SOLVING LARGE NONCONVEX WATER RESOURCES MANAGEMENT MODELS USING GENERALIZED BENDERS DECOMPOSITION

XIMING CAI

International Food Policy Research Institute, 2033 K. Washington D.C., 20006, x.cai@cgiar.org

DAENE C. McKINNEY

Department of Civil Engineering, College of Engineering, The University of Texas at Austin, Austin, Texas 78712, daene@aol.com

LEON S. LASDON

Department of Management Science and Information Systems, Graduate School of Business, The University of Texas at Austin, Austin, Texas 78712, lasdon@mail.utexas.edu

DAVID W. WATKINS, Jr.

Department of Civil and Environmental Engineering, Michigan Technological University, Houghton, Michigan 49931, dwatkins@mtu.edu

(Received June 1998; revision received June 1999; accepted September 1999)

Nonconvex nonlinear programming (NLP) problems arise frequently in water resources management, e.g., reservoir operations, groundwater remediation, and integrated water quantity and quality management. Such problems are usually large and sparse. Existing software for global optimization cannot cope with problems of this size, while current local sparse NLP solvers, e.g., MINOS (Murtagh and Saunders 1987), or CONOPT (Drud 1994) cannot guarantee a global solution. In this paper, we apply the Generalized Benders Decomposition (GBD) algorithm to two large nonconvex water resources models involving reservoir operations and water allocation in a river basin, using an approximation to the GBD cuts proposed by Floudas et al. (1989) and Floudas (1995). To ensure feasibility of the GBD subproblem, we relax its constraints by introducing elastic slack variables, penalizing these slacks in the objective function. This approach leads to solutions with excellent objective values in run times much less than the GAMS NLP solvers MINOS5 and CONOPT2, if the complicating variables are carefully selected. Using these solutions as initial points for MINOS5 or CONOPT2 often leads to further improvements.

For many water resources management models, nonlinear programming (NLP) offers a general mathematical formulation handling a nonseparable objective function and nonlinear constraints. However, in many cases, the NLP problems are nonconvex, but local solvers are applied. Methods for obtaining global solutions to nonconvex mathematical programming problems have not appeared frequently in the water resources literature, even though these problems are often unavoidable in water resources management modeling. Yeh (1985) reviewed some calculus-based nonlinear programming algorithms in reservoir management, including the quasi-Newton method, the gradient projection method, the reduced gradient method, and the Lagrangian dual procedure. He found these NLP algorithms gained practical importance only if the nonlinear problems could be decomposed into separable subproblems. In particular, when the NLP reservoir management model is large, solution becomes difficult due to the trap of infeasibility, or local solutions and slow convergence speed.

In recent years, genetic algorithms (GAs) have been proposed as a promising method to solve nonconvex NLP problems in water resources systems planning (McKinney and Lin 1995, Ritzel et al. 1994). Karatzas and Pinder (1996) used outer approximation (OA) to solve groundwater management problems with concave objective functions and nonconvex feasible regions. Watkins and McKinney (1997) used the Generalized Benders Decomposition (GBD) and OA to solve a mixed-integer nonlinear (MINLP) water resources optimization model with fixed costs. Previous studies have also shown the stochastic extension of GBD to solve stochastic models in water resources. Pereira and Pinto (1985) derived a stochastic extension of GBD and applied the algorithm to solve a multistage optimization model of a multireservoir hydroelectric system. The decomposition scheme was used to accommodate treelike structures of the stochasticity of inflow at each stage, and the expected values of simplex multipliers were used to form Benders cuts. Gorenstin et al. (1993) used a similar approach to study power system expansion planning under uncertainty.

In this paper, we apply the GBD technique to two deterministic, large-scale, nonconvex NLP problems in water resources management, in an approximate way suggested in Floudas et al. (1989) and Floudas (1995). To demonstrate

Subject classifications: Natural Resources: water resources management modeling. Nonlinear Programming: nonconvexity in large nonlinear models. Algorithm: Generalized Benders Decomposition.
Area of review: Environment, Energy and Natural Resources.

the modified approach, we solve two large NLP models in water resources management. One is a reservoir system operation model, considering both water supply and power generation. The other is a river basin water allocation model, including salt control. In §2, we describe these two classes of models. In §3 and §4, we apply the Generalized Benders Decomposition (GBD) to our models, relaxing some of the subproblem constraints to insure feasibility. We show that the true GBD cuts are partially separable piecewise-linear functions, and the GBD relaxed master program is a reverse convex program. Our computations use linear approximations to the true GBD cuts, as suggested in Floudas et al. (1989) and Floudas (1995). Our GAMS implementation is described in §5. The results of §6 show that, for the problem instances we consider, GBD using these cuts leads to objective values only slightly worse than those achieved by MINOS5 and CONOPT2 (Drud 1994), in much less computing time.

1. TWO WATER RESOURCES MODELS

In this section we formulate two typical water resources management models: one a reservoir operation model, and the other a river basin water allocation and salinity control model.

1.1. Definition of Model Network, Data, and Variables

1.1.1. Sets and Indices

T	time periods (months)
dem	demand site nodes
riv	river nodes
rev	reservoir nodes
gw	groundwater nodes
cal	diversion canal nodes
drn	drainage collector nodes
pwst	hydropower station nodes
$n, n_1, n_2 \in N$	general network nodes
(n_1, n_2)	network arcs, all directed
gdlink	$\{(n_1, n_2) \mid n_1 \in gw,\ n_2 \in dem$, arcs from groundwater to demand sites$\}$
before(n)	$\{n_1 \in N \mid (n_1, n)$ is a network arc$\}$
after(n)	$\{n_2 \in N \mid (n, n_2)$ is a network arc$\}$

1.1.2. Data

$wdem(n, t)$	water demand for $n \in dem$ and $t \in T$
$pdem(t)$	power demand for $t \in T$
$ET(n, t)$	crop evapotranspiration rate for $n \in dem$ and $t \in T$
$pcap(n)$	pumping capacity at groundwater node $n \in gw$
$\alpha(n), \beta(n)$	constants in reservoir head-volume and head-surface area equations, $n \in rev$
$k(n), tw(n)$	constants in hydropower generation constraints
$ts(n)$	target salt concentration at node n
$S_0(n), C_0(n)$	initial storage volume and salt concentration at storage nodes $n \in rev$ or $n \in gw$
$Q(n_1, n_2, t),\ C(n_1, t)$	exogenous flows and salt concentrations for nodes n_1 corresponding to exogenous supplies

1.1.3. Decision Variables

$Q(n_1, n_2, t)$	flow on arc (n_1, n_2) in period $t \in T$. At upstream nodes, Q is the source flow (given data)
$C(n, t)$	salt concentration at node $n \in N$ in period $t \in T$. At upstream nodes, C is the source salt concentration (given data)
$RI(n, t)$	ratio of water delivered to that demanded at node $n \in dem$ in period $t \in T$
$S(n, t)$	water stored at reservoir or aquifer node $n \in res$ or gw at the start of period $t \in T$
$A(n, t)$	surface area of reservoir $n \in res$ at the start of period $t \in T$
$G(n, n_1, t)$	water pumped from acquifer node $n \in gw$ to node $n_1 \in dem$ in period $t \in T$
$P(n, t)$	energy generated at reservoir node $n \in rev$ in period $t \in T$
$H(n, T)$	head in reservoir $n \in rev$ in period $t \in T$

1.2. Equations

1.2.1. Objective Function. Water management problems, particularly at the river basin scale, are often complex, multiobjective, and site-specific. As a result, there is no general model formulation for water resources management. To illustrate the types of water resource problems that could benefit from the proposed GBD solution approach, we consider four specific objectives and three specific sets of constraints. Other applications may require other objectives, constraints, and solution procedures. The objectives we consider are the following:

• Maximize the ratio of water delivered to that demanded at demand sites:

$$Z_1 = \sum_{n \in dem} \sum_{t \in T} RI(n, t), \tag{1}$$

where

$$RI(n, t) = \left[\sum_{n_1 \in before(n)} Q(n_1, n, t) + \sum_{(n_1, n) \in gdlink} G(n_1, n, t) \right] \Big/ wdem(n, t), \tag{2}$$

and

$$0 \leqslant RI(n, t) \leqslant 1 \quad \forall n \in dem,\ \forall t \in T,$$

which ensures that no demand is over-satisfied.

• Maximize the smallest water deficits among all demand sites and time periods:

$$Z_2 = \sum_{t\in T} MRIT(t) + \sum_{n\in dem} MRID(n), \quad (3)$$

where

$$MRIT(t) \leqslant RI(n,t) \quad \forall n \in dem, \forall t \in T, \quad (4a)$$
$$MRID(n) \leqslant RI(n,t) \quad \forall n \in dem, \forall t \in T. \quad (4b)$$

This objective is designed to distribute water shortages ($RI < 1$) both over all time periods and among all demand sites.

• Maximize the sum of the total amount of hydropower generated in the basin and the minimum hydropower generation over all periods:

$$Z_3 = \sum_{t\in T}\sum_{n\in pwst} P(n,t) + \omega \cdot RPMIN, \quad (5)$$

where

$$RPMIN \leqslant \sum_{n\in pwst} P(n,t)/pdem(t) \quad \forall t \in T, \quad (6)$$

and ω is a weighting/scaling coefficient. This objective is formulated to maximize total energy generation and to insure that any shortfall in energy supply is distributed in an equitable manner.

• Minimize salt concentration in the system:

$$Z_4 = -\sum_{n\in N}\sum_{t\in T} \frac{C(n,t)}{ts(n)}, \quad (7)$$

where the sum ranges over all nodes representing tributaries, reservoirs, main river nodes, and aquifers, respectively, and Z_4 is maximized.

1.2.2. Constraints. The model constraints comprise water balance constraints, hydroelectric generation constraints, and salinity balance constraints, as follows:

Water Balance Constraints

• Water balance at river, diversion canal, or drainage collector nodes:

$$\sum_{n_1\in before(n)} Q(n_1,n,t) = \sum_{n_1\in after(n)} Q(n,n_1,t) \quad \forall n \in riv, cal, \text{or } drn, \forall t \in T. \quad (8)$$

• Water balance at reservoir nodes:

$$S(n,t+1) = S(n,t) + \sum_{n_1\in before(n)} Q(n_1,n,t) - \sum_{n_1\in after(n)} Q(n,n_1,t) \quad \forall n\in rev, \forall t\in T. \quad (9)$$

• Water balance at groundwater sources:

$$S(n,t+1) = S(n,t) + \sum_{n_1\in before(n)} Q(n_1,n,t) - \sum_{n_1\in after(n)} Q(n,n_1,t) - \sum_{(n,n_1)\in gdlink} G(n,n_1,t) \quad \forall n\in gw, \forall t\in T. \quad (10)$$

• Pumping capacity at groundwater sources:

$$\sum_{(n,n_1)\in gdlink} G(n,n_1,t) \leqslant pcap(n) \quad \forall n\in gw, \forall t\in T. \quad (11)$$

• Water balance at demand nodes:

$$\left[\sum_{n_1\in before(n)} Q(n_1,n,t) + \sum_{(n_1,n)\in gdlink} G(n_1,n,t)\right] * [1 - ET(n,t)] = \sum_{n_1\in after(n)} Q(n,n_1,t) \quad \forall n \in dem, \forall t \in T, \quad (12)$$

where flows out of demand sites represent return flow from demand sites, including surface and subsurface drainage.

Energy Generation Constraints

• Hydroelectric energy (KWH) generation:

$$P(n,t) = k(n) * \left\{\frac{1}{2}[H(n,t) + H(n,t-1)] - tw(n)\right\} * \sum_{n_1\in after(n)} Q(n,n_1,t) \quad (13)$$

and

$$P(n,t) \leqslant PC(n) \quad \forall n \in rev, \forall t \in T,$$

which expresses hydroelectric energy generated as being proportional to the flow through the turbine times the difference between average surface elevation and tail water elevation $tw(n)$. The maximum energy generation in a period cannot exceed the energy generation capacity of a station (PC).

• Reservoir head-volume and head-surface area (A) relationships:

$$S(n,t) = \alpha_1(n) * H(n,t) + \alpha_0(n) \quad \forall n\in rev, \forall t\in T, \quad (14)$$
$$A(n,t) = \beta_1(n) * H(n,t) + \beta_0(n) \quad \forall n\in rev, \forall t\in T. \quad (15)$$

For simplicity, we use linear head-volume and head-surface area relationships in the case study of this paper, since for many reservoirs, linear approximations for these relationships are usually well over the typical operating range. However, these relationships can be nonlinear in general, and the methods suggested in this paper remain applicable.

Salinity Balance Constraints

• Salt balance at river, canal, or drainage nodes:

$$\sum_{n_1\in before(n)} Q(n_1,n,t) * C(n_1,t) = \sum_{n_1\in after(n)} Q(n,n_1,t) * C(n,t) \quad \forall n \in riv, cal, \text{or } drn, \forall t \in T. \quad (16)$$

• Salt balance at reservoir nodes:

$$S(n,t) * C(n,t) + \sum_{n_1\in before(n)} Q(n_1,n,t) * C(n_1,t) = \left[S(n,t+1) + \sum_{n_1\in after(n)} Q(n,n_1,t)\right] * C(n,t) \quad \forall n \in rev, \forall t \in T. \quad (17)$$

• Salt balance at groundwater nodes:

$$S(n,t)*C(n,t)+\sum_{n_1\in before(n)} Q(n_1,n,t)*C(n_1,t) = \left[S(n,t+1)+\sum_{(n,n_1)\in gdlink} G(n,n_1,t+1) + \sum_{n_1\in after(n)} Q(n,n_1,t)\right]*C(n,t) \quad \forall n\in gw,\ \forall t\in T. \tag{18}$$

• Salt balance at demand nodes:

$$\sum_{n_1\in before(n)} Q(n_1,n,t)*C(n_1,t)+\sum_{(n_1,n)\in gdlink} G(n_1,n,t) *C(n_1,t) = \sum_{n_1\in after(n)} Q(n,n_1,t)*C(n,t) \quad \forall n\in dem,\ \forall t\in T. \tag{19}$$

• **Variable bounds.** All variables are nonnegative and some have specific lower and upper bounds to represent some particular physical or policy limits. For example, flows through river nodes have lower bounds for some instream water uses such as flow augmentation and recreation, and upper bounds for flood control; the ratio of water supply to water demand cannot be lower than a specified value in some months; salt concentrations in some locations have upper bounds; and energy generation is constrained by reservoir capacity.

1.3. Models

1.3.1. River Basin Water Allocation and Salinity Control Model. For this model, the objective function is a weighted combination of objectives (1), (3), and (7):

$$\text{Maximize} \quad Z = w_1 Z_1 + w_2 Z_2 + w_4 Z_4, \tag{20}$$

where the nonnegative weights w_i reflect the relative importance of the individual objectives. The constraints are the water and salinity balance constraints (8)–(12) and (15)–(19), omitting the energy generation constraints (13)–(14).

1.3.2. Reservoir Operation Model. For the reservoir operation model, the objective function is a weighted combination of objectives (1), (3), and (5):

$$\text{Maximize} \quad Z = w_1 Z_1 + w_2 Z_2 + w_3 Z_3. \tag{21}$$

The constraints are the water balance constraints (8)–(12) and the power generation constraints (13)–(14), omitting the salinity balance constraints (15)–(19).

2. GENERALIZED BENDERS DECOMPOSITION

The Generalized Benders Decomposition (GBD), developed in Geoffrion (1972), was originally derived for problems of the form:

$$\begin{aligned} \max\quad & f(x,y),\\ \text{s.t.}\quad & g(x,y)\geqslant 0,\\ & x\in X,\ y\in Y, \end{aligned} \tag{22}$$

where g is the vector of coupling constraint functions. The vector y contains complicating variables, in the sense that the problem is assumed to be considerably easier to solve when y is fixed. At iteration r, the procedure involves successive solutions of the subproblem:

$$\begin{aligned} P(y_{r-1}):\quad & \text{maximize } f(x,y_{r-1}),\\ \text{subject to}\quad & g(x,y_{r-1})\geqslant 0,\ x\in X, \end{aligned} \tag{23}$$

whose optimal objective value is $v(y_{r-1})$, and the relaxed master program,

$$\begin{aligned} RMP(r):\quad & \text{maximize } y_0,\\ \text{subject to}\quad & y_0\leqslant L^*(y,u_j),\ j=1,\dots,r,\ y\in Y, \end{aligned}$$

whose solution is y_r. The function L^* is defined as:

$$L^*(y,u_j)=\max_{x\in X}(f(x,y)+u_j^T g(x,y)),$$

and u_j is an optimal multiplier vector for $P(y_j)$.

In the following, we assume that $P(y)$ has an optimal solution and an optimal multiplier vector u for each $y\in Y$. This is guaranteed for our problems by relaxing some constraints of $P(y)$ by introducing "elastic" slack variables where necessary. These slacks enable one to avoid initial infeasibilities from sources such as unmet water demand, and they will usually be zero in a final solution. Under these assumptions, the steps of the GBD algorithm are:

1. Initialize: $r=0$, $y_0\in Y=$ user supplied initial values for y, *lbd* (lower bound) $=-\infty$, *ubd* (upper bound) $=+\infty$, $\varepsilon=$ convergence tolerance.
2. Solve $P(y_r)$, obtaining an optimal solution x_r and an optimal multiplier vector u_r. Update *lbd* by setting $lbd = v(y_r)$.
3. Generate a closed form expression for $L^*(y,u_r)$ and add the constraint $y_0\leqslant L^*(y,u_r)$ to $RMP(r-1)$, creating $RMP(r)$.
4. Solve $RMP(r)$. The optimal solution is y_0^r. Set $ubd=y_0^r$.
5. If $(ubd-lbd)/abs(lbd)<\varepsilon$, stop. (x_r,y_r) is an *ε-optimal* solution to the original problem.
6. Replace r by $r+1$ and go to step 2.

Assumptions which guarantee finite convergence of the procedure are:

1. f and g are concave on x for each fixed $y\in Y$.
2. X is nonempty and convex.
3. It must be possible to solve each $RMP(r)$ globally.

In order for GBD to be computationally attractive, the following additional assumption must be satisfied:

4. It must be possible to obtain a closed form expression (or one that can be evaluated rapidly) for the GBD cut functions $L^*(y,u_r)$ for each fixed u_r. This is "property P" as defined in Geoffrion (1972).

In our problems, as will be shown below, assumptions (1) and (2) are satisfied because $g(x,y)$ is bilinear, X and Y are polyhedral (that is, defined by linear constraints), and f is linear. Some of the coupling constraints (3.1) are equalities, but these define a convex region in x space for fixed y, because g is bilinear. We show in the next section that assumption (4) is also satisfied, but (3) is not, because each RMP is a reverse convex program (Horst 1990), i.e., it is of the form $g(x) \geqslant 0$, where $g(x)$ is a convex function.

3. APPLYING GBD TO WATER RESOURCES MANAGEMENT MODELS

All the constraints described in Section 1.2 are linear except for power generation (13) and salinity balance (15)–(19), which are bilinear, i.e., all nonlinear terms are products of two variables. In the salinity balance constraints, we find products of flows $Q(n, n_1, t)$ and storages $S(n, t)$ with salt concentrations $C(n, t)$, while the hydropower generation relations involve products of reservoir outflows $Q(n, n_1, t)$ and reservoir heads $H(n, t)$. Hence if one member of the set of variables $((Q, S), C)$ is fixed, the salinity balance constraints become linear, and if one member of the set $((Q, S), H)$ is fixed, the power generation constraints are linear. This structure motivates the application of GBD, and the selection of either the (Q, S) or C variables as complicating variables y in the salinity control model, and either (Q, S) or H as complicating variables in the reservoir operation model. As pointed out in Floudas et al. (1989), these selections apply to any problem with bilinear objective and/or constraint functions, and Theorem 1 below applies to any such problem.

3.1. Salinity Control Model

It is natural to choose the salt concentrations C as the complicating variables y, because there are fewer of them. Surprisingly, this is not the best choice. Choosing C as y, GBD using the approximate cuts described by Floudas et al. (1989) and Floudas (1995) did not converge. However, as shown in Section 6, choosing Q as y, these approximate cuts lead to solutions of quality comparable to those achieved by the GAMS solvers, MINOS5 or CONOPT2, using the same starting points in 15 or fewer major iterations.

Let x be the vector of salt concentration variables $C(n, t)$, and let y be the vector of all other salinity model variables (Q, RI, S, G). Then most of the salinity model constraints, i.e., the water balance constraints (8)–(12), and the constraints defining the objective (1)–(4) involve only y, and are linear. Letting Y represent the polyhedron defined by these constraints, they can be written as:

$$y \in Y. \tag{24}$$

The salt balance constraints (15)–(19) involve both x and y, so these are the coupling constraints $g(x, y)$ of the salinity control model. In these constraints, each concentration, x_j, multiplies one or more flow or storage variables (components of y). Hence, we can write the vector of salt balance constraints as

$$g(x, y) = \sum_j x_j a_j(sy_j) - b_i = 0, \tag{25}$$

Table 1. Nonzero coefficients of $C(n, t)$ (eqs. (16) and (17)).

Row	Coefficient
(n, t)	$-\left(\sum_{n_1 \in after(n)} (Q(n, n_1, t) + S(n, t))\right)$
$(n_1, t), n_1 \in after(n)$	$Q(n, n_1, t)$
$(n, t+1)$	$S(n, t)$

where b is a vector of constants, $a_j(sy_j)$ is the column of coefficients associated with x_j in the subproblem $P(y)$, and sy_j is the subvector of y upon which a_j depends. If $x_j \equiv C(n, t)$, a_j has nonzero entries as shown in Table 1.

The variable $S(n, t)$ does not appear if n is not a storage node. No other column a_j involves these variables, because only $C(n, t)$ multiplies flows out of node (n, t) and $S(n, t)$. Hence $a_j(sy_j)$ is a linear function of sy_j, and the subvectors sy_j partition y.

It is desirable to ensure that the subproblem constraints, (25) and $x \in X$, are feasible for all fixed $y \in Y$, in order to avoid having to find an extreme ray of the GBD subproblem when they are infeasible. This is typically done by introducing two nonnegative vectors of "elastic" or "deviation" variables dp and dn, rewriting (25) as

$$\sum_j x_j a_j(sy_j) - b = dp - dn, \quad dp \geqslant 0,\ dn \geqslant 0. \tag{26}$$

Then the objective becomes

$$\max \quad Z = c_1^T x + c_2^T y - M \sum_i (dp_i + dn_i), \tag{27}$$

where M is a sufficiently large positive constant. This form for the penalty term is exact. Thus, if the problem is feasible, there is a positive threshold value for M (the maximum absolute optimal multiplier for (25) such that any value above that threshold yields optimal solutions with all elastic variables equal to zero and the remaining variables optimal for the original problem. To complete the model statement, the only constraints involving the salt concentrations C alone are simple upper and lower bounds, written as

$$x \in X = \{x \mid lx \leqslant x \leqslant ux\}. \tag{28}$$

Applying the GBD algorithm as stated in Section 3 to this form of the model, the subproblem $P(y)$ is to maximize (27) over x, dp, and dn for fixed y subject to (26) and (28). $P(y)$ is a linear program. The Lagrangian function for $P(y)$ is

$$L(x, y, dp, dn, u) = c_1^T x + c_2^T y - M \sum_i (dp_i + dn_i) - u^T \left(\sum x_j a_j(sy_j) - b - dp + dn \right), \tag{29}$$

where u is a vector of Lagrange multipliers for (26), introduced with a minus sign to give them the same signs as the simplex multipliers of $P(y)$. The GBD cut function L^* is

$$L^*(y, u) = \max_{x \in X,\, dp \geqslant 0,\, dn \geqslant 0} L(x, y, dp, dn, u). \tag{30}$$

Collecting terms in y, x, dp, and dn yields:

$$L = u^T b + c_2^T y + \sum_i [dp_i(u_i - M) + dn_i(-u_i - M)] + \sum x_j (c_{1j} - u^T a_j(sy_j)). \quad (31)$$

The coefficient of x_j above is the reduced cost or dual slack corresponding to x_j, defined as

$$ds_j = (sy_j, u) = c_{1j} - u^T a_j(sy_j).$$

Since L is linear in x, dp, and dn for fixed y and u, the max in (30) may be applied separately to each term of (31), so

$$L^*(y, u) = u^T b + c_2^T y + \sum_i \Big[\max_{dp_i \geqslant 0} dp_i(u_i - M) + \max_{dn_i \geqslant 0} dn_i(-u_i - M)\Big] + \sum_j \max_{lx_j \leqslant x_j \leqslant ux_j} x_j ds_j(sy_j, u). \quad (32)$$

By complementary slackness, the maxima over dp and dn in (32) are zero. For any fixed y and u, the maxima over each x_j occurs either at lx_j or ux_j so

$$L^*(y, u) = u^T b + c_2^T y + \sum_j \max\big[lx_j ds(sy_j, u), ux_j ds(sy_j, u)\big]. \quad (33)$$

Thus we characterize L^* as follows:

THEOREM 1. *$L^*(y, u)$ is a convex piecewise linear function of y, and is separable in the subvectors sy_j for any fixed u.*

PROOF. In (33), the max of the pair of linear functions is convex and piecewise linear. Denoting this maxima by $pl_j(sy_j, u)$ we have

$$L^*(y, u) = u^T b + c_2^T y + \sum_j pl_j(sy_j, u). \quad (34)$$

Since L^* is a sum of convex functions, it is convex. Since the sy_j partition y, L^* is separable in these subvectors. □

Our previous discussion of the column a_j showed that sy_j has dimension equal to the number of successors of the node associated with x_j. This dimension is almost always small (e.g., <10), independent of network size. Hence the piecewise linear functions pl_j in (34) are easy to evaluate.

The master program $RMP(r)$ for the water allocation model (or any problem with bilinear nonlinearities), is a nonconvex nonlinear program, known as a "reverse convex" program (Horst 1990). As such, it may have distinct local optima. Its constraints include the inequalities

$$L^*(y, u_j) - y_0 \geqslant 0, \quad j = 1, \ldots, r.$$

Since L^* is convex but, in general, not linear, $L^* - y_0$ is a convex nonlinear function of (y, y_0). The inequalities in $RMP(r)$ thus have the "wrong" direction to define a convex set. Each inequality defines the complement of a convex set. There is no known polynomially-bounded algorithm for solving the reverse convex $RMP(r)$ globally, and GBD cannot guarantee a global solution of the original problem without such a guarantee for $RMP(r)$.

Assume that each upper bound in ux is greater than the corresponding component of lx. Then the maxima in (33) is attained at lx_j if ds_j is nonpositive, and at ux_j if ds_j is nonnegative. By restricting the signs of each ds_j, we force the maxima to occur at a particular bound, and these restrictions define polyhedra over which L^* is linear. Let xe^k be the kth extreme point of X and define

$$P_k(u) = \big\{y \mid ds_j(sy_j, u) \leqslant 0 \text{ if } xe_j^k = lx_j,\ ds_j(sy_j, u) \geqslant 0 \text{ if } xe_j^k = ux_j\big\}. \quad (35)$$

Then, for all y in $P_k(u)$, L^* is the linear function,

$$L_k^*(y, u) = u^T b + c_2^T y + \sum_j x_j^k ds_j(sy_j, u). \quad (36)$$

If x has nx components, there are at most 2^{nx} linear pieces L_k^*. In Visweswaren and Floudas (1996), a branch and bound algorithm for finding a global solution to $RMP(r)$ is derived based on this piecewise linear structure.

Because our problems are so large, we have opted for a simpler but approximate approach. Floudas (1995) has suggested that the functions L^* be approximated as follows: let $\bar{y} \in Y$ and let $(\bar{x}, \bar{p}, \bar{n}, \bar{u})$ be primal and dual solutions for $P(\bar{y})$. Then use

$$L^*(y, \bar{u}) = L(\bar{x}, y, \bar{p}, \bar{n}, \bar{u}). \quad (37)$$

This holds for problems where f and g are separable in x and y. In addition, it is stated in (Floudas 1995, p. 131), that the right hand side of (37) is a supporting function for $v(y)$ at $\bar{y}$ when $v(y)$ is a convex function, hence it can serve as a valid GBD cut. See this reference for further details and some small examples, which are illustrated geometrically. However, the problems considered here are not separable, and there is no guarantee that $v(y)$ is convex. Hence, for our problems, while (37) holds at $\bar{y}$, it does not hold in general at other y, since $\bar{x}$ need not maximize L for y different from $\bar{y}$. In general, the linear function on the right-hand side of (37) does not coincide with any piece of L^*, unless the reduced costs ds_j satisfy special conditions like those in (35). For example, if $\bar{x}_j$ is strictly between its bounds, then $ds_j(\overline{sy}_j, \bar{u}) = 0$. However, as y varies while u remains fixed, $\bar{x}_j$ is no longer a maximizer unless ds_j remains zero. Hence, these approximate cuts need not provide valid upper bounds. Nonetheless, we have obtained good solutions using them, as we discuss in §6. However, our results include instances where the optimal RMP value is lower than the objective value of a locally optimal solution found by another solver.

3.2. Reservoir Operation Model

The previous results also hold for the reservoir operation model. We choose $y = H$ and $x = (Q, RI, G, S, P)$. Using (37) as the GBD cut leads to feasible solutions of high quality, as we describe in §6. With $y = H$, the coupling constraints are the hydropower generation constraints (13) and the reservoir head-volume relationship (14), and these, plus the water balance constraints (8)–(12) are the constraints of the GBD subproblem. They are linear for fixed y. As with the salt balance constraints of the salinity control model, the constraints (13) are relaxed by introducing elastic variables, ensuring that the subproblem is feasible for all y in Y. The set Y is determined by the upper and lower bounds on y. The GBD relaxed master program involves only the GBD cuts in the approximate form (37) and the bounds on y. As a result, both master and subproblem are linear programs.

4. IMPLEMENTATION

This algorithm has been implemented using the algebraic modeling language GAMS (Brooke et al. 1996), using GAMS version 2.50. Both subproblem and relaxed master program (RMP) models are defined, and a loop statement is used to drive the GBD algorithm. This loop contains two SOLVE statements, one for the master and subproblem, both using the solver GAMS/OSL. The GBD cut is created by a GAMS statement which involves the optimal values of the primal and dual variables from the previous subproblem solution, and the new cut is indexed by the loop index. The GBD termination criterion is that of step 5 of §3, with $\varepsilon = 1.0\text{E-}3$.

Our GAMS program exploits the fact that the GBD subproblem structure remains the same at each iteration, but with different parameters (values of y), and that only one equation is added to the RMP. All solutions of these linear programs after the first take advantage of GAMS' automatic warm starting capabilities. The optimal basis from each LP is saving and used as a starting basis for the next SOLVE. This restarting facility saves significant computing time (Brooke et al. 1996).

As to initial values for y, the reservoir model instances solved here have either 12 or 24 complicating variables, the heads of one or two reservoirs in each period, and it is easy to provide good initial guesses for these. The water allocation and salinity control model has more sets of complicating variables, e.g., all flows and storages. Some instances solved in §6 have 1499 complicating variables. Initial values of y are best chosen by first solving the RMP with no GBD cuts, i.e., choosing the flow and storage variables to optimize the portion of the model in which only they appear. Section 6 also shows results where initial y's values are selected by assigning identical "ballpark" values to each variable within a set of related variables, e.g., flows into and out of river nodes and flows from rivers to canals.

Each model discussed in §2 is first expressed directly, and either MINOS5 or CONOPT2 (the new version of CONOPT available with GAMS 2.50) is used as a nonlinear solver for both models. Then the models are implemented in the GBD form described above. The results for the two models, solved by either MINOS5 or CONOPT2 and by GBD, are discussed in the following section.

5. COMPUTATIONAL RESULTS

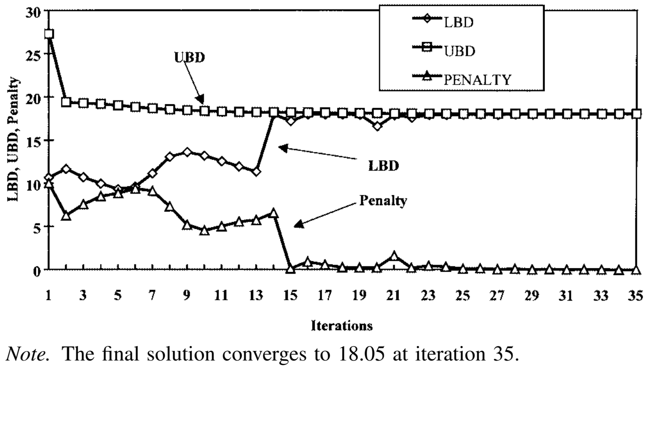
5.1. Reservoir System Operation Model

The instances of the reservoir system operation model used in our computational experiments are described in McKinney and Cai (1997). These models have been used to analyze the tradeoff between upstream power generation and downstream irrigation in the Syrdarya River basin in Central Asia, and the model data is specific to this region (McKinney and Cai 1997). This river basin network includes 13 reservoirs, 30 river reaches, 6 aquifers, 6 demand sites (irrigation, municipal, and industrial water demand), and 5 hydroelectric power generation stations.

Among the 5 hydropower stations, the linear head/storage relations (14) were only defined for the Toktogul Reservoir (14.5 km^3 of active storage volume) which is the major reservoir for flow control and power generation in the basin. The reservoir surfaces of the other hydropower stations are maintained at constant elevation, and thus their heads are treated as constants. The time periods are months, with horizons ranging from 12 to 60 months. The model with 12 time periods consists of 895 equations, 1151 variables, and 2673 nonzero Jacobian elements. Considering the monthly average heads of the Toktogul Reservoir for one year as complicating variables, with all other reservoir elevations taken as constant, there are 12 complicating variables and 24 coupling constraints, 12 power generation constraints (13), and 12 head/volume equations (14). If heads of both the Toktogul and another reservoir (Andijan) are allowed to vary, the number of complicating variables increases to 24, and the number of coupling constraints becomes 48.

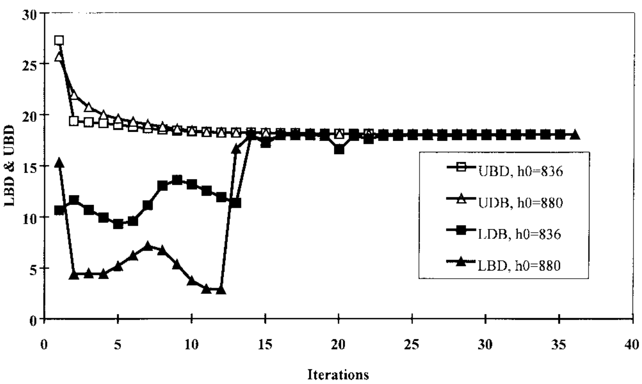
Figure 1 shows the lower bound, upper bound, and penalty cost vs. iteration for the model with 12 complicating variables. In this case, the penalty cost on the slack variables goes to zero, the upper bound always decreases, and the lower bound fluctuates until shortly before the penalty reaches zero, increasing thereafter, until it reaches the upper bound. Figure 2 shows the upper and lower bounds vs. iteration for two sets of initial values for the Toktogul

Figure 1. The lower bound (LBD), upper bound (UBD), and penalty for the reservoir operation model with 12 complicating variables.



Note. The final solution converges to 18.05 at iteration 35.

Figure 2. Upper and lower bounds for various initial values of the complicating variables (reservoir surface elevation in each month).

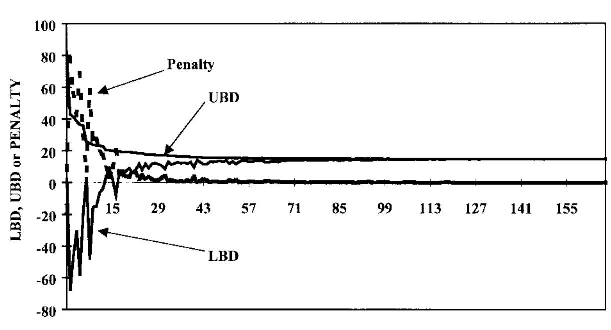


Note. All scenarios converge to the same final value, 18.05.

Reservoir heads in all time periods. In the first scenario, the initial values for elevation in all twelve periods are 836 m, which corresponds to the dead storage of the reservoir. In the second, the initial values are all 880 m, which corresponds to the full storage of the reservoir. In both scenarios, the gap between the two bounds is substantial until around iteration 13, when it shrinks sharply as the lower bounds improve, and then converges slowly to zero at iteration 37. The upper bounds provided by the RMPs are nearly optimal after about 10 iterations. Solving this model directly using MINOS5 with the same starting points leads to the same solution.

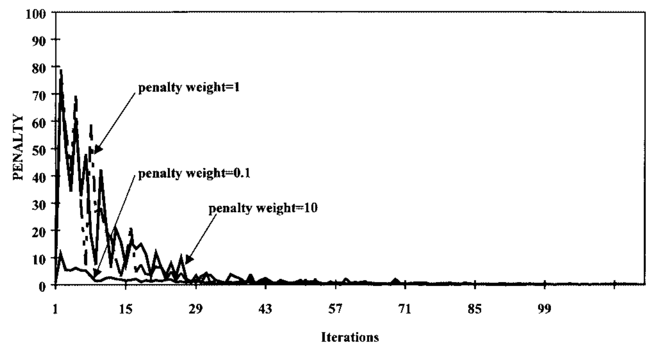
Figure 3 shows the lower bound, upper bound, and penalty cost vs. iterations for the model with 24 complicating variables. Here convergence is much slower, requiring about 155 iterations, with computation time increasing accordingly. The penalty weight, M, also strongly affects GBD convergence and computing time. Figure 4 shows that, for weights of 1 and 10, the penalty cost converges to zero, while for $M = 0.1$ it converges to 1.6. Figure 5 shows the effect of the penalty weight on GBD convergence. The run with $M = 10$ converges fastest, while GBD does not converge for $M = 0.1$.

Figure 3. Lower bound (LBD), upper bound (UBD), and penalty for the reservoir operation model with 24 complicating variables.



Note. The final solution converges to 18.77 at iteration 155.

Figure 4. Penalty vs. iteration for the reservoir operation model with 24 complicating variables under various penalty weights.



In these figures, MINOS5 is used as the LP solver. If MINOS5 is replaced by OSL, the number of iterations does not change significantly (for the model with 24 complicating variables, the number of iterations is 155 for MINOS5, and 137 for OSL), but the total computing time is reduced to about 25% of the MINOS5 time.

5.2. River Basin Water Allocation and Salinity Control Model

The river basin water allocation model used in this paper was developed by McKinney et al. (1997) for the Karshi region of the Amudarya River basin of Uzbekistan. Its mathematical structure is described in §2. It is used to support water allocation decisions with multiple goals, including satisfying water demand, maintaining river flow for ecological protection, balancing water use rights among demand sites, and controlling salinity. The network used to model this basin includes 6 reservoirs, 6 aquifers, 8 river reaches, 2 canals, 7 agricultural drainage water collectors, and 10 demand sites (irrigation, municipal, and industrial water demand). There are 12 monthly time periods. When salt concentrations are included in the objective, this model instance has 1567 constraints, 2039 structural variables (not including the deviation variables), and 9129 nonzero Jacobian elements, of which 4773 (about 52%) are nonconstant. Hence this model is highly nonlinear, due to the presence

Figure 5. Upper bound vs. iteration for the reservoir operation model with 24 complicating variables using various penalty weights.

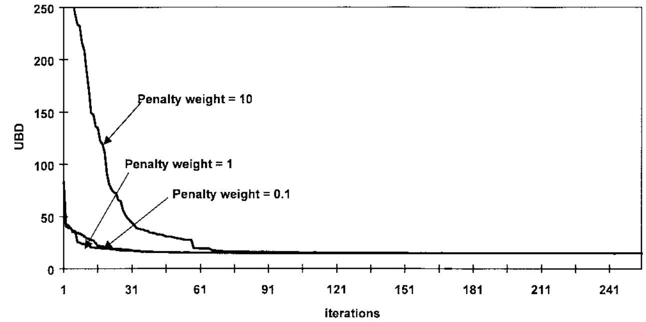


Table 2. GBD and MINOS5 final objective values for four water allocation model cases.

Case	Definition	Significance	GBD Obj	MINOS5 Obj	CONOPT2 Obj
1	No obj salt, no dead storage	Best for GBD	5.289	5.289	5.289
2	No obj salt, yes dead storage	Close to best	5.288	5.288	5.288
3	Yes obj salt, no dead storage	Much farther from best	4.975	4.978	4.978
4	Yes obj salt, yes dead storage	Worst for GBD	4.974	4.975	4.981

of 540 nonlinear salt balance constraints. There are 1499 complicating variables y, 540 salt concentrations, and 540 coupling constraints.

The storage of each reservoir and aquifer is divided into a constant dead storage, and a variable live storage. Only the live storage can be utilized, so only live storage variables appear in water balance equations. However, salt balance equations for reservoirs and aquifers involve both types of storage, so they contain terms in which this constant storage volume is multiplied by variable salt concentrations. Hence these salt balance equations contain linear terms involving x alone. The objective term $c_1^T x$ also introduces a linear term in x into the Lagrangian. We conjectured that such terms would degrade the performance of GBD, because they would force the optimal flows and storages to be further away from their starting values, those which optimize the portion of the model not including salt concentrations. However, the salt concentrations of these aquifers and reservoirs do not change much over time, so deleting the dead storage terms has little impact on the model's solution. In addition, we can delete or include the salt concentration terms in the objective, which has a greater effect. Hence, we can define 4 cases as shown in Table 2, representing increasing distance from the "best" situation of Case 1. In each case, we first solve the problem using GBD and then, using the final GBD solution as an initial point, solve it using MINOS5 and (separately) CONOPT2. In all GBD runs, the initial y values are computed by solving the relaxed master program with no cuts, the "optimal flow" solution. Despite the excellent initial points, MINOS5 and CONOPT2 were unable to improve on the first four significant figures of the final GBD values in Cases 1 and 2, so this value is at least locally optimal to within the rather tight default tolerances of MINOS5 and CONOPT2. There are slight improvements to Cases 3 and 4, confirming our hypothesis that these cases are less favorable to GBD. Still, the largest improvement, by CONOPT2 in Case 4, is only 0.14% better than the GBD value. The CONOPT2 final objective value for Case 4 of 4.981 is not the best we have obtained for this problem instance—CONOPT2 attains 4.9838 using the starting point of case 4-2 in Table 3. It is unclear if these values represent two distinct local optima or if the objective is just very flat near these points.

As discussed above, the most effective way to choose initial y's is to solve the RMP with no GBD cuts. Figure 6 shows the behavior of the lower bound, upper bound and the penalty term versus iteration count for the model of Case 4 of Table 2 when the initial y is chosen in this way. This model, with 1499 complicating variables, converges in 13 iterations, much less than the 155 iterations required for the reservoir operation model, which has only 24 complicating variables. We believe that this faster convergence is due to the "tighter" RMP of the water allocation models. The RMP of the reservoir operation model has only one set of constraints, the GBD cuts, with a new cut added at each iteration. In early iterations, when the RMP has few constraints, the proposals provided by this RMP are far from optimal. For the water allocation model, the RMP includes the flow balance equations, which constrain this RMP more tightly than the RMP of the reservoir operation model. The tighter RMP provides proposals that are of higher quality, resulting in faster convergence.

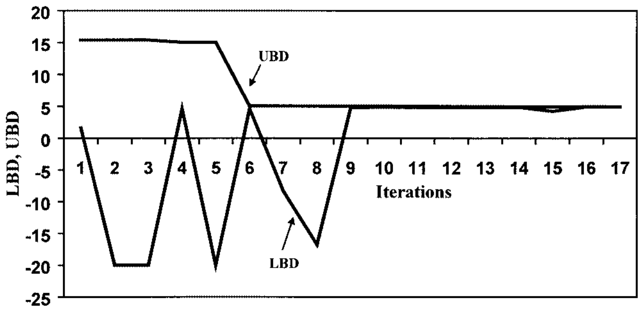
Table 3 shows the final objective values, GBD iterations, and run times of GBD, MINOS5, and CONOPT2

Table 3. Performance of GBD, MINOS, and CONOPT using four different initial points on the problem defined as Case 4 of Table 1.

	GBD			MINOS5				CONOPT2			
	Obj	Iteration	Time	First Run		Third Run		First Run		Third Run	
				Obj	Time	Obj	Time*	Obj	Time	Obj	Time*
Case 4-1	4.9744	13	20.5	4.9809	68.71	4.9809	3.7	4.9812	24.0	4.9812	3.7
Case 4-2	4.9535	10	18.6	4.9837	738.2	4.9837	3.0	Failed	50.2	4.9838	172.8
Case 4-3	4.9426	15	23.9	4.8992	535.1	4.9760	3.0	4.9794	202.5	4.9794	0.44
Case 4-4	4.9653	12	19.8	4.9816	522.7	4.9816	2.3	4.9787	74.5	4.9787	0.54

Notes: Case 4-1: initial y values obtained by solving the GBD master program with no cuts.
Cases 4-2, 4-3, 4-4 use three sets of "ballpark" initial guesses for y.
All programs were run on the same PC-300, Pentium-II. The computational time here is defined as the "resource usage" in the GAMS output file. The time unit is second.
*Additional time since the first run.

Figure 6. GBD lower bound (LBD) and upper bound (UBD) for the problem in Case 4 of Table 1.



when they use the same initial point when solving Case 4 (the most realistic case) in Table 2. MINOS5 and CONOPT2 use all default tolerances and options. Run times shown are the "resource usage" times reported on the GAMS listing file for the solution phase. The computer used is a Pentium II 300 mhz PC. Four different initial points are used. The best initial point, Case 4-1, chooses initial values for y as the "optimal flow" solution described above, and initial x's by solving the GBD subproblem. The other three cases use different "ballpark" initial y values as discussed in §5, again choosing x by solving the GBD subproblem. GBD produces objective values slightly worse that the other two solvers in all cases, but the largest difference, in Case 4-3, is only 0.74%. GBD is much faster than the other two solvers, often by more than a factor of 20 over MINOS5, and by factors of 3 to 9 over CONOPT2. GBD time is affected very little by the starting point. In the column "MINOS5, Third Run", we show the final objective value resulting from 3 successive applications of MINOS5, using 3 consecutive GAMS SOLVE statements. The first solve uses the same initial solution as GBD, and the second and third solves use the result of the previous SOLVE as starting points. This easy-to-implement strategy improves the MINOS5 objective value achieved with one SOLVE by 1.56% in Case 4-3. Using a single SOLVE, CONOPT2 achieves slightly better objective values than the other solvers in two of the four cases, but fails to find a feasible solution in Case 4-2. However, CONOPT2 achieves the best objective value of all using three successive solves in this case.

6. CONCLUSION

Using a relaxed formulation of the GBD subproblems and a sufficiently large penalty weight, GBD has performed well in solving the large, nonconvex, bilinear problems studied here. For the water allocation and salinity control model, GBD solution quality is comparable to that of MINOS5 or CONOPT2, and GBD is considerably faster. Further, GBD can be used to advantage in conjunction with these or any other local solvers, by using the final GBD solution as an initial point for the local solver. In our experiments with the water allocation and salinity control model, MINOS5 and CONOPT2 were able to improve this solution to a small degree. For the reservoir operation models, GBD computation time increases as the number of complicating variables increases from 12 to 24, and in both cases, GBD computing time exceeds that of MINOS5 or CONOPT2. However, experience with the water allocation models indicates that GBD can handle roughly 1500 complicating variables in situations where the RMP is tightly constrained by the constraints defining Y. Hence, analysts considering using GBD to solve nonlinear problems should consider carefully which variables are designated as complicating, and consider reversing the "natural" assignment, which tries to minimize the number of these variables. Our experiments with the water allocation and salinity control model also show that, compared to MINOS5 and CONOPT2, GBD's final results are less sensitive to the starting point.

In summary, we conclude that the approach presented in this paper can be used to search for at least approximate global solutions to models with nonlinear and nonconvex constraints. The computing time depends on the model size, the model structure, and the selection of complicating variables in the GBD formulation. We recommend this approach for large models with special structure, such as the river basin water allocation and salinity control model discussed in this paper.

ACKNOWLEDGMENT

The authors are grateful to Professor Benjamin Hobbs, the Area Editor in *Environment, Energy, and Natural Resources*, and two anonymous reviewers for their very helpful comments and suggestions.

REFERENCES

Brooke A., D. Kendrick, A. Meeraus. 1996. *GAMS: A User's Guide*. GAMS Development Corporation, Washington, DC.

Drud, A. S. 1994. CONOPT—A Large Scale GRG Code. *ORSA J. Comput.* **6** 207–216.

Floudas, C. A., A. Aggarwal, A. R. Ciric. 1989. Global Optimum Search for Nonconvex NLP and MINLP Problems. *Comput. Chem. Engrg.* **13** 1117–1132.

——. 1995. *Nonlinear and Mixed-Integer Optimization*. Oxford Univ. Press, New York.

Geoffrion, A. M. 1972. Generalized Benders Decomposition. *J. Optim. Theory Appl.* **10** 237–259.

Gorenstin B. G., N. M. Campodonico, J. P. Costa, M. V. F. Pereira. 1993. Power System Expansion Planning under Uncertainty, *IEEE Trans. Power Systems* **8**(1).

Horst, R., Hoang Tuy. 1990. *Global Optimization-Deterministic Approaches*. Springer, New York.

Karatzas, G. P., G. F. Pinder. 1996. The Solution of Groundwater Quality Management Problems with a Nonconvex Feasible Region Using a Cutting Plane Optimization Technique. *Water Resources Res.* **32** 3091–3100.

McKinney D. C., M. D. Lin. 1994. Genetic Algorithm Solution of Groundwater Management Models. *Water Resources Res.* **30** 1897–1906.

——, A. Kh. Karimov, X. Cai, 1997. Model development: Aral Sea Regional Allocation Model of the Amu Darya

River. Project Report, Environmental Policy and Technology Project, US Agency for International Development, Central Asian Regional Office, Almaty, Kazakstan.

——, X. Cai, 1997. Multi-Objective Water Resources Allocation Model for the Toktogul Reservoir. Technical Report, Environmental Policy and Technology Project, US Agency for International Development, Central Asian Regional Office, Almaty, Kazakstan.

Murtagh, B. A., M. A. Saunders. 1982. A Projected Lagrangian Algorithm and its Implementation for Sparse Nonlinear Constraints. *Math. Programming Study* **16** 84–117.

Pereira, M. V. F., L. M. V. G. Pinto. 1985. Stochastic Optimization of a Multireservoir Hydroelectric System: A Decomposition Approach. *Water Resources Res.* **21**(6) 779–792.

Ritzel, B. J., J. W. Eheart, S. Ranjithan. 1994. Using Genetic Algorithms to Solve a Multiple Objective Groundwater Pollution Contaminant Problem. *Water Resources Res.* **30** 1589–1603.

Viswesaran, V., C. A. Floudas. 1996. New Formulations and Branching Strategies for the GOP Algorithm I. E. Grossman, ed. *Global Optimization in Chemical Engineering*, 75–100. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Watkins, D. W., D. C. McKinney. 1998. Decomposition Methods for Water Resources Optimization Models with Fixed Costs. *Adv. Water Res.* **21** 283–295.

Yeh, W. W.-G. 1985. Reservoir Management and Operation Models: A State-of-Art Review, *Water Resources Res.* **21**(12) 1797–1818.