Stochastic *p*-Robust Location Problems

Lawrence V. Snyder
Lehigh University

Mark S. Daskin
Northwestern University

Report No. 04T-014

Technical Report
Industrial and Systems Engineering Department, Lehigh University

Lawrence V. Snyder · Mark S. Daskin

Stochastic p-Robust Location Problems

July, 2004

Abstract. Many objectives have been proposed for optimization under uncertainty. The typical stochastic programming objective of minimizing expected cost may yield solutions that are inexpensive in the long run but perform poorly under certain realizations of the random data. On the other hand, the typical robust optimization objective of minimizing maximum cost or regret tends to be overly conservative, planning against a disastrous but unlikely scenario. In this paper, we present facility location models that combine the two objectives by minimizing the expected cost while bounding the relative regret in each scenario. In particular, the models seek the minimum-expected-cost solution that is p-robust; i.e., whose relative regret is no more than $100p\%$ in each scenario.

We present p-robust models based on two classical facility location problems, the P-median problem and the uncapacitated fixed-charge location problem. We solve both problems using variable splitting (Lagrangian decomposition), in which the subproblem reduces to the multiple-choice knapsack problem. Feasible solutions are found using an upper-bounding heuristic. For many instances of the problems, finding a feasible solution, and even determining whether the instance is feasible, is difficult; we discuss a mechanism for determining infeasibility. We also show how the algorithm can be used as a heuristic to solve minimax-regret versions of the location problems.

1. Introduction

Optimization under uncertainty typically employs one of two approaches, stochastic optimization or robust optimization. In stochastic optimization, random parameters are governed by a probability distribution, and the objective is to find a solution that minimizes the expected cost. In robust optimization, probabilities are not known, and uncertain parameters are specified either by discrete scenarios or by continuous ranges; the objective is to minimize the worst-case cost or regret. (Though these approaches are the most common, many other approaches have been proposed for both stochastic and robust optimization; for a review of their application to facility location problems, see [31,40].) The former approach finds solutions that perform well in the long run, but many decision makers are evaluated *ex post*, after the uncertainty has been resolved and costs have been realized; in such situations, decision makers are often motivated to seek minimax regret solutions that appear effective no matter which scenario is realized. On the other hand, minimax objectives tend to be overly pessimistic, planning against a scenario that, while it may be disastrous, is unlikely to occur.

Lawrence V. Snyder: Lehigh University, Dept. of Industrial and Systems Engineering, 200 West Packer Ave., Bethlehem, PA 18015 e-mail: `larry.snyder@lehigh.edu`

Mark S. Daskin: Northwestern University, Department of Industrial Engineering and Management Sciences, 2145 Sheridan Road, MEAS C210, Evanston, IL 60208 e-mail: `m-daskin@northwestern.edu`

In this paper, we discuss facility location models that combine the advantages of both the stochastic and robust optimization approaches by seeking the least-cost solution (in the expected value) that is p-robust; i.e., whose relative regret in each scenario is no more than p, for given $p \geq 0$. This robustness measure will be referred to as *stochastic p-robustness.* The standard min-expected-cost objective can be obtained by setting $p = \infty$. The resulting solution is optimal in the stochastic optimization sense, but may perform poorly in certain scenarios. By successively reducing p, one obtains solutions with smaller maximum regret but greater expected cost. One objective of this paper is to demonstrate empirically that substantial improvements in robustness can be attained without large increases in expected cost.

We consider two classical facility location problems, the P-median problem (PMP) and the uncapacitated fixed-charge location problem (UFLP). The problems we formulate will be referred to as the stochastic p-robust PMP (p-SPMP) and the stochastic p-robust UFLP (p-SUFLP), respectively.[1] Both customer demands and transportation costs may be uncertain; the possible values of the parameters are specified by discrete scenarios, each of which has a specific probability of occurrence. The p-SPMP and p-SUFLP are two-stage models, in that strategic decisions (facility location) must be made now, before it is known which scenario will come to pass, while tactical decisions (assignment of retailers to DCs) are made in the future, after the uncertainty has been resolved. However, by simply multiplying the results by the number of time periods in the planning horizon, one can think of these as multi-period models in which we make strategic decisions now and then make separate tactical decisions in each time period. Thus our models can accommodate parameters that are not time stationary. If we make decisions now that hedge poorly against the various scenarios, we must live with the consequences as parameters change.

We solve both problems using a variable-splitting (or Lagrangian decomposition) formulation whose subproblems reduce to instances of the multiple-choice knapsack problem (MCKP), a variant of the classical knapsack problem. Feasible solutions are found using a simple upper-bounding heuristic. For small values of p, it may be difficult to find a feasible solution, or even to determine whether the problem is feasible. We provide an upper bound that is valid for any feasible solution and use this bound to provide a mechanism for detecting infeasibility early in the algorithm's execution. By solving the problem iteratively with multiple values of p, one can solve minimax-regret versions of the PMP and UFLP. Since our algorithm cannot determine the feasibility of every problem within a given time or iteration limit, this method is a heuristic.

This paper makes the following contributions to the literature. First, we introduce a new measure for optimization under uncertainty (stochastic p-robustness) that requires a given level of performance in each scenario while also minimizing the expected cost. Second, we formulate and solve stochastic p-robust versions of the PMP and UFLP. Our solution method is based on well known methods, but the use of the MCKP in the Lagrangian subproblems is novel, as is the infeasibility test. Third, we provide a heuristic for solving minimax-regret versions of the PMP and UFLP; to date, these problems have been solved optimally for

[1] Throughout this paper, we will use lowercase p to denote the robustness parameter and uppercase P to denote the number of facilities to locate. Though the notation may be somewhat confusing, we have maintained the symbols from two earlier streams of research for the sake of consistency.

special instances (e.g., on tree networks) but only heuristically in the more general case. Our heuristic has the advantages of providing both upper and lower bounds on the optimal minimax regret value and of guaranteeing a given level of accuracy, though running times may be long for small optimality tolerances.

The remainder of this paper is structured as follows. In Section 2, we formally define stochastic p-robustness and review the relevant literature. We formulate the p-SPMP and present a variable-splitting algorithm for it in Section 3 and do the same for the p-SUFLP in Section 4. In Section 5, we discuss a mechanism for detecting infeasibility. The minimax-regret heuristic is presented in Section 6. Section 7 contains computational results, and Section 8 contains our conclusions.

2. Preliminaries and Literature Review

The P-median problem (PMP; [22,23]) and the uncapacitated fixed-charge location problem (UFLP; [4]) are classical facility location problems; for a textbook treatment see [14]. Sheppard [36] was one of the first authors to propose a scenario approach to facility location, in 1974. He suggests selecting facility locations to minimize the expected cost, though he does not discuss the issue at length. Since then, the literature on facility location under uncertainty has become substantial; we provide only a brief overview here and refer the reader to [6,12,26,31,40] for more detailed reviews.

Weaver and Church [42] present a Lagrangian relaxation algorithm for a stochastic version of the PMP that is equivalent to our model with p set to ∞. Their basic strategy is to treat the problem as a deterministic problem with $|I||S|$ customers instead of $|I|$ (where I is the set of customers and S is the set of scenarios) and then to apply a standard Lagrangian relaxation method for it [11,18]. Mirchandani, Oudjit, and Wong [27] begin with the same formulation as [42], explicitly reformulating it as a deterministic PMP with $|I||S|$ customers. They also suggest a Lagrangian relaxation method, but they relax a different constraint to attain a problem that is equivalent to the UFLP. Their method provides tight bounds and requires only a small number of iterations, though each iteration is computationally expensive.

The two most common measures of robustness that have been applied to facility location problems in the literature are minimax cost (minimize the maximum cost across scenarios) and minimax regret (minimize maximum regret across scenarios). The regret of a solution in a given scenario is the difference between the cost of the solution in that scenario and the cost of the optimal solution for that scenario. Robust optimization problems tend to be more difficult to solve than stochastic optimization problems because of their minimax structure. As a result, research on robust facility location generally falls into one of two categories: optimal (often polynomial-time) algorithms for restricted problems like 1-median problems or P-medians on tree networks [3,7,8,41], or heuristics for more general problems [13,33,34].

To introduce the robustness measure we use in this paper, let S be a set of scenarios. Let (P_s) be a deterministic (i.e., single-scenario) optimization problem, indexed by the scenario index s. (That is, for each scenario $s \in S$, there is a different problem (P_s). The structure of these problems is identical; only the data are different.) For each s, let z_s^* be the optimal objective value for (P_s).

Definition 1. *Let $p \geq 0$ be a constant. Let X be a feasible solution to (P_s) for all $s \in S$, and let $z_s(X)$ be the objective value of problem (P_s) under solution X. X is called* p-robust *if for all $s \in S$,*

$$\frac{z_s(X) - z_s^*}{z_s^*} \leq p \tag{1}$$

or, equivalently,

$$z_s(X) \leq (1+p)z_s^*. \tag{2}$$

The left-hand side of (1) is the relative regret for scenario s; the absolute regret is given by $z_s(X) - z_s^*$. In general, the same solution will not minimize the maximum relative regret, the maximum absolute regret, and the maximum cost, though the three objectives are closely related (absolute regret equals relative regret times a constant, and absolute regret equals cost minus a constant). Our definition of p-robustness considers relative regret, but it could easily be modified to consider absolute regret or cost by a suitable change in the right-hand sides of the robustness constraints in the formulations that follow. Thus, without loss of generality we consider only the relative regret case, and we use the term "regret" to refer to relative regret.

The notion of p-robustness was first introduced in the context of facility layout [24] and used subsequently in the context of an international sourcing problem [20] and a network design problem [21]. All three problems are also discussed in Kouvelis and Yu's book on robust optimization [25]. These papers do not refer to this robustness measure as p-robustness, but simply as "robustness." We will adopt the term "p-robust" to distinguish this robustness measure from others. The international sourcing paper [20] presents an algorithm that, for a given p and N, returns either all p-robust solutions (if there are fewer than N of them) or the N solutions with smallest maximum regret. The sourcing problem reduces to the UFLP, so the authors are essentially solving a p-robust version of the UFLP, but unfortunately their algorithm contains an error that makes it return incomplete, and in some cases incorrect, results [38].

It is sometimes convenient to specify a different regret limit p_s for each scenario s—for example, to allow low-probability scenarios to have larger regrets. In this case, one simply replaces p with p_s in (1) and (2). Denoting by $\mathbf{p}$ the vector $(p_1, \ldots, p_{|S|})$, we say a solution is $\mathbf{p}$-robust if the modified inequalities hold for all s. For the sake of convenience, we will assume throughout this paper that $p_s = p$ for all $s \in S$, though all of our results can easily be extended to the more general case.

The p-robustness measure can be combined with a min-expected-cost objective function to obtain problems of the following form:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{s \in S} q_s z_s(X) & \\ \text{subject to} & z_s(X) \leq (1+p)z_s^* & \forall s \in S \\ & X \in \mathbb{X} & \end{array}$$

where q_s is the probability that scenario s occurs and $\mathbb{X}$ is the set of solutions that are feasible for all (P_s). We call the resulting robustness measure *stochastic p-robustness.*

3. The p-SPMP

In this section, we formulate the stochastic p-robust P-median problem (p-SPMP) and then present a variable-splitting algorithm for it.

3.1. Notation

We define the following notation:

Sets

I = set of customers, indexed by i

J = set of potential facility sites, indexed by j

S = set of scenarios, indexed by s

Parameters

h_{is} = demand of customer i under scenario s

d_{ijs} = cost to transport one unit from facility j to customer i under scenarios s

p = desired robustness level, $p \geq 0$

z_s^* = optimal objective value of the PMP under the data from scenario s

q_s = probability that scenario s occurs

P = number of facilities to locate

Decision Variables

X_j = 1 if facility j is opened, 0 otherwise

Y_{ijs} = 1 if facility j serves customer i in scenario s, 0 otherwise

The robustness coefficient p is the maximum allowable regret. The regret is computed using z_s^*, which is an input to the model; the z_s^* values are assumed to have been computed already by solving $|S|$ separate (deterministic) PMPs. Such problems can be solved to optimality using a variety of well known techniques, including Lagrangian relaxation embedded in branch and bound [14, 15]. Note that the assignment variables (Y) are dependent on s while the location variables (X) are not, reflecting the two-stage nature of the problem.

3.2. Formulation

The p-SPMP is formulated as follows:

$$(p\text{-SPMP}) \quad \text{minimize} \quad \sum_{s\in S}\sum_{i\in I}\sum_{j\in J} q_s h_{is} d_{ijs} Y_{ijs} \tag{3}$$

$$\text{subject to} \quad \sum_{j \in J} Y_{ijs} = 1 \qquad \forall i \in I, \forall s \in S \tag{4}$$

$$Y_{ijs} \leq X_j \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{5}$$

$$\sum_{i \in I} \sum_{j \in J} h_{is} d_{ijs} Y_{ijs} \leq (1+p) z_s^* \qquad \forall s \in S \tag{6}$$

$$\sum_{j \in J} X_j = P \tag{7}$$

$$X_j \in \{0,1\} \qquad \forall j \in J \tag{8}$$

$$Y_{ijs} \in \{0,1\} \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{9}$$

The objective function (3) minimizes the expected transportation cost over all scenarios. Constraints (4) ensure that each customer is assigned to a facility in each scenario. Constraints (5) stipulate that these assignments be to open facilities. Constraints (6) enforce the p-robustness condition, while (7) states that P facilities are to be located. Constraints (8) and (9) are standard integrality constraints. Note that, as in the classical PMP, the optimal Y values will be integer even if the integrality constraints (9) are non-negativity constraints. We retain the integrality constraints both to emphasize the binary nature of these variables and because the subproblems formulated below do not have the integrality property; requiring the Y variables to be binary therefore tightens the relaxation.

If $p = \infty$, this formulation is equivalent to the model studied by [27,42] since constraints (6) become inactive.

3.3. Complexity

Proposition 1 below says that the p-SPMP is NP-hard; this should not be surprising since it is an extension of the PMP, which is itself NP-hard [17]. What may be surprising, however, is that the feasibility problem—determining whether a given instance of the p-SPMP is feasible—is NP-complete. This is not true of either the PMP or its usual stochastic or robust extensions (finding a min-expected-cost or minimax-regret solution, respectively), since any feasible solution for the PMP is also feasible for these extensions. Similarly, the feasibility question is easy for the capacitated PMP since it can be answered by determining whether the sum of the capacities of the P largest facilities exceeds the total demand. One would like some mechanism for answering the feasibility question for the p-SPMP in practice to avoid spending too much time attempting to solve an infeasible problem. This issue is discussed further in Section 5.

Proposition 1. *The p-SPMP is NP-hard.*

Proof. If $|S| = 1$ and $p = \infty$, the p-SPMP reduces to the classical PMP, which is NP-hard. □

Proposition 2. *For $|S| \geq 2$, the feasibility problem for the p-SPMP is NP-complete.*

Proof. First, the feasibility problem is in NP since the feasibility of a given solution can be checked in polynomial time. To prove NP-completeness, we reduce the 0–1 knapsack problem to p-SPMP; the 0–1 knapsack problem is NP-complete [17]. Let the knapsack instance be given by

$$\begin{aligned} \text{(KP)} \qquad \text{maximize} \quad & \sum_{k=1}^{n} a_k Z_k \\ \text{subject to} \quad & \sum_{k=1}^{n} b_k Z_k \leq B \\ & Z_k \in \{0,1\} \qquad \forall k = 1, \ldots, n \end{aligned}$$

The decision question is whether there exists a feasible Z such that

$$\sum_{k=1}^{n} a_k Z_k \geq A. \tag{10}$$

We can assume WLOG that $\sum_{k=1}^{n} a_k \geq A$, otherwise the problem is clearly infeasible.

We create an equivalent instance of the p-SPMP as follows. Let $I = J = \{1, \ldots, 2n\}$ and $S = \{1, 2\}$. Transportation costs are given by

$$c_{ijs} = \begin{cases} 1, & \text{if } j = i \pm n \\ \infty, & \text{otherwise} \end{cases} \tag{11}$$

for all $i, j = 1, \ldots, 2n$ and $s = 1, 2$. Demands are given by

$$(h_{i1}, h_{i2}) = \begin{cases} (a_i + \epsilon, \epsilon), & \text{if } 1 \leq i \leq n \\ (\epsilon, b_i \frac{\bar{A}}{B} + \epsilon), & \text{if } n+1 \leq i \leq 2n \end{cases} \tag{12}$$

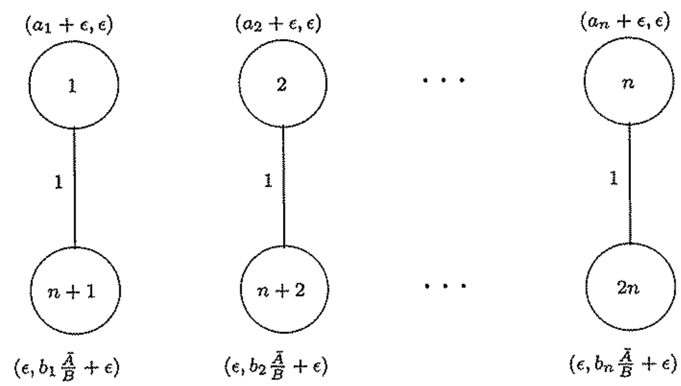
where $0 < \epsilon < \min\{\min_{k=1,\ldots,n}\{a_k\}, \min_{k=1,\ldots,n}\{b_k\}\}$ and $\bar{A} = \sum_{k=1}^{n} a_k - A$. Let $P = n$ and $p = \bar{A}/z_1^*$.

The resulting instance is pictured in Figure 1. The optimal solution in scenario 1 is to set $X_1 = \ldots = X_n = 1$ and $X_{n+1} = \ldots = X_{2n} = 0$, with a total cost of $z_1^* = n\epsilon$. Similarly, the optimal cost in scenario 2 is $z_2^* = n\epsilon$. We prove that the p-SPMP instance is feasible if and only if the knapsack instance has a feasible solution for which (10) holds; therefore, the polynomial answerability of the feasibility question for the p-SPMP implies that of the decision question posed for the 0–1 knapsack problem.

($\Rightarrow$) Suppose that the given instance of the p-SPMP is feasible, and let (X, Y) be a feasible solution. Clearly $X_i + X_{i+n} = 1$ for all $i = 1, \ldots, n$ since the cost of serving i or $i + n$ from a facility other than i or $i + n$ is prohibitive. Let

$$Z_k = \begin{cases} 1, & \text{if } X_k = 1 \\ 0, & \text{if } X_{k+n} = 1 \end{cases} \tag{13}$$

Fig. 1. p-SPMP instance.



for $k = 1, \dots, n$. Since (X, Y) is feasible, it satisfies constraint (6) for scenario 1, namely

$$\begin{aligned} & \sum_{i=1}^{2n}\sum_{j=1}^{2n} h_{i1}c_{ij1}Y_{ij1} \leq (1+p)z_1^* \\ \implies & \sum_{i:X_{i+n}=1} (a_i+\epsilon) + \sum_{i:X_i=1} \epsilon \leq \bar{A} + z_1^* \\ \implies & \sum_{i:X_{i+n}=1} a_i + n\epsilon \leq \sum_{k=1}^{n} a_k - A + n\epsilon \\ \implies & \sum_{k=1}^{n} a_k(1-Z_k) \leq \sum_{k=1}^{n} a_k - A \\ \implies & \sum_{k=1}^{n} a_k Z_k \geq A \end{aligned} \tag{14}$$

In addition, (X, Y) satisfies constraint (6) for scenario 2:

$$\begin{aligned} & \sum_{i=1}^{2n}\sum_{j=1}^{2n} h_{i2}c_{ij2}Y_{ij2} \leq (1+p)z_2^* \\ \implies & \sum_{i:X_i=1} \left(b_i\frac{\bar{A}}{B}+\epsilon\right) + \sum_{i:X_{i+n}=1} \epsilon \leq \bar{A} + z_2^* \\ \implies & \sum_{i:X_i=1} b_k\frac{\bar{A}}{B} + n\epsilon \leq \bar{A} + n\epsilon \\ \implies & \sum_{k=1}^{n} b_k\frac{\bar{A}}{B}Z_k \leq \bar{A} \\ \implies & \sum_{k=1}^{n} b_k Z_k \leq B \end{aligned} \tag{15}$$

Therefore the knapsack instance is satisfied; i.e., there exists a feasible Z whose objective function is greater than or equal to A.

($\Leftarrow$) Now suppose that Z is a feasible solution to the knapsack instance that satisfies (10). Define a solution to the p-SPMP by

$$X_i = \begin{cases} 1, & \text{if } Z_i = 1 \\ 0, & \text{if } Z_i = 0 \end{cases}$$

$$X_{i+n} = 1 - X_i$$

for $k = 1, \dots, n$, and

$$Y_{ijs} = \begin{cases} 1, & \text{if } 1 \leq i \leq n,\ j \in \{i, i+n\}, \text{ and } X_j = 1 \\ 1, & \text{if } n+1 \leq i \leq 2n,\ j \in \{i, i-n\}, \text{ and } X_j = 1 \\ 0, & \text{otherwise} \end{cases}$$

for $i, j = 1, \dots, 2n$, $s = 1, 2$.

Clearly (X, Y) satisfies constraints (4), (5), and (7)–(9). The feasibility of (X, Y) with respect to (6) follows from reversing the implications in (14) and (15). Therefore the p-SPMP instance is feasible, completing the proof. □

Note that Proposition 2 only applies to problems in which $|S| \geq 2$. If $|S| = 1$, then for $p \geq 0$ the problem is trivially feasible by definition since the left-hand side of (6) is the PMP objective function and the right-hand side is greater than or equal to its optimal objective value.

3.4. Variable-Splitting Algorithm

If $p = \infty$, constraints (6) are inactive and the p-SPMP can be solved by relaxing constraints (4); Weaver and Church [42] show that the resulting subproblem is easy to solve since it separates by j and s. But when $p < \infty$, the subproblem obtained using this method is not separable since constraints (6) tie the facilities j together. Instead, we propose an algorithm based on variable splitting.

Variable splitting is used in [5] to solve the capacitated facility location problem (CFLP). The method involves the introduction of a new set of variables to mirror variables already in the model, then forcing the two sets of variables to equal each other via a new set of constraints. These constraints are then relaxed using Lagrangian relaxation to obtain two separate subproblems. Each set of constraints in the original model is written using one set of variables or the other to obtain a particular split. The bound from variable splitting is at least as tight as that from traditional Lagrangian relaxation and is strictly tighter if neither subproblem has the integrality property [10, 19]. In our case, another attraction of variable splitting is that straightforward Lagrangian relaxation does not yield separable subproblems.

The variable-splitting formulation of (p-SPMP) is obtained by introducing a new set of variables W and setting them equal to the Y variables using a new set of constraints:

$$\begin{aligned}
(p\text{-SPMP-VS}) \quad \text{minimize} \quad & \beta \sum_{s\in S}\sum_{i\in I}\sum_{j\in J} q_s h_{is} d_{ijs} Y_{ijs} \\
& +(1-\beta)\sum_{s\in S}\sum_{i\in I}\sum_{j\in J} q_s h_{is} d_{ijs} W_{ijs} && (16)\\
\text{subject to} \quad & \sum_{j\in J} W_{ijs} = 1 \qquad \forall i\in I, \forall s\in S && (17)\\
& Y_{ijs} \le X_j \qquad \forall i\in I, \forall j\in J, \forall s\in S && (18)\\
& \sum_{i\in I}\sum_{j\in J} h_{is} d_{ijs} W_{ijs} \le (1+p) z_s^* \qquad \forall s\in S && (19)\\
& \sum_{j\in J} X_j = P && (20)\\
& W_{ijs} = Y_{ijs} \qquad \forall i\in I, \forall j\in J, \forall s\in S && (21)\\
& X_j \in \{0,1\} \qquad \forall j\in J && (22)\\
& Y_{ijs} \in \{0,1\} \qquad \forall i\in I, \forall j\in J, \forall s\in S && (23)\\
& W_{ijs} \in \{0,1\} \qquad \forall i\in I, \forall j\in J, \forall s\in S && (24)
\end{aligned}$$

The parameter $0 \le \beta \le 1$ ensures that both Y and W are included in the objective function; since $Y = W$, the objective function (16) is the same as that of (p-SPMP). In fact, the two problems are equivalent in the sense that every feasible solution to one problem has a corresponding solution to the other with equal objective value. Therefore, we can refer to the two problems interchangeably in terms of solutions, objective values, and feasibility.

To solve (p-SPMP-VS), we relax constraints (21) with Lagrange multipliers λ_{ijs}. For fixed λ, the resulting subproblem decomposes into an XY-problem and a W-problem:

XY-Problem:

$$\begin{aligned}
\text{minimize} \quad & \sum_{s\in S}\sum_{i\in I}\sum_{j\in J} (\beta q_s h_{is} d_{ijs} - \lambda_{ijs}) Y_{ijs} && (25)\\
\text{subject to} \quad & (18), (20), (22), (23)
\end{aligned}$$

W-Problem:

$$\begin{aligned}
\text{minimize} \quad & \sum_{s\in S}\sum_{i\in I}\sum_{j\in J} \left[(1-\beta) q_s h_{is} d_{ijs} + \lambda_{ijs}\right] W_{ijs} && (26)\\
\text{subject to} \quad & (17), (19), (24)
\end{aligned}$$

The XY-problem is identical to the subproblem in [42]. To solve it, we compute the benefit b_j of opening facility j:

$$b_j = \sum_{s\in S}\sum_{i\in I} \min\{0, \beta q_s h_{is} d_{ijs} - \lambda_{ijs}\}. \qquad (27)$$

We set $X_j = 1$ for the P facilities with smallest b_j and set $Y_{ijs} = 1$ if $X_j = 1$ and $\beta q_s h_{is} d_{ijs} - \lambda_{ijs} < 0$.

The W-problem reduces to $|S|$ instances of the *0-1 multiple-choice knapsack problem* (MCKP), an extension of the classical 0–1 knapsack problem in which the items are partitioned into classes and exactly one item must be chosen from each class [29,37]. The MCKP does not naturally have all-integer solutions, so the W-problem does not have the integrality property. The W-problem can be formulated using the MCKP as follows. For each scenario $s \in S$, there is an instance of the MCKP. Each instance contains $|I|$ classes, each representing a retailer $i \in I$. Each class contains $|J|$ elements, each representing a facility $j \in J$. Item j in class i has objective function coefficient $(1-\beta)q_s h_{is} d_{ijs} + \lambda_{ijs}$ and constraint coefficient $h_{is} d_{ijs}$; including this item in the knapsack corresponds to assigning customer i to facility j. The right-hand side of the knapsack constraint is $(1+p)z_s^*$.

The classical 0–1 knapsack problem can be reduced to the 0–1 MCKP, so the MCKP is NP-hard. However, like the knapsack problem, effective algorithms have been published for the MCKP, including algorithms based on linear programming [2,28,32,37] and Lagrangian relaxation [1]. Any of these algorithms could be used to solve the W-problem, with two caveats. The first is that the data must first be transformed into the form required by a given algorithm (for example, transforming the sign of the coefficients, the sense of the objective function, or the direction of the knapsack inequality), but this can usually be done without difficulty. The second caveat is that either the MCKP must be solved to optimality, or, if a heuristic is used, one must be chosen that can return a *lower bound* on the optimal objective value; otherwise, the Lagrangian subproblem cannot be guaranteed to produce a lower bound for the problem at hand. If the problem is solved heuristically, one might choose to set the variables using the heuristic (upper-bound) solution, but then the lower bound used in the subgradient optimization method does not match the actual value of the solution to the Lagrangian subproblem. We have found this mismatch to lead to substantial convergence problems. A better method is to use a lower-bound *solution*, not just the lower bound itself, to set the variables. Not all heuristics that return lower bounds also return lower-bound solutions, however, so care must be taken when making decisions about which MCKP algorithm to use and how to set the variables.

Since the MCKP is NP-hard, we have elected to solve it heuristically by terminating the branch-and-bound procedure of Armstrong et al. [2] when it reaches a 0.1% optimality gap. Their method can be modified to keep track not only of the best lower bound at any point in the branch-and-bound tree, but also a solution attaining that bound. These solutions, which are generally fractional, are then used as the values of W in the Lagrangian subproblem.

Once the XY- and W-problems have been solved, the two objective values are added to obtain a lower bound on the objective function (3). An upper bound is obtained at each iteration by opening the facilities for which $X_j = 1$ in the optimal solution to (p-SPMP-VS) and assigning each customer i to the open facility that minimizes d_{ijs}. This must be done for each scenario since the transportation costs may be scenario dependent. The Lagrange multipliers are updated using subgradient optimization; the method is standard, but the implementation is slightly different than in most Lagrangian algorithms for facility location problems since the lower-bound solution may be fractional. In addition, the subgradient optimization procedure requires an upper bound to compute the step sizes, but it is possible that no feasible solution has been found; we discuss an alternate upper bound in Section 5. The Lagrangian process is terminated based on standard stopping criteria.

If the process terminates with an UB-LB gap that is larger than desired, or if no feasible solution has been found, branch and bound is used to continue the search. Branching is performed on the unfixed facility with the largest assigned demand in the best feasible solution found at the current node. If no feasible solution has been found at the current node but a feasible solution has been found elsewhere in the branch-and-bound tree, that solution is used instead. If no feasible solution has been found anywhere in the tree, an arbitrary facility is chosen for branching.

4. The p-SUFLP

In this section, we formulate the stochastic p-robust uncapacitated fixed charge location problem (p-SUFLP) and then present a variable-splitting algorithm for it.

4.1. Formulation

In addition to the notation defined in Section 3.1, let f_j be the fixed cost to open facility $j \in J$. Since facility location decisions are scenario independent, so are the fixed costs. (Strictly speaking, the fixed costs might be scenario dependent even if the location decisions are made in the first stage. The expected fixed cost for facility j is then given by $\left(\sum_{s\in S} q_s f_{js}\right) X_j = f_j X_j$, in which case the objective function below is still correct, with f_j simply interpreted as the expected fixed cost. Constraints (31) would also be modified to include f_{js} instead of f_j. These changes would require only trivial modifications to the algorithm proposed below.)

The p-SUFLP is formulated as follows:

$$(p\text{-SUFLP}) \quad \text{minimize} \quad \sum_{j\in J} f_j X_j + \sum_{s\in S}\sum_{i\in I}\sum_{j\in J} q_s h_{is} d_{ijs} Y_{ijs} \tag{28}$$

$$\text{subject to} \quad \sum_{j\in J} Y_{ijs} = 1 \qquad \forall i \in I, \forall s \in S \tag{29}$$

$$Y_{ijs} \le X_j \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{30}$$

$$\sum_{j\in J} f_j X_j + \sum_{i\in I}\sum_{j\in J} h_{is} d_{ijs} Y_{ijs} \le (1+p) z_s^* \qquad \forall s \in S \tag{31}$$

$$X_j \in \{0,1\} \qquad \forall j \in J \tag{32}$$

$$Y_{ijs} \in \{0,1\} \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{33}$$

If $|S| = 1$ and $p = \infty$, this problem reduces to the classical UFLP, so the p-SUFLP is NP-hard. In addition, the feasibility problem for the p-SUFLP is NP-complete; the proof is similar to that of Proposition 2.

4.2. Variable-Splitting Algorithm

The p-SUFLP, too, can be solved using variable splitting, but both the X and Y variables must be split, otherwise the X variables would be contained in both the Y-problem and the W-problem. In particular, we use Z as a doubling variable for X; we also index the location variables X and Z by scenario and add a set of non-anticipativity constraints (37) that force the location decisions to be the same across all scenarios. Doing so allows the ZW-problem to decouple by scenario.

(p-SUFLP-VS)

$$\text{minimize} \quad \beta\left[\sum_{s\in S}\sum_{j\in J} q_s f_j X_{js} + \sum_{s\in S}\sum_{i\in I}\sum_{j\in J} q_s h_{is} d_{ijs} Y_{ijs}\right] + (1-\beta)\left[\sum_{s\in S}\sum_{j\in J} q_s f_j Z_{js} + \sum_{s\in S}\sum_{i\in I}\sum_{j\in J} q_s h_{is} d_{ijs} W_{ijs}\right] \tag{34}$$

$$\text{subject to} \quad \sum_{j\in J} W_{ijs} = 1 \qquad \forall i \in I, \forall s \in S \tag{35}$$

$$Y_{ijs} \le X_{js} \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{36}$$

$$X_{js} = X_{jt} \qquad \forall j \in J, \forall s \in S, \forall t \in S \tag{37}$$

$$\sum_{j\in J} f_j Z_{js} + \sum_{i\in I}\sum_{j\in J} h_{is} d_{ijs} W_{ijs} \le (1+p) z_s^* \qquad \forall s \in S \tag{38}$$

$$Z_{js} = X_{js} \qquad \forall j \in J, \forall s \in S \tag{39}$$

$$W_{ijs} = Y_{ijs} \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{40}$$

$$X_{js} \in \{0,1\} \qquad \forall j \in J, \forall s \in S \tag{41}$$

$$Z_{js} \in \{0,1\} \qquad \forall j \in J, \forall s \in S \tag{42}$$

$$Y_{ijs} \in \{0,1\} \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{43}$$

$$W_{ijs} \in \{0,1\} \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{44}$$

Relaxing constraints (39) and (40) with Lagrange multipliers λ and μ, respectively, we obtain a Lagrangian subproblem that decomposes into an XY-problem and a ZW-problem:

XY-Problem:

$$\text{minimize} \quad \sum_{s\in S}\sum_{j\in J} (\beta q_s f_j - \mu_{js}) X_{js} + \sum_{s\in S}\sum_{i\in I}\sum_{j\in J} (\beta q_s h_{is} d_{ijs} - \lambda_{ijs}) Y_{ijs} \tag{45}$$

subject to (36), (37), (41), (43)

ZW**-Problem:**

$$\text{minimize} \quad \sum_{s\in S}\sum_{j\in J}\left[(1-\beta)q_s f_j + \mu_{js}\right] Z_{js} + \sum_{s\in S}\sum_{i\in I}\sum_{j\in J}\left[(1-\beta)q_s h_{is} d_{ijs} + \lambda_{ijs}\right] W_{ijs} \tag{46}$$

$$\text{subject to} \quad (35), (38), (42), (44)$$

The XY-problem can be solved by computing the benefit of opening facility j:

$$b_j = \sum_{s\in S}(\beta q_s f_j - \mu_{js}) + \sum_{s\in S}\sum_{i\in I}\min\{0, \beta q_s h_{is} d_{ijs} - \lambda_{ijs}\}. \tag{47}$$

We set $X_{js} = 1$ for all $s \in S$ (or, equivalently, set $X_j = 1$ in the original problem) if $b_j < 0$, or if $b_k \geq 0$ for all k but is smallest for j. We set $Y_{ijs} = 1$ if $X_{js} = 1$ and $\beta q_s h_{is} d_{ijs} - \lambda_{ijs} < 0$.

The ZW-problem reduces to $|S|$ MCKP instances, one for each scenario. As in the p-SPMP, there is a class for each customer i, each containing an item for each facility j, representing the assignments W_{ijs}; these items have objective function coefficient $(1-\beta)q_s h_{is} d_{ijs} + \lambda_{ijs}$ and constraint coefficient $h_{is}d_{ijs}$. In addition, there is a class for each facility j, representing the location decisions Z_{js}; these classes contain two items each, one with objective function coefficient $(1-\beta)q_s f_j + \mu_{js}$ and constraint coefficient f_j, representing opening the facility, and one with objective function and constraint coefficient equal to 0, representing not opening the facility. The right-hand side of the knapsack constraint equals $(1+p)z_s^*$.

We note that the p-SUFLP had even greater convergence problems than the p-SPMP did when an upper-bound solution from the MCKP heuristic was used to set the variables, rather than a lower-bound solution. This makes the selection of an MCKP algorithm a critical issue for this problem. The approach outlined in Section 3.4 seems to work quite well.

The upper-bounding, subgradient optimization, and branch-and-bound procedures are as described above for the p-SPMP.

5. Detecting Infeasibility

In this section, we focus our discussion on the p-SPMP for convenience, but all of the results are easily duplicated for the p-SUFLP.

5.1. An A Priori Upper Bound

The subgradient optimization procedure used to update the Lagrange multipliers requires an upper bound (call it UB). Typically, UB is the objective value of the best known feasible solution, but Proposition 2 suggests that finding a feasible solution for the p-SPMP (or p-SUFLP) is difficult, and there is no guarantee that one will be found immediately or at all, even if one exists. We need a surrogate upper bound to use in the step size calculation. Let

$$Q = \sum_{s \in S} q_s (1+p) z_s^*. \tag{48}$$

Proposition 3. *If (p-SPMP) is feasible, then Q is an upper bound on its optimal objective value.*

Proof. Let (X^*, Y^*) be an optimal solution for (p-SPMP) . The objective value under solution (X^*, Y^*) is

$$\sum_{s \in S} q_s z_s (X^*, Y^*) \leq \sum_{s \in S} q_s (1+p) z_s^* = Q \tag{49}$$

by constraints (6). □

We refer to Q as the *a priori* upper bound. Proposition 3 has two important uses. First, if no feasible solution has been found as of a given iteration, we set UB $= Q$ in the step-size calculation. Second, we can use Proposition 3 to detect when the problem is infeasible. In particular, if the Lagrangian procedure and/or the branch-and-bound procedure yield a lower bound greater than Q, we can terminate the procedure and conclude that the problem is infeasible. One would like the Lagrangian procedure to yield bounds greater than Q *whenever* the problem is infeasible, providing a test for feasibility in every case. Unfortunately, there are infeasible instances for which the Lagrangian bound is less than Q. In the next section, we investigate the circumstances under which we can expect to find Lagrange multipliers that yield a bound greater than Q.

5.2. Lagrangian Unboundedness

Let (p-SLR) be the Lagrangian subproblem obtained by relaxing (21) in (p-SPMP-VS); (p-SLR) consists of both the XY-problem and the W-problem. Let $\mathcal{L}_{\lambda,\mu}$ be the objective value of (p-SLR) under given multipliers λ, μ (i.e., the sum of the optimal objective values of the XY- and W-problems). Let $\overline{(p\text{-SPMP-VS})}$ be the LP relaxation of (p-SPMP-VS). It is possible that (p-SPMP-VS) is infeasible but $\overline{(p\text{-SPMP-VS})}$ is feasible. When this is the case, we know that the optimal objective value of (p-SLR) is at least as great as that of $\overline{(p\text{-SPMP-VS})}$ (from standard Lagrangian duality theory), but we cannot say whether it is greater than Q. On the other hand, if $\overline{(p\text{-SPMP-VS})}$ is infeasible, Corrolary 1 below demonstrates that (p-SLR) is either infeasible or unbounded—i.e., if (p-SLR) is feasible, then for any $M \in \mathbb{R}$, there exist multipliers λ, μ such that $\mathcal{L}_{\lambda,\mu} > M$. We first state this result for general linear programs:

Lemma 1. *Let (P) be a linear program of the form*

$$\begin{array}{lll} \text{(P)} & \text{minimize} & cx \\ & \text{subject to} & Ax = b \\ & & Dx \leq e \\ & & x \geq 0 \end{array}$$

with $c \geq 0$, and let (LR) be the Lagrangian relaxation obtained by relaxing the constraints $Ax = b$. If (P) is infeasible, then (LR) is either infeasible or unbounded. That is, either (LR) is infeasible or for any $M \in \mathbb{R}$, there exist Lagrange multipliers λ such that the objective value of (LR) under λ is greater than M.

It is well known that for *feasible* problems, the Lagrangian dual (LR) and the LP dual behave similarly; the lemma verifies that the same intuition holds even for infeasible problems. The proof is similar to standard proofs that the Lagrangian bound is at least as great as the LP bound for integer programming problems [30] and is omitted here.

Corrolary 1. *If $(\overline{p\text{-SPMP-VS}})$ is infeasible, then (p-SLR) is either infeasible or unbounded.*

Proof. By Lemma 1, the LP relaxation of (p-SLR) is either infeasible or unbounded. (The equality in (17) can be replaced by $\geq$ and that in (20) can be replaced by $\leq$ WLOG; (17) can then be multiplied by -1 to obtain an LP in the form used in Lemma 1. The non-negativity of the objective function coefficients follows from the definitions of the parameters.) If the LP relaxation of (p-SLR) is infeasible, then (p-SLR) must be as well, and similarly, if the LP relaxation of (p-SLR) is unbounded, then (p-SLR) must be as well. Therefore (p-SLR) itself is either infeasible or unbounded. □

In most cases, if $(\overline{p\text{-SPMP-VS}})$ is infeasible, then (p-SLR) will be unbounded (not infeasible). (p-SLR) is infeasible if one of the constituent MCKPs is infeasible; since the constraints are independent of λ and μ, the problem is infeasible for any set of Lagrange multipliers and should be identified as such by the MCKP algorithm. If (p-SLR) is infeasible, then clearly (p-SPMP-VS) is infeasible since the constraints of the former problem are a subset of those of the latter. This rarely occurs, however, since most customers have a close (or even co-located) facility to which they may be assigned in the W-problem. Therefore, in most cases, if $(\overline{p\text{-SPMP-VS}})$ is infeasible, the Lagrangian is unbounded, in which case the subgradient optimization procedure should find λ and μ such that $\mathcal{L}_{\lambda,\mu} > Q$ and the algorithm can terminate with a proof of infeasibility. If $(\overline{p\text{-SPMP-VS}})$ is feasible but (p-SPMP-VS) is not, this method cannot detect infeasibility, and an extensive search of the branch-and-bound tree may be required before infeasibility can be proven.

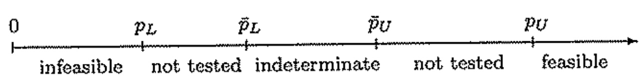
6. The Minimax Regret Heuristic

For a given optimization problem with uncertain parameters, the minimax regret problem is to find a solution that minimizes the maximum regret across all scenarios. Minimax regret is a commonly used robustness measure, especially for problems in which scenario probabilities are unknown. One can solve the minimax regret PMP heuristically using the algorithm discussed above by systematically varying p and solving (p-SPMP) for each value, as outlined below. (We focus our discussion on the (p-SPMP), but the method for the (p-SUFLP) is identical.) If scenario probabilities are unknown, q_s can be set to $1/|S|$ for all s. (p-SPMP) does not need to be solved to optimality: the algorithm can terminate as soon as a feasible solution is found for the current p. The smallest value of p for which the problem is feasible corresponds to the minimax regret value.

We introduce this method as a heuristic, rather than as an exact algorithm, because for some values of p, (p-SPMP) may be infeasible while its LP relaxation is feasible. As discussed in Section 5.2, infeasibility may not be detected by the Lagrangian method in this case, and may not be detected until a sizable portion of the branch-and-bound tree has been explored. Nevertheless, depending on the patience of the modeler, the method's level of accuracy can be adjusted as desired; this point is explicated in the discussion of step 2, below.

Our heuristic for solving the minimax regret PMP returns two values, p_L and p_U; the minimax regret is guaranteed to be in the range $(p_L, p_U]$. The heuristic also returns a solution whose maximum regret is p_U. It works by maintaining four values, $p_L \le \bar{p}_L \le \bar{p}_U \le p_U$ (see Figure 2). At any point during the execution of the heuristic, the problem is known to be infeasible for $p \le p_L$ and feasible for $p \ge p_U$; for $p \in [\bar{p}_L, \bar{p}_U]$, the problem is indeterminate (i.e., feasibility has been tested but could not be determined); and for $p \in (p_L, \bar{p}_L)$ or $(\bar{p}_U, p_U)$, feasibility has not been tested. At each iteration, a value of p is chosen in $(p_L, \bar{p}_L)$ or $(\bar{p}_U, p_U)$ (whichever range is larger), progressively reducing these ranges until they are both smaller than some pre-specified tolerance ϵ.

Fig. 2. Ranges maintained by the minimax-regret heuristic.



Algorithm 1 (MINIMAX-REGRET).

0. Determine a lower bound p_L for which (p-SPMP) is known to be infeasible and an upper bound p_U for which (p-SPMP) is known to be feasible. Let (X^*, Y^*) be a feasible solution with maximum regret p_U. Mark $\bar{p}_L$ and $\bar{p}_U$ as undefined.
1. If $\bar{p}_L$ and $\bar{p}_U$ are undefined, let $p \leftarrow (p_L + p_U)/2$; else if $\bar{p}_L - p_L > p_U - \bar{p}_U$, let $p \leftarrow (p_L + \bar{p}_L)/2$; else, let $p \leftarrow (\bar{p}_U + p_U)/2$.
2. Determine the feasibility of (p-SPMP) under the current value of p.
   - 2.1 If (p-SPMP) is feasible, let $p_U \leftarrow$ the maximum relative regret of the solution found, let (X^*, Y^*) be the solution found in step 2, and go to step 3.
   - 2.2 Else if (p-SPMP) is infeasible, let $p_L \leftarrow p$ and go to step 3.
   - 2.3 Else [(p-SPMP) is indeterminate]: If $\bar{p}_L$ and $\bar{p}_U$ are undefined, let $\bar{p}_L \leftarrow p$ and $\bar{p}_U \leftarrow p$ and mark $\bar{p}_L$ and $\bar{p}_U$ as defined; else if $p \in (p_L, \bar{p}_L)$, let $\bar{p}_L \leftarrow p$; else [$p \in (\bar{p}_U, p_U)$], let $\bar{p}_U \leftarrow p$. Go to step 3.
3. If $\bar{p}_L - p_L < \epsilon$ and $p_U - \bar{p}_U < \epsilon$, stop and return p_L, p_U, (X^*, Y^*). Else, go to step 2.

Several comments are in order. In step 0, the lower bound p_L can be determined by choosing a small enough value that the problem is known to be infeasible (e.g., 0). The upper bound can be determined by solving (p-SPMP) with $p = \infty$ and setting p_U equal to the maximum regret value from the solution found;

this solution can also be used as (X^*, Y^*). In step 1, we are performing a binary search on each region. More efficient line searches, such as the Golden Section search, would work as well, but we use the binary search for ease of exposition. In step 2, the instruction "determine the feasibility..." is to be carried out by solving (p-SPMP) until (a) a feasible solution has been found [the problem is feasible], (b) the lower bound exceeds the *a priori* upper bound Q [the problem is infeasible], or (c) a pre-specified stopping criterion has been reached [the problem is indeterminate]. This stopping criterion may be specified as a number of Lagrangian iterations, a number of branch-and-bound nodes, a time limit, or any other stopping criterion desired by the user. In general, if the stopping criterion is more generous (i.e., allows the algorithm to run longer), fewer problems will be indeterminate, and the range $(p_L, p_U]$ returned by the heuristic will be smaller. One can achieve any desired accuracy by adjusting the stopping criterion, though if a small range is desired, long run times might ensue. (Finding an appropriate stopping criterion may require some trial and error; it is not an immediate function of the desired accuracy.) Our heuristic therefore offers two important advantages over most heuristics for minimax regret problems: (1) it provides both a candidate minimax-regret solution and a lower bound on the optimal minimax regret, and (2) it can be modified to achieve any desired level of accuracy.

7. Computational Results

7.1. Algorithm Performance

We tested the variable-splitting algorithms for the p-SPMP and p-SUFLP on 10 randomly generated data sets, each with 50 nodes and 5 scenarios. All nodes serve as both customers and potential facility sites (i.e., $I = J$). In each data set, demands for scenario 1 are drawn uniformly from $[0, 10000]$ and rounded to the nearest integer, and x and y coordinates are drawn uniformly from $[0, 1]$. In scenarios 2 through 5, demands are obtained by multiplying scenario 1 demands by a random number drawn uniformly from $[0.5, 1.5]$ and rounded off; similarly, x and y coordinates are obtained by multiplying scenario 1 coordinates by a random number drawn uniformly from $[0.75, 1.25]$. (That is, scenario 1 demands are perturbed by up to 50% in either direction, coordinates by up to 25%. This means that nodes may lie outside the unit square in scenarios 2–5. The multiplier for each data element was drawn separately, so that each customer's demand and x- and y-coordinates are multiplied by different random numbers.) Per-unit transportation costs are set equal to the Euclidean distances between facilities and customers. Fixed costs for the p-SUFLP are drawn uniformly from $[4000, 8000]$ and rounded to the nearest integer. Scenario probabilities are set as follows: q_1 is drawn uniformly from $(0, 1)$, q_t is drawn uniformly from $\left(0, 1 - \sum_{s=1}^{t-1} q_s\right)$ for $t = 2, 3, 4$, and q_5 is set to $1 - \sum_{s=1}^{4} q_s$.

The performance measure of interest for these tests is the tightness of the bounds produced at the root node; consequently, no branching was performed. We terminated the Lagrangian procedure when the LB–UB gap was less than 0.1%, when 1200 Lagrangian iterations had elapsed, or when α (a parameter used in the subgradient optimization procedure; see [14]) was less than 10^{-8}. All Lagrange multipliers were initially

Table 1. Algorithm performance.

P	p	Gap	# Lag Iter	CPU Time	MCKP Time	Feas/Inf/Ind
5	∞	1.0%	1176.6	20.7	12.4	10/0/0
5	0.80	0.9%	1152.2	23.7	15.3	10/0/0
5	0.20	2.6%	1182.0	47.0	38.7	10/0/0
5	0.10	2.3%	1200.0	46.0	37.3	3/0/7
5	0.05	—	1058.8	35.5	28.0	0/3/7
5	0.03	—	434.2	15.7	12.4	0/8/2
5	0.01	—	312.4	11.6	9.3	0/10/0
15	∞	0.4%	934.3	16.5	9.8	10/0/0
15	0.40	1.3%	1095.6	66.2	58.1	10/0/0
15	0.30	3.5%	1200.0	77.4	68.5	10/0/0
15	0.20	4.1%	1200.0	85.9	77.3	7/0/3
15	0.10	—	127.0	16.6	15.6	0/10/0
15	0.05	—	34.7	13.8	13.4	0/10/0
15	0.01	—	24.2	11.3	11.0	0/10/0
25	∞	0.4%	709.5	12.7	7.4	10/0/0
25	0.60	1.5%	1041.0	128.9	121.1	10/0/0
25	0.40	2.3%	1098.0	244.8	236.7	10/0/0
25	0.20	—	580.0	134.8	130.4	0/6/4
25	0.10	—	48.6	45.0	44.5	0/10/0
25	0.05	—	23.1	14.1	13.7	0/10/0
25	0.01	—	18.5	19.3	18.9	0/10/0
—	∞	0.9%	1200.0	28.5	20.0	10/0/0
—	0.20	3.4%	1200.0	101.0	92.3	10/0/0
—	0.10	2.2%	1200.0	69.5	60.9	10/0/0
—	0.08	2.8%	1200.0	140.6	132.1	8/0/2
—	0.05	2.0%	1163.9	178.1	169.8	1/1/8
—	0.03	—	660.5	38.2	33.4	0/7/3
—	0.01	—	199.2	121.6	120.1	0/10/0

set to 0. The weighting coefficient β was set to 0.2. We tested our algorithms for several values of p. The algorithm was coded in C++ and executed on a Gateway Profile 4MX desktop computer with a 3.2GHz Pentium 4 processor and 1 GB of RAM.

Table 1 summarizes the algorithms' performance for both the p-SPMP and the p-SUFLP. The column marked "P" gives the number of facilities to locate for the p-SPMP or "—" for the p-SUFLP. The column marked "p" gives the robustness coefficient. "Gap" gives the average value of (UB − LB)/LB among all problems (out of 10) for which a feasible solution was found. Note that for some values of p, no feasible solution was found for any of the 10 test problems. "# Lag Iter" gives the average number of Lagrangian iterations performed, "CPU Time" gives the average time (in seconds) spent by the algorithm, and "MCKP Time" gives the average time (in seconds) spent solving multiple-choice knapsack problems. The final column lists the number of problems (out of 10) for which feasibility was proved (i.e., a feasible solution was found), the number for which infeasibility was proved (i.e., the lower bound exceeded Q), and the number which were indeterminate (i.e., no feasible solution was found but the problem could not be proven infeasible).

The algorithms generally attained gaps of no more than a few percent for feasible instances. (The maximum gap found was 17.6%; the second-highest was less than 10%). Tighter bounds may be desirable, especially if the modeler intends to find a provably optimal solution using branch and bound. These instances generally required close to the maximum allowable number of Lagrangian iterations. On the other hand, the provably infeasible instances generally required fewer than 100 iterations to prove infeasibility. The algorithm generally took only a few minutes to execute on a desktop computer, with the bulk of the time

Table 2. Minimax regret heuristic.

P	p_L	p_U	Gap	Init Gap	CPU Time
5	0.029	0.104	0.074	0.177	296.5
15	0.136	0.206	0.070	0.437	562.4
25	0.202	0.277	0.075	0.656	1758.8
—	0.025	0.063	0.039	0.106	520.5

(83% on average) spent solving multiple-choice knapsack problems. The algorithm was generally successful in proving feasibility or infeasibility, with 36 out of 280 total instances (13%) left indeterminate, generally with mid-range values of p. Corollary 1 implies that for these problems, either the LP relaxation is feasible or it is infeasible but we are not finding good enough multipliers to exceed Q. Further research is needed to establish which is the case.

7.1.1. Minimax Regret Heuristic We tested the minimax regret heuristic discussed in Section 6 for the p-SPMP and p-SUFLP using the 10 randomly generated data sets described above. As above, no branching was performed, and an iteration limit of 1200 was used (this represents the stopping criteria in step 2 of the heuristic.) The results are summarized in Table 2. The first column indicates the value of P (or "—" for the p-SUFLP instances). The columns marked "p_L" and "p_U" indicate the average (among the 10 test problems) lower and upper bounds on the minimax regret value returned by the heuristic. The column marked "Gap" gives the average difference between p_U and p_L, while the "Init Gap" column gives the average gap between the values of p_U and p_L used to initialize the heuristic in step 0. (Since p_L was initially set to 0, the "Init Gap" column equals the average maximum regret of the unconstrained problem, used to initialize p_U.) The column marked "CPU Time" indicates the average time (in seconds) spent by the heuristic.

The heuristic generally returned ranges of about 7% for the p-SPMP and half that for the p-SUFLP. As discussed above, these ranges could be reduced by using a more lenient stopping criterion—for example, using branch-and-bound after processing at the root node. It is our suspicion that p_U is closer to the true minimax regret value than p_L is (that is, the upper bound returned by the heuristic is tighter than the lower bound), though further research is needed to establish this definitively.

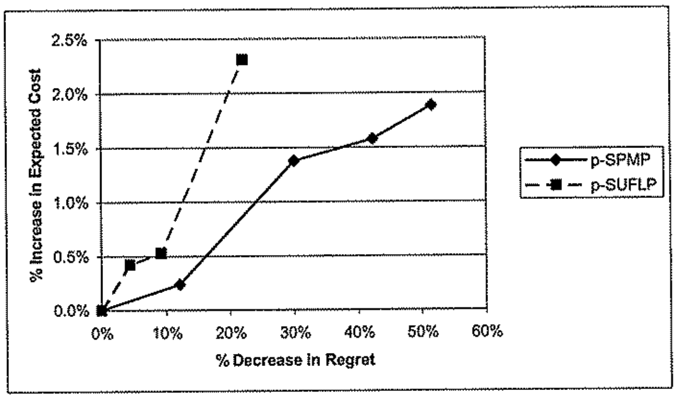
7.2. Cost vs. Regret

The main purpose of the p-SPMP and p-SUFLP is to reduce the maximum regret (by the choice of p) with as little increase in expected cost as possible. To illustrate this tradeoff, we used the constraint method of multi-objective programming [9] to generate a tradeoff curve between the expected cost and the maximum regret. In particular, we solved the problem with $p = \infty$ and recorded the objective value and maximum regret of the solution; we then set p equal to the maximum regret minus 0.00001 and re-solved the problem, continuing this process until no feasible solution could be found for a given value of p. We performed this experiment using one of the data sets described above for the p-SPMP with $P = 15$ and for the p-SUFLP. The results are summarized in Table 3. The column marked "p" gives the value of p used to solve the problem;

Table 3. Expected cost vs. maximum regret.

Problem	p	Obj Value	% Increase	Max Regret	% Decrease
p-SPMP	∞	15408.8	0.0%	0.411	0.0%
p-SPMP	0.4114	15445.4	0.2%	0.362	12.1%
p-SPMP	0.3616	15620.8	1.4%	0.288	30.0%
p-SPMP	0.2879	15651.1	1.6%	0.237	42.4%
p-SPMP	0.2371	15698.8	1.9%	0.199	51.6%
p-SUFLP	∞	70686.3	0.0%	0.095	0.0%
p-SUFLP	0.0950	70982.6	0.4%	0.091	4.4%
p-SUFLP	0.0908	71056.3	0.5%	0.086	9.3%
p-SUFLP	0.0862	72319.5	2.3%	0.074	22.1%

Fig. 3. Expected cost vs. maximum regret.



"Obj Value" is the objective value of the best feasible solution returned by the algorithm; "% Increase" is the percentage by which the objective value is greater than that of the solution found using $p = \infty$; "Max Regret" is the maximum relative regret of the best solution found; and "% Decrease" is the percentage by which the maximum regret is smaller than that of the solution found using $p = \infty$. The percentages by which the cost increases and the regret decreases are plotted in Figure 3 for both problems.

It is clear that large reductions in maximum regret are possible with small increases in expected cost. For example, the last solution for the p-SPMP represents a 52% reduction in maximum regret with less than a 2% increase in expected cost, and the third p-SUFLP solution attains nearly a 10% reduction in maximum regret with only 0.5% increase in expected cost. These results justify the stochastic p-robust approach since it costs very little to "buy" robustness. While we cannot be assured that results like these would be attained for any instance, we have found them to be representative of the general trend. Note that not all problems were solved to optimality (because no branching was performed), so it is possible that for some problems, the optimal expected costs are even smaller than those listed in the table.

8. Conclusions

In this paper we presented models that seek the minimum-expected-cost solution for two classical facility location problems, subject to the constraint that the solution chosen must have relative regret no more than p in every scenario. This robustness measure, called stochastic p-robustness, combines the advantages of traditional stochastic and robust optimization approaches. We presented algorithms for our models that use variable splitting, or Lagrangian decomposition. The Lagrangian subproblems split into two problems, one problem that can be solved by inspection and another that reduces to the multiple-choice knapsack problem. We showed that our algorithm can be used iteratively to solve the minimax-regret problem; this method is approximate, but it provides both lower and upper bounds on the minimax regret value, and it can be adjusted to provide arbitrarily close bounds. In addition, we showed empirically that large reductions in regret are possible with small increases in expected cost. Although our discussion is focused on relative regret, our models and algorithms can be readily applied to problems involving absolute regret or simply the cost in each scenario.

While the bounds provided by our algorithms are reasonable, it would be desirable to tighten them even further. Preliminary exploration of our formulations indicates that the objective values of the IPs grow much more quickly as p decreases than the objective values of their LP relaxations do. This means that for more tightly constrained problems, the LP bounds are increasingly weak. While the Lagrangian bound is necessarily greater than the LP bound (because the subproblems do not have the integrality property), it may not be great enough to provide sufficiently tight bounds. We have begun investigating the reasons for this discrepancy in the bounds and possible methods for improvement; we expect this to be an avenue for future research.

Finally, we note that [39] applies the stochastic p-robustness measure to the location model with risk-pooling (LMRP; [16,35]), a recent supply chain design model that is based on the UFLP but incorporates risk-pooling effects and economies of scale in inventory costs. The stochastic p-robust LMRP formulation does not lend itself to a variable-splitting approach because of the non-linearity of the objective function; instead, [39] solves the problem using straightforward Lagrangian relaxation. We consider the application of stochastic p-robustness to other logistics and supply chain design problems to be another avenue for future research.

9. Acknowledgments

This research was supportd by NSF Grants DMI-9634750 and DMI-9812915. This support is gratefully acknowledged.

References

1. Vijay Aggarwal, Narsingh Deo, and Dilip Sarkar. The knapsack problem with disjoint multiple-choice constraints. *Naval Research Logistics*, 39(2):213–227, 1992.
2. R. D. Armstrong, D. S. Kung, P. Sinha, and A. A. Zoltners. A computational study of a multiple-choice knapsack algorithm. *ACM Transactions on Mathematical Software*, 9(2):184–198, 1983.
3. Igor Averbakh and Oded Berman. Minmax regret median location on a network under uncertainty. *INFORMS Journal on Computing*, 12(2):104–110, 2000.
4. M. L. Balinski. Integer programming: Methods, uses, computation. *Management Science*, 12(3):253–313, 1965.
5. Jaime Barcelo, Elena Fernandez, and Kurt O. Jörnsten. Computational results from a new Lagrangean relaxation algorithm for the capacitated plant location problem. *European Journal of Operational Research*, 53(1):38–45, 1991.
6. Margaret L. Brandeau and Samuel S. Chiu. An overview of representative problems in location research. *Management Science*, 35(6):645–674, 1989.
7. Rainer E. Burkard and Helidon Dollani. Robust location problems with pos/neg weights on a tree. *Networks*, 38(2):102–113, 2001.
8. Bintong Chen and Chin-Shien Lin. Minmax-regret robust 1-median location on a tree. *Networks*, 31(2):93–103, 1998.
9. Jared L. Cohon. *Multiobjective programming and planning*. Mathematics in Science and Engineering. Academic Press, New York, 1978.
10. G. Cornuejols, R. Sridharan, and J.M. Thizy. A comparison of heuristics and relaxations for the capacitated plant location problem. *European Journal of Operational Research*, 50:280–297, 1991.
11. Gerard Cornuejols, Marshall L. Fisher, and George L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.
12. J. Current, M. S. Daskin, and D. Schilling. Discrete network location models. In Zvi Drezner and H. W. Hamacher, editors, *Facility Location: Applications and Theory*, chapter 3. Springer-Verlag, New York, 2002.
13. John Current, Samuel Ratick, and Charles ReVelle. Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach. *European Journal of Operational Research*, 110(3):597–609, 1997.
14. Mark S. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley, New York, 1995.
15. Mark S. Daskin. SITATION software, 2004. Available for download from `users.iems.northwestern.edu/~msdaskin/`.
16. Mark S. Daskin, Collette R. Coullard, and Zuo-Jun Max Shen. An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research*, 110:83–106, 2002.
17. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.
18. A.M. Geoffrion. Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2:82–114, 1974.
19. Monique Guignard and Siwhan Kim. Lagrangean decomposition: A model yielding strong Lagrangean bounds. *Mathematical Programming*, 39:215–228, 1987.
20. Genaro J. Gutiérrez and Panagiotis Kouvelis. A robustness approach to international sourcing. *Annals of Operations Research*, 59:165–193, 1995.
21. Genaro J. Gutiérrez, Panagiotis Kouvelis, and Abbas A. Kurawarwala. A robustness approach to uncapacitated network design problems. *European Journal of Operational Research*, 94:362–376, 1996.
22. S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964.
23. S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475, 1965.
24. Panagiotis Kouvelis, Abbas A. Kurawarwala, and Genaro J. Gutiérrez. Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research*, 63:287–303, 1992.
25. Panagiotis Kouvelis and Gang Yu. *Robust Discrete Optimization and its Applications*. Kluwer Academic Publishers, Boston, 1997.
26. François V. Louveaux. Stochastic location analysis. *Location Science*, 1(2):127–154, 1993.

27. Pitu B. Mirchandani, Aissa Oudjit, and Richard T. Wong. 'Multidimensional' extensions and a nested dual approach for the m-median problem. *European Journal of Operational Research*, 21(1):121–137, 1985.

28. Y. Nakagawa, M. Kitao, M. Tsuji, and Y. Teraoka. Calculating the upper bound of the multiple-choice knapsack problem. *Electronics and Communications in Japan Part 3*, 84(7):22–27, 2001.

29. R.M. Nauss. The 0-1 knapsack problem with multiple choice constraints. *European Journal of Operational Research*, 2:125–131, 1978.

30. George L. Nemhauser and Laurence A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, New York, 1988.

31. Susan Hesse Owen and Mark S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, 1998.

32. David Pisinger. A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research*, 83(2):394–410, 1995.

33. D. Serra, S. Ratick, and C. ReVelle. The maximum capture problem with uncertainty. *Environment and Planning B*, 23:49–59, 1996.

34. Daniel Serra and Vladimir Marianov. The p-median problem in a changing network: The case of Barcelona. *Location Science*, 6:383–394, 1998.

35. Zuo-Jun Max Shen, Collette R. Coullard, and Mark S. Daskin. A joint location-inventory model. *Transportation Science*, 37(1):40–55, 2003.

36. E. S. Sheppard. A conceptual framework for dynamic location-allocation analysis. *Environment and Planning A*, 6:547–564, 1974.

37. Prabhakant Sinha and Andris A. Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, 1979.

38. Lawrence V. Snyder. A note on the robust international sourcing algorithm of Gutiérrez and Kouvelis. working paper, Lehigh University, 2003.

39. Lawrence V. Snyder. *Supply Chain Robustness and Reliability: Models and Algorithms*. PhD thesis, Northwestern University, 2003.

40. Lawrence V. Snyder. Facility location under uncertainty: A review. working paper, Lehigh Unviersity, 2004.

41. George L. Vairaktarakis and Panagiotis Kouvelis. Incorporation dynamic aspects and uncertainty in 1-median location problems. *Naval Research Logistics*, 46(2):147–168, 1999.

42. Jerry R. Weaver and Richard L. Church. Computational procedures for location problems on stochastic networks. *Transportation Science*, 17(2):168–180, 1983.