ELSEVIER

Information Processing Letters 49 (1994) 287–290

Information Processing Letters

Extending the quadrangle inequality to speed-up dynamic programming

Al Borchers *, Prosenjit Gupta

Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, USA

Communicated by M.J. Atallah; received 7 December 1992; revised 27 December 1993

Abstract

Proving that a function satisfies the quadrangle inequality is a powerful and elegant way to show that a dynamic programming algorithm to compute that function can be sped up by a factor of the input size. In this paper we consider two problems that do no fit in the usual general cases of functions that satisfy the quadrangle inequality but for which the proof of the quadrangle inequality still carries through: the multi-peg Tower of Hanoi problem with weighted disks and the problem of constructing a Rectilinear Steiner Minimal Arborescence (RSMA) on a slide. We prove the quadrangle inequality holds for a generalized function that unifies the two problems. This speeds up algorithms for these problems from $O(n^3p)$ to $O(n^2p)$ and from $O(n^3)$ to $O(n^2)$ respectively.

Key words: Analysis of algorithms; Combinatorial problems; Algorithms

1. Introduction

Yao [3] considered the following recurrence:

$$c(i,i) = 0,$$
$$c(i,j) = w(i,j) + \min_{i<k\leqslant j}(c(i,k-1) + c(k,j)).$$

She proved that if the weight function w satisfies the quadrangle inequality

$$w(i,k) + w(j,l) \leqslant w(j,k) + w(i,l), \quad \text{for all } i \leqslant j \leqslant k \leqslant l$$

and is also monotone on the lattice of intervals, that is,

$$w(i,j) \leqslant w(i',j'), \quad \text{if } i' \leqslant i \leqslant j \leqslant j'$$

then the obvious dynamic programming algorithm could be sped up from $O(n^3)$ to $O(n^2)$. She showed first that under these conditions function c also satisfies the quadrangle inequality, and then that this fact can be used to reduce the number of subproblems the dynamic programming algorithm needs to check in searching for a minimum solution, giving the speed up.

We consider two problems that are of a different form than the one above, but for which the proof still carries through. The first problem is the well-known Towers of Hanoi problem, with the addition of auxiliary pegs and weighted disks [1]. The problem is to find the minimum cost solution that moves all the disks from the leftmost peg to the rightmost without ever putting a smaller disk on top of a larger one. The cost of moving a disk from one peg to another (provided it is free to move) is the weight of the disk.

* Corresponding author. Email: borchers@cs.umn.edu.

SSDI 0020-0190(94)00004-I

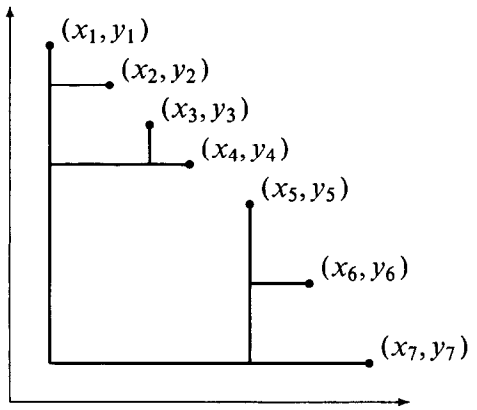

Fig. 1. Rectilinear Steiner arborescence on a slide.

One possible solution to this problem is to calculate the following function (it is unknown if this actually provides the minimum solution). Here $T(i,j,p)$ represents the cost of moving disks $i, i+1, \ldots, j$ from one peg to another with p auxiliary pegs. The weight of disk i is $w(i)$.

$$T(i,j,p) = \begin{cases} w(i), & \text{if } i = j \text{ and } p \geqslant 0, \\ \min\limits_{i \leqslant s < j} (2T(i,s,p) \\ \qquad + T(s+1,j,p-1)), \\ \qquad \text{if } i < j \text{ and } p > 0, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

An $O(n^3 p)$ time dynamic programming algorithm is straightforward. In [1] it was conjectured this could be reduced to $O(n^2 p)$, and in this paper we show this conjecture is true.

The second problem is to construct a minimum length rectilinear directed tree with all edges going up or to the right to interconnect a slide of points [2]. A slide is a set of points (x_i, y_i) where for $i < j$, $x_i < x_j$ and $y_i > y_j$, see Fig. 1.

Since each subtree must be minimal, a dynamic programming solution gives the minimal length tree. Let $L(i,j)$ be the length of the RSMA interconnecting points (x_i, y_i), $(x_{i+1}, y_{i+1}), \ldots, (x_j, y_j)$. Then the length of an RSMA is given by

$$L(i,j) = \min_{i \leqslant s < j} (L(i,s) + L(s+1,j) + x_{s+1} - x_i + y_s - y_j).$$

Here an $O(n^3)$ dynamic programming algorithm is given in [2]. We show this can be reduced to $O(n^2)$.

2. A generalized problem

In both problems we want to compute a function that gives us the total cost of the minimum cost solution. Let $F(i,j,r)$ be the minimum cost of solving a problem on inputs $i, i+1, \ldots, j$ with resource r. We will compute $F(i,j,r)$ by splitting it into two subproblems $F(i,s,\ldots)$ and $F(s+1,k,\ldots)$ at $i \leqslant s < j$. We call the value s where the problem is divided a splitting point. Let $w(i)$ be the cost of $F(i,i,r)$ for $r \geqslant 0$. At each division of the problem into subproblems, we assume the resource does not increase; this is given by the two functions $f(r)$ and $g(r)$, where $r \geqslant f(r)$ and $r \geqslant g(r)$. The function $h(i,s,j)$ is the cost of dividing the problem $F(i,j,r)$ at splitting point s. Finally, a and b are two positive integral coefficients that weigh the cost of the subproblems before and after the splitting point. Our generalized function will then be defined as follows.

$$F(i,j,r) = \begin{cases} w(i), & \text{if } i = j \text{ and } r \geqslant 0, \\ \min\limits_{i \leqslant s < j} (aF(i,s,f(r)) \\ \qquad + bF(s+1,j,g(r)) \\ \qquad + h(i,s,j)), \\ \qquad \text{if } f(r) \geqslant 0 \text{ and } g(r) \geqslant 0, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

The general conditions on h are similar to the quadrangle inequality, for example when $k = l$ below; they can also be interpreted geometrically as a six-point "hexagonal inequality" that follows from the quadrangle inequality. The conditions are, for $i \leqslant j \leqslant t < k \leqslant l$ and $i \leqslant s < l$,

if $t \leqslant s$, then
$h(i,t,k) - h(j,t,k) + h(j,s,l) - h(i,s,l) \leqslant 0$, and
$h(j,s,l) - h(i,s,l) \leqslant 0$;

if $s \leqslant t$, then
$h(j,t,l) - h(j,t,k) + h(i,s,k) - h(i,s,l) \leqslant 0$, and
$h(i,s,k) - h(i,s,l) \leqslant 0$.

For the RSMA problem $h(i,s,j) = x_{s+1} - x_i + y_s - y_j$; in this case it is easy to verify these conditions are met.

A straightforward dynamic programming algorithm can calculate $F(1,n,r)$ in $O(n^3 r)$ time. Using the quadrangle inequality, Lemma 1, we show this can be improved to $O(n^2 r)$ time.

3. Proof of the quadrangle inequality

Lemma 1. (The Quadrangle Inequality) *If* $i \leqslant j \leqslant k \leqslant l$ *then*

$$F(i,k,r) + F(j,l,r) \leqslant F(i,l,r) + F(j,k,r).$$

Proof. The proof is by induction on $l - i$. If $l - i = 0$ then $i = j = k = l$ and the lemma is trivially true.

Assume the lemma is true for $l - i < N$ and assume $l - i = N > 0$. Note that since $l - i \geqslant 1$, $f(r)$ and $g(r)$ must be at least 0 otherwise $F(i,l,r)$ is not defined.

The function $F(i,l,r)$ attains a minimum at some splitting point, say s. Then

$$F(i,l,r) = aF(i,s,f(r)) + bF(s+1,l,g(r)) + h(i,s,l).$$

If $j < k$ then for some t we also have

$$F(j,k,r) = aF(j,t,f(r)) + bF(t+1,k,g(r)) + h(j,t,k).$$

We know $i \leqslant s < l$ and if $j < k$ then $j \leqslant t < k$. There are four cases to consider:

Case 1: $j < k$ and $s \geqslant t$. Since $F(i,k,r)$ and $F(j,l,r)$ are minimal over all splitting points, we have

$$\begin{aligned} &F(i,k,r) + F(j,l,r) \\ &\quad \leqslant aF(i,t,f(r)) + bF(t+1,k,g(r)) \\ &\qquad + h(i,t,k) + aF(j,s,f(r)) \\ &\qquad + bF(s+1,l,g(r)) + h(j,s,l). \end{aligned}$$

Now since $i \leqslant j \leqslant t \leqslant s < l$ we can use the induction hypothesis,

$$F(i,t,f(r)) + F(j,s,f(r)) \leqslant F(i,s,f(r)) + F(j,t,f(r)),$$

to get

$$\begin{aligned} &F(i,k,r) + F(j,l,r) \\ &\quad \leqslant aF(j,t,f(r)) + bF(t+1,k,g(r)) \\ &\qquad + h(i,t,k) + aF(i,s,f(r)) \\ &\qquad + bF(s+1,l,g(r)) + h(j,s,l) \\ &\quad = F(j,k,r) + h(i,t,k) - h(j,t,k) \\ &\qquad + F(i,l,r) + h(j,s,l) - h(i,s,l) \\ &\quad = F(j,k,r) + F(i,l,r) \end{aligned}$$

since s and t are minimal splitting points for $F(i,l,r)$ and $F(j,k,r)$ and by our property on h.

Case 2: $j < k$ and $s < t$. This case is similar to Case 1, splitting $F(i,k,r)$ at s and $F(j,l,r)$ at t and applying the induction hypothesis with $s + 1 < t + 1 \leqslant k \leqslant l$ to the terms with coefficient b.

Case 3: $j = k$ and $s \geqslant j$. Since $r \geqslant f(r)$ it is easy to see that $F(i,k,r) \leqslant F(i,k,f(r))$. Also, $i \leqslant j = k$, so clearly $F(j,k,f(r)) \leqslant F(i,k,f(r))$. Combining these inequalities and using $a \geqslant 1$, we get

$$F(i,k,r) \leqslant aF(i,k,f(r)) - (a-1)F(j,k,f(r)).$$

Then since $F(j,l,r)$ is minimal over all splitting points we get

$$\begin{aligned} &F(i,k,r) + F(j,l,r) \\ &\quad \leqslant aF(i,k,f(r)) - (a-1)F(j,k,f(r)) \\ &\qquad + aF(j,s,f(r)) + bF(s+1,l,g(r)) \\ &\qquad + h(j,s,l). \end{aligned}$$

Using the induction hypothesis on $i \leqslant j = k \leqslant s$,

$$\begin{aligned} F(i,k,r) &+ F(j,l,r) \\ &\leqslant aF(j,k,f(r)) - (a-1)F(j,k,f(r)) \\ &\quad + aF(i,s,f(r)) + bF(s+1,l,g(r)) \\ &\quad + h(j,s,l) \\ &= F(j,k,f(r)) + F(i,l,r) \\ &\quad + h(j,s,l) - h(i,s,l) \\ &\leqslant F(j,k,r) + F(i,l,r). \end{aligned}$$

This last step follows by our conditions on h and the fact that $F(j,k,f(r)) = w(j) = F(j,k,r)$ since $j = k$ and $r \geqslant f(r) \geqslant 0$.

Case 4: $j = k$ and $s < j$. This case is similar to Case 3, splitting $F(i,k,r)$ at s, using the inequality

$$\begin{aligned} F(j,l,r) &\leqslant bF(j,l,g(r)) \\ &\quad - (b-1)F(j,k,g(r)), \end{aligned}$$

and using the induction hypothesis with $s+1 \leqslant j = k \leqslant l$. □

Lemma 2. (The Monotone Property) *Let $M(i,j,r)$ be the largest splitting point for $F(i,j,r)$ that gives a minimum value of F. Then*

$$\begin{aligned} M(i,j,r) &\leqslant M(i,j+1,r) \\ &\leqslant M(i+1,j+1,r). \end{aligned}$$

Theorem 3. *The value of $F(1,n,r)$ can be computed in $O(n^2 r)$ time.*

These last two results follow as in Yao's paper [3]. This shows that for the Tower of Hanoi problem with p auxiliary pegs and n weighted disks we can compute a solution in $O(n^2 p)$ time. For a slide of n points we can compute a Rectilinear Steiner Minimal Arborescence in $O(n^2)$ time.

References

[1] P. Gupta, P.P. Chakrabarti and S. Ghose, The Towers of Hanoi: Generalizations, specializations and algorithms, *Internat. J. Comput. Math.* **46** (1993) 149–161.

[2] S.K. Rao, P. Sadayappan, F.K. Hwang and P.W. Shor, The Rectilinear Steiner Arborescence Problem, *Algorithmica*, **7** (2–3) (1992) 277–288.

[3] F.F. Yao, Efficient dynamic programming using quadrangle inequalities, in: *Proc. 12th Ann. ACM Symp. on Theory of Computing* (1980) 429–435.