Advances in Water Resources 24 (2001) 667–676

Advances in Water Resources

www.elsevier.com/locate/advwatres

Solving nonlinear water management models using a combined genetic algorithm and linear programming approach

Ximing Cai [a,*], Daene C. McKinney [b], Leon S. Lasdon [c]

[a] International Food Policy Research Institute, 2033 K St. N.W., Washington, DC 20006 USA
[b] Department of Civil Engineering, University of Texas, Austin, TX 78712, USA
[c] Department of Management Science and Information Systems, University of Texas, Austin, TX 78712, USA

Abstract

Gradient-based nonlinear programming (NLP) methods can solve problems with smooth nonlinear objectives and constraints. However, in large and highly nonlinear models, these algorithms can fail to find feasible solutions, or converge to local solutions which are not global. Evolutionary search procedures in general, and genetic algorithms (GAs) specifically, are less susceptible to the presence of local solutions. However, they often exhibit slow convergence, especially when there are many variables, and have problems finding feasible solutions in constrained problems with "narrow" feasible regions. In this paper, we describe strategies for solving large nonlinear water resources models management, which combine GAs with linear programming. The key idea is to identify a set of complicating variables in the model which, when fixed, render the problem linear in the remaining variables. The complicating variables are then varied by a GA. This GA&LP approach is applied to two nonlinear models: a reservoir operation model with nonlinear hydropower generation equations and nonlinear reservoir topologic equations, and a long-term dynamic river basin planning model with a large number of nonlinear relationships. For smaller instances of the reservoir model, the CONOPT2 nonlinear solver is more accurate and faster, but for larger instances, the GA&LP approach finds solutions with significantly better objective values. The multiperiod river basin model is much too large to be solved in its entirety. The complicating variables are chosen here so that, when they are fixed, each period's model is linear, and these models can be solved sequentially. This approach allows sufficient model detail to be retained so that long-term sustainability issues can be explored.

Keywords: Water resources management; Nonlinear models; Genetic algorithm; Linear programming

1. Introduction

Gradient-based nonlinear programming (NLP) algorithms are widely available, and can be applied to problems with smooth nonlinear objectives and constraints. Sparsity-exploiting implementations like MINOS [15] and CONOPT2 [4] have solved many problems with thousands of constraints and variables. However, these NLP solvers generally converge to the local solution nearest to the starting point, and their theoretical convergence properties do not hold for nonsmooth problems. Their speed and reliability also decrease as problem size and complexity increase.

Because of these limitations, other methods have been applied to solve large-scale nonlinear reservoir management models in recent years, including evolutionary search methods. These include genetic algorithms (GAs), which often find good approximate solutions even when the model functions are multimodal, discontinuous, or nondifferentiable [17]. Recently, there has been a significant growth of interest in using GAs for water resources planning and design. McKinney and Lin [11] and Huang and Mayer [9] applied a binary-coded GA to models of pump-and-treat groundwater remediation. Savic and Walters [18] applied a GA to least-cost design of water distribution networks, and Halhal et al. [8] studied water network rehabilitation, replacement, and expansion by using a GA. Oliveira and Loucks [16] developed a GA-based approach to search for effective operating policies for dynamic models of multipurpose multi-reservoir systems.

In this paper, we present a combined GA and linear programming (LP) strategy, which is used to solve large nonlinear water resources management models that are difficult, if not impossible, to solve using currently

* Corresponding author.

0309-1708/01/$ - see front matter
PII: S 0 3 0 9 - 1 7 0 8 (0 0) 0 0 0 6 9 - 5

available NLP solvers. The strategy begins by choosing a set of *complicating* variables in the original model. When these variables are fixed, the model becomes much easier to solve (in our examples, it is linear). The complicating variables are varied by a GA, and a linear program is solved to compute the optimal objective value for each set of fixed values suggested by the GA. In principle, another class of search algorithms could replace the GA, for example simulated annealing or tabu search [6].

This strategy has much in common with the widely used benders and generalized benders decomposition (GBD) procedures [5], which vary the complicating variables by solving the benders "master problem". For an application of GBD to multiperiod reservoir models of the form used in this paper, see [3]. However, for the GBD master problem to be tractable, the original model must have special structure (it is sufficient for the complicating and noncomplicating variables to appear separably everywhere). Using a search algorithm to vary the complicating variables, there are no restrictions on how the complicating variables can appear in the model. Of course, this generality is achieved at a price: convergence is not guaranteed, and is often slow in later iterations.

We apply the GA&LP approach to two models: a multiperiod reservoir operation model with nonlinear hydropower generation and reservoir topologic equations, and a long-term dynamic river basin planning model with a large number of nonlinear relationships.

2. Basics of GAs

Detailed descriptions of GAs in water resources literature can be found in McKinney and Lin [11], Huang and Mayer [9], Savic and Walters [18], and Oliveira and Loucks [16]. Only a brief introduction is given here, in order to help readers understand the proposed methodology in this paper.

GAs are a subclass of general artificial-evolution search methods based on natural selection and the mechanisms of population genetics [12]. They belong to a family of optimization techniques in which the solution space is searched by generating candidate solutions with the help of a pseudorandom number generator. These algorithms rely on collective learning processes within a population of individual candidate solutions, each of which represents a point in the space of potential solutions.

There are many variations of GA, but the important features are general. The analogy with nature is established by creating a set of candidate solutions called a population. Each individual in a population is represented by a set of parameters that completely describe a solution. These are encoded into chromosomes, which are, in essence, sets of character strings analogous to the chromosomes found in DNA (see Fig. 1 for an example). Standard GA uses a binary alphabet (characters are 0s or 1s) to form chromosomes. For each generation, a measure of how good each chromosome (or candidate solution) is calculated. This measure is called the *fitness* for each individual in a population. For each individual, its binary alphabet is decoded into parameter values, and then these values are substituted into a program that is used to calculate the fitness. This program may have virtually any form. Next, individuals are selected for "mating" to produce offspring, and this process is called *reproduction*. The reproduction is based on probabilities calculated from the individual's fitness value, which means that strings with a higher value have a higher probability of participating in reproduction and hence of contributing one or more offspring to the next generation. Reproduction consists of *parent selection, crossover* and *mutation*. In crossover, genetic material crosses over from one chromosome to another. However, during crossover, some potentially useful genetic materials may be lost. The process of mutation, which is the occasional random alternation of the value of a string position, protects against such an irrecoverable loss. Crossover plays a primary role in GA, and the probability of crossover is generally set high, while mutation plays a secondary role, and the probability of mutation is set low. For guidance on setting these and other parameters, and on GA strategies in general, see [7,17].

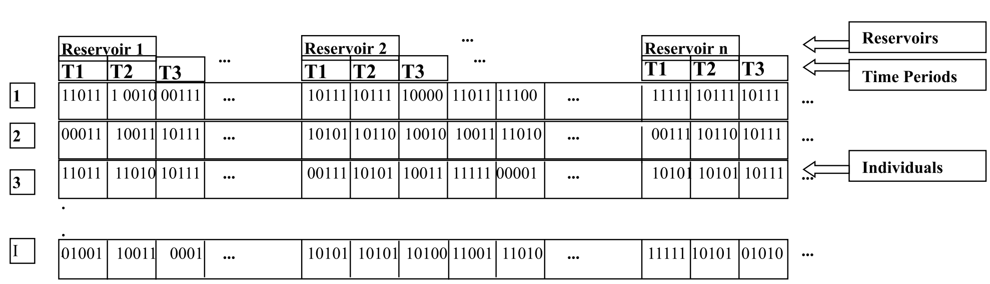

Fig. 1. Variable representation in the GA. A 5-bit binary string is used to represent a single variable. An individual is represented by a string with a length of $5 \cdot N \cdot T$ bites, where N is the number of reservoirs and T is the number of the time periods. The population includes I individuals.

3. Elastic formulation

Most search methods, including GAs, find feasible solutions to constrained problems by penalizing infeasibilities. This requires that constraints which involve complicating variables be formulated in an "elastic" way, by including deviation variables which measure the amount of violation. These are penalized in the objective. The general mathematical statement of the elasticized problem follows. Let x and y be the vectors of noncomplicating and complicating variables, respectively, let p and n be vectors of positive and negative deviation variables, respectively, and let w be a positive penalty weight. The problem to be solved is

$$\text{Maximize} \quad f(x,y) - w(e^{\mathrm{T}}(p+n)) \tag{1}$$

$$\text{subject to} \quad G(x,y) = p - n \tag{2a}$$

$$x \in X, \quad y \in Y, \quad p \geqslant 0, \quad n \geqslant 0, \tag{2b}$$

where e is a vector of ones, G is the vector of "coupling" constraints which involve both sets of variables, and the sets X and Y are determined by those constraints which involve only x and y, respectively. The purpose of the second term in the objective is to force p and n to be small, so that the constraints (2a) are equal to their original right-hand side value of 0, and w must be chosen to be sufficiently large to guarantee this. When y is fixed at some values contained in Y, the above problem, with decision variables x, is called $P(y)$. This problem is assumed to be much easier to solve than the original. For the reservoir operation problem described below, $P(y)$ is a linear program, because all nonlinear terms involve products of components of$x$ and y, or have the form $x_i f_i(y)$, which are linear in x when y is fixed. For the long-term river basin planning model, additional linearizations and approximations must be applied to $P(y)$ to convert it to an LP.

The deviation variables in (2a) are included to deal with values of y such that $P(y)$ is infeasible. Then some components of p or n are positive in the optimal solution to $P(y)$, and the penalty term in the objective causes it to have a much lower value than that for any feasible solution. Any search method which varies y will then have an incentive to avoid infeasible y's.

The optimal objective value of $P(y)$, say $P^*(y)$, need not be a differentiable function of y. Derivative corners can occur whenever the set of active constraints at the optimum of $P(y)$ changes. This nonsmoothness is accounted for in Benders and GBD, where $P^*(y)$ is represented by the point-wise infimum of a collection of functions which overestimate $P^*(y)$. A search method (like a GA) applied to the problem of maximizing $P^*(y)$ will not be affected by its nonsmooth nature.

4. Implementation of the GA

The GA used here discretizes each component of y into 2^L equally spaced values, and represents the discretized variable as a binary string containing L bits, as described in [14]. If we assume that the ith component of y has finite lower and upper bounds l_i and u_i, and that b_j is the value of the jth bit in the string, then the discretized component, denoted by y_i, is given by

$$y_i = l_i + [(u_i - l_i)/(2^L - 1)]\left[\sum_{j=1}^{L} b_j 2^{j-1}\right]. \tag{3}$$

If y has m components, each individual in the GA population is represented by a bit string of length mL, as illustrated in Fig. 1. Using this representation, components of y created by mutation and crossover operations will lie between their lower and upper bounds. Since the GA works in a discretized y-space, it can only approximate the solution of the original problem 1, 2a, 2b.

The initial population consists of randomly generated individuals having the binary form described above. The pair of individuals involved in crossover is determined probabilistically, with selection probabilities proportional to fitness. Fitness is simply the objective value. In moving from one generation to the next, the two fittest individuals are retained, and the others are replaced by new individuals created by crossover and mutation, using the procedure shown below:

1. Select two parents using the selection probability distribution.
2. If crossover (crossover probability varies in the range [0.8,0.95]), produce two offspring using a single point crossover operation. Otherwise, the offspring are the same as the parents.
3. If mutation (mutation probability is 0.01), modify the offspring by reversing a randomly chosen bit.
4. Repeat until enough offspring have been generated to replace all but the two fittest individuals.

Calculating the fitness of an individual whose vector of complicating variables is y requires solving the LP subproblem $P(y)$. For a population of size I (we use $I = 50$ in the following examples), since only the two fittest individuals are retained from the previous generation, $(I - 2)$ LP solutions are required, and this is by far the most time-consuming task in the GA&LP algorithm. This time is reduced by a clustering strategy. Similar individuals are grouped into clusters, where the distance between individuals is taken as the two-norm of the difference of their y-vectors. The fittest individual is chosen as the first cluster center, and all individuals whose distance from this center is less than a threshold become members of the first cluster. The clustering threshold is taken as a small fraction of the sum of the standard deviations of each population variable, over all variables in a given generation. Then the fittest of the unclustered individuals is taken as the next cluster center, and the procedure is repeated until all individuals belong to a cluster. The single LP solved for a cluster uses a discretized y vector determined by rounding the average of the y's for the cluster members. Each member of the cluster receives the fitness value associated with that LP solution. In earlier generations, there may be as many clusters as individuals, but in later generations, there are usually significantly fewer than I clusters, depending on the value of the clustering threshold. For the reservoir operation model of Section 5, using a threshold of 10^{-5} the sum of the standard deviations of each individual and a population size of 50, about 10 clusters are formed in later generations, since later generations tend to become less diverse; while a threshold of 10^{-6} results in 23 clusters in later generations. The choice of the threshold should then be careful, since if the threshold is too high, inferior individuals in a cluster might survive and appear to be fitter than they actually are. The LP solution time is further reduced by starting each clusters' LP after the first from the optimal basis of the previous LP [1].

5. Application to a reservoir operation model

Fig. 2 shows a network diagram of a five-reservoir system, where the reservoirs are capable of hydropower generation, as well as water supply, flood control, and flow augmentation. An optimization model is developed to maximize the production of energy, while satisfying constraints arising from flow augmentation and flood control. Such models are solved by GBD in [3].

The objective is to maximize the sum of the ratios of energy generated to energy demand over all time periods,

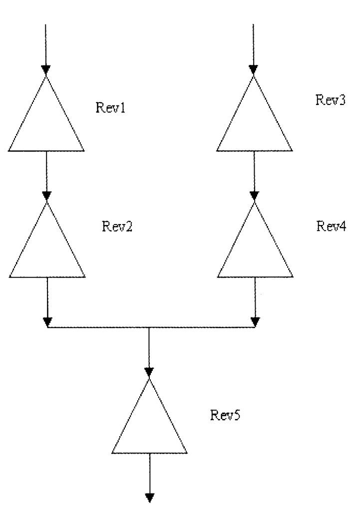

Fig. 2. A hypothetical multi-reservoir system.

$$\max z = \sum_{t\in T}\left(\frac{\sum_{n\in \mathrm{pwst}} \mathrm{PW}(n,t)}{\mathrm{PDEM}(t)}\right), \tag{4}$$

where n and t are the index reservoir/hydropower stations and time periods respectively, $\mathrm{PW}(n,t)$ the energy generated, and $\mathrm{PDEM}(t)$ is the energy demand of the study area in time period t, which is assumed always greater than energy generated. The constraints are:

- Reservoir head–volume relationship:

$$S(n,t) = \alpha_3(n)\cdot H(n,t)^3 + \alpha_2\cdot H(n,t)^2 + \alpha_1\cdot H(n,t) + \alpha_0 \quad \forall n \in \mathrm{rev}. \tag{5}$$

- Reservoir head–area relationship:

$$A(n,t) = \beta_3(n)\cdot H(n,t)^3 + \beta_2\cdot H(n,t)^2 + \beta_1\cdot H(n,t) + \beta_0 \quad \forall n \in \mathrm{rev}, \tag{6}$$

  where S is the reservoir storage, H the reservoir surface elevation, A the reservoir surface area, and a_i, β_i are the constant coefficients, $i = 0, 1, 2, 3$.

- Water balance at reservoirs

$$S(n,t-1) + \mathrm{drn}(n,t) + \sum_{un}\mathrm{RELS}(un,n,t) = S(n,t) + \sum_{ln}\mathrm{RELS}(n,ln,t) + \mathrm{withdw}(n,t) + \mathrm{evap}(n,t)\cdot A(n,t) \quad \forall n \in \mathrm{rev}, \tag{7}$$

  where $\mathrm{drn}(n,t)$ is the natural drainage to reservoirs, constant parameter in the model, $\mathrm{RELS}(un,n,t)$ the flow from upstream reservoir un to reservoir n in period, $\mathrm{RELS}(n,ln,t)$ the flow from reservoir n to downstream reservoir ln in period t, $\mathrm{withdw}(n,t)$ the withdrawal to water demand sites from reservoir n in period t (data item), and $\mathrm{evap}(n,t)$ is the evaporation rate per unit area (data item).

- Hydropower calculation

$$\mathrm{PW}(n,t) \leqslant k(n)\cdot\left\{\frac{1}{2}[H(n,t)+H(n,t-1)] - \mathrm{tw}(n)\right\} \times\left[\sum_{ln}\mathrm{RELS}(n,ln,t) + \mathrm{withdw}(n,t)\right] \quad \forall n \in \mathrm{rev}, \tag{8}$$

$$\mathrm{PW}(n,t) \leqslant \mathrm{PWC}(n) \quad \forall n \in \mathrm{rev}, \tag{9}$$

  where $\mathrm{PWC}(n)$ is the hydropower generation capacity (data item), and $\mathrm{tw}(n)$ is the average tail water level (data item).

  Flood control and downstream flow augmentation are expressed as bounds on reservoir storage and release in each time period

$$S(n,t)^l \leqslant S(n,t) \leqslant S(n,t)^u, \tag{10}$$

$$\mathrm{RELS}(n,t)^l \leqslant \mathrm{RELS}(n,t) \leqslant \mathrm{RELS}(n,t)^u. \tag{11}$$

In the model described above, if the reservoir surface level $H(n,t)$ is fixed, then $S(n,t)$ and $A(n,t)$ can be calculated from (5) and (6), and the generation equations (8), are linear in the remaining variables. These play the role of the coupling constraints (2a) in the general model 1, 2a, 2b. Since S is an increasing function of H, it is easy to calculate upper and lower bounds on H whose satisfaction implies that the storage bounds (10) are satisfied. Therefore it is natural to choose $H(n,t)$ as the complicating variables, and the original model with fixed $H(n,t)$ is an LP. If there are N reservoirs and T time periods, there are $N*T$ complicating variables. The binary coding of $H(n,t)$ is shown in Fig. 1.

We apply the GA&LP approach to six reservoir models with different spatial and temporal dimensions. The simplest has nonlinear relationships (Eqs. (5) (6) and (8)) for only one reservoir (reservoir 1 in Fig. 2), linear relationships for all other reservoirs (see [10] for a linearization method; here we simply assume that energy generated is proportional to reservoir release), and 12 time periods. The most complex model has nonlinear relationships for all five reservoirs and 48 time periods. Table 1 shows the definitions and statistics of all models.

Plots of objective value vs. generations (each GA iteration produces a new generation) for model_1–model_6 are shown in Figs. 3–8, respectively. Objective values in early iterations are negative because the penalty term in (1) is positive, due to positive deviation variables (p and n in (1) and (2a), (2b)) in the LP solution. These infeasibilities are reduced rapidly, as indicated by the fast early improvement in the objective. This is followed by much slower improvement in later iterations, as better feasible solutions are found. We have identified points on each graph roughly where the plateau begins, labeled as the "approximate" objective value. These values, the final objective values, and results from applying the GAMS [1] CONOPT2 nonlinear solver are shown in Table 2. The GA was terminated when the fractional change in objective value in the last 10 generations was less than 10^{-4}. The final objective is significantly better than the "approximate" one, but many more GA iterations are needed to attain it.

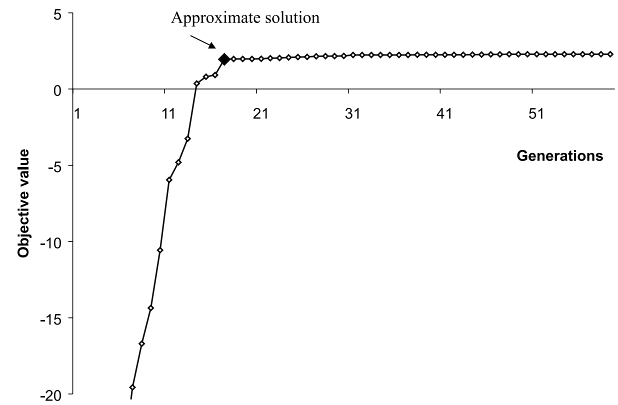

Fig. 3. Objective value vs. generations, model_1, with 1 nonlinear reservoir, 12 time periods.

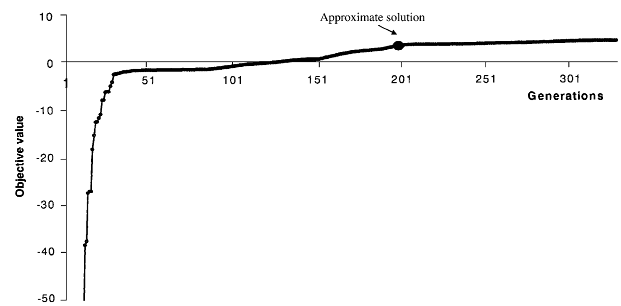

Fig. 4. Objective value vs. generations, model_2, with 1 nonlinear reservoir, 24 time periods.

The last two columns of Table 2 show CONOPT2 runs using both a default starting point, and a starting point which is the final GA&LP solution. We first discuss CONOPT2 results using the default starting point. For the three 12-period models 1, 3, and 5, the final GA&LP objective value is lower (worse) than that obtained by CONOPT2 by 3.5%, 1.5%, and 5.4%, respectively. For each of these three models, we used a real-time based seed for random "gene" generation, ran the GA with the same parameters several times (in a real-time based GA, different runs may result in different solution even with same parameters), and ran the GA for more generations. In all these efforts, however, we

Table 1
Model statistics of the six models for the test of the GA&LP approach

Models	Number of reservoirs with nonlinear relationships	Time periods	Number of equations[a]	Number of variables[a]	Number of nonzero Jacobian elements[a]	Number of non-constant Jacobian elements[a]
Model_1	1 (res1)	12	97	349	655	59
Model_2	1 (res1)	24	193	697	1315	119
Model_3	2 (res1, res2)	12	133	385	762	118
Model_4	2 (res1, res2)	24	265	769	1530	238
Model_5	5 (all res.)	12	241	445	1083	295
Model_6	5 (all res.)	48	1201	2221	4851	1495

[a] From model statistics in GAMS output file.

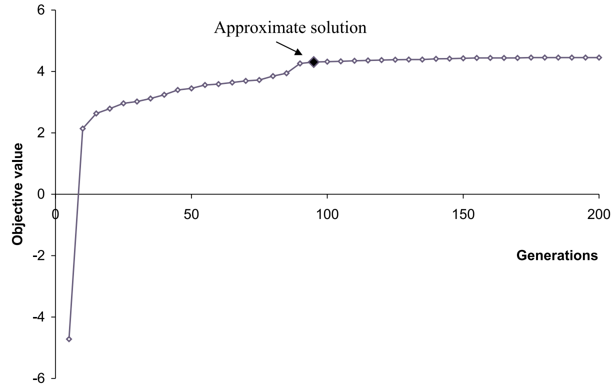

Fig. 5. Objective value vs. generations, model_3, with 2 nonlinear reservoirs, 12 time periods.

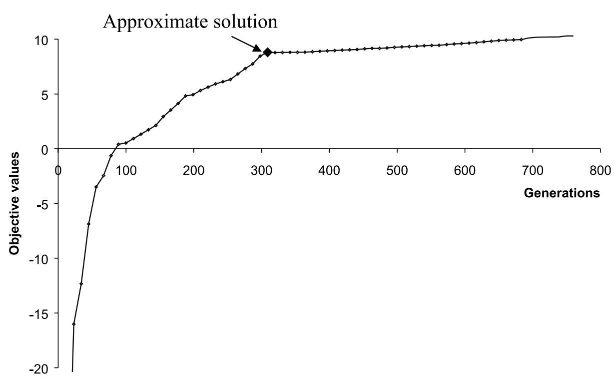

Fig. 6. Objective value vs. generations, model_4, with 2 nonlinear reservoirs, 24 time periods.

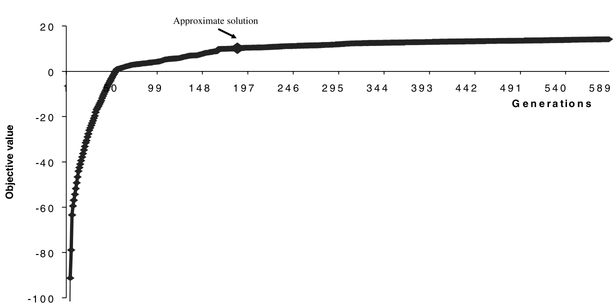

Fig. 7. Objective value vs. generations, model_5, with 5 nonlinear reservoirs, 12 time periods.

never found a solution within 1% of the CONOPT2 solution; while, CONOPT2 yielded the same final objective value using several different starting points.

However, for models 2, 4, and 6, which have 24, 24, and 48 periods, respectively, the GA&LP approach finds solutions that are much better than those attained by CONOPT2 using the default starting point. CONOPT2 finds local solutions to model_2, and model_4 whose objectives are, respectively, 35% and 58% lower than the final GA&LP solutions. For all CONOPT2 runs, rea-

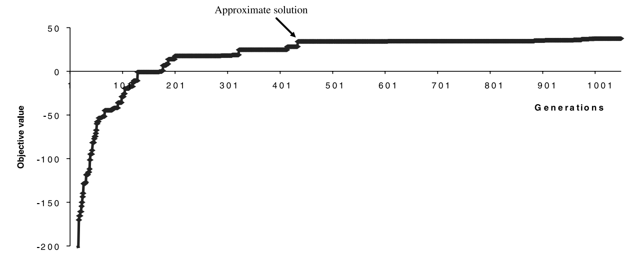

Fig. 8. Objective value vs. generations, model_6, with 5 nonlinear reservoirs, 48 time periods.

sonably relaxed optimality and feasibility tolerances were tested, but no better solution was found than the values presented in Table 2. CONOPT2 is unable to find a feasible solution to model_6 (the largest and most complex one) from the default start. However, as shown in the last column of Table 2, if the final GA&LP solutions to models 2, 4, and 6 are used as a starting point for CONOPT2, CONOPT2 finds significantly improved solutions. Some improvement is almost certain, as CONOPT2 is not restricted to the discrete values for H used by the GA. In this way, the two approaches combine to partially overcome the slow convergence of GA&LP, and the tendency of CONOPT2 to stop at the nearest local solution.

Model size measures and the number of GA iterations to find the approximate and final solutions are shown in Table 3, where all values are ratios using model_1 as the base. GA iterations increase roughly linearly with the number of GA variables.

6. Application to a river basin planning model

The GA&LP approach has also been applied to a very large and complex multiperiod model of long-term irrigation planning and water allocation in the Syr Darya River basin in Central Asia. A more extensive description of the model is in [2]. As illustrated in Fig. 9, the long-term modeling framework is composed of an *inter-year control program* (IYCP) and a series of *yearly models* (YM). The modeling framework is designed to connect intra-year short-term decisions dynamically with inter-year long-term decisions. The IYCP sends 'proposals' to each YM, and the value of the long-term objective resulting from all YMs is fed back to the IYCP,where the GA generates better proposals.

The structure of the yearly model is summarized as follows:

- Time period structure: 12 monthly periods.
- Objective function: maximizing total water use benefit, including net irrigation profit, hydropower generation profit, and benefit from ecological water use.

Table 2
Results from the GA&LP approach and the comparison with CONOPT2

Models	GA&LP					CONOPT2	
	Objective value of first generation[a]	Approximate		Final		Starting from estimated initial	Starting from GA&LP[c]
		Number of generations	Objective value[b]	Number of generations	Objective value[b]		
Model_1	−94.392	17	1.976	60	2.289	2.381	2.381
Model_2	−165.976	201	3.315	330	4.205	2. 577[d]	4.423
Model_3	−4.975	95	4.314	200	4.459	4.529	4.529
Model_4	−78.74	308	8.765	760	10.292	4.513[d]	11.642
Model_5	−203.462	170	9.825	610	14.174	14.994	14.995
Model_6	−423.345	445	40.215	1030	48.216	Failed[d]	52.225

[a] Including penalty on positive deviation variables.
[b] Average value of three runs.
[c] Taking the solution from GA&LP as initial values.
[d] This value was found in multiple CONOPT2 runs, and in a run using the MINOS5 GAMS solver.

Table 3
Ratios of size measures and GA iterations, using Model_1 as a base

Models	Number of equations[a]	Number of variables[a]	Number of non-constant, non-zero Jacobian elements[a]	Number of non-zero Jacobian elements[a]	Number of variables in GA	Number of generations for approximate solution	Number of generations for final solution
Model_1	1	1	1	1	1	1	1
Model_2	1.99	2.00	2.01	2.02	2.00	11.82	5.50
Model_3	1.37	1.10	1.16	2.00	2.00	5.59	3.33
Model_4	2.73	2.20	2.34	4.03	4.00	18.00	12.67
Model_5	2.48	1.28	1.65	5.00	5.00	10.00	10.17
Model_6	12.38	6.36	7.41	25.34	20.00	26.18	17.17

[a] From model statistics in GAMS output file.

- Constraints: hydrologic, agronomic, economic and institutional relationships.
- Major state variables: reservoir storage, groundwater table, water salinity, soil moisture content, and soil salinity.
- Major decision variables, including:
  - Reservoir & aquifer operations.
  - Water withdrawals to demand sites.
  - On-farm water allocation and source blending to specific crop fields.
  - Irrigated crop area under specific climatic conditions.

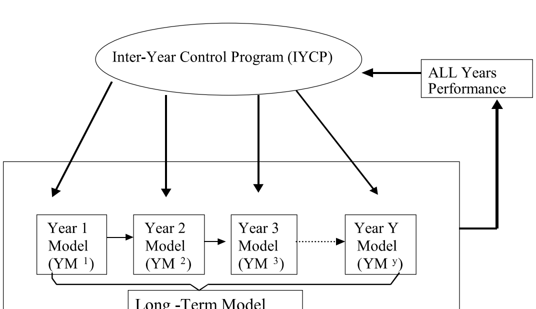

Fig. 9. Structure of the long-term model-overview.

The YM constraints include a large number of hydrologic, agronomic, economic and institutional relationships, many of which are nonlinear. Nonlinearities include product terms in salt conservation equations, where salt concentrations multiply flows, and flow conservation, where crop area multiplies water consumption (crop evapotranspiration) and infrastructure variables (irrigation or drainage efficiency) multiply flows. For example, water distribution efficiency multiplies flow diversion (flow arriving at an irrigated field), irrigation efficiency multiplies flow arriving at an irrigated field (water available for crop), and drainage fraction multiplies irrigated area (drained area). In its full nonlinear form, a typical instance of the YM, with 12 monthly time periods, has of the order of 10,000 constraints and variables. Clearly a 30-yr model, with each YM this size, would be very difficult to solve even if it were linear, so several simplifications are made. The Syr Darya basin has one of the most advanced water storage and distribution systems in the world, so the end-of-year storage levels in key reservoirs have a major effect on future time periods. If these storage levels (WSU in Fig. 10) are chosen as complicating variables, and varied by the IYCP (which uses the long-term objective function), their effect on both current and future

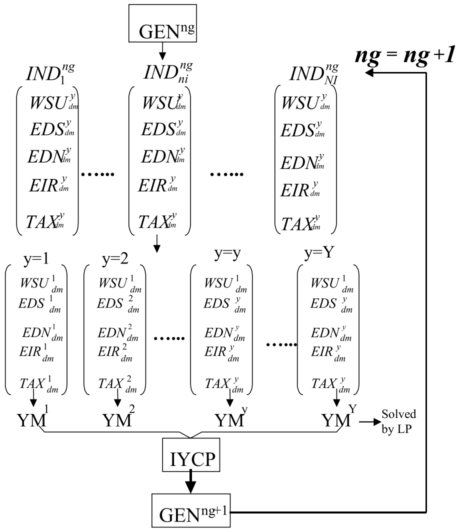

Fig. 10. GA within the IYCP.

years is accounted for. This allows the YMs to be solved sequentially, while retaining the ability of the overall procedure to include the future implications of current decisions. At each agricultural demand site, irrigated crop acreage (IA in Fig. 10), and the efficiencies of water distribution, irrigation, and drainage systems (EDS, EIR, EDN, infrastructure variables) also have a strong long-term effect. If these are selected as complicating variables, many nonlinear YM equations become linear. The YM still has some nonlinear equations, mostly salt balances at river reaches, reservoirs, and aquifers, and at demand sites. These are dealt with by separating the YM into linear flow and salinity submodels, which are solved sequentially in a iterative process. Tax rates on salt discharge (TAX) are also taken as complicating variables, because they too have long-term implications. The result is a sequence of 30- yearly YMs and a GA at the IYCP level, which varies the complicating variables in an effort to minimize a long-term (30-yr) objective.

The long-term objective is a weighted linear combination of seven sustainability criteria, formulated to measure risk minimization in water supply, environmental preservation, equity in water allocation, and economic efficiency in water infrastructure development. The values of these criteria are calculated based on the results from all yearly models.

To search a long-term optimal solution in such a modeling framework, an algorithm dealing with year-by-year dynamic relationships is required. Dynamic programming (DP) is a popular algorithm for problems like the one described above. However, the modeling framework includes a large number of time periods (the yearly model has 12 time periods, and the long-term time horizon is 30 yr for the case study area), and complex spatial dimensions (the basin is comprised of 9 major tributaries, 11 reservoirs, 5 hydropower stations, 6 groundwater sources, and 6 aggregated irrigation distribution systems). Also a large number of decision variables are involved in the model. Therefore, application of a standard DP algorithm is rendered impractical by the "curse of dimensionality", which encourages us to use the combined GA and linear programming (GA&LP) approach.

Fig. 10 shows a diagram of the implementation of the GA&LP program. The GA begins by randomly creating a prescribed number of "individuals" (IND_{ni}^{ng}, $ni = 1, 2, \ldots, NI$), each corresponding to a different set of values for the inter-year control variables.

$$\mathrm{IND}_{ni}^{ng} = (\mathrm{WSU}^{y}, \mathrm{EDS}_{dm}^{y}, \mathrm{EDN}_{dm}^{y}, \mathrm{EIR}_{dm,fd}^{y}, \mathrm{IA}_{dm,fd}^{y}, \mathrm{TAX}_{dm}^{y}). \tag{12}$$

Each generation (GEN^{ng}, $ng = 1, 2, \ldots, NG$) is a group of individuals:

$$\mathrm{GEN}^{ng} = \{\mathrm{IND}_{1}^{ng}, \mathrm{IND}_{2}^{ng}, \mathrm{IND}_{3}^{ng}, \ldots, \mathrm{IND}_{NI}^{ng}\}. \tag{13}$$

The "individual" to be evaluated is input into the yearly LP models YM^{y} ($y = 1, 2, \ldots, Y$), which are solved sequentially. The inter-year control variables are limited by simple bounds and a few additional constraints. For example, all efficiencies are assumed to be nondecreasing over time, and the sum of crop areas over all crop fields must not exceed the total available area in the demand site. These constraints are imposed by examining each individual after its creation, and modifying variable values which do not satisfy these constraints. Any efficiency that is smaller than its value in the proceeding period is set equal to that value, and areas whose sum exceeds the available area are scaled down so the sum equals that area.

The GA&LP approach searches for better solutions at two levels: the best solution among all the individuals within one generation, and improved solutions through several generations. We first consider the search for the best solution within one generation. We choose two individuals which represent the "best" and the "worst" one in one generation, respectively, according to the total objective. These individuals are compared in

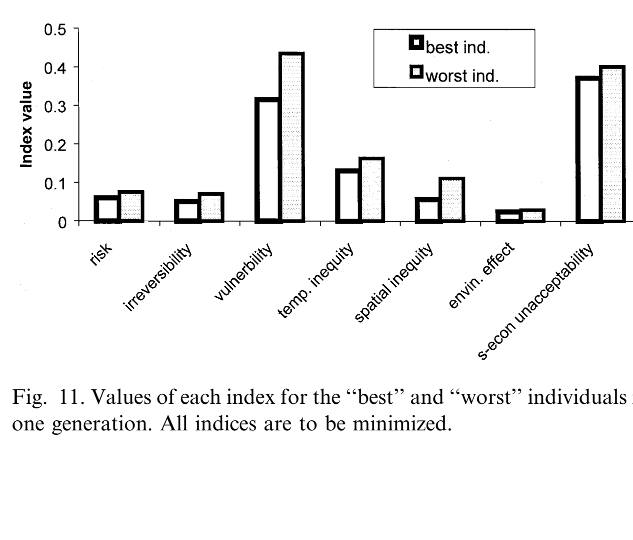

Fig. 11. Values of each index for the "best" and "worst" individuals in one generation. All indices are to be minimized.

Fig. 11, showing the seven indices making up the long-term objective. An index is selected for each of the criteria in the objective to be minimized, including risk, irreversibility, and vulnerability in water supply, spatial and temporal inequity in water allocation, environmental effect, and socio-economic unacceptability (see definitions in [2]). Risk and socio-economic unacceptability are redefined as the reciprocals of reliability and socio-economic acceptability, respectively, so that all can be minimized. As can be seen, the best individual is better than the worst with respect to all criteria in the selected generations. However, this is not necessary for other generations because of the tradeoffs between the criteria, which is shown in Fig. 13.

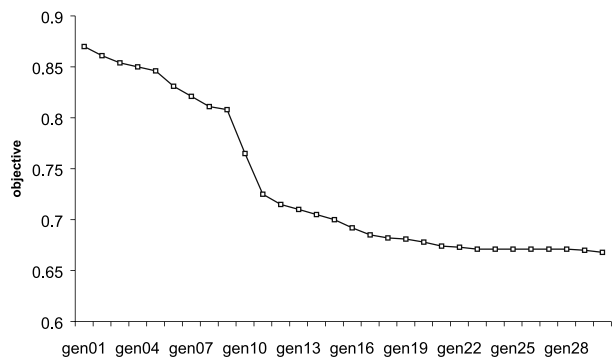

Fig. 12. Long-term objective value in successive generations.

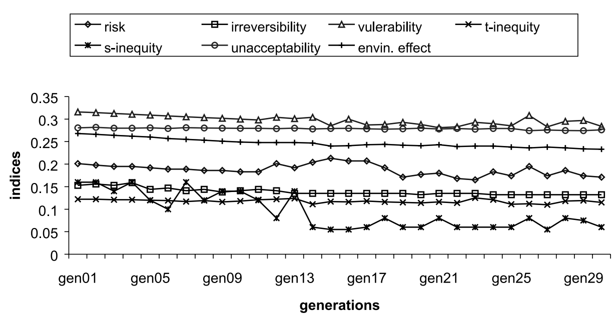

Fig. 13. Values of each criterion in successive generations.

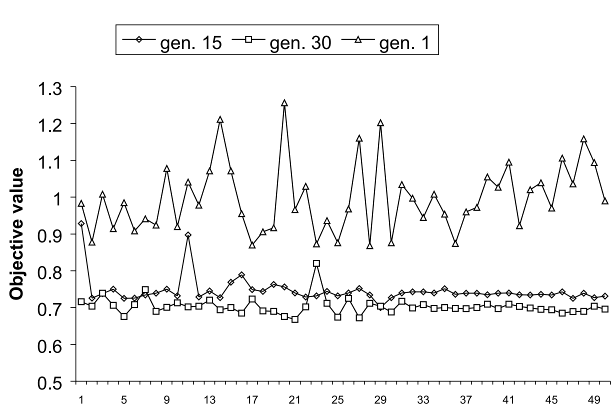

Fig. 14. Comparison of the objective values of generations 1, 15, and 30.

Fig. 12 plots the values of the long-term objective against GA iteration numbers. As with the reservoir model, progress slows as the iterations progress. Fig. 13 shows the values of all indices associated with the best solution in each of the 30 generations. Each has a tendency to improve. The fluctuations of these items from generation to generation show the tradeoff among these items.

The total long-term objective values for all individuals of generations 1, 15, 30 are plotted in Fig. 14. The average solution value improves with iteration number, and there is less variance among the solutions in later generations.

7. Discussion and conclusion

This GA&LP approach combines a modeling strategy and a choice of algorithms. The modeling strategy requires careful selection of the complicating variables. In the reservoir model, these are chosen to render the model linear in the remaining variables. In the long-term planning situation, the complicating variables are chosen to yield both linearity and the ability to solve each yearly model sequentially. If these choices are made properly, computational results show that a fairly standard GA is capable of finding high quality solutions to both classes of models in reasonable computing times. For the larger and more complex reservoir models, these solutions are superior to those found by the CONOPT2 NLP solver from reasonably good starting points. However, CONOPT2 is able to improve the final GA solution, so for problems within its size range, it provides an "end-game" strategy for the GA&LP approach.

As in most GA applications, initial rapid convergence is followed by a long, slow but steady improvement of the objective. However, solutions within 5% of the final value are found early in the process, and computation time does not increase rapidly with problem size for the numbers of complicating variables considered here. The methodology parallelizes very well, since the problems of finding the fitness of each individual are independent of one another and may be solved simultaneously. A parallel implementation will almost surely result in speedups which are proportional to the number of processors [13].

However, with the models considered, the proposed approach converges to a near-global solution very slowly, or perhaps even misses the global solution, and converges to a local solution. The convergence speed depends on setting appropriate parameter values in the GA, as well as the clustering threshold, which must be adjusted by trial-and-error. The probability of crossover and the probability of mutation are the two most sensitive parameters in the GA. Other factors, such as the method of fitness calculation, and the selection of seed for random number generation, also affect the solution. Moreover, the global solution may not be guaranteed due to the stochastic nature of GAs, especially in selecting probability of crossover and probability of mutation.

It is clear that other search methodologies, for example Tabu Search, GRASP, or Simulated Annealing, may be substituted for the GA. Many classes of models are simplified when a relatively small set of complicating variables are fixed, and we believe that the ideas described here are capable of finding good approximate solutions to these models.

Acknowledgements

The authors are grateful to Professor Daniel P. Loucks and another anonymous reviewer for their very helpful comments and suggestions.

References

[1] Brooke A, Kendrick D, Meeraus D. GAMS: a user's guide. Scientific Press; 1988.

[2] Cai X. Modeling framework for sustainable water resources management. Unpublished Ph.D. dissertation, University of Texas at Austin; 1999.

[3] Cai X, McKinney DC, Lasdon LS, Watkins D. Solving large nonconvex water resources management models using generalized benders decomposition. Oper Res 2001 [in press].

[4] Drud AS. CONOPT – a large scale GRG code. ORSA J Comput 1994;6:207–16.

[5] Geoffrion AM. Generalized benders decomposition. J Optim Theory Appl 1972;10:237–48.

[6] Glover F, Laguna M. Tabu search. Dordrecht, MA: Kluwer Academic Publishers; 1997.

[7] Goldberg DE. Genetic algorithms. Reading, MA: Addison-Wesley; 1989.

[8] Halhal DG, Walters A, Ouazar D, Savic DA. Water network rehabilitation with structured messy genetic algorithm. J Water Resour Plng Mgmt ASCE 1997;123(3):137–45.

[9] Huang C, Mayer AS. Pump-and-treat optimization using well locations and pumping rates as decision variables. Water Resour Res 1997;33(5):1001–12.

[10] Loucks DP, Stedinger JR, Douglas H. Water resource systems planning and analysis. Englewood Cliffs, NJ: Prentice-Hall; 1981.

[11] McKinney DC, Lin MD. Genetic algorithm solution of groundwater management models. Water Resour Res 1994;30:1897–906.

[12] Michalewicz Z. Genetic algorithms+data structure = evolution programs. New York: Springer; 1992.

[13] Migdalas AP, Pardalos M, Storoy S, editors. Parallel computing in optimization. Dordrecht, MA: Kluwer Academic Publishers; 1997.

[14] Morshed M, Kaluarachchi J. Enhancements to genetic algorithm for optimal ground-water management. J Hydrol Eng 2000;5(1):67–73.

[15] Murtagh BA, Saunders MA. A projected lagrangian algorithm and its implementation for sparse nonlinear constraints. Math Prog Study 1982;16:84–117.

[16] Oliveira R, Loucks DP. Operating rules for multireservoir systems. Water Resour Res 1997;33(4):839–52.

[17] Reeves CR. Genetic algorithms for the operations researcher. Inform J Comput 1997;9(1):231–50.

[18] Savic DA, Walters GA. Genetic algorithms for least-cost design of water ditribution networks. J Water Resour Plng Mgmt ASCE 1998;123(1):61–77.