Uncertainty in Intelligent and Transportable Agent Systems [1]

Katsuhiro Moizumi and George Cybenko
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755 USA

Abstract

Transportable agents are a relatively new technology for coordinating work within an organization. Transportable agents can migrate from machine to machine, gathering information, analyzing it and interacting with human operators if required. They allow unprecedented flexibility in carrying out tasks by allowing essentially any distributed organizational scheme to be implemented. Much of the technology for implementing transportable agents is presently available and affordable but mathematical models and analysis techniques for understanding the properties of different organizational implementations of agents remains primitive. In this paper, we initiate study of one of the simplest coordination problems in a distributed system, that of resource allocation. We consider a hierarchical organization in which agents aggregate information at different stages and then act to disseminate it. The process of aggregation introduces noise or uncertainty into the combined information. In this paper, we model how the agent architecture affects the quality of the resulting solutions, demonstrating both analytic and simulation results. Our results show that the architecture of an agent implementation can have a significant impact on the accuracy of solutions and that this impact is not always straightforward or intuitive. Specifically, we explain quantitatively why higher levels of a decision structure should have better information integration technology and why agents should minimize the number of information integration steps they perform.

1 Introduction

Transportable agents are software programs that can migrate from machine to machine in a computer network, carrying out information gathering and processing tasks. Transportable agents are most useful in applications where networks are potentially unreliable or when information databases are very large. In the case of unreliable networks, transportable agents can traverse network links and continue to carry out work on behalf of a user even when a connection is down. Moreover, if the information resources are very large databases, it may not be feasible to move the databases to a user's site so that the user must send an agent to explore the data at the remote location using less network resources. Agents can spawn other agents to perform subtasks, meeting at a future time and integrating results for example. Several transportable agent systems have been developed in recent years [17, 20, 19] and their use in real applications is starting to be explored.

In implementing an agent-based solution to a distributed information application, there is much flexibility in how the architecture is designed. For example, suppose there are several sites that contain information that might be relevant to a task and we want to launch agents that will locate, assemble and disseminate the relevant information. Should a single agent visit each site, collating information as it moves around, removing redundant information when it is found? Or should several agents be spawned, acting independently for some time and then meeting occasionally to compare results? In the second case, when should agents be spawned and how many?

[1]Supported in part by AFOSR grant F49620-93-1-0266 and ONR grant N00014-95-1-1204. We thank the Department of Mathematics and Computer Science at Leiden University, the Netherlands for facilities used in part of this research.

The answer to such questions clearly depends on the performance metric we specify – minimize time, maximize robustness, minimize cost (by avoiding redundancy or unnecessary data transmissions) for example. If transportable agents become widely used and can spawn copies of themselves to solve information tasks more efficiently, then some understanding of these issues is required. Models of network performance as well as information resources and their costs in terms of time and money must be available for agent systems and human organizational structures to dynamically optimize their performance.

In this paper, we explore the properties of a simple information gathering and dissemination problem in order to better understand the impact of agent architectures on the accuracy of the solution. By now, there is a sizable body of research that relates organizational structure to decision-making performance, e.g., analysis of primary factors for decision making in organizations [1, 2, 3]. Social-science studies of the effects of different organizational structures on their decision making are described in [4, 5, 6]. In engineering, extensive research has been done in a binary detection problem in a distributed organization [7, 8, 9, 11, 12]. Detection performance in different organization structures are compared in [14, 15]. An optimal solution of the binary detection problem in a multi-level hierarchical organization is derived in [13, 10]. However, those research does not fully address the kinds of problems that confront agent systems because agents are completely flexible and dynamic in their execution behavior. Moreover, essentially no work has been done comparing information processing powers of different organizational structures to our knowledge.

The main results of this paper suggest that architectures for transportable agents should be as flat as possible. That is, the least noise or uncertainty corresponds to network structures that involve the fewest combination steps. More specifically, the largest errors occur at the highest levels of aggregation suggesting that agents or people responsible for combining data at higher levels within an organization must have superior precision and techniques available to them. This analytic observation reinforces the practice already used in many organizations – namely, that combination of information at the highest levels must be done by agents with the highest skills and that organizations themselves are getting flatter.

Section 2 of this paper introduces the model problem that we study together with motivation and examples. Section 3 summarizes the analytic results and compares them with simulations while Section 4 discusses the meaning of the these results and how they can be used in agent systems. The Appendix contains the derivation of the analytic results.

2 Problem Formulation and Motivation

There is much discussion of and expectation for intelligent agent systems that will move through computer networks, searching for information and processing it on behalf of a user. Such systems are still in their infancy and their properties are poorly understood at this time. A simple question for example is how will multiple agents cooperate to collectively solve information processing tasks? To illustrate such issues using a specific problem, suppose we have multiple distributed sites which have "supplies" and "demands" for material resources such as gasoline, machine parts or other types of inventory. We would like software agents to move between and among sites, assessing supplies and demands, making autonomous decisions about resource allocation. How should these agents be organized to make decisions that are most robust in the presence of uncertainty? Uncertainty arises because of time delays and approximate encoding of exact or true quantities.

Moreover, many interesting examples involve resources that are more difficult to quantify, such as health care in emergency and battlefield situations, personnel management and other human resource problems. In such applications, quantification is difficult and combining information results in loss of resolution, therefore an increase in uncertainty. For an analytic study of such problems, the challenge is twofold – how to formulate the problems analytically and how to learn something from those formulations.

In this section we introduce a simple resource allocation model to study how the structure of a distributed information system can influence the quality of decisions as based on the information gathered. We are specifically interested in the implications of these results for intelligent agents but the findings have bearing on general organizational theory as well.

2.1 A Medical System Example

Consider for example the situation depicted in Figure 1. A number of medical centers must respond to a medical emergency distributed over a region. Each medical emergency consists of some number of individual cases, each of which is described using text to convey the specific conditions of the individual cases. Those "demands" for medical services must be combined by intermediate nodes and ultimately compared with the "resources" (physicians and equipment) currently available, also aggregated and described using text. Each aggregation step reduces resolution because it summarizes a larger amount of information into a smaller amount. The ultimate resource allocations are made according to information that has lower resolution and is therefore more uncertain. Figure 1 depicts medical resources (such as hospitals) and field units with medical emergencies that must be assigned to hospitals. Two architectures are shown – one that is *deep* and one that is *shallow*. The deep architecture has more levels while the shallow architecture has few levels. Which architecture leads to better decisions, assuming that agents or people combine information at each node that the information traverses?

2.2 Information Retrieval

Another class of problems involves hierarchies for information and document retrieval. A user or group of users delegate an information retrieval task to agents which then combine their requests, search a network, spawning other agents, combining results and disseminating the retrieved documents. Such applications are currently being developed by several groups. Again, what architecture is most robust and suitable?

2.3 A Formal Model of Resource Allocation

In order to formulate such a problem quantitatively and subsequently analyze it, we assume a simple model for the process of combining information and making decisions based on it. In general, a hierarchical organization (of either people or agents) consists of units which aggregate information from other units in the hierarchy, passing that information through the hierarchy. Once all information is available, resource allocations are made, and they are propagated back through the hierarchy. A fundamental ingredient of this model is that information is condensed as it passes through the hierarchy in order to be digestible by other units. The aggregation process introduces noise or uncertainty into the information. Thus, a unit must make an allocation decision based on uncertain information. In this paper, we develop the resource allocation problem for the binary tree where the supply and demand information accumulate errors during the information propagation process. However, our results generalize to other decision structures as discussed later.

The resource allocation problem is to mediate allocation of supplies of a resource distributed within a system to the nodes which demand it. Our first basic model of the organizational structure is an n level binary tree. Figure 2 shows the organization when $n = 5$. Levels are numbered, starting with the leaves of the tree and increasing as the level becomes higher, culminating in the root.

> Definition – A *resource allocation* problem consists of a graph with nodes $i = 1, ..., n$, supplies s_i and demands d_i at each node, i. The goal is to assign supplies ${s'}_i$ to nodes in order to satisfy the demands, d_i, at each node.

A node in the graph is an organizational unit in which information is aggregated and passed on to higher levels with subsequent action being taken on decisions that propagate back. Each node processes two kinds of information – supply and demand – about resources which are modeled as real numbers. At first, only nodes on the leaves have these two kinds of information. In general, local supplies are different from local demands. In the binary tree model, the supply and demand information propagate into higher nodes with aggregation being implemented by addition as shown in Figure 3(a). At the root of the tree, the supplies are distributed to the lower nodes by proportional splitting according to the demand information as shown in Figure 3(b). In our treatment of this formulation, we consider intermediate decision nodes to be distinct from actual nodes which contain supply and demand information but this is purely for notational simplicity.

To be more precise, let a unit u have two children, c_1 and c_2. These two children have supplies, s_1 and s_2, as well as demands, d_1 and d_2. In the noiseless case, the propagated supply from u is merely $s_1 + s_2$ and the propagated demand is $d_1 + d_2$. These supplies and demands are propagated further up the hierarchy unless u is the root. Resources are allocated according to a proportional splitting rule: if u is allocated a supply of s', which may differ from $d_1 + d_2$, then c_1 receives $\frac{s' d_1}{d_1+d_2}$ and c_2 receives $\frac{s' d_2}{d_1+d_2}$. This rule is applied recursively in the binary tree to each node in the upward and downward directions.

It is easy to verify that this procedure is merely an algorithm for proportionally allocating resources in such a hierarchy. If $s_1, ..., s_n$ are supplies and $d_1, ..., d_n$ are demands at the leaf nodes, then the algorithm computes quantities

$$s' = \sum_{i=1}^{n} s_i \quad \text{and} \quad d' = \sum_{i=1}^{n} d_i$$

and allocates $\frac{s' d_i}{d'}$ to leaf node i in the error-free case. However, our interest is in the noisy, uncertain case where accumulation of information leads to errors in intermediate computed quantities.

Noise is added to the supply and demand information during the aggregation process. Our noise model is multiplicative and is given by:

$$Q = q(1+\varepsilon), \quad -\delta < \varepsilon < \delta$$

where $0 < \delta < 1$ is a bound on the relative error size. Here q is the correct information value and Q is the modeled noisy information. For example, when supply information is aggregated at a node of the tree during the aggregation process, the noisy supply is expressed as:

$$S = (s_1 + s_2)(1+\varepsilon).$$

The *error* in the resource allocation solution is a measure of how far off the allocated supply is from the requested demand. This error is calculated by first normalizing according to:

$$\frac{d_i - s_i'}{d_i}$$

where s_i' is the actual allocated supply for node, i. This normalizes the leaf error as a fraction of the leaf demand. Next, this fraction is thresholded against 0 because a positive fraction indicates that local demand exceeded supply which is counted as an error. If the local demand is less than 0, then no error is counted at that node because demand has been met but our results generalize to the oversupply case as well by symmetry and we discuss this situation later. Finally, these thresholded, normalized errors are added for the overall error metric.

DEFINITION – The overall error is given by

$$ER = \sum_{leaf\ nodes\ i} \max\left\{\frac{d_i - s_i'}{d_i},\ 0\right\}.$$

As previously described, our model allows errors to be introduced at each aggregation step and the proportional distribution is performed according to the information available to the local node. In the general model, errors are cumulative. However, we examine the performance of this distributed resource allocation model by isolating the errors contributed by different levels in the tree, separating supply and demand respectively. This isolation and separation is done to identify the relative contribution of each level in the overall performance. This approach is based on a first-order linear model for the local error as it appears in the total error.

More precisely, the total error can be decomposed as follows:

$$ER = \sum_{L=1}^{n} ER_{L,s} + \sum_{L=1}^{n} ER_{L,d}$$

where $ER_{L,s}$ and $ER_{L,d}$ are the errors arising from uncertainties introduced at level L due to supply and demand uncertainties respectively. In the next section, we summarize the analytic and simulation results isolating the role that each level plays in contributing to the total.

3 Analytical and Experimental Results

This section summarizes how errors in demand and supply affect the error in the overall resource allocation. We isolate the contributions of errors made at each level of aggregation and restrict attention to complete binary tree graphs.

Our main results are:

THEOREM 1. – Assume that the expected values of supplies and demands at leaf nodes are identical. Then, For a graph consisting of 2^n nodes and a binary tree combination and distribution strategy, the resource allocation error contributed by supplies at level L is

$$ER_{L,s} \approx \frac{\delta}{\sqrt{6\pi}\sqrt{2^{n-L}}} \tag{1}$$

while the error contributed by demands at level L is

$$ER_{L,d} \approx \sqrt{1-2^{L+1-n}}\frac{\delta}{\sqrt{6\pi}} \tag{2}$$

PROOF – The proofs are in the Appendix. The results are approximate because we use first-order error models and the Central Limit Theorem to simplify the analytic results leading to the closed form approximations of the theorem.

These two expressions are plotted in Figures 4 and 5 together with the results of computer simulations. The main conclusions from these results are:

- Errors due to supply *increase* as the decision making level increases, meaning that higher levels must have more accurate supply information fusion capabilities to perform similarly to lower levels;
- Errors due to demand *decrease* as the decision making level increases, meaning that higher levels of demand information fusion are less sensitive to errors than at lower levels;
- The analytic approximations are very close to simulation results.

Intuitively, these results can be explained by the way in which errors accumulate. In the supply information case, note that averaging a larger collection of numbers with random fluctuations leads to a smaller average variance than averaging fewer numbers. Because information fusion at higher levels involves combining a smaller amount of information, there is less opportunity for errors to cancel one another.

Conversely, errors in demand affect the distribution part of the resource allocation process. Distribution of the supply is based on the known information about demand which has a larger absolute variation when more information is fused (as opposed to average variation per node which is smaller for a larger number of nodes). Because the absolute variation of errors in demand is larger when the errors occur at lower levels of the decision tree, the overall resource allocation is worse. This explains the downward trend of allocation errors as a function of decision level.

The implications of these results for automated decision making systems, involving both intelligent agents and human decision makers are:

1. High levels of a decision hierarchy must have better decision making tools available to them, especially for supply information in resource allocation;
2. A decision making hierarchy should be as shallow as possible (which is being observed in more and more organizational structures using modern information technology);
3. Information technology, especially intelligent agents, should be and can be used to make decision making hierarchies more shallow.

3.1 Discussion

3.2 Simulation Results

We have performed simulations which compute errors in the final resource allocation solution (that is, total supply deficiencies with respect to demands). The simulations use a binary tree model with 7 levels (64 leaf nodes). In order to isolate the impact of errors at each level on the overall performance, the simulations are executed with an error which occurs on either supply or demand and then only on a fixed level of the hierarchy. Errors are randomly and independently generated from a uniform distribution in the range $-\delta < \varepsilon < \delta$. Because of our analysis, we expect the distribution itself not to make a major difference, only the first and second order statistics. An expected average total leaf error is estimated by repeating the same set of simulations 50 times. The total supply and demand are balanced but simulation results show the same trends for cases where supply both exceeds demand and where demand exceeds supply. The results of the simulations are shown by the solid lines in Figures 4 and 5. The errors are bounded by $\delta = 0.5$. The simulations where carried out using MATLAB.

As predicted by the analytical results, observe the tendency for demand information error in higher levels to causes smaller allocation errors in the overall problem. Conversely, supply information error at higher levels causes larger allocations errors. The analytical predictions for the simulation, given by (1) and (2), are represented by the dashed line in Figures 4 and 5. Notice the close agreement between the simulation and analytic predictions.

The simulation results for supply errors match the analytical predictions well. However, simulations for demand errors differ more from the analytical results because they use both a Taylor expansion (to get first order terms) and the Central Limit Theorem. On the other hand, the supply error simulation results use only the Central Limit Theorem for approximation.

The analysis of the supply case shows that total error depends entirely on the number of errors which occur at some level of the tree. By examining the form of (4), the final error is linear in the number of nodes involved at a given decision level of the tree. Since these errors have the same mean and variance, as the number of error terms becomes larger, the average variance of the errors becomes smaller. This leads to a smaller expected value for the total allocation error.

On the other hand, the analysis of demand information error, the expected total error is not strongly affected by the number of nodes contributing to the demand error. The curve of expected error in the demand case in Figure 5 shows a more gradual dependence on decision levels than the supply case of Figure 4. Examining equation (9) of the proof in the Appendix, the term

$$\frac{\varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_{n-m-2} + \varepsilon_{n-m-1}}{2^{n-m-1}},$$

which is the only factor that depends on the number of error-contributing nodes, is much smaller than 1. It has a small effect on the value of (9).

We have further examined the error tolerance of each level with respect to a fixed total allocation error. That is, for a fixed total allocation error, say ER, how large can δ at level k be without having the total error exceed ER? This value is plotted for each level in Figures 6 and 7. The analytical curves of the bound in the demand and supply case are obtained according to the expressions in (5) and (11).

The analytical and simulation results in the supply case clearly show that higher decision making levels require a smaller error to achieve a fixed total allocation error.

The analytical and simulation results in the demand case (Figure 7) agree fairly well also – note that the bounds are almost constant for all the level. However, a careful comparison of the results for the demand case, shows that the simulation result does not match the analysis well in the higher decision making levels. This mismatch is not surprising because the analytical and simulation result of Figure 5 themselves didn't agree well, especially at higher levels.

3.3 Relationships to Backward Error Analysis

Backward error analysis is a commonly used technique in numerical analysis to study the numerical stability of floating point algorithms [18]. In backward error analysis, computed output data are interpreted as exact solutions to perturbed input problem data. We can apply backward error analysis in this model as well, leading to the same results, just arriving at them differently.

For example, instead of adding errors after aggregating information, $S = (s_1 + s_2)(1 + \varepsilon)$, the same error is interpreted as arising from perturbed input data before aggregating it:

$$S = (s_1)(1 + \varepsilon/2) + (s_2)(1 + \varepsilon/2).$$

Backward error analysis leads to the same results in both supply and demand cases as has been shown in this paper.

3.4 Decision Tree Depth

In Figure 8, we show the total allocation error as a function of the branching factor for a decision tree with 256 nodes using branching factors of 2, 4 and 16. These correspond to trees of height (depth) 8, 4 and 2 respectively. Clearly, with respect to both supply and demand errors, the total allocation errors decrease together with the tree depth supporting our earlier claims that decision trees for both agents and human organizations are better being shallow.

4 Summary

We have studied a simple problem in distributed information gathering and decision making, namely the resource allocation problem, with the goal of understanding how agent and human organizational architectures can affect the quality of the final outcomes. We have modeled information fusion as a noisy process and evaluate the effects of the noise on the quality of the final allocation solutions. We have derived analytic results which have been verified against computer simulations.

Our analytic results show that information gathering and decision making in a hierarchical structure is more susceptible to uncertainty at the higher levels of the hierarchy, in the case of noisy supply data, and more sensitive to noise at the lower levels in the case of noisy demand data. However, the sensitivity with respect to decision levels in demand information is pronouncedly smaller. These effects can be alleviated by using shallower decision trees, that is, using fewer levels in a hierarchy. For agent based systems, this means that spawning many agents earlier on in a process and using fewer agents to fuse the resulting information is superior to less frequent spawning of agents and more aggregation steps. For human decision making structures, this translates into using fewer levels of management which is something that information technology makes possible and is indeed being done in practice.

Our analysis has been for a very specific problem that we could quantify and analyze but the vast majority of distributed information processing and decision making problems remain to be studied. Much work remains to be done towards understanding the relationship between decision making structures, both electronic/digital and human, and the quality of the final results. We hope that this study is a small first step towards understanding and quantifying those relationships.

References

[1] Duncan, R. "The characteristics of organizational environments and perceived environmental uncertainty" *Administrative Science Quarterly,* 17, pp.313-327, 1972

[2] Arthur L. Stinchcombe *Information and Organization*, Berkeley: University of California Press, 1990

[3] James G. March and Herbert A. Simon, *Organizations*, 2nd Edition, Cambridge, MA: Blackwell, 1993.

[4] Kathleen M. Carley "Designing organizational structures to cope with communication breakdowns: a simulation model" *Industrial Crisis Quarterly,* 5, pp. 19-57, 1991

[5] Kathleen M. Carley "Organizational learning and Personnel Turnover" *Organization Science,* vol. 3, No. 1, pp. 20-46, 1992

[6] Henry Ogden Armour, and David J. Teece, "Organizational structure and economic performances: a test of the multidivisional hypothesis," Bell Journal of Economics, vol. 9, pp. 106-122, 1978

[7] Robert R. Tenney, and Nils R. Sandell, Jr., "Detection with distributed sensors," IEEE trans. Aeros. Electr. Syst., vol. AES-17, No4, pp. 501-510, July 1981

[8] A. Pete, K. R. Pattipati, and D. L. Kleinman "Distributed Detection in Teams with Partial Information: A Normative-Descriptive Model" *IEEE Transactions on Systems, Man, and Cybernetics*, vol.23, No. 6, pp. 1626-1648, November/December 1993

[9] A. Pete, K. R. Pattipati and D. L. Kleinman, "Optimal team and and individual decision rules in uncertain dichotomous situations," Public Choice, vol. 75, pp. 205- 239, Mar. 1993

[10] A. Pete, K. R. Pattipati and D. L. Kleinman, "Optimization of detection networks with multiple event structures," IEEE trans. Automat. Contr., vol. 39, pp. 1702-1707, Aug. 1994

[11] J.N. Tsitsiklis, "Decentralized detection," in Advanced Statistical Signal Processing, H. V. Poor and J. B. Thomas, Eds. Greenwich, CT: JAI Press, 1993, vol. 2

[12] Z. B. Tang, K. R. Pattipati, and D. L. Kleinman, "An algorithm for determining the decision threshold in a distributed detection problem," IEEE Trans. Syst., Man, cybern., Vol. 21, pp. 231-237, Jan./Feb. 1991

[13] Z. B. Tang, K. R. Pattipati, and D. L. Kleinman, "Optimization of detection networks: Part II generalized tree structures," IEEE Trans. Syst., Man, Cybern., vol. 23, pp. 211-221, Jan./Feb. 1993

[14] V. V. Veeravalli, T. Basar, and H. V. Poor, "Decentralized sequential detection with a fusion center performing the sequential test," IEEE Trans., Inform. Theory., vol. 39, pp. 433-442, Mar. 1993

[15] Jason D. Papastavrou, and Michael Athans, "On optimal distributed decision architectures in a hypothesis testing environment," IEEE trans. Automat. Contr., vol. 37, pp. 1154-1169, Aug. 1992

[16] Robert L. Mason, Richard F. Gunst, James L. Hess *Statistical design and analysis of experiments: with applications to engineering and science*, New York: Wiley, 1989 (reprinted)

[17] Robert S. Gray, George Cybenko, David Kotz and Daniela Rus. *Agent Tcl*, In William Cockayne and Michael Zypa, editors, Itinerant Agents: Explanations and Examples with CDROM, Manning Publishing, 1996. To appear.

[18] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, NY, USA, 1973.

[19] General Magic, *Telescript Technology: Mobile Agents*, http://www.genmagic.com/Telescript/Whitepapers/wp4/whitepaper-4.html, 1996.

[20] Dag Johansen, Robert van Renesse and Fred B. Schneider, *An Introduction to the TACOMA Distributed System Version 1.0* , Technical Report 95-23. Department of Computer Science, University of Troms, Norway, June 1995.

A Appendix: Proof of Main Theorem

A.1 Uncertainty in Resource Supply

Here we consider the supply error isolated by level. First of all, we study how an error at a parent node propagates to its children nodes when there is no error in demand. Assume that the parent node has noisy supply information S' which should be distributed to its children. The noisy information S' is related to the exact supply information S according to

$$S' = S(1+\varepsilon_1)$$

where ε_1 is the relative error at the node and S is the supply without error.

The supply S' is distributed to children n_i and n_{i+1} according to

$$n_i \quad : \quad S' \frac{d_i}{d_i + d_{i+1}}$$

$$n_{i+1} \quad : \quad S' \frac{d_{i+1}}{d_i + d_{i+1}}.$$

The normalized errors at n_i and n_{i+1} are, respectively,

$$n_i : \frac{S' \frac{d_i}{d_i+d_{i+1}} - S \frac{d_i}{d_i+d_{i+1}}}{S \frac{d_i}{d_i+d_{i+1}}} = \varepsilon_1$$

$$n_{i+1} : \frac{S' \frac{d_{i+1}}{d_i+d_{i+1}} - S \frac{d_{i+1}}{d_i+d_{i+1}}}{S1 \frac{d_{i+1}}{d_i+d_{i+1}}} = \varepsilon_1$$

In the same way, children nodes under both n_i and n_{i+1} get the same normalized error ε_1. Thus, it turns out that all the leaf nodes descendant from a given node with normalized error ε have the same normalized error ε. ¿From this result, it is clear that each leaf node has the same normalized error in supply as does the root. Thus, the total error at leaves is obtained by calculating the normalized supply error at the root times the number of leaves.

Next, we show how to calculate the normalized supply error at the root. Consider the nth level of the binary tree and assume that all the nodes in the kth level from the leaves have supply information error. Given the correct supply at each node in the kth level is s_i, the total supply information which is collected at the root is

$$\sum_{i=1}^{2^{n-k}} s_i(1+\varepsilon_i)$$

including the error.

The normalized supply error at the root will be:

$$\begin{aligned} error_{root} &= \frac{\sum_{i=1}^{2^{n-k}} s_i(1+\varepsilon_i) - \sum_{i=1}^{2^{n-k}} s_i}{\sum_{i=1}^{2^{n-k}} s_i} \\ &= \frac{\sum_{i=1}^{2^{n-k}} s_i \varepsilon_i}{\sum_{i=1}^{2^{n-k}} s_i}. \end{aligned}$$

The expected value of the normalized and thresholded error at each leaf node is

$$E(error_{root}) = E\left(\max\left(\frac{\sum_{i=1}^{2^{n-k}} s_i \varepsilon_i}{\sum_{i=1}^{2^{n-k}} s_i},\ 0\right)\right) \tag{3}$$

This expected value is affected by the sizes of the s_i. In order to make the analysis tractable, we use a constant size for each of the leaf values, s_i. In the simulations, we randomize the supplies and use their expected values as a basis for the corresponding analysis. Using $E(s_1) = E(s_2) = .. = E(s_{2^{n-k}}) = s$, (3) becomes

$$\begin{aligned} E(error_{root}) &= E\left(\max\left(\frac{\sum_{i=1}^{2^{n-k}} s\varepsilon_i}{\sum_{i=1}^{2^{n-k}} s},\ 0\right)\right) \\ &= E\left(\max\left(\frac{\sum_{i=1}^{2^{n-k}} \varepsilon_i}{2^{n-k}},\ 0\right)\right) \end{aligned} \tag{4}$$

Assuming the errors ε_i are uniform random variables whose ranges are $-\delta < \varepsilon_i < \delta$, an estimate of (4) can be based on the *Central Limit Theorem* [16] as follows.

According to the Central Limit Theorem, if $\varepsilon_1, \varepsilon_2, .., \varepsilon_n$ is a sequence of n independent random variables with $E(\varepsilon_i) = \mu_i$, variances $V(\varepsilon_i) = \sigma_i^2$ and we define $X = \varepsilon_1 + \varepsilon_2 + .. + \varepsilon_{n-1} + \varepsilon_n$ then

$$Z_i = \frac{X - \sum_{i=1}^{n} \mu_i}{\sqrt{\sum_{i=1}^{n} \sigma_i^2}}$$

has an approximate $N(0,1)$ distribution as n approaches infinity. Each error random variable ε_i in (4) has the same mean, $E(\delta_i) = \mu = 0$, and variance, $V(\varepsilon_i) = \sigma^2 = \frac{\delta^2}{3}$. Let $X = \sum_{i=1}^{2^{n-k}} \varepsilon_i$ so that

$$\begin{aligned} Z &= \frac{X - 2^{n-k}\mu}{\sqrt{2^{n-k}\sigma^2}} \\ &= \frac{X}{\sqrt{2^{n-k}}\frac{\delta}{\sqrt{3}}} \end{aligned}$$

has approximately an N(0,1) distribution. Therefore, (4) leads to the approximation

$$\begin{aligned} E(error_{root}) &= \frac{1}{2^{n-k}} E(\max(X, 0)) \\ &\approx \frac{1}{2^{n-k}} \int_0^\infty \frac{X}{(\sqrt{2^{n-k}}\sqrt{2\pi}\frac{\delta}{\sqrt{3}})} e^{-(\frac{1}{2})(\frac{X}{\frac{\delta}{\sqrt{3}}\sqrt{2^{n-k}}})^2} dx \\ &= \frac{1}{2^{n-k}} (\sqrt{2^{n-k}}\frac{\delta}{\sqrt{3}}) \int_0^\infty \frac{1}{\sqrt{2\pi}} Z e^{-\frac{Z^2}{2}} dz \\ &= \frac{1}{\sqrt{2\pi}} e^{-\frac{Z^2}{2}}]_0^\infty \\ &= \frac{1}{2^{n-k}} \sqrt{2^{n-k}} \frac{\delta}{\sqrt{3}} \frac{1}{\sqrt{2\pi}} \\ &= \frac{\delta}{\sqrt{6\pi}\sqrt{2^{n-k}}} \end{aligned}$$

As described before, the total leaf error due to noisy supply at a fixed level is the size of the error at the root multiplied by the total number of leaf nodes. We then divide by the number of leaf nodes to get an average. Thus, the expected value of the total average leaf error due to errors in supply information at the kth level will be

$$E(total\ error) \approx \frac{\delta}{\sqrt{6\pi}\sqrt{2^{n-k}}} \tag{5}$$

Note that in our simulations, we show these errors as percentages and so they are multiplied by 100 in the figures.

The general trend is evident from this expression: the larger that k is, the larger the total error will be. This is because at higher levels, there is statistically less opportunity for error cancellation so the variances are larger as expressed by (5). The simulations bear this out.

A.2 Uncertainty in Resource Demand

There are two stages in the propagation of demand errors. First there is the aggregation process of demand information followed by the distribution process in which supply is allocated according to the known demand.

In the aggregation process, errors which occur in a certain level propagate to higher levels as in the supply case. The demand errors are aggregated in the parent nodes and are stored there. Thus, all nodes at a higher level than the one in which the errors originally occurred have errors in demand information.

Let $d_{m,l}$ represent the demand at node l in level m. Isolating the demand errors to level m, the demand information at nodes above level m will be:

$$d_{m+k,i} = \sum_{l=2^{k-1}(i-1)+1}^{2^{k-1}i} d_{m,l}(1+\varepsilon_l).$$

Once the aggregation process reaches the root, the distribution process starts. Aggregated supply information at the root is distributed to the children nodes in proportion to the demand information of the children. Therefore, if there is an error in a demand information, the distributed supply also gets an error. Since all the nodes ranging from the root to the level where errors occur in demand information (say, the mth level) have demand errors, errors in the distributed supplies accumulate until the distribution process reaches the mth level. Below level m, the nodes receive the same normalized errors that their ancestor node in the mth level gets. Thus, the normalized error in the distributed supply on the leaves, will be equal to the normalized error at the nodes in the mth level multiplied by the number of leaves which are descendant of the nodes.

Notice that we can restrict attention to nodes 1 and 2 at each level to simplify notation. The distributed supply at a node in the mth level can be calculated as follows. At node $n_{m,1}$ the delivered supply is:

$$\begin{aligned} s_{m,1} &= S \times \frac{d_{n-1,1}}{d_{n-1,1}+d_{n-1,2}} \times \frac{d_{n-2,1}}{d_{n-2,1}+d_{n-2,2}} \times ... \\ & \times \frac{d_{m+1,1}}{d_{m+1,1}+d_{m+1,2}} \times \frac{d_{m,1}}{d_{m,1}+d_{m,2}} \end{aligned}$$

where S is the supply at the root.

Since

$$d_{m+(k+1),1} = d_{m+k,1} + d_{m+k,2} \tag{6}$$

in (6) except the case where

$$d_{m+1,1} = (d_{m,1}+d_{m,2})(1+\varepsilon_1) \tag{7}$$

(7) reduces to

$$s_{m,1} = S \times \frac{(1+\varepsilon_1)d_{m,1}}{d_{n-1,1}+d_{n-1,2}}. \tag{8}$$

Applying this reduction and using an average value for $d_{m,i}$, that is, $E(d_{m,i}) = d$, (8) becomes

$$\begin{aligned} s_{m,1} &= S\frac{1+\varepsilon_1}{((1+\varepsilon_1)+(1+\varepsilon_2)+....+(1+\varepsilon_{2^{n-m-1}}))} \\ &= S\frac{1}{2^{n-m-1}}\frac{1+\varepsilon_1}{1+\frac{\varepsilon_1+\varepsilon_2+..+\varepsilon_{2^{n-m-1}}}{2^{n-m-1}}} \end{aligned} \tag{9}$$

Since $\frac{\varepsilon_1+\varepsilon_2+...+\varepsilon_{n-m-2}+\varepsilon_{n-m-1}}{2^{n-m-1}} << 1$, using a Taylor expansion, (9) reduces to

$$\begin{aligned} s_{m,1} &\approx S\frac{1}{2^{n-m-1}}(1+\varepsilon_1)(1-\frac{\varepsilon_1-\varepsilon_2-..-\varepsilon_{2^{n-m-1}}}{2^{n-m-1}}) \\ &= S\frac{1}{2^{n-m-1}}(1+\varepsilon_1-\frac{\varepsilon_1+\varepsilon_2+..+\varepsilon_{2^{n-m-1}}}{2^{n-m-1}}-\varepsilon_1\frac{\varepsilon_1-\varepsilon_2-..-\varepsilon_{2^{n-m-1}}}{2^{n-m-1}}) \\ &\approx S\frac{1}{2^{n-m-1}}(1+\varepsilon_1-\frac{\varepsilon_1-\varepsilon_2-..-\varepsilon_{n-m-1}}{2^{n-m-1}}) \\ &= S\frac{1}{2^{n-m-1}}(1-\frac{(2^{n-m-1}-1)\varepsilon_1-\varepsilon_2-\varepsilon_3....-\varepsilon_{2^{n-m-1}}}{2^{n-m-1}}) \end{aligned}$$

where we have eliminated second order terms, specifically

$$\varepsilon_1\frac{\varepsilon_1-\varepsilon_2-...\varepsilon_{2^{n-m-1}}}{2^{n-m-1}}$$

because they are much smaller than 1.

In the same way, other nodes at the mth level have

$$\begin{aligned} s_{m,i} &\approx S\frac{1}{2^{n-m-1}}(1 \\ &- \frac{-\varepsilon_1 - \varepsilon_2 - .. - \varepsilon_{i-1} + (2^{n-m-1}-1)\varepsilon_i - \varepsilon_{i+1} - .. - \varepsilon_{2^{n-m-1}}}{2^{n-m-1}}) \end{aligned}$$

The correct distributed supply is

$$\begin{aligned} s'_{m,i} = S&\frac{d_{m,i}}{d_{m,1} + d_{m,2} + .. + d_{m,2^{n-m-1}}} \\ &= S\frac{1}{2^{n-m-1}}, \end{aligned}$$

again using an average for $d_{m,i}$, that is, $E(d_{m,i}) = d$. Therefore, the normalized error at a node in the mth level is

$$\begin{aligned} e_{m,i} = \max&\left\{\frac{s'_{m,i} - s_{m,i}}{s'_{m,i}}, 0\right\} \\ &\approx \max\left\{1 - (1 - \frac{(2^{n-m-1}-1)\varepsilon_1 - \varepsilon_2 - \varepsilon_3 ... - \varepsilon_{2^{n-m-1}}}{2^{n-m-1}}), 0\right\} \\ &= \max\left\{\frac{(2^{n-m-1}-1)\varepsilon_1 - \varepsilon_2 - \varepsilon_3 ... - \varepsilon_{2^{n-m-1}}}{2^{n-m-1}}, 0\right\} \end{aligned} \tag{10}$$

The reason why the *max* operator is taken in (10) is that the error in distributed supply is considered to be the supply deficiency with respect to the demand. If the delivered supply exceeds the requested demand, then no error occurs.

We can now reduce (10) extended by using the Central Limit Theorem as before. Let $E(\varepsilon_i) = 0$, $V(\varepsilon_i) = \sigma^2 = \frac{\delta^2}{3}$, and

$$Y = (2^{n-m-1}-1)\varepsilon_1 - \varepsilon_2 - \varepsilon_3 .... - \varepsilon_{2^{n-m-2}} - \varepsilon_{2^{n-m-1}}$$

so that $E(Y) = 0$ and

$$V(Y) = (2^{n-m-1}-1)^2\sigma^2 + (2^{n-m-1}-1)\sigma^2 = (2^{n-m-1}-1)(2^{n-m-1})\sigma^2.$$

According to the Central Limit Theorem,

$$\begin{aligned} Z &= \frac{Y - 0}{\sqrt{(2^{n-m-1}-1)(2^{n-m-1})}\sigma} \\ &= \frac{Y}{\sqrt{(2^{n-m-1}-1)(2^{n-m-1})}\frac{\delta}{\sqrt{3}}} \end{aligned}$$

has an approximately N(0,1) distribution. Therefore, (10) can be simplified into

$$\begin{aligned} &\frac{1}{2^{n-m-1}}E(Y) \\ &= \frac{1}{2^{n-m-1}}\int_0^\infty \frac{Y}{(\sqrt{(2^{n-m-1})(2^{n-m-1})}\frac{\delta}{\sqrt{3}})\sqrt{2\pi}} e^{-(\frac{1}{2})(\frac{Y}{\sqrt{(2^{n-m-1}-1)(2^{n-m-1})}\frac{\delta}{\sqrt{3}}})^2} dY \\ &= \frac{1}{2^{n-m-1}}\sqrt{(2^{n-m-1}-1)(2^{n-m-1})}\frac{\delta}{\sqrt{3}}\frac{1}{\sqrt{2\pi}} \\ &= \frac{\sqrt{2^{n-m-1}-1}}{\sqrt{2^{n-m-1}}}\frac{\delta}{\sqrt{3}}\frac{1}{\sqrt{2\pi}} \\ &= \sqrt{1 - 2^{m+1-n}}\frac{\delta}{\sqrt{6\pi}} \end{aligned}$$

As mentioned above, the expected value of the total normalized errors in distributed supply on the leaves is the number of the nodes on the leaves times the expected value of the normalized errors of the nodes at the mth level given by (10). We then divide by the number of leaf nodes to get an average. Thus, the expected value of the total average normalized error on the leaves due to errors in demand occurring at the mth level is

$$\begin{aligned} E(total\ error_m) &= 2^{n-1} \times E(error) \\ &\approx \sqrt{1 - 2^{m+1-n}} \frac{\delta}{\sqrt{6\pi}} \end{aligned} \tag{11}$$

The corresponding quantity for errors arising at the $(m-1)$st layer is

$$E(total\ error_{m+1}) \approx \sqrt{1 - 2^{m-n}} \frac{\delta}{\sqrt{6\pi}}$$

By examining (11), we observe that demand errors at higher levels (that is, closer to the root) are expected to cause a larger error in the final allocation of supply.

Figure 1: Shallow vs. Deep Architectures

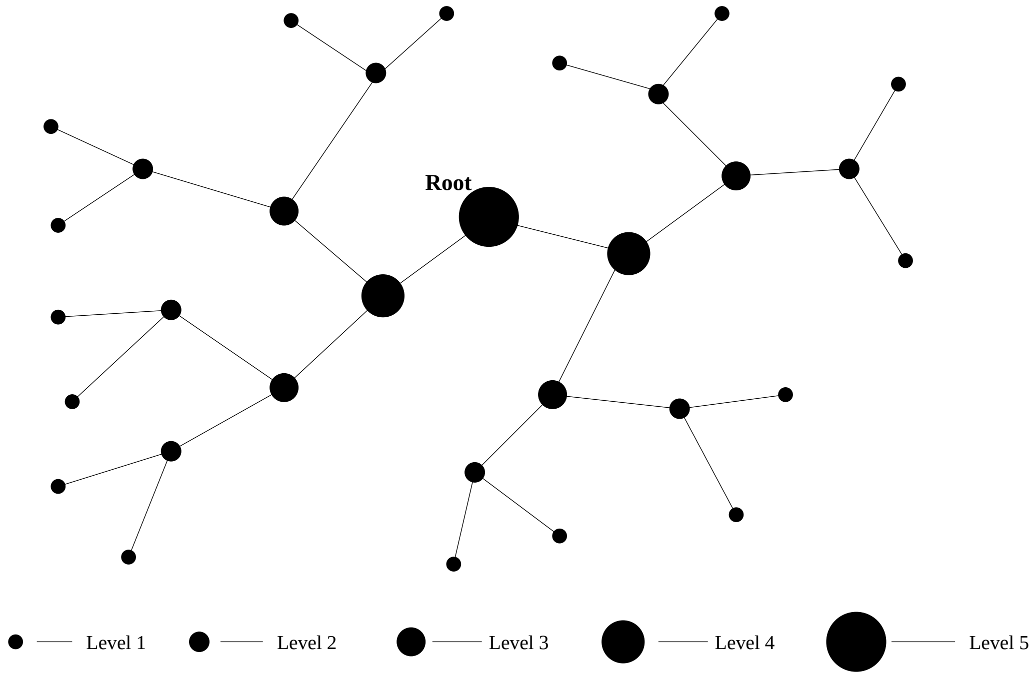

Figure 2: A Sample Hierarchical Agent Organization

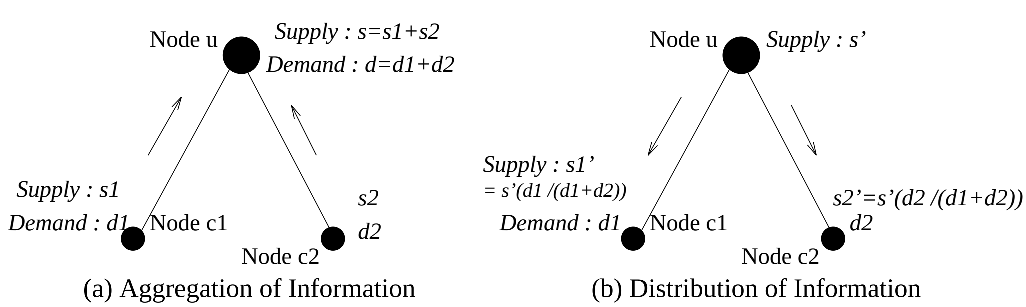

Figure 3: Aggregation and Distribution of information

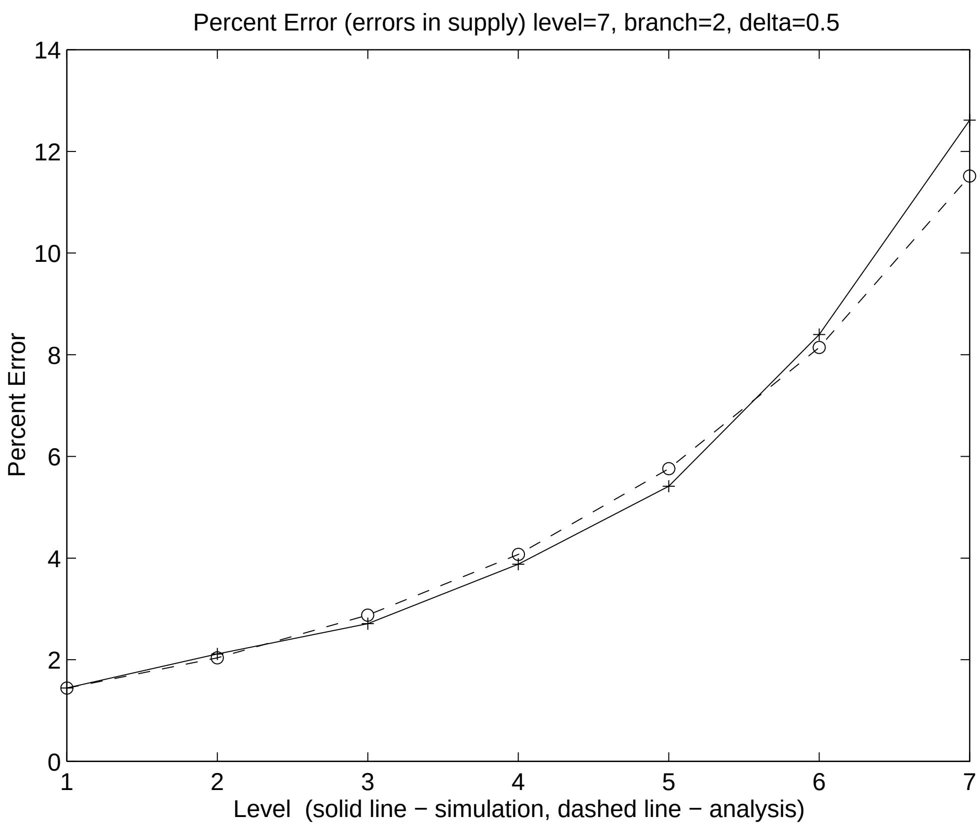

Figure 4: Error as a Function of Decision Level (Supply information)

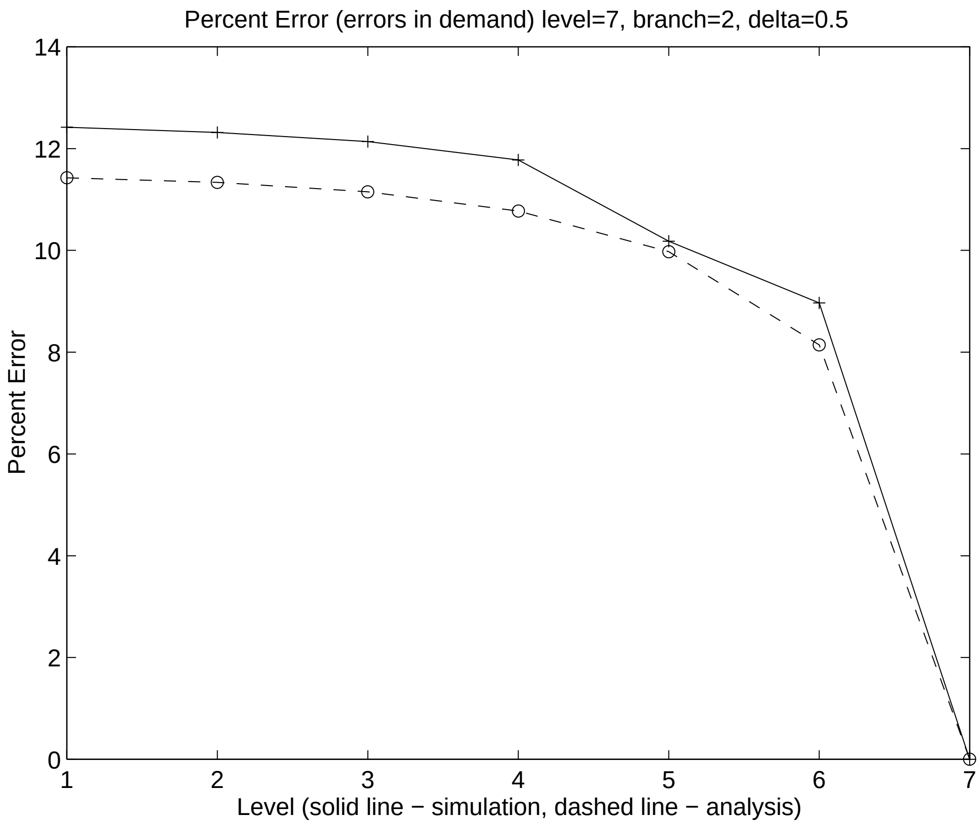

Figure 5: Error as a Function of Decision Level (Demand information)

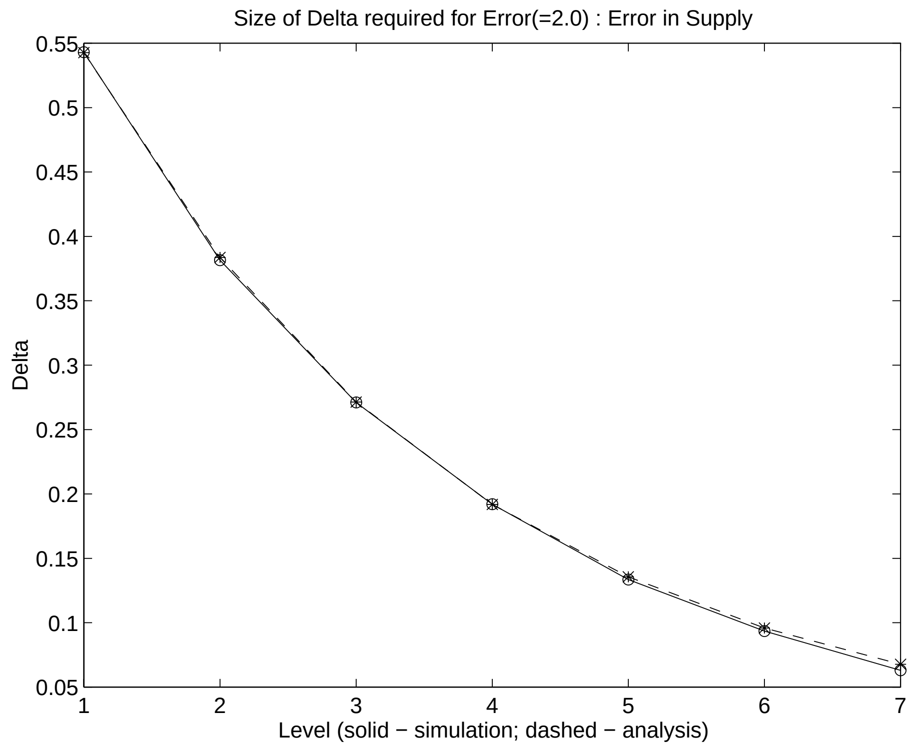

Figure 6: Error bound of supply error for a fixed total error

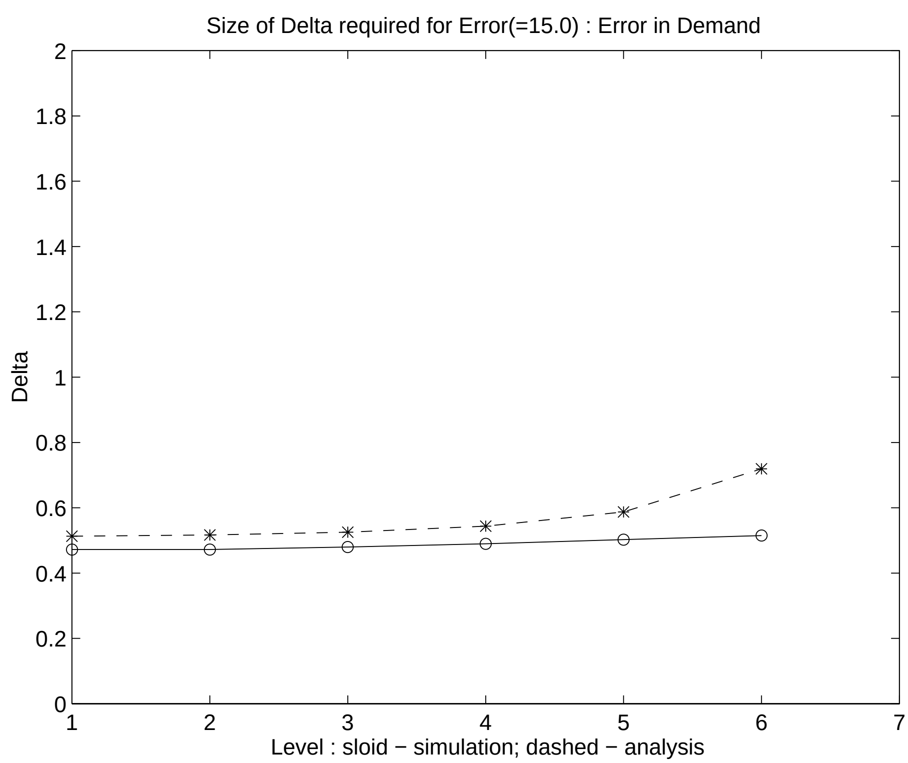

Figure 7: Error bound of demand error for a fixed total error

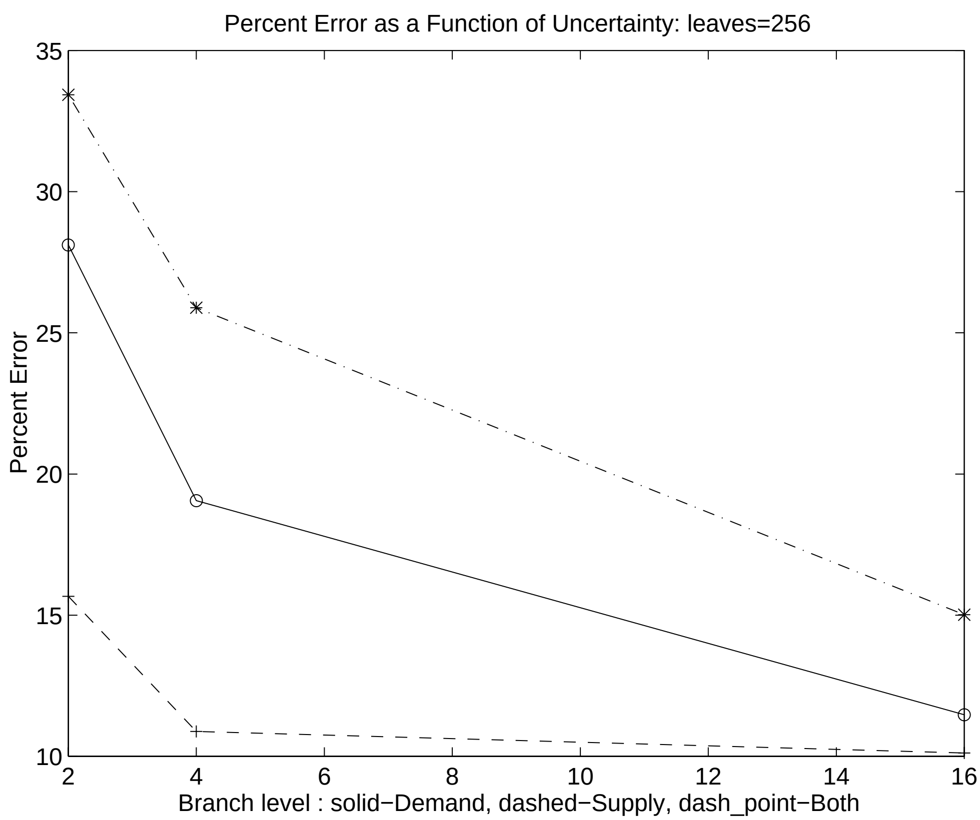

Figure 8: Error as a Function of Branching Factors – 2,4 and 16