



The Dynamic Selection of Coordination Mechanisms

CORA BEATRIZ EXCELENTE-TOLEDO¹

cora@lania.mx

National Laboratory of Advanced Computer Science, LANIA A.C., Rébsamen No. 80, C.P. 91090 Xalapa, Veracruz, México

NICHOLAS R. JENNINGS

nrj@ecs.soton.ac.uk

IAM Group, School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

Abstract. This paper presents and evaluates a decision making framework that enables autonomous agents to dynamically select the mechanism they employ in order to coordinate their inter-related activities. Adopting this framework means coordination mechanisms move from the realm of something that is imposed upon the system at design time, to something that the agents select at run-time in order to fit their prevailing circumstances and their current coordination needs. Using this framework, agents make informed choices about when and how to coordinate and when to respond to requests for coordination. The framework is empirically evaluated, in a grid world scenario, and we highlight those types of environments in which it is effective.

Keywords: multiagent systems, coordination, dynamic reasoning, decision making, coordination mechanisms.

1. Introduction

Effective coordination is essential if autonomous agents are to achieve their goals in a multiagent system (MAS). Such coordination is required to manage the various forms of dependency that naturally occur when the agents have inter-linked objectives, when they share a common environment, or when there are shared resources. To this end, a variety of protocols and structures have been developed to address the coordination problem. These range from long-term social laws [37], through medium term mechanisms such as Partial Global Planning [14], organizational structuring [19] and market protocols [32], to one-shot (short-term) mechanisms like the Contract-Net Protocol [38].

All of these *coordination mechanisms* have different properties and characteristics and are suited to different types of tasks and environments. They vary in the degree to which coordination is prescribed at design time, the amount of time and effort they require to setup a given coordination episode at run-time, and the degree to which they are likely to be successful and produce coordinated behaviour in a given situation. In the majority of cases, these dimensions act as forces in opposing directions; coordination mechanisms that are highly likely to succeed typically have high setup and maintenance costs, whereas mechanisms that have lower setup costs are also more likely to fail. Moreover, a coordination mechanism that works well in a reasonably static environment will often perform poorly in a dynamic and fast changing one. In short, there is no universally best coordination mechanism [20].

Given this situation, we believe it is important for the agents to have a variety of coordination mechanisms, with varying properties, at their disposal so that they can then select a mechanism that is appropriate to the coordination episode at hand. Thus, for particularly important tasks, the agents may choose to adopt a coordination mechanism that is highly likely to succeed, but which will invariably have a correspondingly large setup cost. Whereas for less important tasks, a mechanism that is less likely to succeed, but which has lower setup costs, may be more appropriate. Similarly, when it is difficult or expensive to setup a coordination activity (e.g. because there are not many agents available) is it appropriate to pick a mechanism that is highly likely to succeed (even though it will have a high setup cost). In contrast, if coordination can be setup easily or cheaply then it might be more important to choose a mechanism that promises to achieve the task in the quickest possible way. However, to date, the choice of which coordination mechanism to use in a given situation is something that the designer typically imposes upon the system at design time (e.g., in a given application a particular social law will be used or it will be decided that all coordination activities will be handled by the contract net protocol). This means that in many cases the coordination mechanism that is employed is not well suited to the agents' prevailing circumstances. This inflexibility means that the performance of both individual agents and the overall system may be compromised.

To rectify this situation, our aim in this research is to develop agents that can reason about the process of coordination and then select mechanisms that are appropriate to their current situation. That is, *the choice of coordination mechanism is made at run-time by the agents that need to coordinate*. Thus, we claim that fixing on a single coordination mechanism at design time is inappropriate, because there is no scope for changing or modifying the mechanism to ensure there is a good fit with the prevailing circumstances [2]. We believe this problem is particularly true in open and dynamic environments, because of the wide variety of contexts in which coordination is likely to be needed.

To circumvent this problem, we will present a decision making framework that enables agents to dynamically select the coordination mechanism that is most appropriate to their circumstances. Thus, our work is *not* concerned with developing actual coordination mechanisms that exhibit the varying properties discussed above, nor with classifying existing coordination mechanisms along such dimensions. Rather, we view the coordination mechanisms in the abstract; representing them in a quantitative way. In particular, we model the cost involved in the coordination and the benefits of adopting a particular mechanism (as advocated by Lesser [29]). Our framework is then concerned with deciding whether coordination should be attempted in a given context (given the agent has a particular set of mechanisms at its disposal) and, if so, which of the available mechanisms is the most appropriate to employ in the prevailing situation. Once a particular mechanism has been chosen for a particular coordination episode, the agents involved are expected to adhere to the rules and procedures specified in the mechanism itself. Thus the rules indicate how the coordination should actually proceed.

This work advances the state of the art in the following ways. Firstly, the very idea of letting the agents dynamically select their coordination mechanism has not

explicitly been addressed within the field of multi-agent systems to date. Although some work has been done on flexible reasoning about coordination and the dynamic selection of algorithms, the focus on the reasoning about coordination strategies is new. Secondly, we present a formal framework for capturing the reasoning processes the agents require in order to perform in such a manner (this covers making decisions about when to coordinate and which coordination mechanism to use). Finally, we provide a systematic empirical evaluation of the efficacy of such reasoning in an idealized coordination scenario.

The remainder of this paper is structured as follows. Section 2 details our specific coordination scenario. Section 3 formalises the decision procedures of the agents. Section 4 evaluates the decision making framework. Section 5 deals with related work and, finally, Section 6 concludes and presents the areas of further work.

2. The coordination testbed

The domain in which we perform our evaluations takes the form of a grid world in which a number of autonomous *agents* (A_i) perform tasks for which they receive units of *reward* (R_i). Each agent has a *specific task* (ST_i) which only it can perform; there are other tasks that require several agents to perform them, called *cooperative tasks* (CTs). Each task has a reward associated with it, the rewards for the CTs are higher than those for STs since they must be divided among the coordinating agents.

The agents move around the grid one step at a time, up, down, left or right, or stay still. At any one time, each agent has a single *goal*, either its ST or a CT over which coordination may need to be achieved. On arrival at a square containing its goal, the agent receives the associated reward. In the case of STs, a new one appears, randomly, somewhere in the grid, visible only to the appropriate agent. In the case of CTs, a new one appears, randomly, somewhere in the grid, but this is only visible to an agent who subsequently arrives at that square. If an agent encounters a CT, while pursuing its current goal (i.e., its ST), it takes charge of the CT² and must decide on both whether to initiate coordination with other agents over this task, and which coordination mechanism (CM) it should use. In this context, each agent has a pre-defined range of CMs at its disposal. Each CM is parameterised by two key attributes: setup cost (in terms of time-steps) and chance of success (see Section 2.1 for more details). For example, a CM may take t time-steps to setup (modelled by the agent waiting that number of time-steps before requesting bids from other agents) and have a probability, p , of success (thus when the other agent(s) arrive at the CT square, the reward will be allocated with probability p , with zero reward otherwise). An agent may well decide that attempting to coordinate is not a viable option, in which case it adopts the null CM (i.e. the agent rejects adopting the CT as its goal).

The Agent-in-Charge (AiC) of the coordination selects a CM and, after waiting for the setup period, broadcasts a request for other agents to engage in coordination. The other agents respond with bids composed of the amount of reward they would require in order to participate in the CT and how many time-steps away from the CT square they are situated. If an agent's bid is successful, then it is termed Agent in-Cooperation (AiCoop) to denote the fact that it is a participant (not AiC) for a CT

- [1] Agents arrive at a square. If AiS arrives at its ST cell, its goal is attained, it receives the reward and updates its goal. If AiCoop arrives at the CT cell, it notifies the AiC that it has arrived. The CT is achieved and the rewards are paid to AiCoops.
- [2] If AiS finds a CT it must decide if it wants to become AiC and, if so, which $CM = (t, p)$ it should use. If $t > 0$ it must wait t time-steps before broadcasting a request for coordination. If AiC finds a new CT, it ignores it.
- [3] If AiS receives a request for coordination, it decides whether and what to bid to participate in the CT. The AiC then evaluates all bids. If AiS's bid is accepted, it adopts CT as its new goal. AiC does not respond to requests for coordination.
- [4] Each agent decides on its next move according to its current goal and all agents move simultaneously.

Figure 1. Basic protocol followed by agents.

task. The role Agent-in-ST (AiS) is used to denote the situation where an agent is working towards a ST. Figure 1 highlights the specific decisions which have to be made (see Section 3 for more detail) and gives the protocol the agents follow at each time-step.

Agents might receive more than one proposal at the same time-step, in which case they reply with as many bids as the proposals they receive. However, they will only accept one CT contract at a time. Agreements between AiCs and AiCoops to achieve a particular CT are established via a contracting protocol. This contract-net-like protocol consists of three steps. In the first step, AiC broadcasts a proposal to all agents. It then waits for the bids. The second step involves selecting the bids and contracts from AiCs and AiCoops respectively (both of them have to consider refusals and denials of their corresponding offers). Finally, the third step consists of the commitment about the terms of the contract and the time-step at which AiCoops will arrive at the CT square.

In choosing a scenario in which to evaluate our model we are faced with one of the perennial problems of empirical research (see [9]) for fuller discussion): do we use a concrete real world domain or do we work in an abstract environment? Choosing the former means concerns are raised about the generality of the results. Choosing the latter means there are concerns about the applicability of the developed models or the simplicity of the scenario (see [23] for a discussion of the relative merits of such a choice). Our choice, a grid world scenario, obviously falls into the abstract environment category. This decision was made because the primary objective is to focus on the essential aspects of an agent's decision making about coordination and it was felt that this is best achieved without the extraneous constraints of a real-world application. Nevertheless, it is believed that the scenario models the key features related to making coordination decisions that are present in many real world scenarios and that it incorporates the necessary degree of dynamism and uncertainty to fully evaluate the coordination model. To this end, the scenario has

been deliberately setup to concentrate on the decision making involved in coordination. Thus we minimize the remainder of the agent's decision making capabilities so that differences in performance are *solely* attributable to decisions about coordination and not anything else. For example, we say that an agent can only pursue one goal at a time (meaning our results are not influenced by how effectively an agent can interleave execution of multiple concurrent goals) and that agents cannot renege on commitments (meaning we do not have to reason about commitment strategies and types of sanctions). While adding such functionality would inevitably improve the agents' performance, we believe it may also make the effect of the coordination decisions more difficult to determine. Moreover, the parameterized behaviour of the environment means that the experimental conditions can be fully reproduced in order to allow meaningful comparison between the different coordination techniques.

2.1. *The coordination mechanism abstraction*

The first step in characterising coordination mechanisms is to represent them in a common manner so that they can easily be compared. To this end, we adopt a characterization that covers the following issues: what are the requirements that need to be fulfilled before the technique can be applied (*requirements*)?, what are the rules that agents follow in order to complete their interactions (*coordinating algorithm*)?, and finally, what degree of coordination is likely after the coordinating algorithm has been applied (*level of achievement*)?. Now, to be able to reason about selecting a particular mechanism in a given situation we need to abstract from this basic description some meta information that enables us to distinguish between the alternatives. In this case, we consider the run-time cost to setup the mechanism and the likelihood of it being successful³ (as advocated in [29]). These meta data attributes can be regarded, respectively, as being derived from the description of the requirements and the level of achievement. Moreover, we need to find a simple way of quantifying these values and for illustrative purposes in this section we choose a simple qualitative scale of high, medium and low. Combining this altogether, we produce the generic template of Table 1 for describing and reasoning about coordination mechanisms.

To illustrate the use of our template we show how probably the most common coordination mechanism, the Contract-Net protocol [38], can be represented (characterizations of several others coordination mechanisms are given in [16]. In this mechanism, a contractor agent (the task manager) is responsible for achieving a given goal. It has to decompose this goal into subtasks and indicate to other agents that it requires assistance with some of these subtasks.⁴ The task manager broadcasts a task announcement to all the other agents in the system (along with any special criteria that the bidders must fulfil). Each recipient then decides whether it is interested in taking on the task and, if so, it returns a bid indicating the conditions under which it is willing to undertake the task. Finally, following the expiration of the task announcement, the task manager evaluates the bids (task deliberation) and awards the task to the most appropriate agent. Thus, a contract

Table 1. Generic template of a CM.

Coordination technique components

Requirements. The preconditions that need to be satisfied before the coordinating algorithm can be executed. This covers things such as whether the protocol requires a setup phase (e.g., in multiagent planning a goal decomposition is needed before sub-goals can be assigned to the agents) or whether it needs a particular piece of information to be available (e.g. the number of agents in the system). These requirements might be established at design time and/or during the run-time execution. Examples of the former case are when the agent knows (because it has been hard-wired) how to contact the other agents in the system, what are their corresponding capabilities and so on. Examples of the latter case are when agents acquire the same information, but, as a result of their interaction with others.

Coordinating algorithm. The detailed plan of actions that have to be followed to achieve coordination once the requirements have been fulfilled.

Level of achievement. The degree of coordination that is likely from following the steps specified in the coordinating algorithm.

Meta-data attributes

Cost to setup: A value measuring the run-time costs associated with the requirements noted above.

Probability of success: The likelihood in a given environment that following the steps in the coordinating algorithm will result in a successful coordination.

is established (task allocation) between the manager and the contractors with the winning proposals. From this description, Table 2 shows our representation of this mechanism.

Table 2. Contract-Net Protocol CM.

Coordination technique components

Requirements. To apply the protocol the agents need to have information about how to contact one another (the identification of the possible receivers of the offer) and how to rank incoming bids from potential contractors (the selecting criteria). The information about contractors can be given at design time (in static environments) or determined at run-time (in more dynamic cases).

Coordinating algorithm. The protocol consists of three phases: identifying potential contractors, making a decision about which contractor to select and actually enacting the agreed task. These phases are based on the message interchanges associated with the sending out of a request by the task manager, the handling of the bids from the potential contractees and the contract assignment respectively.

Level of achievement. There are several reasons why, once selected, the protocol may fail to result in a successful coordination. For example, the manager may receive no bids from potential contractees (because they are too busy and unavailable or because they are not interested in the task at hand) or the bids received may not satisfy the manager's requirements, or the chosen contractee may not accept the awarded contract.

Meta-data attributes

Cost to setup: The main setup cost is associated with awarding the contract. This is dependent on how much time is required to determine the set of potential contractors to send the request to (if this is inbuilt the cost is low, if it needs to be determined at run-time it will be more time consuming because it may involve interacting with a broker [12] and the time the agent has to wait for responses before it can make choices (the task announcement expiration-period).

Probability of success: Because of the many eventualities that might occur, the mechanism has a medium likelihood of succeeding in the coordination tasks in the general case. If further information is available about the specifics of the environment (e.g., many agents can provide various subtasks or the agents are generally cooperative and will offer help whenever possible) then this qualification can be refined.

3. The agent's decision making procedures

This section formalises the decision procedures of the agents. To study the average impact of coordination mechanisms, an infinite horizon model of decision making [18] was adopted because we are interested in the long-term performance of agents; a finite horizon model may lead to erratic behaviour as the last time-step approaches. However, there are still two ways to model the agents' decisions: by using average reward per unit time or by discounting future rewards. Here, the former was chosen, since it simplifies the decision analysis.

The agents' aims are to maximize their reward, in particular their average reward per unit time. Each agent keeps track of its own average reward, termed its *reward rate*, and it uses this rate to decide how much to charge for its own services and occasionally to approximate the expected rates of other agents (when it is not able to build up a picture of them). Specifically, each agent uses its reward rate to evaluate and compare the different actions available to it; if it can maintain or improve this rate, it chooses to do so. Of course, this decision model approximates the true relative values of different actions, however, since the environment may change rapidly, we believe that a simple reactive decision procedure is appropriate (see [28]).

Agents may have various dispositions with respect to cooperation and the characterization of sociality used here is captured by an agent's *willingness to cooperate* (WtC) factor (based on Hogg and Jennings [24]). This factor, ω , represents the weight an agent puts on opting to cooperate, relative to collecting its usual reward. When reward units, effectively the agent's utility, are of equal currency, a *neutral* agent ($\omega = 1$) only needs to receive the same reward from a CT as it would from its ST. Thus, if $\omega > 1$ it can be described as *greedy*, asking for more reward than it would normally expect to receive and if $\omega < 1$ it can be described as *self-less*, asking for less than it would normally expect to receive. The decision procedures described in this section will typically assume that agents are neutral, but will include ω to indicate where this factor comes into the calculations.

In our model there are four types of decisions that agents are required to make: (i) the direction to move in; (ii) which CM to adopt, if any; (iii) how much to bid when a request for coordination is received; and (iv) how to determine which bid to accept, if any. Each of these is now dealt with in turn.

3.1. Deciding the direction to move

An agent always has a target square in which its current goal is located. The agent decides to move towards its goal by selecting the direction, up, down, left, or right, probabilistically according to the ratio of up/down to left/right squares away from the goal it is. Formally, if the agent is at square $[x1, y1]$ and its goal is at $[x2, y2]$, the probabilities (pmove) that it will move in any given direction are given by:

$$\text{pmove}(up/down) = \frac{|y1 - y2|}{|x1 - x2| + |y1 - y2|}$$

(up if $y2 < y1$, down otherwise)

$$\text{pmove}(\text{left/right}) = \frac{|x1 - x2|}{|x1 - x2| + |y1 - y2|}$$

(left if $x2 < x1$, right otherwise)

For example, if an agent's ST is located 4 squares up and 3 squares to its right, then with Probability 4/7 it will move up, and with probability 3/7 it will move right.

3.2. Deciding which CM to select

An agent which, while pursuing its current goal, encounters a CT must decide whether to initiate coordination with other agents in order to perform it. To do this, the agent must determine whether there is any advantage in so doing. This depends not only on the reward that is being offered, but also on the CMs available, as well as on various environmental factors that affect the expected demands of the potential coordinating agents.

To model the *expected demands* of the other agents, the AiC assumes they are randomly distributed throughout the grid, and that their current goals are similarly distributed. Thus some agents may be near the CT while others may be far away; likewise, for some agents there would be a significant deviation from their ST to reach the CT, while others may be able to coordinate over the CT enroute to their own goals. The agent then assesses the possible CMs on the basis of how long before the task can be performed and how much reward it is likely to obtain after deducting the expected reward requirement of the other agents. In the former case, it considers both the setup time and the average distance away each agent is situated, whereas the latter value is based on the amount of time the agents must spend deviating from their path and the CM's probability of success. This assessment determines the amount of surplus reward the agent can expect, over and above what it expects to obtain during its normal course of operation, i.e., its own average reward per time-step, r . The agent then selects the CM that maximizes this surplus⁵.

To formalize this decision procedure, consider an $M \times N$ grid with reward size S for STs, and R for CTs, a coordination mechanism, $\text{CM} = (t, p)$, that costs t time-steps to setup and has a probability of success p . In this grid world of known size, the agent can calculate the expected average distance (*avedist*) away of any randomly situated agent from the CT square as well as the likely average deviation (*avedev*) such agents would have to make to get there.

First, the average distance in each direction of a random square from a point $[x, y]$ is given by:

$$\text{x.distance}(x) = \frac{2x(x-1) + M(M+1-2x)}{2M}$$

$$y_distance(y) = \frac{2y(y-1) + N(N+1-2y)}{2N}$$

Hence the `ave_dist` of any given agent from $[x, y]$ is:

$$ave_dist(x, y) = x_distance(x) + y_distance(y)$$

The average distance, `ave_dist`, of an agent from its ST is the average distance between two random points on the grid. This is given by averaging `ave_dist(x, y)` over all x and y :

$$ave_dist = \frac{\sum_{x=1}^M \sum_{y=1}^N ave_dist(x, y)}{M \times N}$$

Finally, the average deviation of an agent to assist in a CT at square $[x, y]$ and then go on to its ST, as compared with going straight to its ST, is given by:

$$ave_dev(x, y) = 2 \times ave_dist(x, y) - ave_dist$$

Based on these figures, the agent can assess the average surplus reward from coordinating over the CT at (x, y) using $CM_j = (t_j, p_j)$. First, it must estimate its own cost in terms of how long the CM will take to setup and how long it expects to wait for the other agents to arrive. Since the AiC would usually expect to receive r reward units per time-step ($r = \frac{S}{ave_dist}$), the cost of CM_j is given by:

$$cost_j(x, y) = r \times (t_j + ave_dist(x, y))$$

Second, the AiC must estimate the average amount of reward the other m agents will require based on their willingness to cooperate (ω). To distinguish an agent's own average reward (r) from that of the others, r_AiCoop is used to refer to the average reward of all the other agents in the environment. When AiC does not have any knowledge of r_AiCoop it uses its own average reward as an approximation (see [17] for further discussion of this matter).

$$ave_bid_j(x, y) = \frac{r_AiCoop \times \omega \times ave_dev(x, y)}{p_j} \quad (1)$$

Third, the AiC estimates the expected surplus (`ave_payoff`) of CM_j from adopting the CT by taking into account the probability of success of the task (represented with the reward R associated to the CT):

$$ave_payoff_j(x, y) = p_j \times R \quad (2)$$

$y \downarrow$					
1	AiS ₂				
2			CT		
3					AiS ₁
4		ST ₁			
5				ST ₂	
$x \rightarrow$	1	2	3	4	5

Figure 2. Example of a coordination world grid.

Using these estimates, the AiC can evaluate the expected surplus reward of adopting CM_{*j*}:⁶

$$\begin{aligned} \text{ave_surplus}_j(x, y) = & \text{ave_payoff}_j(x, y) - (\text{cost}_j(x, y) \\ & + (m \times \text{ave_bid}_j(x, y))) \end{aligned} \quad (3)$$

When deciding which of its CMs to adopt, the agent computes its expected surplus reward from each of them and selects the one that maximizes this value. If the surplus associated with all CMs is negative, the agent adopts the option of the null CM (which is defined to have zero surplus).

To exemplify this decision procedure, consider the simple scenario of Figure 2 at one instant in time with two agents (AiS₁ and AiS₂), two STs, one CT and two CMs: CM₁(3, 0.9) and CM₂(6, 1.0). AiS₂ occupies a 5 × 5 grid and finds a CT requiring one other agent with $R = 6$ at square [3, 2]. Assume all agents have a WtC factor of $\omega = 1$. The average distance of other agents from [3, 2] is 2.6. Since the average distance between two random squares is 3.2, the average deviation of any agent from [3, 2] is 2. Assume that each ST has a reward $S = 2$, then the average reward per time-step of all agents is $\frac{2}{3.2} = 0.625$. The expected surplus reward of adopting each CM is given by:

$$\begin{aligned} \text{cost}_1(3, 2) &= (0.625 \times (3 + 2.6)) = 3.5 \\ \text{ave_bid}_1(3, 2) &= \frac{(0.625 \times (1 \times 2))}{0.9} = 1.389 \\ \text{ave_payoff}_1(3, 2) &= (0.9 \times 6) = 5.4 \\ \text{ave_surplus}_1(3, 2) &= 0.511 \\ \\ \text{cost}_2(3, 2) &= (0.625 \times (6 + 2.6)) = 5.375 \\ \text{ave_bid}_2(3, 2) &= \frac{(0.625 \times (1 \times 2))}{1.0} = 1.25 \\ \text{ave_payoff}_2(3, 2) &= (1.0 \times 6) = 6 \\ \text{ave_surplus}_2(3, 2) &= -0.625 \end{aligned}$$

Under these circumstances, AiS₂ decides to attempt coordination with CM₁ (becoming AiC) because it expects to obtain a profit. Note this is not the case with

CM₂, where the negative result indicates there is not likely to be a surplus. Thus, in this case, if AiS₂ only had CM₂ at its disposal it would choose the null CM (expected surplus zero) and it would continue towards its ST.

3.3. Deciding what to bid to become an AiCoop

When agents receive a request to participate in a CT they submit a bid based on the amount of reward that they would require to compensate them for deviating from their current goal. They also submit their current distance away from the CT square. Thus, an agent's required reward is determined by the amount of time spent in deviating to the CT square, its average reward per time-step and the probability of success of the CM being proposed.⁷

To formalise this, consider an agent, A_i, with ω_i and average reward per time-step r_i . The agent calculates its deviation, i.e., the number of extra time-steps it requires to reach its ST if it goes via the CT square. Note that if, for example, the CT square lies directly on a path to the ST, the agent's deviation would be zero. Clearly, such an agent will be in a position to submit a very attractive bid, since the cost of coordinating is effectively zero.

Again by means of illustration consider the agents depicted in Figure 2. AiS₁ at [5, 3] would take 4 time-steps to reach ST₁ at [2, 4] directly, but 6 steps going via the CT at [3, 2], a deviation of 2 timesteps. However, AiS₂ at [1, 1] would take 7 time-steps to reach ST₂ at [4, 5] directly, and also 7 steps going via the CT at [3, 2]; AiS₂ therefore has a deviation of 0.

To compute the reward A_i requires from engaging in coordination over the CT, it takes into account the compensation both for its deviation and for the possibility that the CM might fail; it also takes into account its willingness to cooperate. Thus, we have:

$$\text{bid}_{ij} = \frac{r_i \times \omega_i \times \text{deviation}_i}{p_j} \quad (4)$$

The agent submits its bid to coordinate and its distance from the CT square. If an agent is selected to coordinate, it adopts the CT as its current goal. Its ST is only re-adopted after the CT has been accomplished.

3.4. Deciding which AiCoop bids to accept

Once the AiC has received bids from all agents, it selects the set that maximizes its surplus reward, given the new (definite) information it has received (cf. the approximation in Section 3.2). For each agent, A_i, the AiC knows the amount of reward it will require (bid_{ij}) and the time it will take to arrive (T_i).

The AiC's selection bid process is based on the calculation of the cost of each bid received. However, when more than two agents are required to achieve a CT, it is

necessary to deal with the fact that an AiCoop may have to wait in the CT cell while the remaining AiCoops arrive (because agents have to travel different distances). There are many ways of dealing with this situation (see discussion below). However to simplify the estimates of expected reward undertaken by the various agents, we assume the AiC pays an additional reward for the time elapsed. Thus, AiC knows the number of time-steps that each AiCoop is likely to have to wait (specified in the bid) and the amount it will pay for waiting time at a specific predefined waiting rate (q). The CT is achieved only when the AiC has received the confirmation of all m agents involved in the cooperation. When an AiCoop notifies the AiC of its arrival at the CT cell, it either receives its share of the CT reward or the waiting rate followed by its share of the CT reward.

Thus, to decide which bids to accept, the general idea is that AiC selects the m proposals with least cost (from the total bids received \mathcal{B}). It does this by considering the reward requested in the bid and the waiting time cost (cost_bid) and then it estimates its expected reward given this cost and its investment. Formally, AiC calculates the cost of each subset b of \mathcal{B} with m elements of the form (bid_{ij}, T_i) . From b , AiC selects the agent that will take the longest time to arrive (i.e., $\max T_b = \max_{(\text{bid}_{ij}, T_i) \in b} [T_i]$), then it can determine the maximum time that each agent will spend in the cell. Finally, it approximates the cost of each bid based on the reward and the waiting time an AiC has to pay:

$$\text{cost_bid}_b = \sum_{(\text{bid}_{ij}, T_i) \in b} (\text{bid}_{ij} + (\max T_b - T_i) \times q) \quad (5)$$

Bringing all this together, AiC estimates the surplus it expects to obtain by taking into account the cost of the selected bids and its own investment to wait for the last AiCoop to arrive. The bids selected belong to the subset b of \mathcal{B} which maximizes the surplus given by:

$$\text{surplus}_j = p_j \times R - \text{cost_bid}_b - r \times (t_j + \max T_b) \quad (6)$$

Now, it may be the case that no bids are received which give a positive surplus. Even though the chosen CM had an expected surplus, by chance it may be that no agents are sufficiently near to provide reasonable bids. In such a situation, the AiC abandons the CT and returns to its ST.

It is important to emphasize the fact that the protocol specified in Figure 1 indicates only the general steps agents follow in this scenario. In order to formulate the agent's decision making procedures, some specific design choices had to be taken. For example, the agent's decision of which bids to select (as described in this section) models the situation in which the AiC pays compensation for the AiCoops's waiting time because all AiCoops have to be in the cell to attain the CT. However, there are other ways of dealing with the fact that various cooperative agents are required to achieve a CT (step [1] of the protocol). For instance, in some circumstances, only two agents might be needed to accomplish a CT or the cooperative agents may not need to wait for the others in order to attain the CT. In both of these cases, the AiC would

not need to pay compensation for the AiCoops's waiting time. Additional situations might occur if the agents are allowed to negotiate the waiting rate. However, many of these alternatives can be modelled using the components and constituent factors introduced in this section. To demonstrate this, in what follows, alternative formulations to calculate surplus are illustrated.

In the situation in which only one more agent is needed to achieve a CT (the simplest case), the bid selected is the one with the cost (cost_bid_b) that maximizes the surplus reward estimated by equation surplus . This time, however, it does not make sense to use \max_T and T_i is used instead.

$$\text{cost_bid}_b = \text{bid}_{ij}$$

If, however, more agents (to be precise m agents) are required to achieve a CT and AiCoops do not need to wait for the rest to achieve the cooperative task, the cost_bid_b must reflect the fact that the bids are not affected by the waiting time. In this case, the cost of each subset b of m elements is approximated by:

$$\text{cost_bid}_b = \sum_{(\text{bid}_{ij}, T_i) \in b} \text{bid}_{ij}$$

Note that for the calculation of surplus , although AiC does not pay compensation for the waiting time, it still has to wait for the furthest AiCoop to arrive ($\max T_b$). Once again, with this new cost calculation, the surplus equation considers the changes needed and the formulation is used transparently.

In summary, the alternative formulations presented for calculating cost_bid_b and surplus illustrate that although there are many situations that agents might encounter as a result of possible interactions, the main components and constituents factors taken into consideration in the agent's decision making are still valid. Moreover, the same concepts can be used to formulate the particulars of alternative applications domains. In particular, the modifications mainly occur in the calculation of cost_bid_b rather than in the whole evaluation of surplus (see [16] for further details of how various rules associated with the achievement of CTs can be modelled).

4. Evaluating the benefits of run-time selection of CMs

Having presented the basic formal framework, we now turn to its evaluation. In particular, we concentrate predominantly on the fundamental hypothesis underlying our work; namely that being able to select the CM at run-time is beneficial. We also explore the impact of the model's main parameters on the performance of the individual agents and on the overall system.

4.1. Experimental setting

The following simulation parameters are fixed for all the experiments: time horizon (duration of 10,000 time units⁸), number of CTs in the grid at any one time (1),

number of agents (5), number of agents needed to achieve a CT (3) and ST reward (1). The agents had the following CMs at their disposal: $CM_1(0, 0.6)$, $CM_2(15, 0.7)$, $CM_3(30, 0.8)$, $CM_4(45, 0.9)$ and $CM_5(60, 1.0)$. These CMs were chosen to be consistent with the observations that there is no best CM and the CMs more likely to succeed take longer to setup (see Section 1).

The experimental variables were the size of the grid and the reward for CTs. The variables that measure the agent's effectiveness are: total agent reward obtained from the accomplishment of its ST and CT tasks, which is termed Agent Utility (AU), and the total number of CTs accomplished (TCT) by an agent.⁹

4.2. Selecting different CMs

The first thing to test is that agents do indeed select different CMs in different circumstances. To this end, Figure 3 shows which CMs were selected in which grid position. Here, the grid size was 20×20 (the remaining three quadrants are simply a mirror of the portion of the upper left shown) and the reward for CTs is 10. In the centre of the grid, $[10, 10]$, the agents choose CMs that minimize the setup cost (even though they have a significant chance of failing to ensure coordination). However, as the agents move further away from the centre, so they increasingly prefer mechanisms that are more likely to succeed (even though they have a correspondingly higher setup cost). The explanation for this behaviour is that as the distance from the centre increases, so does the expected time for another agent to reach the CT square. Thus, to justify its choice of a CT over its ST, the AiC needs to ensure that the cooperation it does enter into do succeed. Whereas, towards the centre of the grid, the time the AiC typically has to wait for another agent to arrive is much smaller and so it can afford to have more cooperations fail. In between are the points where success and setup time are traded off. Figure 4 shows the corresponding expected

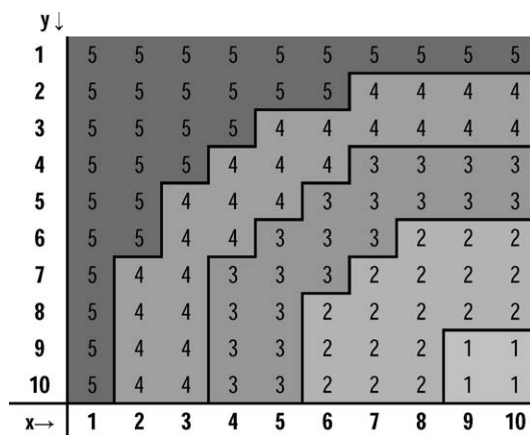


Figure 3. Terrain map showing where the various CMs are selected.

$y \downarrow$											
1	2.20	2.41	2.59	2.74	2.88	2.99	3.08	3.15	3.20	3.22	
2	2.41	2.61	2.79	2.95	3.08	3.20	3.29	3.36	3.41	3.43	
3	2.59	2.79	2.97	3.13	3.26	3.38	3.48	3.55	3.60	3.63	
4	2.74	2.95	3.13	3.29	3.43	3.55	3.65	3.73	3.78	3.81	
5	2.88	3.08	3.26	3.43	3.58	3.70	3.81	3.89	3.94	3.97	
6	2.99	3.30	3.38	3.55	3.70	3.84	3.94	4.02	4.08	4.11	
7	3.08	3.29	3.48	3.65	3.81	3.94	4.05	4.14	4.20	4.23	
8	3.15	3.36	3.55	3.73	3.89	4.02	4.14	4.23	4.28	4.31	
9	3.20	3.41	3.60	3.78	3.94	4.08	4.20	4.28	4.34	4.38	
10	3.22	3.43	3.63	3.81	3.97	4.11	4.23	4.31	4.38	4.41	
$x \rightarrow$	1	2	3	4	5	6	7	8	9	10	

Figure 4. CM’s expected utility in the terrain map.

utility for the various CMs (ave_surplus). Notice that the CM’s surplus expectation decreases as the agent moves away from the centre of the grid.

4.3. Amount of cooperation

To measure the number of times that coordination is attempted, Figure 5 shows the number of cooperative tasks achieved (TCT) by an agent as a function of the reward for achieving a CT and the grid size. The Figure shows that once the reward for CTs

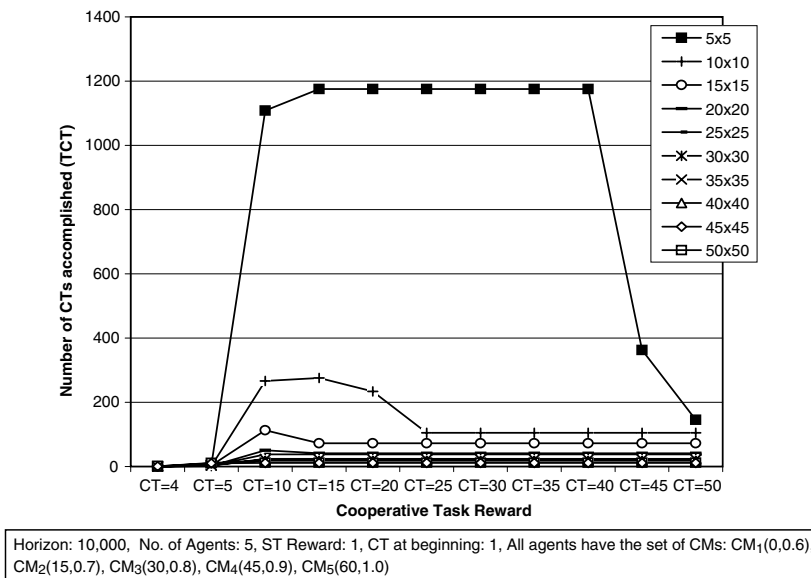


Figure 5. TCT achieved per agent.

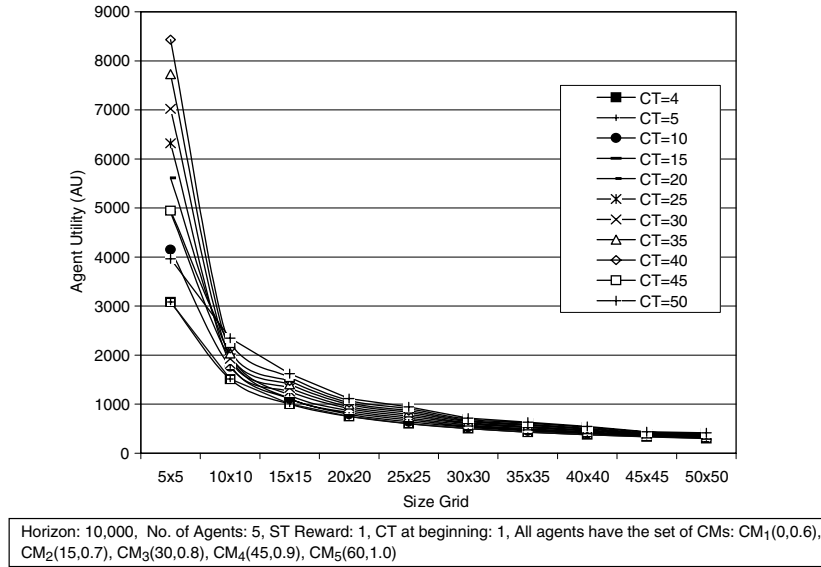


Figure 6. Agent Utility, AU.

is sufficiently high (above 4 in this case) then CTs start getting initiated (i.e. this value is needed before *ave_surplus* becomes positive). At the same time, the TCT increases as the grid size decreases because in small grids agents simply have more chance of finding the CT.

However, it is important to notice that the combination of size of grid and CT reward does have an implication on the selection of the CM and, consequently, on the number of CTs achieved. In the 5×5 grid, for example, the TCT declines when the CT reward is higher than 40. This is because although the agents initiate coordination in exactly the same circumstances, they select a CM that takes more time to set up.¹⁰ The reason for this behaviour is that since the reward is so high, the agents select a CM that guarantees success, regardless of the time invested. As they spend more time establishing coordination, they have less opportunity to find CTs and so with a fixed time horizon they achieve fewer CTs. Similar behaviour is observed in the other grid sizes, although the CT reward level at which coordination starts and the level at which the number of CTs achieved starts to fall obviously varies. While Figure 5 indicates how often the agents cooperated, Figure 6 shows how profitable those decisions are. It is clear that the AU decreases as the grid size increases. Again this is simply because the agents have less opportunity to engage in CTs. Following the same example of the 5×5 grid, it is observed that the agent's utility stops increasing when the CT reward is above 40. Once again, this is explained by the fact that agents do not engage in coordination as often as they do with smaller values of CT reward.

The final key determinant of the amount of cooperation that occurs in the system is the WtC factor. To this end, Figure 7 shows the effect of this factor (the reward for

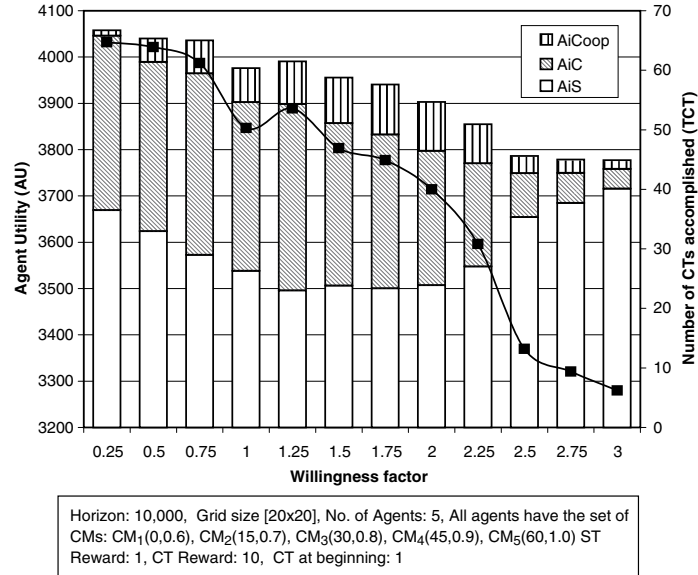


Figure 7. Willingness to cooperate (ω).

CTs is here taken to be 10 and the grid size is 20×20 on the cooperative tasks accomplished. As expected, the more greedy the agents become (increasing ω), the fewer CTs that get initiated and achieved. The same figure (y axis) also illustrates the relation between the TCT achieved and the agent's reward. In this scenario, the more CTs that are accomplished, the more AU that is gained. However, this hides the fact that the various constituents that go towards making AU vary in their relative importance as ω changes. This is because ω affects the amount asked to become a AiCoop (equation bid) and, consequently, the surplus available for the AiC (equation a vesurplus). Thus, for instance, the more selfless an agent becomes, the more reward is gained through the AiC role (the percentage of reward obtained by the AiC and AiCoop roles with $\omega = 0.25$ was 97 and 3% respectively). On the other hand, the more greedy the agents become, the higher the percentage that is obtained by being AiCoop (with $\omega = 3.0$, AiC received 69% of the reward gained through cooperative situations and 31% through being AiCoop).

4.4. Effectiveness of the agent's decision making

Having analysed the effect of the decision making framework's basic parameters, we now consider the impact of being able to dynamically select a CM that is deemed to be appropriate to the prevailing circumstances. A set of experiments were designed as a formal methodology to provide information about the experimental variables. To do this, we employ statistical inference methods; in particular analysis of variance (ANOVA) [9] is used to test hypotheses about differences between the means

collected. The null hypothesis (H0) of equal means can be rejected when the procedure reveals for all experiments that the differences among means are significant ($p < 0.05$) or can be accepted in the contrary case. That is, the execution of an algorithm in a specific environment generates a set of values for the experimental variables that can be analysed under the same circumstances to probe hypotheses using ANOVA. It is important to recall that although ANOVA may reject the equality of means, it does not indicate where the differences between the various elements lie. In our case, because we are interested in a particular agent's performance, it is important to determine which agent (or agents) performs the best. To do this, a postanalysis is applied to the data collected. This test makes a comparison between the data collected and builds groups (as many as necessary) that have statistically homogeneous values. Each group is generated with an associated value (the p value) that indicates the degree of confidence from which each group was built (the higher the number (in a range of 0.0–1.0), the greater the confidence in the grouping).¹¹

To focus on the dynamic selection of CMs, the performance of agents that employ a single CM is contrasted with that of an agent that has the full range of CMs at its disposal. Thus, A_1 only has the $CM_1(0, 0.6)$ to select from, A_2 the $CM_2(15, 0.7)$, A_4 the $CM_4(45, 0.9)$, A_5 the $CM_5(60, 1.0)$, and A_s the whole set of CMs.¹² To test our hypotheses we focus on the variable AU (again averaging by 10 simulations runs) since this measures how much the agent's performance improves depending on a specific environment. Once again the experimental variables were the size of the grid and the reward of the CT.

To evaluate our claim about the benefits of dynamically selecting the CM the following hypothesis (H0) needs to be tested for each agent, in each environment (defined by a specific CT reward and a grid size), using ANOVA.

H0: the AU obtained by A_s in a given environment is the same as that obtained by A_1 , A_2 , A_4 and A_5 .

To start with, Table 3 shows the result of evaluating H0 in an environment with a CT reward of 10 and a grid size of 5×5 . Intuitively, we would expect a different level of performance from the agents that have only one CM to select from and the agent that has a set to select from. The ANOVA result is that the hypothesis is rejected ($p < 0.05$) meaning that the agents' performance does indeed have a statistically significant effect on the AU obtained. To justify this result, Table 4 shows the groups formed by the post analysis. The first thing to notice is that A_s and A_1 are the agents that perform best (they have the highest AU values). This is because, in this environment, the less time that is invested in setting up a CM, the better (because the agents have more opportunity to find CTs). Both A_1 and A_s use the CM that takes the least time to setup (thus, A_s selects CM_1 all of the time).

Table 3. Agent's performance: result of ANOVA.

CT reward = 10 M × N = 5 × 5		
Hypothesis to evaluate	p	Outcome
H0: $AU_{A_1} = AU_{A_2} = AU_{A_s} = AU_{A_4} = AU_{A_5}$	0.000	Rejected

Table 4. Post-analysis of ANOVA.

Agent	AU		
	1	2	3
A ₂	3266.3		
A ₅		3337.7	
A ₄		3353.3	
A ₁			3794.3
A _s			3800.3
p	1.000	0.8728	0.9959

However a good coordination decision maker has to balance the time invested in a CT through the CM selected and the final reward obtained. Thus, to validate this reasoning, it is important to know the total number of CTs achieved by each agent. This is shown in Figure 8. It can be seen that A₁ and A_s did accomplish the most CTs and this corresponds with the reward they obtained. The second best performing group was A₄ and A₅ which have the CMs with the longest time to setup. However from Figure 8 we notice that they did not accomplish any CTs. Instead, they use their time accomplishing STs. This shows that the agent's decision making about when to attempt coordination (and when not to) is as important as selecting the right CM. Thus, A₄ and A₅ gained more AU by not attempting coordination than they would have done by attempting it. Finally the worst performance was by A₂ which attempted coordination some of the time and achieved some CTs, but the reward it gained was not significant enough to make a difference in its AU.¹³

In summary, the post-analysis shows us that the best performance is obtained by the agents which, on the one hand, gain the reward which justifies the time they

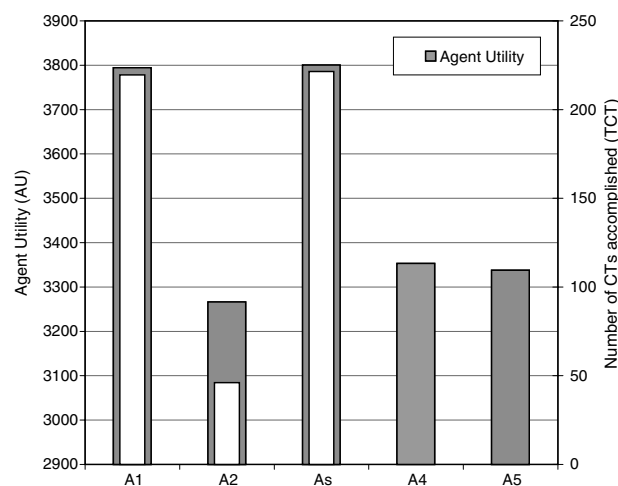


Figure 8. Dynamic selection of CMs.

invest on the cooperative tasks and, on the other, ensures that this reward is better than that they could have obtained by achieving STs alone and having no cooperative attitude.

Having shown the benefits of dynamically selecting CMs in one specific environment, the next step is to evaluate how this generalizes to other environments. In this case, we consider 90 different environments; these have CT rewards in the range of [10, 15, 20, 25, 30, 35, 40, 45, 50] (9 cases) and grid sizes in the range of $[5 \times 5]$, $[10 \times 10]$, $[15 \times 15]$, $[20 \times 20]$, $[25 \times 25]$, $[30 \times 30]$, $[35 \times 35]$, $[40 \times 40]$, $[45 \times 45]$, $[50 \times 50]$ (10 cases). To do this, we simply re-apply the same statistical test to each different environment. Our premise is that the various single CM agents will perform well in different environments, but A_s will perform at least as well as (if not better than) the best of the others in all cases. Being more precise, our interest is to find those environments in which H_0 is rejected¹⁴ and then check (using the same post analysis as described in Table 4) in each environment whether A_s belongs to the group with the best performance (called the *winner group* hereafter).

Our results are as follows. Firstly, there were 55 cases in which H_0 was rejected and 35 cases in which it was accepted. This means that 61% of the time dynamically selecting CMs had a significant effect on the AU in a given environment. In more detail, Figure 9 (left section) shows the number of times that each agent belongs to the winner group. The results are grouped by CT reward (i.e. $CT = 10$ represents the group of 10 environments (of different grid sizes) in which CT was 10). Figure 9 (right section) shows the same information in percentage terms. These results show that A_s obtained a statistically significant better performance than the other agents (61% of the time), A_5 did this 44% of the time and so on. This provides the evidence of the fact that A_s has the dominant behaviour over the other agents in most of the environments.

While previous figures demonstrate A_s 's dominance over the other agents, they do not show in which environments the different CMs are dominant. Again, taking the CT reward of 10 as an example, Table 5 presents the winner agents on a per

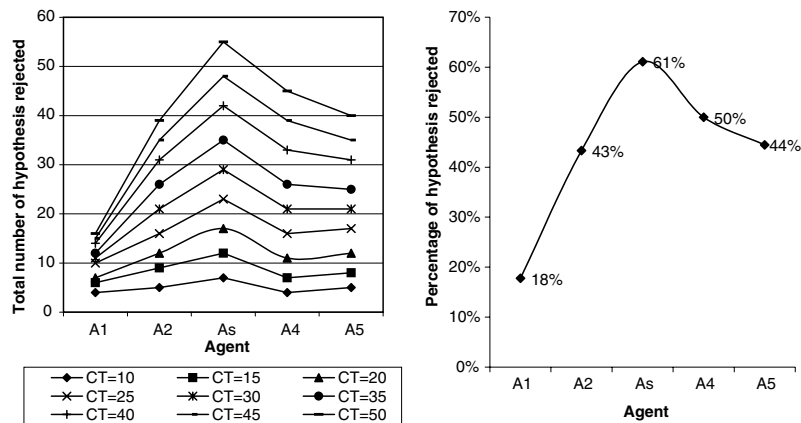


Figure 9. Cases in which the various agents' AU appeared in the winner group.

Table 5. Number of cases in which various agents belong to the winner group.

CT reward = 10					
Grid size	A ₁	A ₂	A _s	A ₄	A ₅
5 × 5	1		1		
10 × 10	1	1	1		
15 × 15	1	1	1		1
20 × 20					
25 × 25					
30 × 30		1	1	1	1
35 × 35			1	1	1
40 × 40	1	1	1	1	1
45 × 45					
50 × 50		1	1	1	1

environment basis.¹⁵ From this it is clear that the CM with lower times to setup (those associated with A₁ and A₂) are selected in smaller grids, while the CMs with higher values are selected in bigger grids. From the same table, we can also observe the grid sizes in which the agents' selections do not have any effect on the AU (those in which H0 was accepted). This figure helps us to clearly illustrate (as previously discussed in Section 4.2) that the combination of CT reward and grid sizes has an effect on the agents' CM selection. Though the figure shows only a few of the environments, the same pattern is followed in the remaining environments. Thus, there are some environments in which some CMs are preferred over others and other environments in which there is no significant difference on the CM selection. However, what is more important is the fact that even though we recognize the situations in which some CMs are preferred over others, what we have shown is that A_s obtains a consistently better performance than the other agents.

5. Related work

There are three broad strands of work that are related to our model of reasoning about coordination:

- work on techniques for coordinating multiple agents (Section 5.1),
- work on the dynamic selection of problem-solving techniques (Section 5.2),
- work on flexible reasoning about coordination (Section 5.3).

Each of these will now be dealt with in turn.

5.1. Techniques for coordinating multiple agents

Coordination has been widely studied by MAS researchers because it is a key form of interaction in many systems. Because of this fundamental role, a number of

techniques exist for coordinating agents' interactions. These include appealing to a higher authority agent in an organizational structure [6, 19, 32], instituting social laws that avoid harmful interactions [4, 37], using contracting protocols or bids exchanged in a market place to allocate tasks [25, 32, 35, 38], iteratively exchanging tentative plans until all constraints are satisfied [14], negotiating to find agreements [26, 35, 36] and so on. In most multiagent systems these coordination mechanisms are hard-wired at design time (and enacted at run-time) as the only means of coordinating actions. Nevertheless, their relevance with our work is that they (and many others not mentioned here) are representative of the set of CMs that agents are likely to have at their disposal in our decision making process. However, in order to incorporate them into our model the meta-data about cost and likelihood of success of the various mechanisms needs to be established (see [16]) for a preliminary attempt in this direction for several of the most common mechanisms).

5.2. The dynamic selection of problem-solving techniques

Comparatively little work addresses run-time selection of particular coordination protocols. However, work that deals with the dynamic selection of problem-solving techniques is somewhat related. This work is concerned with adjusting at run-time, the computational resources made available against the quality of the service provided. This might also involve the dynamic selection of algorithms to optimize the overall system's performance. To this end, several solutions, under different assumptions, have led to the development of a variety of techniques such as design-to-time algorithms [21], algorithm portfolio design [22] and anytime algorithms [40]. Although the similarities consist of selecting at run-time the algorithm with the best performance, the main distinction is that we specifically focus on those aspects involved in reasoning about a coordination decision. In spite of the fact that it would be possible to map each CM as a different algorithm, our concern is more with exploring the necessary elements to make such decisions effectively and on the effects of such considerations. Moreover, we do not believe that optimal performance is possible when dealing with the complexities and uncertainties of coordination problems (even in idealized scenarios such as our own). Thus, our research is more concerned with the flexibility involved in reasoning about coordinating agents, which is the line of research we will discuss in the subsequent paragraphs.

5.3. Flexible reasoning about coordination

Flexible reasoning in this context means agents make more decisions about their coordination activities at run-time so that their choices can better reflect their prevailing circumstances. Thus, our work can be seen as dealing mainly with flexible deliberation. However, flexibility in multiagent systems with respect to coordination can have a broader scope than this [30]. Specifically, it can cover two aspects:

- flexibility in particular protocols (such as multiagent planning or market-like protocols) and
- flexibility in reasoning between alternative coordination mechanisms.

Regarding the first point of flexibility in particular protocols, Durfee and colleagues have been concerned with dynamically deciding on coordination parameters in the context of multiagent planning [14, 15]. Specifically, they have studied two different problems: (i) deciding the level of detail to reason about in multiagent planning [7, 8] and (ii) selecting at run-time the plans with high expected quality [33]. The key issue when dealing with levels of abstraction is that agents derive summary information from plans and decide the right level they should reveal and communicate about. In their work, agents use different levels of representation detail and each agent can work at different levels with different other agents. In short, agents dynamically choose what to represent (what to coordinate over) using planning as the mechanism. To this end, their contribution is more in dynamic reasoning over one mechanism (planning) rather than reasoning over a set of such mechanisms. Nevertheless, the concepts taken into consideration in the summary information at the most abstract level (the preconditions and effects of the application of a plan) are very important in attaining flexibility in coordination. Thus, this research attacks complementary concerns to our research. Turning to the work on evaluating the quality of plans, Pappachan and Durfee [33] associate a variety of measures to plans when taking decisions about the coordination of joint activities. To be more specific, they propose a heuristic to select the plan to execute based on performance metrics of the plan's quality. This takes three components into account: the plan reward (the reward gained after the plans execution), the plan cost (the total cost of all the actions performed) and plan reliability (the likelihood of successfully completing the plan). Because in some situations one component might be more important or desirable than others, several strategies are evaluated by assigning different weights to each criteria. There are clearly several similar aspects to our work: the necessity of taking decisions at run-time, a metric to measure the quality of the outcome and criteria to tradeo. the alternatives. However, once again, the reasoning is performed over only one coordination strategy.

We now turn to the second point of flexibility in reasoning about alternative coordination mechanisms. One of the first attempts to discriminate coordinating techniques was the work of Ramamritham and co-workers in the area of distributed scheduling problems [34]. Specifically, they proposed a mechanism that would deliberate about the selection of various task assignment algorithms. In particular, they assign tasks to nodes by taking into account timing and resource requirements (for example, they would prefer to assign a task to a node with enough resources to deal with it, rather than to one with no resources at all). The objective of the scheduling is to maximize the number of completed tasks before their deadline (compared to the number of invoked tasks). During the scheduling process, nodes receive tasks and decide whether they can complete the task with their own resources. Then if they cannot, all nodes cooperate to locate the node which can guarantee the completion of the task. The authors propose and evaluate four algorithms for selecting the nodes: random, focused addressing, bidding and flexible.

The most interesting of these is the flexible algorithm which consists of deciding whether to use the bidding, the focused addressing or both of them in an opportunistic manner. This is relevant to our work because this algorithm reasons in a flexible manner about when to select a particular protocol (random, focused addressing and bidding) given the particular circumstances in which the decision is taken. Their main result is that the flexible algorithm outperforms the others in most cases. This, in turn, provides some insight into the potential benefits of run-time selection of algorithms by considering the system's circumstances and by taking decisions at the time the choice is made. However, the main drawback of their solution is that agents are assumed to be inherently cooperative and to constantly communicate accurate information. Thus, although the decision making might be performed with imprecise information (for example, the focused algorithms might not select the best node), there is always an alternative way to correct a wrong selection. Additionally, only a very small number of metrics affect the decision-making process (just the node surplus) and general parameters of the system that can be tuned. In short, the underlying approach is similar to our research in that the factors involved in the decision-making process can be out of date and uncertain, though in their research all nodes work together to obtain highly consistent information about the factors that are taken into consideration in the reasoning model.

In terms of introducing flexibility in general, Jennings observed *that all coordination mechanisms can ultimately be reduced to (joint) commitments and their associated (social) conventions* [27]. In demonstrating the feasibility of this hypothesis, he showed that most of the possible interactions and flexibility needed to deal with changing environments can be dealt with by considering commitments and conventions. This analysis is important in two senses. Firstly, it supports our claim in the sense that if all coordination mechanisms can be reduced to commitments and conventions, then it is possible to make a unified evaluation of them. That is, it is possible for all mechanisms to be analysed, measured and evaluated under similar terms. Secondly, however, the problem with his work is that the system designer is responsible for determining which conventions are present in the system, which kinds of interactions particular conventions might be used for, and when to use them. However, as we have argued in this paper, this is not really suitable for dynamic and uncertain environments.

In a similar vein, the Generalized Partial Global Planning framework (GPGP) [10, 11, 31] designed and evaluated a family of algorithms that allows a group of agents to coordinate their interactions through a set of coordination algorithms. In particular, it focuses on two key factors: (i) the representation and reasoning about the features of the task environment (TEAMS) and (ii) the development of coordination mechanisms that can respond to those task environment structures. In this framework, the global problem consists of discovering interactions that are constrained by the activities of the other agents. Hence, GPGP can be seen as an optimization framework in which the agent's local optimization solutions are combined (because of task relationships) into a global problem constrained by quality, time or resource considerations. Specifically, a coordination mechanism in GPGP consists of a protocol that detects and activates actions (communication, information gathering or proposal of commitments) in response to inter-agent task activities. The most

important of these activities are the ones that facilitate the scheduling process (i.e. those that generate commitments). These commitments are at the core of the problem solving process because they solve the task constraints related to time and resources (agents commit to tasks by specifying the time by which the task will be satisfied and the quality with which the action will be done). To this end, GPGP uses TEAMS both to represent the task structure and the aspects taken into consideration when solving the global constraint problem. Thus, the set of protocols in GPGP are one-shot mechanisms that discover where commitments are needed (or can be broken) when the agents' task structures are revised (because the coordination mechanisms indicate the time at which the structures are revised). Hence, the coordination mechanisms are triggered to generate or update those commitments that represent the central feature of solving a coordination problem. And, as can be seen, the representation of the tasks are fundamental to the coordination mechanisms for dealing with coordinating activities. In our research, the agent's take decisions about when to use a particular CM based on the particular circumstances when the decision is taken. In contrast, in GPGP the coordination mechanisms are activated in predefined situations which might occur during execution (no more than one coordination mechanism can be activated in a given situation which is why GPGP does not provide a reasoning mechanism to discriminate between them). In summary, GPGP sees the coordination mechanisms as exclusive protocols whose execution generates alternative outcomes, and, consequently, different benefits to the scheduling. Whereas in our case, the CMs may produce the same outcome and then the problem is how to decide which one is more appropriate.

Thus far, two aspects have been identified as permitting flexibility in the coordination of problem solving. On the one hand, there is the necessity of introducing various 'mechanisms' to solve the coordination problem, and, on the other, the incorporation of a mechanism to reason over the selection of such mechanisms. As previously discussed, some research has investigated the former (the GPGP framework and Jennings's conventions) and some the latter (Ramamritham's research). However, there is one piece of work that studies both aspects and this is [1]. The aim of Barber et al.'s work is to dynamically select the most appropriate coordination mechanism in a given situation. To this end, they present a software engineering framework that enables agents to vary their coordination mechanisms according to the prevailing circumstances. They identify criteria for determining when a particular mechanism is appropriate. The coordination mechanisms analysed are: negotiation, voting, arbitration and self-modification. They identify the following characteristics: (a) constraints associated with the mechanism,¹⁶ (b) cost of communication and execution and (c) quality of solution. Agents associate a weight to each of these characteristics and calculate the cost of each mechanism by adding the cost of all the above mentioned features. Here, the drawback is that calculating the weights for the features is not straightforward. In particular, it is difficult to assess whether or not a given weight will improve overall performance. Also they do not demonstrate whether agents do indeed perform any better by having this capability. Furthermore, the agents do not reason in terms of the other agents in the environment and therefore the decisions about when to select a mechanism are somewhat arbitrary. Finally, even though the authors have analysed each strategy, we consider that more

work is needed to provide a more systematic decision procedure for actually trading off these criteria.

The last line of research to discuss is that of Boutilier [3]. He presents a decision making framework, based on multiagent Markov decision processes (MMDP), that reasons about the state of a coordination mechanism. He proposes the use of coordination mechanisms as protocols and introduces the concept of states of coordination to incorporate them in the MMDP. This protocol is represented as a finite state machine which models the coordination mechanism with states and with possible coordination interactions (decision rules). The decision rules allow agents to constrain their selection of actions. The key insight in this proposal is to allow agents to reason not only in terms of optimal joint actions, but with the state space of coordination in such a way that agents decide the possible next action based upon their particular state of coordination. However, agents with this framework do not reason in terms of their local states, rather, they observe and take decisions based on the global state that they are always assumed to have access to. Thus, this work is concerned with optimal reasoning with respect to a given coordination mechanism, rather than actually reasoning about which mechanism to employ in a particular situation.

6. Conclusions and future work

This paper argued that autonomous agents need to be given the flexibility to dynamically select the mechanism they use for coordinating their actions during cooperative problem solving. We believe this is especially important when the environment in which the decision making takes place is dynamic, open, and heterogeneous. In such situations, making decisions about coordination is difficult because it is impossible to enumerate in advance the wide variety of contexts in which coordination is likely to be needed. To this end, we developed a decision making framework that enables agents to tailor their coordination decisions to the prevailing context. Specifically, this covers deciding when to coordinate (and when not to do it), which coordination mechanism to use and with which agents to cooperate. This framework enables agents to make informed choices about their coordination actions because we abstractly characterize coordination mechanisms in terms of their cost and their expected benefits. Thus, in our framework, this decision is clearly separated from the enactment of the mechanism which is how the task is actually achieved and how the actions of the various agents are actually coordinated. Moreover, we showed that in our grid world scenario the agents are more successful by having the ability to dynamically select the coordination mechanism.

Although the specifics of the decision procedures are clearly related to our particular grid world scenario, we believe that the basic processes and structures we have developed are suitable for reasoning about coordination mechanisms in more general domains. For example, several of the agent's decisions relate to the proximity of potential collaborators. This conception of distance in the grid can easily

and naturally be mapped into a range of analogous concepts that have more general application. The first of these is the notion of trust in social relationships (as represented for example, by the degree of connectivity in social network theory [5]). Thus, cases in which agents are more certain of receiving help, because there is a high degree of trust between them, are similar to the cases in our scenario where potential collaborators are close to hand. In such situations, our results indicate that agents are more likely to attempt coordination using mechanisms that have relatively low times to setup. On the other hand, when collaboration is more difficult to establish, because there is a low degree of trust (equivalent to the agent being far away), the agents are more likely to opt for mechanisms that are more likely to succeed. The second relates to the dynamism in the environment. In more static environments, there is a greater chance of more accurately predicting the behaviour of the various agents present in the system. This corresponds to the case of the agent being near the centre of the grid as this situation involves much less uncertainty. In more dynamic environments, on the other hand, predicting the behaviour of others is more difficult and so corresponds to decision making around the edges of the grid.

Our work also demonstrated that a successful agent is one that takes the right decision about the benefits it will accrue for the time invested in the coordination mechanism vs. what the agent would gain by achieving its individual tasks. The importance of this result is that it corroborates our belief that in some environments an agent that can dynamically select its CM according to its circumstances is able to take more profitable coordination decisions. However, we do not claim that our agents always select the best CM. For example, when agents are at the edges of the grid, they assume it will take a long time for the AiCoops to arrive to the CT cell and consequently they select the CM that has the highest probability of success regardless of the high cost of setting up. This is true even though this is not always the case in practice. To us then, an agent that makes a good decision needs to balance the coordination mechanism setup cost and its likelihood of success with the information predicted. However, in the experimentation reported here good performance is measured simply by the total reward obtained at the end and not by how well this tradeoff was performed. Thus, a potential conclusion from this could be that the constituent factors of the framework need to be refined to more precisely model the environment. However, the process of modelling has the drawback that unless we deal with static environments (which is not the case), this task can imply a great effort and cost [13]. Thus, our research position is that our effort must be directed to, on the one hand, having a reasonable model of those aspects on which agents base their coordinating decisions and, on the other hand, exploring the “right” level of approximation in modelling in order to ensure the agent can coordinate effectively in practice [13]. We believe, our framework is a mature point of departure for this further study because it provides a good approximation about how the key factors that need to be taken into consideration should be combined. Nevertheless, more work is needed to refine the modelling problem and to systematically evaluate the alternatives. The other main area of future work involves systematically classifying coordination mechanisms according to our meta-data criteria. This is necessary in order to put our framework into practical applications.

Acknowledgments

We would like to thank Rachel Bourne for her contribution to the formalisation of our model. The first author also acknowledges the funding of Mexico's National Council of Science and Technology, CONACyT and the National Laboratory of Advanced Computer Science, LANIA.

Notes

1. This work was done while the first author was a PhD student in the Group Intelligence, Agents, Multimedia Group at the University of Southampton.
2. If several agents arrive at a CT square at the same time, one of them is arbitrarily deemed to be in charge and, if an agent finds more than one CT in a given cell, it randomly selects one of them for further analysis.
3. Other attributes could undoubtedly be added to this list (such as quality of coordination, robustness, overhead limitations, communication cost and so on), but here we focus on those that we believe are necessary (if not sufficient) for our purposes.
4. In this protocol, the problems of dividing the problem into subtasks (task decomposition) and how the results from the various subtasks are merged into an overall solution (task synthesis) are not considered a part of the coordination activity. However, in coordination mechanisms such as multiagent planning, task decomposition might be part of the requirements phase.
5. Though this may not be a globally optimal criterion for deciding which CM to use, it makes sense from a self-interested agent's point of view.
6. Note that in order to estimate $ave_surplus$ it is assumed that the m is determined in advance or is part of the agent's knowledge. However, this assumption may not always be valid for cases in which the number of cooperative agents depends on the particulars of the coordination's objective. In such cases, the agents will need to predict this number based on previous experiences or some how estimate this information (e.g., the straightforward solution is that agents maintain an average of the number of helpers each time they accomplish coordination; more complex solutions might involve building a model for each agent each time there is an interaction).
7. Note that the AiS use the actual values of the concepts discussed, whereas the AiC's task is to make a good approximation of these components through equation ave_bid .
8. To set the value of this variable, a utility rate (the agents' total reward divided by the horizon) was calculated for the experiment where the duration was varied from 10,000 to 100,000 time units. The statistical measures of the utility rate in this case had a standard deviation of $2.10E-02$ and a variance of $8.11E-04$. This allows us to conclude that the duration of the experiment in this range does not have a significant effect on the rate of utility since regardless of the horizon, the utility rate is maintained. Therefore we opted for the lower level since this meant the experiments could be conducted more quickly.
9. All experimental variables are calculated by averaging the totals obtained by 10 simulation runs and by all agents in the system. Thus, for example, TCT achieved is the average number of CT tasks achieved by the five (in our case) agents in the system.
10. Being precise, agents in this situation selected the five CMs in the following grid positions:

y ↓					
1	4	3	2	3	4
2	3	1	1	1	3
3	2	1	1	1	2
4	3	1	1	1	3
5	4	3	2	3	4
x →	1	2	3	4	5

11. For example, Table 4 shows the result of this cluster analysis. Here, three groups were generated (labelled 1, 2 and 3) and the respective p values were 1.000, 0.8728 and 0.9959.
12. Note that $CM_3(30, 0.8)$ was not associated to any agent. This is because we decided to maintain a constant number of agents in the environment (for reasons of comparison) rather than change the experimental settings. CM_3 was omitted because it lies in the middle of the range. However, the same experimentation as described here was performed with each of the CMs missing. Although the number of cases rejected were different in each case, the general trend and conclusions remained unchanged [16].
13. Notice that the agents' reward obtained is influenced by the probability of success of the CM which represents the final reward the agents obtain as a percentage of the CT reward. Thus, it can be seen that the number of CTs accomplished gives an indication of the agent's overall performance, but it does not necessarily follow that agents will be more productive if they accomplish more CTs.
14. Recall that the cases in which H_0 is accepted are those environments in which there is no significant effect of the reasoning about coordination on the AU obtained by any agent.
15. Since the post analysis generates groups with inferior and superior thresholds, it is possible to have agents belonging to more than one group (i.e. some AU values might belong to two groups). The results presented here take into account all the members of the winner group regardless of whether some of their AU values are closer to the superior limit of the subsequent group.
16. This involves the particular mechanism's requirements, whether communication is fundamental, whether some specific roles need to be played, whether any authority hierarchy is necessary and so on.

References

1. K. S. Barber, D. C. Han, and T. H. Liu, "Coordinating distributed decision making using reusable interaction specifications," in *Design and Application of Intelligent Agents: Third Pacific Rim International Workshop on Multi-Agents (PRIMA 2000)*, Melbourne, Australia, 2000, pp. 115.
2. R. A. Bourne, C. B. Excelente-Toledo, and N. R. Jennings, "Run-time selection of coordination mechanisms in multi-agent systems," in *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000)*, 2000, pp. 348–352.
3. C. Boutilier, "Sequential optimality and coordination in multiagent systems," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*. Stockholm, Sweden, 1999, pp. 478–485.
4. W. Briggs and D. Cook, "Flexible social laws," in M. E. Pollack, (ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, Morgan Kaufmann publishers Inc.: San Mateo, CA, 1995, pp. 688–693.
5. R. S. Burt, *Toward a Structural Theory of Action. Network models of Social Structure, Perception, and Action*, Academic Press: New York, NY, 1982.
6. K. M. Carley and L. Gasser, "Computational organization theory." in G. Weiss, (ed.), *Multiagent Systems: A Modern Approach To Distributed Artificial Intelligence*, The MIT Press: Cambridge, MA: chapt. 7, pp. 299–330.
7. B. J. Clement and E. H. Durfee, "Theory for coordinating concurrent hierarchical planning agents using summary information," in *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, Orlando, Florida, AAAI Press: Menlo Park, CA, 1999, pp. 495–502.
8. B. J. Clement and E. H. Durfee, "Top-down search for coordinating the hierarchical plans of multiple agents," in O. Etzioni, J. P. Müller, and J. M. Bradshaw, (eds.), *Proceedings of the Third International Conference on Autonomous Agents (AGENTS'99)*, Seattle, Washington, ACM Press, 1999, pp. 252–259.
9. P. R. Cohen, *Empirical Methods for Artificial Intelligence*, The MIT Press: Cambridge, MA, 1995.
10. K. S. Decker, "Environment centered analysis and design of coordination mechanisms," Ph.D. thesis, Department of Computer Science, University of Massachusetts, Amherst, 1995.
11. K. S. Decker and V. R. Lesser, "Designing a family of coordination algorithms," in M. Huhns and M. P. Singh, (eds.), *Readings in Agents*, vol. 4.4 Interactive Agency, Morgan Kaufmann Publishers: San Mateo, CA, chapt. 4. Models of Agency, 1997.

12. K. S. Decker, K. Sycara, and M. Williamson, "Middle-agents for the internet," in M. E. Pollack, (ed.), *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan, Morgan Kaufmann Publishers: San Mateo, CA, 1997, pp. 578–583.
13. E. H. Durfee and T. A. Montgomery, "Coordination as distributed search in a hierarchical behavior space," *IEEE Trans. Syst. Man Cyber.*, vol. 21, no. 6, pp. 1363–1378, 1991.
14. E. H. Durfee and V. R. Lesser, "Partial global planning: A coordination framework for distributed hypothesis formation," *IEEE Trans. Syst., Man Cyber.*, vol. 21, no. 5, pp. 1167–1183, 1991.
15. E. H. Durfee, "Practically coordinating," *AI Mag.*, vol. 20, no. 1, pp. 99–116, 1999.
16. C. B. Excelente Toledo, "The dynamic selection of coordination mechanisms," Ph.D. thesis, Department of Electronics and Computer Science, University of Southampton, 2003.
17. C. B. Excelente-Toledo and N. R. Jennings, "Learning to select a coordination mechanism," in *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, Bologna, Italy, 2002, pp. 1106–1113.
18. J. A. Filar, *Competitive Markov Decision Processes.*, Springer-Verlag: New York, NY, 1997.
19. M. S. Fox, "An organizational view of distributed systems." *IEEE Trans. Syst. Man Cyber.*, vol. 11, no. 1, pp. 70–80, 1981.
20. J. Galbraith, *Designing Complex Organizations*, Addison-Wesley Publishing Company, Inc.: Reading, MA, 1973.
21. A. Garvey and V. Lesser, "Design-to-time real-time scheduling," *IEEE Trans. Syst. Man Cyber. Spl. Issue Plan. Schedul. Cont.*, vol. 23, no. 6, pp. 1491–1502, 1993.
22. C. P. Gomes and B. Selman, "Algorithm portfolio design: Theory and practice," in *Proceedings of the Thirteen Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Morgan Kaufmann Inc: New Providence, RI, 1997.
23. S. Hanks, M. E. Pollack, and P. R. Cohen, "Benchmarks, test beds, controlled experimentation, and the design of agent architectures," *AI Mag.*, vol. 14, no. 4, pp. 17–26, 1993.
24. L. M. J. Hogg and N. R. Jennings, "Socially intelligent reasoning for autonomous agents," *IEEE Trans. Sys. Man Cyber.-Part A*, vol. 31, no. 5, pp. 381–393, 2001.
25. B. A. Huberman and T. Hogg, "Distributed computation as an economic system," *J. Econ. Perspect.*, vol. 9, no. 1, pp. 141–152, 1995.
26. M. N. Huhns and L. M. Stephens, "Multiagent systems and society of agents," in G. Weiss, (ed.), *Multiagent Systems: A Modern Approach To Distributed Artificial Intelligence*, The MIT Press: Cambridge, MA, Chapt. 2, 1999, pp. 79–120.
27. N. R. Jennings, "Commitments and conventions: The foundation of coordination in multi-agent systems," *Knowl. Eng. Rev.*, vol. 8, no. 3, pp. 223–250, 1993.
28. D. N. Kinny and M. P. Georgeff, "Commitments and effectiveness of situated agents," in *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, Sydney, Australia, Morgan Kaufmann publishers Inc.: San Mateo, CA, 1991, pp. 82–88.
29. V. R. Lesser, "Reflections of the nature of multi-agent coordination and its implications for an agent architecture," *Autonom Agents Multi-Agent Syst.*, vol. 1, no. 1, pp. 89–111, 1998.
30. V. R. Lesser, "Cooperative multiagent systems: A personal view of the state of the art," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 1, pp. 133–142, 1999.
31. V. R. Lesser, K. S. Decker, T. Wagner, N. F. Carver, A. Garvey, B. Horling, D. E. Neiman, R. Podorozhny, M. NagendraPrasad, A. Raja, R. Vincent, P. Xuan, and X. Zhang, "Evolution of the GPGP/TAEMS domain-independent coordination framework," Technical Report 02-03, University of Massachusetts/Amherst Computer Science Technical Report., 2002.
32. T. W. Malone, "Modeling coordination in organizations and markets," *Manage. Sci.*, vol. 33, no. 10, pp. 1317–1332.
33. P. M. Pappachan and E. H. Durfee, "A multiagent plan evaluation heuristic for real-time coordination," Personal communication, 2000.
34. K. Ramamritham, J. A. Stankovic, and W. Zhao, "Distributed scheduling of tasks with deadlines and resource requirements," *IEEE Trans. Comput.*, vol. 38, no. 8, pp. 1110–1123, 1989.
35. J. S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*, The MIT Press: Cambridge, MA, 1994.

36. T. W. Sandholm and V. R. Lesser, "Issues in automated negotiation and electronic commerce: Extending the contract net protocol," in V. Lesser and L. Gasser (eds.), *Proceedings of the First International Conference in Multiagent Systems (ICMAS-95)*. San Francisco, CA. The MIT Press, 1995, pp. 328–335.
37. Y. Shoham and M. Tennenholtz, "On the synthesis of useful social laws for artificial agent societies," in *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*. San Jose, California, 1992, pp. 276–281.
38. R. G. Smith and R. Davis, "Frameworks for cooperation in distributed problem solving," *IEEE Trans. Syst. Man Cyber.*, vol. 11, no. 1, pp. 61–70, 1981.
39. M. P. Wellman, "A market-oriented programming environment and its application to distributed multicommodity flow problems," *J. Artif. Intell. Res.*, vol. 1, pp. 1–23, 1993.
40. S. Zilberstein, "Using anytime algorithms in intelligent systems," *AI Mag.*, vol. 17, no. 3, pp. 73–83, 1996.