A MULTIOBJECTIVE PARTICLE SWARM OPTIMIZATION MODEL FOR RESERVOIR OPERATIONS AND PLANNING

Alexandre M. Baltar[1] and Darrell G. Fontane[2]

ABSTRACT

This paper presents an application of an evolutionary optimization algorithm for multiobjective analysis for reservoir operations and planning. A multiobjective particle swarm optimization (MOPSO) algorithm is used to find nondominated solutions with four objectives: (i) maximize annual firm water supply; (ii) maximize annual firm energy production; (iii) minimize flood risk; and (iv) maximize the overall reliability of the system.

The results of this study showed that the MOPSO algorithm was able to find well distributed Pareto solutions in the objective space. An interactive graphical method was also developed to display nondominated solutions in such way that the best compromise solutions can be identified, for different relative importance given to each objective. The method allows the decision maker to explore the Pareto set and visualize not only the best compromise solution but also sets of solutions that provide similar compromises.

KEY WORDS

multiobjective, decision making, particle swarm optimization, pareto set, reservoir planning.

INTRODUCTION

Decision making in water resources planning and management frequently involves multiple objectives. As greater attention is being given to the environmental and social aspects of water resources allocation and management the need for effective multiobjective optimization approaches is increasing. Many of the developments in the area of multiobjective analysis in the United States have come from the field of water resources (Goicoechea et al. 1982).

In multiobjective optimization, a set of nondominated solutions is usually produced instead of a single recommended solution. According to the concept of nondominance, also referred as Pareto optimality, a solution to a multiobjective problem is nondominated, or Pareto optimal, if no objective can be improved without worsening at least one other objective.

Traditional multiobjective optimization methods attempt to find the set of nondominated solutions using mathematical programming. In the case of nonlinear problems, the weighting

[1] PhD Student, Dept. of Civil and Envmtl. Engrg., Colorado State University, 1600 W. Plum St. 26H, Fort Collins, CO 80521, U.S.A., Phone 970/492-9899, FAX 970/491-6787, abaltar@engr.colostate.edu

[2] Professor, Dept. of Civil and Envmtl. Engrg., Colorado State University, B213 Engineering Building, Fort Collins, CO 80521, U.S.A., Phone 970/491-5248, FAX 970/491-6787, fontane@engr.colostate.edu

method and the ε-constraint method are the most commonly used techniques. Both methods transform the multiobjective problem into a single objective problem which can be solved using nonlinear optimization.

With the weighting method, nondominated solutions are obtained if all weights are positive but not all Pareto optimal solutions can be found unless all objective functions as well as the feasible region are convex. Other disadvantage of this method is that many different sets of weights may produce the same solution, compromising the efficiency of the method. When the weights reflect the preferences of the Decision Maker (DM), the method gives the best-compromise solution, i.e. the solution which produces the highest utility to the DM. The ε-constraint method, on the other hand, does not require convexity but only leads to nondominated solutions if certain specific conditions are satisfied (Miettinen 2001).

According to Coello Coello (2001), the first hints on the potential of evolutionary algorithms (EA) for multiobjective optimization occurred in the 1960s but this research area remained largely unexplored until mid 1980s. This author also highlighted two advantages of evolutionary algorithms that make them particularly suitable for multiobjective optimization, when compared to traditional mathematical programming techniques:

- EA work simultanously with a set of possible solutions, the so-called population, and several nondominated solutions may be found in a single run of the algorithm;
- EA are less sensitive to the shape or continuity of the Pareto surface.

Since the mid 1980s, a growing number of evolutionary multiobjective optimization algorithms have been proposed in the literature (Fonseca and Fleming 1995). Although some studies have attempted to compare different algorithms (e.g. Zitzler and Thiele 1999, Coello Coello et al. 2004), these comparisons are always restricted to the type of functions and problems being solved.

Particle swarm optimization – PSO (Kennedy and Eberhart 1995) is one of the newest techniques within the family of evolutionary optimization algorithms. The algorithm is based on an analogy with the choreography of flight of a flock of birds. Due to its fast convergence, PSO has been advocated to be especially suitable for multiobjective optimization (Coello Coello et al. 2004).

There are many variants of the single objective PSO but in most of them the movement of the particles towards the optimum is governed by equations similar to the following:

$$\vec{v}_i(t+1) = w \cdot \vec{v}_i(t) + c_1 \cdot r_1 \cdot \left(\vec{P}_i(t) - \vec{x}_i(t)\right) + c_2 \cdot r_2 \cdot \left(\vec{P}_g(t) - \vec{x}_i(t)\right) \quad (1)$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \quad (2)$$

Where w is an inertia coefficient that has an important role balancing global (a large value of w) and local search (a small value of w), c_1 and c_2 are constants (usually $c_1 = c_2 = 2$), r_1 and r_2 are uniform random numbers in [0,1], P_i is the best position vector of particle i so far, P_g is the best position vector of all particles so far, $x_i(t)$ is the current position vector of particle i, and $v_i(t)$ is the current "velocity" of particle i. Mendes et al. (2004) suggests an inertia coefficient w of less than 1, while other authors recommend to start with larger values and decrease with time, for example from a value of 1.4 to 0.5 (e.g. Elbeltagi et al. 2005, Jung and Karney 2006). Coello Coello et al. (2004) highlighted the sensitivity of the standard PSO algorithm to the value of w and proposed the introduction of a mutation operator that assures an adequate global search while keeping a small value of w (suggested 0.4) which favors a refined local search.

Several applications with evolutionary multiobjective optimization have been recently reported in the water resources literature (e.g. Liong et al. 2001, Muleta and Nicklow 2005, Suen et al. 2005, Tang and Reed 2005, Kapelan et al. 2005). None of these applications, however, used multiobjective PSO. Jung and Karney (2006) compared the performances of single-objective Genetic Algorithm and PSO approaches to optimize the selection, sizing, and placement of hydraulic devices for transient protection. The authors studied six different cases and concluded that both algorithms produced very similar results in most cases but the PSO found better solutions when the same population size and number of iterations were applied.

In this paper, we use a modified version of the multiobjective PSO (MOPSO) proposed by Coello Coello et al. (2004) with an application to reservoir operations and planning. We also propose a graphical procedure to incorporate the DM's preferences, further exploring the set of nondominated solutions and displaying the best-compromise solution as well as families of solutions with similar compromises.

MULTIOBJECTIVE PARTICLE SWARM OPTIMIZATION

In the MOPSO algorithm (Coello Coello et al. 2004), the performances of different particles are always compared in terms of their dominance relations. The main characteristic of this algorithm is the use of an external repository which stores nondominated solutions. The algorithm starts generating an initial population. All the particles of this population are compared to each other and the nondominated particles are stored in the repository. The particles' positions will be subsequently updated using the following:

$$\vec{v}_i(t+1) = w \cdot \vec{v}_i(t) + c_1 \cdot r_1 \cdot \left(\vec{P}_i(t) - \vec{x}_i(t)\right) + c_2 \cdot r_2 \cdot \left(\vec{R}_h(t) - \vec{x}_i(t)\right) \quad (3)$$

Where R_h is a solution selected from the external repository in each iteration t, and the other terms have already been defined, with $w = 0.4$.

The best position vector of particle i, P_i, is initially set equal to the initial position of particle i. In the subsequent iterations, the best position vector is updated in the following way: if the current $P_i(t)$ dominates the new position $x_i(t+1)$ then $P_i(t+1) = P_i(t)$, if the new position $x_i(t+1)$ dominates $P_i(t)$ then $P_i(t+1) = x_i(t+1)$, if no one dominates the other then one of them is randomly selected to be the $P_i(t+1)$.

In MOPSO there is no such thing as the best position vector (P_g) as in the standard PSO. There are several equally good nondominated solutions stored in the external repository. To update the velocity of each particle using Equation (3), the algorithm has to select one of the position vectors stored in the repository. This selection is made in such a way that nondominated solutions located in regions more densely populated in the objective space have lower probabilities of being selected, therefore leading to better distributions of points in the Pareto front. Instead of using the adaptive grid proposed in Coello Coello et al. (2004), the approach followed in this study simply calculates, in the objective space, the density of points around each solution stored in the repository and performs a roulette wheel selection such that the probability of choosing one point is inversely related to its associated density.

In every iteration *t*, the new positions of all particles are compared among themselves and the nondominated ones are then compared with all solutions stored in the repository. The repository is then updated, adding new nondominated solutions and eliminating old solutions that are now dominated. The size of the repository is an important parameter to be set. Once the repository is full and a new nondominated solution is found, then this new solution takes the place of another nondominated solution in the repository which is selected randomly using a similar procedure based on density as described above but now assigning higher probabilities of being selected to solutions located in denser regions of the objective space. The algorithm runs until the maximum number of iterations (cycles) is reached.

The algorithm handles constraints in a very simple and efficient way. When comparing two different solutions, with at least one infeasible, the algorithm does the following: (i) one feasible solution dominates other which is infeasible; (ii) with two infeasible solutions the one with smaller violation of the constraints dominates the other. To implement that when several constraints are imposed, an index is calculated to reflect the aggregated degree of constraint violation.

AN APPLICATION TO RESERVOIR OPERATIONS AND PLANNING

MOPSO is applied to find nondominated solutions for the operation of a single reservoir with up to four of the following objectives: (i) maximize annual firm water supply, (ii) maximize annual firm energy production, (iii) minimize flood risk, and (iv) maximize the overall reliability of the system. The model was implemented in a spreadsheet format using Microsoft Excel© and Visual Basic for Applications.

RESERVOIR SIMULATION MODEL

The reservoir model (Fontane 2002) performs the mass balance in a monthly time step for a sequence of five years of monthly inflows. In this example, 34 years of monthly flow data were used which allows the user to select 30 different sequences of five years to be simulated. The flow data is taken from McMahon and Mein (1986, p. 347). The following parameters must be provided:

- Monthly water use coefficients
- Monthly energy use coefficients
- Parameters *a* and *b* of the area-volume relation as follows

$$Area(t) = a \cdot [V(t)]^b$$

- Monthly average evaporation depths in meters
- Parameters *c*, *d*, and *e* for head calculations as follows

$$Head(t) = c \cdot [V(t)]^d + e$$

- Maximum flow through turbines and minimum required head

The monthly water and energy use coefficients are used to convert annual demands to the monthly time scale. The required water releases to meet the monthly energy demands are then calculated and the mass balance of the reservoir is performed as follows:

$$V(t+1) = V(t) + Inflow(t) - Evap(t) - \mathrm{Re}\,l(t) \quad (4)$$

$$Evap(t) = mthE \cdot Area(t) \quad (5)$$

$$\mathrm{Re}\,l(t) = \max\{Wrel(t), Erel(t)\} \quad (6)$$

Where V is volume, $mthE$ are monthly evaporation depths, $Wrel$ and $Erel$ are the required releases to meet water and energy monthly demands, respectively.

The user can evaluate the performance of the reservoir system for different values of annual water demand, annual energy demand, and active volume. A nonlinear optimization tool using Excel Solver is also included allowing the calculation of: (a) annual firm water for fixed values of active volume and annual energy supply; or (b) annual firm energy for fixed values of active volume and annual water supply. In either case (a) or (b) the annual demand (water or energy) is the objective function (to be maximized) and also the decision variable. The optimization is subject to the following constraints: (i) no water shortages in any month; (ii) no energy shortages in any month; and (iii) ending volume equal to the active volume. The model also assumes the starting volume to be equal to the active volume. Using these optimization tools, an ε-constraint approach can be implemented to find the trade-off curve between annual firm water and annual firm energy, for selected inflow sequences and active volumes. Figure 1 presents such a curve for an active volume of 800 million cubic meters, using the driest inflow sequence.

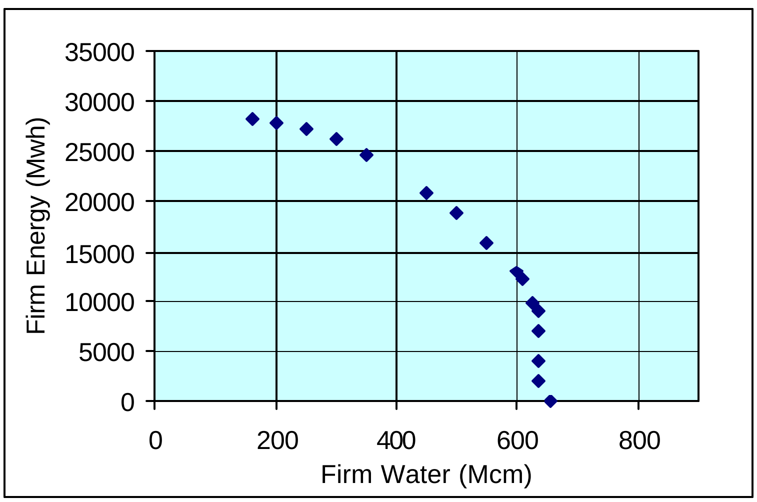

Figure 1: Annual Firm Water vs Annual Firm Energy Trade-off using ε-Constraint Method

MULTIOBJECTIVE OPTIMIZATION PROBLEM

The MOPSO was coded in VBA. The model includes an interface which allows the user to set all MOPSO parameters and choose what objectives should be optimized. If any of the four aforementioned objectives is not selected, the user can specify a fixed value for that objective and this is included as a constraint in the optimization of the remaining selected objectives. If only one objective is selected the MOPSO works as a standard PSO.

Figure 2 presents the water-energy trade-off curve using MOPSO with 100 particles, repository size of 50 solutions, and 50 iterations. The processing time was 40 seconds in a PC AMD Athlon™ 64, 2.2 GHz.

The flood control objective is introduced by minimizing the active volume within a specified interval, [400,800] in this example. In this case, the active volume becomes also a decision variable. Figure 3 presents a 3-d plot of the interpolated Pareto surface and the Pareto solutions obtained by MOPSO with 150 particles, 150 solutions in the repository, and 150 iterations. The processing time was 325 seconds in the same computer. This problem involves a 3-dimensional search space and a 3-dimensional objective space.

The reliability objective is implemented by evaluating each solution for 30 different five-year inflow sequences. The number of sequences with failures (water shortage, energy shortage, or an ending volume less than the full active volume) is counted and a reliability index is calculated by Equation 7 to follow. No dimension is added in the search space, which is still defined by firm water, firm energy, and active volume decision variables. The Pareto front is defined in a 4-dimensional objective space, however.

$$\mathrm{Re}\,ly_i = 1 - \frac{Nfail_i}{30} \tag{7}$$

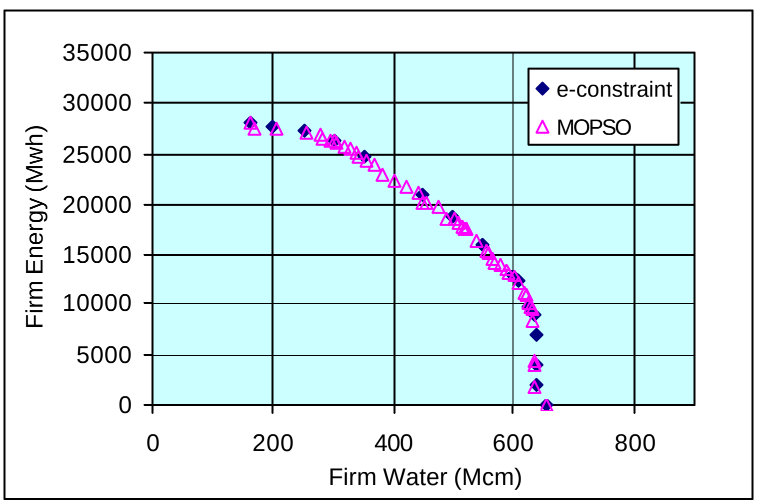

Figure 2: Water-Energy Trade-off Curve by ε-Constraint and MOPSO

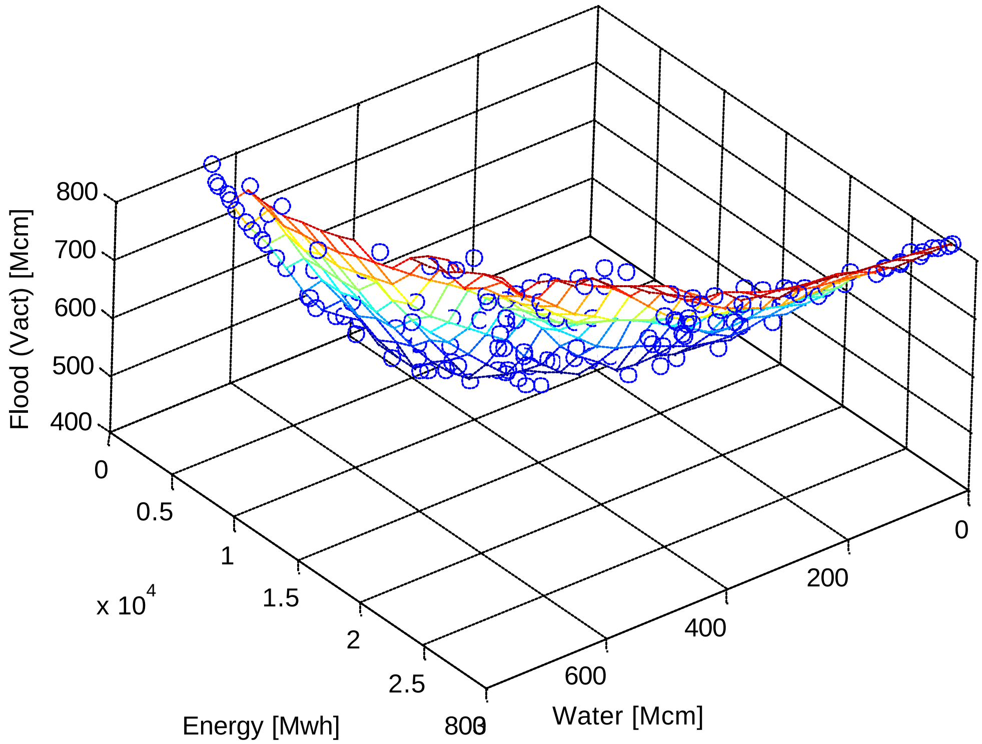

Figure 3: Water-Energy-Flood Trade-off Surface

PROCEDURE TO VISUALIZE AND EXPLORE THE PARETO SET

A procedure was developed to incorporate the preferences of the DM as well as to visualize and explore the Pareto set. First, all Pareto solutions are normalized using a compromise programming approach with a Euclidian norm (L-2 norm). All objectives are placed as vertices equally spaced in a circumference of diameter 1. Let N be the number of objectives. The first objective is arbitrarily assigned to coordinates [0.5,1.0]. The x and y coordinates of the following objectives are given by the following:

$$q = \frac{\left(p - \frac{2p}{N}\right)}{2} \tag{8}$$

$$x_{ob}(k) = x_{ob}(k-1) + \cos q \cdot \cos\left[\frac{-p}{2} + p \cdot k - (2 \cdot k - 3) \cdot q\right] \tag{9}$$

$$y_{ob}(k) = y_{ob}(k-1) + \cos q \cdot \sin\left[\frac{-p}{2} + p \cdot k - (2 \cdot k - 3) \cdot q\right] \tag{10}$$

Where N is the number of objectives and $k = 2..N$.

The normalized metrics are calculated for each objective as follows:

$$CP(i,k)=\frac{[Best(k)-R(i,k)]^2}{[Best(k)-Worst(k)]^2} \tag{11}$$

Where $CP(i,k)$ is the [0,1] normalized metric of particle i for objective k, and $R(i,k)$ is the value of objective k of particle i in the repository.

The normalized metric for each objective are then used to calculate a new coordinate measured in the diameter corresponding to that objective. The new coordinates are given by:

$$x_{CP}(i,k)=x_{ob}(k)+CP(i,k)\cdot\cos\left[\frac{p}{2}+p\cdot k-(2\cdot k-2)\cdot q\right] \tag{12}$$

$$y_{CP}(i,k)=y_{ob}(k)+CP(i,k)\cdot\sin\left[\frac{p}{2}+p\cdot k-(2\cdot k-2)\cdot q\right] \tag{13}$$

Each particle in the repository has now a (x,y) coordinate for each of the N objectives. The particle is them plotted in the centroid defined by these N points. A weighted average of the normalized metrics is then calculated based on weights given by the DM. The DM can change the weights and automatically see the best-compromise Pareto solution, as well as other Pareto solutions with similar compromise. Figure 4 presents the water-energy-flood trade-off graph for two sets of weights, one with equal weights and other with higher importance given to firm water supply. Figure 5 presents the water-energy-flood-reliability trade-off for two sets of weights, one set with higher weights for water and energy, and other with higher weight to reliability.

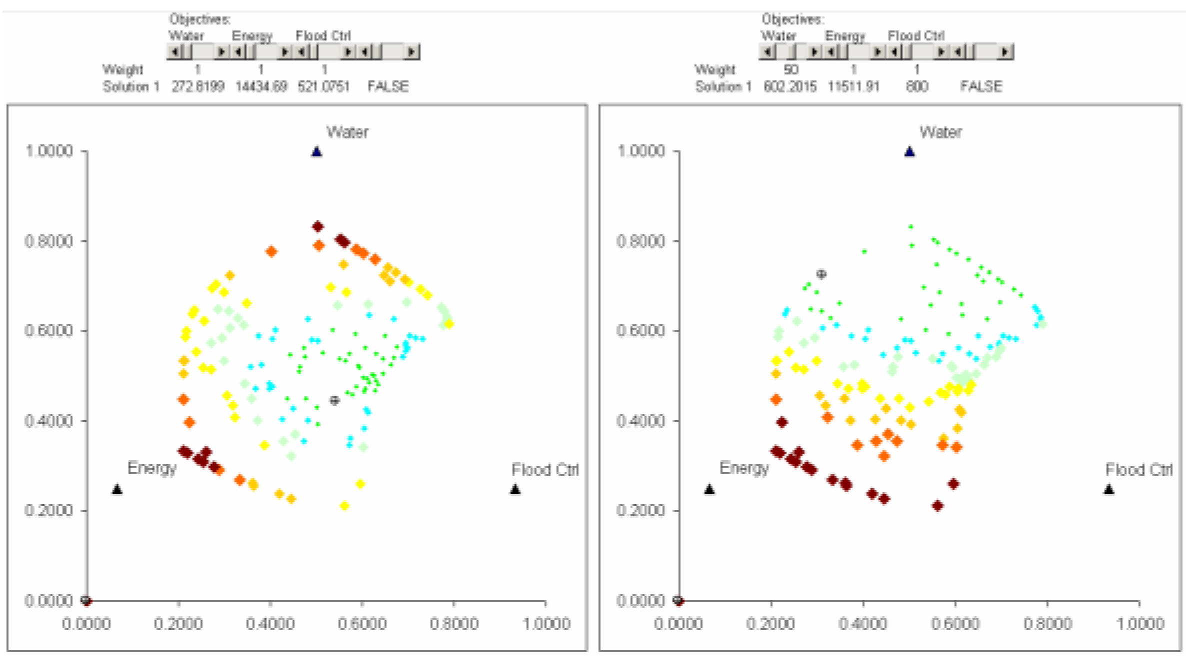

Figure 4: Water-Energy-Flood Trade-off Graphs for Two Sets of Weights

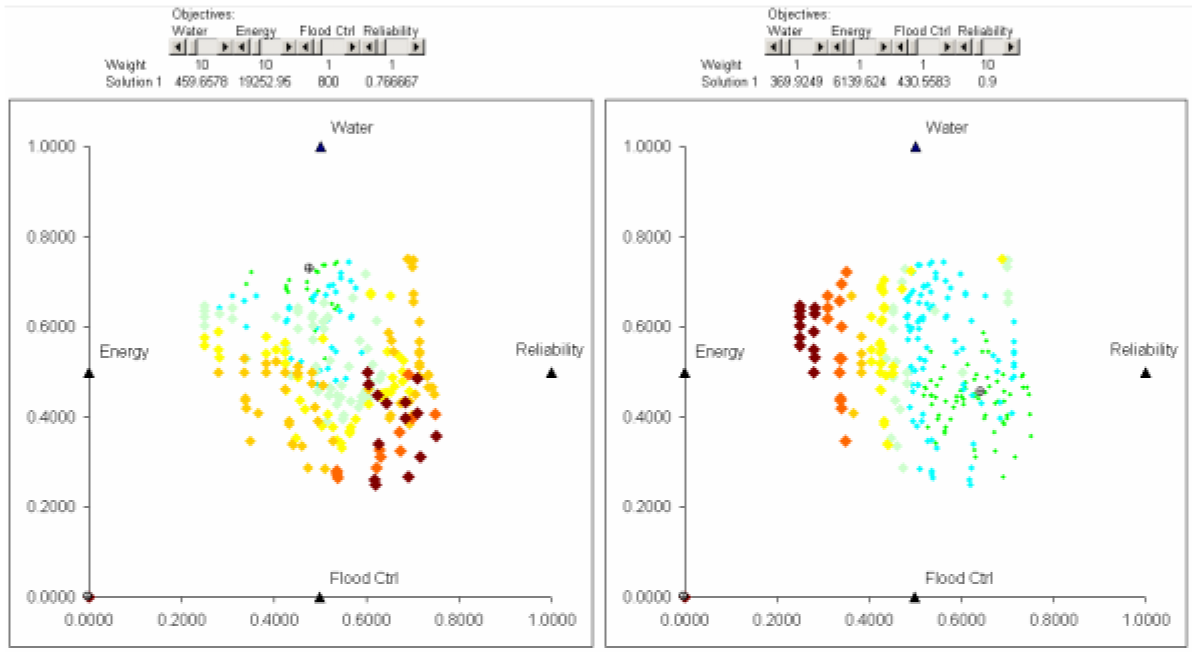

Figure 5: Water-Energy-Flood-Reliability Trade-off Graphs for Two sets of Weights[3]

CONCLUSIONS

A multiobjective particle swarm optimization algorithm was applied in a model for reservoir operations and planning. The algorithm proved to be able to find well-distributed nondominated solutions in the objective space. For the case of two objectives, water supply and energy production, the algorithm was compared to an ε-constraint approach using the Excel Solver. Both methods found the same Pareto front but MOPSO was much faster and did not present any convergence problems. There were convergence problems with the nonlinear optimization using Solver. The Pareto solutions for the cases of three and four objectives were tested individually to see if any objective could be improved without worsening the others. These tests showed that many of the obtained Pareto solutions could still be improved but the average possible improvement was relatively small and, as expected, it decreased with the number of iterations. For three objectives, when the number of iterations was at least 100, the average improvements where about 5%, indicating that the solutions obtained were already very close to the real Pareto front.

The graphical procedure proposed in this paper can help decision makers to explore Pareto sets when three or more objectives are considered. The best-compromise solution may be identified as well as subsets of the Pareto set that provide similar compromises.

ACKNOWLEDGMENTS

The first author of this paper is a sponsored by CNPq, Brazil.

---

3 The processing time for this 4-objective problem was approximately 45 minutes, using 200 particles, repository size of 250 solutions, and 150 iterations.

REFERENCES

Coello Coello, C.A. (2001). "A Short Tutorial on Evolutionary Multiobjective Optimization." Springer-Verlag, *Lect. Notes Comp. Sci. 1993*, Berlin, Germany, 21-40.

Coello Coello, C.A., Pulido, G.T., and Lechuga, M.S. (2004). "Handling Multiple Objectives with Particle Swarm Optimization." *IEEE Transactions on Evolutionary Computation*, IEEE, Piscataway, NJ, 8 (3) 256-279.

Elbeltagi, E., Hegazy, T., and Grierson, D. (2005). "Comparison Among Five Evolutionary-based Optimization Algorithms." *Adv. Engineering Informatics*, Elsevier, 19 (1) 43-53.

Fonseca, C.M. and Fleming, P.J. (1995). "An Overview of Evolutionary Algorithms in Multiobjective Optimization." *Evolut. Comptn.*, MIT Press, Cambridge, MA, 3 (1) 1-16.

Fontane, D.G. (2002). "Multi-objective Simulation and Optimization of Reservoir Operation Using Excel." *Proc. 2nd Joint Fed. Interag. Hydrologic Modlng. Conf.*, Las Vegas, NE.

Goicoechea, A., Hansen, D.R., and Duckstein, L. (1982). *Multiobjective Decision Analysis with Engineering and Business Applications*. J. Wiley & Sons, New York, NY, 519 pp.

Jung, B.S. and Karney, B.W. (2006). "Hydraulic Optimization of Transient Protection Devices Using GA and PSO Approaches." *J. Water Resour. Plng. and Mgmt.*, ASCE, Reston, VA, 132 (1) 44-52.

Kapelan, Z.S., Savic, D.A., and Walters, G.A. (2005). "Optimal Sampling Design Methodologies for Water Distribution Model Calibration." *J. Hydraulic Engrg.*, ASCE, Reston, VA, 131 (3) 190-200.

Kennedy, J. and Eberhart, R. (1995). "Particle Swarm Optimization." *Proc. 4th IEEE Int. Conf. on Neural Networks*, IEEE, Piscataway, NJ, 1942-1948.

Liong, S., Khu, S., and Chan, W. (2001). "Derivation of Pareto Front with Genetic Algorithm and Neural Network." *J. Hydrologic Engrg.*, ASCE, Reston, VA, 6 (1) 52-61.

McMahon, T.A. and Mein, R.G. (1988). *River and Reservoir Yield*. Water Resources Publications, Littleton, CO, 368 pp.

Mendes, R., Kennedy, J., and Neves, J. (2004). "The Fully Informed Particle Swarm: Simpler, Maybe Better." *IEEE Transactions on Evolutionary Computation*, IEEE, Piscataway, NJ, 8 (3) 204-210.

Miettinen, K. (2001). "Some Methods for Nonlinear Multi-objective Optimization." Springer-Verlag, *Lecture Notes in Computer Science 1993*, Berlin, Germany, 1-20.

Muleta, M.K. and Nicklow, J.W. (2005). "Decision Support for Watershed Management Using Evolutionary Algorithms." *J. Water Resour. Plng. and Mgmt.*, ASCE, Reston, VA, 131 (1) 35-44.

Suen, J., Eheart, J.W., and Herricks, E.E. (2005). "Integrating Ecological Flow Regimes in Water Resources Management Using Multiobjective Analysis." *Proc. ASCE/EWRI World Water and Envmtl. Resour. Cong.*, ASCE, Reston, VA.

Tang, Y. and Reed, P.M. (2005). "Multiobjective Tools and Strategies for Calibrating Integrated Models." *Proc. ASCE/EWRI World Water and Envmtl. Resour. Cong.*, ASCE, Reston, VA.

Zitzler, E. and Thiele, L. (1999). "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach." *IEEE Transactions on Evolutionary Computation*, IEEE, Piscataway, NJ, 3 (4) 257-271.