Journal of Hydrology 511 (2014) 776–785

Contents lists available at ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Multiobjective genetic algorithm conjunctive use optimization for production, cost, and energy with dynamic return flow

Richard C. Peralta *, Ali Forghani, Hala Fayad

Civil and Environmental Engineering Dept., Utah State University, 4110 Old Main Hill, Logan, UT 84322-4110, USA

ARTICLE INFO

Article history:
Received 22 November 2013
Received in revised form 21 January 2014
Accepted 22 January 2014
Available online 4 February 2014
This manuscript was handled by Geoff Syme, Editor-in-Chief

Keywords:
Conjunctive use
Genetic algorithm
Multiobjective optimization
Nonlinear aquifer

SUMMARY

Many real water resources optimization problems involve conflicting objectives for which the main goal is to find a set of optimal solutions on, or near to the Pareto front. E-constraint and weighting multiobjective optimization techniques have shortcomings, especially as the number of objectives increases. Multiobjective Genetic Algorithms (MGA) have been previously proposed to overcome these difficulties. Here, an MGA derives a set of optimal solutions for multiobjective multiuser conjunctive use of reservoir, stream, and (un)confined groundwater resources. The proposed methodology is applied to a hydraulically and economically nonlinear system in which all significant flows, including stream-aquifer-reservoir-diversion-return flow interactions, are simulated and optimized simultaneously for multiple periods. Neural networks represent constrained state variables. The addressed objectives that can be optimized simultaneously in the coupled simulation–optimization model are: (1) maximizing water provided from sources, (2) maximizing hydropower production, and (3) minimizing operation costs of transporting water from sources to destinations. Results show the efficiency of multiobjective genetic algorithms for generating Pareto optimal sets for complex nonlinear multiobjective optimization problems.

1. Introduction

Most optimization problems for conjunctive use of surface water and groundwater involve multiple objectives. In single objective optimization, one attempts to obtain the mathematically best decision that is hopefully the global minimum or the global maximum depending on the optimization problem. In an optimization problem having conflicting multiple objectives, no single solution is best (global minimum or maximum) with respect to all objectives. For a typical multiobjective optimization problem, there exists a set of solutions, known as Pareto-optimal solutions, or nondominated solutions (Hans, 1988) that are superior to all other solutions in the search space when all objectives are considered, but are inferior to some solutions in terms of one or more objectives.

Because none of the solutions in the nondominated set are absolutely better than any other, any one of them might be an acceptable solution, depending upon personal preferences or factors not included within the optimization problem. A solution preferred by one water manager may not be acceptable to another manager or in a changed environment. It aids decision makers to know numerous alternative solutions within or near the true Pareto-optimal set (Srinivas and Deb, 1995).

We can classify optimization methods into two general categories; (1) classical methods such as linear programming (LP), nonlinear programming (NLP), and dynamic programming (DP) and (2) evolutionary or heuristic methods. Most published conjunctive use optimization works employ classical optimization methods (Peralta et al., 1992; Ejaz and Peralta, 1995; Belaineh et al., 1999; Barlow et al., 2003; Vedula et al., 2005; Pulido-Velazquez et al., 2006, 2008; Bharati et al., 2008; Peng et al., 2012). However, the classical optimization methods sometimes have difficulties with extremely nonlinear systems and do not directly yield alternative optimal solutions.

On the other hand, evolutionary methods such as Genetic Algorithms (GAs) can solve optimization problems having nonlinear, nondifferentiable, or even discontinuous functions (Goldberg, 1989). GAs do not require derivatives but use the objective function directly. Aly and Peralta (1999) showed that GA computed a better strategy than formal mixed integer nonlinear programming for a groundwater cleanup problem. Rogers et al. (1995) used a combination of artificial neural networks (ANNs) and GA that involved less computational burden and more flexibility than mathematical programming methods. Nicklow et al. (2010) provide a comprehensive review of the applications of GAs in the field of water resources

* Corresponding author. Tel.: +1 435 881 4947; fax: +1 435 797 1185.
E-mail address: peralta.rc@gmail.com (R.C. Peralta).

http://dx.doi.org/10.1016/j.jhydrol.2014.01.044
0022-1694/

management. Simple GAs (SGAs) have been applied to several single objective optimization problems (Safavi et al., 2009).

Traditional multiobjective optimization techniques (weighting and E-constraint), convert a multi-objective problem into a single objective problem, and employ classical or heuristic optimization methods (Mcphee and Yeh, 2004; Vamvakeridou-Lyroudia et al., 2005; Karamouz et al., 2005, 2007). These techniques identify only one Pareto optimal solution per optimization. Furthermore, optimal solutions obtained via weighting method depend upon user-selected weights. Also, because of the need for inputting reasonable bounds for all objectives, as the number of initial objectives increases, E-constraint use can become difficult.

To overcome these weaknesses, multiobjective evolutionary algorithms such as Multiobjective Genetic Algorithms (MGA) have been developed. MGAs can address all objectives simultaneously without the need to convert them into a single objective problem. Also, they can find a set of optimal solutions in a single run. MGAs have been applied to multiobjective water resources problems (Ritzel et al., 1994; Cieniawski et al., 1995; Reed et al., 2000; Reddy and Kumar, 2006; Yang et al., 2009; Penn et al., 2013). Bazargan-Lari et al. (2008) optimized conjunctive use of surface water and groundwater with conflicting objectives using an MGA. However, they did not explicitly model surface water flows during optimization. Fayad et al. (2012) present a single two-objective pareto optimal curve of MGA-developed solutions for managing a nonlinear groundwater and surface water system. For that two-objective problem they could have used the E-constraint method instead of an MGA. Extending the approach of Fayad et al. (2012), here we address a three-objective optimization problem that benefits from multiobjective heuristic optimization for solution. We detail the process of developing multiple ranks of pareto optimal solutions for three scenarios and illustrate visualizations of three-dimensional objective responses. We use a numerical model and artificial neural networks to simultaneously model all significant flows, interactions, and heads in the nonlinear reservoir, stream, and multilayer aquifer system. Decision variables include: reservoir releases, and diversions from reservoir, stream and aquifer. Diversions for a time step are computed while considering all previous and simultaneous dynamic return flows to stream and aquifer resulting from previous decisions. Objective function components are nonlinear due to hydraulics and economics.

Subsequent sections briefly explain the applied MGA, and detail the study area and methodology. Finally, we present the optimization results and conclusion.

2. Employed MGA

MGAs are a subset of MultiObjective Evolutionary Algorithms (MOEAs), which use a population-based search. MOEAs are attractive in multiobjective problems because they find many Pareto optimal solutions in a single run. Developed MOEAs include: Vector Evaluated Genetic Algorithm (VEGA) (Schaffer, 1985), Niched Pareto Genetic Algorithm (NPGA) (Horn et al., 1994), Nondominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb, 1995), Strength-Pareto Evolutionary Algorithm (SPEA) (Zitzler and Thiele, 1999), and Nondominated Sorting Genetic Algorithm-II (NSGA-II) (Deb et al., 2002).

Deb (2001) presents details of multiobjective optimization using evolutionary algorithms. Zitzler et al. (2000) compares various MOEAs, and Reed et al. (2013) diagnostically assesses different MOEAs.

To demonstrate the comprehensive linking of an MGA optimizer with numerical and neural network simulators, the present work employs an NSGA for optimizing conjunctive use. NSGA uses a ranking selection method to emphasize current nondominated points, and a sharing function method to maintain diversity in the population. To do that, NSGA first calculates the objectives values for all population in a generation. Then, current nondominated individuals are (1) identified and selected from the population, and considered as the individuals of the first front (or first level of non-domination), (2) assigned a large dummy fitness value, and (3) *shared* with their fitness values (as described below). The purpose of sharing is to degrade the fitness values of the similar solutions (which exist close to each other in search space) that helps to emphasize the solutions in less crowded regions and maintain population diversity.

For implementing sharing, Eq. (1) is used. The *sharing function value* between individuals i and j in a front is calculated by:

$$\mathrm{sh}(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{\text{share}}}\right)^2 & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

here d_{ij} is the phenotypic distance between two individuals i and j in the current front and σ_{share} is a parameter called niche size. The *niche count* of individual i is calculated by adding the sharing function values (Eq. (1)) for all individuals around individual i (including i itself) in the current front. Finally, the shared fitness value of individual i is calculated by dividing its dummy fitness value by its niche count. In similar process, the shared fitness value of other individuals in the current front is calculated.

In fact, if d_{ij} between two individuals in a front is less than σ_{share}, the fitness values of those individuals will be degraded. Note that niche count for all individuals is equal or more than one (because sharing function value (Eq. (1)) for individual i with itself is $\mathrm{sh}(d_{ii}) = 1$). Then, the more individuals around individual i exist in distance less than σ_{share}, the higher niche count for individual i is obtained, and consequently the lower shared fitness value for individual i is calculated.

After calculating the shared fitness value of the individuals in the first front, these individuals are ignored temporarily while the rest of population is processed to identify individuals for the second nondominated front (or second level of non-domination). These new sets of points are then assigned a new fitness value that is kept smaller than the minimum shared fitness of the previous front and then their fitness values are shared similar to the process mentioned for the first front. This process continues until the entire population is classified into several fronts and all individuals have shared fitness values. Afterwards, normal SGA reproduction occurs (including crossover and mutation) and the population is reproduced according to the shared fitness values of the individuals. Fig. 1 shows that the NSGA algorithm is similar to a simple GA except for the classification of nondominated fronts and the sharing operations (Srinivas and Deb, 1995).

Deb and Goldberg (1989) showed that for an m-parameter function, σ_{share} may be calculated as:

$$\sigma_{\text{share}} = \frac{\sqrt{\sum_{n=1}^{m} (x_{n,\max} - x_{n,\min})^2}}{2\sqrt[m]{q}} \tag{2}$$

here q is the number of peaks in the solution space.

In general, deciding on the optimum population size is a challenging issue in GA. For NSGA, Mahfoud (1995) proposed Eq. (3) to compute a lower bound on population size required to maintain a fixed number of niches for various sharing models.

$$\text{Population size lower bound} = \frac{\ln\left(\frac{1-\gamma^{1/G}}{c}\right)}{\ln\left(\frac{c-r}{c}\right)} \tag{3}$$

here G = number of generations, c = number of niches (peaks), γ = level of confidence to maintain c niches for at least G generations, r = minimum fitness/maximum fitness ($0.0 \leqslant r \leqslant 1.0$).