The Impact of Vertical Specialization on Hierarchical Multi-Agent Systems

Steven Okamoto
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
sokamoto@cs.cmu.edu

Paul Scerri
Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
pscerri@cs.cmu.edu

Katia Sycara
Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
katia@cs.cmu.edu

Abstract

Hierarchies are one of the most common organizational structures observed in multi-agent systems. In this paper we study vertical specialization as a reason for hierarchical structures. In vertically specialized systems, more highly skilled agents are also more costly. By using less capable agents to initially process tasks and forwarding only exceptional tasks to more capable agents, such systems may be able to economize on the number of highly capable agents. The result is a hierarchical structure with least capable agents at the bottom. However, such a structure increases the delay in completing some tasks, because they must pass through multiple levels of control. Thus, vertical specialization presents a tradeoff between economizing on skilled agents and increasing task completion time. We find that for a wide range of settings, vertical specialization induces an optimal hierarchy of height at most three. This suggests that a multi-agent system designer interested in exploiting vertical specialization needs to use at most three levels of specialization in order to reap most of the benefits.

Introduction

Emerging, exciting applications envision using large numbers of robots or software agents to perform a range of dull, dangerous or dirty tasks (Ortiz, Vincent, & Morisset 2005). In many of these applications, the same basic task will vary in difficulty from instance to instance, e.g., calls to a call center or computer help desk will vary in difficulty from routine to highly problematic, as will search or mapping tasks for mobile robots. Vertical specialization is a technique where the overall cost of an organization is reduced by employing many cheaper, less capable agents to handle routine task instances and few highly capable agents for the more problematic task instances. For robot teams or sensor networks, higher capability may reflect additional or more resource-intensive sensors or effectors, with corresponding higher cost. Thus, intuitively, composing a vertically specialized multi-agent or multi-robot team will reduce deployment cost, while maintaining effectiveness.

Specialized agents are often organized so that tasks are first attempted by less capable agents, passing to more capable agents only upon failure. The result is effectively a vertical, hierarchical organization, consisting of layers of equally capable agents, with least capable agents at the bottom accepting all incoming tasks, passing tasks they are incapable of completing up the hierarchy, until they are successfully performed by a sufficiently capable agent. A variety of multi-agent systems have been organized in this way, such as multi-robot teams, information retrieval systems (Zhang & Lesser 2004), manufacturing systems (S. Bussmann & Wooldridge 2004), and distributed sensor networks (Kulkarni *et al.* 2005). Designers of such systems face a tradeoff between two basic constraints. On the one hand, costs can be reduced by more specialization by minimizing the capability of the agent dealing with a task of a particular level of difficulty. On the other hand, since less capable agents must fail before more capable agents attempt a task, greater specialization leads to longer delays for more difficult tasks. The key question addressed by this paper is how to optimally tradeoff between the cost benefits and delay penalties of vertical specialization, i.e., what is the optimal height of a vertically specialized hierarchy?

The tradeoff of vertical specialization in human organizations was elegantly modeled by Beggs (Beggs 2001) with two kinds of costs. The *running cost* is incurred for each agent depending on its skill, and is the cost that vertical specialization reduces. In any case where vertical specialization might be useful, more capable agents are more expensive. The second cost in Beggs's model is a *delay cost* incurred for each completed task depending on the time it took for the task to be completed, and is the cost that inhibits arbitrarily fine specialization. Building on Beggs's model, this paper has two main contributions. First, we apply Beggs's model of vertical specialization to multi-agent systems and explore its implications. Second, we consider three additional rationales for using a hierarchical organizational structure and investigate how these impact the tradeoff for the optimal hierarchy height.

To apply Beggs's model to multi-agent systems, two simplifying assumptions are relaxed. We relax the assumption that the number of agents at a particular skill level may be non-integral. Second, we relax the assumption that the cost of delay is linearly increasing. In many environments, the cost of delay is highly non-linear; e.g., as a building fire spreads, the cost for additional delay in locating trapped victims increases dramatically. Numerical analysis of the resulting model across a wide range of running cost and delay cost functions shows that overwhelmingly, no more than two or three specialization levels provides an optimal tradeoff. This is very surprising, because many human organizations have much finer specialization. Thus, influences other than running cost and delay cost must be responsible for the taller hierarchies observed in human organizations. If those additional influences are not relevant to agent organizations, it is possible for multi-agent system designers to focus exclusively on at most three levels of specialization.

We investigated a number of factors that influence the size of human hierarchies and that may also influence agent or robot vertical specialization. First, we considered agent failures, where failed agents must be replaced through promotion of existing agents or the introduction of new agents, both of which impose replacement costs on the organization. Second, we considered agents that learn, thereby becoming more skillful at handling incoming tasks. Third, we generalized the nature of hierarchical interactions to accommodate cases where agents higher in the hierarchy exert some form of supervision over their subordinates.

Numerical analysis of a model including these additional factors again showed that there is little or no value to more than three specializations or hierarchy levels. This finding is robust even when the system size is scaled to hundreds of agents and holds across a wide range of cost functions. While the reported findings assume a continuous tradeoff between capability and cost, these results are applicable to embodied systems where agent capabilities are often limited to a discrete set based on available hardware. This is simply because drawing from the entire continuum of agent capabilities constitutes an upper bound on the optimal number of layers in any hierarchy based on vertical specialization.

Problem Statement

In Beggs's model, an organization is modeled as an open queueing network, where each agent can process a single task at a time and has a FIFO queue for buffering pending tasks. The organization incurs costs to successfully complete tasks of varying difficulties that arise dynamically. If the average task completion rate exceeds the average task arrival rate at each agent, the distribution of tasks in the organization will converge to a single invariant steady state distribution. The organizational performance is measured by the expected cost incurred per unit time in this steady-state equilibrium, which provides a robust measure of long-term performance independent of initial conditions.

Formally, an organization $O = \langle \mathcal{L}, \lambda, \mu, K, \phi \rangle$, where

- $\mathcal{L} = \{L_1, \ldots, L_n\}$ the set of n layers that partitions the agents. Each layer $L_i = \langle N_i, x_i \rangle$ has $N_i > 0$ agents, each with skill $x_i \in [0, 1]$.
- λ the Poisson arrival rate of tasks to the organization.
- μ the exponential service rate of agents performing tasks.
- $K : [0, 1] \to \mathbb{R}^+$ the running cost function. For each agent with skill x, the organization incurs a cost per unit time of $K(x)$.
- $\phi : \mathbb{R}^+ \to \mathbb{R}^+$ the delay cost function. For each successfully completed task, the organization incurs a cost of $\phi(t)$, where t is the amount of time since the task first arrived to the organization.

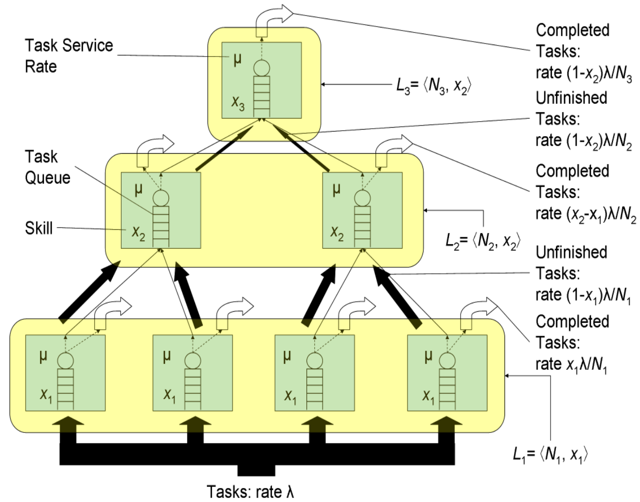

Figure 1: *Tasks arrive at the bottom of the hierarchy to the least skilled agents. At each level, some of the tasks are successfully completed, while the remaining tasks are passed to the next layer, distributing the load evenly. The narrowing black arrows indicate the reduction in flow of tasks as the height in the hierarchy grows. The top layer is required to complete all remaining unfinished tasks.*

The agents are organized into layers of successively greater skill. Tasks arrive at the bottom layer (L_1), with each task characterized by a difficulty $d \in [0, 1]$ drawn independently and uniformly at random. An agent must have skill $x_i \geq d$ to successfully complete a task of difficulty d. If an agent fails to successfully complete a task, the task (with no partial results) is routed to an agent in the next layer.

The organization must be able to process all incoming tasks in finite time. This imposes the following constraints:

- $x_n = 1$. The top-most layer must be able to process all remaining tasks.
- $\lambda_i < N_i \mu$ for all $1 \leq i \leq n$, where λ_i is the arrival rate to layer i defined by

$$\lambda_i = \begin{cases} \lambda & \text{if } i = 1 \\ \lambda(1 - x_{i-1}) & \text{if } 1 < i \leq n \end{cases} \tag{1}$$

This is a stability condition, guaranteeing that no agent's queue grows unboundedly.

The total number of agents must be at least $\lceil \rho \rceil$, where $\rho = \lambda/\mu$ is the loading ratio. This follows immediately from the fact that the lowest layer in the organization must contain at least $\lceil \rho \rceil$ agents to satisfy the stability requirement.

The total per unit time cost in steady-state equilibrium is

$$C = \lambda E_t[\phi(t)] + \sum_{i=1}^{n} N_i K(x_i) \tag{2}$$

where $E_t[\phi(t)]$ is the expected value of $\phi(t)$.

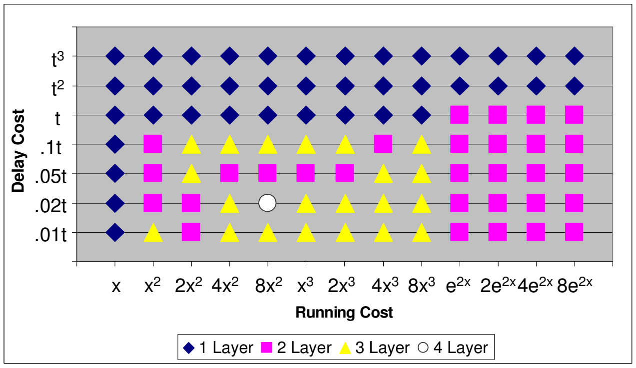

Figure 2: *Number of layers in the optimal hierarchy for varying cost functions. Delay costs increase along the y-axis, and running costs increase along the x-axis. Diamonds, squares, triangles, and circles indicate optimal organizations with one, two, three, and four layers, respectively.*

The vertically specialized hierarchical design problem is to choose the number of layers n and the sizes and skills of the layers N_i and x_i for $1 \leq i \leq n$ such that the system is stable and C is minimized. Generally it is not possible to find analytic solutions because calculating the expected delay cost for non-linear ϕ requires higher moments of the task delay distribution, a notoriously difficult problem in queueing networks. The problem is further complicated by the requirement for integral N_i.

Organization Evaluation and Design

Expected organizational costs cannot be determined analytically in the adapted model. We evaluated costs using Monte Carlo simulation, with the steady-state empirical average as an approximation of the expected value. Organizations were simulated through a warmup period in which 20,000 tasks were completed to reach the steady-state, then the costs were averaged over the next 50,000 tasks completed. The average over 10 repetitions was used to approximate performance, with standard deviations well below 0.1%.

To solve the vertically specialized design problem, we used a genetic algorithm that searches the space of possible organizations for a fixed number of layers. We ran the genetic algorithm for $n = 1, \ldots, 5$ layers, finding the best organization for each number of layers. The genetic algorithm took as input the number of layers n, the running cost K and delay cost ϕ functions, and the arrival and service rate parameters λ and μ.

Experimental Results

We conducted a large number of experiments over a wide range of organization parameter settings. Due to space constraints we present only a representative subset of those results here. The results presented here were obtained with a loading factor of $\rho = 266.67$, so that each organization contained at least 267 members. These results are similar to those obtained for larger and smaller values of ρ.

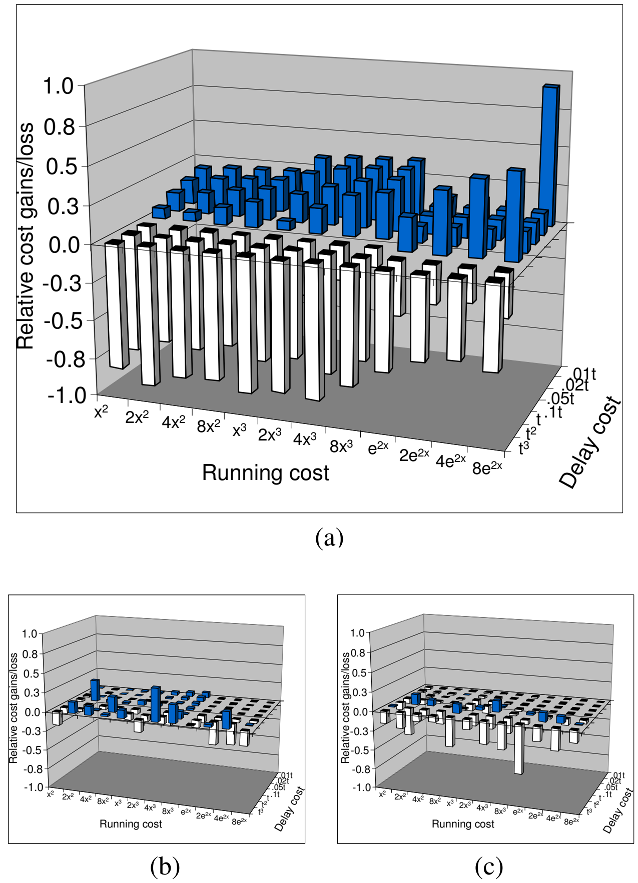

Figure 3: *Relative marginal cost savings in increasing the height of the hierarchy. Unshaded bars indicate worse performance, shaded bars indicate better performance. (a) Savings of 2 layers over 1 layer. (b) Savings of 3 layers over 2 layers. (c) Savings of 4 layers over 3 layers.*

For most running cost and delay cost functions, we found that the optimal hierarchy had a height of no more than three. Figure 2 is a scatter-plot showing the number of layers in the optimal hierarchies for different K and ϕ. In the Figure, K increases from linear to exponential on the x-axis and ϕ increases from linear to cubic on the y-axis. The border of diamonds along the left shows that one-layer organizations are optimal when K is linear. The diamonds along the top show that one-layer organizations are also optimal when ϕ grows quickly, as in quadratic or cubic ϕ, because the high delay cost penalizes taller hierarchies. Aside from the extreme top and left of the Figure, most of the points indicate that the optimal organization has two or three layers.

While the optimal number of layers was varied and sometimes three or even four layers was optimal, the additional reduction in cost is very small. An example of this is the lone optimal hierarchy with four layers, which had $\mathcal{L} = \{\langle 279, 0.239\rangle, \langle 205, 0.685\rangle, \langle 84, 0.997\rangle, \langle 1, 1.0\rangle\}$. In comparison, the best three-layered hierarchy for the same K and ϕ had $\mathcal{L} = \{\langle 273, 0.510\rangle, \langle 131, 0.996\rangle, \langle 1, 1.0\rangle\}$, and cost only 0.1% more than the optimal four layer hierarchy.

Figure 3 shows the relative marginal cost savings obtained by increasing the height of the hierarchy from one layer to two layers (Fig. 3(a)), two layers to three layers (Fig. 3(b)), and three layers to four layers (Fig. 3(c)), for the cost functions in Figure 2. Intuitively, positive values show the fraction of cost that can be saved by adding a layer, while negative values show the fraction of cost that can be saved by removing a layer. Figure 3(a) clearly shows that the difference in performance is most significant between one layer and two layers, with the direction of change primarily dependent on the delay function. In comparison, Figure 3(b) is mostly flat, indicating that two and three layered hierarchies tend to perform very similarly. Figure 3(c) is also mostly flat but with a greater number of negative values, reflecting the finding that organizations with four layers tend to perform worse than those with three layers.

From Figures 2 and 3 it is apparent that ϕ has the greatest impact on the height of the optimal hierarchy. The rate of growth of linear ϕ is strongly correlated with flatter optimal hierarchies ($r = -0.8342$ for quadratic and cubic K), because high delay costs penalize tall hierarchies. For quadratic and cubic K, there is also a weaker trend for faster-growing K to favor taller hierarchies ($r = 0.1764$ for linear ϕ), because faster-growing K provide more opportunity for vertical specialization to reduce running costs. However, the smaller magnitude of correlation between the optimal height and K compared to the magnitude of correlation between optimal height and ϕ suggests that the cost savings that can be obtained through vertical specialization is limited, even for linear ϕ.

Extensions

Because cost savings due to vertical specialization are not sufficient reason to design systems with many specialization levels, we explored other reasons inspired by human organizations. If these factors were responsible for taller hierarchies in human organizations, then they may affect the optimal specialization hierarchy in multi-agent systems as well. The three extensions described in this section, agent failure, agent learning, and agent supervision, are representative of the many extensions we tried.

Agent Failure

One possible reason for taller hierarchies in human organizations is that they allow workers to be promoted to higher positions, which reduces the frequency of resignations and reduces turnover costs. Agent failures in multi-agent systems play a similar role to worker resignations, as failed agents must be replaced at some cost. If reducing turnover is a reason for tall human hierarchies, reducing agent failures may also affect optimal agent specialization.

We extend the model to include a probability of an agent failing that is dependent on the length of time it has spent in its current layer, which indicates the length of time that an agent is using its current capabilities in its current role; e.g., hardware wear-and-tear starts only when the agent is outfitted with the new hardware. We assume that agents fail according to a Weibull distribution with scale parameter $\alpha > 0$ and shape parameter $\beta > 0$. Weibull distributions are commonly used to model failure rates that increase ($\beta > 1$), decrease ($\beta < 1$), or stay constant with time ($\beta = 1$). The results presented here used an increasing failure rate $\beta = 2$ (e.g., wear-and-tear on robots) with $\alpha = 30$, so that for each agent, the mean time before failure was 26.6, where $\mu = 1$.

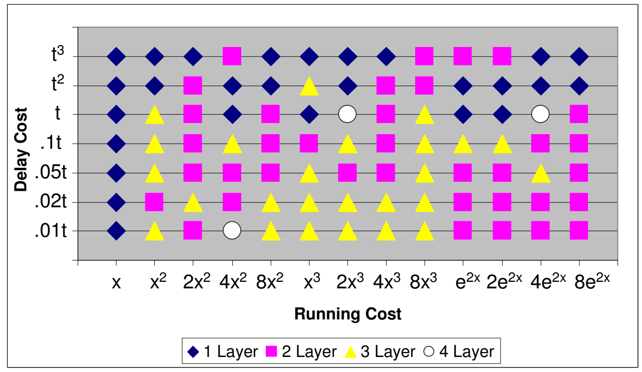

Figure 4: *Number of layers in the optimal hierarchy with agent failure.*

Replacing a failed agent is achieved by promoting an agent from a lower layer. The replacement must be trained from a starting skill x to the skill y of the agent being replaced, which imposes a *training cost* given by the function $\mathcal{T}(y, x)$, e.g., the cost of acquiring a new robot or upgrading an existing sensor. If the failed agent was in the bottom layer, an agent is brought in from outside of the organization, and is assumed to have a starting skill of $x = 0$.

Figure 4 shows the number of layers in the optimal hierarchy under agent failure with $\alpha = 30$, $\beta = 2$, and $\mathcal{T}(y, x) = 10(y - x)^2$. Even with agent failure, hierarchies with three or fewer layers predominate. The addition of the training cost diminishes the relative weights of the running and delay costs to the total cost. This is reflected in a reduced correlation between ϕ and optimal height ($r = -0.262$ for quadratic and cubic K) and between K and optimal height ($r = 0.052$ for quadratic and cubic K).

Agent Learning

Human organizations often reduce their training costs by promoting experienced workers who have already learned some of the skills necessary to conduct their new job. A similar process can work in multi-agent systems, if agents learn as they process tasks. This is most common with software learning, but is applicable for both software and embodied agents: while robots will not "learn" new hardware, most sensors and effectors require online calibration before they can be used to full effect.

In the second extension to the basic model, agents learn as they process jobs. This extension is used in conjunction with the first extension. In addition to its skill x, every agent is also assigned a *potential skill*, denoted χ, which is initially equal to x, and increases with time. Each agent performs tasks at its actual skill x and the organization incurs running costs dependent on x, but χ is used when determining train-

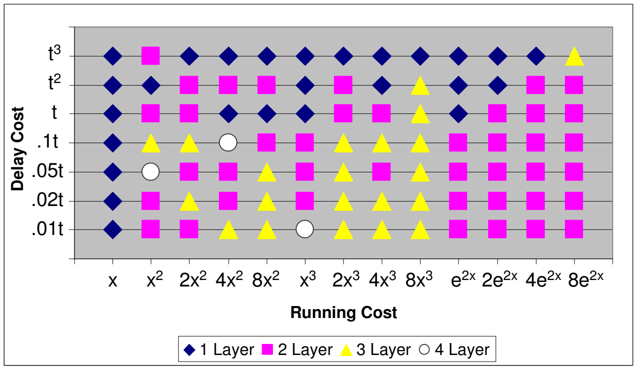

Figure 5: *Number of layers in the optimal hierarchy with agent learning.*

ing costs. Intuitively, this extension may lead to taller optimal hierarchies by reducing training costs, because agents can gain "free training" through learning.

For the results presented here we assume that χ increases linearly with time from the skill of the agent's current layer to the skill of the next layer. This is captured by the formula

$$\chi_t = x_i + \min\{\gamma t(x_{i+1} - x_i), x_{i+1} - x_i\}, \qquad (3)$$

where t is the time the agent has spent at layer i and $\gamma \geq 0$ is the *learning rate*. After $1/\gamma$ time, the agent's potential skill will be equal to the skill of the next layer in the hierarchy.

Figure 5 shows the number of layers in optimal hierarchies with agent learning and $\gamma = 0.0333$. Optimal hierarchies predominantly have at most 3 layers, and even when 4 layers are optimal, there is very little cost saved by using a fourth layer. The greatest cost savings of a four-layered optimal hierarchy over a three-layered hierarchy is only 1.225%.

Agent Supervision

One primary reason hierarchies exist in human organizations is to provide supervision, thus maintaining the productivity of subordinates. The effectiveness of supervision is dependent on how much attention the supervisor can devote to his or her subordinates. In agent organizations, more capable agents may monitor less capable agents. For example, a robot with stereo cameras may provide visual assistance to a robot with only a laser range-finder, helping to speed up a search task of the less capable agent, or a matchmaker agent may monitor requests to an information agent and direct it to a correct information source. We abstractly model the many ways one agent could supervise another by modulating the task processing rate of the subordinate. If supervision has a major effect on the height of human hierarchies, we might expect it to have a similar effect on agent hierarchies.

In the third extension to the basic model, we assume that superiors not only process exceptional tasks from their subordinates, they also interact with subordinates in other ways that directly affect the rate at which subordinates process tasks. The *supervision factor* σ describes whether the effect is positive, negative, or neutral. A positive effect ($\sigma > 0$) increases the service rate of the subordinate and can be used to model beneficial high-level guidance or control that is traditionally associated with hierarchical structures. A negative effect ($\sigma < 0$) slows the service rate of the subordinate and can reflect additional overhead that the subordinate suffers, e.g., when a camera-less robot must wait for visual confirmation. A neutral effect ($\sigma = 0$) has no impact on the service rate and is equivalent to the basic model.

We assume that each agent has a fixed and limited amount of attentional capacity to devote to all of its subordinates, and that it splits this capacity evenly among all subordinates. We denote the sets of superiors and subordinates of an agent a by $\text{Superiors}(a)$ and $\text{Subordinates}(a)$. The total amount of attention that an agent a receives from all of its superiors is termed the *oversight* of a, and is defined as the sum of the attention that it receives from each of its superiors. This in turn has an effect on the agent's service rate, given by

$$\mu_a = \mu + \sigma \sum_{s \in \text{Superiors}(a)} \frac{1}{|\text{Subordinates}(s)|} \qquad (4)$$

where μ is the base service rate. Positive σ values tend to decrease the span of control (to provide each subordinate with greater oversight) and increase the height of the hierarchy. Negative σ values tend to increase the span of control and decrease the height of the hierarchy.

Of all the extensions tried, only agent supervision succeeded in significantly increasing the heights of optimal hierarchies. However, extreme values of σ were required to do this. Figure 6 shows the height of the optimal hierarchies for different values of the supervision factor σ. For sufficiently large positive values, the organization can overcome increased delay costs as tasks move upward through the hierarchy, because the lower layers process tasks so quickly that there is very little delay even when a task must pass through all the levels of the hierarchy. However, the optimal height does not increase beyond 3 layers except for very large values of σ; when $\sigma = 10$, every agent increases the service rate of its subordinates by at least $10/N_1$ (e.g, 3.7% when $N_1 = 267$), and possibly much more. Eventually, there is a limit to the increase in optimal height because running costs limit the number of supervisors an agent can have and the top layer does not get any supervision bonus. For supervision factors beyond those indicated in Figure 6, the height of the optimal hierarchy did not increase beyond four.

When the supervision factor is a sufficiently large negative value, the increased delay costs offset any benefits of vertical specialization, resulting in an optimal hierarchy consisting of one layer. It is worth noting that the gains provided by vertical specialization were substantial enough that a very large negative supervision factor was required to offset it. With $\sigma = -20$, each agent slows the service rates of its subordinates by at least $20/N_1$ (e.g, 7.5% when $N_1 = 267$), and possibly much more. This demonstrates that while vertical specialization may not induce very tall optimal hierarchies, the shallow hierarchies it does induce exploit a substantial and robust cost savings.

Related Work

Organizational design for multi-agent systems has received considerable attention in recent years. It is now well-

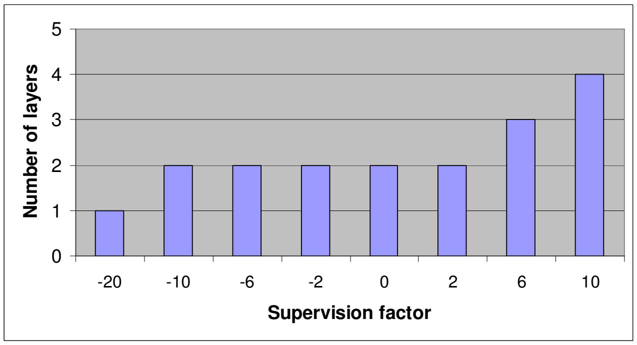

Figure 6: *Number of layers in optimal hierarchy for different values of supervision factor σ. Quadratic running cost $K(x) = 2x^2$, linear delay cost $\phi(t) = 0.01t$.*

established that the organizational design can have significant effects on the performance of the system (So & Durfee 1996; Horling & Lesser 2004). However, the majority of organizational design in multi-agent systems relies on explicit decompositions in the task environment in order to structure the organization (Tambe 1997; Giampapa & Sycara 2002; Lesser *et al.* 2004). Often the resulting hierarchies closely mimic the structure of the problem decomposition. While these techniques are extremely effective, they give no insight into the usefulness of hierarchical structure in other domains that lack this explicit decomposition.

Explanations for the prevalence of hierarchies in human organizations have been extensively studied in economics and organizational theory. Most of these have focused on horizontal parallelization, managerial attention (Geanakoplos & Milgrom 1991), and simplifying coordination (Garicano 2000), all of which are well known to also apply to multi-agent systems. In comparison, vertical specialization has received little attention.

A vertically specialized distributed sensor network was described in (Kulkarni *et al.* 2005), in which the sensor network had to perform four kinds of tasks: object detection, object localization, object recognition, and object tracking. It was found that a three-layered organization substantially reduced power consumption (running costs) compared to a single-layered sensor network, without substantially increasing the delay. This work provided a clear example of vertical specialization, but did not explore the extent to which vertical specialization could be employed.

Conclusions and Future Work

In this paper we examined the extent to which vertical specialization can improve performance and found that in many circumstances it leads to a significant reduction in costs. A surprising finding was that that two or three levels of specialization are often optimal or very close to optimal. Moreover, this result is robust to a wide range of organization and task settings. This suggests that multi-agent system designers need only introduce a small amount of hierarchy in order to enjoy the benefits of vertical specialization.

While this is a significant advance, there are a number of issues that require further work. One key issue is multi-dimensional task complexity, where there are multiple types of tasks and agent capabilities. This may lead to other organizational strategies besides vertical specialization, such as layers including heterogeneous agents, and horizontal specialization of agents into different capabilities. In some cases, tasks may require multiple capabilities, which can necessitate multiple agents working on the same task if they do not have the requisite capabilities individually. It would be especially interesting to see if similar results on degrees of specialization hold for horizontal specialization, where cost reductions by agents specializing for specific types of tasks trade off with increased coordination costs for tasks that require multiple types of capabilities.

Acknowledgments

This research has been sponsored in part by AFOSR FA9550-07-1-0039 and AFOSR FA9620-01-1-0542.

References

Beggs, A. W. 2001. Queues and hierarchies. *Review of Economic Studies* 68:297–222.

Garicano, L. 2000. Hierarchies and the organization of knowledge in production. *J. of Poli. Econ.* 108:874–904.

Geanakoplos, J., and Milgrom, P. 1991. A theory of hierarchies based on limited managerial attention. *J. of the Japanese and International Economies* 5:202–225.

Giampapa, J., and Sycara, K. 2002. Team oriented agent coordination in the RETSINA multi-agent system. In *Agents'02*.

Horling, B., and Lesser, V. 2004. Quantitative organizational models for large-scale agent teams. In *Proc. of the Int'l Work. on Massively Multi-Agent Systems*, 297–312.

Kulkarni, P.; Ganesan, D.; Shenoy, P.; and Lu, Q. 2005. Senseye: a multi-tier camera sensor network. In *MULTIMEDIA '05*, 229–238. New York, NY, USA: ACM.

Lesser, V.; Decker, K.; Wagner, T.; Carver, N.; Garvey, A.; Horling, B.; Neiman, D.; Podorozhny, R.; NagendraPrasad, M.; Raja., A.; Vincent, R.; Xuan, P.; and Zhang, X. Q. 2004. Evolution of the gpgp/taems domain-independent coordination framework. *Autonomous Agents and Multi-Agent Systems* 9(1).

Ortiz, C. L.; Vincent, R.; and Morisset, B. 2005. Task inference and distributed task management in centibots robotic systems. In *AAMAS-05*.

S. Bussmann, N. R. J., and Wooldridge, M. 2004. *Multiagent systems for manufacturing control: A design methodology*. Series on Agent Technology. Springer Verlag.

So, Y., and Durfee, E. 1996. Designing tree-structured organizations for computational agents. *Comp. and Math. Org. Theory* 2(3):219–246.

Tambe, M. 1997. Towards flexible teamwork. *J. of Artificial Intelligence Research* 7:83–124.

Zhang, H., and Lesser, V. 2004. A dynamically formed hierarchical agent organization for a distributed content sharing system. In *IAT '04*, 169–175.