Many algorithms for discrete problems use a variation of the tree-search enumeration technique as a basis for the algorithm. If a solution is the assignment of an attribute from a set of \( m \) attributes to every variable in a set of \( n \) variables, then redundant solutions can be generated if either the attributes or the variables contain some indistinguishable elements. A series of necessary and sufficient techniques are developed to eliminate the production of redundant solutions during enumeration. These techniques can be used to form the foundation of any partial enumeration algorithm where redundant solutions can be produced.

**Key words:** Partial Enumeration, Redundant Solutions, Branch and Bound, Implicit Enumeration.

1. Introduction

Many algorithms for discrete problems use a variation of the tree-search enumeration technique as a basis for the algorithm. The efficiency of these algorithms is greatly increased by techniques designed to show which branches of the tree will not yield a better solution and thus eliminate these potential solutions from further consideration. In this way all potential solutions are implicitly considered. This general process has been given many names by different authors, such as combinatorial programming [9], branch and bound [8], backtrack programming [6], implicit enumeration [5], and subductive programming [1].

In some problems for which partial enumeration algorithms have been devised, each potential solution has been unique. However, for many problems like the chromatic number problem [2], the chromatic reduction problem [3], and the loading problem [4], not all potential solutions are unique and thus some solutions are redundant. This paper will develop techniques to eliminate redundant solutions in partial enumeration algorithms.

Section 2 will develop theorems for preventing redundant solutions and Section 3 will convert the results in the theorems to a set of procedures. Section 3 also contains an example.

Theorem 1 has also appeared in Brown [2] and [3] as a basis for algorithms for the chromatic number and chromatic reduction problems. The loading problem
[4] is to assign a set of discrete objects, each having a weight, to a set of boxes, each of which has a capacity limit, in such a way that every object is assigned to a box and the number of boxes used is minimized. The results in this paper were used as the basis for a loading problem algorithm by Hung and Brown [7].

Before we can precisely define redundant solutions, we must define a finite set $X$ composed of the $n$ numbered variables $x_i$, $X = \{x_1, x_2, \ldots, x_n\}$, and a finite set $A$ composed of the $m$ numbered attributes $a_i$, $A = \{a_1, a_2, \ldots, a_m\}$. A solution is the assignment of one attribute to every variable $x_i$ in the set $X$. For the loading problem, the variables $x_i$ would represent the objects and the attributes $a_i$ would represent the boxes. For the chromatic number problem [5], the variables would represent the nodes of the graph and the attributes $a_i$ would represent the colors. For the 0-1 linear programming problem, only two attributes would be allowed, 0 and 1.

Consider the problem with $X = \{x_1, x_2, x_3\}$ and $A = \{a_1, a_2\}$, then the eight solutions are

<table>
<thead>
<tr>
<th>Variables</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$a_1$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$a_1$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$a_1$</td>
</tr>
</tbody>
</table>

In some problems, $a_1$ and $a_2$ are merely a means to separate the variables into two groups and have no intrinsic or different meaning. The loading problem and the chromatic number problem are two problems of this type. In this case, four of the solutions are redundant because they can be obtained by interchanging $a_1$ and $a_2$ in the other four solutions. One out of each of the following solution pairs is redundant.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$a_1$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$a_1$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$a_1$</td>
</tr>
</tbody>
</table>

In this case, $a_1$ and $a_2$ are called indistinguishable attributes.

Redundant solutions also may be produced by indistinguishable variables as in the loading problem where two or more objects have the same magnitude. Let $A = \{a_1, a_2\}$ with $a_1$ and $a_2$ distinct and $X = \{x_1, x_2, x_3\}$ with $x_1$, $x_2$, and $x_3$ indistinguishable. Then two out of each of the two following solution triplets is redundant.