FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Coordinating a Team of Agents in the Forest Firefighting Domain

Daniel Cardoso de Moura

Licenciado em Engenharia Informática e Computação
pela Faculdade de Engenharia da Universidade do Porto

Dissertação submetida para satisfação parcial dos
requisitos do grau de mestre
em
Inteligência Artificial e Sistemas Inteligentes

Dissertação realizada sob supervisão de
Professor Doutor Eugénio da Costa Oliveira,
do Departamento de Engenharia Electrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto

Porto, Março de 2006

Para a Lúcia, os meus pais, os pais da Lúcia, o meu irmão, e a Bruna

Abstract

The Team Coordination Model proposed in this thesis is especially concerned with spatial coordination of agents performing in a simulated forest firefighting environment. This model relies in a centralised coordination approach that assigns high-level tasks to agents. Agents have local autonomy and are also capable of cooperating with each other to perform local tasks without using communication.

We will start by discussing the motivations behind this work as well as introducing the reader to the firefighting domain by presenting basic knowledge about forest fires and practical firefighting. Then, we will analyse coordination models that have been developed in the area of MAS that we consider to be relevant for this work. We will then introduce Pyrosim, the forest fire simulator were we will test our team coordination model.

Next, we present the team coordination model. We start by specifying a flexible structure that enables to define different Forest Firefighting Tactics. We then instantiate this model by defining several Tactics that may be executed in the Pyrosim simulator. We also propose a mechanism for agents to cooperate locally while executing shared tasks. We will then focus on the specification of a mechanism for managing overall information about the scenario state. Usually, agents are not able to perceive the entire fire because of their perception limitations, which makes difficult to carry out Tactics. This mechanism enables to overcome this difficulty by merging visual information from every agent. Then, we will report the results of our experiments using different Tactics to coordinate a firefighting team in the Pyrosim simulator.

Finally, we present the conclusions of this work as well as some ideas for future work.

Resumo

O Modelo de Coordenação de Equipas proposto nesta tese foi especialmente concebido para coordenação espacial de agentes que actuam num ambiente simulado de fogos florestais. Este modelo baseia-se na utilização de coordenação centralizada para atribuir tarefas de alto nível aos agentes. Por sua vez, os agentes possuem autonomia local e são capazes de cooperar entre eles localmente para executar estas tarefas, sem que para tal tenham de utilizar comunicação.

Começaremos por discutir a motivação para este trabalho e por introduzir o leitor ao domínio do combate de fogos florestais. Para tal, apresentaremos fundamentos da teoria dos fogos florestais e do seu combate. De seguida, analisaremos alguns modelos de coordenação desenvolvidos na área dos Sistemas Multi-Agente que consideramos serem relevantes para este trabalho. Segue-se uma apresentação do simulador de fogos florestais, o Pyrosim, onde iremos testar o nosso modelo de coordenação de equipas.

Prosseguiremos com a proposta do modelo de coordenação de equipas. Começamos por especificar uma estrutura flexível que permite definir Tácticas de Combate a Fogos Florestais. Depois, instanciamos o modelo definindo diversas Tácticas que podem ser executadas no simulador Pyrosim. Neste contexto, também propomos um mecanismo para cooperação local entre os agentes que é utilizado quando estes têm de executar tarefas em comum. De seguida, especificamos um mecanismo que permite gerir informação relativa ao panorama global do cenário de simulação. Normalmente, um agente não é capaz de percepcionar a totalidade do fogo devido às limitações da sua percepção, o que dificulta a execução de Tácticas. Este mecanismo permite superar este problema fundindo informação sensorial de todos os agentes. A seguir, relatamos os resultados das experiências que levamos a cabo no simulador Pyrosim, relativas à utilização de diferentes Tácticas para coordenar uma equipa de bombeiros.

Finalmente, apresentamos as conclusões relativas a este trabalho, assim como algumas possibilidades de trabalho futuro.

Agradecimentos

São muitas as pessoas a que tenho de agradecer pelo apoio que me prestaram no decorrer desta tese.

Em primeiro lugar, gostaria de agradecer ao Professor Eugénio Oliveira pelos constantes desafios que me colocou, pela forma pertinente e objectiva como levantou os problemas e soluções a seguir, pelos conhecimentos que me transmitiu, pela constante motivação e pela sua compreensão.

Gostaria de agradecer a toda a equipa do Núcleo de Inteligência Artificial Distribuída e Robótica (NIAD&R) da Faculdade de Engenharia da Universidade do Porto, pelos comentários e ideias sobre este trabalho, e pelas boas condições e ambiente de trabalho que proporcionaram. Gostaria de destacar em especial o Luís Sarmento, com o qual tive o prazer de trabalhar, pelas inúmeras conversas e trocas de ideias, pela sua disponibilidade, e pelo seu constante apoio e encorajamento. Gostaria também de agradecer ao Luís Paulo pela sua visão sobre coordenação de equipas e pelo seu interesse no meu trabalho.

Agradeço também aos meus colegas da ESTG do IPVC, pelo apoio e compreensão que demonstraram, especialmente nesta fase final.

Agradeço também a todos os meus amigos e família a quem sucessivamente privei da minha presença ou apoio, em especial aos meus ex-colegas de faculdade, ao João André, às minhas primas, entre muitos outros.

Quero agradecer em especial aos meus país pelo constante apoio que me deram a todos os níveis e ao meu irmão pelo seu encorajamento e amizade. Quero também agradecer em particular à D. Soledade pelo apoio fundamental e incondicional que me prestou, e pelo carinho que me demonstrou. A todos eles, muito obrigado.

Por fim, quero agradecer à minha namorada, pela sua paciência e compreensão, pelo seu constante apoio e encorajamento, por tudo e mais alguma coisa. As páginas deste documento não seriam suficientes para descrever tudo o que me deste e tudo aquilo de que te privei no decorrer deste mestrado.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In this thesis we present a model for coordinating a Team of computational Agents that performs in a Forest Firefighting environment. Agents play the role of firefighters that have to combat fire in an organised way in order to control it. With the proposed model it is possible to define Firefighting Tactics similar (although simplified) to the ones that are used by real firefighting teams. Additionally, it is possible to experiment new tactics or variations of commonly used tactics. For accomplishing this, the model provides a flexible structure that enables to define different approaches of the team to the fire. For developing the coordination model we were inspired by existing models used for team coordination in other domains (e.g. Robotic Soccer), and on specific knowledge about the Forest Firefighting domain.

The solution we propose also tackles other problems such as, Local Coordination and Global Situation Awareness. Local Coordination is concerned with coordinating agents that share a common task and are positioned near each other. By coordinating their actions, agents are able to execute their local tasks more effectively than if they act without having in consideration the agents located near them. Overall Situation Awareness tries to create an overall perspective of the current situation by merging the information that every agent has about the environment. This mechanism is fundamental for coordinating the entire team because the environment is partially-accessible, which makes difficult for agents to have an overall evaluation of the scenario using their individual perception only.

1.2 Motivation

Forest fires are an everyday problem of society. Considering Portugal only, in the year of 2005, forest fires burned more than 325 thousand hectares of forest, in a total of more than 35 thousand occurrences [1]. This problem is shared by several other countries, although it particularly affects Portugal, which is not being able to deal with it (figure 1.1).

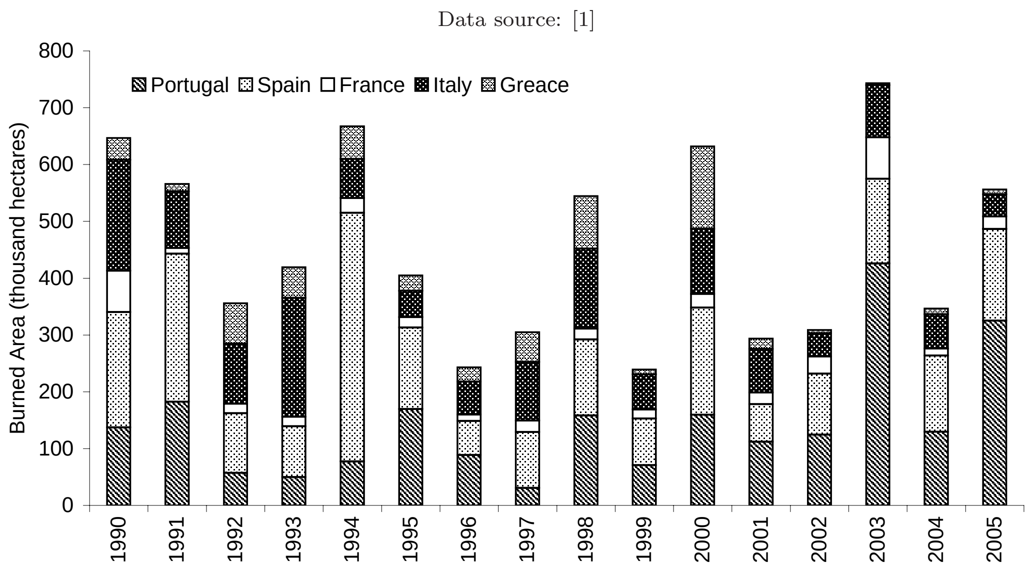


Figure 1.1: Burned areas in forest fires on the South European countries

In spite of the seriousness and challenges of this problem, there is not much work in this domain in the area of Artificial Intelligence. In fact, we only found one work [2, 3] that uses a Multi-Agent System (MAS) to tackle the problem of forest firefighting. The main research involving team coordination in spatial domains is centred around the Robotic Soccer domain on the RoboCup competition [4]. This domain offers several interesting challenges for the MAS area but has some properties specific of the domain that simplify the spatial coordination problem. One of the main limitations is the size of the area under consideration that is fixed (the soccer field has fixed dimensions). On the forest firefighting domain the fire size and shape may vary a lot, which increases the difficulty of positioning the agents and makes many of the models proposed for the Robotic Soccer domain difficult to adapt to firefighting scenarios. More recently, RoboCup started a new competition in the domain of large scale disasters, the RoboCupRescue [5]. With this competition, RoboCup tries to overcome some of the limitations of the robotic soccer domain. In RoboCupRescue the problem of firefighting is present but in an urban context. In the generated environment there are several fires burning simultaneously and the focus of firefighting is centred on target selection and target distribution by firefighters. However, urban fires have a very different behaviour than forest fires, and consequently the tactics that are used are very different too. Additionally, RoboCupRescue tackles the problem of firefighting at a higher-level of abstraction. For a firefighter to attack a fire it only has to communicate to the simulator that he wants to attack that fire [6]. The way that fire is attacked is not taken in consideration, which is an important issue in forest firefighting domain.

With this thesis we intend to contribute to the domain of spatial coordination by proposing a model for coordinating a team of agents. The model we propose is especially designed for the forest firefighting domain but may be used in other domains that require spatial coordination. Furthermore, by implementing the coordination model in the forest firefighting domain, we expect to produce a tool where it will be possible to experiment tactics for fighting forest fires in different scenarios. This would enable to test which tactics results best according to the scenario. However, we realise that this is a long-term goal. The forest fire simulator we are using still needs some improvements, and in this thesis we only consider the problem of coordinating relatively small teams (with no more than twenty agents). Therefore, producing a tool for real tactic experimentation will be left for future work.

1.3 Context

This thesis continues the work of Sarmento and Oliveira [7, 8, 9, 10], which developed an Emotion-Based Agent that act in a Forest Firefighting environment. The previously developed agents act alone and are not able of cooperating with each other. Therefore, in spite of having an intelligent behaviour, a single agent cannot extinguish fires by itself. Nevertheless, the developed agents have an adaptive behaviour produced by their Emotional Mechanisms. These mechanisms try to adapt the information processing capabilities and the operating modes of agents in order to improve their individual performance. These mechanisms are classified as Emotional because they are inspired in Human Emotions and in their influence on the way that humans think and act. Another effect of using Emotional Mechanisms is that the agents behaviour is more similar to the behaviour of human firefighters. This enables to experiment firefighting Tactics with agents that act like humans, which makes simulations more realistic. Yet, we should emphasise that the work presented on this thesis is concerned with the coordination of teams of agents in general, and thereby it does not cover in particular the topic of Emotion-Based Agents coordination.

1.4 Thesis Structure

This thesis is outlined as follows:

Chapter 2 (*Firefighting and MAS*) presents the basics of forest firefighting and identifies the main problems and requirements related to team coordination in the forest firefighting domain.

Chapter 3 (*Team Coordination in MAS*) describes the most relevant work concerning team coordination in MAS that may be adapted to the forest firefighting domain.

Chapter 4 (*Simulation Environment Description*) presents and analysis the platform used for simulating a forest fire environment (Pyrosim) where the proposed coordination module will be tested.

Chapter 5 (*Proposed Model for Team Coordination*) proposes a model for coordinating a team of agents in a spatial domain. In this chapter we will instantiate the proposed model for the firefighting domain.

Chapter 6 (*The Team Coordination Model in Action*) details how we adapted the proposed coordination model to coordinate a team of agents acting on the Pyrosim environment. In this chapter we will give special attention to the problem of limited perception. The chapter ends with a demonstration of a firefighting team using the proposed coordination model acting in the Pyrosim environment.

Chapter 7 (*Experimental Results*) presents the tests' results of using a team implementing the proposed coordination model in Pyrosim. In this chapter, several Firefighting Tactics are experimented in two different scenarios.

Chapter 8 (*Conclusions*) closes this thesis with a summary and discussion of the presented research topics, concluding with some suggestions for future work.

Chapter 2

Firefighting and MAS

In this chapter, we will present the basics of forest firefighting and we will identify the main problems and requirements related to team coordination in the forest firefighting domain. We will start by describing how firefighting teams are organised, and then, we will analyse the behaviour of forest fires, and the factors that affect that behaviour. After this, we will focus on the main techniques that are used by firefighting teams to attack fires. In the end of the chapter, we will analyse the problems of coordinating a team of firefighters and we will define the requirements that a coordination model should satisfy to be implemented in this context.

2.1 Introduction to Firefighting

This section is mainly based in an interview with the Oporto Firefighting Coordination Centre, and in a document produced under two projects supported by European Union and the German Agency for Technical Cooperation [11]. This document proposes a broad framework within which firefighter training can be developed and implemented, and thus provides trusted information about forest fires and forest firefighting. The fire terminology that we will use is also based on this document.

2.1.1 Organization and Command in Firefighting

> "The most dangerous and least efficient ways to fight a fire is for everyone to work by himself or in small groups."
>
> *Marc Nicolas and Grant Beebe*[11]

Firefighting coordination is centralised: one man, the *fire boss*, takes charge of all the people working on a fire. The fire boss has to plan strategy and tactics, and to assign tasks to firefighters in order to control fire and to assure the team safety.

The basic unit for firefighting is a *crew* of 5 to 20 firefighters. For small fires, only a crew is needed and the *crew boss* can also be the *fire boss*. When a crew is not enough, there is an hierarchy of command where the fire boss communicates with the crew's bosses, which then assign tasks to its crew members. In this thesis we will only cover the problem of coordinating a single crew. Therefore, when we refer to fire boss or crew boss we are addressing to the same entity.

2.1.2 Typical Forest Fires

Forest fires may behave very differently depending on three factors [11]: (i) the fuel that is burning, (ii) the weather and (iii) the terrain topography. Table 2.1 describes some of the effects that these factors may have in the fire evolution.

Table 2.1: Main factors that influence the behaviour of a forest fire

Factor	*Property*	*Influence on Fire*
Fuel	Type	Different types of fuel produce different fire behaviours. For instance, trees burn more slowly than grass but produce more heat.
	Condition	Dead plants burns faster and produces more heat than live ones.
	Moisture	Moisture in plants makes them harder to burn. Living plants have more moisture than dead ones.
	Arrangement	Spread vegetation burn slower, however standing trees burn hotter than when they are lying because they have more air available.
Weather	Temperature	The hotter the weather is, the easier it is for a fire to grow and the harder it is to control.
	Wind	Wind helps fire propagate, provides more air to the fire and helps to dry fuels.
	Humidity	Rain and high humidity increase moisture in fuels, which makes them harder to burn.
Topography	Slopes	Rising slopes make fire to move more quickly because cells ahead the fire are warmed before the fire gets there.
	Barriers	Small rivers, rocks, patches, bear earth, roads and other natural or man-made barriers slow fire and may be used to control it.

Although there are a lot of variables that condition fire evolution, forest fires usually have a typical anatomy. In a forest fire, usually it is possible to identify the following parts [11]:

- the *Head*: also denominated as the front of the fire, it is the most rapidly moving portion of the fire;
- the *Tail*: also denominated as the back end of the fire, it is the slowest moving portion of the fire;
- the *Flanks*: are the sides of the fire.

Figure 2.1 illustrates a typical fire with the *Head*, *Tail* and *Flanks* identified. Usually, the fire spreads in the wind direction or up-slope. Therefore, knowing the wind properties (direction and velocity) and the terrain topography is essential to determine where the head and tail of the fire are located.

Source: adapted from [11]

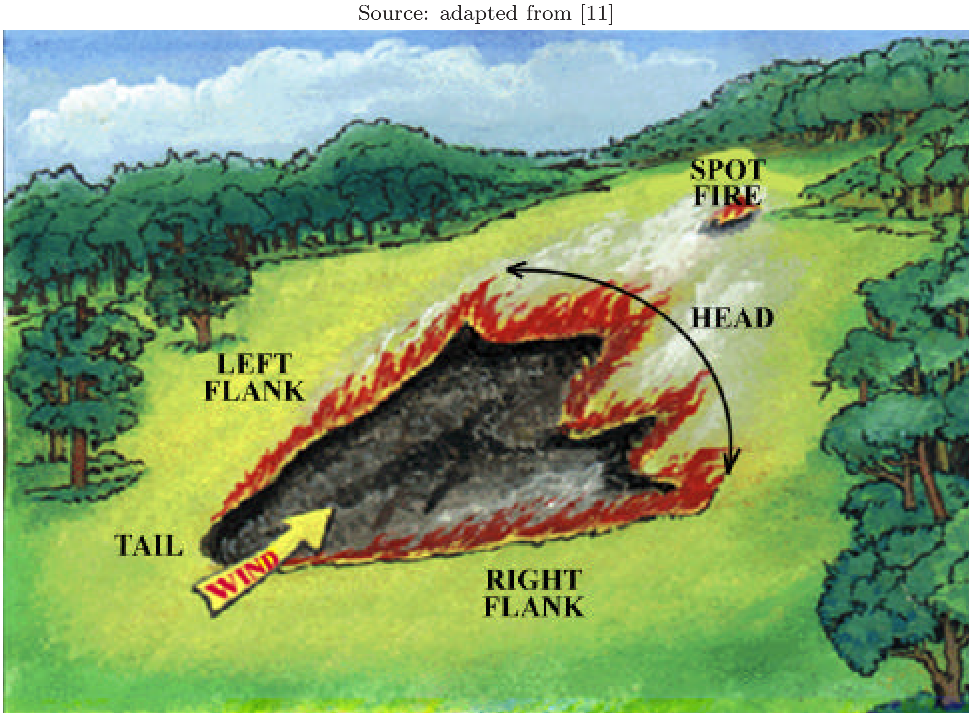


Figure 2.1: Forest Fire Anatomy

2.1.3 Fire Control Techniques

Mainly, there are two kinds of approaches that a firefighting team may perform to control a fire[11, 12]: (i) *Direct Attacks* and (ii) *Indirect Attacks*. During a fire combat, the firefighting crew may use both approaches or only one of them. In the next sections, we will describe in detail direct and indirect Attacks.

Direct Attacks

Direct Attacks involve fighting the fire directly in the flames, using water or manual tools, like shovels, to swat the flames. Direct attacks are usually performed at the edges of the fire tail, where the temperature and danger are lower. The attack then continues moving up to the head of the fire (see figure 2.2).

Source: adapted from [11]

Figure 2.2: A direct attack in both edges of the tail of the fire

However, if the fire is not too strong, it may be possible to attack the fire starting in its head and going down to the tail. When a successful attack is performed in the head, the fire burns a smaller area because firefighters are controlling the part of the fire that spreads faster. Obviously, attacking the head is much more dangerous than attacking the tail, because the escape points of firefighters are located in unburned areas. When attacking the tail, firefighters may run away to the already burned area, where they would be much safer. Because of this same reason, direct attacks performed in the fire tail, are usually safer than indirect attacks.

Indirect Attacks

Indirect Attacks are about fighting fire at distance and are especially used when the fire is too intense for firefighters to approach it. The most common technique is to build a *fireline*. Firelines are built by dogging the ground to remove all the vegetation in front of the fire to starve it out of fuel (see figure 2.3). Firelines may be built starting on a safe point at the tail of a fire, like a road, a natural barrier, or a place where the fire is already out. In case of danger, the crew can always escape towards the safe point. It is also possible to start building the fireline at the front of the fire, although this method is more dangerous and frequently it is impracticable because of the high temperatures. However, if well applied, this technique is more advantageous because prevents fires from growing larger.

Source: adapted from [11]

Figure 2.3: A fireline construction for an indirect attack

In combination with firelines, firefighters usually use another technique known as *Back Firing*. Back firing consists in burning vegetation before the fire gets to it. However, this document will not describe this technique. It is an advanced technique that should only be used after dominating the previous discussed techniques.

Another kind of indirect attack is to build a *wet-line*. Wet-lines have the same function than firelines but demand that firefighters have access to sufficient water to build them.

2.2 Coordination Problems in Forest Firefighting

The forest firefighting domain has a set of challenging problems that are difficult to solve, even when coordinating a single crew. First, there is the problem of the environment to be partially accessible. Usually, firefighters only see part of the fire and must decide what to do with out having access to information about the entire situation. The fire boss must decide which kind of attack to use (direct or indirect), where to position his crew to start the attack, and which part of the fire to attack. Besides the problems of the initial approach, during the firefight the fire boss may need to change the kind of approach, to reposition the crew or even to abort the attack.

In order to execute tactical/strategical decisions, the fire boss must give instructions to his crew. These instructions must specify the task to carry out, while giving some autonomy to firefighters. Fire bosses cannot send instructions every second to every crew member with low level information. Instructions should use higher level information (e.g. attack area, instead of, attack precise location) to reduce message passing frequency, letting the fire boss the time to do other relevant tasks. Furthermore, firefighters win additional autonomy that enables them to adapt better to the particularities of the situation.

The problems we have discussed so far in this section are related to tactical decisions that have influence in the global team behaviour. However, there is also the need for firefighters to coordinate efforts between themselves locally. When a group of firefighters has a given task to execute they must cooperate in order to execute that task effectively. For instance, if a group of firefighters has to attack a given area, they should coordinate themselves to attack the same location or very near locations inside that area. Otherwise, their efforts will be dispersed and the results will be far worst.

2.3 Coordination Requirements for Forest Firefighting

Teamwork is a subarea of MAS coordination defined by a cooperative effort of every team member to accomplish a set of goals shared by the all team [13]. This is the kind of coordination we need to coordinate a team of firefighters that have a common goal, which is putting out fire. However, not every team coordination model is suitable for us. There are some requirements that must be satisfied related to the domain we are working on. One of the most important is that coordination model must be suitable for *spatial coordination*. Spatial coordination is concerned with managing the agent locations in space to accomplish space dependent tasks. This requirement is very relevant because firefighting tactics demand firefighters to be located in strategic positions. For instance, if the tactic is to attack the fire directly from behind, firefighters must position themselves behind the fire tail before starting the attack. Another requirement is that the coordination of the firefighting crew should be done by a single agent, the fire boss [11]. This is how real firefighting teams work, after years of experience. However, this is not a mandatory requirement, but centralising coordination decision-making should enable to accomplish better performances in a partially-accessible environment where a global knowledge of the environment is important to determinate the team positioning and approach to the fire. Finally, the coordination model should enable to implement different firefighting tactics. It should be possible to position firefighters in different strategic positions and performing different kinds of attacks.

In addition to coordinating the entire team, agents also have to coordinate themselves locally. Usually, groups of firefighters share the same task or subgoal and must cooperate efforts in order to carry out the task efficiently.

Table 2.2: Firefighters responsibilities when using a single crew

Fire Boss	*Rest of the crew*
Be aware of the global situation	Report local situation to the boss
Define team positing and approach to the fire	Cooperate locally with teammates
Assign tasks to firefighters	Execute the assigned tasks (if possible)
Make tactical decisions	Make local decisions

In table 2.2 we have summarised the responsibilities of the fire boss and the rest of the firefighting crew.

2.4 Summary and Conclusions

In this chapter, we started by presenting general forest firefighting information to introduce the reader to the firefighting domain. First, we covered team organisation issues and concluded that forest firefighting teams are organised in crews of 5 to 20 elements, which are commanded by a crew boss. In case of a crew is not enough to control a fire, it is possible to use several crews coordinated by a fire boss. In this case fire bosses give orders to crew bosses, creating a hierarchy of command. However, in this thesis we will only address the problem of coordinating a single crew.

We also described forest fires properties and detailed the main factors that influence fire progression and behaviour. We concluded that wind and terrain geometry usually define the fire main propagation direction, and consequently the fire anatomy (head, tail and flanks).

In the end of the firefighting introduction section, we focused on the kind of attacks that a crew (or part of it) may use to control a fire. We have identified two main types of attacks: (i) direct attacks, which are used when it is possible to attack the fire directly with fire or hand tools, and (ii) indirect attacks, which fight fire at distance creating barriers (a fireline or a wet-line) to stop fire from spreading.

At the end of the chapter, we identified the problems we are trying to solve trough the work reported in this thesis, which are related with the coordination of a single firefighting crew. The main problems are (i) a partially-accessible environment that makes global situation understanding and tactical decision difficult, (ii) defining team positioning in the terrain and approach to the fire, (iii) defining tasks for the crew, which cannot be very detailed to give local autonomy to firefighters, and (iv) local coordination between firefighters when sharing the same task. Based on these problems, we introduced some requirements that the coordination model should satisfy. These requirements involve a centralised spatial coordination, which enables for an agent (the fire boss) to be aware of the global situation and decide team positioning and approach mode. Additionally, it should be possible to represent different positioning and approaches in a structured way, in order for the fire boss to decide which alternative tactic to choose, or to experiment which tactics work best for a given situation.

Chapter 3

Team Coordination in MAS

In this chapter, we will expose the most relevant work concerning team coordination using MAS that may be adapted to the forest firefighting domain. Some of the properties of the models that we will describe here will be used in our team coordination model.

In the previous chapter, we have identified the requirements that a coordination model should have for this domain. One of the main requirements is that the coordination model should be able to handle *spatial coordination* with the possibility of creating different kinds of approaches (different positioning in space, and different team behaviour). We are also interested in hierarchical coordination models that enable to implement a chain of command like in the real firefighting teams. Additionally, firefighting also demands that firefighters coordinate their actions locally and therefore this chapter also covers decentralised coordination models. Finally, in the end of the chapter, we will describe the related work in the team coordination area applied to the forest firefighting domain.

3.1 Team Coordination Models

3.1.1 Coordination by a priori Knowledge

Coordination by *a priori* knowledge enhances agents with a common knowledge base that is used to define the agent individual goals based on the current situation [13]. This knowledge may be sent to the agents at the beginning of an episode (e.g. at the beginning of a simulation), or it may be built-in inside the agent.

This coordination technique was successfully used in the simulated soccer domain, first by Stone and Veloso [14, 15] with the *Locker Room Agreement* and then by Reis and Lau [13, 16, 17] with the *Strategic Coordination* model. These works are of particular interest for us because firefighting and simulated soccer share some common characteristics. Examples of these characteristics are: (i) both require spatial coordination, (ii) the team may use different kinds of approaches to the adversary, (iii) agents may play different roles in the team, and (iv) the environment is partially-accessible. In the next subsections, we will describe these two models.

Locker Room Agreement

The *Locker Room Agreement* [14, 15] is a high-level coordination mechanism especially useful in domains with restricted communication. This mechanism is based in a flexible definition of the team structure that is shared by every agent. This structure is composed by the following parts:

- *Roles*: every agent has a different role that specifies the agent behaviour and location in the field. Agents are homogenous and may play any role. When special conditions are satisfied, agents may switch roles between them.
- *Formations*: a formation is a set of roles (one role per agent) that are assigned to agents when a given activation condition is triggered.
- *Units*: roles are also grouped in small units. Units are used to handle local situations without affecting the entire team. Every unit has a leader agent that coordinates the unit in its interventions.
- *Set-Plays*: plans that should be carried out by the team (or part of it) in special situations (e.g. a free-kick). Agents play temporary roles an then return to their original roles.

The locker room agreement specifies an initial formation, it maps the formation's roles to agents, and it defines conditions for formation switching. When the simulation begins, agents know the current formation and every teammate's role. However, during the simulation they may lost track of the current formation. To try synchronizing the team, agents periodically broadcast their role to the other agents.

The locker room agreement may be used in environments where communication is limited or impossible because agents share a predefined team structure that informs them which formation to use based on the current situation. In such environments, agents may not be completely synchronised but they achieve reasonable coordination levels. A prove of that, is that a team implementing this coordination model was able to win two world championships in the robotic soccer simulation league. In environments with no communication restrictions the team is able to stay completely synchronised.

The main limitations of this model are related to its design. First, the number of roles must be equal to the number of agents. This feature does not enable to define a role that is shared by several agents. Besides this, the number of elements in a team must be predefined. In the soccer domain, these drawbacks may not be critical because inside a team there is a considerable variety of roles, and also because a soccer team has a fixed number of elements. Unfortunately, this does not happen in the firefighting domain. The team size may vary a lot, even if we are considering a single crew, which may have between 5 to 20 elements. Additionally, inside a firefighting team many agents share the same role. It is common to happen that in a firefighting crew there is only one ore two kinds of roles. In spite of these disadvantages, the locker room agreement has several interesting ideas that may be useful in the firefighting domain, such as the creation of roles and formations.

Strategic Coordination

Strategic Coordination [13, 16, 17] also uses a flexible team structure shared by every agent, like in the locker room agreement, but more complete and adapted to the robotic soccer reality. This model was especially designed for this domain but may be adapted to other domains, with or without spatial coordination.

In the strategic coordination model, the team structure is defined by the following parts:

- *Roles*: define different behaviours that an agent may play. An agent may play only one role at a time but is able to play any role.

- Formations: define for every agent a position in the field and a role to play. Every position as an importance associated that defines the necessity for that position to be occupied (e.g. the goal keeper position usually has a high importance to force the goal keeper to be near the goal).

- *Flexible Plans*: implement the same idea that set-plays in the locker room agreement, and are composed by activation conditions, termination conditions and a temporal sequence defining the agents roles and positions evolution during the plan execution. This plans are used in special situations like free-kicks.

- *Tactics*: define the team global behaviour trough tactical parameters, a set of formations with activation conditions, and a set of flexible plans. This structure enables, for instance, a team to use a different formation when in possession of the ball than when it is defending.

- *Strategy*: a set of tactics with tactical activation conditions. The team starts with an initial tactic but in special situations the tactic may be changed. For instance, when the team is loosing and the game is almost finishing the team may switch to a more aggressive tactic.

This complex team structure definition enables to define sophisticated team behaviours in environments with limited communication. To use this model efficiently, high-level domain-specific information must be generated that enable agents to easily identify the situations when they should change tactic or formation. The main disadvantageous of this model are related to the differences between robotic soccer and the forest firefighting domains. The first disadvantage is that the strategic coordination model uses a decentralised coordination approach. There are some properties of the robotic soccer domain that simplify some problems of spatial coordination and enable to use decentralised coordination. Examples of such properties are the size of the robotic soccer field that is always the same, and the fact that agents may decide using local information only, when inside of their actuation area. In the firefighting domain the scenario may vary a lot with the terrain properties, and the fire itself may have different shapes, behaviours and sizes. These properties difficult the definition of formations and oblige agents to have a global knowledge of the situation to take proper coordination decisions. Centralised coordination may not be the only solution for these problems, but it solves them. Another disadvantage of the strategic coordination model is that it defines over-flexible team structures that enable to create strategies more complex than the ones that usually are used in the forest firefighting domain. This flexibility is needed in the robotic soccer domain because of the dynamics of the environment. The game is always changing between attack and defence situations that obliges the team to adapt to the current situation. Forest fires have a different kind of dynamics that does not require the team to constantly change formation.

The results of strategic coordination in the simulation league of robotic soccer include two European Championship and one World Championship. The strategic coordination model uses more coordination mechanisms besides the ones described in this section, although the mechanism we have analysed is the one that interest us the most.

3.1.2 Perception-Based Coordination

Perception-based coordination is concerned with using sensory information to help coordinating a team of agents. We have found two main approaches to perception-based coordination. The first is known as *Mutual Modelling* and uses perception information about the other agents to help coordinate their actions. The second is based on perceiving the changes that other agents produce in the environment. We will now analyse these two models separately.

Mutual Modelling

In *Coordination by Mutual Modeling*, agents have a model of the other agents. With this model, an agent can predict what another agent will do in a given situation. Mutual Modeling was first suggested by Genesereth, Ginsberg and Rosenschein in their work about Cooperation without Communication [18]. In their work, agents had information about the preferences of the other agents, which enabled them to estimate the actions of the other agents.

Les Gasser also applied this principle to his MAS but using a more complex model of the other agents. He denominated the knowledge about other agents has *acquaintance knowledge* [19]. This knowledge included information about the role, skills and goals of the other agents, and also the expected plans that agents will carry out to achieve their goals.

In the firefighting domain, mutual modelling could be particularly useful to achieve local coordination without using communication. Usually, groups of firefighters share the same task and mutual modelling may be used to help coordinating their efforts while letting the communication channels free.

Using Perception for Implicit Communication

When using perception for implicit communication, agents usually leave special marks in the environment that may be perceived by other agents. Balch and Arkin [20] were able to coordinate a set of agents that had to cover an entire area using this technique. In their work, agents are capable of leaving a trace in the environment marking the places where they have been, which enables other agents to detect zones that were already covered. The authors claim that in this particular task, adding communication did not improve the team performance.

Manei and Zambonelli use changes in the environment to produce different kinds of team formations [21]. Every agent generates a *co-field*, which is a special kind of signal that is strong in the agent location and becomes weaker as we move way from the agent. These co-fields may be used to attract or repel other agents. A given agent may attract a group of agents and repel other group of agents. Using this mechanism it is possible to form agent formations in spatial domains.

The problem with using perception for implicit communication, is that the environment is the mean of communication and therefore it must be enhanced with features that enables this communication. For instance, in Balch and Arkin work the environment has a matrix used to mark the agents tracks, and in Manei and Zambonelli the environment is responsible for generating the co-fields. These coordination techniques are dependent of the platform, which makes of them less portable to other domains, and less representative of the real world.

3.1.3 Coordination by Partial Hierarchical Control

So far, the models we have analysed are concerned with organising a team of autonomous agents in a decentralised way. No agent is in charge of the all team, and every agent makes tactical/strategical decisions. In *Partial Hierarchical Control*, like in many models of human society, there is an hierarchy of command. Agents placed higher in the hierarchy have more authority than agents placed in lower positions. Additionally, the more high a agent is in the hierarchy the more high-level decisions it makes. This is precisely what happens in the firefighting domain, as we have seen in section 2.1.1. A *fire boss* makes strategic decisions and send directions to *crew bosses*, which send commands to their crew members.

The problem of hierarchical coordination, is that the commands received by agents may not be taken as absolute. Although this models suggests the existence of agents which are commanded by others, all of the agents are still autonomous and consequently have decision power. For instance, during a firefight a crew boss may send an order to an agent to attack a given position. The agent may fulfil that order, or it may refuse to execute if it feels that its safety is put in danger. This happens because the order may put at stake the agent individual goals (in this case, the survival goal) and the team goals (a team with one agent less will have more difficulties to fight the fire).

Among several relevant works in this area we will give special attention to the work of Grosof on conflict handling when receiving advices or instructions [22]. In his work, Grosof suggests four criterions to handle conflicts:

- *Specificity*: instructions with more specific information take precedence over those which covers a more general case. For instance, the instruction *go to location X inside of area Y* is more relevant than the instruction *go to area Y*.
- *Freshness*: more recent instructions have higher priority. If an agent receives two or more instructions from the same source, it should use the newest. For instance, if an agent receive a retreat order after an attack order, it should execute the retreat order.
- *Authority*: instructions from agents with higher hierarchy have higher priority. This is a rule very used in our society built on giving more power to the agents which have a better understanding of the global situation.
- *Reliability*: instructions or advices from more trusted sources should take precedence over those which come from less trusted sources. For instance, an agent who wants to know about the situation in a given position, should trust in the agent that is more close to that position.

These criterions help agents to decide which instructions to execute, and which information to believe the most, in case of conflict between information coming from different sources. This may be especially useful in the forest firefighting domain where agents may receive several instructions in a short period, and where agents may send incomplete or inaccurate information to their superiors about their visual perception.

Coordination by Partial Hierarchical Control was used with success in the RoboCupRescue domain. A team (DAMAS-Rescue) using this coordination method was able to finish in second place on the world championship of the 2004 RoboCup competition in the Rescue simulation league. For every agent group (Firefighters, Ambulances and Police Forces) the DAMAS-Rescue team used a Leader agent to assign high-level tasks to the other agents [23, 24]. Leaders have a better understanding of the overall situation, which makes task assignment more efficient by placing agents in strategically important positions.

3.2 Related Work in the Forest Firefighting Domain

There is not much work in MAS research applied to the forest firefighting domain. The most relevant work we have found is from Wiering *et al.* that uses machine learning to try solving some aspects of agent coordination for the construction of firelines [2, 3]. The authors are especially concerned with the problem of fireline construction by heavy machinery (bulldozers) to prevent large fires from spreading [2]. Agents are bulldozers, and the goal is to determine where the fireline must be constructed in order to prevent fire from spreading and to sacrifice the minimum forest area as possible. In this work, coordination is not a big issue because every agent receives a plan at the beginning of the simulation specifying what part of the fireline the agent must dig, and after that, the agent just follows that plan. This plan is composed by a positioning location that defines the starting point for building the fireline, and a set of subgoals specifying locations for completing the part of the fireline that was assigned to the agent (figure 3.1). Communication only exists at the beginning of the simulation for task assignment.

The problems that Wiering *et al.* focus in their work are: (i) fireline sugoals calculation, (ii) task assignment for distributing subgoals by the agents, and (iii) local subgoal optimization for agents to adjust the subgoal location to the current situation. The first two topics are related to the planning phase, and the last one to the agent real-time adaptation to the situation it is facing. Wiering *et al.* used machine learning techniques to approach all of these topics. The authors claim that they were able to improve the results of using a fixed policy in a scenario using three agents [2]. The work of Wiering is very interesting and has resembles with our work. However, we are aiming to control a crew of firefighters to test first approach techniques and early prevent fire spreading, and on the other hand, Wiering search approaches to largest fires with smaller teams of heavy machinery. Additionally, Wiering centre his work around machine learning, and we focus on

Source: adapted from [2]

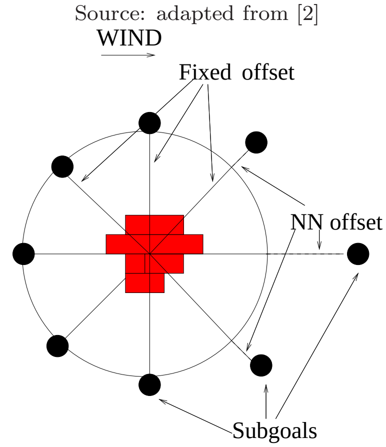


Figure 3.1: Subgoals to be assigned to bulldozers

team coordination techniques to attack fire, although it is our goal in a near future to try machine learning to help tactical/strategical decision-making.

3.3 Summary and Conclusions

In this chapter we have analysed several MAS coordination models that may be applied to the forest firefighting domain. We started by introducing coordination using predefined knowledge that is used to define team behaviours and spatial positioning in domains with restricted communication. These models are very inspiring in terms of tactic and strategy definition. However, they use a fully decentralised model that makes sense in the experimented domain (robotic soccer), but may not be the ideal approach for the forest firefighting domain.

After that, we focused on coordination using perception and analysed two different approaches: *mutual modelling* and *implicit coordination*. Mutual modelling proposes that agents should have models of the other agents to predict their actions, and use this information to coordinate efforts. This approach by itself is not enough for coordinating the entire team because of restrictions in agents perception (agents may have many teammates outside of their visual range), and because of the lack of a structure defining the team approach mode. However, this model may be employed for local coordination when a group of agents share the same task or subgoal. The major benefit of using this technique is that no communication is needed. The second technique that we have analysed for team coordination using perception, suggests using the changes in the environment produced by agents to realise what those agents were doing. This technique may be used in domains where knowing what the rest of the team did is enough (e.g. area covering problems), but in the firefighting domain we are more interested in knowing what agents are currently doing and what will they do next.

Other coordination techniques using perception for implicit coordination need special implementations on the environment for producing information that may be perceived by agents but are not perceived by visual sensors (e.g. repulsive and attractive magnetic fields). These techniques place us farther from the real world because firefighters do not have access to this kind of gear. We also believe that team coordination is possible in the firefighting domain by only using communication and visual perception, and therefore we will not use this kind of approach.

Finally, we have analysed a kind of centralised coordination model that also gives some local autonomy to the operational agents, and therefore is denominated by *partial hierarchical control*. This model resembles very much with the command model used by real firefighting crews, and therefore we will use it for coordinating our team of agents. In the context of centralised coordination, we have described a set of criterions for instruction selection in case of conflict proposed by Grosof.

At the end of this chapter, we have analysed the most relevant related work concerning team coordination in the forest firefighting domain. We presented the work of Wiering *et al.*, which uses machine learning techniques for task optimization and task distribution. The authors presented an interesting approach for generating a fireline, and to distribute tasks to bulldozers in order for every bulldozer to construct a segment of the fireline. However, the main focus of that work was testing machine learning techniques and not coordination, which is limited to an initial task distribution.

Chapter 4

Simulation Environment Description

This chapter presents the platform used for simulating a forest fire environment where the proposed coordination module will be tested. In the first sections, we will describe the simulated environment, and compare it with the real environment as we described it in chapter 2. After that, we will give special attention to the interaction capabilities between the agent and the simulated environment, describing perceptions that agents receive and actions they are able to perform. At this point, the main features of the simulator were presented and it is possible to analyse the kind of fire attacks we are able to simulate. Finally, we will analyse the environment complexity and we will present our conclusions about the adequacy of the Pyrosim simulator to the problem of team coordination in the forest firefighting domain.

4.1 Environment Description

Pyrosim [7] is a real-time simulator developed at LIACC* research centre for experimenting firefighting techniques in a forest fire scenario. This tool was especially developed for testing agents in complex environments. It simulates a three dimensional environment with which user-programmed agents can interact under the role of firefighters. Obviously, agents main goal is to extinguish the fire minimizing the burning area (while keeping themselves alive). In figure 4.1 there is a snapshot of the environment generated by Pyrosim in a 3D visualiser, the PyroViz.

*Laboratótio de Inteligência Artificial e Ciências dos Computadores / Artificial Intelligence and Computer Science Laboratory – Universidade do Porto

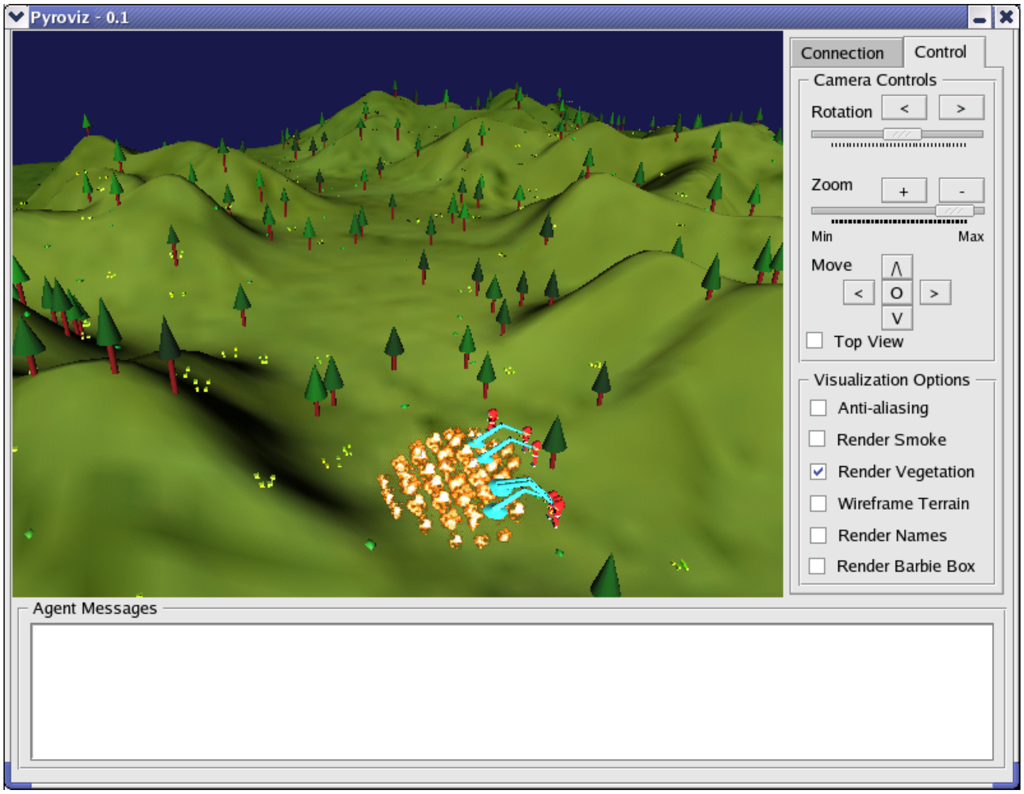


Figure 4.1: The Pyroviz simulation visualiser for the Pyrosim simulator

Pyrosim tries to create a simulation with the complexity of the real world. Agents are presented to an environment where factors like ground vegetation, terrain geography and wind have effect on fire propagation and agent locomotion. Moreover, agents perception is limited and loses detail with the extension of the perceived area (e.g. close range perception is more accurate than medium range perception).

Pyrosim world is described by a two dimensional grid of cells. By default, the world has an extension of approximately 2 per 2 kilometres divided in 16 per 16 metres cells, which result in a 128 per 128 cell grid. Each cell may be described by the following attributes (that remain constant along the cell):

- *Vegetation Distribution*: a terrain cell may have four different kinds of vegetation (underground vegetation, grass, bushes and trees) in different proportions; each vegetation type has different combustion properties (e.g. burning rate, ignition temperature);
- *Geography*: defines the slope of the terrain;
- *Fire Combustion*: indicates the combustion intensity in the cell (zero if the cell is not burning);

- *Burned Area*: percentage of combustible matter already consumed by the fire.

The cell's vegetation and geography are defined in the beginning of the simulation and remain constant. On the other hand, fire combustion and burned area are dynamic attributes about the fire evolution in the cell.

In section 2.1.2 we have presented the main environment factors that influence fire behaviour (table 2.1). Although an ideal simulator would implement every one of those factors, that does not happen in the current version of Pyrosim. Table 4.1 shows which factors are implemented and which are not.

Table 4.1: Factors supported by Pyrosim that influence the fire behaviour

Factor	*Property*	*Supported*	*State of the Implementation*
Fuel	Type	✓	Pyrosim supports 4 different types of fuel with different combustion properties.
	Condition	✗	Only live vegetation is supported.
	Moisture	✓	Every cell as an initial quantity of moisture that decreases with the temperature produced by fire. Firefighters may increase the moisture in the cell using their water jets.
	Arrangement	✗	In Pyrosim, vegetation is always arranged in the same way.
Weather	Temperature	✓	Pyrosim enables to define the ambient temperature.
	Wind	✓	It is possible to define the wind speed and wind direction. However, in the current version, these properties remain constant along all the terrain.
	Humidity	✗	In Pyrosim, there is no rain, or humidity in the ambient.
Topography	Slopes	✓	It is possible to generate random terrains, varying from flat terrains to mountains. Slopes have influence in the fire behaviour and firefighters locomotion.
	Barriers	✗	There is no kind of barriers in Pyrosim.

4.1.1 Agents Perception

Agents have conditioned access to terrain information because of their limited perception capabilities. Every agent percepts two visual maps for each one of the terrain's properties defined above: a *Close-Range Map* and a *Medium-Range Map*. Close-Range Maps are 3x3 matrixes with information about the cell where the firefighter is located and the cells surrounding him (figure 4.2). This corresponds to the maximum detail available because one map cell matches a single terrain cell. On the other hand, Medium-Range Maps cover a much larger area and therefore provide much less detail. Medium-Range Maps are represented by 7x7 matrixes that correspond to a 21x21 world area (figure 4.3). This means that a medium range cell enclose the equivalent area to a Close-Range Map. The value of a medium range cell is equivalent to the arithmetic average of the underlying terrain cells. Close and Medium-Range Maps are always relative to the firefighter, which is placed in the map centre.

Source: adapted from [7]

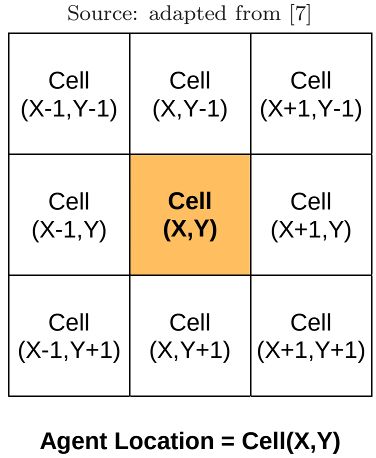


Figure 4.2: A Close-Range Map

Pyrosim simulates the phenomenon of visual occlusion, which happens when the firefighter cannot see a particular area because of the terrain geography. This phenomena is demonstrated in figure 4.4, where the firefighter located in cell 3 does not receive information about cells 1 and 4.

In addition to visual maps, Pyrosim also provides global maps that cover the entire simulation area: one for the terrain geography (named *Pocket Map*) and other for tracking fire evolution (named *Global Fire Map*). The first is sent to firefighters only at the beginning of the simulation and offers low detail information about the terrain geometry. Its aim is to simulate a terrain chart that is usually available to firefighting teams for helping them to avoid difficult areas such as cliffs. The global fire map is a low detail snapshot of the entire area with information about fire intensity. This map is periodically sent to firefighters

Source: adapted from [7]

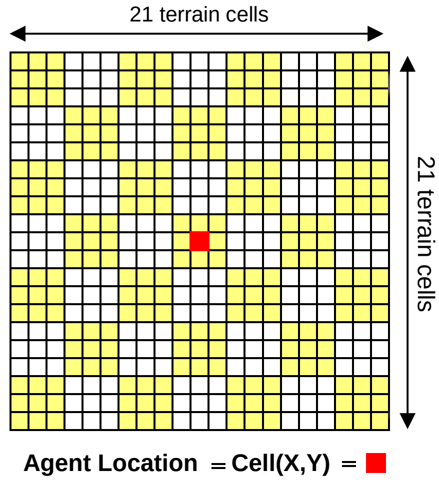


Figure 4.3: A Medium-Range Map

Source: adapted from [7]

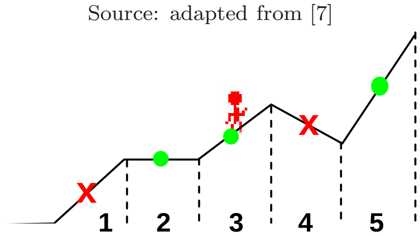


Figure 4.4: Cell Occlusion Phenomenon

with a short frequency and it enables to detect new fires outside of their visual range.

Besides maps, firefighters also perceive other information about the environment, and about themselves. The most relevant perceptions are:

- *Ambient Information*: firefighters feel the air temperature around them, and they have an estimate of the wind speed and direction;

- *Location*: a firefighter knows his location in the terrain and the slope of the terrain where he is standing;

- *Movement*: a firefighter knows his movement direction, speed and acceleration;

- *Energy*: firefighters have limited energy to perform their actions. Firefighters consume energy when walking or running, spending more energy in highly accelerated movements. Furthermore, firefighters lose energy very quickly when exposed to temperatures higher than 50℃, as the result of suffering fire injuries. When firefighters reach low levels of energy, they lose skill in their perception and acting capabilities, simulating the effect of tiredness.

- *Water Hose*: all firefighters have a water hose and a personal water tank. They are able to perceive if the water hose is on, its direction and water flow. They can also inspect their personal water tank to see how much water is available.

- *Teammates*: a firefighter is able to see other firefighters located inside his medium range. In this conditions, a firefighter can see his teammates location, moving direction, moving speed, water hose state (on/off) and water hose target location (if it is on).

4.1.2 Agents Acting Capabilities

Firefighters are able to control their locomotion (acceleration and direction) and their water hose (hose orientation, water jet flow and water jet range). In environments that try to simulate the real world, such as Pyrosim, the result of executing an action may not be exactly the desired. This happens in both locomotion and water hose control actions. In the former case, when a firefighter requests a given acceleration, the obtained result may vary because of the terrain geography. Firefighters can climb hills up to 45 degrees of inclination (or more if they accelerate enough before the hill), experiencing lower acceleration levels in more abrupt terrains. When descending hills if the slope is higher than 45 degrees they lose control on stopping the movement.

As for controlling the water hose, firefighters must also deal with a trade-off between the water jet range and flow (higher distances result in a less powerful water flow). Apart from that, the water hose can be oriented with an angle between 90 degrees to the left of the firefighter and 90 degrees to the right. In the current version of the platform, each firefighter has a personal water tank that does not limit in any way its movements. However, the tank has limited water capacity.

In order to extinguish fire, burning cells must receive sufficient water to suppress the fire. In many cases, especially on cells with intense fire surrounded by other cells in the same conditions, one firefighter may not be enough to extinguish the fire or to control its propagation. In situations like these, teamwork is necessary. To facilitate teamwork, firefighters may send text messages to teammates. Messages can be sent to one teammate in particular or to the all team. Pyrosim leaves the implementation of the message protocol to the agent programmer.

4.1.3 Fire Attacks Supported in Pyrosim

In section 2.1.3, we have introduced the kind of approaches that a firefighting team may experiment to control fire. We will now analyse the Pyrosim simulator to evaluate the possibility of experimenting such approaches.

As we can see in table 4.1, Pyrosim does not support barriers. This is the major drawback from this simulator in terms of tactic experimentation. The environment generated by Pyrosim does not include any kind of barrier, and firefighters are not able to make firelines. However, it is possible to execute indirect attacks by building wet-lines. This is not the most common way for firefighters to operate because water is usually a very limited resource. In this project we will use wet-lines to produce indirect attacks, pretending to be the fairly used firelines. Back firing is not supported too, but this is an advanced technique that would be out of the context of this thesis.

As for direct attacks, firefighters are able to use their water hose to fight fire directly. Using hand tools, like shovels, it is not currently supported in Pyrosim.

In conclusion, although Pyrosim in its current version does not support some important features for experimenting firefighting tactics, it enables to test indirect and direct attacks in unbounded forest-fires.

4.2 Environment Complexity Analysis

Accordingly to [25] environment's classification, the Pyrosim environment is:

- *Partially Accessible*: the agent has limited perception of the world;
- *Non Deterministic*: the outcome of an agent action cannot be determined with certainty;
- *Dynamic*: the world is constantly changing, even when the agent is processing or doing nothing;
- *Continuous*: the agent perceptions and actions are represented by continuous values and cannot be delimited by sets.

This classification demonstrates that Pyrosim is in the class of the most complex environments. In fact, the perception of agents is limited, the outcome of their actions is not guaranteed, the world is constantly changing and the perceptions and actions are (in theory) infinite. In addition, there is another property, not directly covered by this classification, which increases even more the environment complexity: it is a multi-agent environment. Consequently, coordination and cooperation between agents is a mandatory requisite for success.

The Pyrosim environment constantly places agents in situations where decisions (at different levels) must be taken. For example, they must decide the distance to keep from the fire, their position relatively to the fire, which fire spot to attack, the water hose parameters, the paths to reach a desired cell (avoiding hills and fire) and when to escape from a dangerous situation. Furthermore, they must manage their own energy and water reserves. In conclusion, this problem is complex enough even if we are considering controlling a single agent.

For tackling the complexity of the Pyrosim's environment, Sarmento proposed using Emotion-based Agents [7]. The author claims that Emotion-like mechanisms are a requirement for intelligent behaviour for resource-bounded agents operating in complex, dynamical and real-time environments. This is the case of the Pyrosim's environment. The agents developed by Sarmento are capable of adapting their behaviour to the environment in order to achieve better individual performances. However, these agents have no coordination capabilities and therefore are not capable of extinguishing fires. We will use these agents as starting point for developing our own agents, and we will enhance them in both individual and collective skills.

4.3 Summary and Conclusions

In this chapter we have analysed the Pyrosim forest fire simulator. We started by describing the generated environments and then we compared it with real environments. We concluded that not all features that condition fire evolution are present in Pyrosim, although it covers enough features to produce complex environments with properties of the real ones. After that, we described what agents are able to perceive and what actions they are capable to perform. The main issues related to these topics are that agents have access to visual information with different levels of detail. At close range they have maximum detail, and at medium range they can see a much larger area but with far less detail. This characteristic is inspired in reality and in a centralised coordinated team obliges firefighters to have local autonomy to adapt to the local situation, which the coordinator cannot perceive. In addition, there are other simulated phenomena like visual occlusion that difficult perception and adds reality to the simulation. Another important issue related to perception is the capability of firefighters to perceive other firefighters within medium-range. This enables to implement some coordination methodologies based on perception that we discussed in chapter 3.

As for action capabilities, agents are able to perform low level actions (rotate, accelerate) that resemble to robots control actions. Firefighters are able to move at different speed, and to manoeuvre a water hose. In reality, agents have hand tools available but this does not happen in Pyrosim. Agents have a water hose and a personal tank, which lets them free to walk wherever they want. This may be a considerable simplification of reality but the practical effect of using a water hose with a personal water tank may be compared to the effect of one or several firefighters using hand tools. An important property in Pyrosim is that actions may result differently than what is expected. This may happen in both locomotion and water jet control actions. Finally, Pyrosim enables agents to communicate text messages between them with no restrictions. This property enables to use communication-based coordination models like the ones used in reality. However, communication should not be used intensively to avoid flooding the simulator.

We also analysed the kind of attacks that may be experimented in Pyrosim. We concluded that both direct and indirect attacks may be performed, although with some limitations. The most significant limitations are related to indirect attacks. Pyrosim does not support natural or human-made barrier and neither the construction of firelines. However, it is possible to simulate the effect of firelines constructing wet-lines.

At the end of the chapter we analysed the Pyrosim's environment complexity and concluded that Pyrosim is on the most complex class of environments. For cooping with such complexity we will use the Emotion-based Agent architecture developed in [7]. Moreover, Pyrosim creates an environment where team coordination is essential for dealing with the fire, which makes of it a good platform to test the coordination model that we will propose in the next chapters.

Chapter 5

Proposed Model for Team Coordination

In this chapter, we propose a model for coordinating a team of agents in a spatial domain. The model we propose is specially designed for coordinating teams of firefighting agents, but may be used in other domains where spatial coordination is an issue and where agents have communication capabilities although perception is restricted. The proposed model implements team coordination at two levels: *Global Coordination* and *Local Coordination*. Global Coordination is concerned with coordinating the entire team and is responsible for assigning high-level tasks to agents. These tasks should lead agents to occupy strategic positions in the terrain, and to follow a given approach to the addressed problem (in this case, the fire). In order for agents to execute these tasks efficiently, they may need to cooperate locally. In the next sections, we will first describe the Global Coordination model, and then we will detail how agents cooperate to execute some specific tasks.

5.1 Global Coordination

Global Coordination is centralised, which means that the overall team behaviour is controlled by one agent, the *Leader* (in a firefighting crew, the crew boss is the Leader). The Leader responsibilities include being aware of the global situation, reasoning about it, and assigning specific tasks to agents in order to carry out a given plan. There are several reasons that impelled us to use centralised coordination:

- Real firefighting teams use centralised coordination;

- One agent, the Leader, may have access to the global situation and make decisions based on global knowledge. In a decentralised solution, the information would be dispersed, and no agent would have a complete understanding of the situation. The alternative would be for every agent to be aware of the other agents knowledge, but that would increase drastically communications between agents.

- Centralising tactical/strategic decisions let other agents to focus on their tasks execution.

- Using centralised coordination is not synonym of the Leader being a master ruling a set of slaves. Agents may have local autonomy and are not obliged to carry out tasks that go against their own goals. In fact, this is the main principle of Coordination by Partial Hierarchical Control described in chapter 3.

Of course that centralising decision has its drawbacks. First, there is the danger of creating a bottleneck on the Leader agent. To avoid this, teams should not have too many agents, and communication should be used only when needed. Additionally, when coordinating larger teams, one should use an hierarchy of command to ensure that a Leader only controls directly a small set of agents. Another problem with centralised coordination is that if communication fails or if the Leader fails, the team coordination is lost. Therefore, in the firefighting domain, if a Leader participates directly in the firefighting, it should have a more cautious behaviour. Nevertheless, these disadvantages did not presented a problem because the main purpose of this thesis is to test firefighting tactics, and not to produce agents that have to act in the real world or participating in competitions.

In the following subsections, we will propose a model for achieving Global Coordination. This model is concerned with creating a flexible structure that enables the definition of different approaches to the problem (in this case, the burning fire). After defining the coordination model, we instantiate it to the forest firefighting domain.

5.1.1 Coordination Model

As in the work of Stone and Veloso [14, 15] and Reis and Lau [13, 16, 17], every agent has a *Role* (5.3). There is a predefined number of Roles (5.2) which define different agent behaviours. The number of Roles is not related with the number of agents. Teams may be homogenous (all agents have the same Role), or heterogeneous (agents playing different roles).

$$Agents = \{Agent_1, Agent_2, Agent_3, \ldots, Agent_{nagents}\} \tag{5.1}$$

$$Roles = \{Role_1, Role_2, Role_3, \ldots, Role_{nroles}\} \tag{5.2}$$

$$AgentRole_i \in Roles \quad \forall i = 1..nagents \tag{5.3}$$

There are several ways of organising agents to attack fire. In chapter 2 we have seen that fire could be attacked directly in the tail, head or flanks, or that it could be attacked indirectly by constructing firelines or wet-lines. To implement these attacks we have defined a set of *Attack Plans* (5.4) that specify how agents should approach fire.

$$AttackPlans = \{AttackPlan_1, AttackPlan_2, \ldots, AttackPlans_{nattacks}\} \tag{5.4}$$

In the proposed model, an Attack Plan defines a sequence of *Tasks* for a given Role (5.7)). In this way, agents that share the same role will have the same tasks to carry out. In general, an Attack Plan may be carried out by one Role only, although there are Attack Plans that may be executed by a given set of roles (5.6).

$$Tasks = \{Task_1, Task_2, Task_3, \ldots, Task_{ntasks}\} \tag{5.5}$$

$$AdmissibleRoles_i \subseteq Roles \quad \forall i = 1..nattacks \tag{5.6}$$

$$\begin{gathered} AttackPlan_i = \{Role_j, AttackTask_1, \ldots, AttackTask_{nattacktasks}\} \\ \forall i = 1..nattacks \quad Role_j \in AdmissibleRoles_i \quad AttackTask_k \in Tasks \end{gathered} \tag{5.7}$$

A firefighting crew may perform different attacks simultaneously. To define the attacks distribution among the team, the concept of *Tactic* was created (5.8). A Tactic defines for each role how many agents will play that role, and which Attack Plan will be performed by those agents (5.9).

$$Tactics = \{Tactic_1, Tactic_2, Tactic_3, \ldots, Tactic_{ntactics}\} \tag{5.8}$$

$$\begin{gathered} Tactic_{i,j} = \{TacticAttack_j, AgentAssignment_j\} \qquad \forall i = 1..ntactics \\ \forall j = 1..nroles \qquad TacticAttack_j \in AttackPlans \\ AgentAssignmet_j \in [0..1] \qquad \sum_{j=1}^{nroles} AgentAssignment_j = 1 \end{gathered} \tag{5.9}$$

From (5.9), we gather that agent assignment to roles is defined using relative quantities (e.g. allocate half of the team to a given role). However, in our implementation, it is also possible to define agent distribution using absolute quantities (e.g. allocate 2 agents to a given role). Additionally, agent distribution may be defined dynamically at run-time, which enables to product more complex role assignments.

It is also possible to define Tactics that change the team approach over time. We call these tactics *Dynamic Tactics* (5.10), and they are composed by a set of Tactics with *Activation Conditions* (5.11). Activation Conditions define when to switch to a given tactic.

$$DynamicTactics = \{DynamicTactic_1, \ldots, DynamicTactic_{ndynamictactics}\} \tag{5.10}$$

$$\begin{aligned} DynamicTactic_i = \{ & ActivationCondition_1, SubTactic_1, ..., \\ & ActivationCondition_{nsubtactics}, SubTactic_{nsubtactics}\} \\ & \forall i = 1..ndynamictactics \quad SubTactic_j \in Tactics \end{aligned} \tag{5.11}$$

All tactics are defined at the Leader agent level, and therefore, only the Leader has knowledge about tactical information. The other agents only know their role and the tasks that they must execute, which are assigned by the Leader using tactical information.

5.1.2 Implemented Tactics and Attack Plans

For instantiating the model described above, several Tactics, Attack Plans and Roles were defined. We will first describe the Attack Plans (with the associated roles) and then we will focus in the Tactics definition.

Attack Plans

For defining Attack Plans, we have followed an object-oriented approach. This approach allow us to use inheritance to define shared properties and behaviours among Attack Plans. Figure 5.1 illustrates a class diagram of the Attack Plans we implemented, where the classes with no colouring represent Attack Plans that may be instantiated, and the classes coloured in grey represent abstract classes that are used for structuring proposes only. We will now describe every class from this diagram.

Attack The Attack class defines everything that is common to every Attack Plan. Regardless of the kind of attack, Attack Plans always have two stages: the positioning and the attack management. The positioning stage is concerned with placing the agents in positions that enable them to start the attack in good conditions. Once agents are in position, the attack management stage is activated. This stage is concerned with allocating tasks to agents that are performing the attack. Initially, agents receive tasks to attack a given area. Every time agents complete their tasks, they receive new tasks to attack a new target area.

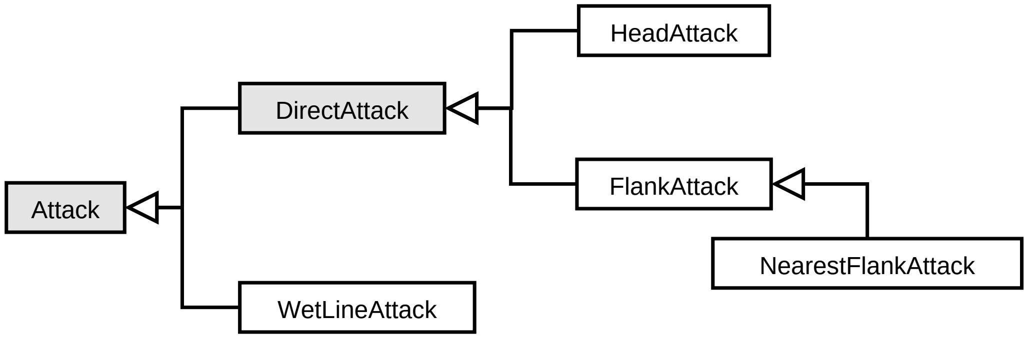


Figure 5.1: Implemented Attacks

Direct Attack In out implementation, Direct Attacks* always have the same behaviour. Therefore, we defined a common attack manager that may be parameterized to produce different kinds of Direct Attacks. This manager receives the desired area to attack, and the preferences about the attack direction. Based on these parameters, and in the team location, the manager computes the best adjacent area to attack using a cost function (5.12). This cost function tries first to approximate agents to the desired attack location, while trying to minimise the distance that the team must travel. In the end, a penalty factor is applied to discourage attacking in directions that may put the team in danger.

$$cost = (distanceToTeam + 2 \times distanceToDesiredArea) \times directionPenalty \tag{5.12}$$

In Direct Attacks, the attack manager also defines situations where the attack must be aborted, such as, when the target area has no fire (because direct attacks fight fire directly in the flames), and when the team attack direction is no longer valid because of eminent danger of the team being caught by fire from behind.

Head Attack Head Attacks are Direct Attacks performed in the head of the fire. Therefore, this Attack Plan starts by placing agents in adjacent area of the fire's head, and then tries to keep agents attacking the head. This Attack Plan may be performed by agents with the *head_attacker* Role.

Flank Attack The Flank Attack is another Direct Attack that is performed over the tail of the fire in one of its two edges. Agents with the Role *left_flank_attacker* or *right_flank_attacker* may perform this kind of Attack Plan. In the first case, agent are positioned behind the left flank and try to keep with it, and in the other case the same thing is done but for the right flank.

*Please refer to section 2.1.3 for more information about Direct Attacks in the firefighting domain.

Nearest Flank Attack This Attack Plan is a special case of the Flank Attack that concentrates all agents in the back of the fire's edge that is closer to the team. Agents with the Role *tail_attacker* may perform this Attack Plan.

Wet-Line Attack The Wet-Line Attack is the only Indirect Attack† that we have implemented. This attack defines an area that should be wetted by agents, which then try to put as much water as they can inside that area in an homogeneous way. There are four Roles that may execute this attack: *head_wet_line_builder*, *tail_wet_line_builder*, *left_wet_line_builder*, and *right_wet_line_builder*. Each of these Roles is concerned with building a line in front of the fire, on its backs or in one of its edges.

Tactics

Using the above Attack Plans, it is possible to define several kinds of approaches to fire. Previously in this chapter, we defined two different types of tactics: Static and Dynamic. The former, defines a fixed distribution of roles and attacks to agents that does not change over time (equation 5.9). The later enables changing this distribution based on activation conditions (equation 5.11).

Static Tactics Table 5.1 summarises the implemented Static Tactics. These tactics are only some examples of the Static Tactics that may be created. We have implemented tactics where all firefighters have the same Role and concentrate their efforts in a single area (ST1 and ST2), where the team is split to directly attack different areas (ST3, ST4, and ST5), where the team is split to surround the fire with water to prevent it from spreading (ST6), and where both Direct and Indirect Attacks are used to prevent fire from spreading while attacking it directly to minimise damage in the forest (ST7). By defining this variety of tactics we aim to have tactics that may succeed in situations that others fail. In these tactics, assignments to roles are defined using a fixed ratio. However, for one of the roles we define that it will get the *rest* of the team. In this way, after calculating the team assignments, if there are any firefighters left they get the most important Role (e.g. In ST6, if the team size is 5, one firefighter goes to the tail, another to the right, another to the left and two to the front of the fire).

Dynamic Tactics At the moment, only two Dynamic Tactics were implemented, which are presented in table 5.2. These tactics try to attack the fire in two stages: (i) a first Indirect Attack to confine the fire, and (ii) when the fire is dominated, the team tries to attack it directly to try saving part of the confined area. The differences between the two tactics are related to the Roles assigned to

† Please refer to section 2.1.3 for more information about Indirect Attacks in the firefighting domain.

Table 5.1: Implemented static tactics

Tactic	*Role*	*Attack*	*Distribution*
All in the Tail (ST1)	tail_attacker	Nearest Flank	All
All in the Head (ST2)	head_attacker	Head	All
Split by Head and Tail (ST3)	head_attacker tail_attacker	Head Nearest Flank	1/2 Rest
Split by Flanks (ST4)	left_flank_attacker right_flank_attacker	Flank Flank	1/2 Rest
Split to Surround (ST5)	left_flank_attacker right_flank_attacker tail_attacker head_attacker	Flank Flank Nearest Flank Head	1/4 1/4 1/4 Rest
Surrounding Wet-Line (ST6)	left_wet_line_builder right_wet_line_builder tail_wet_line_builder head_wet_line_builder	Wet-Line Wet-Line Wet-Line Wet-Line	1/4 1/4 1/4 Rest
Surrounding Wet-Line with Tail Attack (ST7)	left_wet_line_builder right_wet_line_builder tail_wet_line_builder head_wet_line_builder tail_attacker	Wet-Line Wet-Line Wet-Line Wet-Line Nearest Flank	1/5 1/5 1/5 1/5 Rest

firefighters in the second phase: DT1 assignees to each firefighter the part of the fire behind the wet-line he built, and DT2 concentrates all the team in the tail, where the fire is less intense. Much more Dynamic Tactics could be created with more complex behaviours, although this will be left for future work. Another important issue to emphasise is that every tactic we created is for the propose of developing trial-and error experiments, to determine which tactics work best in a given situation. However, it is possible to create Dynamic Tactics that select a Static Tactic based on situation analysis.

Table 5.2: Implemented dynamic tactics

Tactic	*Activation Condition → Sub Tactic*
Wet-Line and then Split (DT1)	When simulation starts → Surrounding Wet-Line When fire intensity stabilises → Split to Surround
Wet-Line and then Attack Tail (DT2)	When simulation starts → Surrounding Wet-Line When fire intensity stabilises → All in the Tail

5.2 Local Coordination

During an attack, agents that share the same role also share the same tasks. To execute these tasks efficiently they must coordinate their actions. For instance, when two firefighters are attacking two adjacent cells simultaneously (one firefighter per cell) they have more difficulties putting the fire down compared to when they first attack together one of the cells and then the other. To achieve this cooperative behaviour we implemented a Local Coordination mechanism for direct attacks. This mechanism uses perception about other agents, and predefined Local Coordination rules.

When agents receive an assignment for directly attacking a given area, they must choose which cell to attack first. This decision may depend on several factors, such as the distance to the target cell, the fire intensity in that cell, the danger of attacking that cell, and if the cell is being attacked by a teammate or not. When an agent enters in the target area, the first thing he does is to verify if another agent is already attacking the cell. If there is, the agent goes to the teammate location in order to help him fighting that fire. Otherwise, the agent selects the nearest cell inside that area and attacks it. If two agents start attacking two different cells simultaneously, two problems arise: (i) the agents are not coordinating efforts, and (ii) the other agents do not know which agent to help. To solve these problems, agents have predefined rules for handling these kind of conflicts. These rules define criterions for evaluating the quality of the agent position to determine which agent to help. The quality of the position is calculated based on the agent progression inside the target area. If two or more agents are in positions with the same quality, the agent's name is used to resolve the conflict.

5.3 Summary and Conclusions

In this chapter we proposed a model for coordinating a team of agents. This model was especially designed for firefighting scenarios but may be applied to other domains where spatial coordination is required and where at least one agent (the Leader) may communicate with every other. The proposed model handles coordination at two distinct levels: Global Coordination and Local Coordination. Global Coordination defines a flexible structure to define different team behaviours. With this coordination structure we were able to implement several firefighting tactics that will enable us to test which approach is the best for a given scenario. A Leader agent is responsible for interpreting this structure and to assign tasks to the other agents in order to carry out a given tactic. Local Coordination enables agents to cooperate locally in order to perform specific tasks. Local Coordination does snot need communication. It is based on a predefined set of rules that have as input perceptions about the other agents and about the agent himself.

Chapter 6

The Team Coordination Model in Action

In this chapter, we will use the model proposed in the previous chapter to coordinate a team of agents performing in the environment generated by the Pyrosim simulator. We will start by describing the perception limitations of agents in this environment. Agents do not have access to overall information about the environment, which makes more difficult carrying out Tactics. To solve this problem, we propose a mechanism for managing global information about the environment. This mechanism supports Global Coordination and provides a global vision of the current situation. Next, we will present a GUI that was developed for monitoring the Leader perspective about the environment and about the other agents. Finally, we describe one of our experiments for showing how the proposed coordination model works in a forest firefighting scenario generated by Pyrosim.

6.1 Perception Restrictions in the Pyrosim Simulator

In section 4.1.1 we analysed the capabilities of agents perception in the Pyrosim environment. We mentioned that agents receive visual information at three distinct levels:

- *Close-Range Maps*: very detailed information about the cell where the agent is located and the cells immediately surrounding the agent.

- *Medium-Range Maps*: cover a much larger area than Close-Range Maps (equivalent to 7x7 Close-Range Maps), but with far less detail (1/9 of the detail). They are also relative to the agent position (the agent is in the centre of the map).

- *Global Maps*: cover the entire simulation area with even less detail than Medium-Range Maps. These maps are only available for information about terrain geography and about the fire. The former is sent at the beginning of the simulation and simulates a terrain chart of the simulation area. The later is sent to agents during the simulation with a short frequency and it enables to detect new fires outside the agents visual range.

By means of analysing the characteristics of these three types of maps, we realise that the information they provide may not be enough to have a global understanding of the current situation. Close-Range Maps may only be used to execute local actions because the area they cover is very small. Medium-Range Maps may be used for coordinating the team for fighting fires that occupy an area considerably smaller than the map's range. However, in fires with a larger area it would be more difficult to coordinate the team, especially in situations where the team is split, because the Leader would not be able to see neither the entire fire nor his teammates. Global Maps cover the entire simulation area but the detail they provide is very low, and they are sent with a short frequency, which delays tactical reactions to changes in the environment. In conclusion, agents perceptions *per se* may not be enough to guarantee a good support for carrying out Tactics. In the next section, we will propose a mechanism for solving this problem.

6.2 Managing Global Knowledge about the Environment

We will now propose a mechanism to manage global information about the environment. This mechanism will be centred on the Leader, and will enable him to conduct Tactics based on the information that he perceives, and the information that his teammates perceive. Additionally, this mechanism will also store the information that was perceived in a recent past, to enable the Leader remember the state of areas that no agent perceives at the moment. We will call this mechanism *World Map*. The World Map covers the entire simulation area and stores information about fire intensity and terrain geography.

To implement the World Map, the first thing to do was to choose the level of detail. There are two options available: Close-Range detail level and Medium-Range detail level. In the previous section, we discussed these two levels of detail when we analysed agents perceptions. We already know that Medium-Range Maps provide information about much larger areas than Close-Range Maps. Therefore, using Medium Range information to update the World Map will allow the Leader to be much more informed about the global situation. Close-Range information is too much localised and the detail that is gained is not very significant for tactical aims. The kind of detail Close-Range Maps provide is very useful when executing local tasks, such as keeping distance from the fire, but for higher-level decisions such detail level may be excessive. Moreover, Medium-Range information is much more stable because it is less sensible to the local dynamics of the fire. This facilitates tactical decisions because it allows the leader to reason in a higher level of abstraction.

Another important issue related to the World Map management is the communication model for gathering information from the firefighting agents. There are two options available: (i) the Leader may decide when to request information from other agents, or (ii) agents may be proactive and decide when to report visual information to the Leader. The former model concentrates decision in the Leader, which may analyse the other agents locations and determine if any agent has important information for updating the World Map. Whenever the Leader discovers agents with useful information he would send them a request for reporting their visual information. On the other hand, using proactive agents frees processing from the Leader and lets the other agents to decide when to send information. Information may be sent periodically whenever an agent is not near his Leader. This was the option that we implemented, although we admit that the first option may reduce communication. The main advantage of the implemented option is that the Leader periodically receives updated information without needing to spend processing time determining when and whom to ask such information. Nevertheless, the parameters that define when to communicate visual information to the Leader must be defined cautiously to avoid excessive communications.

The next step is to define how to merge visual information (from the Leader or from the other agents) in the World Map. This issue will be addressed in the next subsection.

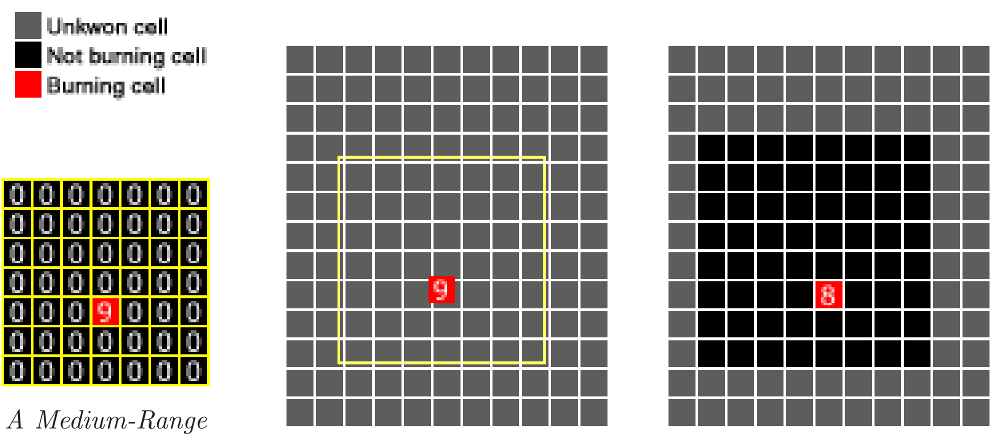
6.2.1 Merging Information

In summary, the World Map is a matrix of Medium-Range cells that covers the entire simulation area. Initially, all the cells are marked as *unknown* to define that no information is available about them. Once the simulation starts, and consequently agents start having perceptions, the World Map is updated with visual information from the Leader and from the other agents.

To manage the World Map, the Leader must deal with information from different times and locations. Even if we consider the problem of managing a World Map that only uses visual information from the Leader, we must merge information that was perceived in a recent past with new information. Moreover, the Leader must merge information that he gathers at different locations. The problem of these situations is that different information will be available about areas that are perceived at different moments with different perspectives. This problem is aggravated when we add visual information coming from other agents. For dealing with this problem, we must set criterions to determine a quality measure that will allow finding out which information is more reliable. We propose managing for each cell a *Confidence* value that is based on three criterions:

- *Novelty*: more recent information is more reliable;
- *Perception Quality*: perceptions that capture the state of the environment more accurately are more reliable;
- *Source*: information that comes from other agents it is less reliable than information that the Leader perceives. The reason for this is that it may take a considerable time from the moment an agent sends its visual information, to the moment the Leader processes it. This makes information coming from other agents less recent.

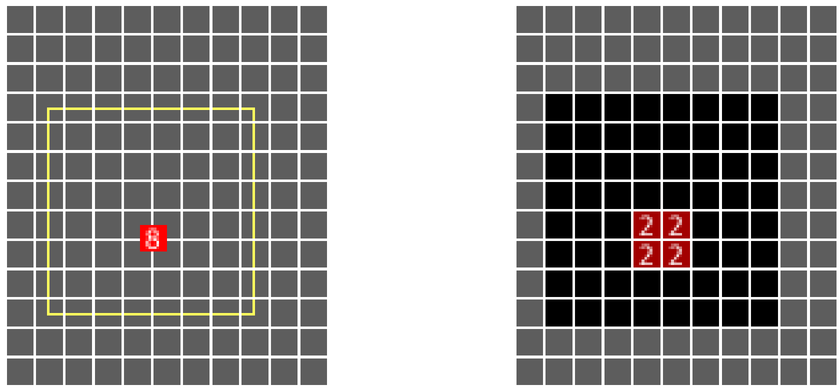
The perception quality of visual information may vary because, usually, cells that an agent is seeing do not exactly match with the cells of the World Map. Therefore, the Leader must calculate the intersections between the cells of Visual Maps with the cells of the World Map in order to have an estimate of the World Map cells. This process is illustrated in figure 6.1 where a Medium-Range Map does not intersect perfectly with the World Map, resulting in estimated values for the covered cells.



A Medium-Range Map *Intersection with the World Map* *Merge result*

Figure 6.1: Merging visual information with the World Map

Large intersection areas between a Visual Map and the World Map indicate that the Visual Map cells almost matches the World Map cells, and therefore minimal estimations will be made. The merging illustrated in figure 6.1 is a good example of a Visual Map with good quality information. On the other hand, figure 6.2 illustrates a worst case scenario because four different cells of the Visual Map contribute equally to the calculation of a single cell of the World Map. To define the quality of the information of a cell of the World Map, we propose using the largest intersection area between that cell and the Visual Map that originated it. For instance, in figure 6.1 the burning cell has a quality of approximately 0.9. On the otherwise, in the worst case scenario (figure 6.2) the quality of each one of the burning cells is approximately 0.25. A perfect match would result in a maximum quality value of 1.



Intersection of a Medium-Range Map with the World Map — *Merge result*

Figure 6.2: Worst case scenario when merging visual information with the World Map

When updating the World Map, we start by determining the Confidence value in the new visual information. The Confidence in the new information for a cell of the World Map is equal to the quality of the cell's information*. If the Confidence on the new information is higher than the Confidence on the stored information, then the cell is updated.

Ensuring information quality is not enough. The environment is dynamic and new information, even having lower quality, may be more relevant than very accurate information that is out of date. In order to open room for more recent information we decrease the levels of Confidence over time. The more time it passes without a cell being updated, the lower is the Confidence value of the information that is stored. If a cell is not updated for a long time, Confidence decreases to very low levels and the cell returns to its original *unknown* state. In our implementation, the cells Confidence decreases a fixed value (0.01) every 4 seconds.

6.2.2 Analysing the World Map

With an updated World Map the Leader is able to compute several useful information for its coordination decisions. First, in order to select the cells that

*When the information comes from agents other than the Leader, a penalty factor is subtracted to the cell's Confidence value.

may be directly attacked, the leader calculates the *Fire Outbound.* The Fire Outbound is a list of cells that are accessible to the team with information about the accessible directions. Besides this, the Leader also keeps track of the *Fire Propagation Direction.* He calculates the main propagation direction comparing the current situation with the initial situation when the team arrived to the fire. Finally, using the Fire Propagation Direction, the leader identifies the *Head* cell (the most advanced burning cell) and the two extremities of the *Tail.* This information is very important for tactical execution because Attack Plans are based in the fire anatomy. When the Leader first arrives to the fire, he has no way of knowing the Fire Propagation Direction and uses the wind direction to identify the anatomic parts of the fire.

6.3 Monitoring the Leader

In order to monitor the Leader knowledge, we developed a GUI (the *Leader Monitor*) that shows the World Map and other information related to the team (figure 6.3).

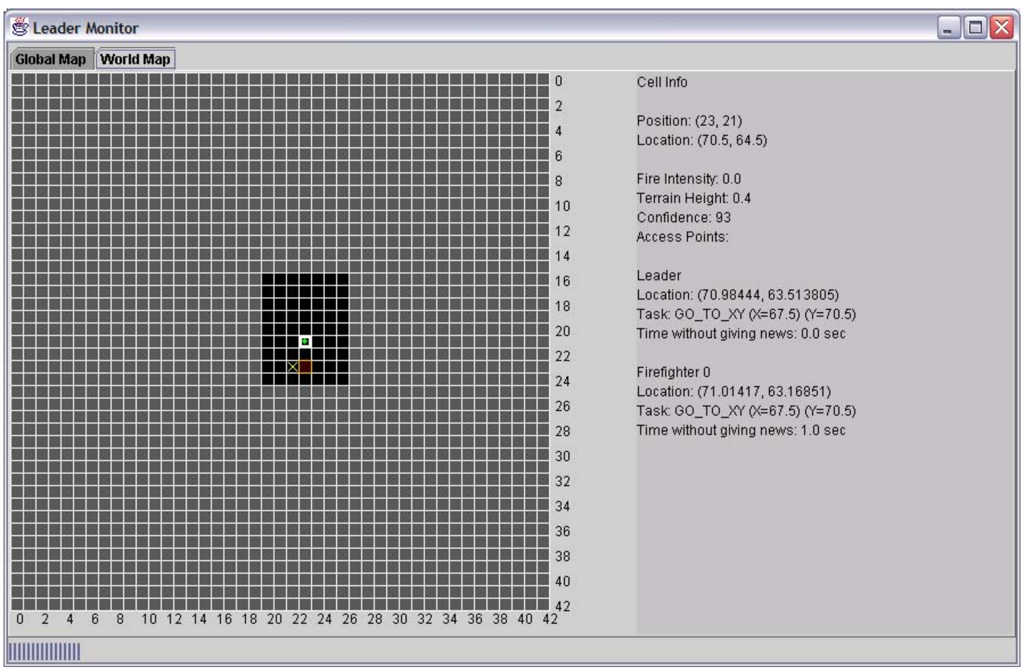


Figure 6.3: The Leader Monitor GUI

In the Leader Monitor it is possible to see every cell of the World Map. In case of cells with information available it is possible to inspect the fire intensity, the terrain height, the Confidence value, and the edges accessible to the team in case of the cell is burning. In addition, it is also possible to monitor the information that the Leader knows about all the team elements, which includes the tasks that agents are executing, their location and the time spent since the last time an agent communicated with the Leader. In addition, several information is displayed graphically in the World Map of the Leader Monitor GUI. Table 6.1 describes the meaning of the symbols that are used for showing such information.

Table 6.1: Symbols used in the World Map of the Leader Monitor

Symbol	*Colour*	*Meaning*
■	Grey Black Red	No information about the cell Cell with no fire Burning cell; the colour intensity is proportional to the fire intensity in the cell
◆	Green	Agent or agents in the cell; this symbol alternates with a number that represent the number of agents in the cell
×	Yellow	A target cell where one or more agents must go to
◁	Yellow	A target cell that one or more agents must attack; the triangle orientation defines the attack direction (up, down, right or left)

6.4 Our Team in Action

To provide a better understanding of how the proposed coordination model works, we will now present the result of a simulation in the Pyrosim platform using one of the Tactics defined in the previous chapter. For monitoring the simulation we will use snapshots from PyroViz† and from the Leader Monitor. For this simulation, we have selected tactic DT2 because it uses both Indirect and Direct Attacks. This Tactic starts by creating a Wet-Line around the fire and then, when fire is under control, all firemen attack the Tail. We will use a team of four agents for this simulation. Bellow is a step by step description of this simulation.

1. Agents connect to Pyrosim and try to find the Leader by broadcasting messages asking who the Leader is. The Leader replies to the messages and registry the agents who done the requests. At the beginning, the team is in the base (start position), waiting for an alarm.

2. The Leader receives information about a fire that is located outside of his visual range. At this moment, the leader only knows that there is a fire in that location, but he does not know its proportions neither the time at

†PyroViz is a tool for visualising simulations in the Pyrosim platform.

which the fire begun. Then, the Leader sends requests to all agents in the team to move to the fire location (figure 6.4).

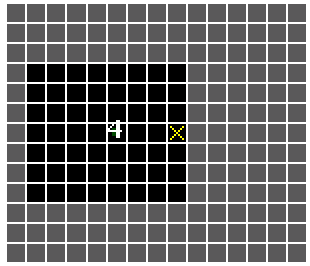


Figure 6.4: The leader World Map indicating the team target location

3. When the leader sees fire (or any one of its teammates report that has seen fire), he tries to analyse the available information in order to identify the Head, and both extremities of the Tail. The tactical mechanism is activated and the Leader assigns Roles to agents according to the Tactic. In this case, every agent will have a different Role (*head_wet_line_builder*, *tail_wet_line_builder*, *left_wet_line_builder*, and *right_wet_line_builder*) but will perform the same kind of Attack Plan (Wet-Line Attack). Using the positioning suggested by the Attack Plan, the Leader defines possible positions to place agents and request them to go to those positions (figure 6.5).

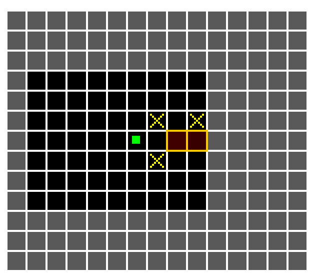


Figure 6.5: The leader World Map indicating the team positioning

4. Finally, agents arrive to their target positions and the Leader requests them to start their attack (figure 6.6). The moment an agent arrives to his target location, he receives the attack order. In this case, the Leader assigns areas (one per role) that must be wetted.

5. Firefighters (agents) execute their tasks, walking from one side to another in the assigned area with their water hose on and set to maximum water flow (figure 6.7).

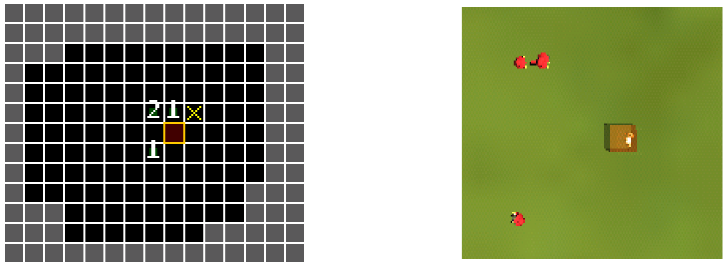


Figure 6.6: Part of the team in position to start building the wet-line

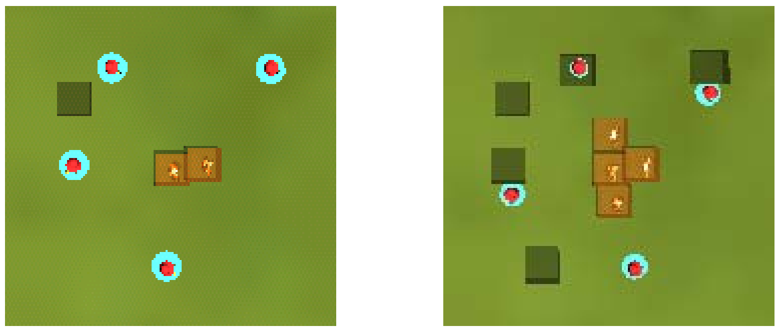


Figure 6.7: Team building the wet-line

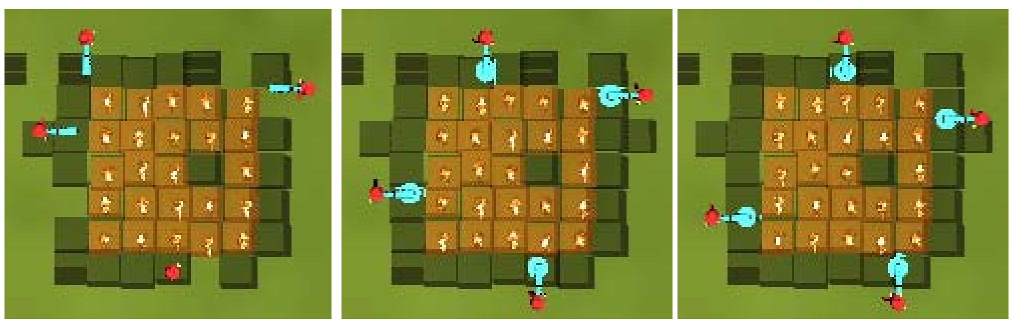


Figure 6.8: Agents moving away from the fire while wetting the ground

6. When it gets too hot for a firefighter, he steps back and continues wetting the ground but at a farther distance, which obliges him to reduce water flow (figure 6.8).

7. When the Leader detects that the fire stops growing or that it is growing very slowly, the condition for switching tactics is activated and the Leader assigns the *tail_attacker_role* to every agent. All the team will be performing a Nearest Flank Attack. The Leader requests agents to go to one of the extremities of the Tail of the fire and to attack it (figure 6.9).

8. Inside the target area, firefighters have full autonomy. They go to the nearest burning cell inside that area and start attacking it together, coordinating

their efforts (figure 6.10).

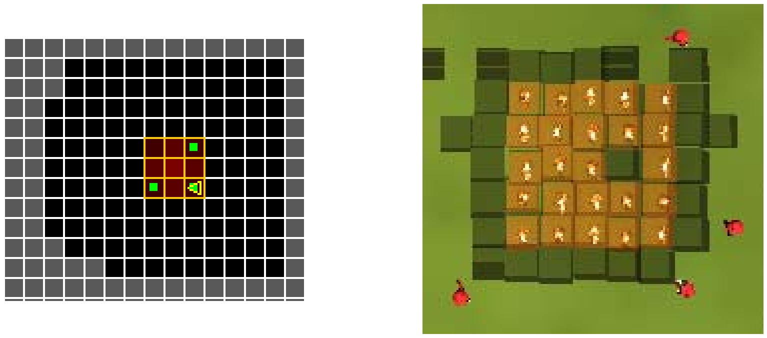


Figure 6.9: The leader World Map indicating the team positioning for a tail attack

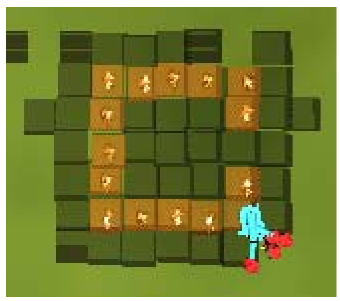


Figure 6.10: Team local coordination attacking the fire

9. When firefighters finish the area, they report it to the leader which assigns them a new area to attack. Eventually, the leader could have detected that the area was finished before the agents reported it. In this simulation this situation did not happen because agents spent all the time trying to put the fire out on the first target area.

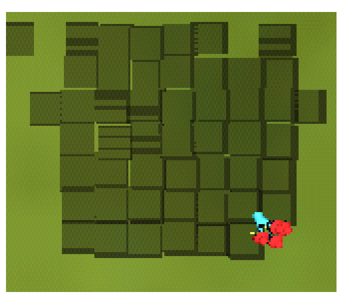


Figure 6.11: Team wetting a fire cell to avoid fire re-ignition

10. When the fire is out, firefighters continue wetting the cell they were fighting to avoid that fire re-ignites (figure 6.11).

In this experiment agents were able to confine the fire by using an Indirect Attack. After that, they tried to minimize damage by attacking the fire directly. The Tactic worked as expected and the fire was successfully extinguished.

6.5 Summary and Conclusions

In this chapter we described how we applied the coordination model proposed in the previous chapter to the scenario created through the Pyrosim forest fire simulator. First, we identified the difficulties of being aware of the global situation in the environment created by Pyrosim. Agents have limited perception with different levels of detail, which make difficult carrying out Tactics. To solve this problem we proposed a mechanism for managing global information about the environment, which we called *World Map*. The World Map is able to store all the information that the Leader and the other agents together perceive. For accomplishing this task successfully, the World Map manages Confidence values for its cells, which enable to know when cells state should be updated. For calculating the Confidence level of visual information we defined three criterions: novelty, perception quality, and source. Using these criterions the Leader is able to decide when to update the World Map information. Next, we described the analysis that the Leader performs on the World Map in order to obtain higher-level information. These analyses process is especially concerned with determining the parts of the Fire Anatomy (the fire Head and both extremities of the Tail). Using this information it is possible carrying out Tactics minimizing the risk of misunderstanding the situation in hands.

We also described a tool that we developed for monitoring the Leader knowledge about the environment and about the other agents. This tool enables to see the Leader's World Map and the tasks he requests to the other agents. Finally, we presented an experiment for showing how the proposed coordination model works in a simulation on the Pyrosim forest fire simulator. We followed this demonstration using the monitoring tool we developed and a visualiser for the Pyrosim simulator.

Chapter 7

Experimental Results

In this chapter we will d test the implemented model for team coordination in Pyrosim. We will use two different scenarios and we will test some of the proposed Tactics on them. We will also use teams with different sizes to see the influence of this factor. Our claim is that there will be tactics that result better in some situations than others, like it happens in reality. We also claim that, in general, larger teams (including more agents) will have a better performance than smaller teams.

We will start this chapter by defining the conditions in which experiments were done, and describing the scenarios we used to test Tactics. Next, we will analyse the results of the experimented Tactics for each scenario. Finally, we conclude this chapter with the evaluation of the obtained results.

7.1 Experimental Conditions

For running experiments we used two machines (table 7.1): a notebook for running the Pyrosim Server and a PC for running all the agents. The two machines were connected in a 100MBps LAN.

Table 7.1: Machines used for running experiments

	Processor	*Memory*	*Operative System*	*Task*
PC	AMD Semprom™ 2800+	512 MB	MS Windows XP	Agents
NB	AMD Athlon™ XP 1800+	512 MB	Red Hat Linux 8.0	Pyrosim

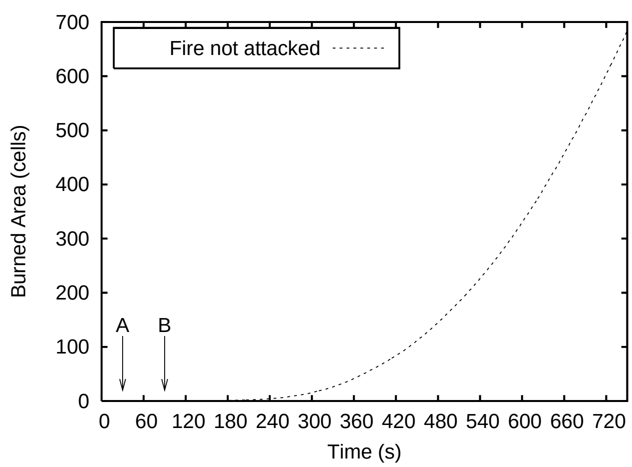
We used the same environment configuration for all experiments. The configuration file may be consulted in appendix B. In short, Pyrosim was configured to produce a fire with high propagation speed in an almost flat terrain. Wind was moderate (25Km/h) and blowing against the agents base. In this way, agents were facing the Head of the fire. The agents base was located near the fire but outside of the visual range of the team, at a distance of 13 cells (208 meters). Agents have a maximum time of 750 seconds (12.5 minutes) to put the fire down. If they cannot control fire within this period the simulation ends.

We will test part of the Tactics proposed on chapter 5 in two different scenarios:

- *Scenario A*: agents receive a fire warning 30 seconds after the simulations starts;

- *Scenario B*: agents receive a fire warning 90 seconds after the simulations starts.

Warning signs may seem too close to generate significantly different scenarios, but in a fast spreading fire, like the one we are using, 1 minute can make the difference. We used a fire with these characteristics in order to accelerate experiments. Experiments must run in real-time and we could not afford to enrol in much longer experiments. In figure 7.1 we illustrate the evolution of the burned area when the fire is not attacked. It is possible to see that after 12.5 minutes of simulation the area burned by fire was approximately 685 cells (17.5 hectares).



Elapsed Time	Burned Area
60	0.30
120	0.60
180	1.14
240	4.36
300	15.43
360	41.72
420	83.73
480	145.19
540	226.52
600	330.42
660	457.03
720	603.16
750	684.72

Figure 7.1: Evolution of the burned area when the fire is not attacked.

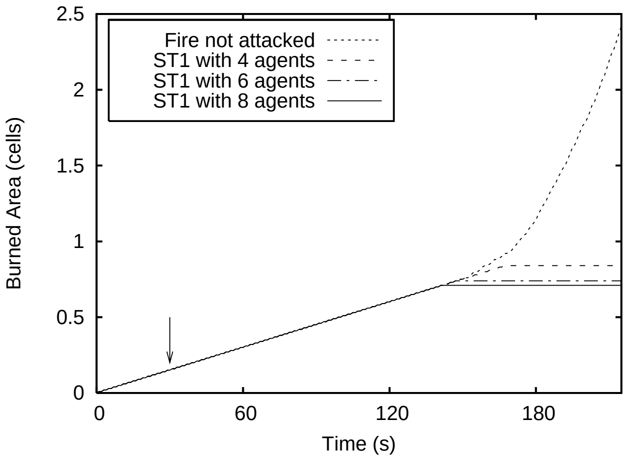
The results we will present in the next sections were obtained using the arithmetic average of a set of 4 tests per Tactic. However, the graphics only illustrate one of the four tests, the one that most approximates the averaged values of the four tests.

7.2 Scenario A: Early Detected Fire

In this scenario, the Leader will detect a fire 30 seconds after it started. We will start by experiment Tactics based on Direct Attacks to see if the fire may be controlled in this way. Next, we will try Indirect Attacks and compare the results of the two approaches.

7.2.1 Experiments with Tactics based on Direct Attacks

We will start by analysing the behaviour of a team using the ST1 Tactic. This Tactic consists in concentrating all agents in the Tail of the fire and attack it with water. We selected this tactic because it is a common used tactic in firefighting when the fire may be attacked directly [11].

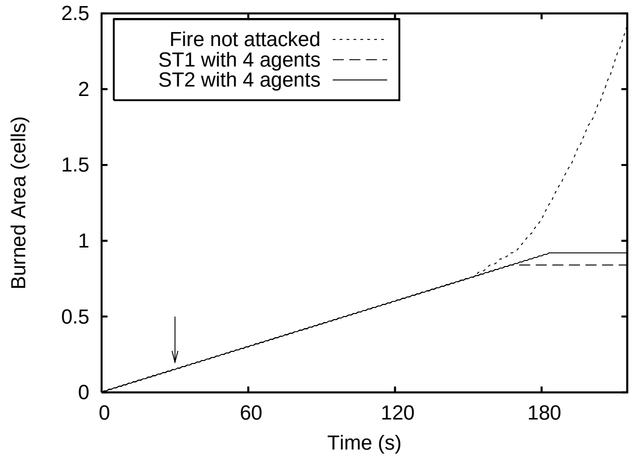


# Agents	Burned Area	Elapsed Time
4	0.83	186
6	0.76	173
8	0.74	169

Figure 7.2: Scenario A: Results for the ST1 tactic (All in the Tail)

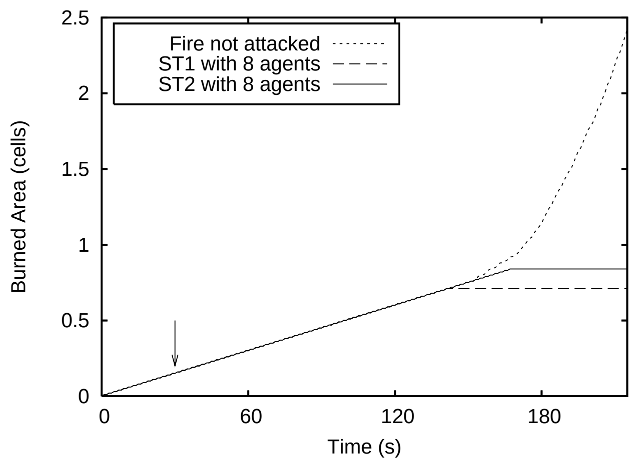
Analysis (figure 7.2): Tactic ST1 obtained good results with every team from 4 to 8 agents. In these scenario, when agents arrive to the fire there is only 1 cell burning. In all experiments, the team was able to prevent fire from spreading and to put the fire down before the cell burned completely. As we expected, we observed that the team performance improves when we increase the number of agents of the team. With more agents, the team is able to reduce the burning area and spends less time fighting the fire. However when we increase the team size from 6 to 8 agents, the performance improvement is not very significant. This kind of analysis may have interest for dimensioning teams when fighting several fires simultaneously.

We will now compare the ST1 and ST2 tactics. The difference between these tactics is that ST1 attacks the Tail of the fire, and ST2 attacks the Head. Head attacks are considered to be more dangerous, but in this scenario the team is facing the Head of the fire and arrives to its attack positions earlier. We will compare these two tactics using teams of 4 and 8 elements.



Tactic	Burned Area	Elapsed Time
ST1	0.83	186
ST2	0.89	194

Figure 7.3: Comparing tactics based on Direct Attacks using a team of 4 agents



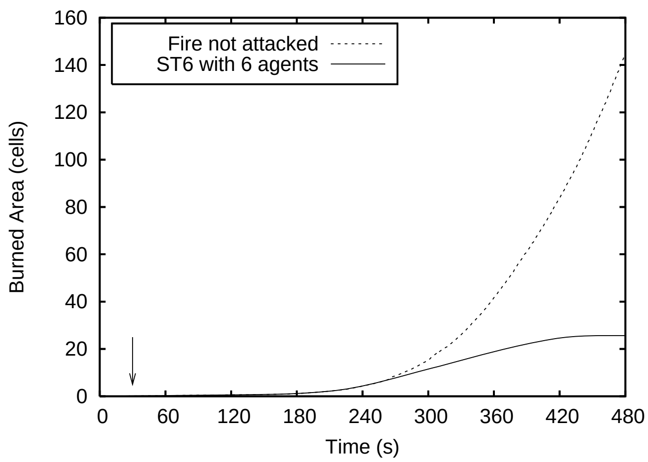
Tactic	Burned Area	Elapsed Time
ST1	0.74	169
ST2	0.89	199

Figure 7.4: Comparing tactics based on Direct Attacks using a team of 8 agents

Analysis (figures 7.3 and 7.4): When using a team of 4 agents, both Tactics achieved similar performance levels, although attacking the Tail (ST1) results is a smaller burned area and needs less time to put the fire down. When raising the number of agents, attacking the Head (ST2) does not produce better results. In fact, the ST2 tactic is not very stable. When attacking the Head of the fire agents are against the wind, and as result the temperature in the Head is considerably higher than in the Tail. This obliges agents to attack fire at a farther distance, and consequently to use a lower water flow in its hose. Additionally, agents run away from fire much more often because of the high temperatures and also because the Head of the fire has a more unpredictable behaviour. Therefore, has it happens in reality, when attacking the fire directly it is usually preferable to attack it in the Tail.

7.2.2 Experiments with Tactics Based on Indirect Attacks

In spite of knowing that Indirect Attacks are usually used when a Direct Attack is not possible, we will test some Tactics based on Indirect Attacks for comparing results with the previously experimented tactics and with the results for the next scenario. We will start by using Tactic ST6, which is the simplest implemented Tactic that is based on an Indirect Attack. This Tactic uses Wet-Line Attacks to surround the fire with water, aiming that the fire will stop spreading when it reaches the water lines.



# Agents	Burned Area	Elapsed Time
4	25.69	473
6	25.57	473
8	25.66	474

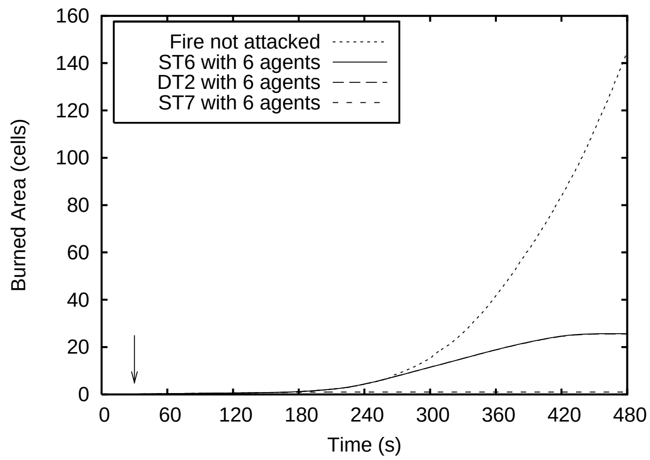
Figure 7.5: Scenario A: Results for the ST6 tactic (Surrounding Wet-Line)

Analysis (figure 7.5): As we expected, the ST6 Tactic was able to control fire spreading. Only 4 firefighters were needed to create a Wet-Line with sufficient water to prevent fire from spreading. Therefore, is not surprising that results obtained using more agents did not improve the results, once the delimited area is always the same. However, this Tactic sacrifices a much wider area than the Direct Attacks tested in the previous section, and spent much more time and water.

We will now make an experiment with two other tactics that try to enhance ST6 tactic:

- ST7 Tactic: assigns part of the team for creating the Wet-Line, while the rest of them team attacks the fire Tail directly;

- DT2 Tactic: has a similar behaviour to ST6, but when it detects that the fire is under control it concentrates all firefighters in the Tail of the fire.



Tactic	Burned Area	Elapsed Time
6 agents		
ST6	25.57	473
DT2	25.53	473
ST7	1.00	220
8 agents		
ST6	25.66	474
DT2	25.61	475
ST7	1.00	221

Figure 7.6: Scenario A: Comparing Tactics based on Indirect Attacks

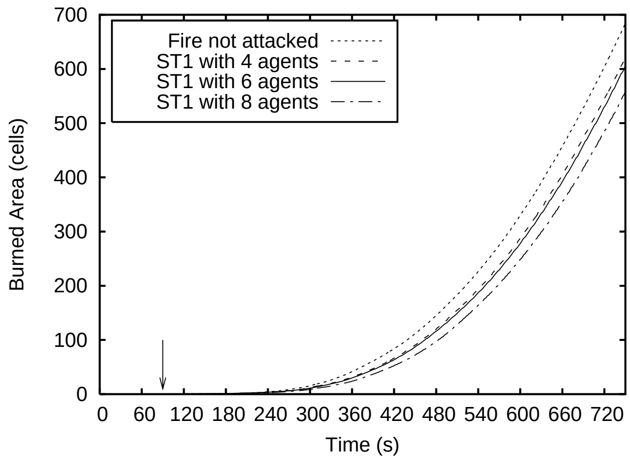
Analysis (figure 7.6): Tactic DT2 did not improve the results of the original tactic ST6. We believe that there are two reasons for this: (i) the Tactic is using a very conservative activation condition for switching to the Tail Attack, and (ii) at the moment that firefighters start attacking the fire directly, the fire is very intense and therefore very hard to fight. As for tactic ST7, using part of the firefighters to attack the fire directly was sufficient to prevent it from spreading. Firefighters were not able to prevent the initial cell from burning completely but the water they used was sufficient for prevent fire from spreading to the other cells. Contrary to our expectations, when using 8 agents there was no performance improvement. Currently, the only explanation we have for this situation is related with the machine where the agents were running that had its processor time fully occupied, which decreases the individual performance of agents. This Tactic achieves results near the ones achieved by Tactics based on Direct Attacks only, but it needs more agents and spends much more water for creating the water-line. However, if the Direct Attack fails, this water-line may stop fire from spreading.

7.3 Scenario B: Not So Early Detected Fire

In this scenario, the Leader will detect a fire 90 seconds after it started (1 minute later than in the previous scenario). We will start by experiment if Tactics based on Direct Attacks achieve positive results or not. Next, we will try Indirect Attacks and compare the results of the two approaches.

7.3.1 Experiments with Tactics based on Direct Attacks

Like in the previous scenario, we will test the ST1 tactic (all firefighters attacking the Tail of the fire). We will not show the results of ST2 Tactic because we have already seen that in a less difficult scenario this Tactic was inferior to ST1.



# Agents	Burned Area	Slow Down
4	625.49	9%
6	606.30	12%
8	569.60	17%

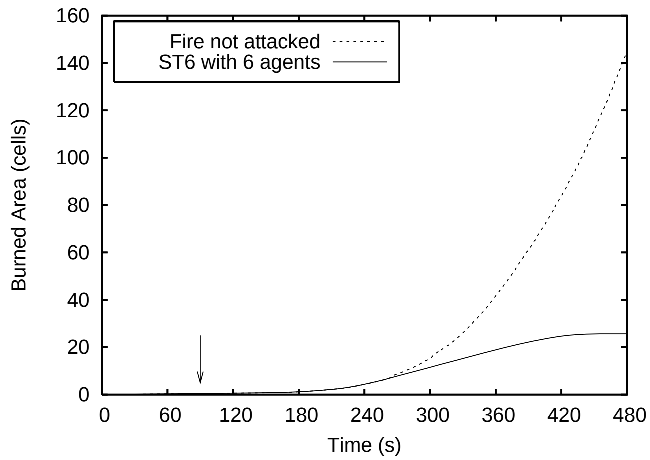
Figure 7.7: Scenario B: Results for the ST1 tactic (All in the Tail)

Analysis (figure 7.7): In this scenario, using the ST1 tactic is not sufficient to put the fire down. We tried this tactic with teams of 4, 6 and 8 agents, but all of them result unsuccessfully. Nevertheless, it is possible to observe that the team performance rise with the number of agents. In particular, when we passed from 6 to 8 agents the team had a significant performance improvement, slowing the fire down for more 5% than with 6 firefighters. This improvement is not only the result of incrementing the team size, it is also a sign that the agents local cooperation is working. More agents result in better performances only if they join efforts. If agents did not concentrate their water jets to try put out the fire, the water effect over fire would be much less noticeable.

7.3.2 Experiments with Tactics based on Indirect Attacks

After failing to control fire using Tactics based on Direct Attacks, we hope that Tactics based on Indirect Attacks will achieve better results. We will start again using tactic ST6, which consists in creating a water-line around the fire.

Analysis (figure 7.8): Once again, ST6 proved to be effective controlling the fire, even when the team was notified later. In fact, the results of applying this tactic in this scenario are very similar to the ones of the previous scenario. This happens because the team is able to wet the ground with sufficient water to stop fire from spreading. Using 4 agents was sufficient to control the fire, although we detected that this team takes a little longer to control it, showing signs of

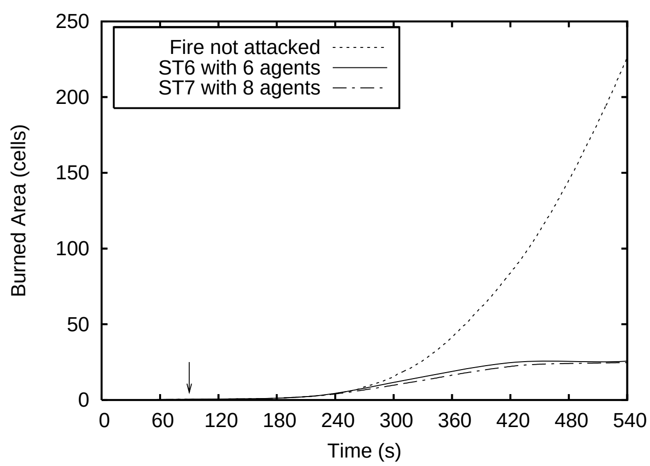


# Agents	Burned Area	Elapsed Time
4	26.10	482
6	25.66	473
8	25.69	474

Figure 7.8: Scenario B: Results for the ST6 tactic (Surrounding Wet-Line)

difficulties. However when we increment the team size to 6 or more agents the fire is controlled with no problems.

We will now try to enhance ST6 Tactic by mobilising part of the team for attacking the Tail of the fire directly (ST7 tactic). In the previous scenario we also used DT2 tactic, but the achieved results were nearly the same than using ST6, and therefore we will skip this Tactic.



Tactic	Burned Area	Elapsed Time
6 agents		
ST6	25.66	473
ST7	25.58	485
8 agents		
ST6	25.69	474
ST7	25,24	522

Figure 7.9: Scenario B: Comparing Tactics based on Indirect Attacks

Analysis (figure 7.9): Using ST7 with 6 agents, slightly improved the results of the ST6 tactic. When increasing the team size to 8 agents this improvement was more noticeable. The agents that were attacking the fire's Tail were able to reduce to burned area and to slow down the fire propagation speed. However, to obtain such results, this tactic needs to use 4 more agents than the original tactic ST6.

7.4 Results Evaluation

We have tested an implementation of the proposed model for team coordination in two different scenarios. Using this model, we have implemented and tested five different Tactics including different collective approaches to fire. Tactics mainly based on Direct Attacks achieved very good results when the team arrives early to the fire (scenario A), but loose control of the fire if they arrive too late (scenario B). Tactics mainly based on Indirect Attacks are more robust. They have a stable behaviour because they predefine an area to sacrifice instead of trying to attack fire directly. The results they obtained in both scenarios are nearly the same and they were always capable of controlling the fire. However, in scenarios where Direct Attacks may be applied successfully (scenario A), Indirect Attacks may sacrifice considerable larger areas than Direct Attacks. Enhancing tactics based in Indirect Attacks by using Direct Attacks enabled to obtain good results when both attacks were used simultaneously. In scenario A, the Indirect Attack worked as a backup plan in case the Direct Attack fails, and in scenario B the Direct Attack slows down the fire which give more time for the other agents to build the wet-line. However, Tactics that combine both kinds of Attacks need more agents than Tactics that use only one kind of Attack in order to achieve similar or better results.

Our experiments resulted as expected according to firefighting theory. When the fire may be controlled using Direct Attacks the results are better than using Indirect Attacks, otherwise Indirect Attacks or a combination of both techniques should be used. This is a sign that our Tactics behave in a similar way than real Firefighting Tactics, which places one step closer of providing a platform for experimenting Firefighting Tactics in simulated Forest Fire scenarios.

We also observed that using teams with more agents usually leads to better performances. This was the case of Tactics that used Direct Attacks, where agents cooperate locally in order to attack fire more effectively. The increase of performance related to the increase of elements in the team is an indicator that local coordination is working. In the case of tactics based in Indirect Attacks, we did not observe this relation because with 4 agents only the Attack could be performed successfully. Using more agents could be useful in larger fires where the sacrificed area is wider, and where longer wet-lines need to be built.

Chapter 8

Conclusions

8.1 Overview

The main goal of this thesis was to develop and test a model for coordinating a team of agents performing in the forest firefighting simulation domain.

For tackling this domain, we first had to study the basics of real firefighting to understand how forest fires behave, what influences their behaviour, how real firefighting teams attack fires, and which techniques do they use. In Chapter 2 we summarised these topics and we identified the problems and requirements for coordinating teams of agents for attacking forest fires.

At the same time, we investigated team coordination models that have been developed in the area of MAS (Chapter 3). Most of the models we found were used in domains other than forest firefighting, and tried to implement coordination with minimal or no communication. The models concerned with spatial coordination were related to the Robotic Soccer domain, which has several domain-specific differences compared to the domain we are working on. However, these models provided a good base for designing the proposed coordination model.

In work preceding this thesis [7], a forest fire simulator was developed, Pyrosim. This was the platform in which we implemented and tested the model we are proposing. To understand the capabilities and limitations of this tool (which would condition the implementation of our model) we decided to analyse Pyrosim before continuing our work (Chapter 4). In [7], Sarmento tackles the complexity of the Pyrosim's environment proposing an Emotion-based Agent architecture. The author claims that Emotion-like mechanisms are a requirement for intelligent behaviour for resource-bounded agents operating in complex, dynamical and real-time environments. The agents developed by Sarmento had no coordination capabilities and therefore were not capable of extinguishing fires. These agents were the starting point of our implementation.

Based on our research in the firefighting domain and in team coordination, we proposed a model for spatial coordination, especially adapted to the coordination of firefighting teams (Chapter 5). This model provides a flexible structure that enables to define different kinds of approaches to fire. Overall coordination is achieved by using a Leader agent that assigns high-level tasks to the other agents. To provide a better understanding of the agent-based proposed model, we defined several Firefighting Tactics for coordinating a team in the Pyrosim simulator. We also proposed a mechanism for agent cooperation while executing local tasks, which does not uses communication.

For effectively testing the proposed model, we implemented a team of agents for the Pyrosim simulator (based on the agent architecture developed in [7]), taking into account problems related with limitations of agents perception. Since agents may only perceive part of the environment, it becomes difficult for the Leader agent to be aware of the overall environment state. To solve this problem, we implemented a mechanism, described in Chapter 6, which merges information from all agents to create a global view of the scenario. This more global information enables the Leader to carry out Firefighting Tactics, even when he cannot perceive the entire fire. However, leader's global information does not prevent agents from using local information.

Finally, when the implementation was completed, we were able to test the proposed model in the Pyrosim simulator. Our goal was to experiment different Firefighting Tactics in alternative scenarios to see how they perform, and to determinate the influence of the team size in the overall team performance. Chapter 7 presents the results we obtained in two different scenarios. These results show that the number of agents in place has direct impact on the team performance in attacks to the fire where local coordination is used. We also observed that the performance of a given Tactic depends of the scenario where it is used in a similar way as it happens in reality. There are Tactics with very good performance when the fire is attacked in an early stage but that fail to control fire when the team arrives too late. Other Tactics achieved a more stable behaviour in the two scenarios being always able to control fire, although achieving lower performance levels in situations were the first Tactics succeed.

8.2 Discussion

In this thesis we proposed and implemented a coordination model that enables to experiment Firefighting Tactics in different forest fire scenarios. The Tactics we implemented are based on Firefighting Tactics that are used by real firefighting teams. The results of our experiments also reflect the firefighting theory we have studied. We therefore believe that we are in a good direction towards an implementation of a platform that enables to experiment Forest Firefighting Tactics in different scenarios. Moreover, using centralised coordination while giving local autonomy to agents as proven to be a good model for this domain, since our agents were always able to execute their tactics correctly.

We also had the opportunity of working with Emotion-based Agents, although this was not the core of our work. During our experiments we were able to observe adaptive behaviours from our agents. For instance, when the agents combat more intense fires they tend to be more cautious, and in the end of the simulation they keep a higher distance from the fire than when they started. This behaviour results from the constant danger at which agents are exposed when fighting intense fires, which affects their Emotional Mechanisms that make agents to adopt a more conservative operation mode. During this work much time was invested on the configuration of the Agents Emotional Mechanisms, and in the development of individual skills in order for our agents to be robust enough to test Firefighting Tactics. The results are gratifying, although we have realised that configuring an Emotion-based Agent is a difficult matter that should be subject of future research.

Finally, during our work we also explored other areas from AI that are not covered in this dissertation for the sake of brevity. We have implemented Path Planning algorithms (for agent locomotion), a Message Protocol as well as the message creation and parsing mechanisms (for agents to communicate), and Optimization algorithms (for improving Greedy tasks assignments), among others.

8.3 Future Work

The next logical step to continue this work is addressing the problem of Tactic Selection. We already observed that for a given scenario there are tactics that achieve better results than others. Tactics that fail in a given scenario may be the best in another one. Defining rules that help deciding which Tactic to select it is a difficult task because there are many variables that affect the fire behaviour, such as the wind, the terrain geometry and the kind of vegetation in the terrain. Additionally, other factors contribute to the success or failure of the tactic, such as, the state of the fire when the team arrives to it, and the number of agents available. One interesting approach to address this problem would be using machine learning over simulation logs to try to find out rules that help the Leader agent to select the best tactics. The rules that result from this selection process may also contribute to the problem of tactical selection in real forest fires.

Another possible direction is to enhance the Pyrosim simulation tool to generate more realistic scenarios and to offer a wider variety of firefighting agents. By including barriers (roads, paths, water streams, etc) in the simulated environment, the simulation would be more realistic and agents would be able to take advantage of them when fighting fires. Moreover, including houses, populations or other critical areas would enable to test Tactics for ensuring the safety of these areas, and would create situations where agents have to attend to multiple and sometime opposing goals (e.g. controlling the fire vs. protecting a critical area; deciding which areas are more important to protect). Also, it would be interesting if Pyrosim offered different kinds of firefighting agents, such as, firefighters with hand tools, bulldozers, aeroplanes, etc. These would allow to test different kind of tactics and to tackle the problem of coordinating heterogeneous teams.

Finally, the field of Emotion-based Agents could be explored in a MAS perspective. In this thesis we developed a MAS of Emotion-based Agents, but we did not explore the effects of the Emotional Mechanisms in the overall team behaviour. Additionally, the output of these Emotional Mechanisms may be used for Tactical decision-making, in particularly for helping to decide when to switch Tactics. We started addressing these problems in [8] but only now it is possible to test our ideas.

Bibliography

[1] Divisão da Defesa da Floresta Contra Incêndios. Incêndios florestais – relatório de 2005. Technical report, Ministério da Agricultura do Desenvolvimento Rural e das Pescas, Portugal, January 2006. http://www.dgrf.min-agricultura.pt/v4/dgf/pub.php?ndx=2271.

[2] Marco Wiering, Fillipo Mignogna, and Bernard Maassen. Evolving neural networks for forest fire control. In M. van Otterlo, M. Poel, and A. Nijholt, editors, *Benelearn '05: Proceedings of the 14th Belgian-Dutch Conference on Machine Learning*, pages 113–120, 2005.

[3] Marco Wiering and Marco Dorigo. Learning to control forest fires. In H. Haasis and K. Ranze, editors, *Proceedings of the 12th international Symposium on 'Computer Science for Environmental Protection'*, pages 378–388, 1998.

[4] RoboCup Federation. Robocup official site. Online, 2006. http://www.robocup.org.

[5] RoboCup Federation. Robocuprescue official site. Online, 2006. http://www.rescuesystem.org/robocuprescue/.

[6] Timo Nüssle, Alexander Kleiner, and Michael Brenne. Approaching urban disaster reality: The resq firesimulator. Technical Report 200, April 2004.

[7] Luís Sarmento. An emotion-based agent architecture. Master's thesis, Faculdade de Ciências da Universidade do Porto, 2004.

[8] Luís Sarmento, Daniel Moura, and Eugénio Oliveira. Fighting fire with fear. In C. Ghidini, P. Georgini, and Wiebe van der Hoek, editors, *EUMAS'04 2nd European Workshop on Multi-Agent Systems*, pages 627–634, Barcelona, December 2004.

[9] Eugénio Oliveira and Luís Sarmento. Emotional advantage for adaptability and autonomy. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*

(AAMAS'03), pages 305–312, New York, USA, 2003. ACM Press. http://doi.acm.org/10.1145/860575.860625.

[10] Eugénio Oliveira and Luís Sarmento. Emotional valence-based mechanisms and agent personality. In Guilherme Bittencourt and Geber Ramalho, editors, *Advances in Artificial Intelligence, Proceedings of SBIA02*, volume 2507 of *Lecture Notes in Computer Science*, Brazil, 2002. Springer. http://www.fe.up.pt/ẽol/PUBLICATIONS/2002/personality7.ps.

[11] Marc Nicolas and Grant Beebe. The training of forest firefighters in indonesia. Technical report, German Agency for Technical Cooperation, European Union, and Government of Indonesia, 1999.

[12] André Buss. Incêndios. Internet page, FURB - Universidade Regional de Blumenau, 2005. http://www.furb.br/monitoramentoflorestal/incendios.html.

[13] Luís Paulo Reis. *Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, June 2003.

[14] Peter Stone and Manuela Veloso. Task decomposition, dynamic role assignment and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110(2):241–273, June 1999.

[15] Peter Stone. *Layered Learning in Multi-Agent System*. PhD thesis, School of Computer Science, Carnegie Mellon University, December 1998.

[16] Luis Paulo Reis, Nuno Lau, and Eugénio C. Oliveira. *Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents*, volume 2103 of *Balancing Reactivity and Social Deliberation in Multi-Agent System - From RoboCup to Real-World Applications*. Springer's Lecture Notes in Artificial Intelligence, Berlin, 2001.

[17] Luis Paulo Reis and Nuno Lau. *FC Portugal Team Description: RoboCup 2000 Simulation League Champion*. RoboCup-2000: Robot Soccer World Cup IV. Springer Verlag Lecture Notes in Artificial Intelligence, Berlin, 2000.

[18] M. Genesereth, M. Ginsberg, and J. Rosenschein. Cooperation without communication. In *Proceedings of the 5th National Conference on Artificial Intelligence, AAAI '86*, pages 51–57. Philadelphia, 1986.

[19] Les Gasser, Carl Braganza, and Nava Herman. Mace: A flexible testbed for distributed ai research. In M. Huhns, editor, *Distributed Artificial Intelligence*, pages 119–152. Pitman Publishers, 1987.

[20] T. Balch and R. Arkin. Communication in reactive multi-agent robotic systems. *Autonomous Robots*, 1(1):27–53, 1994. 1994.

[21] Marco Manei and Franco Zambonelli. Motion coordination in quake 3 arena: a field-based approach. In C. Ghidini, P. Georgini, and Wiebe van der Hoek, editors, *EUMAS 04*, pages 264–278, Barcelona, December 2004.

[22] B. Grosoft. Conflict handling in advice taking and instruction. Technical Report 20123, IBM Research, 1995.

[23] Sébastien Paquet, Nicolas Bernier, and Brahim Chaib-draa. Damas-rescue 2004 report. Technical report, DAMAS laboratory, Laval University, Canada, 2004.

[24] Sébastien Paquet, Nicolas Bernier, and Brahim Chaib-draa. *DAMAS-Rescue Description Paper*. Springer Verlag, Berlin, 2005. (to appear).

[25] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[26] M. Alexander and L. Fogarty. A pocket card for predicting fire behaviour in grasslands under severe burning conditions. Fire Technology Transfer Note 25, Natural Resources Canada, and Canadian Forest Service, June 2002.

[27] National Wildfire Coordinating Group. Glossary of wildland fire terminology, November 1996. USA.

Appendices

Appendix A

Forest Firefighting Glossary

Source: adapted from [27] and [11]

B

BACKFIRING	A tactic associated with indirect attack, intentionally setting fire to fuels inside the control line. Most often used to contain a rapidly spreading fire. Backfiring provides a wide defence perimeter, and may be further employed to change the force of the convection column. Backfiring is a tactic which makes possible a strategy of locating control lines at places where the fire can be fought on the firefighter's terms. Except for rare circumstances meeting specified criteria, backfiring is executed on a command decision made through line channels of authority. See Burning out for difference.
BACKING FIRE	Fire spreading or ignited to spread into the wind and/or down slope.

C

CONFINE A FIRE	To restrict the fire within determined boundaries established either prior to or during the fire.
CONTROL LINE	A comprehensive term for all the constructed or natural fire barriers and treated edges used to control a fire.
CREW	The basic unit of a firefighting team. Usually is formed by 5 to 20 firefighters.

CREW BOSS — Person in charge of a crew. If the firefighting team is composed by only one crew, the Crew Boss is also the Fire Boss

D

DIRECT ATTACK — Any treatment of burning fuel, e.g., by wetting, smothering, or chemically quenching the fire by physically separating the burning from unburned fuel. A suppression strategy in which resources are directed to work close to the fire edge.

F

FIRE ANATOMY — On typical free-burning fires, the spread is uneven with the main spread moving with the wind or up-slope. The most rapidly moving portion is designated the head of the fire, the adjoining portions of the perimeter at right angles to the head are known as the flanks, and the slowest moving portion is known as the rear or the base or the back or the tail.

FIRE BOSS — Person in charge of all people working in a fire. The fire boss has to plan strategy and tactics, and to assign tasks to firefighters in order to control fire and to assure the team safety.

FIREBREAK — Any natural or constructed discontinuity in a fuel bed utilised to segregate, stop, and control the spread of fire or to provide a control line from which to suppress a fire.

FIRELINE — A loose term for any cleared strip used in control of a fire. That portion of a control line from which flammable materials have been removed by scraping or digging down to the mineral soil.

FLANKS OF A FIRE — The parts of a fire's perimeter that are roughly parallel to the main direction of spread. See also Fire Anatomy.

H

HEAD OF A FIRE — See Fire Anatomy.

HEAD FIRE — A fire spreading or set to spread with the wind and/or up-slope.

HOT SPOT — A partially active part of the fire.

I

INDIRECT ATTACK — A method of suppression in which the control line is located some considerable distance away from the fire's active edge. Generally done in the case of a fast-spreading or high-intensity fire and to utilise natural or constructed firebreaks or fuelbreaks and favourable breaks in the topography. The intervening fuel is usually backfired; but occasionally the main fire is allowed to burn to the line, depending on conditions.

NATURAL BARRIER — A naturally occurring obstruction to the spread of fire.

P

FIRE ANATOMY — On typical free-burning fires, the spread is uneven with the main spread moving with the wind or up-slope. The most rapidly moving portion is designated the head of the fire, the adjoining portions of the perimeter at right angles to the head are known as the flanks, and the slowest moving portion is known as the rear or the base or the back or the tail.

R

RATE OF SPREAD — The relative activity of a fire in extending its horizontal dimensions. The forward rate of spread at the fire front or head is usually what is meant by this term.

S

SAFETY ZONE — An area (usually a recently burned area) used for escape in the event the line is outflanked or in case a spot fire causes fuels outside the control line to render the line unsafe.

SPOT FIRE — A fire set outside the perimeter of the main fire by flying sparks or embers.

STRATEGY — An overall plan of action for fighting a fire which gives regard to the most cost-efficient use of personnel and equipment in consideration of values threatened, fire behavior, legal constraints, and objectives established for resource management. Leaves decisions on the tactical use of personnel and equipment to supervisors and leaders in the operations section.

T

TACTICS — Operational aspects of fire suppression. Determining exactly where and how to build a control line and what other suppression measures are necessary to extinguish the fire. Tactics must be consistent with the strategy established for suppressing the fire.

TAIL OF A FIRE — See Fire Anatomy.

W

WET-LINE — Indirect attack that uses water for wetting the ground to try preventing fire from spreading.

Appendix B

Pyrosim Configuration File

```
# SERVER ####################################
SERVER_IP 192.168.2.101
AGENT_SERVER_PORT 5500
VIZ_SERVER_PORT 20000


# GEO #######################################
MAXHEIGHT 1
CR 0
CI 0
SMOOTHLEVEL 0


# AGENT #####################################
BASE_X 58
BASE_Y 70


# FIRE ######################################
BURNING_RATE 5000
INITIAL_ENERGY 1000000
MAX_TEMP 50
TEMP_MOISTURE_EVAP_RATE 0.000011
RANDOM_FIRE 0
N_RANDOM_FIRE 4
FIRE_X 71
FIRE_Y 70
```

```
BROADCAST_COUNT_DOWN 0   #Scenario A: 0   Scenario B: 2
SIMULATION_TIME_OUT 750


# WIND #####################################
MIN_DIR 0
MAX_DIR 0
MIN_SPEED 25
MAX_SPEED 25
TURBOLENCE 1
```