$$(3')\quad S = \{(x_{ij}):\ y_i - y_j + nx_{ij} \le n-1 \quad \text{for } M+1 \le i \ne j \le n \quad \text{for some real numbers } y_i\}.$$

*Heuristic solution techniques for multi-depot VRP's.* Whereas the single depot VRP has been studied widely, the multi-depot problem has attracted less attention. The relevant literature is represented by only a few papers. However, a number of relatively successful heuristic approaches, which we will discuss in this section, have been developed for the multi-depot problem.

*The assignment— sweep approach.* This method, proposed by Gillett and Johnson[273], is an extension of the sweep heuristic that solves the multi-depot problem in two stages. First, locations are assigned to depots. Then, several single depot VRP's are solved using the sweep heuristic. Each stage is treated separately.

Initially, all locations are in an unassigned state.

For any node $i$, let

$t'(i)$ be the closest depot to $i$, and $t''(i)$ be the second closest depot to $i$.

For each node $i$, the ratio

$$r(i) = c_{i,t'(i)}/c_{i,t''(i)}$$

is computed and all nodes are ranked by increasing values of $r(i)$. The arrangement determines the order in which the nodes are assigned to a depot: those that are relatively close to a depot are considered first. After a certain number of nodes have been assigned from the list of $r(i)$, a small cluster is formed around every depot.

Locations $i$ such that the ratio $r(i)$ is close to 1, are assigned as follows. If two adjacent nodes $j$ and $k$ are already assigned to a depot, inserting $i$ between $j$ and $k$ on a route linked to $t$ creates an additional cost equal to $c_{ji} + c_{ik} - c_{jk}$ which represents a part of the total cost (or distance). In other words, the algorithm assigns location $i$ to a depot $t$ by inserting $i$ between two nodes already assigned to depot $t$, in the least costly manner.

The sweep algorithm is used to construct and sequence routes in the cluster about each depot independently. A number of refinements are then made to improve the resulting solution.

*A multi-depot savings approach.* This procedure, due to Tillman and Cain[648] starts with the initial solution consisting of servicing each node exclusively by one route from the closest depot. Let $c_{ij}$ = cost or distance between demand nodes $i$ and $j$; and $c_i^k$ = cost or distance between node $i$ and depot $k$. The total initial cost of all routes is then $D = \sum_{i=1}^{N} 2\min\{c_i^k\}$ where $N$ is the number of demand points. The method successively links pairs of nodes in order to decrease the total cost. One basic rule is assumed in the algorithm: the initial assignment of nodes to the nearest terminal is temporary, but once two or more nodes have been assigned to a common route from a terminal, the nodes are not reassigned to another terminal. In addition, as in the original savings algorithm, nodes $i$ and $j$ can be linked only if neither $i$ nor $j$ are interior to an existing tour.

At each step, the choice of linking a pair of nodes $i$ and $j$ on a route from terminal $k$ is made in terms of the savings when linking $i$ and $j$ at $k$. Nodes $i$ and $j$ can be linked only if no constraints are violated.

The computation of savings is not as straightforward as in the case of a single depot. The savings $s_{ij}^k$ are associated with every combination of demand nodes $i$ and $j$ and depot $k$, and represent the decrease in total cost or distance traveled when linking $i$ and $j$ at $k$. The formula is given by

$$s_{ij}^k = \tilde{c}_i^k + \tilde{c}_j^k - c_{ij}$$

where

$$\tilde{c}_i^k = \begin{cases} 2\min_t\{c_i^t\} - c_i^k & \text{if } i \text{ has not yet been given a permanent assignment} \\ c_i^k & \text{otherwise.} \end{cases}$$

In the first case, node $i$ is being reassigned from its closest depot and the previously assigned cost $2\min_{t}\{c_i^t\}$ is saved; in the second case, node $j$ is being inserted between depot $k$ and node $i$ and the link from $i$ to $k$ is dropped from the current solution.

The savings $s_{ij}^k$ are computed for $i, j = 1, \ldots, N (i \neq j)$ and $k = 1, \ldots, M$ at each step. They can be stored in $M$ matrices, each $N$ by $N$. At each step, the link $i - j$ at $k$ is chosen that maximizes $s_{ij}^k$ and that yields a feasible solution (with regard to capacity and other restrictions).

*A multi-depot savings algorithm for large problems.* The multi-depot VRP can be viewed as being solved in two stages: first, nodes must be allocated to depots; then routes must be built that link nodes assigned to the same depot. Ideally, it is more efficient to deal with the two steps simultaneously as in the multi-depot savings algorithm just discussed. When faced with larger problems, with say 1000 nodes, however, this method is no longer tractable computationally. A reasonable approach would be to divide the problem into as many subproblems as there are depots and to solve each subproblem separately.

The algorithm presented in this section attempts to implement this strategy while using the ratios $r(i)$ introduced in the discussion of the assignment-sweep algorithm. The approach is due to Golden *et al.*[306]. If a given node $i$ is much closer to one depot than any other depot, $i$ will be served from its closest depot. When node $i$ is equidistant from several depots, the assignment of $i$ to a depot becomes more difficult. For every node $i$, we determine the closest depot $k_1$ and the second closest depot $k_2$. If the ratios $r(i) = c_i^{k_1}/c_i^{k_2}$ is less than a certain chosen parameter $(0 \leq \delta \leq 1)$, we assign $i$ to $k_1$; in the case $r(i) \geq \delta$ we say that node $i$ is a border node.

Clearly, if $\delta = 0$, all nodes are declared border nodes and if $\delta = 1$, all nodes are assigned to their closest depot. For a given problem, we can fix the number of border nodes as we wish, by varying $\delta$.

The method proceeds as follows. In the first step, we ignore the nonborder nodes and the multi-depot savings algorithm is applied to the set of border nodes. The algorithm allocates the border nodes to depots and simultaneously builds segments of routes connecting these nodes. At the end of this first step, all nodes of our problem are assigned to some depot and all border nodes are linked on some routes. The solution to the VRP is produced depot by depot using single depot VRP techniques. The segments of routes that are built on border nodes are extended to the remaining nodes.

## 2.7. THE FLEET SIZE AND MIX PROBLEM

The goal of this section is to consider a situation in which the distributor decides, for the purposes of convenience and reliability, to *lease*, rather than *acquire*, vehicles for a certain period of time. The number of vehicles of each type needed to handle the system demand effectively must be determined together with a coherent routing policy for such vehicles. We assume that each vehicle type has a fixed leasing cost and a variable routing cost that is proportional to the distance traveled. The variable cost component encompasses the costs of fuel, maintenance, and manpower. This problem is formulated as a mathematical program by Golden *et al.*[296]. See Ball *et al.*[38] for a related study.

Golden *et al.*[296] also develop heuristics for solving this complex routing problem. We discuss some of these here. The most obvious approach for solving the fleet size and mix problem is to adapt the Clarke and Wright savings technique, discussed in Section 2.5. Remember that two distinct subtours containing customers $i$ and $j$ can be joined if: (a) $i$ and $j$ are not in the same subtour; (b) neither $i$ nor $j$ is an interior point in a subtour; and (c) the combined subtour containing both $i$ and $j$ has a total demand not in excess of vehicle capacity. The savings due to such a combination is $s_{ij} = c_{0i} + c_{0j} - c_{ij}$ and is computed for all $1 \leq i, j \leq n$ $(i \neq j)$. The Clarke and Wright procedure makes combinations in the order indicated by the savings (from largest to smallest), until no further feasible combinations exist.

This procedure is deficient for the fleet size and mix problem due to the fact that it ignores fixed vehicle costs. Since an infinite number of vehicles is assumed to be available, feasibility is certainly no problem. As a result, the savings method will tend to combine subtours until the capacity of the largest vehicle is reached. Unfortunately, a fleet composed almost entirely of largest vehicles may not be the least expensive. We denote this naive application of the Clarke and Wright approach to the fleet size and mix problem by CW.