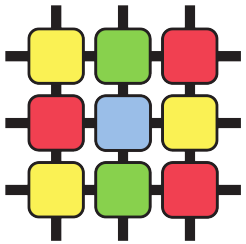
Domain-independent multi-agent planning and plan repair

Antonín Komenda
antonin.komenda@agents.fel.cvut.cz



Agent Technology Center,
Department of Computer Science and Engineering,
Faculty of Electrical Engineering,
Czech Technical University in Prague

Agent Technology Center (ATG)

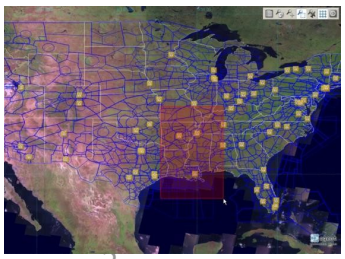
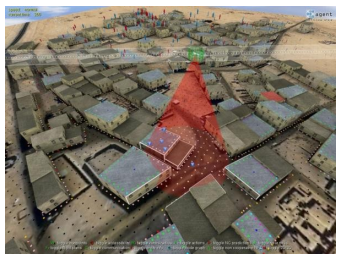
ATG is a university research center performing fundamental and applied research in the field of agent-based computing, multi-agent systems and agent technologies. There are 45 employees including 6 postdocs, 9 PhD students working in ATG.

Research and technology development is funded in parts by the Czech Technical University in Prague, by the Czech Government, European Commission, European and US industry and US research and defense.

Prof. Michal Pěchouček is the founder and head of ATG.

ATG R&D Activities

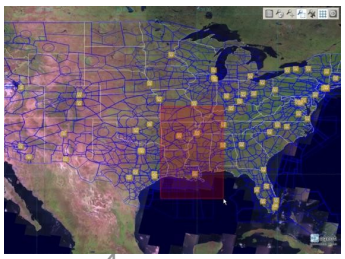
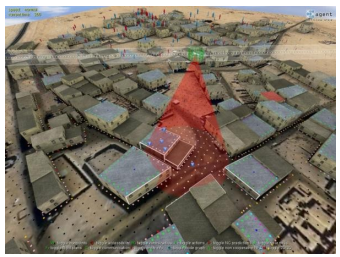
Air Traffic Management	Unmanned Aerial Syst	Critical Infrastructures	Cybersecurity
Mid term deconfliction	Sense & avoid	Game theory	Ensamble classification
Trajectory driven ATM	Tactical planning	Adversarial planning	Network behaviour analysis
NAS modeling	Trajectory planning	Maritime security	Opponent modeling
ATC modeling	HW deployment	Fare evasion in LA	Adversary evasion
	Mixed reality sim	Traffic & mobility nets	Steganography
	Human/Machine interf.		Social networks privacy



ATG R&D Activities

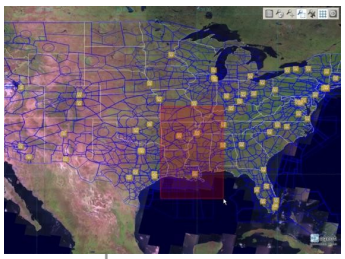
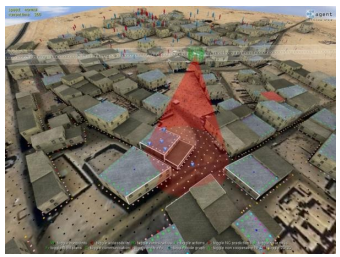
Air Traffic Management	Unmanned Aerial Syst	Critical Infrastructures	Cybersecurity
Mid term deconfliction	Sense & avoid	Game theory	Ensamble classification
Trajectory driven ATM	Tactical planning	Adversarial planning	Network behaviour analysis
NAS modeling	Trajectory planning	Maritime security	Opponent modeling
ATC modeling	HW deployment	Fare evasion in LA	Adversary evasion
	Mixed reality sim	Traffic & mobility nets	Steganography
	Human/Machine interf.		Social networks privacy



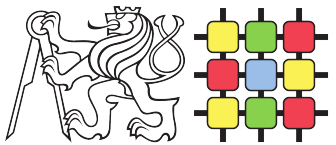
ATG R&D Activities

Air Traffic Management	Unmanned Aerial Syst	Critical Infrastructures	Cybersecurity
Mid term deconfliction	Sense & avoid	Game theory	Ensamble classification
Trajectory driven ATM	Tactical planning	Adversarial planning	Network behaviour analysis
NAS modeling	Trajectory planning	Maritime security	Opponent modeling
ATC modeling	HW deployment	Fare evasion in LA	Adversary evasion
	Mixed reality sim	Traffic & mobility nets	Steganography
	Human/Machine interf.		Social networks privacy



Domain-independent multi-agent **planning**
State of the art overview



Distributed Graphplan

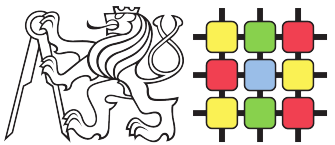
Publications:

- Mark Iwen, Amol Dattatraya Mali: *Distributed Graphplan*, Proceedings of ICTAI, 2002.
- Damien Pellier: *Distributed Planning Through Graph Merging*, Proceedings of ICAART, 2010.

Key concepts and ideas:

- Decomposition of a planning problem (namely goal and action sets without loosing completeness)
- Automatic decomposition process based on interaction-graph
- Merging process of the resulting plans exploiting previously used planning graphs

- Planning graph merging (inter-agent communication of *threats* and *promotions*)
- Individual solution extraction uses CSP (extensible to DCSP)
- Additional planning graph expansions, if a merged solution cannot be found



Based on Multi-agent DCSP+Planning

Publications:

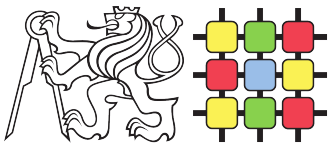
- Ronen I. Brafman, Carmel Domshlak: *From One to Many: Planning for Loosely Coupled Multi-Agent Systems*, Proceedings of ICAPS, 2008 (Best Paper Award).
- Raz Nissim, Ronen I. Brafman, Carmel Domshlak: *A General, Fully Distributed Multi-Agent Planning Algorithm*, Proceedings of AAMAS, 2010.

Key concepts and ideas:

- Complexity analysis of deterministic domain-independent multi-agent planning
- Decomposition of the problem based on public and private actions
- DCSP-based coordination of public sequences of actions (intra-, inter-agent constraints)
- Forward-search-based planning of private action sequences (FastForward planner)
- MA-STRIPS planning complexity upper-bound:

$$f(\mathcal{I}) \cdot \exp(\delta) + \exp(\delta\omega)$$

- No direct exponential dependence on (i) number of agents, (ii) size of the MA planning problem and size of solving joint plan, and (iii) length of individual agent plans



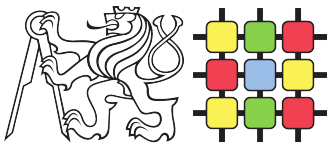
Multi-agent A*

Publications:

- Raz Nissim, Ronen Brafman: *Multi-Agent A* for Parallel and Distributed Systems (Extended Abstract)*, Proceedings of AAMAS, 2012.
- Raz Nissim, Udi Apsel, Ronen Brafman: *Tunneling and Decomposition-Based State Reduction for Optimal Planning*, Proceedings of ECAI, 2012.

Key concepts and ideas:

- Motivation: unknown ways how to use forward search heuristics in DCSP+Planning
- Based on the decomposition of private and public actions
- The state space is "naturally" pruned during the search
- Communication-based search expansion to other agents
- Parallel search on the decomposed problems
- Used with state-of-the-art forward search heuristics (Merge&shring, LM-cut)
- So far promising results (evaluation only considering computational metrics)



Beyond the state of the art

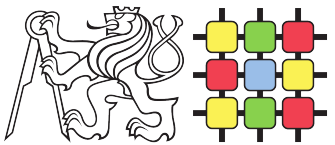
Challenges:

- Domain-independent heuristics specific for multi-agent forward search techniques (Raz Nissim)

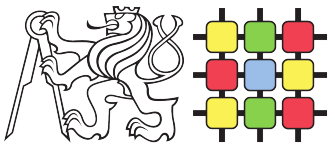
- Communicational complexity is in multi-agent settings often more important than computational complexity
- Resurrect older distinguished techniques and after adaptation for multi-agent settings evaluate their communicational (and computational) complexity

Approach:

- Push a centralized state-of-the-art Graphplanning algorithm (namely LPG planner) with help of the ideas in Distributed Graphplan towards Distributed LPG
- Thoroughly study problem of heuristics transfer from forward search planning to DCSP+(Graph)planning-based techniques
- Study relations of communicational and computational complexities of such heuristics



Domain-independent multi-agent **plan repair**
Towards my PhD thesis



Problem statement

Hypothesis: *Multi-agent plan repair approaches producing more preserving repairs than replanning tend to generate lower communication overhead for tightly coupled multi-agent problems.*

General:

- domain-independent multi-agent setting
- cooperative agents
- achieving a joint objective
- synchronized joint actions

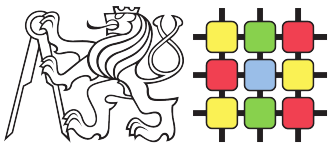
$$P = (\; A_1 : P_1 = \begin{array}{|c|c|c|c|c|} \hline a_1^1 & a_2^1 & a_3^1 & \cdots & a_m^1 \\ \hline \end{array} \\ A_2 : P_2 = \begin{array}{|c|c|c|c|c|} \hline a_1^2 & a_2^2 & a_3^2 & \cdots & a_m^2 \\ \hline \end{array} \\ \vdots \\ A_n : P_n = \begin{array}{|c|c|c|c|c|} \hline a_1^n & a_2^n & a_3^n & \cdots & a_m^n \\ \hline \end{array} \;)$$

Why to repair:

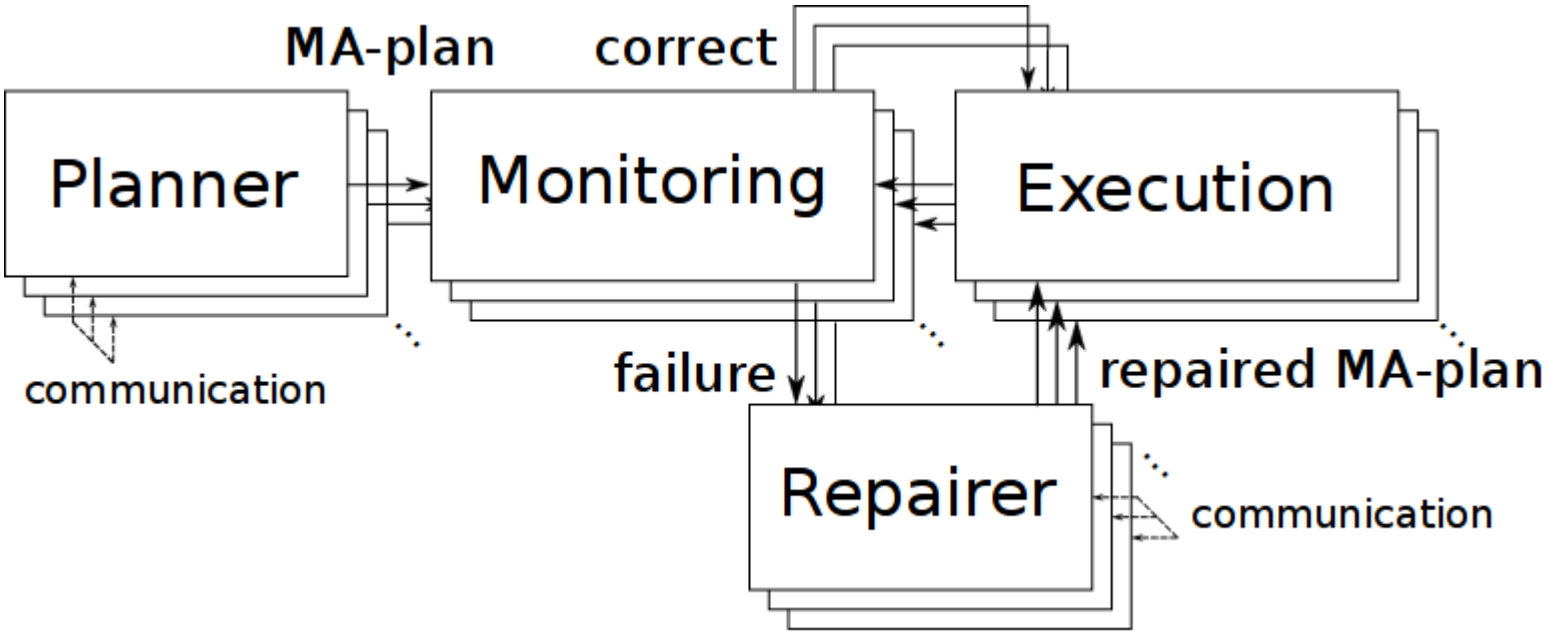
- in dynamic environment (disturbing events during execution)
  - state fact perturbations
  - action failures
  - parametrization of the failures is *a priori* unknown

Challenges:

- requires significant coordination and communication efforts
- communication complexity over computational complexity (vs. classical planning).

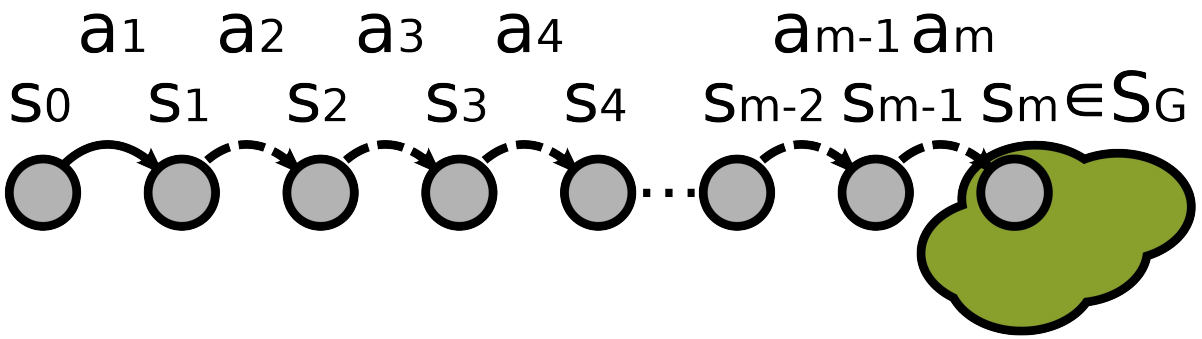


Continuous planning architecture



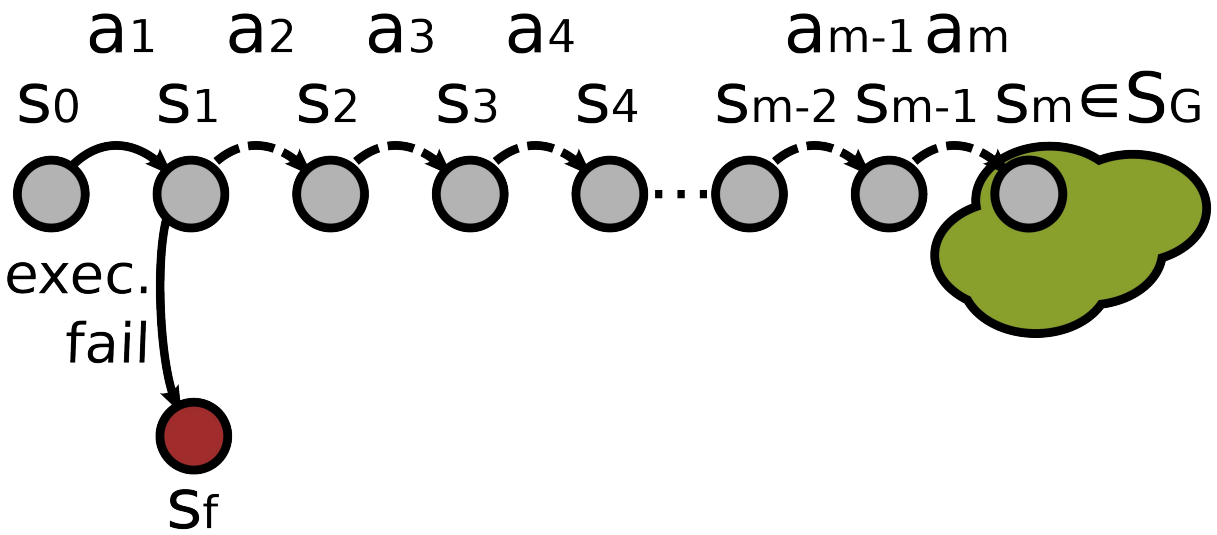
Re-planning from scratch

- Baseline approach
- MA-plan: $P = (a_1, a_2, a_3, \ldots, a_m)$
- Presumed execution sequence: $(s_0, s_1, \ldots, s_m)$



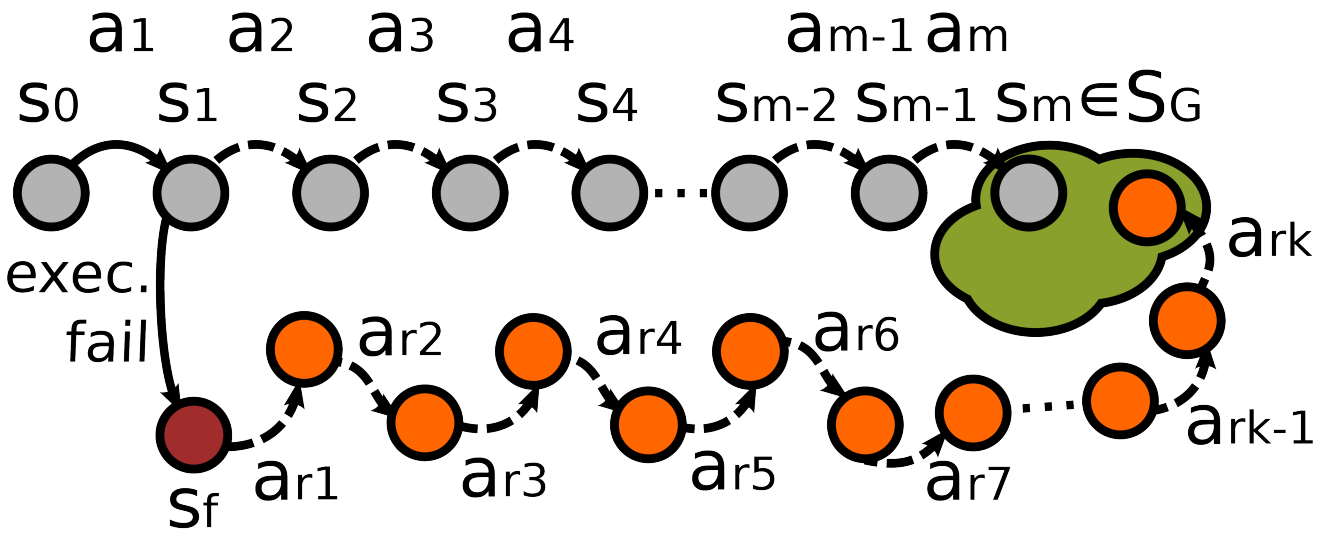
Re-planning from scratch

- Baseline approach
- MA-plan: $P = (a_1, a_2, a_3, \ldots, a_m)$
- Presumed execution sequence: $(s_0, s_1, \ldots, s_m)$



Re-planning from scratch

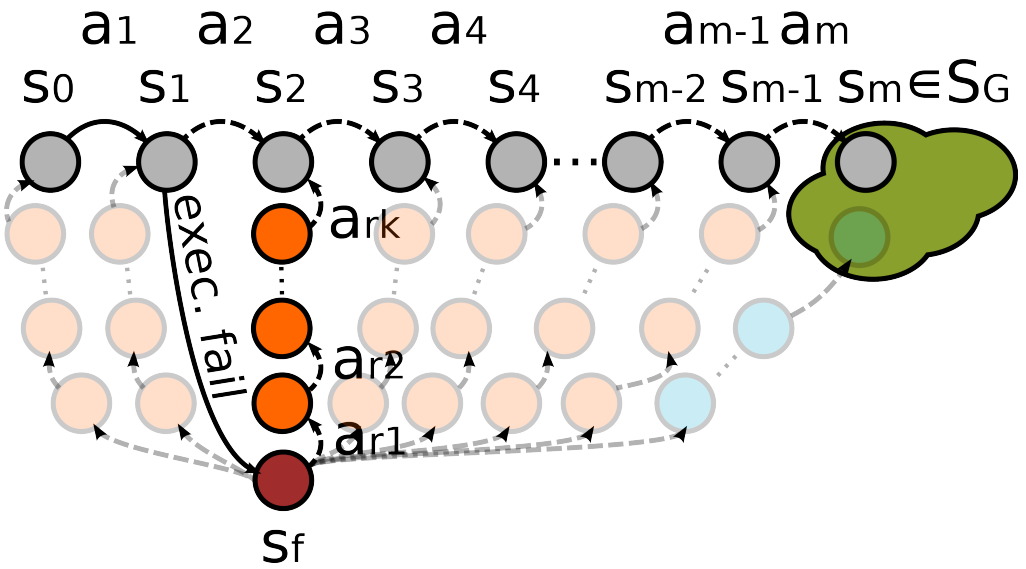
- Baseline approach
- MA-plan: $P = (a_1, a_2, a_3, \ldots, a_m)$
- Presumed execution sequence: $(s_0, s_1, \ldots, s_m)$
- Repair plan: $P^* = (a_{r1}, a_{r2}, a_{r3}, \ldots, a_{rk})$



BackOnTrack & Lazy Repair Techniques

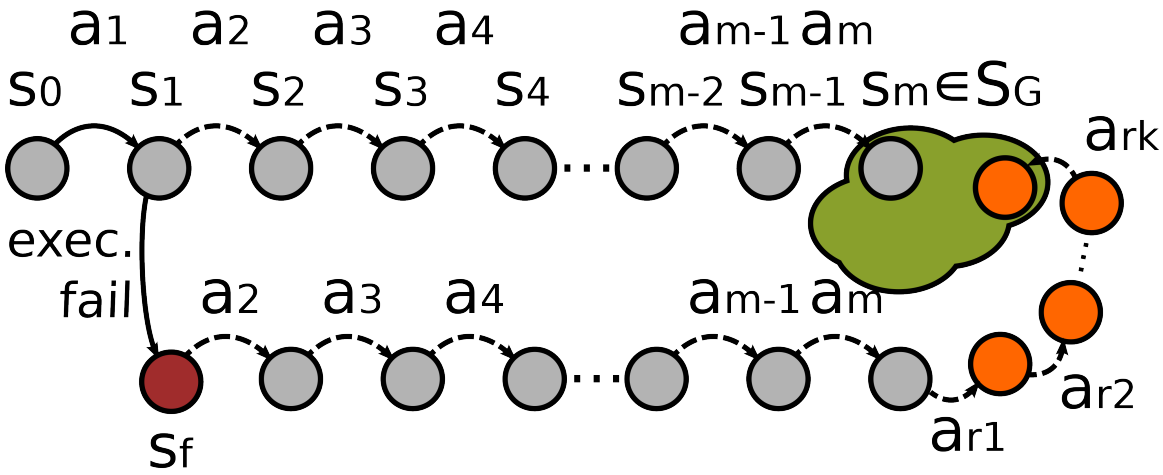
Back On Track Plan Repair

- repairing plan as a prefix
- complete (ends as replanning)

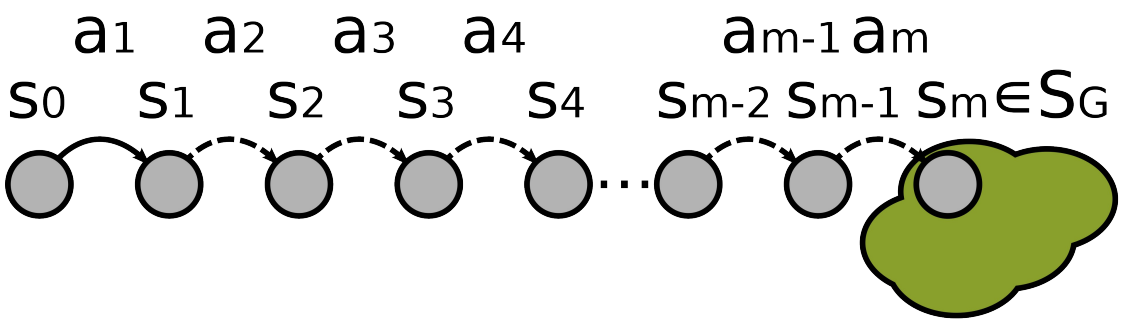


Lazy Plan Repair

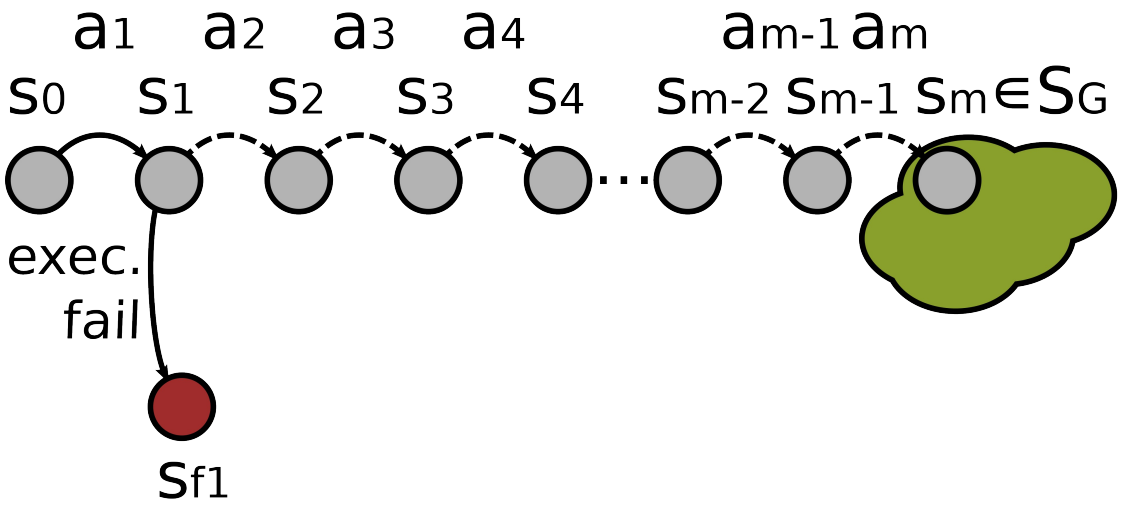
- repairing plan as a suffix
- incomplete (in general)
- complete (if irreversible actions are not allowed)



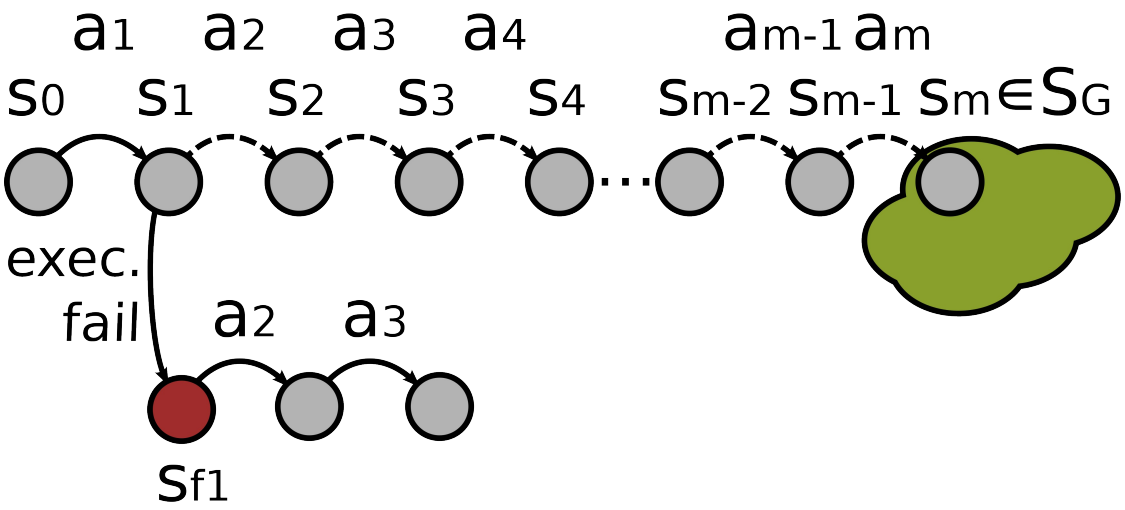
Repeated Lazy Repair Technique

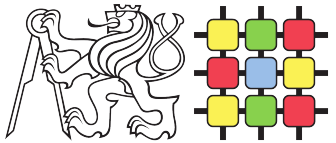


Repeated Lazy Repair Technique

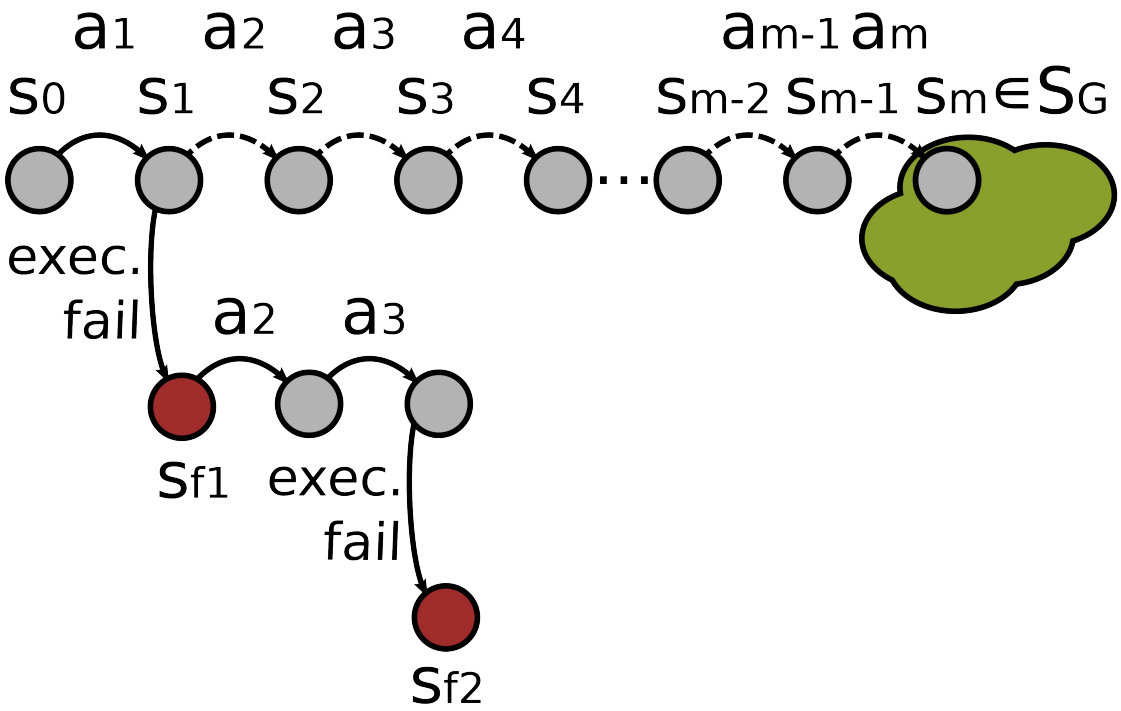


Repeated Lazy Repair Technique

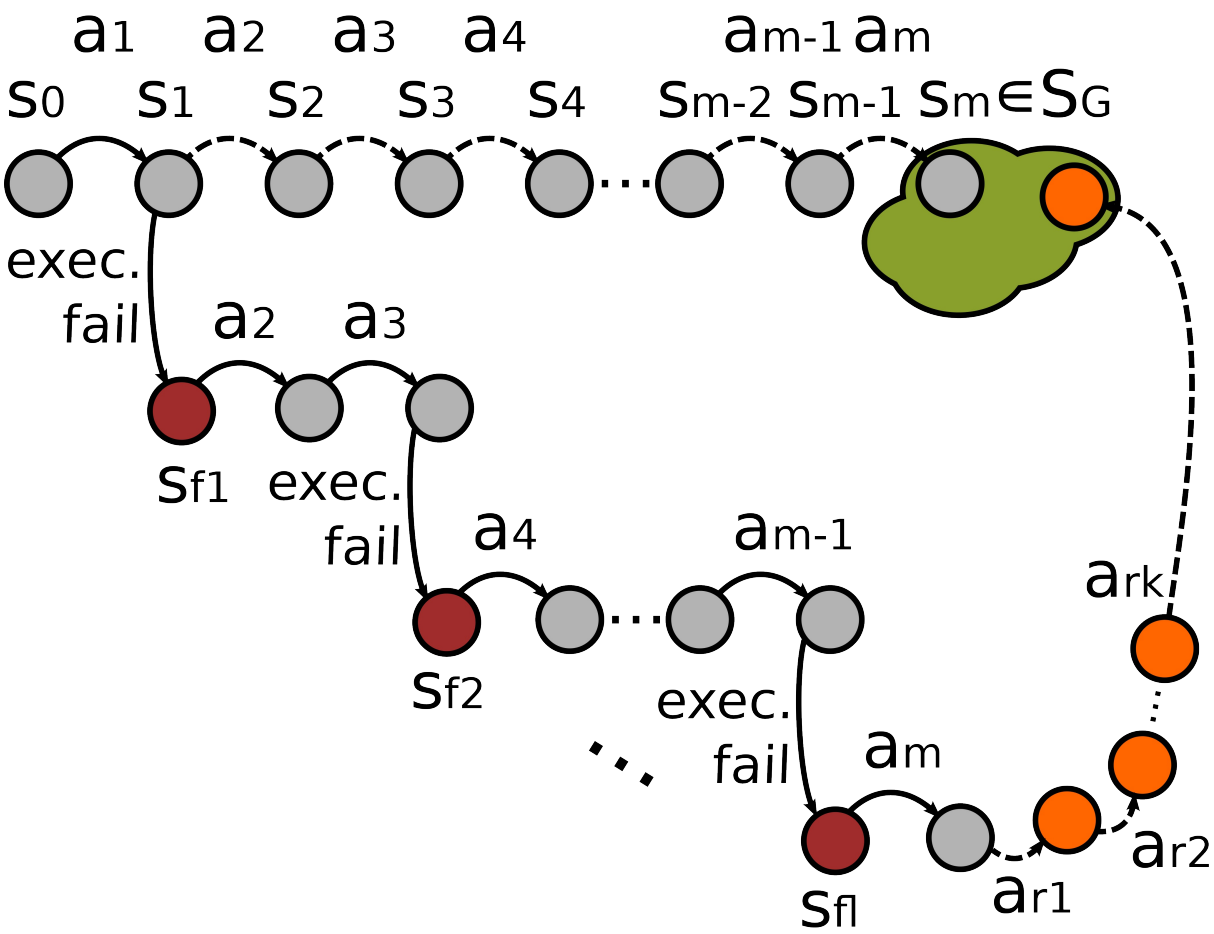




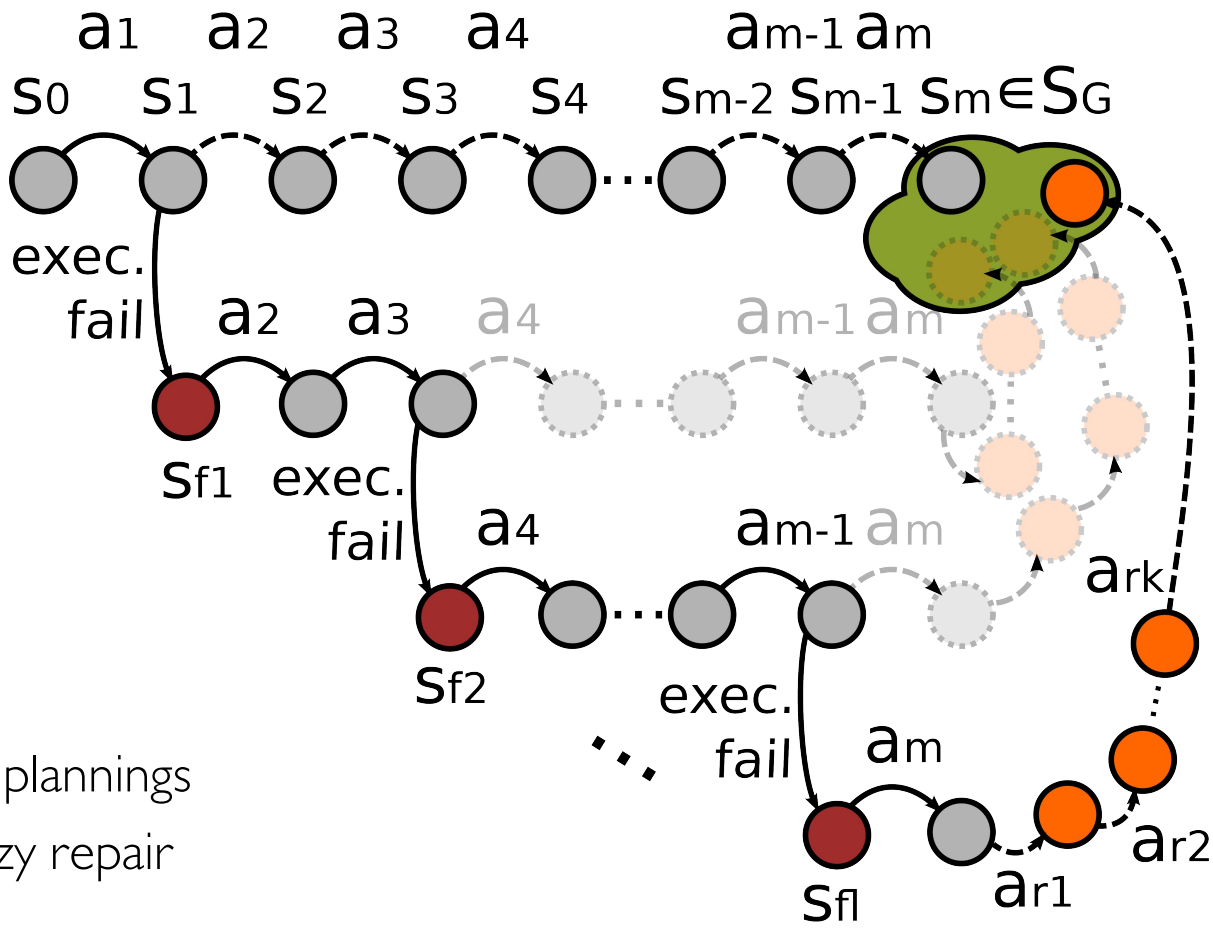
Repeated Lazy Repair Technique



Repeated Lazy Repair Technique



Repeated Lazy Repair Technique



Repeated Lazy Plan Repair

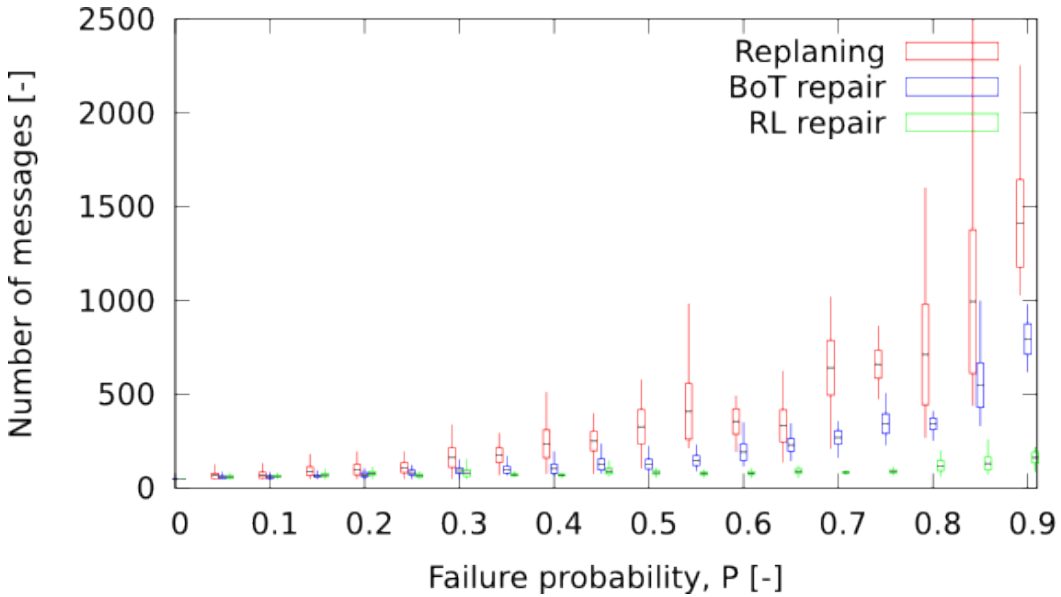
- repairing plan as a suffix
- reduces number of unused plannings
- inherits properties of the lazy repair

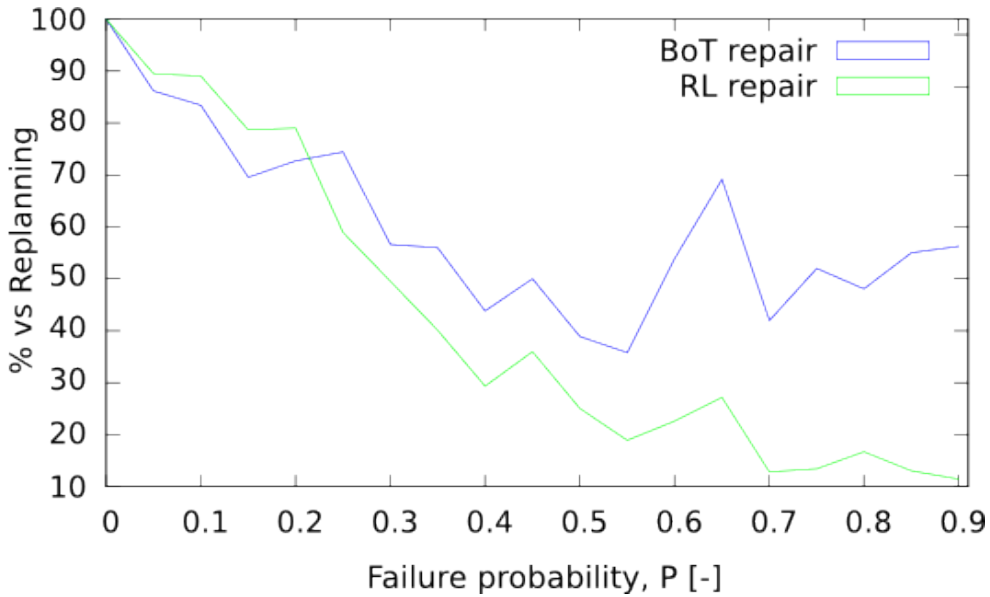
Experimental results (P=0.3)

Domain	Agents	Repairing time [ms]			No. of messages [-]			Communication [kB]			Exec. length [-]		
		RL	BoT	Re	RL	BoT	Re	RL	BoT	Re	RL	BoT	Re
logistics	2	116.1	115.3	145.6	8.2	13.9	10.9	1.7	2.0	2.3	14.3	8.8	10.7
	3	149.6	178.0	257.2	18.0	33.7	59.5	3.6	5.0	6.2	18.7	13.2	15.9
	4	**162.1**	266.2	**479.3**	29.1	89.5	114.7	**6.6**	13.3	**26.5**	21.5	15.5	18.1
logistics (parallel)	5	74.0	73.7	81.3	21.8	23.2	22.5	4.4	4.4	5.0	14.6	13.5	14.9
	6	84.1	126.0	110.7	32.5	49.6	60.9	6.8	9.3	9.6	12.4	11.4	12.0
cooperative pathfiding	2	115.6	23.9	93.2	2.2	2.4	2.2	0.6	0.6	0.6	3.2	2.8	2.6
	3	261.5	28.1	374.6	20.0	12.9	12.7	4.5	0.7	3.4	3.1	2.4	3.4
	4	19568.6	**29.4**	**6529.8**	14k	20.0	19.1	3002.0	**0.9**	**5.1**	4.0	2.3	3.3
rovers	2	179.8	381.3	249.8	7.4	13.0	10.0	1.9	2.7	2.7	20.3	14.7	14.5
	3	300.6	374.5	489.9	13.6	15	22.4	3.6	3.3	6.3	19.5	12.5	14.5
	4	634.4	798.3	650.5	30.0	42.5	29.1	8.3	7.6	8.9	22.1	15.3	13.6
satellites	2	67.5	80.3	67.5	4.9	6.2	3.8	1.1	1.3	0.9	7.5	5.8	5.1
	3	81.9	126.9	139.9	7.6	15.8	15.3	1.8	2.9	3.7	7.0	6.9	6.5
	4	144.4	139.2	176.5	14.5	21.8	18.2	3.6	3.8	4.7	6.9	6.7	5.6
	5	232.9	154.5	222.3	17.7	30.0	25.9	4.8	5.6	7.3	6.7	5.7	5.5
	6	48093.9	1452.8	23027.7	32.5	57.3	38.2	9.4	11.6	11.7	7.1	6.9	5.7

Experimental results (logistics, 4 agents)

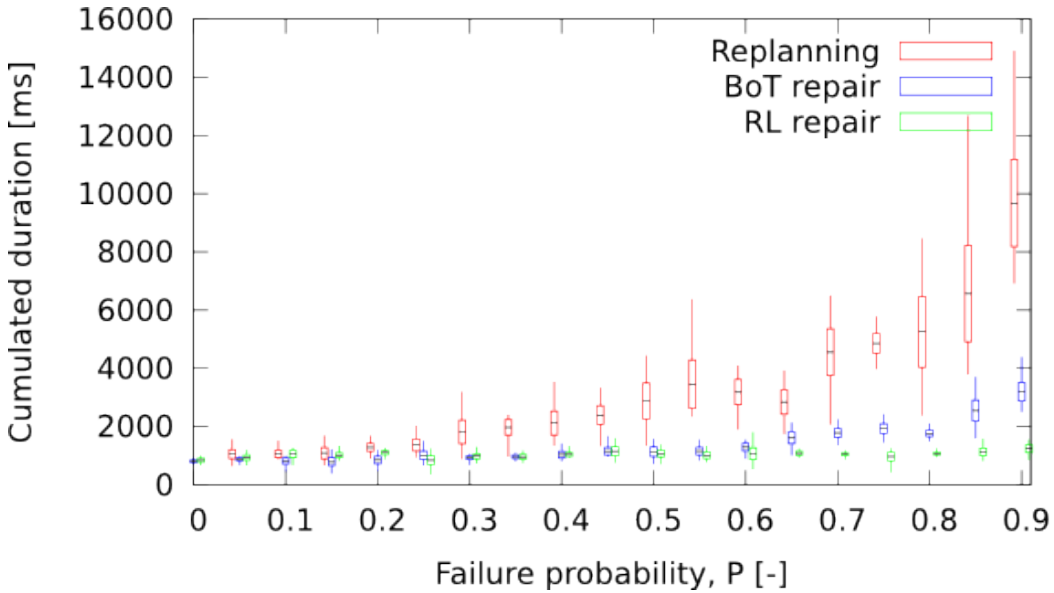
Communication

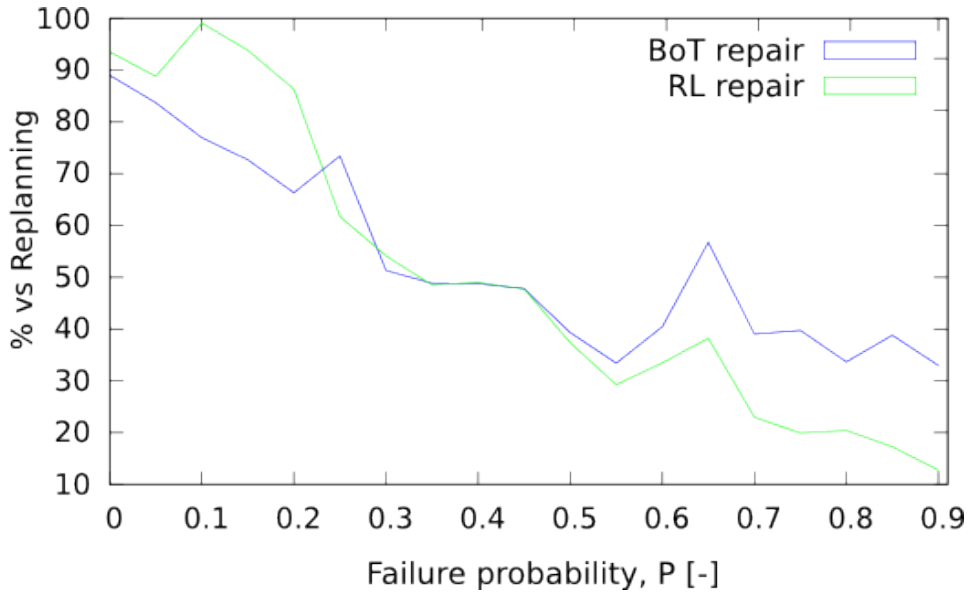




Experimental results (logistics, 4 agents)

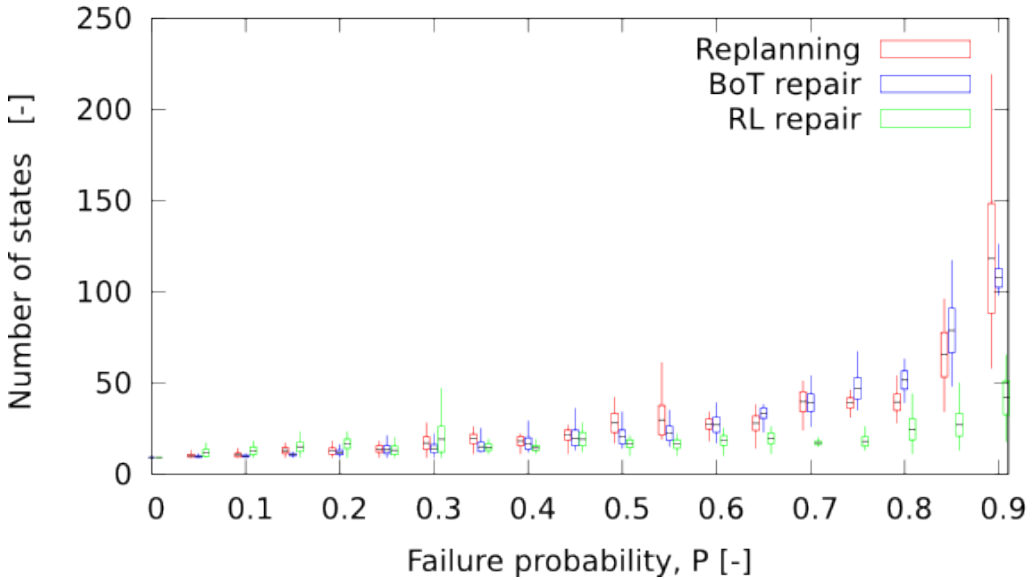
Planning time

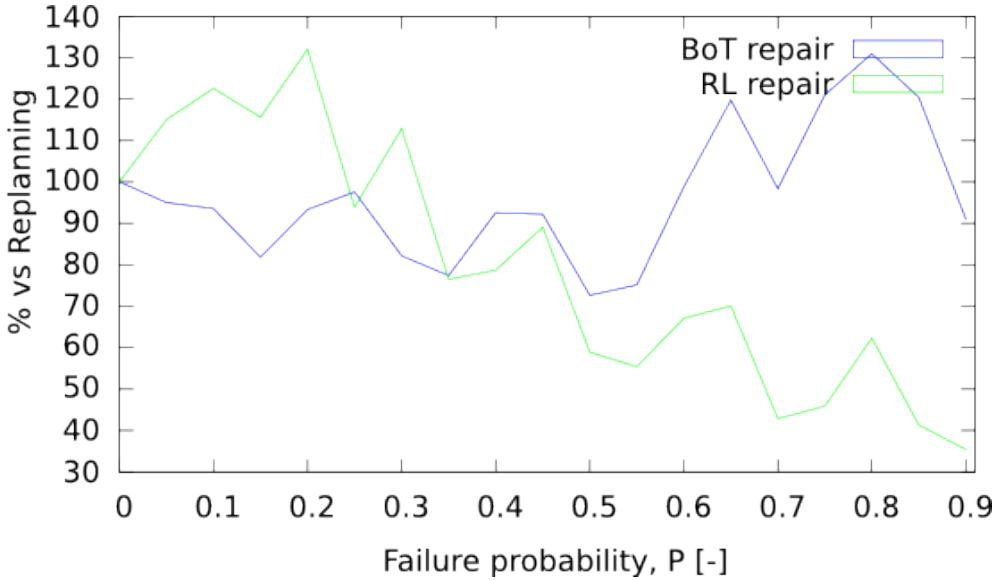




Experimental results (logistics, 4 agents)

Execution length





Generalized plan repair

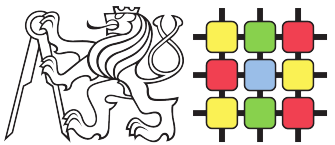
Localized:

- Concept of *k*-public actions
- Iteratively increasing set of involved agents based on their *k*-public actions

Repair strategy parametrization (∂, a, f, g):

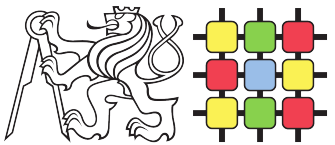
- ∂: number of coordination points
- a: number of agents
- f: number of actions reused from a postfix of the original plan
- g: number of actions reused from a prefix of the original plan

- *Replanning*: $((1,\ldots,\partial_M), a_M, 0, 0)$
- *Back On Track*: $((1,\ldots,\partial_M), a_M, m, 0), ((1,\ldots,\partial_M), a_M, m - 1, 0), \ldots, ((1,\ldots,\partial_M), a_M, 0, 0)$
- *Lazy*: $((1,\ldots,\partial_M), a_M, 0, m)$



Preliminary results (generalized repair)

method	time	pl	mess	NCCCs	exl	pll	a	commB
logistics-e3-ladT.dat	286	6	26	127	13	12	3	4478
logistics-e3-dlaT.dat	288	6	27	126	13	12	3	4526
logistics-e3-dalT.dat	288	6	27	127	13	12	3	4546
logistics-e3-ldaT.dat	292	6	28	128	13	12	3	4579
logistics-e3-adlT.dat	289	6	28	126	13	12	3	4605
logistics-e3-aldT.dat	290	6	28	127	13	12	3	4613
logistics-e3-ldaL.dat	506	18	56	277	23	22	3	6025
logistics-e3-dlaL.dat	824	51	144	282	24	23	3	9136
logistics-e3-dlR.dat	670	22	115	472	13	34	3	10181
logistics-e3-dR.dat	498	13	108	732	16	45	3	10531
logistics-e3-dL.dat	684	13	220	383	25	24	3	19301
logistics-e3-ldL.dat	700	13	227	403	26	25	3	19853
logistics-e3-dlL.dat	717	26	289	420	26	25	3	20081
logistics-e3-dalL.dat	1555	72	327	702	23	22	3	24127
logistics-e3-ladL.dat	2136	26	407	220892	23	22	3	43552
logistics-e3-ldR.dat	3996	35	937	532599	13	35	3	99988
logistics-e3-dlT.dat	1386	6	818	404	13	12	3	102244
logistics-e3-ldT.dat	1401	6	823	414	13	12	3	102832
logistics-e3-dT.dat	1427	6	839	411	13	12	3	104707
logistics-e3-adlL.dat	12296	138	3473	1961140	23	22	3	362068
logistics-e3-aldL.dat	14077	138	3845	2311199	24	23	3	405852



Thank you.