Time Preference for Information in Multi-Agent Exploration with Limited Communication

Victor Spirin, Stephen Cameron, and Julian de Hoog

University of Oxford

Abstract. Multi-agent exploration of unknown environments with limited communication is a rapidly emerging area of research with applications including surveying and robotic rescue. Quantifying different approaches is tricky, with different schemes favouring one parameter of the exploration, such as the total time of exploring 90% of the environment, at the expense of another parameter, like the rate of information update at a base station. In this paper we present a novel approach to this problem, in which agents choose their actions based on the time preference of the base station for information, which it encodes as the desired minimum ratio of base station utility to total agent utility. We then show that our approach performs competitively with existing exploration algorithms while offering additional flexibility, and holds the promise for much improvement regarding incorporation of various information preferences for the base station.

1 Introduction

Multi-agent exploration of environments where communication between agents is limited has been a rapidly emerging area of research in recent years. Several approaches have been suggested, ranging from those that aim to always maintain a communication link between the agents and the base station, ensuring that any new information gets to the base station as soon as it is discovered [8, 1], to approaches where agents are allowed to explore the environment without putting any effort to bring the information back to the base station until the exploration effort is over [13], to strategies that lie in between.[6, 3, 9, 10]. For simplicity of modelling most of the work has assumed a two-dimensional environment where the aim is to provide information at a single base station, but the approaches have natural extensions to more complicated domains.

Each of those approaches has their strengths and weaknesses; they all work best in different specific scenarios. From looking at their performance, we can see that there is one major factor that allows us to decide which approach to choose in favour of another in any given situation - and that factor is the *time preference* of the base station for information about the environment. The base station may value information obtained sooner higher than information obtained later - for example, when rescuing people from a building that is on fire, the human rescue team is likely to prefer to have information about *some* of the environment sooner, than have a complete map of the environment later, when it may already be too late. In other scenarios, however, we may prefer to have information about the whole of the environment sooner, and not be too interested in getting a more constant flow of information. We may also have preferences that lie somewhere in between.

In this paper we present a novel approach that attempts to solve the problem by providing the human operator controlling a team of robots with an intuitive way of specifying what their time preference for new information is, and have the agents automatically adapt their cooperative behaviour accordingly. We describe our approach in section 3, and in section 4 we present results obtained in simulation that show how with our approach, agents can adapt their behaviour to changes in the time preference. We also compare our results with other approaches, and analyse how the performance of the system changes as the time preference is changed. In section 5 we discuss ways in which this approach can be further developed and extended.

2 Related work

Since our approach was developed to primarily deal with scenarios where a team of agents has a goal of exploring an unknown environment, while having limited communication between each other and the base station, in this section we will give an overview of some of the existing approaches to the same problem. Of course, when exploring unknown environments with limited communication, there is always going to be a tradeoff between the speed and frequency of getting new information to the base station, and the time it takes to explore the whole environment. To the best of our knowledge, all existing approaches to the problem optimise for a particular tradeoff between the two. Here we will look at an approach that aims to explore all of the environment as soon as possible, and an approach that aims to minimise the latency in getting new information to the base station while still being able to explore all of the environment in reasonable time. All of the approaches we describe here are built upon the frontier-based exploration framework as described in [11].

2.1 Frontier exploration

Using frontiers to distribute the exploration task among multiple agents is a common approach to multi-agent exploration. A frontier is a boundary between the explored and the unexplored parts of the map. [13] Agents can then allocate frontiers among themselves by estimating the path costs of themselves and other agents in their vicinity to the frontiers, so as to maximise overall exploration utility. However, a frontier on its own gives no information on the potential information gain of exploring the area that lies behind the frontier, which can lead to inefficient allocations. The concept of *frontier polygons* was introduced in [12] to deal with that problem; a frontier polygon is the polygon that is formed between a frontier and the boundaries of *safe space*. Safe space is comprised of

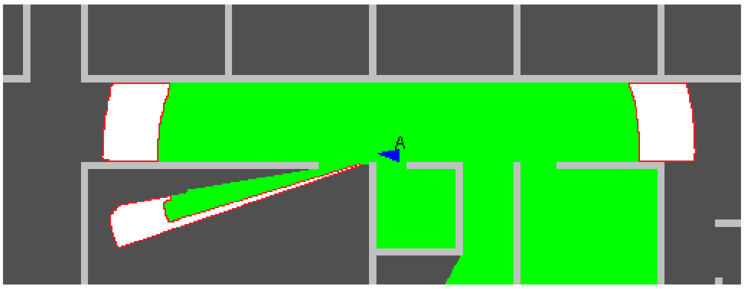

Fig. 1. An example of a map in simulation with visible frontier polygons. A robot is shown in blue, green areas are 'safe space', while white areas need to be further explored and are bounded by 'frontier polygons', shown in red.

the areas of the map that the agent has been closer to than the full range of his sensor (normally, it is set to be around half the sensor's full range).

We can then estimate the potential information gain from a frontier by using the area of the frontier polygon as the estimate. This can be especially important for our proposed approach, as having an estimate of potential information gain from a frontier is crucial when deciding whether to continue to explore the frontier, or to return to the base station; but it can enhance the frontier exploration approach in general by allowing agents to better allocate frontiers among themselves. In particular, we can control the exploration behaviour by setting the constant n in the equation which estimates the potential information gain when calculating the utility of a frontier

$$U(p_i) = A(p_i)/C^n(p_i) \tag{1}$$

where $U(p_i)$ is the utility of the frontier polygon p_i, $A(p_i)$ is the area of the frontier polygon, and $C(p_i)$ is the cost of the path to the frontier polygon's centre. Low values of n mean that agents will favour the exploration of larger frontier polygons, such as corridors or halls, and higher values of n mean that the agents will more often tend to examine nearby smaller areas, such as rooms [11]. In most experiments in this paper we used the value $n = 2$, which tends to provide a good balance between the two in practice [5].

2.2 Return when done

This strategy is the most straightforward, and is the direct application of frontier exploration. All agents continue to explore the environment without having to return to the base station at regular intervals, and only once all the frontiers have been explored do they return to base. In practice, this often ensures that the whole of the environment is explored in shortest time, however that may not always be the case due to the fact that only returning to the base station after the exploration is finished means that not only is the base station not getting frequent updates on the exploration effort, but information sharing between agents may also be reduced, which may lead to the same areas getting explored more than once.

2.3 Role-based Exploration

With this approach, agents are divided into two groups: explorers and relays. The task of the explorers is to explore as much of the environment as possible and return it back to pre-agreed rendezvous points at pre-arranged times. The task of relays is to communicate information between rendezvous points (where they communicate with explorers or other relays), or between a rendezvous point and the base station.

Here, teams of agents have a rigid hierarchy tree which is manually selected before the agents enter the environment; however, agents may switch positions in the tree throughout their mission. However, the shape of the tree itself does not change.

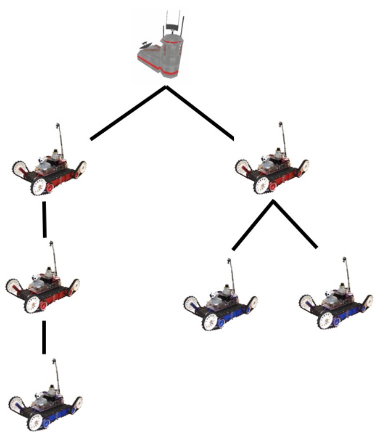

Fig. 2. An example of the agent hierarchy tree. The base station is at the root of the tree; relays are shown in red, and explorers are shown in blue.

When an explorer meets its parent relay, they exchange information about the environment, ensuring that they both have the same knowledge about it. Then, the explorer suggests a rendezvous point, normally near a frontier that it plans to explore next, and a fallback rendezvous point in case the primary one cannot be reached, which is especially useful in dynamic environments (if the explorer and the relay happen to meet before they both reach their meeting point, they act as if they have reached it and proceed to exchange information and to replan their next meeting). Because the explorer and the relay share the same information, the explorer can predict how long it will take for the relay to get back to its own parent, and then travel to the new meeting point, and can therefore decide when it should stop exploring the new frontiers and get back to the agreed rendezvous point [6].

The selection of the rendezvous points is therefore crucial for the performance of the algorithm, as it affects how long the agents get to spend exploring the environment, and how often they deliver their information back to the base station. The further away from the base station the meeting points are, the more biased the exploration effort is to exploring deeper into the environment instead of relaying the information back [4]. Also, selecting meeting points at junctions and in corridors where the communication range is wider leads to increased performance of the exploration approach. [7]. However, this approach does not allow to easily move the rendezvous points closer to the base station / deeper into the environment to favour either faster exploration or more frequent communication with the base station, and as such, when comparing our proposed approach against it, we used the implementation described in [4], where rendezvous points are selected as close as possible to the frontier that the explorer plans to investigate next, while trying to place the meeting point in a junction or an open space, which maximises communication range.

3 Proposed Method

3.1 Overview

When a team of autonomous agents is sent to explore an unknown environment where communication is limited, resources can generally be allocated in two different ways: collecting new information about the environment; and keeping the base station (and other agents) updated of the current progress made by the agents. Usually, an agent would have to allocate its resources to only one of the above tasks at any given time. As a result, the human operator commanding a team of robots would need to somehow specify how the team resources should be allocated between the two tasks.

There are several straightforward ways of doing that. The operator may want to specify maximum latency of information propagation in the team - that is, the maximum time between information exchanges of an agent with the base station. While at first it seems like a useful way of specifying the desired team behaviour and resource allocation, there are several problems with it. How can the operator know apriori what maximum latency is appropriate for a given scenario? It is likely to be desirable for the latency to be very low at the start of the exploration, while the agents are exploring parts of the environment close to the base station, and to increase as the agents progress deeper into the environment. But how should the latency increase, to ensure that agents do not waste their resources communicating with the base station when no new information has been discovered to communicate, and that all of the environment gets explored? The operator may not have the answers to these questions without some information about the environment, and by the time that information is obtained it could be too late to set the behaviour of the team.

Another way of specifying the desired team behaviour could be by setting the rate of information update at the base station to be a function of utility gathered by the agents that has not yet been delivered to the base station. For example, if we are only interested in building a map of the environment, the operator might specify the desired team behaviour by setting the target minimum ratio of information about the environment known by the base station, to the amount of information known by all the agents combined (Fig. 3).

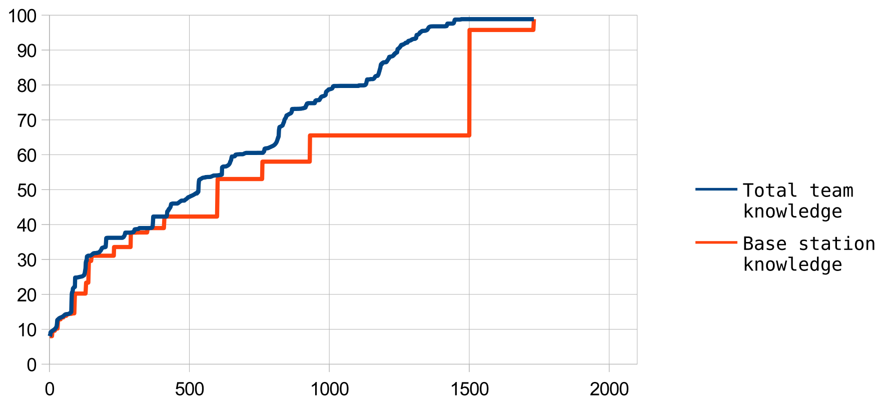

Fig. 3. A graph showing combined team knowledge and base station knowledge changes with time over the course of a simulated exploration mission using role-based exploration strategy.

That single parameter, the target ratio, would then be a real number ranging from 0 to 1. Setting it to 0 would result in greedy exploration behaviour, where all of the team resources are used to gather new information; while setting it to 1 would ensure maximum connectivity to the base station. By setting the ratio to a value between 0 and 1, it is possible for the operator to specify how they want the team resources to be allocated between discovering new information and maintaining communication with the base station in a meaningful way. Of course, as the agents in the team are operating with limited information about the state of other agents in the team and about the environment, their behaviour will not match the target ratio precisely, but as we show in the section describing our simulation results, even with the team using a simple heuristic that crudely approximates the resulting ratio, it provides a promising way of specifying the desired team behaviour, with the team behaviour changing accordingly with the target ratio changes.

3.2 Implementation

For this paper, we used a simple implementation of the approach described above. Before the start of the exploration, the user sets a target information ratio $targetInfoRatio \in [0; 1)$, which gets propagated to all the agents in the team before the exploration begins.

For each agent i, let $infBase_i$ be the information i believes the base station to have at the current time. $infBase_i$ is obtained directly from communicating with the base station, or from communicating with other agents that have communicated with the base station more recently than i.

Let $infNew_i$ be the information about the environment that i knows, excluding $infBase_i$, and excluding the information that i has given to other agents to relay to base. When i gets into communication range with an agent j, and j is closer to base than i, $infNew_i$ is added to $infNew_j$, i marks $infNew_i$ as relayed and hence sets $infNew_i$ to $\varnothing$. This is done to reduce the risk of several agents trying to deliver the same information to base.

Then, during each cycle of the exploration, each agent can be in one of two states: exploring the environment (using the frontier exploration approach outlined in section 2.1), or returning to the base station. An agent i only decides to return to the base station if

$$|infBase_i|/(|infBase_i| + |infNew_i|) < targetInfoRatio, \tag{2}$$

where $|infBase_i|$ and $|infNew_i|$ are the utilities of $infBase_i$ and $infNew_i$ accordingly. Otherwise agent i continues to explore the environment.

4 Simulation Results

4.1 Simulator

We used the MRESim simulator [6] to evaluate our approach. MRESim simulates sensor data, communication, movements and collisions of multiple agents in a 2D environment consisting of free space and obstacles. The actions performed by the simulator at each time step are shown in Algorithm 1.

Actions taken at each time step by MRESim

```
foreach agent do
   nextLoc = requestDesiredLocation(agent);
   if isValid(nextLoc) then
      move(agent, nextLoc);
      sensorData = simulateSensorData(agent, nextLoc);
      sendData(agent,sensorData);
   end
end
foreach agent do
   foreach agent2, agent2 != agent do
```

```
      if isInRange(agent, agent2) then
         communicateData(agent, agent2);
      end
   end
end
updateGUI();
```

The simulator assumes perfect localisation and sensor data. While this assumption is unrealistic, it still allows us to get a good idea about how different agent cooperation strategies perform against each other and see what their strengths and limitations are.

For all of the experiments, we used a standard path loss communication model with a wall attenuation factor [2].

4.2 Set up

We used 4 different maps for our experiments, as shown in Fig. 4: a small room-based map that consists of corridors and a number of rooms to be explored; a cluttered environment; a large "library" map, consisting of many rooms and corridors to be explored; and a large outdoor environment. We initially did 4 runs on each of the maps using 6 different exploration strategies, a total of 96 runs: using our approach with target ratios of 0.95, 0.90, 0.75, 0.50 and 0.30 and using role-based exploration. For each of the runs on the first and second maps, we had a total of 4 agents navigating the environment; for the library map and for the outdoors map we used a total of 8 agents. In the runs where we used role-based exploration, half of those agents were assigned the roles of "relays", each of them relaying information for one other agent.

The results obtained from doing the runs on the 4 maps appeared to be similar to each other, so we decided to focus on running a larger number of simulations on the "rooms" map. We ended up doing 48 runs of each type on the "rooms" map, for a total of 288 runs. We present the results of those simulations below.

4.3 Results

The results of the simulation runs on the "rooms" map are shown in Fig. 5.

As we can see, our approach with a target ratio of 0.95 manages to explore 98% of the environment faster than role-based exploration, while the average ratio of total agent knowledge to base station knowledge is very similar to that of role-based exploration. This was an expected result, as our approach does not designate a number of agents to be used only as relays throughout the simulation runs, which should result in a more efficient use of resources to reach a particular target ratio.

Another interesting observation is that as we decrease the target ratio from 0.95 to 0.3, reducing the average and minimum actual observed ratios between total agent and base station utilities accordingly, it has less and less of an effect on increasing the overall speed of exploration.

Fig. 4. 4 maps used in the simulations: rooms, cluttered, library and outdoors (starting from top left, clockwise)

4.4 Emergent behaviour

During our simulations, we found that with higher values for the target ratio and with higher number of agents (4 or more), at the start of the exploration, all of the agents go off to explore new frontiers and to collect new information. However, as the exploration effort gets deeper into the environment, a number of agents end up acting as dedicated relays, simply going back and forth between the base station and the other exploring agents. Often, agents would behave as chains of relays - at the later stages of the exploration, for example, it is possible for the majority of agents to start acting as relays and only for a few agents to keep exploring. Of course, there is no explicit agreement made between the agents to allocate or assume those roles, and neither do they make agreements about where or when they should meet. The way it appears to happen is as follows:

1. A number of agents meet while they are returning to the base station to deliver their information. This may happen either if the agents flock to an area that has a number of promising frontiers, or if they meet in a corridor while returning to the base station from different areas.
2. They exchange information, and the agent nearest to the base station assumes the responsibility of delivering their combined new information to base.
3. After delivering that information, this "relay" agent proceeds to the area with the most promising frontiers. Since he has the same knowledge as the

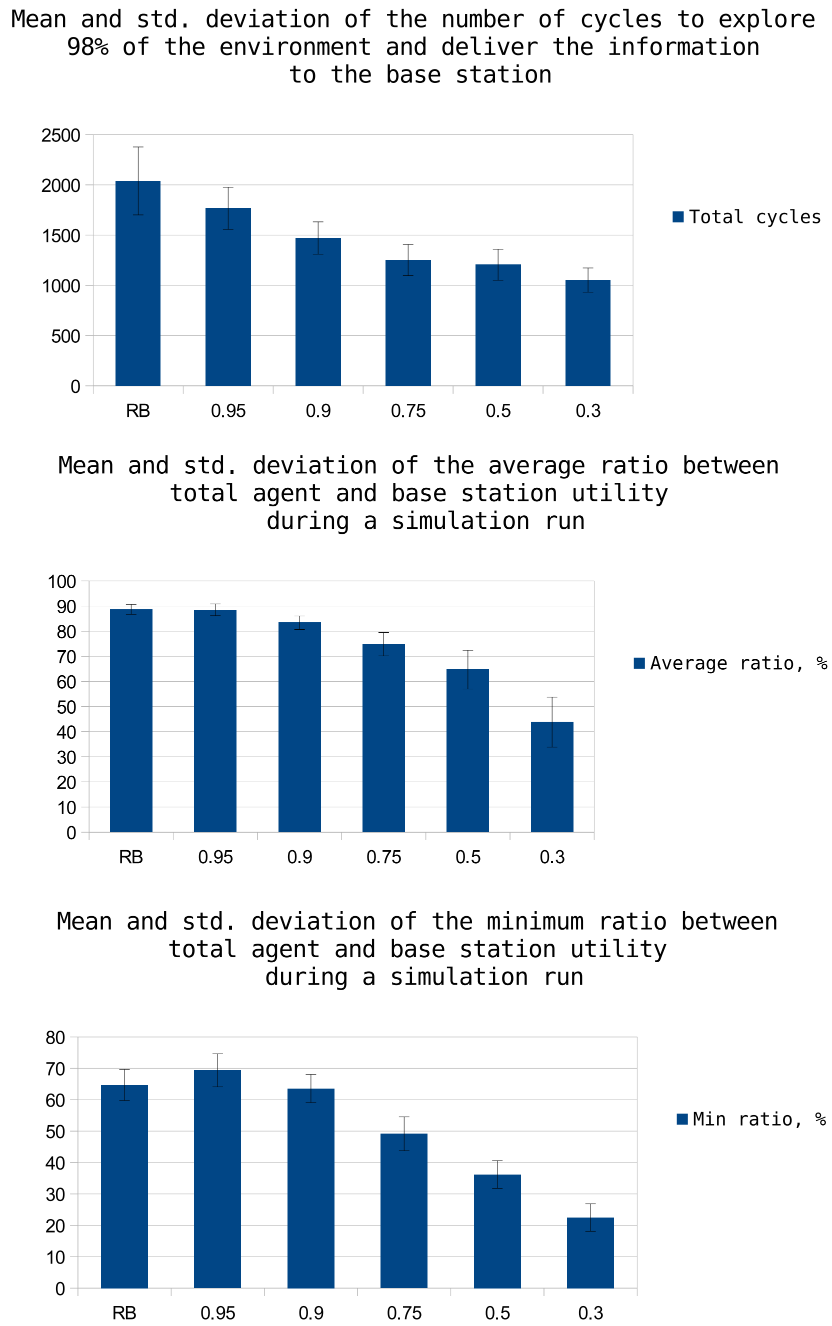

Fig. 5. Graphs showing mean and standard deviation values obtained from the simulation runs using role-based exploration, and our approach using target ratios of 0.95, 0.9, 0.75, 0.5 and 0.3.

other agent had at the time of meeting, he will likely go to the same "promising" frontier as the other agent did, meeting him - or another agent relaying for him and returning to base - on the way.

The cycle above results in the emergent behaviour of chains of relays being formed as they are needed to keep delivering information to the base station at a frequency that is appropriate for the target ratio set up by the operator.

5 Conclusions

We have shown a simple, but effective way of specifying the desired team behaviour by means of setting a single numeric parameter, the target ratio of base station utility to total agent utility. We have presented an implementation of a distributed exploration strategy that takes the target ratio into account and adjusts the behaviour of the team accordingly, and we have shown that it performs competitively with role-based exploration while offering additional flexibility. We have also shown that the gain in the total speed of exploration that we get when we reduce the target ratio seems to get a lot smaller than the corresponding increase in the cost (reduced rate of base station updates) as the target ratio gets closer to 0 - which may be useful when deciding which target ratio should be used for any particular situation.

Some of the extensions we would like to explore include having better estimates by agents of what the total agent knowledge is and how it is going to increase in the future, as well as what the current base station knowledge is. We are also interested in exploring the effects of having a low-bandwidth communication link between all agents, such as VHF radio, that would allow them to communicate their positions and their estimates of how much new information they possess.

References

1. R.C. Arkin and J. Diaz. Line-of-sight constrained exploration for reactive multiagent robotic teams. In *7th International Workshop on Advanced Motion Control. Proceedings (Cat. No.02TH8623)*, pages 455–461. IEEE, 2002.
2. Paramvir Bahl and Venkata N Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. *IEEE Infocom 2000, Tel-Aviv, Israel*, 2:775 – 784, 2000.
3. Tucker Balch and Matthew Powers. Value-Based Communication Preservation for Mobile Robots. *7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.
4. Julian De Hoog. Role-Based Multi-Robot Exploration. *D.Phil Thesis, Department of Computer Science, University of Oxford*, 2011.
5. Julian De Hoog. Using Mobile Relays in Multi-Robot Exploration. *In Proceedings of ACRA (Australian Conference on Robotics and Automation). Melbourne, Australia*, 2011.

6. Julian De Hoog, Stephen Cameron, and Arnoud Visser. Role-Based Autonomous Multi-robot Exploration. *Proceedings of the International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE)*, 2009.
7. Julian De Hoog, Stephen Cameron, and Arnoud Visser. Selection of Rendezvous Points for Multi-Robot Exploration in Dynamic Environments. *Proceedings of AAMAS (Workshop on Agents in Realtime and Dynamic Environments, International Conference on Autonomous Agents and Multi-Agent Systems). Toronto, Canada*, 2010.
8. A. Howard, M.J. Mataric, and G.S. Sukhatme. An incremental deployment algorithm for mobile robot teams. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pages 2849–2854. IEEE, 2002.
9. Alejandro R. Mosteo, Luis Montano, and Michail G. Lagoudakis. Multi-robot routing under limited communication range. In *2008 IEEE International Conference on Robotics and Automation*, pages 1531–1536. IEEE, May 2008.
10. J. Vazquez and C. Malcolm. Distributed multirobot exploration maintaining a mobile network. In *2nd International IEEE Conference on 'Intelligent Systems'. Proceedings (IEEE Cat. No.04EX791)*, pages 113–118. IEEE, 2004.
11. Arnoud Visser and B Slamet. Balancing the information gain against the movement cost for multi-robot frontier exploration. *European Robotics Symposium*, 2008.
12. Arnoud Visser, Merlijn Van Ittersum, Luis A. Gonzalez Jaime, and Laurentiu A. Stancu. Beyond frontier exploration. *Proceedings of the 11th Robocup International Symposium*, 2007.
13. B. Yamauchi. Frontier-based exploration using multiple robots. *AGENTS '98: Proceedings of the second international conference on Autonomous agents. New York, NY, USA: ACM*, pages 47–53, June 1998.