Optimal Cooperative Path-Finding with Generalized Goals in Difficult Cases

Pavel Surynek
Faculty of Mathematics and Physics
Charles University in Prague
Czech Republic

SARA 2013, Leavenworth, WA, USA
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
CPF Formally

- **A quadruple** \((G, R, \alpha^0, \alpha^+)\), where
  - \(G=(V,E)\) is an **undirected graph**
  - \(R = \{r_1, r_2, \ldots, r_\mu\}\), where \(\mu < |V|\) is a **set of robots**
  - \(\alpha^0: R \to V\) is an **initial arrangement of robots**
    - uniquely invertible function
  - \(\alpha^+: R \to V\) 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 \((G, R, \alpha^0, \alpha^+)\)
  - sequence of arrangements of robots
  - \((i+1)\)-th arrangement obtained from \(i\)-th by legal moves
  - **the first arrangement** determined by \(\alpha^0\)
  - **the last arrangement** determined by \(\alpha^+\)
    - 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
- Time step: 1 2 3 4 5 6 7

- \([v_1, v_4, v_7, v_8, v_9, v_9]\)
- \([v_2, v_2, v_1, v_4, v_7, v_8, v_8]\)
- \([v_3, v_3, v_2, v_1, v_4, v_7]\)
Motivation for CPF

- Container rearrangement
  (robot = container)

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

- Data transfer
  (robot = data packet)

- Ship avoidance
  (robot = ship)
Generalization CPF

- Interchangeable robots
  - robots **indifferent** w.r.t. goals

- Motivation
  - formation maintenance

- \( A^+: \mathbb{R} \rightarrow \mathcal{P}(V) - \{\emptyset\} \) instead of \( \alpha^+: \mathbb{R} \rightarrow V \)
  - each robot can have multiple vertices as its goal

- **relaxed** goal
  - problem expected to get easier
CPF as SAT

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

- **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 employed for free

- **Translation**
  - given a CPF $\Sigma=(G, R, \alpha^0, A^+)$ and a makespan $k$
  - construct a formula $\phi$
    - satisfiable iff $\Sigma$ has a solution of makespan $k$
How to encode a question if there is a solution of makespan \( k \)
- Encode arrangements of robots at steps 1,2,...,\( k \)
- **Step 1** ... \( \alpha^0 \)
- **Step \( k \)** ... \( \alpha^+ / A^+ \)

**Integer variables** modeling step \( i \)
- \( A_v^i \in \{0,1,2,..., \mu\} \)
  - \( A_v^i = j \) 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 \( i \)
- \( T_v^i \in \{0,1,2,..., 2\deg(v)\} \)
  - \( 0 < T_v^i \leq \deg(v) \) 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 $A_v^i \in \{0,1,2,..., \mu\}$
    - replaced with $\lceil \log_2(\mu+1) \rceil$ propositional variables
    - extra states are forbidden

  - **⇒ Compact representation**
    - smaller than in SAT-based domain-independent planners
    - knowledge compilation – distance heuristic

| $|A|$ | Makespan | SATPLAN encoding | SASE encoding | INVERSE encoding |
|-----|----------|------------------|---------------|-----------------|
| 4   | 8        | 5.864            | 11.386        | 5.400           |
| 8   | 8        | 10.022           | 19.097        | 5.920           |
| 12  | 8        | 14.471           | 26.857        | 5.920           |
| 16  | 10       | 30.157           | 51.662        | 8.122           |
| 24  | 10       | 43.451           | 73.101        | 8.122           |
| 32  | 14       | 99.398           | 157.083       | 12.396          |

- SARA 2013
- Heuristics **directly built-in into the encoding**
  - **distance** heuristic
    - locations unreachable in a given time are forbidden
    - search space reduced
  - **mutex** heuristic
    - 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 grid of size $6 \times 6$
  - random initial and goal arrangement
  - various sizes of goal sets

![Graphs showing runtime and optimal makespan for different goal sizes and robot counts.](image-url)

N.B. The graphs illustrate the scalability of the system across varying numbers of robots and goal sizes.
Conclusion and Future Research

- CPF with generalized goals
  - set of vertices as a goal
  - makespan optimal solutions via SAT solving
- More complex actions
  - not only moving
- Adversarial version (AAAI 2013)
  - two or more teams competing
  - complexity
  - strategies to gain territory
- Formation preservation
  - motivated by computer games