First Int. Conf. on Maulti-Agent Systems AAAI-ICMAS 95, San Francisco, Ca., June 1995

Hierarchical and Lateral Coordination in MAS: An Analysis of Message Traffic Flow

A. Knoll

University of Bielefeld, Faculty of Technology,
Department of Computer Science,
Postfach 100131, D-33501 Bielefeld, Germany
knoll@techfak.uni-bielefeld.de

J. Meinkoehn

National Research Corporation for
Mathematics and Data Processing (GMD),
Hardenbergplatz 2, D-10623 Berlin, Germany
meinkoehn@fokus.gmd.de

Abstract

The general goal of using multi agent networks for complex problem solving is the maximisation of the quality of the result to be obtained at minimum cost. Both the granularity of the agent society and the competence assigned to each individual agent determine the information flow in the network. The great number of parameters involved make it difficult for the designer to optimally adapt the structure of the network to a given class of tasks. In this paper we outline possible network structures and present an approach for determining a number of important statistical parameters characterising the network at a relatively abstract level. The abstraction enables a comparison of different network structures. The methods for the analysis may, however, be readily refined to evaluate a specific problem. As an example we discuss the use of the multiagent paradigm for structuring the cooperation of sensor networks in robotics. Our analysis is supplemented by simulation results, which prove a superiority of lateral over pure hierarchical coordination, particularly under severe environmental conditions, such as agent failure.

Introduction

There are two main issues to be dealt with when organising teams of interacting agents [Fox 81]. The first of these issues is the *structure* of the team and the second issue is the definition of a *control mechanism* for coordinating the members of the team. Criteria for selecting a structure and a control mechanism for a given network with a specific ensemble of sensor agents are both the *complexity* (e.g. the arrival rate of sensing tasks, the amount of knowledge necessary for resolving the problem and for coordinating the a priori knowledge and the resources) and the *uncertainty* (of acquired data, of the behaviour of the sensor agents and of the behaviour of the environment). The latter determines the number of agents necessary for completing the task.

A. Team Structure

A structure is specified by defining capabilities of the team members and by assigning responsibilities to them. This implies that certain agents may specialise in particular tasks such as sensing; others work on different problems (e.g. pre- or postprocessing data, establishing communication paths or coordinating subordinate agents).

This differentiation of capabilities and responsibilities, however, is valid only for hierarchical structures, whereas in the case of *lateral* structures the agents may be locally disparate but have equal rights and duties (as far as equal duties are possible; for agents interacting with an external environment this is not normally the case). In a *simple hierarchy*, there exist a number of agents on a lower level, which are coordinated by an agent on an upper level. The agents on the lower level are all specialised to unique classes of tasks or they may have universal capabilities. In either case they are subordinated in responsibility to the upper level agent.

In an *extended hierarchy* (such as proposed in [Iyengar 92]) there is more than one level of coordinating agents. Specialised agents may coexist with non-specialised agents in one network. If there are non-specialised agents on the same level, then there is a potential for these agents to coordinate themselves by interchanging information directly without any arbitration by a superior agent, i.e. within one *flat* layer. Both hierarchical and flat structures may coexist in one network: subtrees are structured laterally and organise their cooperation within their layer of the subtree autonomously after receiving a certain task from their superior agent (or an external mandator). Finally, a network in which there are *only* lateral dependencies is called a *cooperative*. In such organizations there is no coordinating authority and agents may be members of different collectives working on different tasks. This structure of overlapping cooperatives forms the basis of our work because it also contains hierarchical structures as a subset of possible specialisations (through the assignment of limited capabilities/ competence to each individual agent).

B. Network Control Mechanism

The control mechanism defines *how* and *when* agents communicate (interact). From an interaction, a transfer of control may result, which in turn is preceded by a selection process. The mechanism for coordinating communication between the agents may be either static, i.e. communication channels and hence groups of agents for working on a certain task are fixed (e.g. [Rao 93]), or it may be dynamic. The latter means that cooperation between agents is agreed upon for a limited period of time and vanishes after completion of the task. During the selection process, an exchange of information with different agents may take place and the decision for or against cooperating with a potential partner may be taken after evaluating the latter's offer in terms of promised result quality, e.g. time of completion and measurement precision.

A simple static control mechanism is the *conservative* selection strategy: An agent which initiates a cooperation for a certain class of tasks for the first time looks for suitable partners and (possibly randomly) selects one of them. When the same task (or class of tasks) reappears, the agent selects the same partner again. After some time, all classes of tasks have caused each agent to "know" each partner for every task class and the partnerships for cooperation are fixed. With a dynamic strategy, current partnerships do not affect future relations. The selection process is repeated each time a cooperation becomes necessary. A simple dynamic strategy is *random* selection: Of all potential partners for an imminent cooperation one is chosen at random. If the momentary state of potential partners (e.g. workload) may be inquired, this information may affect the decision. An example for a strategy presupposing such knowledge is *shortest queue*: The agent with the smallest workload (as expressed by the length of its task queue) is awarded the task. With dynamic strategies the effects of sensor failure are less severe and the addition or removal of agents does not necessitate a complete re-initialisation of the network. Such dynamic strategies are obviously better suited for lateral networks in which agents are less specialised than in hierarchical networks where in certain situations there is only a limited choice of partners. Note that the selection strategy may have a drastic effect on the performance of the network; we shall return to the issue of selecting a suitable control strategy for given network structures below.

Lateral and Dynamic Sensor Agent Coordination

As data fusion methods become more powerful and widespread, there is a natural tendency in the field of manufacturing and robotics to design interconnected sensor systems with an ever increasing number of sensors contributing to the solution of a given sensing task. It is the purpose of these sensor networks to acquire information about the environment which is more comprehensive and more precise than the contribution from any single sensor. The multitude of problems to be dealt with turn this application of MAS into a very attractive area of research.

Each of the sensors is faced with the problem of making decisions based on its observation of a part of the environment and on partial a-priori information. Both the need for transferring information to locally disparate sensors and the need to associate their data require a mechanism for transporting data of different structure at minimal costs. To reduce the amount of data to be transferred, only those sensors that are necessary for the solution of a specific sensing task should be activated. This also makes it possible for the rest to work in parallel on the solution of other tasks. Consider a vision system with cameras of overlapping fields of view (e.g. for distributed vehicle monitoring [Carver 93]). The quality requirements of the task permitting, it is obviously desirable not to focus all cameras to a single specific object at one point in time, but to track different objects (possibly using the same sensor data). This is particularly important when the operations required to refocus a sensor are costly (to model this, a "repair delay" parameter was used in our simulations; see below).

A. Autonomous Sensor Agents

It may be very useful to fuse information on different aspects of one object using a set S_1 of sensors observing a certain spatial area A_1, while the information on a different area A_2 produced by the union of a subset of S_1 and a second set of sensors S_2 is processed by other fusion entities. This suggests another kind of tasks to be accomplished in the network: the coordination of information *processing* entities, i.e. agents that do not necessarily comprise a physical sensor. There are three different interesting classes of the mapping of physical sensors to sensor agents:

- *The $1 \rightarrow M$ mapping:* One physical sensor provides information for M more or less specialised agents. In the field of Computer Vision the extreme view would be "one agent per camera pixel;" realistically, teams of agents are examined, which cooperate on the segmentation of regions [Demazeau 94].
- *The $1 \rightarrow 1$ mapping:* Sensors are equipped with local data (pre-)processing and communication facilities.
- *The $M \rightarrow 1$ mapping:* This is the classical hierarchical network in which M sensors are controlled by one superior agent.

Clearly, a mix of the three is also possible, this would result in an $N \rightarrow M$ mapping, where N agents at a lower level interact with M agents at a higher level of a hierarchy. If a large sensor system is structured according to these schemes, the high number of nodes enforces a strategy for sensor coordination to achieve a common goal with minimal cost. This is the reason, therefore, that architectures must be developed to structure such sensor systems systematically, to organise them efficiently and to ensure a certain degree of fault tolerance by avoiding central controllers or coordinators (as was demanded in [Iyengar 90]); see [Henderson 84] for early work on the centralised approach. It will be shown below that a lateral reconfigurable structure offers significant advantages over hierarchical organizations as the complexity of the network increases, typically as its grows beyond sizes of 15 sensor nodes.

B. Lateral Networks of Autonomous Sensor-Agents

We assume that the network of sensor agents receives a sensing task from an external mandator. The sensor agents then successively agree to form a collective. All of its members are capable of observing the same object feature (or complete object) and were assigned the competence to do so. In principle, each sensor may become a member of any conceivable collective, i.e. a member of whichever collective promises the completion of a certain sensing task. It is assumed that the network forms a grid of K nodes. To establish contact between any two sensor agents of a collective and to coordinate their activities, a bidding scheme similar to the *Contract Net Protocol* introduced in [Davis 83] is utilised (see also [Parunak 89, Ramamritham 89]). In our context, the bidding scheme is applied to the lateral allocation of sensor agents for object identification and localisation tasks. Not only does this scheme enable the dynamic allocation of sensor agents within an individual cooperative, but it can also assign a sensor agent to different cooperatives thus coordinating the activities of overlapping cooperatives. The main benefits of contracting by negotiation in this context are:

a) It enables a sensor network to flexibly adapt to indeterminate environmental and agent specific conditions governing its performance;

b) It also leads to the sequential coordination of only the minimum amount of resources required to solve a task according to given quality constraints.

c) The sensor agents not participating in the processing of a task of a particular collective remain free to employ their resources in other collectives.

d) Due to the lateral relations between sensor agents, sensory results from multiple, possibly disparate sources can be accumulated and integrated.

By contrast, in the hierarchical case a managing agent coordinates only a subset of network capacity for a specific set of tasks. Coordination is efficient only in such a subset because otherwise too many intermediate nodes may be involved, which inhibits tasks from spreading out over the entire network. The relevance of this property increases as the complexity and uncertainty in the network environment grow.

In a concrete implementation and for simulation purposes the status of an agent is described by values expressing its current workload (and sensory precision/variance). The workload is the number of tasks the sensor has successfully bidden for but not yet processed. Due to the generally sporadic time of arrival of individual tasks and the indeterminate amount of time required for their processing, this is an appropriate way of pragmatically measuring the workload of a sensor agent. A *task description* is composed of administrative information, the conditions constraining the cooperative processing of the task as given by a mandator and results generated by sensor agents which have already processed that task. The administrative information consists of a unique task identifier and the communication address of the mandator, both supplied by the agent. The external constraints are composed of a value defining the task processing time limit and the desired quality factors for the object identification and localisation. To facilitate contracting by negotiation among sensor agents, appropriate message types must be defined. In our simulation environment five message types are used:

- A *request for bids*-message describing a task to be processed cooperatively. It initiates a negotiation and selection phase.
- A *bid*-message by which an interested sensor agent offers its capacity to process a task.
- An *award*-message by which an initiating sensor agent transfers task information to the selected bidder.
- A *request for interest*-message by which a sensor agent offers to mandators further processing of a newly arrived task.
- A *result*-message by which a sensor agent currently allocated to a given task returns the available results to a mandator when either the quality requirements for this task have been met or its time limit has expired.

The format of the message types underlying our simulation was specified so as to meet the requirements of a typical system used in robotics to fuse uncertain geometric data acquired from more than one sensor (see [Knoll 93] for details).

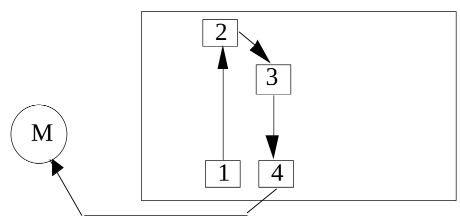

Fig. 1: Multiple agents form a team to process a single specific task. Agent 1 was awarded the task originally, but turned out not to be capable of meeting the requirements.

Fig. 1 shows how a collective of team size $K_S = 4$ agents is assembled if the first agent that was awarded the task turns out to have been too optimistic, i..e. it cannot meet the requirements. It is important to note that, although preferences for choosing particular collectives may exist, the structure of the team is not set a priori.

Evaluating the Performance of Organization Schemes and Selection Strategies

We now turn to the interesting question of how well the architectures perform under several different conditions. The performance of the agent organization is assessed by steady-state simulation, which models an agent network as a set of interconnected service centres equipped with queueing facilities (fig. 2). Each agent is modelled as consisting of two sequentially related service centres, i.e. its local computation component or sensor component (sp) and its coordination component (cp), which controls the interaction between agents. In the context of sensor networks agents without a physical sensor (coordinating agents) lack the sensing component.

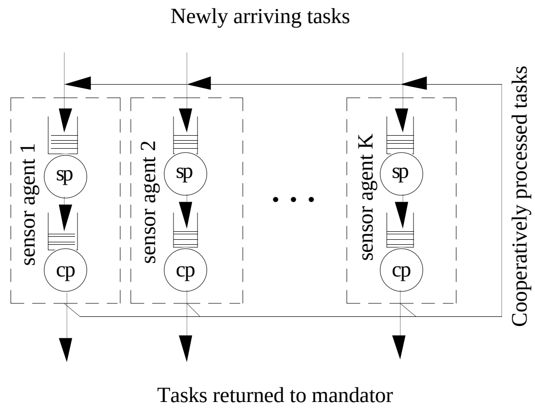

Fig. 2: An MAS as a set of interconnected queues

A. Simulation Model

The sensor component of an agent is represented as an M/M/1-queue, i.e. a service centre with exponentially distributed inter-arrival times of new tasks and exponentially distributed service time. The coordination component is represented as an M/D/1-queue, i.e. with exponentially distributed inter-arrival times and constant service time.

For the purposes of our simulation, a task arriving at an agent is first processed by its sensor component and then by its coordination component. In particular, it is assumed that an external mandator was already located for each newly arriving task. The rate of newly arriving tasks (e.g. due to object movement) λ_i at a sensor agent i with $i = 1,...,$ K is the *external arrival rate* and is assumed to be identical for all agents. Thus, the total external arrival rate is given by $\lambda = \lambda_i K$. A task processed by a sensor agent i is routed to a sensor agent j of the corresponding collective of K_S agents (which are competent to work on the task) with probability q_{ij} where $i, j = 1,\ldots,K_S$. The task exits the network when it was successfully completed with probability

$$q_{i0} = 1 - \sum_{j=1}^{K_s} q_{ij} \quad \text{with } i = 1, \ldots, K$$

The probabilities q_{ij} are the network routing probabilities. The tasks arriving at agent i from other agents j (because of contracting) are a fraction of the total rate of tasks γ_j leaving sensor agent j with $j = 1, \ldots, K_S$. The rate of traffic flowing into agent i is called the *internal arrival rate* of agent i and is given by

$$\sum_{j=1}^{K_s} \gamma_j q_{ji} \quad \text{with } i = 1, \ldots, K$$

Due to the flow balance assumption for queueing systems tasks must *leave* a sensor agent at the same rate at which they *arrive* there. A fraction q_{ij} of the set of tasks arriving at agent i is directed from sensor agent i to sensor agent j with the rate $\gamma_i q_{ij}$. Furthermore, a fraction q_{ji} of tasks is directed from sensor agent j to sensor agent i with the rate $\gamma_j q_{ji}$. Consequently, the total traffic rate γ_i at a sensor agent i is given by the *network traffic equations* (see fig. 3):

$$\gamma_i = \lambda_i + \sum_{j=1}^{K_s} \gamma_j q_{ji} \qquad i = 1, \ldots, K$$

The external arrival of tasks is assumed to be a stationary Poisson process. However, the internal arrival rate is not necessarily such a process: in the case of a dynamic selection strategy (such as selection by smallest workload) arrival rates depend on system history. Moreover, the probability q_{ji} of a task arriving from sensor agent j at sensor agent i is a function of the number of sensor agents which have already processed this task. Further performance evaluation is carried out by means of simulation because the non-Poisson characteristics of the processes significantly complicate an analytical approach.

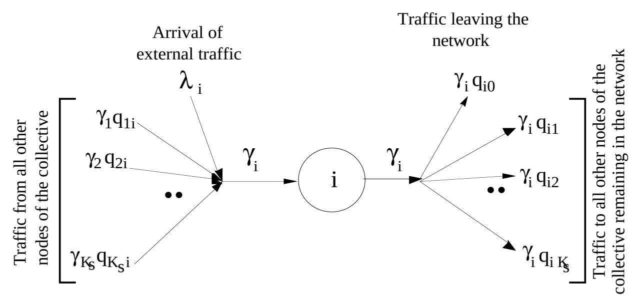

Fig. 3: Task traffic at sensor agent i

The coordination component of an agent decides by means of an evaluation function whether a task processed by the sensor component can be successfully completed. As no further assumptions on the nature of sensor data evaluation were made, the process of K_S agents transferring a given task and accepting it for completion or rejecting it, is viewed as a Bernoulli experiment. After each transfer the task is accepted by the new agent with probability b and rejected with probability $1 - b$. The probability of the k^{th} agent accepting the task for completion is then given by

$$P(k) = (1 - b)^{k-1} b$$

The corresponding geometric probability distribution is given by

$$F(k) = 1 - (1 - b)^k$$

This function determines the probability q_{i0} of a task exiting the network as successfully completed by sensor agent i after passing k agents including i ($k \geq 1$) with $E[k] = 1/b$. Additionally, q_{i0} is set to 1, should the set processing deadline have expired at the time a task arrives. Based on these assumptions, hierarchical and lateral structures were compared in performance. For appropriate performance comparison an extended hierarchy was used which consists of two additional layers of sensor agents (manager-agents) with the original grid of K agents constituting the lowest level. Each agent at the middle level coordinates exactly one row of the lowest agent grid. The middle level agents are coordinated by a single manager-agent at the top level (fig. 4). At the middle and top level a task is processed only by the coordination component of a manager-agent. Specifically, a sensor agent at the middle level coordinates only a subset of the collectives defined by its subordinate agents. The main simulation output parameter of interest and hence the measure of organization performance used for comparison is the percentage *V* of tasks successfully completed within a given deadline. The following variables were among the simulation input parameters, which were introduced to determine the behaviour of the modelled organization and, consequently, the value of V: The *network size K* defining the number of sensor agents in the network; a relative *processing deadline d* within which tasks should be completed; a *failure probability f* which determines whether a sensor agent fails at a specific point in time; a *repair delay r* after which failed sensor agents return into the system and a *completion probability b* determining the mean number of sensor agents required to successfully complete a task.

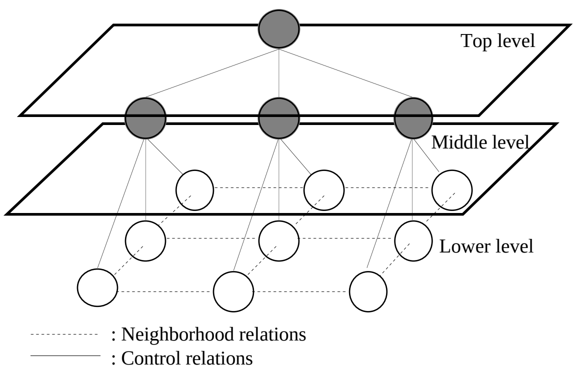

Fig. 4: Hierarchical network used in the comparison

B. Simulation Results – Structures

As a measure of difficulty of the task, the coefficient *b* was varied, a decrease in *b* resulting in an increase in E[*k*], the mean number of sensor agents necessary to successfully complete a task. The node failure probability *f* and repair delay *r* as well as the network size *K* are viewed as measures of complexity. Additionally, the coordination component service time *cp* was varied to represent an increasing complexity in reaching coordination decisions as opposed to the sensor component service time *sp*. For reasons of simplicity the arrival rate λ_i at nodes *i* was assumed to be identical for all nodes and is called *p*.

Figs. I…II depict the effect of increasing the network size *K* while the other parameters remain fixed, except for the failure probability *f*, which was 0.01 and 0.05, respectively. In addition, organization performance (*fl* denotes a lateral and *hi* a hierarchical organization) is shown for different degrees of task uncertainty *b*. In the case of low failure probability *f* (fig. I), the superiority of lateral over hierarchical organization is evident because even with small network sizes the lateral organization provides a higher percentage *V* of tasks completed successfully within the deadline *d*. This advantage increases with growing network size *K* and uncertainty *b*. Particularly, it is shown that for a large network (K = 49) an increase in uncertainty leads to significantly less degraded performance when compared to the hierarchical organization. In this case, with *b* decreased from 0.5 to 0.33 (and hence E[*k*] increased from 2 to 3) the lateral organization suffers from a performance degradation of ca. 5% (as given by the parameter *V*), whereas the hierarchical organization performance degrades by approximately 20% under identical conditions. A further increase in complexity (high failure probability *f* = 0.05; fig. II) yields an important result: initially, i.e. with small network sizes, the hierarchical organization exhibits a better performance than the lateral organization. However, as *K* increases, a break-even point is reached, at which the lateral organization performance exceeds that of the hierarchical organization. Moreover, as uncertainty increases, this break-even point occurs at decreasing network sizes. At first sight, this fact may look contradictory to our argumentation; note, however, that the difference in performance in the two organization types grows with increasing difficulty as the network size increases.

The results displayed in figs. I…II are supported by figs. III…IV, which show the organization performance as a function of the repair delay *r* with high failure probability *f* fixed at 0.05. The repair delay corresponds, for example, to the time it takes to re-focus a sensor if the current focus turns out to be inadequate. The probability *f* indicates how frequently this happens. Initially, with a small network (*K* = 9) and short repair delay, lateral organization is at an advantage over hierarchical organization. Soon, however, with increasing repair delay, lateral organization performance degrades significantly below hierarchical organization performance (fig. III). This situation is completely different in the case of a large network (*K*= 49; fig. IV): Here, even with very long repair delays *r*, the lateral organization significantly outperforms the hierarchical organization. It is also clearly visible that the difference in organization performance increases with growing uncertainty. This sensitivity of the hierarchical organization to increased network size is explained by the bottleneck effect affecting managing agents (see also [Knoll 93]).

A different measure of complexity is the amount of time required by a coordination component to reach a coordination decision. It was considered for varying circumstances and the corresponding results are displayed in fig. V. The network size has no significant effect on lateral organization performance, but very much on hierarchical organization performance. Here, another interesting feature of hierarchical organization performance was encountered: As *cp* is increased (and *sp* correspondingly decreased), lateral organization performance remains relatively stable, rising from nearly 100% to a full 100% of successfully completed tasks. This is largely due to the growing influence of the constant service time *cp* and, correspondingly, the diminishing influence of the exponentially distributed service time *sp*. This is a relevant setting for networks that consist of a large number of coordinators (that do not have a sensing component). However, besides being sensitive to increased network size due to bottleneck potential, hierarchical organization performance rises sharply with increased service time *cp*. It reaches an optimum in the vicinity of the lateral organization performance, and declines just about as sharply as it has risen before reaching the optimum. The results displayed in fig. V suggest that, in contrast to the robustness of lateral organization performance, a hierarchical organization is highly sensitive to the relation of *sp* to *cp* service times. Thus, a hierarchical organization is only justifiable if this relation results in optimal or near-optimal performance. It seems to be increasingly difficult to establish the according range of service time values guaranteeing such performance with increasing network sizes. The right-hand sides of the performance plots of the hierarchical organization are again explained by the bottleneck characteristic of managing agents which is directly amplified by increasing *cp*. The left hand-sides, however, are not so easy to explain: With decreasing *cp* service times, tasks may flow increasingly faster through the hierarchy, leading to saturation effects in the collective subsets coordinated by the middle level managers. This presumption is supported by fig. VI, which displays the mean task population of hierarchical organizations with *cp* service times corresponding to those shown in fig. V. This saturation effect within the collective subsets diminishes with increasing service time *cp* until the performance reaches the optimum, and is afterwards converted to the bottleneck effect mentioned above. The performance increase occurring with increasing service time *cp* is explained by the fact that the traffic originally (with very low *cp*) leading to neighbourhood saturation is increasingly delayed at the corresponding middle level manager. This increase in delay has an advantageous effect on hierarchical organization performance until it reaches the point where the bottleneck effect induced by that delay outweighs this advantage.

C. Simulation Results – Selection Strategies

As mentioned above, besides the network structure the other important distinguishing feature determining the throughput is the selection strategy by which mandating agents choose their cooperation partner. Three different such strategies were compared: conservative *cnsvt*, random *rndm* and shortest queue *shtq*. A further parameter m was introduced to model communication delays inherent to real world communication subsystems and the time it takes to transmit request-for-bids, bidding and award messages. A second (even more important) additional parameter Δ represents updating intervals if the current load information of individual agents is sampled by the communication system only at specific points in time. Figs. VII a…c show the absolute task throughput X for the three strategies as a function of the node arrival rate p with the paramter d (admissible deadline) for a system with negligible communication delay m and continuously available load information ($\Delta = 0$). The conservative strategy, though simple and requiring only minimum overhead, performs particularly badly in the medium load range $p \geq 0.4$. Even very late deadlines ($d = 15.0$) cannot be kept because the network is already in some saturation. The random strategy performs slightly better but cannot compete with the performance of selection by shortest queue in this setting. In the case of *shtq* the rate of successfully accomplished tasks is roughly equal to the rate of incoming tasks $\lambda = p \cdot K$ as indicated by the dashed line in fig. VIIc. This implies that the mean delay time in the agent queues vanishes, i.e. this represents the theoretical maximum.

Fig. VIII shows the effects of a finite updating interval $\Delta > 0$. The status information the initiator has available on the load of its potential cooperation partners is on average $\Delta/2$ time intervals old. Selections are always based on obsolete information. The network designer must therefore know how old this information may become before a significant degradation of throughput is observed. In Fig. VIII the percentage V is shown depending on Δ and the size K. It is surprising at the first glance and important to note that for large values of Δ the performance of *shtq* falls below that of *rndm*. The use of a load-depending strategy is no longer justifiable if it must be based on unreliable information. The comparison between *shtq* and *rndm* in further simulations revealed that the sensitivity of output parameters to a large V decreases as the load increases so that it may still be advantageous to use *shtq*. This, however, can only be studied in detail when the crucial parameters are known for a given application.

Conclusions

The subject of a statistical analysis of agent coordination and control may currently appear esoteric, but it will soon turn out to be quite relevant as the complexity of these systems continues to increase and the prevailing ad hoc approaches will no longer provide adequate solutions. This does not imply, however, that systems with a smaller number of agents, such as sensor systems on current mobile robots, could not profit from a well-structured organization and coordination of their agent system. It was outlined that lateral control in distributed sensor networks is feasible through a corresponding cooperation protocol motivated by considering models from organization theory. Furthermore, simulation studies have revealed not only a general quantitative superiority of lateral over pure hierarchical control structures, but also an increased sensitivity of hierarchical organizations to growing complexity and uncertainty when compared to lateral organizations. However, a clear distinction in performance between lateral and hierarchical organization will not be possible without detailed and comprehensive experimenting by simulation *and* real multi-agent systems. In fact, the simulation results presented here indicate that issues of complexity and uncertainty are closely coupled and can not be studied in isolation.

References

Carver, N., Lesser, V., 1993."Distributed sensor interpretation: Modeling agent interpretations in DRESUN", Comp. Sc. Tech. Report 93-75, University of Massachusetts, Amherst

Davis, R., Smith, R. 1983."Negotiation as a metaphor for distributed problem solving", *Artificial Intelligence 20 (1)*

Demazeau, Y. 1994."From feature extraction to integration of visual modules using agent systems", *Proc. Workshop "Solving Complex Problems with Multi-Agent Systems"*, Univ. of Bielefeld

Fox, M. 1981."An organizational view of distributed systems", *IEEE Trans. on Systems, Man and Cybernetics, SMC-11*

Henderson, T., Fai, W., Hansen, C. 1984."MKS: A multisensor kernel system", *IEEE Trans. on Systems, Man and Cybernetics,* SMC-14

Iyengar, S., Kashyap, R., Madan, R. Thomas, D. 1990. "A tree-structured sensor fusion architecture for distributed sensor networks", *SPIE Vol. 1306 Sensor Fusion III*

Iyengar, S., Sharma, M., Kashyap, R. 1992."Information routing and reliability issues in distributed sensor networks", *IEEE Trans. on Signal Processing*, Vol. 40, No. 12, Dec. 1992

Knoll, A., Meinkoehn, J. 1993."Recent results in Organization and Performance Evaluation of Large Sensor Networks", *Proc. Third Int. Conf on Intell. Auton. Syst.*, IAS-3, Pittsburgh, Feb. 1993

Mitrani, I. 1987."Modeling of computer and communication Systems", Cambridge Computer Science Texts 24, Cambridge University Press

Pete, A., Pattipati, K., Kleinman, D. 1993."Distributed detection in teams with partial information: A normative descriptive model" *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 23, No.6, Dec. 1993

Parunak, H. 1989."Manufacturing experience with the contract net", in: Michael Huhns (Ed.), "Distributed Artificial Intelligence", Pitman

Rao, B., Durrant-Whyte, H., Sheen, J. 93."A fully decentralized multi-sensor system for tracking and surveillance", *Int. J. Rob. Res.*, Vol. 12, No. 1, Feb. 1993

Ramamritham, K., Stankovic, J., Zhao, W. 1989. "Distributed scheduling of tasks with deadlines and resource requirements", *IEEE Transactions on Computers 38 (8)*, 1989

Figures

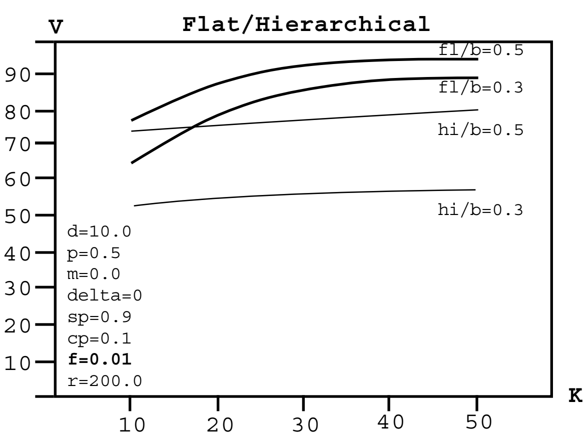

Fig. I. Effect of network size *K* on *V* with varying uncertainty (completion probability) *b*. Low failure probability *f* and long repair delay *r*.

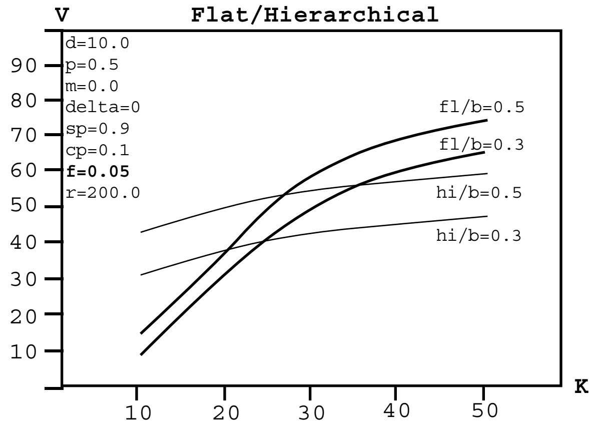

Fig. II. Effect of network size *K* on *V* with varying uncertainty (completion probability) *b*. High failure probability *f* and long repair delay *r*.

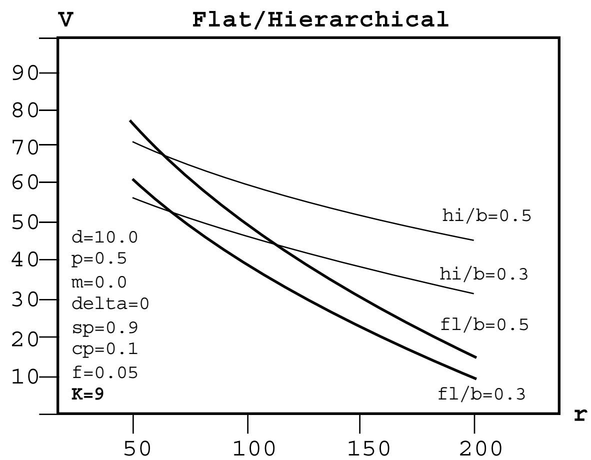

Fig. III. Effect of repair delay *r* ("re-focus delay") on *V* with varying uncertainty (completion probability) *b*. High failure probability *f* and small network size *K*.

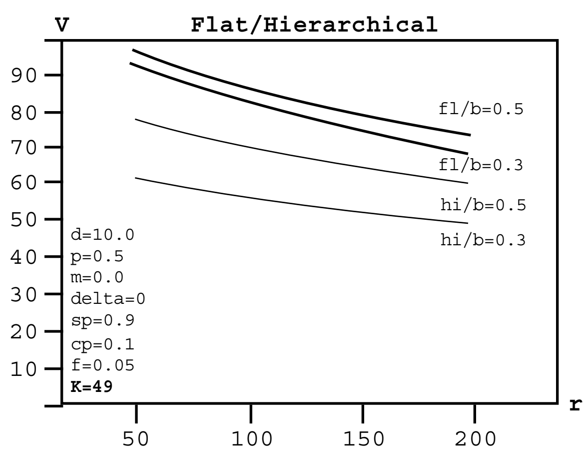

Fig. IV. Effect of repair delay *r* on *V* with varying uncertainty (completion probability) *b*. High failure probability *f* and large network size *K*.

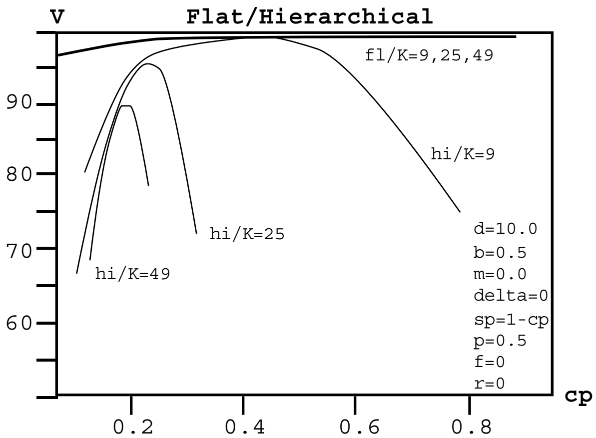

Fig. V. Effect of coordination component service time *cp* on *V*

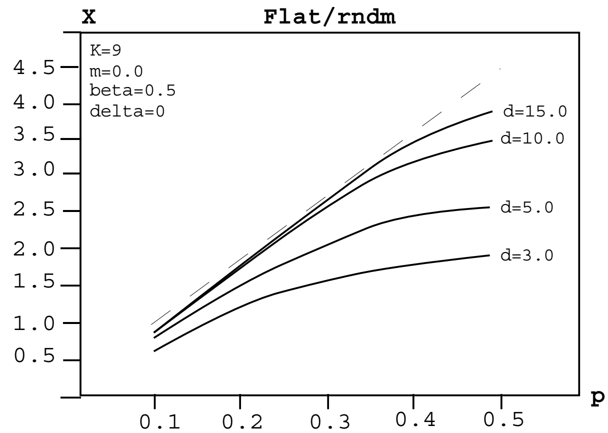

Fig. VIIb. Absolute number of successfully processed tasks. Random Selection strategy.

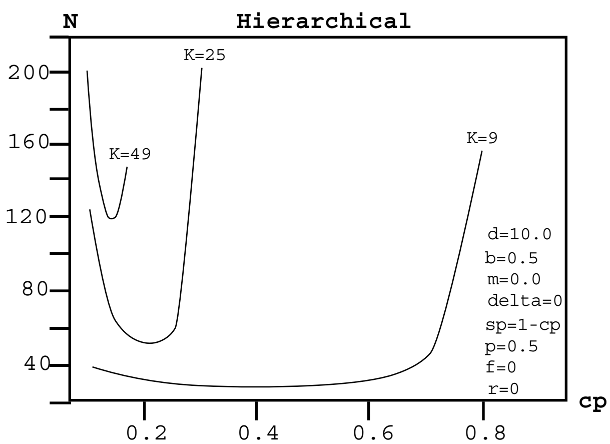

Fig. VI. Effect of coordination component service time *cp* on population *N* (=tasks waiting to be serviced)

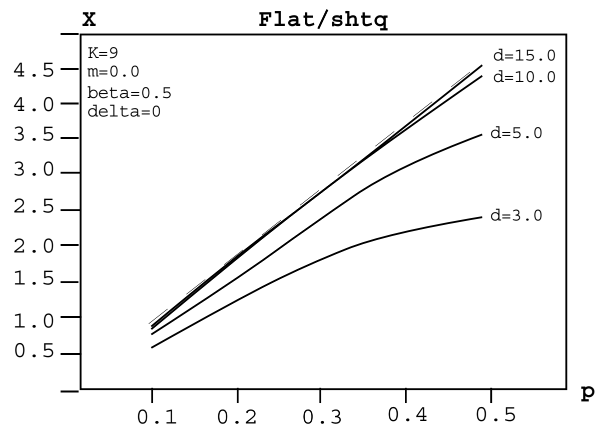

Fig. VIIc. Absolute number of successfully processed tasks. Selection by shortest queue

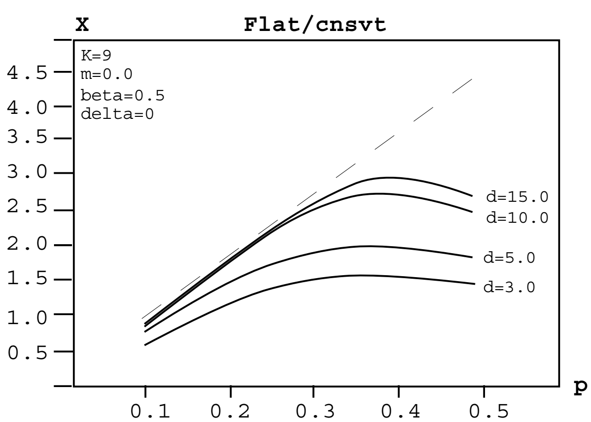

Fig. VIIa. Absolute number of successfully processed tasks. Conservative strategy. Dashed line: Max. poss. throughput

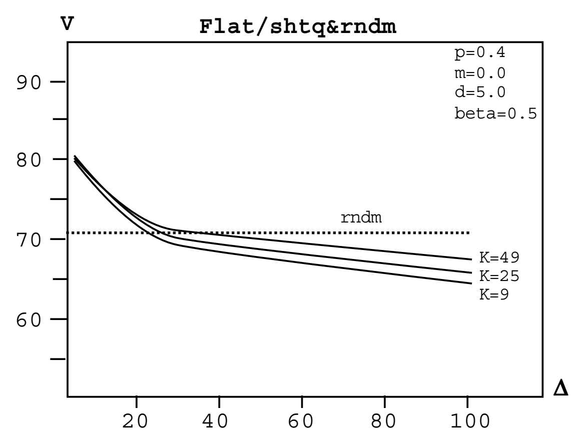

Fig. VIII. Percentage of successfully completed tasks in a lateral network depending upon the updating interval Δ for the selection strategies random and shortest queue.