

A Fast Scaling Algorithm for Minimizing Separable Convex Functions Subject to Chain Constraints

Ravindra K. Ahuja
Industrial and Systems Engineering Department
University of Florida
Gainesville, FL 32611, USA
ahuja@ufl.edu

James B. Orlin
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
jorlin@mit.edu

(Revised April 30, 2001)

A Fast Scaling Algorithm for Minimizing Separable Convex Functions Subject to Chain Constraints

Ravindra K. Ahuja¹ and James B. Orlin²

ABSTRACT

In this paper, we consider the problem of minimizing $\sum_{j \in N} C_j(x_j)$, subject to the following chain constraints $l \leq x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n \leq u$, where $C_j(x_j)$ is a convex function of x_j for each $j \in N = \{1, 2, \dots, n\}$. This problem is a generalization of the *isotonic regression problems* with complete order, an important class of problems in regression analysis that has been examined extensively in the literature. We refer to this problem as the *generalized isotonic regression problem*. In this paper, we focus on developing a fast scaling algorithm to obtain an integer solution of the generalized isotonic regression problem. Let U denote the difference between an upper bound on an optimal value of x_n and a lower bound on an optimal value of x_1 . Under the assumption that the evaluation of any function $C_j(x_j)$ takes $O(1)$ time, we show that the generalized isotonic regression problem can be solved in $O(n \log U)$ time. This improves by a factor of n the previous best running time of $O(n^2 \log U)$ to solve the same problem. In addition, when our algorithm is specialized to the *isotonic median regression problem* (where $C_j(x_j) = c_j |x_j - a_j|$) for specified values of c_j 's and a_j 's, the algorithm obtains a real-valued optimal solution in $O(n \log n)$ time. This time bound matches the best available time bound to solve the isotonic median regression problem, but our algorithm uses simpler data structures and may be easier to implement.

¹ Industrial and Systems Engineering Department, University of Florida, Gainesville, FL 32611, USA.

² Sloan School of Management, MIT, Cambridge, MA 02139, USA.

1. INTRODUCTION

In this section, we study the following separable convex minimization problem subject to chain constraints:

$$\text{Minimize } \sum_{j \in N} C_j(x_j) \tag{1a}$$

subject to

$$l \leq x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n \leq u, \tag{1b}$$

where $C_j(x_j)$ is a strictly convex function of x_j for each $j \in N = \{1, 2, \dots, n\}$. Let u denote an upper bound on the optimal value of x_n , l denote a lower bound of the optimal value of x_1 , and $U = u - l$. Our development assumes $C_j(x_j)$ to be convex functions (not strictly convex functions).

The problem (1) is a generalization of the *isotonic regression problem* defined as follows: Given the vector $a = \{a_1, a_2, \dots, a_n\} \in \mathbb{R}^n$ and an integer number p , find $x = \{x_1, x_2, \dots, x_n\}$, so as to minimize

$$\|x - a\|_p = \sqrt[p]{\sum_{j \in N} (x_j - a_j)^p} . \tag{2}$$

subject to the constraints (1b). The most studied special case of the isotonic regression problem is when $p = 2$. When $p = 1$, this problem is referred to as the *isotonic median regression problem*. Clearly, (1) subsumes the isotonic regression problem for every integer p ; hence we refer to (1) as the *generalized isotonic regression problem*.

The generalized isotonic regression problem finds applications in operations research (see, for example, Maxwell and Muckstadt [1983], Roudy [1986], Kaufman and Tamir [1993], and Ahuja and Orlin [1996]), statistics (see, for example, Barlow, et al. [1972], Lee [1983], and Robertson, Wright, and Dykstra [1988]), and image processing (Restrepo and Bovik [1994]). Here is a simple application of the generalized regression problem: Consider a fuel tank where fuel is being consumed at a slow pace and measurements of the fuel tank are taken at different points in time. Suppose that these measurements are a_1, a_2, \dots, a_n . Due to errors in measurements, these numbers may not be in the non-increasing order despite the fact that the true amounts of fuel remaining in the tank are non-increasing. However, we need to determine these measurements as accurately as possible. One possible way to accomplish this could be to

perturb these numbers to x_1, x_2, \dots, x_n so that $x_1 \geq x_2 \geq \dots \geq x_n$ and the cost of perturbation given by $C_1(x_1 - a_1) + C_2(x_2 - a_2) + \dots + C_n(x_n - a_n)$ is minimum, where $C_j(x_j)$'s are convex functions that give the cost of perturbing the data. This problem may be transformed to the generalized isotonic regression problem by replacing x_j 's by their negatives.

The generalized isotonic regression problem and its special cases have been extensively studied in the literature. The following list of references on this problem demonstrates the level of interest in this problem in the field of operations research and statistics: Ayer et al. [1955], Brunk [1955], Robertson and Waltman [1968], Gebhardt [1970], Barlow et al. [1972], Robertson and Wright [1973, 1980], Ubhaya [1974a, 1974b, 1979, 1987], Casady and Cryer [1976], Goldstein and Kruskal [1976], Dykstra [1981], Lee [1983], Maxwell and Muchstadt [1983], Roundy [1986], Menendez and Salvador [1987], Robertson et al. [1988], Chakravarti [1989], Best and Chakravarti [1990], Stromberg [1991], Best and Tan [1993], Tamir [1993], Eddy et al. [1995], Pardalos et al. [1995], Shi [1995], Best, Chakravarti, and Ubhaya [1996], Liu and Ubhaya [1997], Schell and Singh [1997], and Pardalos and Xue [1998]. It is well known that the isotone regression problem for $p = 1$ can be solved in $O(n \log n)$ time and for $p = 2$ and $p = \infty$, it can be solved in $O(n)$ time (see, for example, Best and Chakravarti [1990], Pardalos et al. [1995], and Liu and Ubhaya [1997]).

In this paper, we focus on developing a faster algorithm to obtain an integer optimal solution of the generalized isotonic regression problem. The Pool Adjacent Violators (PAV) algorithm is the most extensively studied algorithm for the isotonic regression problem. Under the assumption that the evaluation of any function $C_j(x_j)$ takes $O(1)$ time, the PAV algorithm runs in $O(n^2 \log U)$ time (see, for example, Stromberg [1991] and Best, Chakravarti, and Ubhaya [1996]). We use a scaling technique to improve the running time of the PAV algorithm to $O(n \log U)$ time; thereby obtaining a speedup by a factor of n . In addition, when our algorithm is specialized to the isotonic median regression problem (where the objective is to minimize $\sum_{j \in N} w_j |x_j - a_j|$), our algorithm obtains its real-valued optimal solution in $O(n \log n)$ time. This matches the best available time bound to solve the same problem due to several researchers including Pardalos et al. [1995]. However, whereas existing $O(n \log n)$ algorithms use balanced binary trees data structure, our algorithm uses fairly simple data structures.

2. PRELIMINARIES

In this section, we present some background material.

Assumption:

We consider the generalized isotonic regression problem subject to the following assumption:

Assumption 1. Each function $C_j(x_j)$ can be evaluated in $O(1)$ time for a given value of x_j .

This assumption allows us to analyze the worst-case complexity of the algorithms developed in this paper since they all involve evaluating the cost functions.

Lower and Upper Bounds on x^* :

Let θ_j denote the value of x_j at which $C_j(x_j)$ attains the minimum value; if θ_j is non-unique, we choose the minimum among such values. Let $\theta_{\min} = \min\{\theta_j : j \in N\}$ and $\theta_{\max} = \max\{\theta_j : j \in N\}$. It is easy to observe that there exists an optimal solution x^* of (1) where $\theta_{\min} \leq x_j^* \leq \theta_{\max}$ for all $j \in N$. Therefore, we can set $l = \theta_{\min}$ and $u = \theta_{\max}$.

Minimizing Single-Variable Convex Functions:

The PAV algorithm and its variants proceed by finding the minimum of a single-variable convex function $F(\theta)$ that varies in the range $[l, u]$. There are several well known search methods, including binary search and Fibonacci search, that maintain a search interval containing the optimal solution and perform one or two function evaluations to reduce the search interval by a constant factor (see, for example, Bazaraa, Sherali and Shetty [1993]). These search methods terminate when the length of the search interval decreases below some acceptable limit ϵ . The number of iterations performed by these search methods is $O(\log(U/\epsilon))$. Each iteration of these search methods performs $O(1)$ function evaluations; hence, the running time of these search methods is $O(\log(U/\epsilon))$ evaluations of the function $F(\theta)$. In case we want to find an integer optimal solution of the function $F(\theta)$, then we can terminate the search method whenever $\epsilon < 1$. In this case, the running time of the method will be the time taken by $O(\log U)$ function evaluations.

Partitions and Blocks:

The PAV algorithm maintains a partition \mathbf{J} of the set of indices $\{1, 2, 3, \dots, n\}$ into sets of consecutive integers, called *blocks*. For example, if $\mathbf{J} = \{1, 2, 3, \dots, 10\}$, then one possible

partition is $\mathbf{J} = \{[1, 2, 3], [4, 5], [6], [7, 8, 9, 10]\}$ consisting of four blocks: $[1, 2, 3]$, $[4, 5]$, $[6]$, and $[7, 8, 9, 10]$. Since a block consists of a sequence of adjacent integers, we can refer to a block as $[p, q]$ implying that it contains indices p through q inclusive.

We call a block $[p, q]$ to be a *single-valued block* if the following subproblem of (1) has an optimal solution where all variables have the same value; that is, $x_p^* = x_{p+1}^* = \dots = x_q^* = \theta$ for some θ :

$$\text{Minimize } \sum_{j=p}^q C_j(x_j), \text{ subject to } x_p \leq x_{p+1} \leq \dots \leq x_q. \quad (3)$$

The PAV algorithm maintains partitions where each block is a single-valued block. For simplicity, we will henceforth call a single-valued block as a block. We define the function $F(p, q, \theta)$ for a block $[p, q]$ in the following manner:

$$F(p, q, \theta) = \sum_{j=p}^q C_j(\theta). \quad (4)$$

Since each function $C_j(\theta)$ is a convex function of θ , it follows that for fixed values of p and q , $F(p, q, \theta)$ is also a convex function. Let θ_{pq} denote a value of $\theta \in [l, u]$ for which $F(p, q, \theta)$ attains its minimum value. If $F(p, q, \theta)$ does not have a unique value of θ_{pq} , then we can use any of these values; we will use the convention that we choose the minimum among these values as θ_{pq} .

The preceding discussion implies that each partition \mathbf{J} maintained by the PAV algorithm has a unique solution \mathbf{x} associated with it; we obtain this solution by considering each block $[p, q]$ in \mathbf{J} one by one and setting $x_j = \theta_{pq}$ for every $j, p \leq j \leq q$.

We refer to two blocks $[p, q]$ and $[q+1, r]$ as *consecutive*. We refer to two consecutive blocks $[p, q]$ and $[q+1, r]$ as *in-order* if $\theta_{pq} \leq \theta_{q+1,r}$ and *out-of-order* otherwise. The following results are well known (see, for example, Best, Chakravarti, and Ubhaya [1996]):

Lemma 1: *Suppose that each of the two adjacent blocks $[p, q]$ and $[q+1, r]$ are single-valued blocks. If $\theta_{pq} \geq \theta_{q+1,r}$, then the block $[p, r]$ is also a single-valued block.*

Lemma 2: *If \mathcal{J} is a partition where each block is single-valued and every consecutive block is in-order, then the solution associated with this partition is an optimal solution of the generalized isotonic regression problem.*

3. The PAV Algorithm

The PAV algorithm uses the results contained in Lemma 1 and Lemma 2 to solve (1). It maintains a partition \mathcal{J} of (single-valued) blocks. The partition \mathcal{J} may contain consecutive blocks that are out-of-order. In every iteration, the algorithm selects out-of-order blocks $[p, q]$ and $[q+1, r]$ in \mathcal{J} and replaces the blocks $[p, q]$ and $[q+1, r]$ by the block $[p, r]$; we refer to this process as *merging*. The algorithm repeatedly merges out-of-order blocks until there are no out-of-order blocks. It follows by Lemma 2 that the solution corresponding to this partition is an optimal solution of (1). We give in Figure 1 an algorithmic description of this algorithm.

```

algorithm PAV;
begin
   $\mathcal{J} = [[1, 1], [2, 2], \dots, [n, n]]$ ;
  while there exists an out-of-order pair of adjacent blocks in  $\mathcal{J}$  do
    begin
      select a pair of out-of-order blocks  $[p, q]$  and  $[q+1, r]$  in  $\mathcal{J}$ ;
      replace the two blocks  $[p, q]$  and  $[q+1, r]$  by the block  $[p, r]$  and update  $\mathcal{J}$ ;
      compute  $\theta_{pr}$ ;
    end;
  for each block  $[p, q] \in \mathcal{J}$  do  $x_j^* = \theta_{pq}$  for all  $j \in [p, q]$ ;
   $x^*$  is an optimal solution of the generalized isotonic regression problem;
end;

```

Figure 1. The PAV algorithm.

We now analyze the worst-case complexity of the PAV algorithm. First we consider the time needed to identify out-of-order blocks. At the beginning of the algorithm, there are at most n out-of-order blocks. Subsequently, whenever a merge operation is performed, a new out-of-order block may be created involving the newly created block. Using simple data structures, we can easily keep track of the pairs of out-of-order blocks and select them in $O(1)$ time per pair and

in $O(n)$ total time. Consequently, identifying out-of-order blocks is not a bottleneck operation in the algorithm.

We next consider the merge operation. Each merge operation decreases the number of blocks by one; hence, there will be at most $n-1$ merge operations. The bottleneck operation in a merge operation is the computation of θ_{pr} for the block $[p, r]$ and this involves determining the minimum of the convex function $F(p, q, \theta) = \sum_{j=p}^q C_j(\theta)$. Since we are interested in an integer optimal solution of (1), we determine an integer optimal solution of the function $F(p, q, \theta)$. We have seen in Section 2 that finding an integer optimal solution of a convex function $C_j(\theta)$ requires $O(\log U)$ function evaluations. Each evaluation of the function $F(p, q, \theta)$ takes $O(n)$ time since it may involve as many as n function evaluations, each of which can be performed in $O(1)$ time (from Assumption 1). Hence the following theorem.

Theorem 1. *The PAV algorithm obtains an optimal integer solution of the convex ordered set problem in $O(n^2 \log U)$ time.*

It is easy to see that if we want to determine an optimal fractional solution of the convex ordered set problem where the fraction has a denominator of K , then the generalized isotonic regression algorithm would take $O(n^2 \log(UK))$ time.

4. A SCALING APPROACH FOR THE GENERALIZED ISOTONIC REGRESSION PROBLEM

In this section, we will describe an improvement of the PAV algorithm that determines an optimal integer solution of (1) in $O(n \log U)$ time. The improved algorithm uses a scaling technique in the PAV to obtain a speedup by a factor of $O(n)$. Scaling techniques are widely used in the literature to improve the running times of discrete and network optimization problems. We refer the reader to the book of Ahuja, Magnanti and Orlin [1993] for the use of scaling techniques for network optimization problems.

A scaling algorithm typically decomposes an optimization problem into a series of approximate problems and gradually refines the approximation. In the PAV algorithm described in Section 3, the computation of θ_{pq} was a bottleneck operation. We needed θ_{pq} to identify out-of-order pairs of blocks. The scaling algorithm computes θ_{pq} approximately as $\theta_{pq}^\Delta = \Delta \lfloor \theta_{pq} / \Delta \rfloor$, which is the largest integral multiple of Δ less than or equal to θ_{pq} . Since we are interested in the

optimal integer solution of the generalized isotonic regression problem, $\theta_{pq}^1 = \theta_{pq}$. Therefore, if $\Delta = 1$, then $\theta_{pq}^\Delta = \theta_{pq}$.

The scaling version of the PAV algorithm, called the *scaling PAV algorithm*, performs a number of scaling phases. We refer to a scaling phase with a specific value of Δ as the Δ -*scaling phase*. The algorithm starts with $\Delta = 2^{\lfloor \log(U+1) \rfloor}$ and in each subsequent scaling phase decreases Δ by a factor of 2. Eventually, Δ becomes 1 and the algorithm terminates with an optimal integral solution of the generalized isotonic regression problem. The definition of θ_{pq}^Δ implies the following property:

Property 1. $\theta_{pq}^\Delta \leq \theta_{pq} < \theta_{pq}^\Delta + \Delta$.

Our scaling algorithm also uses the following lemma:

Lemma 3. *For a pair of adjacent blocks $[p, q]$ and $[q+1, r]$,*

- (a) *if $\theta_{pq}^\Delta > \theta_{q+1,r}^\Delta$, then $\theta_{pq} > \theta_{q+1,r}$; and*
- (b) *if $\theta_{pq}^\Delta < \theta_{q+1,r}^\Delta$, then $\theta_{pq} < \theta_{q+1,r}$*

Proof: Observe that if $\theta_{pq} \leq \theta_{q+1,r}$, then $\theta_{pq}^\Delta \leq \theta_{q+1,r}^\Delta$. The contrapositive of this result is the result in part (a). The proof of part (b) is similar. ■

Similar to the PAV algorithm described in Section 3, the scaling PAV algorithm maintains a partition \mathbf{J} of blocks. For a partition \mathbf{J} , we associate a solution x_j^Δ in the following manner: for every block $[p, q] \in \mathbf{J}$, we set $x_j^\Delta = \theta_{pq}^\Delta$ for all $j \in [p, q]$. In the Δ -scaling phase, we define a pair of adjacent blocks $[p, q]$ and $[q+1, r]$ to be Δ -*out-of-order* if $\theta_{pq}^\Delta > \theta_{q+1,r}^\Delta$, and Δ -*in-order* otherwise. We call a partition \mathbf{J} to be Δ -*optimal* if it contains no Δ -out-of-order pair of adjacent blocks.

We give an algorithmic description of the scaling PAV algorithm in Figure 2. The algorithm starts with a sufficiently large value of Δ and a partition \mathbf{J} that is Δ -optimal. It then repeatedly calls the procedure `improve-approximation(\mathbf{J}, Δ)` which takes a 2Δ -optimal partition

\mathcal{J} and converts it into a Δ -optimal partition \mathcal{J} . The procedure first computes θ_{pq}^Δ for each block $[p, q] \in \mathcal{J}$. We will show later how it computes θ_{pq}^Δ using $\theta_{pq}^{2\Delta}$ efficiently. It then identifies Δ -out-of-order pair of blocks (say, $[p, q]$ and $[q+1, r]$) and replaces them by the merged block $[p, r]$. It then computes θ_{pr}^Δ . When there is no Δ -out-of-order pair of blocks in the partition \mathcal{J} , the procedure terminates. The algorithm repeats this process until $\Delta = 1$, at which point the solution associated with the partition \mathcal{J} satisfies the conditions in Lemma 2 and the algorithm terminates with an optimal solution of the generalized isotonic regression problem.

algorithm *scaling PAV*;

begin

$\Delta := 2^{\lfloor \log(U+1) \rfloor}$;

$\mathcal{J} := [[1, 1], [2, 2], \dots, [n, n]]$;

if $l < 0$ **then** $\text{temp} := -\Delta$ **else** $\text{temp} := 0$;

for each $i := 1$ **to** n **do** $\theta_{i1}^1 := \text{temp}$;

while $\Delta > 1$ **do** *improve-approximation*(\mathcal{J}, Δ);

for each subset $[p, q] \in \mathcal{J}$ **do** $x_j^* = \theta_{pq}^1$ for all $j \in [p, q]$;

x^* is an optimal solution of the generalized isotonic regression problem;

end;

procedure *improve-approximation*(\mathcal{J}, Δ);

begin

$\Delta := \Delta/2$;

for each subset $[p, q] \in \mathcal{J}$ **do** compute θ_{pq}^Δ ;

while the partition \mathcal{J} is not Δ -optimal **do**

begin

select a Δ -out-of-order pair of blocks $[p, q]$ and $[q+1, r]$;

replace the two blocks $[p, q]$ and $[q+1, r]$ by the block $[p, r]$ and update \mathcal{J} ;

compute θ_{pr}^Δ ;

end;

end;

Figure 2. The improved convex ordered set algorithm.

We will now discuss the worst-case complexity of the algorithm. The algorithm executes the procedure *improve-approximation* $O(\log U)$ times. We will show that the procedure can be implemented in $O(n)$ time, thus giving a time bound of $O(n \log U)$ for the algorithm. The

potential bottleneck step in the algorithm is the computation of θ_{pq}^Δ . The procedure uses $\theta_{pq}^{2\Delta}$ to compute θ_{pq}^Δ . The following lemma establishes a relationship between $\theta_{pq}^{2\Delta}$ and θ_{pq}^Δ .

Lemma 4. $\theta_{pq}^{2\Delta} \leq \theta_{pq}^\Delta \leq \theta_{pq}^{2\Delta} + \Delta$.

Proof: Property 1 implies that $\theta_{pq}^\Delta \leq \theta_{pq}$ (Result 1) and $\theta_{pq} < \theta_{pq}^\Delta + \Delta$ (Result 2). Property 1 also implies that $\theta_{pq}^{2\Delta} \leq \theta_{pq}$ (Result 3) and $\theta_{pq} < \theta_{pq}^{2\Delta} + 2\Delta$ (Result 4). Combining Result 2 and Result 3 yields $\theta_{pq}^{2\Delta} < \theta_{pq}^\Delta + \Delta$, and therefore $\theta_{pq}^{2\Delta} \leq \theta_{pq}^\Delta$ (because both sides are integral multiples of Δ). This establishes the first inequality in the statement of the lemma. Combining Result 1 and Result 4 yields $\theta_{pq}^\Delta < \theta_{pq}^{2\Delta} + 2\Delta$, and therefore $\theta_{pq}^\Delta \leq \theta_{pq}^{2\Delta} + \Delta$, establishing the second inequality in the lemma and completing the proof of the lemma. ■

At the beginning of the procedure improve-approximation in the Δ -scaling phase, we compute θ_{pq}^Δ for every subset $[p, q] \in \mathbf{J}$. From the previous scaling phase, we know the value of $\theta_{pq}^{2\Delta}$. It follows from Lemma 4 that $\theta_{pq}^\Delta = \theta_{pq}^{2\Delta}$ or $\theta_{pq}^\Delta = \theta_{pq}^{2\Delta} + \Delta$. If $\theta_{pq}^{2\Delta} + \Delta > u$, then clearly $\theta_{pq}^\Delta = \theta_{pq}^{2\Delta}$; otherwise we proceed further. It follows from the convexity of the function $F(p, q, \theta)$ and the fact that $F(p, q, \theta)$ attains its minimum at θ_{pq} , that if $\theta_{pq}^{2\Delta} + \Delta \leq \theta_{pq}$ then $\theta_{pq}^\Delta = \theta_{pq}^{2\Delta} + \Delta$; otherwise $\theta_{pq}^\Delta = \theta_{pq}^{2\Delta}$. We check whether $\theta_{pq}^{2\Delta} + \Delta \leq \theta_{pq}$ in the following manner. Let $\beta = \theta_{pq}^{2\Delta} + \Delta$. We compute $F(p, q, \beta-1)$ and $F(p, q, \beta)$. If $F(p, q, \beta) \leq F(p, q, \beta-1)$, then $\theta_{pq}^\Delta = \theta_{pq}^{2\Delta} + \Delta$; otherwise $\theta_{pq}^\Delta = \theta_{pq}^{2\Delta}$. This computation takes $O(p+q+1)$ time for the subset $[p, q]$ and $O(n)$ time for all the blocks in the partition \mathbf{J} .

The algorithm also determines the value of θ_{pr}^Δ for the block $[p, r]$ obtained by merging the blocks $[p, q]$ and $[q+1, r]$. Notice that we merge the blocks $[p, q]$ and $[q+1, r]$ in the Δ -scaling phase only if $\theta_{pq}^\Delta > \theta_{q+1,r}^\Delta$. If this merging occurs then $\theta_{pq}^{2\Delta} = \theta_{q+1,r}^{2\Delta}$; for if $\theta_{pq}^{2\Delta} > \theta_{q+1,r}^{2\Delta}$, we would have merged the blocks in the 2Δ -scaling phase, and if $\theta_{pq}^{2\Delta} < \theta_{q+1,r}^{2\Delta}$ then by Lemma 3(b) $\theta_{pq}^\Delta < \theta_{q+1,r}^\Delta$, giving a contradiction in both the cases. Since $\theta_{pq}^{2\Delta} = \theta_{q+1,r}^{2\Delta}$, it follows that $\theta_{pr}^{2\Delta} = \theta_{pq}^{2\Delta} = \theta_{q+1,r}^{2\Delta}$. Lemma 4 implies that $\theta_{pr}^\Delta = \theta_{pr}^{2\Delta}$ or $\theta_{pr}^\Delta = \theta_{pr}^{2\Delta} + \Delta$, whichever happens to give a lower value of the function $F(p, q, \theta)$. If $\theta_{pr}^{2\Delta} + \Delta > u$, then clearly $\theta_{pr}^\Delta = \theta_{pr}^{2\Delta}$; otherwise, we

proceed further. If $\theta_{pr}^{2\Delta} + \Delta \leq \theta_{pr}$ then $\theta_{pr}^\Delta = \theta_{pr}^{2\Delta} + \Delta$; otherwise $\theta_{pr}^\Delta = \theta_{pr}^{2\Delta}$. We check whether $\theta_{pr}^{2\Delta} + \Delta \leq \theta_{pr}$ in the following manner. Let $\beta = \theta_{pr}^{2\Delta} + \Delta$. We next compute $F(p, r, \beta-1)$ and $F(p, r, \beta)$. Now notice that $F(p, r, \theta) = F(p, q, \theta) + F(q+1, r, \theta)$. Since both $F(p, q, \theta)$ and $F(q+1, r, \theta)$ have been determined earlier in the algorithm for both $\theta = \beta-1$ and $\theta = \beta$, we can compute both $F(p, r, \beta-1)$ and $F(p, r, \beta)$ in $O(1)$ time. If $F(p, r, \beta) \leq F(p, r, \beta-1)$, then $\theta_{pr}^\Delta = \theta_{pr}^{2\Delta} + \Delta$; otherwise $\theta_{pr}^\Delta = \theta_{pr}^{2\Delta}$. We have thus shown that an execution of the procedure improve-approximation takes $O(n)$ time, giving us the following theorem.

Theorem 2. *The scaling PAV algorithm obtains an integer optimal solution of the generalized isotonic regression problem in $O(n \log U)$ time.*

5. SPECIAL CASES OF THE GENERALIZED ISOTONIC REGRESSION PROBLEM

Several special cases of the generalized isotonic problem have been examined in the literature and algorithms have been developed for solving them. It is well known that the PAV algorithm can be implemented to run in $O(1)$ for the quadratic cost case (L_2 norm) and the minimax cost case (L_∞ norm). A straightforward implementation of the PAV algorithm for the rectilinear cost case, called the *isotonic median regression problem*, runs (L_2 norm) in $O(n^2)$ time; however, an $O(n \log n)$ implementation using balanced binary trees has been developed by Pardalos et al. [1995]. We will show that the scaling PAV algorithm also yields an $O(n \log n)$ algorithm to solve the same problem. Our algorithm hence attains the best available time bound for the rectilinear cost case and, we believe that it will be easier to implement since it uses simpler data structures.

In the isotonic median regression problem the objective function is to obtain $x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n$ so as to minimize $\sum_{j=1}^n c_j |x_j - a_j|$, where c_j 's and a_j 's are specified constants and $c_j \geq 0$ for all $1 \leq j \leq n$. For this problem, $F(p, q, \theta) = \sum_{j=p}^q c_j |\theta - a_j|$, and it is well known (see, for example, Francis and White [1976]) that a ‘‘median solution’’ is its optimal solution. A solution θ equal to some a_j with no more than half of the sum of c_j 's on either side is said to be a *median solution*, that is, $\theta = a_k$ for some k satisfying $\sum_{j=1}^{k-1} c_j \leq \frac{1}{2} \sum_{j=1}^n c_j$ as well as $\sum_{j=k+1}^n c_j \leq \frac{1}{2} \sum_{j=1}^n c_j$. We can determine the exact value of θ_{pq} for the block $[p, q]$ in $O(q-p) = O(n)$ time by applying a median finding algorithm. Clearly, in this case the exact computation of θ_{pq} takes

$O(q-p) = O(n)$ time and, consequently, by applying the PAV algorithm we can solve the isotonic median regression problem in $O(n^2)$ time.

We next consider the adaptation of the scaling PAV algorithm. For this case, $U = \max\{a_j: 1 \leq j \leq n\} - \min\{a_j: 1 \leq j \leq n\}$. When the scaling PAV algorithm is applied to this problem, it runs in $O(n \log U)$ time. We will show that a simple transformation can be used to modify the problem so that all data is integer and $U = n$ and, consequently, the scaling PAV algorithm will solve this problem in $O(n \log n)$ time.

The scaling PAV algorithm proceeds by determining θ_{pq} values for the blocks $[p, q]$ obtained during its execution. The θ_{pq} value is the minimum value of the function $F(p, q, \theta) = \sum_{j=p}^q c_j |a_j - \theta|$. As observed above, $\theta_{pq} = a_k$ for some k satisfying $\sum_{a_j < a_k} c_j \leq \frac{1}{2} \sum_{j=p}^q c_j$ and $\sum_{a_j > a_k} c_j \leq \frac{1}{2} \sum_{j=p}^q c_j$. Now observe from this formula that while determining the median solution θ_{pq} , the magnitude of a_j 's is unimportant; it is the relative ordering of the a_j 's with respect to one-another that is important. This observation allows us to use the following method to determine the median solution for any block. We sort a_j 's in the non-decreasing order. Let $\sigma(j)$ denote the position of a_j in this order. For example, if $n = 5$, $a_1 = 50$, $a_2 = 10$, $a_3 = 70$, $a_4 = 20$, and $a_5 = 40$, then $\sigma(1) = 4$, $\sigma(2) = 1$, $\sigma(3) = 5$, $\sigma(4) = 2$, and $\sigma(5) = 3$. Observe that the median solution for the block $[p, q]$ is a_k for some k satisfying $\sum_{\sigma(j) < \sigma(k)} c_j \leq \frac{1}{2} \sum_{j=p}^q c_j$ and $\sum_{\sigma(j) > \sigma(k)} c_j \leq \frac{1}{2} \sum_{j=p}^q c_j$.

We next replace each a_j by $\sigma(j)$ and apply the scaling PAV algorithm. For the modified problem, all data is integer and $U = O(n)$, hence the scaling PAV algorithm would determine an optimal solution of this problem in $O(n \log n)$ time. In the optimal solution y^* , each number varies between 1 to n . We can convert the optimal solution y^* of the modified problem into an optimal solution x^* of the original problem in the following manner: $x_j^* = a_j$ if and only if $y_j^* = \sigma(j)$.

Notice that the optimal solution x^* of the original problem may or may not be integer. We summarize the preceding discussion in the form of the following theorem:

Theorem 3. *The scaling PAV algorithm solves the isotonic median regression problem in $O(n \log n)$ time.*

ACKNOWLEDGMENTS

We thank the referees for their insightful suggestions that led to an improved presentation of our results. The first author gratefully acknowledges the support of NSF Grant DMI-9900087. The research of the second author was supported by the NSF Grant DMI-9820998 and the Office of Naval Research Grant ONR N00014-98-1-0317.

REFERENCES

- Ahuja, R. K., T. L. Magnanti, and J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, NJ.
- Ahuja, R. K., and J. B. Orlin. 1996. Routing and Scheduling Algorithms for ADART. Working Paper, Sloan School of Management, MIT, Cambridge, MA.
- Ayer, M., H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman. 1955. An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics* **26**, 641-647.
- Barlow, R. E., D. J. Bartholomew, D. J. Bremner, and H. D. Brunk. 1972. *Statistical Inference under Order Restrictions*. Wiley, New York.
- Bazaraa, M., H. Sherali, and C. M. Shetty. 1993. *Nonlinear Programming: Theory and Algorithms*. John Wiley and Sons.
- Best, M. J., and N. Chakravarti. 1990. Active set algorithms for isotonic regression: A unifying framework. *Mathematical Programming* **47**, 425-439.
- Best, M. J., N. Chakravarti, and V. A. Ubhaya. 1996. Minimizing separable convex functions subject to simple chain constraints. Research Report, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Submitted for publication.
- Best, M. J., and R. Y. Tan. 1993. An $O(n^3 \log n)$ strongly polynomial algorithm for an isotonic regression knapsack problem. *Journal of Optimization Theory and Application* **79**, 463-478.
- Brunk, H. D. 1955. Maximum likelihood estimates of monotone parameters. *Annals of Mathematical Statistics* **26**, 607-616.
- Casady, R. E., and J. D. Cryer. 1976. Monotone percentile regression. *Computational Statistics and Data Analysis* **5**, 399-406.
- Chakravarti, N. 1989. Isotonic median regression: A linear programming approach. *Mathematics of Operations Research* **14**, 303-308.

- Chakravarti, N. 1992. Isotonic median regression for orders representable by rooted trees. *Naval Research Logistics* **39**, 591-611.
- Dykstra, R. L. 1981. An isotonic regression algorithm. *Journal of Statistical Planning and Inference* **5**, 355-363.
- Eddy, W. F., S. Qian, and S. Sampson. 1995. Isotonic probability modeling with multiple covariates. *Computing Science and Statistics* **27**, 500-505.
- Francis, R. L., and J. A. White. 1976. *Facility Location and Layout*. Addison-Wesley, Reading, MA.
- Gebhardt, F. 1970. An algorithm for monotone regression with one or more independent variables. *Biometrika* **57**, 263-271.
- Goldstein, A. J., and J. B. Kruskal. 1976. Least-square fitting by monotonic functions having integer values. *Journal of American Statistical Association* **71**, 370-373.
- Kaufman, Y., and A. Tamir. 1993. Locating service centers with precedence constraints. *Discrete Applied Mathematics* **47**, 2351-261.
- Lee, C. I. C. 1983. The min-max algorithm and isotonic regression. *Annals of Statistics* **11**, 467-477.
- Liu, M. H., and V. A. Ubhaya. 1997. Integer isotone optimization. *SIAM Journal on Optimization* **7**, 1152-1159.
- Maxwell, W. L., and J. A. Muchstadt. 1983. Establishing consistent and realistic reorder intervals in production-distribution systems. *Operations Research* **33**, 1316-1341.
- Menendez, J. A., and B. Salvador. 1987. An algorithm for isotonic median regression. *Computational Statistics and Data Analysis* **5**, 399-406.
- Pardalos, P. M., G. L. Xue, and Y. Li. 1995. Efficient computation of isotonic median regression. *Applied Mathematics Letters* **8**, 67-70.
- Pardalos, P. M., and G. L. Xue. 1998. Algorithms for a class of isotonic regression problems. To appear in *Algorithmica*.
- Restrepo, A., and A. C. Bovik. 1994. Locally monotonic regression. *IEEE Transactions on Signal Processing* **41**, 2796-2780.
- Robertson, T., and P. Waltman. 1968. On estimating monotone parameters. *Annals of Mathematical Statistics* **39**, 1030-1039.
- Robertson, T., and F. T. Wright. 1973. Multiple isotonic median regression. *Annals of Statistics* **1**, 422-432.
- Robertson, T., and F. T. Wright. 1980. Algorithms in order restricted statistical inference and the Cauchy mean property value. *Annals of Statistics* **8**, 645-651.
- Robertson, T., F. T. Wright, and R. L. Dykstra. 1988. *Order Restricted Statistical Inference*. John Wiley & Sons, New York.

- Roundy, R. 1986. A 98% effective lot-sizing rule for a multi-product multistage production/inventory system. *Mathematics of Operations Research* **11**, 699-727.
- Schell, M. J., and B. Singh. 1997. The reduced monotonic regression method. *Journal of the American Statistical Association* **92**, 128-135.
- Shi, N. -Z. 1995. The minimal L_1 isotonic regression. *Communications in Statistics* **24**, 175.
- Stromberg, U. 1991. An algorithm for isotonic regression with arbitrary convex distance function. *Computational Statistics and Data Analysis* **11**, 205-219.
- Tamir, A. 1993. The least element property of center location on tree networks with applications to distance and precedence constrained problems. *Mathematical Programming* **62**, 475-496.
- Ubhaya, V. A. 1974a. Isotone optimization, I. *Journal of Approximation Theory* **12**, 146-159.
- Ubhaya, V. A. 1974b. Isotone optimization, II. *Journal of Approximation Theory* **12**, 315-342.
- Ubhaya, V. 1979. An $O(n)$ algorithm for discrete n -point convex approximation with application to continuous case. *Journal of Mathematical Analysis and Applications* **72**, 338-354.
- Ubhaya, V. 1987. An $O(n)$ algorithm for least squares quasi-convex approximation. *Computational Mathematics and Applications* **14**, 583-590.