Compact Representations of Cooperative Path-Finding as SAT Based on Matchings in Bipartite Graphs

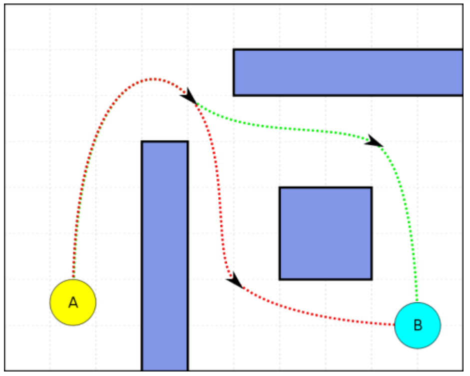
Pavel Surynek
presented by **Filip Dvořák**

Faculty of Mathematics and Physics
Charles University in Prague
Czech Republic

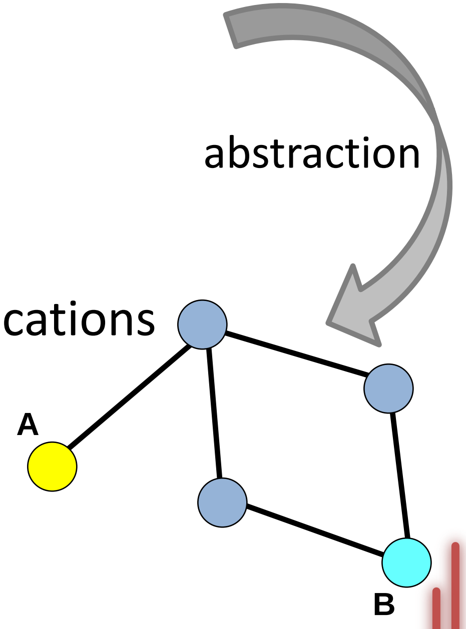
ICTAI 2014, Limassol, Cyprus

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

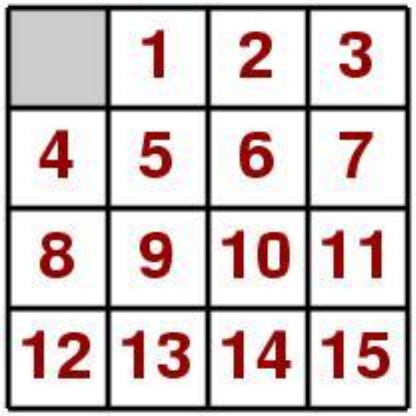


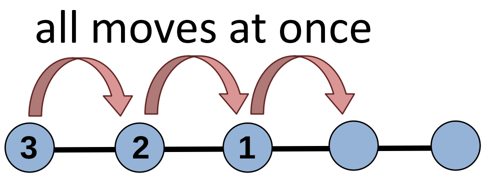
abstraction



CPF Formally

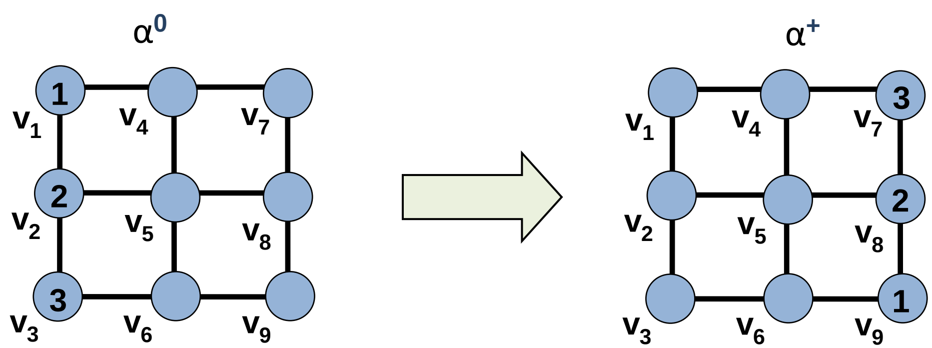
- A **quadruple** (G, A, α^0, α^+), where
  - G=(V,E) is an **undirected graph**
  - A = $\{a_1, a_2, \ldots, a_\mu\}$, where $\mu<|V|$ is a **set of agents**
  - α^0: A $\rightarrow$V is an **initial arrangement of agents**
    - uniquely invertible function
  - α^+: A $\rightarrow$V is a **goal arrangement 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 $(G, A, \alpha^0, \alpha^+)$
  - 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}

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**
  (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 $x = 1$, $y = 1$

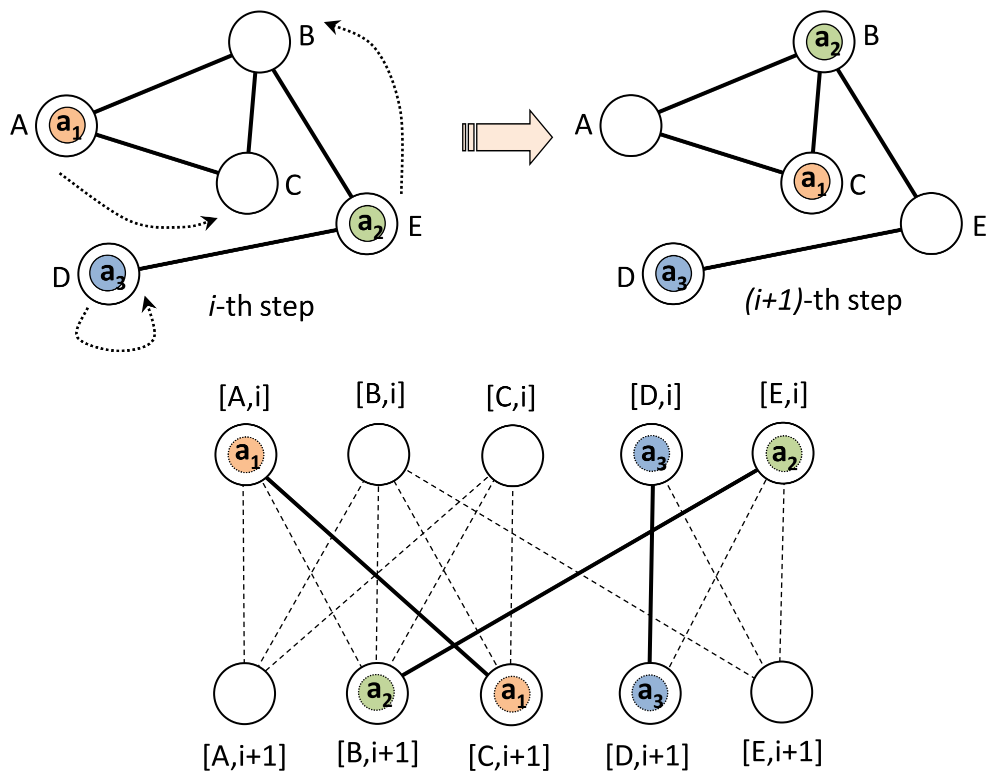
- **SAT solving** and **CPF**
  - powerful SAT solvers
    - MiniSAT, clasp, glucose, glue-MiniSAT, crypto-MiniSAT, …
    - intelligent search, learning, restarts, heuristics, …
  - **CPF $\Rightarrow$ SAT**
    - all the advanced techniques **accessed almost for free**
- **Translation**
  - given a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

MATCHING Encoding of CPF (1)

- **How to encode** a question if there is a solution of makespan ***η*?**
  - Build **time expansion network**
    - Represent arrangements of agents at steps 1,2…,η
      - **step 1** … α^0
      - **step η** … α^+
    - Encode **dynamicity** of CPF
      - consecutive arrangements must be obtainable by valid moves
- Decompose encoding into **two parts** ⇒ **MATCHING Encoding**
  - **(i)** vertex occupancy by anonymous agents
    - occupied vertices in consecutive arrangements form a **matching**
  - **(ii)** mapping of agents to vertices
    - the same agent must be located at both ends of an edge traversed by anonymous agents

MATCHING Encoding of CPF (2)

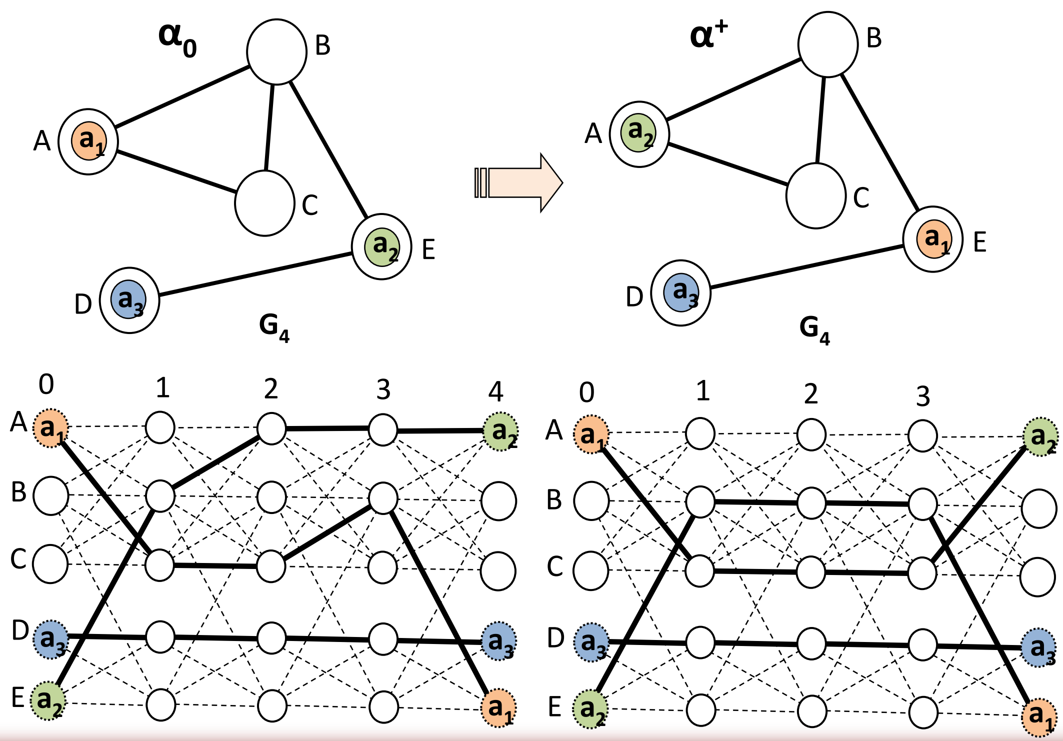
- A **matching** induced by movement of agents between *i*-th and (*i*+1)-th time step



MATCHING Encoding of CPF (3)

- A **series of matchings** corresponding to a solution of CPF of a given makespan
  - existence of a series of matchings is a **necessary condition** for existence of a solution

bCPF $\mathbf{\Sigma}$=(G, {a_1,a_2,a_3}, α_0, α^+), makespan η=4



MATCHING Encoding of CPF (4)

- Agents are **anonymous** within the matching model
  - like a piece of **commodity** (water)
    - an agent at the beginning of a path (initial agent) may **not correspond** to the agent at the end (goal agent)
- **Map** distinguishable agents to anonymous ones (to water)
  - if an edge is selected to the matching then the same agent must be located at both ends
    - distinguishable agents follow paths found by commodity (water)

MATCHING Encoding of CPF (5)

- **Propositional representation**
  - (i) vertex occupancy by **anonymous agents**
    - a **single propositional variable** for occupied vertex/edge at a time step
    - used for the most of constraints regarding **validity of a move**
      - simple representation
  - (ii) vertex occupancy by **distinguishable agents**
    - agent located in a vertex at a time is expressed by a **bit vector**
    - anonymous occupancy at both ends of a selected edge imply equality between agents located its vertices
      - equality between bit vectors

Encoding Size Evaluation

- Comparison with **previous encodings**
  - **INVERSE** [Surynek, PRICAI 2012]
    - based on bit-vectors
    - comparison with domain independent SATPlan [Kautz, Selman, 1999] and SASE encoding [Huang, Chen, Zhang, 2010]
  - **ALL-DIFFERENT** [Surynek, ICTAI 2012]
    - based on bit-vectors and all-different constraint

Setup: 4-connected grid, random initial arrangement and goal, 20% obstacles

16 time steps

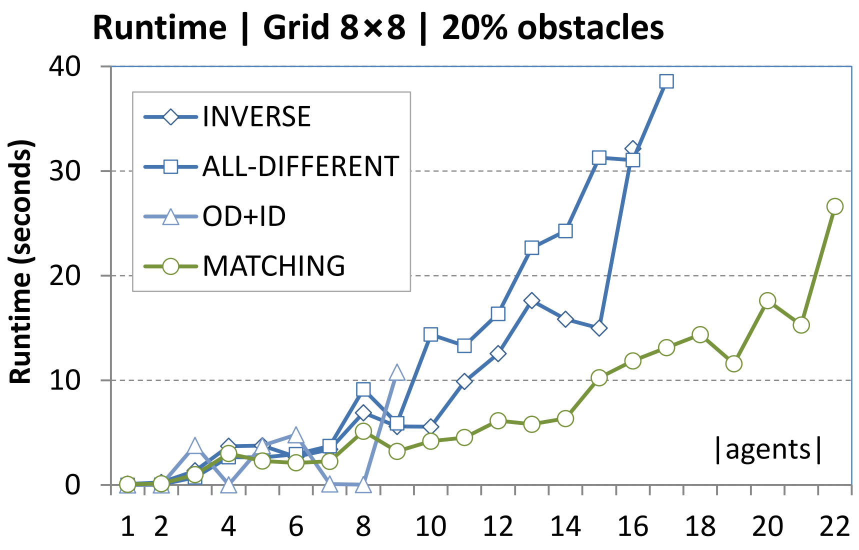
Grid 8×8 / **|Agents|**		INVERSE	ALL-DIFFERENT	**MATCHING**
1	#Variables #Clauses	8 358.7 31 327.9	**1 489.3** **7 930.4**	4 520.3 25 881.1
4		10 019.5 55 437.0	7 834.5 **34 781.9**	**6 181.1** 43 171.0
16		11 680.3 91 344.5	67 088.3 216 745.4	**7 841.9** **72 259.3**
32		12 510.7 122 170.3	230 753.0 646 616.2	**8 672.3** **99 675.5**

32 time steps

Grid 16×16 / **|Agents|**		INVERSE	ALL-DIFFERENT	**MATCHING**
1	#Variables #Clauses	71 974.0 286 764.5	**11 413.6** **82 011.1**	38 328.2 230 572.1
4		85 094.0 496 353.1	**50 978.3** **336 001.7**	51 448.2 377 551.9
16		98 214.0 803 130.0	296 355.6 1 521 163.0	**64 568.2** **621 720.0**
32		104 774.0 1 065 304.0	847 829.1 3 545 489.0	**71 128.2** **852 589.4**

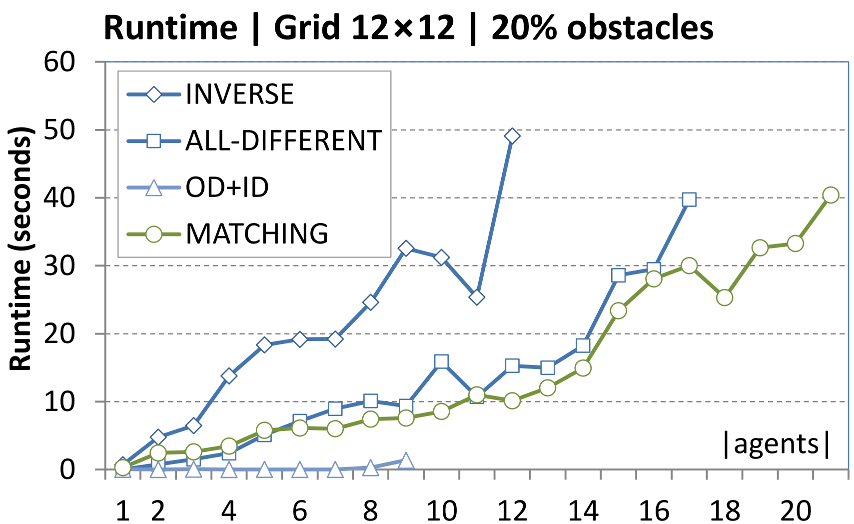
Runtime Evaluation

- Comparison with **previous encodings** + A*-based **ID+OD** [Standley, IJCAI 2011]
  - same setup as in the size evaluation



Grid 8×8 |Agents|	1	2	4	6	8	10	12	16	20
Makespan	5.3	7.4	8.4	8.7	11.0	9.8	11.6	12.4	12.3

Grid 12×12 |Agents|	1	2	4	6	8	10	12	16	20
Makespan	11.3	13.0	14.0	15.5	17.5	18.0	33.8	18.7	21.0



Conclusions and Observations

- **CPF as SAT**
  - **Advantages**
    - search techniques
      - advanced search techniques from SAT solvers accessed
    - modularity
      - exchangeable modules – SAT solver, encoding
  - **Disadvantages**
    - energy extensive solutions
      - agents move too much
- **MATCHING** Encoding
  - **space efficient**
    - small number of variables and clauses
  - **time efficient**
    - can be solved faster than previous encodings
    - SAT-based approach with MATCHING encoding outperforms A*-based approach