Box-Constrained Optimization Methodology and Its Application for a Water Supply System Model

Mashor Housh[1]; Avi Ostfeld, F.ASCE[2]; and Uri Shamir, F.ASCE[3]

Abstract: This study introduces a new search method for box-constrained optimization problems called the search method for box optimization (SMBO). SMBO is a population heuristic-based search methodology that solves global optimization problems. SMBO represents the population as a probability density function (PDF) inside the problem bounds. The PDF shape is dynamically adapted during the process to guide to a "good" search domain. The applicability and the efficiency of the method are demonstrated using two benchmark sets, which include unimodal, multimodal, expanded, and hybrid composition functions. The performance of SMBO is compared with several genetic algorithms (GAs); the first benchmark compares it with nine codes of traditional/classic GAs, and the second compares SMBO with two recent variants of genetic algorithms. The results show that SMBO performs as well as or better than the GAs in both comparisons. The method is demonstrated on a nonlinear model for management of a water supply system (WSS), and the results are compared with the commercial GA toolbox of matrix laboratory (MATLAB). **DOI: 10.1061/(ASCE)WR.1943-5452.0000229.** © *2012 American Society of Civil Engineers.*

CE Database subject headings: Algorithms; Optimization; Water supply; Water distribution systems.

Author keywords: Evolutionary algorithms; Genetic algorithms; Global optimization; Search methods; Water supply systems.

Introduction

Many heuristic techniques were proposed to solve multidimensional global optimization problems (Ahrari et al. 2009; Georgieva and Jordanov 2009; Regis and Shoemaker 2007), among them genetic algorithms (GAs) (Harik et al. 1999; Holland 1975; Tu and Lu 2004) developed from the genetic evolution of biological organisms. Essentially, each variable is represented as a gene, whereas the vector of variables is represented as a chromosome that combines all genes.

The GA starts with an initial population in which each individual has its own chromosome. The population evolves over generations using random selection of the fitness of the individuals. Pairs of individuals are selected out of the current generation and a crossover operation recombined their chromosomes. To ensure diversity, mutation is applied to some individuals. Generally, the efficiency of GAs depends on the values of the algorithm parameters, such as population size, initial population, selection scheme, mutation fraction, and retention of elite members for the next generation (Tvrdík 2009). Identifying good parameter values for any search method is time-consuming and difficult. However, for some applications, such as optimal design of water distribution networks, a parameter-setting-free search method may be useful and was recently proposed by Geem and Cho (2011).

This work presents a search method for global optimization of box-constrained problems. The search method for box optimization (SMBO) represents the population of each generation as a probability density function (PDF) defined within the problem bounds in which each gene has its own triangular-shape PDF.

The details of this method and performance comparison using two benchmark sets are given in the next sections. In section 2, the method and its algorithm are presented, including a 2D example to demonstrate its propagation toward the optimal solution. Section 3 contains comparisons with results obtained by other search methods and section 4 demonstrates the application of SMBO for a WSS model.

Searching Method for Box Optimization

Method Outline

The following optimization problem is considered:

$$\min_{x} f(x) \text{ subject to } LB \le x \le UB \quad (1)$$

where $x \in \mathcal{R}^m$ = variables vector, f = objective function, and $LB \in \mathcal{R}^m$, $UB \in \mathcal{R}^m$ = lower and the upper bounds, respectively.

Population-based search methods are defined as follows:

$$P^{\text{new}} = G(P^{\text{old}}, f(P^{\text{old}})) \quad (2)$$

where P^{old} = a subset of solutions inside the search domain that we call population, f = objective function that returns the fitness for each population member, and G = manipulation function that creates a new population evolved from the previous one.

For instance, in GAs the manipulation function is defined by the operators applied to the old population to create the new one, such as, for example, selection, mutation, and crossover.

[1]Ph.D. Student, Faculty of Civil and Environmental Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel. E-mail: mashor@tx.technion.ac.il

[2]Associate Professor, Faculty of Civil and Environmental Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (corresponding author). E-mail: ostfeld@tx.technion.ac.il

[3]Emeritus Professor, Faculty of Civil and Environmental Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel. E-mail: shamir@tx.technion.ac.il

Note. This manuscript was submitted on July 14, 2011; approved on January 5, 2012; published online on January 12, 2012. Discussion period open until April 1, 2013; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Water Resources Planning and Management***, Vol. 138, No. 6, November 1, 2012. © ASCE, ISSN 0733-9496/2012/6-651-659/$25.00.

J. Water Resour. Plann. Manage. 2012.138:651-659.

SMBO is a population heuristics-based search method in the form of Eq. (2) in which a PDF represents each gene of a population member. For each gene, the PDF is represented as an isosceles triangle inside the problem bounds, which is defined by a center point and base length. With regard to the manipulation function, SMBO draws n independent samples taken from the previous PDF, evaluates these samples, and defines the new population (PDF).

To define the new population: (1) the new center points of the triangular PDFs are calculated as a mean of the elite members of the previous population sample and (2) the new bases lengths of the triangles are reduced to achieve convergence.

Because SMBO deals with box optimization problems, the manipulation function must guarantee that all new population members are within the feasible domain. Therefore, when the vertices of the triangle are outside the problem bounds, the triangle is truncated at the bounds and normalized again to ensure a new population within the given bounds. Fig. 1 shows the shape of the PDF and Fig. 2 contains the pseudo code of the sampling algorithm from the triangular shape PDF.

Base Reduction Scheme

To achieve convergence, the base lengths of the triangles are reduced during each iteration. The base reduction rate is a key issue for the algorithms: rapid base reduction limits the ability to explore the feasible region whereas slow base reduction decreases the overall performance of the algorithms.

Four parameters are considered when controlling the base reduction: *Warming*, *Refining*, $B_{\min}$-minimum half base, and α-linear reduction factor.

The base reduction rate does not have to be constant; a changing rate during the search process may be defined by choosing $0 \leq \alpha < 1$.

The base length cannot be zero because the triangles represent a PDF. Hence, a minimum half base $B_{\min}$ has to be defined.

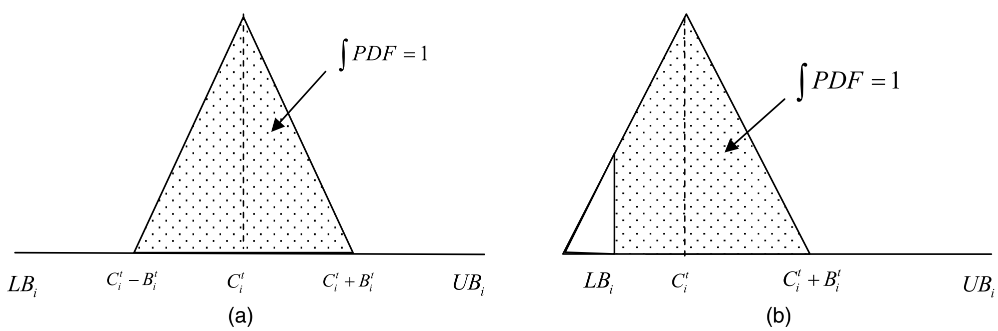

Fig. 1. PDF of variable i at iteration t: (a) the triangular shape is inside the bounds (b) part of the triangular shape is outside the bounds

1) **given** $C \in \Re^m$ – Triangle center points, $B \in \Re^m$ – Half triangle bases
$LB \in \Re^m$ – Lower bound and $UB \in \Re^m$ – Upper bound

2) **generate** m **PDFs, one for each gene**

for $i := 1$ *to* m *do*

$\underline{R}_i = \max(LB_i, C_i - B_i), \overline{R}_i = \min(UB_i, C_i + B_i)$

define the shape Δ_i *using the coordinates*:

$(\underline{R}_i, 0), (\underline{R}_i, \underline{R}_i - (C_i - B_i)), (C_i, B_i), (\overline{R}_i, -\overline{R}_i + (C_i + B_i)), (\overline{R}_i, 0)$

define the PDF $\hat{\Delta}_i = \dfrac{\Delta_i}{\int_{\underline{R}_i}^{\overline{R}_i} \Delta_i}$

3) **generate** n **randomly distributed feasible samples** (Rubinstein, 1981)

for $s := 1$ *to* n *do*

for $i := 1$ *to* m *do*

draw x_i^s *by the inverse transform method out of the PDF* $\hat{\Delta}_i$

Fig. 2. Sampling algorithm

J. Water Resour. Plann. Manage. 2012.138:651-659.

Warming and *Refining* are control parameters that are fractions of the total generations in which there is no base reduction. *Warming* is defined for the first generations, e.g., the first 10% of the total generations, and *Refining* is defined for the last generations, e.g., the last 5% of the total generations.

The parameter *Warming* is used when the feasible region is large; keeping the first iterations without base reduction helps in exploring the solution space.

The parameter *Refining* is used when an accurate solution is desired; keeping the last iterations without base reduction (at the minimum value) helps the method refine the final solution through more sampling in the vicinity of the optimal solution.

Fig. 3 shows the pseudo code of the SMBO algorithm.

$$B_i^t = \begin{cases} B_i^0 & t \geq W \cdot TG \\ B_{\min} & t \geq (1 - RE) \cdot TG \\ \alpha g_1 + (1-\alpha) g_2 & else \end{cases}$$
$$g_1 = B_i^0 - \frac{(B_i^0 - B_{\min})}{RG} \cdot (t - W \cdot TG)$$
$$g_2 = B_i^0 \cdot \left(\frac{B_{\min}}{B_i^0}\right)^{\frac{(t - W \cdot TG)}{RG}} \quad (3)$$

where TG = Total Generation, RE = Refining, W = Warming, $RG = (1 - W - RE) \cdot TG$ and $B_i^0 = \frac{UB_i - LB_i}{2}$.

Propagation to the Optimal Solution (2D Example)

Consider the optimization problem

$$\min_{x,y} x \cdot e^{(-x^2-y^2)} + \frac{(x^2+y^2)}{20} \text{ subject to} \quad (4)$$
$$-10 \leq x \leq 10 \qquad -10 \leq y \leq 10$$

The optimal solution of (4) is $x_{\min} = -0.67$, $y_{\min} = 0$. SMBO starts from an initial population defined by a triangular PDF with centers $C_x^0 = -9.2$, $C_y^0 = -7.8$, and $B^0 = 10$ for both variables. Fig. 4 shows the propagation of the algorithm toward the optimal solution, and Fig. 4a shows the first and second PDFs representing the initial and the second populations during the solution process. Fig. 4b shows all of the PDFs during the search process. The final PDFs are concentrated on the optimal solution because the base of the triangles at the final iteration is 0.02. After normalization, they have a height of 100 whereas the triangle heights during the previous iterations have relativity low heights that are represented by the flat part along axes x and y in Fig. 4b.

1) **given** n – Number of samples in each generation, EC – Elite count,
TG -Total generation, $B_{\min}$ – Minimum base, α - Linear reduction factor,
W – Warming, Re - Refining, LB – Lower bound, UB – Upper bound,

2) **initialize the algorithm**
- set $t = 0$, $B^0 = \frac{UB - LB}{2}$, $X_{\min} = C^0$, $f_{\min} = f(X_{\min})$ where C^0 is a random vector uniformly distributed within the problem bounds.

3) **generate feasible samples**
- apply the Sampling Algorithm in Figure 2 on C^t and B^t
- add the obtained n samples to a set P^t
- add the vectors C^t, $\overline{R}^t$, $\underline{R}^t$ and $X_{\min}$ to the set P^t

4) **evaluate the samples**
- evaluate the members in the set P^t, and sort them in ascending order based on their objective values.
- define X_{best} as the first vector of the sorted vectors, and $f_{best} = f(X_{best})$
- if $f_{best} \leq f_{\min}$ do $X_{\min} = X_{best}$, $f_{\min} = f_{best}$

5) **define the new parameters of the PDFs**
- define C^{t+1} as the average of the first EC vectors of the sorted vectors.
- evaluate B^{t+1} based on equations (3)

6) if $t \leq TG$ do $t = t + 1$, back to step 3
else return $X_{\min} = X_{best}$, $f_{\min} = f_{best}$

Fig. 3. SMBO algorithm

J. Water Resour. Plann. Manage. 2012.138:651-659.

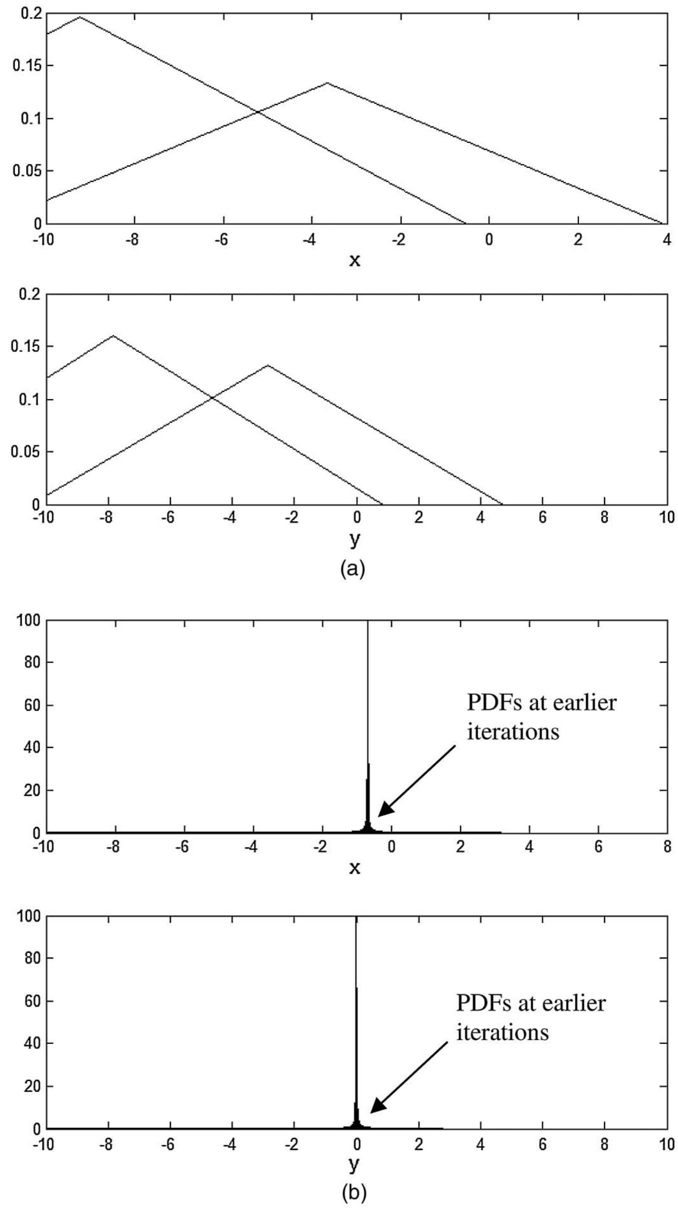

Fig. 4. 2D example: (a) PDFs at the first (left) the second (right) iterations (b) PDFs at the end of the search process

The parameter set considered in this small example is defined in Table 1.

Comparison of Test Results with GAs

To test the ability of the proposed method in efficiently finding the global or at least local minimum, two suits of benchmark functions are optimized by SMBO using the parameters given in Table 1.

De Jong's Test Suite

De Jong's suite contains five test functions (De Jong 1975), listed in the Appendix. F1 is a unimodal function. F2 is a multimodal function. F3 is a discontinuous function. F4 is a unimodal function padded with noise. F5 is a function with many local optima. The performance of SMBO is compared with that obtained in Colorado et al. (1993), which used nine GA codes. For comparison purposes, our algorithm was executed 30 times on all the test functions, each time we used 400 evaluations at each generation and total generations $TG = 1000$, as in Colorado et al. (1993).

Table 1. SMBO Parameters

Parameter	2D example	De Jong's test suite	CEC'05 test suite
TG	50	NA	1000
RE	0	$\{F1, F2, F3, F4 = 0.95\}$, $\{F5 = 0\}$	0.1
W	0	0	0
α	0	0.5	0
$B_{\min}$	0.01	0.01	0.01
n	100	397[a]	147
EC	10	$\min[\lceil 0.1 \cdot (n+3) \rceil, 10]$	$\min[\lceil 0.1 \cdot (n+3) \rceil, 10]$

[a]Total number of evaluations is 400, 397 from sampling and 3 from the triangle vertices; see Fig. 3, step 3.

The stopping criterion $(f^{\text{obtained}} - f^{\text{optimal}}) \le \varepsilon$ was set as in Colorado et al. (1993), where $\varepsilon = 0.01$ except for the function F3 in which $\varepsilon = 0$ and F4 in which $\varepsilon = -2.5$. Table 2 reports the average number of function evaluations and the standard deviation that each algorithm takes to solve De Jong's functions.

CEC-05 Test Suite

In this section, SMBO is applied to a subset of the benchmark functions provided by a CEC-05 special session on real parameter optimization (Suganthan et al. 2005). The chosen subset includes four groups: unimodal, multimodal, expanded, and hybrid composition functions. All of the test functions are listed in the Appendix.

The performance of SMBO is compared with that obtained in Hsieh et al. (2009) using two recent variants of GAs on eighteen test functions with 30 dimensions. For comparison purposes, the number of function evaluations was set to 150,000 and each of the test functions was run 30 times. In SMBO, the evaluations at each generation are set to 150 and the total generations are set to 1,000. Table 3 reports the mean value and the standard deviation of the objective function that each algorithm obtained after 150,000 evaluations. In each function, the algorithm with minimum mean value is shown in bold.

The results show that SMBO obtained better results (lower mean value) than HTGA (Tsai et al. 2004) for all of the test functions except for functions f8 and f11. Moreover, SMBO outperformed SEGA (Hsieh et al. 2009) in fourteen out of the eighteen functions considered, i.e., 78% of the test set.

With regard to function f12, although SMBO obtained a lower mean value in the third decimal place, both SMBO and SEGA are shown in bold. However, one can argue that SEGA obtained a better result because of the lower standard deviation.

Solving a Water Supply System (WSS) problem

A seasonal multiyear model for management of water quantities and salinity for the WSS shown in Fig. 5 was solved. Water is taken from sources including aquifers, reservoirs, and desalination plants and conveyed through a distribution system to consumers who require certain quantities of water under specified salinity constraints.

The objective is to operate the system with minimum total cost of desalination, pumping, delivery, and an extraction levy in the aquifers. The objective function and some of the constraints in

J. Water Resour. Plann. Manage. 2012.138:651-659.

Table 2. De Jong's Test Suite Results

Algorithm	F1 E_ev	F1 Std_ev	F2 E_ev	F2 Std_ev	F3 E_ev	F3 Std_ev	F4 E_ev	F4 Std_ev	F5 E_ev	F5 Std_ev
Cellular	13000	3200	42000	37600	7160	1840	158800	81600	6120	1720
ESGA	11560	2720	33200	22000	6120	1640	61200	19600	5720	17600
Genitor	6800	1640	76000	64000	3280	840	54000	26800	**3160**	**1000**
I-ESGA	12920	3040	32400	16000	7320	2000	150000	78800	5520	1880
I-Genitor	9280	2120	44800	37600	4920	1440	83200	64800	4480	1480
I-pCHC	13280	2960	31200	22800	7520	1760	198000	95600	6520	2120
I-SGA	16520	4480	166800	101200	8800	2120	162000	76800	8120	2760
pCHC	11360	2600	61200	55600	6760	1480	89200	41600	6400	1560
SGA	12280	2960	113600	79200	6680	1680	64400	16000	5840	1760
SMBO	**2760**	**881**	**2013**	**817**	**1747**	**222**	**10427**	**1807**	8107	7763

Note: E_ev, Std_ev are average and standard deviation of function evaluations, respectively.

Table 3. CEC-05 Test Suite Results

Function	SEGA			HTGA			SMBO		
f1	$-4.49E+02$	±	$1.34E-02$	$-4.41E+02$	±	$4.92E+00$	$-4.50E+02$	±	$2.63E-06$
f2	$8.24E+02$	±	$4.33E+02$	$1.51E+04$	±	$1.46E+03$	$-4.50E+02$	±	$2.86E-05$
f3	$1.28E+07$	±	$3.81E+06$	$1.64E+07$	±	$2.66E+06$	$1.09E+07$	±	$5.39E+07$
f4	$1.94E+04$	±	$7.87E+03$	$3.15E+04$	±	$3.30E+03$	$-4.50E+02$	±	$4.97E-05$
f5	$8.90E+02$	±	$1.18E+03$	$4.05E+03$	±	$2.32E+03$	$1.98E+03$	±	$2.81E+03$
f6	$4.52E+03$	±	$1.18E-02$	$4.52E+03$	±	$6.75E-02$	$-1.80E+02$	±	$7.15E-07$
f7	$-1.19E+02$	±	$6.32E-02$	$-1.19E+02$	±	$8.85E-02$	$-1.20E+02$	±	$9.34E-02$
f8	$-3.29E+02$	±	$5.40E-03$	$-3.27E+02$	±	$1.35E+00$	$-3.02E+02$	±	$1.09E+01$
f9	$-2.73E+01$	±	$9.60E+01$	$2.65E+01$	±	$6.75E+01$	$-3.05E+02$	±	$7.55E+00$
f10	$1.24E+02$	±	$3.42E+00$	$1.24E+02$	±	$2.24E+00$	$9.38E+01$	±	$3.19E+00$
f11	$2.01E+04$	±	$1.37E+04$	$2.87E+04$	±	$1.57E+04$	$4.53E+04$	±	$1.55E+05$
f12	$-1.29E+02$	±	$2.13E-01$	$-1.28E+02$	±	$3.82E-01$	$-1.29E+02$	±	$9.68E-01$
f13	$-2.87E+02$	±	$2.40E-01$	$-2.87E+02$	±	$2.97E-01$	$-2.89E+02$	±	$1.04E+00$
f14	$4.97E+02$	±	$1.09E+02$	$4.57E+02$	±	$8.27E+01$	$4.39E+02$	±	$2.66E+02$
f15	$1.14E+03$	±	$3.40E+02$	$1.71E+03$	±	$3.04E+02$	$1.24E+03$	±	$3.41E+01$
f16	$1.53E+03$	±	$7.43E+01$	$1.51E+03$	±	$4.49E+01$	$8.94E+02$	±	$2.54E-04$
f17	$8.17E+02$	±	$5.06E+02$	$9.86E+02$	±	$5.24E+02$	$4.60E+02$	±	$3.37E-04$
f18	$1.55E+03$	±	$6.79E+01$	$1.60E+03$	±	$3.25E+01$	$1.23E+03$	±	$8.00E+00$

Note: Values have been rounded to three significant digits. HTGA-Hybrid Taguchi Genetic Algorithm (Tsai et al. 2004); SEGA-Sharing Evolution Genetic Algorithm (Hsieh et al. 2009).

the model are nonlinear, leading to a nonlinear optimization problem.

The objective is to minimize the operation cost over five years, where each year has two seasons. The optimization problem is

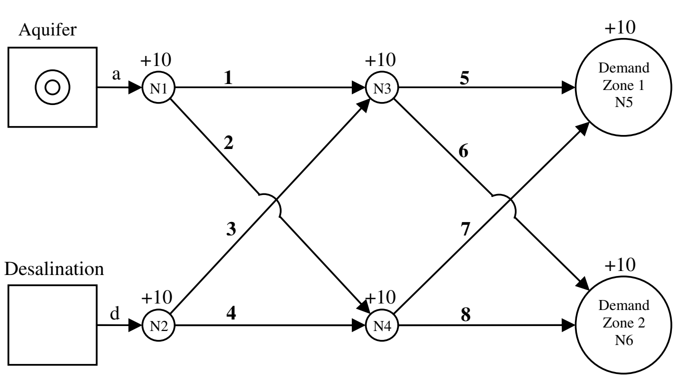

Fig. 5. Network layout

$$\min \text{cost} = \sum_{Y=1}^{5}\sum_{S=1}^{2}\left(\underbrace{\sum_{p=1}^{8} w_p^{S,Y}\cdot (Q_p^{S,Y})^{2.852}}_{\text{Conveyance}} + \underbrace{(\alpha+\beta\cdot h_a^{S,Y})\cdot Q_a^{S,Y}}_{\text{Extraction Levy}} + \underbrace{(\gamma+(C_d^{S,Y})^{-\delta})\cdot Q_d^{S,A}}_{\text{Desalination}}\right) \text{ subject to} \quad (5)$$

State equation for water level in the aquifers $\forall S \forall Y$

$$R_a^{S,Y} - Q_a^{S,Y} = SA_a \cdot (h_a^{S,Y} - h_a^{(S,Y)-1})$$

State equation for water salinity in the aquifers $\forall S \forall Y$

$$(C_R)_a^{S,Y}\cdot R_a^{S,Y} - C_a^{(S,Y)-1}\cdot Q_a^{S,Y} = SA_a \cdot (C_a^{S,Y}\cdot h_a^{S,Y} - C_a^{(S,Y)-1}\cdot h_a^{(S,Y)-1})$$

Desalinated water salinity $\forall S \forall Y$

$$C_d^{S,Y} = C_{\text{sea}}\cdot (1 - RR_d^{S,Y})$$

Water balance in the network $\forall S \forall Y$

J. Water Resour. Plann. Manage. 2012.138:651-659.

$$A \cdot Q^{S,Y} = 0$$

Salinity (mass) balance in the network $\forall S \forall Y$

$$A \cdot \Delta_{Q^{S,Y}} \cdot C^{S,Y} = 0$$

Full dilution in the network nodes $\forall S \forall Y$

$$B \cdot C^{S,Y} = 0$$

Bounds $\forall S \forall Y$

$$(h_{\min})_a^{S,Y} < h_a^{S,Y} < (h_{\max})_a^{S,Y}$$
$$(C_{\min})^{S,Y} < C^{S,Y} < (C_{\max})^{S,Y}$$
$$(Q_{\min})_{\text{Sources}}^{S,Y} < Q_{\text{Sources}}^{S,Y} < (Q_{\max})_{\text{Sources}}^{S,Y}$$
$$(Q_{\min})_{\text{Pipes}}^{S,Y} < Q_{\text{Pipes}}^{S,Y} < (Q_{\max})_{\text{Pipes}}^{S,Y}$$
$$(RR_{\min})_d^{S,Y} \le RR_d^{S,Y} \le (RR_{\max})_d^{S,Y}$$

where p, a, d, z, S, Y denote pipe, aquifer, desalination plant, demand zone, season, and year, respectively.

The model parameters are as follows: $\omega_p^{S,Y}$, α, β, γ, δ are cost parameters; $R_a^{S,Y}$ is recharge (m^3); SA_a is the storativity multiplied by area (m^2); $(C_R)_a^{S,Y}$ is the salinity of the recharge water (mgcl/lit); C_{sea} is the sea water salinity (mgcl/lit); A is the graph matrix of the network; B is the full dilution matrix which indicates equal salinity for two outgoing pipes which share the same node; $Q_{\text{Demand}} = [Q_{z=1}, Q_{z=2}]$ (m^3) is the water demand; and $(.)_{\min}$ and $(.)_{\max}$ are the lower and upper bounds respectively.

The model variables are as follows: $Q_{\text{Sources}} = [Q_a, Q_d]$, $Q_{\text{Pipes}} = [Q_1, \ldots, Q_8]$ are discharges from sources and in pipes, respectively; $C = [C_a, C_d, C_1, \ldots, C_8, C_{z=1}, C_{z=2}]^T$ is a vector combinations of salinity in the system sources, pipes, and demand zones; $RR_d^{S,Y}$ is the removal ratio in the desalination plant; $h_a^{S,Y}$ is the water level in the aquifer; and Δ_Q is a diagonal matrix with a main diagonal $Q = [Q_{\text{Sources}}, Q_{\text{Pipes}}, Q_{\text{Demand}}]^T$.

To reduce the model size, one dependent decision variable was extracted from each equality constraint. Then, the dependent variables were substituted in the objective function and the inequality constraints. For the model previously defined, extracting all dependent variables leads to the following mathematical model that has 50 decision variables, 120 linear constraints, 80 nonlinear constraints, and 50 bounds:

$$\min \sum_{Y=1}^{5} \sum_{S=1}^{2} \text{cost}(Q_{\text{Indep}}^{S,Y}, RR_d^{S,Y}) \text{ subject to} \tag{6}$$

Inequality linear constraints $\forall S \forall Y$

$$\underbrace{\overset{S,Y}{K}}_{(KQ)^{S,Y}} \cdot Q_{\text{Indep}}^{S,Y} \le 0$$

Inequality nonlinear constraints $\forall S \forall Y$

$$g_i(Q_{\text{Indep}}^{S,Y}, RR_d^{S,Y}) \le 0 \quad \forall i = 1 \ldots 8$$

Bounds $\forall S \forall Y$

$$(Q_{\min})_{\text{Indep}}^{S,Y} < Q_{\text{Indep}}^{S,Y} < (Q_{\max})_{\text{Indep}}^{S,Y}$$

$$(RR_{\min})_d^{S,Y} \le RR_d^{S,Y} \le (RR_{\max})_d^{S,Y}$$

where $Q_{\text{Indep}}^{S,Y} = [Q_3^{S,Y}, Q_4^{S,Y}, Q_6^{S,Y}, Q_8^{S,Y}]^T$.

The mathematical model in Eq. (6) was solved using SMBO for which all inequality constraints except for the bounds were added to the objective as penalty terms with penalty factor P. After introducing the penalty terms, the model is a box constrained optimization problem:

$$\min \sum_{Y=1}^{5} \sum_{S=1}^{2} \left[\text{cost}(Q_{\text{Indep}}^{S,Y}, RR_d^{S,Y}) + \text{P} \cdot \sum_{j=1}^{12} \max(0, (KQ)_j^{S,Y}) + \text{P} \cdot \sum_{i=1}^{8} \max(0, g_i^{S,Y}) \right] \quad \text{Subject to} \forall S \forall Y$$

$$(Q_{\min})_{\text{Indep}}^{S,Y} < Q_{\text{Indep}}^{S,Y} < (Q_{\max})_{\text{Indep}}^{S,Y}$$

$$(RR_{\min})_d^{S,Y} \le RR_d^{S,Y} \le (RR_{\max})_d^{S,Y} \tag{7}$$

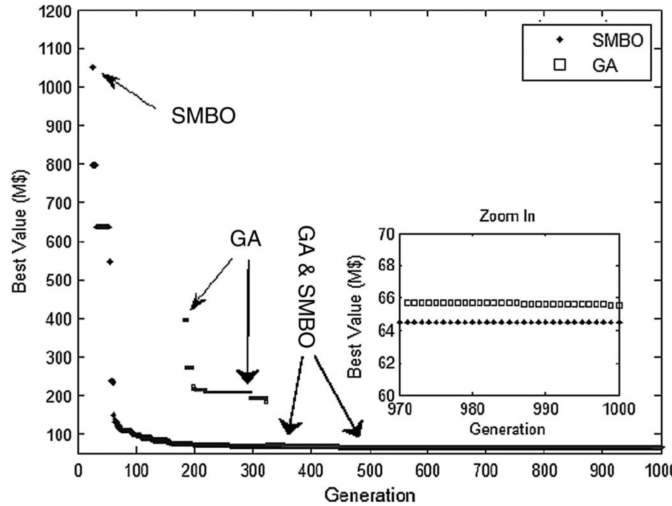

Fig. 6. Progress of SMBO and GA solutions

Problem (7) with penalty factor $\text{P} = 1\text{E}6$ (very high value compared with the values in the cost function) was solved using SMBO and the commercial GA solver of MATLAB. In both algorithms, the evaluations for each generation and the total generations were set to 200 and 1,000, respectively. Both solvers were run without special tuning and the parameters of SMBO are the same as those for the CEC-05 (Table 1), whereas for the MATLAB GA the default options for creation, selection, crossover, and mutation were used.

The solvers were run 10 times each, and Fig. 6 reports the best-run results. SMBO obtained a feasible solution (value below 400 M$) at generation 58 whereas the GA obtained it only at generation 183. The final solution obtained by SMBO (64.51 M$) is also better than that obtained by the GA (65.55 M$). The best-known solution for problem (the original constrained problem) is 64.12 M$. This solution was obtained by the interior point algorithm of the FMINCON solver within the commercial MATLAB optimization toolbox.

Conclusions

A new method for box constrained global optimization, SMBO, was presented. The algorithm searches for the global minimum with a competitive performance compared with other methods. The algorithm was tested on two benchmark problem sets. The number of evaluations needed to obtain the global minimum in De Jong's test suite was the smallest in four functions. Moreover, SMBO achieved a closer solution to the global minimum in most of the CEC'05 test suite problems.

In reference to the WSS management model, SMBO obtained better results than the commercial GA solver of MATLAB and achieved close results to the best-known solution obtained by the gradient optimization solver within the commercial optimization toolbox of MATLAB.

These results demonstrate the promising potential of the method, especially for expensive functions in which each evolution is time consuming.

In the second test, the control parameters of the algorithm were given the same values for all tasks without any tuning. Further research is needed to improve the algorithm's performance through its parameter selection, for example. In the first test suite, further analysis shows that 400 evaluations at each generation, which is specified in the comparison, was too large for SMBO to attain the best performance in the test suite problems. This analysis shows that SMBO obtained even better results when low values of E were selected in the five test functions.

Further research to modify SMBO may be conducted in three areas. First, further research may use discrete PDFs to accommodate discrete variables. Note that because the method is capable of dealing with discontinuous functions, e.g., function F3 in De Jong's test suite, then integer variables may be solved using discontinuous operators such as *floor*, *ceil*, and *round*. Second, research may evaluate the inherent integration of a multiobjective evaluation scheme within an SMBO's algorithm. Third, future research may combine an SMBO with local search algorithms to compose a hybrid global optimization method.

Appendix. Test Problems

This appendix presents all of the test problems used in both comparisons. Tables 4 and 5 present the functions of the first and the second test, respectively.

Table 4. De Jong's Test Suite (De Jong 1975)

Function name	Interval	Function	Global Min
F1 (sphere)	$x \in [-5.12, 5.12]^3$	$f(x) = \sum_{i=1}^{3} x_i^2$	$f_{\min} = 0$
F2 (rosenbrock)	$x \in [-2.048, 2.048]^2$	$f(x) = 100 \cdot (x_1^2 - x_2)^2 + (1 - x_1)^2$	$f_{\min} = 0$
F3 (step)	$x \in [-5.12, 5.12]^3$	$f(x) = \sum_{i=1}^{5} \lfloor x_i \rfloor$	$f_{\min} = -30$
F4 (stochastic)	$x \in [-1.28, 1.28]^{30}$	$f(x) = \sum_{i=1}^{30} i \cdot x_i^4 + \mathrm{Gauss}(0, 1)$	$f_{\min} = 0$
F5 (foxholes)	$x \in [-65.536, 65.536]^2$	$f(x) = \left[0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right]^{-1}$	$f_{\min} = 0.998$

Table 5. CEC'05 Test Suite (30D Problems) (Suganthan et al. 2005)

Function name	Interval	Function	f_{bias} [a]
		Uni-modal functions	
f1-sphere	$x \in [-100, 100]$	$f(x) = \sum_{i=1}^{30} z_i^2 + f_{\text{bias},} z = x - o^{\text{a}}$	−450
f2-schwefel 1.2	$x \in [-100, 100]$	$f(x) = \sum_{i=1}^{30} \left(\sum_{j=1}^{i} \right)^2 + f_{\text{bias},} z = x - o$	−450
f3-rotated high conditioned elliptic	$x \in [-100, 100]$	$f(x) = \sum_{i=1}^{30} (10^6)^{\frac{i-1}{29}} \cdot z_i^2 + f_{\text{bias},} z = (x - o) \cdot M \qquad M - \text{Orthogonal Matrix}$	−450
f4-schwefel 1.2 with noise	$x \in [-100, 100]$	$f(x) = \left(\sum_{i=1}^{30} \left(\sum_{j=1}^{i} z_j \right)^2 \right) (1 + 0.4 \lvert N(0, 1) \rvert) + f_{\text{bias}} \qquad z = x - o$	−450
		Multimodal functions	
f5-rosenbrock	$x \in [-100, 100]$	$f(x) = \sum_{i=1}^{29} (100 \cdot (z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{\text{bias}} \qquad z = x - o + 1$	390
f6-rotated griewank		$f(x) = \sum_{i=1}^{30} \frac{z_i^2}{4000} - \prod_{i=1}^{30} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{\text{bias}} \qquad z = (x - o) \cdot M$ $M - \text{Linear Transformation Matrix, Condition Number} = 3$	−180

J. Water Resour. Plann. Manage. 2012.138:651-659.

Table 5. (*Continued*)

Function name	Interval	Function	f_{bias}[a]
f7-rotated ackley with optimum on bounds	$x \in [-32, 32]$	$f(x) = -20\ \exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} z_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos(2\pi z_i)\right) + 20 + e + f_{\text{bias}}$ $z = (x - o)\dot{c}M$ M-Linear Transformation Matrix, Condition Number = 100	−140
f8-rastrigin	$x \in [-5, 5]$	$f(x) = \sum_{i=1}^{30} z_i^2 \dot{c}(z_i^2 - 10\cos(2\pi z_i) + 10) + f_{\text{bias}}$ $z = z - o$	−330
f9-rotated rastrigin	$x \in [-5, 5]$	$f(x) = \sum_{i=1}^{30} z_i^2 \dot{c}(z_i^2 - 10\cos(2\pi z_i) + 10) + f_{\text{bias}}$ $z = (x - o)\dot{c}M$ M − Linear Transformation Matrix, Condition Number = 2	−330
f10-rotated weierstrass	$x \in [-0.5, 0.5]$	$f(x) = \sum_{i=1}^{30}\left(\sum_{k=0}^{k_{\max}}[a^k\cos(2\pi b^k(z_i + 0.5))]\right) - 30\sum_{k=0}^{k_{\max}}[a^k\cos(2\pi b^k \dot{c}0.5)] + f_{\text{bias}}$ $z = (x - o)\cdot M, a = 0.5, b = 0.3, k_{\max} = 20$ M − Linear Transformation Matrix, Condition Number = 5	90
f11-schwefel 2.13	$x \in [-100, 100]$	$f(x) = \sum_{i=1}^{30}(A_i - B_i(x)^2)^2 + f_{\text{bias}}$ $A_i = \sum_{j=1}^{30}(a_{ij}\sin(\alpha_j) + b_{ij}\cos(\alpha_j))$ $B_i(x) = \sum_{j=1}^{30}(a_{ij}\sin(x_j) + b_{ij}\cos(x_j))$ A, B are two 30×30 matrices. a_{ij}, b_{ij} integer random numbers in the range [-100, 100] α_j are random numbers in the range [-π, π]	−460
		Expanded functions	
f12-griewank + rosenbrock	$x \in [-3, 1]$	$f_a(x) = \frac{x^2}{4000} - \cos(x) + 1$ $f_b(x_1, x_2) = 100\cdot(x_1^2 - x_2)^2 + (1 - x_1)^2$ $f(x) = f_a(f_b(z_1, z_2)) + f_a(f_b(z_2, z_3)) + \cdots + f_a(f_b(z_{29}, z_{30}))$ $+ f_a(f_b(z_{30}, z_1)) + f_{\text{bias}}$ $z = x - o + 1$	−130
f13-rotated expanded scaffer	$x \in [-100, 100]$	$f_a(x_1, x_2) = 0.5 + \frac{\sin^2\left(\sqrt{x_1^2 + x_2^2}\right) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$ $f(x) = f_a(z_1, z_2) + f_a(z_2, z_3) + \ldots + f_a(z_{29}, z_{30})$ $+ f_a(z_{30}, z_1) + f_{\text{bias}}$, $z = (x - o)\cdot M$ M-Linear Transformation Matrix, Condition Number = 3	−300
		Hybrid composition functions[b]	
f14-rotated hybrid composition 1 with noise	$x \in [-5, 5]$	$f(x) = f_1(x)\dot{c}(1 + 0.2\lvert\text{Gauss}(0, 1)\rvert) + f_{\text{bias}} f_1(x)$ is composed using ten functions: two weierstrass, two rastrigin, two ackley, two griewankand, and two sphere functions.[b]	120
f15-rotated hybrid composition 3	$x \in [-5, 5]$	$f(x)$ is composed using ten functions: two rotated expanded scaffer, two (griewank + rosenbrock), two Weierstrass, two rastrigin, and two griewank functions.[b]	360
f16-rotated hybrid composition 3 with high condition number matrix	$x \in [-5, 5]$	The previous function with high condition number matrices in the composition process.	360
f17-rotated hybrid composition 4	$x \in [-5, 5]$	$f(x)$ is composed using ten functions: rotated expanded scaffer, noncontinuous expanded scaffer, Weierstrass, (griewank+ rosenbrock), ackley, rastrigin, noncontinuous rastrigin, and griewank, high conditioned elliptic and sphere function with noise.[b]	260
f18-rotated hybrid composition 4		Same as the previous function.	260

[a]All of the following problems are in 3D and shifted to $f_{\min} = f_{\text{bias}}, o-$, is the shifted global optimum.
[b]For more information on composition functions, see Suganthan et al. (2005, pp. 18–38) (http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC-05/CEC05.htm).

J. Water Resour. Plann. Manage. 2012.138:651-659.

Notation

The following symbols are used in this paper:

A = graph matrix of the network;
B = half triangles bases (SMBO); full dilution matrix (WSS model);
$B_{\min}$ = minimum half triangles bases;
C = triangles center points;
$(C_R)_a^{S,Y}$ = salinity of the recharge water (mgcl/lit);
C_{sea} = sea water salinity (mgcl/lit);
$f(x)$ = objective function;
$f_{\text{best}}, x_{\text{best}}$ = best objective value and best solution in the last iteration, respectively;
$f_{\min}, x_{\min}$ = best objective value and best solution, respectively;
G = manipulation function;
$h_a^{S,Y}$ = water level in the aquifer;
LB, UB = lower bound and upper bound, respectively;
P = population;
Q_{Demand} = water demand (m^3);
$Q_{\text{Sources}}, Q_{\text{Pipes}}$ = discharges from sources and in pipes, respectively;
$R_a^{S,Y}$ = recharge (m^3);
$RR_d^{S,Y}$ = removal ratio in the desalination plant;
Refining = last iterations (%), without base reduction;
S, Y = season and year, respectively;
SA_a = storativity multiplied by area (m^2);
TG = total generations;
Warming = first iterations (%), without base reduction;
x = decision variables;
α = linear reduction factor;
Δ = triangular function;
$\hat{\Delta}$ = triangular probability density function; and
$\omega_p^{S,Y}, \alpha, \beta, \gamma, \delta$ = parameters for the cost function.

References

Ahrari, A., Shariat-Panahi, M., and Atai, A. A. (2009). "Gem: A novel evolutionary optimization method with improved neighborhood search." *Appl. Math. Comput.*, 210(2), 376–386.

Colorado, S. G., Gordon, V. S., and Whitley, D. (1993). "Serial and parallel genetic algorithms as function optimizers." *Proc., 5th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, 177–183.

De Jong, K. A. (1975). "An analysis of the behavior of a class of genetic adaptive systems." Ph.D. dissertation, Univ. of Michigan, Ann Arbor, MI.

Geem, Z. W., and Cho, Y. H. (2011). "Optimal design of water distribution networks using parameter-setting-free harmony search for two major parameters." *J. Water Resour. Plann. Manage.*, 137(4), 377–380.

Georgieva, A., and Jordanov, I. (2009). "Global optimization based on novel heuristics, low-discrepancy sequences and genetic algorithms." *Eur. J. Oper. Res.*, 196(2), 413–422.

Harik, G. R., Lobo, F. G., and Goldberg, D. E. (1999). "The compact genetic algorithm." *IEEE Trans. Evol. Comput.*, 3(4), 287–297.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, MI.

Hsieh, S., Sun, T., and Liu, C. (2009). "Potential offspring production strategies: An improved genetic algorithm for global numerical optimization." *Expert Syst. with Appl.*, 36(8), 11,088–11,098.

Regis, R. G., and Shoemaker, C. A. (2007). "Parallel radial basis function methods for the global optimization of expensive functions." *Eur. J. Oper. Res*, 182(2), 514–535.

Rubinstein, R. Y. (1981). *Simulation and the Monte Carlo method*, Wiley, New York.

Suganthan, P. N., et al. (2005). "Problem definitions and evaluation criteria for the CEC'05 2005 special session on real parameter optimization." *Technical Report*, Nanyang Technological Univ. ⟨http://www.ntu.edu.sg/home/epnsugan⟩.

Tsai, J. T., Liu, T. K., and Chou, J. H. (2004). "Hybrid taguchi-genetic algorithm for global numerical optimization." *IEEE Trans. Evol. Comput*, 8(4), 365–377.

Tu, Z., and Lu, Y. (2004). "A robust stochastic genetic algorithm (STGA) for global numerical optimization." *IEEE Trans. Evol. Comput.*, 8(5), 456–470.

Tvrdík, I. (2009). "Adaptation in differential evolution: A numerical comparison." *Appl. Soft Comput. J.*, 9(3), 1149–1155.

J. Water Resour. Plann. Manage. 2012.138:651-659.