Mutex Reasoning in Cooperative Path Finding Modeled as Propositional Satisfiability

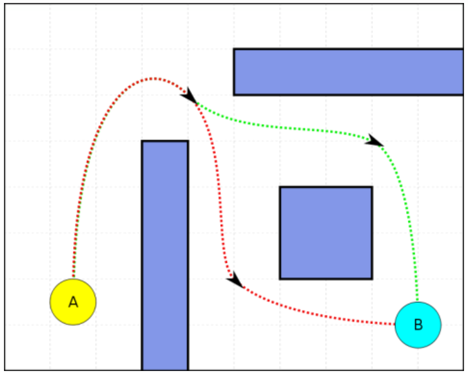
Pavel Surynek

Faculty of Mathematics and Physics
Charles University in Prague
Czech Republic

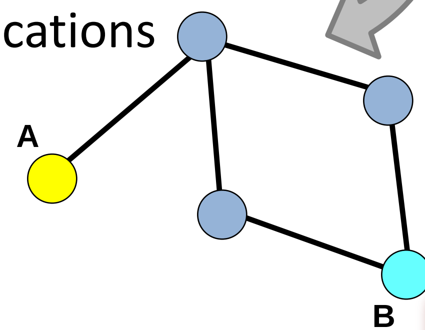
IROS 2013, Tokyo, Japan

Cooperative Path-Finding (CPF)

- Robots can **move only**
  - each robot needs to relocate itself
  - initial and goal location
- **Physical limitations**
  - robots must **not collide** with each other
  - must avoid **obstacles**
- **Abstraction**
  - environment – **undirected graph** G=(V,E)
    - vertices V – **locations** in the environment
    - edges E – **passable** region between neighboring locations
  - robots – entities placed in vertices
    - **at most one** robots per vertex
    - **at least one** vertex empty to allow movements

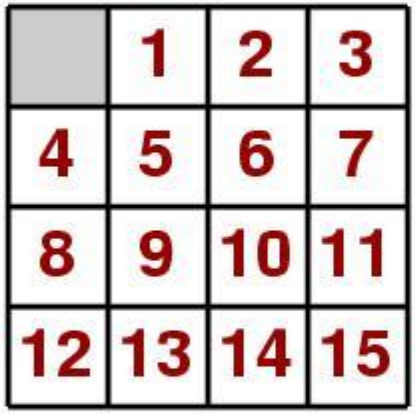


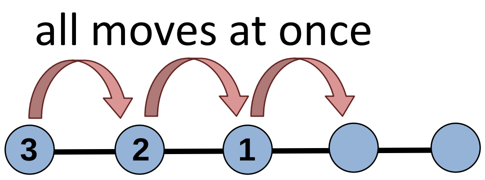
abstraction



CPF Formally

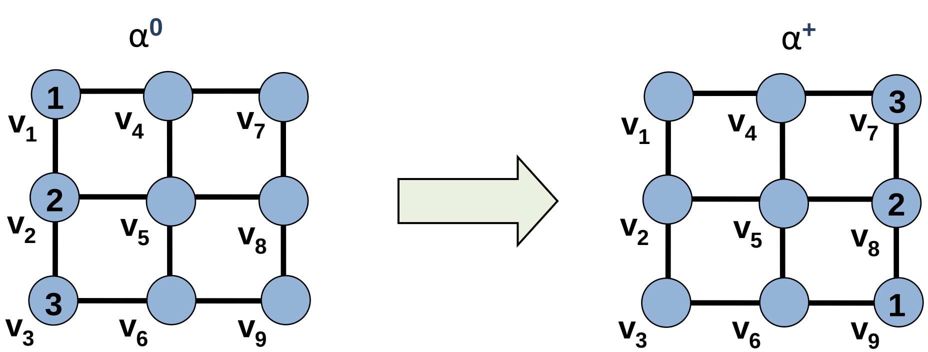
- A **quadruple** (G, R, α^0, α^+), where
  - G=(V,E) is an **undirected graph**
  - R = {r_1,r_2,...,r_μ}, where μ<|V| is a **set of robots**
  - α^0: R $\rightarrow$V is an **initial arrangement of robots**
    - uniquely invertible function
  - α^+: R $\rightarrow$V is a **goal arrangement of robots**
    - uniquely invertible function
- **Time** is discrete – time steps
- **Moves/dynamicity**
  - depends on the model
  - **Robot moves** into unoccupied neighbor
    - no other robot is entering the same target
  - sometimes **train-like** movement is allowed
    - only the leader needs to enter unoccupied vertex





Solution to CPF

- **Solution** of (G, R, α^0, α^+)
  - sequence of arrangements of robots
  - *(i+1)*-th arrangement obtained from *i*-th by legal moves
  - **the first arrangement** determined by α^0
  - **the last arrangement** determined by α^+
    - all the robots in their goal locations
- The length of solution sequence = **makespan**
  - **optimal/sub-optimal** makespan



Solution of an instance of cooperative path-finding on a graph with R={1,2,3}

makespan=7

[$\mathbf{v_1}$, v_4, v_7, v_8, v_9, v_9, $\mathbf{v_9}$]
[$\mathbf{v_2}$, v_2, v_1, v_4, v_7, v_8, $\mathbf{v_8}$]
[$\mathbf{v_3}$, v_3, v_3, v_2, v_1, v_4, $\mathbf{v_7}$]

Time step: **1 2 3 4 5 6 7**

Motivation for CPF

- **Container rearrangement**
  (robot = container)

- **Heavy traffic**
  (robot = automobile (in jam))

- **Data transfer**
  (robot = data packet)

- **Ship avoidance**
  (robot = ship)

CPF as SAT

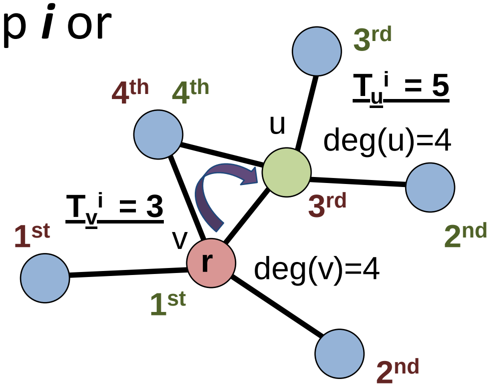
- **SAT = propositional satisfiability**
  - a formula ϕ over 0/1 (false/true) variables
  - Is there a valuation under which ϕ evaluates to 1/true?
    - **NP-complete** problem

$(x \vee \neg y) \wedge (\neg x \vee y)$
Satisfiable for $x = 1$, $y = 1$

- **SAT solving** and **CPF**
  - powerful SAT solvers
    - MiniSAT, clasp, glucose, glue-MiniSAT, crypto-MiniSAT, …
    - intelligent search, learning, restarts, heuristics, …
  - **CPF ⇒ SAT**
    - all the advanced techniques **employed for free**
- **Translation**
  - given a CPF $\Sigma=(G, R, \alpha^0, A^+)$ and a **makespan *k***
  - **construct** a formula ϕ
    - satisfiable iff Σ has a solution of makespan *k*

Encoding CPF as IP

- **How to encode** a question if there is a solution of makespan ***k***
  - Encode arrangements of robots at steps 1,2…,k
  - **Step 1** … α^0
  - **Step k** … α^+ / A^+
- **Integer variables** modeling step ***i***
  - $A_v^i \in \{0,1,2,\ldots,\mu\}$
    - $A_v^i = j$ if robot r_j is located in vertex ***v*** at time step ***i*** or
    - $A_v^i = 0$ if ***v*** is empty at time step ***i***
  - $T_v^i \in \{0,1,2,\ldots,2\deg(v)\}$
    - $0 < T_v^i \leq \deg(v)$ if an robot leaves ***v*** into the (T_v^i)-th neighbor
    - $\deg(v) \leq T_v^i \leq 2\deg(v)$ if an robots enters ***v*** from the $((T_v^i)-\deg(v))$-th neighbor
    - $T_v^i = 0$ if no action taken in ***v***



- Don't forget constraints – valid transitions between time-steps

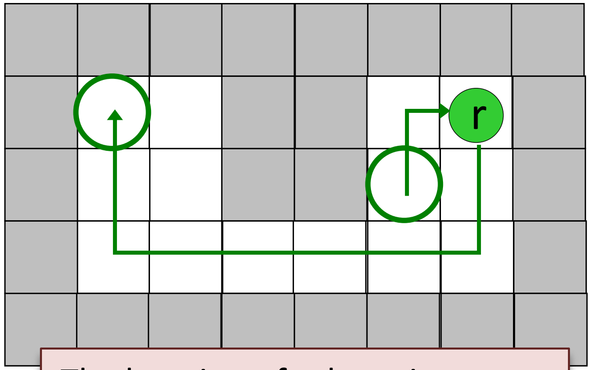
Encoding CPF as SAT

- **Integer variables**
  - replace with bit vectors
  - for example $\mathbf{A_v^i} \in \{0,1,2,\ldots, \mu\}$
    - replaced with $\lceil \mathbf{log_2(\mu+1)} \rceil$ propositional variables
    - extra states are forbidden
- $\Rightarrow$ **Compact representation**
  - smaller than in SAT-based domain-independent planners
  - knowledge compilation – distance heuristic, mutex reasoning

|A| 4-connected grid 8×8	Makespan	**SATPLAN** encoding		**SASE** encoding		INVERSE encoding	
		|Variables|	|Clauses|	|Variables|	|Clauses|	|Variables|	|Clauses|
4	8	5.864	55.330	11.386	53.143	5.400	38.800
8	8	10.022	165.660	19.097	105.724	5.920	48.224
12	8	14.471	356.410	26.857	168.875	5.920	46.176
16	10	30.157	1.169.198	51.662	372.140	8.122	76.192
24	10	43.451	2.473.813	73.101	588.886	8.122	71.072
32	14	99.398	8.530.312	157.083	1.385.010	12.396	137.120

Knowledge Compilation

- Heuristics **directly built-in into the encoding**
  - **distance** heuristic
    - locations unreachable in a given time are forbidden
    - search space **reduced**
  - **mutex** reasoning
    - robots are treated pair-wise
    - computationally difficult



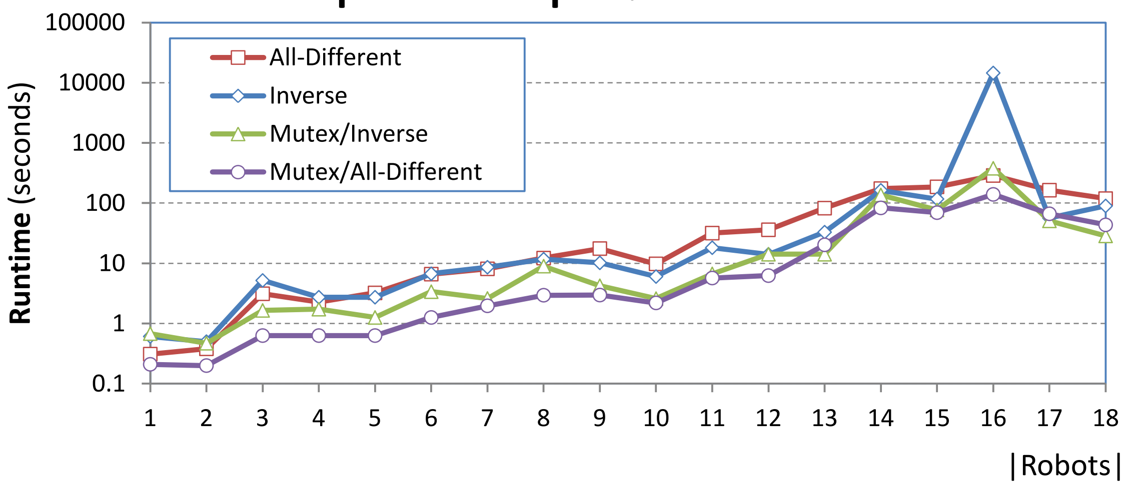
The location of robot **r** is allowed in steps **< k-9** and **> 2**

Although locations of robots **p** and **q** are allowed in steps **< k-11** by distance heuristics, they cannot occur in steps **>= k-20**

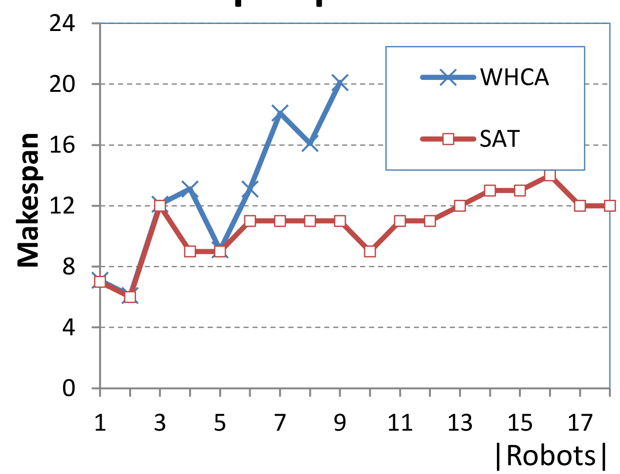
Experimental Evaluation

- **Experimental setup**
  - 4-connected **grids** of size 4×4 to 8×8
  - **random** initial and goal arrangement
  - 20% of cells - obstacles
  - with and without **knowledge compilation**

***Runtime* | Grid 8×8 |** 20% obstacles



Makespan|Grid 8×8



Conclusions and Observations

- Advantages
  - **search techniques**
    - advanced search techniques from SAT solvers employed (almost) for free
  - **modularity**
    - exchangeable modules – SAT solver, encoding
  - **parallelism**
    - knowledge compilation can be done in parallel
- Disadvantages
  - **energy extensive solutions**
    - robots move too much
  - **size of encoded instances**
    - large graphs
    - many time steps