2.4.4 *Classification of solution strategies in vehicle routing*

Most solution strategies for vehicle routing problems can be classified as one of the following approaches: (1) cluster first—route second; (2) route first—cluster second; (3) savings/insertion; (4) improvement/exchange; (5) mathematical-programming-based; (6) interactive optimization; (7) exact procedures. The first four approaches as well as the last one have been used extensively in the past. The other two approaches represent relatively recently developed ideas. A more general framework for heuristic algorithms is given by Ball and Magazine[39].

Cluster first-route second procedures group or cluster demand nodes and/or arcs first and then design economical routes over each cluster as a second step. Examples of this idea are given by Gillett and Miller[274], Gillett and Johnson[273], Chapleau *et al.*[120], and Karp[376] for the standard single depot vehicle routing problem.

Route first-cluster second procedures work in the reverse sequence. First, a large (usually infeasible) route or cycle is constructed which includes all of the demand entities (that is, nodes and/or arcs). Next, the large route is partitioned into a number of smaller, but feasible, routes. Golden *et al.*[296] provide an algorithm that typifies this approach for a heterogenous fleet size vehicle routing problem. Newton and Thomas[506] and Bodin and Berman[82] use this approach for routing school buses to and from a single school, and Bodin and Kursh[87, 88] utilize this approach for routing street sweepers. See also the work of Stern and Dror[623]. These applications are described in greater detail in Chapter 4.

Savings or *insertion* procedures build a solution in such a way that at each step of the procedure (up to and including the penultimate step) a current configuration that is possibly infeasible is compared with an alternative configuration that may also be infeasible. The alternative configuration is one that yields the largest savings in terms of some criterion function, such as total cost, or that inserts least expensively a demand entity not in the current configuration into the existing route or routes. The procedure eventually concludes with a feasible configuration. Examples of savings/insertion procedures for single depot node and arc routing problems are described by Clarke and Wright[145], Golden *et al.*[306], Norback and

Love[514], and Golden and Wong[310]. Hinson and Mulherkar[350] use a variation of this procedure for routing airplanes.

Improvement or *exchange* procedures such as the well-known branch exchange heuristic developed by Lin[444] and Lin and Kernighan[446] and extended by Christofides and Eilon[139] and Russell[572] always maintain feasibility and strive towards optimality. At each step, one feasible solution is altered to yield another feasible solution with a reduced overall cost. This procedure continues until no additional cost reductions are possible. Bodin and Sexton[94] modify this approach in order to schedule mini-buses for the subscriber dial-a-ride problem.

Mathematical programming approaches include algorithms that are directly based on a mathematical programming formulation of the underlying routing problem. An excellent example of mathematical-programming-based procedures is given by Fisher and Jaikumar[219]. They formulate the Dantzig-Ramser vehicle routing problem as a mathematical program in which two interrelated components are identified. One component is a traveling salesman (routing) problem and the other is a generalized assignment (packing) problem. Their heuristic attempts to take advantage of the fact that these two problems have been studied extensively and powerful mathematical programming approaches for their solution have already been devised. We discuss their heuristic in detail later. The algorithm due to Krolak and Nelson[404] is similar in spirit. Christofides *et al.*[143] and Stewart and Golden[628] discuss Lagrangian relaxation procedures for the routing of vehicles (see also Section 2.2). In addition, the article by Christofides, Mingozzi, and Toth[144]] represents a mathematical-programming-based (in particular, dynamic programming) approach for obtaining lower bounds in a variety of combinatorial optimization problems related to vehicles routing. Further discussion on this topic can be found in the article by Magnanti[455].

Interactive optimization is a general-purpose approach in which a high degree of human interaction is incorporated into the problem-solving process. The idea is that the experienced decision-maker should have the capability of setting the revising parameters and injecting subjective assessments based on knowledge and intuition into the optimization model. This almost always increases the likelihood that the model will eventually be implemented and used. Some early adaptations of this approach to vehicle routing problems are presented by Krolak, Felts and Marble[402] and Krolak, Felts and Nelson[403]. The paper by Cullen, Jarvis and Ratliff[156] introduces several rather novel interactive optimization heuristics.

Exact procedures for solving vehicle routing problems include specialized branch and bound, dynamic programming and cutting plane algorithms. Some of the more effective TSP approaches are described by Held and Karp[337, 338], Hansen and Krarup[327], Miliotis[480, 481], Balas and Christofides[29] and Crowder and Padberg[155]. Christofides *et al.*[143] discuss exact algorithms for the VRP. As noted before, any relaxation procedure that can provide good lower bounds on the optimal value of the vehicle routing problem can be embedded within a branch and bound approach to yield an exact procedure.

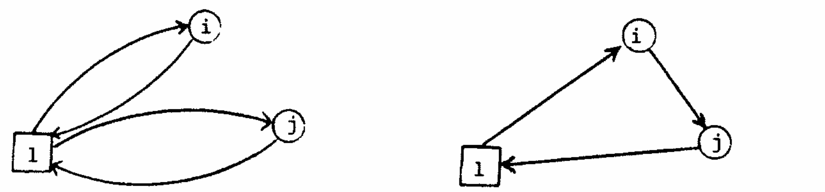
2.5. SELECTED SOLUTION TECHNIQUES FOR SINGLE DEPOT VRP's

As pointed out in the previous section, there are exact algorithms that solve the vehicle routing problem optimally by branch and bound techniques and there are heuristic algorithms that solve the problem approximately. Since the exact algorithms have been viable only for very small problems, we concentrate on heuristic algorithms. To our knowledge, the largest vehicle routing problem of any complexity that has been solved exactly involved about 25 customers[143]. We will now discuss several vehicle routing heuristic methods for a homogeneous fleet that have been used for problems up to 1000 customers.

The savings algorithm. The savings algorithm due to Clarke and Wright[111] is an "exchange" procedure in the sense that at each step one set of tours is exchanged for a better set. Initially, we suppose that every demand point is supplied individually by a separate vehicle (refer to Fig. 2.15 a).

Now, if instead of using two vehicles to service nodes i and j, we use only one, then we obtain a savings s_{ij} in travel distance of

$$(2c_{1i}+2c_{1j})-(c_{1i}+c_{1j}+c_{ij})=c_{1i}+c_{1j}-c_{ij}$$



(a) Initial Setup

(b) Nodes i and j have been linked

Fig. 2.15. Tour aggregation

(see Fig. 2.15 b). We note that if distances are asymmetric then $s_{ij} = c_{i1} + c_{1j} - c_{ij}$. For every possible pair of tour end points i and j there is a corresponding savings s_{ij}. We order these savings from largest to smallest and starting from the top of the list we link nodes i and j with maximum savings s_{ij} unless the problem constraints are violated.

The sweep approach. The sweep approach was originally devised by Gillett and Miller[274]. This approach, which is an efficient algorithm for problems of up to about 250 nodes, constructs a solution in two stages. First, it assigns nodes to vehicles and then it sequences the order in which each vehicle visits the nodes assigned to it. Rectangular coordinates for each demand point are required, from which we may calculate polar coordinates. We select a "seed" node randomly. With the central depot as the pivot, we start sweeping (clockwise or counterclockwise) the ray from the central depot to the seed. Demand nodes are added to a route as they are swept. If the polar coordinate indicating angle is ordered for the demand points from smallest to largest (with seed's angle 0) we enlarge routes as we increase the angle until capacity restricts us from enlarging a route by including an additional demand node (note that the procedure does not explicitly account for timing constraints); this demand point becomes the seed for the following route. Once we have the routes, we can apply TSP algorithms such as the 3-opt heuristic to improve tours. In addition, we can vary the seed and select the best solution. Figure 2.16 illustrates this procedure.

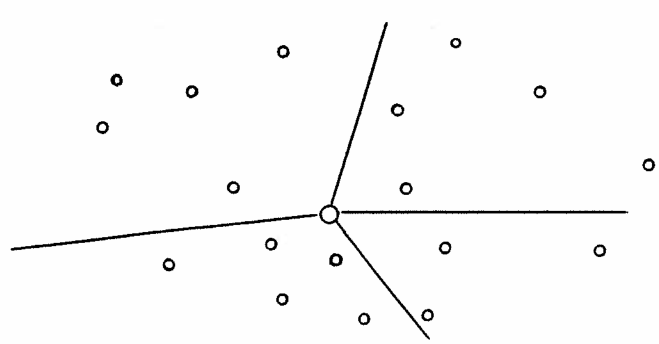


Fig. 2.16. Clusters generated by sweep procedure.