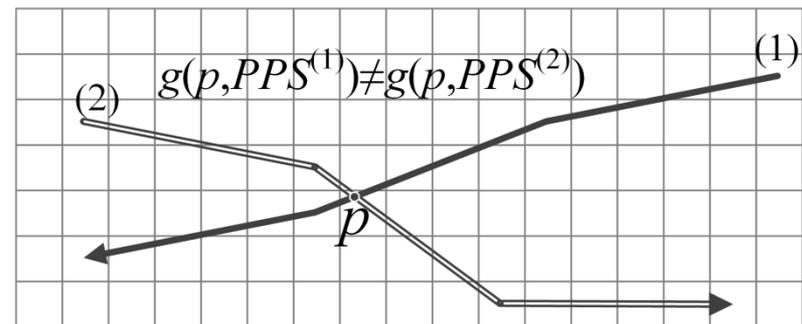
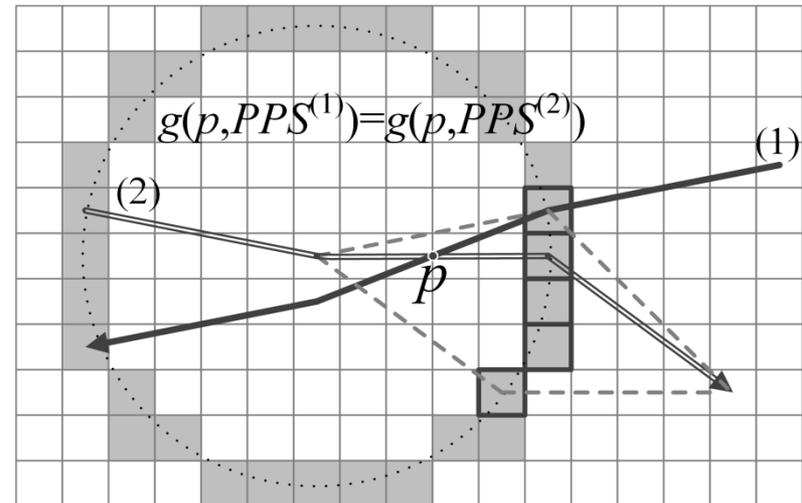
# Local re-planning

## Idea

Construct slightly deviating detour

## Implementation

- Draw a circle  
center - start point of the section being altered  
radius =  $\Delta$
- Prune cells that lead to sharp heading changes  
value of  $\alpha_m$  ( $0 < \alpha_m < 90$ ) is given as an input parameter
- Choose a cell to form a detour



# Conflicts resolution algorithm

---

## Algorithm 1: Conflicts resolution

---

**Input:**  $PS, \Delta, \alpha_m, wait$ ; **Output:**  $PS' \in \text{NoCON}$

```

1:  $\{HEAD, TAIL\} \leftarrow \text{FormHeadAndTail}(PS)$ 
2: while  $TAIL \neq \emptyset$  do
3:    $PPS^{(cur)} = \text{argmin}_{PPS \in TAIL} \text{NumberOfConflicts}(PPS, TAIL \cup HEAD)$ 
4:    $TAIL.\text{remove}(PPS^{(cur)})$ 
5:   while  $(\{e_v^{(cur)}, e_w^{(k)}\} \leftarrow \text{FindFirstConflict}(PPS^{(cur)}, HEAD)) \neq \emptyset$  do
6:      $\pi^{(new)} \leftarrow \text{ComputeLocalDetour}(PPS^{(cur)}, e_v^{(cur)}, PPS^{(k)}, \Delta, \alpha_m)$ 
7:     if  $\pi^{(new)} = \pi^{(cur)}$  then
8:        $t^{(cur)} += wait$ 
9:     else
10:       $\{e_t^{(new)}, e_s^{(m)}\} \leftarrow \text{FindFirstConflict}(PPS^{(new)}, HEAD)$ 
11:      if  $\{e_t^{(new)}, e_s^{(m)}\} = \emptyset$  or  $t > v$  then
12:         $\pi^{(cur)} = \pi^{(new)}$ 
13:      else
14:         $t^{(cur)} += wait$ 
15:    $HEAD.\text{add}(PPS^{(cur)})$ 
16: return  $HEAD$ 

```

Avoiding loops

### Idea

*HEAD* – “good boys” (set of conflict free p-solutions)

*TAIL* – “bad boys” (p-solutions that have at least one conflict with “good boys” each)

### Single step

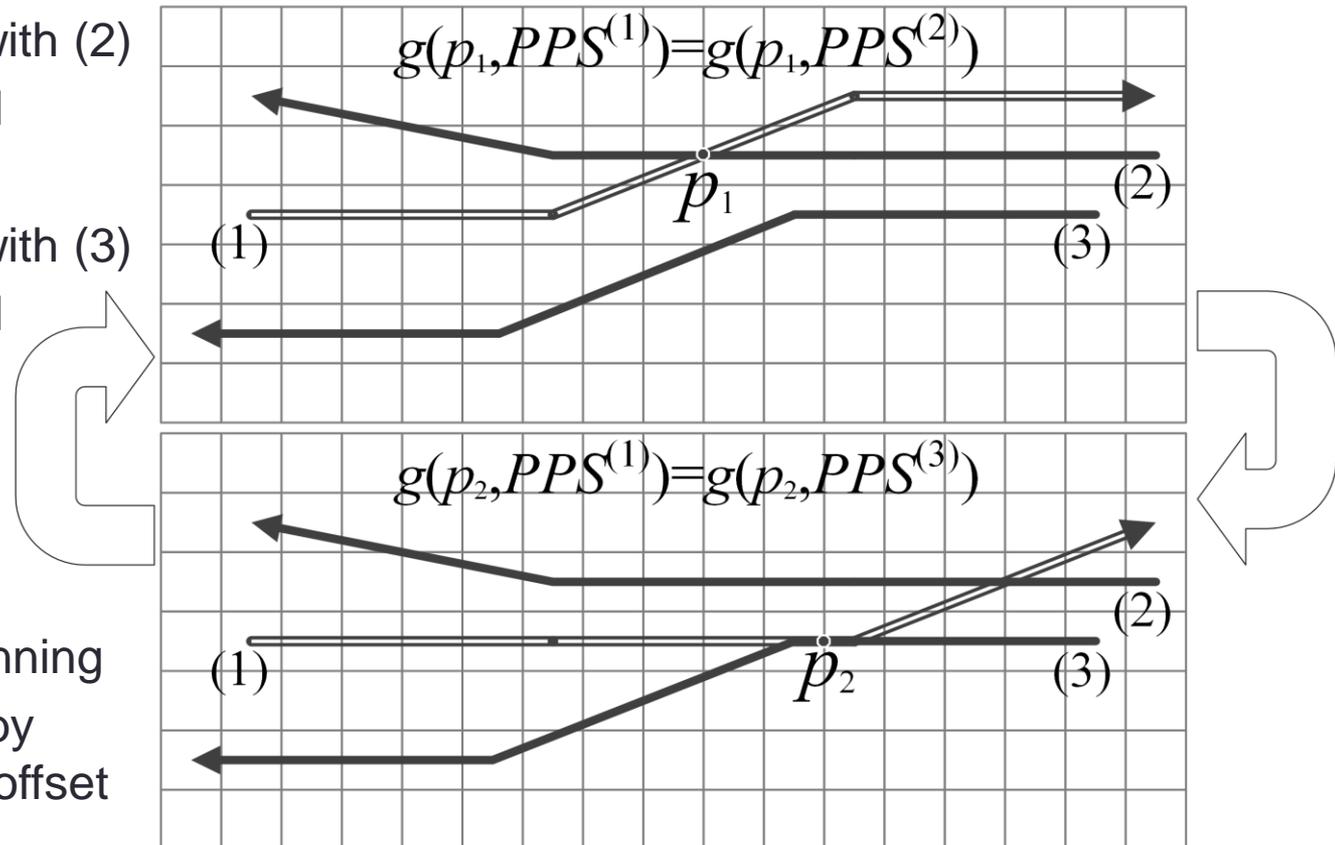
- Choose a “bad” p-solution
- Make it “good”
- Repeat

# Avoiding loops (Lines 10-14) :: Example

- (1) Has a conflict with (2)
- Conflict is resolved
- but
- (1) Has a conflict with (3)
- Conflict is resolved
- but
- It's a **deadlock**

## Solution

1. Roll back re-planning
2. Alter p-solution by incrementing time offset



# Algorithm's properties

- Can be seen as greedy version of CBS [Sharon et al., 2015]
  - Does not maintain conflict tree
  - Never backtracks
- Leads to suboptimal solutions
  - Do we really need optimal solution practically wise?
- Has not been studied theoretically yet
  - Strong arguments exist it is complete (formal proofs are to appear soon)
- Is effective in solving practical tasks
  - So let's proceed to this part...

[Sharon et al., 2015] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant. Conflict-based search for optimal multiagent path finding. *Artificial Intelligence Journal*, (218):40–66, 2015

# Experimental evaluation :: Input

**OpenStreetMaps** → 1347 x 1347 m  
fragments of real city environment

(20-25% buildings)

**100 fragments** in total

1 Fragment ~ **501 x 501 grid**

(1 cell ~ 2,7 m<sup>2</sup>)

**1 task** = allocation of **100** agents

2 types of **allocation**:

**Type 1** (sparse)

**Type 2** (dense)

**2** Type-1 tasks per grid

**2** Type-2 tasks per grid

**400** MAPF tasks in total



**Type 1**

**Type 2**

# Experimental evaluation :: Hardware and algorithms

## Hardware setup

Windows7-operated PC, iCore2 quad 2.5GHz, 2Gb RAM

## Conflict Resolution Algorithm

- $\Delta = 5$
- $\alpha_m = 25^\circ$
- *wait* = 5

## Individual Planners

- Theta \* (slightly modified to produce  $\Delta$ -paths)
- LIAN (produces  $\Delta$ -paths by default, heuristic weight = 2)

All algorithms were coded in C++ from scratch using the same data structures and techniques  
Source code is available online <http://www.pathplanning.ru/public/wompf-2016>

# Experimental evaluation :: Results (1/3)

## Conflicts distribution after path finding

	Theta*		LIAN	
	Type-1	Type-2	Type-1	Type-2
AgentConflicts	42.3	94.83	38.63	98.25
SectionConflicts	76.15	2 423.56	64.92	2 457.41

- agents' allocation really matters
- each agent has conflicts with more than one other agent (actually 2-5 as additional analysis of collected data shows)
- small number of agents produce large number of conflicts between each other.

# Experimental evaluation :: Results (2/3)

## Conflicts resolution stats

	Theta*		LIAN	
	Type-1	Type-2	Type-1	Type-2
	<i>Agents</i>			
Delayed	12.07	64.685	15.3	72.335
Replanned	14.73	58.565	7.77	64.09
Unchanged	79.54	34.33	80.56	26.44
	<i>Attempts to resolve SectionConflicts by</i>			
TimeOffset	21.025	746.085	26.575	1 176.19
Replan	30.935	1 442.185	12.7	638.3

The majority of **re-planning** attempts **fail** to produce a conflict-free p-solution  
Reason: (may be) conflict “moves forward” causing a sequence of unsuccessful re-planning attempts.  
Future work: to investigate the case and handle such resolution patterns.

# Experimental evaluation :: Results (3/3)

## Runtimes and solution costs

PF – path finding; CR – conflict resolution

		Theta*		LIAN	
		Type-1	Type-2	Type-1	Type-2
PF	Time(s)	7.3783	6.9426	9.5827	5.387
	Cost	46 926	46 722	49 636	49 651
CR	Time(s)	0.1466	0.804	0.1441	0.5926
	Cost	47 034 (+0.23%)*	50 477 (+8.04%)	49 772 (+0.27%)	55 566 (+11,91%)

- Resolving conflicts takes less than a second
- Cost overhead is low

(\*) Compared to the cost of initial solution **NOT** the cost of the optimal conflict-free solution

# Experimental evaluation :: Scalability test

- Up to **4 000 agents**
  - Resolving conflicts for more than 4 000 agents takes too much time
- **Theta\*** planner
- 10 maps
  - out of 100 previously used
  - 1 MAPF task per map
- **Type-1** start-goal allocation

## Number of conflicts after path finding

	200	500	1 000	2 000	4 000
AgentConflicts	120 (60%)	427 (85.4%)	962 (96.2%)	1 988 (99.4%)	3 997 (99.9%)
SectionConflicts	327	2 131	8 674	35 217	145 551

Almost every agent is involved in a conflict when number of agents exceeds 1 000

Number of conflicts exhibits quadratic growth in the number of agents.

# Experimental evaluation :: Scalability test

## :: Results (1/2)

### Conflicts resolution stats

	200	500	1 000	2 000	4 000
	<i>Agents</i>				
Delayed	40	191	543	1 395	3 258
Replanned	43	200	557	1 409	3 272
Unchanged	142	261	390	535	680
	<i>Attempts to resolve SectionConflicts by</i>				
PathOffset	86	815	5 162	32 757	176 158
Replan	123	1 158	5 835	28 557	128 253

### “attempt/success” ratio

	200	500	1 000	2 000	4 000
(po+r)/sc	0.64	0.92	1.27	1.74	2.1

po = “PathOffset”  
 r = “Replan”  
 sc = “SectionConflicts”



resolving a conflict leads to creating another one

# Experimental evaluation :: Scalability test

## :: Results (2/2)

Runtimes and solution costs' overheads

	200	500	1 000	2 000	4 000
Time(s)	0.62	6.19	49.69	466.8	5 072.42
Cost overhead *	0.47%	1.78%	5.61%	17,74%	47.59%

(\*) Compared to the cost of initial solution **NOT** the cost of the optimal conflict-free solution

Runtime ~ quadratic growth (in the number of agents).

Solution cost overhead ~ linear growth (in the number of agents).

From practical point of view the reasonable limit of the proposed approach (within the considered navigation scenario) is 1 000 of agents per grid.

# Summary

- MAPF with agents allowed to move into arbitrary directions on a grid map is studied
- Conflict resolution algorithm is proposed
  - Can be seen as greedy modification of CBS
  - Sub-optimal
  - Fast and effective in practice (at least for the scenarios we are interested in)

## Future work

- Experimental evaluation and comparison with CBS
- Theoretical study (proof of completeness)
- Modifications of the algorithm
  - Different prioritization strategies
  - Eliminating “conflict moving forward” problem
  - etc

# Questions?

Demo

<http://pathplanning.ru/public/wompf-2016/Demo/>