Journal of Civil Engineering Research 2019, 9(2): 43-50
DOI: 10.5923/j.jce.20190902.01

A Stochastic Multicriteria Algorithm for Generating Waste Management Facility Expansion Alternatives

Julian Scott Yeomans

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

Abstract While solving waste management (WM) planning problems, it may often be preferable to generate several quantifiably good options that provide multiple, contrasting perspectives. This is because WM planning generally contains complex problems that are riddled with inconsistent performance objectives and contain design requirements that are very difficult to quantify and capture when supporting decision models must be constructed. The generated alternatives should satisfy all of the stated system conditions, but be maximally different from each other in the requisite decision space. The process for creating maximally different sets of solutions is referred to as modelling-to-generate-alternatives (MGA). Simulation-optimization approaches have frequently been used to solve computationally difficult, stochastic WM problems. This paper outlines a stochastic 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 method is demonstrated on a “real world” waste management facility expansion case.

Keywords Modelling-to-generate-alternatives, Simulation-optimization, Waste management planning, Population-based algorithms

1. Introduction

Implementing effective management of waste management (WM) systems has often proven to be contentious and conflict-laden ([1-3]). 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 ([3-6]). Stochastic decision-making typically contains complex design aspects that can be problematic to integrate into mathematical formulations and can be inundated by unquantifiable specifications ([4-6]). WM decision-making frequently possess inconsistent and incompatible design specifications that can be difficult to formulate into supporting mathematical decision-models ([6]). 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 ([7-10]). Although “optimal” solutions to the modelled constructions can be determined, these do not usually deliver the best solution to the “real” problem as there are generally unmodeled components not apparent when the mathematical models are created ([1,2,9]). Moreover, it may not be possible to explicitly convey many of the subjective components 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 ([11]). Consequently, waste management determination proves to be an extremely challenging and complicated endeavour ([11,12]).

Within WM decision-making, there are routinely many stakeholder groups possessing incongruent standpoints, essentially dictating that the waste managers need to build decision frameworks that simultaneously reflect numerous irreconcilable points of view ([4]). As a result, it is more desirable to construct a small number of dissimilar alternatives that permit opposing perspectives to the stated problem ([4,10]). These options should be close-to-optimal with respect to all identified objective(s), but be maximally different from each other in the decision region. The formal

* Corresponding author:
syeomans@schulich.yorku.ca (Julian Scott Yeomans)

Published online at http://journal.sapub.org/jce

process for creating such maximally different solution sets is usually referred to as *modelling-to-generate-alternatives* (MGA) ([8-10]). MGA techniques prescribe a systematic examination of the solution space in order to create a set of alternatives that are considered good when measured by the objective space but as different as possible from each other in the modelled decision space. The resultant solution set should provide alternative viewpoints that perform similarly with respect to the modelled objectives, yet very differently with respect to any potentially unmodelled features ([6]). Subsequently, the decision-makers must perform a comparison of the alternatives to determine which option(s) most closely achieve(s) their specific requirements. In comparison to the more straightforward solution determination approaches inherent in most "traditional" optimization methods, MGA approaches fall necessarily into the decision support category.

Early MGA algorithms employed direct, incremental approaches for constructing their alternatives by iteratively re-running their procedures whenever new solutions needed to be generated ([7-11]). These iterative approaches replicated the seminal MGA work of [8] where, once the initial mathematical formulation has been optimized, all supplementary alternatives are produced one-at-a-time. These approaches all required n+1 iterations of their algorithms – to optimize the original problem in the first step, followed by the subsequent sequential construction of each of the n alternatives ([10,12-14]).

In this paper, it is shown how the set of maximally different options can be created by extending several earlier deterministic MGA approaches to stochastic optimization ([13-19]). The stochastic algorithm provides an MGA process that can be performed by *any* population-based solution mechanism. This algorithm advances earlier procedures ([14-18]) to permit the generation of n distinct alternatives *simultaneously* in a single computational run. Namely, in order 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 ([20-24]). Furthermore, a multicriteria objective is employed that combines a novel data structure into the simultaneous solution approach to create an effective MGA approach. This data structure facilitates the above-mentioned solution generalization to population-based methods. Consequently, this stochastic multicriteria MGA algorithmic approach proves to be very computationally efficient ([25]). The procedure is demonstrated on a WM facilities expansion case that had previously been considered in ([26-28]).

2. 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 ([2,6,8]). While these approaches may provide solutions to the formal mathematical models, whether these outputs are truly the best solutions to the "real" problems remains can be debatable ([1,2,8,9]). 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 stage ([1,6]). Furthermore, most subjective aspects remain unavoidably unmodelled and unquantified in the constructed decision models. 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 ([10]). Several incongruent modelling dualities are discussed in ([7-9,11]).

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 consequences of an unmodelled objective in a decision search, assume that the quantifiably optimal solution for a single-objective, maximization problem is $\boldsymbol{X}^*$ with a corresponding objective value $Z1^*$. Now suppose that a second, unmodelled, maximization objective $Z2$ exists that subjectively incorporates some unquantifiable "politically acceptable" component. Now assume that some solution, $\boldsymbol{X}^{\boldsymbol{a}}$, belonging to the 2-objective noninferior set, exists that represents a potentially best compromise solution for the decision-maker if both objectives had somehow been simultaneously evaluated. While $\boldsymbol{X}^{\boldsymbol{a}}$ could reasonably be considered as the best compromise solution for the real problem, in the quantified mathematical model it would appear inferior to solution $\boldsymbol{X}^*$, since it must be the case that $Z1^{a} \leq Z1^*$. Therefore, when unmodelled components are incorporated into a decision-making process, mathematically inferior options to the modelled problem could actually be optimal for the real underlying problem. By creating these good-but-different solutions, the decision-makers can then examine potentially desirable qualities within the options that may be able to address potentially unmodelled objectives to varying degrees of stakeholder tolerability.

To motivate the MGA process, it is necessary to more formally characterize the mathematical definition of its goals ([9], [10]). Assume that the optimal solution to an original mathematical model is $\boldsymbol{X^*}$ with corresponding objective value $\boldsymbol{Z^*} = F(\boldsymbol{X^*})$. The resultant difference model can then be solved to produce an alternative solution, $\boldsymbol{X}$, that is maximally different from $\boldsymbol{X}^*$:

$$\text{Maximize } \Delta(\boldsymbol{X}, \boldsymbol{X^*}) = \text{Min}_i \ |X_i - X_i^*| \qquad (1)$$

$$\text{Subject to: } \quad \boldsymbol{X} \in D \qquad (2)$$

$$| F(\boldsymbol{X}) - \boldsymbol{Z^*} | \leq T \tag{3}$$

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 $\boldsymbol{Z^*}$. 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 $\boldsymbol{X^*}$ 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 metaheuristic. The survival of solutions depends upon how well the solutions perform with respect to the problem's originally modelled objective(s) and simultaneously by how far away they are from all of the other alternatives generated in the decision space.

3. Simulation-Optimization for Stochastic Optimization

Finding optimal solutions to large stochastic problems proves complicated when numerous system uncertainties must be directly incorporated into the solution procedures ([29-32]). *Simulation-Optimization* (SO) refers to a family of stochastic solution approaches that combines simulation with an underlying optimization component for optimization ([29]). 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 ([31,33]). 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 $\boldsymbol{X} \in D$. When stochastic conditions exist, values for the objective and constraints can be determined by simulation. Any solution comparison between two different solutions ***X1*** and ***X2*** requires the evaluation of some statistic of F modelled with ***X1*** compared to the same statistic modelled with ***X2*** ([29,34]). 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 ([32,34]). 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 ([35]). 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 ([35]).

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 ([34,36]). However, because of the very long computational runs, several approaches to accelerate the search times and solution quality of SO have been examined subsequently ([33]). The next section provides an MGA algorithm that incorporates stochastic uncertainty using SO to much more efficiently generate sets of maximally different solution alternatives.

4. Multicriteria Population-based Simultaneous MGA Computational Algorithm

In this section, a novel data structure is employed that enables a multicriteria MGA solution approach via any population-based algorithm ([25,37,38]). Suppose that it is desired to be able 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 is to contain one possible set of *P* maximally different alternatives. In this representation, let $\boldsymbol{Y}_k$, $k = 1,\ldots, K$, represent the k^{th} solution which is made up of one complete set of *P* different alternatives. Namely, if $\boldsymbol{X}_{kp}$ is the p^{th} alternative, $p = 1,\ldots, P$, of solution *k*, $k = 1,\ldots, K$, then $\boldsymbol{Y}_k$ can be represented as

$$\boldsymbol{Y}_k = [\boldsymbol{X}_{k1}, \boldsymbol{X}_{k2},\ldots, \boldsymbol{X}_{kP}] \ . \qquad (4)$$

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

$$\boldsymbol{X}_{kj} = (X_{kj1}, X_{kj2},\ldots, X_{kjn}) \ . \qquad (5)$$

Consequently, an entire population, $\boldsymbol{Y}$, consisting of *K* different sets of *P* alternatives can be written in vectorized form as,

$$\boldsymbol{Y'} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2,\ldots, \boldsymbol{Y}_K] \ . \qquad (6)$$

The following multicriteria 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 metaheuristic to solve the corresponding, maximal difference problem. Each solution within the population contains one potential set of *p* different alternatives. By exploiting the co-evolutionary solution structure within the metaheuristic, the algorithm collectively evolves each solution toward sets of different local optima within the solution space. In this process, each desired solution alternative undergoes the common search procedure of the metaheuristic. However, the survival of solutions depends both 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 generated in the decision space.

A straightforward process for generating alternatives would be to iteratively solve the maximum difference model by incrementally updating the target *T* whenever a new alternative needs to be produced and then re-running the algorithm. This iterative approach would parallel the original Hop, Skip, and Jump (HSJ) MGA algorithm of [8] in which, once an initial problem formulation has been optimized, supplementary alternatives are systematically created one-by-one through an incremental adjustment of the target constraint to force the sequential generation of the suboptimal solutions. While this approach is straightforward, it requires a repeated execution of the optimization algorithm ([10,13,14]).

To improve upon the stepwise alternative approach of the HSJ algorithm, a concurrent MGA technique was subsequently designed based upon the concept of co-evolution ([14,15,17]). In a co-evolutionary approach, pre-specified stratified subpopulation ranges within an algorithm's overall population were established that collectively evolved the search toward the creation of 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 ([10]). 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 ([16,17]).

While a concurrent approach can exploit population-based solution approaches, the co-evolution process 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 circumvented.

Using the terminology introduced above, the steps in the stochastic multicriteria MGA procedure are as follows ([19-25,28,38]). It should be apparent that the stratification approach outlined in this algorithm can be easily modified to accommodate any population-based solution procedure.

Preliminary Step. In this initialization step, solve the original optimization problem to determine the optimal solution, $\boldsymbol{X^*}$. As with prior solution approaches ([14-18]) and without loss of generality, it is entirely possible to forego this step and construct the algorithm to find $\boldsymbol{X^*}$ as part of its solution processing. However, such a requirement increases the number of computational iterations of the overall procedure and the initial stages of the processing focus upon finding $\boldsymbol{X^*}$ while the other elements of each population solution remain essentially "computational overhead". 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.

Step 1. Create the initial population of size *K* in which each solution is divided into *P* equally-sized partitions. The size of each partition corresponds to the number of variables for the original optimization problem. $\boldsymbol{X}_{kp}$ represents the p^{th} alternative, $p = 1,\ldots,P$, in solution $\boldsymbol{Y}_k$, $k = 1,\ldots,K$.

Step 2. In each of the *K* solutions, evaluate each $\boldsymbol{X}_{kp}$, $p = 1,\ldots,P$, 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 consists 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 $\boldsymbol{Y}_k$, $k = 1,\ldots, K$, 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

$$\mathrm{D}^1_{\mathrm{k}} = \Delta_1(\mathbf{X}_{\mathrm{ka}}, \mathbf{X}_{\mathrm{kb}}) = \underset{a,b,q}{\mathrm{Min}} \;|\, \mathrm{X}_{\mathrm{kaq}} - \mathrm{X}_{\mathrm{kbq}} \,|\,,$$

$$\mathrm{a} = 1,\ldots,\mathrm{P},\ \mathrm{b} = 1,\ldots,\mathrm{P},\ \mathrm{q} = 1,\ldots,\mathrm{n}, \qquad (7)$$

$$\mathrm{D}^2_{\mathrm{k}} = \Delta_2(\mathbf{X}_{\mathrm{ka}}, \mathbf{X}_{\mathrm{kb}}) = \sum\nolimits_{a=1toP} \sum\nolimits_{b=1toP} \sum\nolimits_{q=1\ldots n} |\, \mathrm{X}_{\mathrm{kaq}} - \mathrm{X}_{\mathrm{kbq}} \,|, \ (8)$$

and

$$\mathrm{D}^3_{\mathrm{k}} = \Delta_3(\mathbf{X}_{\mathrm{ka}}, \mathbf{X}_{\mathrm{kb}}) = \sum\nolimits_{a=1toP} \sum\nolimits_{b=1toP} \sum\nolimits_{q=1\ldots n} (\mathrm{X}_{\mathrm{kaq}} - \mathrm{X}_{\mathrm{kbq}})^2, (9)$$

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 measures could be calculated by some other appropriately defined function.

Step 6. Let $D_k = G(D^1_k, D^2_k, D^3_k,\ldots, D^R_k)$ 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 appropriate metaheuristic "change operations" to each of the solutions within the population and return to Step 2.

5. Waste Facility Expansion Case Study

As mentioned earlier, "real world" decision-makers often prefer to choose from a set of "close-to-optimal" options that differ significantly from each other in terms of the structures represented in their decision variables. The capacity of the stochastic multicriteria MGA procedure to produce a set of maximally different alternatives concurrently will be demonstrated using the MSW expansion planning case previously considered in ([26,27,28]).

The region in this facility expansion problem contains three separate municipalities whose MSW disposal needs are collectively met by a landfill and two waste-to-energy (WTE) incinerators. The planning horizon consists of three separate time periods with each of the periods covering an interval of five years. The landfill capacity can only be expanded once throughout the 15-year planning horizon. Each WTE facility can be expanded by any one of four possible options in each of the three time periods. The expansion costs escalate over time to reflect anticipated future conditions and have been discounted to present values for use in the objective function. The MSW waste generation rates and the costs for waste transportation and treatment vary both spatially and temporally. The case requires the construction of the preferred facility expansion alternatives during the different time periods and the effective allocation of the relevant waste flows in order to minimize the total system costs over the planning horizon.

A single best solution to the expansion problem costing \$600.2 million was determined in Yeomans (2012a). However, as discussed, planners generally prefer to be able to select from a set of close-to-optimal alternatives that differ significantly from each other in terms of the system structures characterized by their decision variables. In order to create three alternative planning options, it would be possible to place extra target constraints into the maximal difference model which would force the generation of solutions that were different from this newly determined, optimal solution by target values of, for example, 2.5%, 5%, and 7.5%, respectively. By adding these specific target constraints to the original model, the problem would need to be resolved an additional three times. However, to improve upon the process of running four separate instances of the SO algorithm to determine these solutions, the stochastic population-based multicriteria MGA procedure described in the previous section was run once to produce the objectives for the 4 alternatives shown in Table 1. To illustrate the solution variety created by the MGA procedure, Table 2 displays the system expansion costs of the sets of 4 alternatives for each of the top 15 solutions remaining in the final population.

Table 1. System Expansion Costs ($ Millions) for the 4 Alternatives

	Overall "Optimal" Solution	Best 2.5% Solution	Best 5% Solution	Best 7.5% Solution
System Expansion Costs	600.20	602.92	609.83	613.71

Table 2. System Expansion Costs ($ Millions) for the 15 Best Solutions in the Algorithm's Terminal Population (sorted in non-decreasing order)

Solution in Population	"Optimal" Solution	Solution Within 2.5%	Solution Within 5%	Solution Within 7.5%
1	600.20	602.920	609.834	613.711
2	600.20	602.652	604.659	615.932
3	600.20	603.111	604.910	617.314
4	600.20	603.260	605.156	617.362
5	600.20	603.370	605.161	618.395
6	600.20	604.710	605.445	621.147
7	600.20	604.911	606.054	622.961
8	600.20	605.016	607.000	623.202
9	600.20	606.290	610.976	624.007
10	600.20	606.819	612.814	625.076
11	600.20	610.229	613.611	627.299
12	600.20	610.491	614.759	627.888
13	600.20	611.189	615.285	630.003
14	600.20	611.505	621.664	630.027
15	600.20	615.785	621.832	631.71

This example has demonstrated how the stochastic population-based multicriteria MGA approach could be used to efficiently generate a good set of policy alternatives that satisfy required system performance criteria according to prespecified bounds within stochastic environments and yet remain as maximally different from each other as possible in the decision space. Given the performance bounds established for the objective in each problem instance, decision-makers would be reassured by the stated performance bounds for each of these options while also being aware that the perspectives provided by the set of dissimilar decision variable structures are as maximally different from each other as is feasibly possible. Hence, if there are stakeholders with incompatible standpoints holding diametrically opposing viewpoints, the policy-makers could conduct an assessment of these different options without being myopically constrained by a single overriding perspective based solely upon the objective value. In addition to its alternative generating capabilities, the FA component within the MGA algorithm simultaneously performs extremely well with respect to its role in function optimization. It should be explicitly noted that the overall best solution produced by the MGA algorithm for the case is indistinguishable from the optimal solution determined in Yeomans (2012a).

In the computational testing, the results highlight several important characteristics with respect to the stochastic population-based MGA method: (i) *Any* population-based procedure can be used to direct the fundamental search process in SO; (ii) Due to the nature of the MGA algorithm, the alternatives created are good for planning purposes since all of their structures will be as mutually and maximally different from each other as possible; (iii) The co-evolutionary capabilities within population-based methods can generate more good alternatives than planners would be able to create using other MGA approaches because of the evolving nature of its population-based solution searches; (iv) The approach is very computationally efficient. To generate *n* solution alternatives, the population-based algorithms needs to run exactly once, irrespective of the value of *n*; and, (v) The best overall solutions produced by the stochastic MGA procedure will be identical to the best overall solutions that would be produced by the population-based metaheuristic for function optimization purposes alone.

6. Conclusions

Waste management problems contain multifaceted performance specifications which inevitably possess incongruent performance objectives and unquantifiable modelling features. These decision environments frequently contain incompatible design provisions that are problematic – if not impossible – to incorporate when ancillary decision support models are constructed. Invariably, there are unmodelled elements, not apparent during model formulation, that can significantly affect the adequacy of its 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 approach for waste facility expansion planning using any population-based solution algorithm. This computationally efficient approach establishes how population-based algorithms can simultaneously construct entire sets of close-to-optimal, maximally different alternatives by exploiting the evolutionary characteristics of any population-based solution method. This MGA algorithm establishes how population-based methods can simultaneously construct entire sets of near-optimal, maximally different alternatives by exploiting the evolving solution characteristics in population-based solution approaches. 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. The practicality of this stochastic multicriteria MGA approach can be readily extended into numerous disparate applications and can be clearly modified to suit many "real world" planning situations. Such extensions will be explored in future research.

REFERENCES

[1] Brugnach, M., Tagg, A., Keil, F., and De Lange, W.J., 2007, Uncertainty matters: computer models at the science-policy interface, Water Resources Management, 21, 1075-1090.

[2] Janssen, J.A.E.B., Krol, M.S., Schielen, R.M.J., and Hoekstra, A.Y., 2010, The effect of modelling quantified expert knowledge and uncertainty information on model-based decision making, Environmental Science and Policy, 13(3), 229-238.

[3] Tchobanoglous, G., Thiesen, H., and Vigil S., 1993, Integrated Solid Waste Management: Engineering Principles and Management Issues, New York: McGraw-Hill.

[4] Matthies, M., Giupponi, C., and Ostendorf, B., 2007, Environmental decision support systems: Current issues, methods and tools, Environmental Modelling and Software, 22(2), 123-127.

[5] Mowrer, H.T., 2000, Uncertainty in natural resource decision support systems: Sources, interpretation, and importance, Computers and Electronics in Agriculture, 27(1-3), 139-154.

[6] Walker, W.E., Harremoes, P., Rotmans, J., Van der Sluis, J.P., Van Asselt, M.B.A.P., Janssen, J.A.E.B., and Krayer von Krauss, M.P., 2003, Defining uncertainty – a conceptual basis for uncertainty management in model-based decision support, Integrated Assessment, 4(1), 5-17.

[7] Baugh, J.W., Caldwell, S.C., and Brill, E.D., 1997, A mathematical programming approach for generating alternatives in discrete structural optimization, Engineering Optimization, 28(1), 1-31.

[8] Brill, E.D., Chang, S.Y., and Hopkins, L.D., 1982, Modelling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning, Management Science, 28(3), 221-235.

[9] Loughlin, D.H., Ranjithan, S.R., Brill, E.D., and Baugh, J.W., 2001, Genetic algorithm approaches for addressing unmodelled objectives in optimization problems, Engineering Optimization, 33(5), 549-569.

[10] Yeomans, J.S., and Gunalay, Y., 2011, Simulation-optimization techniques for modelling to generate alternatives in waste management planning, Journal of Applied Operational Research, 3(1), 23-35.

[11] Zechman, E.M., and Ranjithan, S.R., 2007, Generating alternatives using evolutionary algorithms for water resources and environmental management problems, Journal of Water Resources Planning and Management, 133(2), 156-165.

[12] Gunalay, Y., Yeomans, J.S., and Huang, G.H., 2012, Modelling to generate alternative policies in highly uncertain environments: An application to municipal solid waste management planning, Journal of Environmental Informatics, 19(2), 58-69.

[13] Imanirad, R., and Yeomans, J.S., 2013, "Modelling to Generate Alternatives Using Biologically Inspired Algorithms". In Yang, X.S. (Ed.), Swarm Intelligence and Bio-Inspired Computation: Theory and Applications, Amsterdam: Elsevier, 313-333.

[14] Imanirad, R., Yang, X.S., and Yeomans, J.S., 2012, A computationally efficient, biologically-inspired modelling-to-generate-alternatives method, Journal on Computing, 2(2), 43-47.

[15] Imanirad, R., Yang, X.S., and Yeomans, J.S., 2012, A co-evolutionary, nature-inspired algorithm for the concurrent generation of alternatives, Journal on Computing, 2(3), 101-106.

[16] Imanirad, R., Yang, X.S., and Yeomans, J.S., 2013, Modelling-to-generate-alternatives via the firefly algorithm, Journal of Applied Operational Research, 5(1), 2013, 14-21.

[17] Imanirad, R., Yang, X.S., and Yeomans, J.S., 2013, A Concurrent Modelling to Generate Alternatives Approach Using the Firefly Algorithm, International Journal of Decision Support System Technology, 5(2), 33-45.

[18] Imanirad, R., Yang, X.S., and Yeomans, J.S., 2013, A biologically-inspired metaheuristic procedure for modelling-to-generate-alternatives, International Journal of Engineering Research and Applications, 3(2), 1677-1686.

[19] Yeomans, J.S., 2018, "An Efficient Computational Procedure for Simultaneously Generating Alternatives to an Optimal Solution Using the Firefly Algorithm". In Yang X.S. (Ed.), Nature-Inspired Algorithms and Applied Optimization. New York: Springer, 261-273.

[20] Yeomans, J.S., 2017, Simultaneous Computing of Sets of Maximally Different Alternatives to Optimal Solutions, International Journal of Engineering Research and Applications, 7(9), 21-28.

[21] Yeomans, J.S., 2017, An Optimization Algorithm that Simultaneously Calculates Maximally Different Alternatives, International Journal of Computational Engineering Research, 7(10), 45-50.

[22] Yeomans, J.S., 2017, A Computational Algorithm for Creating Alternatives to Optimal Solutions, Transactions on Machine Learning and Artificial Intelligence, 5(5), 58-68.

[23] Yeomans, J.S., 2018, Computationally Testing the Efficacy of a Modelling-to-Generate-Alternatives Procedure for Simultaneously Creating Solutions, Journal of Computer Engineering, 20(1), 38-45.

[24] Yeomans, J.S., 2019, A Simultaneous Modelling-to-Generate-Alternatives Procedure Employing the Firefly Algorithm. In Dey, N. (Ed.), Technological Innovations in Knowledge Management and Decision Support. Hershey, Pennsylvania: IGI Global, 19-33.

[25] Yeomans, J.S., 2019, A Simulation-Optimization Algorithm for Generating Sets of Alternatives Using Population-Based Metaheuristic Procedures, Journal of Software Engineering and Simulation, Forthcoming.

[26] Yeomans, J.S., 2012, Simulation-Driven Optimization in Waste Management Facility Expansion Planning, Journal on

Computational Methods in Science and Engineering, 12(1/2), 111-127.

[27] Yeomans, J.S., 2017, Simulation-Optimization for Stochastic Modelling to Generate Alternatives, International Journal of Engineering Sciences and Management Research, 4(10), 1-8.

[28] Yeomans, J.S., 2019, A Stochastic Simulation-Optimization Method for Generating Waste Management Alternatives Using Population-Based Algorithms, Applied Science and Innovative Research, 3(3), 92-105.

[29] Fu, M.C., 2002, Optimization for simulation: theory vs. practice, INFORMS Journal on Computing, 14(3), 192-215.

[30] Imanirad, R., Yang, X.S., and Yeomans, J.S., 2016, “Stochastic Decision-Making in Waste Management Using a Firefly Algorithm-Driven Simulation-Optimization Approach for Generating Alternatives”. In. Koziel, S., Leifsson, L., and Yang, X.S. (Eds.), Recent Advances in Simulation-Driven Modeling and Optimization, Heidelberg, Germany: Springer, 299-323.

[31] Kelly, P., 2002, Simulation optimization is evolving, INFORMS Journal on Computing, 14(3), 223-225.

[32] Zou, R., Liu, Y., Riverson, J., Parker, A., and Carter, S., 2010, A nonlinearity interval mapping scheme for efficient waste allocation simulation-optimization analysis, Water Resources Research, 46(8), 1-14.

[33] Yeomans, J.S., 2012, Waste management facility expansion planning using simulation-optimization with grey programming and penalty functions, International Journal of Environmental and Waste Management, 10(2/3), 269-283.

[34] Yeomans, J.S., 2008, Applications of simulation-optimization methods in environmental policy planning under uncertainty, Journal of Environmental Informatics, 12(2), 174-186.

[35] Yeomans, J.S., and Yang, X.S., 2014, Municipal waste management optimization using a firefly algorithm-driven simulation-optimization approach, International Journal of Process Management and Benchmarking, 4(4), 363-375.

[36] Linton, J.D., Yeomans, J.S., and Yoogalingam, R., 2002, Policy planning using genetic algorithms combined with simulation: The case of municipal solid waste, Environment and Planning B: Planning and Design, 29(5), 757-778.

[37] Yeomans, J.S., 2019, Waste Management Using Multicriteria Population-Based Simulation-Optimization Algorithms, Journal of Waste Management and Disposal, 2(1), 1-8.

[38] Yeomans, J.S., 2018, An Algorithm for Generating Sets of Maximally Different Alternatives Using Population-Based Metaheuristic Procedures, Transactions on Machine Learning and Artificial Intelligence, 6(5), 1-9.