Improved multiple-objective dynamic programming model for reservoir operation optimization

Tongtiegang Zhao and Jianshi Zhao

ABSTRACT

Reservoirs are usually designed and operated for multiple purposes, which makes the multiple-objective issue important in reservoir operation. Based on multiple-objective dynamic programming (MODP), this study proposes an improved multiple-objective DP (IMODP) algorithm for reservoir operation optimization, which can be used to solve multiple-objective optimization models regardless whether the curvatures of trade-offs among objectives are concave or not. MODP retains all the Pareto-optimal solutions through backward induction, resulting in the exponential increase of computational burden with the length of study horizon. To improve the computational efficiency, this study incorporates the ranking technique into MODP and proposes an efficient IMODP algorithm. We demonstrate the effectiveness of IMODP through a hypothetical test and a real-world case. The hypothetical test includes three cases in which the trade-offs between objectives are concave, convex, and neither concave nor convex. The results show that IMODP satisfactorily captures the Pareto front for all three cases. The real-world test focuses on hydropower and analyzes the trade-offs between total energy and firm energy for Danjiangkou Reservoir. IMODP efficiently identifies the Pareto-optimal solutions and the trade-offs among objectives.

Key words | Danjiangkou Reservoir, Pareto front, ranking technique, trade-off analysis, two-stage formulation

Tongtiegang Zhao
Jianshi Zhao (corresponding author)
State Key Laboratory of Hydro-Science and Engineering,
Department of Hydraulic Engineering,
Tsinghua University,
Beijing,
China
E-mail: *zhaojianshi@tsinghua.edu.cn*

INTRODUCTION

A reservoir system serves multiple operational targets, for example, flood control, water supply, irrigation, and hydropower. Analysis of the trade-offs among multiple objectives has been an important issue in reservoir operation studies (Labadie 2004; Loucks & van Beek 2005). The operational targets exhibit different physical and economic properties and usually result in varying operation decisions (Lund & Guzman 1999; McMahon & Farmer 2004). For example, flood control is risk sensitive because floods cause upstream and downstream damage, and reservoir storage is kept at a low level during the flood season in anticipation of extreme floods (Cheng & Chau 2004; Galelli & Soncini-Sessa 2010; Li *et al.* 2010). Water supply exhibits diminishing marginal utilities and results in a smooth sequence of reservoir release (Draper & Lund 2004; Zhao *et al.* 2011). Irrigation water demand exhibits seasonality, and it is affected by meteorological conditions (Galelli & Soncini-Sessa 2010). Hydropower is affected by the electricity market, and the time-varying electricity demand can induce dramatic changes in reservoir releases (Perez-Diaz *et al.* 2010).

Multiple-objective analysis of a reservoir system is generally difficult, partly due to computational problems involved in modeling the complex hydrological system and partly due to the economic issues in dealing with the trade-offs (Loucks & van Beek 2005; Castelletti *et al.* 2011). Multiple operational targets are characterized by their respective objective functions, forming a vectorial objective. Thus, in contrast to the single-objective problem with only one optimal solution, the multiple-objective problem can have a number of Pareto-optimal solutions with each solution corresponding to a non-dominated objective vector (Lotov *et al.* 2004; Loucks & van Beek 2005).

doi: 10.2166/hydro.2014.004

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

However, most optimization algorithms are designed for single-objective problems. Therefore, the multiple-objective reservoir operation problems are usually converted into single-objective problems, for example, by the weighting method (assigning weights to objectives) or the constraint method (accounting for objectives by constraints), before being solved (Akbari *et al.* 2011; Pianosi *et al.* 2013). By adjusting the weights and modifying the constraints, Pareto-optimal solutions and non-dominated objective vectors can be determined (Galelli & Soncini-Sessa 2010; Castelletti *et al.* 2010a, 2010b). Another approach involves the use of multi-objective evolutionary algorithms (MOEA) to solve the multiple-objective problems directly (Suen & Eheart 2006; Wang & Singh 2007). Compared with mathematical programming based on derivations of gradient, Hessian matrix, and principle of optimality, among others, MOEA methods based on simulation-optimization can handle multiple-objective problems as easily as single-objective ones (Deb *et al.* 2002; Reddy & Kumar 2007; Kim *et al.* 2008).

Dynamic programming (DP) and stochastic DP (SDP) with weighting methods have been widely used in multiple-objective reservoir problems owing to their ability to exploit the sequential nature of reservoir operations and to handle non-linear and non-continuous objectives and constraints (Yakowitz 1982; Labadie 2004). On the other hand, DP and SDP suffer from the curse of dimensionality. To deal with this problem in complex water resource systems, a number of advanced computation methods have been incorporated into DP and SDP, for example, the decomposition method (Archibald *et al.* 1997), meta-modeling (Galelli & Soncini-Sessa 2010), reinforcement learning (Lee & Labadie 2007; Castelletti *et al.* 2010b; Pianosi *et al.* 2013), and response surface methods (Castelletti *et al.* 2010a, 2011). Besides the weighting method, there are some direct approaches to handle multiple-objective problems with DP, for example, treating secondary objectives as state variables (Tauxe *et al.* 1979), applying clustering to reduce the set of Pareto-optimal solution (Rosenman & Gero 1985). These improvements address efficiently the hydrological and economic complexities and exhibit convincing results that aid in the design and management of reservoir systems (Lee & Labadie 2007; Galelli & Soncini-Sessa 2010; Castelletti *et al.* 2011).

Daellenbach & Dekluyver (1980) proposed the multiple-objective DP (MODP) method. One significant advantage of MODP is that it deals directly with multiple objectives regardless of whether the curvatures of trade-offs among objectives are concave or not. On the other hand, MODP retains all the Pareto-optimal solutions in recursive computation, resulting in an exponentially increasing computational burden with the length of study horizon. This study presents a first application of MODP to water resources management. Besides, to improve its computational efficiency, we incorporate the ranking technique to select a representative set of Pareto-optimal solutions (instead of retaining all solutions) and propose an improved MODP (IMODP). We demonstrate application of IMODP through hypothetical and real-world case studies and show its effectiveness in handling non-concave trade-offs.

The remainder of this paper is organized as follows: the two-stage formulation of a multi-objective reservoir operation is presented in the following section; then, the MODP and IMODP algorithms are elaborated. The next two sections respectively present a hypothetical test and a real-world case study to demonstrate the effectiveness and efficiency of IMODP. Finally, discussions and conclusions are presented.

DP-BASED MULTIPLE-OBJECTIVE ANALYSIS

Reservoir operation features multiple periods and sequential decision making. This section illustrates the two-stage formulations of DP and MODP for reservoir operation. To access MODP easily, the formulations of DP are initiated by single-objective analysis and then extended to multiple-objective analyses.

Conventional DP

Let us consider a reservoir operation problem with a study horizon of T periods. The variables are denoted as follows: t, index of time periods; s_t, reservoir storage at the beginning of period t; q_t, reservoir inflow during period t; r_t, reservoir release during period t; $g_t(s_t, q_t, r_t)$, utility function of period t, which depends on s_t, q_t, and r_t; $\underline{s}$, lower bound of reservoir storage; $\overline{s}$, upper bound of reservoir storage;

$\bar{r}$, lower bound of reservoir release; $\bar{r}$, upper bound of reservoir release, for example, maximum release rate of the hydropower turbine and spillway; s_{ini}, initial storage at the beginning of the study horizon; s_{end}, target storage at the end of the study horizon. Using the above variables and selecting r_t as the decision variable, the deterministic single-objective reservoir operation can be formulated as follows (Labadie 2004):

$$
\begin{array}{lll}
\max & \sum_{t=1}^{T} g_t(s_t, q_t, r_t) & (1.1) \\
s.t. & \left\{ \begin{array}{lll}
s_t + q_t - r_t = s_{t+1} & (t = 1, \ldots, T) & (1.2) \\
\underline{r} \le r_t \le \bar{r} & (t = 1, \ldots, T) & (1.3) \\
\underline{s} \le s_t \le \bar{s} & (t = 2, \ldots, T) & (1.4) \\
s_1 = s_{\text{ini}} & & (1.5) \\
s_{T+1} = s_{\text{end}} & & (1.6)
\end{array} \right. &
\end{array}
\qquad (1)
$$

In Equation (1), (1.1) indicates the objective function, that is, the maximization of the sum of utilities from periods 1 to T; (1.2) represents the water-balance relationship; (1.3) and (1.4) denote the capacity constraints of the storage and release, respectively; and (1.5) and (1.6) are the initial and ending storage constraints, respectively. Notably, hydrological conditions $[q_1, q_2, \ldots, q_T]$ are supposed to be perfectly known in Equation (1).

The sequential nature of Equation (1) enables efficient applications of DP to determine the optimal solutions (Yakowitz 1982; Loucks & van Beek 2005). The two-stage formulation of DP is expressed as follows:

$$
\begin{array}{lll}
G_t(s_t) = \max & g_t(s_t, q_t, r_t) + G_{t+1}(s_{t+1}) & (2.1) \\
s.t. & \left\{ \begin{array}{l}
s_t + q_t - r_t = s_{t+1} \\
\underline{r} \le r_t \le \bar{r} \\
\underline{s} \le s_{t+1} \le \bar{s}
\end{array} \right. & \begin{array}{l} (2.2) \\ (2.3) \\ (2.4) \end{array}
\end{array}
\qquad (2)
$$

which is called the recursive function of DP, and $G_t(s_t)$ represents the maximum cumulative utility function from periods T to t. Using backward induction from periods T–1 to 1, the $G_t(s_t)$ values ($t = T$–1, T–2, …, 1) are sequentially determined, and the reservoir operation optimization in Equation (1) is achieved.

The key idea behind why Equation (2) produces the optimal solutions of Equation (1) is called the principle of optimality (Yakowitz 1982; Rust 1997; Kuhn 2006), that is, 'An optimal policy has the property that whatever the initial state and initial solution are, the remaining solutions must constitute an optimal policy with regard to the state from the first solution.' This principle indicates the following: (1) a given value of s_{t+1} corresponds to an optimal release sequence of $[r_{t+1}, \ldots, r_T]$, that is, the optimal remaining solutions from periods $t+1$ to T; and (2) if the optimal release and carried-over storage of s_t are r^*_t and s^*_{t+1} (s^*_{t+1} corresponds to $[r^*_{t+1}, \ldots, r^*_T]$), respectively, then the optimal release sequence of s_t must be $[r^*_t, r^*_{t+1}, \ldots, r^*_T]$. Therefore, through backward induction, the optimal release sequence of s_t ($t = T$–1, …, 1) can be sequentially determined, and the optimal solutions of Equation (1) is included in the case where $s_1 = s_{\text{ini}}$ (Equation (1.5)).

Multiple-objective DP

When Equation (1) is extended to multiple-objective cases, the reservoir operation problem becomes

$$
\begin{array}{lll}
\max & \begin{bmatrix} \sum_{t=1}^{T} g_{1,t}(s_t, q_t, r_t) \\ \vdots \\ \sum_{t=1}^{T} g_{M,t}(s_t, q_t, r_t) \end{bmatrix} & (3.1) \\
s.t. & \left\{ \begin{array}{lll}
s_t + q_t - r_t = s_{t+1} & (t = 1, \ldots, T) & (3.2) \\
\underline{r} \le r_t \le \bar{r} & (t = 1, \ldots, T) & (3.3) \\
\underline{s} \le s_t \le \bar{s} & (t = 2, \ldots, T) & (3.4) \\
s_1 = s_{\text{ini}} & & (3.5) \\
s_{T+1} = s_{\text{end}} & & (3.6)
\end{array} \right. &
\end{array}
\qquad (3)
$$

In Equation (3), $g_{m,t}(s_t,q_t,r_t)$ denotes the mth objective function for period t, which is analogous to $g_t(s_t,q_t,r_t)$ in Equation (1), whereas the value of m ranges from 1 to M (the number of objectives).

Similar to Equation (2), the two-stage formulation of Equation (3) is as follows:

$$
\begin{array}{lll}
\begin{bmatrix} G_{1,t}(s_t) \\ \vdots \\ G_{M,t}(s_t) \end{bmatrix} = \max & \begin{bmatrix} g_{1,t}(s_t, q_t, r_t) + G_{1,t+1}(s_{t+1}) \\ \vdots \\ g_{M,t}(s_t, q_t, r_t) + G_{M,t+1}(s_{t+1}) \end{bmatrix} & (4.1) \\
s.t. & \left\{ \begin{array}{l}
s_t + q_t - r_t = s_{t+1} \\
\underline{r} \le r_t \le \bar{r} \\
\underline{s} \le s_{t+1} \le \bar{s}
\end{array} \right. & \begin{array}{l} (4.2) \\ (4.3) \\ (4.4) \end{array}
\end{array}
\qquad (4)
$$

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

In Equation (4), $[G_{1,t}(s_t), \ldots, G_{M,t}(s_t)]^{\mathrm{T}}$ denotes the objective vectors of the non-dominated cumulative utilities from periods T to t.

Despite the similar forms of Equations (2) and (4), essential differences exist between the two DP models. Equation (2) determines one set of solutions with maximum $G_t(s_t)$ for each s_t. Equation (4) can output multiple sets of Pareto-optimal solutions with non-dominated $[G_{1,t}(s_t), \ldots, G_{M,t}(s_t)]^{\mathrm{T}}$ for each s_t (Daellenbach & Dekluyver 1980). The non-dominated objective vector is identified as follows: for $[G_{1,t}(s_t), \ldots, G_{M,t}(s_t)]^{\mathrm{T}}$ of s_t, if no other objective vector $[G'_{1,t}(s_t), \ldots, G'_{M,t}(s_t)]^{\mathrm{T}}$ of s_t that satisfies the following condition exists

$$\begin{cases} G_{m,t}(s_t) \leq G'_{m,t}(s_t) & (\forall m \in [1, \ldots, M]) \\ G_{m,t}(s_t) < G'_{m,t}(s_t) & (\exists m \in [1, \ldots, M]) \end{cases} \tag{5}$$

then $[G_{1,t}(s_t), \ldots, G_{M,t}(s_t)]^{\mathrm{T}}$ is considered as non-dominated. Notably, the non-dominated $[G_{1,t}(s_t), \ldots, G_{M,t}(s_t)]^{\mathrm{T}}$ is determined by a solution $[r_t, \ldots, r_T]$, that is, $G_{m,t}(s_t) = \sum_{i=t}^{T} g_{m,i}(s_i, q_t, r_i) \quad (m = 1, \ldots, M)$. Correspondingly, $[r_t, \ldots, r_T]$ is called the Pareto-optimal solution of $[G_{1,t}(s_t), \ldots, G_{M,t}(s_t)]^{\mathrm{T}}$. Each non-dominated objective vector corresponds to a set of Pareto-optimal solutions.

The principle of optimality of MODP (Daellenbach & Dekluyver 1980) states that: ‘a non-dominated policy has the property that, regardless of how the process entered a given state, the remaining solutions must belong to a non-dominated sub-policy’. The principle of optimality of MODP is analogous to that of DP. This principle indicates that if the Pareto-optimal release and carried-over storage of s_t are r_t and s_{t+1} (s_{t+1} corresponds to a Pareto-optimal solution $[r_{t+1}, \ldots, r_T]$), respectively, then the Pareto-optimal solution of s_t must be $[r_t, r_{t+1}, \ldots, r_T]$. We must note the following: (1) an s_{t+1} value can correspond to a number of non-dominated objective vectors $[G_{1,t+1}(s_{t+1}), \ldots, G_{M,t+1}(s_{t+1})]^{\mathrm{T}}$ and solutions $[r_{t+1}, \ldots, r_T]$, and s_t also can be associated to multiple Pareto-optimal solutions; and (2) $[r_t, r_{t+1}, \ldots, r_T]$ is a Pareto-optimal solution of s_t when the value $[G_{1,t}(s_t), \ldots, G_{M,t}(s_t)]^{\mathrm{T}} = [g_{1,t}(s_t, q_t, r_t) + G_{1,t+1}(s_{t+1}), \ldots, g_{M,t}(s_t, q_t, r_t) + G_{M,t+1}(s_{t+1})]^{\mathrm{T}}$ ($r_t = s_t + q_t – s_{t+1}$) (i.e. Equation (4.1)) is non-dominated.

MODP AND IMODP

The DP models of Equations (2) and (4) are functionals with the dependent variables as functions, that is, $G_t(s_t)$ depends on $g_t(s_t, q_t, r_t)$ and $G_{t+1}(s_{t+1})$, and $[G_{1,t}(s_t), \ldots, G_{M,t}(s_t)]^{\mathrm{T}}$ depends on $[g_{1,t}(s_t, q_t, r_t), \ldots, g_{M,t}(s_t, q_t, r_t)]^{\mathrm{T}}$ and $[G_{1,t+1}(s_{t+1}), \ldots, G_{M,t+1}(s_{t+1})]^{\mathrm{T}}$. Because of the difficulties involved in determining the analytic solutions of the functional equations (Rust 1997; Grune & Semmler 2004; Kuhn 2006), the DP models are commonly solved by numerical methods (Lee & Labadie 2007; Galelli & Soncini-Sessa 2010; Castelletti *et al.* 2011). This section introduces MODP and IMODP, which employ the discretization method to solve Equation (4). MODP was developed by Daellenbach & Dekluyver (1980), and this study improves this method using the ranking technique (Deb *et al.* 2002).

Illustration of MODP

In multiple-objective analysis, the objective functions map the space of feasible solutions into the space of objectives, and the Pareto optimality of the solutions is identified based on the non-dominated feature of the corresponding objective vectors (Lotov *et al.* 2004). The set of Pareto-optimal solutions maps into the Pareto front. According to the principle of optimality of MODP, the Pareto-optimal solution of s_t must contain the Pareto-optimal solution of s_{t+1}. One direct implication of this principle is that the Pareto front of s_t can be determined after trying all Pareto fronts of s_{t+1} values. Let us consider a reservoir operation case with two objectives, as shown in the upper part of Figure 1. One potential Pareto front of s_t is to transit to s_{t+1}, which results in the following set of releases:

$$r_t = s_t + q_t - s_{t+1} \tag{6}$$

and this transition generates a set of objective vectors by shifting the non-dominated objective vectors of s_{t+1} along the axis of obj 1 by d_1 and along the axis of obj 2 by d_2 (Figure 1(a)), that is

$$\begin{cases} d_1 = g_{1,t}(s_t, q_t, r_t) \\ d_2 = g_{2,t}(s_t, q_t, r_t) \end{cases} \tag{7}$$

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

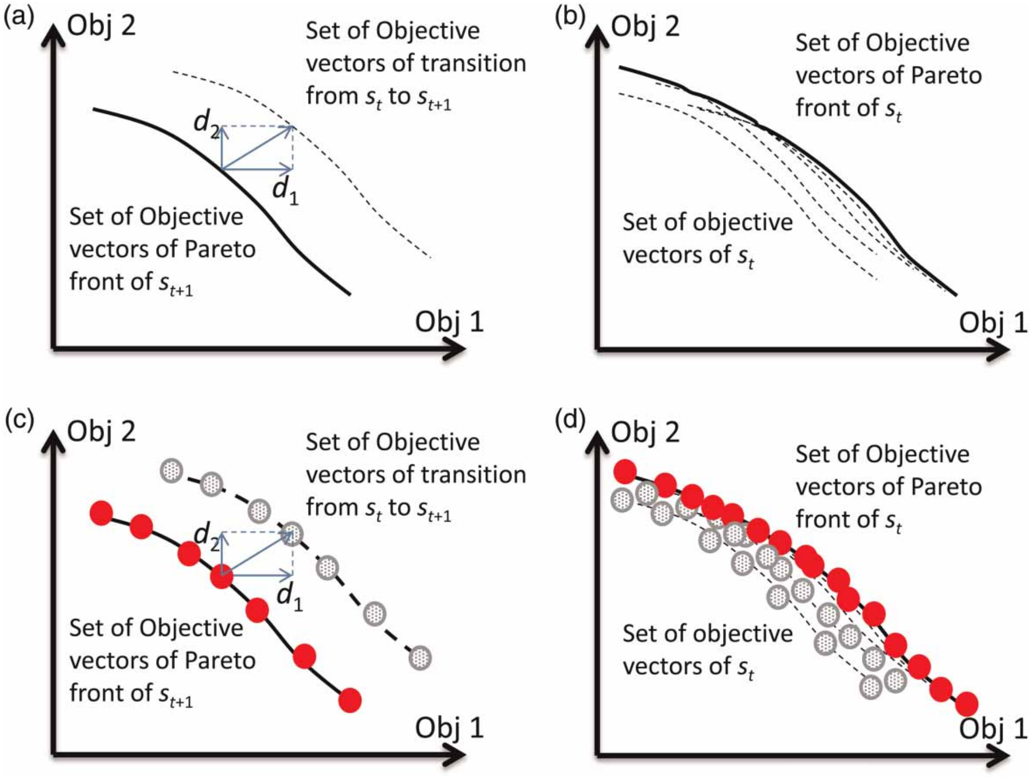

(a)
Obj 2
Obj 1
Set of Objective vectors of transition from s_t to s_{t+1}
d_2
d_1
Set of Objective vectors of Pareto front of s_{t+1}
(b)
Obj 2
Obj 1
Set of Objective vectors of Pareto front of s_t
Set of objective vectors of s_t
(c)
Obj 2
Obj 1
Set of Objective vectors of transition from s_t to s_{t+1}
d_2
d_1
Set of Objective vectors of Pareto front of s_{t+1}
(d)
Obj 2
Obj 1
Set of Objective vectors of Pareto front of s_t
Set of objective vectors of s_t

Figure 1 | Schematic of the two-stage formulation of MODP.

If the objective vectors of each s_{t+1} value are known, then the objective vectors of the transitions from s_t to all s_{t+1} values can be determined similarly. Finally, the non-dominated objective vectors of s_t can be identified and the corresponding Pareto-optimal solutions can be determined (Figure 1(b)).

The idea represented in the upper part of Figure 1 is conceptually simple, but its implementation could be difficult due to the multiple (and possibly uncountable) values of s_{t+1} and the time-consuming identification of the non-dominated objective vectors. Nevertheless, discretization of the state of storage makes this idea practically implementable and tractable. By discretization, instead of employing a continuous curve, the set of objective vectors is represented by discrete points, and each Pareto-optimal solution corresponds to a non-dominated point. If the storage is discretized into L values and the objective vectors of the Pareto front of $s_{t+1,j}$ $(j = 1, \ldots, L)$ are given, the objective vectors of the Pareto front of $s_{t,i}$ can be determined by the following procedures (Figure 2(a)).

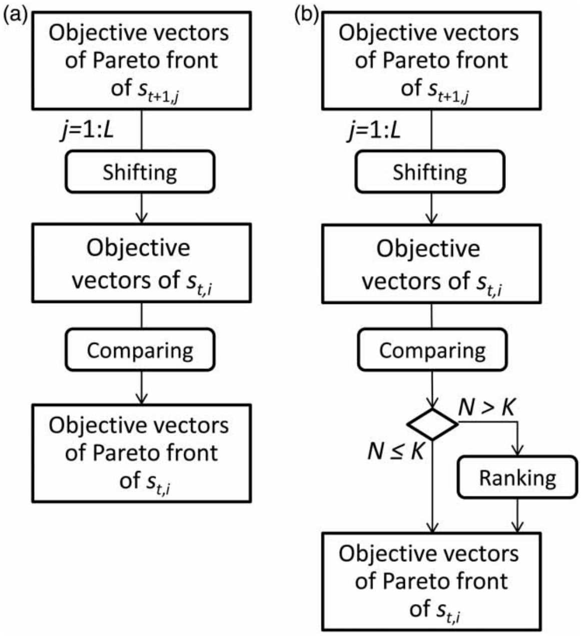

(a)
Objective vectors of Pareto front of $s_{t+1,j}$
j=1:L
Shifting
Objective vectors of $s_{t,i}$
Comparing
Objective vectors of Pareto front of $s_{t,i}$
(b)
Objective vectors of Pareto front of $s_{t+1,j}$
j=1:L
Shifting
Objective vectors of $s_{t,i}$
Comparing
$N > K$
$N \leq K$
Ranking
Objective vectors of Pareto front of $s_{t,i}$

Figure 2 | Flowcharts of MODP (a) and IMODP (b).

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

(1) Shifting the corresponding non-dominated objective vectors by $[g_{1,t}(s_{t,i},q_t,r_t), \ldots, g_{M,t}(s_{t,i},q_t,r_t)]$ $(r_t = s_{t,i} + q_t - s_{t+1,j})$ for $s_{t+1,j}$ $(j = 1, \ldots, L)$, which generates the potential non-dominated objective vectors for $s_{t,i}$ (Figure 1(c)).

(2) Comparing pairwise all the potential objective vectors and identifying the non-dominated objective vectors for $s_{t,i}$ (Figure 1(d)). Given N potential Pareto-optimal objective vectors, steps of the pairwise comparison are as follows:

```
Step 1: Initialization. For $i = 1 : N$, Mark$(i) = 1$ (the vectors are all supposed to be non-dominated and marked as 1).
Step 2: Iteration. For $i = 1 : N–1$
  (a) Set $j = i$.
  (b) While $j < N$ and Mark$(i) = 1$
        (i) Set $j = j + 1$ (checking the dominating relationship between vectors $i$ and $j$).
        (ii) If vector $i$ dominates vector $j$ (Equation (5)), set Mark$(j) = 0$, which indicates that point $j$ has been dominated.
        (iii) Else-if vector $j$ dominates vector $i$ (Equation (5)), Set Mark$(i) = 0$, which indicates that point $i$ has been dominated.
Step 3: The non-dominated solutions, that is, solutions marked as 1, are selected.
```

As illustrated above, the dominating relationships among the candidate Pareto-optimal objective vectors play an important role in reducing the computational burden, because the 'while' cycle will be interrupted before $j = N$ if 'Mark$(i) = 0$'. However, in the extreme case where N points are all non-dominated, the total computational demand is $M \times (N–1) + M \times (N–2) + , \ldots , + M \times 1 = 0.5M \times (N^2–N)$ (M is the number of objectives). Thus, the overall computational complexity of the pairwise comparison algorithm is $O(M \times N^2)$.

The MODP algorithm retains all the Pareto-optimal objective vectors through backward induction (Daellenbach & Dekluyver 1980). As shown in Figure 1, with the L discretized values of s_{t+1}, the number of potential Pareto-optimal solutions of $s_{t,i}$ is L times that of $s_{t+1,j}$. One extreme case happens when all these potential solutions are all Pareto-optimal, that is, the corresponding objective vectors are non-dominated. The number of Pareto-optimal solutions of $s_{t,i}$ will be L times that of $s_{t+1,j}$. Therefore, through backward induction, retaining all the Pareto-optimal solutions could possibly result in an exponential increase in the number of Pareto-optimal solutions ($N = L^T$). Furthermore, as the computational complexity of the pairwise comparison is $O(M \times N^2)$ (M and N are the number of objectives and solutions, respectively), the computational burden is $O(M \times L^{2T})$. Therefore, the MODP algorithm can be quite time consuming.

Illustration of IMODP

The principle of optimality of DP becomes quite complicated in multiple-objective cases, and MODP suffers from the problem of exponential increase in computational burden with length T of the study horizon, because of the multiplication of Pareto-optimal solutions along the backward optimization process. Meanwhile, MOEA based on simulation-optimization can handle multiple-objective problems as easily as single-objective ones (Deb *et al.* 2002; Reddy & Kumar 2007; Kim *et al.* 2008). This condition can be attributed to the fact that MOEA uses heuristic methods, instead of mathematical derivations of principle of optimality, to search for Pareto-optimal solutions. In the meantime, heuristic methods involve some parameters that affect computational efficiency and need empirical specifications (Reddy & Kumar 2007; Kim *et al.* 2008; Nicklow *et al.* 2010). For example, Pianosi *et al.* (2013) developed a multiple-objective fitted Q-iteration (MOFQI) method to deal with Markov decision process. MOFQI employs reinforcement learning to capture the dependence relationship between state variable and decision, which exhibits significant efficiency improvement compared with SDP. The performance of MOFQI is affected by three parameters characterizing the Extra Tree model in reinforcement learning.

This study incorporates a heuristic method into MODP and proposes an IMODP algorithm. IMODP uses selection to alleviate computational burden and involves a parameter K, that is, to select a representative set of K solutions from the whole set. The idea of IMODP is clarified by MOEA, which employs selection to control the number of solutions and thus reduce the computational burden (Rosenman & Gero 1985; Reddy & Kumar 2007; Wang & Singh 2007). We use the ranking technique, which has been used in the popular non-dominated sorting genetic algorithm-II (Deb

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

et al. 2002; Suen & Eheart 2006; Nicklow *et al.* 2010), to perform the selection task (Figure 2(b)).

The ranking technique uses the crowding distance to select representative solutions from the whole set, preserving the diversity of solutions (Deb *et al.* 2002). It consists of two steps: assigning the crowding distance and sorting of the objective vectors. We denote the number of Pareto-optimal objective vectors as N. In the steps of the crowding distance assignment, first, the initial crowding distances are set to zero. Then, the distance for each of the objectives is calculated; the crowding distance is the sum of the individual distances. The pseudo-code is expressed as follows:

```
Step 1: Initialization. For $i = 1 : N$, Distance$(i) = 0$.
Step 2: Iteration. For $m = 1 : M$ (for each objective).
    (a) Sort in ascending order the objective vectors based on the value $G_m(i)$ $(i = 1, \ldots, N)$ of objective $m$ and save the 'Order'.
    (b) Distance(Order(1)) = Distance(Order($N$)) $= +\infty$, that is, the boundary solutions, of which objective $m$ is either maximum or minimum, are assigned an infinite distance value.
    (c) For $i = 2$: $(N–1)$, Distance(Order($i$)) = Distance (Order($i$)) $+ \frac{G_m(\text{Order}(i+1)) - G_m(\text{Order}(i-1))}{G_m(\text{Order}(N)) - G_m(\text{Order}(1))}$, that is, the intermediate solutions are assigned a normalized distance value.
```

After the crowding distance assignment, sorting of the Pareto-optimal objective vectors is conducted. The objective vectors are ranked based on the corresponding *Distance* values in descending order, and the first K solutions are selected. The rationale is that the more distant the solution is from the other solutions, the more representative the solution is. Because the solution with minimum or maximum values for individual objective is assigned an infinite distance value, the ranking technique favourably retains the optimal solutions for each individual objective. By setting $K > 2 \times M$, the IMODP can determine the optimal solutions for each individual objective and also Pareto-optimal solutions that compromise amongst the M objectives. The computational complexity of the crowding distance assignment step is $O(M \times N \times \ln(N))$, and that of the sorting step is $O(N \times \ln(N))$. The overall computational complexity of the ranking technique is $O((M+1) \times N \times \ln(N))$.

The ranking technique has a parameter K, that is, the number of selected non-dominated objective vectors. This study incorporates the ranking technique into MODP and implements the method after the pairwise comparison step. If the number of non-dominated objective vectors of $s_{t,i}$ is larger than K, then the ranking technique is applied; otherwise, all the non-dominated objective vectors and the corresponding solutions are retained. Figure 2 shows the MODP and IMODP procedures.

The MODP algorithm retains all of the Pareto-optimal solutions and suffers from great computational burden. Incorporating the ranking technique into MODP, IMODP selects a representative set of K Pareto-optimal solutions and considerably reduces the computational burden. Notably, IMODP is an approximation of MODP. The parameter K plays an important role in determining computational burden and accuracy of IMODP. A larger K enables the algorithm to retain more Pareto-optimal solutions and preserves more information of the Pareto front. When the K value is set as the maximum number of Pareto-objective vectors, the MODP and IMODP algorithms are equivalent. Notably, IMODP is used for the trade-off analysis of multiple objectives and application of IMODP involves some trade-offs, that is, a larger K is more efficient in identifying the Pareto front and requires more computations.

To summarize, the IMODP has four parameters: M (the number of objective functions), T (the length of study horizon), L (the number of storage state discretization), and K (the threshold number of selected non-dominated objective vectors for each s_t value). The computational complexity of MODP depends on these parameters. For the IMODP steps shown in Figure 2(b), shifting of the non-dominated objective vectors of s_{t+1} takes $M \times L \times K$ computations; the pairwise comparison takes $M \times (L \times K)^2$ computations; and the ranking technique takes $(M+1) \times (L \times K) \times \ln(L \times K))$ computations (Deb *et al.* 2002). Among these steps, the pairwise comparison has the dominating computational complexity. Considering that t ranges from T–1 to 2 and i from 1 to L for $s_{t,i}$, the overall computational complexity of the IMODP algorithm is $O((T \times L) \times (M \times (L \times K)^2))$, that is, $O(M \times T \times L^3 \times K^2)$. Besides the four parameters, computational burden of IMODP also depends on dominating relationships among the solutions. To solve simple problems with few Pareto-optimal solutions involves small

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

computation burden; while to deal with complex problems with many Pareto-optimal solutions can involve considerable computation burden, which also requires a large K to retain information of the Pareto front.

HYPOTHETICAL TEST OF IMODP

IMODP is applicable to solve multiple-objective models with concave or non-concave trade-off curvatures. Using the concave and convex utility functions, this study designs three hypothetical cases to test the effectiveness of IMODP, in which the curvatures of trade-offs between objectives are concave, convex, and neither concave nor convex.

Test settings

The test is based on the power function $g = x^a$, where exponent a determines the function's rate of growth or decay. When $a > 0$, g is an increasing function that depends on x. Moreover, function g is concave when $1 > a > 0$, and it is convex when $a > 1$. The three multiple-objective optimization cases are designed by assigning the values of 0.5 and 2 to a, as shown in Table 1. The two objective functions are concave in case 1 and convex in case 2; in case 3, one objective function is convex, whereas the other is concave. We set three decision variables, namely, x_1, x_2, and x_3. x_1 and x_2 contributed to objectives 1 and 2, respectively; however, x_3 is a disturbing factor, which consumes available resources but does not contribute to either objective. For the three decision variables, a balance relationship of the total available resources exists, analogous to the water balance relationship of the reservoir operation. Lower and upper bounds of x exist, analogous to the capacity constraints.

For the three cases, objectives g_1 and g_2 are increasing functions of x_1 and x_2, respectively. Therefore, the Pareto front can be easily determined, that is, distributing the available resources between x_1 and x_2 and allocating no resource to x_3. The corresponding objectives of the Pareto front can then be determined, as shown in Table 2. Notably, the curvature of the trade-offs among the objectives are concave in case 1, convex in case 2, and neither concave nor convex in case 3.

Formulation and test of IMODP

The MODP for the three cases is set as follows:

(1) The state variable is defined as the cumulative consumption of resources (Liu *et al.* 2012), that is

$$s_{i+1} = s_1 + \sum_{j=1}^{i} x_j \tag{8}$$

Therefore, the state equation is

$$s_{i+1} = s_i + x_i \tag{9}$$

(2) The initial and ending storages are

$$\begin{cases} s_1 = s_{\text{ini}} = 0 \\ s_4 = s_{\text{end}} = 1 \end{cases} \tag{10}$$

(3) The recursive DP formulation is

$$\begin{pmatrix} G_{1,i}(s_i) \\ G_{2,_i}(s_i) \end{pmatrix} = \max \begin{pmatrix} g_{1,i}(x_i) + G_{1,i+1}(s_{i+1}) \\ g_{2,i}(x_i) + G_{1,i+1}(s_{i+1}) \end{pmatrix} \tag{11}$$

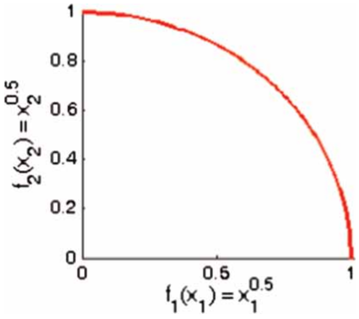
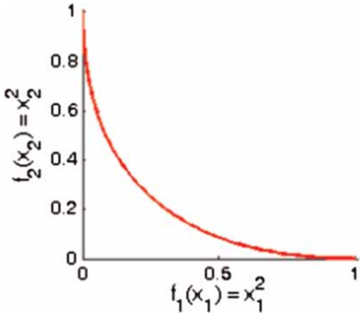
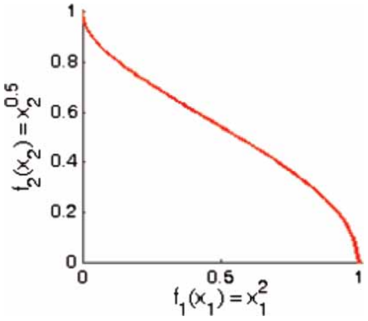
The range of the state variable s_i is [0, 1] as $x_1 + x_2 + x_3 = 1$. The study horizon is three and the backward induction consists of three steps. In step 3, objective values of state transition from s_3 to s_4, which assign resource

Table 1 | Objective functions and constraints of the three hypothetical cases

	Case 1	Case 2	Case 3
Objective functions	$\max \begin{cases} g_1 = x_1^{0.5} \\ g_2 = x_2^{0.5} \end{cases}$	$\max \begin{cases} g_1 = x_1^2 \\ g_2 = x_2^2 \end{cases}$	$\max \begin{cases} g_1 = x_1^2 \\ g_2 = x_2^{0.5} \end{cases}$
Constraints	$\begin{cases} x_1 + x_2 + x_3 = 1 \\ 0 \le x_1 \le 1 \\ 0 \le x_2 \le 1 \\ 0 \le x_3 \le 1 \end{cases}$		

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

Table 2 | Pareto front and the corresponding objectives of the three hypothetical cases

	Case 1	Case 2	Case 3
Pareto front	$\begin{cases} x_1 = \alpha \\ x_2 = 1 - \alpha \;\; (0 \le \alpha \le 1) \\ x_3 = 0 \end{cases}$		
Objectives of Pareto front	$\begin{cases} g_1 = \alpha^{0.5} \\ g_2 = (1-\alpha)^{0.5} \end{cases} (0 \le \alpha \le 1)$	$\begin{cases} g_1 = \alpha^{2} \\ g_2 = (1-\alpha)^{2} \end{cases} (0 \le \alpha \le 1)$	$\begin{cases} g_1 = \alpha^{2} \\ g_2 = (1-\alpha)^{0.5} \end{cases} (0 \le \alpha \le 1)$
Curvature of trade-off among objectives			

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

to disturbing factor x_3, are determined; then in steps 2 and 1, objective values of state transitions from s_2 to s_3 and from s_1 to s_2 are determined, respectively. Pareto-optimal solutions are selected in each step and updated in backward induction (Figures 1 and 2). Notably, in the IMODP computation, the lower bound of x_i is violated if $s_i > s_{i+1}$, and a penalty is applied to the objective functions in this case; the upper bound of x_i is automatically satisfied because both s_i and s_{i+1} are within [0, 1].

In IMODP, the number of objective functions M is two. For the number of state discretization L and the threshold of the maximum number of non-dominated objective vector K, two scenarios are designed:

(1) To discretize s_i into 10 intervals of equal width 0.1, resulting in 11 numerical Pareto-optimal solutions

$$\begin{cases} x_1 = \alpha \\ x_2 = 1 - \alpha \\ x_3 = 0 \end{cases}$$

($\alpha = 0.0, 0.1, 0.2, \ldots, 1.0$); L and K are set as 11.

(2) To set the width of the storage intervals as 0.01, resulting in 101 Pareto-optimal solutions

$$\begin{cases} x_1 = \alpha \\ x_2 = 1 - \alpha \\ x_3 = 0 \end{cases}$$

($\alpha = 0.00, 0.01, 0.02, \ldots, 1.00$); L and K are set as 101.

In the two scenarios, K is set equal to the maximum number of Pareto-optimal solutions. Therefore, IMODP is equivalent to MODP in the hypothetical test (Figure 2). The results of the two scenarios are shown in Figures 3 and 4. As can be seen, IMODP effectively captures the Pareto front for cases with concave, convex, and neither concave nor convex curvatures of the trade-offs among objective functions. The increase in L (more storage discretization) results in more accurate capture of the Pareto front and the non-linear trade-offs.

REAL-WORLD CASE

The hypothetical test demonstrates the ability of IMODP to identify the Pareto front for the cases with concave, convex, and neither concave nor convex trade-off curvatures. Furthermore, we design a case of hydropower reservoir operation with real-world data of Danjiangkou Reservoir (Zhao *et al.* 2012, 2014). The effect of K on computational efficiency of IMODP is analyzed.

Case study settings

Reservoir regulates streamflow variability and serves multiple operational targets. For hydropower reservoir operation, there are two important targets of total energy and firm energy (Tauxe *et al.* 1979; Liu *et al.* 2011, 2012). Firm energy is generally contracted with power grid, and it is sold at premium prices due to its 'firm' availability. On the other hand, there is a trade-off between firm energy and total energy. Due to the complementarity between reservoir storage and release, that is, increase of storage results in higher water head and makes release more productive, it is favourable to keep high reservoir storage (Zhao *et al.* 2014). However, firm energy forces reservoir to release the storage, which would reduce hydropower generation in subsequent periods, particularly during the dry season. Based on IMODP, we analyze the trade-offs between total energy and firm energy for Danjiangkou Reservoir.

Two objectives are set for the case of hydropower reservoir operation. The first objective is the maximization of total energy output

$$\text{Obj1:} \quad \max \sum_{t=1}^{T} p_t \Delta \tag{12}$$

and the second objective is the maximization of firm energy output

$$\text{Obj2:} \quad \max[\min(\, p_1 \quad p_2 \quad \ldots \quad p_T \,)] \tag{13}$$

In Equations (12) and (13), p_t and Δ denote the rate of single-period hydropower generation and the length of time period, respectively. p_t is calculated by

$$p_t = \min\left\{ \eta \left[\frac{(\mathrm{SSR}(s_t) + \mathrm{SSR}(s_{t+1})}{2} - \mathrm{SDR}(r_t)) \right] r_t, \, p_{\max} \right\} \tag{14}$$

In Equation (14), the unit of p_t is million watt (MW). η is the energy conversion coefficient, and its value is 0.00882. *SSR* and *SDR* respectively represent the stage storage

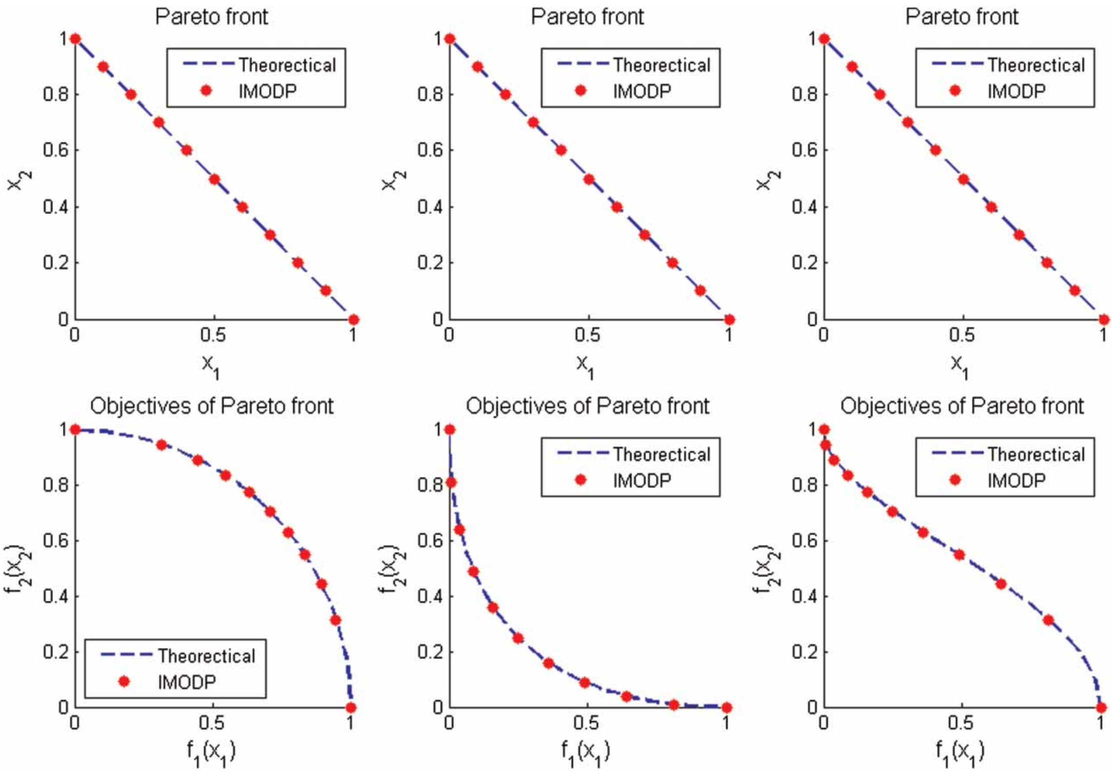


Figure 3 | Theoretical and numerical Pareto fronts (determined by IMODP with the storage state discretized into 10 intervals) and the corresponding objective values for case 1 (left column), case 2 (central column), and case 3 (right column).

relationship and stage discharge relationship of Danjiangkou Reservoir, and they are fitted by power functions, that is, $\mathrm{SSR}(s_t) = 4.5s_t^{0.29} + 77.7$ (storage s_t is in million m^3) and $\mathrm{SDR}(r_t) = 0.045r_t^{0.53} + 88.5$ (release r_t is in m^3/s) (Zhao *et al.* 2014). $p_{\max}$ is the turbine capacity and the value is 940 MW.

Based on Equations (12)–(14), the recursive function of MODP is as follows:

$$\begin{bmatrix} G_{1,t}(s_t) \\ G_{2,t}(s_t) \end{bmatrix} = \max \begin{bmatrix} p_t\Delta + G_{1,t+1}(s_{t+1}) \\ \min\left(p_t,\, G_{2,t+1}(s_{t+1})\right) \end{bmatrix}$$
$$s.t. \quad \begin{cases} s_t + (q_t - r_t)\Delta = s_{t+1} \\ s_{t+1} \geq \underline{s} \\ s_{t+1} \leq \overline{s} \\ r_t \geq \underline{r} \end{cases} \tag{15}$$

In Equation (15), $G_{1,t}(s_t)$ and $G_{2,t}(s_t)$ indicate cumulative hydropower generation and minimum of single-period hydropower generation from period t to the end of study horizon, respectively. Lower bound $\underline{s}$ and upper bound $\overline{s}$ of reservoir storage are set as 12,100 million m^3 and 17,450 million m^3, respectively. Lower bound $\underline{r}$ of reservoir release is set as 350 m^3/s. The case study analyses hydropower reservoir operation for the whole year from January to December and takes a time step of 1 month (Δ represents 720 hours in Equation (12) and 609,408,000 s in Equation (15)). The monthly reservoir inflow is presented in Figure 5. As can be seen, the main flood season last 3 months from July to September. For reservoir operation, both the initial storage s_1 at the beginning of January and the ending storage s_{13} at the end of December are set as 17,000 million m^3.

There are four parameters in IMODP, that is, M (number of objectives), T (length of study horizon), L (number of storage discretization), and K (number of Pareto-optimal solutions selected by the ranking technique). For the case study, M is 2 and T is 12. We set two scenarios of L, that is, 108 and 215, which respectively discretize state of reservoir storage into intervals of equal width of 50 and 25 million m^3. Five scenarios of 10, 20, 30, 40 and 50 are set for K, analyzing effect of K on computational efficiency

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

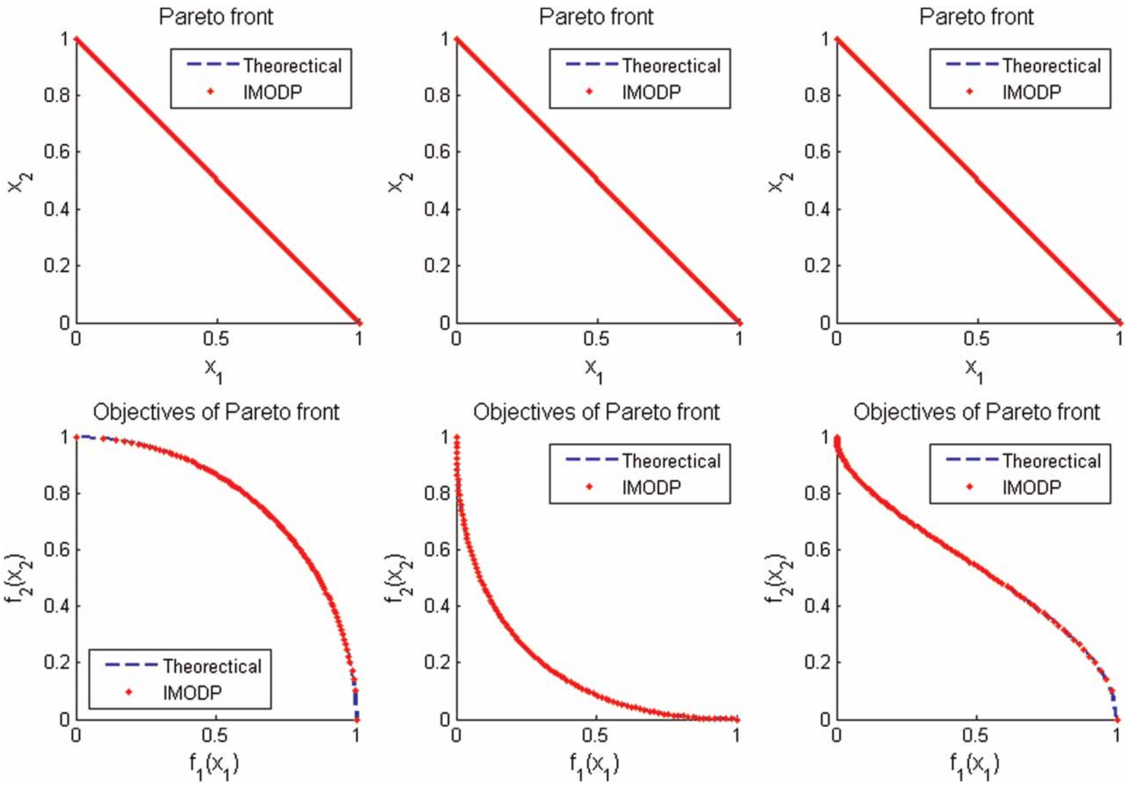


Figure 4 | Theoretical and numerical Pareto fronts (determined by IMODP with the storage state discretized into 100 intervals) and the corresponding objective values for case 1 (left column), case 2 (central column), and case 3 (right column).

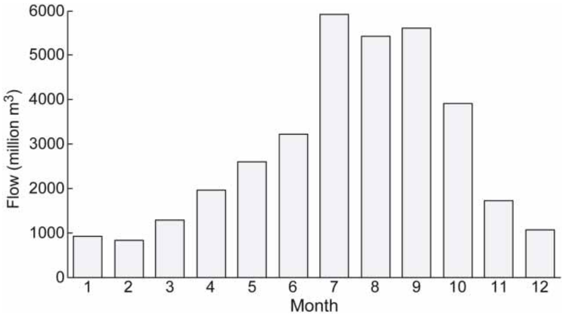


Figure 5 | Monthly reservoir inflow for the case study.

of IMODP. The algorithms are implemented with Matlab R2012b on a desktop with Intel (R) Xeon (R) CPU E3-1230, 8.00 GB of RAM. Besides IMODP, the constraint method is used to analyze trade-offs between total energy and firm energy. We incorporate the objective of firm energy $\underline{p} = \min(p_1 \ldots \quad p_T)$ into the constraint using $p_t \geq \underline{p}$. Twenty-one discrete values ranging from 230 to 330 MW are set for $\underline{p}$. In this way, trade-offs between total energy and firm energy are analyzed by the constraint method.

Result analysis

Based on the IMODP algorithm and the constraint method, we analyze the Pareto-optimal solutions and the corresponding objective vectors for the given streamflow scenario. Trade-offs between total energy and firm energy are shown Figure 6. As can be seen, maximization of total energy leads to minimization of firm energy, that is, [5.331×10^6 MW h, 204.8 MW]; whereas maximization of firm energy results in minimization of total energy, that is, [338.9 MW, 5.313×10^6 MW h]. By setting $K = 2$, IMODP favorably determines the optimal values of total energy and firm energy, which is owing to the ranking technique retaining the boundary solutions. The computing time is slightly longer than that of DP with the constraint method. As the value of K increases, trade-offs between total

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

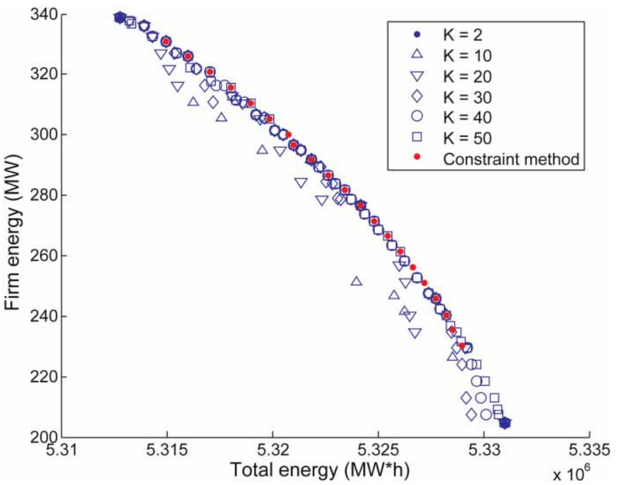


Figure 6 | IMODP-based trade-off analysis between total energy and firm energy ($L = 215$).

energy and firm energy are more accurately captured. When $K = 50$, non-dominated values of total energy and firm energy determined by IMODP match those determined by the constraint method.

The computing time of DP, MODP and IMODP are illustrated in Table 3. Comparing DP with the constraint method, the advantage of IMODP is that it deals directly with multiple objectives and retains the optimal value of individual objective. Notably, the application of the constraint method is strictly limited to cases where each component of the objective that is incorporated into constraints is determined by the aggregate level of the objective (Tauxe *et al.* 1979; Loucks & van Beek 2005). For example, firm energy can be incorporated into the constraint because $p_t (t = 1, 2, \ldots, T)$ are determined by $\underline{p}$, that is, $p_t \geq \underline{p} = \min(\ p_1 \quad p_2 \quad \ldots \quad p_T\ )$; whereas total energy cannot be considered as a constraint because $p_t (t = 1, 2, \ldots, T)$ collectively determine $\sum_{t=1}^{12} p_t \Delta$. Therefore, although DP with the constraint method requires less computing time in the case study, IMODP is applicable to more complicated cases where both objectives cannot be characterized by using a constraint.

When $L = 108$ (215), as K increases from 2 to 50, the computing time of IMODP increases from 14 to 1,096 (from 73 to 15,820) seconds because a larger K retains more trade-off information on the objectives and involves more computation. As regards MODP, which retains all Pareto-optimal solutions, when $L = 108$, the number of Pareto-optimal solutions goes beyond 10,000, and the algorithm cannot finish within 1 day; when $L = 215$, more Pareto-optimal solutions exist and even longer computing time is required. Compared with MODP, IMODP shows considerable improvement on computing time and provides satisfactory results. Notably, parameters of heuristic methods in MOEA can considerably affect computational efficiency, and their values are usually selected based on empirical evaluations (Reddy & Kumar 2007; Kim *et al.* 2008; Pianosi *et al.* 2013). For the case study, $K = 50$ shows satisfactory results. When applying IMODP to other cases, empirical evaluations are required for the selection of an appropriate K value. Nevertheless, the case study demonstrates that the value of K can be significantly smaller than the number of Pareto-optimal solutions, which considerably reduces computing time and yields satisfactory results.

The IMODP algorithm analyses trade-offs among objectives and determines Pareto-optimal decisions. Figure 7 illustrates operation decisions with maximum total energy and firm energy. As can be seen, there is an empty-refill cycle of reservoir storage in 1 year. Generally, reservoir storage is released for hydropower generation and emptied at the beginning of rainy season ($s_7 = \underline{s}$), and it is refilled during the rainy season ($s_{10} = \overline{s}$). Similarities exist among the decisions from July to December, due to there being ample inflow during the rainy season and that reservoir storage is at a high level at the beginning of the dry season.

Table 3 | Comparison of computing time (seconds) of DP, MODP and IMODP

			IMODP					
	DP	**MODP**	***K* = 2**	***K* = 10**	***K* = 20**	***K* = 30**	***K* = 40**	***K* = 50**
$L = 108$	1.5	Longer than 86,400	14	137	404	672	873	1,096
$L = 215$	6.2	Even longer than 86,400	73	1,085	3,676	7,639	11,949	15,820

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

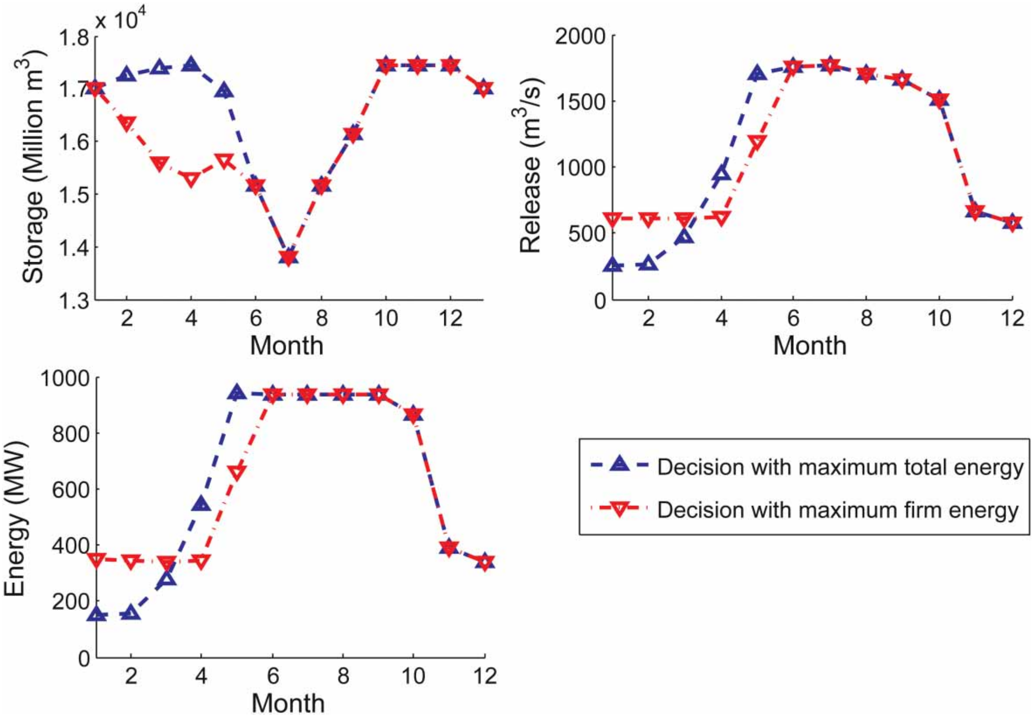


Figure 7 | Pareto-optimal decisions of storage, release in the case study of hydropower generation.

However, significant differences exist among the Pareto-optimal decisions during the empty process from January to June, indicating considerable trade-offs between total energy and firm energy. Notably, maximum total energy decision cuts back on reservoir releases early in the year to maintain reservoir storage at a high level, which provides higher water head and generates more energy (Zhao *et al.* 2014). The opposite is found on the maximum firm energy decision that draws down reservoir storage early on because of limited water availability, which indicates that firm energy output is at the cost of losses of water head and total energy output (Tauxe *et al.* 1979; Liu *et al.* 2011, 2012).

DISCUSSION AND CONCLUSIONS

DP is one of the most popular optimization models used in water resource studies. Adding to the category of DP studies, this study has developed an improved multiple-objective DP algorithm, that is, IMODP, for reservoir operation optimization. The IMODP algorithm is built on multiple-objective DP (Daellenbach & Dekluyver 1980) and the ranking technique (Deb *et al.* 2002). Compared with the weighting method and the constraint method, IMODP deals directly with multiple-objective problems and does not need to assign weights to the objectives or to incorporate some objectives into the constraints. The overall computational complexity of IMODP for a deterministic system with one state variable and one decision variable is $O(M \times T \times L^3 \times K^2)$ and the efficiency of IMODP depends on M (number of objectives), T (length of study horizon), L (number of storage discretization), and K (number of Pareto-optimal solutions selected by the ranking technique).

We design hypothetical and real-world tests to examine the effectiveness of IMODP. The hypothetical test consists of three cases with concave, convex, and neither concave nor convex trade-off curvatures. The results show that IMODP effectively captures the Pareto front of operation decisions and determines the corresponding non-dominated objective vectors. The real-world case focuses on the trade-off between total energy and firm energy for hydropower reservoir operation. IMODP efficiently determines the optimal decisions of individual objectives, as well as compromise decisions. The results highlight the importance of

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

the selection of a proper K in applications of IMODP, which involves trade-offs between computing time and accuracy.

The MODP and IMODP algorithms deal with deterministic single-reservoir problems. In real-world reservoir operations, one important issue is hydrological uncertainty because future stream flows cannot be perfectly predicted. Future efforts can combine IMODP with implicit stochastic programming, determining Pareto-optimal solutions for many stream flow scenarios and deriving operating rules to support decision making under uncertainty. Another major challenge in real-world reservoir operation is multiple reservoirs, which result in heavy computational burden. Further studies can combine MODP and IMODP with advanced computation methods and apply the algorithms to more complex reservoir operation problems.

ACKNOWLEDGEMENTS

We are grateful to the editor and the three anonymous reviewers for their constructive suggestions, which have facilitated major improvements in this paper. This research was supported by the Ministry of Science and Technology of China (Projects 2013BAB05B03 and 2011BAC09B07), the National Natural Science Foundation of China (Projects 51179085 and 91125018), and the China Postdoctoral Science Foundation.

REFERENCES

Akbari, M., Afshar, A. & Mousavi, S. J. 2011 Stochastic multiobjective reservoir operation under imprecise objectives: multicriteria decision-making approach. *J. Hydroinform.* **13** (1), 110–120.

Archibald, T. W., McKinnon, K. I. M. & Thomas, L. C. 1997 An aggregate stochastic dynamic programming model of multireservoir systems. *Water Resour. Res.* **33** (2), 333–340.

Castelletti, A., Pianosi, F., Soncini-Sessa, R. & Antenucci, J. P. 2010a A multiobjective response surface approach for improved water quality planning in lakes and reservoirs. *Water Resour. Res.* **46**, W06502, doi:10.1029/2009WR008389.

Castelletti, A., Galelli, S., Restelli, M. & Soncini-Sessa, R. 2010b Tree-based reinforcement learning for optimal water reservoir operation. *Water Resour. Res.* **46**, W09507, doi:10.1029/2009WR008898.

Castelletti, A., Antenucci, J. P., Limosani, D., Thi, X. Q. & Soncini-Sessa, R. 2011 Interactive response surface approaches using computationally intensive models for multiobjective planning of lake water quality remediation. *Water Resour. Res.* **47**, W09534, doi:10.1029/2011WR010552.

Cheng, C. T. & Chau, K. W. 2004 Flood control management system for reservoirs. *Environ. Model. Softw.* **19** (12), 1141–1150.

Daellenbach, H. G. & Dekluyver, C. A. 1980 Note on multiple objective dynamic-programming. *J. Oper. Res. Soc.* **31** (7), 591–594.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. 2002 A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6** (2), 182–197.

Draper, A. J. & Lund, J. R. 2004 Optimal hedging and carryover storage value. *J. Water Resour. Plann. Manage. ASCE* **130** (1), 83–87.

Galelli, S. & Soncini-Sessa, R. 2010 Combining metamodelling and stochastic dynamic programming for the design of reservoir release policies. *Environ. Model. Softw.* **25** (2), 209–222.

Grune, L. & Semmler, W. 2004 Using dynamic programming with adaptive grid scheme for optimal control problems in economics. *J. Econ. Dyn. Control* **28** (12), 2427–2456.

Kim, T., Heo, J. H., Bae, D. H. & Kim, J. H. 2008 Single-reservoir operating rules for a year using multiobjective genetic algorithm. *J. Hydroinform.* **10** (2), 163–179.

Kuhn, M. 2006 *Notes on Numerical Dynamic Programming in Economic Applications*. CDSEM, University of Mannheim, Denmark.

Labadie, J. W. 2004 Optimal operation of multireservoir systems: state-of-the-art review. *J. Water Resour. Plann. Manage. ASCE* **130** (2), 93–111.

Lee, J. H. & Labadie, J. W. 2007 Stochastic optimization of multireservoir systems via reinforcement learning. *Water Resour. Res.* **43**, W11408, doi:10.1029/2006WR005627.

Li, X. A., Guo, S. L., Liu, P. & Chen, G. Y. 2010 Dynamic control of flood limited water level for reservoir operation by considering inflow uncertainty. *J. Hydrol.* **391** (1–2), 126–134.

Liu, P., Cai, X. M. & Guo, S. L. 2011 Deriving multiple near-optimal solutions to deterministic reservoir operation problems. *Water Resour. Res.* **47**, W08506, doi:10.1029/2011WR010998.

Liu, P., Nguyen, T. D., Cai, X. M. & Jiang, X. H. 2012 Finding multiple optimal solutions to optimal load distribution problem in hydropower plant. *Energies* **5** (5), 1413–1432.

Lotov, A., Bushenkov, V. & Kamenev, G. 2004 *Interactive Decision Maps Approximation and Visualization of Pareto-Frontier*. Springer-Verlag, Heidelberg.

Loucks, D. P. & van Beek, E. 2005 *Water Resources Systems Planning and Management: An Introduction to Methods, Models and Applications*. UNESCO, Paris.

Lund, J. R. & Guzman, J. 1999 Derived operating rules for reservoirs in series or in parallel. *J. Water Resour. Plann. Manage. ASCE* **125** (3), 143–153.

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest

McMahon, G. F. & Farmer, M. C. 2004 Reallocation of federal multipurpose reservoirs: principles, policy, and practice. *J. Water Resour. Plann. Manage. ASCE* **130** (3), 187–197.

Nicklow, J., Reed, P., Savic, D., Dessalegne, T., Harrell, L., Chan-Hilton, A., Karamouz, M., Minsker, B., Ostfeld, A., Singh, A. & Zechman, E. 2010 State of the art for genetic algorithms and beyond in water resources planning and management. *J. Water Resour. Plann. Manage. ASCE* **136** (4), 412–432.

Perez-Diaz, J. I., Wilhelmi, J. R. & Arevalo, L. A. 2010 Optimal short-term operation schedule of a hydropower plant in a competitive electricity market. *Energy Convers. Manage.* **51** (12), 2955–2966.

Pianosi, F., Castelletti, A. & Restelli, M. 2013 Tree-based fitted Q-iteration for multi-objective Markov decision processes in water resource management. *J. Hydroinform.* **15** (2), 258–270.

Reddy, M. J. & Kumar, D. N. 2007 Multiobjective differential evolution with application to reservoir system optimization. *J. Comput. Civil Eng.* **21** (2), 136–146.

Rosenman, M. A. & Gero, J. S. 1985 Reducing the pareto optimal set in multicriteria optimization (with applications to pareto optimal dynamic-programming). *Eng. Optim.* **8** (3), 189–206.

Rust, J. 1997 Using randomization to break the curse of dimensionality. *Econometrica* **65** (3), 487–516.

Suen, J. P. & Eheart, J. W. 2006 Reservoir management to balance ecosystem and human needs: Incorporating the paradigm of the ecological flow regime. *Water Resour. Res.* **42**, W03417, doi:10.1029/2005WR004314.

Tauxe, G. W., Inman, R. R. & Mades, D. M. 1979 Multiobjective dynamic-programming with application to a reservoir. *Water Resour. Res.* **15** (6), 1403–1408.

Wang, L. & Singh, C. 2007 Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm. *Electr. Power Syst. Res.* **77** (12), 1654–1664.

Yakowitz, S. 1982 Dynamic-programming applications in water-resources. *Water Resour. Res.* **18** (4), 673–696.

Zhao, J. S., Cai, X. M. & Wang, Z. J. 2011 Optimality conditions for a two-stage reservoir operation problem. *Water Resour. Res.* **47**, W08503, doi:10.1029/2010WR009971.

Zhao, T. T. G., Cai, X. M., Lei, X. H. & Wang, H. 2012 Improved dynamic programming for reservoir operation optimization with a concave objective function. *J. Water Resour. Plann. Manage. ASCE* **138** (6), 590–596.

Zhao, T., Zhao, J. & Yang, D. 2014 Improved dynamic programming for hydropower reservoir operation. *J. Water Resour. Plann. Manage. ASCE* **140** (3), 365–374.

First received 3 January 2013; accepted in revised form 25 February 2014. Available online 27 March 2014

Downloaded from https://iwaponline.com/jh/article-pdf/16/5/1142/387410/1142.pdf
by guest