

## APPENDIX B

### DYNAMIC PROGRAM ALGORITHM

The purpose of this appendix is to describe in more detail the solution of the dynamic programming problem given in Table 1. First, a general discussion of the solution of dynamic programming models is presented followed by documentation of a computer program developed to specifically solve the dynamic plant location problem (Figure B1).

#### The Solution of Dynamic Programming Problems

Unlike linear programming problems, which can always be solved using a specific algorithm, the solution of a dynamic programming problem is generally specific to the problem itself. No widely distributed software packages are available. The researcher must develop software for each problem encountered.

Dynamic programming refers to a problem solving approach which involves a sequence of interrelated decisions (Hillier and Lieberman, p. 266). The dynamic programming problem in this study is deterministic for which the computational solution is straightforward. A computer program is developed to allow quick solution to alternative specifications of the model.

#### Documentation

The computer program is written in FORTRAN (Figure B1). Comments are placed in the program to provide explanation of the specific steps required.

#### Overview

The number of periods to be considered ( $T$ ), discount factor (interest rate), initial plant configurations and the transition costs of opening new plants and closing old plants are given as data to the program. The best  $R_t$  solutions for each period  $t$ ,  $t = 1, \dots, T$ , have been previously determined. The program starts with the last two time periods and successively works backwards in time to determine the best path from each static solution in period 1 to period  $T$ . The transition cost of moving from the initial configuration to each of the period 1 static solutions is then calculated.

The output of the program includes the optimal path starting from each of the first period solutions and the total cost of that path. The user can compare these paths to find the best overall path.