# UniAGENT: Reduced Time-Expansion Graphs and Goal Decomposition in Sub-optimal Cooperative Path Finding

**Pavel Surynek**
Charles University in Prague, Faculty of Mathematics and Physics, Czech Republic
pavel.surynek@mff.cuni.cz

**Abstract**

Solving cooperative path finding (CPF) by translating it to propositional satisfiability represents a viable option in highly constrained situations. The task in CPF is to relocate agents from their initial positions to given goals in a collision free manner. In this paper, we propose a reduced time expansion that is focused on makespan sub-optimal solving of the problem. The suggested reduced time expansion is especially beneficial in conjunction with a goal decomposition where agents are relocated one by one.

## Cooperative Path Finding (CPF)

The problem of cooperative path-finding (CPF) (Silver, 2005) is a graph theoretical abstraction for many real life problems where the task is to relocate cooperatively a group of agents or other movable objects in a collision free manner. Each agent of the group is given its initial and goal position. The problem consists in constructing a spatial temporal plan for each agent by which it can relocate from its initial position to the given goal.

We further develop solving of CPF by translating it to propositional satisfiability (SAT) (Surynek, 2014). Recent propositional encodings of CPF are based on time expansion of the graph modeling the environment so that the encoding is able to represent arrangements of agents over the graph (in the environment) at all the time steps up to the final step. Since there may be many time-steps before all the agents reach their goals, these encodings may become extremely large and hence unsolvable in reasonable time. We are trying to overcome this limitation by reducing the expansion of the graph in this work.

Let $G = (V, E)$ be an **undirected graph** that models the environment where agents are moving and let $A$ be a finite set of *agents*; then, an arrangement of agents in vertices of graph $G$ is fully described by $\alpha: A \rightarrow V$ with $|\{a \in A | \alpha(a) = v\}| \leq 1$ for each $v \in V$ (at most **one agent** can be located in each vertex).

**Definition 1** (COOPERATIVE PATH FINDING). An instance of *cooperative path-finding* problem is a quadruple $\Sigma = [G = (V, E), A, \alpha_0, \alpha_+]$ where $\alpha_0$ and $\alpha_+$ define the initial and the goal arrangement of agents $A$ in $G$ respectively. □

An arrangement $\alpha_i$ at the $i$-th time step can be transformed instantaneously by a movement of agents in the non-colliding way to form a new arrangement $\alpha_{i+1}$. The transition between $\alpha_i$ and $\alpha_{i+1}$ must satisfy the following *validity conditions*:

- $\forall a \in A$ either $\alpha_i(a) = \alpha_{i+1}(a)$ or $\{\alpha_i(a), \alpha_{i+1}(a)\} \in E$ (agents move along edges or not move at all), (1)
- $\forall a \in A \ \ \alpha_i(a) \neq \alpha_{i+1}(a) \Rightarrow \big(\forall b \in A \ \alpha_i(b) \neq \alpha_{i+1}(a)\big)$ (agents move to vacant vertices only), and (2)
- $\forall a, b \in A \ \ a \neq b \Rightarrow \alpha_{i+1}(a) \neq \alpha_{i+1}(b)$ (no two agents enter the same target/unique invertibility of resulting arrangement). (3)

The task in CPF is to transform $\alpha_0$ using above valid transitions to $\alpha_+$.

**Definition 2** (SOLUTION, MAKESPAN). A *solution* of a *makespan* $m$ to a CPF instance $\Sigma = [G, A, \alpha_0, \alpha_+]$ is a sequence of arrangements $\vec{s} = [\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_m]$ where $\alpha_m = \alpha_+$ and $\alpha_{i+1}$ is a result of valid transition from $\alpha_i$ for every $i = 1,2, \ldots, m-1$ . □

## Reduced Time Expansion Graphs

One approach to makespan optimal CPF solving via SAT is to query the SAT solver (Audemard & Simon, 2013) whether a formula modeling the question if there is a solution of makespan $m$ is satisfiable for growing $m$. This is the process originally suggested for planning by Kautz and Selman (1999). The drawback of makespan optimal CPF solving via SAT is the large size of the formulae that encode queries (Surynek, 2014). The size of encoding formulae becomes especially prohibitive when they encode que-

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The full version of this paper entitled *Reduced Time-Expansion Graphs and Goal Decomposition for Solving Cooperative Path Finding Sub-optimally* was published at IJCAI 2015.

ries if a solution with a large makespan exists. This is due to the fact that existing encodings expands the graph modeling the environment over the time up to the given makespan bound while arrangements of agents are represented at all the time steps in the time expansion.

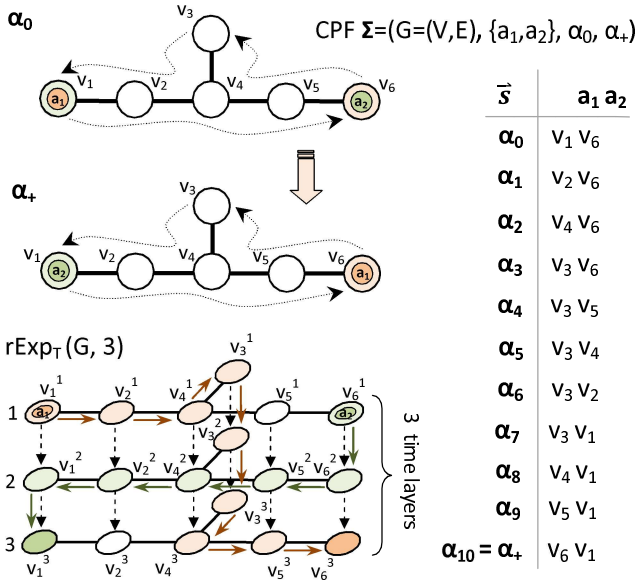



**Figure 1.** *An example of CPF and its solving through* ***reduced time expansion graph***. A reduced time expansion graph consisting of 3 time layers is build. A solution is obtained by collecting vertex disjoint paths connecting the initial positions agents in the first layer with their goal positions in the last time layer.

Hence, our idea was to **reduce** the time expansion by **relaxing** from the requirement of makespan optimality. The key observation is that if there is no need of any complex avoidance between agents (there is no need to visit a single vertex multiple times), no time expansion of the graph is necessary at all. The query if there is a solution (not necessarily makespan optimal) can be stated as a question of existence of vertex disjoint paths connecting initial positions of agents with their goals in the original graph.

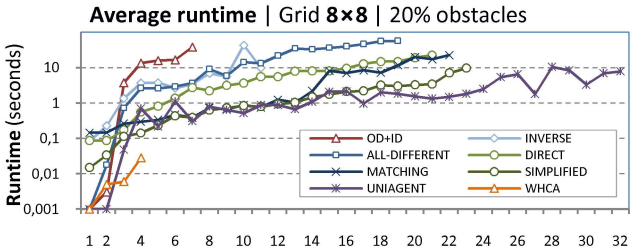
Nevertheless, interactions among agents require complex avoidance in real situations - a single vertex may need to be visited multiple times. This led us to the suggestion of a concept of *reduced time expansion graph*, which combines reduction of the expansion with ability to represent complex avoidance.

**Definition 3** (REDUCED TIME EXPANSION GRAPH - $\mathrm{rExp}_T(G,\vartheta)$). Let $G=(V,E)$ be an undirected graph and $\vartheta \in \mathbb{N}$. A *reduced time expansion graph* with $\vartheta$ *time layers* associated with $G$ is a directed graph $\mathrm{rExp}_T(G,\vartheta)=(V\times\{1,2,\dots,\vartheta\},E')$ where $E'=\{([u,l],\ [v,l])|\{u,v\}\in E; l=1,2,\dots,\vartheta\}\cup\{([v,l],[v,l+1])\mid l=1,2,\dots,\vartheta-1\}$. □

Solving of CPF $\Sigma=[G,A,\alpha_0,\alpha_+]$ can be viewed as a search for *vertex disjoint paths* in $\mathrm{rExp}_T(G,\vartheta)$ that connect initial positions and goals in the first and the last time-layer respectively provided that the number of time-layers $\vartheta$ is sufficiently high (Figure 1).

## UniAGENT Solving Process

It turned out that vertex disjoint paths exist in reduced time expansion graphs with few time layers when the initial arrangement differs little from the goal one. This observation led us to design a method called UniAGENT solving in which agents are placed to their goals one by one. Placing a single agent is solved by extracting vertex disjoint paths from a (small) reduced time expansion graph via SAT. The comparison of the UniAGENT method with other optimal and sub-optimal methods (Figure 2) indicates better scalability of the new method for growing number of agents in highly constrained situations (dense occupancy).




| \|A\| | 1 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 |
|---|---|---|---|---|---|---|---|---|---|
| $\eta$ | 5.3 | 8.4 | 11.0 | 11.7 | 12.4 | 12.3 | - | - | - |
| $\omega$ | 5.6 | 9.3 | - | - | - | - | - | - | - |
| $\vartheta$ | 9.3 | 15.8 | 33.0 | 49.3 | 83.4 | 96.1 | 131.4 | 154.1 | 201.7 |

**Figure 2.** *Runtime and makespan comparison over 8×8 4-connected grid.* UNIAGENT and WHCA* (Silver, 2005) produce makespan sub-optimal solutions; OD+ID (Standley & Korf, 2011) and other SAT based methods (Surynek, 2014) are makespan optimal. Evaluation of runtime and makespan has been done for the growing number of agents (the timeout was 256 seconds). Average optimal makespan is shown as $\eta$; $\vartheta$ and $\omega$ are average makespans of UNIAGENT method and WHCA* respectively.

## References


Audemard, G., Simon, L., 2013. *The Glucose SAT Solver.* http://labri.fr/perso/lsimon/glucose/, [accessed in April, 2015].

Kautz, H., Selman, B., 1999. *Unifying SAT-based and Graph-based Planning.* Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 1999), pp. 318-325, Morgan Kaufmann.

Silver, D., 2005. *Cooperative Pathfinding.* Proceedings of the 1st Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2005), pp. 117-122, AAAI Press.

Standley, T. S., Korf, R. E., 2011. *Complete Algorithms for Cooperative Pathfinding Problems.* Proceedings of Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 668-673, IJCAI/AAAI Press.

Surynek, P., 2014.*Compact Representations of Cooperative Path-Finding as SAT Based on Matchings in Bipartite Graphs.* Proceedings of the 26th International Conference on Tools with Artificial Intelligence (ICTAI 2014), pp. 875-882, IEEE Press.