A DECENTRALISED APPROACH TO MULTI-ROBOT FORMATION INITIALISATION

J. K. Archibald and R. L. Frost
Department of Electrical and Computer Engineering
Brigham Young University, Provo, Utah, 84602 USA
{jka,rickf}@ee.byu.edu

Abstract

This paper presents a novel solution to the formation initialisation problem, in which autonomous mobile robots must arrange themselves into a specific geometric configuration without centralised control, without explicit inter-agent communication, and using only information from local sensors which is necessarily incomplete. Our solution to this fundamental multi-robot coordination problem is based on satisficing theory, a paradigm for decision making in which individual options are evaluated by comparing potential gains with projected costs. An option can be justified as adequate if its benefits outweigh its costs. In multi-agent systems inclined to cooperation, satisficing offers advantages relative to traditional utility-maximisation approaches. We present experimental results that demonstrate cooperative, emergent multi-agent behaviour. We compare the performance of various enhancements within the satisficing framework across a collection of scenarios that differ in starting positions, target formations, number of agents involved, and the presence of static obstacles. The results suggest that satisficing is an attractive alternative for the synthesis of cooperative multi-robot systems.

Keywords

Mobile robots, formation initialisation, satisficing theory, multi-agent coordination, cooperative control.

1 Introduction

There is substantial interest in using groups of coordinating autonomous vehicles to perform such tasks as reconnaissance, remote sensing, and hazard monitoring. For many applications, it is desirable for robots to assume and maintain a desired geometric formation. Formation problems correspond to a variety of multi-agent agreement problems and are therefore representative of more general coordination challenges [1].

In this study we address the *formation initialisation problem* in which a system of n

randomly placed mobile robots arrange themselves into an arbitrary configuration [2]. From the outset, the global coordinates of all n target locations are known to all robots, but target selection and control are completely distributed. Each robot selects a target and moves to it based solely on information acquired through its local sensors – no communication is permitted between robots. Each robot knows its own location and the locations of some number of closest robots. Because the number of robots equals the number of target locations, initialisation is complete when each target location is occupied by exactly one robot.

As a motivating scenario, imagine that the robots are equipped with sensors, dropped from the air, and expected to realize a specific pattern for surveillance. The locations of all targets are programmed into the robots before the start of the mission, but the starting locations of robots cannot be known in advance. Throughout the mission, communication between agents is prohibited to reduce the chance of detection. The challenge is to achieve the desired formation with robots making decisions based solely on local information.

Previously proposed multi-agent formation strategies generally fit into one of three categories: *leader-follower*, *behavioural*, and *virtual structures*. In leader-follower, agents generally move to a specified offset from a leader. Researchers have described control laws for mobile robots using a leader-follower approach [3, 4] and explored a distributed approach to select the best leader [5]. Applications using leader-follower range from working cooperatively to move a box [6] to organising robots into a chain [7].

Behavioural approaches prescribe multiple desired behaviours for each agent; control is determined by a weighted average of the control for each behaviour. Behavioural approaches have been used to maintain formation [8], create line and circle formations [9], transport objects [10], and achieve a desired group behaviour using only local communication [11].

With virtual structures, the entire formation is treated as a single rigid structure. Control is derived by defining the desired dynamics of the overall structure, determining the desired motion for each agent, and then deriving tracking controls for each agent. The virtual structure approach has been applied to formations of mobile robots [12] and to formations of spacecraft [13].

Coordination schemes that rely on centralised planning or extensive inter-agent communication tend to be fragile when operating in uncontrolled real-world environments. To maximise reliability and scalability, it is desirable that group control be distributed – as much as possible, each robot should determine its own actions based solely on its local information. The principal challenge is achieving the desired high-level behaviour of the entire group when only the behaviour of individual agents can be specified. Other researchers have explored the use of distributed algorithms in gathering a group of robots to a single point [1, 14], in covering an area of interest with mobile sensors [15], and in achieving a desired shape and position of a swarm of mobile robots [16].

In this paper, we present a new solution technique for multi-agent coordination problems. The technique is based on a novel theory of decision making under uncertainty, which is briefly described in Section 2. The hardware and software structure of the robot and simulation test-bed used in this study is described in Section 3. Section 4 presents an analysis of various solutions and their performance. Concluding remarks are found in Section 5.

2 Satisficing Theory

Satisficing theory is a rigorous comparative paradigm for decision making that is based on the mathematics of probability theory with different semantics. In this section, we describe the key ideas. The full ntheory is described elsewhere in detail [17].

Given a set of options, satisficing agents identify those that are "good enough" by comparing two utility functions, one referred to as *selectability* representing potential gains, and the second referred to as *rejectability* representing potential losses. If an option's selectability outweighs its rejectability, the option is deemed acceptable. Both utility functions are normalised measures, ensuring comparable units when comparing costs and benefits.

Given utility functions, one can construct the set of all options for which the benefits outweigh the costs. More precisely, we can construct the *satisficing set*, Σ_b, defined as follows:

$$\Sigma_b = \{u \in U\colon p_S(u) \geq b p_R(u)\} \tag{1}$$

where U is the set of options, p_S is the selectability mass function, p_R is the rejectability mass function, and b reflects the agent's relative weighting of resource conservation and goal achievement. If $b \leq 1$ then Σ_b is nonempty. For all studies described in this paper, $b = 1$.

Satisficing theory extended to the multi-agent case offers both a way to express complex inter-agent relationships and a way for individual agents to balance individual and group interests. Utilities can be defined that represent group selectability and rejectability for the group decision space, the set of possible decision vectors across the group, and a group satisficing set can be determined. In this work, we consider cooperative but myopic agents that do not explicitly model group-level utility functions.

Elsewhere we describe how individual and group utilities can be obtained from a complete specification of the relationships in the group [18, 19]. In related studies, we have shown that this approach is effective for systems where the agents are naturally inclined to cooperation, such as aircraft collision avoidance and coordinated search [20, 21, 22, 23]. Our studies suggest that cooperative behaviour exhibited by satisficing agents results from conditioning individual preferences on the *preferences* of other agents, rather than on their *actions*.

2.1 Satisficing for Formation Initialisation

Each of the n robots picks a specific target based on the relative locations of its nearest $m-1$ neighbours and m targets ($m \leq n$). The size of the decision space for each individual agent is thus independent of n, enhancing scalability. Notation used in utility functions is from the point of view of the agent making the decision, labelled A_1. Its $m-1$ nearest neighbours, arranged in order of increasing distance from A_1, are labelled $\{A_2, A_3, \ldots, A_m\}$. A formation F consists of the set of n targets $\{T_1, T_2, \ldots, T_n\}$.

Let U_1 be the decision space for agent A_1. Let $u_i^1 \in U_1$ represent an option vector

$\{t_{i1}, t_{i2}, \ldots, t_{im}\}$, where t_{ij} is the target corresponding to agent A_j in option u_i^1. Finally, let $\delta(t_{ij})$ represent the distance from agent A_j to its target t_{ij}.

Many satisficing formulations are possible. This study uses (2) to determine selectability. The term $\max_j(\delta(t_{ij}))$ is the greatest distance from any coordinating agent to its target in this option, and the term $\max_{k,j}(\delta(t_{kj}))$ is the greatest distance any coordination agent must travel to its target across all options. The denominator normalises the function. In this formulation, selectability is greatest for options that result in the smallest maximum distance travelled by neighbouring agents to their targets.

$$P_s(u_i^1)_{u_i^1 \in U_1} = \frac{1 - \frac{\max_j(\delta(t_{ij}))}{\max_{k,j}(\delta(t_{kj}))}}{\sum_{l=1}^{m!}\left(1 - \frac{\max_j(\delta(t_{lj}))}{\max_{k,j}(\delta(t_{kj}))}\right)} \tag{2}$$

Rejectability is given by (3). The numerator represents the total distance travelled by local agents in moving to their targets, and the denominator normalises the function. The lowest rejectability value will be assigned to options that minimise total distance travelled by neighbour agents.

$$P_r(u_i^1)_{u_i^1 \in U_1} = \frac{\sum_{i=1}^{m} \delta(t_{ij})}{\sum_{k=1}^{m!}\sum_{j=1}^{m} \delta(t_{kj})} \tag{3}$$

The option is chosen with the greatest difference between selectability and rejectability. Once it has selected a target, the robot moves directly to it unless a collision is detected, in which case the target selection algorithm is repeated. Although options include target assignments for multiple robots, the choice determines only the target for the local agent. The likely preferences of neighbouring agents are taken into consideration by each agent, facilitating locally cooperative behaviour.

3 Evaluation Test-bed

Our study used facilities of BYU's MAGICC Lab, including simulation software and custom robots operating with an overhead camera for positioning. Each robot has an on-board computer, two wheels powered by DC motors, and a wireless RF link to a dedicated host computer. The simulation software is tightly integrated into the overall system; system components communicate with others without knowing whether they are real or simulated. Agents are implemented using a three-layered control architecture as illustrated in Fig. 1.

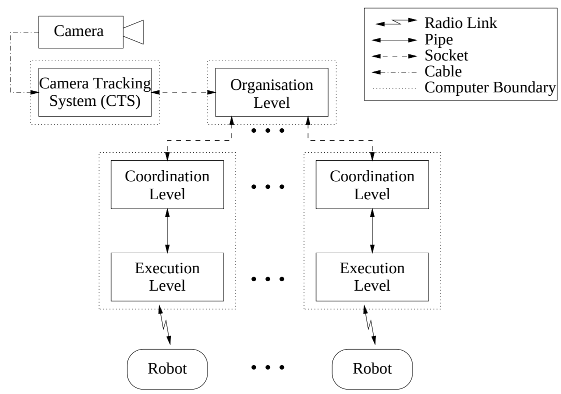

Figure 1: Structure of components in experimental test-bed

The *Organisation Level* contains interfaces to the user and to each agent. User input comes from a text file detailing mission objectives, and output for the user includes a graphical display of robot positions. This level communicates the mission objective to the Coordination Level of all agents. It also simulates proximity sensors and notifies affected agents if a collision is imminent.

The *Coordination Level* of each agent uses a satisficing mechanism to choose an action to achieve the mission. In a three-agent system the mission might be "initialise to formation (X, Y, Z)" where X, Y, and Z are target locations. Once the agent has picked a target, the decision is communicated to the next level with a command of the form "move to Z".

The *Execution Level* translates movement directives into motor commands using a control law and sends voltage commands to the robot via the wireless link. It also merges sensor information from the camera and wheel encoders to determine the actual position of the robot. When running in simulation, a calibrated second-order dynamic model is used to reflect the robot's physical characteristics.

4 Experiments and Results

We begin this section with an example of emergent cooperative behaviour manifest by satisficing agents. Fig. 2 shows initial robot positions, target positions, and the path traced by each robot in one simulation run. In this case, robots are aware of the positions of only their two nearest neighbours and the three nearest targets.

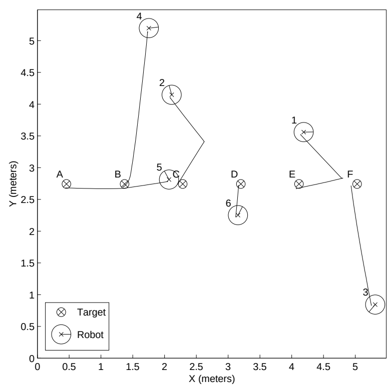

Figure 2: Example scenario: six agents forming a line.

Because robots move directly to selected targets, their choices are evident from their paths. Initially, the choices of all agents are in pairwise conflicts. Agent 2 resolves its conflict with agent 6 for target D when it detects a collision, reevaluates, and picks target C. The resolution of the conflict between agents 4 and 5 is noteworthy: agent 5 moves to B, but when agent 4 approaches, agent 5 leaves B and moves to A instead, since this cooperative solution reduces overall travel distance. A third conflict between agents 1 and 3 is resolved in a manner similar to the first. In the end, all targets are occupied.

4.1 Evaluation Methodology

The results presented in this paper were obtained using simulation, allowing a large number of scenarios to be studied. For each of three goal formations – circles, lines, and randomly placed targets – 250 sets of randomly generated, non-overlapping starting positions were constructed. Reported results are averages over the 250 runs.

Because local information is necessarily incomplete, some decisions made by the robots are suboptimal. At best, this increases the distance the robots travel and the time required; at worst, the robots could fail to realize the formation. The following outcomes are reported for each simulation run:

- **Attainment**. Whether or not the agents were able to complete the formation within fixed time T. For baseline studies, T was 200 seconds, approximately 10 times the interval required to achieve formation in successful runs.
- **Time**. For runs that achieve formation, the total time required.
- **Distance**. For runs that achieve formation, the average distance travelled per agent.
- **Collisions**. The number of actual collisions – those not avoided by the re-decision trigger. To avoid other factors that would influence robot trajectories (and measured outcomes), no other collision avoidance mechanism was used in the baseline model.
- **Decisions**. The number of times each agent executes the procedure to choose a target, including the initial decision.

4.2 Enhancements to Improve Attainment

Fig. 3 shows the fraction of the 250 test runs that achieved attainment using the baseline utility functions. Computational requirements for real-time simulation limited the number of coordinating agents to $m \leq 6$. Because we wanted to model all values of $m \leq n$, this also limited n to 6. Not surprisingly, attainment improves as m increases, since agents have more complete information and can make better decisions.

An analysis of scenarios that did not complete in the allowed time revealed three common problems: agents sometimes repeat the same ineffective decision, the only vacant targets are sometimes well away from the robot, and robots sometimes block each other from reaching goals. To address these, we developed the three specific enhancements described below.

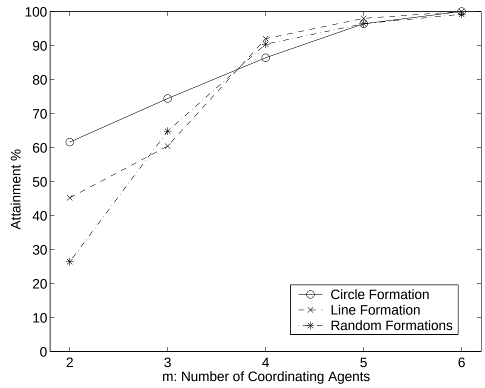

Figure 3: Attainment results for baseline utility functions.

- **Repeat rejection (R)**. A mechanism is added to discourage repeated decisions if unsuccessful: the selectability of each target option is scaled based on the number of times it has been chosen. The scaling factor used is ${\alpha_C}^{x_i}$, where $0 < \alpha_C < 1$ and x_i is the number of times target T_i has been selected as the goal.

- **Target set expansion (T)**. A separate satisficing mechanism is added to determine which m of the n targets will be considered. Each target location $T_i \in F$ is evaluated using the selectability and rejectability in (4) and (5), and the m most satisficing targets are determined. Initially an agent will consider its m nearest targets, but more distant targets will eventually be included; $\delta(T_i)$ is the distance from the agent to target T_i, y_i is the number of times target T_i was previously included, α_T is a constant such that $0 < \alpha_T < 1$, and ϵ is a small constant added to avoid division by zero.

$$P_s(T_i) = \frac{\frac{1}{\delta(T_i)+\epsilon}}{\sum_{j=1}^{n} \frac{1}{\delta(T_j)+\epsilon}} \tag{4}$$

$$P_r(T_i) = \frac{\frac{1}{\alpha_T^{y_i}}}{\sum_{j=1}^{n} \frac{1}{\alpha_T^{y_j}}} \tag{5}$$

- **Path planning (P)**. This enhancement is a simple path planning mechanism employed when an agent's direct path to its goal is obstructed, either by a static obstacle or another agent. Way-points are placed to either side of the straight-line path and adjusted at fixed intervals until a collision-free path is found.

4.3 Convergence

Ideally, initialisation algorithms should achieve desired goal configurations regardless of initial conditions. Because our approach selects goals rather than controlling paths or motion, we assume that control laws exist to allow movement to a goal while avoiding collisions. Thus, once a target is selected, the robot pursues that goal until it is achieved or until a conflict is detected for the same target, in which case the target-selection algorithm is rerun.

Without enhancements, the baseline scheme will not always converge to a solution. (If the only free robot and target are on opposite sides of a large formation, the robot will not perceive the available target and simply cycle between its m nearest targets.) With both repeat rejection and target set expansion, free robots will eventually consider goals beyond the closest m targets, and convergence can be guaranteed. We sketch the proof that all targets eventually become occupied.

First, occupied targets are given up only if another robot is headed for the same target, and only if another free target is nearby. In this case, the original target is occupied by the approaching robot. Now consider any free target during initialisation. With target set expansion, eventually a free robot will pick this target as its goal and approach it. The robot will occupy the target unless another robot gets there first; in either case the target is occupied. Only in the case of a predicted collision between multiple robots moving to the same target will the target selection algorithm be run again before either can occupy the target. Unless the situation is perfectly symmetric for all involved robots (including identical distance to target and history of selecting this target), one of the robots will be allowed to occupy the target. In the unlikely event that both move away from the target (each effectively leaving it to the other), the target will eventually be reconsidered and an agent will succeed in occupying it.

4.4 Simulation Results and Analysis

Results for the circle formation are summarised in Tables 1-4. The top row in each table corresponds to the baseline; subsequent rows correspond to the indicated combination of enhancements. As can be seen, all measures of performance generally improve as the number of coordinating agents increases. (Distance results are similar to those tracking time and not presented.) As expected, coordination is best when agents know the location of all other agents, but this is not always practical.

Table 1: Circle: Attainment %

Options			Number of Coordinating Agents				
R	T	P	2	3	4	5	6
			61.6	74.4	86.4	96.4	100.0
•			96.8	96.4	100.0	100.0	100.0
	•		98.8	100.0	100.0	100.0	100.0
•	•		98.0	99.2	99.6	99.6	100.0
		•	64.4	89.6	92.0	99.6	100.0
•		•	96.8	100.0	100.0	100.0	100.0
	•	•	98.8	100.0	100.0	100.0	100.0
•	•	•	99.2	99.6	99.6	100.0	100.0

Table 2: Circle: Avg. Time (seconds)

Options			Number of Coordinating Agents				
R	T	P	2	3	4	5	6
			18.27	16.86	15.48	14.10	11.76
•			35.08	35.70	18.60	15.23	12.28
	•		42.79	31.16	23.77	15.14	11.76
•	•		43.76	36.55	29.29	17.98	12.28
		•	19.74	29.59	21.13	17.52	13.02
•		•	35.55	30.56	24.57	19.88	13.46
	•	•	39.95	32.05	28.08	18.31	13.02
•	•	•	45.65	34.81	30.72	22.73	13.46

Table 3: Circle: Average Decisions

Options			Number of Coordinating Agents				
R	T	P	2	3	4	5	6
			2.95	2.45	2.17	1.70	1.14
•			7.30	6.22	3.00	2.05	1.26
	•		7.06	4.73	3.47	1.88	1.14
•	•		7.65	6.14	5.05	2.71	1.26
		•	3.01	2.89	1.97	1.53	1.09
•		•	6.73	3.15	2.32	1.79	1.14
	•	•	5.79	2.95	2.37	1.57	1.09
•	•	•	6.41	3.36	2.74	2.02	1.14

Table 4: Circle: Average Collisions

Options			Number of Coordinating Agents				
R	T	P	2	3	4	5	6
			0.50	1.62	0.74	0.60	0.01
•			2.04	10.08	1.63	0.76	0.11
	•		12.61	8.77	3.64	0.92	0.01
•	•		10.04	8.43	4.97	1.23	0.11
		•	0.39	0.39	0.10	0.22	0.02
•		•	1.59	0.89	0.12	0.08	0.01
	•	•	4.61	0.84	0.18	0.10	0.02
•	•	•	3.41	0.62	0.18	0.08	0.01

Simulation results for the line formation are found in Tables 5-6, while Tables 7-8 contain attainment and time results for the random formations. Decision and collision counts are presented only for circle formations; line and random formations exhibit similar tradeoffs.

Table 5: Line: Attainment %

Options R	T	P	Number of Coordinating Agents 2	3	4	5	6
			45.2	60.4	92.0	98.0	100.0
•			94.0	98.8	100.0	100.0	100.0
	•		98.0	97.6	99.6	100.0	100.0
•	•		99.2	99.6	100.0	100.0	100.0
		•	54.0	80.4	99.6	100.0	100.0
•		•	97.6	100.0	100.0	100.0	100.0
	•	•	100.0	99.6	100.0	100.0	100.0
•	•	•	97.6	99.6	100.0	100.0	100.0

Table 6: Line: Avg. Time (seconds)

Options R	T	P	Number of Coordinating Agents 2	3	4	5	6
			17.09	14.22	14.67	13.23	12.72
•			33.05	41.38	15.91	15.33	15.78
	•		24.14	21.23	16.04	13.73	12.72
•	•		24.47	20.53	18.77	17.38	15.78
		•	22.61	27.06	23.59	18.21	15.38
•		•	44.52	35.29	24.59	20.28	16.90
	•	•	27.14	34.91	26.11	19.04	15.38
•	•	•	31.05	36.83	29.56	22.91	16.90

Table 7: Random: Attainment %

Options R	T	P	Number of Coordinating Agents 2	3	4	5	6
			26.4	64.8	90.4	96.4	99.2
•			27.6	74.8	97.2	98.0	99.6
	•		98.4	99.2	99.6	99.6	99.2
•	•		95.2	98.8	99.6	99.2	99.6
		•	33.2	76.8	95.2	97.6	99.6
•		•	42.4	84.8	97.6	99.2	100.0
	•	•	98.0	100.0	100.0	100.0	99.6
•	•	•	93.6	99.2	99.2	100.0	100.0

Table 8: Random: Avg. Time (seconds)

Options R	T	P	Number of Coordinating Agents 2	3	4	5	6
			17.76	17.37	15.93	14.64	13.20
•			19.03	22.96	19.77	17.12	17.50
	•		48.92	31.29	21.47	18.54	13.20
•	•		55.44	35.49	26.56	23.16	17.50
		•	26.19	25.82	21.64	19.33	16.04
•		•	45.68	33.51	24.77	20.97	18.00
	•	•	53.31	33.82	24.39	19.98	16.04
•	•	•	61.40	35.58	27.61	22.97	18.00

A comparison of attainment rates for the baseline scheme reveals that the line formation is more challenging than the circle for a small number of coordinating agents ($m < 4$) despite using the same inter-target spacings. With the circle formation, agents can travel from one target to another (a frequent outcome when a new decision is triggered) without conflicting with robots on other targets in between, as is often the case with the line formation. With four or more coordinating agents, the line actually has a higher attainment rate. If an agent's deliberations are limited to the nearest neighbour, the random scenario is much more difficult than the line or the circle. With more than two coordinating agents, the random scenarios are comparable to line and circle formations in attainment.

Repeat rejection (R) dramatically improves the attainment for small values of m with lines and circles, but it has much less impact on the random formations. The latter are much more likely to contain isolated targets, and repeat rejection can do nothing to help agents consider these targets. Significant increases in attainment are accompanied by significant increases in time to attainment: more runs achieve formation but they take longer on average to do so. As Table 3 shows, large increases in the time to attainment are accompanied by large increases in the average number of decision cycles agents complete.

The positive impact of target set expansion (T) on attainment is even greater than that of repeat rejection. Fig. 4 shows attainment of schemes without path planning averaged over all formations. As can be seen, target expansion has the most dramatic impact on attainment. Especially noteworthy (from Table 7) is its improvement on the attainment of random formations: over 98% of the runs achieve formation for $m > 2$. This improvement is the result of agents eventually including outlying, unoccupied targets in their decision spaces. Where the attainment improves the most, the average time to attainment also increases significantly; it takes some time for the agents to include the most remote targets in their deliberations.

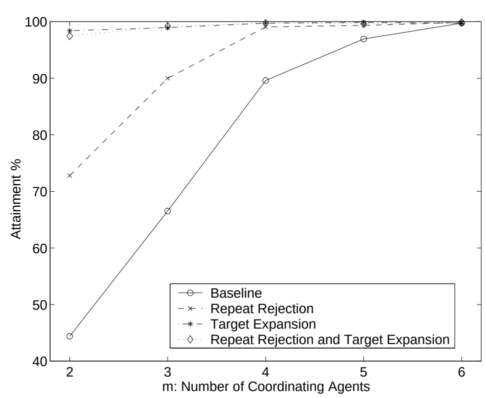

Figure 4: Avg. attainment: all formations

The combination of target set expansion and repeat rejection does little to improve performance relative to repeat rejection alone. In some cases, the combination is worse than repeat rejection by itself. In dynamic environments, simultaneously altering both the decision space and the utility functions results in somewhat chaotic behaviour.

When path planning (P) is added, the principal benefit is a noteworthy reduction in actual collisions (see Table 4), but attainment generally increases as well. As the attainment increases, so does the time required since the alternate paths are longer than straight paths to goal nodes. On average, there is a slight decrease in the number of decision cycles executed by each agent to complete the formation, a natural consequence of reducing collisions that trigger re-decision.

4.5 Formation Initialisation with Static Obstacles

Another 250 scenarios were created with a static obstacle near each target in a line formation. Obstacles were identical to robots in shape and size (circular, 1 foot diameter), and each was placed at a random angle and fixed distance (1.5 feet) from its companion target. Robots are able to sense obstacles within a five foot radius. Results are shown in Tables 9-10.

Table 9: Obstacles: Attainment %

Options			Number of Coordinating Agents				
R	T	P	2	3	4	5	6
			3.6	5.6	7.6	8.4	11.2
•			7.2	17.6	21.2	28.0	32.0
	•		13.2	13.6	13.6	12.4	11.2
•	•		13.2	17.6	22.8	26.0	32.0
		•	28.4	52.4	58.8	59.6	63.2
•		•	50.8	81.6	90.8	96.0	95.6
	•	•	77.6	75.2	79.6	68.4	63.2
•	•	•	78.4	81.2	92.0	93.2	95.6

Table 10: Obstacles: Avg. Time (seconds)

Options			Number of Coordinating Agents				
R	T	P	2	3	4	5	6
			14.08	24.61	14.15	19.57	18.17
•			25.44	46.63	39.00	52.81	53.75
	•		47.44	49.44	56.23	43.85	18.17
•	•		39.28	43.27	52.42	49.72	53.75
		•	39.36	27.79	24.61	21.46	19.23
•		•	65.52	46.94	47.22	43.43	44.01
	•	•	67.09	58.97	49.93	29.76	19.23
•	•	•	65.59	64.00	50.47	57.33	44.01

Without path planning, attainment is poor. Neither repeat rejection nor target set expansion consider obstacles. With path planning, attainment increases markedly. For $m = 2$, 78% attainment is reached with both target set expansion and repeat rejection enabled, and attainment is $\geq 90\%$ with repeat rejection for $m \geq 4$. Consistent with previous results, the increase in attainment is accompanied by an increase in time to achieve formation.

Contrary to performance trends without obstacles, attainment with target set expansion tends to *decrease* as m increases. For small values of m, agents stuck behind obstacles are likely to eventually consider other targets, potentially leading to a decision that reroutes them around the obstacle. When $m = n$, the decision space is not affected by target set expansion, so agents stuck behind obstacles can be stuck indefinitely.

4.6 Formation Initialisation with Many Robots

In our final experiments, robots initialised to square grid formations of to 100 targets. For each grid size, 50 starting scenarios were generated randomly. In addition to measurements listed in Section 4.1, we recorded the percentage of agents on targets over time, amounting to the rate of attainment for each run. To begin, we evaluated all initialisation schemes for the largest (10×10) grid, then simulated only the most promising over the smaller grid sizes. Fig. 5 shows the fraction of targets occupied by robots over time (averaged over 50 runs) of the indicated three schemes, each with path planning and using six coordinating agents ($m = 6$).

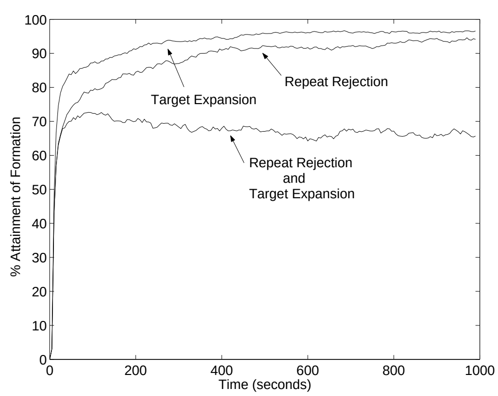

Figure 5: Attainment over time for 10×10 grid formations using path planning.

For all time values, target set expansion results in the greatest percentage of targets covered, but repeat rejection also approaches full coverage. With both target expansion and repeat rejection, coverage peaks at about 70% and then degrades slowly. With 30 to 35 agents constantly in motion, the effects of changing both the decision space *and* the utility functions cancel each other out and somewhat chaotic behaviour results.

The combination of path planning and target set expansion was simulated over a wide range of grid sizes, producing the results in Table 11. For $q \times q$ grid formations of moderate size ($3 \leq q \leq 5$), all of the tested scenarios achieved formation within the allotted time. As q increases from 6 to 10, attainment rates decline significantly. As Fig. 5 suggests, however, even when the entire run does not succeed in covering all targets, the vast majority of targets will be covered by a robot.

Table 11: Grid Formation Results

	Size of Grid Formation							
Data	3x3	4x4	5x5	6x6	7x7	8x8	9x9	10x10
Attainment	100%	100%	100%	92%	78%	74%	44%	42%
Time (sec)	27.2	69.3	179.5	259.9	368.6	419.8	500.4	558.4
Collisions	0.56	1.17	1.10	1.73	1.84	2.05	1.93	1.93
Decisions	3.44	5.18	8.61	10.41	11.09	11.31	12.01	12.00

5 Summary and Conclusions

This paper has discussed the development and implementation of distributed algorithms for formation initialisation. Most importantly, we have presented effective solutions to this problem that are based on satisficing theory, a novel decision making paradigm. Using techniques based on utility functions, groups of agents can successfully initialise into a given formation, even with restricted knowledge of the environment. With the addition of various extensions to the baseline utility functions, which allow the agents to modify their behaviour based on past decisions, even large systems of agents can successfully achieve specific formations. We conclude that satisficing theory offers an attractive alternative for the synthesis of cooperative multi-agent systems.

References

[1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, “Distributed memoryless point convergence algorithm for mobile robots with limited visibility,” *IEEE Trans. on Robotics and Automation*, vol. 15, pp. 818–828, October 1999.

[2] T. B. Gold, J. K. Archibald, and R. L. Frost, “A utility approach to multi-agent coordination,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, (San Francisco, CA), pp. 2052–2057, April 2000.

[3] P. K. C. Wang, “Navigation strategies for multiple autonomous mobile robots moving in formation,” *Journal of Robotic Systems*, vol. 8, no. 2, pp. 177–195, 1991.

[4] J. P. Desai, J. Ostrowski, and V. Kumar, “Controlling formations of multiple mobile robots,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, (Leuven, Belgium), pp. 2864–2869, May 1998.

[5] M. Lemay, F. Michaud, D. Létourneau, and J.-M. Valin, “Autonomous initialization of robot formations,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, (New Orleans, LA), pp. 3018–3023, April 2004.

[6] T. Sugar and V. Kumar, “Decentralized control of cooperating mobile manipulators,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, (Leuven, Belgium), pp. 2916–2921, May 1998.

[7] J. Fredslund and M. J. Matarić, “A general algorithm for robot formations using local sensing and minimal communication,” *IEEE Trans. on Robotics and Automation*, vol. 18, pp. 837–846, October 2002.

[8] T. Balch and R. C. Arkin, “Behavior-based formation control for multi-robot teams,” *IEEE Trans. on Robotics and Automation*, vol. 14, pp. 1–15, December 1998.

[9] X. Yun, G. Alptekin, and O. Albayrak, “Line and circle formation of distributed physical mobile robots,” *Journal of Robotic Systems*, vol. 14, no. 2, pp. 63–76, 1997.

[10] Q. Chen and J. Y. S. Luh, “Coordination and control of a group of small mobile robots,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 2315–2320, 1994.

[11] H. Yamaguchi, “Adaptive formation control for distributed autonomous mobile robot groups,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, April 1997.

[12] K.-H. Tan and M. A. Lewis, “Virtual structures for high-precision cooperative mobile robot control,” *Autonomous Robots*, vol. 4, pp. 387–403, October 1997.

[13] R. W. Beard and F. Y. Hadaegh, “Constellation templates: An approach to autonomous formation flying,” in *World Automation Congress*, (Anchorage, Alaska), pp. 177.7–177.6, May 1998. ISIAC.

[14] M. Cieliebak and G. Prencipe, "Gathering autonomous mobile robots," in *Proc. Int. Colloquium on Structural Information and Communication Complexity*, pp. 57–72, 2002.

[15] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 35, pp. 78–92, January 2005.

[16] R. A. Freeman, P. Yang, and K. M. Lynch, "Distributed estimation and control of swarm formation statistics," in *Proc. of the American Control Conference*, (Minneapolis, MN), pp. 749–755, June 2006.

[17] W. C. Stirling, *Satisficing Games and Decision Making: With Applications to Engineering and Computer Science*. Cambridge, UK: Cambridge Univ. Press, 2003.

[18] W. C. Stirling, "Social utility functions - part I: theory," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, pp. 522–532, November 2005.

[19] W. C. Stirling and R. L. Frost, "Social utility functions - part II: applications," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, pp. 533–543, November 2005.

[20] F. R. Johnson, J. C. Hill, J. K. Archibald, R. L. Frost, and W. C. Stirling, "A satisficing approach to free flight," in *Proc. of Int. Conf. on Networking, Sensing and Control, 2005*, pp. 123–128, IEEE, March 2005.

[21] J. C. Hill, F. R. Johnson, J. K. Archibald, R. L. Frost, and W. C. Stirling, "A cooperative multi-agent approach to free flight," in *AAMAS '05: Proc. of 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pp. 1083–1090, ACM Press, 2005.

[22] J. C. Hill, J. K. Archibald, W. C. Stirling, and R. L. Frost, "A multi-agent system architecture for distributed air traffic control," in *AIAA Guidance, Navigation and Control Conference*, (San Francisco, CA), August 2005. AIAA-2005-6049.

[23] J. C. Hill, M. S. Nokleby, J. K. Archibald, R. L. Frost, and W. C. Stirling, "Cooperative graph search by a system of autonomous agents," in *Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics*, (Waikoloa, HI), pp. 121–126, October 2005.

Author Biographies

James K. Archibald received the B.S. degree in mathematics from Brigham Young University in 1981, and the M.S. and Ph.D. degrees in computer science from the University of Washington in 1983 and 1987 respectively. Since 1987 he has been a faculty member in the Department of Electrical and Computer Engineering at Brigham Young University. He was a visiting professor at the Swiss Federal Institute of Technology in Zurich in 1994-95. His research interests include robotics, autonomous vehicles, and multi-agent systems.

Richard L. Frost received the B.S. degree (Hons.) in physics in 1975, and the M.S.E.E. and Ph.D. degrees from the University of Utah, Salt Lake City, in 1977 and 1979, respectively. He joined the Massachusetts Institute of Technology's Lincoln Laboratory, Lexington, in 1979, then returned to the faculty of the University of Utah from 1981 to 1984. After three years with Sperry Corporation, he joined the Department of Electrical and Computer Engineering, Brigham Young University, in 1987, where he is currently an Associate Professor. His research interests include information theory, especially source coding and quantisation, signal processing, and multi-agent decision theory.