

## Research Article

# Operational Optimization of Large-Scale Parallel-Unit SWRO Desalination Plant Using Differential Evolution Algorithm

Jian Wang,<sup>1,2</sup> Xiaolong Wang,<sup>2</sup> Aipeng Jiang,<sup>2</sup> Shu Jiangzhou,<sup>2</sup> and Ping Li<sup>1</sup>

<sup>1</sup> School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China

<sup>2</sup> Institute of Energy Utilization & Automation, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Ping Li; [pli@iipc.zju.edu.cn](mailto:pli@iipc.zju.edu.cn)

Received 15 November 2013; Accepted 30 December 2013; Published 17 February 2014

Academic Editors: Z. Cui and X. Yang

Copyright © 2014 Jian Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A large-scale parallel-unit seawater reverse osmosis desalination plant contains many reverse osmosis (RO) units. If the operating conditions change, these RO units will not work at the optimal design points which are computed before the plant is built. The operational optimization problem (OOP) of the plant is to find out a scheduling of operation to minimize the total running cost when the change happens. In this paper, the OOP is modelled as a mixed-integer nonlinear programming problem. A two-stage differential evolution algorithm is proposed to solve this OOP. Experimental results show that the proposed method is satisfactory in solution quality.

## 1. Introduction

The shortage of fresh water has become a bottleneck of the economic development in many countries. Seawater desalination is an effective way to solve this problem. Reverse osmosis (RO) desalinating is one of the most popular ways to generate freshwater from seawater and has made a rapid progress over the past four decades [1–3]. The scale of seawater reverse osmosis (SWRO) desalination plant is continually expanding, whose capacity of freshwater has exceeded 100,000 tons per day in recent years.

Now, a kind of large-scale parallel-unit SWRO desalination plant, which is composed of multiple parallel RO units, has appeared. This kind of plant has huge capital cost and more complicated operation processes. So, before it is built, an optimal design is made to select the suitable devices and system performance to match the operating condition [4–8]. These optimal designs are made based on static condition [9, 10], but the actual situation is changing [11]. For example, the seawater temperature varies with changing seasons, the freshwater supply changes according to the user's demand, the permeate rate is declining with the membrane fouling [12], and so on. The result is that these devices would not work at the optimal designed points in most time. Therefore,

an operational optimization scheduling is necessary to make these machines work in a best way under the changed conditions.

In this paper, a mathematical model of operational optimization problem (OOP) for a large-scale parallel-unit SWRO desalination plant, which includes objective function and constraint functions, is made. In order to solve this OOP, a two-stage differential evolution (TSDE) algorithm is proposed. The simulating results show that the TSDE is excellent in searching ability than basic DE and genetic algorithm (GA).

## 2. Problem Description and Formulation

**2.1. SWRO Desalination Plant Representation.** The single SWRO desalination unit is a multistage process, which includes seawater intake, pretreatment, RO desalination, and posttreatment [13]. Figure 1 shows the schematic diagram of the unit.

In the intake stage, the raw seawater is pumped from a deep well, which is located close to the shoreline, to the flocculators to filter; in the pretreatment stage, the most suspended matters and colloids are filtered out of seawater

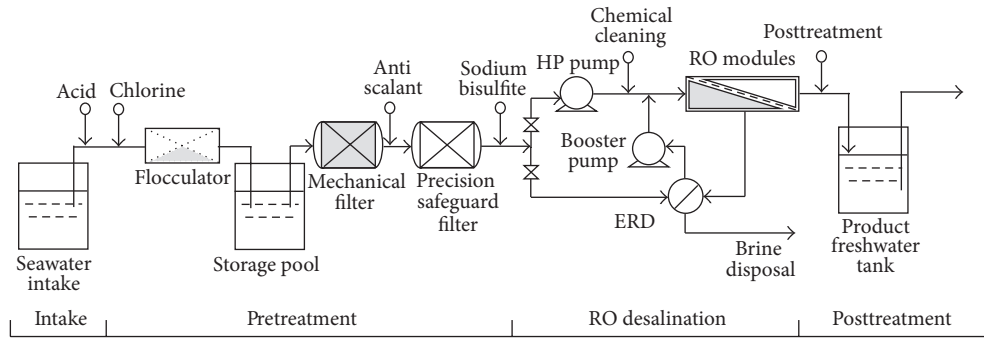


FIGURE 1: Schematic diagram of a single SWRO desalination unit.

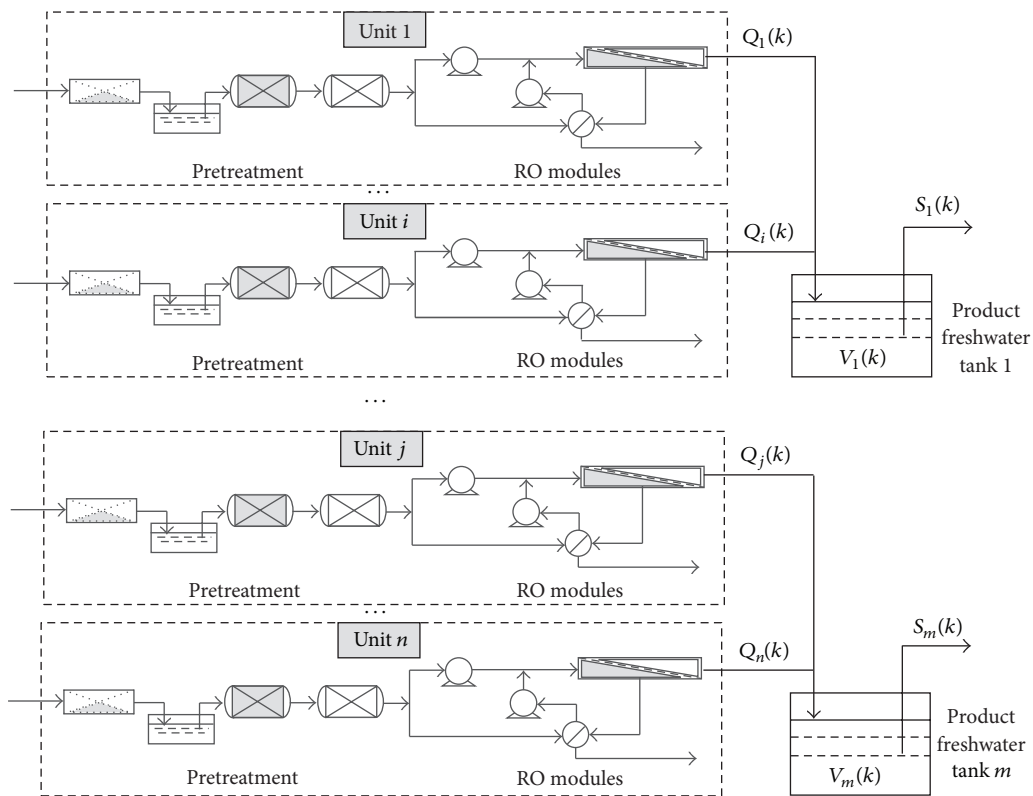


FIGURE 2: Schematic diagram of a parallel-unit SWRO desalination plant.

by flocculators, mechanical filters, and precision safeguard filters successively to ensure that SDI (silt density index) is lower than 5 to meet the RO modules requirement; in the RO desalination stage, one part of the fed seawater is pressurized by the high pressure (HP) pump and the other by energy recovery device (ERD) and booster pump up to 40 ~ 50 bar and then is desalinated by RO modules; in the posttreatment stage, the produced freshwater flows into the product freshwater tanks (PFWT) to supply into the municipal water network; the brine is disposed reasonably.

The schematic diagram of a large-scale parallel-unit SWRO desalination plant is shown in Figure 2, which is structured with a number of independent RO units in parallel.

Each RO unit has different permeate rate. The freshwater, produced by RO Unit  $i$  ( $i \in n_j$ ), flows into PFWT  $j$  and then is supplied for user.

**2.2. Problem Mathematical Formulation.** The OOP of a large-scale SWRO desalination plant is to make an optimal scheduling plan to minimize the plant's total running cost (TRC). This optimal schedule plan determines the on/off status and amount of generated freshwater of each RO unit, and the amount of supplied freshwater by each PFWT at each time period. When computing the TRC, the price factors, such as time-of-use electricity price, the operation cost, and the maintenance cost, are taken into consideration.

**2.2.1. Objective Function of OOP.** A lumped parameter model of this problem is built in this paper. The TRC of the plant consists of capital depreciation cost (CDC), operating cost (OC), labor and chemical cost (LCC), and energy cost (EC), which is presented as

$$\min \text{TRC} = \text{CDC} + \text{OC} + \text{LCC} + \text{EC}. \quad (1)$$

(1) *The Capital Depreciation Cost, CDC, is*

$$\text{CDC} = \text{CC} \times 1.411 \times \eta \times T. \quad (2)$$

The CDC is presented as (2), where CC is the capital cost of the SWRO desalination plant, 1.411 is the coefficient that is used to calculate the practical investment,  $\eta$  is the capital charge rate, which is usually a constant, and  $T$  is the operational optimization periods. When the SWRO desalination plant has been built, the CC is a fixed value; therefore, the CDC is a fixed cost in a certain time period  $T$  under a constant capital charge rate  $\eta$ . So, when computing the minimum value of TRC in this condition, this part cost CDC can be ignored. When the other parts of TRC reach their minimum values, the TRC gets its optimal value by adding this constant cost.

(2) *The Operating Cost, OC.* The OC includes maintenance expense and the repair and replacement expense of devices. The maintenance expense refers to the conventional maintenance expense during machines running; the repair and replacement expense refers to the cost of repair and replacement when the equipment stops. The OC is presented as

$$\text{OC} = \sum_{k=1}^K \sum_{i=1}^n M_i(k), \quad (3)$$

where  $K$  is the time horizon of the operational optimization;  $n$  is the number of RO units in the plant;  $M_i(k)$  is the operating cost of Unit  $i$  at the time  $k$ , which is presented as

$$M_i(k) = \begin{cases} C_1 \times Q_i(k), & \alpha_i(k) = 1, \\ C_2, & \alpha_i(k) = 0, \end{cases} \quad (4)$$

where  $C_1$  is the coefficient that is used to calculate the maintenance cost;  $Q_i(k)$  is the amount of freshwater generated by Unit  $i$  at time  $k$ ;  $C_2$  is the repair and replacement expense when Unit  $i$  stops at time  $k$ ;  $\alpha_i(k)$  is the on/off status of Unit  $i$  at time  $k$ ;  $\alpha_i(k) = 1$  when Unit  $i$  is running;  $\alpha_i(k) = 0$  when it stops.

(3) *The Labor and Chemical Cost, LCC.* Usually, we consider  $\text{LCC} = 12\% \times \text{TRC}$ . This cost includes labor cost, chemical expense, and other expenses. When TRC is the smallest, LCC has its minimum value.

(4) *The Energy Cost, EC.* The energy cost is the energy consumption which is necessary to drive the devices. It is

obviously variable in different scheduling plans. By optimizing the scheduling plan, the plant's energy consumption can achieve the minimum value. The EC is presented as

$$\text{EC} = \sum_{k=1}^K \left( P_e(k) \sum_{i=1}^n (\alpha_i(k) \times C_3 \times Q_i(k)) \right), \quad (5)$$

where  $C_3$  is the coefficient between energy and amount of freshwater generated by Unit  $i$  at time  $k$ ;  $P_e(k)$  is time-of-use electricity price at time  $k$ .

**2.2.2. Constraints of OOP.** The constraints of OPP include technical limitations and the design requirements, which are as follows.

(1) *The Amount of Freshwater,  $Q_i(k)$ .*  $Q_i(k)$  is the amount of freshwater generated by Unit  $i$  at time  $k$ , which is subject to

$$Q_{i,\min} \leq Q_i(k) \leq Q_{i,\max}, \quad i = 1, 2, \dots, n, \quad \forall k, \quad (6)$$

where  $Q_{i,\max}$ ,  $Q_{i,\min}$  are the upper and the lower limit of the amount of freshwater generated by Unit  $i$ , respectively.

(2) *The Capacity of Each PFWT,  $V_j(k)$ .*  $V_j(k)$  is the capacity of PFWT  $j$  at time  $k$ , which is subject to

$$V_{j,\min} \leq V_j(k-1) + \sum_{i=1}^{n_j} \alpha_i(k) \cdot Q_i(k) - S_j(k) \leq V_{j,\max}, \quad (7)$$

$$j = 1, 2, \dots, m; \quad i = 1, 2, \dots, n_j, \quad \forall k,$$

where  $V_j(k-1)$  is the capacity of PFWT  $j$  at the beginning of time  $k$ ;  $m$  is the number of PFWT;  $n_j$  is the number of RO units which feed freshwater to the PFWT  $j$ ;  $S_j(k)$  is the amount of supplied freshwater by PFWT  $j$  at time  $k$ ;  $V_{j,\max}$ ,  $V_{j,\min}$  are the upper limit and the lower limit of capacity of PFWT  $j$ .

(3) *The Amount of Supplied Freshwater,  $S_j(k)$ .*  $S_j(k)$  is the amount of supplied freshwater by PFWT  $j$  at time  $k$ , which is subject to

$$\sum_{j=1}^m S_j(k) = D(k), \quad \forall k, \quad (8)$$

$$S_j(k) \geq 0, \quad j = 1, 2, \dots, m, \quad \forall k.$$

### 3. Two-Stage Differential Evolution

The OOP of the large-scale parallel-unit SWRO desalination plant is a mixed-integer nonlinear programming problem (MINLP) over the time horizon. In the OOP,  $\{0, 1\}$  binary variables represent the on/off statuses of RO units; the real variables represent the amounts of freshwater generated by RO units and supplied freshwater by PFWTs. In order to compute this OOP, a novel differential evolution, TSDE, is proposed in this paper, and the basic steps of this TSDE are addressed in the following subsections.

**3.1. Basic DE.** Differential evolution algorithm was originally proposed by Storn and Price for solving continuous optimization in the mid-1990s [14–18]. It is an evolutionary algorithm, including three important operators: mutation, crossover, and selection [19, 20]. The basic DE works are as follows.

**Step 1 (initialization).** The original population is generated in the search space randomly, which contains NP individual vectors:  $\mathbf{x}_{i,1} = [x_{i1,1}, x_{i2,1}, \dots, x_{iD,1}]$ ,  $i = 1, 2, \dots, \text{NP}$ , where  $D$  is the dimension of individual.

**Step 2 (mutation).** For each individual vector  $\mathbf{x}_{i,G}$ , a mutated solution  $\mathbf{v}_{i,G}$  is created according to a DE/rand/1/bin mutate operator (9) in each generation  $G$  [15]. Consider

$$\mathbf{v}_{i,G} = \mathbf{x}_{r_1,G} + F \times (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}), \quad (9)$$

where  $G$  is the current generation number,  $F$  is a scale factor, and  $r_1, r_2$ , and  $r_3$  are randomly selected integers from the set  $\{1, 2, \dots, \text{NP}\}$  ( $r_1 \neq r_2 \neq r_3 \neq i$ ).

**Step 3 (crossover).** The crossover operator generates an offspring  $\mathbf{u}_{i,G}$  according to

$$u_{ij,G} = \begin{cases} v_{ij,G}, & \text{if } (\text{rand}(j) \leq \text{CR}) \text{ or } j = \text{rnbr}(j), \\ x_{ij,G}, & \text{otherwise} \end{cases} \quad (10)$$

$$(i = 1, 2, \dots, \text{NP}; j = 1, 2, \dots, D),$$

where CR is a crossover rate;  $\text{rnbr}(j)$  is a randomly chosen index ( $\text{rnbr}(j) \in \{1, 2, \dots, D\}$ ), which ensures  $\mathbf{u}_{i,G}$  getting at least one component from  $\mathbf{v}_{i,G}$ .

**Step 4 (selection).** The selection operator is to generate next population  $\mathbf{x}_{i,G+1}$  according to (11). The objective function of  $\mathbf{x}_{i,G}$  is compared to one of  $\mathbf{u}_{i,G}$  and the smaller one is selected as the next generation. Consider

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G}, & \text{if } (f(\mathbf{u}_{i,G}) \leq f(\mathbf{x}_{i,G})), \\ \mathbf{x}_{i,G}, & \text{otherwise} \end{cases} \quad (11)$$

$$(i = 1, 2, \dots, \text{NP}).$$

**Step 5 (stopping criterion).** If the stopping criterion (maximum number of iterations) is satisfied, computation is terminated; otherwise, Steps 3–5 are repeated.

**3.2. Treatment of Constraints.** In the OOP, there are two types of constraints: boundary constraints and technical limited constraint functions. The following is the treatment strategies of them.

**3.2.1. Boundary Constraints.** Sometimes  $\mathbf{u}_{i,G}$  is out of the range of search space. When it happens, it is necessary to

replace this value to guarantee it is into its allowed range (12). Consider

$$u_{ij,G} = \begin{cases} x_{ij}^L + \text{rand} \cdot (x_{ij}^U - x_{ij}^L) & \text{if } (u_{ij,G} < x_{ij}^L), \text{ or } (u_{ij,G} > x_{ij}^U) \\ u_{ij,G}, & \text{otherwise} \end{cases} \quad (12)$$

$$i = 1, 2, \dots, \text{NP}; j = 1, 2, \dots, D,$$

where  $x_{ij}^L$  is the lower bound of  $u_{ij,G}$ ;  $x_{ij}^U$  is the upper bound of  $u_{ij,G}$ .

**3.2.2. Constraint Functions.** Penalty function is one of the most effective methods to solve the evolutionary constraint optimization problem [21–25].

In this paper, the MINLP with constraints (13) is converted into an unconstrained MINLP by using a penalty function shown as (14). Consider

$$\begin{aligned} \min \quad & f(x, y) \\ & g_k(x, y) \leq 0, \quad k = 1, 2, \dots, p \\ & h_l(x, y) = 0, \quad l = 1, 2, \dots, q \\ \text{s.t.} \quad & x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \dots, D \\ & y_k \in \{0, 1\}, \quad k = 1, 2, \dots, m, \end{aligned} \quad (13)$$

$$\min F(x, y) = f(x, y) + M$$

$$\times \left[ \sum_{k=1}^p \max\{0, g_k(x, y)\}^2 + \sum_{l=1}^q \max\{0, |h_l(x, y) - \varepsilon|\}^2 \right], \quad (14)$$

where  $x = [x_1, x_2, \dots, x_n]^T$  is a continuous vector;  $y = [y_1, y_2, \dots, y_m]^T$  is a binary vector;  $g_k(x, y)$  represents the  $k$ th inequality constraint;  $h_l(x, y)$  represents the  $l$ th equality constraint;  $F(x, y)$  represents new objective function, which consists of two parts: the old objective function  $f(x, y)$  and a penalty function;  $p$  is the number of inequality constraints;  $q$  is the number of equality constraints;  $\varepsilon$  is a small positive constant, so that the  $l$ th equality constraint  $h_l(x, y) = 0$  is converted into the inequality constraint  $h_l(x, y) - \varepsilon < 0$ . In addition,  $M$  is defined as penalty coefficient, which is a large positive constant, so that it imposes penalty on unfeasible solutions.

**3.3. Conventional Technique for Binary Variables.** In this paper, the  $\{0, 1\}$  binary variables represent on/off status of RO units, but the DE algorithm is only capable of handling continuous variables. Therefore, some real variables within the range of  $[0, 1]$  are used to represent the statuses of RO units in TSDE instead of these binary variables. When the objective

function and the constraint functions are calculated, these real variables are rounded off to the nearest integer with (15). Consider

$$\bar{y}_i = \text{INT}(y_i), \quad (15)$$

where  $\text{INT}()$  is a function to convert a real number to an integer value by rounding off.

**3.4. Two-Stage DE.** The OOP of a large-scale parallel-unit SWRP desalination plant has such features: the permeate rate of each RO unit is huge; that is, the amount of freshwater generated by each RO unit is usually more than 10000 tons a day; the permeate rate changes within an allowable range, but this range is much smaller than its amount. So, when a RO unit's status changes, the TRC of the plant will change sharply. Usually, this change cannot be made up by adjusting the RO unit's permeate rate within its allowable range.

In this paper, a novel DE, TSDE, is proposed to solve the OOP. The TSDE is divided into two periods: Stage One and Stage Two. In Stage One, the permeate rate of each RO unit is supposed be a constant, which equals the median of its allowable range. A DE algorithm is used to compute the run/off status of each RO unit in this stage. When the DE is satisfying the stopping criterion, a preliminary scheduling will be worked out.

Then, the best 30 percent individuals in Stage One remain to Stage Two. The other 70 percent individuals in Stage Two are generated in the search space randomly (here, these two stages have the same population size NP). All these individuals are as the original values of another DE algorithm to evolve again in Stage Two. When the second DE stops, a final scheduling will be got.

## 4. Experimentation

For numerical experimentation, a large-scale parallel-unit SWRO desalination plant in Liuheng Island, China, which has the capacity of 100,000 m<sup>3</sup> freshwater a day, is considered over a 24-hour time horizon. The basic parameters of this OOP are shown in Tables 1, 2, and 3.

Here, the length of individual  $D = 20$ ; each individual consists of 20 variables. The first 8 variables are  $\{0, 1\}$  binary variables to represent the on/off statuses of 8 RO units. The next 8 variables are real values, which are the amounts of freshwater generated by 8 RO units. And the last 4 variables are real values to represent the supplied freshwater by 4 PFWTs.

The CDC of this plant is considered as a constant and ignored when calculating the TRC.

**4.1. Parameters of TSDE.** The DE's search ability for different problems depends on its parameters [26]. So, before it is working, these parameters must be tuned.

**4.1.1. The Maximum Number of Iterations  $G_m$  and the Population Size NP.** Firstly, the effects of two important parameters, the maximum number of iterations  $G_m$  and the population

TABLE 1: The basic parameters of OOP in Liuheng SWRO desalination plant.

Parameters	Sign	Value
The number of RO units	$n$	8
The number of PFWTs	$m$	4
	$n_1$	2
The number of RO units which feed freshwater to each PFWT	$n_2$	2
	$n_3$	2
	$n_4$	2
	$Q_{1,\max}$	460
	$Q_{2,\max}$	460
	$Q_{3,\max}$	570
The upper limit of the permeate rate of RO Unit $i$ , (m <sup>3</sup> /h)	$Q_{4,\max}$	570
	$Q_{5,\max}$	570
	$Q_{6,\max}$	800
	$Q_{7,\max}$	570
	$Q_{8,\max}$	570
	$Q_{1,\min}$	380
	$Q_{2,\min}$	380
	$Q_{3,\min}$	470
The lower limit of the permeate rate of RO Unit $i$ , (m <sup>3</sup> /h)	$Q_{4,\min}$	470
	$Q_{5,\min}$	470
	$Q_{6,\min}$	655
	$Q_{7,\min}$	470
	$Q_{8,\min}$	470
	$V_{1,0}$	340
The initial value of PFWT (m <sup>3</sup> )	$V_{2,0}$	340
	$V_{3,0}$	340
	$V_{4,0}$	340
	$V_{1,\max}$	1680
The maximum capacity of PFWT $i$ , (m <sup>3</sup> )	$V_{2,\max}$	1680
	$V_{3,\max}$	2280
	$V_{4,\max}$	1680
	$V_{1,\min}$	320
The minimum capacity of PFWT $i$ , (m <sup>3</sup> )	$V_{2,\min}$	320
	$V_{3,\min}$	320
	$V_{4,\min}$	320
The maintenance cost coefficient	$C_1$	11.5
The repair and replacement fees	$C_2$	155
The correlation coefficient of energy consumption and generated freshwater	$C_3$	2.86

size NP, on the search ability of DE for the OOP are explored. A basic DE is used to study the relationship, setting  $F = \text{Rand}[0.1 : 0.2]$ ,  $\text{CR} = \text{Rand}[0.7 : 0.9]$ , where  $\text{Rand}[a : b]$  represents a uniformly distributed random value that ranges from  $a$  to  $b$ .

The different pairs  $\{G_m, \text{NP}\}$  for DE are used to compute the solution of the OOP, and each algorithm is running 10



TABLE 2: The user's demands for freshwater  $D(k)$ .

Time $k$	Demands for flash water ( $\text{m}^3$ )	Time $k$	Demands for flash water ( $\text{m}^3$ )
1	1660	13	2525
2	1520	14	2430
3	1345	15	2300
4	1370	16	2490
5	1865	17	2765
6	2235	18	3030
7	2700	19	3095
8	2830	20	2970
9	2770	21	2655
10	2555	22	2350
11	2705	23	2085
12	2640	24	1870

TABLE 3: The time-of-use electricity price  $P_e(k)$ .

Time $k$	1-8	9-11	12-13	14-19	20-21	22	23-24
Electricity price ( $\text{¥/kWh}$ )	0.27	0.69	0.27	0.69	0.89	0.69	0.27

times. The results are listed in Table 4, in which the bold font is the best solution. It can be observed that the bigger these two parameters are, the stronger the search ability is.

It is worth noting that if  $G_m$  and NP are too big, the search speed of DE will decline sharply; and if the NP is small to a certain value, the algorithm is easy to fall into the locally optimal solution. Therefore, there is a compromise between the search speed and the accuracy when turning these two parameters.

**4.1.2. The Scale Factor  $F$  and Crossover Rate  $CR$ .** The scale factor  $F$  and crossover rate  $CR$  are another two important parameters in DE; the search ability of DE is sensitive to them too. In order to find out the effects of  $F$  and  $CR$  on the search ability of DE in this problem, a sensitivity analysis of these two parameters is presented.

Usually the  $F$  is in the range of  $[0.0, 1.0]$  and  $CR$  is in the range of  $[0.0, 1.0]$  [15]. In order to compare the effects of different  $F$  and  $CR$  fairly, the same initial individuals are used to calculate the optimal values of this OOP under different  $F$  and  $CR$ . The population size  $NP = 100$  and the maximum number of iterations  $G_m = 1000$ . A basic DE is used to do this job, and the results are shown in Figures 3 and 4.

In Figure 3, the effects of different  $CR$  on the search ability of DE when  $F = 0.1, 0.3, 0.5, 0.8, 1.0$ , and  $\text{Rand}[0.1:0.3]$ , respectively, are explored, in which it is found that OOP gets its best solution when  $CR \geq 0.7$ . In Figure 4, the effects of different  $F$  on the search ability of DE when  $CR = 0.1, 0.3, 0.5, 0.8, 1.0$ , and  $\text{Rand}[0.7:0.9]$ , respectively, are studied, and it is found that OOP gets its minimal value when  $F \leq 0.3$ . So, we set  $F = \text{Rand}[0.1:0.3]$  and  $CR = \text{Rand}[0.7:0.9]$  for DE algorithm to compute the best solutions of OOP in the following sections.

**4.1.3. The Two Maximum Numbers of Iterations  $G_1$  and  $G_2$  of TSDE.** As mentioned in the above sections, the TSDE algorithm is divided into two stages. The two maximum iteration numbers of these two stages,  $G_1$  and  $G_2$ , must be determined before the TSDE working. Here, it is supposed that  $G_1 + G_2 = 1000$ ,  $NP = 100$ ,  $F = \text{Rand}[0.1:0.3]$ , and  $CR = \text{Rand}[0.7:0.9]$ , and each algorithm runs 10 times. The results are listed in Table 5, in which the bold font is the best solution. It is known that the TSDE gets its best solution when  $G_1 = 300$  and  $G_2 = 700$ .

**4.2. Pseudocode of TSDE.** The pseudocode of TSDE is shown in Algorithm 1.

### 4.3. Comparative Study

**4.3.1. Comparison between TSDE and Other Global Minimizing Algorithms.** In this paper, genetic algorithm (GA) and basic DE are chosen to compare the search ability with STDE for this problem. The main parameters of GA are set as follows: the maximum number of iterations  $G_m = 1000$ , the length of chromosome  $L_c = 20$ , mutation factor  $F = 0.6$ , and crossover rate  $CR = 0.1$ ; the main parameters of basic DE: the maximum number of iterations  $G_m = 1000$ , the population size  $NP = 100$ , mutation factor  $F = \text{Rand}[0.1:0.3]$ , and crossover rate  $CR = \text{Rand}[0.7:0.9]$ ; the main parameters of TSDE: the maximum two numbers of iterations  $G_1 = 300$ ,  $G_2 = 700$ , the population size  $NP = 100$ , mutation factor  $F = \text{Rand}[0.1:0.3]$ , and crossover rate  $CR = \text{Rand}[0.7:0.9]$ . All algorithms are coded in MATLAB 7.0 and executed in HP desktop 6300 MT with Intel Core i5-3470 CPU @3.2 GHz and 4 GB RAM.

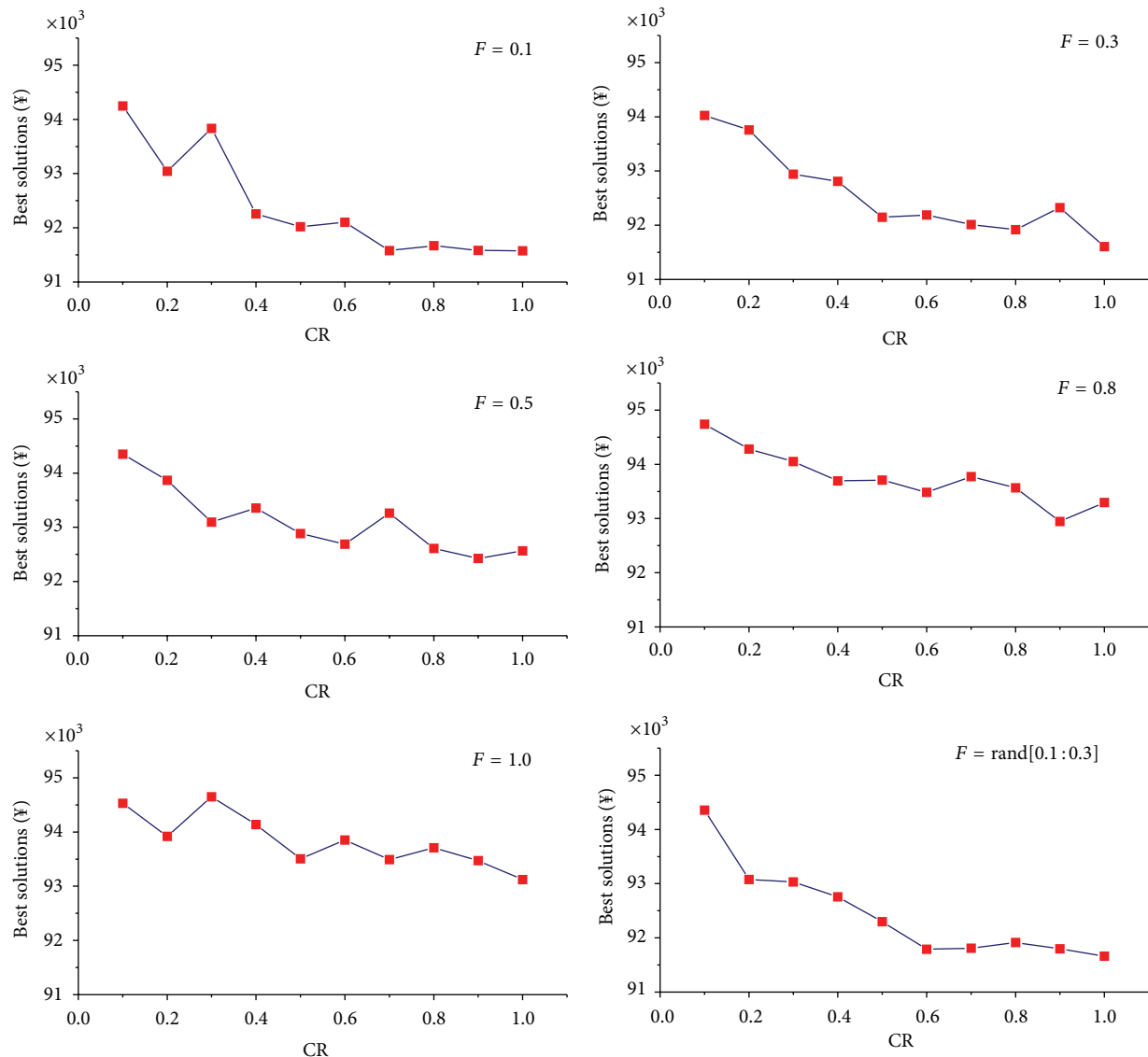
Each of algorithm runs 10 times, and the statistical results are listed in Table 6, in which the bold font are the best solution. The average searching quality of TSDE is better than the others. Moreover, the standard deviations by basic DE and TSDE are much smaller than GA.

**4.3.2. Comparison between Optimal Operation and Manual Operation.** Finally, we compare the optimal operation with manual operation in the OOP. The scheduling strategy of manual operation is that all RO units are running until reaching the high amount limits of the PFWTs from time 1 to time 8 every day, when the time-of-use electricity price is the lowest. When the freshwater amount of each PFWT is at high limit, the corresponding RO units stop. At other time periods, the running status of each RO unit is only determined by the user's freshwater demand, and the time-of-use electricity price will no longer be considered. The optimal operation obeys the scheduling plan computed by TSDE.

Figure 5 is the comparison of freshwater generation of each RO unit between optimal operation and manual operation. As Figure 5 shows, those RO units which have smaller freshwater generation capacity, such as Unit 1 and Unit 2, are almost running because of its smaller running cost. On the contrary, the RO unit which has large capacity (its running cost increases according to its capacity), such as Unit 6, is opened only when necessary. So, it has low

TABLE 4: Analysis of NP and  $G_m$  of TSDE.

NP	$G_m$	Best solution (¥)	Mean value (¥)	Std. dev.
200	250	89758	90376.7	351.6491
250	200	89662	90471.4	556.8511
334	150	89116	89666.7	410.5998
500	100	88822	89926.1	423.0399
625	80	88725	89508.5	451.2566
<b>1000</b>	<b>100</b>	<b>88429</b>	<b>88955.4</b>	<b>376.5983</b>

FIGURE 3: Analysis of CR in different  $F$ .

utilized efficiency. The others which have medium capacity are working intermittently.

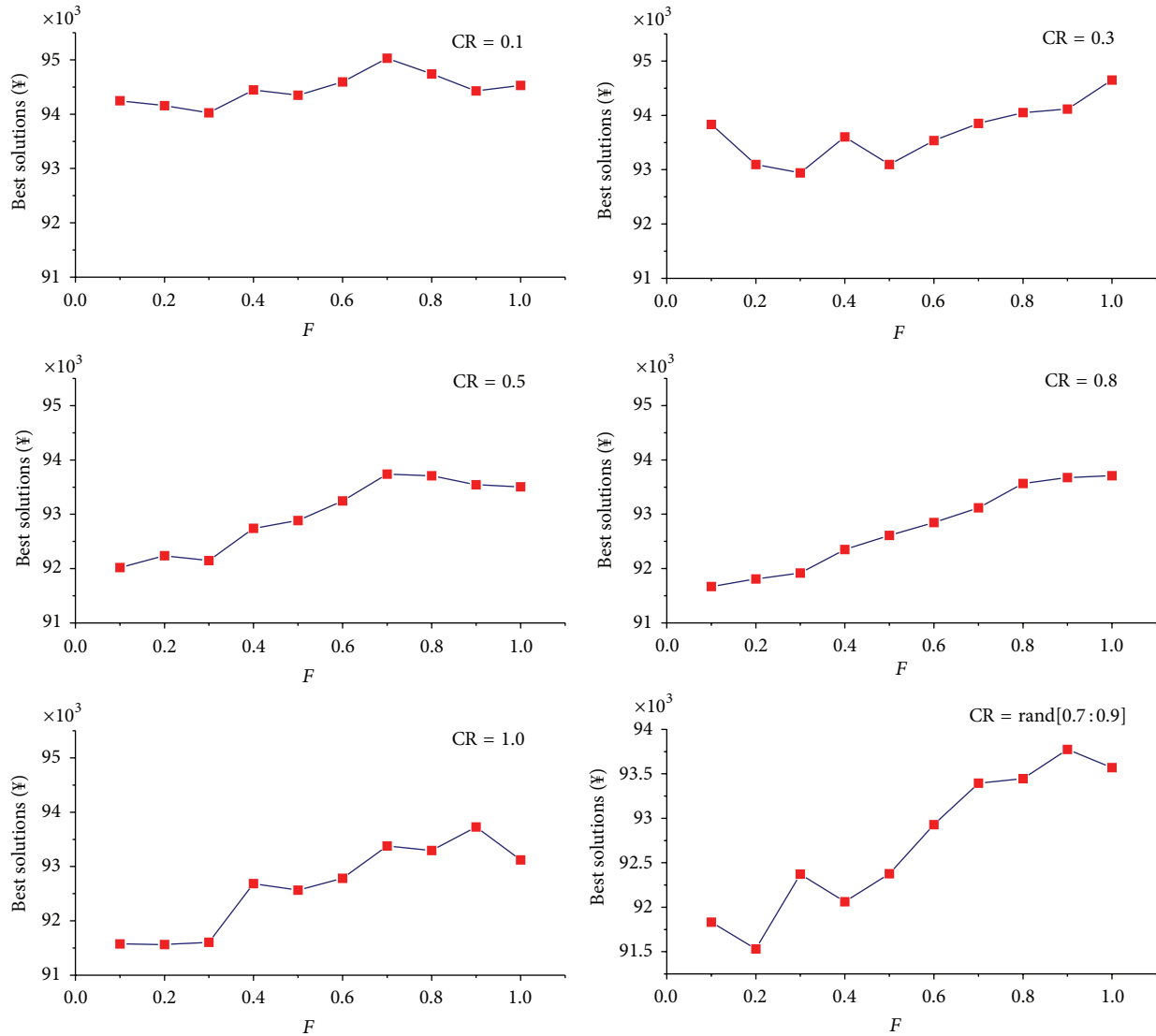
Figure 6 is the comparison of capacity of each PFWT. The sum of capacity of all PFWTs fluctuates with the profile of freshwater demand.

From Figure 6, it is observed as follows.

- (1) Both of these two schedules open the RO units to generate freshwater from time 1 to time 8, and the sum of freshwater reaches its peak at time 8;
- (2) the manual operation does not take the time-of-use electricity price as the cost factor of TRC after time 8, so even at the highest electricity price period, such

TABLE 5: Analysis of  $G_1$  and  $G_2$  of TSDE.

$G_1$	$G_2$	Best solution (¥)	Mean value (¥)	Std. dev.
300	700	88140	88745.38	467.6048
500	500	88464	88878.9	383.1357
700	300	88556	88850	166.2274
1000	0	88275	89052	489.2724

FIGURE 4: Analysis of  $F$  in different CR.

as time 15, the sum of freshwater of PFWTs is still increasing. That is, the RO units are still opening at these time periods, so that the TRC under manual operation cannot get its optimal value;

- (3) the optimal operation takes full use of its advantage in global optimization, so at times 8, 13, and 18, the sum of generated freshwater reaches its local peaks before the time-of-use electricity price gets higher. Therefore, when the electricity price is higher, the storage of freshwater is used to satisfy user's demand,

and the RO units will not be opened unless the storage of freshwater reaches its lower limit. In this way, the TRC of this plant under optimal operation is 5% lower than manual operation (Figure 7).

## 5. Conclusions

The OOP of large-scale parallel-unit SWRO desalination plant is modeled as a MINLP, in which binary-valued vectors indicate the on/off statuses of RO units and real-valued



```

Line      % Pseudocode of TSDE
(1)      Begin
(2)      Set NP = 100; D = 20; G1 = 300; G2 = 700;
(3)      % Stage One
(4)      Initialize population  $x_{i,G}$  ( $i = 1, 2, \dots, NP$ ) (the vectors which represent the status of RO are initialized
        randomly in the range of [0, 1]; the vectors which represent the amount of freshwater are
        initialized as constants, which equal the average capacity of each RO unit)
(5)      G = 1
(6)      For G = 1 to G1
(7)          For k = 1 to K
(8)              For i = 1 to NP
(9)                  Randomly selected three integers  $r_1, r_2$ , and  $r_3$  from the set {1, 2, ..., NP}
                    ( $r_1 \neq r_2 \neq r_3 \neq i$ )
(10)                  $F = 0.2 * \text{rand} + 0.1$ 
(11)                  $v_{i,G} = x_{r_1,G} + F * (x_{r_2,G} - x_{r_3,G})$ 
(12)                 Randomly selected index rnbr(j) ( $\text{rnbr}(j) \in \{1, 2, \dots, D\}$ )
(13)                 CR = rand * 0.2 + 0.7
(14)                 For j = 1 to D
(15)                     If (rand < CR) or ( $j = \text{rnbr}(j)$ )
(16)                          $u_{ij,G} = v_{ij,G}$ 
(17)                     Elseif
(18)                          $u_{ij,G} = x_{ij,G}$ 
(19)                     End If
(20)                     If ( $u_{ij,G} < x_{ij}^L$ ) or ( $u_{ij,G} > x_{ij}^U$ )
(21)                          $u_{ij,G} = x_{ij}^L + \text{rand} * (x_{ij}^U - x_{ij}^L)$ 
(22)                     End If
(23)                 End For
(24)                 If  $F(u_{i,G}) > F(x_{i,G})$ 
(25)                      $x_{i,G+1} = u_{i,G}$ 
(26)                 Else
(27)                      $x_{i,G+1} = x_{i,G}$ 
(28)                 End If
(29)             End For
(30)         End For
(31)     End For
(32)     % Stage Two
(33)     Initialize population  $x_{i,G}$  ( $i = 1, 2, \dots, NP$ ) (30 percent of them are remained the best values
        from Stage One and 70 percent are generated randomly)
(34)     G = 1
(35)     For G = 1 to G2
(36)         For k = 1 to K
(37)             For i = 1 to NP
(38)                 Randomly selected three integers  $r_1, r_2$ , and  $r_3$  from the set {1, 2, ..., NP}
                    ( $r_1 \neq r_2 \neq r_3 \neq i$ )
(39)                  $F = 0.2 * \text{rand} + 0.1$ 
(40)                  $v_{i,G} = x_{r_1,G} + F * (x_{r_2,G} - x_{r_3,G})$ 
(41)                 Randomly selected index rnbr(j) ( $\text{rnbr}(j) \in \{1, 2, \dots, D\}$ )
(42)                 CR = rand * 0.2 + 0.7
(43)                 For j = 1 to D
(44)                     If (rand < CR) or ( $j = \text{rnbr}(j)$ )
(45)                          $u_{ij,G} = v_{ij,G}$ 
(46)                     Elseif
(47)                          $u_{ij,G} = x_{ij,G}$ 
(48)                     End If
(49)                     If ( $u_{ij,G} < x_{ij}^L$ ) or ( $u_{ij,G} > x_{ij}^U$ )
(50)                          $u_{ij,G} = x_{ij}^L + \text{rand} * (x_{ij}^U - x_{ij}^L)$ 
(51)                     End If
(52)                 End For
(53)                 If  $F(u_{i,G}) > F(x_{i,G})$ 

```

```

(54)            $x_{i,G+1} = u_{i,G}$ 
(55)           Else
(56)            $x_{i,G+1} = x_{i,G}$ 
(57)           End If
(58)       End For
(59)   End For
(60) End For
(61) End

```

ALGORITHM 1: Pseudocode of TSDE.

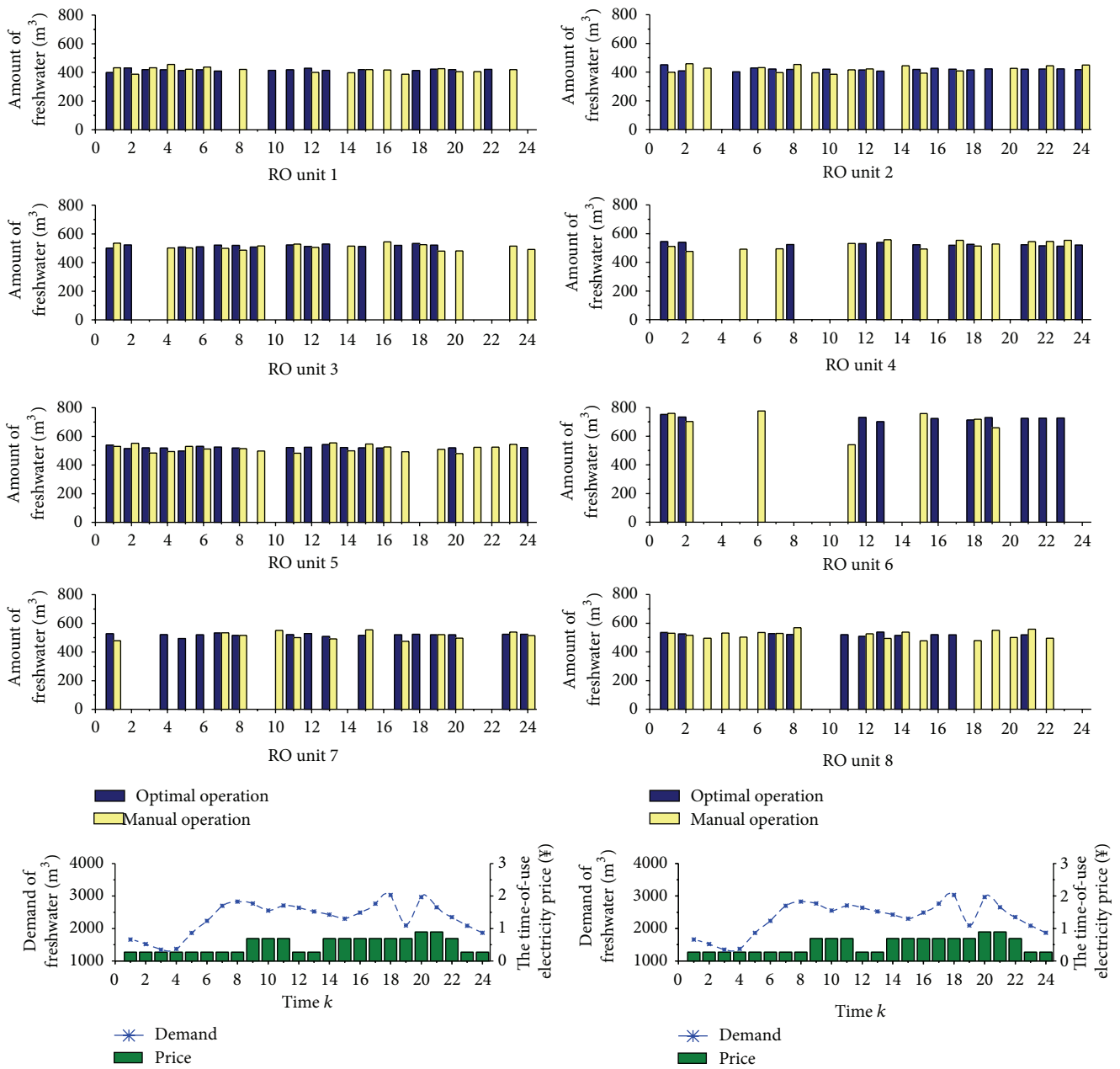


FIGURE 5: The comparison of capacity of each RO unit.

TABLE 6: Comparison of the best solutions of STDE with basic DE and GA (¥).

Algorithm	Best solution	Worst solution	Mean value	Std. dev.
GA	89680	91102	90556	622.9885
Basic DE	88429	89266	88955.4	376.5983
<b>TSDE</b>	<b>88140</b>	<b>89246</b>	<b>88791.59</b>	<b>391.3258</b>

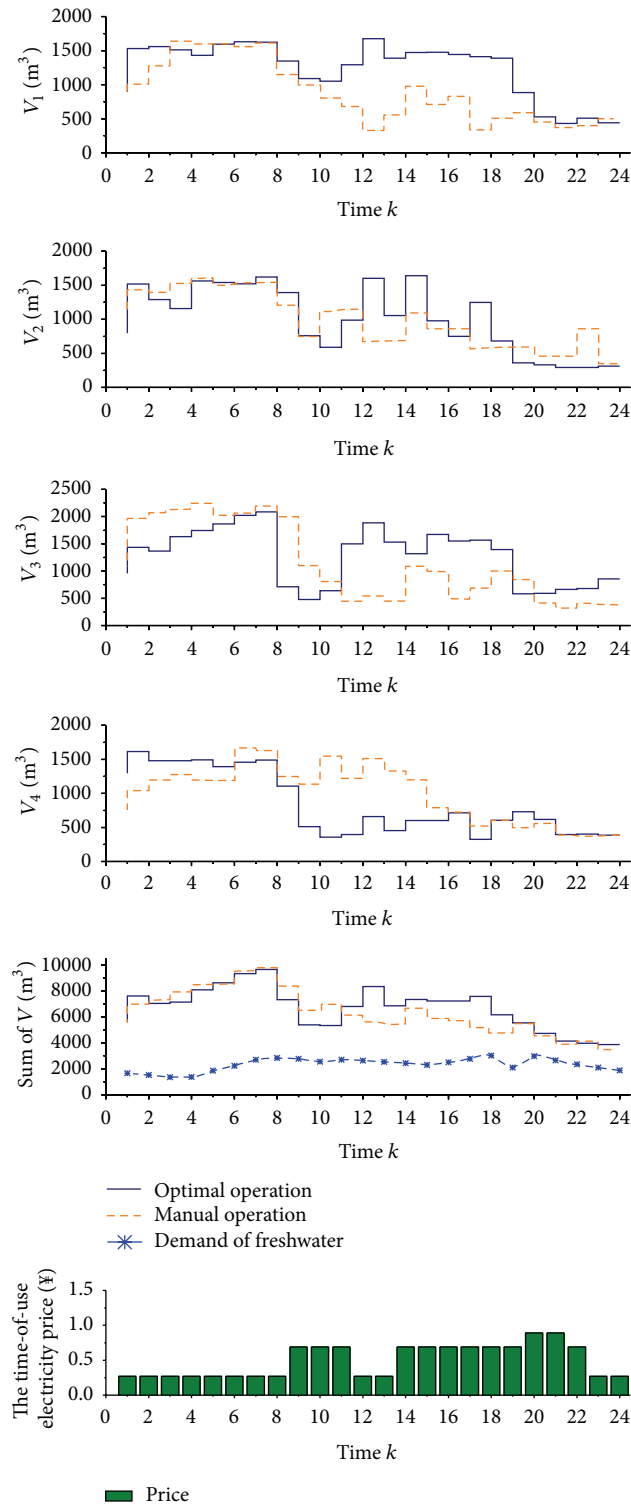


FIGURE 6: The comparison of capacity of each RO unit.

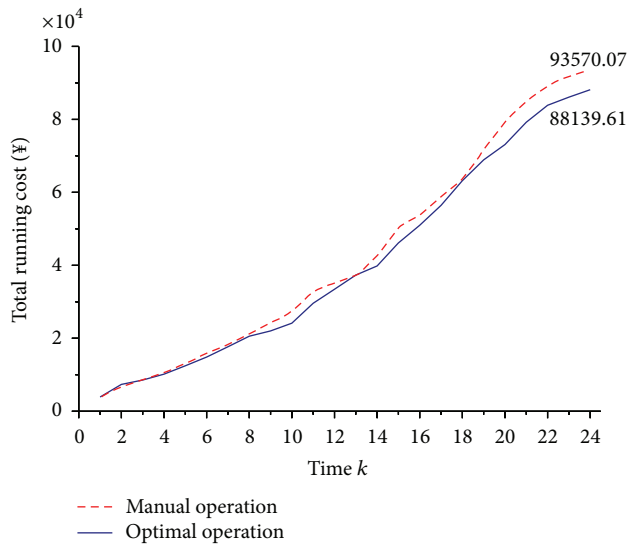


FIGURE 7: The comparison of TRC.

vectors indicate the amounts of freshwater generated by RO units and the amounts of supplied freshwater by PFWFs. The objective function of the OOP is the total running cost of the desalination plant, and the constraint functions include the technical limitations, the design requirements of each RO unit, and the freshwater demand of user. A novel DE, two-stage DE, is presented to solve this OOP, and the effects of its main parameters on the search ability are analyzed in this paper. Applying this TSDE to a 100,000 ton SWRO desalination plant in Liuheng Island, China, it is observed that the proposed TSDE can successfully improve the utilization rate of RO units to reduce the TRC.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work was supported by National Key Technology Research and Development Program of the Ministry of Science and Technology of China (2009BAB47B06) and the grants of National Natural Science Foundation of China (no. 61374142).

## References

- [1] N. M. Wade, "Distillation of plant development and cost update," *Desalination*, vol. 136, no. 1-3, pp. 3-12, 2001.
- [2] E. El-Zanati and S. Eissa, "Development of a locally designed and manufactured small-scale reverse osmosis desalination system," *Desalination*, vol. 165, pp. 133-139, 2004.
- [3] A. Villafafila and I. M. Mujtaba, "Fresh water by reverse osmosis based desalination: simulation and optimisation," *Desalination*, vol. 155, no. 1, pp. 1-13, 2003.

- [4] M. G. Marcovecchio, P. A. Aguirre, and N. J. Scenna, "Global optimal design of reverse osmosis networks for seawater desalination: modeling and algorithm," *Desalination*, vol. 184, no. 1-3, pp. 259-271, 2005.
- [5] Y. Y. Lu, Y. D. Hu, D. M. Xu, and L. Y. Wu, "Optimum design of reverse osmosis seawater desalination system considering membrane cleaning and replacing," *Journal of Membrane Science*, vol. 282, no. 1-2, pp. 7-13, 2006.
- [6] Y. Y. Lu, Y. D. Hu, X. L. Zhang, L. Y. Wu, and Q. Z. Liu, "Optimum design of reverse osmosis system under different feed concentration and product specification," *Journal of Membrane Science*, vol. 287, no. 2, pp. 219-229, 2007.
- [7] Y. Saif, A. Elkamel, and M. Pritzker, "Optimal design of reverse-osmosis networks for wastewater treatment," *Chemical Engineering and Processing*, vol. 47, no. 12, pp. 2163-2174, 2008.
- [8] P. Sarkar, D. Goswami, S. Prabhakar, and P. K. Tewari, "Optimized design of a reverse osmosis system with a recycle," *Desalination*, vol. 230, no. 1-3, pp. 128-139, 2008.
- [9] F. Majali, H. Ettouney, N. Abdel-Jabbar, and H. Qiblawey, "Design and operating characteristics of pilot scale reverse osmosis plants," *Desalination*, vol. 222, no. 1-3, pp. 441-450, 2008.
- [10] M. Verhuelsdonk, T. Attenborough, O. Lex, and T. Altmann, "Design and optimization of seawater reverse osmosis desalination plants using special simulation software," *Desalination*, vol. 250, no. 2, pp. 729-733, 2010.
- [11] K. M. Sassi and I. M. Mujtaba, "Optimal operation of RO system with daily variation of freshwater demand and seawater temperature," *Computer and Chemical Engineering*, vol. 59, pp. 101-110, 2013.
- [12] Y. G. Lee, A. Gambier, E. Badreddin, S. Lee, D. R. Yang, and J. H. Kim, "Application of hybrid systems techniques for cleaning and replacement of a RO membrane," *Desalination*, vol. 247, no. 1-3, pp. 25-32, 2009.
- [13] D. W. Jing, *Optimization Design of Reverse Osmosis System*, Chemical Industrial Press, Beijing, China, 2006.
- [14] R. Storn and K. Price, "Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces," Tech. Rep. TR-95-012, International Computer Science Institute, Berkeley, Calif, USA, 1995.
- [15] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.
- [16] R. Storn and K. Price, "Differential evolution—a simple evolution strategy for fast optimization," *Dr. Dobbs Journal*, vol. 78, pp. 18-24, 1997.
- [17] R. Storn and K. Price, *Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*, University of California, Berkeley, Calif, USA, 2006.
- [18] R. Storn, K. Price, and J. Lampinen, *Differential Evolution—A Practical Approach to Global Optimization*, Springer, New York, NY, USA, 2005.
- [19] Y. W. Zhong, L. J. Wang, C. Y. Wang, and H. Zhang, "Multi-agent simulated annealing algorithm based on differential evolution algorithm," *International Journal of Bio-Inspired Computation*, vol. 4, pp. 217-228, 2012.
- [20] X. S. Yang and S. Deb, "Two-stage eagle strategy with differential evolution," *International Journal of Bio-Inspired Computation*, vol. 4, pp. 1-5, 2012.

- [21] C. W. Carroll, "The created response surface technique for optimizing nonlinear restrained systems," *Operations Research*, vol. 9, pp. 169–184, 1961.
- [22] A. V. Fiacco and G. P. McCormick, "Extensions of SUMT for nonlinear programming: equality constraints and extrapolation," *Management Science*, vol. 12, pp. 816–828, 1968.
- [23] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.
- [24] F.-Z. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 340–356, 2007.
- [25] R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561–579, 2010.
- [26] A. Ponsich and C. A. C. Coello, "Differential evolution performances for the solution of mixed-integer constrained process engineering problems," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 399–409, 2011.

