Swarm-Inspired Solution Strategy for the Search Problem of Unmanned Aerial Vehicles

By

Xingbo Li

A thesis submitted by partial fulfilment of the requirements for the Degree of Doctor of Philosophy in Engineering

University of Warwick, Warwick Manufacturing Group

December 2010

List of Figures

List of Tables

Acknowledgement

I wish to express my gratitude to my supervisor, Professor Rajat Roy, for his supervision and advice on the completion of my PhD research. I would also like to thank my tutor and colleague, Mrs Margaret Low, for her support and encouragement. And finally, I sincerely thank my parents for their understanding, care and support throughout all stages of my research.

Declaration

I hereby declare that this thesis is of my own work.

I hereby confirm that this thesis has not been submitted for a degree at another university.

XINGBO LI

Abstract

Learning from the emergent behaviour of social insects, this research studies the influences of environment to collective problem-solving of insect behaviour and distributed intelligent systems. Literature research has been conducted to understand the emergent paradigms of social insects, and to investigate current research and development of distributed intelligent systems. On the basis of the literature investigation, the environment is considered to have significant impact on the effectiveness and efficiency of collective problem-solving. A framework of collective problem-solving is developed in an interdisciplinary context to describe the influences of the environment to insect behaviour and problem-solving of distributed intelligent systems. The environment roles and responsibilities are transformed into and deployed as a problem-solving mechanism for distributed intelligent systems.

A swarm-inspired search strategy is proposed as a behaviour-based cooperative search solution. It is applied to the cooperative search problem of Unmanned Aerial Vehicles (UAVs) with a series of experiments implemented for evaluation. The search environment represents the specification and requirements of the search problem; defines tasks to be achieved and maintained; and it is where targets are locally observable and accessible to UAVs. Therefore, the information provided through the search environment is used to define rules of behaviour for UAVs. The initial detection of target signal refers to modified configurations of the search environment, which mediates local communications among UAVs and is used as a means of coordination. The experimental results indicate that, the swarm-inspired search strategy is a valuable alternative solution to current approaches of cooperative search problem of UAVs. In the proposed search solution, the diagonal formation of two UAVs is able to produce superior performance than the triangular formation of three UAVs for the average detection time and the number of targets located within the maximum time length.

Chapter 1. Introduction

1.1 The Present Research and Related Subject Areas

The present research studies the collective problem-solving of social insects and agent-based intelligent systems [1] in an interdisciplinary context, and applies the problem-solving mechanisms to the cooperative search problem of Unmanned Aerial Vehicles (UAVs).

Collective problem-solving of social insects describes how individual insects cooperate with each other to accomplish complex tasks in their day-to-day life routines. Such a metaphor has raised major research interests over the years due to its capabilities of solving complex problems with rather simple rules of behaviour (Théraulaz 1995; Marco Dorigo 2000; Johnson 2001). Such kind of problem-solving is carried out with distributed intelligent systems, which is an important subject in the research of artificial intelligence (AI). Major research streams of distributed intelligent systems are multi-agent systems and multi-robot systems. Ongoing researches have been conducted in each of these research streams to explore decentralised problem-solving approaches through cooperation and coordination of independent agents with basic rules of activities (Ferber 1999; Simmons, Apfelbaum et al. 2000; Hayes 2002; Wooldridge 2002).

The cooperative search of multiple UAVs is a typical problem of collective problem-solving, which has been investigated by various researchers for a range of optimal solutions (Baum and Passino 2002; Richards, Whitley et al. 2005; Ruini and Cangelosi 2008). The emergent behaviour of social insects and agent-based problem-solving mechanisms provide essential

[1] Well-known as multi-agent systems (MAS)

contributions to resolving the UAV search problem with improved effectiveness and efficiency.

In the following sections, the research and development of traditional AI approaches are briefly introduced, and then provides an overview for the collective problem-solving in social insects and multi-agent systems.

1.1.1 Centralised Problem-Solving: Traditional AI Approaches

The research of artificial intelligence (AI) was founded in 1956 by a group of researchers attending the Dartmouth Conference (J. McCarthy, M. L. Minsky et al. 1956; McCorduck, Minsky et al. 1977) and has been explored for decades. A variety of subject areas have been developed in the AI research, ranging from machine learning, expert systems, artificial neural networks, to evolutionary computation, hybrid intelligent systems, as well as knowledge engineering and data mining. All of these subjects of AI research suggest that the nature of artificial intelligence is to solve complex problems intelligently. From individual human behaviour to emergent phenomenon, the term "intelligence" describes "the quality of a good mind" (James Kennedy and Russell C Eberhart 2001). In addition, (Negnevitsky 2001) refers to "intelligence" as "the ability to learn and understand, to solve problems and to make decisions" and indicates that "a machine is thought intelligent if it can achieve human-level performance in some cognitive tasks".

Traditional approaches of artificial intelligence integrate a range of problem-solving capabilities from hardware to software, which centralises all levels of problem-solving processes. Such highly-integrated and centralised problem-solving of such kind requires enormous computational resources and computer memories, which is time consuming and generates intensive computational complexity. Moreover, centralised approaches of problem-solving have high demand on the completeness and quality of the information resources available. Despite later research and development into the methods for dealing with uncertain and incomplete information (Ng and Abramson 1990; Joya, Frias et al. 1993; Russell and Norvig 2003; Carpenter, Martens et al. 2005), the strict requirements for information available in solving complex problems remained and resulted in constraining further achievements for centralised problem-solving approaches.

1.1.2 Collective Problem-Solving: Emergent Intelligence

The problem-solving patterns of social insects are distributed, flexible and robust, which are facilitated with multiple interactions (direct and indirect interactions between insects and the environment), stimulated activities and self-organisation. This form of artificial intelligence is popularly known as swarm intelligence (Eric Bonabeau 1999; James Kennedy and Russell C Eberhart 2001), which refers to the emergent collective intelligence of groups of simple agents. Therefore, collective problem-solving simulates the emergent behaviour of social insects to solve complex problems in real world.

The emergent behaviour of social insects shows potential features of solving complex problems through the emergence of individual activities. In-depth research (Bernstein 1975; Downing and Jeanne 1988; Franks 1991; Van Dyke Parunak 1997; Susi and Ziemke 2001; Detrain and Deneubourg 2008) has been conducted to investigate the problem-solving phenomenon in the behavioural patterns of social insects. The observation and modelling of the emergent patterns of ant foraging behaviour (Hahn M 1985; Deneubourg and Goss 1989;

Marco Dorigo 2000) and nest construction of termites (Grasse 1959; Deneubourg 1977; Théraulaz 1995) indicate that the environment and environment-related elements have influential contributions at both individual and cooperative levels of insect behaviour.

Meanwhile, a series of research regarding multi-agent systems (Odell, Parunak et al. 2003; Danny Weyns 2005; Simonin and Gechter 2006; Mirko Viroli 2007; Platon, Mamei et al. 2007; Weyns, Omicini et al. 2007) also suggests that the environment has influential roles and responsibilities in the design and applications of multi-agent systems. Through the environment and environment-related influential elements, simple individual agents are able to develop emergent patterns of problem-solving using limited resources and local information.

Inspired by the problem-solving of social insects, swarm intelligence offers a new perspective to solve complex problems through building distributed intelligent systems. Compared to centralised intelligent systems, researchers (Ferber 1999; Wooldridge 2002) consider the following aspects that enable distributed intelligent systems to become an advantageous option for complex problem-solving:

- Problems are physically distributed. For example, traffic control systems such as air/road traffic management, requires robust responses in order to adapt to the fast changing situations in real time applications.
- Problems have great diversity in terms of their functionalities. A typical example is the design, production and assembly of aircraft. Different parts and components of the aircraft such as engine, wings, computer control systems, and cockpit are designed and manufactured by various industrial teams at different locations around the world. Such diversified industrial collaboration generates great complexity of problem-solving, which requires distributed intelligence to accomplish with.

- The vast development of computer networks and software systems requires a distributed view. Distributed intelligent systems such as multi-agent systems are thus considered to be a potential approach for the construction of “open, distributed, diversified and flexible architectures, capable of offering high-quality services for collective work” (Ferber 1999), without acquiring any knowledge in advance.
- Local and more specific perspectives are also needed due to the complexity of problems. Distributed intelligence facilitates complex problem-solving by distributing multiple tasks to simplify the problem, so that each task can be carried out individually to minimise various constraints.
- Systems must have sufficient robustness and flexibilities to adapt to changes in the context and structure of the environment.
- Software engineering, such as the design and development of multi-agent systems, which is more focused on the relationships between autonomous agents, which are mainly referring to the direct and indirect interactions between agents and the application environment.

Existing research of insect behaviour and the design and applications of multi-agent systems show that the influences of environment have essential contributions in collective problem-solving. There has been significant growth of research interests in both of the subjects and intensive research works have been carried out with each of them in parallel. However, to the author’s knowledge, so far there has been no work done to look into the impact of environment-related elements in the collective problem-solving from an interdisciplinary point of view – regarding the social insects and the multi-agent systems.

Therefore, this research takes a step further to consider the influential contributions of environment in an interdisciplinary context. A framework of collective problem-solving is constructed to present the influences of environment in the collective problem-solving of social insects and multi-agent systems. The framework describes the common characteristic properties of collective problem-solving in the two subjects, and explains how the influences of environment are represented by a set of problem-solving mechanisms each for the insect behaviour and multi-agent systems. Such representation indicates that the influences of environment contribute to problem-solving mechanisms in both social insects and multi-agent systems. Based on the interdisciplinary knowledge of collective problem-solving, a swarm-inspired search strategy is proposed and deployed in the cooperative search of Unmanned Aerial Vehicles (UAVs) with related problem-solving mechanisms. The cooperative search of UAVs reflects major characteristics of distributed problem-solving; for instance, UAVs and targets are physically distributed in the search field, and each UAV operates as an individual till communications occur based on specific stimulations. Hence the author uses the cooperative search of UAVs as the specific research problem to investigate the proposed swarm-inspired search strategy.

1.2 Research and Development of UAVs

UAVs have been deployed as a semi-autonomous platform for surveillance and reconnaissance, search, intelligence data collection, remote sensing and other similar operations in both military and civil applications. Compared to manned aircraft, UAVs are considered to have distinctive advantages such as persistence, expendability, increased manoeuvrability and survivability, and lower reliability standards. Originally built for military applications, the systems of UAVs have been explored over decades for both military and civil applications (Li 2003). The current UAV family has three classes of member – standard, medium, and micro-sized UAVs. The applications of UAVs can be categorised according to their sizes, which refer to centralised hardware and software systems for standard and medium UAVs, and distributed hardware and software systems for micro UAVs. More recently, one of the most popular subjects of UAV research is to investigate cooperative search mechanisms of multiple micro-sized UAVs. The system of multiple UAVs is a classic example of distributed intelligent systems, which provides an ideal platform to investigate collective problem-solving.

The cooperative search of UAVs has become one of the major applications for collective problem-solving. Intensive research and development has been conducted over the years in order to identify effective and robust search strategies and related mechanisms for this application. The problem scenario of a cooperative search of UAVs refers to a group of UAVs searching for multiple mobile targets as an individual and through cooperative activities with each other. Below are the major subjects related to the cooperative search of UAVs: cooperative flight formations of multiple UAVs (Bayraktar, Fainekos et al. 2004; Vincent and Rubin 2004; Beard, McLain et al. 2006; Altshuler, Yanovsky et al. 2008), coordinated path planning (Marios M. Polycarpou 2001; Bellingham, Tillerson et al. 2002;

Sujit and Ghose 2004; Geyer 2008), exploration and mapping (Fox, Ko et al. 2006; Rudol, Wzorek et al. 2008; Bryson and Sukkarieh 2009), wireless communications between UAVs (Palat, Annamalau et al. 2005; Morris, Mullins et al. 2006; Allred, Hasan et al. 2007), sensing and image processing techniques (Parunak, Brueckner et al. 2003; Sevcik, Green et al. 2005; Doherty and Rudol 2007), and so on.

The well-established understanding of distributed intelligent systems and especially the emerging study of swarm intelligence offer potential support to further enhancement for a variety of UAV applications. The aim of this research is to construct a swarm-inspired search strategy to accomplish the cooperative search of UAVs. A better solution for the search problem is explored through the design and deployment of individual search behaviour of UAVs and emergent behaviour of multiple UAVs.

1.3 The Aim and Research Objectives

Having understood the influences of the environment on insect behaviour and multi-agent systems, the aim of this research is to propose a swarm-inspired solution methodology with a set of related problem-solving mechanisms to solve the search problem of UAVs. To accomplish this aim, the following objectives are set:

1. Understand the influences of environment that contribute to the emergent behaviour of social insects and the collective problem-solving of multi-agent systems;
2. Establish the knowledge of collective problem-solving regarding the influential contributions of the environment in social insects and multi-agent systems;
3. Use the mechanisms of collective problem-solving to develop a swarm-inspired search strategy in the cooperative search of UAVs;
4. Evaluate the proposed search strategy through the performance of UAVs in the cooperative search problem.

1.4 Thesis Outline

The thesis is structured as below:

Chapter 1 has introduced the overall research, highlighting the research context and related subject areas, aim of the research and research objectives.

Chapter 2 reviews the background literature of collective problem-solving, describing and analysing current research works, focusing on the emergent behaviour of social insects and collective problem-solving in multi-agent systems. Based on the literature study, an interdisciplinary framework of collective problem-solving is presented.

Chapter 3 describes the two main research streams of collective problem-solving: multi-agent systems and multi-robot systems. Key concepts and fundamental problems of each of the research streams are explained, and also emphasised is the deployment of agent-based methodologies in multi-robot systems.

Chapter 4 presents a specific subject of research in collective problem-solving – the cooperative search of UAVs. The author describes and analyses selected examples of the current approaches to the problem. Based on existing research of the cooperative search of UAVs and the framework of collective problem-solving, the author then proposes a swarm-inspired search strategy with a brief introduction.

Chapter 5 provides a detailed description and explanation for a swarm-inspired search strategy. The proposed search solution refers to the framework of collective problem-solving, which is composed of a set of mechanisms designated for the cooperative search of UAVs. The mechanisms are transformed into programmable procedures, which are written in the multi-agent modelling language NetLogo. The design and purpose of each of the mechanisms are explained accordingly.

Chapter 6 explains the design and implementation of experiments, analyses the experimental data and evaluates the proposed search strategy. The results of experiments are further discussed by considering the proposed search solution together with existing search approaches for the cooperative search of UAVs.

Chapter 7 concludes the thesis by emphasising the differences between the cooperative control architecture presented in the literature and the swarm-inspired search strategy presented in this research; categorising the proposed search solution; describing the methodological features of the proposed search solution; and highlighting the research contributions. The conclusion is then finalised with the discussion of potential future work.

Chapter 2. Background Literature: Collective Problem-Solving

This chapter presents the background literature for collective problem-solving. The interest is in two major subjects of research. One of them is the emergent behaviour of social insects, especially the ant foraging behaviour and the nest construction of termites. A series of observations and modelling of insect behaviour (Deneubourg and Goss 1989; Théraulaz 1995; Deneubourg 1999; Eric Bonabeau 1999; Susi and Ziemke 2001; A. C. Mailleux 2003; Karla Vittori, Jacques Cautrais et al. 2004) indicate that the emergent behaviour of many species of insects (such as ants, honeybees, termites, paper wasps, and so on) is a natural system of collective problem-solving. The emergent behaviour of social insects presents important problem-solving characteristics in a decentralised pattern, which are considered to be effective and robust to solve complex problems (Traniello 1989; S. Portha 2004; Detrain and Deneubourg 2008). Section 2.1 describes the ant foraging behaviour and the nest construction of termites to explain the collective problem-solving in social insects.

The other subject of interest is the collective problem-solving in multi-agent systems (Ferber 1999; Weiss 2000; Wooldridge 2002). As a typical example of decentralised intelligent systems, multi-agent systems (MAS) represent important features of collective problem-solving. It becomes one of the most exploited development platform for investigating the solution methodologies of collective problem-solving. The problem-solving characteristics of ant foraging behaviour and nest construction of termites are transformed into solution methodologies of collective problem-solving. Section 2.2 introduces the solution methodologies for the collective problem-solving in multi-agent systems. Based on the understanding of collective problem-solving in insect behaviour and multi-agent systems, section 2.3 presents a framework of collective problem-solving. The framework constructs an interdisciplinary knowledge of the solution methodologies for the collective problem-solving.

2.1 Emergent Behaviour of Social Insects

In-depth research works (Eric Bonabeau 1999; Marco Dorigo 2004; Zomaya 2006) have been conducted to understand the emergent behaviour of social insects in the context of collective problem-solving. Many species of social insects, such as ants, honeybees, wasps, and termites, are observed to have evolving capabilities in terms of solving complex tasks with very simple and basic individual behaviour. The ant foraging pattern (Traniello 1989; Deneubourg 1999; Detrain and Deneubourg 2008) and the nest construction of termites (Downing and Jeanne 1988; Théraulaz 1995) have attracted ongoing research interests. The aim is to learn from their emergent behaviour and to construct problem-solving mechanisms for collective problem-solving that are exploitable with multi-agent systems.

2.1.1 Ant Foraging Behaviour

The observation and modelling of ant foraging behaviour show the following important mechanisms of collective problem-solving: the laying and following of pheromone trails, food distribution and abundance, and self-organisation.

2.1.1.1 Pheromone

Ants forage for food and construct foraging patterns[2] between available food sources and their nests. Initially a few ants set out to search for food sources in a random pattern. Each ant deposits a chemical substance called pheromone to mark the path as it moves around. Pheromone is defined as below (Eric Bonabeau 1999):

"A pheromone is a chemical used by animals to communicate. In ants, a pheromone trail is a trail marked with pheromone. Trail-laying trail-following behaviour is widespread in ants."

Once it finds a food source, it either carries the food back to the nest or return to the nest without the food if the food is too heavy and/or large to carry on its own. In both situations the ant uses the same pheromone-marked path to travel back to the nest and lays pheromone on the path. Meanwhile, if other ants smelled the pheromone, they would be stimulated to follow the same path and lay pheromone as usual. The stronger the pheromone become, the more ants are stimulated to follow the pheromone-marked paths. As a result, the ant foraging behaviour is emerged from the pheromone-laying and pheromone-following behaviour of ants.

2.1.1.2 Food Distribution and Abundance

In addition to pheromone strength, the food distribution and abundance also contribute to the foraging patterns of ants (Bernstein 1975; Eric Bonabeau 1999; Mailleux AC 2000; S. Portha 2004). Depending on the dietary preferences of different ant species, the food distribution and abundance each refers to the food type and number of food sources. The food type (Portha, Deneubourg et al. 2002; S. Portha 2004) and number of food sources (Deneubourg 1999; Nicolis, Detrain et al. 2003) are two of the characteristics of the search environment (Karla

[2] The foraging pattern of ants is also known as swarm raid pattern, especially when referring to army ants.

Vittori, Jacques Cautrais et al. 2004), and each of them is considered to have influences on the collective behaviour of insects.

Food distribution defines the accessibility of food sources, and food abundance refers to the quantity and quality of food sources. (Deneubourg 1989) indicated that food distribution is an important representation of the search environment for ants. Food abundance is another element affecting the foraging strategies of ants (Bernstein 1975). For some species of desert ants, such as *Pogonomyrmnex cali-fornicus* and *P. rugosus,* their foraging patterns become long and narrow when the food density is low. If the food sources become relatively abundant, ants intend to search for food independently instead of converging onto several fixed routes.

Figure 2.1 Foraging Patterns of Three Species of Army Ants: Eciton. hamatum, E. rapax, and E. burchelli

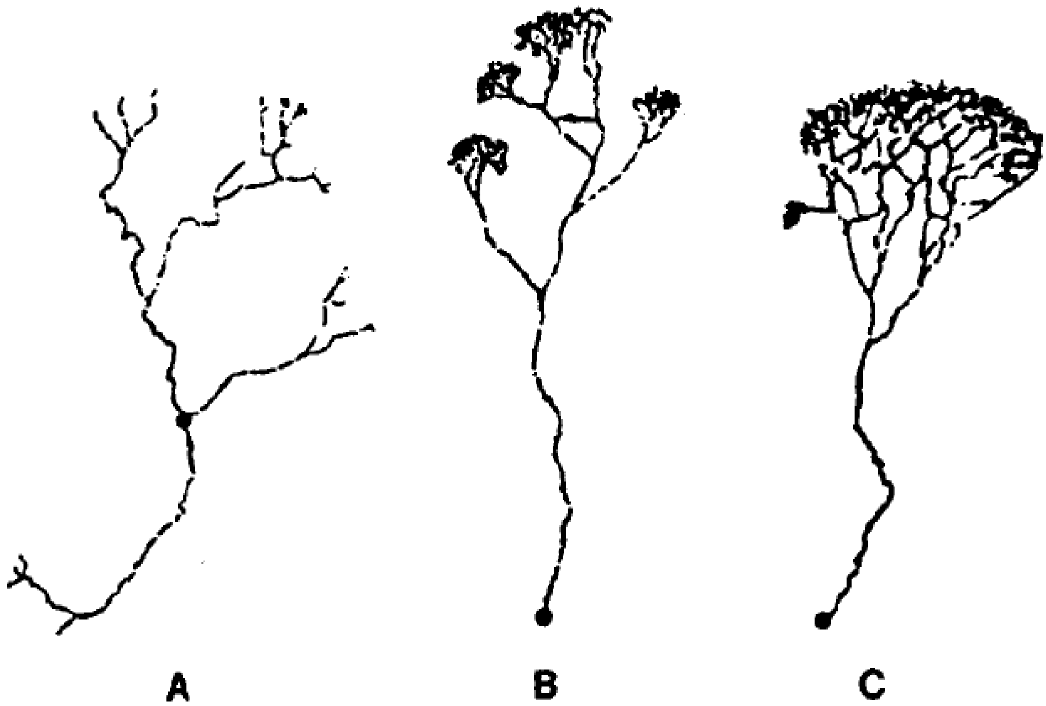


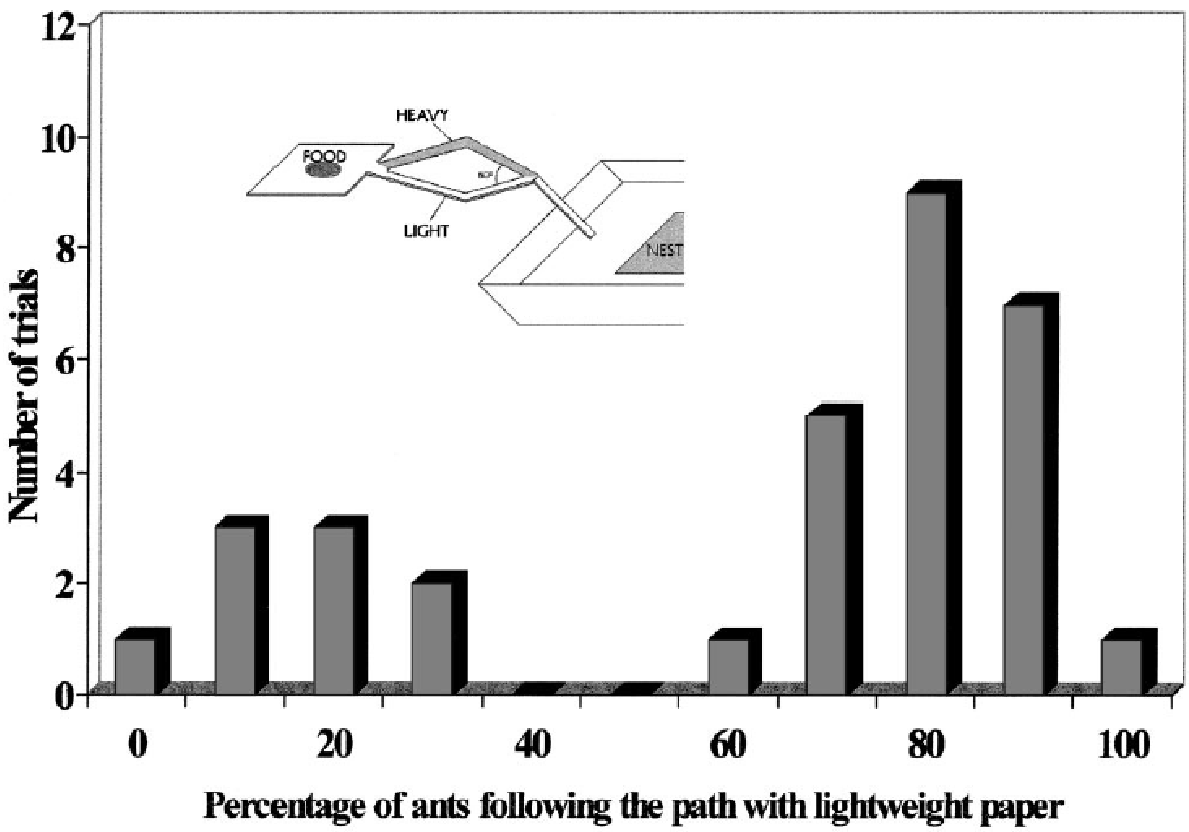
Fig. 1. Foraging patterns of three army ant species, (A) *Eciton hamatum*, (B) *E. rapax*, and (C) *E. burchelli*. Redrawn from Burton and Franks (1985).

Furthermore, different groups of researchers (Burton and Franks 1985; Deneubourg 1989; Hӧlldobler and Wilson 1990; Franks 1991) observed and modelled the foraging patterns of three species of army ants: *Eciton hamatum*, *E. rapax,* and *E. burchelli*. These three species of army ants have different diet preferences, which determine the accessibility of food sources and the quantity and quality of food sources. For instance, as shown in Figure 2.1 (Burton and Franks 1985), the food sources of Eciton hamatum are rarely distributed but large in quantity; the food sources of E. rapax are both easy to be found and in good quantities; the food sources of E. burchelli are easy to be found but only in small quantities each time. Each foraging pattern of the three species of army ants varies depending on the different spatial distributions of food sources and abundance.

In addition to the food type and the number of food sources, the colony size (A. C. Mailleux 2003) and the nature of substrate (C. Detrain 2001) are considered to be the other two environment characteristics (Karla Vittori, Jacques Cautrais et al. 2004), which are also considered to be influential to the emergence of insect behaviour. The colony size represents the inner environment of *Lasius niger* ants, which is demonstrated to be an essential factor that determines the exploratory and foraging responses of *Lasius niger* ants. In particular, the increase and decrease of colony size directly result in changes in communication and foraging behaviour of *Lasius niger* ants (A. C. Mailleux 2003).

Substrate is defined in biology (1997) as "The material on which a sedentary organism (such as a barnacle or a plant) lives or grows. The substrate may provide nutrients for the organism or it may simply act as a support." From the definition the author can infer that, substrate represents physical conditions of the environment. Regarding the nature of substrate, (C. Detrain 2001) designed a two-bridge experiment to investigate the foraging patterns of *Lasius niger* ants under different conditions of substrates. As shown in Figure 2.2 (C. Detrain 2001), the two bridges connect the nest and food sources, with each is covered with lightweight paper and heavyweight paper. The paper bridges are designed as artificial substrate, which provide an experimental platform to study the influence of the substrate on collective choices of ant foraging path. The experimental data shows that more than half of the ants forage on the lightweight paper bridge. The study explained that the non-biotic environmental factors such as substrate have the impact on the collective decision-making of ants. Such kind of impact is independent from the changes of individual behaviour of ants (C. Detrain 2001; Detrain and Deneubourg 2002).

Figure 2.2 Collective Choices of a Foraging Path by *Lasius niger* ants (C. Detrain 2001)



Furthermore, (Detrain and Deneubourg 2002) studied how social insects make comprehensive decisions by processing complex information of the environment. They also indicated that the environment is able to contribute to the emergence of collective patterns as an independent element, without engaging any adjustments of individual behaviour. In addition to the food distribution and abundance, the presence of competitors (Hölldobler 1976; Acosta, Lopez et al. 1995) and the existence of predators (Nonacs and Dill 1988) are also considered as the influential factors of the environment.

2.1.1.3 Self-Organisation

The foraging patterns of ants have been further investigated and validated in a self-organisation model constructed by (Franks 1991). The model shows that ants organise their foraging behaviour through multiple interactions. While searching for food sources, ants carry out multiple interactions via continuous pheromone-laying and pheromone-following processes. Ant foraging behaviour is considered as self-organised foraging activities, which require no centralised supervision and are emerged from simple rules of behaviour at individual level.

(Eric Bonabeau 1999) describes self-organisation to be "a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components. The rules specifying the interactions among the systems' constituent units are executed on the basis of purely local information, without reference to the global pattern".

(James Kennedy and Russell C Eberhart 2001) define self-organisation as "the ability of some systems to generate their form without external pressures, either wholly or in part. It can be viewed as a system's incessant attempts to organise itself into ever more complex structures, even in the face of the incessant forces of dissolution described by the second law of thermodynamics."

Previous research (Hölldobler 1976; Burton and Franks 1985; Rissing 1988; C. Detrain 2001) indicates that the environment contributes to the collective decision-making of social insects through self-organised behaviour. Self-organisation enables cooperative behaviour to be emerged from constrained individual activities, and ultimately generates complex problem-solving capabilities at group level. Therefore, the self-organised phenomenon is an essential problem-solving mechanism in the collective behaviour of social insects.

Below are the four basic features related to self-organised mechanisms (Eric Bonabeau 1999):

- Positive Feedback, such as recruitment and reinforcement. For example, recruitment to a food source is a positive feedback that relies on pheromone laying and pheromone following in the foraging behaviour of some species of ants;
- Negative Feedback, which counterbalances positive feedback to stabilise the collective pattern. In the foraging behaviour of ants, for instance, negative feedback could be the limited number of foraging ants, exhaustion of food sources, and competition between food sources, and so forth;
- Randomised Processes, such as random walks and errors, which initiate new opportunities of exploring new food sources;
- Multiple Interactions; Self-organisation is a continuous process. Although an individual insect is able to carry out self-organised activities, by itself it would unable to maintain continuous processes of self-organisation. Thus, through the interaction with the environment and with other individuals, insects are able to deliver self-organised behaviour continually.

Self-organising mechanisms enable individual insects to accomplish complex tasks at group level by carrying out simple rules of behaviour at individual level. As a decentralised problem-solving strategy, self-organisation is widely exploited by social insects such as ants, honeybees, termites and wasps in a variety of emergent behaviour. The influential factors of environment contribute to the emergence of self-organised behaviour of insects. Self-organisation is considered to be one of the most important problem-solving mechanisms that are exploited in the emergent behaviour of social insects. Not only in the emergent behaviour of social insects, the phenomenon of self-organisation is also considered as an essential component of collective problem-solving in other computational contexts such as collective robotics and multi-agent systems. A survey of current research for the collective problem-solving in multi-agent systems is presented in Section 2.2.

Briefly, ants forage by laying and following pheromone on the routes between food sources and the nest. Pheromone is a chemical substance that works like a messenger to enhance effective communications between ants. The communications of such kind trigger self-organised activities to occur, and as a result, ants are stimulated to converge on the pheromone-marked paths. The structures of ant foraging patterns vary depending on different spatial distributions and abundance of food sources. Food distribution and abundance represent the search environment of ant colonies, and this suggests that the search environment impacts on the foraging patterns of ants.

2.1.2 Nest Construction of Termites

Nest construction is another emergent behaviour of social insects that have inspired the recognition and development of collective problem-solving. The nest construction behaviour of termites was originally described by (Grasse 1959) and similar works have been conducted to study the building behaviour of paper wasps (Downing and Jeanne 1988). Based on the observation of the construction behaviour of termites and paper wasps, a model of self-assembly has been developed and followed by a series of related research that intends to understand the construction behaviour of social insects from the perspective of collective problem-solving (Deneubourg 1977; Smith 1978; Théraulaz 1995; Eric Bonabeau 1999; Camazine, Deneubourg et al. 2003). In particular, the construction of royal chamber of termites has been studied by various research groups over the years (Bruinsma and Wageningen 1979; Grasse 1984; Bonabeau, Henaux et al. 1998) and is used to build the model of self-assembly. The model of self-assembly shows that the construction of royal chamber is accomplished through the combination of self-organisation and template. In addition to self-organising mechanisms, template and stigmergic communications are the two essential components contributing to the construction behaviour of termites.

2.1.2.1 Template

The model of self-assembly indicates that the self-organising activities are constrained by templates. A template[3] is defined to be "a pattern that is used to construct another pattern", which is "a kind of prepattern in the environment, used by insects – or by other animals – to organise their activities" (Eric Bonabeau 1999). In the construction of royal chamber, termites make use of the body of termite queen to build up the royal chamber. The size and shape of termite queen are the template pattern for the construction behaviour of termites. It initialises stimulating configurations of the royal chamber by diffusing pheromone-type of messages to organise the building activities of termites. Termites respond to the pheromone information by carrying out random walks to deposit soil pellets at a constant distance from the queen. The template pheromone also specifies the location and quantity of soil pellets to

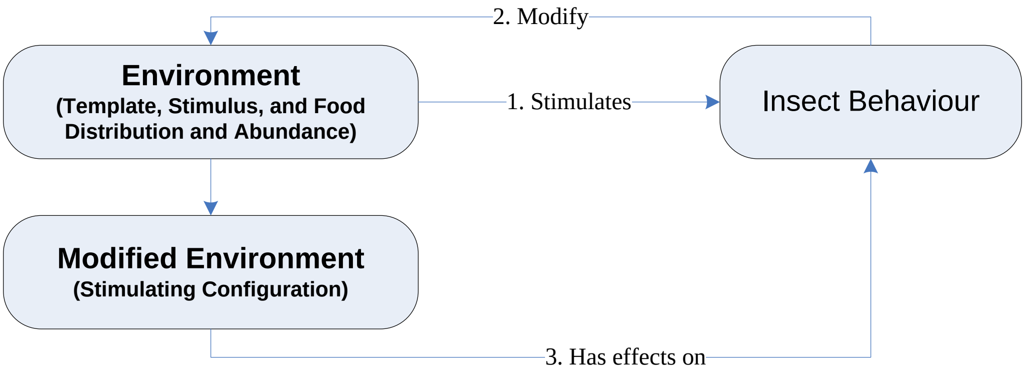
[3] In biology, template refers to "Any molecule that acts as a pattern for the synthesis of a new molecule" (1997). Oxford Concise Colour Science Dictionary, Oxford University Press.

be deposited each time. The model shows that the random walk of termites, the stimulation of the termite queen, and the diffusion of the pheromone determine the structures of the royal chamber.

Similar to the pheromone in the ant foraging behaviour, the template is locally observable and accessible to termites. It defines the context and structure of construction and initialises self-organised building activities of termites. The combination of template and self-organisation exhibits snowball effect, and at the same time produces a perfectly predictable pattern of termite behaviour.

Moreover, a template represents the context and structure of the building environment and is used as shared resources by all termites. It specifies the rules of construction behaviour and stimulates self-organised activities of termites. Hence, through the template, the building environment influences the construction behaviour of termites. As the template structure, the termite queen influences the construction behaviour of termites by initially defining the size and shape of the nest structure. Termites modify the nest structure through continually depositing soil pellets around the termite queen. Hence, the construction behaviour is taking place in a feedback loop of modifying the nest structure and responding to the modified nest structure. Figure 2.3 illustrates the relationships of the environment and insect behaviour.

Figure 2.3 Relationships between the Environment and Insect Behaviour



To insects, the natural environment is represented by the template structure for construction, the modified context and structure delivered via stimuli, as well as food distribution and abundance. Such kind of relationships not only exists in the nest construction of termites, but is also observed in other emergent behaviour of social insects. For instance, the pheromone-laying and pheromone-following activities of ants show the same relationship between the search environment and the ant foraging behaviour. Table 2.1 outlines the examples of the influential relationships between the environment and the emergent behaviour of social insects. Ants forage to search for food sources, thus the food distribution and abundance represent the search environment for ants. The structures of ant foraging patterns vary depending on the food distribution and abundance. The nest construction of termites is initiated by building up the royal chamber for the termite queen. Thus the size and shape of termite queen represent the building environment and are used as a template by termites. This template represents the context and structure of the building environment, which defines the context and structure of construction behaviour for termites.

Table 2.1 The Environment and the Emergent Behaviour of Social Insects

Environment	Insect Behaviour
The Search Environment	Ant Foraging Behaviour
Represented by Food Distribution and Abundance	Influences the structures of foraging patterns
The Building Environment	The Construction of Royal Chamber
Represented by Template: the size and shape of termite queen	Defines the context and structure of construction behaviour

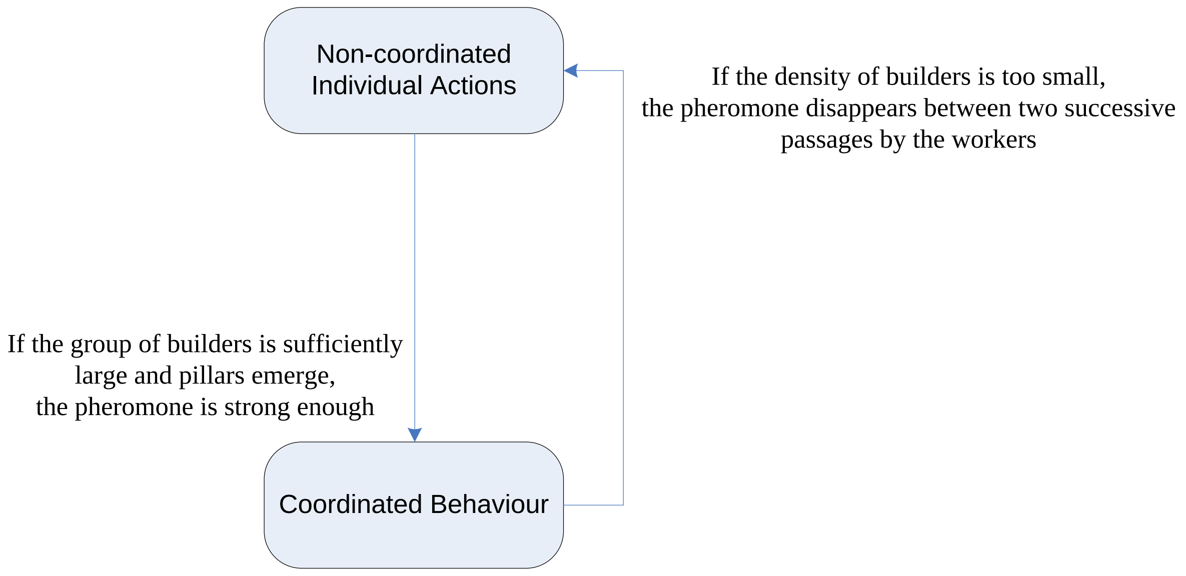
For ants and termites, the environment refers to the task environment that they operate in – the search environment for ant foraging behaviour and the building environment for termites. Therefore, the author considers the task environment as influential in determining the context and structure of insect behaviour. The influence of task environment is represented by influential elements, such as the food distribution and abundance for the search environment, and the template for the building environment.

2.1.2.2 Stimulus

In addition to template and food distribution and abundance, stimulus is another element related to the environment. In biology, stimulus is defined as “Any change in the external and internal environment of an organism that provokes a physiological or behavioural response in the organism” (1997). It is observed in ant foraging behaviour and nest construction of termites, in which it is used to describe and deliver the dynamic context and structure of the environment. In ant foraging behaviour, there are two types of "stimulus" – food smell/scent and pheromone (Eric Bonabeau 1999). Ants search for food sources and respond to the strength of food scent with various activities. The pheromone concentration also affects ant behaviour in a way that, the stronger the pheromone becomes, the more ants are recruited (reinforced) to the pheromone-marked path(s). In the nest construction of termites, stimulus is generated from the individual actions of termites and appears to be the modification of the building environment (Downing and Jeanne 1988). As described in (Eric Bonabeau 1999), "a stimulus is transformed into another, qualitatively different, stimulus under the action of an insect". Beginning with the construction template, termites modify the nest structure each time they deposit soil pellets. The processes of nest construction are continuously stimulating configuration of the template structure.

Furthermore, Marco Dorigo et al. (Marco Dorigo 2000) describe the nest construction behaviour as two successive phases – non-coordinated individual activities and coordinated behaviour. Pheromone enables coordinated behaviour to emerge from non-coordinated, simple rule-based individual activities. Constructed according to their description (Marco Dorigo 2000), Figure 2.4 illustrates the emergent processes of the coordinated behaviour, which is emerged from non-coordinated individual activities when the pheromone aggregates to a sufficiently strong level.

Figure 2.4 The Role of Pheromone in Nest Construction of Termites



The density of builders and the volume of pillars deposited develop positive pheromone that stimulates more builders to converge and more pillars to be deposited. Hence, builders communicate through the change of pheromone. Pheromone in the form of builder density and pillar deposition represent the external environment to individual insects. By reacting to the changes of builder density and the volume of pillars deposited, individual builders are responding to the influences of environment. Here pheromone works as stimulus, which is task-associated and mediates stigmergic communications between insect workers.

Stimulus and stigmergic communications belong to the phenomenon of stigmergy. The next section explains stigmergic communications in detail.

2.1.2.3 Stigmergic Communications

Stigmergic communications refer to indirect interactions between insects while carrying out self-organised behaviour. Compared to direct interactions, indirect interactions occur between two individual insects "when one of them modifies the environment and the other responds to the new environment at a later time" (Eric Bonabeau 1999). Indirect communications of such kind is an example of stigmergy (Theraulaz and Bonabeau 1999).

(James Kennedy and Russell C Eberhart 2001) define stigmergy as "communication by altering the state of the environment in a way that will affect the behaviour of others for whom the environment is a stimulus".

In addition to template and self-organisation, stigmergy reflects the influences of environment through a set of stimulating configurations. These configurations appear to be the modifications of environment. By modifying the environment and responding to the modification of the environment, insects are able to self-organise their activities. The change of environment structure and context mediates insect behaviour, and ultimately enables the coordinated pattern to emerge. The indirect communication of this kind is described as the phenomenon of stigmergy. Stigmergy is considered to represent another influential role of the environment.

Stigmergy was first introduced by (Grasse 1959; Grasse 1984) in his study of the nest construction of termites. Grassé explained how termites coordinate and regulate their activities based on the nest structure. The size and shape of termite queen are used as the initial stimulating configuration of the construction behaviour. An individual termite responds to the stimulating configuration by depositing a soil pellet at one location, and the modified nest structure triggers another termite to deposit a soil pellet at a different location. The initial stimulating configuration is continuously modified and termites respond to the new structure by depositing soil pellets in various quantities at different locations. As a result, termites self-organise their building activities by modifying and responding to modified nest structures.

The above description of the construction behaviour of termites indicates that the nest structure represents the building environment of termites, which indirectly mediates communications between individual termites. Such kind of indirect communications are taking place through the stimulating configuration, and thus are called stigmergic communications. Stigmergic communications describe the nest construction of termites to be stimulus-response behaviour, which enables self-organised activities to occur.

In brief, the author reviewed current studies for the two typical examples of insect behaviour – ant foraging behaviour and the nest construction of termites. The review describes that the environment contains essential information of the tasks to be achieved by social insects, which is considered to have both direct and indirect influences on the emergent behaviour of insects. There are two influential factors that represent the task environment. One is food distribution and abundance of the search environment that influence the structures of ant foraging patterns; the other is the template of the building environment that influences the context and structure of nest structure.

Furthermore, the emergent behaviour of insects presents characteristic properties of collective problem-solving. Ants lay and follow pheromone trails to forage on the paths between food sources and the colony. Termites are stimulated by the size and shape of termite queen to build up the royal chamber and the entire nest. Pheromone and template are the two stimulating factors that trigger cooperative behaviour to emerge. They are combined with two emergent mechanisms: self-organisation and stigmergic communication to generate effective collective problem-solving of social insects.

Therefore, predefined by a template construction pattern, the construction behaviour of social insects is initiated with random walk of individual insects, which is then continuously stimulated by building pheromone and stimulating configurations through stigmergic communications.

2.2 Collective Problem-Solving in Multi-Agent Systems

Having understood the collective problem-solving of social insects, this section describes collective problem-solving in multi-agent systems (MAS). The objective is to explore how the collective problem-solving of social insects can be transformed into exploitable solution methodologies in multi-agent systems.

Collective problem-solving offers novel perspectives of solving complex problems in the real world, in which uncertain conditions are present all the time. (Bogatyreva and Shillerov 2006) presented a mathematical model to describe a "goal-directed, or programmed, behaviour, interacting with uncertainty of environment". As background knowledge to their model, they analysed the two concepts related to the systems that are "self-organised, self-dependent, self-adapted and self-regulating". The first main concept describes a top-down approach with hierarchical control (Johnson 2001). The other concept is a bottom-up approach that is constructed on the understanding of emergent phenomenon of insects and human beings (James Kennedy and Russell C Eberhart 2001). Along with the research and development of distributed intelligent systems and bio-inspired problem-solving mechanisms, the emergent approaches are more popularly applied in collective problem-solving (De Wolf and Holvoet 2005). Table 2.2 identifies the five essential principles each for the top-down and bottom-up approaches of intelligent systems (Bogatyreva and Shillerov 2006).

Table 2.2 The Comparison of Two General Methodological Approaches to Complex System Study (Bogatyreva and Shillerov 2006)

Bottom-up	Top-down
1. More is different (emergent effect) 2. Agent ignorance is useful 3. No predefined order (random encounters) 4. Order from chaos as complexity grows 5. Local information leads to global wisdom	1. More is different (emergent effect) 2. Agent ignorance is harmful 3. Predefined order 4. Growing complexity increases chaos and leads to a new order 5. Global information leads to local wisdom

As shown above, both approaches present emergent phenomenon. The bottom-up approach is built on limited local knowledge of individual agents, while the top-down approach requires each agent to have a certain level of situational awareness; the bottom-up approach encourages solving complex problems from stochastic situations, while the top-down approach emphasises the deployment of orders; the bottom-up approach generates global solution for the problem from a collection of local information, while the top-down approach uses central intelligence to organise the problem-solving activities of individual agents. Based on the understanding of the two approaches of intelligent problem-solving, (Bogatyreva and Shillerov 2006) proposed a merged concept that has the following features:

- Apply hierarchical structures to support the interaction of agents instead of hierarchy of complexity;
- Emphasise the independence of each agent;
- The rules of activities are designed with respect to emergent phenomenon;
- Being able to deploy certain level of predictability to the changes in a system;

- Agents have global perspectives of the problem while acting locally.

The merged methodological framework intends to maximise the beneficial components of the bottom-up and top-down approaches with the aim to achieve controllable self-organisation and predictable adaptability. It emphasises the value of global knowledge in the processes of problem-solving, and indicates that predictable adaptability is achieved from the local actions of individual agents on the basis of global knowledge. The framework is further described in a mathematical model, which has potential applications in self-assembled mobile robots (Bogatyreva and Shillerov 2006).

2.2.1 Fundamental Concepts

Ant colonies and termites are considered to be natural agent systems (Van Dyke Parunak 1997), which have system behaviour (the goal of operations), individual responsibilities (rules of behaviour) and integration (cooperation emerged from individual activities). From the computational perspective, multi-agent system is considered to best represent characteristic properties of collective problem-solving. Therefore, the author uses multi-agent systems to further investigate potential solution methodologies of collective problem-solving. Below the author presents two fundamental concepts of multi-agent systems: the definition of agents, and the definition of multi-agent systems in various contexts.

As defined in (Ferber 1999), an agent is a physical or virtual entity with following features:

- Capable of acting in an environment;
- Can communicate directly with other agents;
- It is driven by individual objectives such as desirable operations or survival functions;
- It possesses resources of its own;

- Capable of perceiving its environment to a limited extent;
- Has none or a partial representation of its environment;
- Possess skills and can offer services;
- May reproduce itself;
- Makes use of available resources and skills and depending on its perception, its representations and the communications to accomplish the objectives.

A multi-agent system refers to a system comprising the following components (Ferber 1999):

- An environment, E, that is, a space which generally has a volume;
- A set of situated objects, O, which can be perceived, created, destroyed and modified by the agents;
- A group of agents, A, which are specific objects ($A \subseteq O$) representing the active entities of the system;
- An assembly of relations, R, which link objects (and thus agents) to each other;
- An assembly of operations, Op, making it possible for the agents (A) to perceive, produce, consume, transform and manipulate objects from O;
- "Laws of Universe", Operators with the task of representing the application of these operations and the reaction of the world to this attempt at modification.

Multi-agent systems provide a dynamic platform suitable for exploring potential solution methodologies of collective problem-solving. Such a platform is popularly used to design and model complex systems in which sets of random activities, multiple interactions, as well as goal-driven operations are taking place at all times. Ant colonies and termites are considered natural agent systems, while multi-agent systems are similar systems that are composed of computerised agents. Therefore, the author investigates swarm-inspired solution methodologies of collective problem-solving in multi-agent systems.

2.2.2 The Environment

As stated above (Ferber 1999), the environment is the primary component of a multi-agent system. In the emergent behaviour of social insects, the environment presents influential characteristics that contribute to the mechanisms of collective problem-solving. In multi-agent systems, environment is considered as an exploitable component and possesses equally important roles and responsibilities in the collective problem-solving. The term "environment" is vague and has different meanings depending on different contexts. Next the author describes the concept of environment in various disciplines and main definitions from different research perspectives of multi-agent systems. The author then explains the roles and responsibilities of environment in multi-agent systems.

2.2.2.1 Define "Environment"

The term "environment" is defined in a variety of disciplines. Below are the definitions of environment in the literature.

1) Literature Definitions of the Environment

a. In ecology (1997),[4] environment is defined as:

"The physical, chemical, and biological conditions of the region in which an organism lives."

To social insects, the environment contains various information for survival, such as food, shelter and light. With constant changes of environmental conditions, insect swarms must be sufficiently robust in order to adapt into the changed environment conditions so that they are

[4] Oxford Concise Colour Science Dictionary, pp255

able to survive through generations. The process of adaptation reflects the influences of environment in insect behaviour.

b. In multi-agent systems, the environment is defined in the following six major perspectives:

- Definition 1): Environment as a "generic environment program"

Russell and Norvig (Russell and Norvig 2003) describe environment using a "generic environment program". The program illustrates the basic relationship between agents and their environment. Agents perceive the environment and respond to it with actions, and then the result of agent actions updates the state of the environment;

- Definition 2): Environment described through its characteristics

Rao et al. (Rao, Georgeff et al. 1992) describe the typical characteristics of environment. The environment here is external to individual agents, which present different characteristics in a broad range of application domains;

- Definition 3): Environment = (State, Process)

Parunak (Van Dyke Parunak 1997) provides a point of view that environment is composed of states and processes. The environment state includes agents and objects that have a set of values, and the environment process indicates the environment dynamics of itself. Such a definition underlines the active nature of the environment;

- Definition 4): Environment defined to be "a space E, which generally has a volume"

Ferber (Ferber 1999) specifies environment in a container function and emphasizes the separation of agent actions from the reaction to those actions in the environment;

- Definition 5): Environment with structures and the nature of mediation

Odell et al. (Odell, Parunak et al. 2003) present a perspective in terms of the physical, communicative and social structures of the environment;

- Definition 6): Environment is “a first-class abstraction”

On the basis of the above definitions of the environment, Weyns et al. (Danny Weyns 2005; Weyns, Omicini et al. 2007) present their definition of environment as “a first-class abstraction” and a design block from the perspective of engineering environment:

“The environment is a first-class abstraction that provides the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources.”

Weyns et al.’s definition is focused on the context of multi-agent systems, which indicates that the structure and the dynamic nature of the environment define agent perception, rules of behaviour, communications and coordinated activities. They view environment as an independent and exploitable component from the software engineering’s perspective. The environment is an essential part of the world, which the agents exist in, interact with, and have access to. Also the environment mediates the interactions among agents and between agents and resources.

2) Application Environment

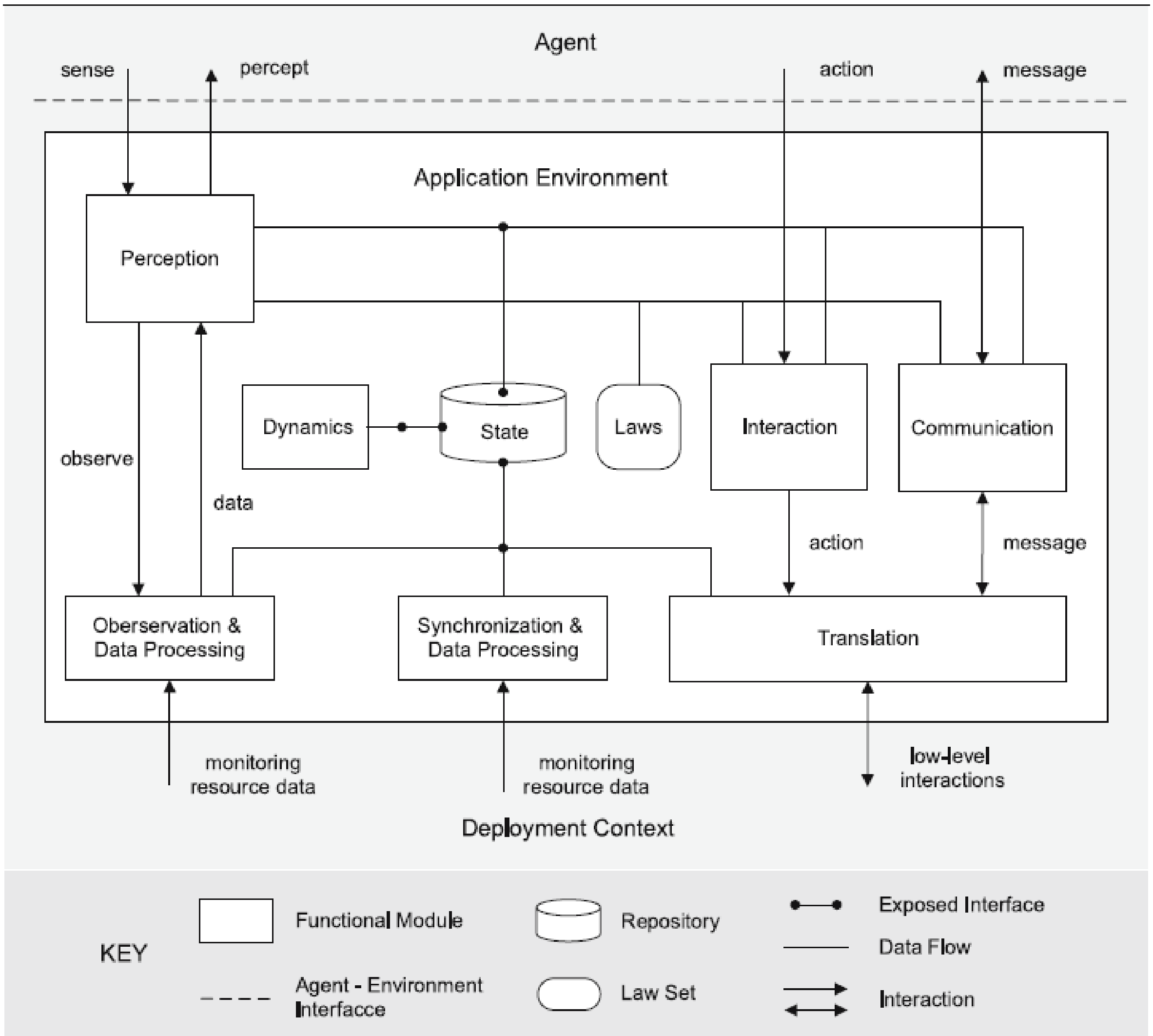
The term *application environment* was firstly introduced by (Weyns, Omicini et al. 2007), which describes the environment as an exploitable component to be designed for specific applications. Figure 2.5 shows a reference model for the environment. The reference model is composed of sets of modules that represent a range of functionalities of the environment.

These functionalities refer to the roles and responsibilities of environment in the collective problem-solving of multi-agent systems.

Similar to the task environment described in (Wooldridge 2002), the application environment defines the characteristics of the environment in which the group of agents are to operate in, and the set of tasks that agents are to implement in the environment. However, the task environment is constrained to agent-environment interaction, which focuses on the tasks to be achieved and the tasks to be maintained by agents. The application environment develops further on the basis of the task environment. It represents the specific problem to be solved by agents and identifies the problem specifications through the predefined context and structure. Agents are able to perceive and interact with the context and structure of the application environment with different actions. The context and structure of the application environment define the tasks to be carried out by agents and hence contribute to the design and specification of the agent behaviour, as well as emergent problem-solving mechanisms.

The environment is also referred in the present research context to be an application environment. In addition to the scope described above, the author emphasises that the application environment describes the characteristic properties of the problem. The description of the problem is used to define agent perception, rules of behaviour, communication and coordinated activities. It provides a generalised space to construct mechanisms for collective problem-solving in multi-agent systems. As a consequence, the application environment contributes to the design and specification of agent behaviour. Learning from the emergent behaviour of social insects, the author considers the individual rules of behaviour, communication and cooperation of agents to be the problem-solving mechanisms. The reference model in Figure 2.5 indicates that the roles and responsibilities possessed by the environment are influential to the agent behaviour.

Figure 2.5 A Reference Model for the Environment in Multi-Agent Systems (Weyns, Omicini et al. 2007)

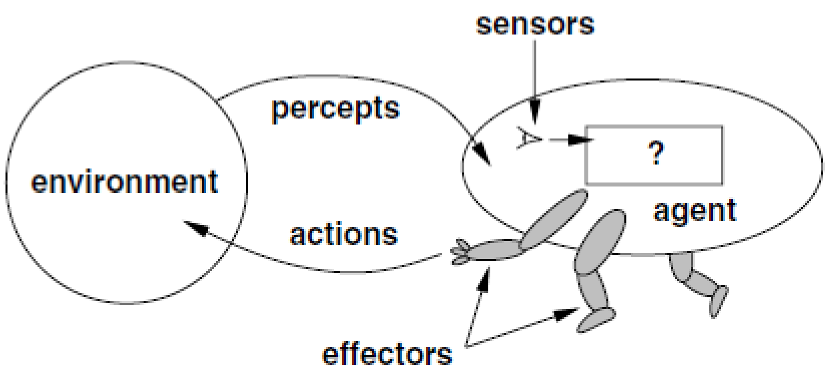


2.2.2.2 Roles and Responsibilities of Environment

The definitions of environment imply that there are potential capabilities offered by this component to accomplish effective and robust collective problem-solving. (Danny Weyns 2005) has presented a detailed survey on environments in multi-agent systems. The survey discussed important research works on environments and identified two key concerns for environments in multi-agent systems: the structure of the environment and the activities in the environment. The survey overviewed some key definitions of the environment in the form of general models (Ferber 1999; Odell, Parunak et al. 2003; Russell and Norvig 2003), communication infrastructure between agents and the environment, models for indirect interactions (e.g. stigmergic communications (Paul VALCKENAERS 2001)), environment used as an organisational layer, as well as environments in agent-oriented methodologies.

Figure 2.6 illustrates the agent interaction with the environment (Russell and Norvig 2003).

Figure 2.6 Interactions between the Agent and the Environment (Russell and Norvig 2003)



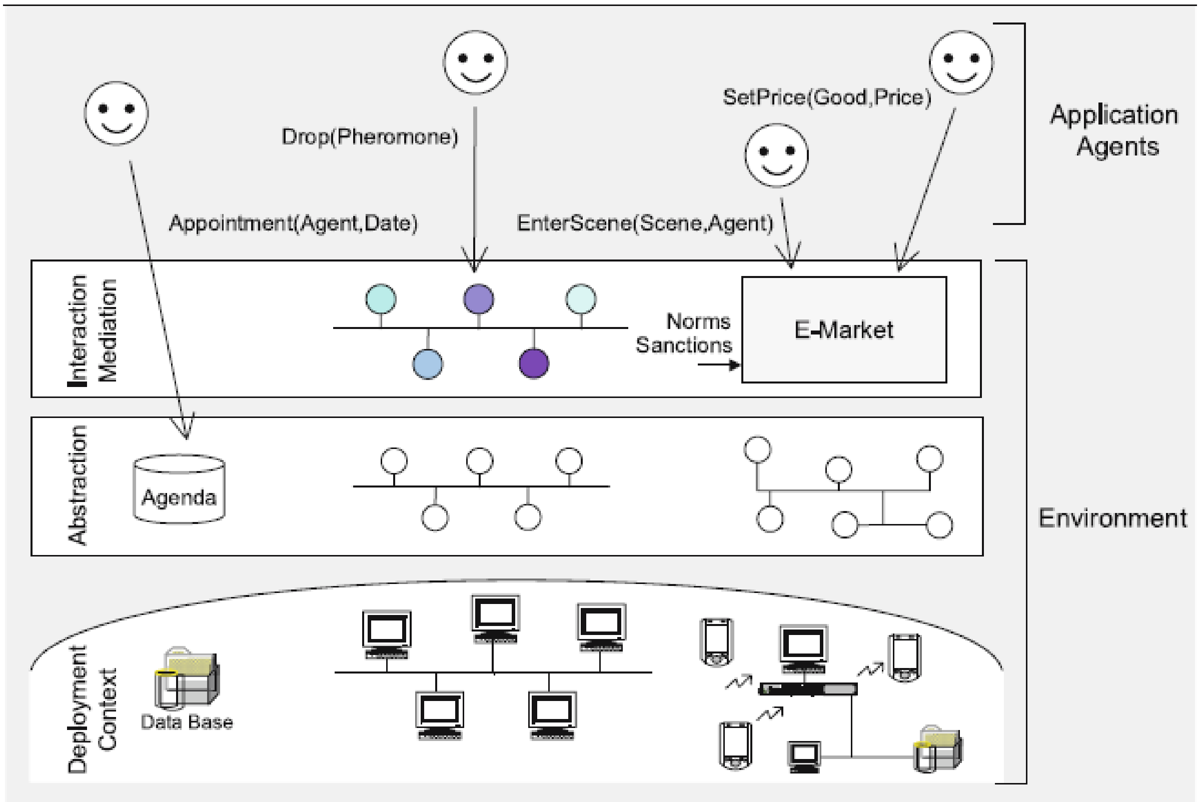
As illustrated in the figure above that, there are multi-way interactions between the agent and the environment. The agent operates in the environment, it perceives the information provided in the environment via the sensor system and responds with appropriate actions. Depending on the actual circumstances of the environment, the agent either operates as an individual or cooperates with other agents through communications. The communications of such kind are triggered by the dynamic conditions of the environment, which indicate that the environment provides a medium for coordination and thus is a means for indirect communications between agents. Based on the survey (Danny Weyns 2005), the environment is considered to contribute in different roles of collective problem-solving and to possess a number of important responsibilities (Weyns, Omicini et al. 2007). The roles and responsibilities of the environment are described as below.

The roles of environment highlight the interrelationship between the agent's activities and the environment that the agent is situated in. The environment is described to have following roles (Weyns, Omicini et al. 2007):

- An "external world" to agents;
- A medium for coordination;
- A means for communication;
- An explicit architectural abstraction with specific responsibilities that differ from agent responsibilities.

Figure 2.7 provides an example of the role of environment as a means for communication (Weyns, Omicini et al. 2007). The example describes a scenario of electronic market, where the environment acts as a medium to allocate tasks and resources among agents. A set of application agents are assigned with different tasks and the environment provides an interface for these agents to interact with three levels of infrastructures.

Figure 2.7 The environment mediates the interaction with resources and among agents (Weyns, Omicini et al. 2007)



Having investigated the emergent behaviour of social insects and existing mechanisms of collective problem-solving in multi-agent systems, effective communication mechanisms offer essential basis for coordinated activities to occur between agents. The author also emphasises the mediation role of the environment in her research, because it supports effective and robust communications to occur among agents, and the coordinated activities of agents rely on effective communications. Therefore, as a medium for coordination and a means for communication, the author considers the environment has significant contributions to the strategies of collective problem-solving.

Furthermore, regarding the definition of “first-class abstraction”, the environment has the following responsibilities (Danny Weyns 2005; Weyns, Omicini et al. 2007):

- Structures the multi-agent systems, which defines rules for the multi-agent systems – based on the relationships between agents, resources and services;
- Embeds resources and services that are situated in a physical structure;
- Maintains dynamics that are independent from agent activities such as states, processes, and conditions. Typical examples include the evaporation, aggregation, and diffusion of computational pheromones; and a self-managed field in a network;
- Locally observable and accessible to agents.

These responsibilities further describe the strategic contributions of the environment as an exploitable component in the collective problem-solving of multi-agent systems.

Briefly, the emergent behaviour of social insects and the collective problem-solving of multi-agent systems share common characteristic properties such as self-organisation, stigmergic communications (indirect interactions via the environment), template-defined behaviour, and stimulating elements such as pheromone. These characteristic properties indicate that the roles and responsibilities of the environment have influential contributions to the strategies and mechanisms of collective problem-solving. Section 2.3 presents a framework of collective problem-solving, describing the influences of the environment and how the influential factors are transformed to strategic mechanisms for collective problem-solving in multi-agent systems.

2.3 The Framework of Collective Problem-Solving

The literature review presented in Section 2.1 and Section 2.2 explained that the environment has influential contributions in the strategic mechanisms of collective problem-solving in social insects and multi-agent systems. Based on the literature definitions of the environment, and the knowledge of the environment roles and responsibilities, the author now presents a framework of collective problem-solving. The framework establishes an interdisciplinary knowledge of the environment proposing in a comprehensive form that the influences of environment contribute to the strategies of collective problem-solving.

First of all, the author explains her understanding of the environment in an interdisciplinary context, describing and analysing the influences of the environment that contribute to the strategies and mechanisms of collective problem-solving. The knowledge of environmental influences is established on the understanding of collective problem-solving in social insects and multi-agent systems. After that, the author illustrates the framework in Figure 2.8.

Table 2.3 below compares the definitions, the roles and responsibilities of environment in the collective problem-solving of social insects and multi-agent systems. In the literature, the environment is defined in biology and in multi-agent systems, where descriptions are given to the various aspects of the environment. The environment roles and responsibilities in multi-agent systems are reflected in the relationships between insects and the environment. Such a reflection indicates that the influences of the environment contribute to the collective problem-solving in an interdisciplinary context.

Table 2.3 The Environment in Insect Behaviour and In Multi-Agent Systems

Literature	In Social Insects	In Multi-Agent Systems
	The Environment is defined as: A physical structure that provides chemical and biological conditions of the region in which the colony requires to live. (1997)	1. Defined in various application-domains by describing the application-specific characteristics of environment (Rao, Georgeff et al. 1992)
		2. Environment = (State, Process) (Van Dyke Parunak 1997)
		3. Described using a container function and is defined as “a space E” with a volume (Ferber 1999)
		4. “a generic environment programme” illustrates the basic relationship between agents and their environment (Russell and Norvig 2003)
		5. Environment has structures and the nature of mediation (Odell, Parunak et al. 2003)
		6. Environment is an exploitable element in developing multi-agent systems (Danny Weyns 2005)
	Environment:	Environment Responsibilities:

	- Structures the insect colony - Provides resources and living conditions - Maintains dynamics through stimulating configurations - Is locally observable and accessible to insects - Defines rules for the colony	- Structures the MAS - Embeds resources and services - Maintains dynamics - Locally observable and accessible - Defines rules for the MAS
	Environment is:	Environment Roles:
	- A medium of coordination (through template, self-organisation and stimulating configurations) - A means for communication (through template and modified environment) - Task-defined environment	- A medium of coordination - A means for communication - Application-specific environment - Physical, communicative and social environment

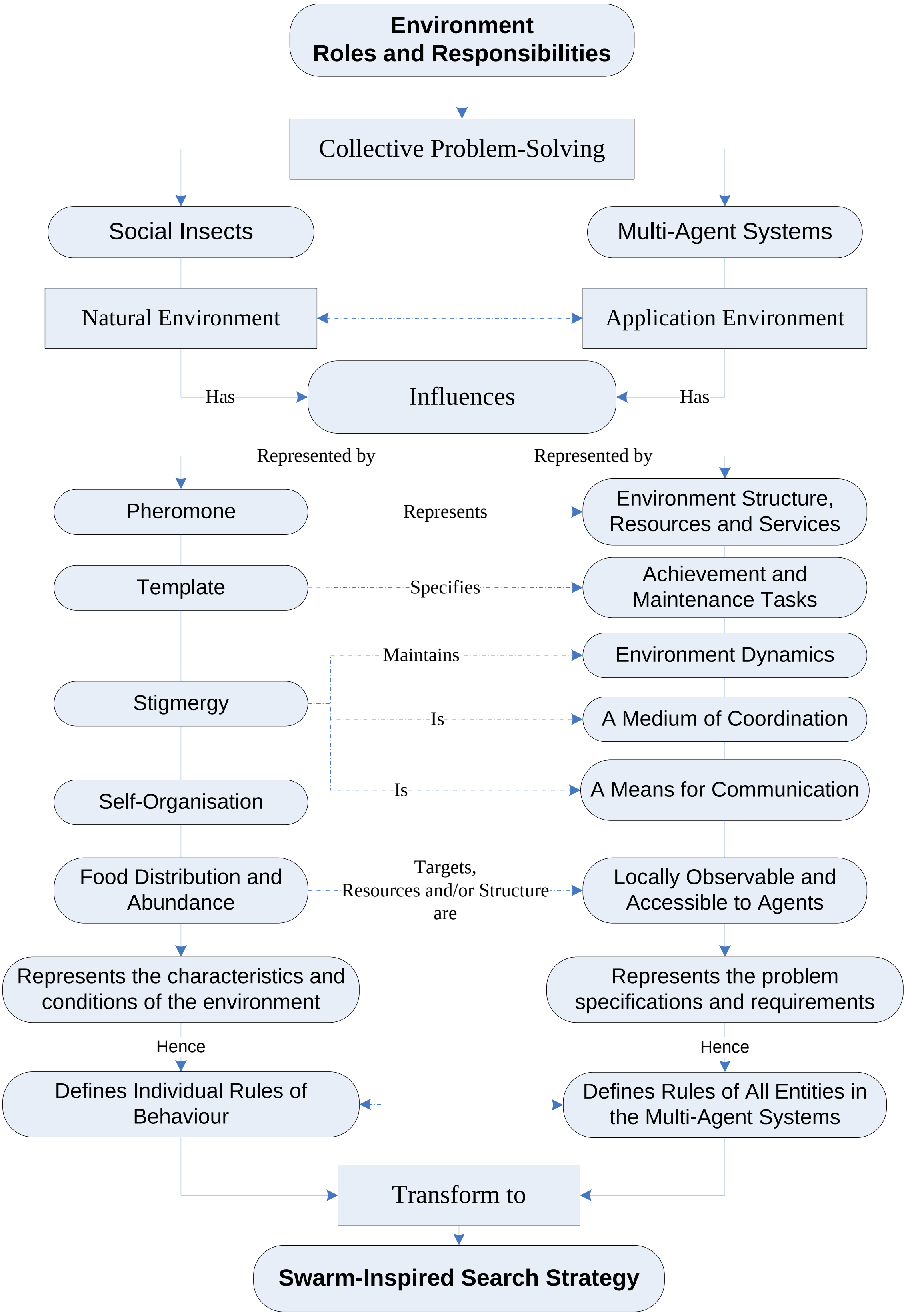
The environment is a physical structure. It provides resources and conditions for collective problem-solving. The environment is an essential element in enhancing effective problem-solving through the processes of self-organisation, emergent coordination and stigmergic communication. The environment has effects on the strategies and mechanisms of problem-solving, which are presented to be: application-specific characteristic properties, stigmergic communications via the modifications of environment, as well as the effectiveness of interaction between insects/agents and the environment is altered by the nature of substrate. In particular, the environment responsibilities and roles outline the influences of environment on the individual and coordinated behaviour of social insects and multi-agent systems.

The 6 definitions represent the main understanding of environment in multi-agent systems over the last decade. None of them has officially defined the term, and instead, researchers intended to describe the various aspects of environment of their own interests. For instance, Rao et al. (Rao, Georgeff et al. 1992) describe the characteristic properties of the environment of different application domains; Parunak (Van Dyke Parunak 1997) underlines the dynamic nature of the environment by assigning values to its states and processes; Ferber (Ferber 1999) uses a container function to specify the space and volume of an environment; Russell and Norvig (Russell and Norvig 2003) present the interactive relationship between agents and the environment; Odell et al. (Odell, Parunak et al. 2003) are also interested in the structure of environment and describe it to be physical, communicative and social.

In multi-agent systems, the knowledge of environment has been systematically developed. However, the inspiration of insect colonies in agent-based systems, especially the roles of environment in collective problem-solving, has been rather implicitly recognised. Because of its involvement in enabling flexible and robust performance of emergent systems, it is important to present the environment explicitly regarding its influential roles in collective problem-solving. To accomplish this, a framework of collective problem-solving is constructed to present the interdisciplinary knowledge of the influences of environment in social insects and multi-agent systems.

Figure 2.8 shows the framework of collective problem-solving, which describes the interdisciplinary knowledge of the environment through illustrating its characteristic properties and related mechanisms of problem-solving, in the emergent behaviour of social insects and the collective problem-solving of multi-agent systems. This framework provides a theoretical ground for the research work presented here on cooperative search of Unmanned Aerial Vehicles (UAVs). Learning from the problem-solving mechanisms of social insects and related applications in multi-agent systems, a set of swarm-inspired mechanisms of collective problem-solving are designed and deployed to accomplish the cooperative search problem of UAVs.

Figure 2.8 The Framework of Collective Problem-Solving



The review so far has explored a series of similarities between insects and multi-agent systems in collective problem-solving, however, the two also differ from each other in some aspects. For instance, it is well-known that individual ants and termites are very basic and simple beings. The idea of mimicking their behaviour in multi-agent systems is to solve complex problems through the cooperation and coordination of simple individuals with limited capabilities (Van Dyke Parunak 1997). However, real-time applications of multi-agent systems have encountered concerns regarding the level of intelligence each individual agent should be designed with. Regarding their relationships with the environment, agents can accomplish problem-solving processes with either cognitions or reactions (Ferber 1999). Cognitive agents have prior knowledge that enables them to solve certain problems by themselves and to also anticipate future events intentionally; whereas reactive agents simply respond to the modifications of the environment and have no prior knowledge of the problem. The former exhibit higher level of intelligence that demands more comprehensive design of the multi-agent system, and the latter intend to solve problems through emergent phenomenon.

Despite the emergent behavioural patterns, individual insects present synthesised characteristics of cognitive agents and reactive agents. Real-world applications consist of sophisticated tasks that cannot be accomplished by purely cognitive agents or purely reactive agents. Therefore, it is important to identify a synthesised form of cognition and reaction for each agent according to the goals and objectives of different problems. A variety of studies have been carried out to investigate design methodologies and infrastructures that fulfil the demands of solving complex problems (Parunak, Brueckner et al. 2003; Sierra, Rodriguez-Aguilar et al. 2004; Park and Sugumaran 2005; Moya and Tolk 2007; Weyns, Omicini et al. 2007; Daneshfar and Bevrani 2009).

Chapter 3. Multi-Robot Systems and Multi-Agent Systems

In Chapter 2 an in-depth literature review of collective problem-solving was presented, describing the research and development of emergent paradigms in social insects and multi-agent systems. There are two main application areas where such paradigms have attracted considerable research interest. One is in the use of UAVs to search for mobile targets; this is the specific subject area for this research and is reviewed in the next chapter. The other is in the area of multiple mobile robotics collectively carrying out tasks, such as the cooperative search and rescue.

This chapter reviews key concepts and fundamental problems of multiple mobile robotics; constructs the taxonomy of multi-agent systems based on current research works in agent and agent-related subjects; and lastly describes recent deployment of agent-based methodologies in multi-robot systems.

3.1 Multiple Mobile Robotics

Extended from the study of single robot systems, multiple mobile robotics have mainly developed in the following application areas (Parker 2000): exploration and mapping, communication, search and rescue, object transport and manipulation, motion coordination, as well as reconfigurable robotics. In the application of search and rescue, for instance, the goal is to find and rescue victims. To achieve this goal, multiple robots need to accomplish these objectives: 1) to explore the unknown search area intending to locate victims; 2) to transport victims to safety. Hence, the search and rescue tasks require the team of mobile robots to explore the unknown environment, to detect intermittent signals emitted from the victims and to localise their coordinates. Important mechanisms of cooperation deployed in the application areas of multi-robot systems include self-organisation, stigmergy, communications and coordination, learning and adaptability.

3.1.1 Key Concepts in the Robotic Literature

A mobile robot is autonomous and physically independent, and a system of multiple mobile robotics is composed of dozens of identical mobile robots. The research and development of multiple mobile robotics have become a potential area of research for the collective problem-solving. The major objective of such research is to deliver effective and efficient performance of problem-solving through applying distributed intelligent mechanisms. Investigating in hardware and software, the robotic research aims to study group architecture, resource and/or task allocation, cooperation and coordination, learning and adaptation, as well as geometric problems. This section describes the key concepts of multiple mobile robotics, as well as main streams of research for the collective problem-solving.

3.1.1.1 Taxonomy of Swarm Robotics

There are two main streams of multiple mobile robotics. One is collective robotics, where a group of fully autonomous robots cooperate with each other to accomplish high-level tasks (Kube 1997; Kube and Bonabeau 2000; Dutta, Bogobowicz et al. 2004). Applications of collective robotics include cooperative transport (Labella, Dorigo et al. 2004), box-pushing

(Mataric, Nilsson et al. 1995; Zhang, Fang et al. 2005), and two-robot bar carrying task (Hara, Sasajima et al. 1998; Cheng, Fink et al. 2009). The other is metamorphic robotics, where the robots are semi-autonomous as they are physically connected to each other; a well-known example is Swarm-Bots project, which describes a novel distributed concept in robotics adapting swarm intelligence paradigm in multiple mobile robotics. Swarm-Bots refer to a team of metamorphic robotics called s-bot, which are homogeneous and physically embodied to carry out various tasks at the group level (Francesco, Giovanni et al. 2004; Marco, Vito et al. 2004; Mondada, Pettinaro et al. 2004).

Swarm-Bots project is a typical example of swarm robotics research. Swarm robotics are defined by (Dorigo and Sahin 2004) as below:

"Swarm robotics is the study of how a large number of relatively simple physically embodied agents can be designed such that a desired collective behaviour emerged from the local interactions among the agents and between the agents and the environment."

(Sahin 2005; Sahin, Spears et al. 2007) described the following characteristics of swarm robotics research:

- Individual robots are situated, physically connected and/or disconnected with each other, and are able to have physical interactions with the environment;
- Robots are homogeneous, both hardware and software;
- The individual capabilities of robots are limited, to enhance group accomplishment for the tasks;
- Individual robots are constrained to local interactions among them and with the environment.

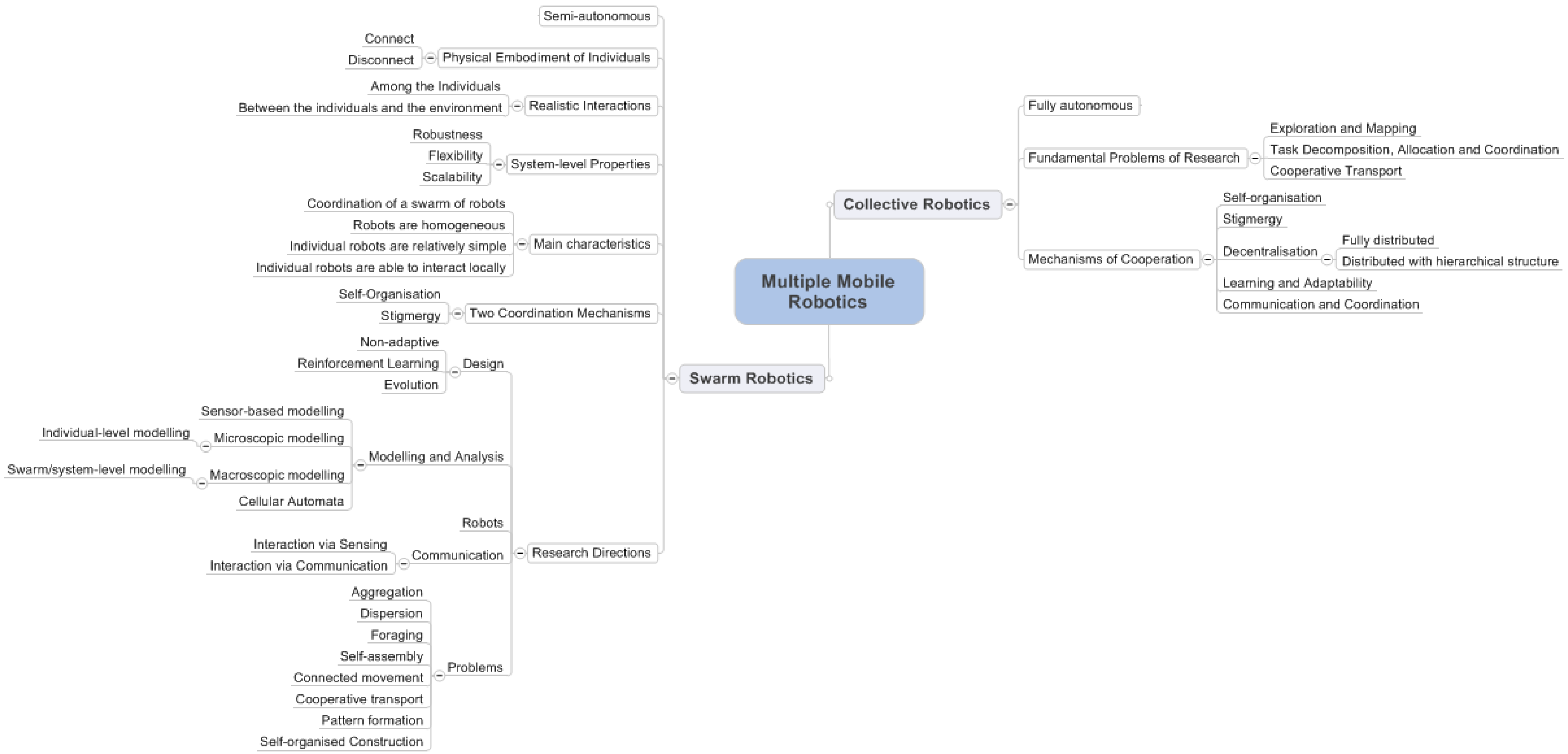
One of the main features of swarm robotics is the physical embodiment, which is engaged with problems such as aggregation (William, Alcherio et al. 2004; Garnier, Gautrais et al. 2009) and dispersion (McLurkin and Smith 2007), connected movement (Trianni and Dorigo 2005), self-assembly (Groß, Dorigo et al. 2006) and cooperative transport (Kube and Bonabeau 2000). The physical embodiment and physical connection with the environment are considered as the most important features that distinguish swarm robotics from collective robotics, where the latter consists of independent and fully autonomous individual robots. The homogeneous design and specification, limited individual capabilities and local communications are equally applicable to collective robotics.

Furthermore, collective robotics and swarm robotics also share common characteristics such as decentralisation, limited individual capabilities and local communications, cooperation and coordination through self-organisation and stigmergy, multiple interactions among individual robots and between robots and the environment. The two streams of robotic research complement each other and have various research works conducted on the same or similar problems, as well as problem-solving mechanisms; for instance, the cooperative transport, pattern formation, exploration and mapping. Figure 3.1 on the next page illustrates the two main streams of robotic research: collective robotics and swarm robotics.

The design and applications of multi-robotic systems make use of swarm-inspired mechanisms, such as a combination of self-organisation and stigmergic communications, to accomplish tasks. In the prey retrieval of collective robotics (Labella, Dorigo et al. 2004), the design of mechanisms is inspired by the ant foraging behaviour aiming to increase group capability. Related activities in the foraging processes of multiple robotics include efficient exploration of the search environment, communications among robots, as well as self-organised task allocation and coordination. Each robot of the group has a behaviour-based individual controller, and the overall operation is supported by distributed control algorithms.

Having understood the construction behaviour of social insects, researchers (Holland and Melhuish 1999; Gianluca, Domenico et al. 2006) deployed the model of self-assembly and related mechanisms to explore the design and applications of multiple mobile robotics. Examples include mimicking the nest construction of insects (Mondada, Guignard et al. 2003; Parker, Hong et al. 2003), exploration and mapping (Batalin and Sukhatme 2003; Ko, Stewart et al. 2003; Fox, Ko et al. 2006), pattern formation (Fredslund and Mataric 2001; Nouyan and Dorigo 2006), search and rescue (James S. Jennings 1997; Wolf, Brown et al. 2003; Vincent, Fox et al. 2008), communication networks (Dutta, Bogobowicz et al. 2004; Cazangi, Von Zuben et al. 2005), and so on.

Figure 3.1 Taxonomy of Multiple Mobile Robotics



3.1.1.2 Mechanisms of Cooperation

Like all other distributed intelligent systems, multiple robotics exhibit cooperative behaviour and intends to resolve complex problems using emergent approaches. The mechanisms such as self-organisation and stigmergic communications result in cooperation among robots, and such kind of cooperation describes issues regarding the task distribution, cooperative mechanisms, and also emergent performance of the multi-robot system. The cooperative behaviour in robotics literature is defined as below:

"Given some tasks specified by a designer, a multiple robot system displays cooperative behaviour if, due to some underlying mechanisms (i.e, the 'mechanism of cooperation'), there is an increase in the total utility of the system." (Cao, Fukunaga et al. 1997)

The mechanisms of cooperation take into account the following aspects:

- Group Architecture

"The group architecture of a cooperative robotic system provides the infrastructure upon which collective behaviour are implemented and determines the capabilities and limitations of the system." (Cao, Fukunaga et al. 1997)

The cooperative mechanisms of multiple robots define the group architecture to be decentralised. The decentralised approaches can be further divided into fully distributed architectures in which all robots have the same level of responsibilities, and hierarchical architectures in which robots have different levels of responsibilities in the cooperative behaviour. Also included in the architectural design are communication structures that refer to the interactions among individual robots via environment, via sensing, as well as via direct communications with each other. Decentralised group architectures offer promising features such as self-organised emergence and stigmergic communications that have significant contributions to collective problem-solving.

- Resource Conflict

A typical example of resource conflict in the cooperative robotic system is physical collision between robots. Applications related to resource conflict mainly refer to path planning and collision avoidance of multiple robots. A range of research has been conducted to explore optimal approaches of path planning while avoiding possible collisions.

- Geometric Problems

Geometric problems of cooperative robotic system include path planning and terrain exploration, mapping, generating and maintaining formations, as well as self-assembled patterns. Later research of mobile robotics makes use of decentralised approaches and a variety of swarm intelligent mechanisms have been investigated to accomplish above tasks. Some of the problems such as exploration and mapping, self-assembled patterns, as well as communications issues encountered in each of these problems have become fundamental subjects of research in multiple mobile robotics. Details related to the problem-solving approaches of these problems are provided in the next section.

- Reinforcement Learning

The mechanisms of cooperation also consider a form of reinforcement learning, which enables robots to learn relevant control parameters in order to improve their performance and to adapt to changes in the environment.

3.1.2 Fundamental Problems of Multiple Mobile Robotics

3.1.2.1 Coordinated Exploration and Mapping

The problem of exploring and mapping an unknown environment is a fundamental subject of multiple mobile robotics. The goal is to complete the exploration process in a minimum period of time and with increased mapping accuracy (Simmons, Apfelbaum et al. 2000; Burgard, Moors et al. 2005; Fox, Ko et al. 2006). The multi-robot exploration and mapping are generally studied and applied as an essential part of robotic operations. Related applications of the multi-robot exploration and mapping include: reconnaissance and surveillance (Saptharishi, Spence Oliver et al. 2002), search and rescue (Murphy 2004), mowing (Sahin and Guvenc 2007) and cleaning (Altshuler, Yanovski et al. 2009).

In the processes of exploration and mapping, each robot keeps a record of explored area – the local map of the environment and continually updates the mapping data while it is exploring different parts of the environment. These localised maps are then combined into a global map by a central agent. As the exploration process continues, the coverage of this global map is extended till the entire environment is completely navigated by robots. A hierarchical structure is applied here when the maps of individual robots are combined by a centralised agent. In addition, the central agent also manages a bidding scheme to assign different robots to explore different areas of the environment (Simmons, Apfelbaum et al. 2000). The central agent makes the decisions of assignment using current local maps obtained by individual robots. The task allocation of such kind coordinates robot behaviour, which minimises overlapping areas of exploration and, hence, enhance group efficiency of robots. The central agent is expected to be one of the robots, which coordinates the exploration and mapping tasks of the team of robots (Fox, Ko et al. 2006).

The distributed approach of exploration and mapping has been further developed to enhance effective coordination among multiple robots, to produce a shared map at the early stage of the exploration and to deal with limited communications between robots and the central agent (Fox, Ko et al. 2006). To improve the coordination of robots, the global map is not only produced as a result of the exploration, but also shared among robots during the process to provide real-time information of the explored parts of the environment. When exploring large area of environment, the communications among robots, between robots and the central agent are constrained and when failures occur, individual robots are expected to be able to carry on exploring on their own. The distributed exploration approach provides robots with certain level of independence to deal with the failure caused by limited communications. In this way the team of robots is sufficiently robust to adapt into ad hoc situations.

Furthermore, (Franchi, Freda et al. 2007) presented a fully decentralised random strategy that uses a multi-robot Sensor-based Random Trees (SRT) method to record and maintain the local maps of explored area with an associated safe region. The multi-robot SRT method is a multiple version of the single-robot SRT method, which provides a collection of data structures to keep track of explored area of multiple robots. Robots are able to conveniently access the data structure of such kind to make use of the information for their further exploration and mapping tasks. With respect to the nature of decentralisation, the randomised strategy is accomplished with cooperation to increase efficiency, coordination to avoid conflicts, as well as communications to support effective cooperation and coordination among robots. Different from the hierarchical structure applied via a central agent in (Simmons, Apfelbaum et al. 2000; Fox, Ko et al. 2006), the full decentralisation require no central management of tasks. The global map of the environment is generated through the multi-robot SRT method, and there is no task decomposition and/or allocation involved in the process. As a result, robots are able to operate effectively with limited communication range.

Moreover, the randomly generated data structures provide robots with reliable and up-to-date information of the explored area, which enable robots to be flexible and robust when dealing with individual robot malfunctions and communication failures.

3.1.2.2 Pattern Formations of Robotics

Pattern formation is the emergent result of local actions and multiple interactions among individual robots and between robots and the environment. Generating and maintaining a pattern formation is another popular subject of research in the multi-robot systems. The generation processes of pattern formations vary between collective robotics and swarm/metamorphic robotics, as the former is composed of individually independent robots and the latter is a team of physically connected robots. Both are decentralised, and are emerged from individual behaviour of robots through swarm-inspired mechanisms of cooperation – self-organisation, stigmergy and multiple communications.

(Bayindir and Sahin 2007) refer pattern formation to be one of the problems of interest in swarm robotics research. Existing research indicate that the processes of generating pattern formations involve both individual level adaptation and group level adaptation (Bahceci, Soysal et al. 2003). Information received from local sensing and limited communications among individuals is used to generate an open-loop style of formation of robots (Fredslund and Mataric 2002). Additionally, earlier research of robot formations learn from the flocking of birds, schooling of fish and ant foraging pattern, and construct goal-directed reactive control approach to generate and maintain different formation patterns (Balch and Arkin 1998).

The research works mentioned here exhibits some general requirements of collective problem-solving:

- The problem-solving approaches require effective cooperation of individual robots, and the group capabilities are emerged from simple individual behaviour of robots;
- Robots should be able to interact at multiple levels, both among individuals and with the environment;
- Acquiring and accessing real-time information on the environment is important to ensure effective and efficient accomplishment of multi-robot systems;
- The goal and goal-defined tasks can be exploited and programmed in a form of individual behaviour of robots, as well as mechanisms for cooperation and coordination to fulfil the problem requirements with respect to the nature of decentralised intelligent systems.

Briefly, in the context of collective problem-solving, multi-robot systems and multi-agent systems are two independent subjects but also share common characteristic properties in the design, modelling, communications, analytical studies and related problems. Next section presents the mutual contributions of the two subjects by firstly describing the taxonomy of multi-agent systems, and then explaining how agent-based methodologies are deployed in the construction and development of multi-robot systems.

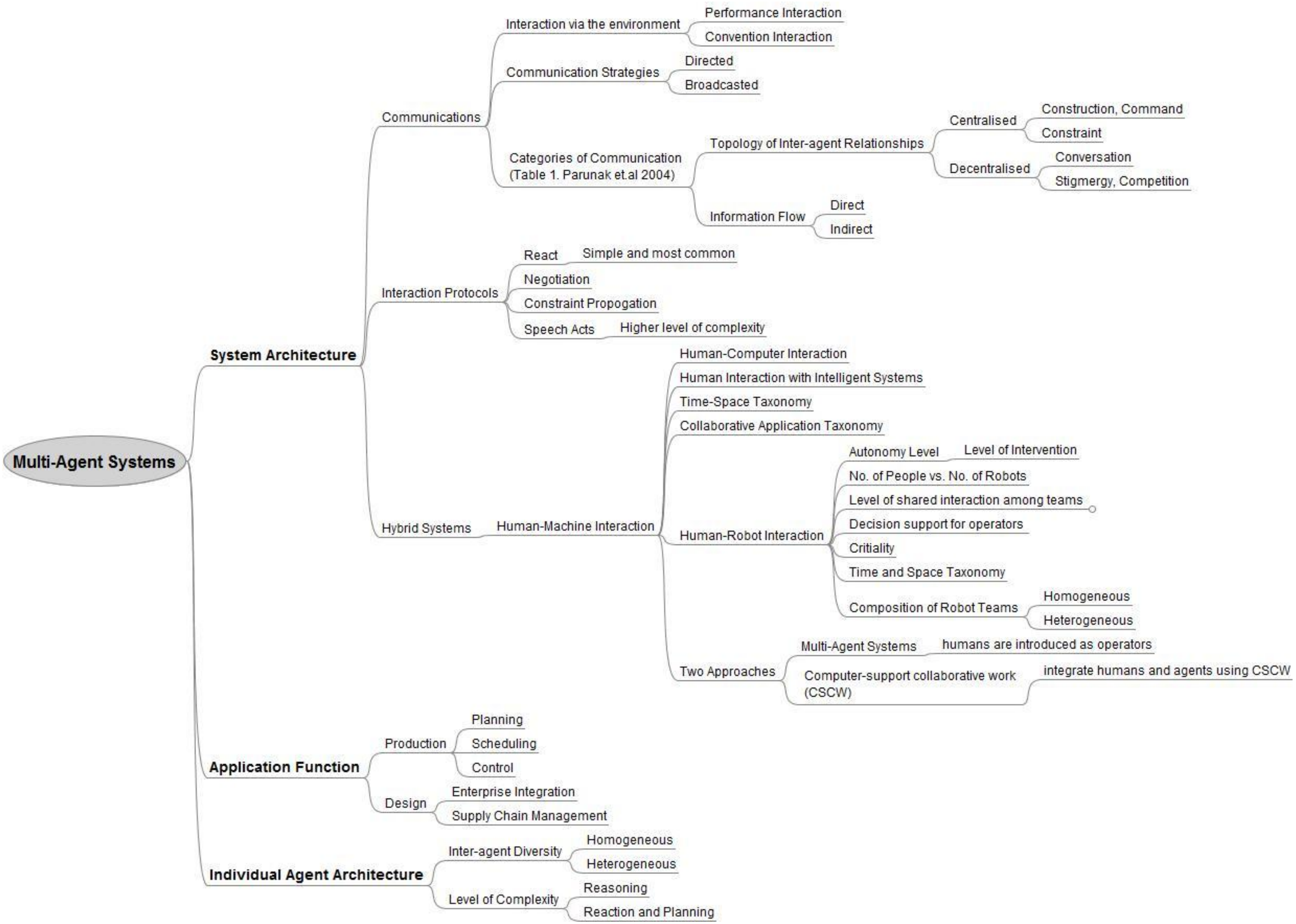
3.2 Multi-Robot Systems and Multi-Agent Systems

As presented in Section 3.1, the design and applications of multi-robot systems deploy mechanisms of swarm-inspired emergent problem-solving and for instance, self-organised rules of behaviour, multiple interactions especially stigmergic communications, group capabilities of solving problems emerged from rather simple individual behaviour. The system of multiple mobile robotics also exhibits distinguishing characteristics of distributed intelligent systems such as decentralisation, homogeneous and/or heterogeneous individuals with limited individual knowledge, multiple interactions among individuals and with the environment, and so on. Therefore, a system of multiple mobile robots is a typical example of multi-agent systems. A series of research have been conducted to apply agent-based methodologies in the design and development of multi-robot systems. This section firstly takes a look at the taxonomy of multi-agent systems from the application-oriented perspective, and then presents current applications of multi-agent systems in the design and development of multi-robot systems.

3.2.1 Taxonomy of Multi-Agent Systems

Figure 3.2 illustrates the taxonomy of multi-agent systems. The taxonomy is constructed based on previous works on the understanding of agent and agent-based intelligent systems (Moya and Tolk 2007), an organisational view of multi-agent systems (Ferber, Gutknecht et al. 2003), as well as a classification for human-machine and human-robot interaction (Yanco and Drury 2002; Parunak, Brueckner et al. 2003).

Figure 3.2 Taxonomy of Multi-Agent Systems


Multi-Agent Systems
System Architecture
Communications
Interaction via the environment
Performance Interaction
Convention Interaction
Communication Strategies
Directed
Broadcasted
Categories of Communication (Table 1. Parunak et.al 2004)
Topology of Inter-agent Relationships
Centralised
Construction, Command
Constraint
Decentralised
Conversation
Stigmergy, Competition
Information Flow
Direct
Indirect
Interaction Protocols
React
Simple and most common
Negotiation
Constraint Propogation
Speech Acts
Higher level of complexity
Hybrid Systems
Human-Machine Interaction
Human-Computer Interaction
Human Interaction with Intelligent Systems
Time-Space Taxonomy
Collaborative Application Taxonomy
Human-Robot Interaction
Autonomy Level
Level of Intervention
No. of People vs. No. of Robots
Level of shared interaction among teams
Decision support for operators
Critiality
Time and Space Taxonomy
Composition of Robot Teams
Homogeneous
Heterogeneous
Two Approaches
Multi-Agent Systems
humans are introduced as operators
Computer-support collaborative work (CSCW)
integrate humans and agents using CSCW
Application Function
Production
Planning
Scheduling
Control
Design
Enterprise Integration
Supply Chain Management
Individual Agent Architecture
Inter-agent Diversity
Homogeneous
Heterogeneous
Level of Complexity
Reasoning
Reaction and Planning

As a decentralised intelligent system, multi-agent systems offer a bottom-up approach that deals with complexity and carry out real time applications in a pattern of emergence and aggregation, coordination and cooperation. Earlier applications of intelligent agents and multi-agent systems are popularly seen in manufacturing design. For instance, in production planning, scheduling and control; as well as enterprise integration and supply chain management (Parunak 1996; Shen and Norrie 1999). The design and specification of individual agents consider 1) the inter-agent diversity – agents are either homogeneous or heterogeneous; 2) the level of complexity is defined through the reasoning, reaction and planning capabilities. In addition, (Parunak 1996) indicated three main aspects of the system architecture: multiple interactions and communications, interaction protocols, and human-machine interactive systems. Each aspect has then been further developed in a series of research work including the categories of communication (Parunak, Brueckner et al. 2003) and detailed description and analysis of human-robot interaction (Yanco and Drury 2002). The multi-agent systems (MAS) are here classified to provide a coherent presentation regarding their applications in intelligent manufacturing, architectural perspectives both at individual level and at system level.

3.2.2 Deployment of Multi-Agent Systems

(Daneshfar and Bevrani 2009) reviewed a series of concepts and technical aspects of multi-agent systems (MAS) in control engineering applications. Their survey shows that MAS are mainly used in two ways. First of all, it is used to construct both hardware and software systems that are able to provide sufficient robustness, flexibility, and extensibility in order to fulfil the requirements of real-time engineering applications (Bond and Gasser 1988; Parunak 1996). The other way is that the multi-agent system (MAS) itself can be used as a modelling approach (Da Silva and De Lucena 2004; Richard Hill, Simon Polovina et al. 2005). The exploitations of MAS require autonomy of agents, adaptability, concurrent task processing capabilities, multiple communication architectures, platform for distributed systems, mobility to enable agents to travel between platforms and environments, as well as fault tolerance through providing redundancy to the system (Oprea 2004).

3.2.2.1 MAS used for System Construction

The majority of design and applications of multi-agent systems are deployed as a bottom-up approach, while there are also researches being carried out to explore multi-agent systems in a top-down approach with hierarchical structures. Both approaches intend to enhance problem-solving performance of the agent-based system in a variety of engineering and other applications. The main applications of MAS in industry include the management and control of manufacturing systems, congestion control systems such as traffic control (Balbo and Pinson 2001) and telecommunication networks, distributed control deployed to manage power engineering (McArthur, Davidson et al. 2007) and environmental control systems

(Schreinemachers, Berger et al. 2007), and also the construction and deployment of multi-robot systems (Burgard 2008; Cervera 2008).

The agent-based technologies differ from traditional control technologies in the way that multi-agent systems provide decentralised, emergent and concurrent approaches to improve problem-solving capabilities while minimising individual complexity of agents, decomposing hardware requirements and reducing overall computational complexity. Table 3.1 presents the key aspects of the two approaches and outlines the advantages and disadvantages for each of them (Parunak 1996).

Table 3.1 Agent-Based Technologies vs. Conventional Technologies (Parunak 1996)

Issue	Autonomous Agents	Conventional
Model	Economics, Biology	Military
Issues favoring conventional systems		
Theoretical optima?	No	Yes
Level of prediction	Aggregate	Individual
Computational Stability	Low	High
Issues favoring autonomous agents		
Match to reality	High	Low
Requires central data?	No	Yes
Response to change	Robust	Fragile
System reconfigurability	Easy	Hard
Nature of software	Short, simple	Lengthy, complex
Time required to schedule	Real time	Slow

When the centralised control technologies have encountered constraints, and integrated computational pattern is no longer able to efficiently respond to real-time requirements of industrial applications, multi-agent systems offer a novel approach to problem-solving. Each agent is designed as a problem-solving unit, and the problem and/or task is decomposed and allocated to agents. Agents, either heterogeneous or homogeneous, accomplish assigned tasks both as an individual and as a team through multiple interactions and coordinated mechanisms. The decentralised approach respects characteristic properties of real world applications, which is easy to design and reconfigure, robust in responding to ad hoc circumstances. Also, data collection and processing are ongoing and decentralised, software programmes are simple to write and maintain.

3.2.2.2 MAS as a Modelling Approach

The aim of presenting a comprehensive perspective of multi-agent systems and multi-robot systems is to show the role of multi-agent systems in application-focused industrial problems, and how the two agent-based problem-solving approaches are inter-related. As illustrated in Figure 3.2, one exploitation of multi-agent systems is as a modelling approach. A system of intelligent agents is used to represent different entities and entity-related relationships in real world applications. In addition to this, agent-based modelling methodologies have been widely deployed in collective problem-solving. Having investigated existing research framework of both multi-agent systems and multi-robot systems, multi-agent systems as a modelling approach in robotic research is briefly explored.

A series of research have been undertaken to apply agent-oriented modelling methodologies to multiple robotics applications, including the Multi-Agent Systems Engineering (MaSE) methodology to design a team of autonomous and heterogeneous robots for search and rescue missions (DeLoach, Matson et al. 2003). Different from most system development of robotic applications, MaSE assigns a hierarchical structure to represent the roles of heterogeneous robots and emphasises high level cooperative activities of robots. Such a top-down approach enhances the maintenance and modification of cooperative robotic systems. (Menezes 2004) proposed the use of agent-based modelling language StarLogo[5] to simulate the collective behaviour of multiple robotics. Agent-based modelling methodologies intend to provide abstractions, as well as sufficient details at both individual level and group level of multi-robot systems. (Chia-How, Kai-Tai et al. 2005) presented agent-based robot control architecture (ARC), which is used to coordinate and control the cooperative target searching of two mobile robots. Additionally, agent-oriented software patterns are deployed as an integrated method to increase both generality and application-focused usability of multi-robotic systems (Chella, Cossentino et al. 2010).

To summarise, multi-agent systems and multi-robot systems are the two main subjects of distributed intelligence and have been explored intensively as approaches to collective problem-solving. Existing taxonomies of multi-robot systems and multi-agent systems in the literature have been used as the basis of presenting a coherent classification for each of the two approaches.

[5] StarLogo and NetLogo are amongst the most popular agent-based modelling languages, which provide a simple yet powerful development environment to represent various emergent paradigms and to build agent-oriented simulation programmes.

Chapter 4. The Cooperative Search of UAVs

Chapter 4 presents a specific literature review for the cooperative search of UAVs. The objective is to investigate existing research regarding the mechanisms used to deliver effective cooperative search of UAVs. First of all, the current approaches of the cooperative search of UAVs are reviewed, and then a set of swarm-inspired cooperative search mechanisms are proposed on the basis of the framework of collective problem-solving. In Chapter 2, the framework as an interdisciplinary knowledge of collective problem-solving was described. Here the framework is further studied through designing and deploying a set of swarm-inspired search mechanisms with the aim to accomplish the problem of cooperative search of UAVs.

The cooperative search problem has been studied in-depth with various distributed intelligent systems, such as collective robotics (James S. Jennings 1997; Burgard, Moors et al. 2005; Fox, Ko et al. 2006; Franchi, Freda et al. 2007), multi-agent systems (Van Dyke Parunak 1997; Teodorovic 2003; Sierra, Rodriguez-Aguilar et al. 2004; Dasgupta 2008), as well as in the case of a group of Unmanned Aerial Vehicles (UAVs) (Ryan, Zennaro et al. 2004; Doherty and Rudol 2007; Altshuler, Yanovsky et al. 2008). The cooperative search of UAVs has attracted continuous research interests over time, and has become one of the major subjects in the research area of the collective problem-solving. The system of UAVs presents typical characteristics of decentralised systems and is considered to be an ideal platform for studying the cooperative search problem.

The problem scenario for the cooperative search of UAVs has been well-established, which deploys a group of identical UAVs to search, detect and locate multiple non-stationary targets through UAV cooperation (Ablavsky and Snorrason 2000; Passino, Polycarpou et al. 2002;

Vincent and Rubin 2004). The goal is to locate the target positions with maximum accuracy and minimum costs. A series of research have been conducted to achieve such a goal. For instance, the generation and maintenance of flight formations of multiple UAVs (Ryan, Zennaro et al. 2004; Vincent and Rubin 2004; Altshuler, Yanovsky et al. 2008), path planning (Ablavsky and Snorrason 2000; Marios M. Polycarpou 2001; Geyer 2008), exploration and mapping (Fox, Ko et al. 2006), wireless communications between UAVs (Morris, Mullins et al. 2006), sensing and image processing techniques (Sevcik, Green et al. 2005; Doherty and Rudol 2007), and so on.

Path planning has been of research interests over years, which aiming to maximise the search performance of UAVs by optimising the search path of UAVs and efficiently allocating the search tasks (Ablavsky and Snorrason 2000). Optimal search path is essential for UAVs to avoid obstacles in complex terrains such as urban environment (Geyer 2008). Also, the process of recording the trajectory history and calculating an optimal search path provides real-time information of the search environment (Marios M. Polycarpou 2001). Thus UAVs are able to "learn" the information on environment and use this to guide their search activities.

Exploration and mapping are mainly referring to autonomous robotics, which is considered to be fundamental in the problem-solving infrastructure for the cooperative search of UAVs. It aims to investigate different solutions of efficient exploration and mapping of unknown environments. (Fox, Ko et al. 2006) presented a distributed approach that allows a team of robotics to explore an unknown environment from different locations. Initially each robot explores the environment independently and obtains the sensor data using a particle filter. They exchange the information of their current locations and update the shared search maps as the exploration is taking place.

Wireless communications show impacts on cooperative search algorithms for multiple UAVs. The research conducted by (Morris, Mullins et al. 2006) suggest that the communication range and the number of UAVs have impact on the group capability to accomplish the search task. This study contributes to the 4-stage cooperative search infrastructure of UAVs (Pack and York 2005; Toussaint, De Lima et al. 2007), through focusing on the realistic communication constraints in the global search stage. The results show that a trade-off exists between different combinations of the communication range and the number of UAVs. Despite this, the global search is able to deliver reasonably effective performance when the realistic communication constraints are applied.

Sensing and image processing are one of the popular research areas of the cooperative search algorithms of multiple UAVs. The subjects of research varies from traditional sensing techniques such as thermal and colour (Doherty and Rudol 2007), infrared, sonar and vision (Sevcik, Green et al. 2005), to integrated lightweight sensor package (e.g. Very Large Scale Integrated (VLSI) techniques), bio-mimetic sensing, as well as wireless image acquisition and wireless mote localisation (Sevcik, Green et al. 2005).

In addition to the range of research areas above, some of the major research interests are focusing on the flight formations of UAVs. Different formation patterns of cooperative UAVs are considered to be an important technique of locating target positions, particularly in the research by (Vincent and Rubin 2004; Pack and York 2005; Toussaint, De Lima et al. 2007; York, Pack et al. 2007; Altshuler, Yanovsky et al. 2008; Pack, DeLima et al. 2009). This is also the focus of the research work presented in this thesis.

Section 4.1 presents an overview of the prior research of cooperative search strategies of UAVs, especially the flight formations of multiple UAVs. Section 4.2 proposes the swarm-inspired cooperative search approach, including the behaviour-based cooperative search strategies and the flight formations of UAVs. And finally, Section 4.3 summarises the existing research for the cooperative search of UAVs and the proposed swarm search solution.

4.1 Prior Research

4.1.1 The Cooperative Search of UAVs

4.1.1.1 The Goal of the Cooperative Search of UAVs

A variety of search methodologies have been investigated to accomplish the cooperative search of UAVs. The ultimate goal of UAVs is to locate the positions of mobile targets, which can be accomplished through one or more of the following five objectives (Vincent and Rubin 2004):

- Maximise the probability of detection;
- Minimise the expected detection time;
- Minimise the number of UAVs employed in the search operation;
- Enhanced robustness to ensure the search operation would not be interrupted by individual UAV failure;
- Minimise the amount of information to be exchanged between UAVs.

4.1.1.2 Categories of the Cooperative Search Approaches

In order to achieve the five objectives above, the search strategies of the cooperative search of UAVs are expected to be efficient and effective, and require minimum communications between UAVs. Depending on how the behaviour of UAVs are designed to carry out the cooperative search, (Vincent and Rubin 2004) categorised the cooperative search approaches indicating that UAVs carry out either cooperative search patterns or non-cooperative search patterns. Table 4.1 presents the categories of the cooperative search approaches. It also shows that for each of the search patterns, UAVs operate with either predefined flight paths or dynamic flight paths.

Table 4.1 The Categories of the Cooperative Search Approaches (Vincent and Rubin 2004)

	Non-cooperative Search	Cooperative Search
Predefined Flight Paths	Example: Each UAV flies parallel sweeps, independent of other UAVs.	Example: UAVs coordinate to fly a cooperative sweep *pattern*.
No Predefined Flight Paths	Example: Each UAV flies a random path, independent of other UAVs.	Example: UAVs each fly dynamically computed independent paths, but share information about the environment.

The examples of the cooperative search with predefined flight path of UAVs include the line formation introduced by (Vincent and Rubin 2004) and the fixed flight formation presented in (Ryan, Zennaro et al. 2004). In a predefined line formation, for instance, UAVs scan through the rectangular search area in either parallel search patterns (Vincent and Rubin 2004) or angled search patterns (Altshuler, Yanovsky et al. 2008). When UAVs operate a random flight path, each of them either flies independently (Marios M. Polycarpou 2001) or cooperatively (Pack and York 2005; Dasgupta 2008) by sharing the information of the search environment.

Recent research interests have shifted from non-cooperative search approaches (Ablavsky and Snorrason 2000) to cooperative search approaches (Ryan, Zennaro et al. 2004). The flight path of UAVs is predefined as part of the cooperative search strategies of UAVs. As a consequence, later study (Ryan, Tisdale et al. 2007; Dasgupta 2008) explored more in the category of cooperative search pattern and the non-predefined flight paths of UAVs.

The UAV cooperation is the most significant component in this presented research. As presented in Chapter 2, the knowledge framework of the collective problem-solving describes how the characteristics of distributed intelligent systems can be applied as effective and robust problem-solving strategies to solve dynamic problems such as the cooperative search of UAVs. Therefore, the proposed swarm search algorithm is in the category of cooperative search approach and has no predefined flight path for UAVs.

4.1.1.3 The Cooperative Control Architecture of UAVs

In order to accomplish the goal of the cooperative search of UAVs, (Pack and York 2005) presented a 4-state control architecture to support the overall search operation: 1) Global Search (GS), 2) Approach detected Target (AT), 3) Orbit and Locate Target (LT), and 4) Local Search for lost mobile targets (LS). UAVs carry out the cooperative search operation by going through each of the four states of the control architecture. The decisions of switching from one state to another state are made based on the search and cost functions. For example, in the Global Search state, each UAV calculates its flight path and flies to a point with the minimum explored history (Pack and Mullins 2003). Computations are also used in the Locate Target state to determine the optimal number of UAVs involved in locating targets, as well as to decide whether or not to initiate the Local Search in order to continue searching for the target. Referring to Table 4.1, this search strategy is categorised to be a cooperative search with no predefined flight path. UAVs operate independently from each other, yet they share the information about the search environment.

The detection of target signal and the identification of target location are the two major processes involved in the cooperative search of UAVs. They both require strong sensing and image processing techniques, such as the triangulation of multiple UAVs, the angle-rate algorithm and the Kalman filtering techniques (Toussaint, De Lima et al. 2007; Pack, DeLima et al. 2009). A range of experiments is implemented to demonstrate these techniques, and the results indicate that there are various constraints for each of the techniques. First of all, despite the advancement of sensor hardware and image processing techniques, individual UAVs can only offer limited fields of view due to various obstructions to their line-of-sight caused by the variations of search terrains (Vincent and Rubin 2004). Second, the angle-rate algorithm is generally restricted to fixed targets and has constraints on the angle-of-approach. And finally, Kalman filters require preliminary information of the targets, and also the hardware specifications of the sensors used (Pack, DeLima et al. 2009).

As for the triangulation of multiple UAVs, it has emerged that the flight formations of cooperating UAVs are capable of delivering promising performance in the detection of multiple mobile targets (Altshuler, Yanovsky et al. 2008; Pack, DeLima et al. 2009). Different flight formations of UAVs are generated and maintained through UAV cooperation, which enable UAVs to detect the target signal, track the target movement and finally to locate the target position (Vincent and Rubin 2004; Pack and York 2005; Toussaint, De Lima et al. 2007; York, Pack et al. 2007).

There are two types of flight formations of UAVs being studied aiming to enhance the search and detection of mobile targets – the line formation (Vincent and Rubin 2004; Altshuler, Yanovsky et al. 2008) and the triangular formation (Pack and York 2005; Toussaint, De Lima et al. 2007; York, Pack et al. 2007; Pack, DeLima et al. 2009). Section 4.1.2 and Section 4.1.3 each presents an overview for the two types of the flight formations of UAVs.

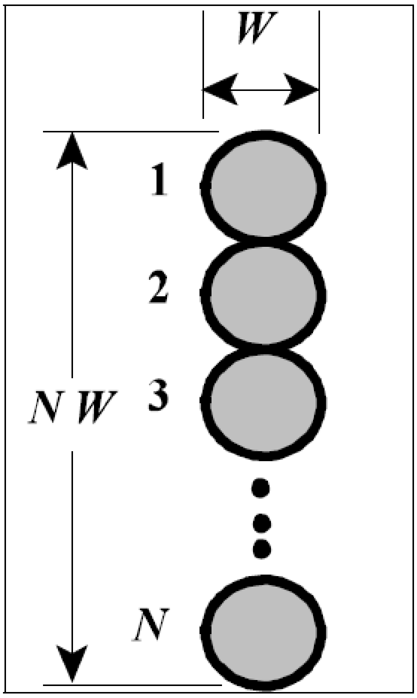
4.1.2 The Line Formation of UAVs

The line formation of UAVs was originally proposed in (Vincent and Rubin 2004) as part of the cooperative search algorithm to search, detect and locate "mobile and evasive" targets. The cooperative search algorithm is composed of the line formation pattern and the search pattern of UAVs. An improved cooperative search algorithm has been presented by (Altshuler, Yanovsky et al. 2008) to accomplish the same cooperative search problem. The improved cooperative search algorithm proposed a modified search pattern while the line formation pattern remains the same.

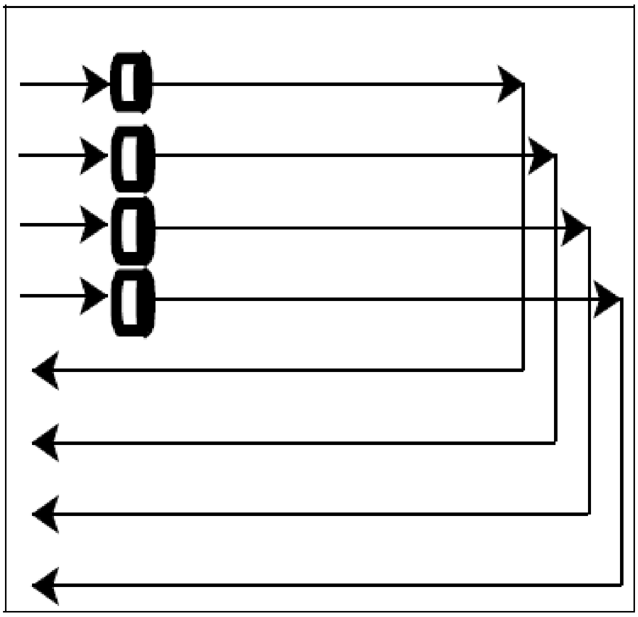
The line formation is predefined and maintained unchanged throughout the search operation. It consists of a limited number of UAVs. UAVs scan through the rectangle search area while moving between the area boundaries. Figure 4.1 illustrates the line formation of UAVs for the original (Vincent and Rubin 2004) and the improved search patterns (Altshuler, Yanovsky et al. 2008).

Figure 4.1 The Line Formation of UAVs and Search Patterns

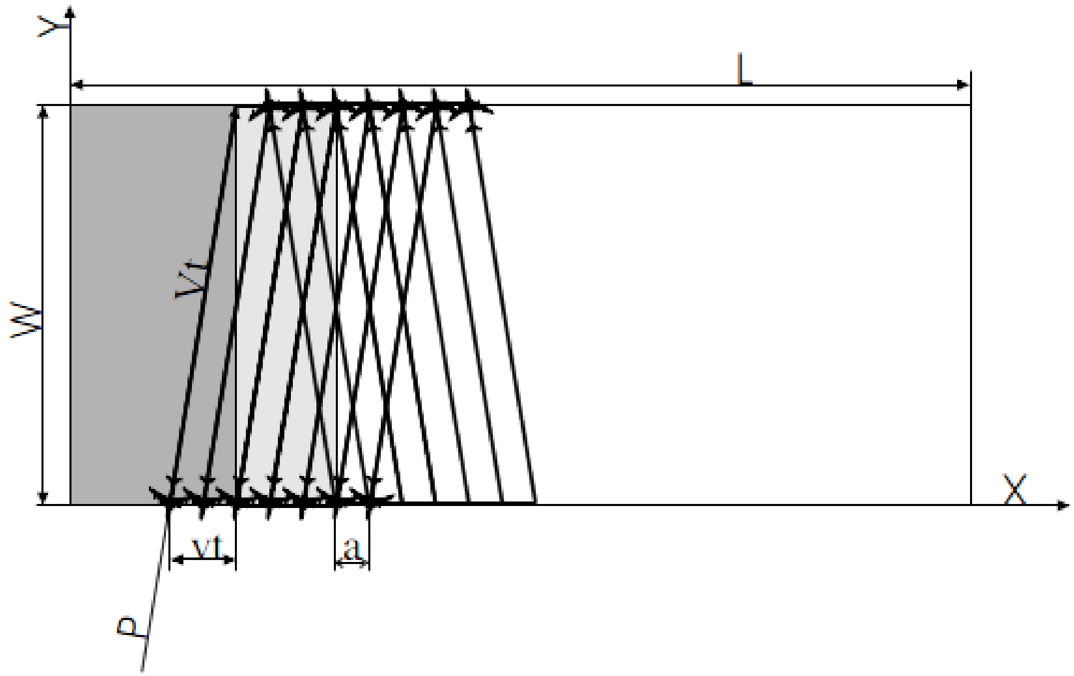
a. **The Line Formation, N UAVs (Vincent and Rubin 2004; Altshuler, Yanovsky et al. 2008)**



b. **Parallel Path Search (4 UAVs) (Vincent and Rubin 2004)**



c. Improved Search Pattern (Altshuler, Yanovsky et al. 2005; Altshuler, Yanovsky et al. 2008)



The figure shows that the same line formation is deployed with two search algorithms, in which each uses different patterns to scan through the rectangular search area. The original line formation of UAVs uses a parallel search pattern in order to cover all parts of the search area. Since the targets are moving and evading, the search strategy is to define a region as "swept clean" if all targets within the region have been detected. To restrain the search complexity, undetected targets are prevented from entering the sweep-clean region. Thus, while the UAVs in the line formation scan between the boundaries of the search area, the sweep-clean region is also expanding at the same time. As a result, UAVs are expected to locate all the targets when they have finished scanning the entire search area. For the same line formation, (Altshuler, Yanovsky et al. 2008) have improved the search pattern of UAVs by adding angles to the scanning pattern. The purpose is to produce additional sweep-clean region and thus to increase the efficiency of the target detection with only half the number of UAVs deployed in the original line formation.

Furthermore, the line formation of UAVs also considers the impact of the individual failure of UAVs to the entire search operation. The loss of UAVs includes the situations such as the loss of communications, hardware malfunction, and UAVs shot down by enemy fire. It proposed to manage the loss of one or more UAVs by reconfiguring the formation pattern. Having formed into the line formation, there are three types of control messages exchanged between UAVs to maintain the formation pattern, to update and to reconfigure the formation pattern in case when one or more UAVs are lost. The original cooperative search algorithm (Vincent and Rubin 2004) proposed to retain the pattern of line formation and the number of UAVs in order to minimise the amount of information that must be communicated, and also to keep the reconfiguration as simple as possible.

However, the argument is that because the number of UAVs is fixed and UAVs are restricted from reconfiguring into different formation patterns, the sensor coverage of the UAV formation would be reduced by the loss of UAVs. Reduced sensor coverage implies increased detection time since it will take longer for remaining UAVs in the line formation to sweep clean the search area. Therefore, the fixed number of UAVs and the restricted reconfiguration of the formation pattern are unable to guarantee efficient handling of the loss of UAVs.

In addition, the search pattern of UAVs is pre-specified and is to be carried out from the south boundary to the north boundary of the rectangle search area. This means that the search area must be known to the UAVs before they initiate the search operation. It requires UAVs to be provided with basic but accurate information of the search field, which is considered to be disadvantageous as the cooperative search algorithm would not be robust enough to adapt into different situations of target detection. For instance, UAVs would be unable to accomplish the search operation if they are placed in an unknown search area or there is only little or out-of-date terrain information provided before the search operation is initiated.

Briefly, the line formation of UAVs is predefined and consists of a fixed number of UAVs. To accomplish the goal of locating all the mobile targets in the shortest period of time, the UAVs maintain the line formation and scans between the south and north boundaries of a rectangular search area. If one or more UAVs are lost, the remaining UAVs are reconfigured into the same line formation and continue to search for targets. The line formation requires prior knowledge of the search area, and the restrictions apply to minimise the message exchange required for reconfiguration as well as to simplify the process of reconfiguration.

It might be argued that, firstly, in the case of lost UAVs, the fixed number of UAVs and the restricted formation pattern for the reconfiguration would cause a reduction of UAVs' sensor coverage. Ultimately it may result in additional time consumption of completing the search operation. Secondly, both the formation pattern and the search pattern of UAVs are predefined based on the information of the search area. This, on the other hand, might also degrade the robustness and flexibility of the cooperative search algorithm if only little, or out-of-date or no information could be provided before the search operation initiates

4.1.3 The Triangular Formation of UAVs

The triangular formation of UAVs refers to the cooperative pattern of multiple UAVs, which is generated between two to three UAVs by orbiting around the target. It aims to collect sufficient angle-to-target estimations from multiple UAVs in order to obtain accurate and reliable target location.

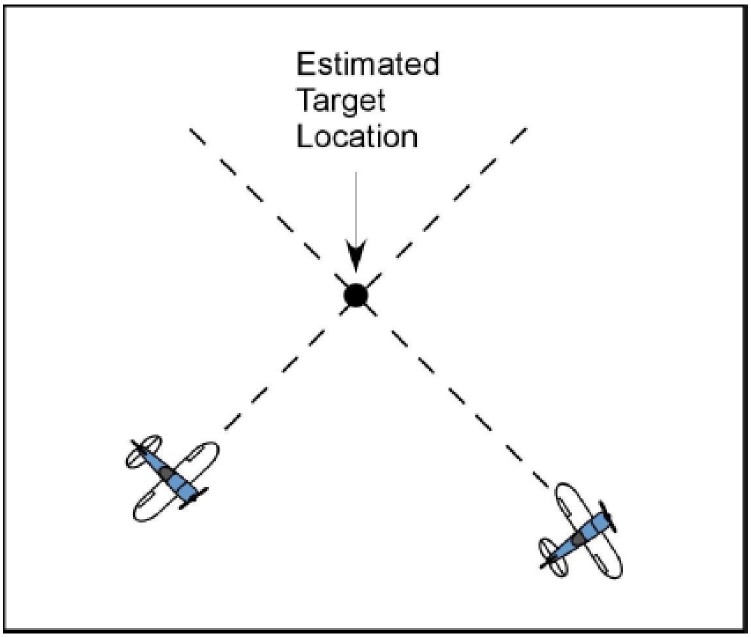
A series of study have been conducted (Pack and York 2005; Toussaint, De Lima et al. 2007; York, Pack et al. 2007; Pack, DeLima et al. 2009) to investigate the triangular formation of UAVs. This technique enables UAVs to work together in order to obtain an estimation of target locations with maximum accuracy. Individual UAVs are able to detect target signal with their angle-of-approach sensors. Such kind of sensors provides coarse estimation of the angle-of-approach between the UAV and the target. Based on the initial estimation of target location, a team of two to three UAVs converge to an orbit around the target and collect additional data on the estimated angles towards the target. UAVs triangulate in such a formation to obtain multiple estimations of the target location. Compared to the independent estimation of the target location, the estimated angles collected from multiple UAVs produce more accurate and reliable data of the target location (Toussaint, De Lima et al. 2007).

The triangulation of two and three UAVs is generated through a coordinated leader-follower approach (Toussaint, De Lima et al. 2007). The leader is the UAV that first detects the target signal, which determines the orbit direction for the follower UAVs to converge on; also the leader UAV identifies the position for the follower UAVs in the triangulation. This leader-follower coordination enables UAVs to converge onto the formation. However, approaching the target at different time scales may also produce errors in collecting sensor data.

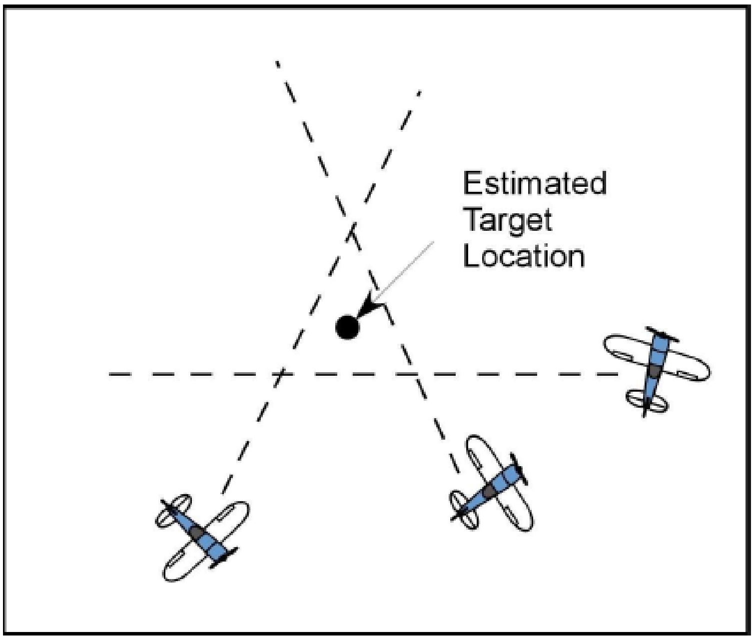
Figure 4.2 illustrates the process of estimating target locations with multiple UAVs (Toussaint, De Lima et al. 2007). In (a.), the target location is estimated by the intersection of the two angle bearing lines of two UAVs. In (b.), three UAVs are deployed to produce a triangulation around the target and estimate the target is located in the centre of the triangulation.

Figure 4.2 Estimating the Target Location with Multiple UAVs (Toussaint, De Lima et al. 2007)

a. **With Two UAVs**



b. **With Three UAVs**



The two cases of target location estimation are demonstrated through simulations and the results indicate that, the average deviation between the estimated target location and the actual target location varies more with three UAVs compared to that with two UAVs. The smaller deviation between the estimated target location and the actual target location, the more accurately the detected target is located. As a consequence, increasing the number of UAVs from two to three may not necessarily reduce the average deviation to the estimated target locations. Later study (Pack, DeLima et al. 2009) also showed that triangulation with two UAVs also encountered difficulties in creating correct sensor coverage around the target. As shown in Figure 4.2 (a.), the two UAVs intend to estimate the target location at the intersection of their angle bearing lines. However, due to no predefined flight path and the activities of approaching the detected target are not synchronised between the two UAVs, they could be either flying too close to each other or unable to read sensor data at appropriate time (Pack, DeLima et al. 2009). As a result, the two UAVs would be unable to produce a proper intersection of their angle bearing lines to estimate the target location.

The recent study of the flight formations of UAVs suggests that such kind of approaches to locating target positions have potential benefits but also have constraints regarding the accuracy of estimated target locations. In this research, alternative approaches of the flight formations with multiple UAVs are explored in order to enhance the effectiveness of identifying target locations and to eliminate related constraints.

Briefly, the existing cooperative search strategy of UAVs consists of the decentralised cooperative search architecture and the techniques used to determine the locations of mobile targets. The decentralised cooperative search architecture provides UAVs with four states of search behaviour, which supports UAVs to carry out the cooperative search operation. In the state of locating target positions, different techniques are used to estimate the target locations. Amongst them, the flight formations of cooperating UAVs are reviewed in detail to show the strengths and weaknesses. A swarm-inspired cooperative search solution is next proposed to explore an alternative approach for the cooperative search of UAVs. Section 4.2 presents an outline for the proposed cooperative search solution.

4.2 Swarm-inspired Search Strategy

Learning from the emergent behaviour of social insects and based on the study of the collective problem-solving of multi-agent systems, a swarm-inspired cooperative search solution is proposed for the cooperative search of UAVs. The proposed solution presents behaviour-based search strategies, which facilitates the UAV cooperation to be emerged from individual behaviour of UAVs. Section 4.2.1 explains the behaviour-based cooperative search using the framework of collective problem-solving. Section 4.2.2 describes the new flight formations of UAVs deployed in the proposed search strategy.

4.2.1 Behaviour-Based Cooperative Search

The solution specifies each UAV with simple and identical rules of behaviour. At the beginning of the search operation, UAVs initiate the search operation with randomised flight pattern. Once an UAV has obtained an initial detection of target signal, it communicates with neighbouring UAVs to exchange information on signal detection. The signal-stimulated communications enables UAVs to cooperate with each other in a manner of "occur-on-demand", which minimises the information to be exchanged between UAVs. Through such kind of communications, UAVs are converged onto predefined formation patterns.

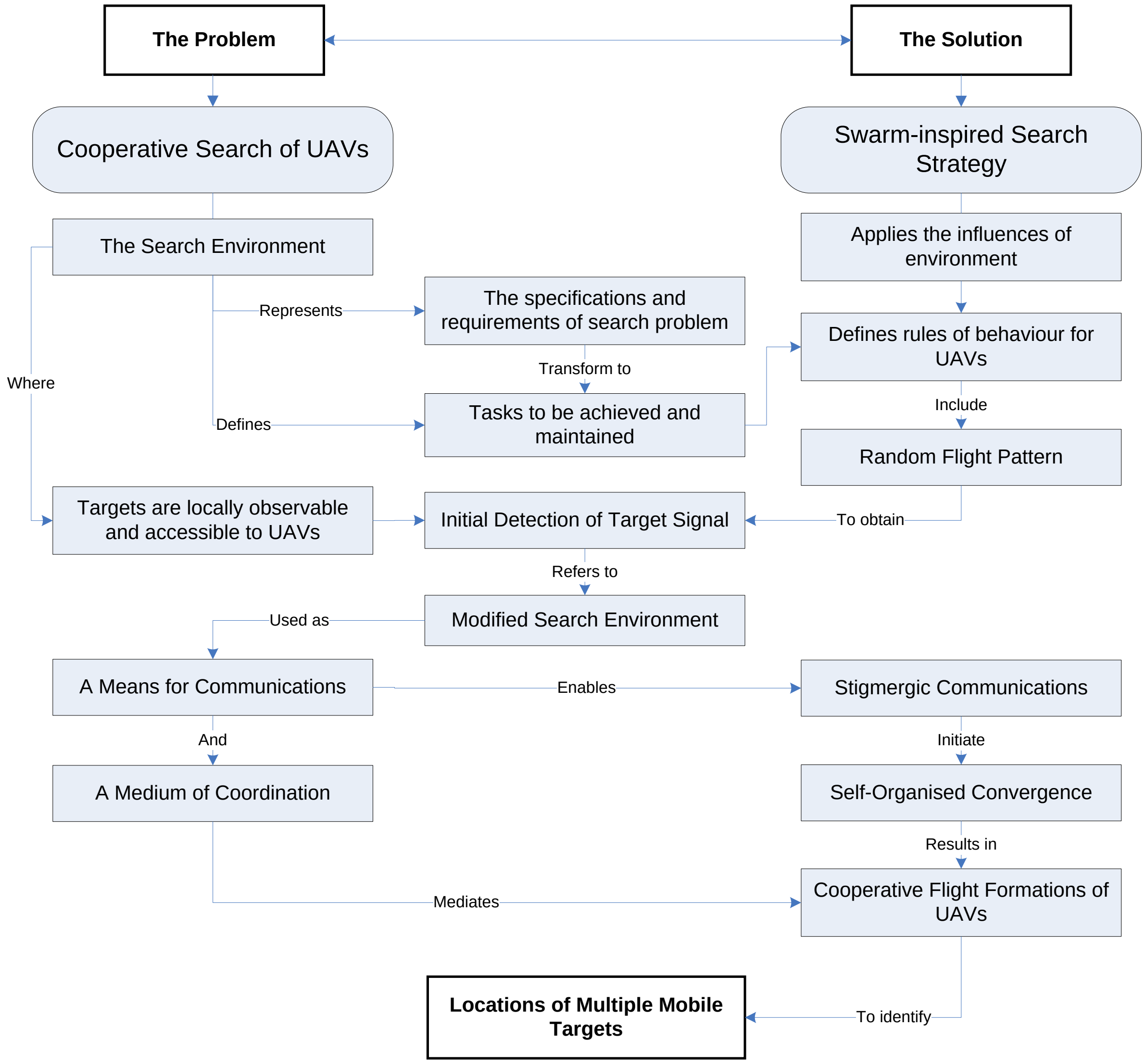
According to the framework of collective problem-solving, the cooperative search of UAVs is an example of collective problem-solving in a multi-agent system. The behavioural rules of UAVs are designed to make use of the influences of environment. The environment here refers to a search environment, where the cooperative search of UAVs is taking place. More importantly, like the application environment for multi-agent systems, the search environment represents the problem specifications and requirements of cooperative search of UAVs.

As illustrated in the framework, the search environment has following influences on the cooperative search of UAVs:

The search environment defines the tasks to be achieved by UAVs, such as to detect and locate the coordinates of mobile targets. Targets continuously emit radio frequency signal that can be detected by the sensors of UAVs. Thus as long as they are in the senor range of UAVs, the target distribution and signal presence are locally observable and accessible to UAVs. The initial detection of target signal indicates that the original search environment has been changed, which then stimulates communications between UAVs. Referring to the stimulating configurations in the nest construction of termites and the model of self-assembly, the modified search environment becomes a means for communications. Then the signal-stimulated communications enable more UAVs to be recruited to converge into a cooperative flight formation. In above processes, the search environment mediates the UAV cooperation to occur through signal-stimulated communications.

Figure 4.3 illustrates how the problem of the cooperative search of UAVs and the proposed solution constructed according to the framework of collective problem-solving.

Figure 4.3 The Swarm-inspired Search Strategy



4.2.2 New Flight Formations of UAVs

The formation patterns vary depending on the number of UAVs. Two UAVs are converged onto a diagonal formation, and three UAVS are converged onto triangular formation. The new flight formations differ from the recent approaches in the following ways: firstly, UAVs are converged in the predefined patterns of flight formations. The two UAVs are converged onto a diagonal position, and the three UAVs are converged onto a triangular formation. Secondly, rather than estimating the target location with the intersection of the angle bearing lines, the new flight formations enable overlapping sensor coverage to be emerged between converged UAVs. The target locations are to be identified at unique coordinates (X-coordinate and Y-coordinate) within the overlapping sensor coverage of UAVs.

If the converged UAVs are unable to identify the target location immediately, they initiate synchronised circling behaviour in order to track the moving target and locate the target in a wider area. While circling around together, UAVs maintain the same formation pattern to ensure their sensor range is overlapped properly.

Referring to Table 4.1, the proposed search solution is in the category of cooperative search approaches with UAVs operating a randomised and independent flight path. The search environment changes once UAVs obtain an initial detection of target signal. They share the information of target detection via the local communications between each other. The signal-stimulated communications enable UAVs to effectively respond to the various dynamic situations during the search operation. A successful communication facilitates further cooperative search activities of UAVs, such as the flight formations and synchronised circling behaviour.

4.3 Summary

The cooperative search of UAVs has become an important subject of research in the collective problem-solving. It has attracted continuous research interests in a range of subject areas. This chapter reviewed prior research of the cooperative search of UAVs to show that in-depth studies have been conducted to investigate strategies for the cooperative search of UAVs. A four-state control architecture (Pack and York 2005) has been constructed as an overall solution methodology for the cooperative search of UAVs. It defines four states of search for each UAV: the Global Search state, the Approach Target state, the Locate Target state, and the Local Search state. In the Locate Target state, three techniques (Toussaint, De Lima et al. 2007; Pack, DeLima et al. 2009) are used to estimate the target location – the triangulation of UAVs, the angle-rate algorithm, and the Kalman filter for the sensor measurement of multiple UAVs. The experimental results for each of the three target localisation techniques show strengths and weaknesses. Amongst them, the triangulation of multiple UAVs seems to have the most potential capabilities that would enable UAVs to identify mobile target locations efficiently and effectively.

Learning from the emergent behaviour of social insects, various features that are essential to the collective problem-solving have been explored. The problem-solving patterns of social insects are compared with the collective problem-solving in multi-agent systems. From the two subject areas, a framework of swarm-inspired search strategies is proposed. The framework presents a cooperative search solution, which defines each individual UAV with identical rules of behaviour. The behaviour-based search solution consists of the following strategies: random search pattern, signal-stimulated communications, new flight formations of cooperating UAVs, and synchronised circling behaviour.

The proposed search solution presents two formation patterns of UAVs, one is a diagonal formation for two UAVs, and the other is a triangular formation for three UAVs. Unlike the recent approaches of the UAV triangulation, the new flight formations of UAVs are predefined and generated via the signal-stimulated communications. The locations of mobile targets are to be identified in the overlapping sensor coverage of the converged UAVs. UAVs maintain the same formation throughout the identification process, if necessary, by carrying out additional search by circling around synchronically to locate the target.

Details of the swarm-inspired cooperative search solution are described and explained in Chapter 5.

Chapter 5. Swarm-Inspired Search Strategy of Unmanned Aerial Vehicles

In the literature, where cooperative UAVs search for targets has been considered (Toussaint, De Lima et al. 2007; York, Pack et al. 2007; Altshuler, Yanovsky et al. 2008; Pack, DeLima et al. 2009), the assumption has been that three cooperating UAVs are needed to triangulate and locate the targets. No alternatives have been investigated or evaluated. Here the author proposes the flying formation of two and three cooperating UAVs to locate the targets; and evaluates their relative performances through the design and implementation of a series of experiments (Chapter 6).

The swarm-inspired search strategy is constructed based on the framework of collective problem-solving, which emphasises the influences of the search environment on the UAV performance. It makes use of the search environment to provide UAVs with real-time information of the cooperative search, and mediates UAV cooperation. Therefore, UAVs operate on identical rules of behaviour and are able to respond to changes of the search circumstances with appropriate actions.

Section 5.1 defines the cooperative search problem of UAVs, and Section 5.2 presents an overview of the proposed solution methodology. Section 5.3 introduces the development environment applied to simulate the cooperative search operation of UAVs. Section 5.4 further explains and specifies the solution methodology in the form of the detection algorithm. And finally, Section 5.5 summarises the proposed search solution.

5.1 The Problem Definition

The problem under study explores deployment of multiple Unmanned Aerial Vehicles (UAVs) to search, detect and locate multiple mobile targets in an unknown area. The problem is defined by referring to the existing search model of UAVs (Vincent and Rubin 2004; Morris, Mullins et al. 2006; York, Pack et al. 2007), and thus represents general context and structure of the cooperative search of UAVs.

The UAVs are micro-sized aircrafts with identical specifications of hardware and software. Each UAV is equipped with a 360 °sensor and programmed with a series of behavioural rules. The sensor enables UAVs to detect the target signal. The rules of behaviour are designed so that multiple UAVs are able to either operate as an individual or cooperate with each other aiming to accomplish the multiple target detection. According to these rules, UAVs carry out different flight patterns and search activities at different stages of the search operation.

The targets are moving randomly on the ground of the search area. They continually send out radio frequency (RF) signal, which can be detected by the sensors of UAVs.

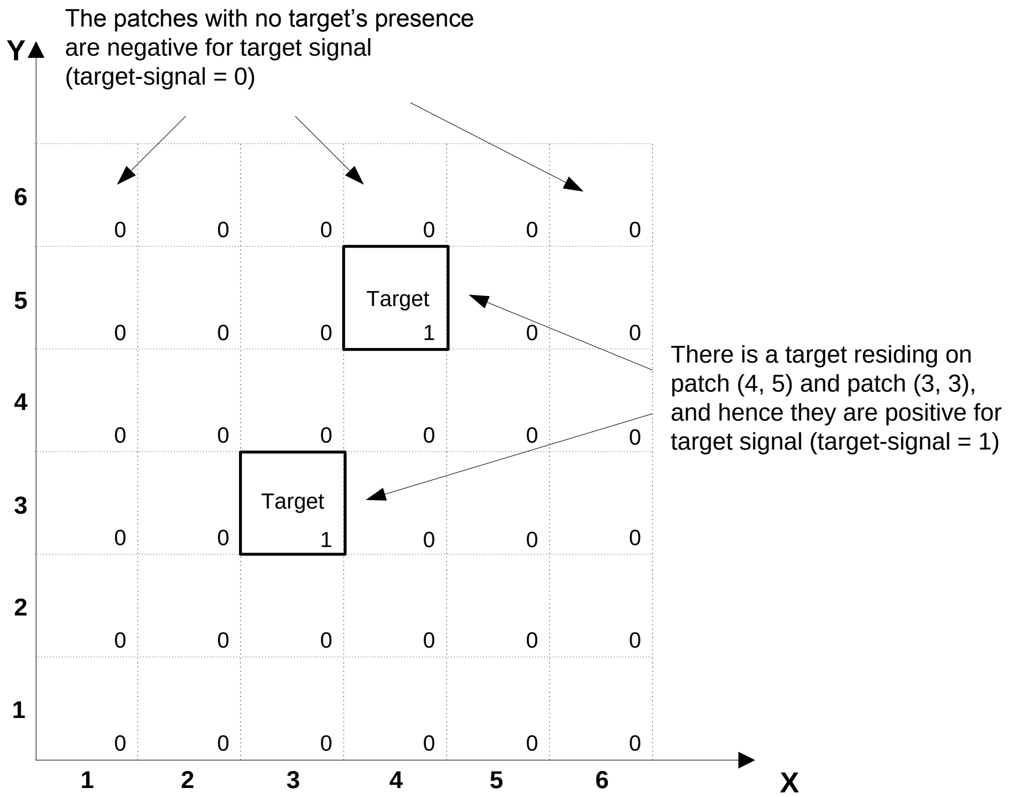
The search area is two-dimensional with X and Y coordinates. It is divided up into a grid of patches. Targets move on the ground from one patch to another patch. UAVs operate in the air and thus they are in the third dimension (Z) of the search world.

During the search operation, targets and UAVs interact with patches in different ways as two types of agents. They present their own characteristics and have their own rules of activities.

5.1.1 The Target

The target is moving from patch to patch at a speed of 0.005 steps[6] per time, and continuously emit signal. Every time when the target is residing on a patch, that patch has a positive target signal and thus given a signal value of 1. Other patches with no target present have no signal and thus given a signal value of 0. Figure 5.1 shows an example of the presence of targets and target signals. If there is a target currently residing on patch (4, 5), the target signal of patch (4, 5) is given a signal value of 1. If there is a target currently residing on patch (3, 3), the target signal of patch (3, 3) is given a signal value of 1. The patches with no target present are negative for target signal, and therefore the target signal of these patches is given a signal value of 0.

Figure 5.1 An Example of the Targets and Target Signal



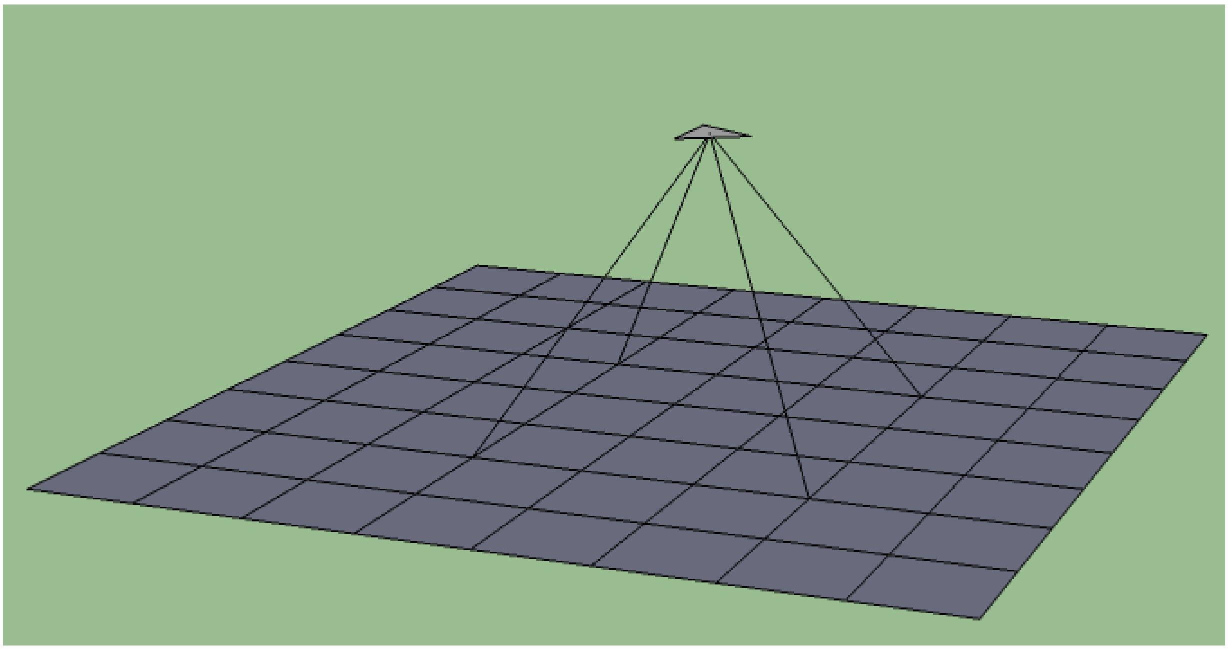
[6] In NetLogo models, time passes in discrete steps and is recorded by a reporter called ticks.

5.1.2 The UAV

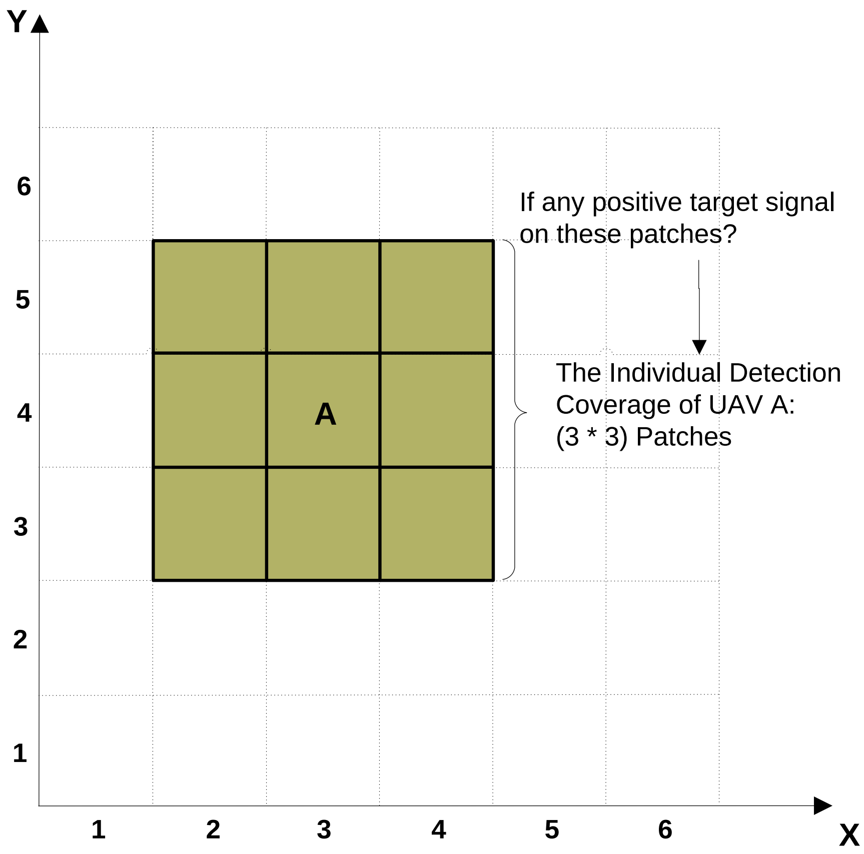
While they are operating in the air randomly, UAVs look to detect target signal with their 360 ° sensor onboard. The 360 ° sensor range enables each individual UAV to scan 3 * 3 patches at a time to look for positive target signal. Hence, the 3 * 3 patches of the ground area define the individual detection range of UAVs. A positive target signal can be picked up by UAVs if there is a target present on one of the 9 patches of the UAV's individual detection range. The precision of detection is governed by the altitude of the UAV. The lower the UAV operates, the more precise the UAV is able to identify individual patches of the search area. Figure 5.2 (a.) and (b.) each shows the three-dimensional illustration and the two-dimensional illustration of the individual detection range of UAVs.

Figure 5.2 The Individual Detection Range of UAVs

a. Three-Dimensional Illustration



b. Two-Dimensional Illustration



5.1.3 Assumptions

Aware of Geographical Boundaries

It is assumed that UAVs are aware of the geographical boundaries of the search area and only operate inside of these boundaries.

Free from Collisions

A series of research works (Shim, Hoam et al. 2006; Vrba, Mařík et al. 2007; Viquerat, Blackhall et al. 2008; Yu, Beard et al. 2010) are being conducted in terms of generating collision-free flight formation of UAVs. Thus it is not included in the current search problem.

It is assumed that UAVs operate free from collisions, both with each other and with any types of obstacles. The same assumption also applies to targets.

Sensor

The RF sensor detects the intermittent radio frequency signal emitted from mobile targets (Pack and York 2005; Toussaint, De Lima et al. 2007; Pack, DeLima et al. 2009). This type of sensor is capable of direction finding and estimates the angle to target with a precision range of +/- 7 degrees (York, Pack et al. 2007). Despite its low precision range, the common assumption in research focusing on the cooperative mechanisms of UAVs is that, such kind of sensor technology is sufficiently reliable to detect the target signal. The target signal is assumed continuous in this work, so each UAV is assumed to be able to detect the target signal reliably. With the major interests in the detection algorithm of UAVs, this research assumes the RF sensor is able to detect the target signal within a 3 * 3 metres area. Referring to the sensor range of 3 * 3 patches defined for each UAV in Section 5.1.2, each patch equals to 1 square metre in real-time unit.

Multiple Dimensions

Because of the development environment, the search operation is modelled in two dimensions but assuming UAVs operate above the search ground.

Briefly, targets and UAVs have different characteristics due to their roles in the search operation. Their characteristics are specified as part of the problem definition. The patch with target present is defined to have positive target signal and given a signal value of 1, whereas other patches with no target present are given a signal value of 0. The 360° sensor defines the individual detection range of UAVs to be 3 * 3 patches on the search ground. It enables each

UAV to detect positive target signal if a target present on one of the patches inside of this range.

Having defined the search problem and the characteristics of targets and UAVs, Section 5.2 presents an overview of the solution methodology.

5.2 Overview of The Solution Methodology

Inspired by the emergent behaviour of insects, the proposed solution is composed of the independent behaviour of UAVs and the cooperative behaviour emerged from the independent behaviour. The independent behaviour of UAVs includes the random flight pattern and the initial detection of target signal. An individual UAV is unable to identify the target location as it cannot form a diagonal and/or triangular formation. Therefore, triggered by the initial detection of target signal, the detecting UAV cooperates with other UAVs and together they generate cooperative flight formations. Depending on the number of cooperating UAVs, they converge onto either a diagonal formation if there are two UAVs or a triangular formation if there are three UAVs.

The aim of UAV cooperation is to locate the mobile targets on unique coordinates by generating cooperative formations of UAVs. Referring to the line formation and triangulation of UAVs, it intends to provide alternative ways of UAV formations with improved performance to locating multiple mobile targets.

Emergent from the independent behaviour of UAVs, the cooperative behaviour of UAVs consists of the detection-stimulated communications, the convergence of UAV formation and the circling behaviour. The UAV formation is the key component of the proposed search solution, which is supported by the random search pattern of individual UAVs. UAVs are converged onto the formation in order to identify the target-located patch through cooperation. The target-located patch is the only patch with target signal that is commonly covered by all of the UAVs of the formation. The search operation of UAVs is terminated if all targets are located and/or the predefined maximum time length has expired.

5.2.1 Independent Behaviour of UAVs

At the initial stage of the search operation, UAVs operate in a randomised flight pattern to search for target signal. Targets are randomly distributed and moving on the ground of the search area. Taking the inspiration from the random walk of ants, the random flight pattern enables UAVs to explore the entire search area effectively. Each UAV scans the ground of search area in a range of 3 * 3 patches at a time to detect the target signal. When there is a target present, the UAV is able to detect a positive target signal. Once a target signal is detected, the detecting UAV records the 9 patches of its current detection range into a list. The list of 9 patches becomes the initial detection data of the detecting UAV. The initial detection of target signal triggers the cooperative search activities of multiple UAVs.

Having had an initial detection of target signal, the detecting UAV is to cooperate with other UAVs in order to identify the target location through either the diagonal formation or the triangular formation. Hence, it switches from the current random flight pattern and initiates a series of cooperative activities. Section 5.2.2 describes the details of the cooperative behaviour of UAVs.

5.2.2 Cooperative Behaviour of UAVs

First of all, the initial detection initiates local communications between the detecting UAV and the nearby UAVs. Through the communications, the detecting UAV recruits the closest UAV and the two are converged onto a formation. The proposed search solution presents two scenarios of UAV formation: one is the diagonal formation of two UAVs and the other is the triangular formation of three UAVs. When the UAVs are converged onto a formation, their individual detection range is overlapped. The UAV formation enables unique common patches to be generated between UAVs. Then if both of the UAVs in the formation have the detection of target signal, the unique common patch is identified to be the target-located patch. The coordinates of the target-located patch is the coordinates of the target location.

In the case of multiple target detection, it is common for UAVs to have detected different target signals at the same time. If there are two or more UAVs that initially detect targets in their local neighbourhoods, it would be necessary to prioritise one of the target detections on random basis. Thus, in the same local area, available UAVs only respond to one of the detecting UAVs at a time. This enables local UAVs to focus on one target at a time, and therefore ensures further UAV cooperation to engage effectively.

In addition to the UAV formation, if no common patch with target signal can be identified and the maximum time length is not exceeded, UAVs start to circle in a clockwise pattern in order to track down the target movement in a wider area. While circling, UAVs continue to carry out the procedure of identifying the target-located patch.

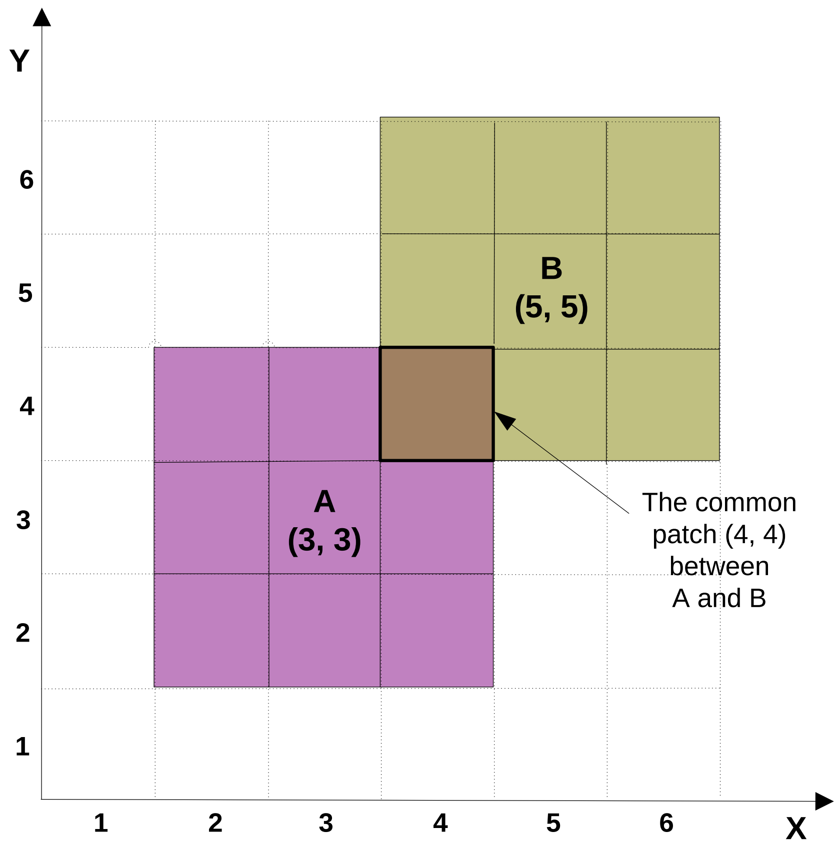
Section 5.2.2.1 describes the search solution in the scenario of the two-UAV formation, and Section 5.2.2.2 describes the scenario of the three-UAV formation. And finally, Section 5.2.2.3 introduces the circling behaviour of UAVs.

5.2.2.1 Two-UAV Formation

In this scenario, the two UAVs are converged in a diagonal formation. An initial detection of target signal stimulates the local communications between the detecting UAV and its closest neighbour. Through the detection-stimulated communications, the detecting UAV recruits the closest UAV. The recruited UAV has no initial target detection of itself at the time of recruiting. Having been recruited, the UAV moves onto the diagonal patch of the detecting UAV, and the recruited UAV are converged onto a diagonal formation. The diagonal formation generates unique common patch between the two UAVs as their individual detection range overlapped. Like the detecting UAV, the recruited UAV also records the 9 patches of current detection range in a list. Thus, each of the two UAVs in the diagonal formation has its own list of detection data. The objective is to find the unique common patch between the two lists of detection data of the UAVs. Figure 5.3 shows an example of the diagonal formation of two UAVs.

As shown in Figure 5.3, if the detecting UAV is currently hovering above patch (3, 3), the diagonal patch is defined to be the patch at (5, 5). The diagonal patch is identified two patches away from the detecting UAV. The diagonal formation enables overlapped detection coverage to emerge between the detecting UAV and the recruited UAV. In the overlapping detection coverage, there is only one patch that is covered by both of the UAVs (patch (4, 4) in Figure 5.3). Having identified the unique common patch, if both of the two UAVs have the detection of target signal, the patch is identified to be the target-located patch. Once the target is located, the two UAVs report the target location and then resume random search to explore other parts of the search area for new targets.

Figure 5.3 An Example of the Diagonal Formation of Two UAVs

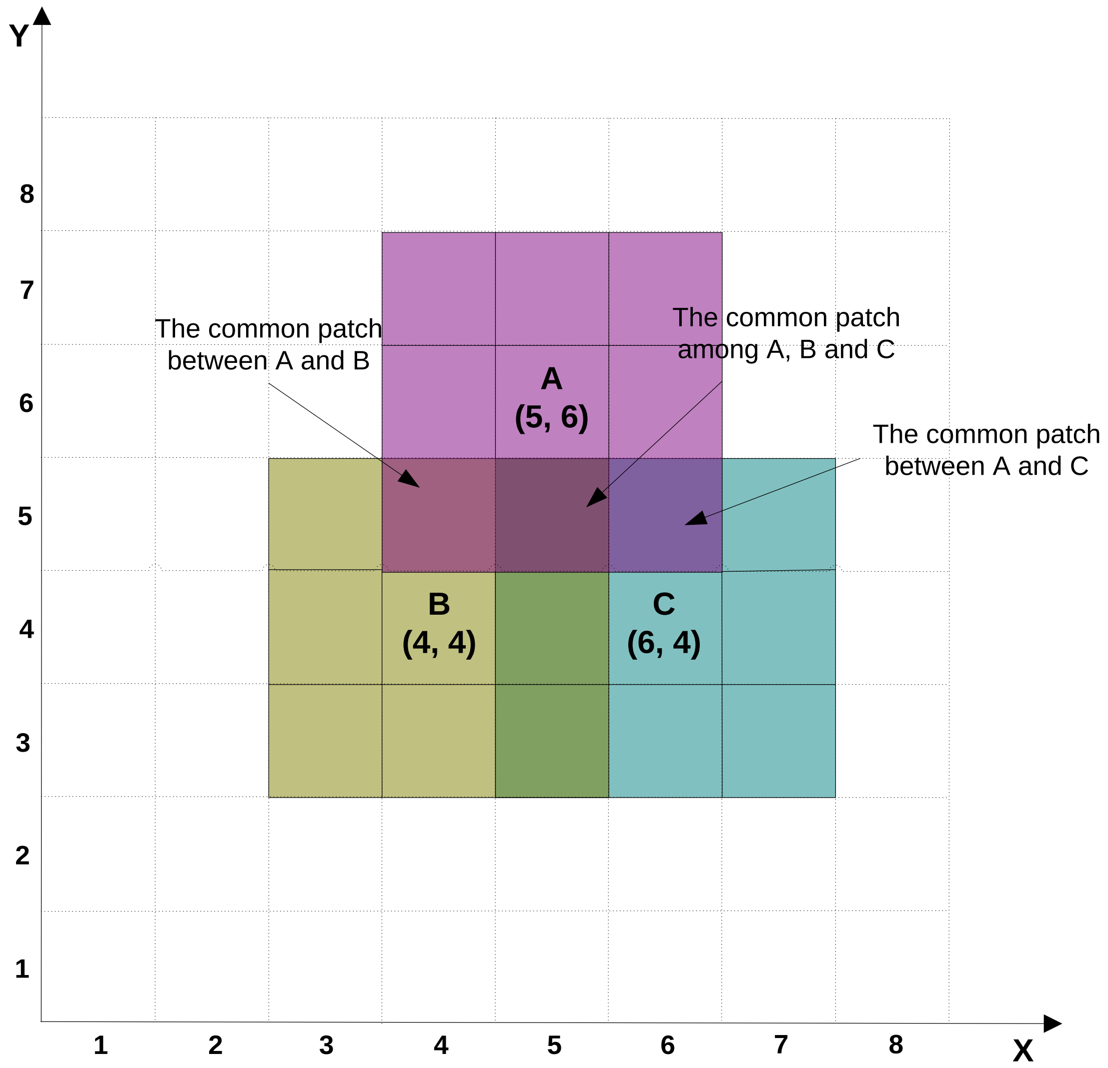


5.2.2.2 Three-UAV Formation

The rules of search behaviour described in the two-UAV formation also apply to the scenario of three-UAV formation. The detecting UAV recruits two nearby UAVs to converge onto a triangular formation. Figure 5.4 shows an example of the triangular formation of three UAVs.

Through the detection-stimulated communications, the detecting UAV recruits two nearby UAVs to converge onto two diagonal patches. In Figure 5.4, the detecting UAV A currently resides at patch (5, 6); UAV B and UAV C are recruited by A to each converge to patch (4, 4) and patch (6, 4). Thus a triangular formation emerges among the three UAVs. When they are converged, an overlapping detection range is established among the three UAVs. Each UAV records its current detection range in a list of 9 patches. The three lists of detection data are then used to find the unique common patch amongst two or all of the three UAVs. If the patch has target signal, it is identified to be the target-located patch.

Figure 5.4 An Example of the Triangular Formation of Three UAVs



For the three-UAV formation, the identification of the target-located patch is carried out between UAV A and UAV B, between UAV A and UAV C, and also among the three UAVs A, B and C. It uses the combination of A and B, and the combination of A and C because the diagonal formation between each of the two UAVs generates a common patch between them. If the detecting UAV A and either the recruited UAV B or UAV C have a common patch, plus both of the two UAVs have the detection of target signal, they would be able to identify the target-located patch. In addition, the triangular formation also generates a unique common patch amongst the three UAVs (Figure 5.4).

5.2.2.3 The Circling Behaviour of UAVs

UAVs carry out a predefined circling behaviour when they are unable to identify common patch with the detection of target signal from the diagonal formation and the triangular formation. UAVs circle around in a clockwise pattern in order to track the moving target in a wider area. The circling pattern is predefined and identical to all UAVs. Throughout the whole circling process there are a total of 25 patches that each circling UAV is to move to. While circling, UAVs move one patch per step. At each step of the circling behaviour, each UAV records its current detection range and updates the list of detection data as it moves from patch to patch.

In both of the two scenario of the UAV formation, every time the new lists of detection data are generated, they are used to identify the common patch with target signal between UAVs. If UAVs identify the target-located patch, they report the coordinates of the patch as the target location. If there is still no common patch with target signal identified when the UAVs move to the last patch of the circling process, they have lost track of the target movement as the target has moved away from the UAVs' detection coverage. In such a case, if there are more targets to be detected and the maximum time length has not been exceeded, the UAVs are dismissed from the formation and initiate a new round of target detection.

In brief, the proposed search solution presents the two scenarios of UAV cooperation – the diagonal formation of two UAVs and the triangular formation of three UAVs. The purpose of

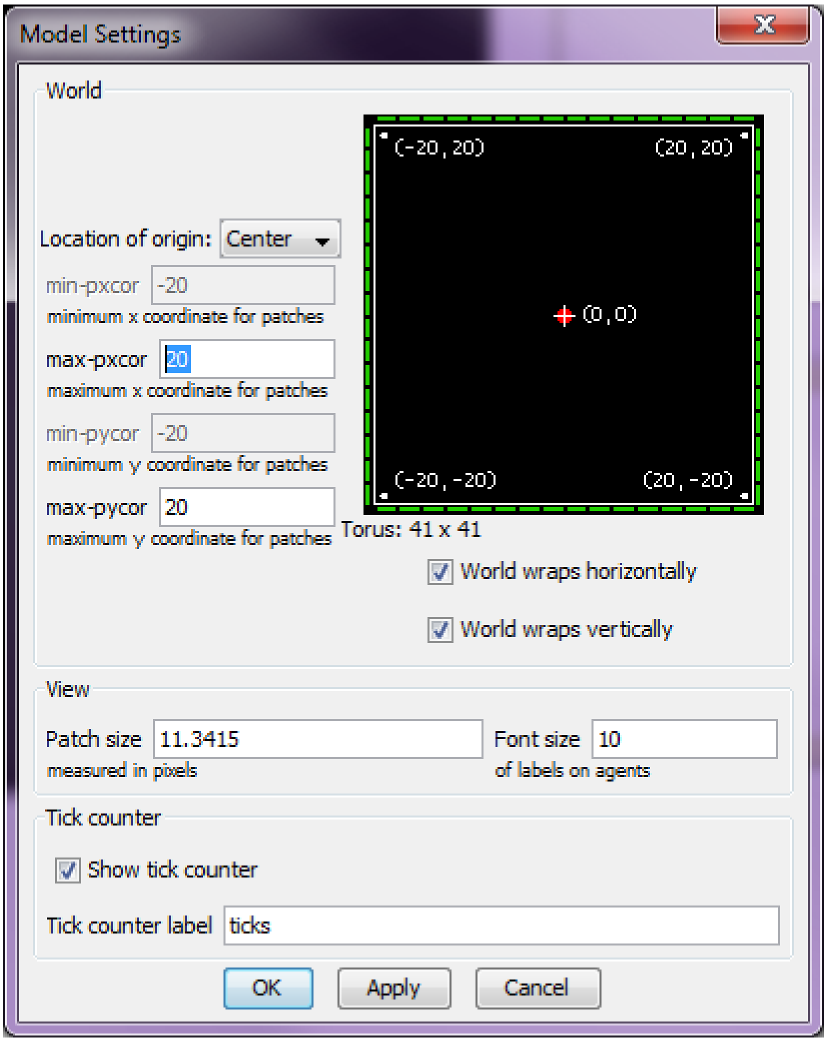
UAV formation is to identify the unique common patch in the overlapping detection coverage of converged UAVs. If the converged UAVs all have the detection of target signal, the unique common patch is the target-located patch. The performance of the research is to evaluate the proposed search solution through measuring the performance of the two cooperative scenarios. For each of the two scenarios, the performance is to be measured with the average detection time and the number of all-located target detections.

The search operation of UAVs is simulated in NetLogo. NetLogo is a popular distributed modelling language that provides an ideal development environment for implementing with the search operation of UAVs. Details of the simulation are introduced in Section 5.3.

5.3 The Development Environment

The search operation of UAVs is simulated in Java-based multi-agent modelling language NetLogo (Wilensky 1999). The search model is written in NetLogo 4.1. Figure 5.5 shows the model setting of the simulated UAV search operation.

Figure 5.5 Model Settings of the Simulated UAV Search Operation



In NetLogo, the search area is a two-dimensional world (X * Y) and divided up into a grid of equal-sized patches. It is wrapped up both horizontally and vertically and hence is a torus world. Torus is the default topology of NetLogo world, which means when UAVs or targets move past the edge of the search area, it disappears and reappears on the opposite edge and every patch has the same number of neighbouring patches. The size of the search area is determined by the number of patches, which is determined by the maximum and minimum values of X and Y coordinates. For the search operation of UAVs, the size of search area is 41 * 41 and hence there are a total number of 1681 patches. NetLogo offers scalable features so that the space and the size of patches can be scaled to represent various sizes in real-time applications, such as 1 patch = 1 square metre or 1 acre. The detection coverage of individual UAVs is defined to be 3 * 3 patches (Section 5.1.2) and as stated in the assumption (Section 5.1.3) the sensor coverage is 3 * 3 metres. Thus 1 patch = 1 square metre.

Patches and turtles are two types of agents in NetLogo. Agents are individuals that can be programmed with activities. Patches cannot move while turtles are agents that move around in the world. There are two breeds of turtles defined to each represent UAVs and targets. As described in Section 5.1, UAVs and targets have their own characteristics. Thus, each of them is specified with different rules of behaviour, as well as different colours and shapes to characterise themselves. UAVs and targets carry out their own rules of behaviour interactively.

The NetLogo procedures are user-defined commands and reporters. A command specifies details of an action for turtles to execute, and a reporter reports a computed result. NetLogo also provides built-in commands and reporters that can be used as needed. There are two kinds of procedures: one is a "setup" procedure and the other is a "go" procedure. The "setup" procedure is where the characteristics of UAVs and targets are defined. The "go" procedure consists of a series of procedures which specify the rules of search behaviour.

The independent behaviour of UAVs includes the following procedures: `detect-target-signal`, `find-closest-UAV`, `find-diagonal-patch`, and `resume-random-search`. Emerged from the independent behaviour, the cooperative behaviour includes procedures `communicate-and-converge` and `circle-clockwise`. Reporter `match` determines the existence of a common patch with target signal and if the result is true, procedure `find-patch` identifies the common patch where the target is located.

5.4 The Detection Algorithm

The detection algorithm is designed in the form of behavioural rules of UAVs. The rules of search behaviour are identical to all UAVs and determine different search activities at different stages of the search operation. At the initial stage of search operation, UAVs operate in a randomised pattern to detect target signal. Section 5.4.1 explains the details of UAVs' independent behaviour and related NetLogo procedures. The independent behaviour of UAVs refers to the random flight pattern, the initial detection of target signal, recruitment, and finding the diagonal patch. Section 5.4.2 describes the cooperative behaviour of UAVs and to explain how it is transformed into NetLogo procedures. The UAV cooperation includes the detection-stimulated communications, the diagonal formation of two UAVs, the identification of the unique common patch with target signal, and the circling behaviour of UAVs.

5.4.1 Independent Behaviour of UAVs

5.4.1.1 Random Flight Pattern

The randomised search pattern enables UAVs to explore the search area in all directions. It maximises the geographical range of search. UAVs are assigned with a constant speed, and randomised angles on the left and right. Before the search operation initiates, UAVs are evenly distributed in a circle at the centre of the search area. Once initiated, they scatter out in randomised pattern at the same time. The code below specifies the random flight pattern of UAVs:

```
ask UAVs [ rt random 40 lt random 40 fd 0.05 ]
```

Each UAV turns either right or left to the selected number of degrees by picking a random whole number between 0 and 40. Flying from the centre of one patch to the centre of another patch, UAVs operate at a constant speed of 0.05 steps at a time, which is set to be 10 times faster than the speed of targets (0.005 steps at a time). Upon an initial detection of target signal, the speed difference of UAVs and targets ensures that it would take the target an extended period of time to move away from the detecting UAV's detection range (3 * 3 patches), allowing sufficient time for the detecting UAV to initiate further cooperative activities with other UAVs in order to locate the mobile target. This significantly large difference between the speeds of the UAVs and the mobile target is essential for a successful search and is a common assumption in similar research (and is indeed found in practice) (Ryan, Zennaro et al. 2004; Vincent and Rubin 2004; DeLima, York et al. 2006).

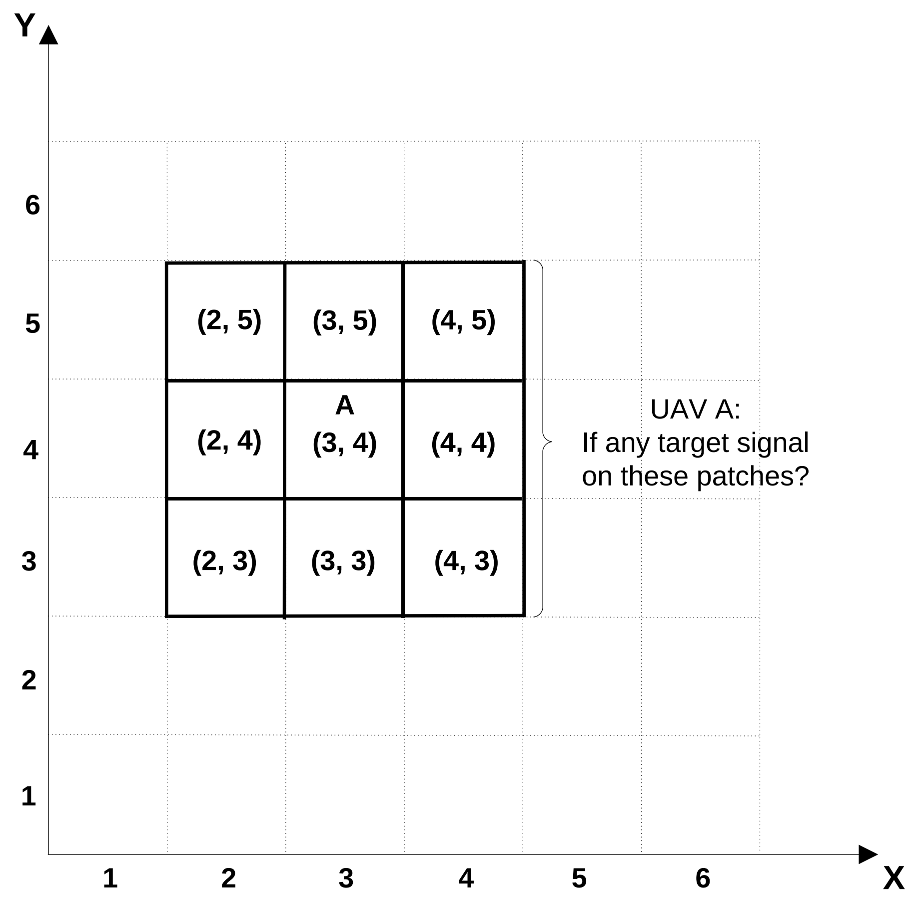
5.4.1.2 Individual Behaviour for Target Detection

Each UAV is initialised with identical rules of individual behaviour, including the random flight behaviour, the initial detection of target signal, and recruiting the closest UAV. The UAV detects target signal with equipped sensor. Having initially detected a target signal, the UAV records the detection data and intends to recruit the closest UAV. The recruited UAV converges onto the diagonal patch of the detecting UAV. The diagonal patch is identified based on the patch coordinates of the detecting UAV.

1) Initial Detection of Target Signal

If the UAV detects an initial target signal, the target is located on any of the 9 patches in the UAV's detection range. The UAV records all the 9 patches of its individual detection range as the target can be on any of the 9 patches. As shown in Figure 5.6, if UAV A is on top of the patch (3, 4) at the time of initial detection, it records the current patch (3, 4) and the 8 surrounding patches (2, 5), (3, 5), (4, 5), (2, 4), (4, 4), (2, 3), (3, 3) and (4, 3).

Figure 5.6 The Initial Detection of Target Signal



In NetLogo, the initial detection of target signal is encoded as below:

```
if target-signal = 1
    [ set my-patch patch-here
      set my-pxcor [ pxcor ] of my-patch
```

```
set my-pycor [ pycor ] of my-patch
set detection-data (list patch (my-pxcor – 1) (my-pycor + 1),
                         patch (my-pxcor) (my-pycor + 1),
                         patch (my-pxcor + 1) (my-pycor + 1),
                         patch (my-pxcor – 1) (my-pycor),
                         my-patch,
                         patch (my-pxcor + 1) (my-pycor),
                         patch (my-pxcor – 1) (my-pycor - 1),
                         patch (my-pxcor) (my-pycor - 1),
                         patch (my-pxcor + 1) (my-pycor - 1)) ]
```

2) Recruiting The Closest UAV

Having recorded the initial detection, the detecting UAV then recruits the closest UAV in order to generate a diagonal formation between them. First of all the detecting UAV has to find the closest UAV. The closest UAV is one that is currently operating in the shortest distance from the detecting UAV, and has no detection of itself at the time of recruitment. The procedure `find-nearby-UAVs` defines the group of nearby UAVs with no initial detection of themselves. And the closest UAV is then identified among this group of nearby UAVs (`find-closest-UAV`).

```
to find-nearby-UAVs
  set nearby-UAVs other UAVs in-radius 2 with [ signal? = false ]
end
to find-closest-UAV
  set the-closest-UAV min-one-of nearby-UAVs [ distance myself ]
end
```

When the closest UAV is identified, it is recruited by the detecting UAV through the detection-stimulated communications. The details of detection-stimulated communications are further described in Section 5.4.2 as an essential procedure of UAV cooperation.

3) Find the Diagonal Patch

The diagonal patch is defined based on the current patch of the detecting UAV. To generate a diagonal formation of two UAVs, the diagonal patch is defined to be the second patch along the diagonal line. Therefore, in the two-UAV formation, based on the current patch of the detecting UAV `patch (my-pxcor, my-pycor)`, the diagonal patch is defined to be `patch (my-pxcor + 2, my-pycor + 2)`.

In the three-UAV formation, the two diagonal patches are defined based on the current patch of the detecting UAV: `patch (my-pxcor - 1, my-pycor - 2)` and `patch (my-pxcor + 1, my-pycor - 2)`. Unlike the diagonal patch defined in the two-UAV formation, the three-UAV formation is specified so that overlapping detection coverage can be generated among the three UAVs.

In brief, a group of identical UAVs operate in a random pattern to detect initial target signal with the sensor onboard. The 360 ° sensor enables each UAV to scan a range of 3 * 3 patches on the ground. The sensor capacity defines the individual detection range of UAVs. When an initial target signal is detected, it indicates that the target could be located on any of the 9 patches. Hence the UAV records its current detection range as a list of 9 patches. Individual UAVs are unable to locate the target position on their own. Therefore, the proposed solution deploys multiple UAVs to collaborate with each other to determine the target location. The purpose of UAV cooperation is to identify target location through the diagonal formation of two UAVs and the triangular formation of three UAVs. Section 5.4.2 describes the cooperative behaviour of UAVs, and the specification of UAV formation.

5.4.2 UAV Cooperation

Initiated by an initial detection of target signal, UAVs are to carry out the following cooperative behaviour in order to locate mobile targets: the detection-stimulated communications, the converged formation of UAVs, the identification of common patch with a target signal, and the synchronised circling behaviour. Details of the UAV cooperation are described in the four sections below.

5.4.2.1 Detection-stimulated Communications

Triggered by the initial detection of target signal, the detection-stimulated communications occur to enable the detecting UAV to recruit its closest UAV. The communications are established between the detecting UAV and the closest UAV with no detection of itself. The code below shows the key procedures to be engaged in the detection-stimulated communications. When the communications succeed, the closest UAV is recruited to converge into a predefined diagonal patch.

```
if num-of-converged < 2
  [ find-closest-UAV
    if the-closest-UAV != nobody
    [ find-diagonal-patch
      if (diagonal-patch != nobody)
      [ communicate-and-converge ] ] ]
```

Hence the diagonal formation is generated between the two UAVs. Once converged, the recruited UAV updates its detection data by recording the 9 patches of its individual detection range if it detected a target signal. The procedure `communicate-and-converge` presents the process of convergence. When both UAVs are converged properly, the detection data of them is processed to see if both have a target signal. This would indicate that the target is residing on the patch which is common to their detection range.

```
to communicate-and-converge
  ask the-closest-UAV
  [ set signal? true
    set color blue
    face diagonal-patch
    move-to diagonal-patch
    update-detection-data
    set converged? true ]
end
```

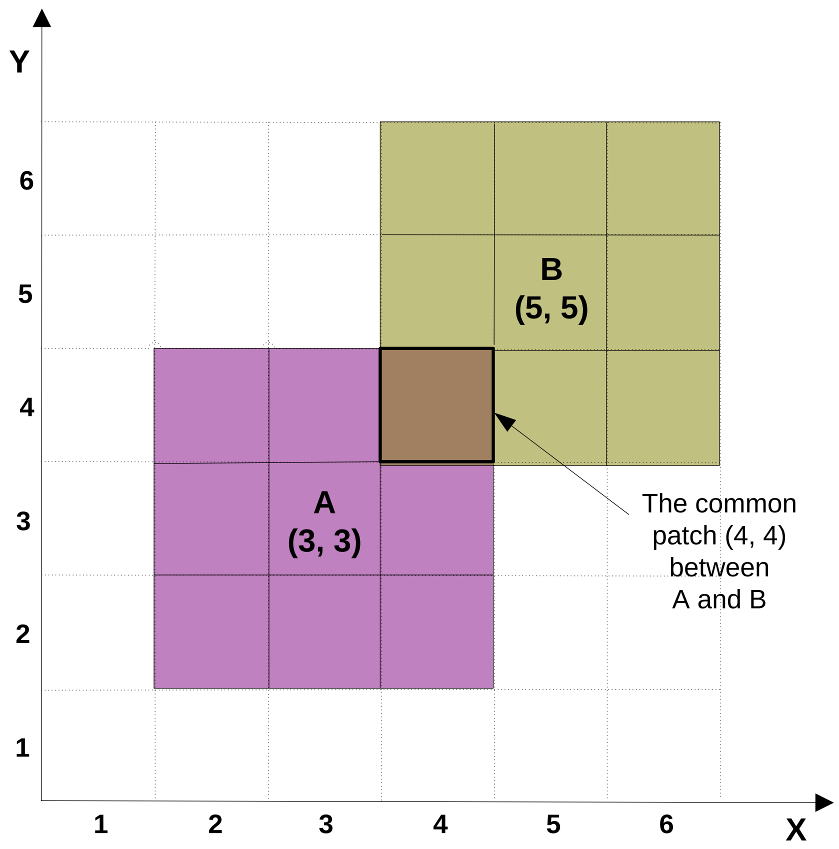
If, however, none of the UAVs are found available for recruitment and the communications cannot be established, the detecting UAV resumes random search to explore other parts of the search world.

Briefly, an initial detection of target signal triggers communications between the detecting UAV and the closest UAV. Through the communications, the detecting UAV recruits the closest UAV and the two of them converge onto a diagonal formation. If the communications have not been successful, no UAV is recruited and the detecting UAV resumes random search to explore other parts of the search world. The detection-stimulated communications are essential in terms of initiating the rest of the cooperative behaviour of UAVs.

5.4.2.2 Converged Formation of UAVs

There are two scenarios of converged formation of UAVs: the diagonal formation of two UAVs and the triangular formation of three UAVs. For the two-UAV formation, if the recruitment is successful, the detecting UAV and the recruited UAV are converged onto a diagonal formation. When converged, the individual detection range of the two UAVs is overlapped with one common patch. Figure 5.7 shows an example of the detection coverage of two-UAV formation.

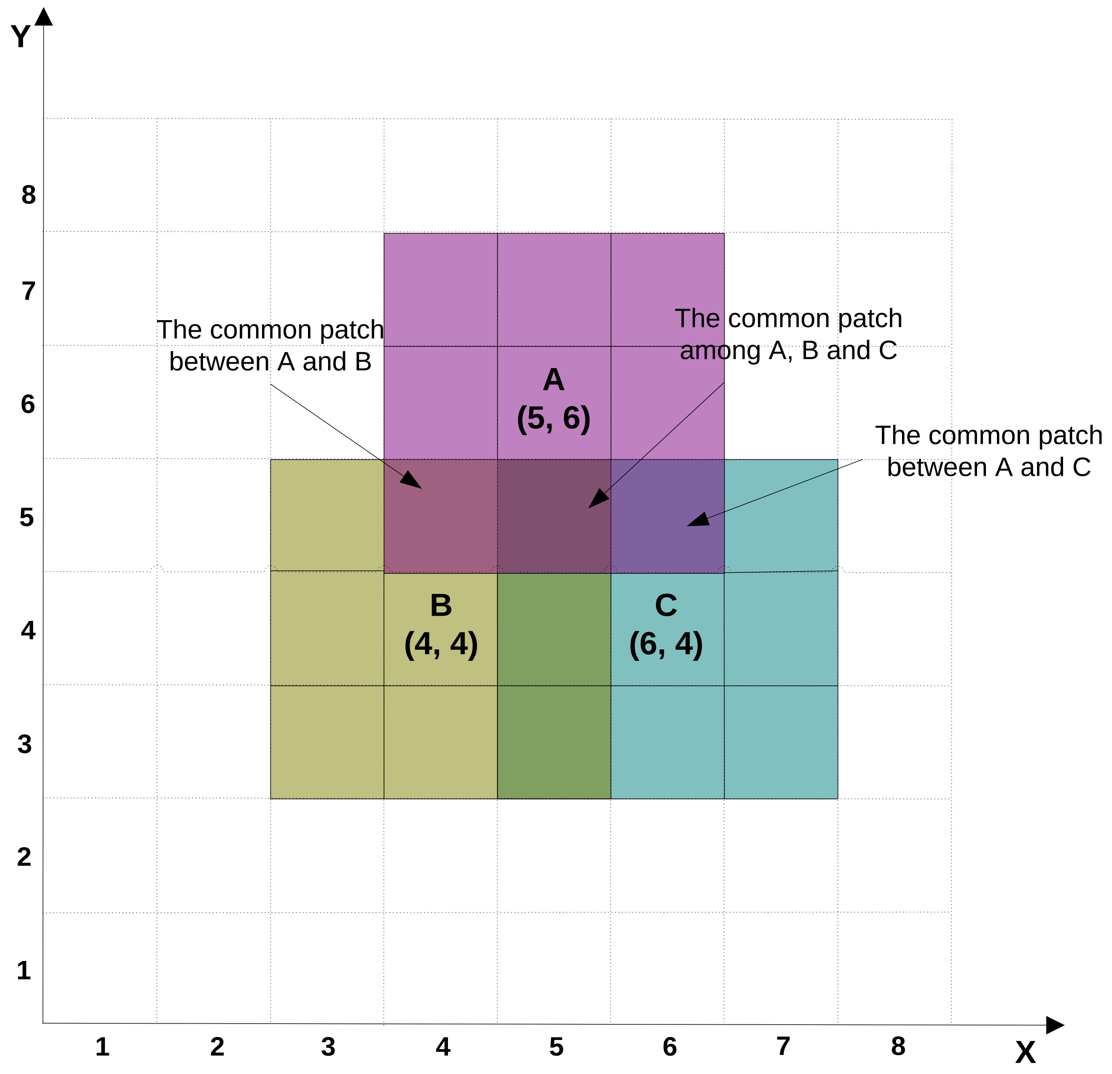
Figure 5.7 The Detection Coverage of Two-UAV Formation



In Figure 5.7, UAV A detected a target signal when it is on patch (3, 3). Through the detection-stimulated communications, it recruited UAV B to converge onto patch (5, 5).

UAV B is found to be the closest neighbour of A and has no detection of its own at the time of recruitment. Based on A's current position of patch (3, 3), patch (5, 5) is identified to be the diagonal patch. As a result, A and B are converged onto a diagonal formation. The outlined squares indicate the individual detection range of A and B.

Figure 5.8 The Detection Coverage of Three-UAV Formation



The same rules of convergence apply to the three-UAV formation. Figure 5.8 illustrates an example of the three-UAV formation. UAV A detected the target signal on patch (5, 6). It found two nearby UAVs B and C that have no initial detection of themselves at the time.

Hence, UAV B is recruited to move onto the patch (4, 4), and UAV C is recruited to move onto the patch at (6, 4). As a result, the three UAVs are converged onto a triangular formation. The formation is emerged in such a way that unique common patches are generated among the three UAVs.

5.4.2.3 Detection of the Target

The target-located patch is identified through the following procedures: first of all, the diagonal formation and the triangular formation of UAVs generate unique common patches between them. Second, if both of the UAVs have the detection of target signal, the unique common patch is identified to be the target-located patch. The two scenarios of UAV formation have different procedures of identifying the target-located patch.

1) Two-UAV Formation

As shown in Figure 5.7, patch (4, 5) is the only patch that is covered by the detection range of both UAV A and UAV B. If A and B both have the detection of target signal, the target is confirmed to be located on patch (4, 5).

2) Three-UAV Formation

For the three-UAV formation, the target-located patch is found by identifying a common patch between UAVs A and B, or UAVs A and C, both with target signal. If no unique common patch with the detection of target signal is identified between A and B, it continues to check A and C. If no unique common patch with the detection of target signal is identified between A and C, the three UAVs are unable to identify the target location and they continue with the synchronised circling operation (see Section 5.4.2.4).

In the example shown in Figure 5.8, patch (4, 5) and patch (5, 5) are the common patches between A and B. If both A and B have the detection of target signal, then it indicates that the target is located on either patch (4, 5) or patch (5, 5). One of the two patches is the unique common patch between A and B, and hence can be identified as the target-located patch. If A and B have no detection of target signal, the target-located patch could not be confirmed and UAVs A and C are next checked.

In Figure 5.8, patch (6, 5) and patch (5, 5) are common patches between A and C. If both A and C have the detection of target signal, either patch (6, 5) or patch (5, 5) can be identified to be the target-located patch (Note: Patch (5, 5) would have been already eliminated in the previous step). Otherwise, the three UAVs are unable to locate the target and they continue with the synchronised circling operation (see Section 5.4.2.4).

The identification process of the target-located patch is transformed into the NetLogo procedures and the section below describes the details of the NetLogo specification.

3) NetLogo Specification

In NetLogo, the identification of target-located patch is specified in reporter `match` and procedure `find-patch`. In the two-UAV formation, UAV A and UAV B generate two lists of patches (`List1 and List2`) that each record their individual detection range. The two lists of patches are processed to identify a unique common patch between A and B.

In the procedure `find-patch`, if a unique common patch is identified with the detection of target signal, the patch is confirmed to be the target-located patch. Therefore, the coordinates of the identified patch is the coordinates of located target.

```
to find-patch
 let List1 [ my-detection-data ] of one-of converged-UAVs
 let List2 [ my-detection-data ] of one-of other converged-UAVs
 if (List1 != 0) and (List2 != 0)
 [ set found-patch? match List1 List2
   if found-patch? = true
   [ let combined-list sentence List1 List2
     set matched-patches modes combined-list
     foreach matched-patches
     [ ask ? [ if target-signal = 1 [ set located-patch ? ] ] ]
     if located-patch != 0
     [ ask converged-UAVs
       [ output-show (word "Found a target on " located-patch " at time " ticks) ] ]
       ;; When not all targets are located, increment the number of located targets by 1
       if located-targets-count < num-of-targets
       [ set located-targets-count located-targets-count + 1 ]
     ] ] ]
end
```

In the three-UAV formation, procedure `find-patch` each checks the detection data of A (`the-detecting-UAV`) and B (`UAV-one`), A (`the-detecting-UAV`) and C (`UAV-two`), in order to identify the target location.

```
to find-patch
  let List1 [ my-detection-data ] of the-detecting-UAV
  let List2 [ my-detection-data ] of UAV-one
  let List3 [ my-detection-data ] of UAV-two
```

```
if (List1 != 0) and (List2 != 0) and (List3 != 0)
[
  set found-patch-one? match-two List1 List2
  ifelse found-patch-one? = true
  [ let combined-list sentence List1 List2
    set matched-patches modes combined-list
    foreach matched-patches
    [ ask ? [ if target-signal = 1 [ set located-patch ? ] ] ]
    if located-patch != 0
    [ ask the-detecting-UAV
      [ output-show (word "Found a target on " located-patch " at time "
    ticks) ]
      ask UAV-one
      [ output-show (word "Found a target on " located-patch " at time "
    ticks) ]
…
    if located-targets-count < num-of-targets
    [ set located-targets-count located-targets-count + 1 ]
  ]
  [ set found-patch-two? match-two List1 List3
    ifelse found-patch-two? = true
    [ let combined-list sentence List1 List3
      set matched-patches modes combined-list
      foreach matched-patches
      [ ask ? [ if target-signal = 1 [ set located-patch ? ] ] ]
      if located-patch != 0
      [ ask the-detecting-UAV
        [ output-show (word "Found a target on " located-patch " at time
    "           ticks) ]
        ask UAV-two
        [ output-show (word "Found a target on " located-patch " at time
    " ticks) ]
…
      if located-targets-count < num-of-targets
      [ set located-targets-count located-targets-count + 1 ]
```

```
      ]
    ] ]
end
```

When one of the targets is located, it is removed from the current operation. If there are more targets to be located and the maximum time length has not yet expired, the converged UAVs resume random search to continue exploring other parts of the search area. The following code shows the situations when UAVs resume random search:

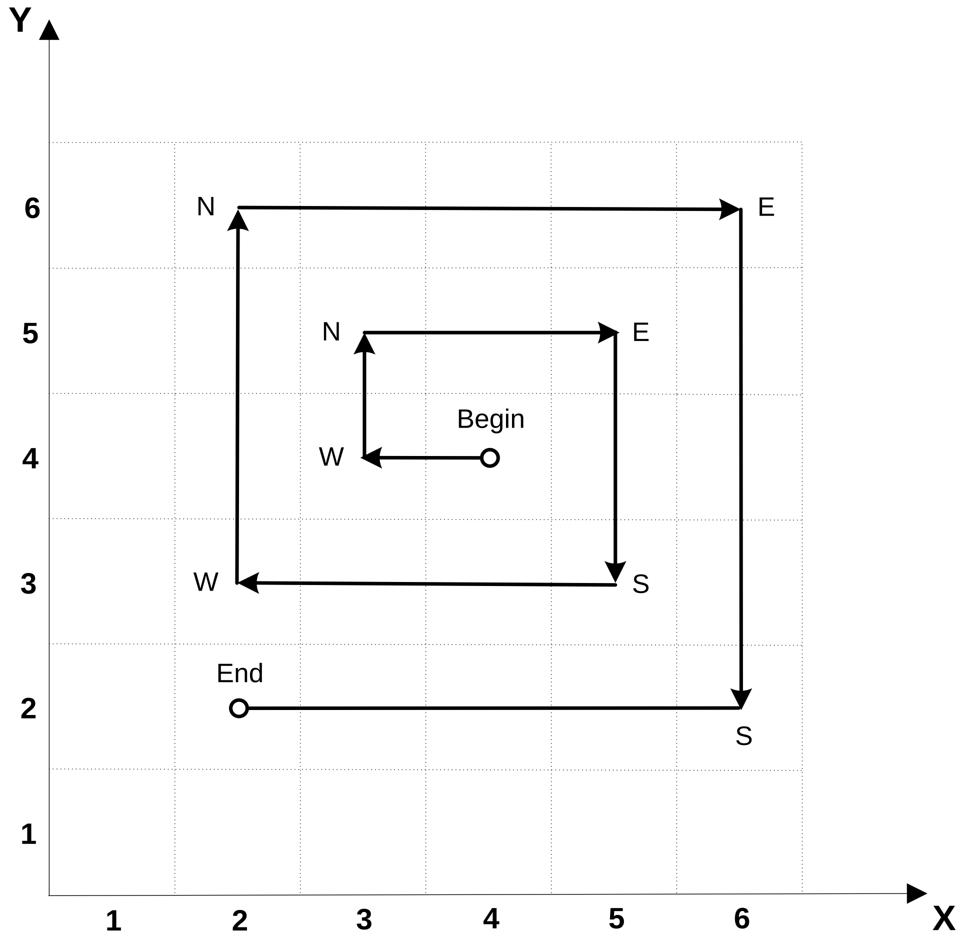
```
if located-targets-count < num-of-targets
   [ resume-random-search ]
```

For both of the two scenarios, if the target-located patch could not be identified and the maximum time length has not expired, the UAVs initiate synchronised circling behaviour aiming to track the target in a wider area. Section 5.4.2.4 describes the circling behaviour of UAVs.

5.4.2.4 Synchronised Circling Behaviour of UAVs

The circling behaviour is predefined and identical to all UAVs. While circling, the UAVs retain the original formation in order to keep the unique common patch between each other. Figure 5.9 illustrates the clockwise pattern of UAVs' circling behaviour. The UAV circles clockwise in four directions – west, north, east and south. From the UAV's perception, the circling behaviour begins from the patch where the target signal is initially detected. Depending on the current heading, the UAV moves one patch forward (west) and turns right to move one patch upwards (south). It continues to move two patches to the right (east), two patches down (south), and turns right again to move three patches forward (west). Then it keeps turning right and moves three patches upwards (north), four patches on the right (east), four patches down (south), turning right again (west) for another four patches to complete the circling.

Figure 5.9 The Clockwise Pattern of UAVs Circling Behaviour



In total there are 25 patches defined for each UAV throughout the circling process. According to the turning angles and distances, each of the patches is defined as below:

```
let patch-one patch-right-and-ahead -90 0.5
let patch-two patch-right-and-ahead 0 0.5
let patch-three patch-right-and-ahead 90 0.5
let patch-four patch-right-and-ahead 90 1
let patch-five patch-right-and-ahead -180 0.5
let patch-six patch-right-and-ahead -180 1
```

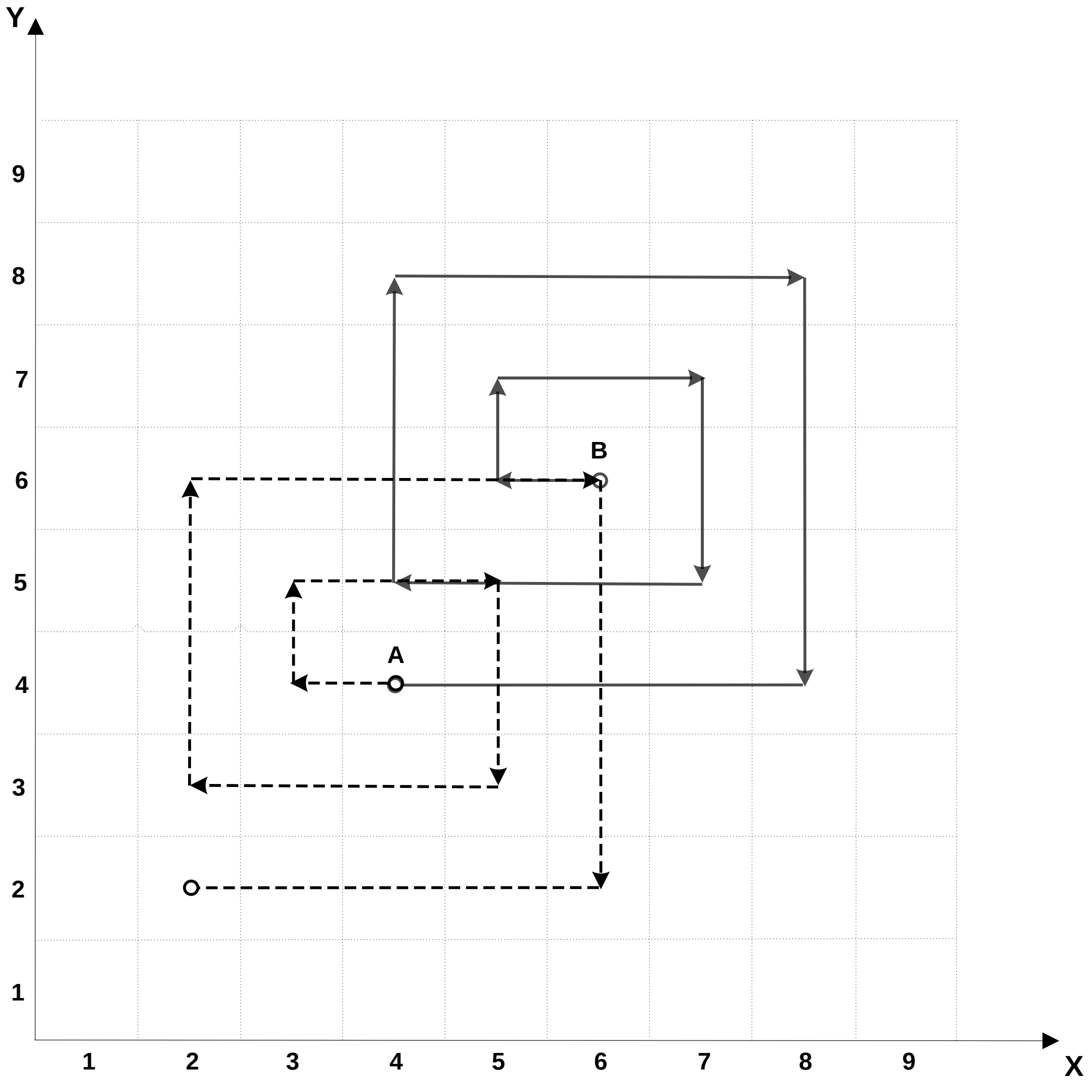
…

Moving from one patch to the next patch, at every step the UAV is able to cover different ranges of patches as its individual detection range shifts accordingly. Hence, each time the UAV moves onto a new patch, it updates the individual detection range with a new list of 9 patches. The following code specifies how one of the circling UAVs is moving from patch to patch.

```
face patch-one move-to patch-one
update-detection-data
find-patch
if found-patch? = false
  [ face patch-two move-to patch-two
    update-detection-data
    find-patch
    if found-patch? = false
      [ …
```

Figure 5.10 shows the synchronised circling behaviour of the two UAVs in the diagonal formation. Converged in a diagonal formation, UAV A and UAV B circle clockwise at the same time. In such a way they are able to retain the diagonal position while circling. Every time the detection data of A and B is updated, it is used to identify the unique common patch between the two UAVs. Once the patch is found with a target signal, it is reported to be the target-located patch. When the target is located, the UAVs are dismissed from the current circling activity to resume a new search operation.

Figure 5.10 The Synchronised Circling Behaviour of UAV A and UAV B



If, however, the unique common patch cannot be identified when A and B moved to the last patch of the circling pattern, it indicates that the target has moved away from their detection range. A and B switch from the current circling pattern to resume random search.

```
if found-patch? = false
  [ face patch-twenty-four move-to patch-twenty-four
    update-detection-data
    find-patch
    if found-patch? = false
    [ resume-random-search ] ]
```

And finally, the search operation terminates when all targets are located.

```
if located-targets-count = num-of-targets
  [ output-show (word "The total detection time is " ticks)
    if num-of-success = num-of-targets
    [ set full-success-detection full-success-detection + 1
      output-show (word "All targets are located successfully") ]
    stop ]
```

Furthermore, the search operation is also terminated if UAVs are unable to complete the target detection within the maximum time length. In such a case, the maximum time length is recorded as the total search time of the target detection.

5.5 Summary

To summarise, the research problem is about multiple UAVs deployed to locate multiple mobile targets within a specific time. The proposed solution defines following search behaviour of UAVs:

- Initial Detection of Target Signal
- Detection-Stimulated Communications
- Diagonal Formation and Triangular Formation of Converged UAVs
- Identification of the Target-Located Patch
- Synchronised Circling Behaviour of UAVs

At the initial stage of the search operation, UAVs search for target signal in a random flight pattern. An initial detection of target signal stimulates local communications between the detecting UAV and its closest neighbours. The detection-stimulated communications enable multiple UAVs to converge onto two scenarios of the UAV formation.

In the scenario of two-UAV formation, the detecting UAV recruits the closest UAV through the detection-stimulated communications. If successful, the two UAVs are converged in a diagonal formation. In the scenario of three-UAV formation, the detecting UAV recruits two nearby UAVs and the. The different formations of UAVs are designed to generate unique common patch amongst converged UAVs. If the unique common patch is identified and all of the converged UAVs have the detection of target signal, the patch is the target-located patch. The coordinates of the target-located patch is the coordinates of the target position.

If no common patch can be identified, converged UAVs initiate synchronised circling behaviour aiming to track the moving target in a wider area. UAVs retain the original formation while circling. The individual detection range of each UAV changes at every step of the circling behaviour, so that UAVs continue to update their detection data while circling. At each step of the circling process, the detection data of UAVs is processed to identify the target-located patch. Once the target-located patch is identified, if there are more targets to be located and the maximum time length is not expired, UAVs resume random search operation to continue looking for new targets. If no patch can be identified at the end of the circling process, UAVs resume random flight pattern and initiate a new round of target detection.

The search operation terminates when all targets are located and/or the predefined maximum time length is exceeded.

In the next chapter, the author evaluates the proposed search solution through running a series of experiments of the search model, as well as measuring the search performance of UAVs in the two cooperative scenarios. Chapter 5 introduces the experimental setups, the implementation of experiments, and also the analysis and discussion of the experimental results.

Chapter 6. Design of Experiments and Results

Last chapter proposed a swarm-inspired search solution for the cooperative search of multiple UAVs and constructed a search model in a simulation programme. In this chapter the proposed search solution is evaluated through the design and implementation of a series of experiments. The evaluation is carried out according to two performance criteria: one is the average detection time, and the other is the number of targets detected within the maximum time length (also known as all-located target detection). The former is to assess the efficiency of target detection, and the latter is to assess the detection effectiveness of UAVs. The experiments are setup with different combinations of multiple targets and multiple UAVs. For each combination of targets and UAVs, the experiments are implemented each with the two scenarios of UAV cooperation: the diagonal formation of two UAVs and the triangular formation of three UAVs. For each of the scenarios, the experiments are setup in the following two ways:

1. Fixed Numbers of Targets vs. Increasing Numbers of UAVs

Firstly, increase the number of UAVs on a constant basis while the number of targets remains the same. It aims to show the impact of the numbers of UAVs on the performance of target detection.

2. Fixed Numbers of UAVs vs. Increasing Numbers of Targets

Secondly, increase the number of targets while the number of UAVs remains unchanged. It intends to explore the maximum search capacity of different numbers of UAVs.

The experiments of UAV search model are setup and implemented using the BehaviorSpace in NetLogo. BehaviorSpace provides a platform to set up the experiments of models, and automatically runs a model as many times as needed. When the experiments are setup, BehaviorSpace adjusts the model's settings and records the results of each model run. It enables the model behaviour to be observed ad hoc, and hence to explore potential capability of the model. Additionally, the tool supports parallel runs of the experiments and provides different output formats to record the experimental data.

Section 6.1 presents the experimental setup and implementation of the search model. The simulation model of the cooperative search of UAVs is setup for two cases: the single target detection and the multiple target detection. For each search case, the model is implemented in the two scenarios of UAV cooperation. The experimental data is recorded in a spreadsheet format and presented in tables and figures. Section 6.2 analyses the experimental data and discusses the search performance of the proposed search solution based on the two criteria of performance measurement.

6.1 Experimental Setup and Implementation

Section 6.1.1 describes the experimental setup and the implementation of the search model in the case of the single target detection. The experiments increase the number of UAVs while the number of targets remains the same. Section 6.1.2 describes the experimental setup and the implementation of the search model in the case of multiple target detection. Following each implementation of the experiments, it presents and explains the experimental data of the two scenarios of UAV cooperation. And finally, the two scenarios are compared based on the two performance criteria.

6.1.1 Search Case 1: Single Target Detection

Experimental Setup:

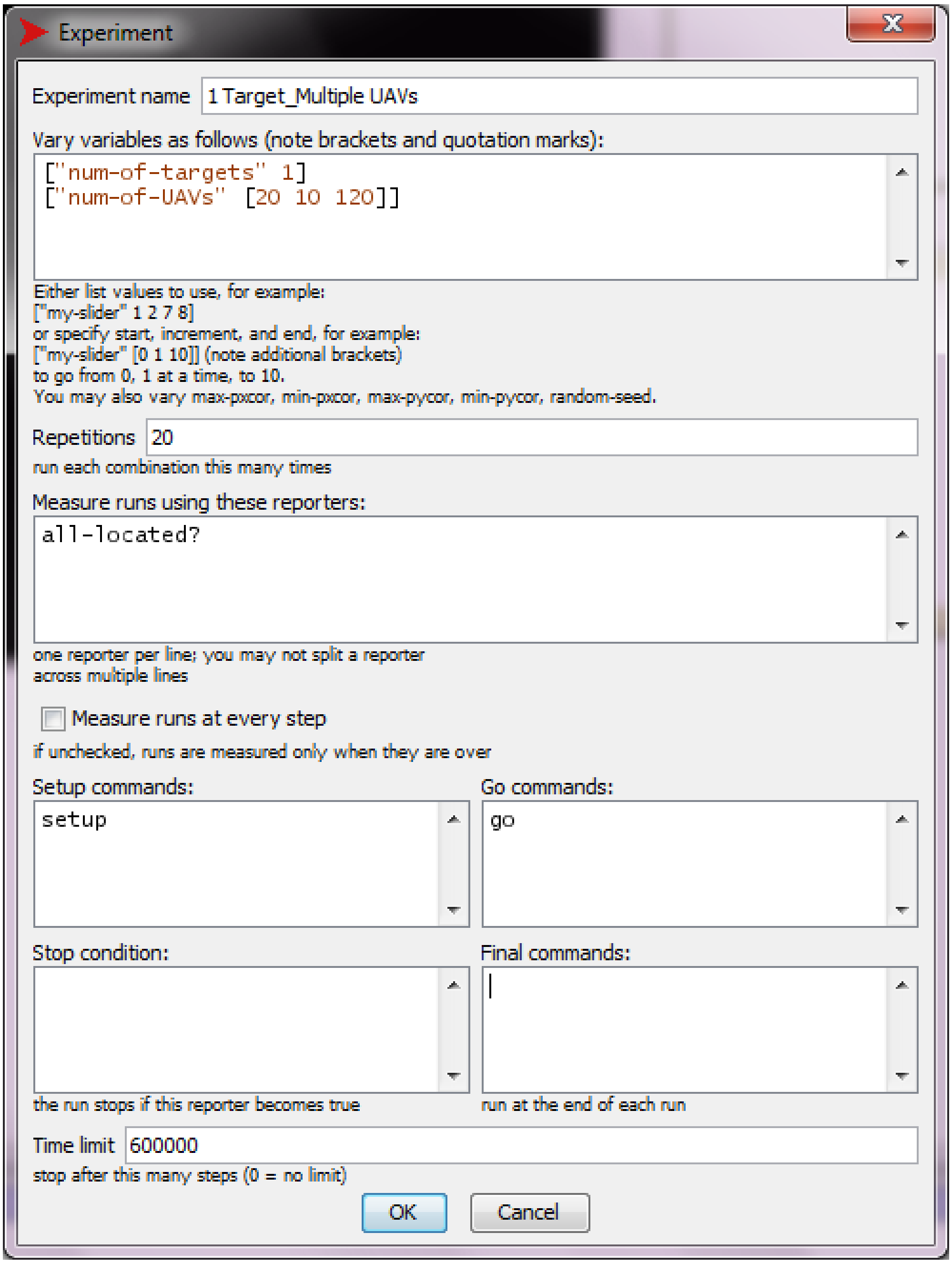
Fixed Numbers of Targets vs. Increasing Numbers of UAVs

The experiments of single target detection are implemented to investigate the impact of the number of UAVs on the search performance of the single target detection; first to show the relationships between the search performance and the number of UAVs for each scenario of the UAV cooperation; and second to compare the search performance of the two scenarios of UAV cooperation.

6.1.1.1 The Setup of Experiments

Multiple UAVs are deployed to search for the single target. The number of UAVs is incremented by 10 each time from a minimum of 20 UAVs to a maximum of 120 UAVs. Thus it makes up a total of 11 groups of UAVs to run the experiments with. The number of targets equals to 1 and remains the same for all the 11 groups of UAVs. Figure 6.1 shows the experimental setup of the single target detection in BehaviorSpace.

Figure 6.1 Experimental Setup for the Single Target Detection



Each combination of the experiment is replicated for 20 runs. The detection time[7] is recorded at the end of each experimental run. When all the 20 experimental runs are completed, the average detection time for each combination of experiments is calculated. For the single target detection, the average detection time refers to the average time consumption of UAVs

[7] The time passes in discrete steps in the search model, which is recorded by a built-in tick counter in NetLogo. The tick counter is an integer, which starts at 0 and advances 1 at a time.

in order to locate the single target over the 20 experimental runs; for the multiple target detection (introduced in Section 6.1.2), the average detection time refers to the average time consumption of UAVs in order to locate all the targets in the search area over the 20 experimental runs. In each case, if a target is not located within the maximum time length, its detection time is set to the maximum time length (600,000) for simplicity (NB: the average detection time in such cases would thus be underestimated).

In addition, each experimental run is also measured by a reporter called "`all-located?`" that reports TRUE if the target is located by UAVs within the maximum time length. Otherwise it reports FALSE. The number of targets detected within the maximum time length (i.e. the number of "TRUE"s) is recorded for all of the 20 experimental runs of each combination; as one of the two criteria of the experiments it is recorded as the all-located target detections.

In the BehaviorSpace, the maximum time length is predefined as the time limit that is used to terminate each of the experimental runs. While experimenting with the two-UAV formation and three-UAV formation, it was found that UAVs are able to locate all or most of targets within 600,000 steps. Lengthy but ineffective experimental runs would not produce meaningful data. Trial experiments showed that 600,000 steps is an 'optimal' value. If it is less than 600,000 steps, three-UAV formation would not have sufficient time to converge and to locate targets in most cases. If it is longer than 600,000 steps, the implementation of experiments would become lengthy without adding much value. Therefore, the maximum time length is set to 600,000 steps. If the maximum time length is exceeded, the target detection would be terminated whether or not UAVs have located all of the targets. In such a case, the maximum time length is recorded to be the detection time of that target detection. Each step could stand for any appropriate physical time such as hours, minutes and seconds, and thus can be deployed in real-time applications.

At the end of each combination of the single target detection, the average detection time and the number of targets detected within the maximum time length are recorded to measure the search performance of UAVs.

Once setup, the experiments are implemented in the order shown in Table 6.1.

Table 6.1 The Order of Experimental runs of the Single Target Detection

Order Number	Number of Targets	Number of UAVs	Repetitions
1	1	20	20
2	1	30	20
3	1	40	20
4	1	50	20
5	1	60	20
6	1	70	20
7	1	80	20
8	1	90	20
9	1	100	20
10	1	110	20
11	1	120	20

6.1.1.2 Implementation and Results of Experiments

The experiments of the single target detection are implemented with the two scenarios of UAV cooperation. Section 1) describes the experimental results of the two-UAV formation, and Section 2) describes the experimental results of the three-UAV formation.

For both scenarios, the average detection time is calculated from 20 repetitions for each combination of targets and UAVs, which records the time passes in discrete steps. The number of targets detected with the maximum time length (i.e. the all-located target detections) records the number of "TRUE"s reported by the reporter "`all-located?`" for each experiment combination.

The experimental data of all the figures in this chapter is recorded in table format and is presented in Appendix A – 2.

1) Two-UAV Formation

Figure 6.2 The Average Detection Time with Two-UAV Formation

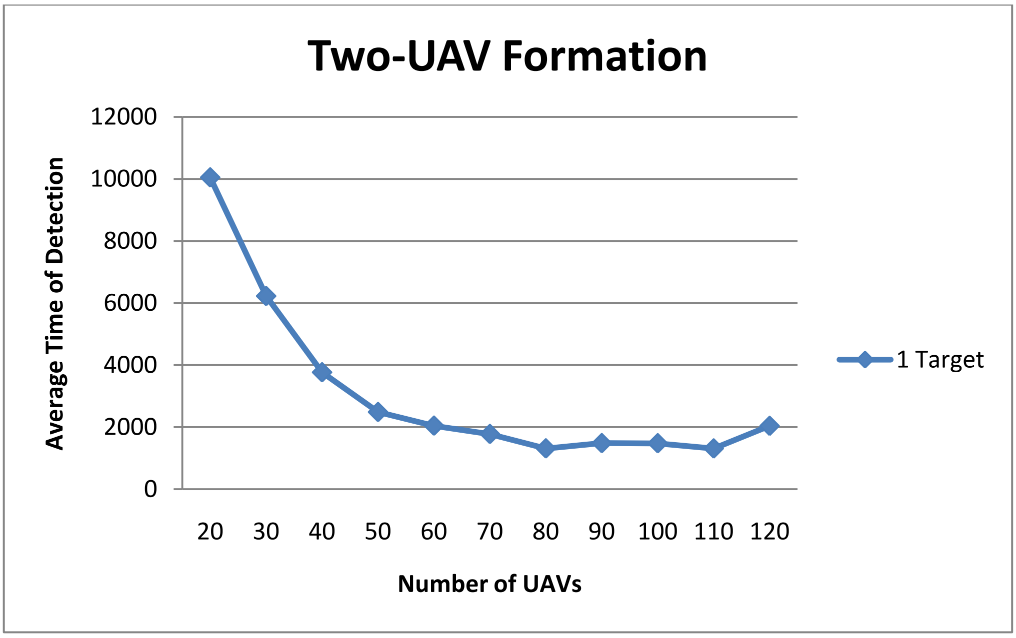


Figure 6.2 illustrates the average detection time of UAVs to show the relationship between the number of UAVs and the average time consumption. The data pattern indicates that the

average detection time decreases as the number of UAVs continues to increase. Both the data table and the figure indicate that, the more UAVs deployed the less time to be consumed to locate the target. This is true till the number of UAVs has increased to a certain value (also see Table A1 in Appendix A – 2).

Regarding the all-located target detections, the experimental data indicates that UAVs are able to locate the target for all the experiments of the single target detection.

2) Three-UAV Formation

Figure 6.3 The Average Detection Time with Three-UAV Formation

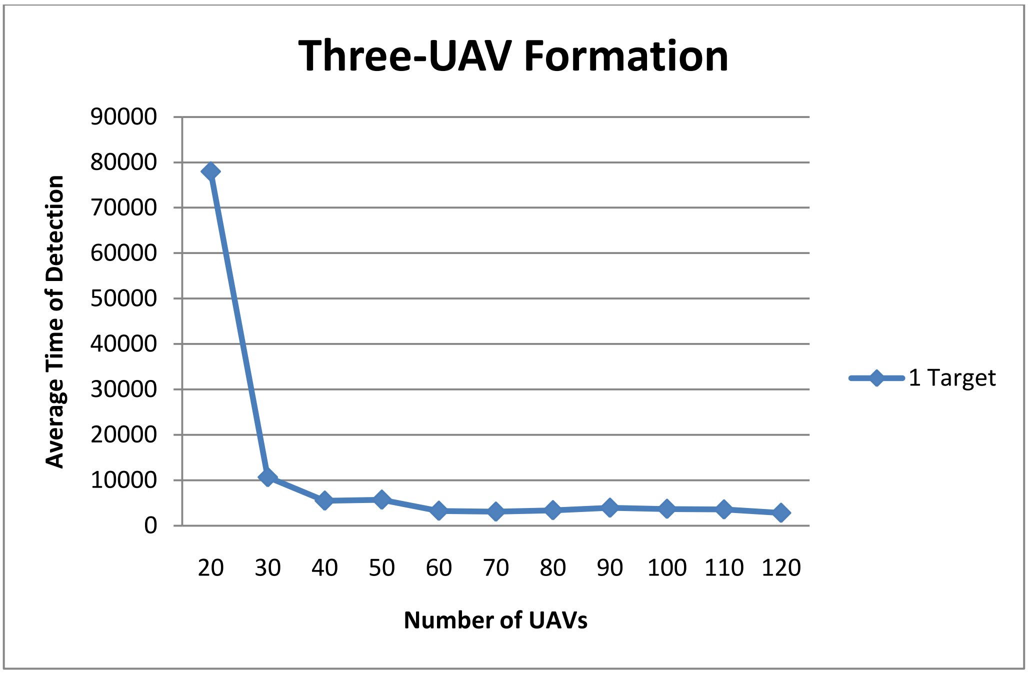


Figure 6.3 illustrates the average detection time consumed by UAVs to carry out the single target detection with the three-UAV formation. It shows that there is a significant decrease in the average detection time when the number of UAVs has increased from 20 to 30. It took 20

UAVs more than 78000 steps on average to locate the target, and the average detection time decreased to 10654.85 steps when the number of UAVs increased to 30. The reduction of average detection time continues, but to a lesser extent when there are more UAVs deployed in the search operation. It also shows that the impact of increasing UAV quantities has little significance on the decrease of time consumption when there are more than 60 UAVs (see Table A2 in Appendix A – 2).

The same as the two-UAV formation, UAVs are able to locate the target within the maximum time length for each experimental run.

3. Comparison of the Two Scenarios in the Single Target Detection

Criteria One: Average Detection Time

The average detection time of the two scenarios of UAV cooperation are first compared to identify which of the scenarios is more efficient to accomplish the single target detection. Table 6.2 presents the average detection time of 20 UAVs, 70 UAVs and 120 UAVs each with the two-UAV formation and the three-UAV formation. These three groups are selected as each of them represents the minimum, middle and maximum values of UAV quantities.

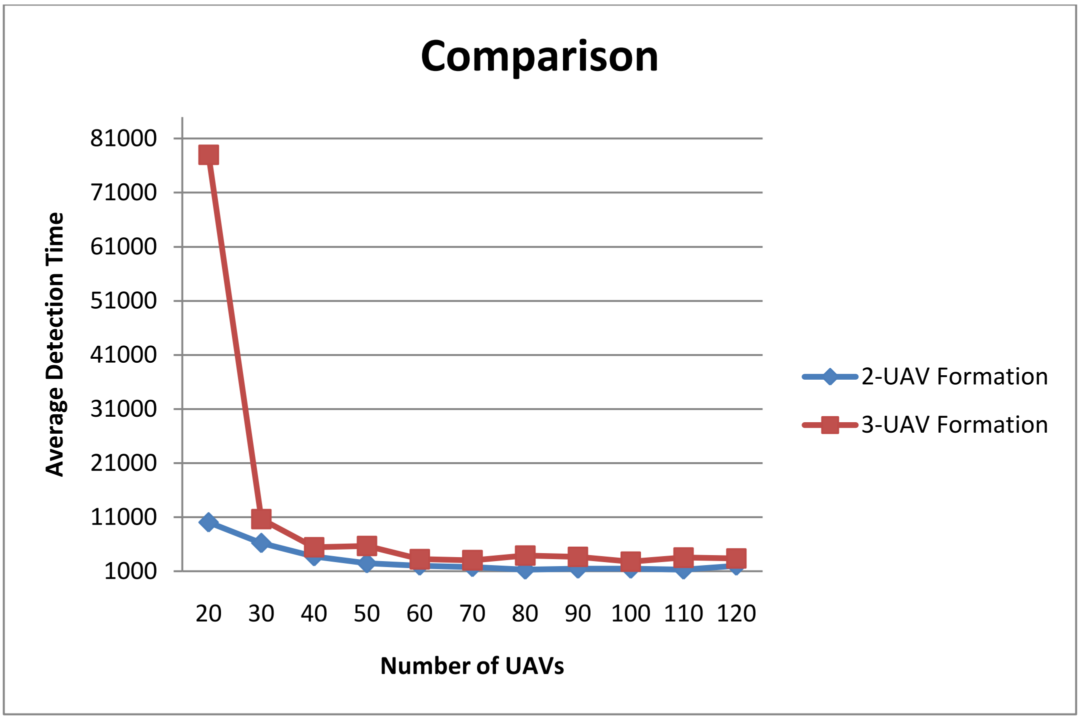
Table 6.2 The Average Detection Time of the Two Scenarios

	Two-UAV Formation			**Three-UAV Formation**		
Number of UAVs	20	70	120	20	70	120
Average Detection Time	10045.9	1778.55	2039	78007.8	3053.7	2794.35

As indicated in the table, with the same numbers of UAVs, the two-UAV formation consumed much less time on average than the three-UAV formation to complete the single target detection. Such a difference of the two scenarios is particularly distinctive when there are only 20 UAVs. The data indicate that the average detection time of 20 UAVs in the three-UAV formation is 7 times more than that for the two-UAV formation.

Figure 6.4 illustrates graphically the average detection time of the two scenarios in the single target detection. It shows that the two-UAV formation requires lower detection time on average and less number of UAVs than the three-UAV formation to accomplish the single target detection. Therefore, the two-UAV formation is more efficient than the three-UAV formation in the single target detection. Incrementing the number of UAVs decreases the average detection time with both scenarios of the UAV cooperation.

Figure 6.4 The Average Detection Time of the Two Scenarios



Criteria Two: All-Located Target Detections

The experimental data of the single target detection indicate that, the two-UAV formation and the three-UAV formation have equally successful performance on the number of all-located target detections. Both of the two scenarios enable UAVs to locate the target within the maximum time length.

6.1.2 Search Case 2: Multiple Target Detection

There are two ways to setup the experiments of the multiple target detection. The first experimental setup is the same as the single target detection, in which the number of UAVs is incremented on a constant basis while the number of targets remains the same. The second experimental setup is to increment the number of targets while the number of UAVs remains unchanged. For each of the cooperative scenarios of UAVs, the objective is to show the relationships between the search performance of different numbers of UAVs and the number of targets, and then to compare the search performance of the two cooperative scenarios of UAVs.

6.1.2.1 The Setup of Experiments

Experimental Setup One:

Fixed Numbers of Targets vs. Increasing Numbers of UAVs

Having implemented the first experimental setup in the single target detection, the next step is to run the same experimental setup in the case of multiple target detections. The number of targets is varying from a minimum of 2 targets to a maximum of 5 targets. The same as in the case of the single target detection, there are 11 groups of UAVs deployed to carry out the multiple target detection. For each combination of targets and UAVs, the number of targets remains the same while the number of UAVs increments from 20 to 120. Each of the 11 groups of UAVs is to carry out the same detection of 2 targets, as well as the detection of 3 targets, 4 targets and 5 targets.

Table 6.3 lists out the order of the experimental runs of the multiple target detection. There are a total of 220 experimental runs to carry out. At the end of each experimental run, the detection time and the reporter of the all-located target detections are recorded as the two criteria of the performance measurement. The maximum time length (i.e. the time limit) is also set to be 600,000 steps.

Table 6.3 The Order of Experimental runs for Multiple Target Detection

Order Number	Number of Targets	Number of UAVs	Repetitions[8]	Total Runs
1, 2, …, 11	2	20, 30, 40, …, 120	20	220
1, 2, …, 11	3	20, 30, 40, …, 120	20	220
1, 2, …, 11	4	20, 30, 40, …, 120	20	220
1, 2, …, 11	5	20, 30, 40, …, 120	20	220

[8] Repetitions per combination of targets and UAVs

Experimental Setup Two:

Fixed Numbers of UAVs vs. Increasing Numbers of Targets

In addition to the first experimental setup, the second experimental setup is also applied to the multiple target detection. It increases the numbers of targets while the numbers of UAVs remain unchanged. Each group of UAVs is to detect multiple targets and the numbers of targets increase from a minimum of 2 targets to maximum of 5 targets.

Table 6.4 (a.) uses the group of 20 UAVs as the example to show the order of experimental runs of the second setup. The numbers of targets change from 2 to 5 while the number of UAVs remains at 20 for each case. Table 6.4 (b.) presents the order of experiment runs for all of the 11 groups of UAVs.

Table 6.4 The Order of Experiment Runs of Experimental Setup Two

a. **For the Group of 20 UAVs**

Order Number	Number of UAVs	Number of Targets	Repetitions
1	20	2	20
2	20	3	20
3	20	4	20
4	20	5	20

b. **For the 11 Groups of UAVs**

Order Number	Number of UAVs	Number of Targets	Repetitions[9]	Total Runs

[9] Repetitions per combination of UAVs and targets

1, 2, 3, 4	20	2, 3, 4, 5	20	80
1, 2, 3, 4	30	2, 3, 4, 5	20	80
1, 2, 3, 4	40	2, 3, 4, 5	20	80
1, 2, 3, 4	50	2, 3, 4, 5	20	80
1, 2, 3, 4	60	2, 3, 4, 5	20	80
1, 2, 3, 4	70	2, 3, 4, 5	20	80
1, 2, 3, 4	80	2, 3, 4, 5	20	80
1, 2, 3, 4	90	2, 3, 4, 5	20	80
1, 2, 3, 4	100	2, 3, 4, 5	20	80
1, 2, 3, 4	110	2, 3, 4, 5	20	80
1, 2, 3, 4	120	2, 3, 4, 5	20	80

Having explained how the experiments are designed for multiple target detection, the next section presents the experimental results of the two scenarios of cooperative search strategies. Each scenario is implemented for the two experimental setups and results are collected accordingly.

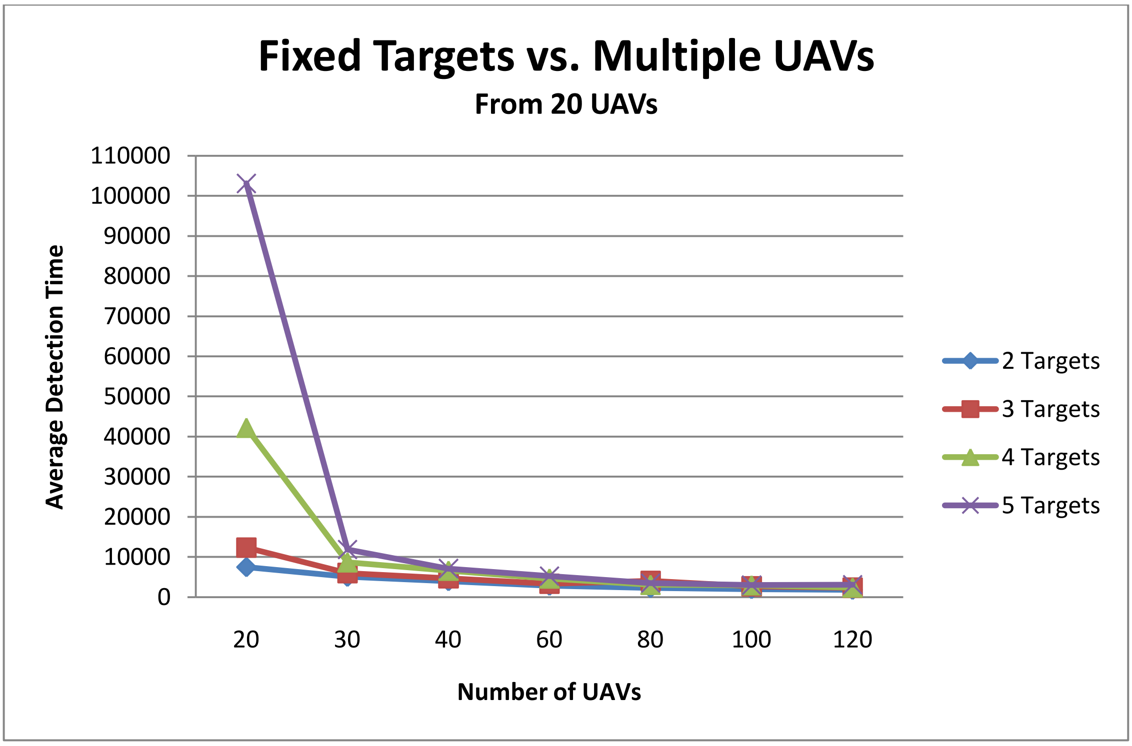
6.1.2.2 Implementation and Results of Experiments

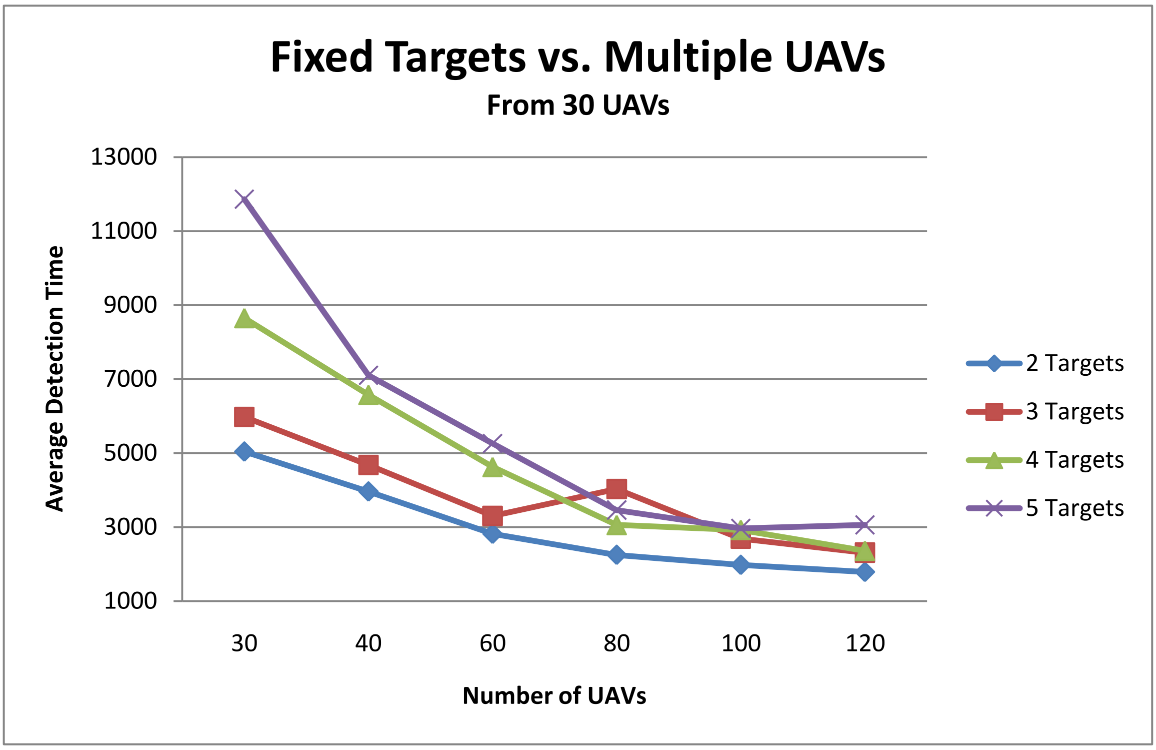
1) Two-UAV Formation

Figure 6.5 illustrates the average detection time of UAVs with the two-UAV formation for the two experimental setups. The results of both experimental setups indicate that, as the number of UAVs is incremented by 10 at a time, the average detection time decreases continually. For instance, a significant decrease of average detection time occurred when the number of UAVs increased from 20 to 30 in the case of 5-target detection. The experimental data (refer to Table A3a in Appendix A – 2) indicate that, it took 20 UAVs over 40,000 steps on average to locate all of the 5 targets while it only took 30 UAVs less than 15,000 steps to complete the same operation. For a better view of the data pattern, Figure 6.5a presents the results of experimental setup one in two different ranges: the first chart includes 20 UAVs and the second chart starts from 30 UAVs onwards. The second chart provides a detailed perspective to show the decrease of average detection time with the increase of numbers of UAVs. Figure 6.5b illustrates the results of experimental setup two, which also indicates that compared to 30 UAVs, it took 20 UAVs a prolonged period of time on average to locate multiple targets.

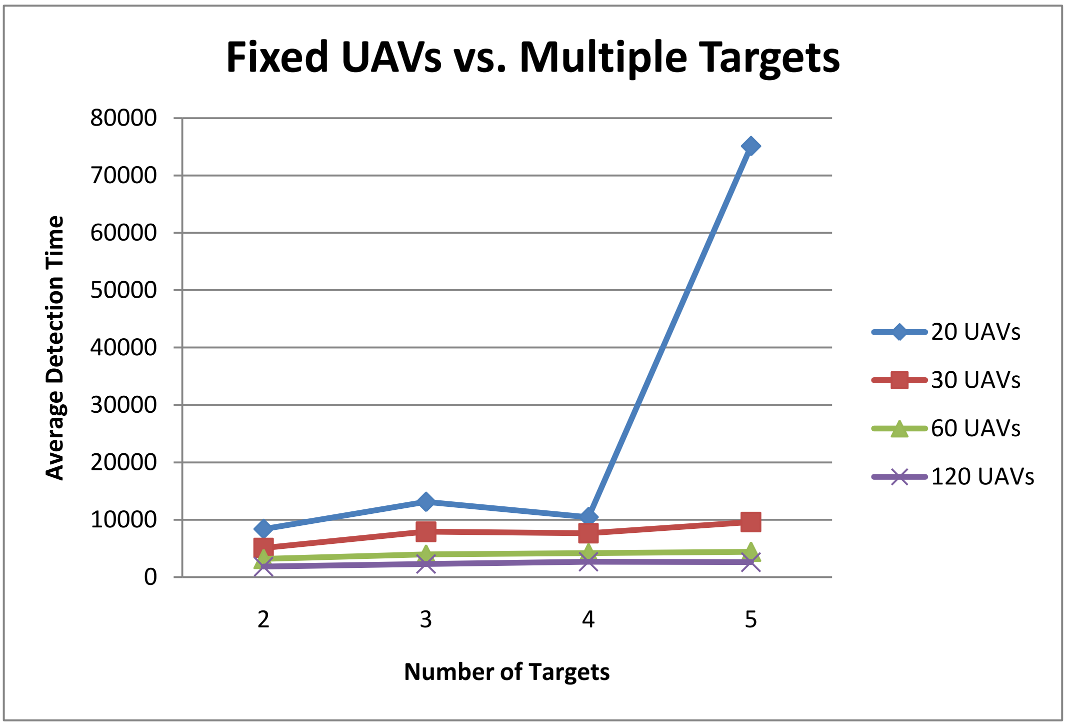
Figure 6.5 The Average Detection Time with Two-UAV Formation

a. Experimental Setup One





b. Experimental Setup Two



As for the all-located target detections, one of the experimental runs for the 20 UAVs/5 targets combination in the case of the fixed targets vs. multiple UAVs setup (i.e. Setup One) consumed more than 600,000 steps and was able to locate only 4 targets within the maximum specified time (see Tables A3a and A3b in Appendix A-2). This reflects constrained capabilities caused when the minimum number of UAVs (20) is used to locate the maximum number of targets (5). Except in this one experimental run, all other cases of multiple target detection with the two-UAV formation were successful, i.e. UAVs are able to locate all the targets within the maximum time length for the scenario of two-UAV formation.

The experimental data suggest that, the same as in the single target detection, the more UAVs deployed the less time is consumed to complete the multiple target detection. Furthermore, such a result can also be used to estimate the optimal value of UAV quantities needed in order to locate different numbers of targets within a specific length of time. For instance, as indicated by the experimental data (see Table A3 in Appendix A – 2), in the case of 5-target detection, it needs at least 40 UAVs to locate all of the 5 targets within 10,000 steps on average. To complete the 5-target detection within 5,000 steps on average, it needs a minimum of 80 UAVs to be deployed in the operation.

Next, the scenario of three-UAV formation is implemented.

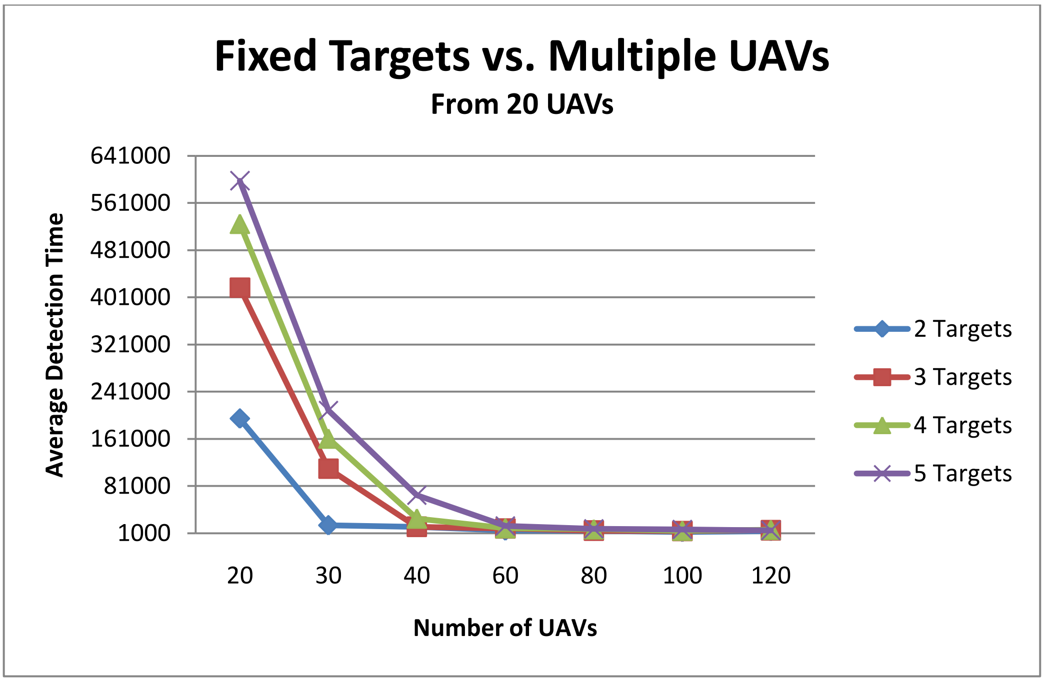
2) Three-UAV Formation

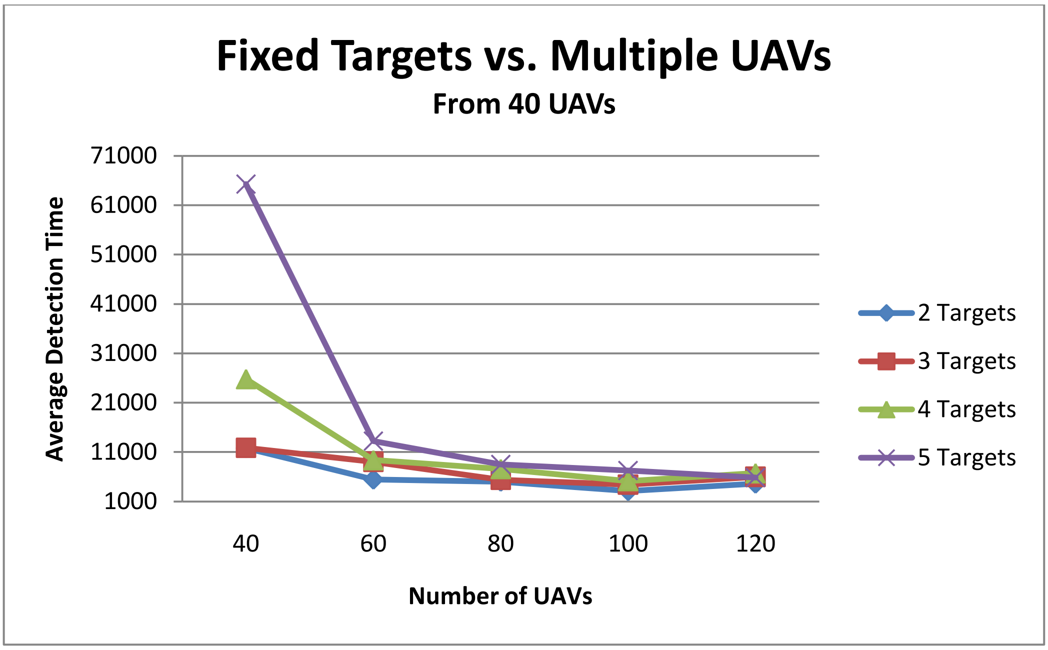
In the scenario of the three-UAV formation, not all the targets are located within the maximum time length. The search operation is terminated when the maximum time length is exceeded. In those cases the maximum time length is exceeded, the total detection time is recorded to be the pre-specified value of the maximum time length. The detection time is then calculated to produce the average detection time at the end of 20 experimental runs for each combination of experiments.

Figure 6.6 shows the average detection time consumed by UAVs to complete the multiple target detection with the two experimental setups. To provide a better visualisation of the illustration, two ranges of UAVs each from 20 UAVs and 40 UAVs are presented for each experimental setup. The same as in the case of the two-UAV formation, the experimental data indicate that the average detection time decreases while the number of UAVs increases (see Table A4 in Appendix A – 2).

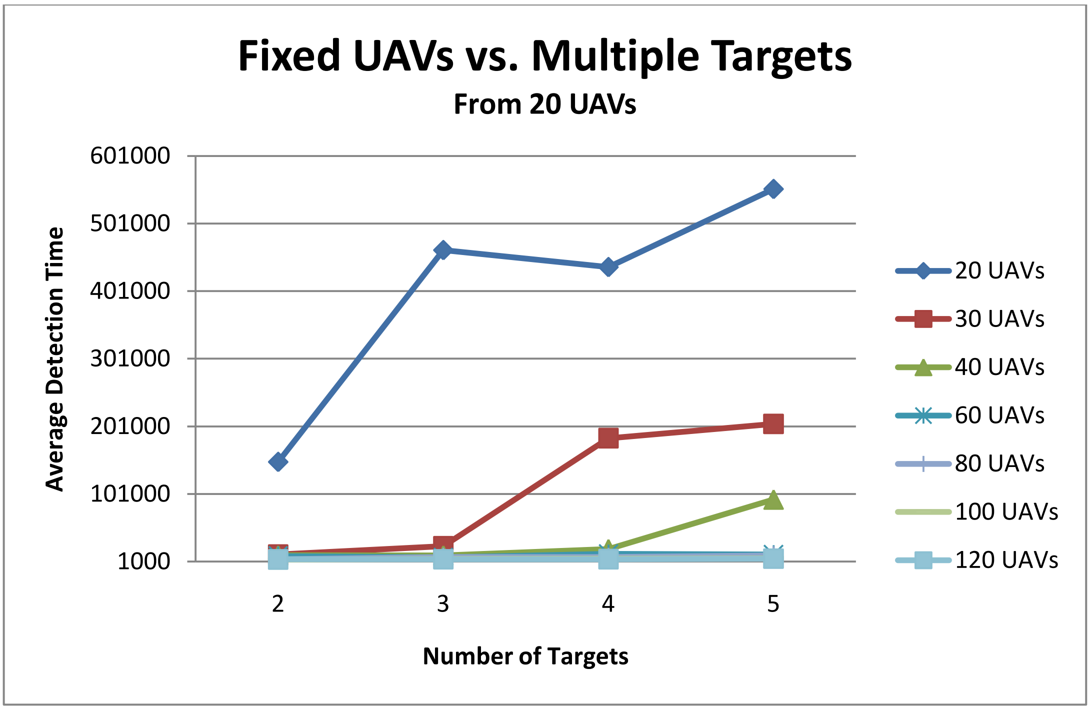
Figure 6.6 The Average Detection Time with Three-UAV Formation

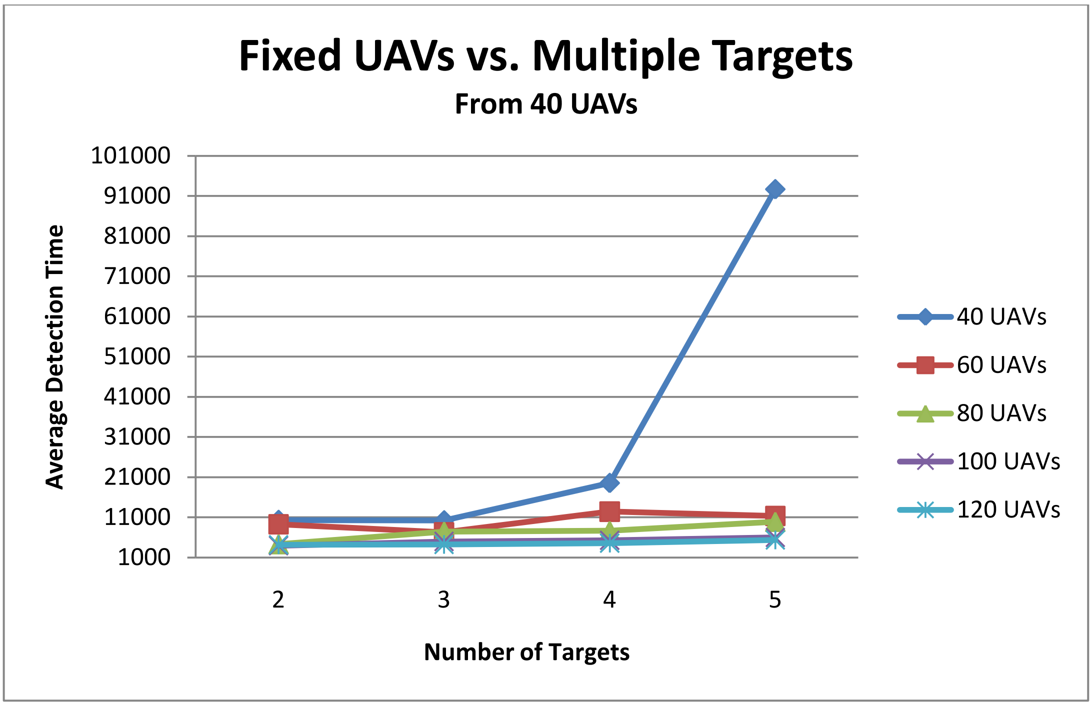
a. **Experimental Setup One**





b. Experimental Setup Two



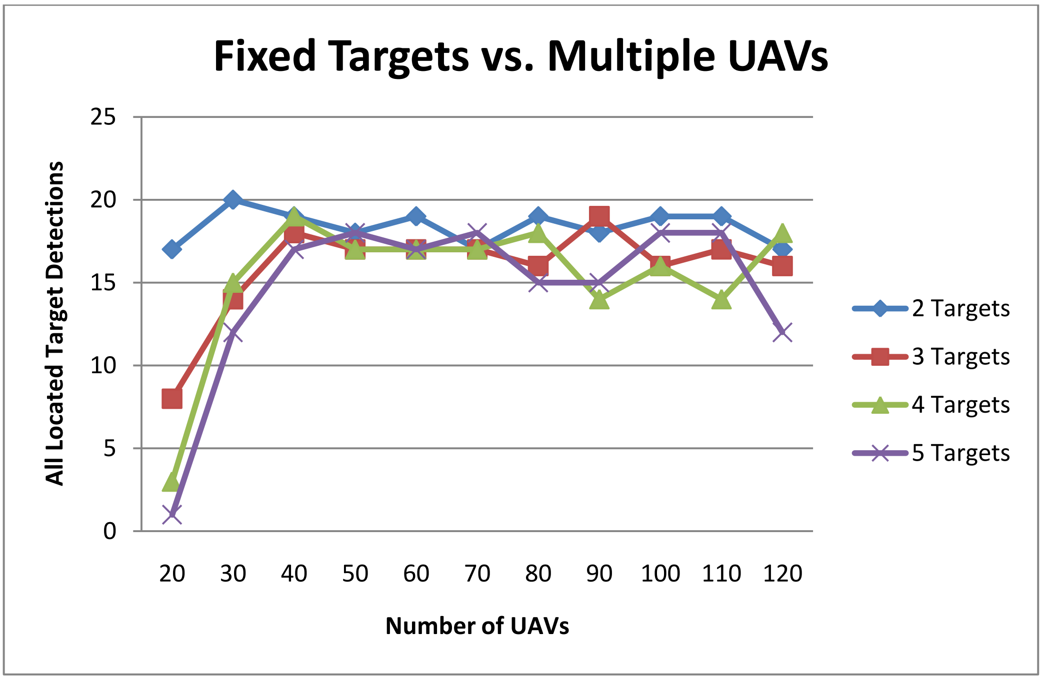


In addition to the average detection time, Figure 6.7 illustrates the number of all-located target detections of the three-UAV formation. The number of all-located target detections records the number of experimental runs that located all of the targets within the maximum time length. The number of all-located target detections varies from 0 to 20; 0 refers to none of the 20 experimental runs have successfully located all the targets, and 20 refers to all the 20 experimental runs are successfully located all the targets (see Tables A1 – A4).

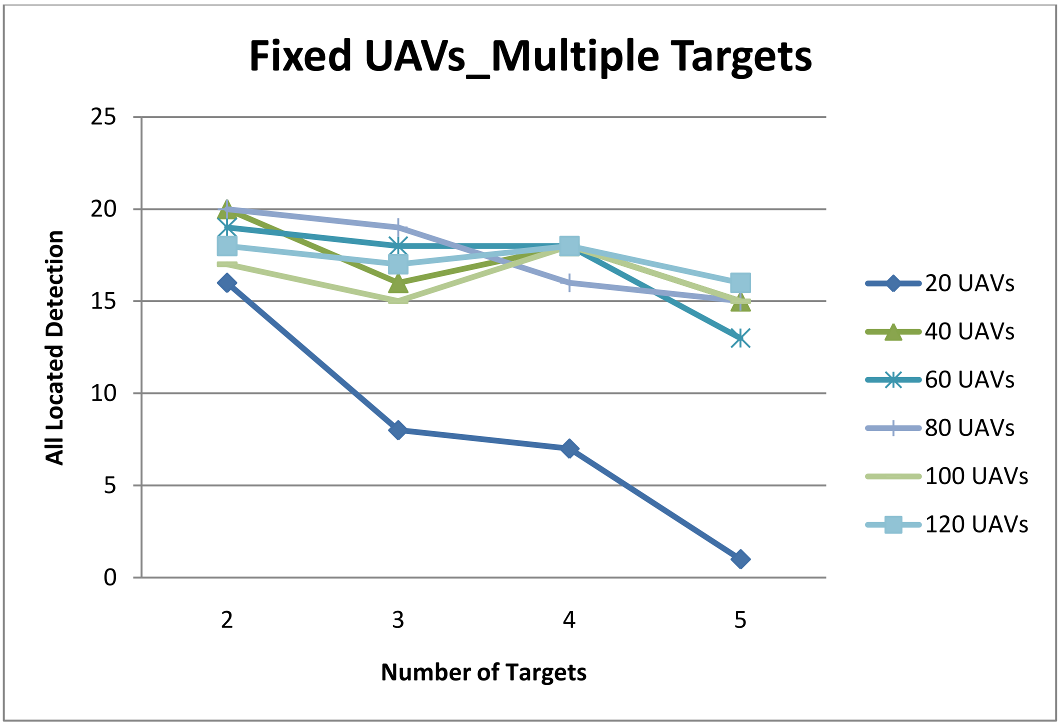
The illustration indicates that when the number of UAVs is limited, especially for the groups of 20 and 30 UAVs, the number of targets has an effect on the performance of achieving all-located target detections. The more targets to be located, the less number of all-located target detections can be achieved for every 20 experimental runs of the search model. The effect can be reduced by increasing the number of UAVs. To provide a better perception, Figure 6.7b presents the number of all-located target detections for selected groups of UAVs.

Figure 6.7 The All-Located Target Detections with the Three-UAV Formation

a. **Experimental Setup One**



b. **Experimental Setup Two**



The data indicate that, the three-UAV formation enables UAVs to locate some of the targets but the performance varies amongst different combinations of UAVs and targets. The maximum time length is often exceeded before UAVs are able to locate all of the targets. Taking the group of 20 UAVs in the 2-target detection as an example, UAVs are able to achieve 17 all-located target detections from 20 experimental runs. The number of all-located target detections reduced when there are more targets to be located by the group of 20 UAVs. This could be improved by deploying more UAVs to carry out the 2-target detection. The less effective performance of three-UAV formation could be caused by 1) the randomised search patterns of individual UAVs, which may generate clusters of UAVs in one part of the search area and miss out other parts; 2) limited UAV quantities, as for every initial detection of target signal, the three-UAV formation needs to engage three UAVs to verify and locate the actual target.

6.1.2.3 Comparison of the Two Scenarios in the Multiple Target Detection

The performances of the two cooperative scenarios of UAVs are compared regarding the average detection time and the number of all-located target detections.

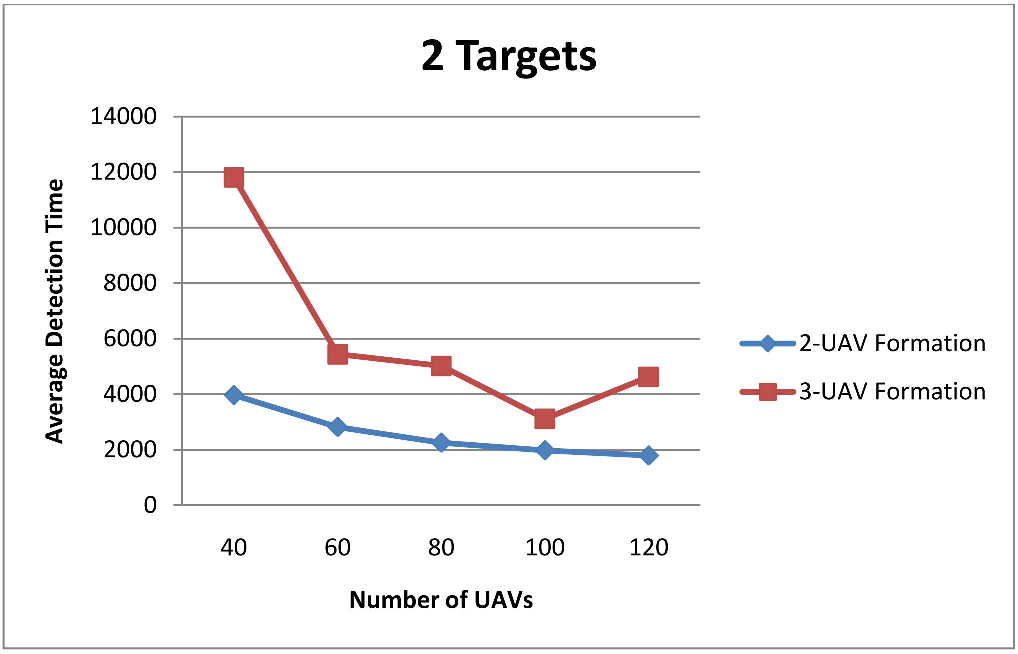
Criteria One: Average Detection Time

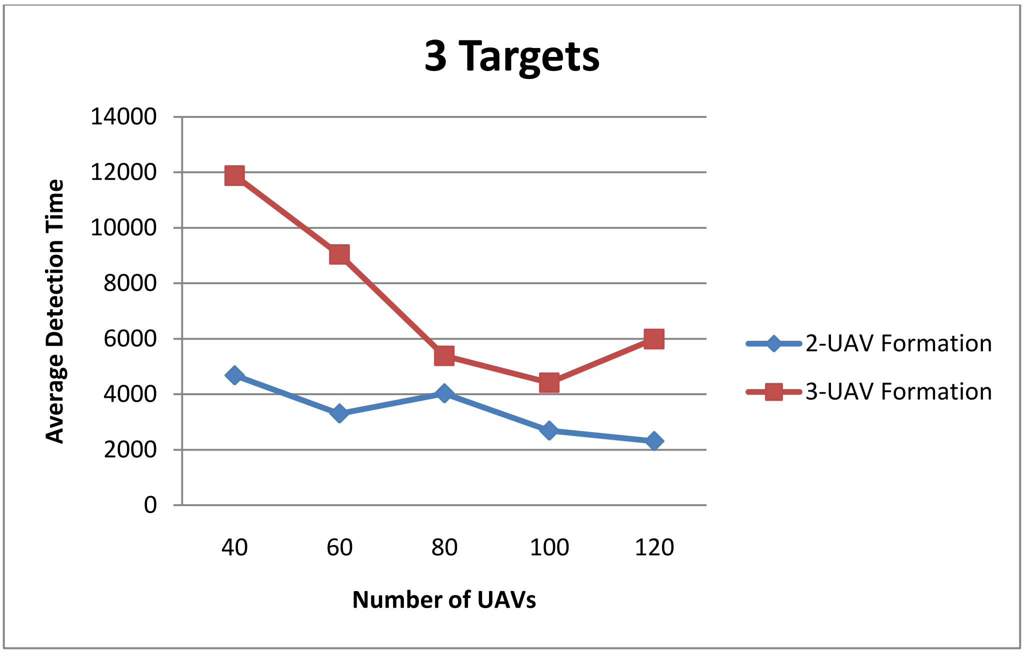
The average detection time consumed by UAVs in each of the two scenarios is first compared. The experimental data indicate that, the number of UAVs has significant impact on the overall search performance. When there are limited numbers of UAVs, the two scenarios of cooperative strategies would become less effective as it takes time to engage required numbers of UAVs to generate proper formations. The group of 20 UAVs and 120 UAVs are the two typical examples to explain the relationships between the UAV quantities and the average detection time. The group of 20 UAVs consumed the longest detection time on average with both of the scenarios of UAV cooperation. For 120 UAVs, the two-UAV formation consumed less detection time on average than the three-UAV formation.

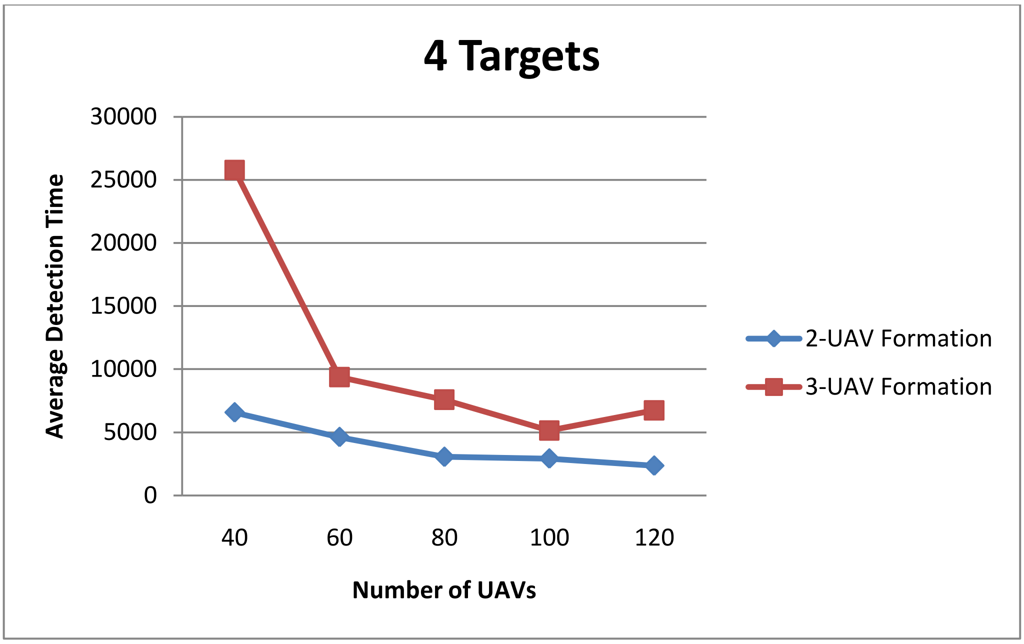
Figure 6.8 compares the average detection time of the two scenarios for the 2-target detection, the 3-target detection, the 4-target detection and the 5-target detection. The groups of 40 UAVs, 60 UAVs, 80 UAVs, 100 UAVs and 120 UAVs are selected for the comparison of the two experimental setups. The experimental data suggest that, the two-UAV formation consumed significantly lower detection time on average than the three-UAV formation for all the combinations of UAVs and targets. Therefore, regarding the average detection time, the diagonal formation of two UAVs is more efficient than the triangular formation of three UAVs in the multiple target detection.

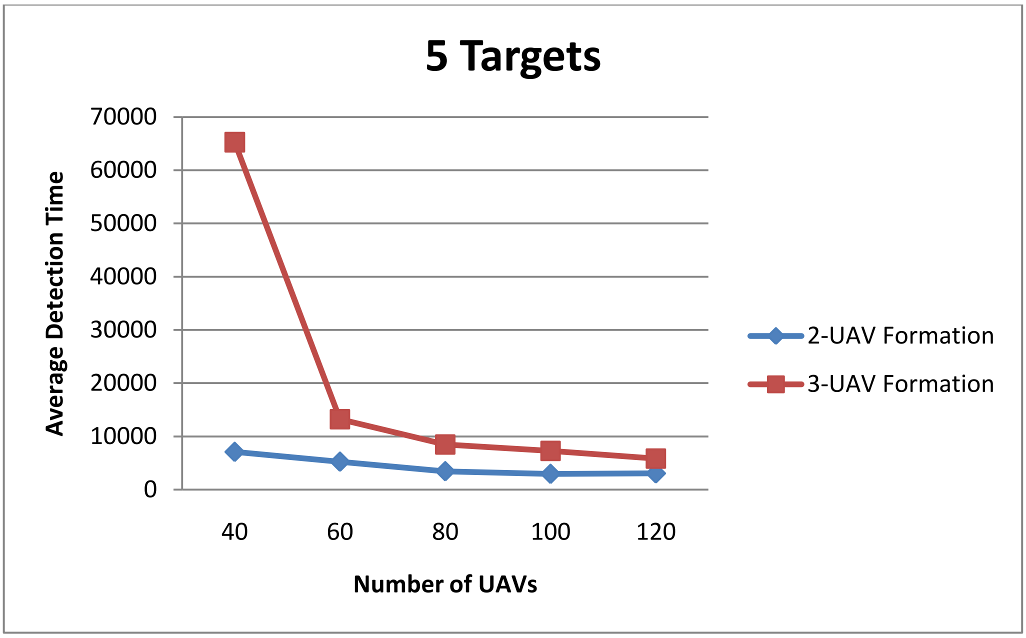
Figure 6.8 The Average Detection Time of the Two Scenarios

a. Experimental Setup One: Fixed Targets vs. Multiple UAVs

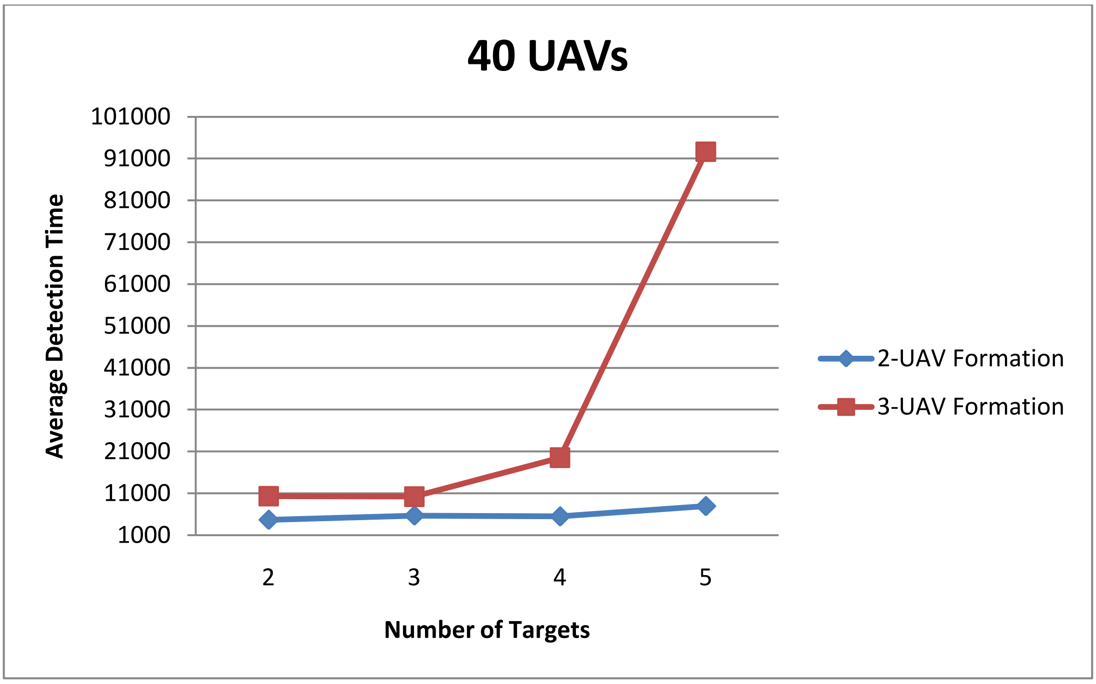

2 Targets
Average Detection Time
14000
12000
10000
8000
6000
4000
2000
0
40
60
80
100
120
Number of UAVs
2-UAV Formation
3-UAV Formation

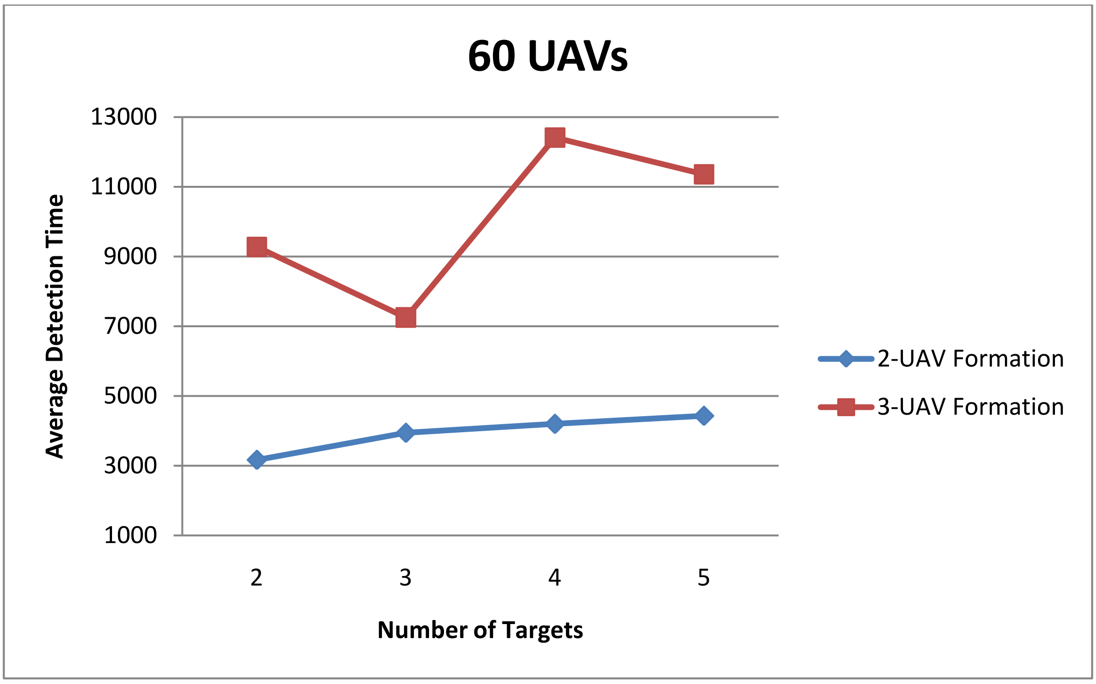

3 Targets
Average Detection Time
14000
12000
10000
8000
6000
4000
2000
0
40
60
80
100
120
Number of UAVs
2-UAV Formation
3-UAV Formation

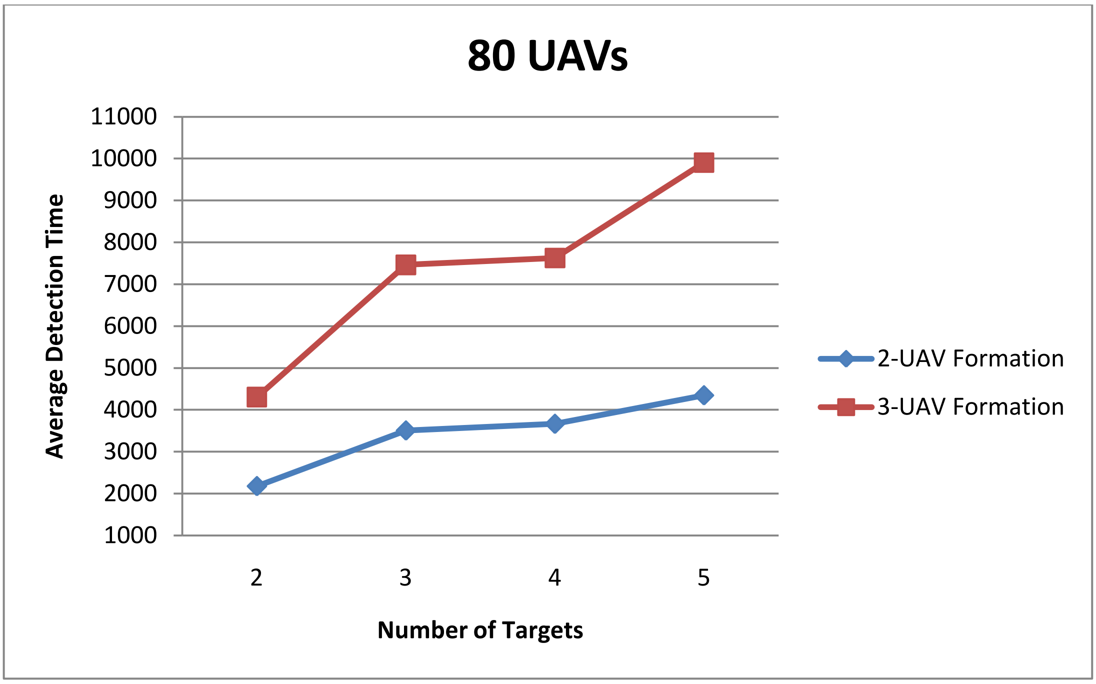

4 Targets
Average Detection Time
30000
25000
20000
15000
10000
5000
0
40
60
80
100
120
Number of UAVs
2-UAV Formation
3-UAV Formation

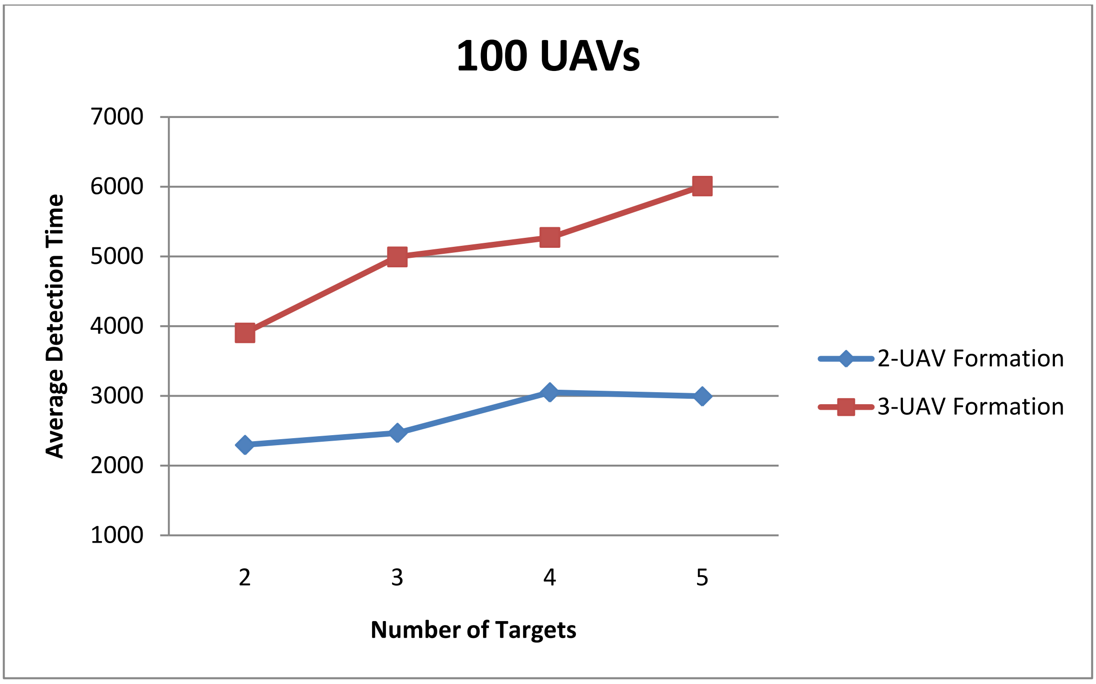

5 Targets
Average Detection Time
70000
60000
50000
40000
30000
20000
10000
0
40
60
80
100
120
Number of UAVs
2-UAV Formation
3-UAV Formation

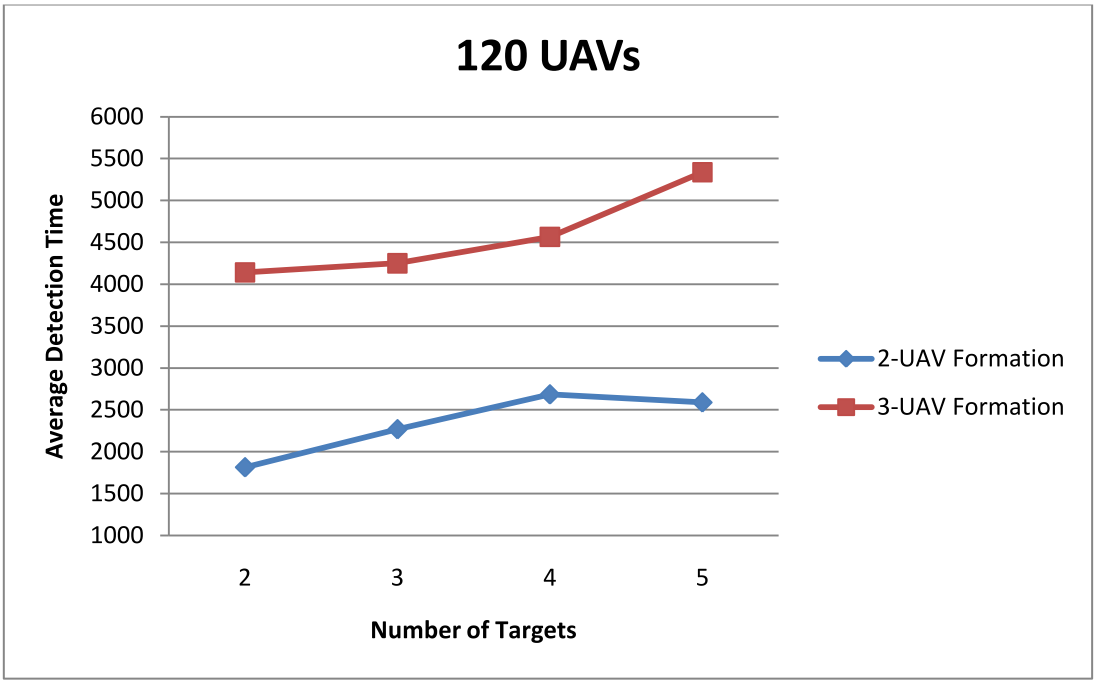
b. Experimental Setup Two: Fixed UAVs vs. Multiple Targets






80 UAVs
Average Detection Time
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
2
3
4
5
Number of Targets
2-UAV Formation
3-UAV Formation


100 UAVs
Average Detection Time
7000
6000
5000
4000
3000
2000
1000
2
3
4
5
Number of Targets
2-UAV Formation
3-UAV Formation



Criteria Two: All-Located Target Detections

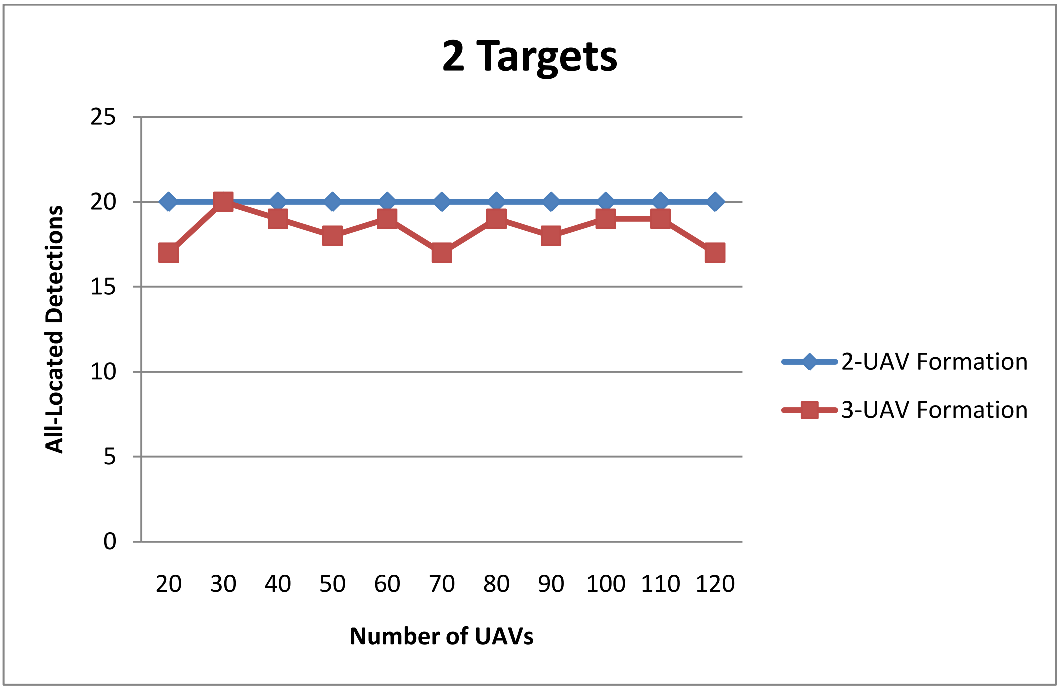
In addition to the average detection time, the two scenarios of UAV cooperation are also compared in terms of the number of all-located target detections for each combination of UAVs and targets. Figure 6.9a compares the number of all-located target detections of the two-UAV formation and the three-UAV formation in the multiple target detection. Figure 6.9 b compares the number of all-located target detections of the two scenarios for select groups of UAVs.

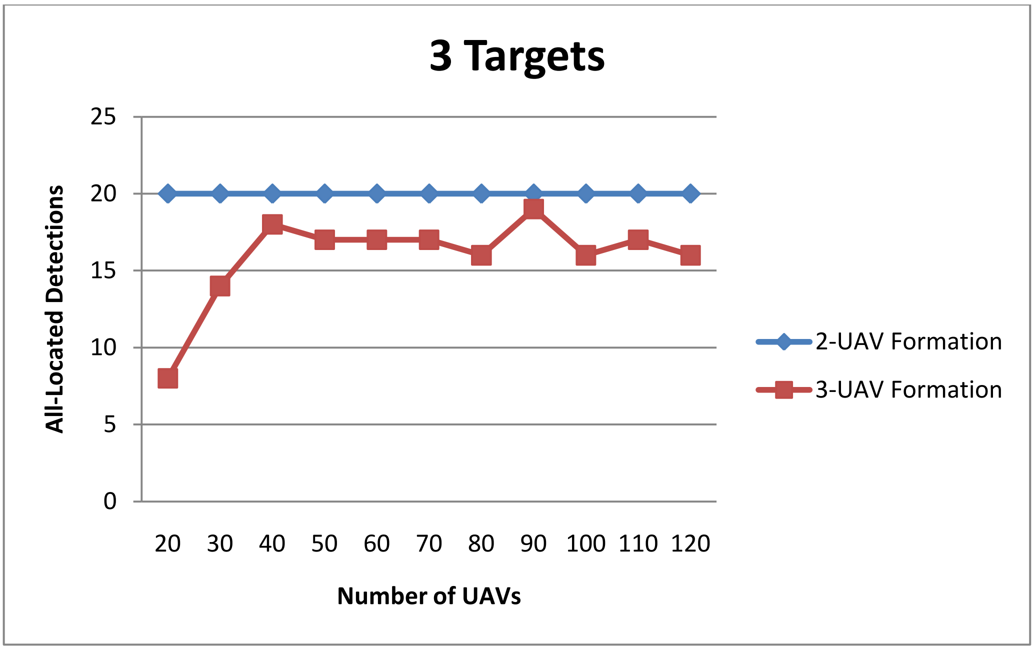
For the 20 UAVs / 5 targets combination, one of the experimental runs of the two-UAV formation was unable to locate all the targets within the maximum time length. This case occurred with the first experimental setup: fixed targets vs. multiple UAVs (see Figure 6.9 a). Apart from this one case, the two-UAV formation located all of the targets within the maximum time length in each case of the multiple target detection. With the three-UAV formation, however, UAVs are only able to locate some of the targets within the maximum time length. The number of all-located target detections varies depending on the number of targets and the number of UAVs. In the 2-target detection, for example, the three-UAV formation enables UAVs to achieve 17 successful experimental runs out of the total 20 experimental runs (Table A4 in Appendix A-2 and Figure 6.9). As the number of targets increases, the number of the all-located target detections decreases especially when there are only 20 UAVs.

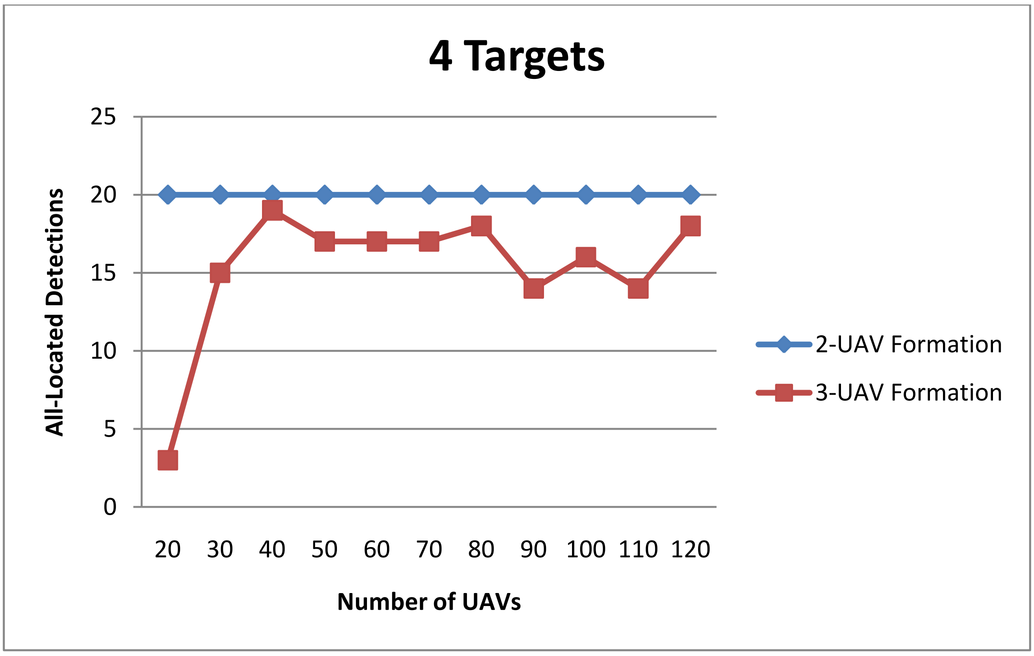
Therefore, the performance of three-UAV formation is significantly affected by limited UAV quantities. This is because the three-UAV formation requires formation of three UAVs to locate each of the targets, which is time-consuming and when there are more targets to be located, it would become increasingly difficult for UAVs to accomplish the task effectively and efficiently.

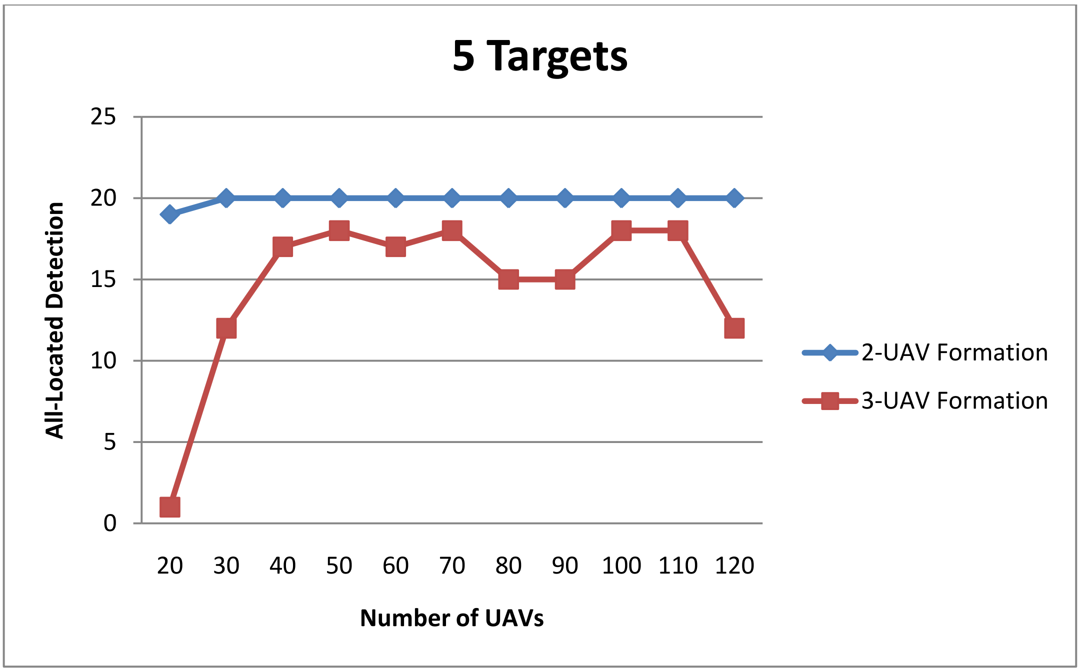
Figure 6.9 The All-Located Target Detections of the Two Scenarios

a. Experimental Setup One: Fixed Targets vs. Multiple UAVs

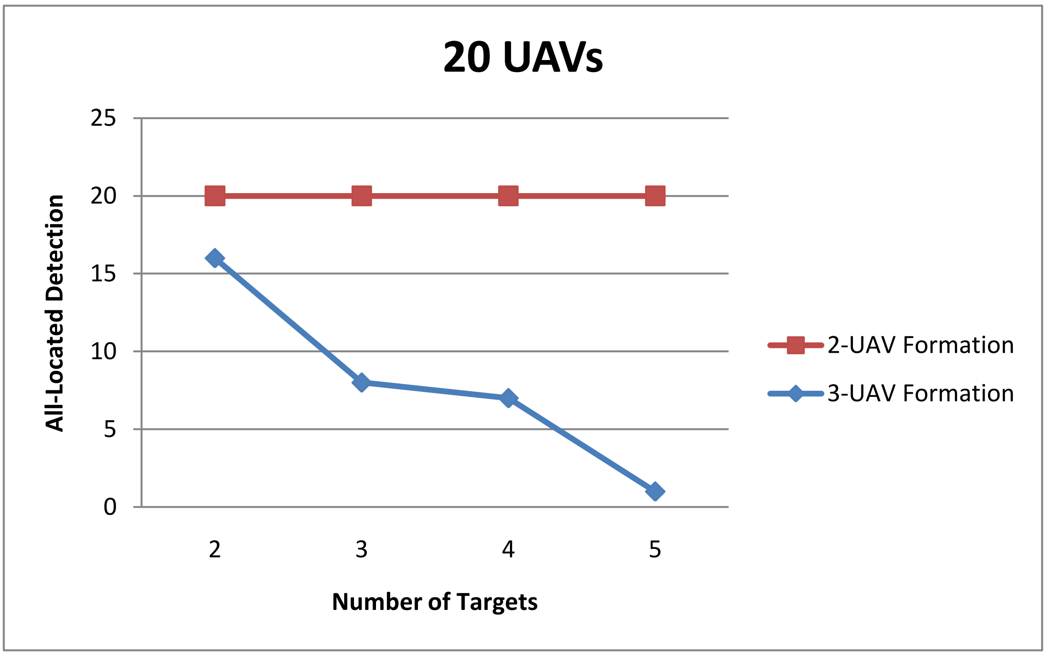


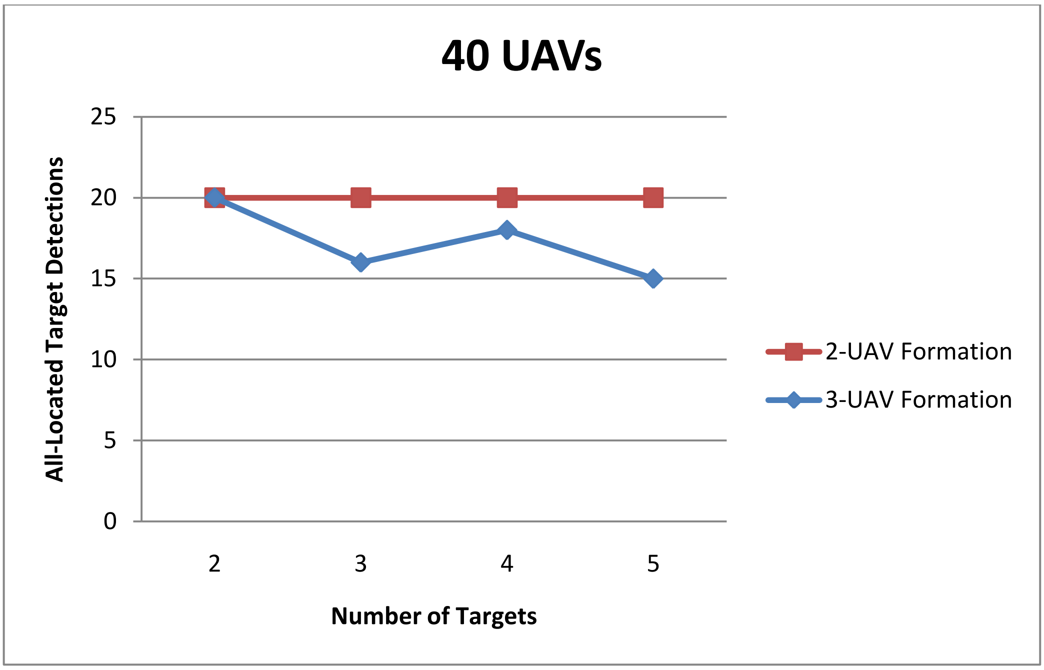

3 Targets
25
20
15
10
5
0
All-Located Detections
20 30 40 50 60 70 80 90 100 110 120
Number of UAVs
2-UAV Formation
3-UAV Formation

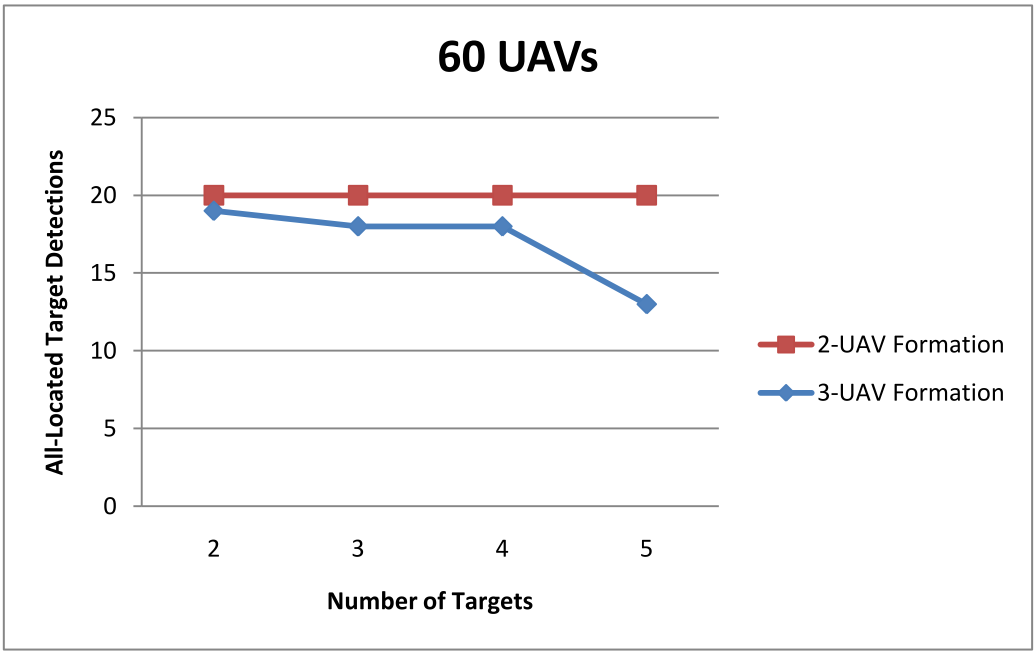

4 Targets
25
20
15
10
5
0
All-Located Detections
20 30 40 50 60 70 80 90 100 110 120
Number of UAVs
2-UAV Formation
3-UAV Formation

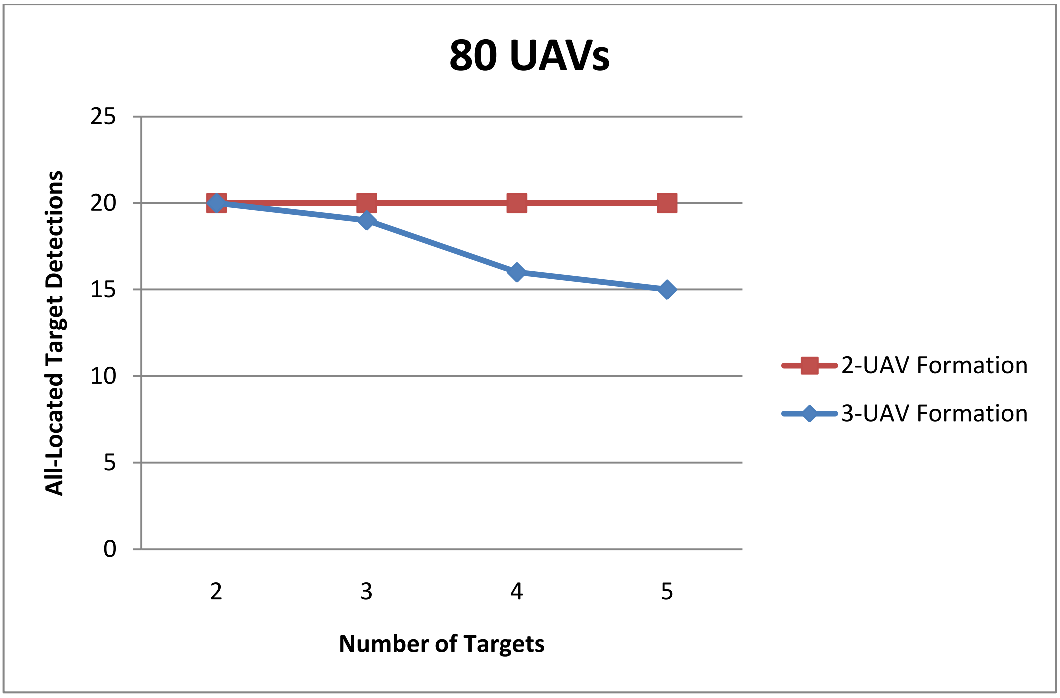


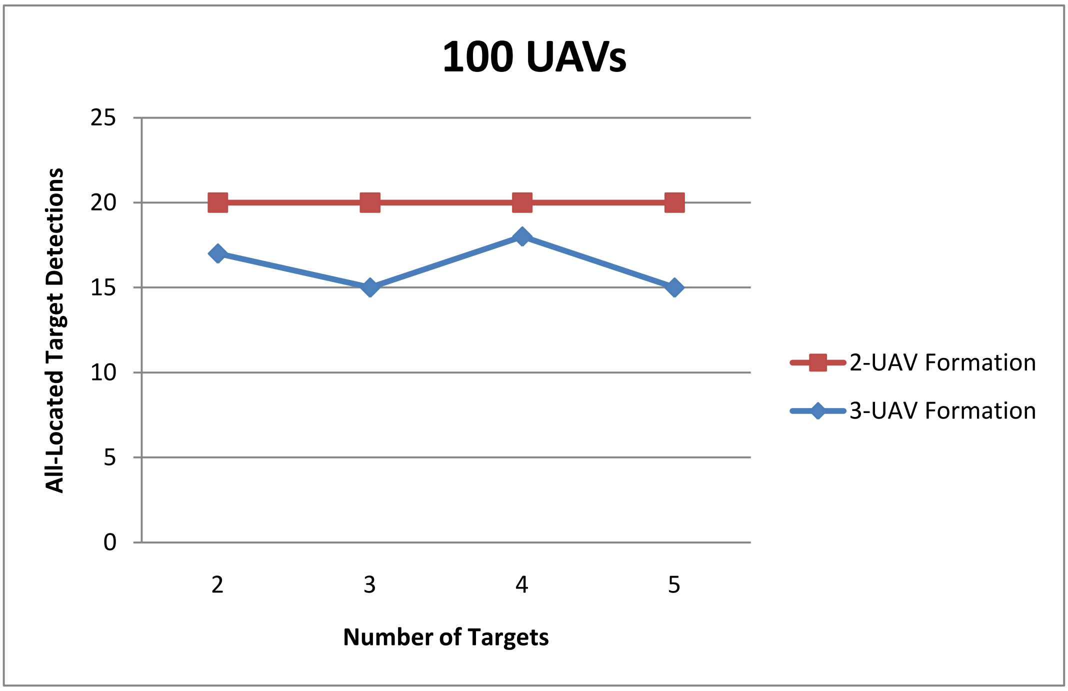
b. Experimental Setup Two: Fixed UAVs vs. Multiple Targets

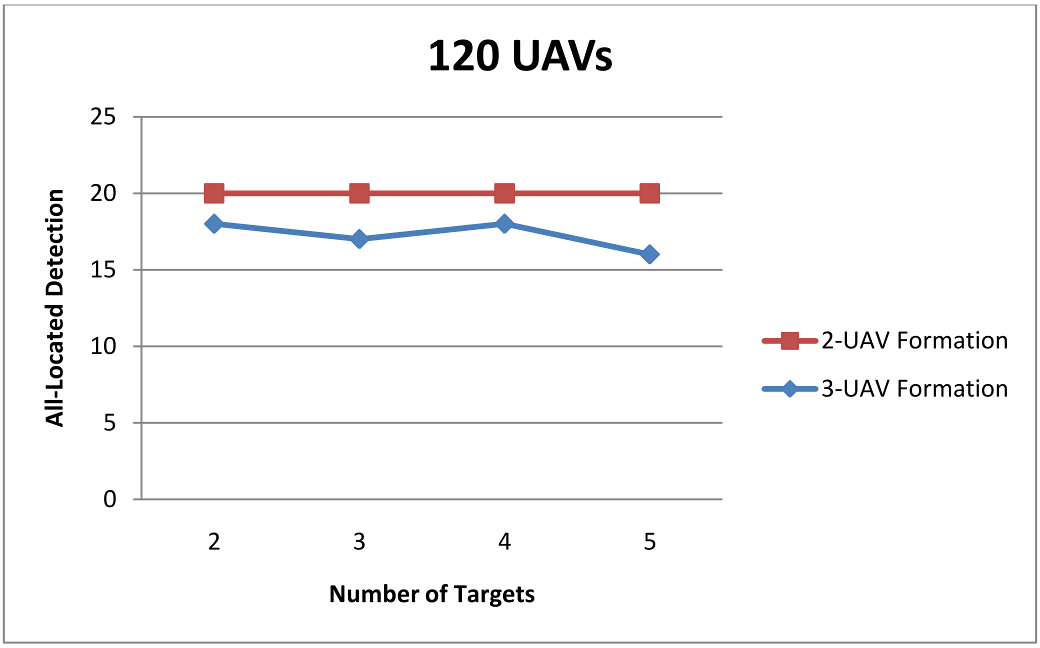



40 UAVs
All-Located Target Detections
25
20
15
10
5
0
2
3
4
5
Number of Targets
2-UAV Formation
3-UAV Formation


60 UAVs
All-Located Target Detections
25
20
15
10
5
0
2
3
4
5
Number of Targets
2-UAV Formation
3-UAV Formation


80 UAVs
All-Located Target Detections
25
20
15
10
5
0
2
3
4
5
Number of Targets
2-UAV Formation
3-UAV Formation


100 UAVs
All-Located Target Detections
25
20
15
10
5
0
2
3
4
5
Number of Targets
2-UAV Formation
3-UAV Formation



6.1.3 Statistical Analysis of the Results

The analysis so far has shown that the 2-UAV formation consistently outperforms the 3-UAV formation on average detection time. In this section, the relative performance on detection time is studied in greater detail. However, note that for any given combination of number of targets and number of UAVs, the two experimental set-ups each provide 20 replicates (the differences in results in the two cases for the same combination is due to different random number streams in use). Hence, the results from the two set-ups are combined to base the statistical analysis on 40 replicates.

Tables A5 and A6 (in Appendix A-2) present the maximum and minimum values observed around the average detection time for the two formations. They show a large spread of values for both, but the spread tends to reduce considerably as the number of UAVs increase; hence, the reliability of detection within a given time improves as the number of UAVs increase. To further show the different performances of the two-UAV formation and three-UAV formation, t-test was carried out in Excel on the average detection time. Independent t-test is selected that does not assume equal variances for the two scenarios (type 3 in Excel, Aspin-Welch test). Table 6.5 presents the p-values for the t-test comparing the average detection times for the two and three UAV formation in the case of each combination of number of targets and number of UAVs. The results clearly show that the two-UAV formation very significantly outperforms the triangular 3-UAV formation in terms of average detection time (p-value < 0.1 in each case, considerably lower in most cases).

Table 6.5 T-test: P-Values for the Comparison of Two- and Three-UAV Formations

	1 Target	**2 Targets**	**3 Targets**	**4 Targets**	**5 Targets**
UAVs					
20	0.000804883	4.8274E-05	7.92873E-14	1.38035E-16	4.56435E-26
30	0.000350403	0.000134969	0.009359101	2.26305E-05	5.86794E-06
40	0.004017002	1.86072E-05	6.63043E-07	4.351E-05	0.001418003
50	0.000199959	0.000117927	5.82137E-07	2.61426E-07	1.45314E-07
60	0.001516112	8.69473E-06	4.67745E-06	2.8708E-08	6.81616E-09
70	0.000133916	0.000116199	2.85509E-05	1.3162E-06	7.29633E-06
80	0.00430245	2.98537E-05	0.000341308	3.06339E-08	2.83261E-07
90	0.000243071	0.000268164	1.3456E-05	0.000367585	1.16386E-08
100	0.000272115	2.36086E-05	2.07363E-05	0.000164166	2.33279E-08
110	3.19146E-05	1.6849E-07	5.9479E-05	1.13153E-05	9.86981E-08
120	0.005066751	2.38008E-07	9.95562E-06	4.41508E-07	8.82722E-08

6.2 Discussion

6.2.1 Swarm-inspired Cooperative Search

The cooperative search operation of multiple UAVs is implemented in two scenarios of target detection – single target detection and multiple target detection. Each scenario is evaluated with two performance criteria: the average detection time and the number of targets located within the maximum time length (all-located target detection). The number of UAVs and the number of targets are the two variables used to implement the experiments. The experiments are setup with different combinations of UAVs and targets, which alters one of the variables while the other remains unchanged. The experimental data suggest that given the same numbers of UAVs and the same numbers of targets, the diagonal formation of two UAVs consumed much less detection time in average and was able to successfully locate all the targets within the maximum time length in all but one experimental run for the 20 UAVs / 5 targets combination. The triangular formation of three UAVs consumed longer detection time on average and was unable to locate all the targets within the maximum time length. The results of t-test also indicate that in all combinations of targets/UAVs, the two-UAV formation significantly outperforms the three-UAV formation.

For three-UAV formation, it takes time to engage three UAVs to converge onto the initial detection of target signal. The performance is significantly constrained by the number of UAVs available at the time of generating formation. Once the target signal is initially detected, the detecting UAV needs to recruit two other UAVs. In particular, when there is only a limited number of UAVs, it would take much longer time for the detecting UAV to find qualified UAVs to recruit. Additionally, when there are more targets to be detected, a limited number of UAV would be unable to cover them all on time. This resulted in prolonged detection time on average, as well as reduced number of located targets within the maximum time length.

6.2.2 Comparison with Current Approaches of Cooperative Search

The swarm-inspired search strategy proposes to accomplish the cooperative search of UAVs through the following mechanisms: random search pattern of individual UAVs, signal-stimulated communications, recruitment and convergence, and the generation and maintenance of flight formations of UAVs. Table 6.6 below outlines the context and features each for the swarm-inspired search strategy and existing approaches of the cooperative search of UAVs.

Table 6.6 The Cooperative Control Architecture and Swarm-inspired Search Strategy

The Cooperative Control Architecture	Swarm-inspired Search Strategy
4-State Control Architecture (Pack and York 2005; York, Pack et al. 2007): Global Search, Approach located Target, Orbit and Locate Target, Local Search for lost mobile Target.	Behaviour-based Search Mechanisms: Random Search Pattern of Individual UAVs, Signal-stimulated Communications, Recruitment, Emergence of Flight Formations, Synchronised Circling.
Detecting and Locating targets with sensing and image processing techniques, e.g.: 1) The Line Formation (Vincent and Rubin 2004; Altshuler, Yanovsky et al. 2008); 2) The Triangulation of Multiple UAVs (Toussaint, De Lima et al. 2007; Pack, DeLima et al. 2009)	Flight Formations of UAVs: 1) Diagonal Formation of Two UAVs; 2) Triangular Formation of Three UAVs.

The Line Formation (Vincent and Rubin 2004; Altshuler, Yanovsky et al. 2008): Pre-defined with fixed number of UAVs in the formation pattern; Require prior knowledge of the search area; Search and locate mobile targets with parallel sweep pattern;	Diagonal/Triangular Formation of UAVs: Any of the two/three UAVs are able to converge onto cooperative formations; The initial detection of target signal stimulates the communications and recruitment between UAVs; The formation is generated on an ad hoc basis and no advanced information of the search area is required; UAVs retain the formation and circle synchronously to track and locate the mobile target.
The Triangulation of Multiple UAVs (Toussaint, De Lima et al. 2007; Pack, DeLima et al. 2009): With two UAVs, the target location is estimated by the intersection of the two angle bearing lines of UAVs; With three UAVs, the target location is estimated in the centre of the triangulation.	Diagonal/Triangular Formation of UAVs: The target location can be identified at unique coordinates within the overlapping sensor coverage of UAVs; Synchronised circling behaviour enables UAVs to track the target movement and locate the target in a wider area.

The line formation of UAVs is predefined and consists of a fixed number of UAVs (Vincent and Rubin 2004). It requires UAVs to be aware of the search area in advance and hence demands basic but accurate information to be provided before initiating the search operation. This involves additional ground work beforehand and is not robust enough to be adaptable with different conditions of search area. In the case of UAV failure, retaining the original formation pattern with no new UAVs joining in could also reduce the detection coverage of the UAV formation. Reduced detection coverage is likely to cause extra time consumption and less efficiency of target detection. Additionally, although the parallel sweep pattern (both the original sweep pattern (Vincent and Rubin 2004) and the angled sweep pattern (Altshuler, Yanovsky et al. 2008)) is able to fully cover the entire search area from one end of the field to the other, it could consume extended period of time unnecessarily if targets are clustering on one side of the search field; and the effectiveness and efficiency may also be constrained to the size and shape of the search field.

The triangulation of UAVs engages a team of two and/or three UAVs to locate the target with their angle-to-target estimations (Pack and York 2005; Toussaint, De Lima et al. 2007; Pack, DeLima et al. 2009). However, the estimated target locations have various deviations from the actual target locations. Also, the leader-follower approach coordinates UAVs to join the formation but in the mean time, it requires additional calculations of the followers' trajectories and reliable angle-to-target estimations can only be produced when all cooperating UAVs are in a stabilised orbit.

The swarm-inspired search strategy proposes behaviour-based mechanisms that assign UAVs with identical rules of behaviour. Stimulated by the initial detection of target signal, the cooperative flight formations are generated between any of the two and/or three UAVs. No advanced information of the search area is needed as the UAV formations are generated on an ad hoc basis, and are only taking place when there is a target signal detected. In this way, other UAVs are able to explore different parts of the search area, and thus to enhance detection effectiveness of UAVs. The diagonal formation of two UAVs and the triangular formation of three UAVs generate overlapping sensor coverage in order to track and locate the mobile target. Instead of estimating the target location, the goal is to locate the target at unique coordinates in the overlapping sensor coverage of UAVs. If UAVs are unable to identify the target location via the formation initially, they then circle around together while retaining the formation pattern intending to track the target in a wider area. The cooperative formations of UAVs are emerged from randomised individual detection of target signal via signal-stimulated communications. Such kind of mechanisms increases robustness and efficiency of target detection, and enables UAVs to effectively respond to various conditions and circumstances of the search operation.

The experimental results indicate that the swarm-inspired search strategy has delivered promising performance of searching for multiple mobile targets. The behaviour-based search mechanisms enable UAVs to explore the search area in a dynamic pattern. The two formation patterns of UAVs are able to achieve the unique coordinates of targets and accomplish the search operation effectively in terms of both detection accuracy and time consumption. The diagonal formation of two UAVs has outperformed the triangular formation of three UAVs with very significantly lower time of detection on average and the number of located targets within the maximum time length. Thus, it is shown that, with appropriate formations, an efficient search can be carried out by two UAVs.

Chapter 7. Conclusion and Future Work

7.1 Conclusion

In-depth research was conducted to understand the emergent behaviour of social insects and related problem-solving mechanisms that are applicable to collective problem-solving in multi-agent systems. The observation and modelling of insect behaviour, such as the ant foraging behaviour and the nest construction of termites, indicate that simple individuals are able to accomplish complex problems through cooperation and coordination. The problem-solving of such kind presents distinctive features in terms of robustness, adaptability, and efficiency. As a result of the research, an interdisciplinary knowledge of collective problem-solving in social insects and multi-agent systems was established. Such knowledge emphasises the influences of environment, and especially how the environment roles and responsibilities contribute to the collective problem-solving of both insect behaviour and multi-agent systems. A framework has been developed to describe the collective problem-solving in the above interdisciplinary context.

On the basis of the knowledge of collective problem-solving, a swarm-inspired strategy was proposed for the UAV search problem. It is composed of randomised search pattern of individual UAVs, signal-stimulated communications, self-organisation, recruitment and convergence, cooperative flight formations and synchronised circling behaviour. The proposed search solution aimed to investigate how the environment roles and responsibilities facilitate the cooperative search of UAVs. The cooperative search problem of UAVs is a classic subject of collective problem-solving, which has attracted ongoing research and development over the years (Bellingham, Tillerson et al. 2002; Yanli, Minai et al. 2004; Beard, McLain et al. 2006; Dasgupta 2008; Bryson and Sukkarieh 2009).

Amongst a variety of research of the cooperative search of UAVs, the 4-state cooperative control architecture presented one of the major achievements of resolving the problem (Pack and York 2005; Toussaint, De Lima et al. 2007; York, Pack et al. 2007). UAVs carry out the following four states of search in order to locate mobile targets: Global Search (GS), Approach Target (AT), Locate Target (LT), and Local Search (LS). At each state of the search operation, UAVs use the predefined search and cost functions to make decisions on, for example, flight path, switching between search states, whether or not to initiate local search and engaging in cooperation, and so forth. The swarm-inspired search strategy presented here proposes to randomise the target detection of individual UAVs; no flight path is predefined to individual UAVs and UAV cooperation occurs upon initial detection of target signal via signal-stimulated communications among involved UAVs.

In the 4-state cooperative control architecture, each UAV keeps its own track of the exploration record and the ones of all neighbouring UAVs. The data is used to calculate the search cost at the global search state in order to identify optimal flight path at minimum costs. Calculations are also required when making decisions on switching from the global search state to the approach target state. In the local search state, UAVs need to calculate and adjust their orbiting positions to generate accurate detection data of target locations. The results show the triangulating orbit generates various deviations from the actual target locations. The proposed solution presented here simplifies the global search by randomising the search pattern of individual UAVs. No data record needs to be retained unless there is a detection of target signal. No calculations are required for generating UAV cooperation, as the communications and cooperative activities of UAVs are triggered by the detection of target signal and thus are carried out on a stimulating basis. This enables UAVs to be flexible, adaptable and robust. The experimental results indicate that the swarm-inspired search strategy provides a simplified solution to the cooperative search of UAVs with improved effectiveness and efficiency.

Table 7.1 categorises the proposed search strategy according to the categories of cooperative search approaches presented in the literature (Vincent and Rubin 2004).

Table 7.1 Categories of the Swarm-Inspired Search Strategy

Categories presented in the literature[10]	The Swarm-Inspired Search Strategy
Non-cooperative search	No predefined flight path, UAVs operate in a randomised search pattern to obtain the initial detection of target signal
Cooperative search	The initial detection of target signal stimulates local communications among UAVs; Via communications the detecting UAV recruits closest UAVs to converge onto formations; The patterns of UAV formations are predefined, which aim to locate moving targets on unique coordinates; If UAVs are unable to locate the target immediately, they are to carry out circling behaviour in order to track the moving target and to identify its coordinates.

In addition, the proposed search strategy presents merged characteristic properties of the bottom-up and top-down approaches of complex problem-solving. Referring to the literature (Bogatyreva and Shillerov 2006), Table 7.2 outlines the methodological features of the swarm-inspired search strategy.

[10] Please refer to Table 4.1 in Chapter 4, pp 77

Table 7.2 Methodological Features of the Swarm-Inspired Search Strategy

Merged Features of Intelligent Problem-Solving[11]	Swarm-Inspired Search Strategy
Hierarchical structure applied to support the agent interactions	The detecting UAV initiates signal-stimulated communications to recruit local UAVs. It is the coordinator of UAV cooperation, and thus holds the top position in the hierarchy.
Emphasise the Independence of Agents	Individual UAVs are provided with sufficient independence, so that they are able to operate as individuals effectively, as well as to conveniently switch between the cooperative state and the independent state.
The rules of activities are designed with respect to emergent phenomenon	The rules of UAV behaviour are designed so that the cooperative and coordinated activities are emerged from individual target detection via signal-stimulated communications, recruitment, etc.
Being able to predict the changes in a system	Upon an initial detection of target signal, the detecting UAV is expected to recruit local UAVs and cooperative formations are expected to be emerged from engaged UAVs. Such predictability to changes is implemented through ad hoc stimulating processes.
Situational awareness at global level while acting locally	Local responses of UAVs are the result of their situational awareness at global level, i.e. immediate and appropriate actions are taken regarding the real-time changes of the search circumstances.
Emergent Phenomenon via Limited Communications among individual agents of local neighbourhood	Multiple interactions are essential to produce emergent coordination and cooperation of UAVs. Such kind of interactions is taking place locally to minimise the generation and maintenance complexity. Using the search environment as a medium of communication, local UAVs interact with each other indirectly by modifying the search

[11] Refer to Table 2.2 in Chapter 2 (pp36) and the merged methodological framework on pp37

	environment and responding to the modified search environment. Direct interaction also occurs amongst UAVs in situations such as the detecting UAV recruits nearby UAVs to converge onto cooperative formations.

In the 4-state cooperative control architecture, the cooperative formations of UAVs performed a key contribution in locating target positions (Altshuler, Yanovsky et al. 2008; Pack, DeLima et al. 2009). Two typical UAV formations are presented in the literature: the line formation (Vincent and Rubin 2004; Altshuler, Yanovsky et al. 2008) and the triangulation of multiple UAVs (Pack and York 2005; Toussaint, De Lima et al. 2007). The line formation is predefined and consists of a fixed number of UAVs. It is able to locate mobile targets with one of the simplest formation patterns of UAVs via sweeping through the search region. However, it requires basic but accurate information of the search field to be available to UAVs in advance. Also the individual UAV failure may result in reduced sensor coverage as no new UAVs are configured to join in and the formation pattern remains unchanged.

The triangulation UAVs produces an estimation of target locations. For two UAVs, they estimate the target location using the intersection of their angle bearing lines; for three UAVs, they estimate the target location using the centre point of the triangulation. Results show that the estimation by three UAVs produced more deviations from the actual target location than the two UAVs, and thus was a less effective approach (Toussaint, De Lima et al. 2007). Further study also indicated that the triangulation of two UAVs encountered difficulties on producing a proper intersection of their angle bearing lines (Pack, DeLima et al. 2009).

Alternative cooperative formations of UAVs for locations of mobile targets was explored and evaluated in the swarm-inspired search strategy. Two scenarios of cooperative formations of

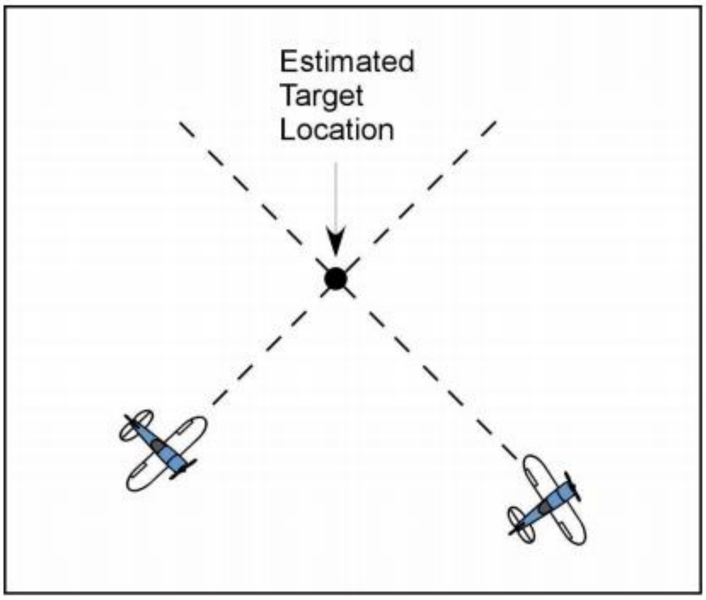
UAVs have been proposed, which are the diagonal formation of two UAVs and the triangular formation of three UAVs. Different from the triangulation of multiple UAVs presented in the literature, the two scenarios of UAV cooperation intend to locate mobile targets on unique coordinates based on predefined formations. For each of the two-UAV formation and the three-UAV formation, Figure 7.1 presents graphical illustrations each for the literature approaches and for approaches deployed in the proposed solution.

A simulation model of the cooperative search of UAVs has been developed and a series of experiments have been implemented to evaluate the performance of UAVs. The two scenarios of UAV cooperation have been evaluated and compared for their performances in accomplishing the cooperative search of UAVs. The experimental data suggest that within the architecture of proposed search solution, the diagonal formation of two UAVs is able to produce superior performance than the triangular formation of three UAVs. Experimental results show that the diagonal formation of two UAVs has outperformed the triangular formation of three UAVs with very significantly lower time of detection on average, and on the number of located targets within the maximum time length.

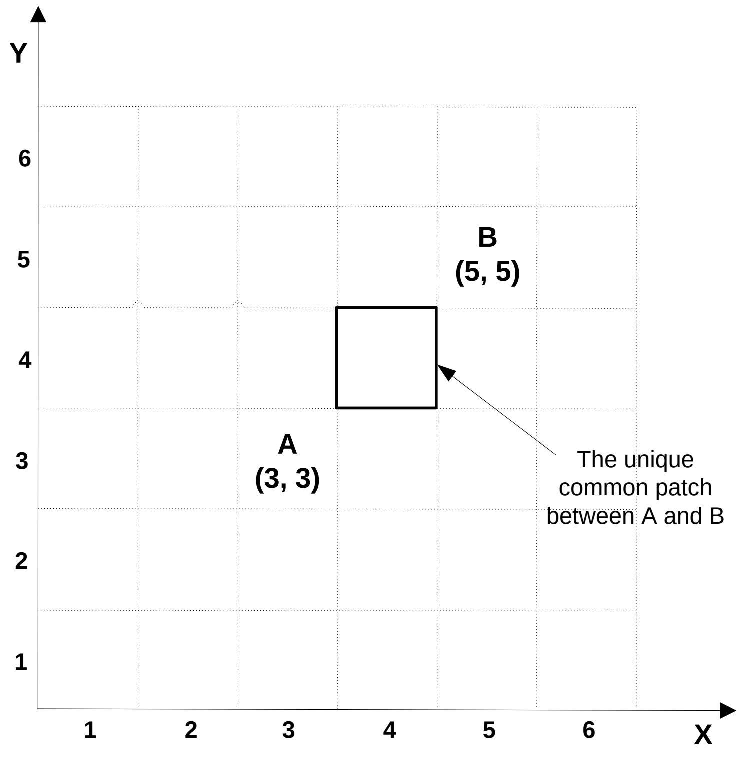
Figure 7.1 Current UAV Formations and Proposed UAV Formations

a. **Two-UAV Formation**

   1) **Literature: Intersection of angle bearing lines (Toussaint, De Lima et al. 2007)**

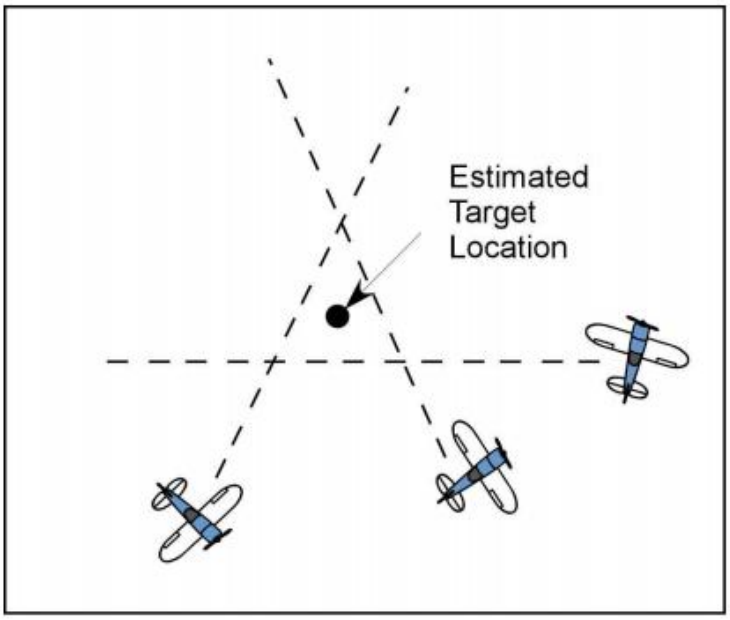


2) Proposed Solution: The Diagonal Formation of Two UAVs

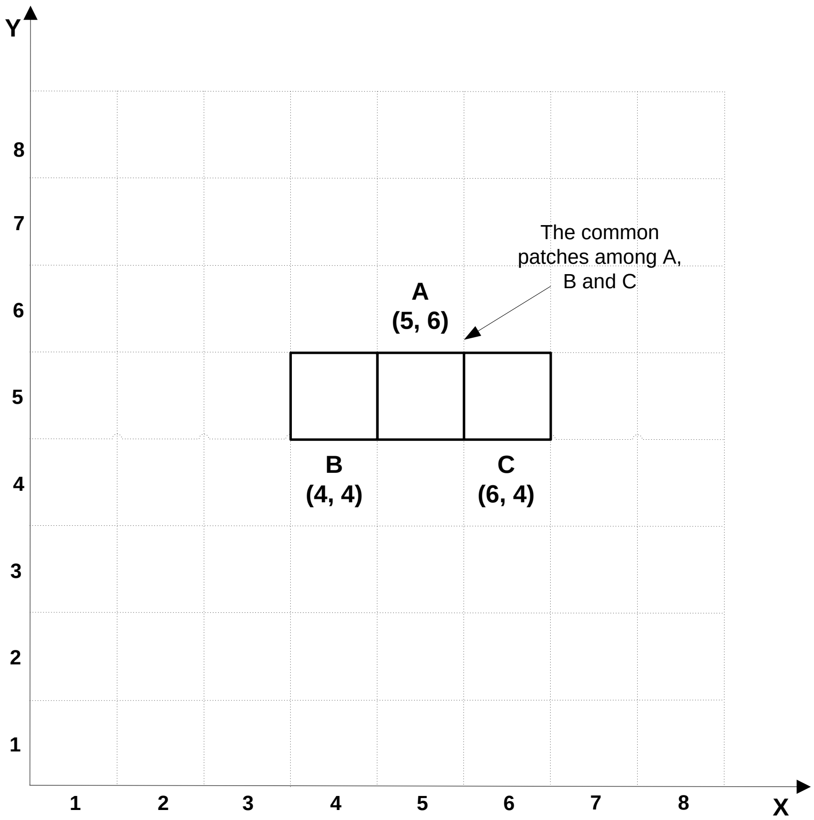


b. Three-UAV Formation

1) Literature: Multiple Intersections of Three UAVs (Toussaint, De Lima et al. 2007)



2) **Proposed Solution: The Triangular Formation of Three UAVs**



7.2 Research Contributions

This research has achieved the research objectives identified in Chapter 1, Section 1.3, which has following contributions to related subject areas:

1. In-depth study has been carried out to investigate the influences of environment that contribute to the problem-solving mechanisms of insect behaviour and multi-agent systems;

2. An interdisciplinary knowledge of collective problem-solving is established in the form of a framework, which describes the influential contributions of the environment in the emergent behaviour of social insects and the collective problem-solving of multi-agent systems;

3. Using the framework, a swarm-inspired search strategy with a set of problem-solving mechanisms have been proposed for the application to accomplish the cooperative search problem of UAVs;

4. Experimental results indicate that the proposed two-UAV formation is very efficient and effective in achieving the objective of the search problem.

7.3 Future Work

This research suggests the following future work:

1) Improve the line-of-sight of individual UAVs;

Further development on the sensing and image processing is expected to eliminate both hardware and software constraints, and hence to improve the line-of-sight of individual UAVs with minimum obstructions caused by the unknown environment.

2) Adjust the altitude of UAVs to obtain a more precise detection of targets;

In addition to the sensor range, the detection precision is also governed by the altitude of UAVs. A new procedure of adjusting the altitude of UAVs can be added to the local search aiming to achieve a more precise location of the target on the patch. The UAV altitude would become an important variable in a 3D search environment.

3) Handling UAV failure;

Additional mechanisms are required for dealing with UAV failure. For instance, if one of cooperating UAVs failed while in a formation, possible actions to be taken for other UAVs may include either resuming random flight pattern or recruiting a replacement.

4) Improve the two-UAV formation;

Existing research and the proposed research both suggest that the two-UAV formation is more effective and efficient than the three-UAV formation. Hence further investigation of potential improvements of the two-UAV formation should be considered. For instance, to assess whether this scenario is adaptable to the search problem with increased complexity, such as requiring UAVs to distinguish between false targets and the real target; recognition for different identities of different targets, e.g. detection for human body, ground vehicles, and ships, etc; and to develop the search model in a 3D simulation environment to better represent the search operation of UAVs.

5) Eliminate disadvantages of randomisation.

The random flight pattern of UAVs could lead UAVs to be clustered in certain parts of the search field and not being able to search for targets evenly. This would compromise the detection efficiency of UAVs. One solution is to divide the search field into smaller and equal-sized areas. UAVs are then distributed into each area of the search field before the search operation initiates. Each area may have identical number of UAVs, and these UAVs are only responsible for exploring the area in which they are allocated to. Thus, UAVs still operate randomly within the allocated areas of search field. They do not move beyond the boundary of allocated area. Other rules of search behaviour remain the same.

References

(1997). Oxford Concise Colour Science Dictionary, Oxford University Press.

A. C. Mailleux, J. L. Deneubourg and C. Detrain (2003). "How Does Colony Growth Influence Communication in Ants?" Insectes Sociaux 50: 24 - 31.

Ablavsky, V. and M. Snorrason (2000). Optimal search for a moving target: a geometric approach, Citeseer.

Acosta, F., F. Lopez, Serrano, JM (1995). "Dispersed versus central-place foraging: intra-and intercolonial competition in the strategy of trunk trail arrangement of a harvester ant." American Naturalist 145(3): 389-411.

Allred, J., A. Hasan, Saroch Panichsakul, William Pisano, Peter Gray, Jyh Huang, Richard Han, Dale Lawrence and Kamran Mohseni (2007). SensorFlock: an airborne wireless sensor network of micro-air vehicles, ACM.

Altshuler, Y., V. Yanovski, Wagner, I. A. and Bruckstein, A. M. (2009). Swarm ant robotics for a dynamic cleaning problem - upper bounds. 4th International Conference on Autonomous Robots and Agents, 2009. ICARA 2009. .

Altshuler, Y., V. Yanovsky, Wagner, I. A. and Bruckstein, A. M. (2005). The cooperative hunters-efficient cooperative search for smart targets using uav swarms, Citeseer.

Altshuler, Y., V. Yanovsky, Wagner, I. A. and Bruckstein, A. M. (2008). "Efficient cooperative search of smart targets using UAV Swarms." Robotica 26(04): 551-557.

Bahceci, E., O. Soysal and Sahin, E (2003). "A review: Pattern formation and adaptation in multi-robot systems." Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-43.

Balbo, F. and S. Pinson (2001). Toward a Multi-agent Modelling Approach for Urban Public Transportation systems. Engineering Societies in the Agents World II. A. Omicini, P. Petta and R. Tolksdorf, Springer Berlin / Heidelberg. 2203: 160-174.

Balch, T. and R. Arkin (1998). "Behavior-based formation control for multirobot teams." IEEE Transactions on Robotics and Automation 14(6): 926-939.

Batalin, M. A. and G. S. Sukhatme (2003). "Coverage, Exploration, and Deployment by a Mobile Robot and Communication Network." Information Processing in Sensor Networks: Second International Workshop, Ipsn 2003, Palo Alto, Ca, Usa, April 22-23, 2003: Proceedings.

Baum, M. and K. Passino (2002). A search-theoretic approach to cooperative control for uninhabited air vehicles.

Bayindir, L. and E. Sahin (2007). "A review of studies in swarm robotics." Turkish Journal of Electrical Engineering 15(2): 115-147.

Bayraktar, S., G. E. Fainekos and Pappas, G. J. (2004). Experimental cooperative control of fixed-wing unmanned aerial vehicles. 43rd IEEE Conference on Decision and Control.

Beard, R. W., T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson. (2006). "Decentralized Cooperative Aerial Surveillance Using Fixed-Wing Miniature UAVs." Proceedings of the IEEE 94(7): 1306-1324.

Bellingham, J. S., M. Tillerson, Alighanbari, M. and How, J. P. (2002). Cooperative path planning for multiple UAVs in dynamic and uncertain environments. Proceedings of the 41st IEEE Conference on Decision and Control, 2002.

Bernstein, R. A. (1975). "Foraging strategies of ants in response to variable food density." Ecology 56: 213 - 219.

Bogatyreva, O. and A. Shillerov (2006). "Robot swarms in an uncertain world: Controllable adaptability." Arxiv preprint cs/0601060.

Bonabeau, E., F. Henaux, Guérin, Sylvain; Snyers, Dominique; Kuntz, Pascale and Theraulaz, Guy (1998). Routing in Telecommunications Networks with Ant-Like Agents. Intelligent Agents for Telecommunication Applications. S. Albayrak and F. Garijo, Springer Berlin / Heidelberg. 1437: 60.

Bond, A. and L. Gasser (1988). "An analysis of problems and research in DAI." Readings in distributed artificial intelligence: 3-35.

Bruinsma, O. and L. Wageningen (1979). An analysis of building behaviour of the termite Macrotermes subhyalinus (Rambur), Landbouwhogeschool te Wageningen.

Bryson, M. and S. Sukkarieh (2009). "Architectures for Cooperative Airborne Simultaneous Localisation and Mapping." Journal of Intelligent Robotic Systems 55(4): 267-297.

Burgard, W. (2008). "Agent Standard Protocols for Building Multi-Robot Applications." Intelligent Autonomous Systems 10: IAS-10: 1.

Burgard, W., M. Moors, Stachniss, C. and Schneider, F. E. (2005). "Coordinated multi-robot exploration." IEEE Transactions on Robotics [see also IEEE Transactions on Robotics and Automation] 21(3): 376-386.

Burton, J. L. and N. R. Franks (1985). "The foraging ecology of the army ant Eciton rapax: an ergonomic enigma?" Ecological Entomology 10(2): 131-141.

C. Detrain, C. Natan. and J. L. Deneubourg (2001). "The Influence of the Physical Environment on the Self-organised Foraging Patterns of Ants." Naturwissenschaften 88: 171-174.

Camazine, S; Deneubourg, JL; Franks, NR; Sneyd, J; Theraula, G and Bonabeau, E (2003). Self-organization in biological systems, Princeton University Press.

Cao, Y. Uny, A. S. Fukunaga and Kahng, Andrew (1997). "Cooperative Mobile Robotics: Antecedents and Directions." Autonomous Robots 4(1): 7-27.

Carpenter, G. A., S. Martens and Ogi J. Ogas (2005). "Self-organizing information fusion and hierarchical knowledge discovery: a new framework using ARTMAP neural networks." Neural Networks 18(3): 287-295.

Cazangi, R. R., F. J. Von Zuben and Figueiredo, M. F. (2005). "Autonomous navigation system applied to collective robotics with ant-inspired communication." Proceedings of the 2005 conference on Genetic and evolutionary computation: 121-128.

Cervera, E. (2008). Practical multi-robot applications with player and JADE. 10th International Conference on Control, Automation, Robotics and Vision, 2008. ICARCV 2008.

Chella, A., M. Cossentino, Gaglio, Salvatore, Sabatucci, Luca and Seidita and Valeria (2010). "Agent-oriented software patterns for rapid and affordable robot programming." Journal of Systems and Software 83(4): 557-573.

Cheng, P., J. Fink, Kim, S and Kumar, V (2009). "Cooperative towing with multiple robots." Algorithmic Foundation of Robotics VIII: 101-116.

Chia-How, L., S. Kai-Tai and Anderson, G. T. (2005). Agent-based robot control design for multi-robot cooperation. IEEE International Conference on Systems, Man and Cybernetics.

Da Silva, V. T. and C. J. P. De Lucena (2004). "From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language." Autonomous Agents and Multi-Agent Systems 9(1): 145-189.

Daneshfar, F. and H. Bevrani (2009). "Multi-agent systems in control engineering: a survey." Journal of Control Science and Engineering. 2009: 2-2.

Danny Weyns, H. V. D. P., Fabien Michel, Tom Holvoet and Jacques Ferber (2005). "Environments for Multiagent Systems State-of-the-Art and Research Challenges." E4MAS 2004 LNAI 3374: 1 - 47.

Danny Weyns, M. S., Alessandro Ricci, Mirko Viroli and Tom Holvoet (2005). "Environments in multiagent systems." The Knowledge Engineering Review 20(2): 127 - 141.

Dasgupta, P. (2008). "A Multiagent Swarming System for Distributed Automatic Target Recognition Using Unmanned Aerial Vehicles." IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 38(3): 549-563.

De Wolf, T. and T. Holvoet (2005). Emergence Versus Self-Organisation: Different Concepts but Promising When Combined. Engineering Self-Organising Systems. S. A. Brueckner, G. D. M. Serugendo, A. Karageorgos and R. Nagpal, Springer Berlin / Heidelberg. 3464: 1-15.

DeLoach, S., E. Matson and YONGHUA LI (2003). "Exploiting agent oriented software engineering in cooperative robotics search and rescue." International Journal of Pattern Recognition and Artificial Intelligence 17(5): 817-835.

Deneubourg, J.-L. (1977). "Application de l’order par Fluctuations á la Description de Certaines étapes de la Construction de nid chez les Termites." Insectes Sociaux. 24: 117 - 130.

Deneubourg, J.-L., S. Aron, S. Goss, and J.-M. Pasteels (1989). "The Blind leading the Blind: Modelling Chemically Mediated Army Ants Raid Patterns." Journal of Insect Behaviour 2: 719 - 725.

Deneubourg, J. L. and S. Goss (1989). "Collective patterns and decision-making." Ethology Ecology & Evolution 1(4): 295 - 311.

Deneubourg, J. L. and S. C. Nicolis (1999). "Emerging Patterns and Food Recruitment in Ants: an Analytical Study " Journal of theoretical Biology 198: 575 - 592.

Detrain, C. and J.-L. Deneubourg (2008). Collective Decision-Making and Foraging Patterns in Ants and Honeybees. Advances in Insect Physiology. S. J. Simpson, Academic Press. Volume 35: 123-173.

Detrain, C. and J. Deneubourg (2002). "Complexity of environment and parsimony of decision rules in insect societies." The Biological Bulletin 202(3): 268.

DeLima, P., G. York and D. Pack. (2006). Localization of ground targets using a flying sensor network. Sensor Networks, Ubiquitous, and Trustworthy Computing, Taichung, IEEE.

Doherty, P. and P. Rudol (2007). "A UAV search and rescue scenario with human body detection and geolocalization." Lecture notes in computer science 4830: 1 - 13.

Dorigo, M. and E. Sahin (2004). "Special issue: swarm robotics." Autonomous Robots 17(2-3): 111–113.

Downing, H. A. and R. L. Jeanne (1988). "Nest construction by the paper wasp, Polistes: a test of stigmergy theory." Animal Behaviour 36(6): 1729-1739.

Dutta, I., A. D. Bogobowicz and Gu, J. J. (2004). "Collective robotics-a survey of control and communication techniques." Proceedings. 2004 International Conference on Intelligent Mechatronics and Automation, 2004: 505-510.

Eric Bonabeau, Marco Dorigo and Guy Theraulaz (1999). Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press.

Ferber, J. (1999). Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence Addison Wesley

Ferber, J., Olivier Gutknecht and Fabien Michel (2003). From Agents to Organizations: An Organizational View of Multi-agent Systems. Agent-Oriented Software Engineering IV. P. Giorgini, J. P. Müller and J. Odell, Springer Berlin / Heidelberg. 2935: 443-459.

Fox, D., J. Ko, Konolige, K., Limketkai, B., Schulz, D. and Stewart, B. (2006). "Distributed multi-robot exploration and mapping." Proceedings of the IEEE 94(7): 1325-1339.

Francesco, M., C. P. Giovanni, Andre, Guignard; Ivo, W. Kwee; Dario, Floreano; Jean-Louis, Deneubourg; Stefano, Nolfi; Luca Maria, Gambardella and Marco, Dorigo (2004). "Swarm-Bot: A New Distributed Robotic Concept: Swarm Robotics (Guest Editors: Marco Dorigo and Erol Sahin)." Autonomous Robots 17: 193-221.

Franchi, A., Freda, L., Oriolo, G. and Vendittelli, M. (2007). "A Randomized Strategy for Cooperative Robot Exploration." IEEE International Conference on Robotics and Automation: 768-774.

Franks, N. R., N. Gomez, S. Goss, and J.-L. Deneubourg (1991). "The Blind leading the Blind in Army Ant Raid Patterns: Testing A Model of Self-organisation (Hymenoptera: Formicidae)." Journal of Insect Behaviour 4: 583 - 607.

Fredslund, J. and M. J. Mataric (2001). Robot formations using only local sensing and control. IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2001. Proceedings.

Fredslund, J. and M. J. Mataric (2002). "A general algorithm for robot formations using local sensing and minimal communication." IEEE Transactions on Robotics and Automation 18(5): 837-846.

Garnier, S., J. Gautrais, Masoud, A, Christian, J and Guy, Theraulaz (2009). "Self-Organized Aggregation Triggers Collective Decision Making in a Group of Cockroach-Like Robots." Adaptive Behavior 17(2): 109-133.

Geyer, C. (2008). Active Target Search from UAVs in Urban Environment. IEEE International Conference on Robotics and Automation 2366 - 2371.

Gianluca, B., P. Domenico and N. Stefano (2006). "Distributed coordination of simulated robots based on self-organization." Artificial Life 12(3): 289-311.

Grasse, P.-P. (1959). "La Reconstruction du nid et les Coordinations Inter-Individuelles chez Bellicositermes Natalensis et Cubitermes sp. La theorie de la Stigmergie: Essai d'interpretation du Comportement des Termites Constructeurs." Insectes Sociaux 6: 41 - 80.

Grasse, P.-P. (1984). Termitologia, Tome II: Fondation des Societes. Construction. Paris, Masson.

Groß, R., M. Dorigo and M. Yamakita (2006). Self-assembly of mobile robots—from swarm-bot to super-mechano colony, Citeseer.

Hahn M, M. U. (1985). "Foraging strategies and recruitment behaviour in the European harvester ant Messor refitarsis." Oecologia 68: 45 - 51.

Hara, F., K. Sasajima and T. Sawada (1998). Morpho-functional characteristics of SSA-based two-mobile-robots system carrying a long bar on wavy road. IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998. Proceedings.

Hayes, A. T. (2002). "How Many Robots? Group Size and Efficiency in Collective Search Tasks " Proc. Int'l Symp. on Distributed Autonomous Robotic Systems, Springer-Verlag: pp 289 - pp 298.

Holland, O. and C. Melhuish (1999). "Stigmergy, self-organization, and sorting in collective robotics." Artificial Life 5(2): 173-202.

Hölldobler, B. (1976). "Recruitment behavior, home range orientation and territoriality in harvester ants, Pogonomyrmex." Behavioral Ecology and Sociobiology 1(1): 3-44.

Hölldobler, B. and E. O. Wilson (1990). The ants. Cambridge, Mass., Belknap Press of Harvard University Press.

J. McCarthy, M. L. Minsky, N. Rochester and Shannon, C.E. (1956). A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE. Dartmouth College in Hanover, New Hampshire.

James Kennedy and Russell C Eberhart, with Yuhui Shi (2001). Swarm Intelligence, Morgan Kaufmann.

James S. Jennings, Whelan, G. and Evans, W.F. (1997). Cooperative Search and Rescue with a Team of Mobile Robots. 8th International Conference on Advanced Robotics, Monterey, CA, IEEE.

Johnson, S. (2001). Emergence: The Connected Lives of Ants, Brains, Cities, and Software (Scribner, New York).

Joya, G., Frias, J.J., Mar Marin, M. and Sandoval, F (1993). "New learning strategies from the microscopic level of an artificial neural network." Electronics Letters 29(20): 1775-1777.

Karla Vittori, Jacques Cautrais, Aluizio F.R. Araújo; Vincent Fourcassié and Theraulaz, Guy (2004). "Modelling Ant Behaviour Under a Variable Environment." ANTS 2004 LNCS 3172: 190 - 201.

Ko, J., B. Stewart, Fox, D., Konolige, K. and Limketkai, B (2003). "A practical, decision-theoretic approach to multi-robot mapping and exploration." 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 4.

Kube, C. and E. Bonabeau (2000). "Cooperative transport by ants and robots." Robotics and autonomous systems 30: 85-101.

Kube, C. R. (1997). "Task Modelling in Collective Robotics." Autonomous Robots 4(1): 53-72.

Labella, T. H., M. Dorigo and Deneubourg, Jean-Louis (2004). Efficiency and Task Allocation in Prey Retrieval. Biologically Inspired Approaches to Advanced Information Technology. A. J. Ijspeert, M. Murata and N. Wakamiya, Springer Berlin / Heidelberg. 3141: 274-289.

Li, X. (2003). To investigate the current capability and feasibility of a swarm of micro UAVs for surveillance, reconnaissance, search, weather data collection, remote sensing and other similar operations, University of Warwick

Mailleux AC, D. J.-L. and Detrain C (2000). "Do ants assess food volume? ." Animal Behaviour 59: 1061 - 1069.

Marco, D.; T. Vito; Erol, ahin; Roderich, Gro; szlig; Thomas, H. Labella; Gianluca, Baldassarre; Stefano, Nolfi; Jean-Louis, Deneubourg; Francesco, Mondada; Dario, Floreano and Luca, M. Gambardella (2004). "Evolving Self-Organizing Behaviors for a Swarm-Bot: Swarm Robotics (Guest Editors: Marco Dorigo and Erol Sahin)." Autonomous Robots 17: 223-245.

Marco Dorigo, Eric Bonabeau and Guy Theraulaz (2000). "Ant Algorithms and Stigmergy." Future Generation Computer Systems 16: 851-871.

Marco Dorigo, M. B., Christian Blum, Luca M. Gambardella, Francesco Mondada and Thomas Stützle (Eds.) (2004). Ant Colony Optimisation and Swarm Intelligence. 4th International Workshop, ANTS 2004, Brussels, Belgium Springer.

Marios M. Polycarpou, Y. Y. and Kevin M. Passino (2001). A Cooperative Search Framework for Distributed Agents. Proceedings of the 2001 IEEE Inernational Symposium on Intelligent Control Mexico City, Mexico, IEEE.

Mataric, M. J., M. Nilsson and Simsarin, K. T. (1995). Cooperative multi-robot box-pushing. IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. .

McArthur, S. D. J., Davidson, E. M., Catterson, V. M., Dimeas, A. L., Hatziargyriou, N. D., Ponci, F. and Funabashi, T (2007). "Multi-Agent Systems for Power Engineering Applications Part I: Concepts, Approaches, and Technical Challenges." IEEE Transactions on Power Systems 22(4): 1743-1752.

McCorduck, P., M. Minsky, Selfridge, O and Simon, HA (1977). History of artificial intelligence, Morgan Kaufmann Publishers Inc.

McLurkin, J. and J. Smith (2007). Distributed Algorithms for Dispersion in Indoor Environments Using a Swarm of Autonomous Mobile Robots. Distributed Autonomous Robotic Systems 6. R. Alami, R. Chatila and H. Asama, Springer Japan: 399-408.

Menezes, Andy Tinkham and Ronaldo (2004). Simulating Robot Collective Behaviour using StarLogo. ACM - SE 2004, Huntsville, Alabama ACM.

Mirko Viroli, T. H., Alessandro Ricci, Kurt Schelfthout and Franco Zambonelli (2007). "Infrastructures for the environment of multi-agent systems." Autonomous Agents and Multi-Agent Systems 14: 49 - 60.

Mondada, F., Guignard, A., Bonani, M., Bar, D., Lauria, M. and Floreano, D. (2003). SWARM-BOT: from concept to implementation. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings.

Mondada, F., G. C. Pettinaro, Pettinaro, Andre Guignard, Ivo W. Kwee, Dario Floreano, Jean-Louis Deneubourg, Stefano Nolfi, Luca Maria Gambardella and Marco Dorigo (2004). "Swarm-Bot: A New Distributed Robotic Concept." Autonomous Robots 17(2): 193-221.

Morris, K. M., Mullins, B. E., Pack, D. J., York, G. W. P. and Baldwin, R. O. (2006). Impact of Limited Communications on a Cooperative Search Algorithm for Multiple UAVs. Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, 2006. ICNSC '06.

Moya, L. and A. Tolk (2007). Towards a taxonomy of agents and multi-agent systems, Society for Computer Simulation International.

Murphy, R. R. (2004). "Human-robot interaction in rescue robotics." IEEE Transactions on Systems, Man and Cybernetics, Part C 34(2): 138-153.

Negnevitsky, M. (2001). Artificial Intelligence: a guide to intelligent systems, Addison Wesley.

Ng, K. C. and B. Abramson (1990). "Uncertainty management in expert systems." IEEE Expert 5(2): 29-48.

Nicolis, S., C. Detrain, Demolin, D. and Deneubourg, J. (2003). "Optimality of collective choices: A stochastic approach." Bulletin of mathematical biology 65(5): 795-808.

Nonacs, P. and L. Dill (1988). "Foraging response of the ant Lasius pallitarsis to food sources with associated mortality risk." Insectes Sociaux 35(3): 293-303.

Nouyan, S. and M. Dorigo (2006). Chain Based Path Formation in Swarms of Robots. Ant Colony Optimization and Swarm Intelligence. M. Dorigo, L. Gambardella, M. Birattariet al, Springer Berlin / Heidelberg. 4150: 120-131.

Odell, J. J., H. V. D. Parunak, Fleischer, Mitch and Brueckner, Sven (2003). Modeling agents and their environment. Proceedings of the 3rd international conference on Agent-oriented software engineering III. Bologna, Italy, Springer-Verlag: 16-31.

Oprea, M. (2004). "Applications of multi-agent systems." Information Technology: 239-270.

Pack, D. J., DeLima, P., Toussaint, G. J. and York, G. (2009). "Cooperative Control of UAVs for Localization of Intermittently Emitting Mobile Targets." IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39(4): 959-970.

Pack, D. J. and B. E. Mullins (2003). Toward finding an universal search algorithm for swarm robots. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings.

Pack, D. J. and G. W. P. York (2005). Developing a Control Architecture for Multiple Unmanned Aerial Vehicles to Search and Localize RF Time-Varying Mobile Targets: Part I. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.

Palat, R. C., Annamalau, A. and Reed, J.R. (2005). Cooperative relaying for ad-hoc ground networks using swarm UAVs. Military Communications Conference, 2005. MILCOM 2005. IEEE.

Park, S. and V. Sugumaran (2005). "Designing multi-agent systems: a framework and application." Expert Systems with Applications 28(2): 259-271.

Parker, C. A. C., Hong, Zhang and Kube, C. R. (2003). Blind bulldozing: multiple robot nest construction. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings.

Parker, L. (2000). "Current state of the art in distributed autonomous mobile robotics." Distributed Autonomous Robotic Systems 4: 3–12.

Parunak, H. (1996). "Applications of distributed artificial intelligence in industry." Foundations of Distributed Artificial Intelligence 4: 139–164.

Parunak, HVD, Brueckner, S and Odell, JJ (2003). Swarming coordination of multiple UAV's for collaborative sensing. 2nd AIAA Unmanned Unlimited Systems Technologies and Operations Aerospace Land and Sea Conference and Workshop & Exhibit, San Diego, CA, Citeseer.

Parunak, H. V. D., S. Brueckner, Fleischer, Mitch and Odell, James (2003). A Design Taxonomy of Multi-agent Interactions. Agent-Oriented Software Engineering IV. P. Giorgini, J. P. Müller and J. Odell, Springer Berlin / Heidelberg. 2935: 123-137.

Passino, Kevin; Polycarpou, Marios; Jacques, David; Pachter, Meir; Liu, Yang; Yang, Yanli; Flint, Matt and Baum, Michael (2002). Cooperative Control for Autonomous Air Vehicles. Cooperative Control and Optimization. R. Murphey and P. M. Pardalos, Springer US. 66: 233-271.

Paul VALCKENAERS, M. K., Hendrik VAN BRUSSEL, Olaf BOCHMANN, C. ZAMFIRESCU (2001). The Design of Multi-Agent Coordination and Control Systems Using Stigmergy. In Proceedings of the 3rd International Workshop on Emergent Synthesis. Bled, Slovenia.

Platon, E., M. Mamei, Sabouret, N., Honiden, S. and Parunak, H. V. D. (2007). "Mechanisms for environments in multi-agent systems: Survey and opportunities." Autonomous Agents and Multi-Agent Systems 14(1): 31-47.

Portha, S., J.-L. Deneubourg and Detrain, Claire (2002). "Self-organized asymmetries in ant foraging: a functional response to food type and colony needs." Behavioral Ecology 13(6): 776-781.

Rao, A., M. Georgeff and Sonenberg, EA (1992). "Social plans: A preliminary report." Decentralized AI 3: 57-76.

Richard Hill, Simon Polovina and Beer, Martin (2005). "From Concepts to Agents: Towards a Framework for Multi-Agent System Modelling " AAMAS.

Richards, M., D. Whitley, Beveridge, JR, Mytkowicz, T, Nguyen, D and Rome, D (2005). Evolving cooperative strategies for UAV teams, ACM.

Rissing, S. W. (1988). "Dietary similarity and foraging range of two seed-harvester ants during resource fluctuations." Oecologia 75(3): 362-366.

Rudol, P., M. Wzorek, Conte, G. and Doherty, P. (2008). Micro Unmanned Aerial Vehicle Visual Servoing for Cooperative Indoor Exploration. Aerospace Conference, 2008 IEEE.

Ruini, F. and A. Cangelosi (2008). Distributed control in Multi-Agent Systems: A preliminary model of autonomous MAV swarms. 11th International Conference on Information Fusion.

Russell, S. and P. Norvig (2003). Artificial intelligence: a modern approach, Prentice hall.

Ryan, A.; J. Tisdale; Ryan, A.; Tisdale, J.; Godwin, M.; Coatta, D.; Nguyen, D.; Spry, S.; Sengupta, R. and Hedrick, J. K. (2007). Decentralized Control of Unmanned Aerial Vehicle Collaborative Sensing Missions. American Control Conference, 2007. ACC '07.

Ryan, A.; M. Zennaro; Howell, A.; Sengupta, R. and Hedrick, J.K. (2004). An overview of emerging results in cooperative UAV control. 43rd IEEE Conference on Decision and Control Atlantis, Paradise Island, Bahamas, IEEE.

Stéphane Portha, Jean-Louis Deneubourg and Claire Detrain (2004). "How food type and brood influence foraging decisions of Lasius niger scouts." Animal Behaviour 68: 115 - 122.

Sahin, E. (2005). "Swarm robotics: From sources of inspiration to domains of application." Swarm Robotics Workshop: State-of-the-art Survey: 10–20.

Sahin, E., W. Spears and Alan F. T. Winfield (2007). Swarm robotics, Springer-Verlag Berlin Heidelberg.

Sahin, H. and L. Guvenc (2007). "Household robotics: autonomous devices for vacuuming and lawn mowing [Applications of control]." Control Systems Magazine, IEEE 27(2): 20-96.

Saptharishi, M.; C. Spence Oliver; Diehl, C.P.; Bhat, K.S.; Dolan, J.M.; Trebi-Ollennu, A. and Khosla, P.K. (2002). "Distributed surveillance and reconnaissance using multiple autonomous ATVs: CyberScout." IEEE Transactions on Robotics and Automation 18(5): 826-836.

Schreinemachers, P.; T. Berger and Aune, Jens B. (2007). "Simulating soil fertility and poverty dynamics in Uganda: A bio-economic multi-agent systems approach." Ecological Economics 64(2): 387-401.

Sevcik, K. W.; W. E. Green and Oh, P. Y. (2005). Exploring search-and-rescue in near-earth environments for aerial robots. 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings.

Shen, W. and D. Norrie (1999). "Agent-based systems for intelligent manufacturing: a state-of-the-art survey." Knowledge and Information Systems 1(2): 129-156.

Shim, D. H.; C. Hoam and Sastry, S. S. (2006). "Conflict-free navigation in unknown urban environments." Robotics & Automation Magazine, IEEE 13(3): 27-33.

Sierra, C.; J. A. Rodriguez-Aguilar; Noriega, P.; Esteva, M. and Arcos, J. L. (2004). "Engineering multi-agent systems as electronic institutions." European Journal for the Informatics Professional 4: 33–39.

Simmons, R.; D. Apfelbaum; Burgard, W; Fox, D; Moors, M; Thrun, S and Younes, H (2000). Coordination for multi-robot exploration and mapping, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Simonin, O. and F. Gechter (2006). "An environment-based methodology to design reactive multi-agent systems for problem solving." Environments for Multi-Agent Systems II 3830: 32–49.

Smith, A. P. (1978). "An investigation of the mechanisms underlying nest construction in the mud wasp Paralastor sp. (Hymenoptera: Eumenidae)." Animal Behaviour 26(Part 1): 232-240.

Sujit, P. B. and D. Ghose (2004). "Search using multiple UAVs with flight time constraints." IEEE Transactions on Aerospace and Electronic Systems 40(2): 491-509.

Susi, T. and T. Ziemke (2001). "Social cognition, artefacts, and stigmergy: A comparative analysis of theoretical frameworks for the understanding of artefact-mediated collaborative activity." Cognitive Systems Research 2(4): 273-290.

Teodorovic, D. (2003). "Transport modeling by multi-agent systems: a swarm intelligence approach." Transportation Planning and Technology 26: 289-312.

Théraulaz, G. and E. Bonabeau (1995). "Coordination in Distributed Building." Science 269: 686 - 688.

Théraulaz, G. and E. Bonabeau (1995). "Modelling the Collective Building of Complex Architectures in Social Insects with Lattice Swarms." Journal of Theoretical. Biology 177: 381 - 400.

Theraulaz, G. and E. Bonabeau (1999). "A Brief History of Stigmergy." Artificial Life 5(2): 97-116.

Toussaint, G. J., P. De Lima and Pack, D.J (2007). Localizing RF Targets with Cooperative Unmanned Aerial Vehicles. American Control Conference, 2007. ACC '07.

Traniello, J. (1989). "Foraging strategies of ants." Annual Review of Entomology 34(1): 191-210.

Trianni, V. and M. Dorigo (2005). Emergent collective decisions in a swarm of robots. Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE.

Van Dyke Parunak, H. (1997). "" Go to the ant": Engineering principles from natural multi-agent systems." Annals of Operations Research 75: 69-101.

Vincent, P. and I. Rubin (2004). A framework and analysis for cooperative search using UAV swarms, ACM New York, NY, USA.

Regis Vincent, Dieter Fox, Jonathan Ko, Kurt Konolige, Benson Limketkai, Benoit Morisset, Charles Ortiz, Dirk Schulz and Benjamin Stewart (2008). "Distributed multirobot exploration, mapping, and task allocation." Annals of Mathematics and Artificial Intelligence 52(2): 229-255.

Andrew Viquerat, Lachlan Blackhall, Alistair Reid, Salah Sukkarieh and Graham Brooker (2008). Reactive collision avoidance for unmanned aerial vehicles using doppler radar, Springer: 245-254.

Pavel Vrba, Vladimír Mařík, Libor Přeučil, Miroslav Kulich and David Šišlák (2007). Collision Avoidance Algorithms: Multi-agent Approach. Holonic and Multi-Agent Systems for Manufacturing. V. Marík, V. Vyatkin and A. Colombo, Springer Berlin / Heidelberg. 4659: 348-360.

Weiss, G., Ed. (2000). Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence, The MIT Press.

Danny Weyns, Andrea Omicini and James Odell (2007). "Environment as a first class abstraction in multiagent systems." Autonomous Agents and Multi-Agent Systems 14(1): 5-30.

Wilensky, U. (1999). NetLogo.

William Agassounon, Alcherio Martinoli and Kjerstin Easton (2004). "Macroscopic Modeling of Aggregation Experiments using Embodied Agents in Teams of Constant and Time-Varying Sizes: Swarm Robotics (Guest Editors: Marco Dorigo and Erol Sahin)." Autonomous Robots 17: 163-192.

Wolf, A., H. B. Brown, Casciola, R., Costa, A., Schwerin, M., Shamas, E. and Choset, H (2003). A mobile hyper redundant mechanism for search and rescue tasks. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings.

Wooldridge, M. (2002). An Introduction to Multi-Agent Systems, John Wiley & Sons, Ltd.

Yanco, H. and J. Drury (2002). A taxonomy for human-robot interaction.

Yanli, Y., A. A. Minai and Polycarpou, M.M (2004). Decentralized cooperative search by networked UAVs in an uncertain environment. Proceedings of the American Control Conference, 2004.

York, G. W. P., D. J. Pack and Harder, J (2007). "Comparison of Cooperative Search Algorithms for Mobile RF Targets Using Multiple Unmanned Aerial Vehicles." LECTURE NOTES IN ECONOMICS AND MATHEMATICAL SYSTEMS 588: 387.

Yu, H., R. Beard and Byrne, J (2010). "Vision-based navigation frame mapping and planning for collision avoidance for miniature air vehicles." Control Engineering Practice 18(7): 824-836.

Zhang, Dandan; Fang, Yimin; Xie, Guangming; Yu, Junzhi and Wang, Long (2005). Coordinating Dual-Mode Biomimetic Robotic Fish in Box-Pushing Task. Advances in Artificial Life. M. S. Capcarrere, A. A. Freitas, P. J. Bentley, C. G. Johnson and J. Timmis, Springer Berlin / Heidelberg. 3630: 815-824.

Zomaya, Stephan Olariu and Albert Y. (2006). Handbook of Bioinspired Algorithms and Applications Taylor & Francis Group, LLC.

Appendix

A-1. NetLogo Simulation Programmes

1. Two-UAV Formation

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;; Two-UAV Formation ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;; Multiple UAVs search for Single/Multiple Moving Target(s) ;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

breed [ UAVs UAV ]
breed [ targets target ]

globals [ this-target
          diagonal-patch   ;; The patch in diagonal position of the calling UAV's patch
          converged-UAVs  ;; UAVs converged on potential target area
          num-of-converged  ;; Number of converged UAVs
          found-patch?  ;; Report true if found matched patches, false otherwise
          matched-patches  ;; The list used to record common patches
          located-patch  ;; The patch found by circling UAVs
          located-targets-count  ;; The number of reported target location
          num-of-located  ;; The number of individual located targets
          all-located?   ;; True if all targets are located successfully, false otherwise
        ]

UAVs-own [ signal?  ;; True if the UAV detected a target signal, false otherwise
           converged?   ;; True if the UAV is converged onto formation, false otherwise
           circled?  ;; True if converged UAVs complete circling, false otherwise
           my-patch  ;; Records the UAV's position
           my-pxcor my-pycor  ;; X, Y coordinates of my-patch
           my-detection-data
```

```
          nearby-UAVs  ;; Find nearby UAVs to communicate with
          the-closest-UAV  ;; Identify the closest UAV in the neighbourhood
        ]
patches-own [ target-signal ]

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;  SETUP PROCEDURES  ;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to setup
  clear-all
  set-default-shape UAVs "airplane"
  set-default-shape targets "target"
  set found-patch? false
  set all-located? false
  set located-targets-count 0
  set num-of-located 0
  create-UAVs num-of-UAVs
  [ set signal? false  ;; Signal detection is null at the beginning
    set color white  ;; No signal detection
    set converged? false  ;; Not converged yet
    set circled? false  ;; Haven't circled yet
    layout-circle sort UAVs 2 ]

  create-targets num-of-targets
  [ set color yellow
    ;; Randomly distribute targets and ensure they are
    ;; evenly located on the four parts of the search area
    if (distancexy 0 0) > -10
    [ if num-of-targets = 1
      [ setxy random-pxcor random-pycor ]
      if num-of-targets = 2
      [ ask n-of 2 targets
        [ ask one-of other targets
          [ setxy random-float max-pxcor random-float max-pycor ]
```

```
          setxy random-float min-pxcor random-float min-pycor ]
      ]
      if num-of-targets > 2
      [ let other-targets other targets in-radius max-pxcor
        if any? other-targets
        [ ask one-of other other-targets
          [ setxy random-float max-pxcor random-float min-pycor ]
          setxy random-float min-pxcor random-float max-pycor ]
      ] ] ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;  GO PROCEDURE  ;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to go
  ;; Terminate the search operation when all targets are located
  if located-targets-count = num-of-targets
  [ output-show (word "The total time of detection is " ticks)
    if num-of-located = num-of-targets
    [ set all-located? true
      output-show (word "All targets are located") ]
    stop ]

  ask UAVs [ ifelse signal? = false
             [ rt random 40
               lt random 40
               fd 0.05
               detect-target-signal ]
              ;; UAVs resume random search if not all targets are located
              [ if located-targets-count < num-of-targets
                [ resume-random-search ] ]
           ]
  ask targets [ rt random 30
              lt random 30
```

```
               fd 0.005 ]
  ;; Set the target signal to 1 if there is a target on patch
  ;; 0 if no target is in presence
  ask patches [ if any? targets-here [ set target-signal 1 ] ]
  ;; Record the time of detection
  tick
  ;; Plot the Time vs. Number of UAVs with Detection
  plot-performance
end

to detect-target-signal  ;; UAV procedure
  ;; Scan patches for the target signal
  if target-signal = 1
  [ set signal? true
    set color blue
    if not any? other UAVs-on patch-here
    [ face patch-here move-to patch-here
      ;; Update the individual detection range
      ;; Record the detection data as a list of 9 patches
      update-detection-data ]
    set this-target min-one-of targets [ distance myself ]
    if this-target != nobody
    [ set converged-UAVs UAVs-on [ neighbors ] of this-target
      set num-of-converged count converged-UAVs ]
    ;; When there are two UAVs converged in diagonal formation,
    ;; check if there is a common patch between them
    if num-of-converged = 2
    [ find-patch
      ;; If no common patch is found, the two UAVs start to circle together
      ;; in order to find a common patch between them.
      if found-patch? = false
      [ ask-concurrent converged-UAVs [ circle-clockwise ] ]
    ]
    ;; If required number of UAVs is not converged,
    ;; find the closest UAV to share the detection data
```

```
    if num-of-converged < 2
    [ find-nearby-UAVs
      if any? nearby-UAVs
      [ find-closest-UAV
        if (the-closest-UAV != nobody)
        [ ;output-show (word "Found the closest UAV " the-closest-UAV)
          ;; Identify the diagonal-patch based on my-patch
          find-diagonal-patch
          ;output-show (word "Identified diagonal patch " diagonal-patch)
          ;; Through communications, recruit the closest UAV to converge onto the
diagonal patch
          if diagonal-patch != 0
          [ communicate-and-converge ]
        ] ] ] ]
end

to update-detection-data
  ;; Reocrd the current patch of the UAV
  set my-patch patch-here
  set my-pxcor [ pxcor ] of my-patch
  set my-pycor [ pycor ] of my-patch
  ;; Record the individual detection range as a list of 9 patches
  set my-detection-data (list patch (my-pxcor - 1) (my-pycor + 1)
                              patch (my-pxcor) (my-pycor + 1)
                              patch (my-pxcor + 1) (my-pycor + 1)
                              patch (my-pxcor - 1) (my-pycor)
                              my-patch
                              patch (my-pxcor + 1) (my-pycor)
                              patch (my-pxcor - 1) (my-pycor - 1)
                              patch (my-pxcor) (my-pycor - 1)
                              patch (my-pxcor + 1) (my-pycor - 1))
end

to find-nearby-UAVs  ;; UAV procedure
  ;; Suitable UAVs are within the local neighbourhood and
```

```
  ;; shall have no signal detection of themselves at the time of communication
  set nearby-UAVs other UAVs in-radius 2 with [ signal? = false ]
end

to find-closest-UAV  ;; UAV procedure
  set the-closest-UAV min-one-of nearby-UAVs [ distance myself ]
end

to find-diagonal-patch
;; Such a diagonal formation is to ensure the target is covered by
;; the cooperative detection range of two converged UAVs.
 if target-signal = 1
 [ set diagonal-patch patch (my-pxcor + 2) (my-pycor + 2) ]
end

to communicate-and-converge ;; UAV procedure
  ;; The closest UAV copies the detection data
  ;; and moves to the diagonal patch.
  ;; Once converged in a diagonal formation with the detecting UAV,
  ;; the closest UAV carries out the same circling activity
  ask the-closest-UAV
  [ set signal? true
    set color blue
    face diagonal-patch
    move-to diagonal-patch
    ;output-show (word "I moved to patch " patch-here)
    update-detection-data
    set converged? true ]
end

to circle-clockwise
  ;; Define the 24 patches to be scanned for target signal.
  ;; Initiated from the UAV's current patch, these patches are in clockwise order.
  let patch-one patch-right-and-ahead -90 0.5
  let patch-two patch-right-and-ahead 0 0.5
```

```
let patch-three patch-right-and-ahead 90 0.5
let patch-four patch-right-and-ahead 90 1
let patch-five patch-right-and-ahead -180 0.5
let patch-six patch-right-and-ahead -180 1
let patch-seven patch-right-and-ahead -90 0.5
let patch-eight patch-right-and-ahead -90 1
let patch-nine patch-right-and-ahead -90 1.5
let patch-ten patch-right-and-ahead 0 0.5
let patch-eleven patch-right-and-ahead 0 1
let patch-twelve patch-right-and-ahead 0 1.5
let patch-thirteen patch-right-and-ahead 90 0.5
let patch-fourteen patch-right-and-ahead 90 1
let patch-fifteen patch-right-and-ahead 90 1.5
let patch-sixteen patch-right-and-ahead 90 2
let patch-seventeen patch-right-and-ahead -180 0.5
let patch-eighteen patch-right-and-ahead -180 1
let patch-nineteen patch-right-and-ahead -180 1.5
let patch-twenty patch-right-and-ahead -180 2
let patch-twenty-one patch-right-and-ahead -90 0.5
let patch-twenty-two patch-right-and-ahead -90 1
let patch-twenty-three patch-right-and-ahead -90 1.5
let patch-twenty-four patch-right-and-ahead -90 2

;; The circling process is made visible by drawing up
;; the pattern of this behaviour
pen-down
;; Before moving towards a new patch, the UAV check if the new patch
;; is occupied by another UAV. If not, it moves onto it.
;; Then the UAV updates detection data as its detection range changes every time
;; when it moves to a new patch.
;; The new detection data is processed to find common patches between two UAVs
;; If found a common patch, the target-located patch is identified.
;; If not, the two UAVs continue to move to the next patch and repeat the process
;; till the target-located patch is identified.
;; In total there are 24 patches throughout the circling process.
```

```
if not any? UAVs-on patch-one
[ face patch-one move-to patch-one
  update-detection-data
  find-patch
  if found-patch? = false
  [ if not any? UAVs-on patch-two
    [ face patch-two move-to patch-two
      update-detection-data
      find-patch
  if found-patch? = false
  [ if not any? UAVs-on patch-three
    [ face patch-three move-to patch-three
      update-detection-data
      find-patch
  if found-patch? = false
  [ if not any? UAVs-on patch-four
    [ face patch-four move-to patch-four
      update-detection-data
      find-patch
  if found-patch? = false
  [ if not any? UAVs-on patch-five
    [ face patch-five move-to patch-five
      update-detection-data
      find-patch
  if found-patch? = false
  [ if not any? UAVs-on patch-six
    [ face patch-six move-to patch-six
      update-detection-data
      find-patch
  if found-patch? = false
  [ if not any? UAVs-on patch-seven
    [ face patch-seven move-to patch-seven
      update-detection-data
      find-patch
```

```
if found-patch? = false
[ if not any? UAVs-on patch-eight
  [ face patch-eight move-to patch-eight
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-nine
  [ face patch-nine move-to patch-nine
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-ten
  [ face patch-ten move-to patch-ten
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-eleven
  [ face patch-eleven move-to patch-eleven
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-twelve
  [ face patch-twelve move-to patch-twelve
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-thirteen
  [ face patch-thirteen move-to patch-thirteen
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-fourteen
  [ face patch-fourteen move-to patch-fourteen
    update-detection-data
    find-patch
```

```
if found-patch? = false
[ if not any? UAVs-on patch-fifteen
  [ face patch-fifteen move-to patch-fifteen
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-sixteen
  [ face patch-sixteen move-to patch-sixteen
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-seventeen
  [ face patch-seventeen move-to patch-seventeen
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-eighteen
  [ face patch-eighteen move-to patch-eighteen
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-nineteen
  [ face patch-nineteen move-to patch-nineteen
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-nineteen
  [ face patch-nineteen move-to patch-nineteen
    update-detection-data
    find-patch
if found-patch? = false
[ if not any? UAVs-on patch-twenty
  [ face patch-twenty move-to patch-twenty
    update-detection-data
    find-patch
```

```
    if found-patch? = false
    [ if not any? UAVs-on patch-twenty-one
      [ face patch-twenty-one move-to patch-twenty-one
        update-detection-data
        find-patch
    if found-patch? = false
    [ if not any? UAVs-on patch-twenty-two
      [ face patch-twenty-two move-to patch-twenty-two
        update-detection-data
        find-patch
    if found-patch? = false
    [ if not any? UAVs-on patch-twenty-three
      [ face patch-twenty-three move-to patch-twenty-three
        update-detection-data
        find-patch
    if found-patch? = false
    [ if not any? UAVs-on patch-twenty-four
      [ face patch-twenty-four move-to patch-twenty-four
        update-detection-data
        find-patch
    if found-patch? = false
    [ resume-random-search ] ] ] ] ] ]
  ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ]
 ] ] ]
  ;; Stop drawing at the end of the circling process
  pen-up
  set circled? true
end

to-report match [ list1 list2 ]
  ;; Report true if found matched patches between the two UAVs in formation
  ;; false if no match has been found
  foreach list1 [ if (member? ? list2) and (target-signal = 1)
                  [ report true ] ]
  report false
```

```
end


to find-patch
 let List1 [ my-detection-data ] of one-of converged-UAVs
 let List2 [ my-detection-data ] of one-of other converged-UAVs
 ;; Check if there is any common patches in the patch-list of converged UAVs
 ;; Report true if found matched patches, false otherwise
 if (List1 != 0) and (List2 != 0)
 [ set found-patch? match List1 List2
   ;; If a matched patch is found, the target is located on the patch.
   ;; The coordinates of this patch is the coordinates of the target.
   if found-patch? = true
   [ let combined-list sentence List1 List2
     set matched-patches modes combined-list
     foreach matched-patches
     [ ask ? [ if target-signal = 1
               [ set located-patch ? ] ] ]
     if located-patch != 0
     [ ask converged-UAVs
       [ output-show (word "Found a target on " located-patch " at time " ticks) ]
       if this-target != nobody
       [ ;if located-patch = [ patch-here ] of this-target
         set num-of-located num-of-located + 1
         ask this-target [ ;output-show (word "Currently located on " patch-here "
at time " ticks)
                           die ] ]
       ;; When not all targets are located, increment the number of located targets
by 1
       if located-targets-count < num-of-targets
       [ set located-targets-count located-targets-count + 1 ]
     ] ] ]
end


to resume-random-search  ;; UAV procedure
  set signal? false
  set color white
```

```
  set circled? false
end

to plot-performance
  set-current-plot "UAVs in Circling Procedure"
  set-current-plot-pen "UAVs Engaged"
  plotxy ticks count UAVs with [ circled? = true ]
end
```

2. Three-UAV Formation

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;; Three-UAV Formation ;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;; Multiple UAVs search for Single/Multiple Moving Target(s) ;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

breed [ UAVs UAV ]
breed [ targets target ]

globals [ the-detecting-UAV  ;; The first UAV detected target signal
          nearby-UAVs  ;; Find nearby UAVs to communicate with
          UAV-one  UAV-two ;; Two nearby UAVs of the-detecting-UAV
          this-target
          diagonal-patch-one diagonal-patch-two
          ;; The two patches in diagonal position of the detecting UAV's patch
          converged-UAVs  ;; UAVs converged on potential target area
          num-of-converged  ;; Number of converged UAVs
          found-patch? found-patch-one? found-patch-two?
          ;; Report true if found matched patches, false otherwise
          matched-patches  ;; The list used to record common patches
          located-patch  ;; The patch found by circling UAVs
          located-targets-count  ;; The number of reported target location
          num-of-located  ;; The number of located targets
          all-located?  ;; The number of successful detection ]
```

```
UAVs-own [ signal?  ;; True if the UAV detected a target signal, false otherwise
           converged?   ;;  True  if  the  UAV  is  converged  onto  formation,  false
otherwise
           circled?  ;; True if converged UAVs complete circling, false otherwise
           my-patch  ;; Records the UAV's position
           my-pxcor my-pycor  ;; X, Y coordinates of my-patch
           my-detection-data ]

patches-own [ target-signal ]

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;  SETUP PROCEDURES  ;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to setup
  clear-all
  set-default-shape UAVs "airplane"
  set-default-shape targets "target"
  set found-patch? false
  set found-patch-one? false
  set found-patch-two? false
  set all-located? false
  set located-targets-count 0
  set num-of-located 0

  create-UAVs num-of-UAVs
  [ set signal? false  ;; Signal detection is null at the beginning
    set color white  ;; No signal detection
    set converged? false  ;; Not converged yet
    set circled? false  ;; Haven't circled yet
    layout-circle sort UAVs 2 ]

  create-targets num-of-targets
  [ set color yellow
```

```
    ;; Randomly distribute targets and ensure they are
    ;; evenly located on the four parts of the search area
    if (distancexy 0 0) > -10
    [ if num-of-targets = 1
      [ setxy random-pxcor random-pycor ]
      if num-of-targets = 2
      [ ask n-of 2 targets
        [ ask one-of other targets
          [ setxy random-float max-pxcor random-float max-pycor ]
          setxy random-float min-pxcor random-float min-pycor ]
      ]
      if num-of-targets > 2
      [ let other-targets other targets in-radius max-pxcor
        if any? other-targets
        [ ask one-of other other-targets
          [ setxy random-float max-pxcor random-float min-pycor ]
          setxy random-float min-pxcor random-float max-pycor ]
      ] ] ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;  GO PROCEDURE  ;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to go
  ;; Terminate the search operation when all targets are located
  ;; Otherwise record the number of located targets
  if located-targets-count = num-of-targets
  [ output-show (word "The total time of detection is " ticks)
    if num-of-located = num-of-targets
    [ set all-located? true
      output-show (word "All targets are located") ]
    stop ]

  ask UAVs [ ifelse signal? = false
```

```
               [ rt random 40
                 lt random 40
                 fd 0.05
                 detect-target-signal ]
                ;; UAVs resume random search if not all targets are located
                [ if located-targets-count < num-of-targets
                  [ resume-random-search ] ]
           ]
  ask targets [ rt random 30
                lt random 30
                fd 0.005 ]
  ;; Set the target signal to 1 if there is a target on patch
  ;; 0 if no target is in presence
  ask patches [ if any? targets-here [ set target-signal 1 ] ]
  ;; Record the time of detection
  tick
  ;; Plot the Time vs. Number of UAVs with Detection
  plot-performance
end

to detect-target-signal  ;; UAV procedure
  ;; Scan patches for the target signal
  if target-signal = 1
  [ set signal? true
    set color blue
    set this-target min-one-of targets [ distance myself ]
    if this-target != nobody
    [ set converged-UAVs UAVs-on [ neighbors ] of this-target
      set num-of-converged count converged-UAVs ]
    set the-detecting-UAV self
    face patch-here move-to patch-here
    ;; Record the current detection range as individual detection data
    update-detection-data
    ;; If required number of UAVs is not converged,
    ;; find the closest UAV to share the detection data
```

```
    if num-of-converged < 3
    [ find-nearby-UAVs
      ;; Identify the diagonal-patch based on my-patch
      find-diagonal-patch
      if (count nearby-UAVs > 1)
      [ ;; Find two nearby UAVs in the neighbourhood
        set UAV-one one-of nearby-UAVs
        set UAV-two one-of other nearby-UAVs
        if (UAV-one != nobody) and (UAV-one != the-detecting-UAV) and (UAV-two != UAV-one)
        [ ;; Through communications, recruit the closest UAV to converge onto the diagonal patch
          ask UAV-one [ converge-to-one ]
          ask UAV-two [ converge-to-two ]
        ] ] ]
    ;; When there are three UAVs converged in a triangular formation,
    ;; identify a unique common patch amongst them
    if num-of-converged = 3
    [ if (UAV-one != nobody) and (UAV-one != the-detecting-UAV) and (UAV-two != UAV-one)
      [ find-patch
        ;; If no patch is found, the three UAVs circle around
        ;; to search a wider area for target.
        if found-patch-two? = false
        [ ask-concurrent converged-UAVs [ circle-clockwise ] ]
      ] ] ]
end


to update-detection-data
  ;; Reocrd the current patch of the UAV
  set my-patch patch-here
  set my-pxcor [ pxcor ] of my-patch
  set my-pycor [ pycor ] of my-patch
  ;; Record a list of 9 patches as the individual detection data
  set my-detection-data (list patch (my-pxcor - 1) (my-pycor + 1)
                              patch (my-pxcor) (my-pycor + 1)
```

```
                              patch (my-pxcor + 1) (my-pycor + 1)
                              patch (my-pxcor - 1) (my-pycor)
                              my-patch
                              patch (my-pxcor + 1) (my-pycor)
                              patch (my-pxcor - 1) (my-pycor - 1)
                              patch (my-pxcor) (my-pycor - 1)
                              patch (my-pxcor + 1) (my-pycor - 1))
end

to find-nearby-UAVs  ;; UAV procedure
  ;; Suitable UAVs are within the local neighbourhood and
  ;; shall have no signal detection of themselves at the time of communication
  set nearby-UAVs other UAVs in-radius 2 with [ signal? = false ]
end

to find-diagonal-patch
  ;; Such a diagonal formation is to ensure the target is covered by
  ;; the cooperative detection range of two converged UAVs.
  if target-signal = 1
  [ set diagonal-patch-one patch (my-pxcor - 1) (my-pycor - 2)
    set diagonal-patch-two patch (my-pxcor + 1) (my-pycor - 2) ]
end

to converge-to-one ;; UAV procedure
  ;; The closest-UAV-one is recruited to move to the diagonal-patch-one
  set signal? true
  set color blue
  face diagonal-patch-one
  move-to diagonal-patch-one
  update-detection-data
  set converged? true
end

to converge-to-two
  ;; The closest-UAV-two is recruited to the diagonal-patch-two
```

```
  set signal? true
  set color blue
  face diagonal-patch-two
  move-to diagonal-patch-two
  update-detection-data
  set converged? true
end

to circle-clockwise
  ;; Define the 24 patches to be scanned for target signal.
  ;; Initiated from the UAV's current patch, these patches are in clockwise order.

  let patch-one patch-right-and-ahead -90 0.5
  let patch-two patch-right-and-ahead 0 0.5
  let patch-three patch-right-and-ahead 90 0.5
  let patch-four patch-right-and-ahead 90 1
  let patch-five patch-right-and-ahead -180 0.5
  let patch-six patch-right-and-ahead -180 1
  let patch-seven patch-right-and-ahead -90 0.5
  let patch-eight patch-right-and-ahead -90 1
  let patch-nine patch-right-and-ahead -90 1.5
  let patch-ten patch-right-and-ahead 0 0.5
  let patch-eleven patch-right-and-ahead 0 1
  let patch-twelve patch-right-and-ahead 0 1.5
  let patch-thirteen patch-right-and-ahead 90 0.5
  let patch-fourteen patch-right-and-ahead 90 1
  let patch-fifteen patch-right-and-ahead 90 1.5
  let patch-sixteen patch-right-and-ahead 90 2
  let patch-seventeen patch-right-and-ahead -180 0.5
  let patch-eighteen patch-right-and-ahead -180 1
  let patch-nineteen patch-right-and-ahead -180 1.5
  let patch-twenty patch-right-and-ahead -180 2
  let patch-twenty-one patch-right-and-ahead -90 0.5
  let patch-twenty-two patch-right-and-ahead -90 1
  let patch-twenty-three patch-right-and-ahead -90 1.5
```

```
let patch-twenty-four patch-right-and-ahead -90 2

;; The circling process is made visible by drawing up
;; the pattern of this behaviour
pen-down

;; Before moving towards a new patch, the UAV check if the new patch
;; is occupied by another UAV. If not, it moves onto it.
;; Then the UAV updates detection data as its detection range changes every time
;; when it moves to a new patch.
;; The new detection data is processed to find common patches between two UAVs
;; If found a common patch, the target-located patch is identified.
;; If not, the two UAVs continue to move to the next patch and repeat the process
;; till the target-located patch is identified.
;; In total there are 24 patches throughout the circling process.

if not any? UAVs-on patch-one
[ face patch-one move-to patch-one
  update-detection-data
  find-patch
  if found-patch-two? = false
  [ if not any? UAVs-on patch-two
    [ face patch-two move-to patch-two
      update-detection-data
      find-patch
  if found-patch-two? = false
  [ if not any? UAVs-on patch-three
    [ face patch-three move-to patch-three
      update-detection-data
      find-patch
  if found-patch-two? = false
  [ if not any? UAVs-on patch-four
    [ face patch-four move-to patch-four
      update-detection-data
      find-patch
```

```
if found-patch-two? = false
[ if not any? UAVs-on patch-five
  [ face patch-five move-to patch-five
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-six
  [ face patch-six move-to patch-six
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-seven
  [ face patch-seven move-to patch-seven
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-eight
  [ face patch-eight move-to patch-eight
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-nine
  [ face patch-nine move-to patch-nine
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-ten
  [ face patch-ten move-to patch-ten
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-eleven
  [ face patch-eleven move-to patch-eleven
    update-detection-data
    find-patch
```

```
if found-patch-two? = false
[ if not any? UAVs-on patch-twelve
  [ face patch-twelve move-to patch-twelve
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-thirteen
  [ face patch-thirteen move-to patch-thirteen
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-fourteen
  [ face patch-fourteen move-to patch-fourteen
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-fifteen
  [ face patch-fifteen move-to patch-fifteen
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-sixteen
  [ face patch-sixteen move-to patch-sixteen
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-seventeen
  [ face patch-seventeen move-to patch-seventeen
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-eighteen
  [ face patch-eighteen move-to patch-eighteen
    update-detection-data
    find-patch
```

```
if found-patch-two? = false
[ if not any? UAVs-on patch-nineteen
  [ face patch-nineteen move-to patch-nineteen
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-nineteen
  [ face patch-nineteen move-to patch-nineteen
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-twenty
  [ face patch-twenty move-to patch-twenty
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-twenty-one
  [ face patch-twenty-one move-to patch-twenty-one
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-twenty-two
  [ face patch-twenty-two move-to patch-twenty-two
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-twenty-three
  [ face patch-twenty-three move-to patch-twenty-three
    update-detection-data
    find-patch
if found-patch-two? = false
[ if not any? UAVs-on patch-twenty-four
  [ face patch-twenty-four move-to patch-twenty-four
    update-detection-data
    find-patch
```

```
    if found-patch-two? = false
    [ resume-random-search ] ] ] ] ] ] ] ] ]
  ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ]
  ;; Stop drawing at the end of the circling process
  pen-up
  set circled? true
end

to-report match-all-three [ list1 list2 list3 ]
  ;; Report true if found the patch among all three UAVs
  ;; false otherwise
  foreach list1
  [ if (member? ? list2) and (member? ? list3) and (target-signal = 1)
    [ report true ] ]
  report false
end

to-report match-two [ list1 list2 ]
  ;; Report true if found matched patches between the two UAVs in formation
  ;; false if no match has been found
  foreach list1 [ if (member? ? list2) and (target-signal = 1)
                [ report true ] ]
  report false
end

to find-patch
  let List1 [ my-detection-data ] of the-detecting-UAV
  let List2 [ my-detection-data ] of UAV-one
  let List3 [ my-detection-data ] of UAV-two
  ;; Check if there is any common patches in the patch-list of converged UAVs
  ;; Report true if found matched patches, false otherwise
  if (List1 != 0) and (List2 != 0) and (List3 != 0)
  [ ;; Three steps to identify the target-located patch
    ;; If there is a unique common patch with target signal, it is the target-
located patch
```

```
    ;; Put the three UAVs into two groups and check each group for detection of target signal

    ;; If both have a detection of target signal, the patch is where the target's residing

    set found-patch? match-all-three List1 List2 List3

    ifelse found-patch? = true

    [ let combined-list (sentence List1 List2 List3)

      set matched-patches modes combined-list

      foreach matched-patches

      [ ask ? [ if target-signal = 1 [ set located-patch ? ] ] ]

      if located-patch != 0

      [ ask the-detecting-UAV

        [ output-show (word "Found a target on " located-patch " at time " ticks) ]

        ask UAV-one

        [ output-show (word "Found a target on " located-patch " at time " ticks) ]

        ask UAV-two

        [ output-show (word "Found a target on " located-patch " at time " ticks) ]

        if this-target != nobody

        [ ;if located-patch = [ patch-here ] of this-target

          set num-of-located num-of-located + 1

          ask this-target [ ;output-show (word "Currently located on " patch-here " at time " ticks)

                            die ] ]

        if located-targets-count < num-of-targets

        [ set located-targets-count located-targets-count + 1 ]

      ]

    ]

    [ set found-patch-one? match-two List1 List2

      ifelse found-patch-one? = true

      [ let combined-list sentence List1 List2

        set matched-patches modes combined-list

        foreach matched-patches

        [ ask ? [ if target-signal = 1 [ set located-patch ? ] ] ]

        if located-patch != 0

        [ ask the-detecting-UAV

          [ output-show (word "Found a target on " located-patch " at time " ticks) ]
```

```
          ask UAV-one
          [ output-show (word "Found a target on " located-patch " at time " ticks) ]
          if this-target != nobody
          [ ;if located-patch = [ patch-here ] of this-target
            set num-of-located num-of-located + 1
            ask this-target [ ;output-show (word "Currently located on " patch-here " at time " ticks)
                              die ] ]
          if located-targets-count < num-of-targets
          [ set located-targets-count located-targets-count + 1 ]
        ]
      ]
      [ set found-patch-two? match-two List1 List3
        if found-patch-two? = true
        [ let combined-list sentence List1 List3
          set matched-patches modes combined-list
          foreach matched-patches
          [ ask ? [ if target-signal = 1 [ set located-patch ? ] ] ]
          if located-patch != 0
          [ ask the-detecting-UAV
            [ output-show (word "Found a target on " located-patch " at time " ticks) ]
            ask UAV-two
            [ output-show (word "Found a target on " located-patch " at time " ticks) ]
            if this-target != nobody
            [ ;if located-patch = [ patch-here ] of this-target
              set num-of-located num-of-located + 1
              ask this-target [ ;output-show (word "Currently located on " patch-here " at time " ticks)
                                die ] ]
            if located-targets-count < num-of-targets
            [ set located-targets-count located-targets-count + 1 ]
          ] ] ] ] ]
end
```

```
to resume-random-search  ;; UAV procedure
  set signal? false
  set color white
  set circled? false
  set found-patch? false
  set found-patch-one? false
  set found-patch-two? false
end

to plot-performance
  set-current-plot "UAVs in Circling Procedure"
  set-current-plot-pen "UAVs Engaged"
  plotxy ticks count UAVs with [ circled? = true ]
end
```

A-2. Experimental Data

Used in Chapter 6, the following tables list the number of UAVs, the average detection time and the number of targets detected within the maximum time length.

Table A1. Results of the Single Target Detection with Two-UAV Formation

Number of UAVs	Average Detection Time	All-Located Target Detections
20	10045.9	20
30	6223.2	20
40	3766.15	20
50	2485.95	20
60	2045.2	20
70	1778.55	20
80	1313.4	20
90	1483.45	20
100	1479.45	20
110	1311.15	20
120	2039	20

Table A2. Results of the Single Target Detection with Three-UAV Formation

Number of UAVs	Average Detection Time	All-Located Target Detections
20	78007.8	20
30	10654.85	20
40	5473.15	20

50	5693.75	20
60	3245.7	20
70	3053.7	20
80	3372.85	20
90	3915.5	20
100	3673.15	20
110	3540.55	20
120	2794.35	20

Table A3 and A4 present two sets of experimental data that each contains the average detection time and the number of all-located target detections generated from the experimental setup one (a.) – fixed numbers of targets versus increasing numbers of UAVs, and the experimental setup two (b.) – fixed numbers of UAVs versus increasing numbers of targets. For each combination of experiments, the average detection time is listed with the number of UAVs respectively. As for the all-located target detections, all targets are located within the maximum time length at each experimental run of the multiple target detection.

Table A3. Experimental Data of the Two-UAV Formation

a. Results of Experimental Setup One

Number of Targets	Number of UAVs	Average Detection Time	All-Located Target Detections
2	20, 30, 40, …, 120	11135.45, 5535.7, 5166.85, 3903.3, 2658, 2804.9, 2557.4, 2155.15, 2400.65, 2006.45, 2260.8	20
3	20, 30, 40, …, 120	13290.85, 7240.4, 4709.1, 4158.7, 3946.85, 3023.15, 3040.95, 2623.85, 2087.25, 2239.8,	20

		2064.85	
4	20, 30, 40, …, 120	14102.45, 8068.25, 7662.05, 4910.35, 3902.6, 3691.4, 3854.9, 2789.65, 2913.75, 2281.45, 2751.9	20
5	20, 30, 40, …, 120	45151.2, 13935, 6554.8, 5608.6, 4476.65, 4598.55, 3240.2, 3251, 3103.95, 3032.95, 2220.1	19, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20

b. Results of the Experimental Setup Two

Number of UAVs	Number of Targets	Average Detection Time	All-Located Target Detections
20	1, 2, 3, 4, 5	7775.25, 8348.95, 14516.2, 25570.25, 60206.85	20
30	1, 2, 3, 4, 5	3942.55, 4175.1, 8428.65, 10692.3, 9722.55	20
40	1, 2, 3, 4, 5	3196.55, 5121.05, 6327.35, 6919.2, 6914.45	20
50	1, 2, 3, 4, 5	2191.05, 2730.35, 4363.3, 5436.75, 6869.95	20
60	1, 2, 3, 4, 5	2030.4, 3662.15, 3961.3, 4409.4, 4881.25	20
70	1, 2, 3, 4, 5	2157.1, 3689.05, 3201.6, 4629.55, 3736.7	20
80	1, 2, 3, 4, 5	2061.65, 2152.3, 2738.35, 4131.65, 3075.2	20
90	1, 2, 3, 4, 5	2141.4, 2369.6, 3342.4, 3180.75, 3455.1	20
100	1, 2, 3, 4, 5	1734.55, 2058.65, 2685.7, 3209.6, 3284.3	20

110	1, 2, 3, 4, 5	1277.2, 1987.95, 2375.55, 3251.3, 2923.15	20
120	1, 2, 3, 4, 5	1280.1, 2553.65, 1961.4, 2292.45, 2921.05	20

Table A4.Experimental Data of the Three-UAV Formation

a. Results of the Experimental Setup One

Number of Targets	Number of UAVs	Average Detection Time	All-Located Target Detections
2	20, 30, 40, …, 120	195188.2, 14396.45, 11803.2, 9246.9, 5437.55, 7787.9, 5014.35, 3825.15, 3114.5, 4287.2, 4621.75	17, 20, 19, 18, 19, 17, 19, 18, 19, 19, 17
3	20, 30, 40, …, 120	417254.9, 110417.9, 11881.7, 7355.7, 9038.35, 9938.05, 5383, 6532.2, 4413.8, 4253.65, 5989.75	8, 14, 18, 17, 17, 17, 16, 19, 16, 17, 16
4	20, 30, 40, …, 120	525653.1, 161384.2, 25790.15, 12190, 9377.4, 7019.65, 7589.65, 6076.35, 5150.1, 4884.95, 6733.2	3, 15, 19, 17, 17, 17, 18, 14, 16, 14, 18
5	20, 30, 40, …, 120	598779.2, 208927.5, 65265.75, 18053.5, 13217.5, 10236.55, 8481.85, 6699.3, 7272.65, 7314.95, 5818.5	1, 12, 17, 18, 17, 18, 15, 15, 18, 18, 12

b. Results of the Experimental Setup Two

Number of UAVs	Number of Targets	Average Detection Time	All-Located Target Detections
20	1, 2, 3, 4, 5	15248.55, 148342.3, 461567.6, 436465.3, 551947.9	20, 16, 8, 7, 1

30	1, 2, 3, 4, 5	12395.45, 11477.3, 23588.1, 183538.9, 204693.5	20, 19, 17, 11, 10
40	1, 2, 3, 4, 5	7382.35, 10308.5, 10207.5, 19521.2, 92664.3	20, 20, 16, 18, 15
50	1, 2, 3, 4, 5	5892.25, 6422.4, 8732.2, 14251.5, 14707.6	20, 20, 16, 17, 15
60	1, 2, 3, 4, 5	5538.3, 9273.5, 7249.75, 12407, 11353.2	20, 19, 18, 18, 13
70	1, 2, 3, 4, 5	5076.75, 5299.7, 6859.5, 8407.5, 9649.75	20, 18, 18, 17, 14
80	1, 2, 3, 4, 5	3670.4, 4304.6, 7462.35, 7627.1, 9905.2	20, 20, 19, 16, 15
90	1, 2, 3, 4, 5	4862, 4208.05, 5058.3, 5159.55, 5803.85	20, 17, 15, 15, 14
100	1, 2, 3, 4, 5	4120.6, 3901.95, 4992.05, 5271.35, 6009.25	20, 17, 15, 18, 15
110	1, 2, 3, 4, 5	4331.1, 4203.85, 4159.25, 5396.6, 7971.45	20, 19, 18, 18, 16
120	1, 2, 3, 4, 5	2306.7, 4140.05, 4250.55, 4565.9, 5336.35	20, 18, 17, 18, 16

Table A5. Average Time Consumption for Two-UAV Formation

UAVs		**1 Target**	**2 Targets**	**3 Targets**	**4 Targets**	**5 Targets**
20						
	max	29011	32334	36963	279245	>600000
	avg	6787.65	7927.05	12717.95	26322.15	89041.925
	min	217	178	1976	2442	4309
30						
	max	10718	15034	16550	17527	28527

	avg	3264.6	5056.525	6935.85	8151.65	10728.65
	min	180	448	1412	1684	2970
40						
	max	16592	13342	13781	18106	17145
	avg	3294.5	4312.425	5180.675	6022.625	7522.2
	min	87	873	1048	2021	1549
50						
	max	6567	7612	9891	11020	11881
	avg	2175.575	3337.525	3773.825	4441.3	5184.85
	min	76	145	1130	1454	1178
60						
	max	10029	9030	10359	14948	9945
	avg	2547.775	2992	3622.05	4410.125	4842.8
	min	187	193	816	1509	1465
70						
	max	5560	8081	6679	11312	15492
	avg	2053.8	3022.7	3034.475	3621.875	4247.775
	min	170	410	1154	1050	753
80						
	max	6455	4908	13268	7819	10866
	avg	2019.725	2212.975	3767.65	3360.6	3904.1
	min	136	237	1088	1502	1117
90						
	max	4200	4794	5771	10258	8583
	avg	1594.35	2142.875	3093.55	3741.875	2991.95
	min	152	558	502	1474	1131
100						
	max	5934	4617	5842	7159	6276
	avg	1534.45	2136.475	2577.5	2987.6	2977.95
	min	190	452	1165	983	1307
110						
	max	4115	4332	6041	7685	9505
	avg	1599.2	1786.125	2743.05	2791.45	3059.575
	min	291	239	468	1233	953
120						

	max	4304	4906	6075	5980	5036
	avg	1805.1	1800.75	2289.425	2523.075	2824.35
	min	284	183	842	975	1413

Table A6. Average Time Consumption for Three-UAV Formation

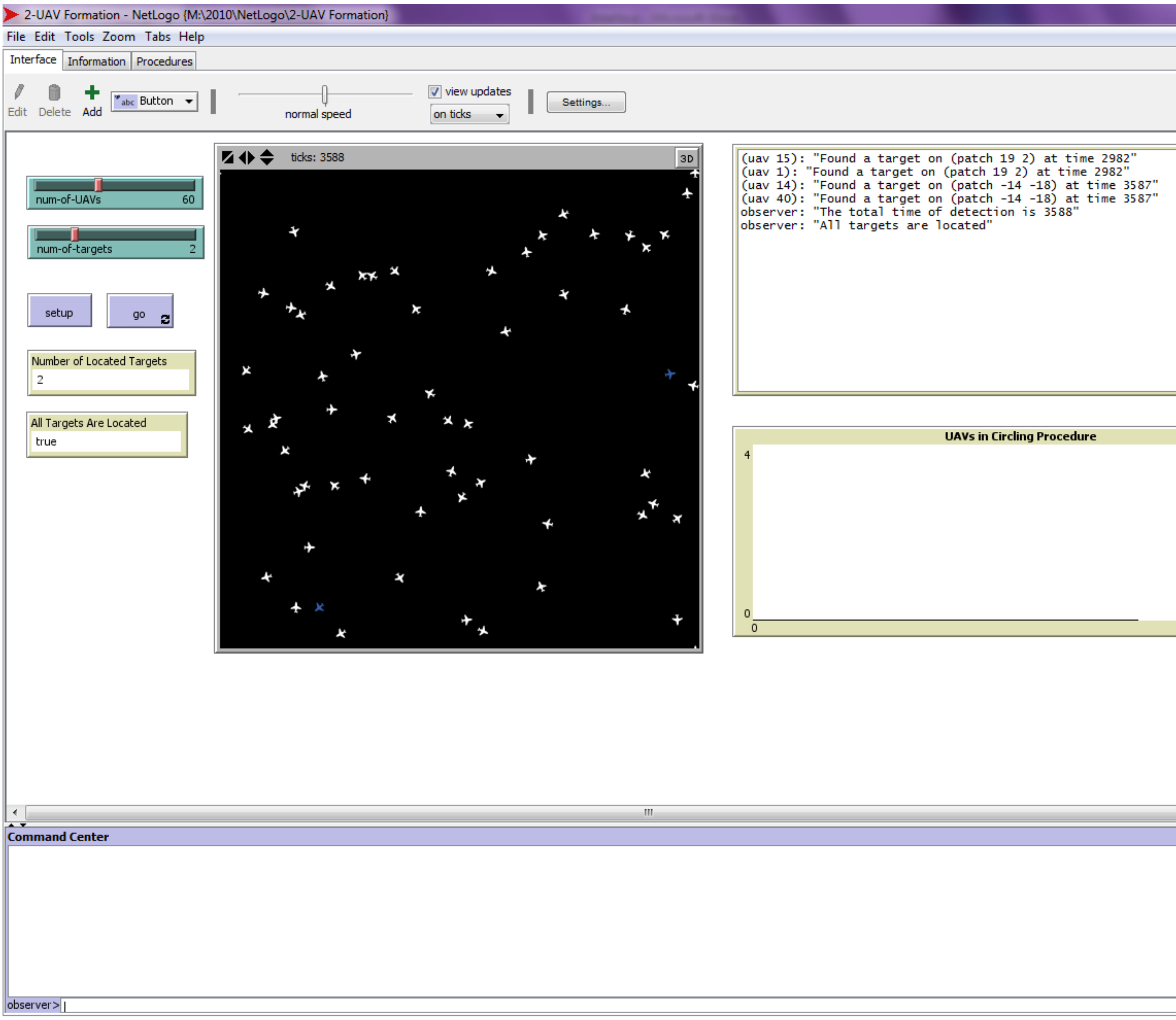
		1 Target	**2 Targets**	**3 Targets**	**4 Targets**	**5 Targets**
20						
	max	47913	>600000	>600000	>600000	>600000
	avg	12773.075	171765.25	439411.2	481059.18	575363.53
	min	787	327	6527	8987	4385
30						
	max	63147	49247	>600000	>600000	>600000
	avg	9973.65	12936.875	67003	172461.53	206810.5
	min	1162	34	1648	2610	3420
40						
	max	65698	34278	30582	148214	>600000
	avg	8454.7	11055.85	11044.6	22655.675	78965.025
	min	184	572	3832	4200	4286
50						
	max	41309	31043	20734	43216	53448
	avg	7814.25	7834.65	8043.95	13220.75	16380.55
	min	194	924	626	1428	2730
60						
	max	27472	26753	25574	30672	28493
	avg	5718.55	7355.525	8144.05	10892.2	12285.35
	min	262	1142	2055	1378	3109
70						
	max	25134	21924	37017	19675	38579
	avg	4999.575	6543.8	8398.775	7713.575	9943.15
	min	230	513	1282	1453	1018
80						
	max	15952	15468	15951	24226	26332
	avg	3522.825	4659.475	6422.675	7608.375	9193.525

	min	291	460	207	2103	1932
90						
	max	23574	14087	14533	13541	15782
	avg	4125.525	4016.6	5795.25	5617.95	6251.575
	min	533	867	439	2270	896
100						
	max	14636	8068	12282	21336	19062
	avg	3524.825	3508.225	4702.925	5210.725	6640.95
	min	375	326	864	1685	1823
110						
	max	21394	10284	9190	16723	20212
	avg	4517.1	4245.525	4206.45	5140.775	7643.2
	min	675	283	748	1327	2778
120						
	max	7168	14040	15164	17790	12637
	avg	2719.625	4380.9	5120.15	5649.55	5577.425
	min	192	1350	460	1851	2137

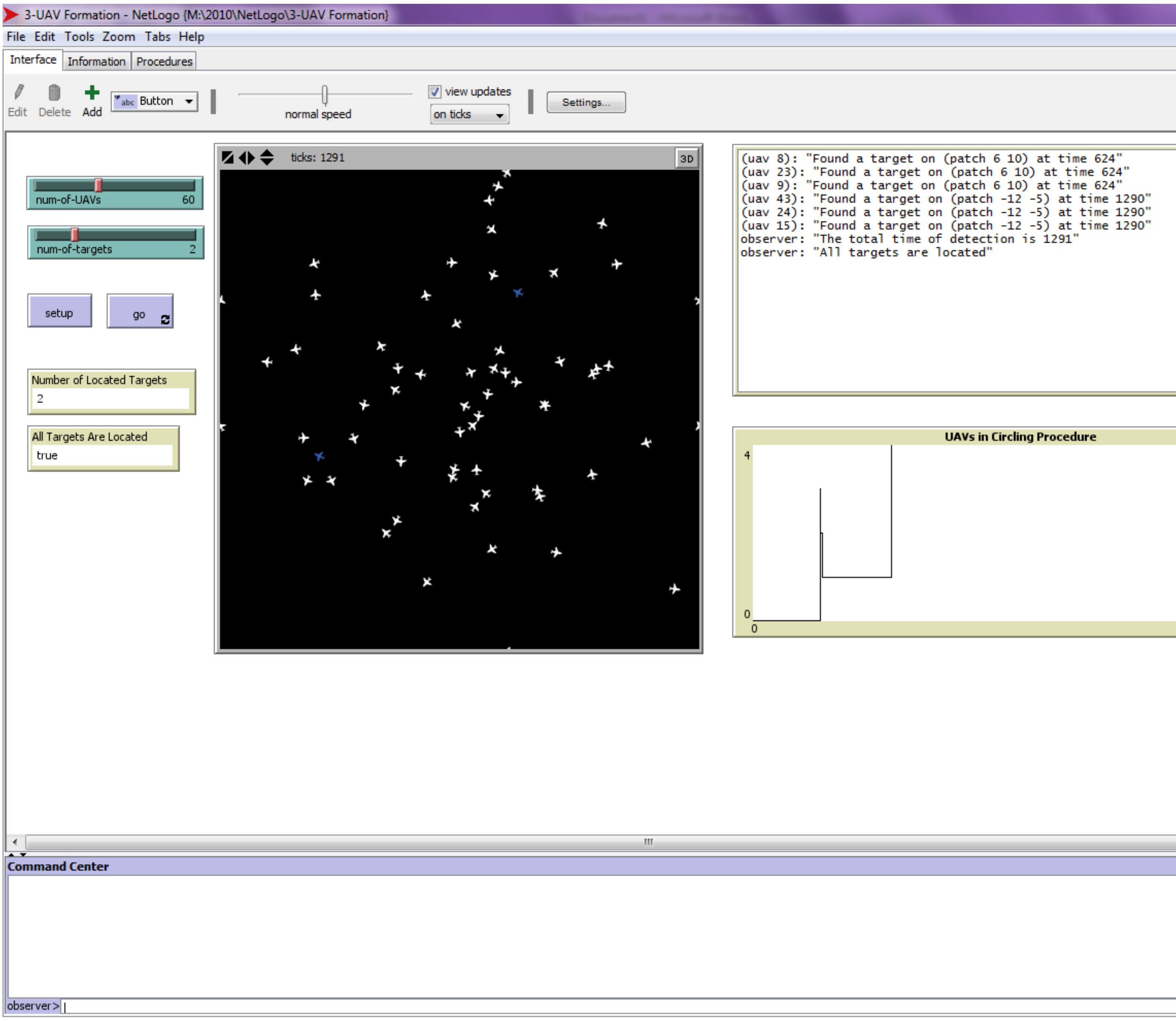
A-3. Programme Output

1. Two-UAV Formation

2. Three-UAV Formation



2-UAV Formation - NetLogo {M:\2010\NetLogo\2-UAV Formation}
File Edit Tools Zoom Tabs Help
Interface
Information
Procedures
Edit
Delete
Add
Button
normal speed
view updates
on ticks
Settings...
num-of-UAVs 60
num-of-targets 2
setup
go
Number of Located Targets
2
All Targets Are Located
true
ticks: 3588
3D
(uav 15): "Found a target on (patch 19 2) at time 2982"
(uav 1): "Found a target on (patch 19 2) at time 2982"
(uav 14): "Found a target on (patch -14 -18) at time 3587"
(uav 40): "Found a target on (patch -14 -18) at time 3587"
observer: "The total time of detection is 3588"
observer: "All targets are located"
UAVs in Circling Procedure
4
0
0
Command Center
observer>



3-UAV Formation - NetLogo {M:\2010\NetLogo\3-UAV Formation}
File Edit Tools Zoom Tabs Help
Interface
Information
Procedures
Edit
Delete
Add
Button
normal speed
view updates
on ticks
Settings...
num-of-UAVs 60
num-of-targets 2
setup
go
Number of Located Targets
2
All Targets Are Located
true
ticks: 1291
3D
(uav 8): "Found a target on (patch 6 10) at time 624"
(uav 23): "Found a target on (patch 6 10) at time 624"
(uav 9): "Found a target on (patch 6 10) at time 624"
(uav 43): "Found a target on (patch -12 -5) at time 1290"
(uav 24): "Found a target on (patch -12 -5) at time 1290"
(uav 15): "Found a target on (patch -12 -5) at time 1290"
observer: "The total time of detection is 1291"
observer: "All targets are located"
UAVs in Circling Procedure
4
0
0
Command Center
observer>