Adversarial Cooperative Path-Finding: A First View

Marika Ivanová and **Pavel Surynek**

Charles University in Prague, Faculty of Mathematics and Physics
Malostranské náměstí 25, 118 00, Praha, Czech Republic
marikaivanova@seznam.cz, pavel.surynek@mff.cuni.cz

Abstract

An adversarial version of the problem of cooperative path-finding (CPF) in graphs is addressed. Two or more teams of agents compete in finding paths in the graph to given destination vertices. The task is to control agents of a selected team so that its agents reach their destinations before agents of adversaries. Tactics such a blocking agents of the adversary or protecting own agents are allowed by the suggested formal definition of the problem.

Introduction and Motivation

Adversarial Cooperative Path-Finding (ACPF) can be regarded as a generalization of CPF (Silver, 2005; Ryan, 2008) where team of agents compete in reaching their destinations. Apart from probably the most obvious utilization of the concept in computer games (real time strategies), we can mention planning, simulations, and verification of police interventions, military actions, or security operations. The problem is a game of a certain number of players on a graph at certain level of abstraction.

Formal Definition of ACPF

The following definition introduces the ACPF problem in a graph formally.

Definition 1 *(ACPF problem)*. Let us consider a 6-tuple $\Sigma = (G, A, \mathcal{T}, \lambda_0, \lambda_+, \hat{\alpha})$ where $G = (V, E)$ is an undirected graph, $A = \{a_1, a_2, \dots, a_k\}$ is a set of agents, and $\mathcal{T} = \{T_1, T_2, \dots, T_t\}$ denotes a finite set of teams with $t \leq k$. Teams are disjunctive sets of agents and every agent belongs to exactly one team. T_1 is our team; other teams are our adversaries. An injective mapping $\lambda_0: A \rightarrow V$ assigns an initial vertex to each agent (a starting position). Goal positions are more general; $\lambda_+: A \rightarrow \mathcal{P}(V)$ assigns a set of target vertices to each agent. Finally, $\hat{\alpha}$ denotes a mapping, that determines the next placement of all the agents belonging to adversary teams. It takes a sequence of all the previous placements of agents as its input. □

Agents are placed in vertices of G so that at most one agent is placed in each vertex. The placement of agents at time step i is thus an injective mapping $\lambda_i: A \rightarrow V$.

Agent movements must observe several rules that correspond to real physical limitations. That is, λ_i can be transformed to λ_{i+1} only in a defined way as follows:

- Agents move along edges or stay at a vertex.
- An agent a can move to an unoccupied vertex v or if v was occupied at time step i by agent a' different from a, agent a' must move away. The second condition allows agents to move around a cycle without any free vertex. A trivial case of position swapping along an edge is an exception and is forbidden.
- Teams alternate in their moves. That is, only agents of team $T_{(i+1) \bmod t}$ moves between times steps i and $i + 1$.

Our goal is to find a sequence of moves for agents of team T_1 that reacts on moves of agents of adversaries and leads agents of T_1 to their targets.

Definition 2 *(solution)*. A sequence $[\lambda_0, \lambda_1, \dots, \lambda_{m-1}]$ of mappings, where $\lambda_{m-1}(a) \in \lambda_+(a)$ holds for every agent belonging to team T_1 is a *solution* of given ACPF problem Σ. The sequence satisfies all the movement constraints and there is no other team whose all agents has already reached their goal position. □

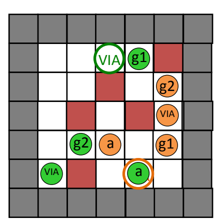


Figure 1. *ACPF on 4-connected grid.* The green team protects its VIA agent by two guard agents g_1 and g_2 while simultaneously attacks target location of the orange VIA agent.

The problem definition does not guarantee the existence of a solution since the graph is not required to be connected or the agents can block each other so they can never reach their target locations. Hence in practice we should focus on weaker objectives such as relocating a certain number of agents to their targets.

A First View of Theoretical Properties

ACPF seems to be a computationally difficult. We sketch a proof that ACPF is PSPACE-hard in this section. More precisely, we will consider a question if there exists a solution (winning strategy) for a selected team in the given ACPF regardless of actions taken by adversaries. So far we have found that the reduction from QBF (Coste-Marquis et al., 2006) to ACPF can be used to show the result. Techniques suggested in (Surynek, 2010) will be used.

Proposition 1. *A question if there exist a solution for a selected team in ACPF is PSPACE-hard.* ■

Sketch of proof: Consider an instance of QBF $\mathcal{F}$ where each variable has the same number of positive and negative literals. This problem is known to be PSPACE-complete. We will construct an instance of ACPF where a solution exists if and only if the given QBF $\mathcal{F}$ is valid.

Let us start with a graph that contains a vertex for each literal in $\mathcal{F}$. Other vertices and edges will be added to the graph by construction. There will be two teams of agents T_1 and T_2. T_1 is the player; that is, existential quantification in $\mathcal{F}$ is relevant for him; while T_2 is an adversarial team; that is, universal quantification in $\mathcal{F}$ is relevant for her. Agents of T_1 correspond to variables (there is one agent per positive literal) and to clauses (one agent per clause). Agents of T_1 corresponding to variables must go through literal vertices at certain time step i while their objective is to leave at least one vertex in each clause free at i. This free vertex is reserved for clause agents of T_1 that each need to go through it at i. Enforcing that agents must go through some vertices at given time step is done by techniques such as vertex locking shown in (Surynek, 2010). For simplicity we omit details here.

Team T_2 consists only of agents corresponding to variables which are trying to thwart intentions of team T_1. Every time team T_1 decides to enter either positive or negative literal vertices corresponding to an existentially quantified propositional variable of its turn, T_2 is trying make a decision with its agents corresponding to the next universally quantified variable that eventually leads to occupation of all the vertices of a certain clause at time step i. If team T_2 succeed then at least one clause agent of T_1 will be delayed at completely occupied clause at time step i and T_1 loses. ■

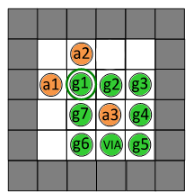
The important technical step is enforcing that agents corresponding to certain variable are never divided between positive and negative literal vertices – they all go to either positive or negative side.

Whether ACPF belongs to PSPACE, that is, if it is PSPACE-complete is currently an open question. The problematic point is that we do not have any polynomial upper bound on the length of the sequence of moves in the winning strategy.

Practical Offensive and Defensive Tactics

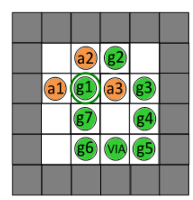
Various tactics can be used when ACPF is solved in practice. Our first view suggestions are motivated by a security operation with a protected person (VIP) and guards who are trying to secure relocation of VIP to its destination. We suggest 3 different roles for agents, let us call them *VIA* (very important agent), *guard* and *attacker*. Roles merely indicate that agents are treated differently by the planning algorithm. An agent without any target can be used as guard or attacker, while an agent with only one target should be treated as VIA and protected by other agents.

Although the agent placement on a graph can be very diverse, we try to identify whether it is possible to enforce reachability of a target location for a given VIA. Suppose a VIA a located at vertex s and some path $(s = v_0, v_1, \ldots, v_n = \lambda_+(a))$, where every vertex is filled by a guard agent from the team of a. Let us call such formation a *bridge*. A simple observation shows that if there exist two bridges between s and $\lambda_+(a)$, then the vertex $\lambda_+(a)$ can be taken by agent a in number of time steps equal to the length of the shorter bridge by rotation of the circle formed by the two bridges.



The motivation of bridges is that it is more difficult to block such formation than a single agent. The requirement for two bridges is quite strong, but the presence of a single bridge itself does not ensure achievement of the target vertex as additional conditions must hold: vertices of the bridge must be adjacent to sufficient number of unoccupied vertices because a has to be shifted towards its target. The shifting can be carried out either by moving whole bridge or clearing the way by a guard agent.

The aim of the attackers is to prevent the adversary VIAs from achieving their goals. They can surround adversary agents or occupy their target vertices.



Conclusion and Future Work

We have introduced a novel concept of ACPF, a problem derived from traditional CPF, in this short paper. Its PSPACE-hardness was shown and several solving techniques were proposed.

The ACPF problem offers new area for research. Finding a solving algorithm, development of heuristics, and future study of the problem complexity is expected.

There are many research opportunities concerning special cases of ACPF, e.g. algorithms designed for particular types of graphs, bounded numbers of agents or teams, symmetric vs. asymmetric situations and a lot more.

Acknowledgement

This work is supported by the Charles University in Prague within the PRVOUK project (section number P46) and by the Czech Science Foundation under the contract number GAP103/10/1287.

References

Coste-Marquis, S., Le Berre, D., Letombe, F., Marquis, P., 2006: Complexity Results for Quantified Boolean Formulae Based on Complete Propositional Languages. *Journal on Satisfiability, Boolean Modeling and Computation* (JSAT), Volume 1(1): pp. 61-88, IOS Press.

Ryan, M. R. K., 2008. Exploiting Subgraph Structure in Multi-Robot Path Planning. *Journal of Artificial Intelligence Research* (JAIR), Volume 31, pp. 497-542, AAA Press.

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

Surynek, P., 2010. An Optimization Variant of Multi-Robot Path Planning is Intractable. *Proceedings of the 24th AAAI Conference on Artificial Intelligence* (AAAI 2010), pp. 1261-1263, AAAI Press.