ELSEVIER

The Journal of Systems and Software 56 (2001) 231–246

The Journal of Systems and Software

www.elsevier.com/locate/jss

Autonomous agents for coordinated distributed parameterized heuristic routing in large dynamic communication networks

Armin R. Mikler [a,*], Vasant Honavar [b], Johnny S.K. Wong [b]

[a] *Department of Computer Sciences, University of North Texas, P.O. Box 311366, Denton, TX 76203-1366, USA*
[b] *Department of Computer Science, Iowa State University, Ames, IA 50011, USA*

Received 6 February 1999; received in revised form 24 July 1999; accepted 26 May 2000

Abstract

Parameterized heuristics offers an elegant and powerful theoretical framework for design and analysis of autonomous adaptive traffic management agents in communication networks. Routing of messages in such networks presents a real-time instance of a multi-criterion optimization problem in a dynamic and uncertain environment. This paper describes the analysis of the properties of heuristic routing agents through a simulation study within a large network with grid topology. A formal analysis of the underlying principles is presented through the incremental design of a set of autonomous agents that realize heuristic decision functions that can be used to guide messages along a near-optimal (e.g., minimum delay) path in a large network. This paper carefully derives the properties of such heuristics under a set of simplifying assumptions about the network topology and load dynamics and identify the conditions under which they are guaranteed to route messages along an optimal path, so as to avoid hotspots in the load landscape of the network. The paper concludes with a discussion of the relevance of the theoretical results to the design of intelligent autonomous adaptive communication networks and an outline of some directions of future research.

1. Introduction

With the unprecedented growth in size and complexity of modern communication networks, the development of intelligent and adaptive approaches to system management (including such functions as routing, congestion control, traffic/load management, etc.) have assumed considerable theoretical as well as practical significance. Knowledge representation and heuristic techniques (Pearl, 1984) of artificial intelligence, decision-theoretic methods, as well as techniques of adaptive control offer a broad range of powerful tools for the design of intelligent, adaptive, and autonomous communication networks. This paper develops and analyzes heuristic decision functions in support of adaptive routing in large high-speed communication networks.

Routing (Bertsekas and Gallager, 1992) in a communication network refers to the task of propagating a message from its source towards its destination. For each message received, the routing algorithm at each node must select a neighboring node to which the message is to be sent. Such a routing algorithm may be required to meet a diverse set of often conflicting performance requirements (e.g., average message delay, network utilization, etc.), thus making it an instance of a multi-criterion optimization problem.

For a network node to be able to make an optimal routing decision, as dictated by the relevant performance criteria, it requires not only up-to-date and complete knowledge of the state of the entire network but also an accurate prediction of the network dynamics during propagation of the message through the network. This, however, is impossible unless the routing algorithm is capable of adapting to network state changes in almost real time.

Consequently, routing decisions in large communication networks are based on imprecise and uncertain knowledge of the current network state. This imprecision is a function of the network dynamics, the memory available for storage of network state information at each node, the frequency of, and propagation delay associated with, update of such state information. Thus, the routing decisions have to be based on knowledge of

* Corresponding author. Tel.: +1-940-565-4279; fax: +1-940-565-2799.
E-mail address: mikler@cs.unt.edu (A.R. Mikler).

0164-1212/01/$ - see front matter
PII: S0164-1212(00)00100-X

network state over a local neighborhood supplemented by a summary of the network state as viewed from a given node. Motivated by these considerations, a class of adaptive heuristic routing algorithms have been developed over the past few years (Mikler et al., 1997). Experiments demonstrate routing by autonomous routing agents that embody such algorithms displays several interesting properties including: automatic load balancing and message delay minimization. The results presented in this paper constitute a step toward the development of a theoretical framework for the design and the analysis of self-managing communication networks that are managed by interacting, proactive, and reactive, autonomous intelligent agents.

The rest of is paper is organized as follows. Section 2 briefly describes the Quo Vadis framework (Mikler et al., 1997, 1998) for heuristic routing in large communication networks. Section 3 presents some of the results of simulation experiments which motivated the theoretical analysis presented in this paper. Section 4 presents the design and analysis of various routing heuristics with the emphasis on hotspot avoidance. Section 5 concludes with a discussion of the relevance and limitations of the main results and some directions for further research.

2. A framework for heuristic routing

Any intelligent traffic management mechanism capable of performing in a large communication environment must include an effective knowledge representation (KR) mechanism as well as an efficient knowledge acquisition (KA) engine, that minimizes the overhead that is associated with acquiring and maintaining network state information. In addition, adaptive decision making methods are needed which are designed to optimize the network performance.

The underlying framework for heuristic routing, consists of two closely coupled modules: the KR module which is primarily responsible for maintaining and updating the network state information as viewed from each node; and the decision module which implements routing and control algorithms. Both these modules instantiate a family of parameterized heuristics that follow from the design philosophy of Quo Vadis (Mikler et al., 1997). The objective of using parameterized heuristics is to adapt the behavior of the routing function to the dynamics of the network without having to propagate complex network state information. Decision making based on parameterized heuristics is realized by relatively simple, autonomous, proactive as well as reactive agents that are situated at the individual nodes. The collection of such agents form a multi-agent system which displays several interesting emergent properties including load balancing.

2.1. Knowledge representation mechanisms

The KR mechanism in Quo Vadis is designed to provide to the routing agent at each node, a locally computed view that includes precise information about the node itself, supplemented by a spatially and temporally averaged summary of the state of the network as viewed from that node.

The routing agents in this framework do not maintain routing tables. Thus, they lack information about network topology which is implicitly encoded in the routing tables. Since the discussion in this paper is restricted to networks that have regular topologies (e.g., the grid) an alternative scheme was used for addressing nodes and for computing their positions relative to each other. Each network node is assigned a unique coordinate which reflects its location in the grid. Thus, each node n_i is addressed by its respective coordinates (x_i, y_i). Although this paper assumes the network topology to be a regular grid, it is possible to embed network with other topologies into a grid.

The routing agent at node n_i maintains a view $V_i(t)$ of the network from its vantage point at time t. This view is decomposed into four components, one for each of the four directions – north, south, east, and west. Thus we have: $V_i(t) = [V_i^{\mathrm{N}}(t), V_i^{\mathrm{S}}(t), V_i^{\mathrm{E}}(t), V_i^{\mathrm{W}}(t)]$. Each element $V_i^d : (d \in \{\mathrm{N, S, E, W}\})$ of the view $V_i(t)$ is computed from information received from neighbors n_k. This information consists of the corresponding view components $V_k^d(t-\tau)$ (where τ is the interval between view updates) together with local measurements $\rho_k(t)$, indicating the load at n_k.

The view component $V_i^d(t)$ at node n_i at time t is given by

$$V_i^d(t) = \sum_{n_k \in H_i} \alpha \times \rho_k(t) + (1-\alpha) \times V_k^d(t-\tau); \quad 0 < \alpha \leqslant 1, \tag{1}$$

where H_i represents the set of all neighboring nodes n_k of n_i. Here, the parameter α determines the degree to which the effects of an event (i.e. load change) can impact routing decisions at other network nodes. α governs the relative significance attached to the local measurements as opposed to the (spatially and temporally averaged) global view of the network as seen from the node in question. α is a candidate for adaptation to cope with changes in network dynamics (see Section 3).

2.2. Routing and control mechanisms

Upon receiving (or generating) a message to be routed, the routing agent node n_i makes a routing decision based on the destination of the message and its current routing information. The task of the routing agent at n_i is to forward the message through one of its neighbors, such that the resulting path of the message optimizes some desired performance criteria (e.g., average path length, average delay, or other suitable routing metrics). The routing agent at node n_i does this by selecting one of the nodes in its neighborhood H_i that appears to best serve this objective. Choosing the best neighbor is based on the use of an evaluation function (in much the same spirit as the heuristic evaluation functions used in state space search in artificial intelligence problems (Pearl, 1984)). The routing agent at node n_i computes the utility U_k of each node $n_k \in H_i$ and chooses one that has the largest utility. If two or more nodes in the neighborhood have the same utility, one of them is selected at random.

The utility of a neighbor node n_k, U_k, is a function of two separate components: the load liability L_k which estimates the load likely to be encountered by the message on its way to its destination n_d if it were to be routed through n_k; and the path liability P_k that assigns a value to each neighbor n_k so that neighbors that are closer to the destination of the message being routed reflect lower values of P_k. Note that in practice there may be many different suitable heuristics that will guide packets along an optimal path.

In principle, the overall utility U_k of the node n_k is given by

$$U_k = -(\beta \times P_k + (1-\beta) \times L_k); \quad 0 \leqslant \beta \leqslant 1, \tag{2}$$

where the parameter β determines the emphasis placed on finding the shortest path to the destination relative to the desire of avoiding heavily loaded paths.

Given this general framework for computing the utility of nodes, several different choices exist for the exact form of the expressions used to compute L_k and P_k. We define the load liability L_k of node n_k as follows:

$$L_k = \gamma \times \rho_k(t) + (1-\gamma) \times v_k(t); \quad 0 \leqslant \gamma \leqslant 1, \tag{3}$$

where $v_k(t)$ is the sum of the projections of the appropriate components of the view V_k of the neighbor node n_k onto the vector connecting n_k to the destination node n_d. The tunable parameter γ determines the relative emphasis placed on the load (as measured by $\rho_k(t)$) versus the appropriate projections of $V_k(t)$ (as reflected by $v_k(t)$).

The path liability of a node n_k with respect to a message passing through n_i on its way to a destination n_d is given by

$$P_k = \frac{D_{k,d}}{D_{i,d}} \times \rho_i(t), \tag{4}$$

where $D_{i,j}$ is the Euclidean distance between n_i and n_j. Clearly, choice of a neighbor node that has the smallest P_k biases the decision mechanism to route messages along paths that cover the largest fraction of the remaining distance to the destination (provided other things being equal).

Other formulations that share the spirit of the examples shown above for the calculation of load and path liabilities are certainly possible. In what follows, we present the effects of each of the parameters, α, β, and γ used in the framework and described above. A detailed description of Quo Vadis and the corresponding mechanisms can be found in Mikler et al. (1997).

3. Properties of parameterized routing heuristics

A prototype implementation was used to conduct a number of experiments to explore the effects of the various parameters used in the framework. These experiments were conducted in simple regular $m \times n$ grid networks. We anticipate that more general network topologies might present several additional specific issues that will have to be investigated. However, our primary objective in this paper is to analyze the behavior of routing mechanisms based on parameterized heuristics within a relatively simple setting.

A detailed description of the simulation environment can be found in Mikler et al. (1998). Initial experiments were focused on the study of the effects of the parameters α, β, and γ, as used in Eqs. (1)–(3) in an $m \times n$ grid network.

3.1. Routing in the presence of single hotspots

The parameter α determines how the composite load landscape of the network is captured by the views maintained by the individual routing agents. Therefore, both the distance over which a specific load condition can have impact on routing decisions as well as the extent of this impact are governed by α.

As all parameters in the framework are tightly coupled, a demonstration of the effects of α with respect to the view computation required the decoupling of the knowledge representation from the routing mechanism. For this simulation experiment, a 10×10 grid network was set in a particular state corresponding to a pre-determined load distribution. The underlying motivation of this approach is to statically model various load conditions and to determine their impact on the view $V_i(t)$ as acquired by node n_i. In order to eliminate the effects of routing decisions on the load distribution in the network, single hotspots were created at individual nodes by generating self-traffic at a constant rate. Thus, messages did not have to be routed among network nodes but could be *delivered* to the node itself at a node's service rate. As a consequence, the values for parameters β and γ were rendered irrelevant for this experiment. The network together with its corresponding load distribution is shown in Fig. 1. While the a single hotspot may appear to be somewhat restrictive, it models many

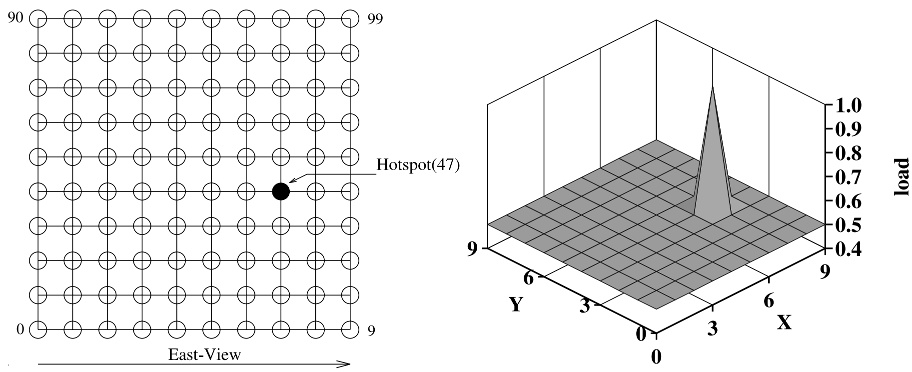

Fig. 1. 100 node network with its corresponding load landscape.

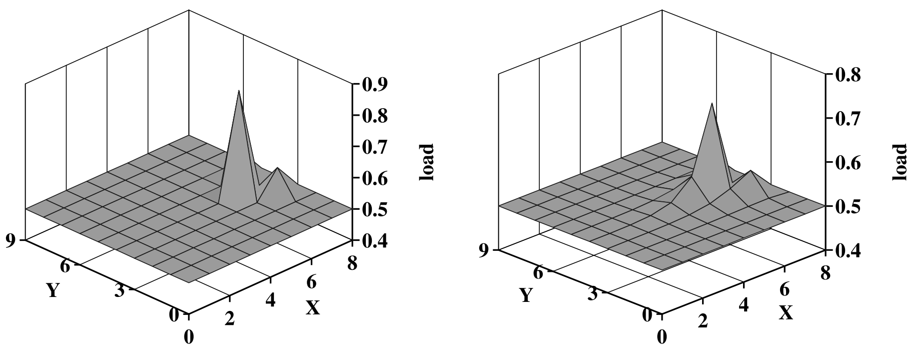

Fig. 2. V_i^E for $\alpha = 1.0$ and $\alpha = 0.6$, respectively.

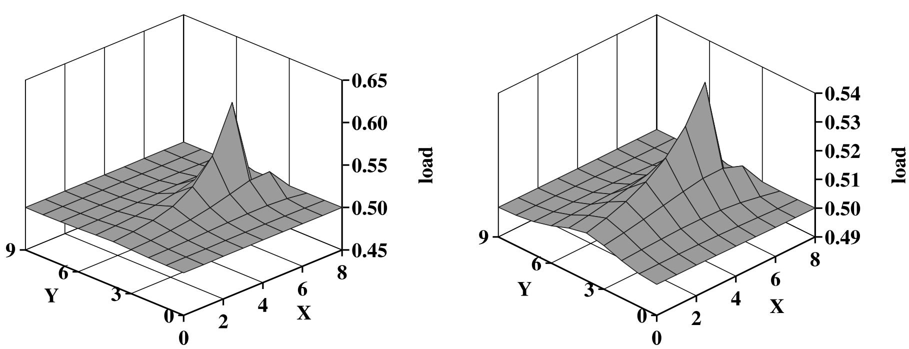

Fig. 3. V_i^E for $\alpha = 0.3$ and $\alpha = 0.1$, respectively.

adverse situation, such as node failures, link failures, local congestion, etc.

Adverse load conditions (hotspots) were simulated by increasing the message generation rate at a single node (or a small number of nodes). Since no messages were sent across the network, the only information communicated among network nodes was view and load information. The views in each of the four directions (East, West, North, and South) as acquired by the individual routing agents were computed after the the convergence of view computation. This experiment was repeated for different values of α. Figs. 2 and 3 show the east-views, V_i^E, as acquired at every node n_i in the network after T seconds of simulation for different values of α. It should be noted that Figs. 2 and 3 do not display view values for nodes $\{9, 19, 29, 39, 49, 59, 69, 79, 89, 99\}$, as the east-views in these nodes are undefined.

From Eq. (1) it is apparent that for $\alpha = 1.0$, the routing agent at node n_i computes its east-view V_i^E solely as the weighted average of local load values ρ_j obtained from neighbor nodes $n_j \in H_i$. The views, V_j^E, computed in neighbors n_j do not contribute to V_i^E. For smaller values of α (i.e., $\alpha = 0.6$), Eq. (1) takes the view V_j^E of neighbors into account thus computing V_i^E as an average of view and load measures of nodes in an extended neighborhood. That is, network nodes $n_k \notin H_i$ affect the magnitude of V_i^{E}. These effects are clearly displayed in Fig. 2. As $\alpha \to 0$ a load condition in a single node n_k affects the views maintained by a larger set of routing agents. However, the magnitude of impact on the view V_i^{E} is significantly reduced. Fig. 3 shows the change of magnitude as a function of distance from n_k.

How the individual views V_i^d, can be used to optimize performance in an anticipatory fashion is further highlighted in the study of effects of parameter γ.

3.2. Compromizing between distance and delay

For the study of the effects of β on the selection of routes, message routing was simulated in a 1024-node grid network. The destination nodes for messages are chosen at random during message creation. Each node in the network has equal probability of being selected as the destination node for a particular message. Self-traffic, however, is prohibited. It is further assumed that links have sufficient bandwidth so that transmission delays are negligible. Message delays are thus assumed to be caused solely by queuing delays encountered in network nodes. A more detailed description of the simulation setup for the study of the effect of β is presented in Mikler et al. (1997).

The following simulation results clearly demonstrate the effects of parameter β. The parameterized routing heuristics used in the framework selects routes so as to reactively as well as pro-actively avoid highly utilized network areas. This behavior is governed primarily by the setting of the parameter β in Eq. (2). To isolate the effect of β on the performance of the decision mechanism other parameters were maintained constant ($\alpha = \gamma = 0.5$).

3.2.1. Shortest path vs. load sensitive routing

From Eq. (2) it is apparent that choosing parameter $\beta = 1.0$ forces the routing mechanism to select routes so as to optimize with respect to the path liability by minimizing the remaining distance to the destination node. This is equivalent to what is generally referred to as *shortest path routing*. In a grid topology, the number of shortest paths between a node n_i and the destination node n_d depends on their relative hop-distance. As one might expect, not all nodes in the grid network experience the same amount of traffic. In fact, nodes in the center of the grid network have to route a larger number of messages on average as compared to nodes at the fringes of the grid. This is due to the fact that a larger number of shortest paths between randomly chosen source–destination pairs pass through nodes in the center of the grid. The corresponding load-graph is shown in Fig. 4.

As the message delay in a network node increases exponentially with its load, it follows that nodes in the center of the grid contribute most to the overall message delay along path traversed by the message. Thus, load at these nodes impacts the total message delay to a much higher degree than nodes at the fringes of the grid. This effect is amplified as the average network load increases. A heuristic routing function with the appropriate parameter setting delays the onset as well as reduces the impact of this effect given an appropriate setting of β. While a shortest path routing algorithm makes a random decision among neighbors with equal *path utility* (Eq. (4)), taking network load into account, biases the selection towards neighbors with better *load utility* (Eq. (3)). The price paid for the ability to circumvent a highly utilized network area is an increase in mean path length $\bar{h}$.

Table 1 indicates the existence of an optimal value for β, β^*, that minimizes the mean message delay. An increase in the mean delay is observed for $\beta < \beta^*$ as the routing decisions are dominated by the *load liability* L_k. For $\beta \ll \beta^*$ the behavior of the mechanism may approach that of random routing. For $\beta > \beta^*$, the routing mechanism approaches shortest path routing thereby causing an increased mean message delay as discussed above.

The load distribution in the network with different values of β is shown in Fig. 4.

Clearly, a load sensitive setting of β results in a more balanced distribution of load, thus preventing a single network area from becoming overutilized. If load vigilance is high (i.e. small β), routing decisions may result in extended path length. However, this does not necessarily lead to an increase in total message delay along the path if the message is routed through a lightly loaded

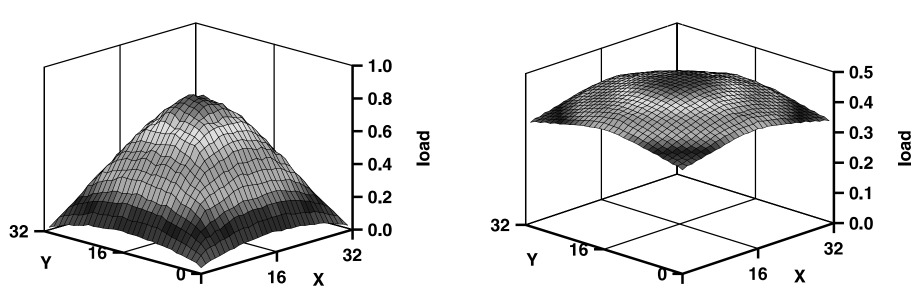

Fig. 4. Load distribution in a 1024 node grid network using shortest path routing, i.e., $\beta = 1.0$ and $\beta = 0.4$, respectively.

Table 1
Mean hop count ($\bar{h}$) and mean message delay ($\bar{d}$) for different values of β ($n > 85\,700$ messages)

β	$\bar{h}$	$\bar{d}$
0.3	23.07	2.43
0.4	22.76	2.41
0.5	22.44	2.36
0.6	22.15	2.34
0.7	21.89	2.33
0.8	21.58	2.35
0.9	21.33	2.51
1.0	21.29	2.79

area. The exponential increase in delay with increasing load justifies such a tradeoff.

3.2.2. Autonomous load balancing properties

The term hotspot refers to a single node or a small group of nodes in the network that experience a sudden increase in utilization. Such hotspots may be caused due (among other things) to:

- localized increases in arrival rate, or
- localized node or link failures.

One of the desirable properties of a routing mechanism is its ability to react to such load changes. A good routing algorithm should attempt to route messages around the hotspot, thereby reducing the message delay, perhaps at the expense of increasing the total length of the route.

The ability to adapt to such localized load changes quickly has been deliberately designed into the routing mechanism. Nodes in the neighborhood of a suddenly *over-utilized* node start to divert traffic as soon as the load increase is made known to them. High load in an affected node (as in highly loaded network areas) has a repulsive effect on traffic and routing decisions are automatically biased towards avoiding that node. Again, the extent of this bias is determined by β. Such dispersion of traffic is accomplished with minimal impact on nodes that are sufficiently distant from those that are affected by local increases in load.

While the increase in a node's load should clearly repel messages from being routed though it, a sudden load decrease should be utilized by nodes in the neighborhood in their effort to distribute network load more uniformly.

Sudden load changes were simulated by increasing and decreasing a node's service rate. The effects of such a change when shortest path routing is in place are shown in Fig. 5. The effects of adaptive measures taken when the parameterized heuristics represent a higher load vigilance are shown in Fig. 6.

Shortest path routing (i.e., $\beta = 1.0$) does not attempt to reduce the influx of traffic into the affected area in order to normalize the load conditions at the hotspot. Heuristics tuned towards load sensitive routing, however, tend to balance load conditions in the network in a relatively short time. This is accomplished by the dispersion of traffic which would otherwise have been routed through the hotspot area.

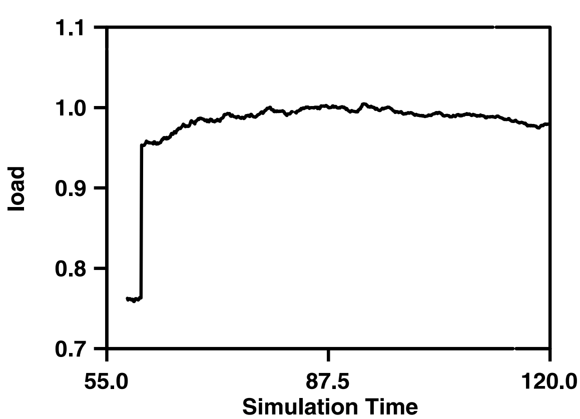

Fig. 5. Effects of sudden load increase in node n_i under shortest path routing.

3.3. From local information to a global view

In Eq. (3), γ defines the significance of load measures ρ_k versus v_k, the projections of a node's view $V_k(t)$ with respect to a particular destination. The underlying motivation is to enable network nodes to make routing decisions in either reactive or anticipatory fashion. For $\gamma = 1$, only ρ_k determines the load liability of n_k, thereby enabling n_i to route messages so as to circumvent the

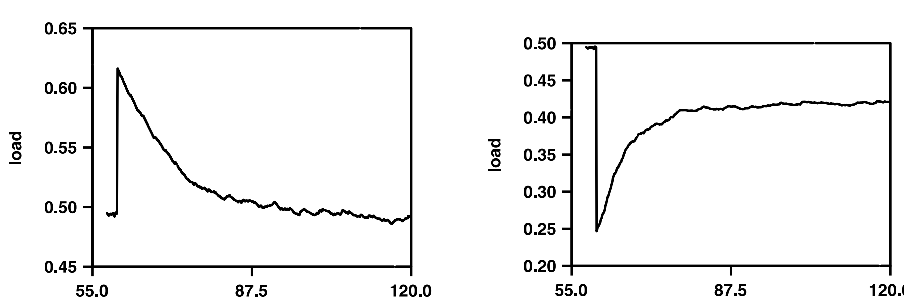

Fig. 6. Effects of load sensitive routing on sudden load changes in node n_i.

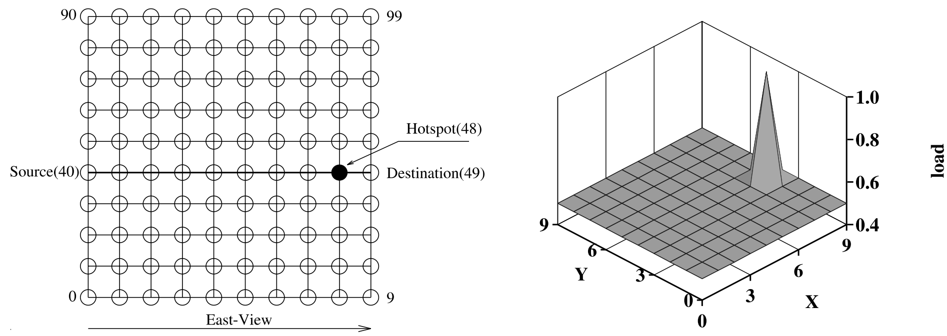

Fig. 7. 100 node network with its corresponding load landscape.

neighbor node $n_k \in H_i$ with the highest utilization, thus reacting to adverse load conditions in the immediate neighborhood. On the other hand, for small values of γ (i.e., $\gamma \to 0$) node n_i will base its evaluation of neighbors n_k on a load summary as represented by $V_k(t)$ with respect to the relative location of the destination. Hence, adverse load conditions on the path towards the destination can be sensed by n_i so as to adjust the routing decision.

As for the evaluation for α, the isolation of the effects of γ required the network to remain in a pre-determined state. The corresponding network and load graph are shown in Fig. 7.

In addition, nodes 40 and 49 were selected to serve as source and destination nodes for a single message which is traced on its journey through the network. The purpose of the trace is to identify all nodes that are visited by the message thus revealing the routing decisions made by intermediate nodes. This experiment was repeated for various values of γ. Since β controls the significance of the load liability, it was chosen so as to amplify the effects of γ, i.e., β was maintained constant at 0.2. The value of α was set to 0.3, thus making the effects of adverse load condition visible at distant nodes.

The different routes traveled by a test message are presented in Table 2 for various values of γ. Clearly, the shortest path between source node 40 and destination node 49 is given by (40, 41, 42, 43, 44, 45, 46, 47, 48, 49). However, the high utilization of node 48 forces the route to deflect. The nodes at which deflection occurs are shown in **bold**. Table 2 shows that for large values of γ deflection takes place only when adverse load conditions are encountered in the immediate neighborhood; (i.e. n_{47} deflects as $n_{48} \in H_{47}$ experiences a high utilization.) Small values of γ, force Eq. (3) to attach a higher significance to the view projection $v_k(t)$, which reflects the adverse load conditions at node 48. As a consequence, nodes can take anticipatory action and deflect earlier.

Table 2
Points of deflection for different values of γ

γ	Route
0.0	(**40**, 30, 31, 32, 33, 34, 35, 36, 37, 38, 48, 49)
0.1	(40, **41**, 31, 32, 33, 34, 35, 36, 37, 38, 39, 49)
0.2	(40, **41**, 31, 32, 33, 34, 35, 36, 37, 38, 39, 49)
0.3	(40, 41, **42**, 32, 33, 34, 35, 36, 37, 38, 39, 49)
0.4	(40, 41, 42, **43**, 33, 34, 35, 36, 37, 38, 39, 49)
0.5	(40, 41, 42, **43**, 33, 34, 35, 36, 37, 38, 39, 49)
0.6	(40, 41, 42, 43, 44, **45**, 55, 56, 57, 58, 59, 49)
0.7	(40, 41, 42, 43, 44, 45, 46, **47**, 57, 58, 59, 49)
0.8	(40, 41, 42, 43, 44, 45, 46, **47**, 57, 58, 59, 49)
0.9	(40, 41, 42, 43, 44, 45, 46, **47**, 57, 58, 59, 49)
1.0	(40, 41, 42, 43, 44, 45, 46, **47**, 57, 58, 59, 49)

The experimental results summarized here demonstrate the promise of control and decision mechanisms based on parameterized heuristics for routing in large communication networks. This raises the question as to whether it is possible to come up with a more systematic characterization of the properties of these heuristics to shed more light on the experimental results as well as to guide the design of similar heuristics. In the following section we will develop theoretical framework based on utility theory and decision theory for the design and implementation of intelligent agents for routing and control.

4. Design and analysis of parameterized heuristics

Routing messages in large communication networks so as to optimize some desired set of performance criteria presents an instance of resource-bounded, multi-criteria, real-time, optimization problem. Our proposed solution to this problem involves the use of *utility-theoretic heuristics* (Mikler et al., 1996). *Utility* is a measure that quantifies a decision maker's preference for one action over another (relative to some criteria to be maximized) (French, 1986). When the result of an action is uncertain, it is convenient to use the *expected* utility of each action to pick actions which maximize the expected utility. The heuristic function enables the routing agent at each node n_j in the network to select a *best* neighbor in its neighborhood to route a message M (which it has received or generated) towards its destination.

The utility U_i^d of node n_i (with respect to a destination n_d) is computed by the routing agent at a neighboring node, n_j, as n_j attempts to route a message M that it has received, along a desired (e.g., minimum delay) path, to M's destination, n_d. A routing agent at node n_j preference-orders its neighbors n_i according to their respective utilities. We say that the routing agent at n_j is *indifferent with respect to* the choice between two neighbors n_k and say n_l if $U_k^d = U_l^d$ (where n_d is the destination of the message M being routed by n_j). We denote the indifference between two nodes by $n_k \sim n_l$. We say that a neighboring node n_k is preferred by the routing agent at n_j over another neighbor n_l if $U_k^d > U_l^d$. We denote this preference by $n_k \succ n_l$.

For the purpose of the analysis that follows, it is assumed that the network is a regular rectangular grid (with adjacent nodes being at unit distance from each other). Additional assumptions concerning load and load dynamics are made as necessary. A suitably defined *reward* function provides the directional guidance necessary to route each message towards its destination.

In the regular grid network, let $D_{i,d}$ denote the Manhattan distance between a node n_i and n_d. Other topologies may require the use of other distance measures. We define the *partial reward* for node n_i as $R_i^d = f_R(D_{i,d})$, where f_R is a *reward function* chosen such that $\forall i \forall j \; D_{i,d} \leqslant D_{j,d} \iff f_R(D_{j,d}) \leqslant f_R(D_{i,d})$.

There are many possible choices for the reward function $f_R(\cdot)$. A particular example of $f_R(\cdot)$ is given by $f_R(D_{i,d}) = (m+n) - D_{i,d}/(m+n)$, where n and m are the dimensions of the grid network (see Fig. 8). Note that the results that follow are independent of particular choices of $f_R(\cdot)$ so long as the reward is an inverse function of the distance to the destination.

We define a *cumulative reward* R^P obtained by a message M traveling along a path P (from its source n_s to its destination n_d) as $R^P = \sum_{n_i \in P} R_i^d$. At each node n_i along path P, the delay encountered by a message M is modeled by a non-negative, bounded cost C_i. That is, $\forall i, \; 0 \leqslant C_i \leqslant \xi$. It is further assumed that the penalty C_i remains constant during the time it takes to make a routing decision for message M at node n_i. If cumulative delay is to be minimized, a natural interpretation of C_i is the delay (on account of load) at n_i. However, since delays can become unbounded when there is queuing, it may be necessary to discard some messages in order to keep the delay bounded at the expense of message loss. If cumulative load is to be minimized, C_i is guaranteed to be bounded by the maximum utilization $\rho \leqslant 1$.

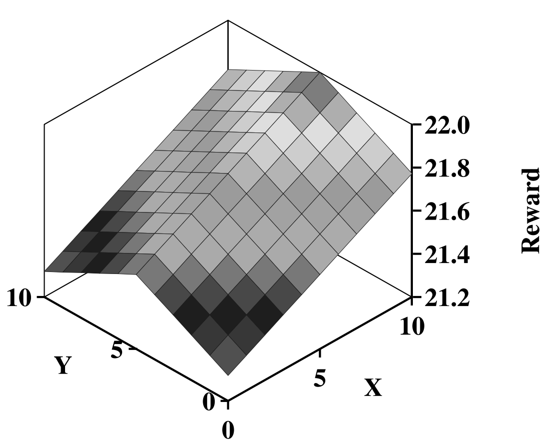

Fig. 8. Reward function $f_R(D_{i,d}) = (m+n) - D_{i,d}/(m+n)$ for destination $n_d = (10, 5)$.

The total cost incurred by a message along a path P is given by $C^P = \sum_{n_i \in P} C_i$. We can now define the net *partial payoff* Z_i^d received by a message M when it reaches the node n_i on its way to its destination n_d as $Z_i^d = R_i^d - C_i$. Correspondingly, the total payoff along a path P is given by $Z^P = R^P - C^P$. Note, that the payoff received at a node n_i may not be the same for different destinations. Hence, Z_i^d represents the payoff with respect to a particular destination node n_d. Let Π be a minimum cost path from a source n_s to a destination n_d. The cost C^Π along this path is given by $C^\Pi = \min_{\forall P}\{C^P\}$.

In the discussion that follows, in order to simplify our analysis, we proceed under the assumption that the network is uniformly loaded. This assumption is captured by the following definition.

Definition 1. If $\forall i, \; C_i = \kappa \; (0 \leqslant \kappa \leqslant \xi)$, we refer to the network as a *uniform cost* network.

Lemma 1. *In a uniform cost network, a simple utility function $U^0 = Z_i^d$ is sufficient to route each message along a minimum cost path to its destination.*

Proof of Lemma 1. Since in a uniform cost network, $\forall i, \; C_i = \kappa$, the partial reward Z_i^d can be written as $Z_i^d = R_i^d - \kappa$. Thus, Z_i^d can be maximized at each intermediate node along path P simply by maximizing R_i^d. Let Λ_P be the number of nodes on a path P. As message M is propagated along a P such that R_i^d is maximized at every intermediate step, in a regular grid network, the property of the reward function (i.e., $\forall i \forall j \; D_{i,d} \leqslant D_{j,d} \iff f_R(D_{j,d}) \leqslant f_R(D_{i,d})$) guarantees that M is propagated along *a shortest path* (as measured by the number of hops) P from the source n_i of the message M to its destination n_d and thus we have $C^P = \kappa \Lambda_P$. Since P is a minimum hop (shortest) path, it follows that $C^P = C^\Pi$.

Here, the uniform cost assumption renders the cost component in the payoff function irrelevant for making the routing decisions. This assumption is no longer valid when the network is not a uniform cost network. In what follows, we relax the uniform cost assumption by allowing a single *hotspot* (a node with a high load relative to its neighbors) in an otherwise uniform cost network.

4.1. Routing in presence of a single hotspot

Definition 2. A hotspot, n_h, in an otherwise uniform cost network is a single network node which has a higher load than its neighbors so that a message M traveling through it incurs a cost $C_h > \kappa$ (where $C_i = \kappa \; \forall i \neq h$).

Note that since the costs C_i are bounded by ξ, it follows that $C_h \leqslant \xi$. Further note that the above definition of a hotspot says nothing about the relative difference in costs C_h and C_i. A more realistic definition of a hotspot might require that the cost of routing a message through a hotspot is *significantly larger* than that of routing the same message through a node in the neighborhood of the hotspot. Also, when a network deviates substantially from the uniform cost assumption, it is more useful to focus on the load distribution in the vicinity of a node rather than hotspots. However, to make the analysis mathematically tractable, the discussion that follows focuses on routing in an otherwise uniform cost networks with a single hotspot.

As the uniform cost assumption is relaxed by allowing a single hotspot n_h with cost $C_h > C_j \; \forall j \neq h$ in the network, it is easy to show that relying on partial payoffs alone as utilities for routing messages can result in suboptimal routes. Consider a grid network with node coordinates increasing as a message M travels east and south. From the uniform cost assumption, we have $C_i = C_j = \kappa \; \forall i, j \neq h$. Let x_s, y_s, x_d, and y_d be the x and y coordinates of M's source and destination, respectively. Let x_h and y_h be the x and y coordinates of a hotspot in one of the following configurations:

1. $x_s \leqslant x_h \leqslant x_d \wedge y_s \leqslant y_h \leqslant y_d$,
2. $x_s \geqslant x_h \geqslant x_d \wedge y_s \geqslant y_h \geqslant y_d$.

Here, the probability that a shortest path from n_s to n_d passes through the hotspot n_h is non-zero. That is, unless n_h coincides with either n_s, n_d, or both, there exists a node n_i in the neighborhood of hotspot n_h that must decide how to route M so as to minimize the total cost incurred by M. As we show below, if this decision is based on a preference ordering induced by the naive utility function $U^0 = Z_i^d$, messages can be routed through the hotspot thereby incurring a higher cost than they would have otherwise.

Assumption 1. For the discussion below, we assume that the reward functions chosen guarantee that for any two nodes, n_k, n_i, in the network the difference in reward with respect to destination n_d is greater than ξ (i.e., $|R_i^d - R_k^d| > \xi$), whenever $D_{i,d} \neq D_{k,d}$.

Assumption 1 ensures that the cost C_i of a node n_i, (and n_h in particular) does not offset the *guidance* provided through R_i^d unless two nodes with equal rewards are being compared.

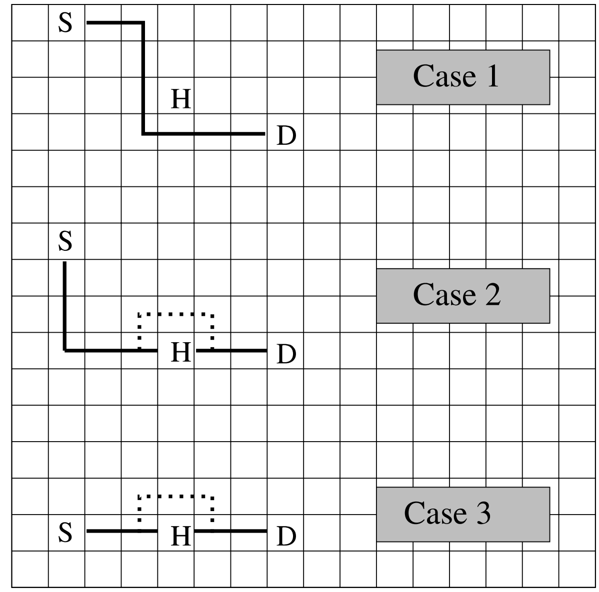

Fig. 9. Sample node placement.

In the following we distinguish four canonical cases (see Fig. 9). We focus in our analysis on configuration 1 above. Similar arguments hold for configuration 2.

Case 0. This case combines four scenarios of placing nodes n_s, n_d, and n_h in the grid network, each of which presents a trivial routing problem. In these scenarios, at least two of the nodes n_s, n_d, and n_h are identical. That is, $n_s = n_d = n_h$, $n_s = n_d$, $n_s = n_h \neq n_d$, and $n_s \neq n_h = n_d$. Clearly, in the first two scenarios, no routing decisions are needed as the message source coincides with the destination. Whenever the message source coincides with the hotspot as in the third scenario, the routing algorithm will select a neighbor $n_k \in H_i$ with the highest utility. Hence, the routing algorithm performs as in the case of a uniform cost network (without hotspots). For the fourth scenario, Assumption 1 assures that n_d yields the highest partial reward R_i^d, $\forall i$, despite the fact that the cost incurred by hotspot conditions reduces its partial payoff. Hence, routing decisions can be made without taking cost C_i into consideration, as in the case of a network without hotspots.

Case 1. Let $P\Delta_{i,j}$ denote the number of minimum hop paths from a node n_i to node n_j. This case encompasses all placements of nodes n_s, n_h, and n_d, such that

1. $P\Delta_{s,h} > 1 \wedge P\Delta_{h,d} > 1$ or
2. $P\Delta_{s,h} = 1 \wedge P\Delta_{h,d} \geqslant 1$, where $P\Delta_{s,d} > 1$.

For scenario 1, the hotspot n_h does not share either the x or y coordinates of n_s or n_d. That is, $(x_s < x_h < x_d) \wedge (y_s < y_h < y_d)$. Scenario 2 represents a hotspot placement such that n_h occupies either the same x or y coordinate as n_s.In either scenario, the partial minimum hop paths from n_s to n_h may be part of a minimum cost path from n_s to n_d if all nodes n_i that neighbor the hotspot take action to route M so as to circumvent n_h. Thus, the utility function $U^0 = Z_i^d$ is guaranteed to route M on a minimum cost path to its destination n_d.

Lemma 2. *In a uniform cost network with a single hotspot n_h located such that $(x_s < x_h < x_d) \wedge (y_s < y_h < y_d)$, a routing algorithm which propagates a message M such that U^0 is maximized at every intermediate step will yield an optimal path Π with cost C^Π.*

Proof of Lemma 2. Clearly, the only nodes at which a decision has to be made to circumvent n_h are n_i or n_j $(x_h - 1, y_h)$ and $(x_h, y_h - 1)$, respectively. Since $x_h < x_d \wedge y_h < y_d$, there exist nodes n_k and n_l with coordinates $(x_h - 1, y_h + 1)$ and $(x_h + 1, y_h - 1)$, respectively, that lie on a minimum hop path from n_s to n_d. Since $C_k = C_l = \kappa < C_h$ it follows that $Z_k^d = Z_l^d > Z_h^d$. Hence, a routing decision in n_i or n_j that maximizes the partial payoff will choose n_k or n_l to propagate M towards n_d. Since $C_i = \kappa\ \forall i \neq h$, and M is propagated along a minimum hop path, Lemma 1 guarantees that M is routed along an optimal path Π.

Case 2. Here, n_s, n_d, and n_h are placed such that $(x_s < x_h < x_d) \wedge (y_s < y_h = y_d)$ or $(x_s < x_h = x_d) \wedge (y_s < y_h < y_d)$, i.e.; $(P\Delta_{s,h} > 1) \wedge (P\Delta_{h,d} = 1)$.

Assuming the former, there exists a node n_i with coordinates (x_i, y_i) with $(x_s < x_i < x_h) \wedge (y_i = y_h = y_d)$ from which the number of minimum hop routes $P\Delta_{i,d} = 1$. Since in a uniform cost network $n_k \sim n_l\ \forall k, l \neq h$ the naive utility function U^0 can guide a message M through n_i, thereby committing to a path P with cost $C^P > C^\Pi$. Assuming that M is only routed using utilities to choose among minimum hop routes, the additional cost $(C^P - C^\Pi)$ is inflicted on M by n_h. If M is permitted to deflect from a minimum hop route, the additional cost $(C^P - C^\Pi)$ is inflicted by n_h itself or due to the extended length of P in circumventing n_h. As we will show in Section 4.2, Case 2requires the most intricate design of decision functions in order to circumvent the hotspot and forward the messages on an optimal path.

Case 3. This scenario consists of all placements of n_s, n_d, and n_h such that $(x_s = x_h = x_d) \wedge (y_s \leqslant y_h \leqslant y_d)$ or $(x_s \leqslant x_h \leqslant x_d) \wedge (y_s = y_h = y_d)$. Since there is only a single optimal path Π from n_s to n_d, i.e., $P\Delta_{s,d} = 1$, message M must either visit n_h or deflect from the minimum hop path in order to circumvent n_h. U^0, however is not sufficiently informative to guarantee an optimal routing decision. Hence, M may be routed along a path P for which $C^P > C^\Pi$.

Assumption 2. In the following we assume that a node n_j upon receiving a message M from a neighbor node $n_i \in H_j$ will refrain from propagating M back to n_i.

This is a natural assumption that is meant to avoid the so-called *bouncing of messages* back to a node from which it was routed.

Lemma 3. *In a uniform cost network with a single hotspot n_h, a routing algorithm based on U^0 will deflect a message M at most once in order to circumvent n_h provided bouncing is avoided* (*via Assumption* 2).

Proof of Lemma 3. Consider a node n_i with coordinates (x_i, y_i) such that $x_s < x_i = x_h - 1 < x_d \wedge y_s < y_i = y_h = y_d$ (similar analysis holds for the case where $x_s = x_i = x_h < x_d \wedge y_s < y_i = y_h - 1 < y_d$). Node n_i can deflect M to a node n_j with coordinates (x_j, y_j), such that $x_s < x_j = x_h - 1 < x_h < x_d \wedge y_s < y_j = y_h \pm 1$. Clearly, $P\Delta_{j,h} = 2$. Since $x_h < x_d$, $P\Delta_{j,d} > 2$. Hence there must exist a node n_k with $n_k \in H_j$ which lies on a minimum hop path P from n_j to n_d such that $n_h \notin P$. Our particular choice of the reward function (see above) guarantees that $R_i^d = R_k^d > R_j^d$. In a grid, $\nexists n_l,\ l \neq i, k$ such that $n_l \in H_j \wedge R_l^d \geqslant R_k^d$ (since the reward function ensures that the rewards vary inversely with the Manhattan distance). Since $C_j = C_k = C_i = \kappa$, $Z_i^d = Z_k^d > Z_j^d$. This limits the routing choices for message M at n_j to n_i and n_k, of which, by Assumption 2, n_k has to be chosen (since otherwise M will be bounced back to n_i, which had routed the message to n_j to begin with, thereby violating Assumption 2). This ensures that from n_j, M is sent along a minimum hop path P to the destination n_d. Since $n_h \notin P$, Lemma 1 guarantees that M is propagated along P without further deflection.

The analysis of the performance of a routing algorithm based on U^0 for each of the four cases above yields the following theorem.

Theorem 1. *In a uniform cost network with a single hotspot n_h with $C_h > \kappa$ (where $\forall i \neq h$, $C_i = \kappa$), a routing algorithm which propagates a message M such that U^0 is maximized at every intermediate step is guaranteed to yield a path P with cost C^P such that $C^P - C^\Pi \leqslant$* $\max((C_h - \kappa), 2\kappa)$.

Proof of Theorem 1. 1In Case 1, Lemma 2 guarantees that a routing algorithm based on U^0 will find a minimum cost path if n_s, n_h, and n_d are placed such that $(x_s < x_h < x_d) \wedge (y_s < y_h < y_d)$. Hence, $C^P = C^\Pi$ and thus $C^P - C^\Pi = 0 < \max((C_h - \kappa), 2\kappa)$.

Case 2 involves a node n_i with coordinates (x_i, y_i) such that $x_s < x_i = x_h - 1 < x_d \wedge y_s < y_i = y_h = y_d$ or $x_s = x_i = x_h < x_d \wedge y_s < y_i = y_h - 1 < y_d$. Now n_i must decide whether to route message M through n_h or to deflect M from a minimum hop path. Routing through n_h will result in a path cost C^P which is sub-optimal by an amount $C_h - \kappa$. That is, $C^P - C^\Pi = C_h - \kappa \leqslant$

$\max((C_h - \kappa), 2\kappa)$. If n_i chooses to deflect M so as to circumvent n_h, M is propagated along a path P'. Let Λ_P be the length (in number of hops) of the minimum hop path P from n_i to n_d via n_h and $\Lambda_{P'}$ be the length of path P'. Deflecting from path P in a grid topology yields a path P' with $\Lambda_{P'} = \Lambda_P + 2$. Lemma 3 guarantees that M is deflected at most once, $C^{P'} = C^{\Pi} + 2\kappa$. Hence $C^{P'} - C^{\Pi} = 2\kappa \leqslant \max((C_h - \kappa), 2\kappa)$.

In Case 3, the cost C^{Π} for a minimum cost path Π between n_s and n_d is given by $C^{\Pi} = \Lambda_P \kappa + \min(C_h - \kappa, 2\kappa)$. Hence, $C^{P'} - C^{\Pi} \leqslant \max((C_h - \kappa), 2\kappa)$.

If n_h coincides with either n_s or n_d, the hotspot cannot be circumvented and $C^P = C^{\Pi}$ (i.e., the minimum cost path has to necessarily pass through the hotspot in this case). Clearly, $C^P = C^{\Pi}$ and $0 < \max((C_h - \kappa), 2\kappa)$. Therefore, $C^P - C^{\Pi} \leqslant \max((C_h - \kappa), 2\kappa)\ \forall P$.

4.2. Eliminating sub-optimality using a modified utility function

Sub-optimal routing scenarios as discussed above arise primarily as a result of a lack of knowledge at n_i at the time it is routing a message M to a neighbor n_j, regarding the likely cost of completing the path from n_j to the destination of M, namely, n_d. Source–hotspot–destination configurations corresponding to scenarios described in Cases 2 and 3 can result in sub-optimal routes (i.e., $C^P > C^{\Pi}$) when routing decisions are based on the naive utility function U^0. In what follows, we derive more complex utility/decision functions which would eliminate sub-optimal routing in Cases 2 and 3.

4.2.1. Eliminating sub-optimality in Case 3

In order to eliminate sub-optimal routing in scenarios corresponding to Case 3, additional constraints must be added to the utility function U^0.

Definition 3. Let U^1 be a utility function given by:

$$U^1 = \begin{cases} R_j^d & \text{if } \kappa < C_j < 3\kappa \wedge \nexists k\ (R_j^d = R_k^d) \wedge (n_j \neq n_d), \\ Z_j^d & \text{otherwise.} \end{cases}$$

U^1 exploits the fact that messages are to be routed in a uniform cost network with a single hotspot. If routing decisions are based on the preference ordering induced by U^1 in an otherwise uniform cost network with a single hotspot, every message originating in a source n_s and a destination n_d that correspond to a source–hotspot destination placement described in Case 3 is guaranteed to be propagated along an optimal path Π between n_s to n_d. Using U^1, n_i can decide whether or not to propagate M through a hotspot n_h in its neighborhood or to circumvent the hotspot by routing M through a different neighbor $n_k \neq n_h$. In other words, the preference ordering induced by U^1 ensures that at a node neighboring a hotspot in a Case 3 scenario we have:

- $(C_h - C_k) = (C_h - \kappa) > 2\kappa \Longleftrightarrow n_k \succ n_h$,
- $(C_h - C_k) = (C_h - \kappa) < 2\kappa \Longleftrightarrow n_h \succ n_k$.

Thus all routing decisions based on U^1 in Case 3 scenarios result in optimal (minimum cost) routes. However, it is easy to see that U^1 does not eliminate the possibility of a sub-optimal route in a source–hotspot–destination configurations corresponding to the scenario in Case 2.

4.2.2. Eliminating sub-optimality in Case 2

As shown by the preceding analysis, U^1 can result in a sub-optimal routing decision in a source–hotspot–destination configuration corresponding to the scenario in Case 2. In particular, any routing decision in a configuration corresponding to Case 2 will result in a sub-optimal path P if it results in the propagation of a message M to a node $n_k \in P$ such that $x_k < x_h < x_d \wedge y_k = y_h = y_d$ or $x_k = x_h = x_d \wedge y_k < y_h < y_d$. Routing decisions based on a preference ordering induced by U^1 can lead to such a situation since in a neighborhood H_i of n_i such that $n_h \notin H_i$, $\forall n_j$, $n_k \in H_i$, $n_k \sim n_j$ provided $R_k^d = R_j^d$. Note that Case 2 scenarios include all placements of n_s, n_h, and n_d, such that $\forall\ \{n_i \mid x_i \neq x_d \wedge y_i \neq y_d\}\ \exists\ k, l$, such that $(n_k \in \Pi) \vee (n_l \in \Pi)$.

These observations suggest the possibility of using an estimate of the cost along paths from n_k to n_d as a component of a modified utility function U^2 so as to induce a preference ordering between nodes (where no such preference ordering is induced by U^1) so as to eliminate sub-optimal routing decisions altogether. In other words, U^2 should be able to induce a preference ordering among nodes n_k and n_l in the neighborhood of a node n_i (the node making the routing decision for a message M) such that: $(n_k \in \Pi) \wedge (n_l \notin \Pi) \Rightarrow n_k \succ n_l$. We now proceed to define a cost estimator function E_k^d as follows.

Definition 4. A cost estimator function $E_k^d(\cdot)$ estimates the cost E_k^d of a minimal cost path to a destination n_d from a node n_k.

It would be nice if the cost estimator function defined above helps U^2 to induce the desired preference ordering necessary to guarantee routing along an optimal path in the scenario corresponding to Case 2. We capture this property by defining what are called *admissible* cost estimator functions.

Definition 5. A cost estimator function is said to be *admissible* if $\forall$ nodes n_i in the network, for all nodes n_k, n_l in the neighborhood H_i of n_i, it is guaranteed that $(n_k \in \Pi) \wedge (n_l \notin \Pi) \Rightarrow E_k^d < E_l^d$.

Definition 6. We define a utility function U^2 as follows:

$$U^2 = \begin{cases} U^1 & \text{if } x_s = x_d \vee y_s = y_d, \\ U_j^d = R_j^d - C_j - E_j^d & \text{otherwise.} \end{cases}$$

In the discussion that follows, it is assumed that the cost estimator function E_k^d is admissible.

The estimate returned by $E_k^d(\cdot)$ must be based, at the very least, on some knowledge of the current cost distribution in the network. More precise estimates would require knowledge of the network dynamics. If costs associated with each node are allowed to change with time, as would be the case in a more realistic routing task, since E_k^d is computed at the time a message M is being considered for propagation through n_k, to a destination n_d, E_k^d has to reflect changes in network load over time. We need to represent at each node, the cost distribution over the network in a form that is independent of specific destination nodes (because the destinations become known only after arrival of the respective messages). Any such representation, in order to be useful in practice in large networks, must not require the storage and update at (or broadcast to) each node, of cost values for all the nodes in the large regions of the network. Ideally, it must adequately summarize the load values in large regions of the network as viewed from a given node.

These considerations (among others) led us to define a *view*, V_k, which is maintained in every node in the network (Mikler et al., 1997). In a rectangular grid network, this view consists of four components, one for each of the four directions - north, south, east, and west. Thus, we have: $V_k = [V_k^{\mathrm{N}}, V_k^{\mathrm{S}}, V_k^{\mathrm{E}}, V_k^{\mathrm{W}}]$.

Each component $V_k^{\delta} : (\delta \in \{\mathrm{N}, \mathrm{S}, \mathrm{E}, \mathrm{W}\})$ represents a weighted average of costs C_i along the minimum hop path from n_k to the border of the grid network in the direction specified by δ. Consider two nodes, n_i and n_k, located such that $n_k \in H_i$ and n_k is to the east of n_i, i.e., $x_i < x_k \wedge y_i = y_k$. Then V_i^{E} is given by

$$V_i^{\mathrm{E}} = \frac{C_k + V_k^{\mathrm{E}}}{2}, \tag{5}$$

V_i^{N}, V_i^{S}, and V_i^{W} are computed using analogous formulae.

In the discussion that follows, we assume that sufficient time has elapsed for the view computation to stabilize following major load changes in the network before the view is used in the computation of cost estimates using $E_k^d(\cdot)$.

In practice, this assumption need not be satisfied exactly so long as the views are adequately precise to ensure the admissibility of the cost estimator function defined below. Assuming that n_d is located such that $x_s < x_d \wedge y_s < y_d$. Let $D_i^x = |x_i - x_d|$ and $D_i^y = |y_i - y_d|$ denote the distance from n_i to n_d in x and y direction, respectively. $E_i^d(\cdot)$ is given by

$$E_i^d(\cdot) = \frac{D_i^x V_i^{\mathrm{E}} + D_i^y V_i^{\mathrm{S}}}{D_i^x + D_i^y}. \tag{6}$$

It is easy to verify that this estimator (which is one of several alternatives that are possible) is admissible.

Lemma 4. *For all nodes n_i in the network, for each message M from a source n_s to a destination n_d that reaches a node n_i, the routing decision at n_i based on the preference ordering induced by U^2 will route M along a path P selected only from the set of minimum hop paths from n_i to n_d, unless $P\Delta_{i,d} = 1$ and $(n_h \in P) \wedge (n_h \in H_i)$.*

Proof of Lemma 4. Consider a routing decision to be made for message M by a node n_i. Since $P\Delta_{i,d} > 1$ and $n_h \notin P$, there must exist at least one node $n_k \in H_i$ such that $n_k \neq n_h$ and $R_i^d < R_k^d$ (i.e., n_k is closer to the destination (n_d) than n_i). For Lemma 4 to hold, we have to show that the router at n_i, based on the preference ordering induced by U^2, will necessarily route M to such a node n_k. That is, U^2 must ensure that n_i will not route M through a node $(n_j \in H_i) \wedge (n_j \neq n_h)$ such that $D_{j,d} > D_{i,d} > D_{k,d}$. In other words, in this scenario we have to show that $n_k \succ n_j$ as per the preference ordering induced by U^2.

Note that by Assumption 1 $(R_i^d - R_j^d) > \xi$ and $(R_k^d - R_i^d) > \xi$, and $R_k^d - R_j^d > 2\xi$ (this follows from the fact that n_i and n_j are one hop from each other, n_k and n_i are one hop from each other, and n_j and n_k are two hops from each other). Since $\forall i,\ C_i \leqslant \xi$, Eq. (5) guarantees that $V_i^{\delta} \leqslant \xi$. By Eq. (6), $E_i^d(\cdot) \leqslant \xi$, and thus $(C_i + E_i^d) \leqslant 2\xi$. Thus we have $(R_k^d - C_k - E_k^d) - (R_j^d - C_j - E_j^d) > 0$ which implies $U_k^d > U_j^d$. This implies that n_i routes M through n_k. Since $n_k \in P$ and $n_j \notin P$ (where P is a minimum hop path from n_i to n_d), this proves Lemma 4.

The preceding discussion sets the stage for Theorem 2 that establishes a major property of the utility function U^2, namely, that it eliminates sub-optimal routes in an otherwise unformly loaded grid network with a single hotspot.

Theorem 2. *In a uniform cost network with a single hotspot n_h with an associated cost $C_h > \kappa$ (where $\forall i \neq h$, $C_i = \kappa$), a routing algorithm which makes routing decisions at each node based on a preference ordering induced by U^2 is guaranteed to propagate each message M along a minimum cost path Π.*

Proof of Theorem 2. Consider the placement of n_s and n_d, such that $(x_s \leqslant x_d) \wedge (y_s \leqslant y_d)$ (analogous arguments hold for other source–destination configurations). For nodes n_i, n_j, and n_k for which $(x_i, x_j, x_k < x_h)$, $(y_i, y_j, y_k < y_h)$, $(n_j, n_k \in H_i)$ and $R_i^d < R_j^d = R_k^d$, as per preference ordering induced by U^2 for a message to be propagated from n_i, $n_j \sim n_k$. Hence, a message will be propagated through the network along a minimum cost partial path until a routing decision has to be made which involves a node n_k with coordinates $x_k = x_h \wedge y_k < y_h$ or $x_k < x_h \wedge y_k = y_h$. At this point, the utility of n_k is below that of some n_j with coordinates $x_j < x_h \wedge y_j < y_h$ on account of the relative values of the cost estimates E_j^d and E_k^d. This causes the message M to be propagated to a node n_l with coordinates $x_l = x_h - 1 \wedge y_l = y_h - 1$. We can now show that M will always circumvent n_h and is propagated along Π. We will consider each of the four cases in turn.

Since routing in Case 0 scenarios is equivalent to routing in the absence of hotspots, we have $U^2 = U^1 = U^0$. Hence, a message M will travel along a minimum delay path Π.

As an example for Case 1 scenarios, we have $x_h < x_d \wedge y_h < y_d$. Consider the two possible routing decisions n_j and n_k with coordinates $x_j = x_h - 1 \wedge y_j = y_h$ and $x_k = x_h \wedge y_k = y_h - 1$, respectively. Since both n_j and n_k offer a minimum cost path to n_d, either decision will cause the message M to be routed along an optimal path Π. Since $C_h > C_l = \kappa$ for $n_l \in H_k$ or $n_l \in H_j$ and $x_h < x_d \wedge y_h < y_d$, M will circumvent n_h while approaching n_d. Lemma 4 assures us that U^2 will propagate messages only along a minimum hop path and since given the same number of hops, a path that circumvents a hotspot is necessarily of a lower cost than a path that goes through a hotspot, we can say that for all source–hotspot–destination configurations that correspond to the Case 1 scenario, U^2 guarantees that M is propagated along an optimal path Π.

In a Case 2 scenario, the routing algorithm has to choose at a node n_l, a neighbor from among nodes n_j and n_k with coordinates $x_j = x_h - 1 \wedge y_j = y_h = y_d$ and $x_k = x_h \wedge y_k = y_h - 1$. Clearly, a routing decision that would yield n_j will result in a sub-optimal path P since $x_j = x_h - 1 < x_d \wedge y_j = y_h = y_d$. We can now prove that a routing decision based on the preference ordering induced by U^2 will necessarily select n_k over n_j thereby circumventing n_h. Clearly, $C_j = C_k = \kappa\ \forall i, k \neq h$. Since all nodes to the east of n_h have cost κ, Eq. (5) yields $V_h^{\mathrm{E}} = \kappa$. It follows therefore that the east view computed at n_j is $V_j^{\mathrm{E}} = (C_h + \kappa)/2$. Correspondingly, the south view computed in n_k is $V_k^{\mathrm{S}} = (C_h + \kappa)/2$. As n_h does not impact the south view of n_j or the east view of n_k, we have $V_j^{\mathrm{S}} = V_k^{\mathrm{E}} = \kappa$. Since n_j and n_k have the same distance from n_d, we have $R_j^d = R_k^d$. Therefore the preference ordering between n_j and n_k for routing decisions in n_l is determined by the relative values of $E_j^d(\cdot)$ and $E_k^d(\cdot)$. In other words, n_k is preferred over n_j if $E_j^d(\cdot) - E_k^d(\cdot) > 0$. $E_j^d(\cdot)$ and $E_k^d(\cdot)$ are given by:

$$E_k^d(\cdot) = \frac{D_k^x \kappa + D_k^y((\kappa + C_h)/2)}{D_k^x + D_k^y},$$

$$E_j^d(\cdot) = \frac{D_j^x((\kappa + C_h)/2) + D_j^y \kappa}{D_j^x + D_j^y} = \frac{(D_k^x + 1)((\kappa + C_h)/2) + (D_k^y - 1)\kappa}{D_j^x + D_j^y},$$

$E_j^d(\cdot) - E_k^d(\cdot)$ is then given by

$$\frac{(D_k^x + 1)((\kappa + C_h)/2) + (D_k^y - 1)\kappa}{D_j^x + D_j^y} - \left(\frac{D_k^x \kappa + D_k^y((\kappa + C_h)/2)}{D_k^x + D_k^y}\right).$$

Since $(D_j^x + D_j^y) = (D_k^x + D_k^y)$ it suffices to consider the difference

$$(D_k^x + 1)\left(\frac{\kappa + C_h}{2}\right) + (D_k^y - 1)\kappa - \left(D_k^x \kappa + D_k^y\left(\frac{\kappa + C_h}{2}\right)\right)$$

which simplifies to

$$r = \left(\frac{C_h - \kappa}{2}\right)(D_k^x - D_k^y + 1).$$

Now, $r > 0 \Rightarrow E_j^d(\cdot) > E_k^d(\cdot)$ and n_k should be preferred over n_j. This is the case when $(D_k^x - D_k^y + 1) > 0$.

Since $x_k < x_d$, $D_k^x \geqslant 1$. As $y_h = y_d$ and $y_k = y_h - 1$, we must have $D_k^y = 1$. Therefore, $(D_k^x - D_k^y + 1) > 0$ and a routing decision based on U^2 will route the message M to n_k on its way to the destination n_d.

For Case 3 scenarios, U^2 uses U^1 which will yield an optimal path Π (as shown in Section 4.2.1). This proves Theorem 2.

5. Summary and discussion

Routing and control mechanisms which are based on parameterized heuristics can significantly reduce the resource requirement for storage, acquisition, and use of network state information while achieving the desired performance (as defined by the criteria such as average message delay). Conventional routing and control mechanisms rely on relatively up-to-date information about the state of the entire network. Hence, in large communication networks with thousands of nodes distributed over a wide area, they entail tremendous resource overhead in terms of memory needed at individual nodes, computation time for making decisions, and network bandwidth needed to keep the information up-to-date. The overall effect of this phenomenon include: reduced utilization of the network (in terms of network bandwidth used to actually transmit messages as opposed to information needed for network management), deterioration in the quality of routing and control decisions as measured by some performance metric, or both.

Against this background, we have presented in this paper, a framework for adaptive heuristic routing by a collection of autonomous, reactive, and proactive routing agents. The size of the knowledge base $S_i(t)$ maintained by the routing agent at each node n_i depends solely on the number of neighbors in its neighborhood H_i and is independent of the size of the network. Thus if M is the total number of nodes in the network and h the average connectivity (i.e., the average cardinality of H_i), then the storage required at each node is $O(h)$. This constitutes a significant reduction in storage and processing overhead (especially in very large networks where $M \gg h$) over conventional routing mechanisms (e.g., those that use global routing tables) which require $O(M)$ storage at each node.

As the routing agents at neighboring nodes n_j and n_i communicate only local measurements $\rho_j(t)$ and the view vector $V_j(t)$ the bandwidth requirement is small compared to conventional routing mechanisms. The routing mechanism realized by the collection of interacting routing agents does not attempt to construct a precise picture of the network state as imprecision increases with distance and uncertainty of routing decisions is inevitable. Instead, it utilizes local information, supplemented by a weighted summary of the global network state. Directional orientation is provided through a global coordinate system. Thus, costly validity check of information as required by routing methods that use the link state protocol become unnecessary.

The experimental results summarized in this paper demonstrate that utility-theoretic routing heuristics are largely successful in meeting its primary design objectives, at least when it is used within the relatively simple regular grid network. Particularly noteworthy is the ability to pro-actively as well as reactively avoid congestion in the network while simultaneously minimizing message delay. More systematic parametric study of routing heuristics in a dynamic environment with emphasis on parameters such as, α, β, γ,and update interval τ (and the interrelationships among them as well as β) is a topic of ongoing research.

Extensive research by other researchers on both link state and distance vector routing algorithms have uncovered many issues that need to be considered in the design of new routing mechanisms. Examples of such design issues are bandwidth and storage overhead, performance in the presence of failure (Merlin and Segall, 1979; Jaffe and Moss, 1982; Wong and Kang, 1990), message looping and bouncing. The approach presented in this paper is aimed at reducing resource overhead. Issues such as message looping, message bouncing, as well as mechanisms to deal with node and link failures are currently under study.

A long-term objective of this research is the design of multi-agent systems consisting of autonomous, intelligent agents for self-managing, low-overhead, robust and adaptive traffic management in very large high speed communication networks of the future. Towards this objective, mechanisms that dynamically adapt the tunable parameters in response to changes in network dynamics are of interest. This, however, requires an understanding of the complex interactions that exist between different measures of network performance and resource requirements and the development of a coherent framework that facilitates a graceful tradeoff of some of the performance measures and resource requirements against others on demand. Variations of techniques drawn from adaptive control and machine learning – especially reinforcement learning (Kaelbling et al., 1996; Barto et al., 1995; Bertsekas and Tsitsiklis, 1996) are currently under investigation. For examples of preliminary work by other investigators on this topic, the reader is referred to Littman and Boyan (1993) and Lehmann et al. (1993).

Decision theory and artificial intelligence provide a range of tools that can be useful in the design of intelligent, adaptive, self-managing communication networks. Decision and control tasks that arise in such networks (e.g., routing decisions made at each node, actions taken to balance the load across the entire network, etc.) have to attempt to satisfy as closely as possible, multiple, and often conflicting, performance criteria. Examples of such performance criteria include: network throughput, maximum tolerable delay, maximum tolerable message loss, average delay, degree of load balancing, etc.

In this paper, we have formulated some simple utility-theoretic heuristic decision functions for guiding messages along a near-minimum-delay path in a large network. We have analyzed some of the interesting properties of such heuristics under a set of simplifying assumptions regarding network topology and load dynamics. For a regular grid network with uniform load (with the exception of a single hotspot), we have identified the precise conditions under which a simple and computationally efficient utility-theoretic heuristic decision function is guaranteed to route a message along a minimum delay path when it is assumed that the change in network load is negligible during the time it takes to make a routing decision. We have derived an upper bound on the sub-optimality of a path and have established an upper bound on the probability that a path between a randomly chosen source–destination pair is sub-optimal by considering configurations of uniformly loaded grid networks with single hotspots under the assumption that each source–destination pair is equally likely. We have modified the underlying heuristic function such that it is guaranteed to yield optimal routes under the same set of assumptions about network topology, load, and load dynamics. The study of utility-theoretic heuristics which is described in this paper, was, at least in part, motivated by a desire to formulate the heuristic routing functions and to understand the experimental results in more precise mathematical terms.

Some natural questions to ask at this point include: How realistic or practical are the various assumptions that were made in our development and analysis of utility-theoretic heuristics for routing? How can the results be applied (if at all) to more realistic communication network environments in which assumptions regarding network topology, load, and load dynamics do not hold? How can the analysis be extended to such scenarios? How can computationally efficient utility-theoretic heuristics be designed for different sets of performance criteria for such complex and dynamic networks so that they become essentially autonomous and self-managing? Although this paper does not provide complete and satisfactory answers to all these questions, we believe that it constitutes a useful (albeit perhaps tentative) first step in that direction. In this context, a few comments are in order.

The simulation results of experiments using heuristics that are very similar in spirit to U^2 display the property of automatic *load balancing*. This suggests that the simplifying of *uniform network load* (except at a hot spot) is useful at least as a crude first approximation of a more realistic scenario. A hotspot is typically caused in such a network due to extensive influx of traffic to a particular network node (or group of nodes) or a node or link failure (which is generally assumed to be rare in modern communication networks). However, the behavior of the routing functions compensates for this change by redistributing traffic away from the hotspot. Also, given this behavior, it is reasonable to assume that the probability of several hotspots occurring simultaneously within close proximity of each other in such a network is generally quite small. A possible exception to this scenario would be a hotspot region (caused for example, by a failure of an entire sub-network as could occur in the event of a major natural disaster). When the hotspots are not in close proximity of each other, the single hotspot assumption holds at least locally in a large network. Similarly, the uniform load assumption is also likely to hold (given the load-balancing tendency of the heuristic routing functions), at least locally (except for the discontinuity introduced by a hotspot), in a large network. These observations suggest that our analytical results are likely to be useful (at least in qualitative terms) to guide the design of utility-theoretic heuristics for a more complex network. Of course, this does not mean that it is not worthwhile to extend our analysis to a range of increasingly complex scenarios by removing some of the simplifying assumptions. Some obvious cases to consider include: allowing irregular grids; allowing non-uniform (but relatively smooth) load distribution – except at a hotspot, allowing multiple hotspots or contiguous hotspot regions (of various shapes), etc.

It is perhaps worth emphasizing that the utility function U^2 developed in this paper yields minimum delay paths if certain assumptions regarding network topology, load, and load dynamics hold – by making use of the measured uniform load in the network (and hence the delay per link). Thus, the performance of such utility-theoretic heuristics critically depends on the existence of an adequately precise *estimator* of delay (or some other performance measure) that would result from a particular routing choice. A wide range of such estimators are possible, depending (among other things) on what can be assumed regarding the network topology, load, and network dynamics. It might be useful to analyze a range of such estimators and the resulting heuristics based on different sets of such assumptions – especially since a useful strategy for designing good heuristics for complex problems is based on solution of *simplified* or *relaxed versions* of the original problem (Pearl, 1984). Other interesting research directions include: investigation of methods for adaptation that enable the tuning of heuristics – perhaps parameterized in some manner – using appropriate measurements of network performance as feedback in real-time – drawing upon the rich literature on adaptive control and techniques for learning (Honavar et al., 1998; Mitchell, 1998) for constructing new heuristics or modify existing heuristics as a function of measured network behavior or as a function of information gathered through directed experiments initiated by the network during otherwise idle periods.

The task of making decisions based on incomplete and uncertain information is by no means limited to communication networks. Load distribution and task scheduling in distributed computing environments are other examples of decision mechanisms that are attempting to maximize certain performance criteria without having access to global information upon which their decisions can be based. The tradeoff between the quality of decisions and the resource overhead associated with knowledge acquisition and maintenance is critically important to understand in any complex dynamic environment. Examples of such complex dynamic systems include power systems, transportation systems, distributed computing systems, sensor networks, and manufacturing systems. Development of distributed intelligent information networks consisting of large numbers of intelligent, autonomous, adaptive, communicating, and cooperating agents for monitoring and control of complex dynamic systems and related applications is a topic of ongoing research (Honavar et al., 1998; Silvescu and Honavar, 2000; Sharma et al., 2000).

Acknowledgements

Honavar was supported in part by the National Science Foundation through grant NSF IRI-9409580.

References

Barto, A., Bradtke, S., Singh, S., 1995. Learning to act using real-time dynamic programming. Artificial Intelligence 72, 81–138.

Bertsekas, D., Gallager, R., 1992. Data Networks. Prentice-Hall, Englewood Cliffs, NJ.

Bertsekas, D., Tsitsiklis, J., 1996. Neuro-dynamic Programming. Athena Scientific, New York.

French, S., 1986. Decision Theory: An Introduction to the Mathematics of Rationality. Halsted Press, New York.

Honavar, V., Miller, L., Wong, J., 1998. Distributed knowledge networks. In: Proceedings of the IEEE Information Technology Conference. pp. 87–90.

Kaelbling, L., Littman, M., Moore, A., 1996. Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237–285.

Jaffe, J.M., Moss, F.H., 1982. A responsive distributed routing algorithm for computer networks. IEEE Transactions on Communications Com-30 (7), 1758–1762.

Lehmann, F., Seising, R., Walther-Klaus, E., 1993. Simulation of learning in communication networks. Simulation Practice and Theory 1 (1), 41–48.

Littman, M., Boyan, J., 1993. A distributed reinforcement learning scheme for network routing. In: Alspector, J., Goodman, R., Brown, T.X. (Eds.), Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications. pp. 45–51.

Merlin, P.M., Segall, A., 1979. A failsafe distributed routing protocol. IEEE Transactions on Communications Com-27 (9), 1280–1287.

Mikler, A.R., Honavar, V.G., Wong, J.S.K., 1996. Analysis of utility-theoretic heuristics for intelligent adaptive network routing. In: Proceedings of the 13th National Conference on Artificial Intelligence. Portland, OR.

Mikler, A.R., Wong, J.S.K., Honavar, V.G., 1997. Quo vadis – a framework for intelligent routing in large communication networks. The Journal of Systems and Software 37 (1), 61–73.

Mikler, A.R., Wong, J.S.K., Honavar, V.G., 1998. An object-oriented approach to simulating large communication networks. The Journal of Systems and Software 40 (2), 151–164.

Mitchell, T., 1998. Machine Learning. McGraw-Hill, New York.

Pearl, J., 1984. Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, Reading, MA.

Silvescu, A., Honavar, V., 2000. MIGories: an abstract model for inter-agent interaction. In: Proceedings of the Fourth International Conference on Coordination Models and Languages. Limassol, Cyprus (in press).

Sharma, T., Silvescu, A., Honavar, V., 2000. Algorithms for learning from distributed data sets. In: Proceedings of the Workshop on Distributed and Parallel Knowledge Discovery, Conference on Knowledge Discovery and Data Mining (KDD). Boston, MA (in press).

Wong, J.S.K., Kang, Y., 1990. Distributed and fail-safe routing algorithms in toroidal-based metropolitan area networks. Computer Networks and ISDN Systems 18, 379–391.

Armin R. Mikler received his Diploma in Informatik from Fachhochschule Darmstadt, Germany in 1988. He received his M.S. and his Ph.D.(Computer Science) from Iowa State University in 1990 and 1995, respectively. From 1995 to 1997, he worked as a postdoctoral research associate in the Scalable Computing Laboratory at the Ames Laboratory, USDOE. Since 1997, Dr. Mikler has been an assistant professor in the Department of Computer Sciences at the University of North Texas (UNT). His research interests include: Intelligent Traffic Management, Coordination of Intelligent Mobile Agents, Decision-Making in Distributed Environments, and Cluster Computing. At UNT, Dr. Mikler founded and currently directs the Network Research Laboratory (NRL) for research on Intelligent Mobile Agents, Autonomous Distributed Systems, Mobile and Wireless Network Management, and Distributed Simulation (see http://www.nrl.csci.unt.edu).Dr. Mikler is a member of UNT's Institute of Applied Science and has been conducting collaborative research in computational science. Prof. Mikler is an associate editor for Telecommunication Systems and a member of ACM and the IEEE Computer Society.

Dr. Vasant Honavar received a B.E. in Electronics Engg. from Bangalore University, India, an M.S. in Electrical and Computer Engg. From Drexel University, and an M.S. and a Ph.D. in Computer Science from the University of Wisconsin, Madison. He founded and directs the Artifical Intelligence Research Laboratory www.cs.iastate.edu/~honavar/aigroup.html in the Department of Computer Science at Iowa State University (ISU) where he is currently an associate professor. Honavar is also a member of the Lawrence E. Baker Center for Bioinformatics and Biological Statistics, the Virtual Reality Application Center, and the faculty of Bioinformatics and Computational Biology at ISU. His research and teaching interests include Artifical Intelligence, Machine Learning, Bioinformatics and Computational Biology, Intelligent Agents and Multi-agent systems, Distributed Intelligent Information Networks, Data Mining, Knowledge Discovery and Visualization. He has published over 100 research articles in refereed journals, conferences and books, and has co-edited three books. He is a co- editor-in-chief of the Journal of Cognitive Systems Research published by Elsevier. His research has been partially funded by grants from the National Science Foundation, the National Security Agency, the Defense Advanced Research Projects Agency (DARPA), the US Department of Energy, the John Deere Foundation, the Carver Foundation, Pioneer Hi-Bred Inc., and IBM. Prof. Honavar is a member of ACM, AAAI, IEEE, and the New York Academy of Sciences.

Dr. Johnny Wong is a Full Professor of the Computer Science Department, Iowa State University at Ames, Iowa, USA. Before he came to Iowa State University, he was on the faculty of the Computer Science Department at the University of Sydney. His research interests include Operating Systems, Distributed Systems, Telecommunication Networks, Integrated Services Digital Networks (ISDN), Concurrency Control and Recovery, Object-Oriented Databases, Multilevel Database and Network Security, Multi-Agent Systems for Intrusion and Countermeasures.He has been an investigator for research contracts with Telecom Australia from 1983 to 1986, studying the performance of network protocols of the ISDN. During this period, he has contributed to the study and evaluation of the communication architecture and protocols of ISDN. From 1989 to 1990, he was the Principal Investigator for a research contract with Microware Systems Corporation at Des Moines, Iowa. This involved the study of Coordinated Multimedia Communication in ISDN. In Summer 1991 and 1992, Dr. Wong was supported by IBM corporation in Rochester. While at IBM, he worked on the Distributed Computing Environment (DCE) for the Application Systems. This involved the integration of communication protocols and distributed database concepts. Dr. Wong is also involved in Coordinated Multimedia System (COMS) in Courseware Matrix Software Project, funded by NSF Synthesis Coalition Project to enhance engineering education. From 1993 to 1996, he was working on a research project on a knowledge-based system for energy conservation education using multimedia project communication technology, funded by the Iowa Energy Center. From 1995 to 1996, he was supported by the Ames Laboratory of the DOE, working in Middleware for Multidatabase systems.Currently, he is working on several projects, including Intelligent Multi-Agents for Intrusion Detection and Countermeasures funded by the Department of Defense (DoD), Database-generating and X-ray Displaying World Wide Web Applications funded by Mayo Foundation, and CISE Educational Innovation: Integrated Security Curricular Modules funded by National Science Foundation (NSF).