Journal of the Operational Research Society (1999) 50, 44–51
http://www.stockton-press.co.uk/jor

Determination of lockbox collection points via simulated annealing

PR McMullen and RA Strong

University of Maine, USA

This research presents a heuristic to solve the lockbox location problem via a search-based technique known as simulated annealing. In the past, more traditional mathematical programming techniques have been used to address this problem, but with limited success due to its combinatorial nature. Because simulated annealing is a search-based technique, an optimal solution is not guaranteed, but past research has demonstrated that search-based heuristics can provide reasonable solutions without the difficulties associated with the more traditional formulations. In this paper, the simulated annealing methodology is used to solve a large lockbox location problem at several differing levels of cost. The results compare favourably to solutions obtained from a K-means clustering heuristic.

Keywords: simulated annealing; heuristics; lockbox; optimisation

Introduction and simulated annealing

Determination of cash collection points is an important component of accounts receivable management. These cash collection points are frequently referred to as lockboxes — a destination for customer remittances. Proper location selection for these lockboxes can help reduce travel time (frequently referred to as float). Reduction of this travel time in turn reduces the time required for the check to clear the bank, ultimately reducing the firm's opportunity costs.

The more lockboxes a firm has the more it will be able to reduce float because the remittances do not, typically, have as far to travel. On the other hand, while this variable cost is decreased, the fixed cost component is increased due to the increased number of lockboxes. It is also important that however many lockboxes are used they be positioned according to some type of policy that considers the population densities of the prospective lockbox locations.

Because of this tradeoff between variable and fixed costs as well as the travel time considerations previously mentioned, the lockbox problem is far from trivial. Levy[1] was one of the first to address this problem with a 'greedy' heuristic that was intended to find 'good' but not necessarily optimal solutions to the lockbox problem. Nauss and Markland[2,3] followed with approaches dedicated to finding optimal solutions to the lockbox location problem. Stone[4] has also offered a heuristic to solve the lockbox location problem that does not guarantee optimality but does consider computational efficiency. Feilitz and White[5,6] extended the work of both Stone and Nauss-Markland to obtain optimal solutions to the lockbox problem with reasonable computational efficiency. Maier and Vender Weide[7,8] also presented approaches to the lockbox location problem accompanied with general discussion of other solution approaches. Davis *et al*[9,10] addressed this general type of spatial optimisation problem for bank management by providing methodology for determining locations for check processing centers, implementation has shown the Davis *et al* methodology to be successful.

Regardless of which of these approaches is used for problems of this nature, a general problem presents itself when trying to find an optimal solution. This problem is that of computational efficiency. The lockbox location problem can be thought of as a combinatorial optimisation problem — an integer programming type of problem where there are many different combinations of feasible solutions. The presence of these many different combinations greatly complicates both the formulation of the model as well as its solution. These problems are referred to as nondeterministic-polynomial hard (NP-hard).

Recently, search-based heuristics (sometimes referred to as metaheuristics) have been successfully deployed to address these types of combinatorial optimisation problems. Simply put, these search-based heuristics examine many feasible solutions to the problem of interest and select the solution providing the most favorable objective function value.

Here, a search-based heuristic referred to as simulated annealing is used to address the lockbox problem. In the physical sense, annealing is the heating of a solid to a very high temperature and slowly cooling it. The objective of this is to have the solid attain some type of desirable property, such as strength or hardness. *Simulated* annealing is where several solutions to an optimisation problem are explored, with the central objective being to optimise some sort of objective function. Informative (and readable)

Correspondence: Dr PR McMullen, University of Maine, Maine Business School, 5723 Donald P.Corbett Business Building, Orono, Maine 04469-5723 USA.
E-mail: patmc@maine.maine.edu

descriptions of simulated annealing are available from both Eglese[11] and Kirkpatrick *et al.*[12]

This research presents a methodology to solve the lockbox problem via simulated annealing for a reasonably large-sized problem, the methodology is used to solve the problem for several levels of variable and fixed costs. These results are then compared to the results of a more informal heuristic solution to the same problem. Concluding remarks and general observations are then made.

Methodology

Prior to presentation of the actual details of the heuristic, some definitions are given:

Data:

N = number of regions serviced by the solution
S_i = annual sales in region serviced by lockbox i
S = total annual sales
P_i = population of region serviced by lockbox i
P = total population of all regions
$A(L_{ij})$ = land area of region j serviced by lockbox i
d_{ij} = travel distance from lockbox i to region j (in days)
Mileage (ij) = distance from lockbox i to region j (in miles)
r = interest rate
DOC = daily opportunity cost of remittances
FC = fixed cost for lockbox

Simulated annealing parameters:

T_I = initial temperature for simulated annealing
T_F = final temperature for simulated annealing
CR = cooling rate for simulated annealing
Iter = number of iterations for simulated annealing
s = iteration counter for simulated annealing
FIter = number of iterations for infeasible neighboring solutons
t = iteration counter for infeasible neighboring solutions

Decision variables:

R_i = Set of regions serviced by lockbox i
L_{ij} = the jth region serviced by lockbox i
x_{ij} = 1 if the jth region is serviced by lockbox i, 0 otherwise
n = number of lockboxes in the solution
m_i = number of regions serviced by lockbox i
E = total cost of objective function

The general formulation of this problem is as follows[13]:

$$\min: E = \sum_{i=1}^{n} \sum_{j=1, j\in R_i}^{m_i} [(S_i \cdot d_{ij} \cdot \mathrm{DOC})] + (n \cdot \mathrm{FC}) \quad (1)$$

subject to:

$$\sum_{i=1}^{n} \sum_{j=1, j\in R_i}^{m_i} x_{ij} = N \quad (2)$$

where

$$S_i = S \cdot (P_i/P) \quad (3)$$

$$\mathrm{DOC} = r/365 \quad (4)$$

The constraint simply implies that each region must be a member of exactly one lockbox.

Initialisation

An initial solution is generated where all regions needing to be serviced by the solution are lockboxes ($n = N$). This objective function value also becomes the objective function value for the current solution (E_C) and the best solution (E_B). It is worth noting that if a region accommodates a lockbox, that region is referred to as the host region for a lockbox. For example, if there is a lockbox in Texas that services Texas, Oklahoma and Arkansas, Texas is considered the host region for the lockbox. The travel distance in days for this particular region (serviced by lockbox i) is assumed to be the following:

$$d_{i1} = (0.5 \cdot \sqrt{A(L_{i1})/\pi})/100 \quad (5)$$

The subscript shows that the host region is the first region processed with respect to (1). For other regions that are not the host for the lockbox and are serviced by lockbox i, the distance in days is as follows:

$$d_{ij} = \mathrm{mileage}(L_{i1}, L_{ij})/100, \qquad \text{for all } j \in R_i \quad (6)$$

Both (5) and (6) have denominators of 100 here to show the assumption that one day is required for a remittance to travel 100 miles. For the initial solution here, each region serves as a host for a lockbox, hence, $n = N$. Parameters specific to simulated annealing are also selected (T_I, T_F, Iter and CR), and the iteration counter is also initialised.

Search sequence

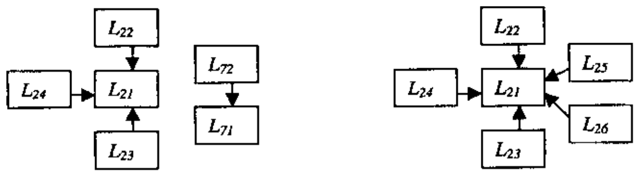
The search procedure commences by determining what type of neighboring solution is generated. Here, there are two general types of neighboring solutions — solutions involving compressions and solutions involving transfers, compressions are when two individual lockboxes serving single regions each, are fused into one. Transfers are when a single region from within a lockbox system becomes a new, unique lockbox, the probability of compression *P(C)* is determined by the following relationship:

$$P(C) = n/N \quad (7)$$

A random number is then generated (Rand). If Rand < *P(C)*, compression is performed, otherwise a transfer is performed, when compression is performed, a lockbox is randomly selected and is referred to as a host lockbox. The lockbox nearest to the host lockbox is then found

(a) before compression (b) after compression

Figure 1 Before and After Compression.



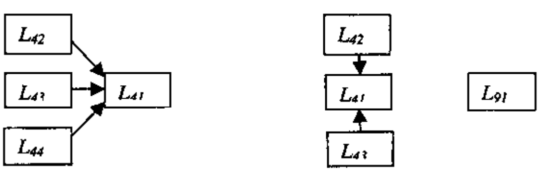
(a) before hyper-compression (b) after hyper-compression

Figure 3 Example of Hyper-Compression.

(referred to as the target region), compression is then performed by having only that region serviced by the host lockbox, the number of lockboxes (n) is then decreased by 1. Figure 1 shows an example of two lockboxes prior to and after compression. Notice in (a) that lockbox 6 (L_{61}) and lockbox 3 (L_{31}) are different entities altogether. After the compression (b), lockbox L_{61} is renamed L_{32} to indicate that it is now serviced by L_{31}.

If Rand $\geqslant P(C)$, a transfer is performed, with a transfer, a lockbox servicing two or more regions is randomly selected. The region in this lockbox system that is furthest from the host region (in miles), is removed and becomes a stand-alone lockbox servicing only the region, the number of lockboxes (n) is increased by 1 as the result of the transfer. Figure 2 shows an example of before and after a transfer. Here, lockbox 4 is randomly selected and the region being served by the lockbox (L_{44}) is transferred so that it becomes a lockbox serving only the region in which it resides.

There are occasions when neither compression nor a transfer will be possible. For example, compression will not be feasible when the randomly selected host lockbox also services other regions. A transfer will not be possible when all lockbox systems service single regions. The heuristic keeps count of the number of consecutive times when neither compression nor transfer was feasible (t). When the value of t equals FIter, an action referred to as hyper-compression is attempted. Hyper-compression is simply the merging together of two lockbox systems where each lockbox system has two or more regions served by the host region. Hyper-compression works by randomly selecting the host lockbox that will also serve as the host for the new lockbox. The target lockbox for hyper-compression is then selected as the lockbox whose host region is nearest (in miles) to the recently selected host. All regions serviced by the target lockbox then become regions serviced by the newly created, larger lockbox system in addition to the units



(a) before transfer (b) after transfer

Figure 2 Before and After Transfer.

already serviced by the existing host lockbox. Figure 3 shows an example of before and after hyper-compression. In this example, lockbox 2 is randomly selected as the host lockbox for hyper-compression. Lockbox 7 is then determined to be the lockbox nearest lockbox 2. As a result, hyper-compression places the regions originally serviced by lockbox 7 into lockbox 2 and these regions are then serviced by lockbox 2 (L_{71} and L_{72} become L_{25} and L_{26} respectively).

Hyper-compression is performed when, after several attempts, neither regular compression nor transfers are feasible. This scenario can be thought of as stagnation — new solutions are not being found. Hyper-compression can generate a new, more aggressive solution when compared to the other two types of maneuvers, thereby placing the new solution at a different location on the continuum of feasibility and hopefully a more desirable location on this feasibility continuum, once hyper-compression is performed the value of t should be adjusted to zero.

Simulated annealing

The solutions obtained from the search strategies previously described will be considered 'test solutions.' These solutions will be evaluated by the objective function in (1) and will have the objective function value of E_t. Once a feasible test solution is found, its objective function value is compared to the objective function value of the current solution. If the objective function value of the test solution is superior to that of the current solution ($E_t < E_C$), then the test solution replaces the current solution, the value of E_C is replaced by E_t, and the increment counter, s, is incremented by 1. If the value of E_t is not superior to that of E_C, then the metropolis criterion for acceptance is explored.[14]

Investigation of the metropolis criterion for accepting a test solution that is inferior to that of the current solution involves calculating the probability of the inferior test solution being accepted, firstly, the following difference in objective functions is determined:

$$\delta E = E_t - E_C \tag{8}$$

This difference will always be non-negative when determined because when determined, $E_t > E_C$. According to the metropolis criterion, the probability of this inferior solution being accepted as current $P(A)$ is as follows:

$$P(A) = e^{-\delta E/T} \quad (9)$$

A random number (Rand) is then generated. If Rand $< P(A)$ then the metropolis criterion is met and the inferior solution is accepted as the current solution, otherwise, the test solution is not accepted and the current solution remains. Regardless of whether-or-not the metropolis criterion is met, the iteration counter, s, is incremented by 1.

If the current solution is replaced by the test solution via regular acceptance, this new current solution is compared against the best solution found so far, if the new value of E_C is less than that of E_B, then the best solution is replaced by the current solution.

Each time the iteration counter s is incremented, it is compared against the value of Iter. As soon as the value of s equals the value of Iter, s is reset to 0 and the control parameter T is adjusted according to the following relationship:

$$T = T^*CR \quad (10)$$

The search heuristic continues in this fashion until the value of T reaches its stopping value T_F.

The presented pseudocode is intended to help one visualise the flow of steps as a part of this procedure:

```
Initialisation
While T > T_F
  For s = 1 to Iter
    Generate neighboring solution (compression, transfer
    or hyper compression)
  Update t
  Determine cost of neighboring solution
  If E_t < E_C then
    Replace current solution with test solution
    If E_t < E_B then replace best solution with test solution
  Else
    Examine metropolis criterion
    If Rand < e^(-δE/T) then replace current solution with
    test solution
  End if
  Next s
  T = T*CR
End while
Present best solution
End
```

Experimentation

To support the presented methodology, an experimental problem was solved via the described heuristic. The experimental problem consisted of 49 regions (each of the 48 contiguous US states and the District of Columbia). The travel distance from one region to another was determined as a function of the mileage from the largest city in one region to another via the use of (6). For example, the travel time from Wisconsin to Michigan is a function of the Euclidean distance from Milwaukee to Detroit, the largest cities in each respective state. The US Census Bureau[15] provided populations of cities and states; the mileage matrix was constructed via latitudinal and longitudinal coordinates as provided by the US Gazetteer.[16] As a result, then, this particular lockbox location problem has the largest city in each state (and the District of Columbia) serving as potential lockbox sites.

To illustrate, the 49-region problem was solved according to the following values of the variables: sales $(S) = 5000$ M, fixed-cost per lockbox (FC) = 150 K (where M is millions and K is thousands), interest rate $(i) = 10\%$, initial temperature $(T_1) = 75$, final temperature $(T_F) = 2$, cooling rate (CR) = 95%, iterations for simulated annealing (Iter) = 25, iterations without finding feasible solution (FIter) = 4. The results of the best solution found via simulated annealing are presented in Table 1.

The objective function value was \$3 803 267. The states in boldface are the host regions for each respective lockbox — it is important to note how the individual lockboxes are grouped into homogeneous geographic regions, the heuristic was constructed with Microsoft Visual Basic 5. The above solution required 2.85 CPU minutes on a Pentium

Table 1 Solution to Example Problem

Lockbox	*State(s)*	*Lockbox*	*State(s)*
1	**DC**	12	**MD,** DE, PA
2	**ME,** NH, MA	13	**WY**
3	**TX**	14	**NY,** CT
4	**CA,** AZ, NV, NM	15	**IL**
5	**MT**	16	**IA,** KA, MO, SD, NE, ND, MN, OK, WI
6	**VT**	17	**AR,** TN, MS, LA
7	**MI**	18	**NJ**
8	**ID,** UT	19	**IN**
9	**WA,** OR	20	**NC,** GA, KY, WV
10	**OH**	21	**AL**
11	**CO**	22	**RI**

120 processor. Figure 4 presents the iteration history of this problem, showing objective function values of the current and best solutions.

From Figure 4, note the converging pattern of the objective function value of the best solution found to that point, also note how the current solutions 'meanders' while exploring the feasibility continuum for improved solutions. Early in the process, the best solution is frequently updated, but this 'flattens out' by the time the iterations have been completed.

Factors of sales and fixed-cost per lockbox

The above problem was solved several times — at varying levels of the factors of annual sales (S) and fixed-cost per lockbox (FC). Sales varied from 500 M to 10 000 M in increments of 500 M. Fixed cost per lockbox varied from 10 K to 200 K in increments of 10 K. Sales and fixed cost per lockbox were varied to determine if the objective function value of the best solution found was sensitive to these factors. ANOVA and linear regression will be used to address this research question.

Simulated annealing vs a K-means clustering heuristic

It was also desired to compare the result of this simulated annealing heuristic with that of another, somewhat less formal, heuristic. Cluster analysis was used to provide the comparison heuristic, cluster analysis is the grouping of observations into categories. Observations whose attributes are similar to one another are likely to be placed in the same group. In the language of cluster analysis, 'similarity' is frequently measured by euclidean (straight-line) distance. Because of the frequent usage of euclidean distance in cluster analysis, this technique could be adapted to group geographic regions into categories where close proximity is a priority.[17] For the purpose of this research, these 'categories' could function as lockboxes.

K-means clustering could easily be used to design a lockbox system. First of all, the decision-maker decides how many lockboxes are to exist in a solution (analogous to a user deciding how many clusters to interpret). Then, by using K-means clustering, a solution is generated based upon euclidean (straight-line) distances between geographic regions. Standardized longitudinal and latitudinal coordinates are used for euclidean distance computations. From the K-means cluster analysis output, the members of each unique cluster are simply the regions in each unique lockbox, the host region of each lockbox is simply the region closest to the centroid of each cluster. In the event where there are exactly two regions that are members of a lockbox (where each region is equidistant from the centroid), the larger region in terms of population is selected to serve as the host for that particular lockbox.

For the experimental problem presented here, this K-means clustering approach to lockbox design is performed for when the decision-maker wants as few as fifteen lockboxes and as many as thirty, with all integer values between fifteen and thirty explored as well, this means that the K-means clustering approach is performed sixteen times. For each K-means clustering solution, objective function values are obtained via the variation of the parameters of S and FC as it was with the simulated annealing heuristic. The research question of interest here is whether-or-not there is a meaningful difference in the objective function values of the simulated annealing heuristic as compared to that of the K-means clustering heuristic, if so, which provides more desirable results. This research question is addressed via ANOVA.

Database

To summarise the experimentation component of this research, there are twenty levels of the factor S, twenty levels of the factor FC, and seventeen levels of the heuristic factor (one level is the simulated annealing heuristic and the other sixteen being the K-means clustering solutions). These three experimental factors above yield a database of 6800 observations ($20 \times 20 \times 17 = 6800$). It is also appropriate to note that the Simulated Annealing parameters of T_1, T_F, CR, Iter and FIter as the same as in the above example.

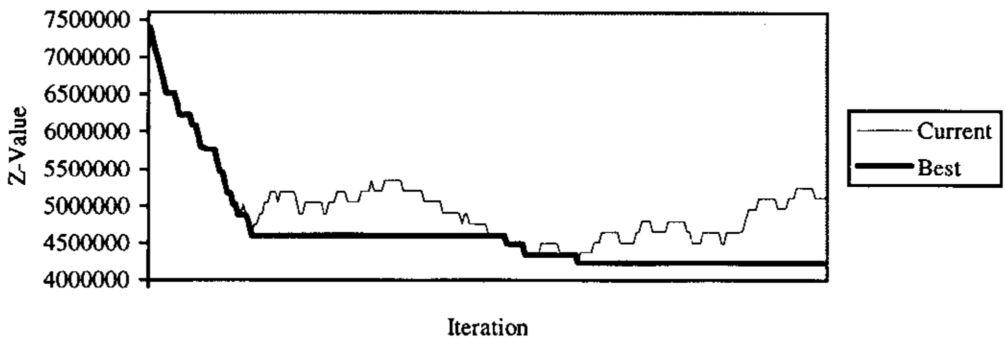


Figure 4 Iteration History of Example Problem.

Experimental results

Simulated annealing solution response to fixed cost and annual sales

ANOVA showed that for the simulated annealing approach, the objective function value of total cost was sensitive to both fixed cost per lockbox and annual sales (F-statistics of 391.04 and 241.48, respectively, with both P-values <0.001). Linear regression shows a meaningful positive relationship between fixed cost per lockbox and total cost ($t=23.94$, $P<0.001$ and $R^2=0.590$). Linear regression also shows a meaningful positive relationship between annual sales and total cost ($t=15.11$, $P<0.001$ and $R^2=0.364$). The fact that as the fixed cost per lockbox and the annual sales is increased, the objective function value of total cost also increases should not be surprising.

Another non-surprising result is that the number of lockboxes obtained from the simulated annealing solution is very dependent on the fixed cost per lockbox. Specifically, as FC increases, fewer lockboxes are needed, and vice versa ($t=-25.46$, $P<0.001$ and $R^2=0.620$). The simulated annealing heuristic is 'motivated' to find a solution with fewer lockboxes due to the increasing fixed cost of the lockboxes— a common characteristic of 'fixed-charge' problems. Perhaps more thought-provoking, however, is that the simulated annealing heuristic results in more lockboxes when the level of annual sales is increased and vice versa ($t=8.76$, $P<0.001$ and $R^2=0.162$). This is because as annual sales are increased, the heuristic is more 'motivated' to have more lockboxes to minimise the additional daily opportunity cost of remittances. This phenomena should enable the user to take some level of comfort in that the heuristic will find a 'near-optimal' solution despite variation in fixed cost and annual sales.

It is also worth noting that larger geographic regions will typically serve as host regions for lockboxes. This is because the simulated annealing heuristic is 'motivated' to have these regions serve as hosts because these hosts will better reduce the daily opportunity costs as compared to smaller regions, because the larger states in terms of population make larger contributions to total annual sales. The example problem, whose results are presented in Table 1 support this claim, where the larger states of Texas, California, Michigan, Ohio, New York, Illinois and New Jersey all serve as hosts of their respective lockboxes.

Simulated annealing vs K-means clustering

Comparing the simulated annealing heuristic with the K-means clustering heuristics (via ANOVA) showed that the objective function value of the solutions was sensitive to the heuristic used ($F=13.87$, $P<0.001$). Specifically, it was determined that the simulated annealing heuristic resulted in the most desirable mean objective function value — the simulated annealing heuristic resulted in a mean cost of \$3 934 947, while the K-means clustering approach resulted in a mean cost of \$4 265 123. As a more rigorous comparison between the simulated annealing heuristic and the K-means clustering approach, ANOVA was performed to compare the cost obtained from the simulated annealing heuristic with the means of *all* sixteen of the K-means clustering approaches. Table 2 shows the cost means and Tukey's pairwise comparisons for each of these seventeen heuristics.

From inspection of Table 2, it is clear that the simulated annealing heuristic is superior to all of the K-means heuristics in terms of mean cost with the exception of the K-means heuristics utilising 17, 18 and 19 lockboxes (CL17, CL18 and CL19). Despite the relative inferiority of the simulated annealing here, the differences are not meaningful at the $\alpha=0.05$ level of significance. The simulated annealing heuristic has a lower mean cost at the $\alpha=0.05$ level of significance when compared to the K-means clustering heuristics for 15 and 16 lockboxes (CL15 and CL16) and for 25 through 30 lockboxes (CL25–CL30). In summary, the simulated annealing heuristic provides statistically significant superior solutions in terms of cost for eight of the sixteen K-means clustering approaches. The K-means clustering approach never provides solutions superior to the simulated annealing approach at the $\alpha=0.05$ level. Figure 5 provides a chart showing the mean cost of the simulated annealing heuristic and the mean cost of each of the sixteen K-means clustering approaches along with the variation of ± one standard deviation.

Table 2 Presentation of Means for K-means Clustering Heuristics

Rule	*Mean*	*P-value*	*Rule*	*Mean*	*P-value*
CL15	4 629 297	< 0.001	CL23	4 141 100	0.774
CL16	4 617 192	< 0.001	CL24	4 285 703	0.110
CL17	3 917 707	> 0.999	CL25	4 340 312	0.031
CL18	3 902 017	> 0.999	CL26	4 409 950	0.004
CL19	3 865 440	> 0.999	CL27	4 509 049	< 0.001
CL20	3 950 194	> 0.999	CL28	4 509 060	< 0.001
CL21	4 033 174	0.995	CL29	4 478 275	< 0.001
CL22	4 105 136	0.868	CL30	4 548 360	< 0.001

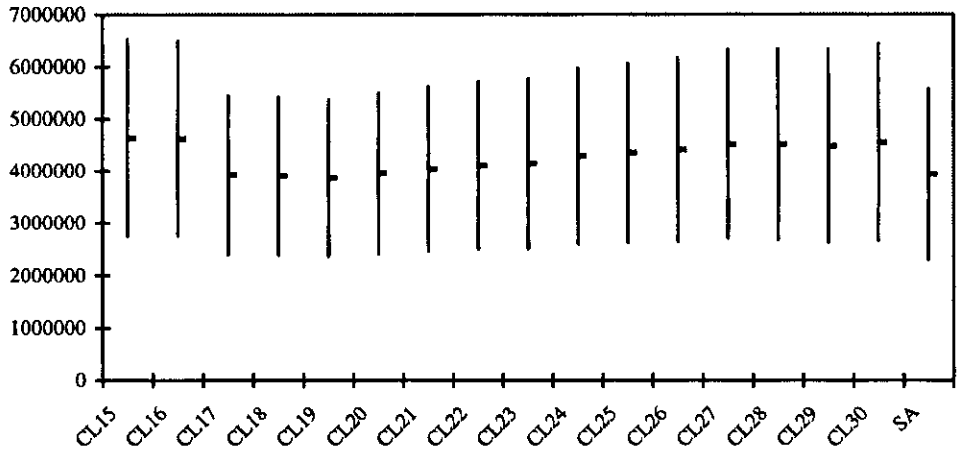


Figure 5 Cost means ± One Standard Deviation.

From inspection of Table 2 and Figure 5, it should become clear that while the K-means clustering approach does provide some formidable solutions (especially for seventeen through twenty-one lockboxes), the simulated annealing approach generally provides lower cost lockbox solutions, perhaps, however, the greatest relative advantage of using the simulated annealing approach is subjective. When using the simulated annealing approach, the user can take comfort in the fact that they are not responsible for deciding how many lockboxes to use — the heuristic addresses that by providing 'near-optimal' solutions given the pre-specified values of S and FC. The number of lockboxes with respect to the simulated annealing approach can then be thought of as endogenous — determined by the model. When using the K-means clustering approach, the user must decide how many lockboxes to specify. For the experimental problem used for this research, it is clear that relatively desirable solutions are found when the user decides on using between seventeen and twenty-one lockboxes (CL17–CL21) — when other values of lockboxes are specified, relatively inferior cost solutions result. This approach essentially, then, forces the user to go on a 'hunting expedition' to find the best approach via the K-means clustering approach. In addition, then, to the relative desirability of mean cost performance associated with the simulated annealing solution, its advantage of not having to pre-specify the number of lockboxes should not be overlooked.

Summary and conclusions

It is shown that simulated annealing can be effectively used to solve the lockbox location problem. The heuristic starts with a feasible solution and continues to modify this solution until some stopping criterion is found. During this modification process, solutions with objective function values inferior to their predecessors are sometimes accepted as current. This occasional acceptance of inferior solutions is done in an attempt to avoid being trapped at local optima—a distinguishing characteristic of simulated annealing.

Through experimentation, it can be seen that the objective function value found via simulated annealing is sensitive to both the level of annual sales and fixed costs, it can also be seen that the solutions found via simulated annealing are generally superior to a variety of those attained via a K-means clustering heuristic. While the simulated annealing heuristic does not guarantee optimality, the experiment does demonstrate its relative desirability over the more informal heuristics for problems of this type.

The simulated annealing search heuristic can be thought of as a subset of a more general type of search heuristic referred to as a metaheuristic, there are several types of metaheuristics used to solve challenging optimisation problems such as the one presented here.[18] An interesting opportunity for future research would be to compare the objective function performance of the simulated annealing heuristic with that of other types of metaheuristics, such as tabu search or genetic algorithms.

References

1 Levy FK (1966). An application of heuristic problem solving to accounts receivable management. *Mgmt Sci* **12:** B236–B244.
2 Nauss RE and Markland RM (1979). Solving lockbox location problems. *Financial Mgmt* **Spring:** 21–27.
3 Nauss RE and Markland RM (1981). Theory and application of an optimizing procedure for lock box location analysis. *Mgmt Sci* **27:** 855–865.
4 Stone BK (1981). Design of a receivable collection system: sequential building heuristics. *Mgmt Sci* **27:** 866–880.
5 Feilitz BD and White DL (1981). A two stage solution procedure for the lock box location procedure. *Mgmt Sci* **27:** 881–886.
6 Feilitz BD and White DL (1982). An evaluation and linking of alternative solution procedures for the lock box location problem. *J Bank Res* **Spring:** 17–27.

7 Maier SF and Vander Weide JH (1976–1977). A unified location model for cash disbursement and lockbox collections. *J Bank Res* **Summer:** 166–172.
8 Maier SF and Vander Weide JH (1983). What lockbox disbursement models really do. *J Finance* **38:** 361–371.
9 Davis SG(1986). Strategic planning for bank operations with multiple check-processing locations. *Interfaces* **16:** 1–12.
10 Davis SG *et al* (1989). Processing center location analysis for multiple branch banks. *OMEGA* **17:** 169–175.
11 Eglese RW (1990). Simulated annealing: a tool for operational research. *Eur J Opl Res* **46:** 271–281.
12 Kirkpatrick S, Gelatt CD and Veechi MP (1983). Optimization by simulated annealing. *Science* **220:** 671–679.
13 Maness TS and Zeitlow JT (1993). *Short-Term Financial Management: Texts, Cases, and Readings*. West: St. Paul, MN.
14 Metropolis N *et al* (1953). Equation of state calculations by fast computing machines. *J Chem Phys* **21:** 1087–1092.
15 http://www.census.gov/ftp/pub/population/www/
16 http://www.census.gov/cgi-bin/gazetteer
17 O'Kelley ME (1992). A clustering approach to the planar hub location problem. *Annals Opns Res* **40:** 339–353.
18 Glover F and Greenburg HJ (1989). New approaches for heuristic search: a bilateral linkage with artificial intelligence. *Eur J Opl Res* **39:** 119–130.

Received December 1997;
accepted August 1998 after one revision