Lessons Learned from the Effort to Solve Cooperative Path-Finding Optimally

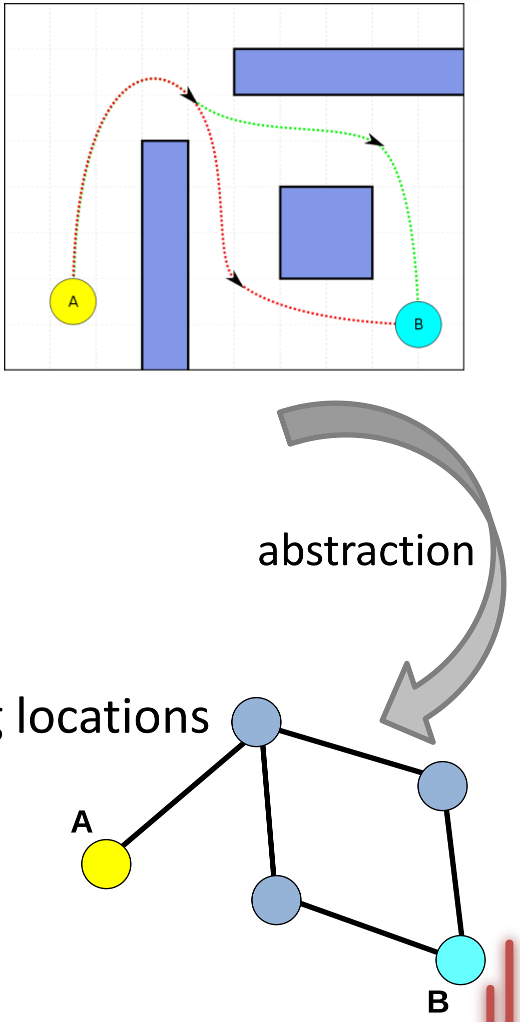
Pavel Surynek

Faculty of Mathematics and Physics
Charles University in Prague
Czech Republic

CJS 2014, Kitakyushu, Japan

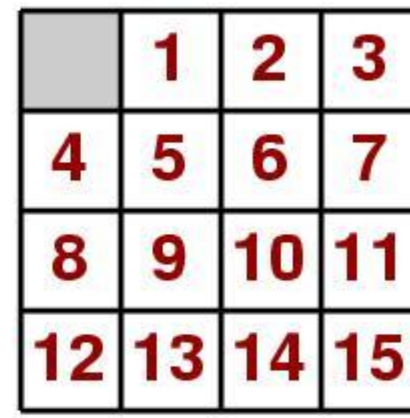
Cooperative Path-Finding (CPF)

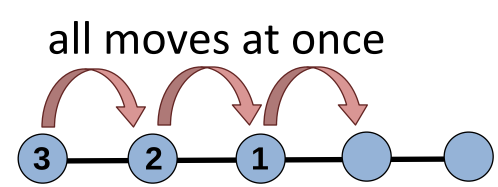
- agents can **move only**
  - each agent needs to relocate itself
  - initial and goal location
- **Physical limitations**
  - agents must **not collide** with each other
  - must avoid **obstacles**
- **Abstraction**
  - environment – **undirected graph** G=(V,E)
    - vertices V – **locations** in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



CPF Formally

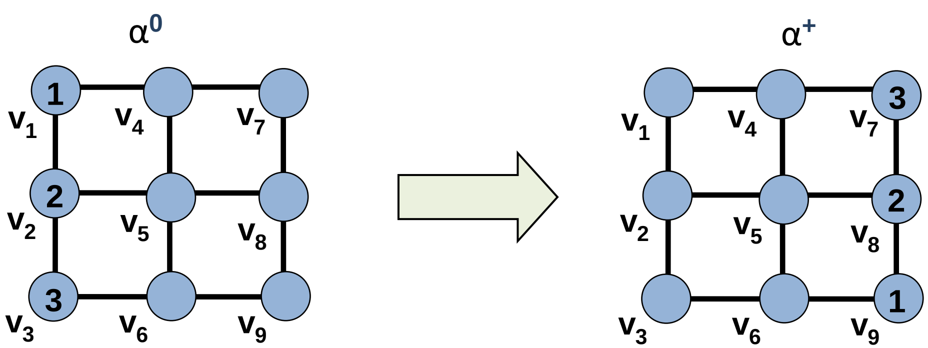
- A **quadruple** $(G, A, \alpha^0, \alpha^+)$, where
  - $G=(V,E)$ is an **undirected graph**
  - $A = \{a_1,a_2,\ldots,a_\mu\}$, where $\mu<|V|$ is a **set of agents**
  - $\alpha^0: A \rightarrow V$ is an **initial arrangement of agents**
    - uniquely invertible function
  - $\alpha^+: A \rightarrow V$ is a **goal arrangement of agents**
    - uniquely invertible function
- **Time** is discrete – time steps
- **Moves/dynamicity**
  - depends on the model
  - **agent moves** into unoccupied neighbor
    - no other agent 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, A, α^0, α^+)
  - sequence of arrangements of agents
  - *(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 agents 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 A={1,2,3}

	[$\mathbf{v_1}$, v_4, v_7, v_8, v_9, v_9, $\mathbf{v_9}$]
makespan=7	[$\mathbf{v_2}$, 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**
  (agent = container)

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

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

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

CPF as SAT

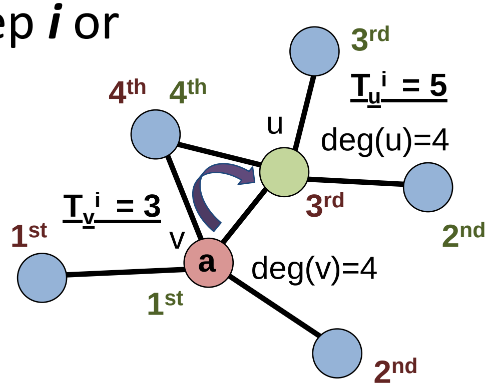
- **SAT = propositional satisfiability**
  - a formula ϕ over 0/1 (false/true) variables
  - Is there a valuation under which ϕ evaluates to 1/true?
    - **NP-complete** problem

$(x \vee \neg y) \wedge (\neg x \vee y)$
Satisfied 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 **accessed almost for free**
- **Translation**
  - given a CPF $\Sigma=(G, A, \alpha^0, A^+)$ and a **makespan** $\boldsymbol{\eta}$
  - **construct** a formula ϕ
    - satisfiable iff Σ has a solution of makespan η

INVERSE Encoding of CPF

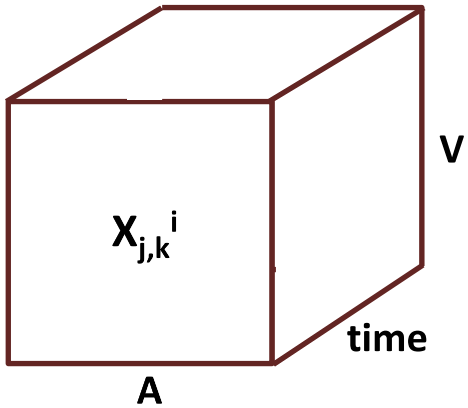
- **How to encode** a question if there is a solution of makespan ***η***
  - Encode arrangements of agents at steps 1,2…,η
  - **Step 1** … α^0
  - **Step η** … α^+
- **Integer variables** modeling step ***i***
  - $A_v^i \in \{0,1,2,\ldots, \mu\}$
    - $A_v^i = j$ if agent a_j is located in vertex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



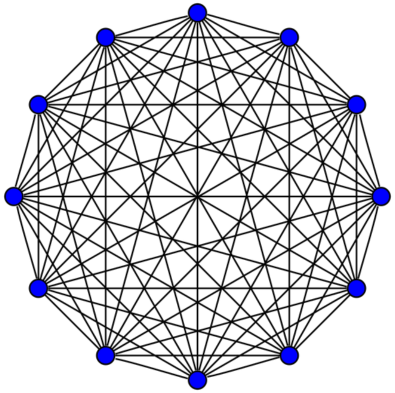
- Don't forget constraints – valid transitions between time-steps

DIRECT Encoding of CPF

- Use **propositional variables directly** instead of integer ones
  - **A =** $\{a_1, a_2, \ldots, a_\mu\}$
    - a set of agents
  - **V**=$\{v_1, v_2, \ldots, v_n\}$
    - a set of vertices
  - **time steps** 1,2…,η



- $X_{j,k}^{i} \in$ {true, false}
  - TRUE iff agent a_k appears in v_j at time step i
    - allow to represent invalid states



- **Constraints**
  - rule out invalid states
  - enforce valid transitions between time steps
    - many binary clauses
      - **at most one** agent is placed in a vertex at each time step
      - support **unit propagation**

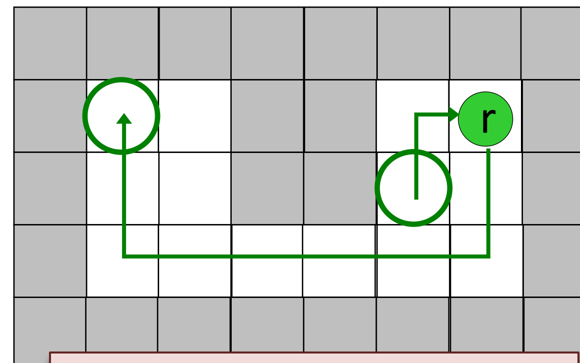
Size of Encodings

- **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

Grid 8×8 **|Agents|**			**INVERSE**		ALL-DIFFERENT		**DIRECT**		SIMPLIFIED	
1	#Variables #Clauses	Ratio Length	8 358.7 31 327.9	3.748 2.616	1 489.3 7 930.4	5.325 3.057	814.4 23 241.9	28.539 2.149	1 628.8 3 384.6	2.078 2.550
4			10 019.5 55 437.0	5.532 2.641	7 834.5 34 781.9	4.440 3.103	3 257.6 115 934.3	35.589 2.272	4 072.0 17 997.8	4.420 2.374
16			11 680.3 91 344.5	7.820 3.127	67 088.3 216 745.4	3.231 3.147	13 030.4 840 540.6	64.506 2.505	13 844.8 150 259.2	10.853 2.180
32			12 510.7 122 170.3	9.765 3.733	230 753.0 646 616.2	2.802 3.168	26 060.8 2 738 584.7	105.084 2.621	26 875.2 510 672.1	19.002 2.111

Knowledge Compilation

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

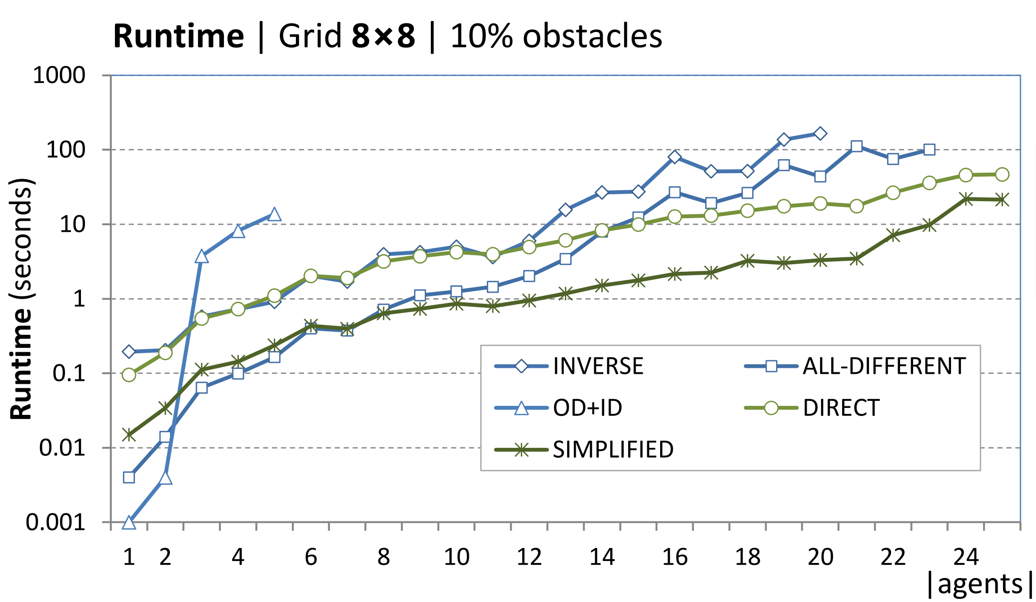


The location of agent **r** is allowed in steps **< η-9** and **> 2**

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

Runtime Evaluation

- **Experimental setup**
  - 4-connected **grids** of size 6×6, 8×8, 12×12
  - **random** initial and goal arrangement
  - 10% of cells - obstacles
    - comparison with an A*-based ID+OD



Grid 8×8 / |A|	1	2	4	8	12	16	20	24
Makespan	6.4	6.1	8.1	10.5	9.8	11.0	11.9	12.7

Conclusions and Observations

- **CPF as SAT**
  - Advantages
    - **search techniques**
      - advanced search techniques from SAT solvers accessed
    - **modularity**
      - exchangeable modules – SAT solver, encoding
    - **knowledge compilation**
  - Disadvantages
    - **energy extensive solutions**
      - agents move too much
    - **size of encoded instances**
      - large graphs
      - many time steps

- Encoded **integer variables (INVERSE)** vs. **propositional variables (DIRECT)**
  - **INVERSE**
    - smaller size of encoding
  - **DIRECT**
    - more shorter clauses - supports unit propagation
    - over constrained