The Thirteenth International Symposium
on Combinatorial Search (SoCS 2020)

# Moving Agents in Formation in Congested Environments*

**Jiaoyang Li,[1] Kexuan Sun,[1] Hang Ma,[2] Ariel Felner,[3] T. K. Satish Kumar,[1] Sven Koenig[1]**
[1]University of Southern California, [2]Simon Fraser University, [3]Ben-Gurion University
jiaoyanl@usc.edu, kexuansun@usc.edu, hangma@sfu.ca, felner@bgu.ac.il, tkskwork@gmail.com, skoenig@usc.edu

## Abstract

We study the Moving Agents in Formation problem, that combines the tasks of finding short collision-free paths for multiple agents and keeping them in close adherence to a desired formation. We develop a two-phase complete algorithm, called SWARM-MAPF, whose first phase is inspired by swarm-based algorithms (in open regions) and whose second phase is inspired by multi-agent path finding algorithms (in congested regions). Empirically, we show that SWARM-MAPF scales well and finds close-to-optimal solutions.

## Introduction

Moving a team of agents in formation without collisions in known congested environments is an important problem that arises in many applications of multi-agent systems. For example, unmanned vehicles have to move in specific formations in order to transport large objects or maintain a communication network. Game characters or army personnel have to move in specific formations in order to protect vulnerable agents. These applications involve two key tasks: (a) planning collision-free paths for multiple agents, and (b) keeping the agents in formation. Task (a) can be addressed with multi-agent path finding (MAPF) algorithms, which typically minimize one of several possible metrics on the path costs. Task (b) can be addressed with formation-control algorithms, which try to restore the desired formation in case it is compromised because of obstacles.

We study the *Moving Agents in Formation (MAiF)* problem in congested environments to bridge the gap between algorithms that focus on tasks (a) or (b) exclusively. MAiF is a problem related to MAPF where a desired formation is given and the task is to plan collision-free paths for all agents that balance between the minimization of the makespan and a close adherence to the desired formation at all times. We propose a complete and efficient MAiF algorithm, called SWARM-MAPF, which leverages ideas from both formation control and MAPF. It first chooses an agent as the *leader* and plans a path for it such that the number of timesteps when the desired formation has to be compromised is minimized. Then, the path of the leader is divided into segments of two types. For segments where all agents can move in the desired formation, SWARM-MAPF uses a swarm-based algorithm from formation control. For segments where the desired formation has to be compromised, SWARM-MAPF uses a MAPF-based algorithm to move the agents around the obstacles as quickly as possible while still trying to keep a close adherence to the desired formation. SWARM-MAPF is not guaranteed to provide optimal solutions for either of the two objectives, but we demonstrate experimentally that it often produces solutions that keep all agents in close adherence to the desired formation with only a small loss of optimality in the makespan.

## Problem Definition

In a MAiF problem, we are given an undirected graph $G = (V, E)$ in a $d$-dimensional Cartesian system. The vertices $V$ correspond to locations, and the edges $E$ correspond to transitions between locations. A location $v_i \in V$ can be recognized by its coordinates $\mathbf{v_i} = (\mathbf{v_{i1}}, \ldots, \mathbf{v_{i}}_d) \in \mathbb{R}^d$. We are also given a set of $M$ agents $\{a_i | i = 1, \ldots, M\}$, each with a start location $s_i \in V$ and a goal location $g_i \in V$. At every discrete timestep, an agent can either move to an adjacent location or wait at its current location. A *path* $\pi_i$ for agent $a_i$ is a sequence of locations $\pi_i(t) \in V$, one for each timestep $t$, that moves agent $a_i$ from its start location $s_i$ to its goal location $g_i$. A *collision* happens when two agents are at the same location at the same timestep or traverse the same edge in opposite directions at the same timestep. A *solution* is a set of collision-free paths, one for each agent. The quality of a solution is evaluated by both its *makespan* $T = \max_{1 \le i \le M} |\pi_i|$ (i.e., the maximum length of all paths) and its *total formation deviation* $\sum_{t=0}^{T} \mathscr{F}(t)$, where the formation deviation $\mathscr{F}(t)$ at timestep $t$ is defined below.

The *formation* at timestep $t$ is an $M$-tuple $\ell(t) = \langle \boldsymbol{\pi_1}(\mathbf{t}), \ldots, \boldsymbol{\pi_M}(\mathbf{t}) \rangle$ specified by the coordinates of the locations of all agents at timestep $t$. The *desired formation*

*This paper is a short verion of (Li et al. 2020). The research at the University of Southern California was supported by the National Science Foundation (NSF) under grant numbers 1409987, 1724392, 1817189, and 1837779 as well as a gift from Amazon. The research was also supported by the United States-Israel Binational Science Foundation (BSF) under grant number 2017692.

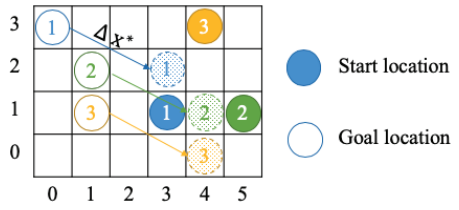


Figure 1: A MAiF instance on a 4-neighbor grid. The hatched circles show the locations of all agents after applying the optimal translation $\mathbf{\Delta x}^*$ to the goal locations.

is an $M$-tuple $\ell_g = \langle \mathbf{g_1}, \dots \mathbf{g_M} \rangle$ specified by the coordinates of the goal locations of all agents. The *formation deviation* $\mathscr{F}(t)$ at timestep $t$ characterizes the least effort needed to transform the formation $\ell(t)$ to the desired formation $\ell_g$. It is defined as the sum of the $L_1$-distances over all agents between the two locations of the same agent after applying any translation $\mathbf{\Delta x}$ to $\ell_g$, minimized over all such translations. Formally, $\mathscr{F}(t) = \min_{\mathbf{\Delta x}} \sum_{i=1}^{M} ||\boldsymbol{\pi_i}(\mathbf{t}) - (\mathbf{g_i} + \mathbf{\Delta x})||_1 = \sum_{j=1}^{d} \min_{\mathbf{\Delta x}_j} \sum_{i=1}^{M} |(\boldsymbol{\pi_i}(\mathbf{t})_j - \mathbf{g_i}_j) - \mathbf{\Delta x}_j| = \sum_{j=1}^{d} \sum_{i=1}^{M} |(\boldsymbol{\pi_i}(\mathbf{t})_j - \mathbf{g_i}_j) - \mathbf{\Delta x}_j^*|$, where, for each dimension $j$, $\mathbf{\Delta x}_j^*$ is the median of all differences $\boldsymbol{\pi_i}(\mathbf{t})_j - \mathbf{g_i}_j$. Figure 1 shows an example. The differences between the coordinates of the start and goal locations for every agent in every dimension are $\{\mathbf{s_i}_1 - \mathbf{g_i}_1\} = \{3, 4, 3\}$ and $\{\mathbf{s_i}_2 - \mathbf{g_i}_2\} = \{-2, -1, 2\}$. Therefore, the optimal translation is $\mathbf{\Delta x}^* = (3, -1)$, and the formation deviation at timestep 0 is thus $\mathscr{F}(0) = 5$.

## SWARM-MAPF

SWARM-MAPF first chooses a leader among all agents and partitions its path into open and congested segments. In each open segment, the agents form the desired formation and follow the leader, while, in each congested segment, their paths are planned by a dedicated MAPF-based algorithm.

### Choosing the Leader and Its Path

A *formation-blocking* location for an agent is one where the desired formation cannot be kept by the remaining agents when the agent is at it. The *total formation-blocking value* of a path is the number of formation-blocking locations on it, which captures the minimum number of timesteps when agents cannot form the desired formation. SWARM-MAPF chooses the agent as the leader whose path minimizes the total formation-blocking value among all paths for all agents of lengths no larger than $wB$, where $w \geq 1$ is a user-provided parameter and $B = \max_{1 \leq i \leq M} d(s_i, g_i)$ is a lower bound on the makespan ($d(s_i, g_i)$ is the distance between locations $s_i$ and $g_i$). To do so, SWARM-MAPF performs a best-first search for each agent to find a path with the minimum total formation-blocking value subject to the constraint that the path length is no larger than $wB$. It breaks ties in favor of shorter paths. Then, SWARM-MAPF chooses the path with the minimum total formation-blocking value among the paths of all agents as the leader's path and the corresponding agent as the leader. It breaks ties in favor of shorter paths.

### Partitioning the Path of the Leader into Segments

Once the leader has been chosen, its path $\pi^*$ is partitioned into *open segments* and *congested segments* alternately. Each open segment is a maximum segment $[\pi^*(t_b), \pi^*(t_e)]$ ($t_b \leq t_e$) such that, for all $t_b \leq t \leq t_e$, location $\pi^*(t)$ is not formation-blocking. Each remaining segment is a congested segment. Assume that there are $K$ open segments. Let $p_1^*, \dots, p_{2K}^*$ denote the first and last locations of all open segments, i.e., the $k$-th open segment is $[p_{2k-1}^*, p_{2k}^*]$. Let $p_0^*$ denote the start location of the leader. $p_0^*$ and $p_1^*$ are identical iff the start locations of all agents are in the desired formation. There are also $K$ congested segments, and the $k$-th congested segment is $[p_{2k-2}^*, p_{2k-1}^*]$.

Let $\ell_0^*$ denote the $M$-tuple of the start locations of all agents and $\ell_k^*$ ($1 \leq k \leq 2K$) denote the $M$-tuple of the locations of all agents that form the desired formation around location $p_k^*$ of the leader. Each open segment specifies a sub-MAiF instance where all agents need to move from locations $\ell_{2k-1}^*$ to locations $\ell_{2k}^*$. SWARM-MAPF obtains a sub-solution for each such sub-MAiF instance for free since all agents move in the desired formation along the path segment $[p_{2k-1}^*, p_{2k}^*]$ of the leader. In each congested segment $[p_{2k-2}^*, p_{2k-1}^*]$, any location except for locations $p_{2k-2}^*$ and $p_{2k-1}^*$ is formation-blocking. Each congested segment specifies a sub-MAiF instance where all agents need to move from locations $\ell_{2k-2}^*$ to locations $\ell_{2k-1}^*$. SWARM-MAPF obtains a sub-solution for each such sub-MAiF instance with CBS-M, a dedicated MAPF algorithm adapted from Conflict-Based Search (Sharon et al. 2015) that minimizes the makespan and breaks ties by preferring small total formation deviations. Finally, SWARM-MAPF concatenates the sub-solutions for all sub-MAiF instances of both types of segments to obtain a solution for the overall MAiF instance.

## Experiments

We implement SWARM-MAPF in C++ and test it on $30 \times 30$ 4-neighbor grids with 10% blocked cells. We generate 100 random instances for each number of agents and use a runtime limit of 5 minutes for each instance, When $w = 1$, SWARM-MAPF solves all instances and computes solutions in real-time (i.e., less than one second) on average for 10 and 20 agents. It also solves 91% and 65% of instances for 30 and 40 agents, respectively. As $w$ increases, as expected, the average makespan of the solutions increases and the average total formation deviation decreases. The average runtime also increases. See more results in (Li et al. 2020) and some videos of the execution of the solutions of SWARM-MAPF with $w = 1$ at `http://idm-lab.org/formation`.

## References

Li, J.; Sun, K.; Ma, H.; Felner, A.; Kumar, T. K. S.; and Koenig, S. 2020. Moving agents in formation in congested environments. In *AAMAS*.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. R. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219:40–66.