SEMI-GREEDY HEURISTICS: AN EMPIRICAL STUDY

J. Pirie HART

Department of Management Science, University of Washington, Seattle, WA 98195, USA

Andrew W. SHOGAN

School of Business Administration, University of California, 350 Barrows Hall, Berkely, CA 94720, USA

Received January 1986
Revised February 1987

Given p or c, a semi-greedy heuristic chooses each iteration's decision randomly from among those decisions resulting in objective value improvements either within p% of the best improvement or among the c best improvements. In the context of vehicle routing, we empirically compare the single use of a greedy heuristic with repeated use of a semi-greedy heuristic.

heuristic solution * vehicle routing

1. Introduction

The importance of heuristics is well established in the practice and theory of operations research. A heuristic frequently provides a suboptimal solution that is acceptable with respect to some criterion such as worst case error, asymptotic error, or average error in a well designed empirical test. Included among the ways to improve the performance of a heuristic are:

- *Use of several heuristics.* Instead of using just one heuristic, several different heuristics may be employed to generate serveral suboptimal solutions, with the best chosen for implementation. For example, in solving a vehicle routing problem, one could employ both the savings heuristic of Clarke and Wright [3] and the sweep heuristic of Gillett and Miller [4].
- *Use of several mathematical functions to govern the heuristic.* The decision made at each iteration of a heuristic often depends upon some mathematical function involving the problem's data. Instead of using just one function of the data, several different functions may be employed to generate several suboptimal solutions, with the best chosen for implementation. Balas and Ho [1] examined this approach for the set covering problem. The classic greedy heuristic for the set covering problem adds at iteration k the column that has the lowest ratio c_j/n_j, where c_j is the cost associated with the column j and n_j is the number of as yet uncovered rows that would be covered if column j were added to the current partial cover. Balas and Ho reported the results of empirical tests not only of a heuristic that used the ratio c_j/n_j but also of heuristics that used the ratios $c_j/(\log_2 n_j)$, $c_j/(n_j \log_2 n_j)$, and $c_j/(n_j \log n_j)$.
- *Reverting to a partial solution and reapplying the heuristic.* After applying a heuristic to obtain a suboptimal solution, it is sometimes possible to revert to a partial solution and then reapply the heuristic to obtain a different suboptimal solution. After this is done several times, the best of the suboptimal solutions can be implemented. Balas and Ho also examined this approach for the set covering problem. They first used one of the greedy heuristics described above to obtain a cover. Then they considered the columns in the order of their inclusion into the cover and removed from the cover all columns that covered at least one row that was covered more than once. Starting from the resulting partial cover, they then completed the cover using the greedy heuristic.
- *Perturbing the data and reapplying the heuristic.*

Instead of applying the heuristic to only the original data, the heuristic can also be applied to several minor perturbations of the data. The best of the suboptimal solutions obtained can then be implemented. To improve the performance of a heuristic for a vehicle routing problem, Beltrami and Bodin [2] examined the data perturbation obtained by increasing the distance between the depot and one or more locations. (There is no apparent need for restricting the distance perturbation to only increases or to changes in the distances involving the depot; Section 2 contains further discussion of this.)

- *Use of randomization within the heuristic.* Randomization can be used within a heuristic in several ways. One way is possible when the heuristic (e.g., the sweep heuristic of Gillet and Miller for a vehicle routing problem) has an arbitrary starting point or 'seed'. By randomly selecting the seed, several suboptimal solutions can be generated, with the best chosen for implementation. Golden [5] applied such a strategy to the travelling salesman problem. Another way to use randomization within a heuristic is simulated annealing, as illustrated by its application to the travelling salesman problem by Kirkpatrick, Gelatt and Vecchi [9]. They obtained a heuristic whose repeated application leads to a variety of suboptimal tours because an iteration may randomly decide to rearrange the tour in a manner resulting in a temporary increase in the tour's length. A final example of the use of randomization within a heuristic is one that is most similar to what we present in this paper. In the context of the capacitated Chinese postman problem, Golden, DeArmon and Baker [6] experimented with repeated applications of a heuristic that selected each iteration's course of action by randomly selecting from among the top three alternatives. They reported only minor improvement in the objective value.

In this paper, we present an approach whose use of randomization within a heuristic is similar to, but more general than that used by, Golden, DeArmon and Baker. The approach is an alternative to a greedy heuristic, that is, a heuristic that, at each iteration, chooses the decision resulting in the best improvement of the objective value during the current iteration. In contrast to a greedy heuristic, consider the following definitions of two types of what we call a semi-greedy heuristic:

- *A percentage-based semi-greedy heuristic with parameter p.* For a specified value of p, the decision at each iteration is chosen randomly from among those decisions resulting in objective value improvements within p% of the best possible improvement.
- *A cardinality-based semi-greedy heuristic with parameter c.* For a specified value of c, the decision at each iteration is chosen randomly from among those decisions resulting in objective value improvements among the c best improvements.

To illustrate, suppose that at some iteration of a heuristic, there exist seven alternative decisions that result in objective value improvements of 100, 98, 95, 91, 89, 73 and 58. Whereas a greedy heuristic automatically chooses the first alternative, a percentage-based semi-greedy heuristic with parameter $p = 10$ randomly chooses the decision from among the first four alternatives (since they are all within 10% of the best alternative), and a cardinality-based semi-greedy heuristic with parameter $c = 2$ randomly chooses the decision from among the first two alternatives. Observe that setting $p = 0$ and $c = 1$ results in a semi-greedy heuristic behaving like a greedy heuristic.

Consider the following strategy for obtaining a heuristic solution to a problem: Specify values for p (or c) and m, perform m repetitions of a percentage-based (or cardinality-based) semi-greedy heuristic with parameter p (or c), and implement the best of the suboptimal solutions obtained during the m repetitions.

Two questions arise with respect to the above strategy:

- For m fixed, what is the 'best' value for p (or c)?
- For p fixed, what is the trade-off as m increases between the 'benefit' obtained by an improved objective value versus the 'cost' of increased computational effort?

Most of the remainder of this paper presents empirically-based answers to these questions in the context of the capacitated vehicle routing problem.

2. A semi-greedy heuristic for the capacitated vehicle routing problem

In this paper, we use the terminology capacitated vehicle routing problem to describe a problem with the following characteristics:

- There exist $N+1$ locations having indices 0, 1, 2,..., N.
- Location 0 is a depot that houses a fleet of homogeneous vehicles; in particular, each vehicle has the same capacity C (weight or volume).
- Location i demands a single vehicle to make a delivery of size a_i. (Alternatively, all demands may be pick-ups.)
- The distance between location i and location j is d_{ij}.
- A feasible vehicle route is a sequence of locations starting and ending at the depot for which the summation of a_i over the locations serviced by the route does not exceed C.
- The problem is to determine a set of feasible vehicle routes that minimizes the total distance travelled while servicing all demands.

A well-known and frequently used heuristic for the capacitated vehicle routing problem is the savings heuristic of Clarke and Wright (hereafter referred to as the C–W heuristic). The following is a summary of the steps of the C–W heuristic:

Initialization steps. Initialization consists of the following three steps:

(1) Begin with the N trivial routes obtained by having each location serviced by a different vehicle that simply makes a round trip between the depot and the location.

(2) For $i \neq 0$, $j \neq 0$ and $i \neq j$, compute

$$S_{ij} = d_{i0} + d_{0j} - d_{ij}.$$

S_{ij} is the savings that results if the route that ends with a vehicle travelling from location i to the depot is combined with the route that begins with a vehicle travelling from the depot to location j.

(3) Sort by non-ascending order the savings S_{ij}.

Iterative step. Suppose S_{ij} is the maximum savings remaining on the list of savings. If combining the two routes would result in the total size of the single route's deliveries exceeding the vehicle capacity, then do *not* combine the routes but delete S_{ij} from the savings list. Otherwise, combine the two routes and delete from the savings list the following: (a) S_{ik} for all k (since there no longer exists a route that ends with travel from location i to the depot), (b) S_{kj} for all j (since there no longer exists a route that begins with travel from the depot to location j), and (c) S_{k_1,k_2} where k_1 and k_2 are, respectively, the last and first locations on the combined route (since the route that ends with travel from location k_1 to the depot and the route that begins with travel from the depot to location k_2 is now one and the same route).

Termination rule. Terminate execution when the remaining entries on the savings list all have negative values or when the savings list is empty.

Golden, Magnanti and Nguyen [7] pointed out that a major flaw in the C–W heuristic is that once a location is assigned to a route, it cannot be subsequently reassigned to another route. At a particular iteration of the algorithm, there may be little difference among the entry at the top of the savings list and several of the immediately following entries. Despite the potentially small differences among the entries at or near the top of the savings list, the C–W heuristic greedily and irrevocably selects the maximum savings. To introduce variety into the C–W heuristic, Golden, Magnanti and Nguyen replaced the expression for savings with

$$S_{ij} = d_{i0} + d_{0j} - \alpha d_{ij},$$

where α is a so-called route shape parameter. Golden, Magnanti and Nguyen conducted empirical experiments in which, instead of using the C–W heuristic once, they repeated the modified C–W heuristic for several values of α in the interval [0,2] and selected the best. The variety introduced by using a route shape parameter resulted in less expensive routes. However, one drawback of this approach is that every time α is changed, the values of S_{ij} change and the savings list must be resorted. Since sorting time is the major component of the C–W heuristic's execution time, it would be desirable to introduce variety into the C–W heuristic without having to resort the savings list.

Another means of introducing variety into the C–W heuristic is through use of data perturbation, as suggested by Beltrami and Bodin. They restricted the data perturbation to increases in the distances between the depot and one or more locations. A generalization of this would be to obtain alternative realizations of the data by perturbing each distance d_{ij} by selecting it from a uniform distribution either on the interval $[(1-\epsilon)d_{ij}, (1+\epsilon)d_{ij}]$ or the interval $[d_{ij}-\epsilon, d_{ij}+\epsilon]$, where ϵ is a small constant. Like the route shape parameter approach of Golden, Magnanti and Nguyen, the data perturbation approach of Beltrami and Bodin, as well as its generalization, suffers from the drawback that every data perturbation requires a resorting of the savings list.

We can achieve the goal of not having to resort the savings list by using a semi-greedy heuristic. Once the savings have been computed (using $S_{ij} = d_{i0} + d_{0j} - d_{ij}$) and sorted, we can perform any desired number of repetitions of a semi-greedy heuristic without having to resort the savings list. Working with the identical sorted savings list, every repetition of a semi-greedy version of the C–W heuristic would at each iteration decide which two routes to combine by randomly selecting from among those savings within p% of the maximum savings (if using a percentage-based semi-greedy heuristic) or from among those savings among the c largest (if using a cardinality-based semi-greedy heuristic).

3. Notation

To describe the empirical experiments we conducted using semi-greedy heuristics, we need the following notation for quantities obtained when a percentage-based semi-greedy heuristic is applied to a specific set of data:

P: A set of alternative values for p, the semi-greedy heuristic's percentage parameter. It will always be the case that $0 \in P$, so that one of the alternative semi-greedy heuristics is always the greedy heuristic.

$V(m, p)$: The best objective value obtained after m repetitions of a percentage-based semi-greedy heuristic with parameter p, where $p \in P$.

$V(m, P)$: The minimum of $V(m, p)$ over $p \in P$.

$R(m, p)$: The percentage reduction achieved by the best objective value obtained after m repetitions of a percentage-based semi-greedy heuristic with parameter p in comparison with the objective value obtained from m repetitions of a greedy heuristic where ties are randomly broken. Formally, $R(m, p) = \{[V(m, 0) - V(m, p)]/|V(m, 0)|\} \times 100\%$, where $|\cdot|$ denotes absolute value. (This assumes minimization; the numerator's terms should be reversed if maximizing.)

$D(m, p)$: The percentage deviation of the best objective value obtained after m repetitions of the percentage-based semi-greedy heuristic with specific parameter p in comparison with the best objective value obtained after m repetitions of every percentage-based semi-greedy heuristic for every parameter $p \in P$. Formally, $D(m, p) = \{[V(m, p) - V(m, P)]/|V(m, P)|\} \times 100\%$. (This assumes minimization; the numerator's terms should be reversed if maximizing.)

$N(m, p)$: A 'counter' that equals 100% if parameter p results in the best objective value over all members of the set P and equals 0% otherwise. Formally, $N(m, p) = 100\%$ if $V(m, p) = V(m, P)$ and equals 0% otherwise.

The above notation for $R(m, p)$, $D(m, p)$ and $N(m, p)$ apply only to a single data set. To denote the means of these quantities over several data sets, we place a bar over the notation (e.g., $\bar{R}(\ )$, $\bar{D}(\ )$, and $\bar{N}(\ )$). Also, to obtain equivalent notation for a cardinality-based semi-greedy heuristic, we replace p, $p = 0$, and P everywhere by c, $c = 1$, and C, where C is a set of alternative values for the cardinality parameter.

4. Empirical results

To gain computational experience with a semi-greedy C–W heuristic, we generated random vehicle routing problems characterized by the following three attributes:

- *Number of locations.* The number of locations (excluding the depot) was either 10, 25, 50, 75 or 100. The locations were randomly placed on the integer coordinates of a 1000×1000 grid.
- *Choice of distance metric.* Distances between every pair of locations (including the depot) were computed using either the Euclidean metric or the rectangular metric.
- *Noise factor applied to distances.* The actual distances computed using the metric were dis-

torted by multiplying each distance by a distinct value selected from a uniform distribution on the interval $(1-\epsilon, 1+\epsilon)$, where ϵ is a 'noise' factor satisfying $0 \leqslant \epsilon < 1$. Observe that setting $\epsilon = 0$ produces a symmetric distance matrix and setting $\epsilon = 0.99$ is equivalent to stating that the distances have little relationship to the locations' grid-coordinates.

For any combination of the above three attributes, the service requirement for each location was selected using a uniform distribution over the integers in the interval [1, 999]. Also, a common vehicle capacity was selected using a uniform distribution over the integers in an interval $[(M+S)/2, S]$, where M and S respectively denote the maximum and the sum of the locations' service requirements.

Space limitations preclude a detailed summary of every combination of the above attributes we considered. Tables 1 and 2 display a representative sample of our empirical results. For a percentage-based semi-greedy C–W heuristic, Table 1 displays $\bar{R}(m, p)$, $\bar{D}(m, p)$, $\bar{N}(m, p)$ for $1 \leqslant m \leqslant 50$ and p an element of the set $P = \{0, 1, 2, \ldots, 8\}$, where the mean was taken over 50 data sets with the following attributes: (a) the number of locations was 50, (b) distances were computed using the Euclidean metric, and (c) each distance was distorted by using a noise factor of $\epsilon = 0.1$. Table 2 displays the analogous values obtained by applying to the identical data sets a cardinality-based semi-greedy C–W heuristic with c an element of the set $C = \{1, 2, 3, 4, 5\}$.

To illustrate the interpretation of Table 1, consider the triple in the row corresponding to $m = 50$ and the column corresponding to $p = 4\%$. The triple (5.0, 1.2, 24) indicates that 50 repetitions of a percentage-based semi-greedy C–W heuristic with $p = 4\%$ yielded the following results: (a) a 5% mean reduction in the objective value in comparison with the objective value obtained from the greedy C–W heuristic, (b) a 1.2% mean deviation from the best of the objective values obtained over all the semi-greedy C–W heuristics with $p = 0, 1, \ldots, 8$, (c) a 24% mean occurrence rate for the event that $p = 4\%$ produced the best of the objective values among those objective values obtained using the other eight values of p.

Examination of Table 1 leads to the following observations and conjectures about a percentage-based semi-greedy C–W heuristic:

(1) For $m = 1$, $\bar{R}(m, p)$ and $\bar{N}(m, p)$ attain maximums at $p = 0$, and $\bar{D}(m, p)$ attains a minimum at $p = 0$. Consequently, if only a single repetition is to be performed, the greedy C–W heuristic performs better than any percentage-based semi-greedy C–W heuristic.

(2) For the number of repetitions fixed at a

Table 1
Summary for a percentage-based semi-greedy heuristic (each triple of numbers represents, respectively, $\bar{R}(m, p)$, $\bar{D}(m, p)$ and $\bar{N}(m, p)$)

m	$p = 0\%$	$p = 1\%$	$p = 2\%$	$p = 3\%$	$p = 4\%$	$p = 5\%$	$p = 6\%$	$p = 7\%$	$p = 8\%$
1	(0.0, 3.0, 18)	(−0.6, 3.4, 10)	(−0.8, 3.7, 18)	(−1.4, 4.4, 18)	(−2.7, 5.7, 10)	(−2.1, 5.0, 16)	(−3.9, 7.0, 4)	(−5.3, 8.3, 4)	(−6.3, 9.4, 2)
2	(0.0, 3.3, 6)	(0.7, 3.0, 22)	(0.5, 3.2, 20)	(0.5, 3.2, 20)	(−0.4, 4.2, 14)	(−0.3, 4.2, 10)	(2.0, 5.8, 4)	(−3.0, 6.8, 4)	(−4.0, 7.8, 2)
3	(0.0, 4.0, 12)	(0.9, 3.0, 22)	(0.8, 3.0, 14)	(0.7, 3.1, 18)	(0.3, 3.6, 16)	(0.1, 3.8, 8)	(−1.4, 5.3, 4)	(−2.9, 6.9, 2)	(−2.6, 6.5, 4)
4	(0.0, 4.3, 10)	(1.5, 2.7, 20)	(1.8, 2.3, 16)	(1.6, 2.5, 18)	(1.1, 3.1, 16)	(0.8, 3.3, 12)	(−0.5, 4.7, 4)	(−2.0, 6.2, 0)	(−1.8, 6.1, 4)
5	(0.0, 4.8, 8)	(1.7, 2.9, 16)	(2.2, 2.3, 20)	(2.2, 2.4, 20)	(2.2, 2.4, 18)	(1.2, 3.4, 8)	(0.1, 4.6, 6)	(−1.4, 6.2, 2)	(−1.6, 6.4, 2)
6	(0.0, 4.9, 8)	(2.0, 2.8, 16)	(2.6, 2.1, 16)	(2.5, 2.2, 22)	(2.4, 2.3, 20)	(1.6, 3.2, 6)	(0.5, 4.2, 8)	(−0.7, 5.5, 2)	(−1.2, 6.0, 2)
7	(0.0, 5.1, 8)	(2.2, 2.6, 18)	(2.7, 2.1, 16)	(2.7, 2.2, 22)	(2.6, 2.3, 18)	(1.8, 3.1, 4)	(0.9, 4.0, 12)	(−0.3, 5.2, 2)	(−0.8, 5.8, 2)
8	(0.0, 5.2, 6)	(2.3, 2.6, 18)	(2.8, 2.1, 16)	(2.8, 2.1, 20)	(2.7, 2.2, 20)	(1.8, 3.2, 4)	(1.4, 3.6, 14)	(0.0, 5.1, 2)	(−0.4, 5.5, 2)
9	(0.0, 5.3, 6)	(2.4, 2.7, 16)	(3.0, 2.1, 18)	(3.1, 2.0, 20)	(2.8, 2.3, 20)	(2.0, 3.2, 4)	(1.5, 3.7, 12)	(0.2, 5.0, 4)	(−0.4, 5.6, 2)
10	(0.0, 5.4, 6)	(2.5, 2.6, 14)	(3.1, 2.0, 16)	(3.2, 1.9, 20)	(3.0, 2.1, 22)	(2.0, 3.1, 4)	(1.6, 3.6, 14)	(0.4, 4.8, 4)	(−0.3, 5.6, 2)
15	(0.0, 5.7, 6)	(2.7, 2.7, 10)	(3.6, 1.8, 20)	(3.7, 1.7, 18)	(3.8, 1.6, 22)	(2.7, 2.7, 8)	(2.3, 3.1, 14)	(1.2, 4.4, 4)	(0.2, 5.3, 2)
20	(0.0, 5.8, 6)	(2.9, 2.7, 12)	(3.7, 1.8, 12)	(3.9, 1.6, 20)	(4.0, 1.5, 26)	(3.0, 2.6, 8)	(2.7, 2.8, 14)	(1.8, 3.8, 4)	(0.7, 5.0, 2)
25	(0.0, 6.0, 4)	(3.0, 2.8, 14)	(4.0, 1.6, 8)	(4.2, 1.4, 22)	(4.3, 1.4, 28)	(3.4, 2.3, 10)	(3.0, 2.6, 10)	(2.2, 3.5, 4)	(1.0, 4.8, 2)
30	(0.0, 6.2, 2)	(3.1, 2.9, 12)	(4.3, 1.5, 14)	(4.5, 1.3, 20)	(4.5, 1.3, 30)	(3.6, 2.3, 10)	(3.2, 2.8, 8)	(2.3, 3.7, 4)	(1.6, 4.4, 2)
35	(0.0, 6.4, 0)	(3.1, 3.1, 6)	(4.4, 1.6, 10)	(4.7, 1.3, 26)	(4.7, 1.3, 30)	(3.8, 2.3, 12)	(3.5, 2.6, 14)	(2.4, 3.8, 2)	(2.0, 4.2, 0)
40	(0.0, 6.5, 0)	(3.2, 3.0, 4)	(4.5, 1.6, 8)	(4.8, 1.3, 30)	(4.8, 1.3, 26)	(4.2, 2.0, 18)	(3.7, 2.5, 14)	(2.8, 3.4, 2)	(2.0, 4.2, 0)
45	(0.0, 6.6, 0)	(3.2, 3.1, 2)	(4.6, 1.6, 8)	(4.8, 1.4, 28)	(5.0, 1.2, 28)	(4.2, 2.0, 16)	(3.9, 2.4, 16)	(2.9, 3.4, 4)	(2.1, 4.3, 0)
50	(0.0, 6.7, 0)	(3.2, 3.2, 2)	(4.7, 1.5, 12)	(4.9, 1.3, 28)	(5.0, 1.2, 24)	(4.4, 1.9, 16)	(4.0, 2.4, 16)	(3.1, 3.2, 4)	(2.4, 4.0, 0)

Table 2
Summary for a cardinality-based semi-greedy heuristic (each triple of numbers represents, respectively, $\bar{R}(m, c)$, $\bar{D}(m, c)$ and $\bar{N}(m,c)$)

m	$c=1$	$c=2$	$c=3$	$c=4$	$c=5$
1	(0.0, 0.9, 76)	(−4.1, 4.9, 20)	(−7.3, 8.2, 0)	(−12.0, 13.0, 4)	(−16.0, 17.0, 0)
2	(0.0, 1.6, 60)	(−1.8, 3.3, 28)	(−4.7, 6.3, 8)	(−10.2, 11.8, 4)	(−12.5, 14.2, 0)
3	(0.0, 1.8, 54)	(−0.8, 2.5, 32)	(−4.2, 5.9, 10)	(−8.5, 10.4, 4)	(−11.0, 12.9, 0)
4	(0.0, 2.2, 46)	(0.0, 2.1, 36)	(−3.2, 5.2, 12)	(−7.9, 10.1, 6)	(−10.2, 12.5, 0)
5	(0.0, 2.2, 44)	(0.3, 1.8, 36)	(−2.3, 4.4, 14)	(−7.3, 9.6, 6)	(−9.4, 11.6, 0)
6	(0.0, 2.2, 42)	(0.5, 1.6, 38)	(−2.0, 4.1, 16)	(−6.8, 9.1, 6)	(−8.8, 11.1, 0)
7	(0.0, 2.4, 40)	(0.8, 1.4, 42)	(−1.5, 3.8, 16)	(−6.4, 8.8, 4)	(−8.4, 10.9, 0)
8	(0.0, 2.5, 38)	(1.0, 1.4, 44)	(−1.1, 3.5, 14)	(−5.8, 8.3, 4)	(−8.0, 10.6, 0)
9	(0.0, 2.7, 34)	(1.4, 1.2, 50)	(−1.0, 3.5, 12)	(−5.3, 8.0, 4)	(−7.6, 10.3, 0)
10	(0.0, 2.8, 30)	(1.4, 1.2, 54)	(−0.8, 3.4, 12)	(−4.9, 7.7, 4)	(−7.4, 10.2, 0)
15	(0.0, 3.1, 26)	(2.2, 0.8, 60)	(−0.2, 3.2, 12)	(−3.6, 6.8, 2)	(−6.7, 9.9, 0)
20	(0.0, 3.3, 24)	(2.4, 0.6, 62)	(0.2, 3.0, 10)	(−3.2, 6.5, 4)	(−6.2, 9.6, 0)
25	(0.0, 3.5, 24)	(2.7, 0.6, 64)	(0.5, 2.9, 12)	(−2.5, 6.0, 0)	(−6.0, 9.6, 0)
30	(0.0, 3.6, 22)	(2.8, 0.5, 66)	(0.8, 2.6, 12)	(−2.4, 5.9, 0)	(−5.6, 9.2, 0)
35	(0.0, 3.7, 20)	(2.9, 0.5, 64)	(1.1, 2.4, 16)	(−2.2, 5.9, 0)	(−5.5, 9.2, 0)
40	(0.0, 3.8, 18)	(3.1, 0.5, 68)	(1.2, 2.5, 14)	(−2.2, 5.9, 0)	(−5.1, 9.0, 0)
45	(0.0, 3.9, 18)	(3.2, 0.5, 68)	(1.3, 2.4, 14)	(−2.0, 5.8, 0)	(−4.9, 8.8, 0)
50	(0.0, 4.0, 16)	(3.3, 0.4, 72)	(1.4, 2.3, 12)	(−1.9, 5.8, 0)	(−4.7, 8.7, 0)

value $m \geq 2$, $\bar{R}(m, p)$, $\bar{D}(m, p)$ and $\bar{N}(m, p)$ attain optimal values at a value of p greater than 0 but less than the maximum value in P. For example, for $m = 20$, the optimal value of p is 4%.

(3) As m increases, the optimal value of p increases. (There is sufficient evidence to conclude whether, as m increases, the optimal value of p continues to increase or approaches an asymptotic value.)

(4) For p held constant at a value $p > 0$, $\bar{R}(m, p)$ is an increasing function of m that increases at a decreasing rate.

Examination of Table 2 leads to similar observations and conjectures about a cardinality-based semi-greedy C–W heuristic. In particular, items (1) and (4) above remain valid with p replaced everywhere by c and $p > 0$ replaced by $c > 1$. Item (2) also remains valid, except that it takes about five repetitions (instead of two) until the greedy C–W heuristic performs worse than a semi-greedy C–W heuristic for at least one value of c. A significant contrast occurs with respect to item (3). For a percentage-based semi-greedy C–W heuristic, the optimal value of p increases from $p = 0$ for $m = 1$, to $p = 1$ for $2 \leq m \leq 3$, to $p = 2$ for $4 \leq m \leq 7$, to $p = 3$ for $8 \leq m \leq 11$, to $p = 4$ for $12 \leq m \leq 50$. In contrast, from Table 2, we see that, for a cardinality-based semi-greedy heuristic, $c = 2$ replaces $c = 1$ as optimal after about five iterations and remains optimal by a wide margin through $m = 50$. Thus, $c = 2$ appears to introduce just the right amount of variety into the C–W heuristic, whereas $c > 2$ introduces too much variety if only 50 repetitions will be performed.

The above observations and conjectures are based on a random sample of 50 data sets having the attributes of 50 locations, a Euclidean distance metric, and a distance noise factor of $\epsilon = 0.10$. The observations and conjectures are also supported by the tables similar to Tables 1 and 2 obtained when the number of locations was varied, the distance metric was changed to the rectangular metric, and the distance noise factor was varied. Empirical evidence (not reported here) suggests that varying one attribute (number of locations, distance metric, or distance noise factor) while keeping the others the same has the following effects:

- If m is fixed, the optimal value of p for a percentage-based semi-greedy C–W heuristic decreases as the number of locations increases. For example, when the Euclidean metric was used, the noise factor was $\epsilon = 0.10$, and the number of locations was 10, 25, 50, 75 and 100, the respective optimal values of p for $m = 50$ were $p = 13$, 5, 4, 2 and 1, respectively.

- The distance metric (Euclidean or rectangular) has no discernible effect on the empirical results.
- If m and p are fixed, $\bar{R}(m, p)$ and $\bar{D}(m, p)$ increase as the noise factor increases. For example, when the number of locations was 50, the Euclidean metric was used, and the noise factor ϵ was 0, 0.05, 0.1, 0.50 and 0.99, the respective values of $\bar{R}(50, 4)$ were 3.79, 4.22, 5.05, 6.94 and 10.83, and the respective values of $\bar{D}(50, 0)$ were 5.93, 5.85, 6.67, 9.87 and 15.41. Thus, as the distances between locations become less and less related to the locations' grid-coordinates, use of a semi-greedy C–W heuristic becomes more and more attractive as an alternative to a greedy C–W heuristic.

5. Concluding remarks

The question of whether to perform several repetitions of a semi-greedy C–W heuristic or a single repetition of the greedy C–W heuristic must be answered in the context of the trade-off between the benefit obtained by an improved objective value versus the cost of the increased computational effort. Our empirical evidence indicates that the increased computational effort is negligible, primarily because the savings S_{ij} do not require resorting at every repetition.

More specifically, our experiments were conducted in FORTRAN on an IBM PC-XT equipped with an 8087 numeric coprocessor. Sorting was performed using a quicksort routine outlined by Kernighan and Plauger [8], with an enhancement made to avoid the worst case running time. For a problem with 100 locations, the mean time (over 50 data sets) to sort the savings S_{ij} was 4.54 minutes, and the mean execution time for one repetition of a percentage-based semi-greedy heuristic was 0.20 minutes. Thus, whereas a single repetition of the greedy C–W heuristic requires an average of $4.54 + 0.20 = 4.74$ minutes, 50 repetitions of the semi-greedy C–W heuristic requires an average of $4.54 + (50)(0.20) = 14.54$ minutes. This investment of about 10 minutes of execution time was rewarded with a reduction of 4% in the objective value. Note that, if the savings S_{ij} must be resorted for every repetition, as is the case for the C–W heuristic modified by the use of the route shape parameter of Golden, Magnanti and Nguyen, then the mean execution time for a problem with 100 locations would be approximately 50 $(4.54 + 0.20) = 237$ minutes or almost four hours. Thus, it is significantly less time-consuming to introduce variety into the C–W heuristic by executing 50 repetitions of a semi-greedy C–W heuristic than executing the approach of Golden et al. with 50 different route shape parameters.

The question of whether one should employ a percentage-based semi-greedy C–W heuristic or a cardinality-based semi-greedy C–W heuristic has no definitive answer. For example, our empirical results (as illustrated by Tables 1 and 2) indicate that a fixed number of repetitions of a percentage-based semi-greedy C–W heuristic using the optimal value of p yields a better objective value than an equal number of repetitions of a cardinality-based C–W heuristic with the optimal value of c. However, whereas $c = 2$ was optimal for all experiments, p's optimal value depended on such things as the number of repetitions and the number of locations. Hence, it may be better to use a cardinality-based semi-greedy C–W heuristic with $c = 2$ than to use a percentage-based semi-greedy heuristic with a non-optimal value of p.

Unanswered questions include the following:

- Although there exists a semi-greedy version of any greedy heuristic, are Section 4's observations and conjectures supported by empirical evidence gathered from experiments with other types of problems? We have performed experiments using a semi-greedy variant of the 'greedy close' heuristic applied to the uncapacitated facility location problem. The empirical evidence supports Section 4's observations and conjectures. What about other types of problems (e.g., set covering)?
- Which of Section 4's observations and conjectures can be formally proved?
- In addition to being used as a 'stand-alone' technique, how useful will a semi-greedy heuristic be in the context of a branch-and-bound algorithm? For example, many branch-and-bound algorithms based on Lagrangian relaxation use a greedy heuristic to convert (when necessary) each optimal solution to be relaxed problem into a primal feasible solution, thereby obtaining an upper bound on the primal's objective value (when minimizing). With the use of a semi-greedy heuristic, it might be possible to quickly identify several upper bounds from

the same primal infeasible solution.

We intend to explore these and others issues in future research.

References

[1] E. Balas and A. Ho, "Set covering algorithms using cutting planes, heuristics, subgradient optimization: A computational study", *Mathematical Programming* **12**, 37–60 (1980).

[2] E.J. Beltrami and L.D. Bodin, "Network and vehicle routing for municipal waste collection", *Networks* **4**, 65–94 (1974).

[3] G. Clarke and J.W. Wright, "Scneduling of vehicles from a central depot to a number of delivery points", *Operations Research* **12**, 568–581 (1964).

[4] B. Gillett and L. Miller, "A heuristic algorithm for the vehicle dispatch problem", *Operations Research* **22**, 340–349 (1974).

[5] B.L. Golden, "A statistical approach to the TSP", *Networks* **7**, 209–225 (1977).

[6] B.L. Golden, J. DeArmon and E.K. Baker, "Computational experiments with algorithms for a class of routing problems", *Computers and Operations Research* **10**, 47–59 (1983).

[7] B.L. Golden, T.L. Magnanti and H.Q. Nguyen, "Implementing vehicle routing algorithms", *Networks* **7**, 113–148 (1977).

[8] B.W. Kernighan and P.J. Plauger, *Software Tools*, Addison-Wesley, Reading, MA, 1976.

[9] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, "Optimization by simulated annealing", *Science* **220**, 671–680 (1983).