Memory Allocation in Information Storage Networks[1]

Anxiao (Andrew) Jiang and Jehoshua Bruck
Dept. of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, U.S.A.
{jax, bruck}@paradise.caltech.edu

I. Introduction

Given a graph $G = (V, E)$ whose edges have lengths, we denote the distance between two vertices u and v by $d(u, v)$. For any vertex $u \in V$ and any real number r, we define $N(u, r)$ as the set of vertices within distance r from u. We study the following problem in this paper:

Definition 1: The Memory Allocation Problem

INSTANCE: A graph $G = (V, E)$. Every edge $e \in E$ has a length $l(e)$. Every vertex $v \in V$ is associated with a set $R(v) = \{(r_i(v), k_i(v)) | 1 \leq i \leq n_v\}$, which is called the '*requirement set*' of v. Each vertex $v \in V$ is also associated with a parameter $W_{min}(v)$, which is called the '*minimum memory size*' of v, and a parameter $W_{max}(v)$, which is called the '*maximum memory size*' of v.

QUESTION: How to assign a number $w(v)$ ($W_{min}(v) \leq w(v) \leq W_{max}(v)$) to each vertex $v \in V$, with the value of $\sum_{v \in V} w(v)$ minimized, such that for any vertex $u \in V$ and for $1 \leq i \leq n_u$, $\sum_{v \in N(u, r_i(u))} w(v) \geq k_i(u)$? (Here $w(v)$ is called the '*memory size* of v'. A solution to this memory allocation problem is called an *optimal memory allocation*.)

COMMENTS: In the definition of this problem, both $l(e)$ and $r_i(v)$ are non-negative real numbers, and $k_i(v)$, $W_{min}(v)$, $W_{max}(v)$, and $w(v)$ are all non-negative integers. □

The memory allocation problem is a sub-problem of a file storage scheme we propose, which bounds file-retrieving delays in a heterogeneous information network, under both fault-free and faulty circumstances. (For details, see [1].)

II. Main Results

In the rest of this paper, we assume $G = (V, E)$ is a rooted tree, and assume for any $v \in V$ and for $1 \leq i \leq n_v$, $\sum_{u \in N(v, r_i(v))} W_{max}(u) \geq k_i(v)$ — so a feasible memory allocation solution exists. For any vertex $v \in V$, let $Des(v)$ denote the set of descendants of v. A set $\{w(v) | v \in V\}$ is called an *optimal memory basis* if there exists an optimal memory allocation for the tree $G = (V, E)$ which assigns memory size $w_{opt}(v)$ to every vertex $v \in V$, such that for any $v \in V$, $W_{min}(v) \leq w(v) \leq w_{opt}(v)$.

Lemma 1 *u_1 is a child of u_2 in the tree $G = (V, E)$. And $\{w_1(v) | v \in V\}$ is an optimal memory basis. Assume the following conditions are true for the memory allocation problem: for any vertex $v \in Des(u_1)$, the 'requirement set' $R(v) = \emptyset$; $R(u_1)$ has an element (r, k), namely, $(r, k) \in R(u_1)$.*

We define S_1 as $S_1 = N(u_2, r - d(u_1, u_2))$, and define S_2 as $S_2 = N(u_1, r) - S_1$. We compute the elements of a set $\{w_2(v) | v \in V\}$ in the following way (step 1 to step 3): Step 1: for all $v \in V$, let $w_2(v) \leftarrow w_1(v)$. Step 2: Let $X \leftarrow \max\{0, k - \sum_{v \in S_1} W_{max}(v) - \sum_{v \in S_2} w_1(v)\}$, and let $C \leftarrow S_2$. Step 3: Let v_0 be the vertex in C that is the closest to u_1—namely, $d(v_0, u_1) = \min_{v \in C} d(v, u_1)$; let $w_2(v_0) \leftarrow \min\{W_{max}(v_0), w_1(v_0) + X\}$; let $X \leftarrow X - (w_2(v_0) - w_1(v_0))$, and let $C \leftarrow C - \{v_0\}$; repeat Step 3 until X equals 0.

Then the following conclusion is true: $\{w_2(v) | v \in V\}$ is also an optimal memory basis. □

Lemma 2 *u_1 is a child of u_2 in the tree $G = (V, E)$. And $\{w_0(v) | v \in V\}$ is an optimal memory basis. Assume the following conditions are true for the memory allocation problem: for any vertex $v \in Des(u_1)$, the 'requirement set' $R(v) = \emptyset$; for any element in $R(u_1)$—say the element is (r, k)—we have $\sum_{u \in N(u_2, r - d(u_1, u_2))} W_{max}(u) + \sum_{u \in N(u_1, r) - N(u_2, r - d(u_1, u_2))} w_0(u) \geq k$.*

We compute the elements of a set $\{\hat{R}(v) | v \in V\}$ in the following way (step 1 and step 2): Step 1: for all $v \in V$, let $\hat{R}(v) \leftarrow R(v)$. Step 2: let (r, k) be an element in $\hat{R}(u_1)$; if $\sum_{v \in N(u_1, r)} w_0(v) < k$, then add an element $(r - d(u_1, u_2), k - \sum_{v \in N(u_1, r) - N(u_2, r - d(u_1, u_2))} w_0(v))$ to the set $\hat{R}(u_2)$; remove the element (r, k) from $\hat{R}(u_1)$; repeat Step 2 until $\hat{R}(u_1)$ becomes an empty set.

Let's call the original memory allocation problem, in which the 'requirement set' of each vertex $v \in V$ is $R(v)$, the 'old problem'. *We derive a new memory allocation problem—which we call the* 'new problem'*—in the following way: in the* 'new problem' *everything is the same as in the* 'old problem', *except that for each vertex $v \in V$, its 'requirement set' is $\hat{R}(v)$ instead of $R(v)$, and its 'minimum memory size' is $w_0(v)$ instead of $W_{min}(v)$. Then the following conclusions are true: (1) The* 'new problem' *has a solution (an optimal memory allocation). (2) An optimal memory allocation for the* 'new problem' *is also an optimal memory allocation for the* 'old problem'. □

Based on Lemma 1 and Lemma 2, we present an algorithm of complexity $O(q|V|^3)$ which finds an optimal memory allocation for a tree network $G = (V, E)$, where $|V|$ is the number of vertices and q is the average cardinality of a requirement set, namely, $q = \frac{1}{|V|} \sum_{v \in V} |R(v)|$. When $W_{max}(v) = \infty$ for every $v \in V$, an algorithm of complexity $O(q|V|^2)$ is presented, solving the same problem. A third algorithm of complexity $O(q|V|^3 log(Y - \frac{X}{|V|}))$ is presented, which finds a solution that minimizes the greatest memory size of single nodes, among all the optimal memory allocations. Here Y is the greatest memory size of single nodes in some optimal memory-allocation solution, and X is the total memory size in that solution. For an extended version of this paper, please see [1].

References

[1] A. Jiang and J. Bruck, "Memory allocation in information storage networks," technical report, http://www.paradise.caltech.edu/papers/ETR048.ps, 2002.

[1]This work was supported in part by the Lee Center for Advanced Networking at the California Institute of Technology, and by NSF grant CCR-TC-0209042.