



ELSEVIER

European Journal of Operational Research 117 (1999) 157–174

EUROPEAN
JOURNAL
OF OPERATIONAL
RESEARCH

www.elsevier.com/locate/orms

Theory and Methodology

A heuristic to minimax absolute regret for linear programs with interval objective function coefficients

Helmut E. Mausser^{*}, Manuel Laguna

Graduate School of Business Administration, University of Colorado at Boulder, Boulder, CO 80309-0419, USA

Received 10 March 1997; accepted 26 August 1997

Abstract

Decision makers faced with uncertain information often experience regret upon learning that an alternative action would have been preferable to the one actually selected. Models that minimize the maximum regret can be useful in such situations, especially when decisions are subject to ex post review. Of particular interest are those decision problems that can be modeled as linear programs with interval objective function coefficients. The minimax regret solution for these formulations can be found using an algorithm that, at each iteration, solves first a linear program to obtain a candidate solution and then a mixed integer program (MIP) to maximize the corresponding regret. The exact solution of the MIP is computationally expensive and becomes impractical as the problem size increases. In this paper, we develop a heuristic for the MIP and investigate its performance both alone and in combination with exact procedures. The heuristic is shown to be effective for problems that are significantly larger than those previously reported in the literature. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Probabilistic programming; Heuristics; Interval programming; Regret

1. Introduction

In an increasingly volatile business environment, managers are often faced with uncertain or imperfect information when making decisions. Since decisions based on models that ignore variability in key input data can have devastating consequences, models that can deliver plans that will perform well regardless of future eventualities are appealing. Furthermore, when decisions are subject to ex post review, managers may want to avoid being viewed as having exercised poor judgement. The maximum possible regret (i.e., the distance from optimality once uncertainty resolves) is a useful criterion for evaluating decisions in such cases. Regret can be measured in an absolute or relative sense. The former is simply the difference between

^{*} Corresponding author. Address: Algorithmics Incorporated, 185 Spadina Avenue, Toronto, Ont., Canada M5T 2C6. E-mail: hmausser@algorithmics.com

the actual and best possible outcomes, while the latter expresses this difference on a percentage basis. In this paper, we present a heuristic method to minimize maximum absolute regret for linear programs with interval objective function coefficients. An instance of such a problem might arise when constructing an investment portfolio to maximize its future value and uncertain stock prices are assumed to lie between specified upper and lower limits.

Empirical evidence suggests that the anticipation of regret can influence the choices of decision makers (see, for example, Loomes and Sugden, 1987; Ritov, 1996; Zeelenberg et al., 1996). Bell (1985) and Simonson (1992) described actual marketing strategies that targeted regret on the part of consumers. Formal theories of regret (Bell, 1982; Loomes and Sugden, 1982) have been advanced to explain observed violations of expected utility theory. Thus, while the regret criterion may be difficult to justify from a normative sense, the fact remains that regret avoidance is a demonstrated concern of decision makers and, therefore, warrants further consideration.

In the mathematical programming literature, regret has often been associated with the idea of *robustness*. Gupta and Rosenhead (1968) and Rosenhead et al. (1972) proposed the concepts of robustness and stability for evaluating sequential decisions made under uncertainty. Robustness referred to the flexibility afforded by the current choice of action, in terms of its likelihood of resulting in a desirable end-state, while stability considered performance relative to the best possible outcome (i.e., regret). Subsequent work has somewhat blurred the distinction between these two terms, removing the focus on sequential decisions, while maintaining the relevance of regret as a measure of performance. In solving a plant layout problem, Rosenblatt and Lee (1987) measured a solution's robustness by the number of demand scenarios for which the solution was within a specified percentage of optimal (i.e., preferred solutions were those most likely to satisfy some relative regret threshold). Similarly, Kouvelis et al. (1992) found robust plant layouts that were within a specified percentage of optimal for all scenarios. The requirement that a robust solution satisfies some worst-case percentage deviation from optimality was also used by Gutierrez and Kouvelis (1995) in their design of international sourcing networks. Mulvey et al. (1995) have used the term *robust optimization* to describe a general model for stochastic programming, that explicitly trades off the conflicting objectives of remaining close to feasible (model robustness) and close to optimal (solution robustness) across multiple scenarios. Another interesting example of a regret criterion in stochastic optimization is due to Barlow and Glover (1987), who introduced a variable scaling factor for relative regret.

The aforementioned research assumed that uncertainty was represented by a finite set of scenarios. Alternatively, one can replace scenarios by uncertainty intervals, which simply specify an upper and lower bound for each uncertain quantity. This model has been examined by Daniels and Kouvelis (1995) in the context of production scheduling. Their procedure finds the minimax absolute regret schedule, in terms of total flow time, for a set of jobs with uncertain, bounded processing times.

Inuiguchi and Sakawa (1995) considered minimizing maximum absolute regret for general linear programs with interval objective function coefficients (for brevity, we will refer to objective function coefficients simply as *costs*). Their approach was based on an iterative relaxation procedure developed for *min-max* problems by Shimizu and Aiyoshi (1980). Specifically, at each iteration, the algorithm first solves a relaxed linear program to obtain a candidate solution and then finds the cost vector that maximizes regret for the candidate, thereby generating a new constraint for the linear program. Obtaining the regret-maximizing costs required finding optimal solutions for all *extremal* cost vectors (i.e., those in which all uncertain costs are at a bound). Since a problem with n uncertain costs can have up to 2^n such solutions, this task quickly becomes computationally intractable. Their approach was refined by Inuiguchi and Sakawa (1996) and Mausser and Laguna (1997), who used mathematical programming to obtain the regret-maximizing costs. The complexity of the resulting mixed integer programs severely limits the size of problems that can be addressed. For example, previous work has considered problems with up to 36 uncertain costs, and a maximum size of approximately 100 variables and 100 constraints (Mausser and Laguna, 1997). Thus,

heuristics for finding the regret-maximizing costs are expected to play an important role in extending the applicability of the minimax regret criterion within the context of uncertain cost coefficients. In this paper we undertake to develop such a heuristic and investigate its performance, both alone and in combination with exact methods, for problems significantly larger and more difficult than those previously reported in the literature.

In Section 2, we first give a formal statement of the problem and review the iterative minimax regret algorithm. We then focus on the mixed integer programming formulation of Mausser and Laguna (1997) and derive a heuristic solution procedure in Section 3. Section 4 describes our computational experiments, and conclusions and possible extensions are presented in Section 5.

2. An iterative procedure to minimax regret

Consider the linear program

$$\begin{aligned} \max \quad & cx \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \tag{1}$$

where Ω is a non-empty, bounded polyhedron, and let the optimal solution to (1) be x_c . Suppose that costs are uncertain and that $c \in \Gamma = \{c \mid \underline{c}_i \leq c_i \leq \bar{c}_i, i = 1, \dots, n\}$, where \underline{c}_i and \bar{c}_i are known constants. The resulting problem is a so-called *linear program with interval objective function coefficients*. Our objective is to find an $x \in \Omega$ that minimizes the maximum absolute deviation from optimality (i.e., the regret) over all possible costs.

For any x , the maximum regret associated with cost vector c is $R(c, x) = \max_{y \in \Omega} (cy - cx)$. Let $R_{\max}(x) = \max_{c \in \Gamma} \{R(c, x)\}$ be the maximum regret for x over all possible costs. We want to find x^* satisfying $R_{\max}(x^*) \leq R_{\max}(x)$ for all $x \in \Omega$. That is, x^* is the optimal solution to the minimax regret problem (MMR):

$$\begin{aligned} \min_x \quad & \{\max_{c,y} cy - cx\} \\ \text{s.t.} \quad & x, y \in \Omega, \\ & c \in \Gamma. \end{aligned} \tag{2}$$

Our assumption that Ω and c are bounded implies that $R_{\max}(x^*)$ is finite.

MMR is a *min-max* problem that can be solved by an iterative relaxation procedure (Shimizu and Aiyoshi, 1980). Consider the relaxed linear program (MMR') obtained by replacing $c \in \Gamma$ with a finite set of cost vectors, or scenarios, $C = \{c^1, c^2, \dots, c^m\}$:

$$\begin{aligned} \min \quad & r \\ \text{s.t.} \quad & c^k x + r \geq c^k x_{c^k}, \forall c^k \in C, \\ & x \in \Omega, \\ & r \geq 0. \end{aligned} \tag{3}$$

We refer to each constraint $c^k x + r \geq c^k x_{c^k}$ associated with an element of C as a *regret cut*. Let the solution to MMR' be \hat{x} , with corresponding regret \hat{r} . Note that $\hat{r} \leq R_{\max}(x^*)$ is a lower bound for the minimax regret and this lower bound is non-decreasing as more regret cuts are added to MMR'. Since MMR' contains $(n + 1)$ variables, an optimal (non-degenerate) basic solution has $(n + 1)$ binding constraints, and it follows that at most $(n + 1)$ regret cuts are actually necessary to determine the optimal solution to MMR. However, finding the minimal set of necessary regret cuts is not a trivial task, and therefore we can typically expect MMR' to contain a number of non-binding regret cuts.

The set C of cost scenarios can be constructed iteratively. Given any candidate solution \hat{x} to MMR', the cost scenario that maximizes regret can be found by solving the following candidate maximum regret problem (CMR):

$$\begin{aligned}
R_{\max}(\hat{x}) \equiv \max \quad & cx - c\hat{x} \\
\text{s.t.} \quad & x \in \Omega, \\
& c \in \Gamma.
\end{aligned} \tag{4}$$

Note that $R_{\max}(\hat{x}) \geq R_{\max}(x^*)$ is an upper bound for MMR. Thus, using MMR' and CMR to generate candidate solutions and regret cuts, respectively, yields the following algorithm to minimize the maximum regret.

- Step 0.* (Initialization). Set $LB = 0$ and choose $\hat{x} \in \Omega$.
Step 1. (Solve CMR). Find \hat{c} and $R_{\max}(\hat{x})$. If $R_{\max}(\hat{x}) \leq LB$, then go to Step 4.
Step 2. (Add cut). Add the cut $\hat{c}x + r \geq \hat{c}\hat{x} + R_{\max}(\hat{x})$ to MMR'.
Step 3. (Solve MMR'). Find \hat{F} and \hat{x} . Set $LB = \hat{F}$. Go to Step 1.
Step 4. (End). STOP (\hat{x} minimizes maximum regret).

Ω is a non-empty, compact set defined by linear constraints. Also, $cy - cx$ is continuous in x and y , and has continuous partial derivatives. Thus, it follows from Theorem 3 of Shimizu and Aiyoshi (1980) that the above algorithm terminates in a finite number of iterations.

Solving CMR in Step 1 is the most computationally demanding part of the algorithm. Since CMR is an indefinite quadratic program, previous work has sought to exploit properties of the regret maximizing solution in order to solve CMR efficiently. Inuiguchi and Sakawa (1995) identified two such properties that are particularly relevant for our approach.

First, they showed that in (4), x can be restricted to those vertices of Ω that are optimal for some $c \in \Gamma$, or so-called *possibly optimal solutions*. We will refer to this property as *x-optimality* and note that it follows from the fact that for any $c \in \Gamma$ and $\hat{x} \in \Omega$, $R(c, \hat{x}) = \max_{x \in \Omega} (cx - c\hat{x}) = (\max_{x \in \Omega} cx) - c\hat{x}$ implies $x = x_c$. Second, c can be chosen from among extremal cost vectors. Furthermore, the regret-maximizing cost vector \hat{c} satisfies the following condition, which we call *c-consistency*:

$$\hat{c}_i = \begin{cases} \underline{c}_i & \text{if } (x_{\hat{c}})_i < \hat{x}_i, \\ \bar{c}_i & \text{if } (x_{\hat{c}})_i > \hat{x}_i. \end{cases}$$

The method of Inuiguchi and Sakawa (1995) first constructs the set E of all possibly optimal solutions. Then, to solve CMR for a given \hat{x} , *c-consistency* is used to quickly calculate the regret for each element of E ($R_{\max}(\hat{x})$ is simply the largest of these values). The difficulty with this approach is that finding all possibly optimal solutions is intractable, even for problems of moderate size.

Subsequently, Inuiguchi and Sakawa (1996) formulated CMR to explicitly enforce *x-optimality*, thereby eliminating the need to construct the set E . They developed a branch and bound method to solve the resulting mathematical program. More recently, Mausser and Laguna (1997) exploited *c-consistency* to obtain the following mixed integer programming formulation of CMR, where M_i is an upper bound for x_i :

$$\begin{aligned}
R_{\max}(\hat{x}) \equiv \max \quad & \bar{c}z - cy \\
\text{s.t.} \quad & x \in \Omega, \\
& x + y - z = \hat{x}, \\
& y_i - \hat{x}_i b_i \leq 0 \quad \text{for } i = 1, \dots, n, \\
& z_i - (M_i - \hat{x}_i)(1 - b_i) \leq 0 \quad \text{for } i = 1, \dots, n, \\
& y, z \geq 0, \\
& b \in \{0, 1\}.
\end{aligned} \tag{5}$$

This formulation requires one integer variable (b_i) for each uncertain cost. The regret-maximizing costs are given by $\hat{c}_i = \bar{c}_i + b_i(\underline{c}_i - \bar{c}_i)$.

3. A heuristic method for CMR

Previous attempts to minimize maximum regret have concentrated on finding the optimal solution to CMR in Step 1 of the algorithm. Clearly, the termination criterion requires that $R_{\max}(\hat{x})$ be an upper bound for MMR (which can only be guaranteed by solving CMR to optimality). However, any $R(c, \hat{x}) > \text{LB}$ gives rise to a regret cut for MMR' that can increase the lower bound and allow the algorithm to continue. Mausser (1997) investigated accepting the first branch and bound solution whose regret exceeded the lower bound. While this reduced the processing time per iteration, it also generated so many additional “weak” regret cuts (i.e., for which $R(c, \hat{x}) < R_{\max}(x^*)$) that the overall processing time increased.

A heuristic method, able to quickly find a good solution for CMR, may be an attractive alternative to simply accepting the first valid branch and bound solution. When the heuristic solution does not exceed the lower bound, solving CMR as a mixed integer program will either generate a regret cut or satisfy the termination condition. A heuristic approach may also represent the only alternative for problems whose size makes impractical the solution of CMR as a mixed integer program.

3.1. A greedy procedure

The x -optimality and c -consistency conditions motivate the following greedy search procedure for CMR. Let (c^0, x^0) be an initial candidate solution, where without loss of generality we assume that x -optimality is satisfied. If c^0 is c -consistent, we are done. Otherwise, choose a new cost vector c^1 that is c -consistent with x^0 , and find x^1 that is x -optimal. Note that $c^1(x^1 - \hat{x}) \geq c^1(x^0 - \hat{x}) > c^0(x^0 - \hat{x})$, and so (c^1, x^1) is an improved solution. If (c^1, x^1) is c -consistent, we are done, otherwise we repeat the above procedure to find an improved solution (c^1, x^1) and continue as necessary. The procedure returns a solution that is both x -optimal and c -consistent, though not necessarily the optimal solution $(\hat{c}, x_{\hat{c}})$.

Note that c_i can attain either bound when $x_i = \hat{x}_i$ without violating c -consistency. In this case, we select the one that is most likely to be valid should x_i change. That is, if $\hat{x}_i > M_i/2$ we will require $c_i = \underline{c}_i$, and $c_i = \bar{c}_i$ otherwise.

In practice, many feasible solutions to CMR are both x -optimal and c -consistent, and so they can only be viewed as possessing a certain “local optimality”. Thus, performance of the heuristic may benefit from a good initial solution (c^0, x^0) as well as mechanisms to diversify the search after finding a locally optimal solution. We consider these issues in the following sections.

3.2. Finding an initial solution

Our goal is to choose an initial cost vector c^0 so that the resulting linear program yields a good initial candidate solution x^0 . Mausser and Laguna (1997) showed that CMR can be formulated as a linear program with convex piecewise-linear costs. Specifically, each cost consists of two segments with slope \underline{c}_i (for $x_i < \hat{x}_i$) and \bar{c}_i (for $x_i > \hat{x}_i$). We first consider constructing c^0 as a linear approximation to these costs.

Suppose that the true cost

$$c = \begin{cases} \underline{c} & \text{for } 0 \leq x \leq \hat{x}, \\ \bar{c} & \text{for } \hat{x} < x \leq M, \end{cases}$$

is to be approximated by k (Fig. 1). We are interested only in the slope k and not in the intercept of the approximation (the intercept represents a constant term that is irrelevant for optimization purposes). Note that if $\hat{x} = 0$ or M , then the true cost is in fact linear. Four possible approximations, and the corresponding values of k , are as follows (see Mausser, 1997, for details):

- Connecting the endpoints $(0, -\underline{c}\hat{x})$ and $(M, \bar{c}(M - \hat{x}))$,

$$k = \bar{c} + \frac{\hat{x}}{M}(\underline{c} - \bar{c}).$$

- A least squares fit using the points $(0, -\underline{c}\hat{x})$, $(\hat{x}, 0)$, and $(M, \bar{c}(M - \hat{x}))$,

$$k = \frac{3M\bar{c}(M - \hat{x}) - (\hat{x} + M)(\bar{c}(M - \hat{x}) - \underline{c}\hat{x})}{3(\hat{x}^2 + M^2) - (\hat{x} + M)^2}.$$

- Minimizing the squared error over the entire interval,

$$k = \bar{c} - \left(\frac{\hat{x}}{M}\right)^2 \left(3 - \frac{2\hat{x}}{M}\right) (\bar{c} - \underline{c}).$$

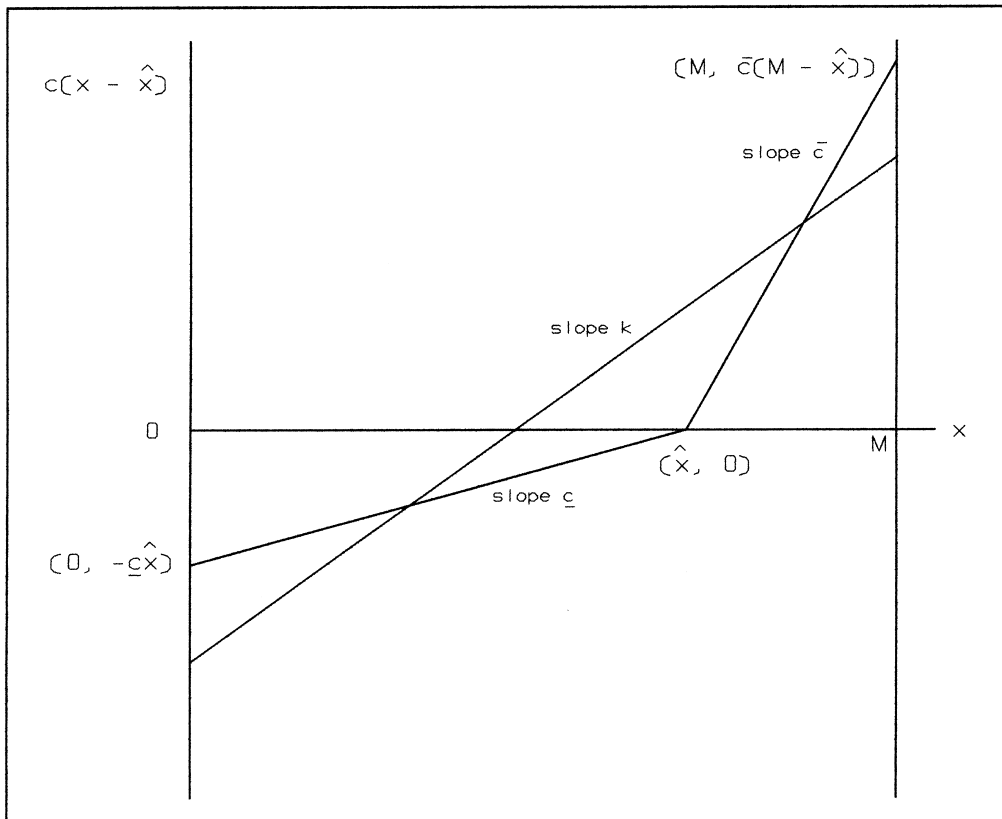


Fig. 1. Linear cost approximation.

- Minimizing the error over the entire interval (i.e., the area between the true costs and the linear approximation should be as small as possible),

$$k = \begin{cases} \bar{c} & \text{if } \hat{x} \leq \frac{M}{4}, \\ \frac{2}{M} \left[\bar{c} \left(\frac{3M}{4} - \hat{x} \right) + \underline{c} \left(\hat{x} - \frac{M}{4} \right) \right] & \text{if } \frac{M}{4} < \hat{x} < \frac{3M}{4}, \\ \underline{c} & \text{if } \hat{x} \geq \frac{3M}{4}. \end{cases}$$

Instead of a linear cost approximation, one can also solve the linear programming relaxation of (5) to obtain x^0 . By replacing the integrality requirement for the b -variables with $0 \leq b \leq 1$, we can obtain both $y_i > 0$ and $z_i > 0$ for some i . The solution given by $x^0 = \hat{x} - y + z$ satisfies $x^0 \in \Omega$ while c^0 is a weighted average of the bounds \underline{c} and \bar{c} .

3.3. Search diversification

Recall that a solution (c^k, x^k) satisfying x -optimality and c -consistency can be viewed as being locally optimal. The benefits of accepting non-improving moves to escape local optimality are well established, and form the basis for techniques such as tabu search (Glover and Laguna, 1997) and simulated annealing (Kirkpatrick et al., 1983). We propose to diversify the search as follows. After obtaining a locally optimal solution (c^k, x^k) , which becomes the *incumbent*, we “flip” one of the costs to its opposite bound, thereby violating c -consistency. We then find x^{k+1} that is x -optimal for the new cost vector c^{k+1} and continue with the greedy search until a new local optimum is found. If the regret of this new solution exceeds that of the incumbent, the incumbent is replaced and we diversify the search in the neighborhood of this new local optimum. Otherwise, we continue searching the neighborhood of the incumbent until all costs have been flipped without improving the solution. Note that if $\hat{x}_i = 0$ or M_i , then there is no need to flip c_i since the cost is \bar{c}_i or \underline{c}_i , respectively, for all values of x_i .

The objective of diversification is to obtain an x far enough away from the current local optimum so that the greedy search does not simply return to the incumbent solution. It is reasonable to assume that the larger the magnitude of the cost change, the greater the chance of obtaining a significantly different x . Thus, we propose to order costs by decreasing uncertainty interval size, and select flips based on this list. One possibility is to initiate each search by flipping the cost with the largest uncertainty interval. In other words, as each new incumbent is found, we proceed from the beginning of the list so that the most promising costs are flipped first. A potential drawback of this strategy is that the same costs are flipped repeatedly, while those with small uncertainty intervals are selected only if previous flips did not improve the solution. An alternative is to treat the list as circular and begin each new search with the next cost immediately after the last one used. We will investigate both options in our computational experiments.

3.4. Formal description

We now give a complete description of the heuristic procedure for CMR, assuming there are n uncertain costs. In the following, k is the iteration counter, f is the cost flip counter, g gives the index of the cost to be flipped, and d specifies the final cost flip.

- Inputs: A set of costs $\Gamma = \{c \mid \underline{c}_i \leq c_i \leq \bar{c}_i, i = 1, \dots, n\}$; a candidate minimax regret solution \hat{x} , a non-empty, bounded polyhedron Ω .
- Outputs: A set of regret maximizing costs c^d ; the maximum regret R .

- Step 0.* (Initialization). Choose $c^0 \in \Gamma$. Construct a list σ that orders the costs by decreasing uncertainty interval size (i.e., $\sigma(i) = h$, where $\bar{c}_h - \underline{c}_h \geq \bar{c}_{\sigma(j)} - \underline{c}_{\sigma(j)}$ for $i < j \leq n$). Set $k = 0$, $f = 0$, $R = 0$.
- Step 1.* (x -optimality). Find $x^k \in \Omega$ maximizing $c^k x$.
- Step 2.* (c -consistency). If c^k is c -consistent then go to Step 3. Otherwise, set
- $$c_i^{k+1} = \begin{cases} \underline{c}_i & \text{if } x_i^k < \hat{x}_i \text{ or } x_i^k = \hat{x}_i > M_i/2, \\ \bar{c}_i & \text{if } x_i^k > \hat{x}_i \text{ or } x_i^k = \hat{x}_i \leq M_i/2. \end{cases}$$
- Set $k = k + 1$ and go to Step 1.
- Step 3.* (local optimality). If $R(c^k, x^k) > R$ then set $R = R(c^k, x^k)$, $x^l = x^k$, $c^l = c^k$, and go to Step 4. Otherwise, go to Step 5.
- Step 4.* (diversify search). If circular flips are being used then set $d = f + n$. Otherwise, set $f = 0$, $d = n$. Go to Step 6.
- Step 5.* (re-install incumbent). If $f < d$ then set $c^k = c^l$ and go to Step 6. Otherwise go to Step 7.
- Step 6.* (flip cost). Set $f = f + 1$, $g = f$. If $g > n$ then set $g = g - n$. If $\hat{x}_{\sigma(g)} = 0$ or $\hat{x}_{\sigma(g)} = M_{\sigma(g)}$ then go to Step 5. Set
- $$c_{\sigma(g)}^{k+1} = \begin{cases} \underline{c}_{\sigma(g)} & \text{if } c_{\sigma(g)}^l = \bar{c}_{\sigma(g)}, \\ \bar{c}_{\sigma(g)} & \text{if } c_{\sigma(g)}^l = \underline{c}_{\sigma(g)}. \end{cases}$$
- Set $k = k + 1$ and go to Step 1.
- Step 7.* (terminate). STOP. (c^l, x^l) is the best solution found for CMR, with regret R .

4. Computational experiments

Our computational experiments have two primary goals. First, we evaluate the heuristic as a stand-alone method for solving CMR and investigate the effects that initialization, flip type, and level of uncertainty have on the quality of the final minimax regret solution. Second, we combine the heuristic approach with exact methods to determine whether such a strategy is more efficient than one that relies exclusively on mixed integer programming to solve CMR.

As a performance measure, we have chosen the percentage gap between the regret \hat{r} of the final MMR' solution and the maximum regret $R_{\max}(\hat{x})$ of the final candidate \hat{x} , which respectively represent lower and upper bounds for the optimal minimax regret. The percentage gap is calculated as $(R_{\max}(\hat{x}) - \hat{r})/\hat{r}$. $R_{\max}(\hat{x})$ is obtained by solving CMR exactly (i.e., using (5)) for the final candidate \hat{x} . Note that this is the CMR problem for which the heuristic could not find a cost scenario whose regret exceeded \hat{r} , thereby terminating the algorithm. Since $(R_{\max}(\hat{x}) - \hat{r})$ represents the amount by which the true maximum regret of \hat{x} has been underestimated, we will refer to the percentage gap as *error*.

Instead of arbitrarily generating random test problems, we followed the controlled randomization approach (Greenberg, 1990). As described below, we selected problems from the NETLIB collection (Gay, 1985) and generated uncertainty intervals of various sizes for a random subset of the cost coefficients.

We used the GNU Project C compiler (V 2.6) with maximum code optimization to implement the algorithm on a DEC Alpha 2000 running DEC OSF/1 V3.2. All mathematical programs were solved by the CPLEX Version 3.0 runtime library routines using the parameter values listed in Fig. 2. To account for rounding errors, we used a tolerance of 10^{-6} where applicable (e.g., when testing for the equality of $R_{\max}(\hat{x})$ and \hat{r} , x_i and \hat{x}_i , etc.). The initial candidate \hat{x} was found by solving the original linear program (1) with costs $c_i = \underline{c}_i$ if $-\underline{c}_i \geq \bar{c}_i$ and $c_i = \bar{c}_i$ if $\bar{c}_i > -\underline{c}_i$.

4.1. Sample problems

We selected nine problems from the NETLIB dataset that were sufficiently large to make their solution by the exact method (i.e., solving (5) optimally in each iteration, which we will refer to as OPTCUT) impractical. The need for solving the final CMR to calculate the percentage gap for evaluation purposes limited the number of uncertain costs that could be handled. We therefore restricted uncertainty to some random subset of the cost coefficients. Specifically, for each problem we specified a target value P for the proportion of costs to be made uncertain. For each non-zero cost in turn, a uniform (0, 1) variate was generated and if it was less than or equal to P , the cost was made uncertain. The value P was obtained by experimentation to yield problems of an appropriate level of difficulty. In general, our goal was to generate problems that could not be solved by the exact approach within one hour of processing time, yet that still allowed the final CMR to be solved in a reasonable amount of time. We selected uncertainty intervals of plus-or-minus 10%, 80%, and 150%.

Since none of the problems contained upper bounds for the decision variables, it was necessary to provide this information in order to guarantee a finite minimax regret. In actual practice, appropriate bounds typically may be deduced from the underlying problem. However, lacking sufficient background knowledge of the test problems and to avoid arbitrarily large values (which can increase the time required to solve (5)), we elected to set the upper bounds based on the optimal solution (x_D) of the original deterministic problem. Specifically, if \bar{x}_D is the average of all optimal solution values, we set the upper bound for x_i to the greater of $2(x_D)_i$ and $\bar{x}_D/4$.

Table 1 summarizes the problem characteristics. The difficulty of each problem can be assessed from Table 2, which reports the results of solving CMR with the OPTCUT approach. We imposed a time limit of one hour, but allowed the iteration in progress at that time to finish (hence the total time often exceeded 3600 s). The optimal minimax regret solution was found and verified within the one-hour limit for BEACONFD ($\pm 80\%$), SCAGR7 ($\pm 150\%$), and SCORPION ($\pm 150\%$). In two other instances (BEACONFD ($\pm 10\%$) and SCAGR7 ($\pm 10\%$)), optimality was verified during the evaluation phase. The problems E226 ($\pm 10\%$), ISRAEL ($\pm 10\%$), and SCORPION ($\pm 10\%$, $\pm 80\%$) presented particular difficulty for OPTCUT.

Although the results were not obtained as part of a rigorous statistical experiment, we choose to comment on two seemingly apparent relationships. The average error suggests that the performance of OPTCUT improves as the uncertainty level increases. However, disregarding SCORPION results in average errors of 6.8%, 3.4%, and 5.8% for uncertainty levels of $\pm 10\%$, $\pm 80\%$, and $\pm 150\%$, respectively. Thus, evidence of a relationship between performance and uncertainty level is less than conclusive. The uncer-

Simplex Parameters

LIMITS ITERATIONS = 500000

MIP Parameters

BACKTRACK = 2

LIMITS SOLUTIONS = 10000

LIMITS NODES = 1000000

TOLERANCES RELOBJDIFFERENCE = 0 (set to 1 for *FIRSTCUT*)

Preprocessing and Starting Parameters

ADVANCE = 0 (set to 1 for heuristic solution)

PRESOLVE = 0

Fig. 2. CPLEX parameter values.

Table 1
Sample problems

Problem	Variables	Constraints	Uncertain costs	Average half interval		
				±10%	±80%	±150%
ADLITTLE	97	56	53	70.29	562.29	1054.29
AGG	163	488	72	4.25	34.01	63.76
BANDM	472	305	23	0.32	2.56	4.80
BEACONFD	262	173	36	0.65	5.18	9.71
E226	282	223	45	0.19	1.52	2.85
ISRAEL	142	174	53	16.64	133.15	249.65
SCAGR7	140	129	22	16.71	133.64	250.57
SCORPION	358	388	28	2.35	18.77	35.19
SCTAPI	480	300	50	1.87	14.98	28.08

tainty level does appear to influence the time required to solve a single instance of CMR. One might speculate that a greater difference between the upper and lower limits for each cost improves the ability of branch-and-bound to fathom partial solutions.

4.2. Stand-alone heuristic

In this section, we investigate the performance of the heuristic under various operating conditions. In addition to establishing the general quality of heuristic solutions, we also want to determine if a certain initialization method or search strategy consistently yields the best results. A three-factor randomized complete block experimental design (see, for example, Ch. 5 in Anderson and McLean (1974)) and an analysis of variance was used to examine these issues. We elected to test hypotheses at a significance level of $p = 0.05$.

4.2.1. Experimental design

The fixed factors and their corresponding treatments include initialization type (endpoint connect, least squares, minimum squared error, minimum error, LP relaxation, and random), method of flipping costs (linear, circular), and uncertainty level ($\pm 10\%$, $\pm 80\%$, $\pm 150\%$). We include random initialization, which sets $c_i^0 = \underline{c}_i$ or \bar{c}_i with equal probability, to provide a benchmark for the other treatments. The nine sample problems are blocks, with each block receiving all treatment combinations. Since we want our results to be valid for all possible problems, not just for this particular sample, we allow the blocks to be random instead of fixed. For practical reasons, all nine problems are solved in sequence for each treatment combination. This randomization restriction precludes testing any observations that might be time-dependent, such as processing time in our multi-user computing environment.

Let Q_i be the effect of the i th initialization type, F_j be the effect of the j th flip method, U_k be the effect of the k th uncertainty level, and N_l be the effect due to the l th problem (block). The linear statistical model is

$$y_{ijkl} = \mu + Q_i + F_j + U_k + N_l + QF_{ij} + QU_{ik} + QN_{il} + FU_{jk} + FN_{jl} + UN_{kl} + QFU_{ijk} \\ + QFN_{ijl} + QUN_{ikl} + FUN_{jkl} + QFUN_{ijkl} + \epsilon_{(ijkl)h},$$

where the subscript h denotes replication. Note that the block and block interaction effects are random and mixed, respectively.

The appropriate test statistic (i.e., the ratio of mean squares that determines the F -value) is determined by the expected mean squares. To derive the expected mean squares in Table 3, we used the approach described by Montgomery (1976). Note that lower case letters represent the number of treatments, or levels of the corresponding factor.

Table 2
Error and processing times (s) for OPTCUT

Problem	±10%			±80%			±150%					
	Error	Total time	Per CMR	Eval time	Error	Total time	Per CMR	Eval time	Error	Total time	Per CMR	Eval time
ADLITTLE	0.085	3659	104	175	0.046	3674	63	159	0.067	3699	63	102
AGG	0.002	3902	299	920	0.002	3868	167	294	0.021	3640	86	250
BANDM	0.005	4034	334	548	0.095	8551	775	2439	0.144	4868	537	1610
BEACONFD		4119	684	933		2917	363	0	0.086	3809	345	494
E226	0.305	4314	860	1692	0.040	4019	125	314	0.021	3925	108	314
ISRAEL	0.137	13527	2254	738	0.037	4315	139	162	0.045	3712	74	146
SCAGR7		5338	1776	3147	0.001	3662	281	619		642	53	0
SCORPION	0.858	3656	1815	1190	0.721	3679	1825	2599		2029	167	0
SCTAPI	0.013	3621	105	134	0.051	3733	88	131	0.081	3655	75	135
Average	0.156	5130	915	1053	0.110	4269	425	746	0.052	3331	168	339

4.2.2. Experimental results

During our computational experiments, special action was sometimes necessary to solve and evaluate certain problems. We found that poor scaling occasionally resulted in an “unscaled infeasibilities” message (CPLEX, 1994) while solving CMR heuristically. In such cases, the solution was simply ignored and the heuristic allowed to continue. Evaluating several instances of SCORPION ($\pm 80\%$) and BANDM ($\pm 150\%$) caused CPLEX to terminate with insufficient memory. When necessary, these problems were re-evaluated after changing the following CPLEX parameters, which reduced memory requirements at the expense of execution time: ADVANCE = 0, BACKTRACK = 5, PRESOLVE = 1.

Table 4 shows the analysis of variance for solution error. There is strong evidence that the initialization method affects performance, but no other main or interaction effects are significant. Since the data indicated that random initialization performed relatively poorly, we conducted a second ANOVA that included only the other five initialization treatments (Table 5). In this case, there is no evidence that the initialization method, the flip type, or the uncertainty level influence the performance of the algorithm. Thus, while obtaining a good initial candidate does improve the performance of the heuristic, there is no clearly preferred approach among those strategies described in Section 3.2.

When the random initialization results were included, the overall mean error was 5.70%. This was reduced to 5.34% when these results were excluded. For the nine sample problems, obtaining an initial candidate from the LP relaxation and using circular flips for diversification gave the best results, with average and worst-case errors of 5.0% and 14.1%, respectively (Table 6). These results compare favorably to those of Table 2. Solution quality was significantly improved for the four difficult problems mentioned earlier, with the heuristic algorithm actually finding the optimal solution for SCORPION ($\pm 10\%$), in a fraction of the time allotted to the OPTCUT approach.

As mentioned previously, our experimental design does not allow us to draw conclusions about treatment effects on processing time. However, it is notable that the time needed to solve each instance of CMR

Table 3
Degrees of freedom and expected mean squares

Term	Degrees of freedom	(F)	(F)	(F)	(R)	(R)	Expected mean square
		<i>q</i>	<i>f</i>	<i>u</i>	<i>n</i>	1	
		<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>h</i>	
Q_j	$(q-1)$	0	<i>f</i>	<i>u</i>	<i>n</i>	1	$(fun \sum Q_j^2)/(q-1) + fu\sigma_{QN}^2 + \sigma^2$
F_j	$(f-1)$	<i>q</i>	0	<i>u</i>	<i>n</i>	1	$(qun \sum F_j^2)/(f-1) + qu\sigma_{FN}^2 + \sigma^2$
U_k	$(u-1)$	<i>q</i>	<i>f</i>	0	<i>n</i>	1	$(qfn \sum U_k^2)/(u-1) + qf\sigma_{UN}^2 + \sigma^2$
N_l	$(n-1)$	<i>q</i>	<i>f</i>	<i>u</i>	<i>l</i>	1	$qfu\sigma_N^2 + \sigma^2$
QF_{ij}	$(q-1)(f-1)$	0	0	<i>u</i>	<i>n</i>	1	$un\sigma_{QF}^2 + u\sigma_{QFN}^2 + \sigma^2$
QU_{ik}	$(q-1)(u-1)$	0	<i>f</i>	0	<i>n</i>	1	$fn\sigma_{QU}^2 + f\sigma_{QUN}^2 + \sigma^2$
QN_{il}	$(q-1)(n-1)$	0	<i>f</i>	<i>u</i>	1	1	$fu\sigma_{QN}^2 + \sigma^2$
FU_{jk}	$(f-1)(u-1)$	<i>q</i>	0	0	<i>n</i>	1	$qn\sigma_{FU}^2 + q\sigma_{FUN}^2 + \sigma^2$
FN_{jl}	$(f-1)(n-1)$	<i>q</i>	0	<i>u</i>	1	1	$qu\sigma_{FN}^2 + \sigma^2$
UN_{kl}	$(u-1)(n-1)$	<i>q</i>	<i>f</i>	0	1	1	$qf\sigma_{UN}^2 + \sigma^2$
QFU_{ijk}	$(q-1)(f-1)(u-1)$	0	0	0	<i>n</i>	1	$n\sigma_{QFU}^2 + \sigma_{QFUN}^2 + \sigma^2$
QFN_{ijl}	$(q-1)(f-1)(n-1)$	0	0	<i>u</i>	1	1	$u\sigma_{QFN}^2 + \sigma^2$
QUN_{ikl}	$(q-1)(u-1)(n-1)$	0	<i>f</i>	0	1	1	$f\sigma_{QUN}^2 + \sigma^2$
FUN_{jkl}	$(f-1)(u-1)(n-1)$	<i>q</i>	0	0	1	1	$q\sigma_{FUN}^2 + \sigma^2$
$QFUN_{ijkl}$	$(q-1)(f-1)(u-1)(n-1)$	0	0	0	1	1	$\sigma_{QFUN}^2 + \sigma^2$
$\epsilon_{(ijkl)h}$	0	1	1	1	1	1	σ^2 (no estimate)

Table 4
Analysis of variance for error (including random initialization)

Source of variation	Sum of squares	Degrees of freedom	Mean square	<i>F</i>	<i>p</i> -value
<i>Q</i>	0.0209	5	0.0042	7.9054	0.000030
<i>F</i>	0.0000	1	0.0000	0.1688	0.691957
<i>U</i>	0.0221	2	0.0110	0.6287	0.545964
<i>N</i>	0.3291	8	0.0411		
<i>QF</i>	0.0011	5	0.0002	1.4930	0.213754
<i>QU</i>	0.0065	10	0.0007	1.3028	0.243415
<i>QN</i>	0.0212	40	0.0005		
<i>FU</i>	0.0004	2	0.0002	0.8053	0.464263
<i>FN</i>	0.0012	8	0.0001		
<i>UN</i>	0.2808	16	0.0175		
<i>QFU</i>	0.0017	10	0.0002	1.0104	0.442204
<i>QFN</i>	0.0057	40	0.0001		
<i>QUN</i>	0.0400	80	0.0005		
<i>FUN</i>	0.0044	16	0.0003		
<i>QFUN</i>	0.0133	80	0.0002		
Total	0.7483	323			

using a circular list was approximately half that required by the alternative method. We speculate that this is due in part to the reduced number of (non-improving) cost flips that are immediately reversed.

4.3. Combined heuristic – MIP approach

We now consider using the heuristic approach in conjunction with mixed integer programming. Specifically, whenever the heuristic is unable to find a solution to CMR that exceeds the current lower bound, the algorithm solves CMR as a mixed integer program to either generate a regret cut or verify optimality.

Table 5
Analysis of variance for error (excluding random initialization)

Source of variation	Sum of squares	Degrees of freedom	Mean square	<i>F</i>	<i>p</i> -value
<i>Q</i>	0.0006	4	0.0002	1.9504	0.126071
<i>F</i>	0.0000	1	0.0000	0.1024	0.757174
<i>U</i>	0.0159	2	0.0079	0.5627	0.580552
<i>N</i>	0.2479	8	0.0310		
<i>QF</i>	0.0011	4	0.0003	1.6101	0.195753
<i>QU</i>	0.0015	8	0.0002	1.7238	0.109891
<i>QN</i>	0.0026	32	0.0001		
<i>FU</i>	0.0005	2	0.0003	1.1117	0.232020
<i>FN</i>	0.0009	8	0.0001		
<i>UN</i>	0.2255	16	0.0141		
<i>QFU</i>	0.0016	8	0.0002	1.1097	0.367438
<i>QFN</i>	0.0052	32	0.0002		
<i>QUN</i>	0.0072	64	0.0001		
<i>FUN</i>	0.0026	16	0.0002		
<i>QFUN</i>	0.0112	64	0.0002		
Total	0.5242	269			

Table 6
Error and processing times (s) for heuristic with LP relaxation initialization and circular diversification

Problem	±10%			±80%			±150%					
	Error	Total time	Per CMR	Eval time	Error	Total time	Per CMR	Eval time	Error	Total time	Per CMR	Eval time
ADLITTLE	0.095	19	0.33	162	0.085	88	0.64	94	0.068	128	0.75	110
AGG	0.083	21	0.27	384	0.071	34	0.53	124	0.141	73	0.63	77
BANDM	0.041	44	1.89	487	0.008	102	2.54	28968	0.007	154	3.00	49731 ^a
BEACONFD	0.075	10	0.30	862	0.004	11	0.32	625	0.008	15	0.30	570
E226	0.045	63	1.32	17894	0.024	194	2.29	1279	0.031	251	2.79	878
ISRAEL	0.071	32	0.85	287	0.102	84	1.91	90	0.079	279	2.16	35
SCAGR7	0.000	4	0.16	2951	0.001	7	0.22	587	0.055	7	0.27	6
SCORPION		28	0.39	26248	0.005	31	0.39	14271	0.085	31	0.60	28
SCTAPI	0.101	51	0.93	82	0.021	342	2.03	2012	0.035	500	2.60	217
Average	0.057	30	0.72	5484	0.036	99	1.21	5339	0.057	219	1.46	5739

^a Modified CPLEX parameters for evaluation.

We will consider finding the optimal solution to CMR (OPTCUT strategy) as well as finding the first solution whose regret exceeds the current lower bound (FIRSTCUT strategy). We solve the LP relaxation of CMR to obtain an initial candidate and employ circular flips for diversification, based on our previous experimental results. Our primary interest is to determine if the combined heuristic/OPTCUT or heuristic/FIRSTCUT approaches yield better solutions than a pure OPTCUT or FIRSTCUT strategy in a comparable amount of time.

4.3.1. Experimental design

Table 6 shows that, in most cases, the time spent obtaining and evaluating a heuristic solution was still less than the one hour allotted to the OPTCUT method. Thus we decided to impose an identical time limit on the combined and FIRSTCUT approaches (again allowing any iteration in progress after one hour to finish). In all five cases where the evaluation time exceeded one hour, the stand-alone heuristic solution was already better than that of OPTCUT and so we did not apply the heuristic/OPTCUT approach to these problems. Since our control of processing time is somewhat imprecise, both because of the multi-user computing environment and our allowing the final iteration to finish, we feel that a formal statistical analysis of the results is not appropriate in this case.

4.3.2. Experimental results

Table 7 shows the errors for the combined and FIRSTCUT approaches. FIRSTCUT recorded the best solution for problem E226 ($\pm 10\%$) only, although the solution found by the heuristic/FIRSTCUT method could not be evaluated for this problem and may in fact be superior. In all other cases, one of the combined approaches matched or exceeded the solution quality of OPTCUT (see Table 2) and FIRSTCUT.

To provide a more meaningful comparison that takes into account differences in processing time, we subtracted the error and total processing time for the OPTCUT method in Table 2 from the respective values for each method in Tables 6 and 7. The results are plotted in Fig. 3. Any point in the lower left quadrant signifies a clear domination of OPTCUT (i.e., smaller error and shorter processing time) while points in the upper right quadrant indicate clearly inferior performance. Based on these results, the combined approaches appear to be viable alternatives to both OPTCUT and FIRSTCUT. However, more detailed experimentation is required to ascertain which, if any, of the combined approaches is superior.

Table 7
Errors FIRSTCUT (F), Heuristic/OPTCUT (H/O), and Heuristic/FIRSTCUT (H/F) approaches

Problem	$\pm 10\%$			$\pm 80\%$			$\pm 150\%$		
	F	H/O	H/F	F	H/O	H/F	F	H/O	H/F
ADLITTLE	0.052	0.020	0.036	0.058	0.026	0.012	0.112	0.015	0.027
AGG		0.001		0.007	0.001	0.032	n/a ^c	0.003	n/a ^c
BANDM	0.018		0.002	0.085	n/a ^a	0.007	0.081	n/a ^a	n/a ^c
BEACONF	0.297			0.569			0.805	0.000	0.002
E226	0.026	n/a ^a	n/a ^c	0.112	0.011	0.006	0.093	0.013	0.015
ISRAEL	0.055	0.044 ^b	n/a ^c	0.041	0.058	0.031	0.055	0.029	0.032
SCAGR7				0.001					
SCORPION		n/a ^a		n/a ^d	n/a ^a	n/a ^d			
SCTAPI	0.042		0.012	0.085	0.004	0.005	0.067	0.025	0.027

^a Not attempted.

^b Stopped after 43 cuts (total time exceeded 24 h).

^c Evaluation failed.

^d MIP failed during heuristic.

^e Unscaled infeasibilities in MMR'.

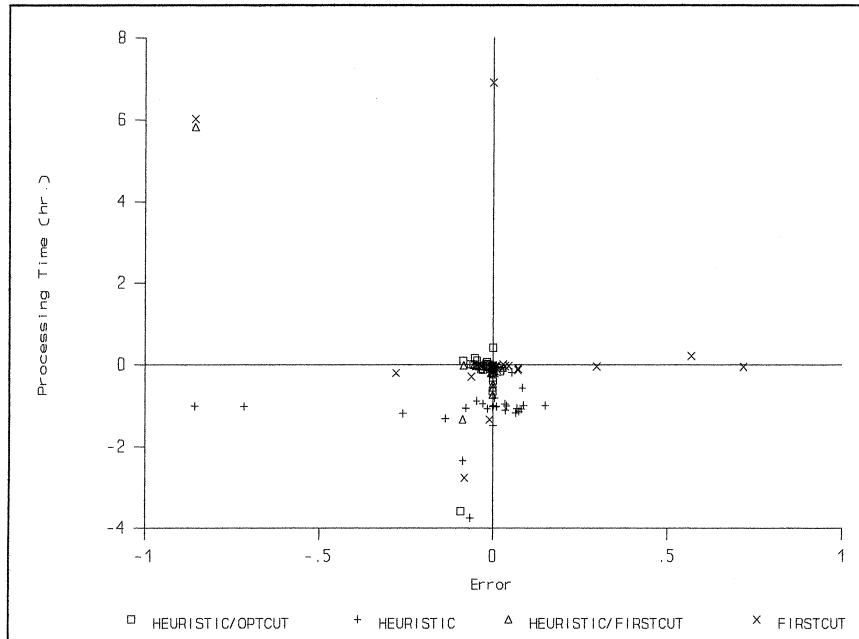


Fig. 3. Differences from OPTCUT.

5. Conclusions and extensions

The minimax regret solution for a linear program with interval objective function coefficients can be found using an iterative algorithm that solves a mixed integer program (CMR) to generate cuts for a linear program (MMR'). This paper has presented a heuristic method for solving CMR, based on a greedy search that derives from satisfying the conditions of x -optimality and c -consistency. We proposed several techniques for finding a good initial solution for the greedy search, along with a simple diversification strategy to extend the search beyond a local optimum.

The purpose of our computational experiments was twofold. First, we showed that when the heuristic was used as a stand-alone procedure for CMR, the iterative algorithm found solutions that were within 5% of optimal on average. These results compare favorably with those of the exact method, and were obtained in a fraction of the processing time allotted to OPTCUT. While a good initial solution had a significant affect on final solution quality, we did not detect a clear preference among the proposed techniques. Neither the level of uncertainty nor the variant of our diversification strategy had an impact on solution quality.

Second, we investigated a combined approach that used the heuristic solution to CMR whenever possible and solved the corresponding mixed integer program only when necessary. Under a similar one-hour time allotment, the combined approach outperformed both OPTCUT and FIRSTCUT. Our results suggest that the trade-off between finding the strongest cuts (by solving CMR optimally) and generating a larger number of possibly weaker cuts (by solving CMR heuristically) can be manipulated to coincide with the available processing time, or to match limitations imposed by the size of the problem.

The heuristic described in this paper extends the usefulness of the minimax regret criterion in practice by allowing larger problems to be addressed. The heuristic was shown to be effective for problems containing several hundred variables and constraints, and up to 72 uncertain costs. In particular, as the number of

uncertain costs increases, the ability to solve CMR as a mixed integer program diminishes and the stand-alone heuristic procedure remains the only available option.

A useful topic for future research relates to strengthening our diversification procedure, which is extremely simple in light of the powerful local search techniques currently available. We feel that a tabu search procedure can significantly improve the quality of CMR solutions and as a result, allow the heuristic to generate more cuts before terminating or invoking mixed integer programming in a combined approach.

Acknowledgements

The authors thank two anonymous referees for their helpful comments that improved an earlier version of this paper.

References

- Anderson, V.A., McLean, R.A., 1974. *Design of Experiments: A Realistic Approach*. Marcel Dekker, New York.
- Barlow, J., Glover, F., 1987. A multicriteria stratification framework for uncertainty and risk analysis. *International Journal of Policy and Management* 12 (3), 77–85.
- Bell, D.E., 1982. Regret in decision making under uncertainty. *Operations Research* 30, 961–981.
- Bell, D.E., 1985. Putting a premium on regret. *Management Science* 31 (1), 117–120.
- CPLEX, 1994. CPLEX Optimization Inc., Using the CPLEX Callable Library (Version 3), CPLEX Optimization, Inc., Incline Village, NV.
- Daniels, R.L., Kouvelis, P., 1995. Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management Science* 41 (2), 363–376.
- Gay, D.M., 1985. Electronic mail distribution of linear programming test problems. *Mathematical Programming Society Committee On Algorithms (COAL) Newsletter*.
- Glover, F., Laguna, M., 1997. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA.
- Greenberg, H.J., 1990. Computational testing: Why, how and how much. *ORSA Journal on Computing* 2 (1), 94–97.
- Gupta, S.K., Rosenhead, J., 1968. Robustness in sequential investment decisions. *Management Science* 15 (2), B18–B29.
- Gutierrez, G.J., Kouvelis, P., 1995. A robustness approach to international sourcing. *Annals of Operations Research* 59, 165–193.
- Inuiguchi, M., Sakawa, M., 1995. Minimax regret solution to linear programming problems with an interval objective function. *European Journal of Operational Research* 86, 526–536.
- Inuiguchi, M., Sakawa, M., 1996. Maximum regret analysis in linear programs with an interval objective function. In: *Proceedings of IWSCI'96*, pp. 308–317.
- Kirkpatrick, S., Gelatt, C., Vecchi, M., 1983. Optimization by simulated annealing. *Science* 220, 671–680.
- Kouvelis, P., Kurawarwala, A.A., Gutierrez, G., 1992. Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research* 63, 287–303.
- Loomes, G., Sugden, R., 1982. Regret theory: An alternative theory of rational choice under uncertainty. *The Economic Journal* 92, 805–824.
- Loomes, G., Sugden, R., 1987. Testing for regret and disappointment in choice under uncertainty. *The Economic Journal* 97, 118–129.
- Mausser, H.E., 1997. *Minimizing Maximum Regret for Linear Programs with Interval Objective Function Coefficients*. Unpublished Ph.D. Thesis, Graduate School of Business, University of Colorado, Boulder, CO.
- Mausser, H.E., Laguna, M., 1997. A new mixed integer formulation for the maximum regret problem. Working Paper, Graduate School of Business, University of Colorado, Boulder, CO.
- Montgomery, D.C., 1976. *Design and Analysis of Experiments*. Wiley, New York.
- Mulvey, J.M., Vanderbei, R.J., Zenios, S.A., 1995. Robust optimization of large scale systems. *Operations Research* 43, 264–281.
- Ritov, I., 1996. Probability of regret: Anticipation of uncertainty resolution in choice. *Organizational Behavior and Human Decision Processes* 66 (2), 228–236.
- Rosenblatt, M.J., Lee, H.L., 1987. A robustness approach to facilities design. *International Journal of Production Research* 25 (4), 479–486.
- Rosenhead, J., Elton, M., Gupta, S.K., 1972. Robustness and optimality as criteria for strategic decisions. *Operational Research Quarterly* 23 (4), 413–431.

- Shimizu, K., Aiyoshi, E., 1980. Necessary conditions for min–max problems and algorithms by a relaxation procedure. *IEEE Transactions on Automatic Control* AC-25, 62–66.
- Simonson, I., 1992. The influence of anticipating regret and responsibility on purchase decisions. *Journal of Consumer Research* 19, 105–118.
- Zeelenberg, M., Beattie, J., Van Der Pligt, J., De Vries, N.K., 1996. Consequences of regret aversion: Effects of expected feedback on risky decision making. *Organizational Behavior and Human Decision Processes* 65 (2), 148–158.