A Game Theoretic Analysis of Multi-Robot-On-the-Grid Environments

Elena Camuffo, Luca Gorghetto[†]

Abstract—**Automation has revolutionized manufacturing and established robots in the industrial assembly across a wide range of applications. Nevertheless, in most practical scenarios, robots are limited to repetitive tasks and there is still a long way to go in terms of enabling them to operate in populated environments, especially when dealing with robot motion planning. The research progress of multi-agent decision-making strategies based on reinforcement learning provides a solution for solving the problems faced by multi-robot systems in practical scenarios. In this paper, we propose two original game theoretic models applied to multi-robot environments. Simulations are led through the Nash-Q learning algorithm in order to prove the theoretical analysis and confirm their predicted trajectory dynamics. The results of this work can be exploited as a tool to provide insights for multi-robot control where agents are placed in similar scenarios.**

I. Introduction

The application of robotic technologies in working environments has led in the last decades to significant benefits for what concerns productivity. In particular, mobile robots have been introduced to assist humans in order to reduce fatigue, increase precision, and improve the quality of products. During assistance tasks, a robot must be capable of performing basic autonomous operations involving both navigation and motion planning. Moreover, the task of controlling the motion between multiple agents becomes challenging when many robots are required to work together in the same environment. This is due to the fact that, in practical scenarios, usually different robots have different objectives to pursue. In the past two decades, a wide range of these scenarios has been studied extensively in the field of control theory. In fact, even if the origin of control theory is related to the control of a single system through different control methodologies, the attention in this field has shifted to the control of multiple interconnected systems since many benefits can be obtained by replacing a unique complex system with several simple systems [1]. However, recently, also techniques related to the field of game theory have started to gain attention [2]–[7]. The application of game theory in engineering as a control technique allows to model the interaction among different agents, which make individual local decisions pursuing a global and common objective, the Nash equilibrium. The most common and natural setup exploiting game theory in the robot framework involves multi-robotic systems where a number of non-adversarial robots (but not necessarily explicitly cooperative) are aimed to accomplish a task while being limited in communication and having limited resources, such as available energy. Typically, these kinds of multiagent environments are modeled as stochastic games. A stochastic game is a theoretical model representing multi-state multi-agent environments having Markovian property and a stochastic inter-state transition rule, and can be used to model inter-agent interactions in such environments. Current frameworks for stochastic games assume perfectly rational agents, which is an assumption that is violated in a variety of real-world scenarios, e.g., human-robot interaction. However, when multi-robot scenarios are concerned, this assumption is more realistic. In practice, typically agents don't know their reward functions or state transition probabilities, which means that in order to find the Nash Equilibrium of the game, a learning problem arises. If an agent directly learns about its optimal policy without knowing either the reward function or the state transition function, such an approach is called model-free reinforcement learning [13]. Research investigating reinforcement learning techniques in the context of multi-agent robotic systems have been particularly interesting in the last years since a wide range of algorithms have been developed for solving these games [8]–[13]. So far, the standard setting which has been used as a reference for the learning of these algorithms is the one of two-dimensional grid games, where a number of robots are placed on square cells and are allowed to move in 4 directions: *Up, Down, Left* and *Right*. On the other hand, this simple structure lacks in modeling aspects that sometimes are fundamental when dealing with the dynamics in trajectory control.

In this paper, we examine and simulate two original game-theoretical models, where new topological scenarios are considered. In particular, the first one concerns two drones in a three-dimensional grid environment, where cells are considered as voxels, whereas the second involves two robots in a two-dimensional grid, where cells are defined as hexagons. The specific choice of these settings is based on the fact that similar scenarios can be encountered in real-world situations when multiple robots are located in the same environment. This project aims at applying the strength of game theory in order to infer some insights regarding the robots' coordinated motion control and provide an extension to the results developed so far in this field. The rest of this paper is organized as follows. Section II reviews the applications of game theory related to robot motion control problems already existing in the literature. Section III provides the background theory related to stochastic games which have been exploited in order to solve the games. Section IV and V present two original scenarios and discuss some

[†]Department of Information Engineering, University of Padova, email: {elena.camuffo, luca.gorghetto}@studenti.unipd.it

results. Section VI concludes the paper and outlines the main guidelines and possible expansions for future work.

II. Related Work

In the past few years, the use of game theory has grown extensively in the robotic research community. In fact, it has been exploited for representing, comparing, and providing insights to a wide class of problems in robotics. As an example, game theory has been used in robust control for the landing an aircraft [6] or Robotic Manipulators [7]. LaValle and Hutchinson [2] were among the first who proposed game theory for the high-level planning of multiple robot coordination. In this context, relevant applications include a multi-robot search for several targets [3], the shared exploration of structured workspaces like building floors [4], or coalition formation [5]. In parallel, recently the interest in the implementation of game theoretical techniques in conjunction with reinforcement learning (RL) increased significantly. Single-Agent RL has seen wide application in robotics, such as in robotic arm control [14], service robots [15] and autonomous robotic nanofabrication [16]. In particular, single-agent methods have been applied in a straightforward fashion also to each agent in multiagents domains such as robotic soccer [17]. However, as recognized in [13], by doing so the familiar theoretical guarantees no longer apply since the environment can no longer be considered as stationary. The actual extension of reinforcement learning techniques from single-agent to multi-agent robotic environments includes two main classes of learning algorithms, called *adaptive learning algorithms* and *equilibrium learning algorithms*. The main difference between these two is that in the latter case agents are calculating an equilibrium solution assuming that their opponents are rational, and their convergence is limited to a number of cases where these equilibria are identifiable. The adaptive learning agents, on the contrary, make no assumptions about their opponents' rationality, learning capabilities and the solution type they are searching. These learning algorithms are proven to be able to converge in self-play (i.e., when learning "against" agents that are using the same learning algorithm) to an equilibrium solution in a wide variety of repeated matrix games. Among adaptive algorithms, the ones which have been exploited more frequently are Infinitesimal Gradient Ascent (IGA) [8], Policy Hill-Climbing (PHC) [9] and Adaptive Play Q-learning (APQ) [10]. Regarding equilibrium learning algorithms instead, it is possible to mention popular techniques such as Minimax Q-learning [11], Friend-or-foe Q-learning [12] and Nash Q-learning [13]. In particular, the latter was the one that was exploited in our simulations. All these techniques were empirically tested by their respective authors on the different test benches. However, although these algorithms were tested on a number of repeated matrix games and on some examples of stochastic games [13], a number of questions is remaining whether these algorithms are well extensible to the general form of stochastic games. Our study brings the contribution of using more advanced underlying models than those already existing in the literature, providing thus a first proof of their effectiveness in a wide variety of possible new settings.

III. Theoretical Background

In this section, we provide the theoretical framework which is needed in order to understand the applications investigated in this paper. According to [26]:

Definition 1. A robot is a physically situated intelligent agent, i.e., a system that perceives its environment and takes actions which maximize its chances of success.

A robot can move in the environment and its position at time t is generally denoted with x_t. The temporal sequence of locations, or *path*, is given as $X_T = \{x_0, x_1, x_2, \ldots, x_T\}$, where $T \leq \infty$ denotes the terminal time, at which the game ends. The initial location x_0 often serves as a point of reference for the estimation algorithm. We define u_t the odometry that characterized the motion between time $t-1$ and time t, obtained from the robot's wheel encoders or from the controls given to those motors. Therefore the sequence $U_T = \{u_0, u_1, u_2, \ldots, u_T\}$ characterizes the relative motion of the robot, given by the sequence of its actions. Let m denote the map of the environment, which is supposed to be static, i.e., time-invariant. The robot measurements establish information between features in m and the robot location x_t. If we assume, without loss of generality, that the robot takes exactly one measurement (observation) at each point in time, the sequence of measurements is given as $Z_T = \{z_1, z_2, z_3, \ldots, z_T\}$.

Definition 2. The *localization problem* [21] is the problem to obtain the path or current position of the robot $x_{0:T}$ given the robot's controls $u_{1:T}$ and observations $z_{1:T}$.

Localization is needed in order to perform motion planning, i.e., the ability for an agent to compute its own collision-free path motion towards certain goal. Motion planning is performed knowing the robot own geometry and kinematics, its initial and goal positions, and the geometry of the environment supposing static obstacles. These definitions show that robots' scenarios fit the framework of stochastic games, explained below.

Stochastic games model multi-agent systems with discrete-time and non-cooperative nature, meaning that players pursue their individual goals and cannot form an enforceable agreement on their joint actions. In a stochastic game, agents choose actions simultaneously. The state space and action space are assumed to be discrete. A formal definition is the following:

Definition 3. An n-player stochastic game Γ is a tuple $\langle S, A^1, \ldots, A^n, r^1, \ldots, r^n, p \rangle$, where S is the state space, A^i is the action space of player i, $r^i : S \times A^1 \times \cdots \times A^n \rightarrow R$ is the payoff function for player i, $p : S \times A^1 \times \cdots \times A^n \rightarrow \Delta(S)$ is the transition probability map, where $\Delta(S)$ is the set of probability distributions over state space S.

Given state s, agents independently choose actions $a_1, \cdots, a_n$, and receive rewards $r^i(s, a_1, \cdots, a_n)$, $i = 1 \cdots n$.

The state then transits to the next state s' based on fixed transition probabilities, satisfying the constraint:

$$\sum_{s' \in S} p\left(s' \mid s, a^1, \ldots, a^n\right) = 1 \tag{1}$$

General-sum stochastic games allow the agents' rewards to be arbitrarily related. As special cases, *zero-sum stochastic games* are instances where agents' rewards are always negatively related. In a *discounted stochastic game*, the objective of each player is to maximize the discounted sum of rewards, with discount factor $\beta \in [0, 1)$. A strategy $\boldsymbol{\pi}$ is defined as a plan for playing a game. Here $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_t, \ldots)$ is defined over the entire course of the game, where π_t is called the decision rule at time t. A decision rule is a function $\pi_t : \mathbf{H}_t \rightarrow \Delta(A)$, where $\mathbf{H}_t$ is the space of possible histories at time t, with each $H_t \in \mathbf{H}_t, H_t = (s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$, and $\Delta(A)$ is the space of probability distributions over the agent's actions. $\boldsymbol{\pi}$ is called a stationary strategy if $\pi_t = \bar{\pi}$ for all t, that is, the decision rule is independent of time. $\boldsymbol{\pi}$ is called a behavioral strategy if its decision rule may depend on the history of the game play, $\pi_t = f_t(H_t)$. If we let $\boldsymbol{\pi}^i$ be the strategy of player i, then for any given initial state s, player i tries to maximize:

$$v^i\left(s, \boldsymbol{\pi}^1, \boldsymbol{\pi}^2, \ldots, \boldsymbol{\pi}^n\right) = \sum_{t=0}^{\infty} \beta^t E\left(r_t^1 \mid \boldsymbol{\pi}^1, \boldsymbol{\pi}^2, \ldots, \boldsymbol{\pi}^n, s_0 = s\right) \tag{2}$$

At this point, we can formally define the concept of Nash Equilibrium (NE) for stochastic games.

Definition 4. In a stochastic game Γ, a Nash equilibrium is a tuple of n strategies $\left(\boldsymbol{\pi}_*^1, \ldots, \boldsymbol{\pi}_*^n\right)$ such that for all $s \in S$ and $i = 1, \ldots, n$,

$$v^i\left(s, \boldsymbol{\pi}_*^1, \ldots, \boldsymbol{\pi}_*^n\right) \geq v^i\left(s, \boldsymbol{\pi}_*^1, \ldots, \boldsymbol{\pi}_*^{i-1}, \boldsymbol{\pi}^i, \boldsymbol{\pi}_*^{i+1}, \ldots, \boldsymbol{\pi}_*^n\right)$$

for all $\boldsymbol{\pi}^i \in \boldsymbol{\Pi}^i$, where $\boldsymbol{\Pi}^i$ is the set of strategies available to agent i.

The meaning of a NE is that of a joint strategy where each agent's is a best response to the others'. In general, the strategies that constitute a NE can be behavioral strategies or stationary strategies. The result of Fink [18] proved that every n-player discounted stochastic game possesses at least one NE in stationary strategies. In this paper, we limit our study to stationary strategies. Therefore if a state is visited multiple times, the players' choices would be the same each time. Non-stationary strategies, which allow conditioning of action on history of play are relatively less studied in this framework. Finally, the theoretical foundations of multi-agent agent Q-learning for general-sum stochastic games can be given. For an n-agent system, we define a Nash Q-value as the expected sum of discounted rewards when all agents follow specified Nash equilibrium strategies from the next period on. More precisely:

Definition 5. Agent i's Nash Q-function is defined over $\left(s, a^1, \ldots, a^n\right)$, as the sum of Agent i's current reward plus its future rewards when all agents follow a joint Nash equilibrium strategy. That is,

$$\begin{aligned} Q_*^i\left(s, a^1, \ldots, a^n\right) = r^i\left(s, a^1, \ldots, a^n\right) + \\ + \beta \textstyle\sum_{s' \in S} p\left(s' \mid s, a^1, \ldots, a^n\right) v^i\left(s', \boldsymbol{\pi}_*^1, \ldots, \boldsymbol{\pi}_*^n\right) \end{aligned} \tag{3}$$

where $\left(\boldsymbol{\pi}_*^1, \ldots, \boldsymbol{\pi}_*^n\right)$ is the joint Nash equilibrium strategy, $r^i\left(s, a^1, \ldots, a^n\right)$ is agent i's one-period reward in state s and under joint action $\left(a^1, \ldots, a^n\right), v^i\left(s', \boldsymbol{\pi}_*^1, \ldots, \boldsymbol{\pi}_*^n\right)$ is agent i's total discounted reward over infinite periods starting from state s' given that agents follow the equilibrium strategies.

Before giving basic idea of the Nash Q-learning algorithm (details can be found in [19]), which allows to evaluate these Nash-Q functions, we need to clarify the distinction between Nash equilibria for a stage game (one-period game), and for the stochastic game (many periods).

Definition 6. An n player stage game is defined as $\left(M^1, \ldots, M^n\right)$, where for $k = 1, \ldots, n, M^k$ is agent k's payoff function over the space of joint actions, $M^k = \left\{r^k\left(a^1, \ldots, a^n\right) \mid a^1 \in A^1, \ldots, a^n \in A^n\right\}$, and r^k is the reward for agent k.

If now we let σ^{-k} be the product of strategies of all agents other than $k, \sigma^{-k} \equiv \sigma^1 \cdots \sigma^{k-1} \cdot \sigma^{k+1} \cdots \sigma^n$, we have:

Definition 7. A joint strategy $\left(\sigma^1, \ldots, \sigma^n\right)$ constitutes a Nash equilibrium for the stage game $\left(M^1, \ldots, M^n\right)$ if, for $k = 1, \ldots, n$

$$\sigma^k \sigma^{-k} M^k \geq \hat{\sigma}^k \sigma^{-k} M^k \quad \text{for all } \sigma^k \in \hat{\sigma}\left(A^k\right)$$

In the Nash Q-Algorithm, at each time t, the i-th agent observes the current state and takes its action. After that, it observes its own reward, actions taken by all other agents, others' rewards, and the new state s'. It then calculates a NE $\pi^1(s') \cdots \pi^n(s')$ for the stage game $\left(Q_t^1(s'), \ldots, Q_t^n(s')\right)$, and updates its Q-values according to:

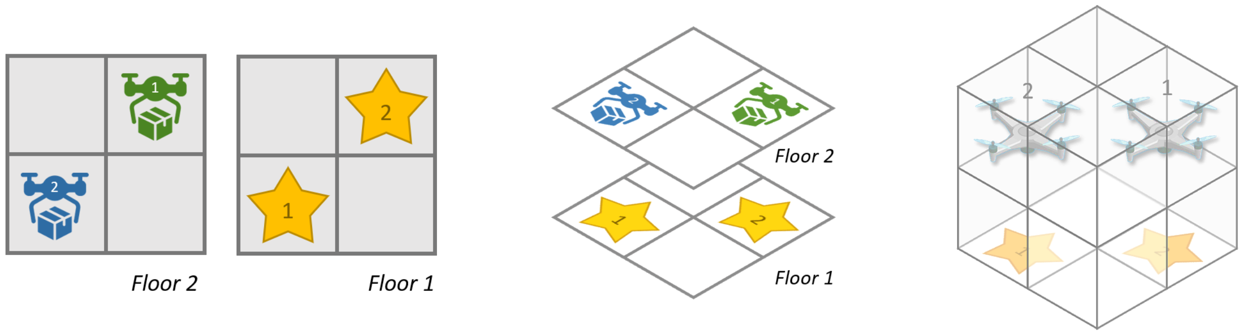
$$\begin{aligned} Q_{t+1}^i\left(s, a^1, \ldots, a^n\right) = (1 - \alpha_t) Q_t^i\left(s, a^1, \ldots, a^n\right) + \\ \alpha_t\left[r_t^i + \beta \operatorname{Nash} Q_t^i(s')\right] \end{aligned} \tag{4}$$

$$\text{where: } \operatorname{Nash} Q_t^i(s') = \pi^1(s') \cdots \pi^n(s') \cdot Q_t^i(s')$$

This general framework will be used in the following sections to analyze the considered games.

IV. Scenario I

The first scenario consists of a game involving two drones in a three-dimensional environment. This choice is due to the fact that generally in literature, only cases with mobile robots are considered for game theory analyses. This scenario, instead, provides an approach of solving problems of different type that are as much common in robotics. The considered example shows a couple of drones, that need to pass into a tight area to get their respective own way in order to safely deliver their packets. They need to avoid the other drone staying safely with no collisions, but they need also to overcome the issue in the lowest amount of time possible, therefore following the shortest path to achieve their goals. The drones are chosen for their easy representation, but this scenario can fit well



a: 2D representation

b: Floors representation

c: 3D representation

Fig. 1: Scenario I.

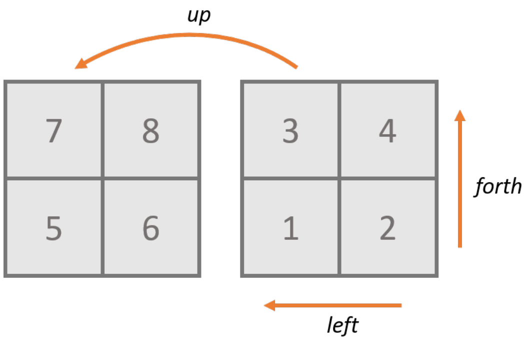


Fig. 2: Enumeration and moves, Scenario I.

also a manipulators' environment. In particular if the two manipulators have a sufficient level of autonomy, they must decide their own trajectory in order to pick their respective object. If the two considered manipulators are required to share a small space, the problem occurs and a game theoretical analysis comes into play. The game is dealt first assuming conditions like in grid-game-1 of [13]; then the model is adapted to another similar situation. Finally conclusions are drawn on the results obtained.

A. Setting

The grid-game considered in scenario I is built on a 3D grid, where each voxel represents a position, the robot can occupy. Each robot can move only one voxel at a time, in six possible directions: *Up, Down, Left, Right, Forth, Back*. The two robots are placed in the opposite corners of the upper floor and try to reach their goal on the opposite corner in the lower floor. If they attempt to move into the same cell (excluding a goal cell), they are bounced back to their previous cells. The game ends as soon as a robot reaches its goal. Reaching the goal earns a positive reward. In case both robots reach their goal cells at the same time, both are rewarded with positive payoffs. Note that, when the scenario is deterministic, the two shortest paths that do not interfere with each other constitute a Nash equilibrium, since each path (strategy) is a best response to the other. Therefore, the objective of each robot is to reach its goal with the minimum number of steps without colliding. The robots do not know the location of their goal at the beginning of the learning period. Furthermore, the two robots choose their actions simultaneously. They observe the previous actions of both, the current state (current position of both) and their immediate rewards.

Figure 1 shows this game using three different representations. The action space of robot i, $i = 1, 2$, is $A^i = \{$*Up, Down, Left, Right, Forth, Back*$\}$; the state space is $S = \{(1, 2), (1, 3), \ldots, (8, 7)\}$, where a state $s = (l^1, l^2)$ represents the two agents' joint location. Robot i's location is represented by a position index, as shown in figure 2. The state transitions are deterministic and the rewards that each robot can receive are:

- 100 if it reaches the goal position.
- -1 if it collides with the other robot.
- 0 otherwise.

B. Analysis

Let the initial state be $s_0 = (8, 5)$, as in figure 1, and the discount factor $\beta = 0.99$. The value of the game can be computed for both players. In this case, as the game is symmetric, we can restrict the analysis to robot 1 only.

The value of the game for robot 1 is defined, as in (2), as its accumulated reward when both agents follow their Nash equilibrium strategies,

$$\begin{aligned} v^1(s_0) &= 0 + 0.99 \cdot 0 + 0.99^2 \cdot 0 + 0.99^3 \cdot 100 \\ &= 97.0 \end{aligned}$$

This same value can be yield by different strategies. For example figure 3 shows that, fixing the path of robot 2, robot 1 can reach the Nash Equilibrium following one of the proposed different strategies.

Based on the values of each state, it is possible to derive the Nash Q-values for robot 1 in state s_0 using (3),

$$\begin{aligned} Q^1_*(s_0, \textit{Left, Forth}) &= -1 + 0.99 \cdot v((8, 5)) = 95.1 \\ Q^1_*(s_0, \textit{Down, Forth}) &= 0 + 0.99 \cdot v((6, 7)) = 97.0 \end{aligned}$$

The whole set of Nash Q-values found is shown in table 1. There are seven Nash Equilibria given by the couple of payoffs

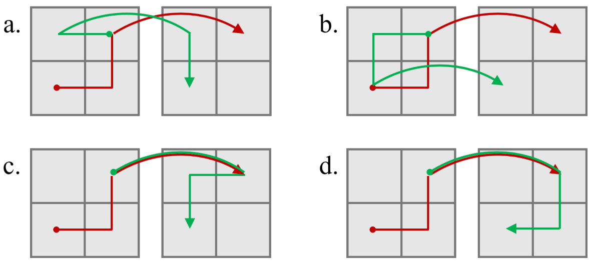


Fig. 3: Different Nash Equilibrium Paths, Scenario I.

$(97.0, 97.0)$.

Let's assume now that the robots know their goal position. The robots always knows also their own and the other robot's location and the moves are deterministic; consequently, the game can be modeled as a stage game with complete and perfect information. Figure 4 represents the first two stages of such game.

At the beginning, the drones are placed on the same floor. This means that in the first stage they are lead both to choose the move *Down*. Then (*Down*, *Down*) is the dominant strategy. This would also be the case if the number of floors would be higher, since only in the last stage, where the two are in the same floor of their objective, they would choose the strategy that brings them to their goal.

The game is a *collaborative game* where the collaborative strategy at stage 1 is to move *Down* until the lowest floor (the one with the goals) is reached. That strategy is also the stage Nash Equilibrium. At the second stage, the game becomes no longer collaborative and no Nash Equilibrium in pure strategy can be found. The game becomes an *Odds and Evens* game; therefore the Nash Equilibrium is achieved using mixed strategies, playing each move a half of the times.

This *discoordination game* of the second stage leads to two different results: if the discoordination strategy is achieved, the robots go to different cells; in this case the game ends in the next following stage with payoff 100β for both. If instead they attempt to go to the same cell, the stage game is repeated until discoordination is achieved; in the latter case the payoff is given by $-1 + \sum_{t=1}^{T-1} \beta^t \cdot (-1) + 100\beta^T$ for both.

This means that the best strategy that the two drones can adopt is to reduce the problem to a coin flip, and postpone the inevitable fight to the last round, which is discounted and has therefore a lower impact on the final payoff. The result suites well the drones' situation, but fits almost better the manipulators' one. It suffices to think about that with a random strategy the two robotic arms can collide and loose rounds (additional rounds, not only the ones due to the collision of the end effectors); instead if they first go down towards their objective, then the movement is straightforward despite the probability of collision of the end effectors.

V. Scenario II

The second scenario considered is instead a classical bi-dimensional grid-game for mobile robots. The big difference introduced here lies in the shape of the cells in the map. The hexagons were chosen since they represent the best shape to cover a surface with the lowest number of edges. This shape leads also to a largest set of moves for the players, and a more complicate but more realistic design of the game. In addition, this scenario can also be modeled as one of the state-of-art problems in the field of robotics: given a formation of robots, they have to pass through a tight passage and the original formation cannot be preserved; one of the robots has to enter the tight passage first, and the others after. There is no way to decide which of the robot has to enter first, since they are all equal, equally rational and initially at the same distance from the passage; that is where game theory comes into play. The hexagonal scenario models well the problem, as the goal position is bounded at the top and reachable from multiple cells, thing that would be not possible in a squared scenario. The transition probabilities for some states are modeled to represent the intrinsic cost to perform a non trivial maneuver, as the agents are assumed to be non-holonomic mobile robots.

	Right	*Forth*	*Down*
Left	$97.0, 97.0$	$95.1, 95.1$	$97.0, 97.0$
Back	$95.1, 95.1$	$97.0, 97.0$	$97.0, 97.0$
Down	$97.0, 97.0$	$97.0, 97.0$	$97.0, 97.0$

TABLE 1: Scenario I: Nash Q-values in state $(8, 5)$

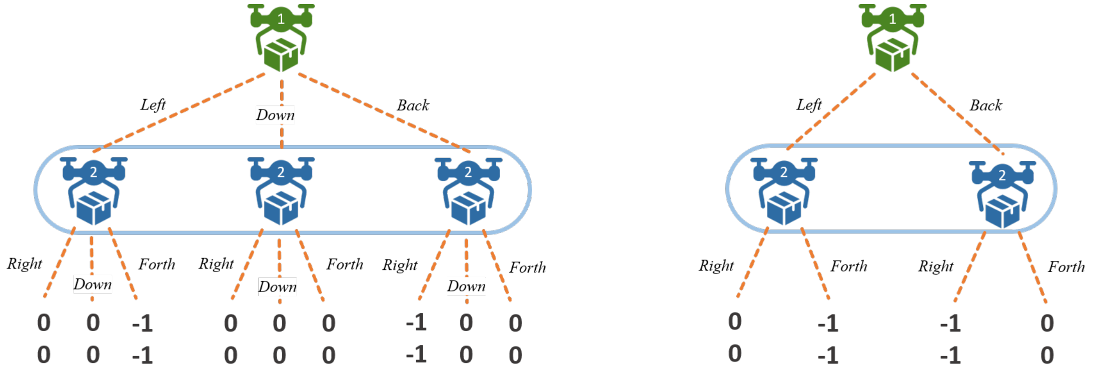
A. Setting

The map of scenario II is made of seven hexagonal cells. Each robot can move only one cell at a time, in 6 possible directions: *North* (N), *South* (S), *North-East* (NE), *North-West* (NW), *South-East* (SE), *South-West* (SW). As in the first setting, if the two robots attempt to move into the same cell, they are bounced back to their previous cells. The game ends as soon as a robot reaches the goal. Reaching the goal earns a positive reward. The two robots cannot reach the goal simultaneously, but in case one robot reaches the goal cell, both are rewarded with positive payoffs. The robots know the location of their goal at the beginning and choose their actions simultaneously. They observe the previous actions of both, the current state (current position of both) and their immediate rewards.

The action space of robot i, $i = 1, 2$, is $A^i = \{$*North* (N), *South* (S), *North-East* (NE), *North-West* (NW), *South-East* (SE), *South-West* (SW)$\}$; the state space is $S = \{(1,2), (1,3), \ldots, (7,6)\}$, where a state $s = (l^1, l^2)$ represents the two agents' joint location. Robot i's location is represented by a position index. Figure 5 shows this game and the enumeration of the cells. The rewards that each robot can receive are:

- 100 if it reaches the goal position.
- 50 if the other reaches the goal position.
- -1 if it collides with the other robot.
- 0 otherwise.

The state transitions are deterministic except the followings: if the current state is $(5, 6)$ and the robots attempt both to reach the goal, they succeed or remain in the previous state and they



a: Stage Game 1

b: Stage Game 2

Fig. 4: Extensive form representation of Scenario I as a Stage Game.

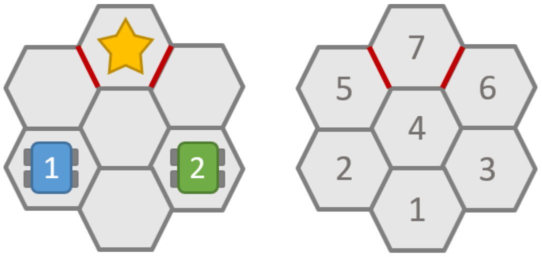


Fig. 5: Scenario II.

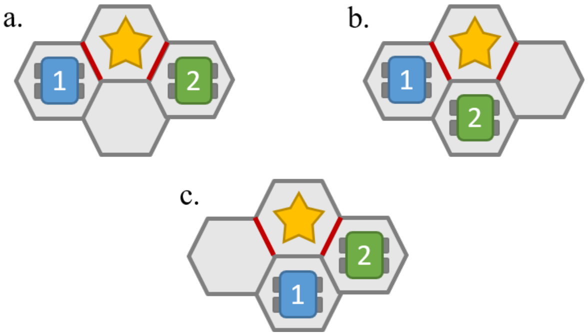


Fig. 6: Scenario II, stage 2.

never collide. Both succeed with probability p, and remain in the previous location with probability $1-2p$, where $p=1/3$. If instead they are in state $(4,6)$, they have probability $q=1/2$ to move to state $(7,6)$ and probability $1-q$ to remain in the same state; in the latter case they receive a null reward since the collision occurs in the goal position. Analogous reasoning holds for state $(5,4)$.

B. Analysis

Let's consider the initial state as $s_0 = (2,3)$ and the discount factor $\beta = 0.99$ as before. The optimal values of the game for both players must be computed in order to understand which strategy is convenient to be adopted. Since the game is symmetric, the values are computed for player 1 only. Some states hold transition probabilities, thus only the deterministic values are immediate to be computed:

$$v^1((x,y)) = 0.99 \cdot 100 = 99, \qquad x=4,5,6,\ \ y=1,2,3$$
$$v^1((x,y)) = 0.99 \cdot 50 = 45.5, \qquad x=1,2,3,\ \ y=4,5,6$$

Then, if one of the players goes to position 1, the game becomes no longer probabilistic and comes to an end in the next following stage. In the case both players choose position 1 or position 4, they earn negative rewards and the game is repeated, discounted.

The other Nash Q-values can be computed only in expectation. However, if in the first stage neither of the deterministic strategies is played, the game falls into one of the configurations represented in figure 6. In these states, the players aim at reaching their objective as soon as possible, and have no incentive to go back to their previous state. The total value of the game in the initial state can be computed using backward induction: the analysis starts from the states of figure 6, the optimal values are computed and used to draw conclusions on the overall scenario.

Let's define $A_1 = v^1((5,6))$, $A_2 = v^2((5,6))$, $B_1 = v^1((5,4))$, $B_2 = v^2((5,4))$. The Nash Q-values in state $(5,6)$ are reported in table 2, where the strategy of coming back is not considered, as dominated. Here, the only Nash equilibrium is the joint strategy (*NE, NW*), bringing the implication:

$$50 + \frac{1}{3}0.99A_i = A_i \rightarrow A_i = \frac{5000}{67} \approx 74.63, \quad i=1,2$$

Thus, the only Nash Equilibrium gives payoffs $(\frac{5000}{67}, \frac{5000}{67})$, and implies that both players attempt to reach the objective, since they cannot receive a negative reward, and one third of the times they reach the goal, but another third of the times they receive a positive reward. This strategy is not collaborative, but selfish and myopic.

The Nash Q-values for case $(4,6)$ are shown in table 3. For state $(5,4)$ they are analogous with the row player as the column one and viceversa. Also here the Nash Equilibria is unique, and given by the joint strategy (*SE, NW*). In fact the strategy (*NE, NW*) is penalized by the discount factor and its

	NW	*SW*
NE	$50 + \frac{1}{3}0.99A_1, 50 + \frac{1}{3}0.99A_2$	$100, 50$
SE	$50, 100$	$-1 + 0.99A_1, -1 + 0.99A_2$

TABLE 2: Scenario II: Nash Q-values in state $(5, 6)$

	N	*NE/NW*
NE	$25 + \frac{1}{2}0.99B_1, 50 + \frac{1}{2}0.99B_2$	$100, 50$
SE	$50, 100$	$0.99B_2, 0.99B_1$

TABLE 3: Scenario II: Nash Q-values in state $(4, 6)$

	N	*NW*	*SW*
N	$\frac{4950}{67}, \frac{4950}{67}$	$45.5, 99$	$99, 45.5$
NE	$99, 45.5$	$-1 + 0.99R_1, -1 + 0.99R_2$	$99, 45.5$
SE	$45.5, 99$	$45.5, 99$	$-1 + 0.99R_1, -1 + 0.99R_2$

TABLE 4: Scenario II: Nash Q-values in state $(2, 3)$

implication

$$25 + \tfrac{1}{2}0.99B_1 = B_1 \rightarrow B_1 = \tfrac{5000}{101} \approx 49.50$$
$$50 + \tfrac{1}{2}0.99B_2 = B_2 \rightarrow B_2 = \tfrac{10000}{101} \approx 99.01$$

cannot hold, since to be a Nash Equilibrium, the strategy requires the condition:

$$25 + \frac{1}{2}0.99B_1 \geq 50 \rightarrow B_1 \geq \frac{5000}{99} \approx 50.50$$

which is not met.

Therefore only the joint strategy (*SE, N*) is a Nash Equilibrium and gives payoffs $(50, 100)$. This configuration underlines that the central player has a huge advantage, and certainly reaches the goal first. The lateral player is led to leave the floor to the other, in order to maximize its own payoff.

Observe that for the first subgame (referred to initial state $(5, 6)$) the value of p is almost irrelevant. In fact the parametric implication leads to:

$$100p + 50p + 0.99R_i(1 - 2p) = R_i, \quad i = 1, 2$$
$$\rightarrow R_i = \frac{150p}{0.01 + 0.99 \cdot 2p}$$

which is a Nash Equilibrium if satisfies the condition:

$$R_i = \frac{150p}{0.01 + 0.99 \cdot 2p} \geq 50 \rightarrow p > \frac{1}{102} \approx 9.80 \times 10^{-3}$$

Then the analysis led is valid in every case, except if the probability to win is extremely low.

Also in case the initial state is $(5, 4)$, the probability value has not an high impact on the analysis. In fact:

$$B_1 = \frac{50q}{1 - 0.99(1 - q)} \geq 50 \rightarrow q \geq 1$$

$$B_2 = \frac{100q}{1 - 0.99(1 - q)} \geq 50 \rightarrow q \geq \frac{1}{299} \approx 3.34 \times 10^{-3}$$

and the first inequality cannot hold. Thus the strategy (*NE, NW*) is never a Nash equilibrium.

At this point, the Nash Q-values in state $s_0 = (2, 3)$ can be computed, and they are shown in table 4. There are multiple Nash Equilibria, given by the joint strategies (*NE, N*), (*N, NW*), (*SE, NW*) and (*NE, SW*). The payoffs given are always $(45.5, 99)$ or $(99, 45.5)$ depending on which of the players chooses the leading strategy of occupy the central position. However contrarily to what is straightforward to think, the best strategy is a mixed strategy that includes the possibility of going to position 1, which is incredibly a strategy as good as going *North*. On the other hand the central position guarantees the higher payoff, but it is not to be played always, since the probability of collision with the other agent is relevant.

The result is robotics terms is very accurate. It proves that in a similar situation, the robots collaboration is essential to cope with an issue like this, and pass through the tight road in the least time possible. Nevertheless, observing the result of state $(5, 4)$, if an agent has to do a complex maneuver to reach the goal, it has better to leave the floor to the other agent and come after him. Finally the result of state $(5, 6)$ proves that the selfish strategy of getting closer to the goal trying to take it first is never the best strategy for the global objective, since in that case both the robots need to operate maneuvers, and there is a big probability to loose rounds for nothing.

VI. Experimental Results

In this section the results of the experiments lead are given. The game tested is scenario I, which is provided with the initial configuration, supposing the players do not know their goal location at the beginning of the learning. We developed a version of Nash-Q learning algorithm originally proposed by Hu and Wellman. The choice of using this type of equilibrium learning algorithm is motivated (as stated in section I) by the fact that, the rationality assumption is reasonable when dealing with robots. Matlab was used for the implementation, which resulted not straightforward and required adjustments to obtain

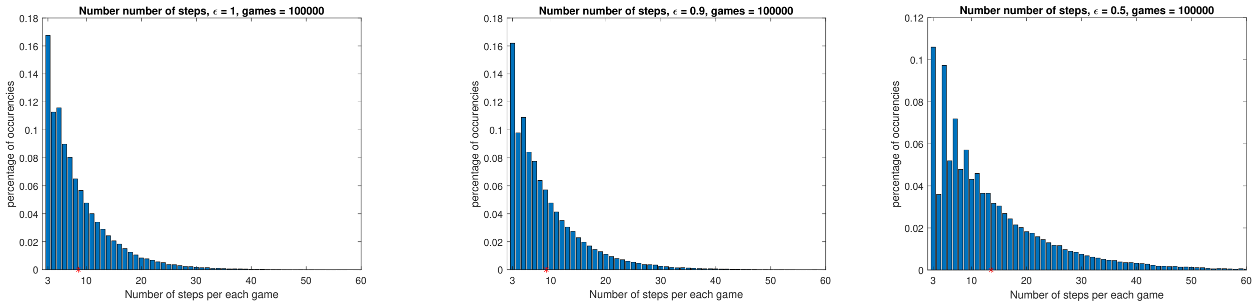


Fig. 7: Path length with different ϵ values.

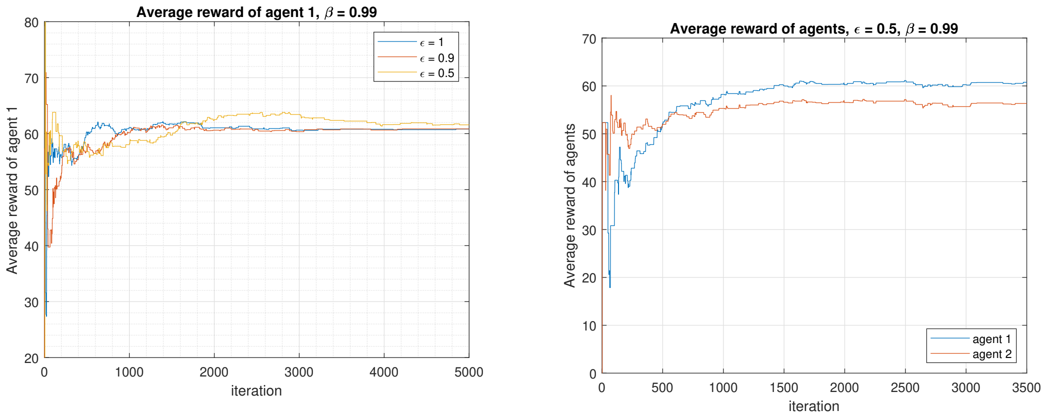


Fig. 8: Average value of the Q-function vs number of iterations.

good results. The function takes advantage of the Lemke Howson algorithm [23] to find the First Nash Equilibrium, as in the original version. The algorithm uses a multiplayer ϵ-greedy exploration strategy, then the strategies to adopt can be *explore*, *exploit*, or *explore and exploit*. In the implemented version, the value of parameter ϵ, controls the probability of choosing the *exploit* strategy. The average length of the path to reach the goal is measured with different ϵ and the results are given in figure 7. Notice that the average path length is lower if the *exploit* strategy is used; moving the robot at random in the space, keeps it into the play for a long time and implies a waste. In addition, the number of steps per path results being distributed according to a Geometric distribution, which generally models the waiting time of an event, in this case the reaching of the goal. The average length is quite high on average because they do not know where their goals are, but choosing always the best move of the situation the average path length shortens. This result is also influenced by the discount factor.

The results for the Q-function are reported in figure 8. The behaviour is always convergent, but the speed of convergence depends on the method chosen: for players following totally an *exploit* strategy the convergence is faster with respect to the case in which they play $exploit$ half of the times and $explore$ the other times. The average values, computed for both players, are similar to the grid-game-1 of [13], reported in [28], to which scenario I is inspired. However the convergence to a Nash equilibrium is not always guaranteed by the Lemke Howson algorithm, used by the Nash-Q learning during the game, and repeated tries have required to lead all the configurations to convergence. Nevertheless, in the final Q-matrix obtained, the Nash equilibria are computed neglecting the moves that cannot be performed in that state. The result is that, when the convergence is reached, it gives always one of the seven Nash Equilibria reported in table 2.

VII. Conclusions

In this paper, two possible original extensions of the classic two-dimensional grid games are investigated. The development includes a simulation of the first game, led with our own implementation of the Nash Q-algorithm. The results obtained are promising as the Nash equilibria derived in the theoretical analysis coincide with the ones given in the simulation. This way, the prediction capability of the Nash Equilibria has been verified, as well as the functioning of the Nash-Q technique in this extended scenario. Moreover, the algorithm provides a mean to evaluate the average path length metric in the first game for different values of ϵ and prove the convergence of the Q function. On the other hand, the impact of these analysis on the actual meaning of the real situations is to be considered. The overall result proves that the collaboration between the agents is always the best choice even in scenarios which can seem very different in appearance.

What we obtained so far is pretty satisfactory, but there is still room for improvement, for example considering different configurations for the transition probabilities or different discount factors. In addition the analysis can be extended to other situations. As a first extension, it is possible to simulate also the second scenario, developed only theoretically, to confirm the results. At the same time, these two games can be tested with different types of reinforcement learning algorithms to compare their performances. Possible developments involve also the consideration of different scenarios, including for example further elements of uncertainty, like obstructions in cells or even additional numbers of players.

References

[1] F. Garin and L. Schenato. *A survey on distributed estimation and control applications using linear consensus algorithms*. In: Network Control Systems. Ed. by Springer. 2010, pp. 75107.

[2] LaValle S, Hutchinson S (1993) *Game theory as a unifying structure for a variety of robot tasks*. In: Proceedings of IEEE international symposium on intelligent control, pp 429-434

[3] Meng Y (2008) *Multi-robot searching using game-theory based approach*. Int J Adv Robot Syst 5(4):341-350

[4] Skrzypczyk K (2005) *Game theory based task planning in multi robot systems*. Int J Simulat 6(6):50-60

[5] Ghazikhani A, Mashadi HR, Monsefi R (2010) *A novel algorithm for coalition formation in multi-agent systems using cooperative game theory*. In: Procedings of IEEE Iranian conference on electrical engineering, pp 512-516

[6] Ganebny S, Kumkov S, Patsko V (2006) *Constructing robust control in game problems with linear dynamics*. In: Petrosjan L, Mazalov V (eds) Game theory and applications, 11th edn. Nova Science Publishers, New York

[7] Hamano F. (1991) *Robust Control of Robotic Manipulators*. In: Jordanides T., Torby B. (eds) Expert Systems and Robotics. NATO ASI Series (Series F: Computer and Systems Sciences), vol 71. Springer, Berlin, Heidelberg.

[8] Singh, S., Kearns, M., Mansour, Y.: *Nash convergence of gradient dynamics in general-sum games*. In: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI'94), San Francisco, CA, Morgan Kaufman (1994) 541-548.

[9] Bowling, M., Veloso, M.: *Multiagent learning using a variable learning rate*. Artificial Intelligence 136(2) (2002) 215-250.

[10] Gies, O., Chaib-draa, B.: *Apprentissage de la coordination multiagent : une methode basee sur le Q-learning par jeu adaptatif.* Revue d'Intelligence Artificielle 20(2-3) (2006) 385-412.

[11] Littman, M.: *Markov games as a framework for multi-agent reinforcement learning*. In: Proceedings of the Eleventh International Conference on Machine Learning (ICML'94), New Brunswick, NJ, Morgan Kaufmann (1994) 157-163.

[12] Michael L. Littman, *Friend-or-Foe Q-learning in General-Sum Games*

[13] Junling Hu, Michael P.Wellman, *Nash Q-Learning for General-Sum Stochastic Games*, Journal of Machine Learning Research 4 (2003) 1039-1069.

[14] Franceschetti Andrea, Tosello Elisa, Castaman Nicola, Ghidoni Stefano. (2021). *Robotic Arm Control and Task Training through Deep Reinforcement Learning*.

[15] Y. Jiang, F. Yang, S. Zhang and P. Stone, *Task-Motion Planning with Reinforcement Learning for Adaptable Mobile Service Robots*, 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019, pp. 7529-7534, doi: 10.1109/IROS40897.2019.8967680.

[16] Leinen, Philipp Esders, Malte Schutt, Kristof Wagner, Christian Muller, Klaus-Robert Tautz, F (2020). *Autonomous robotic nanofabrication with reinforcement learning*. Science Advances.

[17] Peter Stone and Richard S. Sutton. *Scaling reinforcement learning toward RoboCup soccer*. In Proc. 18th International Conf. on Machine Learning, pages 537-544. Morgan Kaufmann, San Francisco, CA, 2001.

[18] Fink, A. M. Equilibrium in a stochastic n-person game. J. Sci. Hiroshima Univ. Ser. A-I Math. 28 (1964), no. 1, 89–93. doi:10.32917/hmj/1206139508.

[19] Hu, J., Wellman, M. P, *Multiagent reinforcement learning: Theoretical framework and an algorithm*. Proceedings of the Fifteenth International Conference on Machine Learning (pp. 242-250), 1998.

[20] C. J. C. H. Watkins and P. Dayan, *Q-learning*, Machine Learning, vol. 8, no. 3, pp. 279-292, 1992.

[21] Cyrill Stachniss, John J. Leonard, Sebastian Thrun, *Simultaneous Localization and Mapping*, chapter 46, *Handbook of Robotics*, Springer 2016.

[22] Busoniu, Lucian and Babuska, Robert and De Schutter, Bart and Ernst, Damien, *Reinforcement learning and dynamic programming using function approximators*, CRC Press 2010.

[23] C. E. Lemke and J. T. Howson, Jr. *Equilibrium Points of Bimatrix Games*, Journal of the Society for Industrial and Applied Mathematics, Vol. 12, No. 2 (Jun., 1964), pp. 413-423.

[24] Lloyd S. Shapley. *A note on the Lemke-Howson algorithm*, Pivoting and Extension: Mathematical Programming Studies, Vol. 1, 1974, pp 175-189.

[25] Bruno Codenotti, Stefano De Rossi, Marino Pagan, *An experimental analysis of Lemke-Howson algorithm*.

[26] R. Murphy, *Introduction to AI Robotics*, The MIT Press, Second Edition, 2019.

[27] T. Jaakkola, M. Jordan, and S. Singh, *On the convergence of stochastic iterative dynamic programming algorithms* Neural Computation, vol. 6, no. 6, pp. 1185-1201, 1994.

[28] Pascal De Beck-Courcelle, *Study of Multiple Multiagent Reinforcement Learning Algorithms in Grid Games*, 2013.