Solving a nonlinear integer program for allocating resources

Y.H. Liu\textsuperscript{a,*}, J.M. Chuang\textsuperscript{b}, Ming-Jiu Hwang\textsuperscript{c}

\textsuperscript{a} Department of Marketing and Distribution Management, Hsuan Chuang University, Taiwan
\textsuperscript{b} Department of Mechanical Engineering, Dalhousie University, Canada
\textsuperscript{c} Library, National Chiao Tung University, Taiwan

Received 9 November 2005; accepted 18 January 2006

Abstract

An algorithm for solving a nonlinear integer program for resource allocation is proposed. This algorithm is efficient and relatively simple.

Keywords: Nonlinear integer program; Resource allocation; Mathematical induction

1. Introduction

The nonlinear integer program has been studied extensively by many researchers (see Refs [1–5], for example). In this paper, the nonlinear integer program (RIP) as shown below is solved by using a proposed novel and relatively simple algorithm:

\[
\min f(x_1, x_2, \ldots, x_N) = \sum_{j=1}^{N} \frac{a_j}{x_j}
\]

(RIP) s.t. \( \sum_{j=1}^{N} x_j \leq K \)

\( x_j \geq 1 \)

\( x_j \) is an integer, \( a_j > 0 \).

The model (RIP) is obtained from the integer program (HIP) below. It is a special form of resource allocation problem (a human resource distribution problem).

\[
\max \sum_{j=1}^{N} P_j \left( T_j - \frac{a_j}{x_j} \right)
\]

* Corresponding author.
E-mail addresses: dr-liu@hotmail.com (Y.H. Liu), Jimmy.Chuang@Dal.Ca (J.M. Chuang), mjhwang@hcu.edu.tw (M.-J. Hwang).

0895-7177/S - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2006.01.023
(HIP) s.t. $\sum_{j=1}^{N} x_j \leq K$

$$x_j \geq 1, \quad x_j \text{ is a positive integer}$$ (2)

where $T_j, a_j$ and $P_j$ are given positive real numbers.

To solve (HIP), one can solve (RIP) first, and the solution of (HIP) follows immediately, since (RIP), in fact, can be obtained from (HIP) by some simple substitutions.

In this paper, Section 2 introduces the main theorem which is the theoretical foundation of our algorithm. The algorithm of a computational procedure is described in Section 3. Section 4 gives a simple example. The last section, Section 5, gives a conclusion and remarks.

2. Main theorem

In this section, we will prove the main theorem. Based upon this theorem, an algorithm will be developed for constructing a sequence of feasible solutions iteratively. The algorithm starts with a feasible solution $x_j = 1$, for all $j = 1, 2, \ldots, N$ and the algorithm terminates when an optimal solution is reached.

Let the feasible region of (HIP) be

$$X = \left\{(x_1, x_2, \ldots, x_N) : \sum_{j=1}^{N} x_j \leq K, x_j \geq 1 \forall j = 1, 2, \ldots, N\right\}.$$ (3)

Also let

$$X_p = \left\{(x_1, x_2, \ldots, x_N) : \sum_{j=1}^{N} x_j = N + p, x_j \geq 1 \right\}, \quad p = 0, 1, \ldots, L \quad \text{where } L = K - N.$$ (4)

Clearly, $\{X_p : p = 0, 1, 2, \ldots, L\}$ is a partition for $X$.

For $p = 0, 1, 2, \ldots, L$, we let

$$x(p) = (x(p, 1), x(p, 2), \ldots, x(p, N)) \in X_p \quad \text{(5)}$$

$$d(p) = (d(p, 1), d(p, 2), \ldots, d(p, N)) \quad \text{(6)}$$

where

$$d(p, j) = \begin{cases} 1, & \text{if } j = M, \text{ for exactly one given } M \in \{1, 2, \ldots, N\} \\ 0, & \text{if } j \neq M, \text{ for all } j \in \{1, 2, \ldots, N\}. \end{cases} \quad \text{(7)}$$

Then, when $p = 0$, all components of $x(p)$ are 1, i.e.

$$x(p) = (1, 1, \ldots, 1)$$ (8)

and

$$x(p + 1) = x(p) + d(p), \quad \text{for all } p = 0, 1, 2, \ldots, L.$$ (9)

Consequently,

$$f(x(p)) - f(x(p + 1)) = \frac{a_M}{x(p, M)(x(p, M) + 1)} > 0, \quad \text{for all } 0 \leq p \leq L, \quad x(p) \in X_p.$$ (10)

Thus, $\{f(x(p))\}$ is a decreasing sequence of $p$, for $0 \leq p \leq L$.

In addition, if $M$ is chosen so that

$$f(\tilde{x}(p)) - f(\tilde{x}(p + 1)) = \frac{a_M}{x(p, M)(x(p, M) + 1)} = \max_{1 \leq j \leq N} \frac{d_j}{x(p, j)(x(p, j) + 1)}.$$ (11)

Then, we have the following Lemma, and the Theorem:
For all 1 ≤ j ≤ N, suppose
\[
\frac{a_M}{\bar{x}(p, M)(\bar{x}(p, M) + 1)} < \frac{a_j}{x(p, j)(x(p, j) + 1)}
\]
then there exists a q such that y(k, q) = x(k + 1, q) and for all j ≠ q, y_j = x_j. Also by the Lemma, it follows that
\[
\frac{a_M}{\bar{x}(k, M)(\bar{x}(k, M) + 1)} = \max_{1 ≤ j ≤ N} \frac{a_j}{x(k, q)(x(k, q) + 1)}.
\]

**Proof.** For all 1 ≤ j ≤ N, suppose
\[
\frac{a_M}{\bar{x}(p, M)(\bar{x}(p, M) + 1)} < \frac{a_j}{x(p, j)(x(p, j) + 1)}
\]
then
\[
\frac{a_j}{\bar{x}(p, j)(\bar{x}(p, j) + 1)} ≤ \frac{a_M}{\bar{x}(p, M)(\bar{x}(p, M) + 1)} < \frac{a_j}{x(p, j)(x(p, j) + 1)}
\]
i.e.
\[
\frac{a_j}{\bar{x}(p, j)(\bar{x}(p, j) + 1)} < \frac{a_j}{x(p, j)(x(p, j) + 1)}
\]
and, consequently, we have
\[
\bar{x}(p, j)(\bar{x}(p, j) + 1) > x(p, j)(x(p, j) + 1).
\]
Since \(\bar{x}(p, j), (\bar{x}(p, j) + 1)\) and \(x(p, j), (x(p, j) + 1)\) are two pairs of consecutive integers,
\[
\bar{x}(p, j) > x(p, j) \quad \text{for all } j.
\]
This contradicts the fact that
\[
\sum_{j=1}^{N} \bar{x}(p, j) = \sum_{j=1}^{N} x(p, j) = N + p.
\]
Therefore, there exists a q such that y(k, q) = x(k + 1, q) and for all j ≠ q, y_j = x_j. Also by the Lemma, it follows that
\[
\frac{a_M}{\bar{x}(k, M)(\bar{x}(k, M) + 1)} ≥ \frac{a_q}{x(k, q)(x(k, q) + 1)}.
\]

2.2. **Theorem**

Let the sequence \{\(\bar{x}(p)\in X_p : p = 0, 1, 2, \ldots, L\) and \(d(p)\) be defined as in (5)–(9). Then
\[
f(\bar{x}(p)) ≤ f(x) \quad \text{for all } x \in X_p \quad \text{and for all } p = 0, 1, 2, \ldots, L.
\]

**Proof.** An induction proof is given as follows:

1. When \(p = 0, X_p = \{(1, 1, \ldots, 1)\}, it is trivial.
2. Assuming \( f(x(k)) \leq f(x) \) for all \( x \in X_k \), we have to prove that \( f(\bar{x}(k + 1)) \leq f(x) \) for all \( x \in X_{k+1} \). To show this, we know that for all \( x(k + 1) = (x(k + 1, 1), x(k + 1, 2), \ldots, x(k + 1, N)) \in X_{k+1} \), there are \( y(k) = (y(k, 1), y(k, 2), \ldots, y(k, N)) \in X_k \) and a \( q, 1 \leq q \leq N \), such that \( y_q + 1 = x_q \) and for all \( j \neq q \), \( y_j = x_j \). Also by the Lemma, it follows that

\[
\frac{a_M}{\bar{x}(k, M)(\bar{x}(k, M) + 1)} \geq \frac{a_q}{x(x, q)(x(k, q) + 1)}.
\]

Therefore, \( \forall x \in X_{k+1} \), it can be seen that

\[
f(x(k + 1)) = \sum_{j \neq q} \frac{a_j}{x(k + 1, j)} + \frac{a_q}{y(k, q) + 1} \\
\geq f(y(k)) - \frac{a_q}{y(k, q)} + \frac{a_q}{y(k, q) + 1} \\
\geq f(\bar{x}(k)) - \frac{a_q}{y(k, q)(y(k, q) + 1)} \\
\geq f(\bar{x}(k)) - \frac{a_M}{\bar{x}(k, M)(\bar{x}(k, M) + 1)} \\
= f(\bar{x}(k + 1)).
\]

This completes the proof of the theorem.

Moreover, the Main Theorem below can be seen immediately.

**Main Theorem.** \( \bar{x}(L), L = K - N \), is an optimal solution of the integer program (RIP).

3. **Algorithm**

A procedure for the algorithm for the computation is described below:

1. Let \( x_j = 1 \), for \( j = 1, 2, \ldots, N \).
2. Compute \( t_j \) for \( j = 1, 2, \ldots, N \), where

\[
t_j = \frac{a_j}{x_j(x_j + 1)}.
\]
3. Find the corresponding \( j \), say \( j_{max} \), of \( \max\{t_j, j = 1, 2, \ldots, N\} \). Notice that if \( j_{max} \) is multiple, the first one is chosen.
4. Increase \( x_{j_{max}} \) by 1, i.e. \( x_{j_{max}} = x_{j_{max}} + 1 \).
5. Repeat the process (from item 2 to item 4), until

\[
\sum_{j=1}^{N} x_j = K.
\]
6. Calculate \( f(x_1, x_2, \ldots, x_N) \), which is

\[
f(x_1, x_2, \ldots, x_N) = \sum_{j=1}^{N} \frac{a_j}{x_j} = \min_{x \in X} f(x_1, x_2, \ldots, x_N).
\]

4. **Example**

4.1. **Computational Example**

Let

\[
\min f(x_1, x_2, \ldots, x_4) = \frac{20}{x_1} + \frac{18}{x_2} + \frac{100}{x_3} + \frac{30}{x_4}
\]

s.t. \( x_1 + x_2 + x_3 + x_4 \leq 15 \)
\( x_1, x_2, x_3, x_4 > 0 \).
Table 1

<table>
<thead>
<tr>
<th>( t_{ij} )</th>
<th>( t_{i1} )</th>
<th>( t_{i2} )</th>
<th>( t_{i3} )</th>
<th>( t_{i4} )</th>
<th>( t_{imax} )</th>
<th>( x_{i1} )</th>
<th>( x_{i2} )</th>
<th>( x_{i3} )</th>
<th>( x_{i4} )</th>
<th>( \sum_j x_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{1j} )</td>
<td>10/1 = 10</td>
<td>10/1 = 10</td>
<td>20/1 = 20</td>
<td>30/1 = 30</td>
<td>30</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1 + 1 = 2</td>
</tr>
<tr>
<td>( t_{2j} )</td>
<td>10/1 = 10</td>
<td>10/1 = 10</td>
<td>20/1 = 20</td>
<td>30/2 = 15</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1 + 1 = 2</td>
</tr>
<tr>
<td>( t_{3j} )</td>
<td>10/1 = 10</td>
<td>10/1 = 10</td>
<td>20/2 = 10</td>
<td>30/2 = 15</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2 + 1 = 3</td>
</tr>
<tr>
<td>( t_{4j} )</td>
<td>10/1 = 10</td>
<td>10/1 = 10</td>
<td>20/2 = 10</td>
<td>30/3 = 10</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( t_{5j} )</td>
<td>10/2 = 5</td>
<td>10/1 = 10</td>
<td>20/10 = 10</td>
<td>30/3 = 10</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>1 + 1 = 2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>( t_{6j} )</td>
<td>10/2 = 5</td>
<td>10/2 = 5</td>
<td>20/10 = 10</td>
<td>30/3 = 10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2 + 1 = 3</td>
</tr>
<tr>
<td>( t_{7j} )</td>
<td>10/2 = 5</td>
<td>10/2 = 5</td>
<td>20/3 = 6.667</td>
<td>30/3 = 10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Using the algorithm developed in Section 3, the solution of this problem can be obtained easily in a few mathematical operations:

1. The minimum is 39.333; and
2. \( x_1 = 3, x_2 = 3, x_3 = 6, x_4 = 3 \).

4.2. Example of hand calculation

\[
\min f(x_1, x_2, \ldots, x_4) = \frac{10}{x_1} + \frac{10}{x_2} + \frac{20}{x_3} + \frac{30}{x_4}
\]

s.t. \( x_1 + x_2 + x_3 + x_4 \leq 10 \)

\( x_1, x_2, x_3, x_4 \geq 1 \).

This problem can be easily solved by hand calculation with Table 1.

Consequently, \( x_1 = x_2 = 2, x_3 = x_4 = 3 \) and \( \min f(x_1, x_2, x_3, x_4) = 26.6667 \).

5. Conclusion and remarks

A relatively new theorem and algorithm for solving a nonlinear integer program for resource allocation is proposed in this research. It is very simple and efficient in comparison to the traditional approach. More specifically, for solving a nonlinear integer problem for allocating resources, there are only \( K - N \) iterations in which there are \( N \) divisions and \( K - N \) comparisons for finding the maximum involved.

References