

Applications of Simulation-Optimization Methods in Environmental Policy Planning Under Uncertainty

J. S. Yeomans*

Operations Management and Info. Systems, Schulich School of Business, York University, Toronto, ON, M3J 1P3, Canada

Received 6 March 2008; revised 17 July 2008; accepted 31 August 2008; published online 1 December 2008

ABSTRACT. Environmental policy formulation can prove especially complicated, since in general, system components contain considerable degrees of uncertainty. However, simulation-optimization (SO) techniques can be adapted to model a wide variety of problem types in which system components are stochastic. In this paper, it is shown how multiple environmental policy alternatives meeting required system criteria, or modelling-to-generate-alternatives (MGA), can be quickly and efficiently created using SO. The efficacy of this MGA approach is illustrated using two case studies. Furthermore, since SO techniques can be adapted to problems in which many system components are stochastic, the practicality of this approach can be extended into many other operational and strategic planning applications containing significant sources of uncertainty.

Keywords: environmental decision making under uncertainty, simulation-optimization, planning and strategy, modeling to generate alternatives

1. Introduction

Environmental policy formulation can prove to be an especially complicated process, since the components of environmental systems generally contain considerable degrees of uncertainty. As a result, the environmental decision-making arena has supplied an ideal background for testing the diverse spectrum of modelling techniques used to support policy formulation (Linton et al., 2002; Rubenstein-Montano et al., 2000). Numerous deterministic mathematical programming and optimization techniques have been used for environmental policy determination (e.g. Ferrell and Hizlan, 1997; Hasit and Warner, 1981; Haynes, 1981; Lund, 1990; Lund et al., 1994; Marks and Liebman, 1971; Walker, 1976; Wenger and Cruz, 1990). However, the abundance of stochastic uncertainty within environmental systems renders most deterministic methods relatively unsuitable for practical implementation purposes, since they provide no effective mechanism with which to directly incorporate the system uncertainties into their solutions (Brown et al., 1974; Coyle, 1973; Liebman, 1975; Gottinger, 1986; MacDonald, 1996; Tchobanoglous et al., 1993).

When input data cannot be expressed with appropriate certainty, the quality of the solutions produced by deterministic techniques need to be viewed somewhat dubiously. To circumvent these uncertainty shortcomings, various Monte Carlo

simulation approaches have been introduced into environmental planning processes (Bodner et al., 1970; Baetz, 1990; Wang et al., 1994; Openshaw and Whitehead, 1985). Unfortunately, while simulation provides an effective mechanism for comparing stochastic system behaviours, it does not provide a formal means for producing best system solutions. Hence, to incorporate data uncertainty into “optimal” environmental planning, two studies implemented the techniques of grey programming (GP) (Huang et al., 1998) and simulation-optimization (SO) (Yeomans et al., 2003). GP provides a computationally efficient optimization technique which processes system uncertainties expressed as interval estimates (Huang et al., 1994a, b), while SO directly incorporates inherent system uncertainties expressed as probability distributions (Fu, 2002; Kelly, 2002). Both of these approaches focused upon the “function optimization” aspects of the modelled systems, with the goal being to determine single best system policies.

GP has been extensively applied to a variety of stochastic environmental planning problems, since it circumvents many parameter uncertainty difficulties experienced in discrete optimization techniques (Bass et al., 1997; Chang and Wang, 1995; Chang et al., 1996; Huang, 1996; Huang et al., 1994a, b, 1995, 1996a, b, 1997; Yeh, 1996). In GP, problems containing interval parameter estimates are transformed into pairs of deterministic submodels which generate stable upper and lower limits on the solution when solved in tandem. Unlike more straightforward best-case/worst-case analyses, the submodel transformation has to be executed in a prescribed order using the output from the first submodel as direct input into the second. The resulting output from GP is a set of stable interval values for both the objective function and the decision variables.

* Corresponding author. Tel.: +1 416 7365074; fax: +1 416 7365687.
E-mail address: syeomans@schulich.yorku.ca (J. S. Yeomans).

In contrast to GP, SO incorporates inherent stochastic parameters directly into its solution construction by representing all uncertainties as probability distributions within the modeled system (Fu, 2002; Kelly, 2002). While SO holds considerable potential for application to a wide range of stochastic problems, it cannot be considered universally applicable due to related solution time issues. Because its search procedures involve probabilistic processes, SO's solution times are stochastic and can vary considerably from one implementation to the next. As a result, the major difficulty experienced by SO is the length of time required for its solution search to converge to optimality (Lacksonen, 2001).

Usually optimization-based techniques are designed to create only single best solutions to problems. However, due to the presence of considerable system uncertainty and to the possibility that opposition from a dominant stakeholder can actually eliminate any single (even an optimal) solution from further consideration, environmental policy-makers faced with difficult and potentially controversial choices prefer to have the capability of selecting from a set of alternatives (Huang et al., 1996b). Preferably all of these alternatives possess near-optimal objective measures, but would differ from each other in terms of the system structures characterized by their decision variables. Hence, from an environmental policy formulation perspective, it often proves preferable to be able to generate several good alternatives that provide multiple different perspectives to the same problem.

In response to this option-generation requirement, several approaches collectively referred to as *modelling-to-generate-alternatives* (MGA) have been developed (Baetz et al., 1990; Brill, 1979; Brill et al., 1981; Chang et al., 1980, 1982; Church and Huber, 1979; Falkenhausen, 1979; Gidley and Bari, 1986; Rubenstein-Montano and Zandi, 1999; Rubenstein-Montano et al., 2000). The goal for all MGA methods is to create an optimal solution together with a set of several near-optimal alternatives (Gidley and Bari, 1986). Policy-makers can then perform a subsequent comparison of these alternatives to determine which option most closely satisfies their specific circumstances.

Current MGA formulation practice dictates that policy-designers propose a number of technologically feasible policy alternatives, which are then evaluated by estimating their performance and effect on the system. This stage is followed by an in-depth comparison of these alternatives in which the policy-designer selects the option that best achieves the established system criteria. The disadvantage to this approach is that policy-makers can only ever realistically generate a finite subset of design alternatives, while the number of feasible options could prove to be extremely numerous. This restriction to a small subset of possibilities leads to the significant possibility of overlooking many better system design alternatives. Another shortcoming of MGA techniques arises from the fact that most are designed using deterministic mathematical programming methods, thereby not effectively incorporating uncertainty in their solutions.

To address this MGA deficiency, Yeomans (2002) demon-

strated that SO could be used to generate multiple policy options that would never have been considered by decision-makers, while simultaneously integrating system uncertainty directly into each generated alternative. SO could therefore replace the naturally myopic design tendencies of policy-makers with a formal MGA mechanism for generating numerous policies that would not have been considered otherwise. Yeomans and Huang (2003) and Linton et al. (2002) have shown SO to be an effective MGA technique in environmental policy formulation. Unfortunately, other researchers have shown that the stochastic aspects of SO's solution time can impact negatively upon its results (Azadivar, 1999; Fu, 2002; Kelly, 2002; Lacksonen, 2001) and that the solution quality for a given problem varies considerably from one implementation of SO to another (Fu, 2002; Lacksonen, 2001). Since the solution time of SO impacts negatively on its ability to determine optimal solutions, this difficulty clearly extends into its use as an MGA procedure. Huang et al. (2005) and Yeomans (2005, 2006) have proposed different approaches to improve the search times and solution quality of SO. In this paper, it is illustrated how these approaches have impacted the MGA capabilities of SO using the environmental case studies taken from Yeomans and Huang (2003) and Yeomans et al. (2003). Since these techniques can be adapted to many other stochastic systems, the practicality of these MGA improvements can be readily modified to satisfy numerous other environmental planning situations.

2. Simulation-Optimization for Function Optimization

In general, it is often exceedingly difficult to find optimal solutions to large stochastic problems when system uncertainties have to be incorporated *directly* into the solution procedure (Fu, 2002). Suppose the mathematical representation of an optimization problem contains n decision variables, X_i , expressed in vector form as $\mathbf{X} = [X_1, X_2, \dots, X_n]$. If the objective function is represented by F and the problem's feasible region is given by D , then the mathematical programming problem is to optimize $F(\mathbf{X})$ subject to $\mathbf{X} \in D$. In modelling stochastic systems, any uncertain elements are represented by stochastic functions. For instance, uncertain parameters within the objective or constraints appear as probability distributions. Additionally, D could contain a set of r constraints of the form $h_j(\mathbf{X}) \leq 0, j = 1, \dots, r$, that need not *always* be satisfied by every feasible solution instance. Such stochastic conditions in the constraints create solution difficulties because the resulting systems possess only fuzzy feasible region boundaries. Azadivar and Tompkins (1999) considered multiobjective formulations, in which F consisted of a vector-valued function containing q functions $F_k, k = 1, \dots, q$, while Pierreval and Tautou (1997) introduced instances in which D could simultaneously contain decision options from the real, integer and qualitative domains.

When stochastic conditions exist, values for the constraints and objective can only be efficiently estimated by simulation. Thus, any solution comparison between two distinct decisions

X1 and *X2* necessitates the evaluation of some statistic of *F* modelled with *X1* to the same statistic modelled with *X2* (Pierreval and Tautou, 1997). These statistics are calculated by a simulation performed on the solutions, in which each candidate solution provides the decision variable settings in the simulation. While simulation presents a mechanism for *comparing* results, it does not provide the means for *determining* optimal solutions to problems. Hence, simulation, by itself, cannot be used as a stochastic optimization procedure.

SO is a broadly defined set of solution approaches that combine simulation with some type of optimization method for stochastic optimization (Fu, 1994, 2002; Law and Kelton, 2000). In SO, all unknown objective functions, constraints, and parameters are replaced by one or more discrete event simulation models in which the decision variables provide the settings under which the simulation is performed. Since all measures of system performance are stochastic, every potential solution, *X*, examined would necessarily need to be evaluated via simulation and simulation is a computationally intensive process.

As simulation is computationally intensive, an optimization component is used to guide the solution search through the problem's feasible region using as few simulation runs as possible. Azadivar (1999) and Fu (1994) identified four broad classes of optimization search strategies that have guided SO searches; (i) gradient-based methods, (ii) stochastic approximation methods, (iii) response surface methods, and (iv) heuristic methods. Lacksonen (2001) contrasted the performance of SO search strategies and found that the performance of evolutionary procedures (a stochastic approximation method) clearly proved to be the most robust. For the remainder of the paper, only SO search strategies directed by evolutionary algorithms are considered.

Evolutionary SO consists of two alternating phases; an evolutionary module and a simulation module. Evolutionary SO maintains a set, or *population*, of candidate solutions throughout its execution. The evolutionary module considers the entire population of solutions during each generation of the search and *evolves* from a current population to a subsequent one. Because of the system's stochastic components, all performance measures are necessarily statistics calculated from the responses generated in the simulation module. The quality (or fitness for survival) of each solution in the population is found by having its performance criterion, *F*, evaluated by simulation. After simulating each candidate solution, the respective fitness values are returned to the evolutionary module to be utilized in the creation of the next generation of candidate solutions. The fitness of each candidate solution within the population is ranked in comparison to every other candidate solution. These fitness measures are inputs to the evolutionary module where the next population of candidate solutions is created using an evolutionary algorithm.

One primary principle of evolutionary algorithms is that fitter solutions in the current population possess a greater likelihood for survival and progression into the subsequent generation. The evolutionary module evolves the system toward improved solutions in subsequent populations and ensures that

the solution search does not become fixated at some local optima. After generating a new candidate solution set in the evolutionary module, the new population is returned to the simulation module for comparative evaluation. This alternating, two-phase search process terminates when an appropriately stable system state has been attained (Azadivar and Tompkins, 1999; Pierreval and Tautou, 1997). The optimal solution produced by the procedure is the single best solution found over the course of the entire search.

3. Using Grey Programming to Improve SO

Because stochastic system problems contain many possible solutions, solution quality can be highly variable unless an extensive search has been performed throughout the problem's entire feasible domain. Evolutionary methods are conducive to these extensive searches because the complete set of candidate solutions maintained in their populations permit searches to be undertaken throughout multiple sections of the feasible region, concurrently.

However, since the search processes of evolutionary procedures are probabilistic; their actual solution times are stochastic and can vary considerably from one implementation to the next. In fact, the major difficulty experienced by SO procedures has been the length of time required for them to converge to optimality (Fu, 2002; Lacksonen, 2001). Expressed equivalently, SO's solution quality can be highly variable for a search conducted over any fixed period of time.

In performing its search, SO commences from the set of candidate solutions of its initial population and then evolves from one population to a subsequent one in its searching process. Consequently, an important consideration is the determination of the initial population, since this can bias the initial search direction of the procedure. In general cases (i.e. evolutionary searches that do not possess a simulation component), initial populations for evolutionary searches have been randomly generated for numerous justifiable reasons (Caudill and Butler, 1990; Goldberg, 1989, 1991; Holland, 1992). However, Reeves (1993) suggested that a directed generation of the initial population can sometimes prove more efficient than this traditional random approach in accelerating solution convergence.

If a computationally efficient method can be found to generate settings for a good starting population, then this population can be used to direct the solution search into more preferred regions of a large feasible domain; thereby producing better solutions faster. An effective determination of this initial population biasing reduces the sole reliance on the evolutionary aspects of the search procedure for finding good solutions. If an appropriate initial solution can be found, then the resulting directed search could significantly reduce the solution time while simultaneously increasing the solution quality. Because of its inherent efficiencies, the computationally efficient GP procedure can be used to quickly generate this requisite good initial population from which to start SO (Huang et al., 2005).

4. Using Penalty Functions Minimization to Improve SO

As mentioned above, since evolutionary search procedures are probabilistic, their solution times and convergence times are stochastic (Fu, 2002; Lacksonen, 2001). In SO, the feasibility of each candidate solution is assessed during the simulation analysis performed on the current population, and in strictly enforced (or hard-constrained) optimization situations, any solution not satisfying the stated constraints has to be discarded. Since simulation is computationally intensive, should significant portions of search time be expended examining infeasible solutions, then considerable portions of search effort would be devoted to pursuing fruitless avenues. Gendreau (2002) has indicated that evolutionary algorithms, in general, (those that do not include a simulation component during their search) expend considerable computational effort in correcting the many infeasible or meaningless solutions that can be created by their evolutionary operators. However, incorporating infeasibilities via penalty functions reflecting degrees of constraint violation can rapidly force evolutionary searches toward more preferable solutions (Goldberg, 1989).

If SO received feedback from solutions that nearly satisfied its constraints, then its search would evolve more quickly toward optimal regions. For constrained deterministic optimization problems, Goldberg (1989) suggested implementing penalty-functions to reflect the degree of solution violation and this approach can be readily modified to SO.

Suppose that the constrained optimization problem has been expressed as minimizing objective $F(\mathbf{X})$ subject to $g_i(\mathbf{X})$, $i = 1, \dots, n$. In order to incorporate penalties into this formulation, a measurement of the degree of constraint violation must be expressed for each constraint i , $i = 1, \dots, n$. Depending upon the direction of the constraint, a constraint violation, $\eta_i \geq 0$, can be computed as either $g_i(\mathbf{X}) - \eta_i \leq \mu_i$ for less-than-or-equal-to constraints or $g_i(\mathbf{X}) + \eta_i \geq \lambda_i$ for greater-than-or-equal-to constraints. Each constraint's proportional deviation, Δ_i , can then be calculated as either η_i/μ_i or η_i/λ_i , so that the penalty function, Φ_i , applied to each violated constraint i is $\Phi_i[g_i(\mathbf{X})] = e^{\Delta_i} - 1$. Using this approach, the constrained optimization problem can be transformed into an unconstrained penalty format of minimizing $F(\mathbf{X}) + \sum_{i=1}^n \pi_i \Phi_i[g_i(\mathbf{X})]$, where π_i is an objective weighting coefficient applied to penalty function Φ_i .

Incorporating infeasible solutions into the search population via these penalty functions forces the evolutionary search to proceed toward more preferable feasible scenarios in order to reduce the negative impact from the penalties, while at the same time reducing the computational burden required in having to evaluate and discard infeasible instances. Yeomans (2005, 2006) has shown how the penalty function approaches from discrete optimization can be extended into SO.

5. Modelling to Generate Policy Alternatives with Simulation-Optimization

Since environmental policy formulation problems contain

so many uncertain components, reality dictates that they most likely possess more than one acceptable solution. Fortunately, in addition to its optimization capabilities, SO provides a mechanism for generating multiple policy options that might never have been considered by decision-makers, while simultaneously integrating uncertainty directly into each generated alternative (Yeomans, 2002). SO can therefore be used to replace any myopic design tendencies of policy-makers with a formal MGA mechanism for generating numerous good policy options.

As described earlier, evolutionary algorithms maintain a population of solutions throughout their searching phase. Each solution in a population corresponds to one specific policy option and, therefore, the population of candidate solutions represents an entire set of policy alternatives. When evolving from one population to a subsequent one, relatively weaker candidate solutions within a population become progressively replaced by better solutions in an evolutionary survival-of-the-fittest analogy (Holland, 1992; Goldberg, 1989, 1991; Caudill and Butler 1990). In SO, the comparative ranking of these solutions is determined by the statistics calculated for their respective objectives during the simulation phase and surviving solutions from the evolutionary processes are retained using the survival-of-the-fittest methodology based upon the value of their statistical measures.

Therefore, upon completion of its optimization search, SO's final resident population would necessarily correspond to a highly "fit" population. Thus, when SO terminates, it has not only found the single optimal solution from its search, but has also created the set of fit solutions resident in the terminal population (Yeomans, 2002). This final, fit population corresponds to a set of good policy alternatives that could be considered for actual implementation. Hence, SO actually generates a collection of good solution alternatives for MSW planning in addition to having determined its best solution and can therefore be considered a *de facto* MGA technique.

By adopting this MGA methodology, multiple design options can be created that meet established system criteria, while simultaneously remaining acceptable and implementable in practice. These solutions can be considered good from two perspectives: (i) based upon the evolving nature of the search process, all surviving solutions will be extremely fit – corresponding to policy solutions that have necessarily achieved near-optimal measures for their objective functions; and, (ii) the solutions found will closely satisfy the required system criteria that each policy option must possess, as represented by the constraints stated within the problem.

Consequently, SO procedures used in an MGA context automatically generate a set of very good policy alternatives and most of these options would not have been considered by planners during a normal policy setting phase. Furthermore, all of these policy solutions will have their inherent planning uncertainty directly integrated into the generation of these alternatives. SO's direct integration of uncertainty into option generation produces major practical benefits in comparison to solutions created by any deterministic MGA procedure. Hence,

SO methods can be used directly as MGA procedures for problems containing uncertainty (Yeomans, 2002).

6. Case Studies of SO used for MGA

Obviously every solution difficulty encountered by SO in its function optimization role must carry-over into its use as an MGA, policy creation procedure. As described, SO optimization can be improved by including penalty functions and initial population biasing. In this section, these optimization improvements extended into MGA are directly illustrated using the case studies of (i) municipal solid waste planning from Yeomans et al. (2003) and (ii) the expansion of waste management facilities from Yeomans and Huang (2003).

6.1. Case 1: Policy Generation for Municipal Solid Waste Planning

Yeomans et al. (2003) studied Municipal Solid Waste planning in the Municipality of Hamilton-Wentworth which is located at the western-most tip of Lake Ontario. Within this MSW system, decisions had to be made regarding whether waste materials would be recycled, landfilled or incinerated and additional determinations had to be made as to which specific facilities would process the discarded materials. Included within these decisions was a determination of which one of the multiple possible pathways that the waste would flow through in reaching the facilities. Conversely, specific pathways selected for waste material flows determined which facilities processed the waste. It was possible to subdivide the various waste streams with each resulting substream sent to a different facility. Since cost differences from operating the facilities at different capacity levels produced economies of scale, decisions had to be made to determine how much waste should be sent along each flow pathway to each facility. Therefore, any single MSW policy option was composed of a combination of many decisions regarding which facilities received waste material and what quantities of waste were sent to each facility. All of these decisions were compounded by overriding system uncertainties.

The complete mathematical model used for MSW planning appears in the subsequent section. This mathematical formulation was used not only to examine the existing municipal MSW system, but also to prepare the municipality for several potentially enforced future changes to its operating conditions. Yeomans et al. (2003) examined three likely future scenarios, with each scenario involving potential incinerator operations (Huang et al., 1998). Scenario 1 considered the existing MSW management system and corresponded to a *status quo* case. Scenario 2 examined what would occur should the incinerator operate at its upper design capacity; corresponding to a situation in which the municipality would landfill as little waste as possible. Scenario 3 permitted the incinerator to operate anywhere in its design capacity range; from being closed completely to operating up to its maximum capacity.

6.1.1. Mathematical Model for Solid Waste Planning in the Municipality of Hamilton-Wentworth

This section provides the complete mathematical model for MSW planning in Hamilton-Wentworth. Extensive details and descriptions of it can be found in Huang et al. (1998) and Yeomans et al. (2003). Any uncertain parameter A is represented by \vec{A} in the model. Formulating any single MSW policy corresponds to finding a decision variable solution satisfying constraints (2) through (31), with cost determined by objective (1):

$$\text{Minimize Cost} = \sum_{p=1}^5 T_p + \sum_{q=1}^6 P_q - \sum_{r=2}^3 R_r \quad (1)$$

Subject to:

$$T_1 = \sum_{i=1}^{17} \sum_{j=1}^3 \vec{t}x_{ij}x_{ij}\vec{W}_i \quad (2)$$

$$T_2 = \sum_{i=1}^{17} \sum_{k=1}^3 \vec{t}z_{ik}z_{ik}\vec{W}_i \quad (3)$$

$$T_3 = \sum_{i=1}^{17} \sum_{j=1}^3 \sum_{k=1}^3 \vec{t}y_{jk}y_{jk}x_{ij}\vec{W}_i \quad (4)$$

$$T_4 = (\vec{t}sl)\vec{Q}_2 \sum_{i=1}^{17} \vec{W}_i \cdot [z_{i2} + \sum_{j=1}^3 y_{j2}x_{ij}] \quad (5)$$

$$T_5 = (\vec{t}tl)\vec{Q}_3 \sum_{i=1}^{17} \vec{W}_i \cdot [z_{i3} + \sum_{j=1}^3 y_{j3}x_{ij}] \quad (6)$$

$$P_1 = \vec{\rho}_1 \sum_{i=1}^{17} \vec{W}_i \sum_{k=1}^3 [\vec{Q}_k z_{ik} + \sum_{j=1}^3 x_{ij}y_{jk}] \quad (7)$$

$$P_2 = \vec{\rho}_2 \sum_{i=1}^{17} \vec{W}_i \cdot [z_{i2} + \sum_{j=1}^3 x_{ij}y_{j2}] \quad (8)$$

$$P_3 = \vec{\rho}_3 \sum_{i=1}^{17} \vec{W}_i \cdot [z_{i3} + \sum_{j=1}^3 x_{ij}y_{j3}] \quad (9)$$

$$P_4 = \vec{\delta}_1 \sum_{i=1}^{17} x_{i1}\vec{W}_i \quad (10)$$

$$P_5 = \vec{\delta}_2 \sum_{i=1}^{17} \vec{W}_i \cdot [x_{i2} + \vec{Q}_3 \{z_{i3} + \sum_{j=1}^3 x_{ij}y_{j3}\}] \quad (11)$$

$$P_6 = \vec{\delta}_3 \sum_{i=1}^{17} x_{i3}\vec{W}_i \quad (12)$$

$$R_2 = \vec{r}_2 \sum_{i=1}^{17} \vec{W}_i \cdot [z_{i2} + \sum_{j=1}^3 x_{ij}y_{j2}] \quad (13)$$

$$R_3 = \vec{r}_3 \sum_{i=1}^{17} \vec{W}_i \cdot [z_{i3} + \sum_{j=1}^3 x_{ij}y_{j3}] \quad (14)$$

$$\sum_{i=1}^{17} x_{i1} \vec{W}_i \leq \vec{M}_1 \tag{15}$$

$$\sum_{i=1}^{17} \vec{W}_i \cdot [x_{i2} + \vec{Q}_3 \{z_{i3} + \sum_{j=1}^3 x_{ij} y_{j3}\}] \leq \vec{M}_2 \tag{16}$$

$$\sum_{i=1}^{17} x_{i3} \vec{W}_i \leq \vec{M}_3 \tag{17}$$

$$\sum_{i=1}^{17} x_{i1} \vec{W}_i \geq \vec{K}_1 \tag{18}$$

$$\sum_{i=1}^{17} \vec{W}_i \cdot [x_{i2} + \vec{Q}_3 \{z_{i3} + \sum_{j=1}^3 x_{ij} y_{j3}\}] \geq \vec{K}_2 \tag{19}$$

$$\sum_{i=1}^{17} x_{i3} \vec{W}_i \geq \vec{K}_3 \tag{20}$$

$$\sum_{i=1}^{17} \vec{W}_i \sum_{k=1}^3 [\vec{Q}_k z_{ik} + \sum_{j=1}^3 x_{ij} y_{jk}] \leq \vec{N}_1 \tag{21}$$

$$\sum_{i=1}^{17} \vec{W}_i \cdot [z_{i2} + \sum_{j=1}^3 x_{ij} y_{j2}] \leq \vec{N}_2 \tag{22}$$

$$\sum_{i=1}^{17} \vec{W}_i \cdot [z_{i3} + \sum_{j=1}^3 x_{ij} y_{j3}] \leq \vec{N}_3 \tag{23}$$

$$\sum_{i=1}^{17} \vec{W}_i \cdot [z_{i2} + \sum_{j=1}^3 x_{ij} y_{j2}] \geq \vec{L}_2 \tag{24}$$

$$\sum_{i=1}^{17} \vec{W}_i \cdot [z_{i3} + \sum_{j=1}^3 x_{ij} y_{j3}] \geq \vec{L}_3 \tag{25}$$

$$\sum_{j=1}^3 x_{ij} + \sum_{k=1}^3 z_{ik} = 1 \quad (i = 1, 2, \dots, 17) \tag{26}$$

$$\sum_{j=1}^3 x_{ij} \vec{R}W_j + z_{i3} \leq \vec{a}_i \quad (i = 1, 2, \dots, 17) \tag{27}$$

$$\sum_{k=1}^3 y_{jk} = 1 \quad (j = 1, 2, 3) \tag{28}$$

$$\sum_{i=1}^{17} \vec{W}_i \cdot [x_{i2} + \vec{Q}_3 \{z_{i3} + \sum_{j=1}^3 x_{ij} y_{j3}\}] = \sum_{i=1}^{17} \sum_{k=1}^3 x_{i2} \vec{W}_i y_{2k} \tag{29}$$

$$\sum_{i=1}^{17} x_{ij} \vec{W}_i y_{j3} = \vec{R}W_j \sum_{i=1}^{17} x_{ij} \vec{W}_i \quad (j = 1, 2, 3) \tag{30}$$

$$x_{ij} \geq 0, \quad y_{jk} \geq 0, \quad z_{ik} \geq 0 \quad (\forall i, j, k) \tag{31}$$

tem (Scenario 1), a solution that would never cost more than \$20.6 million was constructed. For Scenarios 2, and 3, Yeomans et al. (2003) produced optimal solutions costing \$22.1 million and \$18.7 million, respectively. In all of these scenarios, SO was used exclusively as a function optimizer with the goal being to produce only single best solutions.

6.1.2. Generating Policy Alternatives Using Simulation-Optimization

As described, optimizing the problem required running SO on the stochastic model to find the minimum system cost and the resulting terminal population provides the set of policy alternatives determined by SO in its capacity as an MGA procedure. To accelerate the search times and to improve solution quality, both penalty function penalization and GP were integrated into the solution process (Huang et al., 2005; Yeomans, 2005). This adaptation was accomplished with two modifications to the original model. Firstly, GP was used to generate the initial starting population. Secondly, the original hard-constrained SO model was re-formulated to penalize any candidate solution lying outside constraint limits (2) to (30). After incorporating these two modifications, a computational study was undertaken to investigate their potential performance improvement for MGA in comparison to the hard-constrained process.

Eighteen separate computational experiments were undertaken to compare the relative performance of the reformulated penalty-function/GP SO to the original constrained-SO approach. In this experimentation, initial populations of size 40 were created either by random generation for constrained-SO or by GP for the penalty-function procedure. Both procedures were run for fixed time intervals of 30 and 90 minutes. In addition, recognizing that constrained-SO might expend considerable computational effort in creating and discarding infeasible solutions, a third “time” period was included that required the search to evolve through exactly 50 population generations. A distinct set of policy options was generated for each of the three problem scenarios considered by the Municipality. Hence, each of the eighteen experimental settings corresponded to one specific combination of: (i) the two different SO solution procedures, (ii) the three different time periods permitted for solution search, and (iii) one of the three specific scenarios under investigation. Upon termination of each experimental run, the entire surviving population would represent the set of 40 different policy options generated for MSW management within the municipality.

Table 1 contrasts the values of the cost objectives for each of the 40 policy options in the terminal populations under the six respective experimental settings performed for Scenario 1. The lack of variability within each penalty function/GP column indicates that all solution costs fall within similar ranges, indicating that these populations are comparatively equivalent from a cost standpoint. A separate detailed examination of the resulting decision variables indicated that most of these options provided MSW system structures that were quite distinct from each other. From a practical perspective, this finding demon-

Yeomans et al. (2003) ran SO for a 24-hour period to determine best solutions for each scenario. For the existing sys-

Table 1. Maximum System Costs (\$ Millions) for the 40 Solutions in Terminal Populations under Scenario 1 (Costs have been Sorted in Non-Decreasing Order)

ESO Procedure	Penalty Function	Constrained ESO	Penalty Function	Constrained ESO	Penalty Function	Constrained ESO
Starting Solution:	GP	Random	GP	Random	GP	Random
Solution Time:	30 Minutes	30 Minutes	90 Minutes	90 Minutes	50 Iterations	50 Iterations
1	20.71	22.64	20.60	20.97	20.69	20.76
2	20.71	22.79	20.60	21.00	20.69	20.76
3	20.71	22.83	20.60	21.04	20.69	20.76
4	20.71	22.87	20.60	21.06	20.69	20.78
5	20.71	22.90	20.61	21.40	20.69	20.78
6	20.71	22.94	20.61	21.44	20.69	20.78
7	20.72	22.94	20.61	21.48	20.70	20.78
8	20.72	23.01	20.61	21.52	20.70	20.79
9	20.72	23.01	20.61	21.58	20.70	20.79
10	20.72	23.05	20.61	21.60	20.70	20.79
11	20.72	23.09	20.61	21.62	20.70	20.79
12	20.72	23.13	20.61	21.62	20.70	20.79
13	20.72	23.16	20.61	21.70	20.70	20.80
14	20.72	23.24	20.61	21.71	20.70	20.80
15	20.72	23.28	20.61	21.71	20.71	20.80
16	20.74	23.28	20.61	21.78	20.72	20.81
17	20.75	23.31	20.61	21.97	20.73	20.81
18	20.75	23.39	20.61	22.00	20.73	20.81
19	20.76	23.39	20.61	22.01	20.74	20.81
20	20.76	23.43	20.61	22.03	20.74	20.81
21	20.76	23.43	20.62	22.08	20.74	20.81
22	20.76	23.46	20.62	22.09	20.75	20.81
23	20.76	23.46	20.62	22.10	20.75	20.82
24	20.77	23.46	20.62	22.16	20.75	20.82
25	20.78	23.46	20.62	22.20	20.75	20.82
26	20.78	23.50	20.62	22.21	20.76	20.82
27	20.78	23.61	20.62	22.23	20.76	20.82
28	20.78	23.61	20.62	22.25	20.76	20.83
29	20.78	23.65	20.62	22.29	20.76	20.85
30	20.78	23.72	20.62	22.30	20.77	20.85
31	20.78	23.80	20.63	22.37	20.77	20.85
32	20.78	23.80	20.63	22.39	20.77	20.85
33	20.78	23.84	20.63	22.55	20.77	20.85
34	20.78	23.87	20.63	22.56	20.77	20.86
35	20.78	23.87	20.65	22.56	20.77	20.86
36	20.78	23.91	20.65	22.66	20.77	20.86
37	20.78	23.91	20.69	23.35	20.77	20.88
38	20.80	23.95	20.70	23.67	20.78	20.89
39	20.80	23.99	20.70	23.74	20.79	20.90
40	20.81	23.99	20.75	23.80	20.80	20.90

strates that penalty function/GP SO can generate considerably more lower cost policy alternatives than the limited number of options that might be produced by a planner using an optimization technique.

Since each alternative provides different waste flow policies within the system, this leads to different utilizations of the various waste processing facilities in the Municipality. In its MGA capacity, SO has produced 40 different policy alternatives possessing the requisite system characteristics with ea-

ch option providing a different planning perspective. It should be noted that *all* 40 alternatives in each penalty function/GP setting have system costs that are at least \$200,000 cheaper than the existing municipal system. Because each alternative represents a distinct policy option of system utilization and *every one* of these options possesses a cost that is lower than the existing MSW policy, this clearly indicates that SO has created an entire set of improved policies for MSW processing in the municipality.

The constrained-SO columns in Table 1 show the solution alternatives created when starting from randomly generated initial populations. In comparison to penalty function/GP, the constrained-SO populations exhibit considerably more cost variability, representing a significantly more diverse set of policy options for the planners to choose from. Thus, if the goal of an MGA approach is to produce solution alternatives that vary significantly from one another, the experimentation indicates that SO produces more desirable results when starting from random populations. However, the best cost of all of the solutions found in these populations are higher than the worst solutions produced by the GP starting point, and the highest cost solutions are considerably higher. Therefore, it appears quite apparent that the general solution quality for the SO procedure beginning from a random initial population has deteriorated substantially in comparison to having the same procedure commence from a GP generated population.

Evaluating the population characteristics from Table 1 from a search time perspective highlights certain other differentiating features between the SO approaches. For the 30 and 90 minute search times, penalty-function/GP produced vastly superior populations than constrained-SO. While the two populations generated by penalty-function/GP resemble each other in both time periods, in constrained-SO the 90 minute search produced a markedly superior population relative to the 30-minute search. In the 50-generation experiment, penalty-function/GP produced a similar population to its 30 minute trial. Most significantly, the 50-generation constrained-SO experiment demonstrated considerable solution improvement over its 30- and 90-minute runs; providing alternatives that appear comparable in value to the penalty-function/GP solutions. However, an extremely important observation from the 50-generation experiments was that constrained-SO required in excess of 3 hours to iterate through its 50 generations, while penalty-function/GP took only 45 minutes. Hence, this experimentation reveals that longer solution searches in constrained-SO produce markedly superior populations and that penalty-function/GP produces better populations than constrained-SO in much shorter time-spans. Scenarios 2 and 3 were also tested using the same experimental settings applied to Scenario 1 and produced analogous observations to all of those noted above.

ve.

In addition to the various populations of alternatives created, it is also instructive to examine the best overall solution produced under each experimental setting (i.e. solution number 1 in each column of Table 1) and to directly compare these to the optimal values of \$20.6 million, \$22.1 million and \$18.7 million found by Huang et al. (1998) and Yeomans et al. (2003) for Scenarios 1, 2, and 3, respectively. Table 2 summarizes the single, best objective values obtained for Scenario 1 under the indicated settings. The table indicates that the constrained-SO solutions improve with longer search times and that the 3-hour, 50-generation experiment produced a solution within \$300,000 of optimality. In stark contrast, all of the penalty function/GP experiments produced near-optimal solutions with the 30-minute and 50-generation runs within \$100,000 of the best-known solution, and the 90-minute trial achieving optimality. These findings confirm the search-time benefits of penalty function/GP in that the 30-minute trial produced a solution costing \$200,000 less than the 3-hour constrained-SO trial. Similar patterns demonstrating the superiority of penalty function/GP over constrained SO were found in the solutions to Scenarios 2 and 3. Table 3 contrasts the optimal solutions for each scenario with the best solutions produced by penalty-function/GP in the 90-minute experiments. This table demonstrates that penalty function/GP generated an optimal solution for each scenario.

Analogously to issues surrounding solution quality, it had been mentioned that the search times of evolutionary methods are stochastic. Hence, constrained-SO commencing from random starting points should prove capable of producing solutions comparable to penalty-function/GP if longer search periods had been permitted. To illustrate this phenomenon, an additional 24-hour, constrained-SO experiment was performed for Scenario 1 and the last column of Table 3 shows the best solution found from this extended-time experimentation. The solution demonstrates a dramatic value improvement when contrasted against the earlier results and suggests that with extended search times, the best solutions found by constrained-SO do, indeed, become comparable to penalty-function/GP SO.

In totality, these observations underscore numerous impor-

Table 2. Optimum Annual MSW Planning Costs (in millions of \$) for the Existing System Structure (Scenario 1)

SO Method & Starting Condition Employed	Solution Search Time			
	30 Minutes	90 Minutes	50 Iterations	24 Hours
Constrained SO & Random Starting Solution	22.6	20.9	20.8	20.7
Penalty Function SO & GP Starting Solution	20.7	20.6	20.7	N/A

Table 3. Annual Cost (in millions of \$) of Best Solution found in 90 Minutes by Penalty Function SO for Each Scenario Compared to Optimal Solutions in Yeomans *et al.* (2003)

Method	Scenario		
	Scenario 1	Scenario 2	Scenario 3
Yeomans et al. (2003) using SO	20.6	22.1	18.7
Penalty Function SO & GP Starting Solution	20.6	22.1	18.7

tant points for contrasting the two SO approaches when used either for MGA or even for optimization: (i) That the combination of GP and penalty function minimization in directing the evolutionary search toward good solutions early-on in the procedure clearly proves extremely beneficial both from a solution time and solution quality viewpoint; (ii) That in producing sets of near-optimal alternatives in addition to essentially overall optimal solutions, penalty-function/GP acts as an extremely efficient MGA approach for large-scale, stochastic problem settings; (iii) That for the same number of generations, penalty-function/GP produces many more better solution alternatives than constrained-SO; (iv) Constrained-SO must clearly expend considerable search effort evaluating and discarding infeasible solutions, and; (v) Constrained-SO requires on the order of 4 times longer to create and evaluate the same number of generations as the penalty-function/GP. Consequently, it becomes readily apparent from this experimentation that superior quality solution alternatives are produced by penalty-function/GP procedures in direct comparison to having constrained-SO procedures start from randomly-generated initial populations and that these better solutions are produced in considerably less time.

6.2. Case 2: Policy Generation for the Expansion of Waste Management Facilities

After optimizing an initial problem formulation, the deterministic Hop, Skip, and Jump (HSJ) MGA technique creates supplementary problem instances by systematically adding target constraints on both the objective function value and the decision variables to force the generation of solution alternatives (Brill, 1979; Brill et al., 1981). Huang et al. (1996b) combined GP with HSJ modelling to construct a procedure referred to as the Grey, Hop, Skip and Jump (GHSJ) method. Yeomans and Huang (2003) extended GHSJ by incorporating evolutionary SO into its solution process. The utility of each of these approaches were demonstrated on an MSW facility expansion application.

6.2.1. Mathematical Model for the Expansion of Waste Management Facilities

This section outlines the complete mathematical model used for the municipal waste management facility expansion case from Huang et al. (1996b), while more extensive details for it can also be found in Yeomans and Huang (2003).

$$\text{Minimize Cost} = \sum_{k=1}^3 \bar{F}LC_k y_k + \sum_{i=2}^3 \sum_{m=1}^4 \sum_{k=1}^3 \bar{F}TC_{imk} z_{imk} + \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 L_k \bar{C}_{ijk} \bar{W}G_{jk} x_{ijk} \quad (32)$$

Subject to:

$$\sum_{j=1}^3 \sum_{k=1}^{k'} L_k \cdot [\bar{W}G_{jk} x_{1jk} + \sum_{i=2}^3 \bar{W}G_{jk} x_{ijk} \cdot FE] \leq \sum_{k=1}^{k'} \bar{\Delta}LC_k y_k + \bar{L}C \quad (k' = 1, 2, 3) \quad (33)$$

$$\sum_{j=1}^3 \bar{W}G_{jk} x_{ijk'} \leq \sum_{m=1}^4 \sum_{k=1}^{k'} \bar{\Delta}TC_{imk} z_{imk} + \bar{T}C_i \quad (i = 2, 3; k' = 1, 2, 3) \quad (34)$$

$$\sum_{i=1}^3 \bar{W}G_{jk} x_{ijk} \leq \bar{W}G_{jk} \quad (k = 1, 2, 3; j = 1, 2, 3) \quad (35)$$

$$\sum_{m=1}^4 z_{imk} \leq 1 \quad (i = 2, 3; k = 1, 2, 3) \quad (36)$$

$$\sum_{k=1}^3 y_k \leq 1 \quad (37)$$

$$\sum_{i=1}^3 x_{ijk} = 1 \quad (k = 1, 2, 3; j = 1, 2, 3) \quad (38)$$

$$1 \geq x_{ijk} \geq 0 \quad (i = 1, 2, 3; j = 1, 2, 3; k = 1, 2, 3) \quad (39)$$

$$y_k = 0 \text{ or } 1 \quad (k = 1, 2, 3) \quad (40)$$

$$z_{imk} = 0 \text{ or } 1 \quad (i = 2, 3; k = 1, 2, 3; m = 1, 2, 3, 4) \quad (41)$$

In the MSW facility expansion case, the studied region contains three different municipalities whose MSW disposal needs are served by a landfill and two waste-to-energy (WTE) incineration facilities. At the start of the planning period, the landfill possesses an existing maximum capacity and both WTE facilities have given processing capacities. Three time periods are considered with each period covering an interval of five years. The landfill capacity can be expanded only once over the entire 15 year planning horizon, while each of the WTE facilities can be expanded by any one of four possible options in each of the three time periods. The expansion costs escalate over time reflecting anticipated future conditions and both the waste generation rates and transportation and treatment costs vary temporally and spatially.

The case requires the determination of the preferred facility expansion alternatives during the different time periods and the determination of an effective allocation of the relevant waste flows that minimizes the total system costs over the planning horizon. Using GHSJ, Huang et al. (1996b) optimized the problem and then created three alternative expansion options by placing a technical constraint on the objective function that increased system costs by steps of 2%, 5%, and 8%, respectively. To improve upon the 4 alternatives created by GHSJ, Yeomans and Huang (2003) ran the SO MGA procedure separately using each of the 4 GHSJ alternatives as the initial evolutionary population.

6.2.2. Policy Generation for the Expansion of Waste Management Facilities

In this section, the MSW facility expansion case is further extended by adding penalty functions (Yeomans, 2005) to

Table 4. System Expansion Costs (\$ Millions) for the 40 Solutions in Terminal Populations

Starting Solution	Randomly Generated	Optimal GHSJ Solution	2% GHSJ Solution	5% GHSJ Solution	8% GHSJ Solution
ESO Procedure	Hard Constrained	Penalty Function	Penalty Function	Penalty Function	Penalty Function
1	668.89	602.19	600.19	604.19	606.19
2	670.95	602.19	600.19	604.33	607.26
3	673.16	602.40	600.57	604.50	607.47
4	674.02	602.57	600.66	604.63	607.58
5	694.71	602.58	600.76	604.66	611.46
6	697.20	602.65	600.77	604.91	612.65
7	699.32	603.11	600.77	605.16	613.28
8	701.86	603.26	600.81	605.16	615.42
9	705.75	603.37	601.94	605.44	615.66
10	706.71	603.38	601.95	606.05	615.91
11	707.72	603.47	601.98	606.30	615.93
12	707.81	603.47	602.10	606.37	617.31
13	712.66	603.49	602.14	606.46	617.36
14	713.33	603.54	602.21	607.00	618.40
15	713.52	603.54	602.23	610.98	621.15
16	717.65	603.63	602.25	612.81	621.15
17	728.64	603.63	602.33	613.61	621.66
18	730.70	603.64	602.60	613.62	621.83
19	731.38	603.67	603.05	613.72	622.19
20	732.82	603.69	603.24	614.09	622.60
21	735.74	603.76	603.38	614.38	622.93
22	735.89	603.76	603.62	614.42	622.96
23	736.85	603.80	603.68	614.76	623.20
24	740.06	603.81	603.74	615.29	624.01
25	742.66	604.09	603.92	616.15	624.31
26	743.47	604.22	605.14	616.95	624.66
27	744.34	604.39	605.16	617.59	624.86
28	745.58	604.43	606.20	617.70	625.08
29	748.32	604.54	607.53	617.84	627.30
30	748.99	604.64	608.11	617.85	627.54
31	753.21	604.71	608.19	617.86	627.89
32	754.17	604.91	611.49	618.07	630.00
33	764.01	605.02	611.49	618.24	630.03
34	764.11	606.29	611.50	618.53	630.12
35	764.49	606.82	611.50	618.54	631.52
36	770.06	610.23	611.91	618.54	632.91
37	812.06	610.49	612.07	618.56	642.03
38	830.96	611.19	612.19	620.87	646.76
39	835.44	611.51	619.58	622.03	647.89
40	838.79	615.79	624.48	624.35	648.72

the evolutionary SO approach used by Yeomans and Huang (2003) for the model of Section 6.2.1. Consequently, the effectiveness of directing the start of SO from the solutions created by GHSJ in combination with penalty function minimization will be demonstrated.

Five separate policy-generation experiments for MSW expansion were conducted. To establish a comparative base case, the first experiment ran SO without penalty functions and with the search starting from a randomly generated initial population. The remaining four experiments included penalty functions and used each of the four GHSJ solution alternatives from

Huang et al. (1996b), respectively, as the starting population. These four starting points represent the overall optimal solution for the original GHSJ model, together with the 2%, 5%, and 8% alternatives generated when target constraints had been included during the alternative creation phase. In each experiment, SO was run with for exactly 50 population iterations. The columns in Table 4 show the objectives found for each of the 40 policy options residing in the terminal populations of each experiment sorted from largest to smallest. An analysis of the output highlights several interesting characteristics regarding the SO search procedure.

One key evaluation criteria involves a comparison of the search times between the two different SO approaches. Although both the hard-constrained and penalty-function approaches each iterated through exactly 50 generations, the former procedure required in excess of 3 hours to complete, while the latter required on the order of 10 to 15 minutes. This indicates that hard-constrained SO clearly expends a considerable portion of its search efforts in the evaluation, disposal, and correction of infeasible solutions. Obviously, the penalty-function transformation to an unconstrained problem holds distinct practical advantages with respect to solution time.

In comparison to the 4 expansion options created by GHSJ, each implementation of SO produces 40 different alternatives. In Table 4, all of the costs cluster at the lower end of the cost ranges produced by GHSJ. This demonstrates that SO can be used to not only generate considerably more decision options than GHSJ, but also to create these policy alternatives such that their costs are all lower than those produced by GHSJ. The experimentation also demonstrates an apparent benefit of commencing SO from a relatively good, but not necessarily optimal, GP starting population. This characteristic became evident even when starting from the slightly suboptimal, 2% GHSJ alternative. The suboptimal starting point seems to allow more leeway in permitting the search to evolve, while at the same time creating more structurally diverse policy alternatives. The final population sets generated from the 5% and 8% GHSJ alternatives produce even more variety in the policy options created, and could quite conceivably have determined comparable cost values to the other starting points had additional search iterations been permitted. Good practice might, therefore, dictate that starting SO from a suboptimal GP alternative would produce more solution variety in the policies generated.

In totality, these findings highlight several important observations regarding the use of SO for MGA: (i) SO generates many more good alternatives than planners would have created using other MGA approaches; (ii) All solutions produced by SO directly incorporate system uncertainty during their creation unlike earlier MGA methods; (iii) The combination of penalty functions with GP is highly effective in rapidly allowing SO to generate a number of good solutions; (iv) The combination of penalty functions with GP directs the evolutionary search toward good solutions early-on in the procedure, providing major solution time and solution quality benefits; (v) Hard-constrained SO procedures clearly devote (or waste) a considerable proportion of their search energies in evaluating and correcting infeasible solutions; (vi) A good GP starting point proves valuable in directing SO toward better solution regions earlier in the search; (vii) Penalty functions permit a perturbation about good solutions without becoming "bogged down" in correcting infeasibilities, thereby allowing a much more extensive and productive search; (viii) After the same number of iterations, SO with penalty-functions produces better alternatives than hard-constrained SO; (ix) Penalty function SO is at least an order of magnitude faster than hard-constrained SO, and; (x) By producing sets of near-optimal alternatives in addition to essentially overall optimal solutions, the

penalty-function approach provides an extremely efficient MGA method for stochastic problems. Consequently, SO procedures run in conjunction with penalty-functions and GP produce superior quality alternatives in considerably less time in direct comparison to hard-constrained SO.

7. Conclusions

In summary, environmental policy formulation is a complicated process that can be impacted upon by a multitude of uncertain factors. Any ancillary technique to support policy generation must address all of these features and must be flexible enough to encapsulate the impacts from inherent planning uncertainty. It has been recognized that SO can be improved by examining solutions lying outside its feasible region via penalty functions and that the computationally efficient GP procedure can be used to quickly generate a good starting population for SO. In combination, penalty function minimization together with GP improve SO's search time and solution quality when solving stochastic problems.

In this paper, it was shown how SO could be used to efficiently generate multiple near-best policy alternatives for difficult, stochastic, environmental problems and the effectiveness of this MGA approach was illustrated using two case studies. In its MGA capacity, SO produces many solutions possessing the requisite characteristics of the system, with each alternative providing a different planning perspective. Since SO techniques can easily be adapted to many different stochastic problems, the practicality of this approach can clearly be extended into many other environmental planning applications containing significant sources of uncertainty.

Acknowledgments. This work was supported in part by grant OGP0155871 from the Natural Sciences and Engineering Research Council.

References

- Azadivar, F. (1999). Simulation Optimization Methodologies, *Proceedings of the 1999 Winter Simulation Conference*, December 5-8, Phoenix, AZ, 93-100.
- Azadivar, F., and Tompkins, G. (1999). Simulation Optimization with Qualitative Variables and Structural Model Changes: A Genetic Algorithm Approach, *European Journal of Operational Research*, 113, 169-182, doi:10.1016/S0377-2217(97)00430-X.
- Baetz, B.W. (1990). Optimization/Simulation Modeling for Waste Management Capacity Planning, *ASCE J. Urban Plann. Dev.*, 116 (2), 59-79.
- Baetz, B.W., Pas, E.I., and Neebe, A.W. (1990). Generating alternative solutions for dynamic programming-based planning problems, *Socio-Economic Planning Sciences*, 24, 27-34, doi:10.1016/0038-0121(90)90025-3.
- Bass, B., Huang, G.H., and Russo, J. (1997). Incorporating Climate Change into Risk Assessment using Grey Mathematical Programming, *J. Environ. Manage.*, 49, 107-124.
- Bodner, R.M., Cassell, A., and Andros, P.J. (1970). Optimal Routing of Refuse Collection Vehicles, *ASCE Journal of the Sanitary Engineering Division*, 96, 893-903.
- Brill, E.D. (1979). The use of optimization models in public sector planning, *Manage. Sci.*, 25, 413-422.

- Brill, E.D., Chang, S.Y., and Hopkins, L.D. (1981). Modelling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning, *Manage. Sci.*, 27, 314-325.
- Brown, R.V., Kahr, A.S., and Peterson, C. (1974). *Decision Analysis for the Manager*, Holt, Rinehart and Winston, New York, NY.
- Caudill, M. and Butler, C. (1990). *Naturally Intelligent Systems*, MIT Press, Cambridge, MA.
- Chang, S. Y., Brill, E.D., and Hopkins, L.D. (1980). Efficient random generation of feasible alternatives: a land use planning example. Staff Paper 12, *Institute for Environmental Studies*, University of Illinois at Urbana-Champaign, Urbana, IL.
- Chang, S.Y., Brill, E.D., and Hopkins, L.D. (1982). Use of mathematical models to generate alternative solutions to water resources planning problems, *Water Resour. Res.*, 18, 58-64.
- Chang, N.B., and Wang, S.F. (1995). A Grey Nonlinear Programming Approach for Planning Coastal Wastewater Treatment and Disposal Systems, *Water Science and Technology*, 32, 19-29.
- Chang, N.B., Wen, C.G., and Yong, Y.C. (1996). A Grey Fuzzy Multiobjective Programming Approach for Optimal Planning of a Reservoir Watershed, *Water Res.*, 30, 2329-2334.
- Church, R.L., and Huber, D.L. (1979). On finding close to optimal solutions as well as determining a noninferior set of tradeoff envelopes for multi-objective location models, *Research Series 34*, Department of Civil Engineering, University of Tennessee, Knoxville, TN.
- Coyle, R.G. (1973). *Computer-based Design for Refuse Collection Systems*, In R. Deininger (Ed.), *Models for Environmental Pollution Control* (pp307-325), Ann Arbor Science, Ann Arbor, MI.
- Falkenhausen, K. (1979). Facility location planning for regional waste treatment systems by branch and bound vs. evolution strategy concepts, *Presented at the ORSA/TIMS Milwaukee Conference*.
- Ferrell, W.G., and Hizlan, H. (1997). South Carolina Counties Use a Mixed-Integer-Programming Based Decision Support Tool for Planning Municipal Solid Waste Management, *Interfaces*, 27, 4, 23-34, doi:10.1287/inte.27.4.23.
- Fu, M.C. (1994). Optimization for Simulation: Theory vs. Practice, *INFORMS Journal on Computing*, 14 (3), 192-215, doi:10.1287/ijoc.14.3.192.113.
- Fu, M.C. (2002). Optimization via Simulation: A Review, *Annals of Operational Research*, 53, 199-248, doi:10.1007/BF02136830.
- Gendreau, M. (2002). Metaheuristics in Vehicle Routing, *Presented at the Canadian Operational Research Society (CORS/SCRO) Meeting*, Toronto.
- Gidley, J.S., and Bari, M.F. (1986). Modelling to Generate Alternatives, *ASCE Water Forum '86*, 1366-1373.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Reading, MA.
- Goldberg, D.E. (1991). *Genetic Algorithms*, Addison-Wesley, Reading, MA.
- Gottinger, H.W. (1986). A Computational Model for Solid Waste Management with Applications, *Applied Mathematical Modelling*, 10, 330-338.
- Hasit, Y., and Warner, D.B. (1981). Regional Solid Waste Planning with WRAP, *J. Environ. Eng.*, 107, 511-525.
- Haynes, L. (1981). A Systems Approach to Solid Waste Management Planning, *Conservation and Recycling*, 4 (2), 67-78.
- Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems 2nd Edition*, MIT Press, Cambridge, MA.
- Huang, G.H. (1996). IPWM: An Interval Parameter Water Quality Management Model, *Eng. Optimiz.*, 26, 79-103, doi:10.1080/03052159608941111.
- Huang, G.H., Baetz, B.W., and Patry, G.G. (1994a). *A Grey Chance-Constrained Programming Approach for Waste Management Planning under Uncertainty*, In Hipel, K.W. and Fang, L. (Eds.), *Effective Environmental Management for Sustainable Development* (267-280), Kluwer Academic Publishers, Dordrecht, Netherlands.
- Huang, G.H., Baetz, B.W., and Patry, G.G. (1994b). Grey Dynamic Programming for Solid Waste Management Planning under Uncertainty, *ASCE J. Urban Plann. Dev.*, 120, 132-156, doi:10.1061/(ASCE)0733-9488(1994)120:3(132).
- Huang, G.H., Baetz, B.W., and Patry, G.G. (1995). A Grey Integer Programming Approach for Waste Management Planning, *European Journal of Operational Research*, 83, 594-620.
- Huang, G.H., Baetz, B.W., and Patry, G.G. (1996a). A Grey Hop, Skip and Jump Method for Generating Decision Alternatives: Planning for the Expansion/Utilization of Waste Management Facilities, *Can. J. Civ. Eng.*, 23, 1207-1219.
- Huang, G.H., Cohen, S.J., Yin, Y.Y., and Bass, B. (1996b). Incorporation of Inexact Integer Programming and Fuzzy Relation Analysis for Integrated Environmental Impact Assessment and Adaptation Study under Uncertainty, *J. Environ. Manage.*, 48, 45-68.
- Huang, G.H., Jorgensen, S.E., Xu, Y., and Guo, H.C. (1997). Integrated Environmental Planning for Sustainable Development in Lake Erhai Basin - with a Diagnostic Study for Local Environmental Concerns, *United Nations Environment Programme*, Nairobi, Kenya.
- Huang, G.H., Baetz, B.W., and Patry, G.G. (1998). Trash-Flow Allocation: Planning under Uncertainty, *Interfaces*, 28 (6), 36-55.
- Huang, G., Linton, J., Yeomans, J.S. and Yoogalingam, R. (2005). Policy Planning Under Uncertainty: Efficient Starting Populations for Simulation-Optimization Methods Applied to Municipal Solid Waste Management, *J. Environ. Manage.*, 77(1), 22-34.
- Kelly, P. (2002). Simulation Optimization is Evolving, *INFORMS Journal on Computing*, 14 (3), 223-225, doi:10.1287/ijoc.14.3.223.108.
- Lacksonen, T. (2001). Empirical Comparison of Search Algorithms for Discrete Event Simulation, *Computers and Industrial Engineering*, 40, 133-148, doi:10.1016/S0360-8352(01)00013-4.
- Law, A.M. and Kelton W.D. (2000). *Simulation Modeling and Analysis (3rd edn)*, McGraw-Hill, New York, NY.
- Liebman, J.C. (1975). *Models in Solid Waste Management*, In Gass, S. and Sisson R. (Eds.), *A Guide to Models in Governmental Planning and Operations*, Sauger Books, Potomac, MD.
- Linton, J.D., Yeomans, J.S., and Yoogalingam R. (2002). Policy Planning using Genetic Algorithms Combined with Simulation: The Case of Municipal Solid Waste, *Environ. Plann. B*, 29 (5), 757-778.
- Lund, J.R. (1990). Least Cost Scheduling of Solid Waste Recycling, *ASCE J. Environ. Eng.*, 116, 182-197, doi:10.1061/(ASCE)0733-9372(1990)116:1(182).
- Lund, J.R., Tchobanoglous, G., Anex, R.P., and Lawver, R.A. (1994). Linear Programming for Analysis of Material Recovery Facilities, *ASCE J. Environ. Eng.*, 120, 1082-1094, doi:10.1061/(ASCE) 0733-9372(1994)120:5(1082).
- MacDonald, M. (1996). Bias Issues in the Utilization of Solid Waste Indicators, *Journal of the American Planning Association*, 62, 236-242, doi:10.1080/01944369608975687.
- Marks, D.H., and Liebman, J.C. (1971). Location Models: Solid Waste Collection Example, *ASCE J. Urban Plann. Dev.*, 97 (1), 15-30.
- Openshaw, B.W., and Whitehead, P. (1985). A Monte Carlo Simulation Approach to Solving Multicriteria Optimization Problems Related to Plan Making, Evaluation, and Monitoring in Local Plan-

- ning, *Environ. Plann. B*, 12, 321-334.
- Pierreval, H., and Tautou, L. (1997). Using Evolutionary Algorithms and Simulation for the Optimization of Manufacturing Systems, *IIE Transactions*, 29(3), 181-189, doi:10.1023/A:1018562927155.
- Reeves, C.R. (1993). *Modern Heuristic Techniques for Combinatorial Problems*, John Wiley and Sons, New York, NY.
- Rubenstein-Montano, B., Anandalingam, G., and Zandi, I. (2000). A Genetic Algorithm Approach to Policy Design for Consequence Minimization, *European Journal of Operational Research*, 124, 43-54.
- Rubenstein-Montano, B., and Zandi, I. (1999). Application of a Genetic Algorithm to Policy Planning: the Case of Solid Waste, *Environ. Plann. B*, 26, 893-907.
- Tchobanoglous, G., Thiesen, H., and Vigil, S. (1993). *Integrated Solid Waste Management: Engineering Principles and Management Issues*, McGraw-Hill, New York, NY.
- Walker, W.E. (1976). A Heuristic Adjacent Extreme Point Algorithm for the Fixed Charge Problem, *Manage. Sci.*, 22, 587-596.
- Wang, F.S., Richardson, F.A., Richardson, A.J., and Curnow, R.C. (1994). SWIM-Interactive Software for Continuous Improvement of Solid Waste Management, *Journal of Resource Management and Technology*, 22 (2), 63-72.
- Wenger, R.B., and Cruz-Urbe, B.W. (1990). Mathematical Models in Solid Waste Management: A Survey, *Presented at the TIMS/ORSA Conference Las Vegas*.
- Yeh, S.C. (1996). *Grey Linear Programming and Its Application to Water Resources Management*, Doctoral dissertation, School of Civil and Environmental Engineering, Cornell University, Ithaca, NY.
- Yeomans, J.S. (2002). Automatic Generation of Efficient Policy Alternatives Via Simulation-Optimization, *J. Oper. Res. Soc.*, 53(11), 1256-1267, doi:10.1057/palgrave.jors.2601439.
- Yeomans, J.S. (2005). Planning Using Evolutionary Simulation-Optimization Combined with Penalty Functions, *Working Paper*, York University, Toronto, Ontario.
- Yeomans, J.S. (2006). Solid Waste Policy Planning Under Uncertainty Using Evolutionary Simulation-Optimization, *Socio-Econ. Plann. Sci.*, 40(3).
- Yeomans, J.S., and Huang, G. (2003). An Evolutionary Grey, Hop, Skip, and Jump Approach: Generating Alternative Policies for the Expansion of Waste Management Facilities, *J. Env. Inform.*, 1(1), 37-51, 2003, doi:10.3808/jei.200300005.
- Yeomans, J.S., Huang, G., and Yoogalingam, R. (2003). Combining Simulation with Evolutionary Algorithms for Optimal Planning Under Uncertainty: An Application to Municipal Solid Waste Management Planning in the Regional Municipality of Hamilton-Wentworth, *J. Env. Inform.*, 2(1), 11-30, doi:10.3808/jei.200300014.