Open Access

SCHOLARENA

Journal of Waste Management and Disposal

RESEARCH ARTICLE

ISSN: 2641-8827

Waste Management Using Multicriteria Population-Based Simulation-Optimization Algorithms

Yeomans JS*

OMIS Area, Schulich School of Business York University, Toronto, Canada

***Corresponding author:** Yeomans JS, OMIS Area, Schulich School of Business York University, 4700 Keele Street Toronto, ON, M3J 1P3 Canada, Tel: (416) 736-5074, E-mail: syeomans@schulich.yorku.ca

Citation: Yeomans JS (2019) Waste Management Using Multicriteria Population-Based Simulation-Optimization Algorithms. J Waste Manag Disposal 2: 109

Article history: Received: 15 June 2019, **Accepted:** 25 June 2019, **Published:** 27 June 2019

Abstract

When resolving waste management (WM) planning problems, it is often preferable to construct a number of quantifiably good alternatives that provide multiple, disparate perspectives. This is because solid waste planning generally involves complicated problems that are riddled with incompatible performance objectives and contain inconsistent design requirements that are very difficult to quantify and capture when supporting decision models must be constructed. These potential alternatives need to satisfy the required system performance criteria and yet be maximally different from each other in the decision space. The approach for creating maximally different sets of solutions is referred to as modelling-to-generate-alternatives (MGA). Simulation-optimization approaches have frequently been employed to solve computationally difficult problems containing the significant stochastic uncertainties in waste management. This paper outlines a multicriteria MGA approach for WM planning that can generate sets of maximally different alternatives for any simulation-optimization method that employs a population-based solution algorithm. This algorithmic approach is computationally efficient because it simultaneously produces the prescribed number of maximally different solution alternatives in a single computational run of the procedure. The efficacy of this stochastic MGA approach for creating alternatives is demonstrated using a "real world" waste management planning case.

Keywords: Waste management; Modelling-to-generate-alternatives; Simulation-Optimization; Population-based algorithms

Introduction

Planners have faced complex questions related to the processing and management of waste for many decades [1-3]. Implementing effective management of waste management (WM) systems has proven to be both notoriously contentious and conflict-laden. Since WM systems generally contain all of the characteristics associated with complex planning situations, waste management problems have provided an ideal backdrop for the testing of an extensive assortment of decision support techniques used for decision-making [4-8]. WM decision-making frequently possesses inconsistent and incompatible design specifications, that can be difficult to formulate into supporting mathematical decision-models [1-7,9-13]. This situation commonly occurs when final decisions must be constructed based not only upon clearly articulated specifications, but also upon environmental, political and socio-economic objectives that are either fundamentally subjective or not clearly articulated [14-17]. Although "optimal" solutions can be determined for the formulated mathematical models, whether these can be considered the best solution to the "real" problem remains somewhat dubious. Moreover, it may not be possible to explicitly convey many of the subjective considerations because there are numerous competing, adversarial stakeholder groups holding diametrically opposed perspectives. Therefore, many of the subjective aspects remain unknown, unquantified and unmodelled in the construction of the corresponding decision models. WM policy formulation can prove even more complicated when the various system components also contain stochastic uncertainties [17,18]. Consequently, waste management determination proves to be an extremely challenging and complicated enterprise [17-19].

Within WM decision-making, there are routinely many stakeholder groups holding completely incongruent standpoints, essentially dictating that waste managers need to construct decision frameworks that somehow simultaneously reflect numerous irreconcilable points of view [4,10,11,20,21]. Under such circumstances, it is often more desirable to construct a small number of distinct alternatives that provide dissimilar viewpoints for the particular problem [4,13,14]. These dissimilar solutions should be close-to-optimal with respect to the specified objective(s), but be maximally different from each other within the decision domain. Numerous approaches collectively referred to as *modelling-to-generate-alternatives* (MGA) have been created to address this multi-solution requirement [7-15,22-27]. The principal motivation behind MGA is the production of a set of alternatives that are "good" with respect to the specified objective(s), but fundamentally dissimilar from each other in the decision space [6]. Decision-makers then perform a subsequent appraisal of this set of alternatives to determine which option(s) most closely satisfy their specific goals. Consequently, MGA approaches are classified as decision support methods rather than as solution creation processes as assumed in traditional optimization.

Early MGA algorithms employed direct, incremental approaches for constructing their alternatives by iteratively re-running their procedures whenever new solutions needed to be generated [3,7-17,21,28-30]. These iterative approaches replicated the seminal MGA technique of Brill, *et al.* [15] where, once the initial mathematical formulation has been optimized; all supplementary alternatives are produced one-at-a-time. Therefore, these approaches all required n+1 iterations of their respective algorithms – firstly to optimize the original problem, then to construct each of the n subsequent alternatives [14,30-37].

In this paper, it is demonstrated how a set of maximally different solution alternatives can be generated by extending several earlier MGA approaches to stochastic optimization [31-37]. The stochastic algorithm provides an MGA process that can be performed by any population-based solution mechanism. This algorithm advances earlier concurrent procedures [32,34-37] by permitting the simultaneous generation of n distinct alternatives in a single computational run. Specifically, to generate n maximally different alternatives, the algorithm runs exactly the same number of times that a function optimization procedure needs to run (i.e. once) irrespective of the value of n [38-42]. A multicriteria objective is employed that combines a novel data structure into the simultaneous solution approach to create an effective MGA approach. The use of this data structure permits the solution generalization to all population-based methods. Consequently, this stochastic, multicriteria MGA algorithmic approach proves to be extremely computationally efficient. The efficacy of this method for creating waste management alternatives is demonstrated by extending the MGA procedure to the "real world" WM optimization case taken from [43,44].

Modelling to Generate Alternatives

Mathematical optimization has fixated almost entirely on determining single optimal solutions to single-objective problems or constructing sets of noninferior solutions for multi-objective formulations [2,6,15]. While these approaches may create solutions to the mathematical models, whether these outputs are the best solutions to the "real" problems remains can be debatable [1,2,7,15]. Within most "real world" decision-making environments, there are countless system requirements and objectives that will never be explicitly apparent or included in the model formulation [1,6,45]. Furthermore, most subjective aspects unavoidably remain unmodelled and unquantified in the decision models constructed [46]. This regularly occurs where final decisions are constructed based not only on modelled objectives, but also on more subjective stakeholder goals and socio-political-economic preferences [14]. Several examples of these incongruent modelling dichotomies are discussed in [7,15-17].

When unmodelled objectives and unquantified issues exist, non-traditional methods are required for searching the decision region not only for noninferior sets of solutions, but also for alternatives that are evidently sub-optimal to the modelled problem. Namely, any search for alternatives to problems known or suspected to contain unmodelled components must concentrate not only on a non-inferior set of solutions, but also necessarily on an explicit exploration of the problem's inferior solution space.

To demonstrate the implications of unmodelled objectives in a decision search, assume that an optimal solution for a maximization problem is $\mathbf{X}^*$ with objective value $Z1^*$ [47]. Suppose a second, unquantified, maximization objective Z2 exists that represents some "politically acceptable" factor. Assume that the solution, $\mathbf{X}^a$, belonging to the 2-objective noninferior set, exists that corresponds to the best compromise solution if both objectives could have been simultaneously considered. Although $\mathbf{X}^a$ would be considered as the best solution to the real problem, in the actual mathematical model it would appear inferior to solution $\mathbf{X}^*$, since $Z1^a \leq Z1^*$. Therefore, when unquantified components are included in the decision-making process, inferior decisions to the mathematically modelled problem could be optimal to the underlying "real" problem. By creating good-but-different solutions, the decision-makers are able to explore potentially desirable qualities within the alternatives that might be able to satisfy the unmodelled objectives to varying degrees of stakeholder acceptability.

To motivate the MGA process, it is necessary to more formally characterize the mathematical definition of its goals [7,14]. Assume that the optimal solution to an original mathematical model is $\mathbf{X}^*$ with corresponding objective value $\mathbf{Z}^* = F(\mathbf{X}^*)$. The resultant difference model can then be solved to produce an alternative solution, $\mathbf{X}$, that is maximally different from $\mathbf{X}^*$:

Maximize $$\Delta(\mathbf{X}, \mathbf{X}^*) = \underset{i}{\mathrm{Min}} \left| X_i - X_i^* \right| \quad (1)$$

Subject to: $$X \in D \quad (2)$$

$$\left| F(X) - Z^* \right| \leq T \quad (3)$$

Where Δ represents an appropriate difference function (shown in (1) as an absolute difference) and T is a tolerance target relative to the original optimal objective value $\mathbf{Z}^*$. T is a user-specified limit that determines what proportion of the inferior region needs to be explored for acceptable alternatives. This difference function concept can be extended into a difference measure between any set of alternatives by replacing $\mathbf{X}^*$ in the objective of the maximal difference model and calculating the overall minimum absolute difference (or some other function) of the pairwise comparisons between corresponding variables in each pair of alternatives-subject to the condition that each alternative is feasible and falls within the specified tolerance constraint.

The population-based MGA procedure to be introduced is designed to generate a pre-determined small number of close-to-optimal, but maximally different alternatives, by adjusting the value of T and solving the corresponding maximal difference problem instance by exploiting the population structure of the algorithm.

Simulation-Optimization for Stochastic Optimization

Finding optimal solutions to large stochastic problems prove complicated when numerous system uncertainties must be directly incorporated into the solution procedures [47-50]. Simulation-Optimization (SO) is a broadly defined family of stochastic solution approaches that combines simulation with an underlying optimization component for optimization [47]. In SO, all unknown objective functions, constraints, and parameters are replaced by simulation models in which the decision variables provide the settings under which simulation is performed.

The general steps of SO can be summarized in the following fashion [48,51]. Suppose the mathematical model of the optimization problem contains n decision variables X_i, represented in the vector $\boldsymbol{X} = [X_1, X_2, \ldots, X_n]$. If the objective function is expressed by F and the feasible region is designated by D, then the mathematical programming problem is to optimize $F(\boldsymbol{X})$ subject to $X \in D$. When stochastic conditions exist, values for the objective and constraints can be determined by simulation. Any solution comparison between two different solutions $\boldsymbol{X1}$ and $\boldsymbol{X2}$ require the evaluation of some statistic of F modelled with $\boldsymbol{X1}$ compared to the same statistic modelled with $\boldsymbol{X2}$ [47,52]. These statistics are calculated by simulation, in which each $\boldsymbol{X}$ provides the decision variable settings employed in the simulation. While simulation provides a means for comparing results, it does not provide the mechanism for determining optimal solutions to problems. Hence, simulation cannot be used independently for stochastic optimization.

Since all measures of system performance in SO are stochastic, every potential solution, $\boldsymbol{X}$, must be calculated through simulation. Because simulation is computationally intensive, an optimization algorithm is employed to guide the search for solutions through the problem's feasible domain in as few simulation runs as possible [49,52]. As stochastic system problems frequently contain numerous potential solutions, the quality of the final solution could be highly variable unless an extensive search has been performed throughout the entire feasible region. A stochastic SO approach contains two alternating computational phases; (i) an "evolutionary" module directed by some optimization (frequently a metaheuristic) method and (ii) a simulation module [53]. Because of the stochastic components, all performance measures are necessarily statistics calculated from the responses generated in the simulation module. The quality of each solution is found by having its performance criterion, F, evaluated in the simulation module. After simulating each candidate solution, their respective objective values are returned to the evolutionary module to be utilized in the creation of ensuing candidate solutions. Thus, the evolutionary module aims to advance the system toward improved solutions in subsequent generations and ensures that the solution search does not become trapped in some local optima. After generating new candidate solutions in the evolutionary module, the new solution set is returned to the simulation module for comparative evaluation. This alternating, two-phase search process terminates when an appropriately stable system state (i.e. an optimal solution) has been attained. The optimal solution produced by the procedure is the single best solution found throughout the course of the entire search process [53].

Population-based algorithms are conducive to SO 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. For population-based optimization methods, the evolutionary phase evaluates the entire current population of solutions during each generation of the search and evolves from a current population to a subsequent one. A primary characteristic of population-based procedures is that better solutions in a current population possess a greater likelihood for survival and progression into the subsequent population.

It has been shown that SO can be used as a very computationally intensive, stochastic MGA technique [52,54]. However, because of the very long computational runs, several approaches to accelerate the search times and solution quality of SO have been examined subsequently [51]. The next section provides an MGA algorithm that incorporates stochastic uncertainty using SO to much more efficiently generate sets of maximally different solution alternatives.

Multicriteria Population-based MGA Computational Algorithm

In this section, a data structure is employed that enables a multicriteria MGA solution approach via any population-based algorithm [55-57]. Suppose that it is desired to produce P alternatives that each possess n decision variables and that the population algorithm is to possess K solutions in total. That is, each solution contains one possible set of P maximally different alternatives. Let $\boldsymbol{Y_k}$, $k = 1,\ldots, K$, represent the k^{th} solution which consists of one complete set of P different alternatives. Specifically, if $\boldsymbol{X_{kp}}$ corresponds to the p^{th} alternative, $p = 1,\ldots, P$, of solution k, $k = 1,\ldots, K$, then $\boldsymbol{Y_k}$ can be represented as

$$Y_K = [X_{k1}, X_{k2}, \ldots, X_{kP}] \tag{4}$$

If X_{kjq}, $q = 1,\ldots, n$, is the q^{th} variable in the j^{th} alternative of solution k, then

$$X_{kj} = (X_{kj1}, X_{kj2}, \ldots, X_{kjn}) \tag{5}$$

Consequently, an entire population, Y, consisting of K different sets of P alternatives can be written in vectorized form as,

$$Y' = [Y_1, Y_2, \ldots, Y_K] \tag{6}$$

The following population-based MGA method produces a pre-determined number of close-to-optimal, but maximally different alternatives, by modifying the value of the bound T in the maximal difference model and using any population-based method to solve the corresponding, maximal difference problem. The multicriteria MGA algorithm that follows constructs a pre-determined number of maximally different, near-optimal alternatives, by modifying the bound value T in the maximal difference model and using any population-based technique to solve the corresponding maximal difference problem. Each solution in the population comprises one set of p different alternatives. By exploiting the co-evolutionary aspects of the algorithm, the algorithm evolves each solution toward sets of dissimilar local optima within the solution domain. In this processing, each solution alternative mutually experiences the search steps of the algorithm. Solution survival depends upon both how well the solutions perform with respect to the modelled objective(s) and by how far apart they are from every other alternative in the decision space.

A straightforward process for generating alternatives solves the maximum difference model iteratively by incrementally updating the target T whenever a new alternative needs to be produced and then re-solving the resulting model [55]. This iterative approach parallels the seminal Hop, Skip, and Jump (HSJ) MGA algorithm [15] in which the alternatives are created one-by-one through an incremental adjustment of the target constraint. While this approach is straightforward, it entails a repetitive execution of the optimization algorithm [14,31,32]. To improve upon the stepwise HSJ approach, a concurrent MGA technique was subsequently designed based upon co-evolution [32,34,36]. In a co-evolutionary approach, pre-specified stratified subpopulation ranges within an algorithm's overall population are established that collectively evolve the search toward the specified number of maximally different alternatives. Each desired solution alternative is represented by each respective subpopulation and each subpopulation undergoes the common processing operations of the procedure. The survival of solutions in each subpopulation depends simultaneously upon how well the solutions perform with respect to the modelled objective(s) and by how far away they are from all of the other alternatives. Consequently, the evolution of solutions in each subpopulation toward local optima is directly influenced by those solutions contained in all of the other subpopulations, which forces the concurrent co-evolution of each subpopulation towards good but maximally distant regions within the decision space according to the maximal difference model [14]. Co-evolution is also much more efficient than a sequential HSJ-style approach in that it exploits the inherent population-based searches to concurrently generate the entire set of maximally different solutions using only a single population [31,36].

While concurrent approaches can exploit population-based algorithms, co-evolution can only occur within each of the stratified subpopulations. Consequently, the maximal differences between solutions in different subpopulations can only be based upon aggregate subpopulation measures. Conversely, in the following simultaneous MGA algorithm, each solution in the population contains exactly one entire set of alternatives and the maximal difference is calculated only for that particular solution (i.e. the specific alternative set contained within that solution in the population). Hence, by the evolutionary nature of the population-based search procedure, in the subsequent approach, the maximal difference is simultaneously calculated for the specific set of alternatives considered within each specific solution – and the need for concurrent subpopulation aggregation measures is avoided.

Using the data structure terminology, the steps for the multicriteria MGA algorithm are as follows [33,38-42,55-57]. It should be readily apparent that the stratification approach employed by this method can be easily modified for any population-based algorithm.

Initialization Step: Solve the original optimization problem to find its optimal solution, $\boldsymbol{X}^*$. Based upon the objective value $F(\boldsymbol{X}^*)$, establish P target values. P represents the desired number of maximally different alternatives to be generated within prescribed target deviations from the X^*. Note: The value for P has to have been set a priori by the decision-maker.

Without loss of generality, it is possible to forego this step and to use the algorithm to find $\boldsymbol{X}^*$ as part of its solution processing in the subsequent steps. However, this significantly increases the number of iterations of the computational procedure and the initial stages of the processing become devoted to finding $\boldsymbol{X}^*$ while the other elements of each population solution are retained as essentially "computational overhead".

Step 1: Create an initial population of size K where each solution contains P equally-sized partitions. The partition size corresponds to the number of decision variables in the original optimization problem. $\boldsymbol{X}_{kp}$ represents the p^{th} alternative, $p = 1,\ldots,P$, in solution Y_k, $k = 1,\ldots,K$.

Step 2: In each of the K solutions, evaluate each X_{kp}, $p = 1,\ldots,P$, using the simulation module with respect to the modelled objective. Alternatives meeting their target constraint and all other problem constraints are designated as feasible, while all other alternatives are designated as infeasible. A solution can only be designated as feasible if all of the alternatives contained within it are feasible.

Step 3: Apply an appropriate elitism operator to each solution to rank order the best individuals in the population. The best solution is the feasible solution containing the most distant set of alternatives in the decision space (the distance measures are defined in Step 5).

Note: Because the best solution to date is always retained in the population throughout each iteration, at least one solution will always be feasible. A feasible solution for the first step can always consist of P repetitions of X^*.

Step 4: Stop the algorithm if the termination criteria (such as maximum number of iterations or some measure of solution convergence) are met. Otherwise, proceed to Step 5.

Step 5: For each solution Y_k, $k = 1,\ldots, K$, calculate R Max-Min and/or Max-Sum distance measures, D^r_k, $r = 1,\ldots, R$, between all of the alternatives contained within the solution.

As an illustrative example for calculating the multicriteria distance measures, compute

$$D_k^1 = \Delta^1\left(X_{ka}, X_{kb}\right) = \underset{a,b,q}{Min}\left|X_{kaq} - X_{kbq}\right|, \qquad a = 1,\ldots,P, b = 1\ldots P, b = 1\ldots n, \tag{7}$$

$$D_k^2 = \Delta^2\left(X_{ka}, X_{kb}\right) = \sum_{a=1toP} \sum_{b=1toP} \sum_{q=1\ldots n}\left|X_{kaq} - X_{kbq}\right|, \tag{8}$$

and

$$D_k^3 = \Delta^3\left(X_{ka}, X_{kb}\right) = \sum_{a=1toP} \sum_{b=1toP} \sum_{q=1\ldots n}\left(X_{kaq} - X_{kbq}\right)^2, \tag{9}$$

D^1_k denotes the minimum absolute distance, D^2_k represents the overall absolute deviation, and D^3_k represents the overall quadratic deviation between all of the alternatives contained within solution *k*.

Alternatively, the distance functions could be calculated by some other appropriately defined function.

Step 6: Let $Dk = G(D^1_k, D^2_k, D^3_k,\ldots, D^R_k)$ represent the multicriteria objective for solution *k*. Rank the solutions according to the distance measure D_k objective – appropriately adjusted to incorporate any constraint violation penalties for infeasible solutions. The goal of maximal difference is to force alternatives to be as far apart as possible in the decision space from the alternatives of each of the partitions within each solution This step orders the specific solutions by those solutions which contain the set of alternatives which are most distant from each other.

Step 7: Apply applicable algorithmic "change operations" to each solution within the population and return to Step 2.

Waste Management Case Study

As indicated throughout the previous sections, WM decision-makers faced with situations containing numerous uncertainties often prefer to select from a set of "near best" alternatives that differ significantly from each other in terms of the system structures characterized by their decision variables. The efficacy of the multicriteria, population-based MGA procedure will be illustrated using the WM stochastic optimization case of Hamilton-Wentworth taken from [43] and [44]. While this section briefly summarizes the optimization model, more explicit details, data, and descriptions can be found in [43].

Located at the Western-most edge of Lake Ontario, the Municipality of Hamilton-Wentworth covers an area of 1,100 square kilometers and includes six towns and cities; Hamilton, Dundas, Ancaster, Flamborough, Stoney Creek, and Glanbrook. The Municipality is considered the industrial centre of Canada, although it simultaneously incorporates diverse areas of not only heavy industrial production, but also densely populated urban space, regions of significant suburban development, and large proportions of rural/agricultural environments. Prior to the study of Yeomans, *et al.* [43], the municipality had not been able to effectively incorporate inherent uncertainties into their planning processes and, therefore, had not performed effective systematic planning for the flow of wastes within the region. The WM system within the region is a very complicated process which is affected by economic, technical, environmental, legislational and political factors.

The WM system within Hamilton-Wentworth needs to satisfy the waste disposal requirements of its half-million residents who, collectively, produce more than 300,000 tons of waste per year, with a budget of $22 million. The region had constructed a system to manage these wastes composed of: a waste-to-energy incinerator referred to as the Solid Waste Reduction Unit (or SWARU); a 550 acre landfill site at Glanbrook; three waste transfer stations located in Dundas (DTS), in East Hamilton at Kenora (KTS), and on Hamilton Mountain (MTS); a household recycling program contracted to and operated by the Third Sector Employment Enterprises; a household/hazardous waste depot, and; a backyard composting program.

The three transfer stations have been strategically located to receive wastes from the disparate municipal (and individual) sources and to subsequently transfer them to the waste management facilities for final disposal; either to SWARU for incineration or to Glanbrook for landfilling. Wastes received at the transfer stations are compacted into large trucks prior to being hauled to the landfill site. These transfer stations provide many advantages in waste transportation and management; these include reducing traffic going to and from the landfill, providing an effective control mechanism for dumping at the landfill, offering an inspection area where wastes can be viewed and unacceptable materials removed, and contributing to a reduction of waste volume because of the compaction process. The SWARU incinerator burns up to 450 tons of waste per day and, by doing so, generates about 14 million kilowatt hours of electricity per year which can be either used within the plant itself or sold to the provincial electrical utility. SWARU also produces a residual waste ash which must subsequently be transported to the landfill for disposal.

Within this WM system, decisions have to be made regarding whether waste materials should be recycled, landfilled or incinerated and additional determinations have to be made as to which specific facilities would process the discarded materials. Included within these decisions is 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 determine which facilities process 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 have to be made to determine how much waste should be sent along each flow pathway to each facility. Therefore, any single WM policy option is composed of a combination of many decisions regarding which facilities received waste material and what quantities of waste are sent to each facility. All of these decisions are compounded by overriding stochastic system uncertainties. The complete mathematical model used for WM planning appears in [43]. This mathematical formulation was used not only to examine the existing municipal WM system, but also to prepare the municipality for several potentially enforced future changes to its operating conditions.

Yeomans, *et al.* [43] examined three likely future scenarios, with each scenario involving potential incinerator operations. Scenario 1 considered the existing WM 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 within its design capacity range; from being closed completely to operating up to its maximum capacity. Yeomans, *et al.* [43] ran SO for a 24-hour period to determine optimal solutions for each scenario. For the existing system (Scenario 1), a solution that would never cost more than $20.6 million was constructed. For Scenarios 2 and 3, Yeomans, *et al.* [43] 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 (Table 1).

	Scenario 1	Scenario 2	Scenario 3
Specific WM Settings for the Incinerator Operations	Status Quo system. Incinerator must operate at or above a minimum floor capacity level.	Incinerator operating at its maximum capacity	Incinerator can operate at any level within its design capacity range. [0,Max]
Optimal Scenario Cost found by [43]	$20.6 million	$22.1 million	$18.7 million

Table 1: Specific WM Settings for the 3 Scenarios and Optimal WM Costs ($ Millions) for Each Scenario Determined in Yeomans, *et al.* [43]

As noted, WM planners faced with difficult and controversial selections generally prefer to choose from a set of near-optimal alternatives that differ significantly from each other in terms of the system structures characterized by their decision variables. In order to create these alternative planning options for the three WM system scenarios, it would be possible to place extra target constraints into the original optimization model which would force the generation of solutions that were different from their respective, initial optimal solutions. Suppose for example that four additional planning alternative options were created through the inclusion of a technical constraint on the objective function that increased the total system cost of the original model from 2.5% up to 10% in increments of 2.5%. By adding these incremental target constraints to the original SO model and sequentially resolving the problem 4 more times for each scenario (i.e. another 12 additional computational runs of the SO procedure), it would be possible to create a specific number of alternative policies for WM planning.

However, to improve upon the need to run fifteen separate instances of the computationally intensive SO algorithm to generate these solutions, the multicriteria population-based MGA procedure described in the previous section was run only once to produce the 15 alternatives shown in Table 2. Each column of the Table shows the overall system costs for the 5 maximally different options generated for each of the three scenarios.

Annual WM Costs	Scenario 1	Scenario 2	Scenario 3
Best Solution Found	20.61	22.10	18.71
Alternative Within 2.5%	21.01	22.72	18.98
Alternative Within 5.0%	21.48	23.08	19.53
Alternative Within 7.5%	22.10	23.91	19.99
Alternative Within 10%	22.45	24.06	20.31

Table 2: Annual WM Costs ($ Millions) for 5 Maximally Different Alternatives for Each Scenario Produced by the Multicriteria MGA algorithm

Given the performance bounds established for the objective in each problem instance, the decision-makers can feel reassured by the stated performance for each of these options while also being aware that the perspectives provided by the set of dissimilar decision variable structures are as different from each other as is feasibly possible. Hence, if there are stakeholders with incompatible standpoints holding diametrically opposing viewpoints, the policy-makers can perform an assessment of these different options without being myopically constrained by a single overriding perspective based solely upon the objective value.

Furthermore, it can be shown that the multicriteria MGA procedure does indeed produce very good solution values for the originally modelled optimization problem, itself. Table 3 clearly highlights how the "Best Solution Found" by the MGA procedure for each scenario is actually identical to the one found by function optimization alone in [43].

	Scenario 1	Scenario 2	Scenario 3
Yeomans, *et al.* [43] using SO	20.6	22.1	18.7
Best Solution Found using MGA	20.6	22.1	18.7

Table 3: Comparison of Best Annual WM Costs ($ millions) Determined by MGA and in [43]

The computational example highlights several important aspects with respect to the MGA approach: (i) Population-based algorithms can be effectively employed as the underlying solution search procedure for SO routines; (ii) Population-based solution searches can simultaneously generate more good alternatives than planners would be able to create using other MGA approaches; (iii) By the design of the MGA algorithm, the alternatives generated are good for planning purposes since all of their structures are guaranteed to be as mutually and maximally different from each other as possible; (iv) The approach is very computationally efficient since it need be run only once to generate its entire set of multiple, good solution alternatives (i.e. to generate *n* maximally different solution alternatives, the MGA algorithm would run exactly the same number of times that the SO would need to be run for function optimization purposes alone - namely once - irrespective of the value of *n*); and, (v) The best overall solutions produced by the MGA procedure will be identical to the best overall solutions that would be produced for function optimization purposes alone.

Conclusions

Waste management problems contain multidimensional performance specifications which inevitably include incongruent performance objectives and unquantifiable modelling features. These problems also often possess incompatible design specifications which are impossible to completely formulate into the supporting decision models. Consequently, there are unmodelled problem components, generally not apparent during model construction that can significantly influence the acceptability of any model's solutions. These competing and ambiguous components force WM decision-makers to incorporate many conflicting requirements into their decision process prior to the final solution determination. Consequently, waste management decision-makers generally prefer to select from a set of distinct planning perspectives.

This paper has employed a computationally efficient multicriteria population-based MGA procedure for WM planning. This MGA approach establishes how population-based algorithms can simultaneously construct entire sets of near-optimal, maximally different alternatives by exploiting the evolving solution characteristics in population-based solution algorithms. In an MGA role, a multicriteria objective can efficiently generate the requisite set of dissimilar alternatives, with each generated solution suggesting an entirely different perspective to the problem. Since population-based algorithms can be extended to an eclectic variety of problem settings, the practicality of this multicriteria MGA method can be used on a diverse spectrum of "real world" applications. These extensions will be considered in future studies.

Acknowledgment

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

References

1. Brugnach M, Tagg A, Keil F, De Lange Lange W (2007) Uncertainty matters: computer models at the science-policy interface. Water Resour Manage 21: 1075-90.

2. Janssen J, Krol M, Schielen R, Hoekstra A (2010) The effect of modelling quantified expert knowledge and uncertainty information on model-based decision making. Environ Sci Policy 13: 229-38.

3. Tchobanoglous G, Thiesen H, Vigil S (1993) Integrated solid waste management: engineering principles and management issues. New York: McGraw-Hill 978.

4. Matthies M, Giupponi C, Ostendorf B (2007) Environmental decision support systems: Current issues, methods and tools. Environ Model Software 22: 123-7.

5. Mowrer HT (2000) Uncertainty in natural resource decision support systems: Sources, Uncertainty in natural resource decision support systems: Sources, interpretation, and importance. Comput Electron Agric 27: 139-54.

6. Walker W, Harremoes P, Rotmans J, Van der Sluis J, Van Asselt M, et al. (2003) Defining uncertainty - a conceptual basis for uncertainty management in model-based decision support. Integr Assess 4: 5-17.

7. Loughlin D, Ranjithan S, Brill E, Baugh J (2001) Genetic algorithm approaches for addressing unmodelled objectives in optimization problems. Eng Optim 33: 549-69.

8. Lund J, Tchobanoglous G, Anex R, Lawver R (1994) Linear programming for analysis of material recovery facilities. ASCE J Environ Eng 120: 1082-94.

9. Castelletti A, Galelli S, Restelli M, Soncini-Sessa R (2012) Data-driven dynamic emulation modelling for the optimal management of environmental systems. Environ Model Software 34: 30-43.

10. De Kok J, Wind H (2003) Design and application of decision support systems for integrated water management; lessons to be learnt. Phys Chem Earth 28: 571-8.

11. Hipel K, Walker S (2011) Conflict analysis in environmental management. Environmetrics 22: 279-93.

12. Lund J (2012) Provoking more productive discussion of wicked problems. J Water Resour Plann Manage 138: 193-5.

13. Walker S, Hipel K, Inohara T (2012) Attitudes and preferences: approaches to representing decision maker desires. Appl Math Comput 218: 6637-47.

14. Yeomans J, Gunalay Y (2011) Simulation-optimization techniques for modelling to generate alternatives in waste management planning. J Appl Oper Res 3: 23-35.

15. Brill ED, Chang S, Hopkins LD (1982) Modelling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning. Manage Sci 28: 221-35.

16. Baugh J, Caldwell S, Brill E (1997) A mathematical programming approach for generating alternatives in discrete structural optimization. Eng Optim 28: 1-31.

17. Zechman E, Ranjithan R (2007) Generating alternatives using evolutionary algorithms for water resources and environmental management problems. J Water Resour Plann Manage 133: 156-65.

18. Kasprzyk J, Reed P, Characklis G (2012) Many-objective de novo water supply portfolio planning under deep uncertainty. Environ Model Software 34: 87-104.

19. Gunalay Y, Yeomans J, Huang G (2012) Modelling to generate alternative policies in highly uncertain environments: An application to municipal solid waste management planning. J Environ Inf 19:58-69.

20. Fuerst C, Volk M, Makeschin F (2010) Squaring the circle? Combining models, indicators, experts and end-users in integrated land-use management support tools. Environ Manage 46: 829-33.

21. McIntosh B, Ascough J, Twery M (2011) Environmental decision support systems (EDSS) development - challenges and best practices. Environ Model Software 26: 1389-402.

22. Caicedo JM, Yun GJ (2011) A novel evolutionary algorithm for identifying multiple alternative solutions in model updating. Struct Health Monit- Int J 10: 491-501.

23. DeCarolis J (2011) Using modeling to generate alternatives (MGA) to expand our thinking on energy futures. Energy Econ 33: 145-52.

24. Rubenstein-Montano B, Zandi I (1999) Application of a genetic algorithm to policy planning: The case of solid waste. Environ Plann B: Plann Design 26: 791-907.

25. Trutnevyte E, Stauffacher M, Schlegel M (2012) Context-specific energy strategies: coupling energy system visions with feasible implementation scenarios. Environ Sci Technol 46: 9240-8.

26. Ursem R, Justesen P (2012) Multi-objective distinct candidates optimization: locating a few highly different solutions in a circuit component sizing problem. Appl Soft Comput 12: 255-65.

27. Zarate B, Caicedo J (2008) Finite element model updating: multiple alternatives. Eng Struct 30: 3724-30.

28. He L, Huang G, Zeng G-M (2009) Identifying optimal regional solid waste management strategies through an inexact integer programming model containing infinite objectives and constraints. Waste Manag 29: 21-31.

29. Sun W, Huang G (2010) Inexact piecewise quadratic programming for waste flow allocation under uncertainty and nonlinearity. J Environ Inf 16: 80-93.

30. Thekdi S, Lambert J, Source J (2012) Decision analysis and risk models for land development affecting infrastructure systems. Risk Anal 32: 1253-69.

31. Imanirad R, Yeomans JS (2013) Modelling to generate alternatives using biologically inspired algorithms. In "Yang X, Swarm Intell Bio-Inspired Comput: Theory Appl". Amsterdam: Elsevier 313-33.

32. Imanirad R, Yang X, Yeomans J (2012) A computationally efficient, biologically-inspired modelling-to-generate-alternatives method. J Comput 2: 43-7.

33. Yeomans JS (2018) An efficient computational procedure for simultaneously generating alternatives to an optimal solution using the firefly algorithm. In "Yang X, Nat-inspired algorithms appl optim". New York: Springer 261-73.

34. Imanirad R, Yang X, Yeomans J (2012) A co-evolutionary, nature-inspired algorithm for the concurrent generation of alternatives. J Comput 2: 101-6.

35. Imanirad R, Yang X, Yeomans J (2013) Modelling-to-generate-alternatives via the firefly algorithm. J Appl Oper Res 5: 14-21.

36. Imanirad R, Yang X, Yeomans JS (2013) A concurrent modelling to generate alternatives approach using the firefly algorithm. Int J Decis Support Syst Technol 5: 33-45.

37. Imanirad R, Yang X, Yeomans J (2013) A biologically-inspired metaheuristic procedure for modelling-to-generate-alternatives. Int J Eng Res Appl 3: 1677-86.

38. Yeomans J (2017) Simultaneous computing of sets of maximally different alternatives to optimal solutions. Int J Eng Res Appl 7: 21-8.

39. Yeomans JS (2017) An optimization algorithm that simultaneously calculates maximally different alternatives. Int J Comput Eng Res 7: 45-50.

40. Yeomans J (2018) Computationally testing the efficacy of a modelling-to-generate-alternatives procedure for simultaneously creating solutions. J Comput Eng 20: 38-45.

41. Yeomans JS (2017) A computational algorithm for creating alternatives to optimal solutions. Trans Mach Learn Artif Intell 5: 58-68.

42. Yeomans JS (2019) A simultaneous modelling-to-generate-alternatives procedure employing the firefly algorithm. In "Dey N, Technological Innovations in Knowledge Management and Decision Support". Hershey, Pennsylvania: IGI Global 19-33.

43. Yeomans J, Huang G, 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 Environ Inf 2: 11-30.

44. Yeomans JS (2004) Improved Policies for Solid Waste Management in the Municipality of Hamilton-Wentworth, Ontario. Can J Administrative Sci 21: 376-93.

45. Wang L, Fang L, Hipel K (2007) On achieving fairness in the allocation of scarce resources: measurable principles and multiple objective optimization approaches. IEEE Syst J 1: 17-28.

46. Martinez LJ, Joshi NN, Lambert JH (2011) Diagramming qualitative goals for multiobjective project selection in large-scale systems. Syst Eng 14: 73-86.

47. Fu MC (2002) Optimization for simulation: theory vs. practice. INFORMS Journal on Computing 14: 192-215.

48. Kelly P (2002) Simulation optimization is evolving. INFORMS J Comput 14: 223-5.

49. Zou R, Liu Y, Riverson J, Parker A, Carter S (2010) A nonlinearity interval mapping scheme for efficient waste allocation simulation-optimization analysis. Water Resour Res 46: 1-14.

50. Imanirad R, Yang X, Yeomans J (2016) Stochastic decision-making in waste management using a firefly algorithm-driven simulation-optimization approach for generating alternatives. In "Koziel S, Leifsson L, Yang X, Recent Adv Simul-Driven Model Optim". Heidelberg: Springer 299-323.

51. Yeomans JS (2012) Waste management facility expansion planning using simulation-optimization with grey programming and penalty functions. Int J Environ Waste Manage 10: 269-83.

52. Yeomans JS (2008) Applications of simulation-optimization methods in environmental policy planning under uncertainty. J Environ Inf 12: 174-86.

53. Yeomans J, Yang X (2014) International Journal of Process Management and Benchmarking 4: 363-75.

54. Linton JD, Yeomans JS, Yoogalingam R (2002) Policy planning using genetic algorithms combined with simulation: The case of municipal solid waste. Environ Plann B: Plann Des 29: 757-78.

55. Yeomans JS (2018) An algorithm for generating sets of maximally different alternatives using population-based metaheuristic procedures. Trans Mach Learn Artif Intell 6: 1-9.

56. Yeomans JS (2019) A bicriterion approach for generating alternatives using population-based algorithms. WSEAS Trans Syst 18: 29-34.

57. Yeomans JS, Yang XS (2019) A simulation-optimization algorithm for generating sets of alternatives using population-based metaheuristic procedures. J Software Eng Simul Forthcoming.