Distributed Scheduling of Multiagent Communication

Y. Xiang
Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2, E-mail: yxiang@cs.uregina.ca

Abstract

We consider a homogeneous cooperative multiagent system organized as a multiply sectioned Bayesian network (MSBN). Earlier work has shown that (1) multiagent MSBNs can be applied to distributed interpretation tasks; and (2) a distributed communication operation can be used to ensure the global consistency among agents.

In this paper, we address the following problem: During a communication operation, each agent is unavailable to process new evidence for a time interval (called off-line time). We consider the minimization of the total length of off-line time of the entire system. To concentrate on the factors affecting the off-line time, we abstract communication in MSBNs into a graphical model for off-line time study. Using the model, we present the optimal schedules when communication is initiated from an arbitrarily selected agent. We show how the optimal schedules can be constructed in a distributed fashion.

Topic areas: Distributed artificial intelligence, communication issues.

1 Introduction

Multiply sectioned Bayesian network (MSBN) was developed originally for probabilistic reasoning in a single-agent oriented knowledge-based system in a large domain (Xiang, Poole, & Beddoes 1993; Xiang *et al.* 1993). Earlier work (Xiang 1994a) showed that the modular structure of MSBN allows a natural extension of its semantics to multiagent distributed interpretation tasks as defined in (Lesser & Erman 1980), and a communication operation was proposed to regain the global consistency after multiagents have acquired local evidence asynchronously in parallel. Infrequent communication was shown to be necessary and adequate. Thus the framework can be characterized as one of functionally accurate, cooperative distributed systems (Lesser & Corkill 1981).

The basic syntactic requirement for MSBNs to be applicable to a domain is that the information dependency of agents can be organized into a hypertree structure such that agents A and B mediated in the hypertree by a third agent C are conditionally independent given the information contained in C. Whenever such a hypertree organization is feasible for a distributed interpretation domain, the MSBN framework can be applied. Xiang et al. (1993) showed how such an organization can be achieved in a medical domain (representing a natural system). Srinivas (1994) proposed a hierarchical approach for model-based diagnosis, which can be viewed as a special case of MSBNs[1]. His work showed that how probabilistic knowledge about electronic circuits (an artificial system) can be organized into a hypertree structure.

Given the applicability of MSBNs to multiagent distributed interpretation and the communication operation that regains the global consistency, there remain the issues of improving the efficiency of communication operation. This paper addresses the following issue: During communication, each agent is not available to process new evidence for a period of time (called *off-line time*). Proper scheduling of communication should minimize the length of this off-line time. How can we perform the scheduling in a distributed fashion?

Since different agents have different off-line time, some measurement for the entire system is necessary. This paper consider the minimization of the total length of off-line time for the entire multiagent system. To facilitate the study of the optimal communication schedules, we abstract the activities during communication into a graphical model. We then present the communication schedules that minimize the total length of off-line time when communication is initiated

[1]For example, the set of input node I, output node O, mode node M, and dummy node D (Srinivas 1994), which collectively form an interface between a higher level and a lower level in the hierarchy, is a d-sepset (Xiang, Poole, & Beddoes 1993). The 'composite joint tree' (Srinivas 1994) corresponds to the 'hypertree' (Xiang, Poole, & Beddoes 1993). The way in which inference is performed in the composite join tree corresponds to the operation *ShiftAttention* (Xiang, Poole, & Beddoes 1993).

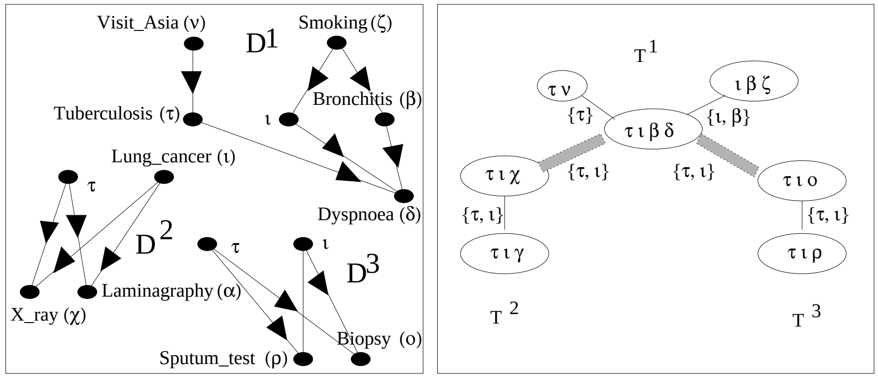

Figure 1: Left: A three-agent MSBN in a medical domain. D^1: clinical subnet, D^2: radiological subnet, D^3: biological subnet. Right: The LJF for the multi-agent MSBN. Sepsets between cliques of each JT are shown in solid lines. Hyperlink between JTs are shown in dotted bands.

from an arbitrarily selected agent. Finally, we show how these schedules can be obtained in a distributed fashion.

Section 2 reviews the communication operation in MSBNs. Section 3 introduces the off-line time problem. Section 4 proposes the graphical model used to focus the study of the problem. Section 5 presents the algorithms for construction of the optimal communication schedules. Section 6 shows how to obtain the optimal schedules by distributed operations of multiagents.

2 Communication in Multiagent MSBNs

Readers are referred to (Pearl 1988; Charniak 1991; Henrion, Breese, & Horvitz 1991; Neapolitan 1990; Shachter 1988; Lauritzen & Spiegelhalter 1988; Jensen, Lauritzen, & Olesen 1990) for the syntax, semantics and common inference algorithms of Bayesian networks, and to (Xiang, Poole, & Beddoes 1993; Xiang *et al.* 1993; Xiang 1994a) for formal presentation of the syntax, semantics and inference algorithms in MSBNs. To make this paper self-contained, we briefly review the communication operation in multiagent MSBNs.

A multiagent MSBN consists of a set of interrelated Bayesian subnets, each of which represents one agent's perspective of the entire domain. A set of variables interfacing a pair of subnets are chosen such that the two subnets are conditionally independent given the set (called a *d-sepset*). The MSBN is compiled into a *linked junction forest* (LJF) for run time inference computation. The forest consists of a set of junction trees (JTs) each of which is compiled from an original subnet. The JTs are linked into a *hypertree* structure. Each hypernode is a JT, and each hyperlink corresponds to a d-sepset. The LJF is associated with a probability distribution, equivalent to that of the original MSBN, defined in terms of belief tables (BTs) of individual JTs. The hypertree is so organized that, if A, B and C are three nodes in the hypertree which form a chain (with B in the middle), then A and C are conditionally independent given B.

Example 1 Figure 1 (left) shows a three-agent MSBN representing diagnostic knowledge of tuberculosis and lung cancer from three perspectives: clinical, radiological, and biological. The clinical agent may need the help from the other two agents in reaching a diagnosis since itself does not have the expertise to process radiological and biological evidence. The compiled LJF are shown in Figure 1 (right).

Example 2 Figure 2 (left) depicts a general hypertree structured MSBN. Each box represents a subnet. The boundaries between boxes represent the d-sepsets. Figure 2 (right) illustrates the compiled LJF. Ignore the arrows for the moment.

In a multiagent MSBN, each agent/subnet acquires local evidence asynchronously in parallel, which causes inconsistency among agents. A distributed operation `CommunicateBelief` (Xiang 1994a) is performed from time to time to regain the global consistency. Readers are referred to the above reference for the definition of the operation. A formal treatment about its correctness (i.e., it guarantees the global consistency) can be found in (Xiang 1994b). We illustrate here informally how the operation works.

Example 3 Figure 2 (right) shows how belief propagates through a LJF during `CommunicateBelief`. Suppose the operation is initiated at an arbitrarily selected agent represented by T^1. `CommunicateBelief` consists of two steps: The first step `CollectNewBelief`

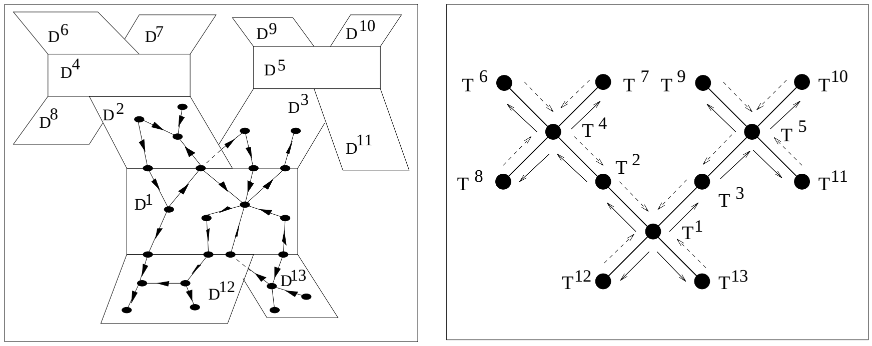

Figure 2: Left: A MSBN with a hypertree structure. Right: The compiled LJF, and directions of belief propagation during `CommunicateBelief`.

proceeds by first propagating control from T^1 towards terminal agents along solid arrows, and then propagating belief from terminal agents back to T^1 along dotted arrows. The second step `DistributeBelief` proceeds by propagating belief from T^1 towards terminal agents along solid arrows. Compared with the time for belief propagation, the time required for the control propagation can usually be ignored. The operation of belief propagation from one JT to a neighbor JT is called `UpdateBelief`.

3 Off-line Time During Communication

3.1 Off-line time of each agent

Given an arbitrary agent chosen to initiate the communication activity (call it the communication *root*), the hypertree LJF can be viewed as a rooted tree. In Figure 2 (right), for example, T^1 is the root, T^2 has a child T^4, and T^2's parent is T^1.

During the operation `CommunicateBelief`, the BT of a JT T (denoted $B(T)$) may be changed through `CollectNewBelief` and `DistributeBelief` in the following ways: (1) During `CollectNewBelief`, `UpdateBelief` performed by T relative to its children may change $B(T)$; (2) During `DistributeBelief`, `UpdateBelief` performed by T relative to its parent may change $B(T)$.

In order to guarantee that `CommunicateBelief` regain the global consistency, it is necessary (see (Xiang 1994b) for proof) that $B(T)$ be not modified by new evidence between the first `UpdateBelief` (during `CollectNewBelief`) and the last `UpdateBelief` (during `DistributeBelief`) in the process of `CommunicateBelief`. This implies that T can not process new evidence and answer queries accordingly between the above mentioned two `UpdateBelief`s. Therefore the length of this time interval should be minimized.

If t is the instant of time when the first `UpdateBelief` involved by T is started, and τ is the instant of time when the last `UpdateBelief` involved by T is completed. We define $\Delta(T) = \tau - t$ as the **`off-line time`** of T during the communication.

3.2 Off-line time of a multi-agent system

Different JTs in a LJF may have different off-line times during a communication depending on several factors. Assuming the communication root is given, this paper considers the following three factors. One factor is the order in which `CollectNewBelief` is performed by each agent relative to its neighbors.

Example 4 Consider T^7 in Figure 2 (right) where the root is T^1. During `CollectNewBelief`, T^4 must perform `UpdateBelief` relative to T^6, T^7 and T^8 sequentially. If T^7 is first selected, T^7 must become off-line before T^6 and T^8, and its off-line time will be prolonged accordingly.

We refer to the order in which multiple neighbors are selected by an agent to perform `UpdateBelief` against, during `CollectNewBelief`, as the *collection order* of the agent. Similarly, we refer to the order in which multiple neighbors are selected by an agent to perform `UpdateBelief`, during `DistributeBelief`, as the *distribution order* of the agent. It is another factor affecting each agent's off-line time.

Example 5 Consider T^7 in Figure 2 (right) where the root is T^1. During `DistributeBelief`, T^6, T^7 and T^8 must perform `UpdateBelief` relative to T^4 *sequentially.* If T^7 is first selected, T^7 can become available before T^6 and T^8, and its off-line time will be shortened accordingly.

The third factor is the time complexity of `UpdateBelief` by a JT T^i relative to a neighbor JT

T^k. This time complexity is fixed once the two JTs and their d-sepset are determined. However the time complexity of `UpdateBelief` by T^i relative to T^k may not be the same as the time complexity of `UpdateBelief` by T^k relative to T^i since both the size of the d-sepset and the size of the belief-receiving JT are relevant (Xiang 1994b).

The difference of off-line time across agents calls for a measurement of off-line time of the entire system. This paper considers the following measurement.

Definition 6 (Absolute Off-line Time) *Let F be a LJF. Let t be the instant of time when a first JT in F becomes off-line during a* `CommunicateBelief` *operation. Let τ be the instant of time when the last JT becomes available again. The* `absolute off-line time` *of F is $\Delta_{abs} = \tau - t$.*

4 A Graphical Model for Off-line Time Study

To concentrate on the factors that determine the off-line time, we abstract the communication in a LJF into a graphical model:

Model 7 (Graphical communication model)

Given an undirected and weighted tree, and an arbitrary node A as the root, the tree is converted to a rooted tree R.

For each node X of R, if $X \neq A$, place an *in-agent* at X. For each node Y of R, if Y has k children, place k *out-agents* at Y.

The agents traverse R according to the following rules.

1. To start with, each parent node Y with leaf children selects one child X, according to some order $O_{in}(Y)$. Once selected, X sends its in-agent to move from X to Y, which takes time $w_{in}(X)$ that is the weight associated with the link (X, Y) in the inward direction (leaf towards root). After one child's in-agent arrives at Y, the next child, selected according to $O_{in}(Y)$, sends its in-agent to Y.

   After a parent Y has received all the in-agents from its children, Y is ready for selection by its own parent Z according to $O_{in}(Z)$. Once selected by Z, Y sends its in-agent to Z. The inward movement (called *collection* of in-agents continues in this fashion.

2. After the root A receives all in-agents from its children, collection is completed, and an outward movement (called *distribution*) starts.

   A selects one child X, according to some order $O_{out}(A)$. A then sends one out-agent to move from A to X, which takes time $w_{out}(X)$ that is the weight of the link (A, X) in the outward direction. After an out-agent of A reaches the destination, A selects another child according to $O_{out}(A)$ and sends another out-agent to the child. The process continues until all out-agents of A reach their destinations.

   After an out-agent from A reaches a child X, X selects its own children, according to $O_{out}(X)$, and sends its out-agents to child nodes in sequence. The process continues in this fashion until the last out-agent in R reaches its leaf destination.

The model characterizes `CommunicateBelief` for off-line time study correctly: The undirected tree corresponds to the LJF. Each node corresponds to a JT of the LJF. The root A corresponds to the communication root. Collection corresponds to `CollectNewBelief`, and distribution corresponds to `DistributeBelief`. Given a parent node Y and a child node X, $w_{in}(X)$ corresponds to the time required for Y to perform `UpdateBelief` relative to X, and $w_{out}(X)$ for X relative to Y. $O_{in}(X)$ corresponds to the collection order of X, and $O_{out}(Y)$ corresponds to the distribution order of Y. The time instant when the in-agent of a node X leaves X corresponds to the time instant when the corresponding JT becomes off-line. The time instant when the last out-agent of X arrives at its destination (a child of X) corresponds to the time instant when the corresponding JT becomes available (on-line) for entering evidence. The interval between the two instants thus corresponds to the off-line time of the JT represented by X. Do not confuse the in(out)-agents with multiagents. The former corresponds to the belief to be propagated, and the latter corresponds to nodes of the graphical model.

We shall say that a non-leaf node X is *off* at the time instant when the in-agent from the first child selected by X starts moving to X. We use $t_{off}(X)$ to denote the instant. If X is a leaf node, then X is off as soon as its in-agent leaves X.

We shall say that a non-leaf node X is *on* at the time instant when its last out-agent arrives at one of X's children. We use $t_{on}(X)$ to denote the instant. If X is a leaf, then X is on when it receives the out-agent from its parent. We shall say that the off-line time of the node X is $\Delta(X) = t_{on}(X) - t_{off}(X)$.

For collection, we use $t_{rdy}(Y)$ to denote the time instant when a non-leaf node Y receives the last in-agent from its children and is *ready* for its parent to select. For a leaf node Y, we assign $t_{rdy}(Y)$ to be the instant when collection starts. We use $t_{wat}(Y)$ to denote the time instant when Y's in-agent arrives at its parent and Y starts to *wait* for an out-agent to

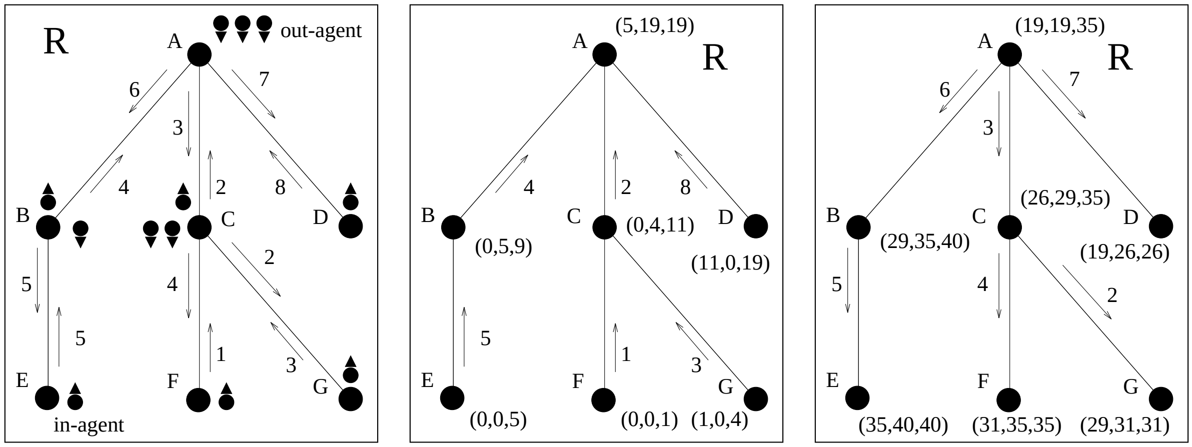

Figure 3: Graphical model for communication in a seven-agent MSBN. The in-weight (out-weight) of a link is indicated by an upward (downward) arrow and the associated label. Left: The weighted tree R rooted at A. Middle: Collection schedule in R. Each node X is labeled with $(t_{off}(X), t_{rdy}(X), t_{wat}(X))$. Right: Distribution schedule in R. Each node X is labeled with $(t_{sel}(X), t_{cpt}(X), t_{on}(X))$.

be sent from its parent. For the root A, we assign $t_{wat}(A) = t_{rdy}(A)$.

For distribution, we use $t_{sel}(X)$ to denote the time instant when a node X is *selected* by its parent Y such that Y is about to send an out-agent to X. For the root A, we assign $t_{sel}(A)$ to be the instant when distribution starts. We use $t_{cpt}(X)$ to denote the time instant when X receives the out-agent from Y (distribution relative to Y is *completed*). For the root A, we assign $t_{cpt}(A) = t_{sel}(A)$.

We shall call a complete specification of the above defined timing of every node during collection (distribution) as a collection (distribution) *schedule*. Figure 3 illustrates the graphical communication model for a seven-agent system (left), a collection schedule (middle) using a left-to-right order for each node, and a distribution schedule (right) using a right-to-left order. The absolute off-line time is $\Delta_{abs} = 40 - 0 = 40$.

Our goal is to find schedules with the minimum off-line time. In both schedules of Figure 3, we have assumed that a node engages in its activity as soon as the activity is possible without any delay. Since unnecessary idling can not contribute positively to our goal, we will exclude from our consideration those schedules in which some nodes delay their activities unnecessarily (*no-delay* assumption). On the other hand, if there is any practical reason to delay the belief propagation, e.g., computer network delay, we assume that the delay has been modeled in the link weights.

The schedule in Figure 3 is *not* optimal. In the remaining part of this paper, we use the graphical model to study the minimization of off-line time with an arbitrarily given communication root.

5 Minimal Absolute Off-line Time Schedule

Let collection start at $t = t_0$ and terminate at $t = t_1$. Let distribution start at $t = t_1$ and terminate at $t = t_2$. Denote the interval between t_0 and t_1 by Δ_{0-1}, and denote the interval between t_1 and t_2 by Δ_{1-2}. We have $\Delta_{abs} = \Delta_{0-1} + \Delta_{1-2}$. Since Δ_{0-1} is independent of Δ_{1-2}, $min(\Delta_{abs}) = min(\Delta_{0-1}) + min(\Delta_{1-2})$ where the minimization in the left-hand side of the equation is over all collection and distribution schedules, the first minimization in the right-hand side is over all collection schedules, and the second is over all distribution schedules. This implies that the optimal communication schedule can be obtained by independently obtaining the optimal collection schedule and the optimal distribution schedule.

To determine the optimal collection schedule, it is sufficient to determine the collection order for each node in the rooted tree, given our no-delay assumption. We therefore present our result in terms of Algorithm 9 that rearranges the left-right order for each node such that the collection order becomes topologically explicit. In the algorithm, the depth of root is 0. Theorem 8 establishes the optimality, whose proof can be found in (Xiang 1994b).

Theorem 8 (Optimal collection schedule)
Let R be a tree for collection rooted at A.

The absolute off-line time Δ_{0-1} for collection is minimized if R is arranged according to Algorithm 9 and then collection at each node is performed according to the left-to-right order.

The minimum value of Δ_{0-1} is given by $t_{rdy}(A)$ as computed by Algorithm 9.

Algorithm 9 (Order arrangement for collection)

Input: *A rooted tree of depth M with the in-weight of each link defined.*
begin
  D := M;
  for each node Z of depth D, do $t_{rdy}(Z) := 0$;
  D := D-1;
  while $D \geq 0$, do
    for each node Y of depth D with n child nodes, do
      arrange children of Y and index them from left to right as
        $X_1, \ldots, X_n$ such that $t_{rdy}(X_1) \leq \ldots \leq t_{rdy}(X_n)$;
      $t_{rdy}(Y) := max(t_{rdy}(X_1) + \sum_{i=1}^{n} w_{in}(X_i), \ldots, t_{rdy}(X_n) + \sum_{i=n}^{n} w_{in}(X_i))$;
    for each leaf Z of depth D, do $t_{rdy}(Z) := 0$;
    D := D-1;
end

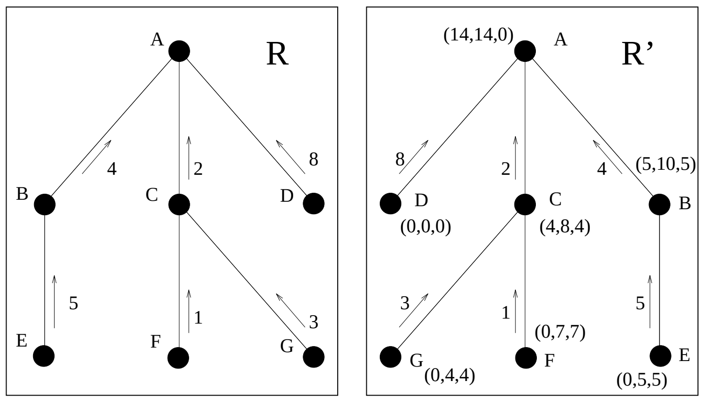

Figure 4: R: A rooted tree for collection. R': R after processed according to Algorithm 9. Each node X is labeled with $(t_{off}(X), t_{rdy}(X), t_{wat}(X))$.

Example 10 The depth of the tree in Figure 4 is $M = 2$. The t_{rdy} for each node as computed by Algorithm 9 as well as the optimal collection schedule determined by Theorem 8 are shown in the figure. After the first iteration of the *while* loop, $t_{rdy}(B) = 5$, $t_{rdy}(C) = 4$, and $t_{rdy}(D) = 0$. After the second iteration of the *while* loop, the children of A is arranged in the order D, C and B from left to right, and $t_{rdy}(A) = max(0+8+2+4, 4+2+4, 5+4) = 14$. The minimum absolute off-line time $min(\Delta_{0-1})$ for R is obtained using R' and the left-to-right collection order: $min(\Delta_{0-1}) = t_{rdy}(A) = 14$. It is a 26% improvement over $t_{rdy}(A) = 19$ in Figure 3 (middle).

Note that the same minimum Δ_{0-1} as from Example 10 can be obtained if the collection order for root is $O_{in}(A) = (D, B, C)$ instead of $O_{in}(A) = (D, C, B)$ as in the example. Given a rooted tree, the optimal collection schedule is not unique in general.

Algorithm 13 and Theorem 11 establishes the optimal distribution schedule. The proof of Theorem 11 can be found in (Xiang 1994b).

Theorem 11 (Optimal distribution schedule)

Let R be a tree for distribution rooted at A. The absolute off-line time Δ_{1-2} for distribution is minimized if R is arranged according to Algorithm 13 and the distribution order for each node is right-to-left. The minimum Δ_{1-2} is given by $v(A)$ as computed by Algorithm 13.

Example 12 Figure 5 (left) shows the same rooted tree R for distribution as Figure 3. It is rearranged into R' (right) according to Algorithm 13. The optimal schedule as determined by Theorem 11 is also labeled in the figure. The distribution starts at $t = 0$. The minimum Δ_{1-2} is $\Delta_{1-2} = v(A) = max(0+7+6+3, 5+6+3, (4+2)+3) = 16$ which is a 24% improvement over $\Delta_{1-2} = 40 - 19 = 21$ in Figure 3 (right).

Algorithm 13 (Order arrangement for distribution)

Input: *A rooted tree of depth M with the out-weight of each link defined.*
begin
D := M;
for each node Z of depth D, do $v(Z) := 0$;
D := D-1;
while $D \geq 0$, do
for each node Y of depth D with n child nodes, do
arrange children of Y and index them from left to right as
$X_1, \ldots, X_n$ such that $v(X_1) \leq \ldots \leq v(X_n)$;
$v(Y) := max(e_1, \ldots, e_n)$ where $e_i = (\sum_{k=i}^{n} w_{out}(X_k)) + v(X_i)$;
for each leaf Z of depth D, do $v(Z) := 0$;
D := D-1;
end

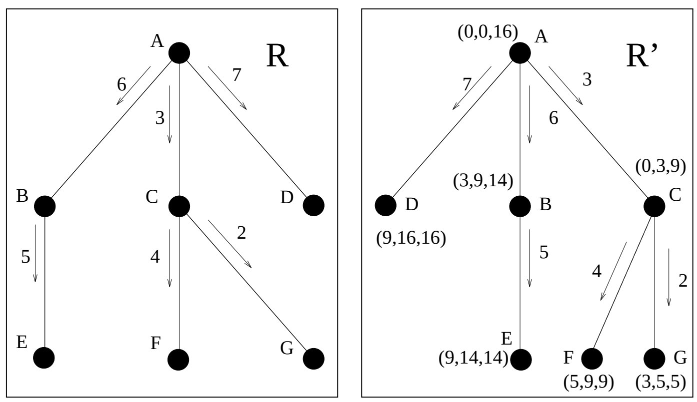

Figure 5: R: A rooted tree for distribution. R': R with the left-right order of nodes rearranged according to Algorithm 12. The distribution schedule determined by Theorem 11 is shown by the label $(t_{sel}(X), t_{cpt}(X), t_{on}(X))$ at each node X.

6 Distributing Communication Scheduling

The optimal schedules in Section 5 are presented as if there is a centralized scheduler. It is not necessary. The scheduling of collection can be distributed as follows:

Operation 14 (ScheduleCollection) *Let T be a JT in a LJF. Let* `caller` *be either the LJF or a neighbor JT. When* `ScheduleCollection` *is called in T, T performs the following.*

1. *If T has no neighbor except* `caller`*, T returns $t_{rdy}(T) = 0$ to* `caller`*. Otherwise, T performs the following.*

2. *T calls* `ScheduleCollection` *in all neighbors except* `caller`*.*

3. *After each neighbor X being called has returned $t_{rdy}(X)$, T indexes them as $X_1, \ldots, X_n$ such that $t_{rdy}(X_1) \leq \ldots \leq t_{rdy}(X_n)$. T returns $t_{rdy}(Y) := max(t_{rdy}(X_1) + \sum_{i=1}^{n} w_{in}(X_i), \ldots, t_{rdy}(X_n) + \sum_{i=n}^{n} w_{in}(X_i))$ to* `caller`*. The collection order of T is $O_{in}(T) = (X_1, \ldots, X_n)$.*

`ScheduleCollection` *is equipped at each JT.*

Similarly, we can distribute the scheduling of distribution.

Operation 15 (ScheduleDistribution) *Let T be a JT in a LJF. Let* `caller` *be either the LJF or a neighbor JT. When* `ScheduleDistribution` *is called in T, T performs the following.*

1. *If T has no neighbor except* `caller`*, T returns $v(T) = 0$ to* `caller`*. Otherwise, T performs the following.*

2. *T calls* `ScheduleDistribution` *in all neighbors except* `caller`*.*

3. *After each neighbor X being called has returned $v(X)$, T indexes them as $X_1, \ldots, X_n$ such that $v(X_1) \leq \ldots \leq v(X_n)$. T returns $v(Y) := max(v(X_1) + \sum_{i=1}^{n} w_{out}(X_i), \ldots, v(X_n) + \sum_{i=n}^{n} w_{out}(X_i))$ to* `caller`*. The distribution order of T is $O_{out}(T) = (X_n, \ldots, X_1)$.*

`ScheduleDistribution` *is equipped at each JT.*

The optimal communication schedule of the entire system can then be obtained as follows.

Operation 16 (ScheduleCommunication) *When* `ScheduleCommunication` *is initiated at a LJF, the following are performed.*

1. *A JT T is selected.*

2. `ScheduleCollection` *is called in T.*

3. *When T has finished* `ScheduleCollection`*,* `ScheduleDistribution` *is called in T.*

Theorem 17 *Let F be a LJF. If the collection order and distribution order obtained from performing* `ScheduleCommunication` *are followed during* `CommunicateBelief`*, the resultant schedule has the minimum absolute off-line time Δ_{abs}.*

Theorem 17 can be proven by comparing `ScheduleCollection` with Algorithm 9, comparing `ScheduleDistribution` with Algorithm 13, and then applying Theorem 8 and Theorem 11.

Acknowledgement

This work is supported by the Dean's Research Funding from Faculty of Science, University of Regina, the General NSERC Grant from University of Regina, and Research Grant OGP0155425 from NSERC. Anonymous reviewers provided useful feedback.

References

Charniak, E. 1991. Bayesian networks without tears. *AI Magazine* 12(4):50–63.

Henrion, M.; Breese, J.; and Horvitz, E. 1991. Decision analysis and expert systems. *AI Magazine* 12(4):64–91.

Jensen, F.; Lauritzen, S.; and Olesen, K. 1990. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly* (4):269–282.

Lauritzen, S., and Spiegelhalter, D. 1988. Local computation with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B* (50):157–244.

Lesser, V., and Corkill, D. 1981. Functionally accurate, cooperative distributed systems. *IEEE Transactions on Systems, Man and Cybernetics* SMC-11(1):81–96.

Lesser, V., and Erman, L. 1980. Distributed interpretation: a model and experiment. *IEEE Transactions on Computers* C-29(12):1144–1163.

Neapolitan, R. 1990. *Probabilistic Reasoning in Expert Systems*. John Wiley and Sons.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Shachter, R. 1988. Probabilistic inference and influence diagrams. *Operations Research* 36(4):589–604.

Srinivas, S. 1994. A probabilistic approach to hierarchical model-based diagnosis. In *Proc. Tenth Conf. Uncertainty in Artificial Intelligence*, 538–545.

Xiang, Y.; Pant, B.; Eisen, A.; Beddoes, M. P.; and Poole, D. 1993. Multiply sectioned bayesian networks for neuromuscular diagnosis. *Artificial Intelligence in Medicine* 5:293–314.

Xiang, Y.; Poole, D.; and Beddoes, M. P. 1993. Multiply sectioned bayesian networks and junction forests for large knowledge based systems. *Computational Intelligence* 9(2):171–220.

Xiang, Y. 1994a. Distributed multi-agent probabilistic reasoning with bayesian networks. In Ras, Z., and Zemankova, M., eds., *Methodologies for Intelligent Systems*. Springer-Verlag. 285–294.

Xiang, Y. 1994b. A probabilistic framework for multi-agent distributed interpretation and optimization of communication. Technical Report TR 94-08, University of Regina.