Investigation of Practical, Robust and Flexible Decisions for Facility Location Problems Using Tabu Search and Simulation
Author(s): K. Sörensen
Published by: Palgrave Macmillan Journals on behalf of the Operational Research Society
Stable URL: http://www.jstor.org/stable/30132730

REFERENCES
Linked references are available on JSTOR for this article:

You may need to log in to JSTOR to access the linked references.
Investigation of practical, robust and flexible decisions for facility location problems using tabu search and simulation

K Sörensen*

University of Antwerp, Antwerp, Belgium

We investigate how robust and flexible solutions of stochastic capacitated facility location problems (CFLPs) can be obtained by combining metaheuristic optimization with Monte Carlo sampling techniques. To this end, we develop a tabu search procedure for the CFLP, and use this to solve an extensive set of stochastic versions of this problem.

Published online 13 December 2006

Keywords: facility location; stochastic; optimization; metaheuristics; robustness; flexibility

Introduction

The location of production or distribution facilities is typically a very important and cost-intensive decision, that can only be reversed or changed at extremely high cost. However, this decision is usually based on very volatile information such as customer demand and customer location. It is therefore paramount that the uncertainty of this information is taken into account when determining the actual location of facilities. In other words, the solution to a (stochastic) facility location problem (FLP) should in most cases be robust, flexible or both. A number of different definitions and taxonomies of robustness and/or flexibility have appeared in the literature, see for example, Kouvelis and Yu (1997); Mulvey et al (1995); Roy (1998); Vincke (1999); Dias and Climaco (1999); Branke (2001).

Consistent with the most frequently used definitions in the literature, we call a solution robust if its quality, as measured by some objective function value, is relatively insensitive to the actual realization of the stochastic parameters of the problem. In other words, a robust solution has a high quality across all potential realizations of the stochastic parameters. A solution is called flexible if it can be successfully adapted to any potential realization of the stochastic parameters of the problem. In other words, to obtain a flexible solution, it is necessary that the solution can be changed after the actual values of the stochastic parameters have been observed. This is done using some procedure usually called a repair procedure or recourse procedure. Summarizing, a solution is robust if it has a high quality across all potential outcomes of the stochastic parameters without first being adapted to these outcomes.

A solution is flexible if it has a high quality across all potential outcomes after being adapted to them. The division between robustness and flexibility is not clear-cut as some repair procedures may be so trivial that they are not considered as such.

To find robust and flexible solutions of difficult optimization problems, a strand of literature has focused on the stochastic FLP, and a number of solution methods for such problems have been developed.

Frank (1966) considers network problems with random demand and discusses some techniques to locate a facility at the absolute expected center or the absolute expected median. The nodes for which the maximum and the average expected demand-weighted distance are minimal are called the absolute expected center and the absolute expected median, respectively.

The objective of the m-median problem is to locate m facilities (medians) in order to minimize the sum of the demand-weighted distance from each customer to its nearest facility. Mirchandani and Odoni (1979) and Weaver and Church (1983) discuss the stochastic m-median problem. The objective of this problem is to locate m facilities on a stochastic network (i.e. a network in which travel costs and demands may be stochastic). Mirchandani et al (1985) present a multi-dimensional m-median problem where the objective is to locate m facilities when the travel and demand costs are stochastic, multiple commodities are considered and multiple objective functions exist. Using a modified version of a standard algorithm for the static uncapacitated FLP (due to Erlenkotter (1978)), Mirchandani et al (1985) develop an efficient way to solve this problem.

Jucker and Carlson (1976) consider the simple plant-location problem under uncertainty. Both price and demand can be uncertain in this problem and the objective is to maximize the expected utility for end-of-period profit. This
value, denoted by \( z \), is assumed to be given by the formula 
\[
z = E - \bar{\lambda}V \]
where \( E \) is the expected value of the profit, \( V \) is the variance of the profit and \( \bar{\lambda} \) is a parameter that represents the rate at which a firm will substitute variance for expected value, thus incorporating the risk-aversion of the decision maker into the model. Hodder (1984) uses a similar approach and incorporates elements of financial market models to diversify risks.

Louveaux (1986) and Louveaux and Peeters (1992) consider a stochastic location problem in which demands, variable production and transportation costs, as well as selling prices, can be random. Whereas the standard uncapacitated FLP assumes that facilities are built so that their capacity is equal to the total demand they serve, Louveaux (1986) argues that this is not a feasible assumption when demand is stochastic. The author therefore introduces a selling price, which makes it possible to let the location of the facilities depend on a trade-off between the cost of increasing the size of the facilities, the expected profit from sales and the probability of the various demand levels. Louveaux and Peeters (1992) present a dual-based heuristic, relying on the procedure proposed by Erlenkotter (1978), to solve the stochastic FLP. The authors use a scenario-based approach, finding solutions in the case of one, three or five scenarios.

Drezner (1987) develops heuristic solution methods for the unreliable \( p \)-median and the \((p, q)\)-center problem. In both problems, the probability that a certain facility will become inactive is given. In the \((p, q)\)-center problem, at most \( q \) facilities may become inactive at the same moment. The proposed solution procedure is based on the ideas found in Cooper (1963, 1964).

Most methods used to solve these problems have several disadvantages, that make them rather unsuitable for use in practical situations. First, they are very inflexible, in that they are developed for a specific stochastic facility location model and adapting them to the complex requirements of real-life applications is difficult. Secondly, many of these models use exact methods that are guaranteed to find the optimal solution. As a result, they are not suitable when the problem size is large. To tackle these problems, we develop an approach that uses metaheuristics to find robust and/or flexible solutions. Our approach uses Monte Carlo simulation to sample the underlying probability distributions of the stochastic parameters and calculate a measure of robustness for each solution encountered. A repair function is used to adapt the solution to a realization of the stochastic parameters, in order to be able to evaluate the flexibility of a solution. The approach explicitly recognizes the fact that the robustness of a solution may refer to a more complex reality than simply its average or its worst case performance. It therefore encourages the use of metaheuristics to generate a set of diverse solutions, from which a choice can be made using a multi-criteria decision making method, taking into account, for example, the decision maker’s preferences and risk averseness. An application of this approach to a single-machine scheduling problem was published in Sevaux and Sörensen (2004). Another approach that uses tabu search for stochastic optimization is due to Løkketangen and Woodruff (1996).

In stochastic programming, several sampling-based methods have been proposed, such as stochastic decomposition (Higle and Sen, 1991) and importance sampling (Infanger, 1994) for stochastic linear problems. For discrete stochastic problems, Norkin et al. (1998a,b) develop a sampling branch-and-bound. Whereas these methods use sampling at different steps of the optimization method to estimate, for example, function values, the sample average approximation method (Kleywegt et al., 2001) uses sampling to generate a set of sample scenarios and then attempts to optimize the corresponding deterministic expected-value problem. This method has been successfully applied to supply chain design problems (Goetschalckx et al., 2001) and routing problems (Verweij et al., 2003), among others. Our approach adopts a rather pragmatic view on stochastic optimization, approximating the stochastically optimal solution on two levels. First, a simulation- or sampling-based approach is used to estimate the robustness of a solution. Second, a metaheuristic optimization procedure is used to find a solution that is as close as possible to the solution with the best approximate robustness and flexibility. On the one hand, the sampling-based approach allows us to quickly evaluate the robustness or flexibility of each solution even when it has complex stochastic properties, such as interdependencies between the distributions of the stochastic parameters. On the other hand, the use of metaheuristics allows us to efficiently solve large-scale, real-life problems.

This paper is structured as follows. In the next section, we briefly discuss the (deterministic) CFLP and develop a simple tabu search procedure to solve it. We then describe the approach for robust and flexible optimization using metaheuristics and discuss its advantages over more traditional methods. The next section describes an elaborate set of experiments that was performed to show the effectiveness of the approach in solving different stochastic variants of the CFLP.

The capacitated FLP

Mathematical formulation

Among the most useful location models for the design of distribution systems is the so-called FLP. The objective of the FLP is to select an unspecified number of facilities from a finite number of candidate locations so as to minimize the sum of total fixed setup cost and total transportation cost required to serve the demand of a given set of customers. The demand of each customer has to be completely served from one or more open facilities. In the capacitated facility location problem (CFLP), each facility can only serve a limited amount of demand (called its capacity). Moreover, each candidate location has a certain fixed setup cost that is incurred when a facility is actually located there. The routing cost is the total demand-weighted distance multiplied by the cost per unit distance per unit demand.
Table 1 Facility location symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Explanation</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>Index for facilities</td>
<td>$j \in {1, \ldots, m}$</td>
</tr>
<tr>
<td>$i$</td>
<td>Index for customers</td>
<td>$i \in {1, \ldots, n}$</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of potential facilities</td>
<td></td>
</tr>
<tr>
<td>$n$</td>
<td>Number of customers</td>
<td></td>
</tr>
<tr>
<td>$x_j$</td>
<td>Facility status</td>
<td></td>
</tr>
<tr>
<td>$y_{ij}$</td>
<td>Amount of product delivered to customer $i$ from facility $j$</td>
<td></td>
</tr>
<tr>
<td>$f_j$</td>
<td>Fixed cost of opening facility $j$</td>
<td></td>
</tr>
<tr>
<td>$q_i$</td>
<td>Demand of customer $i$</td>
<td></td>
</tr>
<tr>
<td>$C_j$</td>
<td>Capacity of facility $j$</td>
<td></td>
</tr>
<tr>
<td>$c_{ij}$</td>
<td>Cost of delivering one unit of demand from facility $j$ to customer $i$</td>
<td></td>
</tr>
<tr>
<td>$n_e$</td>
<td>Number of evaluations per robust evaluation</td>
<td>See section An approach for robust and flexible optimization</td>
</tr>
</tbody>
</table>

The symbols used in the mathematical formulation of the CFLP are discussed in Table 1.

A mathematical formulation of the CFLP is given by Daskin (1995).

$$\min_{x_j,y_{ij}} \left[ \sum_j f_j x_j + \sum_j \sum_i c_{ij} y_{ij} \right]$$ (1)

s.t.

$$\sum_j y_{ij} = q_i \quad \forall i$$ (2)

$$\sum_i y_{ij} \leq C_j x_j \quad \forall j$$ (3)

$$x_j = 0, 1; \quad y_{ij} \geq 0 \quad \forall i, j$$ (4)

Equation (1) is the cost function that consists of location costs and routing (distribution) costs. Equation (2) ensures that the demand $q_i$ of each customer is completely served. Equation (3) ensures that the capacities $C_j$ of the facilities are not exceeded. These constraints also enforce the logical constraint that customers can only be served from open facilities. If the binary variables $x_j$ are fixed, the CFLP reduces to a transportation problem.

A tabu search algorithm for the CFLP

This section briefly describes a simple tabu search procedure for the CFLP. A solution is represented as a binary string, representing the status $x_j$ of all potential locations. Given a set of open facilities, the corresponding optimal distribution of demand over the open facilities can be quickly calculated using an algorithm for the transportation problem. Our tabu search procedure uses the Ford–Fulkerson algorithm.

At the basis of the tabu search procedure lies a steepest-descent local search mechanism that finds the facility that yields the largest cost decrease when it changes its status. When the search is stuck in a local optimum, that is, no facility can be opened or closed so that the total cost decreases, the local search returns the facility that yields the smallest cost increase.

The algorithm alternates between intensification and diversification strategies. Both strategies are implemented by applying appropriate memory structures. This is done in two loops, an outer loop and an inner loop. The inner loop is mainly used for intensification of the current solution. It implements a simple local search that is prevented from getting stuck in a local optimum too soon by a tabu list. The outer loop is used for diversification purposes and uses frequency-based memory. A schematic representation of the tabu search procedure can be found in Algorithm 1. An interesting feature of this algorithm is that it is completely deterministic—every action the algorithm takes is predictable and repeatable.

Algorithm 1. Facility location tabu search algorithm

1: initialize: empty tabu list, $\forall j \in \{1, \ldots, m\}: x_j \leftarrow 0$
2: repeat
3: repeat
4: repeat
5: find best facility $j$ to open/close according to local search
6: until facility $j$ is not tabu
7: if $x_j = 0$ then
8: $x_j \leftarrow 1$
9: else
10: $x_j \leftarrow 0$
11: end if
12: make $j$ tabu
13: update frequencies
14: until no solution improvement for a fixed number of iterations
15: open facility $f$ with lowest frequency
16: make $f$ tabu
17: until stopping conditions
The algorithm was evaluated on the capacitated facility location instances of the OR library (Beasley, 1988, 1990). These instances all have 50 customers. Thirteen instances have 16 potential locations, 12 have 25 and 12 have 50. The proposed tabu search algorithm is very effective in that a single run with tabu tenure 5 finds the optimal solution in all but two cases. In both of those cases, the solution found is less than 0.1% from optimal. Average computing times are approximately 2 s for the instances with 16 potential locations, 8 s for those with 25 potential locations and 21 s for those with 50.

An approach for robust and flexible optimization

A minimization problem can be written as

\[ f(x^*; p) = \min_{x \in X(p)} f(x; p), \]  

where \( f \) is the objective function and \( p \) is the set of parameters for the given problem instance. The optimal solution \( x^* \) should belong to the set of all feasible solutions for this problem instance (the domain \( X(p) \)). If the set of problem data contains some uncertain elements, we represent this set by \( \pi \). For a given solution \( x \), the objective function value \( f(x; \pi) \) now is a random variable that cannot be minimized. Also, the set of feasible solutions \( X(\pi) \) is stochastic, and as a result the feasibility of a given solution \( x \) depends on the realizations of the stochastic parameters.

A robust solution is characterized by the fact that it has a high quality across the set of potential realizations of the stochastic parameters of the problem. It is however important to note that the preferred formalization of robustness can differ between decision makers. A flexible solution is one that can be easily adapted to the realizations of the stochastic parameters. Flexibility implies the existence of some procedure to adapt a solution to the specific outcomes of the stochastic parameters. We call such a procedure a repair procedure and require of it that it is several orders of magnitude faster (in the computational sense) than the optimization procedure used to find the solution. The notion of repair function is related to the stochastic programming concept of second-stage or recourse decision, but differs from it in that the repair function is allowed to be any procedure, even a stochastic or heuristic one.

Overview

Our approach for robust and flexible optimization can be summarized as follows:

1. Use an optimization procedure that generates many diverse solutions. Most metaheuristics will do this.
2. Evaluate each solution generated with a robust evaluation function. Perform the following steps \( n_e \) times and combine the evaluations into a measure of robustness/flexibility.

(a) Sample the stochastic parameters of the problem for the given solution.
(b) (Only for flexible solutions) Use the repair procedure to improve the solution.
(c) Calculate the quality of the (repaired) solution.
3. Pick the solution that performs best with respect to this robust evaluation function.

Robust evaluation function

A typical robust evaluation function has the following form:

\[ f^*(x) = \frac{1}{n_e} \sum_{i=1}^{n_e} f(x; \mathcal{S}_i(\pi)) \]  

The sampling function \( \mathcal{S}_i \) generates a potential outcome of the stochastic data \( \pi \) of the problem. \( \mathcal{S}_i(\pi) \) is the \( i \)th sampling of the stochastic problem data. The solution \( x \) is evaluated on \( n_e \) samples and these evaluations are then combined into \( f^*(x) \)—a measure of the robustness of \( x \)—by taking the average.

If a repair function \( \mathcal{R} \) is used, the solution is first repaired before being evaluated. A robust evaluation function is

\[ f^*(x) = \frac{1}{n_e} \sum_{i=1}^{n_e} f(\mathcal{R}(x; (\mathcal{S}_i(\pi)))) \]  

Incorporating the decision maker’s risk-averseness

If the decision maker is relatively risk-averse, he might prefer to evaluate the worst-case performance of the solution. The robust evaluation function then becomes

\[ f^*(x) = \max_{i=1}^{n_e} f(x; \mathcal{S}_i(\pi)) \]  

A measure of the risk associated with a given solution is given by the standard deviation of all evaluations of the solution, given by

\[ \sigma^*(x) = \sqrt{\frac{1}{n_e-1} \sum_{i=1}^{n_e} [f(x; \mathcal{S}_i(\pi)) - f^*(x)]^2} \]  

Several of these measures of robustness can be calculated for each solution and a solution can be chosen using a multi-objective decision making process. Other, even more complex expressions of robustness can be considered, such as the probability that the quality of the solution falls below a certain threshold.

Penalty functions

Some solutions might become infeasible for some realizations of the stochastic parameters. The decision maker might therefore decide to only allow solutions that are feasible across all potential realizations. This would imply that a solution that is infeasible in at least one of the \( n_e \) cases, would receive an infinitely large robust evaluation function value. A less drastic approach is to allow for some infeasibility through the use...
of penalty functions. A robust evaluation function that incorporates penalty functions is of the form

\[ f^*(x) = \frac{1}{n_e} \sum_{i=1}^{n_e} [f(x; \mathcal{P}_i(\pi)) + \mathcal{P}(x; \mathcal{P}_i(\pi))] \]  

(10)

where \( \mathcal{P} \) is a penalty function that should reflect the ‘severity’ of the constraint violation of solution \( x \) under the realization \( \mathcal{P}_i(\pi) \) of the stochastic parameters.

The number of evaluations to perform per robust evaluation, \( n_e \), has an important impact on the performance of the approach. A value of \( n_e \) that is too small will result in a large uncertainty about the expected quality of the solutions. On the other hand, a value that is too large will unnecessarily increase the computation time required. After the experiments section, under ‘Results’, we discuss how an appropriate value for \( n_e \) can be obtained using a limited pilot study.

**Advantages of the approach**

The approach provides an answer to several problems that cannot be adequately dealt with by more traditional methods based on stochastic programming. These include the following:

- **Ease of use**: The approach uses metaheuristic optimization procedures designed for deterministic problems and adapts them in a rather straightforward way to stochastic problem formulations. The metaheuristic itself is left almost unchanged—only the evaluation function is adapted.

- **Flexibility**: The approach is—at least in principle—applicable to any stochastic optimization problem. There is no limit on the number and type of stochastic parameters that can be entered into the problem formulation. This implies that all information that is available can be added to the problem formulation. If a decision maker has precise data on the distribution of some of the problem data, these can be added. If scenarios can be created that reflect realistic situations that can be expected, these too can be used.

- **Extensibility**: The approach can easily be extended to include penalty functions for infeasible solutions or to take the risk preference of the decision maker into account.

- **Applicability to large-scale stochastic problems**: Finding the optimum (usually defined as ‘optimizing the expected value’ or ‘optimizing the worst-case performance’) of large-scale stochastic problems is often intractable. This is due to two reasons. First, the deterministic equivalent of these problems is often \( NP \) hard (especially in the case of integer programming problems), making it impossible to find the optimal value to the deterministic problem in a reasonable amount of time. Second, as indicated by Birge (1997), the complexity of stochastic programs grows proportionally to the number of possible realizations of the stochastic parameters, which in turn grows exponentially with the number of stochastic parameters. Because in realistic cases this number is usually very large or even infinite (as in the case of continuous distributions), only very small problems can be solved to ‘optimality’.

- **Objective independence**: Robustness and flexibility are terms that can be used to express a number of different properties of a solution, of which average-case and worst-case performance are probably the most widely used. However, many others can be considered. The approach does not force a choice of objective onto the decision maker. Using the robust evaluation function, decision makers can determine their preferred type of robustness or flexibility and find the best solution according to their preferences. Different expressions of the concepts of robustness can be evaluated simultaneously.

**Robust and flexible facility location experiments**

The versatility of our approach for robust and flexible optimization is demonstrated by the fact that several rather different versions of the stochastic FLP are tackled. These include problems with stochastic costs, problems with stochastic demand and problems with stochastic customers. In each of these experiments, all stochastic parameters are independently distributed. A separate section discusses an experiment in which scenarios are used to express potential realizations of the stochastic parameters. After the experiments, we discuss briefly how a ‘reasonable’ amount of evaluations to calculate a robust evaluation function can be determined. We also discuss the computing times that are required to solve robust and flexible FLPs by means of the modified tabu search procedure. It is shown that a relatively modest increase in computing time results from using the robust evaluation function only for evaluation purposes. When using the robust evaluation function to guide the search, computing time increases become rather large, but even then this might be justified in real-life situations. As mentioned, facility location decisions are usually long-term decisions and the few hours (or even days) of computation can be considered negligible.

**Tabu search parameters**

For all problems discussed in this paper, the tabu search procedure developed in the previous section uses the same parameter settings. Unless otherwise indicated, the tabu tenure is 5, the inner loop (using the tabu list) is performed until 10 non-improving moves, and the outer loop (using the frequencies memory) is performed until 10 non-improving inner loops. All experiments are run on an AMD Athlon 1100 PC with 512 Mb RAM.

**General experiment setup**

Data sets with stochastic parameters are derived from the 37 data sets for the deterministic CFLP, obtained from the OR Library (Beasley, 1990) (http://people.brunel.ac.uk/~mastjjb/ jeb/info.html, accessed 6 June 2006). The stochastic parameters are usually defined as a probability distribution having the deterministic parameter value as expected value. Only
summary results are presented, but detailed results can be obtained from the author.

For each data set and parameter setting, the tabu search procedure finds the solution that scores best according to the robust evaluation function. This solution is denoted by \( x_r \). In most cases, this solution is compared to the best solution found using the ordinary evaluation function, denoted by \( x_o \). Unless otherwise indicated, this solution optimizes the mean value formulation of the stochastic problem, that is, this solution is the best one encountered for the (deterministic) problem with all stochastic parameters set to equal their expected values. Solutions are compared both in terms of their score on the robust evaluation function and their standard deviation across all samples. The standard deviation can be regarded as a measure of the risk associated with a given solution. As mentioned, many other expressions of robustness can be computed and compared using multi-objective decision making techniques, but we only calculate two for reasons of simplicity. In all cases (except for the experiment using scenarios), we set the number of evaluations \( n_e \) to 100.

If we express robustness by the expected objective function value, we can test statistically whether \( x_r \) is more robust (or flexible) than \( x_o \) by comparing \( f^*(x_r) \) to \( f^*(x_o) \). Since the number of evaluations \( n_e \) is sufficiently large, both \( f^*(x_r) \) and \( f^*(x_o) \) are approximately normally distributed. We can therefore formulate and test the null hypothesis \( H_0 \) that the robust evaluation function values of \( x_r \) and \( x_o \) are equal. This hypothesis is tested against the alternate hypothesis \( H_1 \) that the robust evaluation function value of \( x_r \) is smaller than that of \( x_o \). Since \( x_r \) is the solution that minimizes \( f^* \), it follows by definition that \( f^*(x_r) \leq f^*(x_o) \).

The hypotheses to test can be formulated as

\[
H_0: \quad f^*(x_r) = f^*(x_o)
\]

\[
H_1: \quad f^*(x_r) < f^*(x_o)
\]

To test \( H_0 \), we construct the following statistic

\[
Z = \frac{f^*(x_r) - f^*(x_o)}{s}
\]

where \( s \) is the standard error of \( f^*(x_r) - f^*(x_o) \), which can be calculated as

\[
s = \sqrt{\frac{(\sigma(x_r))^2 + (\sigma(x_o))^2}{n_e}}
\]

The null hypothesis can be rejected at the \( \alpha \) significance level if \( Z > z_\alpha \), where \( z_\alpha \) is the upper \( \alpha \) critical point of the standard normal distribution. We generally test at the 0.05 and the 0.01 significance levels (\( z_{0.05} \approx 1.65 \) and \( z_{0.01} \approx 2.33 \)).

Problems with stochastic costs

This section describes a set of experiments in which the cost of delivering a unit of demand from facility \( j \) to customer \( i \), represented by \( c_{ij} \), is stochastic. The corresponding deterministic cost is given by \( \bar{c}_{ij} \). For a given solution, changing the delivery costs changes the objective function value. This potentially raises the need for robust and flexible solutions. Two different variants can be discussed:

- Problems in which no changes can be made to the assignments after the actual costs have been observed. This type of problem requires robust solutions.
- Problems in which the assignments can be changed after the costs have been measured. This type of problem requires flexible solutions.

In the first case, the assignments of customer demand to the various facilities is decided before the distribution costs are realized and left unchanged once the actual distribution costs are observed. The robust evaluation function is equal to

\[
f^*(x) = \sum_j f_j x_j + \frac{1}{n_e} \sum_i \sum_j c_{ijk} y_{ij}
\]

where \( k \) is the index for the evaluations in the robust evaluation function. \( c_{ijk} \) is the \( k \)th random number drawn from the distribution of \( c_{ij} \) with \( k \in [1, n_e] \). It can be shown that the optimal solution of this FLP with stochastic cost is equal to the optimal solution of its (deterministic) expected-value problem, in which all stochastic costs have been replaced by their expected values. Indeed, some experiments show that the best solutions found by our approach are either the same or very similar to the optimal solutions for the deterministic problem. We therefore focus on the more interesting case, in which the decision maker is allowed to redistribute the demand over the different open facilities after the actual values of the distribution costs have been observed.

Experiment setup. The data sets for problems with stochastic costs are derived from the data sets for deterministic problems by assuming that the deterministic distribution cost represents the mean of a distribution. The costs to deliver a unit of product to customer \( i \) from facility \( j \) is uniformly distributed between an upper and a lower bound; this can be written as \( c_{ij} \sim U(c_{ij}^{\text{up}}, c_{ij}^{\text{lo}}) \). We express the upper and lower limit (UL and LL) as a fraction of the deterministic cost and ensure that the deterministic cost is equal to the expected value of the uniform distribution by setting (UL + LL) / 2 = 1. The expected value of the uniform distribution of \( c_{ij} \), denoted by \( \bar{c}_{ij} \), is the transportation cost between customer \( i \) and facility \( j \) found in the deterministic data set. The assumption that all costs are distributed according to a uniform distribution with the same relative upper and lower limits can be easily relaxed.

In this case, the assignment of customer demand to the facilities can change after the demand has been observed. For a given solution (ie a set of open and closed facilities), the repair procedure uses the Ford–Fulkerson algorithm for the transportation problem to calculate the optimal assignment of demand to customers, given the actual realizations of the transportation costs.
The robust evaluation function, including the repair procedure is

$$f^*(x) = \sum_j f_j x_j + \frac{1}{n_c} \sum_k \min_{y_{ijk}} \left( \sum_i \sum_j c_{ijk} y_{ijk} \right)$$

(16)

s.t.

$$\sum_j y_{ijk} = q_i \quad \forall i, k$$

(17)

$$\sum_j y_{ijk} \leq C_j x_j \quad \forall j, k$$

(18)

$$x_j = 0, 1; \quad y_{ijk} \geq 0 \quad \forall i, j, k$$

(19)

In order to compute the robust evaluation function, $n_c$ transportation problems must be solved. The repair function finds the optimal allocation of demand to the various open facilities. Although it is a desirable property of a repair procedure to always find the best possible solution, given a certain realization of the stochastic parameters, this is not required. Other, less computing-time intensive procedures may be used instead. In four experiment sets, distribution costs are allowed to deviate from the average by 20, 50, 80 and 100%, respectively. The number of evaluations per robust evaluation is equal to 100 in all cases.

**Results.** Table 2 contains summary results of the experiments with stochastic costs and repair function.

Table 2  Experiment with stochastic costs and repair, summary

<table>
<thead>
<tr>
<th>$LL$</th>
<th>$UL$</th>
<th>$f^<em>(x_r) &lt; f^</em>(x_o)$</th>
<th>Avg.</th>
<th>$z = 0.05$ (%)</th>
<th>$z = 0.01$ (%)</th>
<th>CPU (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.80</td>
<td>1.20</td>
<td>40.54</td>
<td>18.92</td>
<td>49.57</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>1.50</td>
<td>75.68</td>
<td>48.65</td>
<td>41.30</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>1.80</td>
<td>89.19</td>
<td>81.08</td>
<td>44.32</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>2.00</td>
<td>94.59</td>
<td>78.38</td>
<td>43.70</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>All</td>
<td></td>
<td>75.00</td>
<td>56.76</td>
<td>44.72</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The robust evaluation function, including the repair procedure is

In order to compute the robust evaluation function, $n_c$ transportation problems must be solved. The repair function finds the optimal allocation of demand to the various open facilities. Although it is a desirable property of a repair procedure to always find the best possible solution, given a certain realization of the stochastic parameters, this is not required. Other, less computing-time intensive procedures may be used instead. In four experiment sets, distribution costs are allowed to deviate from the average by 20, 50, 80 and 100%, respectively. The number of evaluations per robust evaluation is equal to 100 in all cases.

**Results.** Table 2 contains summary results of the experiments with stochastic costs and repair function.

Table 2 shows that the number of significantly more flexible solutions rises as the range of the stochastic costs widens. When stochastic costs are allowed to range from 0 to $2\hat{c}_{ij}$, a significantly more flexible solution can be found in almost 95% of the cases (0.05 significance level). The value of $f^*(x_r)$ is often considerably lower than that of $f^*(x_o)$, indicating that there is a strong need for flexible solutions. Also note that $f^*(x_r)$ is often considerably lower than $f^*(x_o)$. This indicates that important cost savings can be made by re-distributing customer demand across the various open facilities.

A decision maker does not necessarily have to choose the solution that has the best robust evaluation function value. The standard deviation can also be considered and a multi-objective decision can be taken. This is demonstrated in Figure 1. This figure shows all solutions—except the infeasible ones—encountered during a run of the tabu search algorithm. The problem data file used in this figure is cap71, cost parameters are $c_{ij} = 0.5$ and $c_{ij} = 1.5$.

As can be seen in this figure, the optimal solution of the deterministic problem is dominated by a number of other solutions, that is, it has both a worse average quality (robust evaluation function value) and a larger standard deviation. The optimal solution of the deterministic problem therefore clearly is not a good solution to the problem with stochastic costs.

The leftmost point in Figure 1 corresponds to the solution that has the best robust evaluation function value. However, several other solutions are encountered that have a lower standard deviation. For almost all problems, there is at least one solution that dominates the best solution of the deterministic problem (i.e., performs better with respect to both the robust evaluation function and the standard deviation). It is possible to adapt the tabu search procedure so that it searches for a solution that simultaneously minimizes the robust evaluation and the standard deviation. A possible way to do this is to find the solution that minimizes $f^*(x) + \lambda \sigma^*(x)$, where $\lambda$ is a parameter that determines the relative importance of the standard deviation. Another—and in our opinion better—approach is to find the set of non-dominated solutions and pick one using a multi-criteria method. This approach is preferable because it does not require the decision maker to fix the value of $\lambda$ beforehand. However, it requires supplementary information about the preference structure of the decision maker, with respect to the various measures of robustness used.

**Problems with stochastic demand**

In a second set of experiments, customer demand is assumed to be stochastic. When the actual customer demand becomes known, it needs to be distributed across the open
facilities. This is done using the algorithm for the transportation problem.

Experiment setup. As usual, the value of the robust evaluation function is calculated by taking the average of \( n_e \) evaluations of the solution on a sample of the stochastic parameters. At each evaluation, a random sample is drawn from the demand distribution of each customer. It is then checked whether the total demand exceeds the total capacity of all open warehouses. If it does, the current configuration of open warehouses is declared infeasible and rejected. If total demand can be satisfied, the transportation algorithm is used to distribute the demand of the customers to the open facilities and the cost of this solution is stored. These steps are repeated \( n_e \) times.

The robust evaluation function, including the repair procedure, for this problem can be calculated as follows:

\[
f^*(x) = \sum_j f_j x_j + \frac{1}{n_e} \left[ \sum_i \left( \sum_j C_{ij} y_{ijk} \right) \right]
\]

(20)

s.t.

\[
\sum_j y_{ijk} = q_{ik} \quad \forall i, k
\]

(21)

\[
\sum_i y_{ijk} \leq C_{j} x_{j} \quad \forall j, k
\]

(22)

\[
x_{j} = 0, 1; \quad y_{ijk} \geq 0 \quad \forall i, j, k
\]

(23)

where \( q_{ik} \) is the \( k \)th random number drawn from the distribution of \( q_i \). \( \bar{q}_i \) is the deterministic demand, equal to the mean of \( q_i \). Let the number of evaluations be \( n_e \). Then, calculating a robust evaluation function involves calculating \( n_e \) transportation problems. As mentioned for each set of samples from the demand distributions, it is possible that total demand will be greater than the total capacity, that is, that for some \( k \in [1 \ldots n_e] \),

\[
\sum_i q_{ik} \geq \sum_j C_{j}
\]

In this case, demand cannot be satisfied and the solution is infeasible for this sampling of the stochastic parameters. We adopt a strict approach: a solution that becomes infeasible for at least one sampling is considered an infeasible solution and is rejected.

As in the previous sets of experiments, the data sets for the experiments with stochastic demand are derived from the data sets for the deterministic FLP. A lower limit and an upper limit on the demand of each customer is given by \( q_i^L \) and \( q_i^U \), respectively. As in the previous experiments, we express these values as a fraction of the expected value, and require that \((LL + UL)/2 = 1\). In four sets of experiments, demand of each customer is uniformly distributed between varying lower and upper limits, respectively, 20, 50, 80 and 100% from the expected value \( \bar{q}_i \). The number of evaluations per robust evaluation \( n_e \) is equal to 100 in all experiments.

Results. A summary of the results is presented in Table 3.

In this table, we make a comparison between the best solution found by the tabu search procedure with the robust evaluation function \( (x_r) \) and the best solution found by the tabu search procedure with the ordinary evaluation function \( (x_o) \). Solution \( x_r \) corresponds to the solution that would be found by the tabu search procedure if we would ignore all stochastic information and assume that all stochastic parameters were in fact deterministic and equal to their expected value. In the columns labelled ‘\( x_r \) feas’ and ‘\( x_o \) feas’, we indicate the percentage of experiments in which \( x_r \) and \( x_o \) were feasible solutions for all \( n_e \) evaluations. At the 0.05 significance level, \( x_r \) is more robust than \( x_o \), in on average 55.41% of all cases. At the 0.01 significance level, this number drops to 32.43%. This shows that the solutions found by the tabu search procedure with the robust evaluation function are more robust than those found with the ordinary evaluation function. The percentage of cases in which \( x_r \) is significantly more robust than \( x_o \) also seems to rise with widening demand ranges.

The number of times the best solution found with the ordinary evaluation function is infeasible rises with the amount of allowed deviation. If the allowed deviation is 100% (ie individual customer demand can vary between 0 and \( 2\bar{q}_i \)), the number of feasible solutions found by the ordinary evaluation function drops below 68%. The number of feasible solutions found by the robust evaluation function is still almost 92%. The only cases in which the robust evaluation function is unable to find a solution that remains feasible across all evaluations is when the problem has extremely small slack, that
is, the total capacity of all facilities is only slightly larger than the total demand of all customers. In those problems, even the solution that opens all facilities can become infeasible in some cases.

An explanation for the fact that a solution that is more robust than \( x_r \) can be found in only 50% of the cases 0.05 significance level) is the following. In some cases, the capacity of one or two facilities is enough to serve the demand of all customers. In those cases, almost all facilities exhibit a large excess capacity and the solutions do not become infeasible. Moreover, since the capacity of each single facility is very large, most customers can be serviced entirely from their nearest facility, that is, the problem behaves as in the uncapacitated case. Even when some customers increase their demand, they can still be served from the same facility. Because the cost matrix does not change, this results in a solution of the transportation problem that is approximately the same and as a result in a distribution cost that is largely the same. In those cases, it is likely that no solution exists that has a smaller robust evaluation function value than the best solution of the deterministic problem.

In about 70% of the cases, the tabu search with robust evaluation function finds a solution that has a better robust solution value and a lower standard deviation. In these cases, \( x_r \) dominates \( x_o \). An example is shown in Figure 2. The data file used in this example is cap134. Costs are stochastic between 0 and 2\( q_c \). It is clear that the optimal solution of the mean value formulation of the stochastic problem is a very bad solution—both in terms of the average solution quality and the standard deviation of the solution quality—for the problem with stochastic demands. Several efficient solutions are encountered during the search, among which can be chosen by the decision maker according to his or her preferences.

**Penalty function for unmet demand.** A rather straightforward extension of the previous problem with stochastic demand is to include a penalty function for undelivered demand. For reasons of simplicity, we assume that the penalty is proportional to the amount of unmet demand. We further assume that not delivering demand is only allowed if there is an actual shortage, that is, all demand that can be delivered should be delivered, regardless of the cost.

The penalty function \( \mathcal{P} \) for this problem can be defined as

\[
\mathcal{P}(x) = \gamma \frac{1}{n_c} \sum_k \min \left( \sum_j C_j x_j - \sum_i \sum_j y_{ijk}, 0 \right)
\]

where \( \gamma \) is the penalty factor, that is, the extra cost per unit of undelivered demand. The penalty factor is added to the robust evaluation function value, which can be calculated as in Equations (20)–(23).

A limited set of experiments was performed for this problem with \( \gamma = 100 \). Allowed deviations from the average demand were between 10 and 100%. Average computing times per run were 192 s.

The difference between \( f(x_r) \) and \( f(x_0) \) is significant in only 54.05 and 40.54% of the cases at the 0.05 and 0.01 significance level, respectively. These figures are very close to those reported in the experiment without penalty function. In instances of the experiments without penalty function where \( x_o \) was infeasible, that is, instances with relatively small slack, relatively high penalties will now be incurred. In cases where there is a lot of slack, penalties will be negligible and the best solution according to the ordinary evaluation function will be robust with respect to changing customer demand.

**Problems with stochastic customers.**

In the experiments described in this section, customers are assumed to be stochastic, that is, they have a fixed probability of requiring service. We assume that the location of facilities needs to be determined before the list of customers to service is known, but that demand can be redistributed afterward. Demand is assumed to be known and constant for each customer.

**Experiment setup.** We assume that each customer has the same probability \( p_c \) of requiring service. This can however easily be relaxed.

The robust evaluation function is given by

\[
f^*(x) = \sum_j f_j x_j + \frac{1}{n_c} \sum_k \min \left( \sum_{i \in I_k} \sum_j c_{ij} y_{ijk} \right)
\]

s.t.

\[
\sum_j y_{ijk} = q_{ik} \quad \forall i \in I_k, k
\]

\[
\sum_{i \in I_k} y_{ijk} \leq C_j x_j \quad \forall j, k
\]

\[
x_j = 0, 1; \quad y_{ijk} \geq 0 \quad \forall i \in I_k, j, k
\]

---

**Figure 2** Robust facility location with stochastic demand.
Table 4: Experiment with stochastic customers, summary

<table>
<thead>
<tr>
<th>$p_c$ (%)</th>
<th>$x_r$ feas (%)</th>
<th>$x'_o$ feas (%)</th>
<th>$f^<em>(x_r) &lt; f^</em>(x'_o)$</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\alpha = 0.05$ (%)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\alpha = 0.01$ (%)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU (s)</td>
<td></td>
</tr>
<tr>
<td>80.00</td>
<td>100.00</td>
<td>89.19</td>
<td>86.40</td>
<td>159.76</td>
</tr>
<tr>
<td>60.00</td>
<td>100.00</td>
<td>78.38</td>
<td>83.78</td>
<td>87.46</td>
</tr>
<tr>
<td>40.00</td>
<td>100.00</td>
<td>81.08</td>
<td>83.78</td>
<td>36.27</td>
</tr>
<tr>
<td>20.00</td>
<td>100.00</td>
<td>62.16</td>
<td>72.97</td>
<td>14.35</td>
</tr>
<tr>
<td>All</td>
<td>100.00</td>
<td>77.70</td>
<td>81.76</td>
<td>74.46</td>
</tr>
</tbody>
</table>

$I_k$ is the $k$th set of randomly selected customers. Each customer is selected independently with a probability $p_c$. Calculating a robust evaluation function value requires the calculation of $n_c$ transportation problems, but the size of the transportation problems is almost always smaller than in the deterministic case (because the set of customers is almost always smaller).

Of course, comparing a solution of the problem with stochastic customers to a solution of the corresponding deterministic problem (i.e., with all customers present) constitutes an unfair comparison. In the solution of the deterministic problem, much more demand has to be assigned and the cost of this problem therefore will be much higher. It can be argued that there does not exist a valid mean value formulation of this problem. We therefore compare $x_r$, the solution that has the highest robust evaluation function value, to the solution $x'_o$ that has the best evaluation on any single random instantiation of the problem data. In other words, $x'_o$ is the solution that has the highest ordinary evaluation function value on a single sampling of the stochastic parameters of the problem.

When, for a set of customers $I_s$, total demand of all open facilities is insufficient, the solution is infeasible. With respect to the robust evaluation function, a solution is called infeasible if there is at least one evaluation for which the solution is infeasible. In four experiment sets, $p_c$ is set equal to 80, 60, 40, and 20%, respectively.

Results. Table 4 summarizes the results for the experiments with stochastic customers.

The solution found by the tabu search procedure with robust evaluation function is always feasible, contrary to $x'_o$, which is infeasible in over 22% of the cases. Moreover, the modified tabu search procedure finds a significantly better solution in almost 82% of the cases (0.05 significance level) and 72% of the cases (0.01 significance level). Not surprisingly, the need for flexibility rises with a decreasing probability for customer presence.

Scenario-based flexible optimization

The main goal of this final experiment is to show the versatility of the method. Instead of assigning a probability distribution to each of the stochastic parameters of the problem individually, a set of scenarios is created. The number of customers can vary across scenarios, as well as the demand of each customer and the cost of serving a unit of demand from the different facilities.

In other words, each scenario $s$ contains a set of customers $I_s$ of cardinality $|I_s|$, a demand $q_{is}$ of each customer and a set of costs $c_{ij}$ between customers and facilities.

Because the set of customers is allowed to vary per scenario, no solution can remain feasible across all scenarios. This implies that a repair function is needed to determine which amount of customer demand to serve from which facility. Like in the previous experiments, the repair function is the transportation algorithm. The robust evaluation function value is obtained by calculating a transportation problem per scenario, and combining these evaluations. Like in the problems discussed before, we wish to find a good average performance of the solution across all scenarios. We therefore calculate the following robust evaluation function for each solution:

$$f^*(x) = \sum_j f_j x_j + \min_{y_{is}} \left[ \frac{1}{n_s} \sum_s \sum_{i \in I} \sum_j c_{ij} y_{is} \right]$$

s.t.

$$\sum_j y_{is} = q_{is} \quad \forall i \in I_s, s$$

$$\sum_i y_{is} \leq c_{ij} x_j \quad \forall j, s$$

$$x_j = 0, 1 \quad \forall j$$

$$y_{is} \geq 0 \quad \forall i \in I_s, j, s$$

The values of $q_{is}$ and $c_{ij}$, and $I_s$, the set of indexes of customers present in scenario $s$, are determined in each scenario. $n_s$ is the number of scenarios. The values of $y_{ij}$ are calculated by the repair function. Note that this robust evaluation function implicitly assumes that all scenarios are equally likely to occur. If each scenario has a different probability $p_s$, then the factor $1/n_s$ in Equation (29) is replaced by $p_s$.

Scenario creation. According to several authors, for example, Kouvelis and Yu (1997), the act of formulating scenarios has the advantage that it forces the decision maker to consider several potential options for the future. Moreover,
interdependencies between different problem variables can be easily entered into the scenarios.

Based on a 50-customer deterministic data set, five scenarios are created with 30, 40, 50, 60 and 70 customers, respectively. For each scenario, demand of each customer is a normally distributed random variate, distributed as follows.

\[ q_{ij} \sim \mathcal{N}\left( \bar{q}, \frac{5}{5} \right) \]  

where \( \bar{q} \) is the average demand corresponding to the deterministic data set of 50 customers. Costs are also randomly generated as

\[ c_{ij} \sim \mathcal{N}\left( \frac{50}{|I_j|} \bar{c}, \frac{5}{|I_k|} \bar{c} \right) \]  

In these equations, \( \bar{c} \) is the average cost to deliver the demand of a customer from a facility and \(|I_i|\) is the number of customers in scenario \( s \). This ensures that the five scenarios are approximately equally important (ie for a given solution, the total cost of each scenario is approximately equal). Hence, no single scenario can dominate the search process. The set of facilities is the same for each scenario. Like in the deterministic data set, the number of potential facilities is equal to 50. We set the value of \( \bar{q} \) 1000 and the value of \( \bar{c} \) to 50000.

**Results.** Given the fact that the computation of the robust evaluation function value for a given solution requires the calculation of only five deterministic evaluation functions, the robust evaluation function can be used to determine the best possible move without excessive computation time requirements. Therefore, this experiment uses the robust evaluation not only for evaluation purposes, but also to guide the search. A more elaborate discussion follows.

The best solution found by using the robust evaluation function is compared to the best (deterministic) solution found for each of the scenarios. As an example, the first solution in Table 5 is the one that has the lowest cost for scenario 1. We also record the solution that has the lowest value \( \sigma^*(x) \).

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Customers</th>
<th>( f^*(x) )</th>
<th>( \sigma^*(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>30</td>
<td>1747592.145</td>
<td>85574.743</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>40</td>
<td>1758065.312</td>
<td>59875.474</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>50</td>
<td>1754753.284</td>
<td>76525.394</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>60</td>
<td>1760438.904</td>
<td>40767.742</td>
</tr>
<tr>
<td>Scenario 5</td>
<td>70</td>
<td>1770545.989</td>
<td>62156.218</td>
</tr>
</tbody>
</table>

any of the scenario-optimized solutions are significant at the 0.01 significance level. In addition, a measure like the EVPI (expected value of perfect information) can be used to determine the amount of money a decision maker would be willing to spend to find out in advance which scenario will occur. Total computing time needed to determine the robust solutions is 163 s.

**Number of evaluations**

An important question is what constitutes a sensible number of evaluations per robust evaluation. On the one hand, increasing the number of evaluations also increases the computing time. This is especially true when a repair procedure is used to improve the quality of the solutions after sampling the stochastic parameters of the problem, and is discussed later. On the other hand, an insufficient number of evaluations might result in an arbitrary solution receiving a good robust evaluation function value by chance.

Some practical guidelines will be mentioned to help the decision maker determine an appropriate number of evaluations. Three main criteria to determine whether the number of evaluations is sufficient are:

*The consistency of the evaluation:* While a robust evaluation function generally involves randomness, a given solution should always have approximately the same robust evaluation function value. A large variance in the robust evaluation function values of a single solution is an indication that the number of evaluations should be increased.

*Confidence interval of the evaluation:* This criterion is closely related to the previous one. As a robust evaluation function value is based on a limited number of samples, it is only an approximation of the ‘real’ average performance of a given solution. For a sufficiently large number of evaluations (eg \( n_e \) > 30), the central limit theorem states that \( f^*(x) \) is approximately normally distributed. From this, it follows that an \( 100(1 - \alpha) \) confidence interval for the real average evaluation of a solution is given by

\[ f^*(x) \pm z_{1-\alpha/2} \sqrt{\frac{(\sigma^*(x))^2}{n_e}} \]

See Law and Kelton (1999) for a discussion. Of course, this expression can only be used when \( f^*(x) \) is the average of several evaluations, not when it is, for example, the maximum. Moreover, this expression gives an indication of the confidence that one can have in the average performance of a solution. It says nothing about the actual performance of a solution, or how far this actual performance can deviate from the expected average.

*The consistency of the solution:* When repeatedly using the same solution procedure with the same number of evaluations, the solution found to be the most robust should always be approximately the same. Another indication of a sufficient number of evaluations is that all solutions that receive a good
Table 6 Three best solutions for cap132 according to robust evaluation function

<table>
<thead>
<tr>
<th>Open facilities</th>
<th>$f^*(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 4 7 8 9 11 12 15 17 23 25 27 30 31 34 42 45 50</td>
<td>479720.426</td>
</tr>
<tr>
<td>1 4 7 8 9 11 12 15 17 23 25 27 31 34 40 42 50</td>
<td>485483.452</td>
</tr>
<tr>
<td>1 4 7 8 9 10 11 12 15 17 23 27 30 31 34 42 45 48</td>
<td>488920.971</td>
</tr>
</tbody>
</table>

Table 7 Increase in computing times

<table>
<thead>
<tr>
<th>$m \times n$</th>
<th>Avg. determ. (s)</th>
<th>Avg. stoch. (s)</th>
<th>Avg. stoch/Avg. determ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 × 50</td>
<td>6.46</td>
<td>59.90</td>
<td>9.27</td>
</tr>
<tr>
<td>25 × 50</td>
<td>24.67</td>
<td>180.17</td>
<td>7.30</td>
</tr>
<tr>
<td>50 × 50</td>
<td>63.00</td>
<td>224.15</td>
<td>3.56</td>
</tr>
<tr>
<td>All</td>
<td>30.70</td>
<td>157.35</td>
<td>5.13</td>
</tr>
</tbody>
</table>

robust evaluation function value during the course of a single run, are relatively similar.

Example: As an example, Table 6 shows the three best solutions, according to the robust evaluation function for the data file cap132 with stochastic customers, $p$, being 40%. The number of evaluations per robust evaluation is equal to 100. It can be easily seen that the solutions are all very similar. This is an indication that no solution has accidentally received a favourable robust evaluation function value.

Computing times

Computing a robust evaluation function value obviously requires more time than computing an ordinary evaluation function value. The question therefore arises whether this increase in computing time is prohibitive. In this section, we show that for the experiments in this paper, the increase in computing time is rather modest.

Of course, the increase in computing time is very dependent on the type of repair function used. In the experiments with stochastic cost and no repair function, average computing time increased from 30.70 s in the deterministic case to 44.72 s in the stochastic case. This increase of about 50% can be considered insignificant.

When a repair function is introduced, computing a robust evaluation function value requires calculating $n_e$ transportation problems, where $n_e$ is the number of evaluations per robust evaluation.

The average computing times of the experiments with stochastic demand are given in Table 3. The global average computing time over all experiments is 152.18 s. In Table 7, computing times are reported for different instance sizes (experiments with limits 100% from the deterministic demand). It can be seen that the computing times in the stochastic case are on average about five times those in the deterministic case. Good news is that the computing time increase does not rise with the problem size. On the contrary, the computing time increase is lower for larger problems. This is due to the fact that, for larger problems, the algorithm spends a relatively large fraction of time evaluating moves. As the time to evaluate a move remains the same (the ordinary evaluation function is used for this), the computing time increase is smaller for larger problems.

The main reason for the relatively limited rise in computing times is the fact that the robust evaluation function is only used for evaluation purposes, that is, it is not used to guide the search. For the tabu search procedure, this means that the ordinary evaluation function is used to determine which move should be made by the steepest descent mechanism.

Using the robust evaluation function to determine the best move is possible but time-consuming. Tests show that total computing time rises approximately with a factor $n_e$. This is not surprising, considering the fact that most of the computing time is spent evaluating solutions, even in the deterministic case, and that a single robust evaluation requires $n_e$ ordinary evaluations. Using the robust evaluation function to determine the best move and using 100 evaluations per robust evaluation would increase the average computing time for a single problem to over 30 min.

It should be noted, however, that facility location is almost always an important and costly decision, which warrants the most careful computational treatment. In real-life situations, there is no need to perform a multitude of experiments on different data sets, and as a result all computational resources can be focused exclusively on robustly optimizing a single problem. In such situations, it becomes interesting to use the robust evaluation function to guide the search, regardless of the increase in computation time that this causes. Moreover, several of the efficient solutions produced by the algorithm should be examined in detail.

In the experiments with stochastic customers, computing times are dependent on the probability that each customer is present. A lower probability for each customer to be present results in a lower computational effort to calculate the robust evaluation function value. This is a result of the fact that a reduced amount of customers also causes a reduced effort to calculate the transportation problems.

Conclusion

In this paper, we have developed an approach to find robust and flexible solutions for several different stochastic versions of the capacitated FLP. Our approach combines Monte Carlo sampling with a simple tabu search algorithm for the deterministic CFLP. Using various experiments, it has been shown...
how robust and flexible solutions can be found using the concepts of robust evaluation function and repair function. We discussed some of the advantages over more traditional methods of stochastic programming. We also discussed how a sensible number of evaluations can be determined and showed that the approach can be applied without prohibitive computing time increases.

References


Received August 2004; accepted July 2006 after three revisions