Environmental Modelling & Software 43 (2013) 5–25

Contents lists available at SciVerse ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

# A model-independent Particle Swarm Optimisation software for model calibration

Mauricio Zambrano-Bigiarini [a], Rodrigo Rojas [b,*]

[a] Water Resources Unit, Institute for Environment and Sustainability, Joint Research Centre, European Commission, Via E. Fermi 2749, TP261, 21027 Ispra (VA), Italy
[b] Climate Risk Management Unit, Institute for Environment and Sustainability, Joint Research Centre, European Commission, Via E. Fermi 2749, TP261, 21027 Ispra (VA), Italy

ARTICLE INFO

*Article history:*
Received 30 March 2012
Received in revised form
18 December 2012
Accepted 7 January 2013
Available online 26 February 2013

*Keywords:*
Global optimisation
Evolutionary algorithm
Surface water modelling
Groundwater modelling
SWAT-2005
MODFLOW-2005
R

ABSTRACT

This work presents and illustrates the application of *hydroPSO*, a novel multi-OS and model-independent R package used for model calibration. *hydroPSO* allows the modeller to perform a standard modelling work flow including, sensitivity analysis, parameter calibration, and assessment of the calibration results, using a single piece of software. *hydroPSO* implements several state-of-the-art enhancements and fine-tuning options to the Particle Swarm Optimisation (PSO) algorithm to meet specific user needs. *hydroPSO* easily interfaces the calibration engine to different model codes through simple ASCII files and/or R wrapper functions for exchanging information on the calibration parameters. Then, optimises a user-defined goodness-of-fit measure until a maximum number of iterations or a convergence criterion are met. Finally, advanced plotting functionalities facilitate the interpretation and assessment of the calibration results. The current *hydroPSO* version allows easy parallelization and works with single-objective functions, with multi-objective functionalities being the subject of ongoing development. We compare *hydroPSO* against standard algorithms (SCE_UA, DE, DREAM, SPSO-2011, and GML) using a series of benchmark functions. We further illustrate the application of *hydroPSO* in two real-world case studies: we calibrate, first, a hydrological model for the Ega River Basin (Spain) and, second, a groundwater flow model for the Pampa del Tamarugal Aquifer (Chile). Results from the comparison exercise indicate that *hydroPSO* is: i) effective and efficient compared to commonly used optimisation algorithms, ii) "scalable", i.e. maintains a high performance for increased problem dimensionality, and iii) versatile to adapt to different response surfaces of the objective function. Case study results highlight the functionality and ease of use of *hydroPSO* to handle several issues that are commonly faced by the modelling community such as: working on different operating systems, single or batch model execution, transient- or steady-state modelling conditions, and the use of alternative goodness-of-fit measures to drive parameter optimisation. Although we limit the application of *hydroPSO* to hydrological models, flexibility of the package suggests it can be implemented in a wider range of models requiring some form of parameter optimisation.

© 2013 Elsevier Ltd. All rights reserved.

## Software availability

Name of the software: *hydroPSO*
Version: 0.3-0
Developers: Mauricio Zambrano-Bigiarini & Rodrigo Rojas
Contact E-mail: mzb.devel@gmail.com; Rodrigo.RojasMujica@gmail.com
Available since: April 2012
Available from: http://www.rforge.net/hydroPSO/; http://cran.r-project.org/web/packages/hydroPSO/
Tutorial: http://www.rforge.net/hydroPSO/files/hydroPSO_vignette.pdf
Program language: R
Cost: Free
Package size: ca. 150 Kb

## 1. Introduction and scope

A long tradition of modelling expertise in environmental sciences and the advent of more powerful computers have resulted in a large suite of computer models being available in recent years. In

* Corresponding author.
*E-mail addresses:* Mauricio.Zambrano@jrc.ec.europa.eu (M. Zambrano-Bigiarini), Rodrigo.Rojas@jrc.ec.europa.eu, rodrigo.rojasmujica@gmail.com (R. Rojas).

1364-8152/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.envsoft.2013.01.004

addition to critical requirements, such as data quality and availability, every model must be able to reasonably reproduce observations of system state variables. This, in order to guarantee a minimum level of confidence in the predictive capacity of these tools, which is essential for end-users of model predictions.

The process of constraining model outputs to measured ranges of observed variables is termed *model calibration* (see, e.g., Refsgaard and Henriksen, 2004). By model calibration we understand a process by which key (and sensitive) parameters of a model are adjusted within physically feasible ranges and which aims to achieve a good fit between observed variables and their simulated equivalents. In the specialised literature, model calibration is also referred to as "parameter optimisation/estimation", "inverse problem" or "history matching" and, thus, these terms are used interchangeably hereafter.

Models generally suffer from over-parametrisation, i.e. an excessive number of (redundant) parameters that are not fully identifiable (see, e.g., Beven, 2006). A procedure to formally restrict the number of parameters involved in model calibration is, therefore, required in most cases. Amongst the different techniques available (e.g., eigen-based, factor, and principal components analyses), sensitivity analysis is a valuable tool for identifying relevant model parameters and for improving the stability of the model calibration (see, e.g., Hill and Tiedeman, 2007). An extensive overview of sensitivity analysis, however, is beyond the scope of this article and hence we refer the reader to Saltelli et al. (2000) for an excellent review.

To date, a vast collection of calibration algorithms can be found in the literature (see, e.g., Matott et al., 2009). In general, model calibration techniques can be grouped in (i) *semi-intuitive/manual* calibration (trial-and-error), which is time-consuming and cumbersome (see, e.g., Wagener et al., 2004); and (ii) *automated* calibration (optimised goodness-of-fit measures), which is faster and based on *search* algorithms with explicit "parameter-updating" rules (see, e.g., Duan et al., 1992; Vrugt et al., 2003; Poeter et al., 2005; Matott, 2005; Vrugt et al., 2008; Doherty, 2010).

At the same time, several well-known problems (e.g., multiple local optima, numerical granularity and non-convex response surfaces, non-linear interdependence of parameters, interactions between parameter limits, saddle points where first derivatives vanish, non-informative data, and bias, autocorrelation, and heteroscedasticity – inconstant variance – in the residuals) undermine the search for the "optimum parameter set" (see, e.g., Duan et al., 1992; Beven and Binley, 1992; Yapo et al., 1996; Uhlenbrook et al., 1999; Beven, 2006). One consequence of these problems is parameter non-uniqueness, i.e. when several parameter sets might provide an equally good fit (e.g., Beven, 2006).

Advanced (global) optimisation/calibration techniques aimed at handling (and partially overcoming) the limitations listed above include, for example, Simulated Annealing (SA) (Kirckpatrick et al., 1983), Genetic and Evolutionary Programming (GP and EP) (Goldberg, 1989), including the widely-used Shuffled Complex Evolution algorithm (SCE_UA) (Duan et al., 1992), Particle Swarm Optimisation (PSO) (Kennedy and Eberhart, 1995), Ant Colony Optimisation (ACO) (Dorigo and Stutzle, 2004), Differential Evolution (DE) (Storn and Price, 1997), and an adaptive multi-method search procedure termed as AMALGAM (Vrugt and Robinson, 2007). These techniques, however, are generally implemented in customised pieces of software that have to be strongly modified to set up the calibration of different model codes (see, e.g., Matott et al., 2009), require the explicit definition of the links between model inputs (and outputs) and the optimisation algorithm and, more often than not, require strong modifications to alleviate the computational burden through parallelization. These limitations deprive the user of the flexibility to reuse these calibration techniques without having to invest considerable programming efforts in customising a given algorithm to a new calibration problem.

Within the recently developed global optimisation strategies, Particle Swarm Optimisation (PSO) has received a surge of attention given its flexibility, ease of implementation, and efficiency (Poli et al., 2007; Poli, 2008). PSO is a population-based stochastic optimisation technique inspired by social behaviour of bird flocks, which shares a few similarities with other evolutionary optimisation techniques such as Genetic Algorithms (GA) (Poli et al., 2007). In PSO, however, the search space is explored on the basis of individual and neighbourhood-based best-known "particle positions" with no presence of evolutionary operators such as mutation or crossover. Over the years numerous variants of the PSO algorithm have appeared in the literature aimed at improving its performance. This has resulted in a large and dispersed collection of codes covering from the basic PSO to recent adaptive algorithms requiring less user intervention (see e.g., Cooren et al., 2009).

The aim of this work is to present a new PSO-based optimisation/calibration software, benchmark it against standard optimisation algorithms, and illustrate its application to the calibration of hydrological models. This software, termed *hydroPSO*, is developed as an R package (R Development Core Team, 2011) and some of its key features are: a) multi-OS and model-independent; b) parallel-capable; c) fully compatible with other PEST-like software (Doherty, 2010); d) frugal user intervention to interface *hydroPSO* and any (R-external) model code; e) a full suite of fine-tuning options to customise the calibration engine; f) built-in sensitivity analysis using Latin Hypercube One-factor-At-a-Time (LH-OAT) (van Griensven et al., 2006); and g) availability of functions to perform customised post-process analyses. It should be noted, however, that the current version of *hydroPSO* works only with single-objective functions and a multi-objective version is under development. We benchmark the performance of *hydroPSO* against five standard algorithms (SCE_UA, DE, DREAM, SPSO-2011, and GML). We further illustrate the application of *hydroPSO* in two real-world case studies: calibration of a hydrological model for the Ega River Basin (Spain); and calibration of a groundwater model for the regional aquifer of the Pampa del Tamarugal Basin (Chile). We limit the application of *hydroPSO* to hydrological models as this is the authors' area of expertise. However, based on the flexibility of the package and the benefits added by programming it in R, we believe *hydroPSO* can be implemented in a wider range of models requiring parameter optimisation. This claimed flexibility as well as the ease of use are easily verifiable in the extensive package vignette (tutorial), where the reader can find examples of simple R scripts interfacing *hydroPSO* with complex models such as SWAT-2005 and MODFLOW-2005 (see http://www.rforge.net/hydroPSO/files/hydroPSO_vignette.pdf). These R scripts and the implementation details described in the package tutorial provide the basis for the two case studies analysed in this work.

It should be noted that other model-independent software tools performing not only parameter estimation but also sensitivity and uncertainty analyses are available in the literature (see Matott et al. (2009) for an excellent review). In particular, tools such as OSTRICH (Matott, 2005) and squads/MADS (Vesselinov and Harp, 2012) implement a basic PSO configuration and an adaptive PSO version enhanced with a local-search algorithm. These tools, however, do not allow the user to fine-tune the PSO algorithm and do not comply with the most recent theoretical development expressed in the "Standard PSO 2011" algorithm (Clerc, 2012).

The remainder of this paper is arranged as follows. Section 2 provides details on the basics of the PSO algorithm, whereas Section 3 details the *hydroPSO* R package. Section 4 shows the comparison of *hydroPSO* against standard optimisation algorithms,

while study cases are presented in Section 5. Section 6 provides an in-depth discussion and we finalise presenting concluding remarks in Section 7.

## 2. Particle Swarm Optimisation – PSO

### 2.1. Canonical PSO algorithm

The canonical PSO algorithm starts with a random initialisation of the particles' positions and velocities within the parameter space. Considering a $D$-dimensional search space, *position* and *velocity* for the $i$-th particle are represented by $\vec{X}_i = x_{i1}, x_{i2}, \ldots, x_{iD}x$ and $\vec{V}_i = v_{i1}, v_{i2}, \ldots, v_{iD}$, respectively. The performance of each particle is assessed through a problem-specific "goodness-of-fit" measure, which is the basis for updating $\vec{X}_i$. The best-known position of the $i$-th particle, termed as *personal/previous best*, is represented by $\vec{P}_i = p_{i1}, p_{i2}, \ldots, p_{iD}$, whereas the best-known position within the particle's neighbourhood, termed as *local best* (Clerc, 2010), is represented by $\vec{G} = g_1, g_2, \ldots, g_D$. Velocity and position of the $i$-th particle are updated according to the following equations,

$$\vec{V}_i^{t+1} = \omega \vec{V}_i^t + c_1 \vec{U}_1^t \otimes \left(\vec{P}_i^t - \vec{X}_i^t\right) + c_2 \vec{U}_2^t \otimes \left(\vec{G}^t - \vec{X}_i^t\right) \quad (1a)$$

$$\vec{X}_i^{t+1} = \vec{X}_i^t + \vec{V}_i^{t+1} \quad (1b)$$

where $i = 1, 2, \ldots, N$, with $N$ equal to the swarm size, and $t = 1, 2, \ldots, T$, with $T$ equal to the maximum number of iterations. $\omega$ is the *inertia weight*, $c_1$ and $c_2$ are the *cognitive* and *social* acceleration coefficients, and $\vec{U}_1$ and $\vec{U}_2$ are independent and uniformly distributed random vectors in the range [0,1] (note that $\otimes$ denotes element-wise vector multiplication).

The inclusion of $\omega$ in equation (1a) aims to prevent *swarm explosion*, i.e. an uncontrolled increase of the magnitude of velocities (particle displacement) (Shi and Eberhart, 1998b). In addition to this, Eberhart and Shi (2000) suggest constraining particle velocity to the range $[-\vec{V}^{\max}, \vec{V}^{\max}]$ with $\vec{V}^{\max} = \vec{X}^{\max}$, and $\vec{X}^{\max}$ as the limits of the search space.

### 2.2. Topologies

Particles in the swarm interact by defining a common set of links, which control the exchange of information and are defined as the *swarm topology*. The set of particles "informing" the $i$-th particle is termed the particle's *neighbourhood*, and includes the particle itself as a member (Clerc, 2007).

Two of the most common topologies correspond to the so-called *gbest* and *lbest* (Poli et al., 2007). In the *gbest* topology all particles are inter-connected. This topology is recognised to have a fast convergence but is highly vulnerable to sub-optimal solutions and premature convergence (see, e.g., Mendes, 2004; Poli et al., 2007; Clerc, 2010). In the *lbest* topology, on the contrary, each particle is connected to only two immediate neighbours. The *lbest* topology allows parallel searches in different regions of the search space (Poli et al., 2007), which results in a more thorough search strategy.

One of the main findings by Kennedy and Mendes (2002) is the relative superiority of the *von Neumann* topology, i.e. four neighbours. This topology shows some parallelism with *lbest* but benefits from a larger neighbourhood (Poli et al., 2007).

Finally, Clerc (2007) proposes the *random* topology where each particle informs $k$ particles chosen randomly, with $k$ usually set to 3 (Clerc, 2012). In this topology the "connections" between particles randomly change when the global optimum shows no improvement.

### 2.3. Standard PSO 2011

In a series of technical reports (see Clerc, 2007, 2009, 2012), a clear need to establish a common benchmark to assess the performance of the numerous PSO improvements proposed in the literature has been identified.

The most recent benchmark proposed corresponds to the Standard PSO 2011 (SPSO-2011) (Clerc, 2012). SPSO-2011 starts by defining a centre of gravity ($Gr$) around the current, previous, and local best as follows,

$$\vec{p}_i^t = \vec{X}_i^t + c_1 \vec{U}_1^t \otimes \left(\vec{P}_i^t - \vec{X}_i^t\right) \quad (2a)$$

$$\vec{l}_i^t = \vec{X}_i^t + c_2 \vec{U}_2^t \otimes \left(\vec{G}^t - \vec{X}_i^t\right) \quad (2b)$$

$$\vec{Gr}_i^t = \frac{\left(\vec{X}_i^t + \vec{p}_i^t + \vec{l}_i^t\right)}{3} \quad (2c)$$

then, a random point is defined in the hypersphere $\mathscr{H}(\vec{Gr}_i^t, \|\vec{Gr}_i^t - \vec{X}_i^t\|)$ and the velocity is updated as follows,

$$\vec{V}_i^{t+1} = \omega \vec{V}_i^t + \mathscr{H}\left(\vec{Gr}_i^t, \left\|\vec{Gr}_i^t - \vec{X}_i^t\right\|\right) - \vec{X}_i^t \quad (3)$$

The particle's position is updated following Equation (1b).

Despite the fact that SPSO-2011 is an improvement to the previous SPSO-2007 in terms of accounting for rotational invariance, SPSO-2007 could perform better than SPSO-2011 for separable and (least likely) for non-separable functions (Maurice Clerc, developer SPSO-2007 & 2011, personal communication, 2012). This is the main reason for including both SPSO-2007 & 2011 in the *hydroPSO* R package.

## 3. The *hydroPSO* R package

### 3.1. R: a software environment for environmental sciences

R is an open-source programming language and a "*free software environment for statistical computing and graphics*" (Ihaka and Gentleman, 1996). R uses a command line interface for maximizing power and flexibility, and its functionality excels in exploratory data analysis, ease of producing publication-quality plots, available documentation, and ease of customisation/extensibility via *packages*. To date, the large collection of user-contributed R packages containing state-of-the-art functions/algorithms used in many different fields (see e.g., http://cran.r-project.org/web/views/) continues to grow exponentially (Fox, 2009). These packages are freely available for public scrutiny, thus, resulting in a continuously peer-based quality-control system.

Some of the main advantages of R for the *hydroPSO* package are that is free and open-source, thus actively promoting reproducible research (see, e.g., Ince et al., 2012), is fairly robust on a wide range of operating systems, and presents the ideal environment to benefit from state-of-the-art mathematical/statistical algorithms and data manipulation/visualization techniques.

### 3.2. hydroPSO package description

*hydroPSO* is a multi-OS and model-independent R package that has been designed to allow the user to perform sensitivity analysis, model calibration, and assessment of the results. Unlike other R packages recently developed for similar purposes (see e.g., *hydromad*, Andrews et al., 2011, *R-SWAT-FME*, Wu and Liu, 2012, and *pso*,

Bendtsen, 2012), *hydroPSO* is not restricted to a limited number of hydrological models, can be interfaced with any model with a relatively low programming effort, is fully compatible with calibration tools employing PEST-like template files, and allows parallelization.

In principle, *hydroPSO* only needs to know *which* model parameters need to be calibrated, *where* they need to be written, and *from where* and *how* to read the main model output (see Fig. 1). The calibration engine of *hydroPSO* communicates with any model through simple ASCII files and/or R wrapper functions, which read model inputs and outputs, and compute the model's performance if required. For examples of basic (ASCII files) and advanced (R functions) interfacing we refer the reader to the accompanying *hydroPSO* tutorial available from http://www.rforge.net/hydroPSO/.

### 3.3. Main hydroPSO functions

Fig. 2 shows the interaction among the main functions comprising the *hydroPSO* package. These functions have been specifically developed for *hydroPSO* and are summarised in Table 1 and further described below.

1. The lhoat() function implements the Latin Hypercube One-factor-At-a-Time (LH-OAT) sensitivity analysis technique (van Griensven et al., 2006). lhoat() produces a ranking with the parameter having the largest effect receiving a rank of 1 and the one(s) with the smallest effect receiving a rank equal to the total number of parameters ($D$).
2. The hydromod() function is one of the key components of *hydroPSO*. It has a unique role linking and controlling the execution of a user-defined model with *hydroPSO*. hydromod() first reads a set of parameter values, which are then written into the corresponding model input file(s) by using the information provided in the user-defined *ParamFiles.txt* file. *ParamFiles.txt* defines the name of the model parameters, input files, and specific location of the parameters in the input files. Unlike tools such as PEST (Doherty, 2010), UCODE-2005 (Poeter et al., 2005), OSTRICH (Matott, 2005) or MADS (Vesselinov and Harp, 2012), this is done in one file only with no need for creating numerous *template* files replicating model input files. Then, hydromod() calls the model executable(s) file(s) to obtain the corresponding model outputs, which are read through the out.FUN() R function. Finally, simulated equivalents are assessed against user-provided observations of system state variables through a gof.FUN() R function (For readers unfamiliar with R, a single command line would allow reading two numeric columns with number of rows equal to nobs after $n$ lines of text in the file fname: tmp<-read.table(file=fname, skip=n,header=FALSE,nrows=nobs, colClasses=c("NULL","NULL","numeric","numeric"),fill=TRUE). A simple command like this is the basis for the out.FUN R function. In addition, the function assessing the model performance may be easily defined as gof.FUN="NSE" to select the commonly used Nash-Sutcliffe Efficiency). hydromod() returns the model simulated equivalents and model performance (goodness-of-fit).
3. The main calibration engine is implemented in the hydroPSO() function. This function is another key component of *hydroPSO* that includes state-of-the-art PSO variants and numerous fine-tuning options to customise the calibration engine to specific user needs (see Table 2). At the first iteration, parameters are sampled from a feasible range obtained from the user-defined *ParamRanges.txt* file. Then, hydromod() is called to obtain a goodness-of-fit measure for each particle in the swarm. Particle velocity and position evolve according to the user-defined PSO configuration until some termination criteria are met (e.g., maximum number of iterations (maxiter), or relative (reltol) or absolute (abstol) tolerances are less than a user-defined threshold). hydroPSO() returns the optimum parameter set, all sampled parameters and their corresponding goodness-of-fit, model outputs, particles' velocities, and convergence measures.
4. Results from the hydroPSO() function are post-processed using the plot_results() function, which delivers a series of user-friendly high-quality plots to facilitate the assessment of the model calibration.
5. The verification() function is designed to validate (one or more) user-defined parameter sets. It returns a goodness-of-fit measure for each parameter set defined, the best parameter set, and the goodness-of-fit measure corresponding to the best parameter set.
6. Function PEST2hydroPSO() imports existing PEST files (i.e. *.tpl, *.pst, and *.ins) into the input files required to set up a *hydroPSO* optimisation (i.e. ParamFiles.txt, ParamRanges.txt and hydroPSO-Rscript.R).
7. Function hydroPSO2PEST() exports the *hydroPSO* files into the input files required for performing a PEST local-calibration. This function is particularly useful for obtaining detailed information on sensitivity and uncertainty of parameters once a global optimisation has been successfully implemented with *hydroPSO*. In particular, this function would allow linking *hydroPSO* with tools using PEST-based templates such as OSTRICH, UCODE-2005, and squads/MADS.
8. The *hydroPSO* package includes a series of n-dimensional functions commonly used as benchmarks to assess the performance of optimisation algorithms. These functions are described in the test_functions() function.

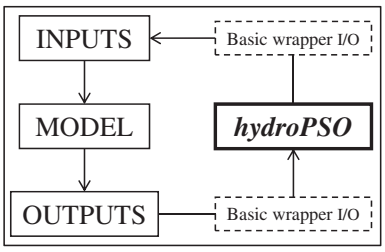


**Fig. 1.** Flowchart of the interaction between *hydroPSO* and the model code to be calibrated. Dashed-line boxes represent basic I/O wrapper functions (not strictly necessary) to read/write model files.

### 3.4. PSO variants and fine-tuning options included in hydroPSO

In addition to the Canonical PSO and the SPSO-2007 & 2011 discussed in Section 2, *hydroPSO* includes three PSO variants that could easily be customised by potential users: (i) the Fully Informed Particle Swarm (FIPS), (ii) the weighted FIPS (wFIPS) by Mendes et al. (2004), where information drawn from all the particles in the neighbourhood contributes to adjusting the particle's velocity, and (iii) the Improved PSO (IPSO) by Zhao (2006), where the term representing the social influence (i.e., $\overrightarrow{G}^t - \overrightarrow{X}_i^t$) is enhanced by using information drawn from a subset containing the $n_{gb}$ best performing particles in the neighbourhood. This last PSO variant shows some similarity with an adaptive PSO algorithm termed TRIBES (Cooren et al., 2009).

At the same time, *hydroPSO* includes numerous fine-tuning options for customizing the calibration engine to specific user requirements. All these options as well as the corresponding *hydroPSO* argument are summarised in Table 2.

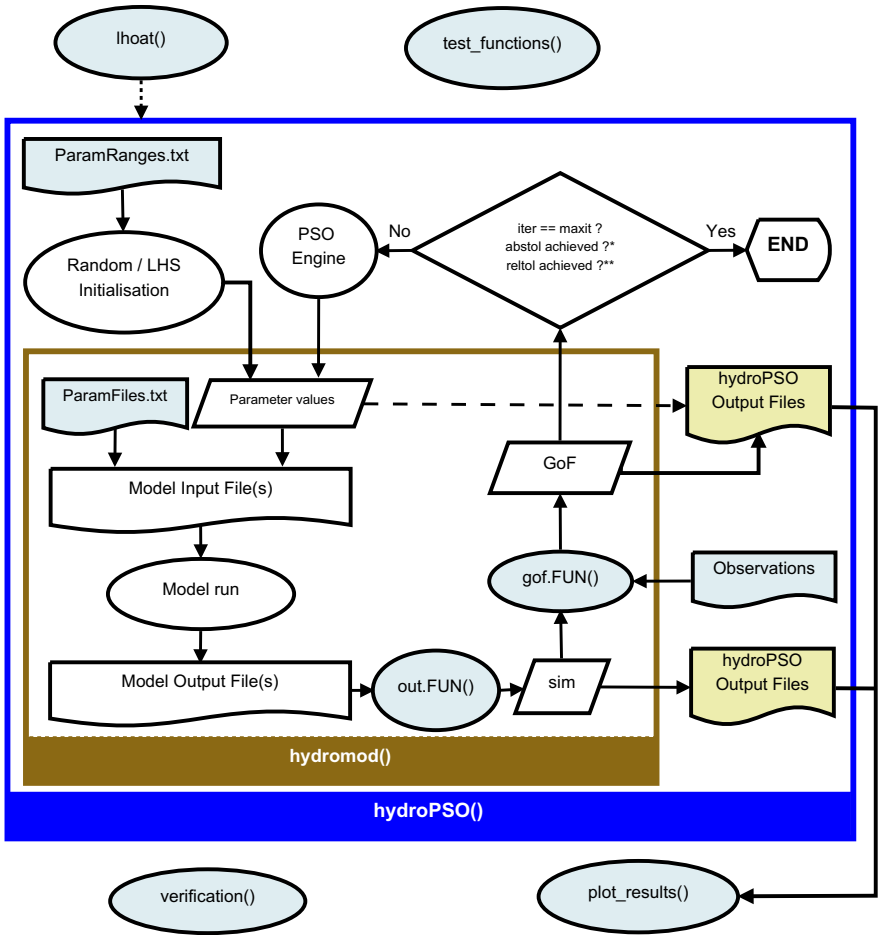


**Fig. 2.** Flowchart describing the interaction of the main *hydroPSO* functions. User-defined files ParamRanges.txt and ParamFiles.txt provide information on the parameters to be calibrated, whereas out.FUN(), gof.FUN(), and *observations* are used to assess the quality of the particles' positions through a Goodness-of-Fit measure. Light-blue shaded boxes require user intervention.

**Table 1**
Main functions included in the *hydroPSO* R package.

| Function | Short description |
|---|---|
| lhoat() | Sensitivity analysis using LH-OAT (van Griensven et al., 2006) |
| hydromod() | Control of the model code to be calibrated |
| hydroPSO() | Multi-OS and model-independent enhanced PSO calibration engine |
| read_results() | Reading of results produced by *hydroPSO*. It is a wrapper to: |
| read_particles() | Reading "*Particles.txt*" output file |
| read_velocities() | Reading "*Velocities.txt*" output file |
| read_out() | Reading "*Model_out.txt*" output file |
| read_convergence() | Reading "*ConvergenceMeasures.txt*" output file |
| read_GofPerParticle() | Reading "*Particles_GofPerIter.txt*" output file |
| plot_results() | Plotting of results produced by *hydroPSO*. It is a wrapper to: |
| plot_particles() | Plotting parameters |
| plot_velocities() | Plotting the evolution of particles' velocities |
| plot_out() | Plotting model outputs (simulated equivalents) |
| plot_convergence() | Plotting the evolution of the global optimum and $\delta^t_{\text{norm}}$ |
| plot_GofPerParticle() | Plotting the evolution of the goodness-of-fit per particle |
| verification() | Run the model code with one or more parameter sets specified by the user |
| PEST2hydroPSO() | Import PEST (Doherty, 2010) files into input files for *hydroPSO* |
| hydroPSO2PEST() | Export *hydroPSO* files into input files for PEST |
| test_functions() | Implementation of a set of *n*-dimensional test functions for benchmarking |

## 4. Comparison of *hydroPSO* against standard algorithms

The aim of this section is to assess the performance of *hydroPSO* against commonly used optimisation algorithms. In particular, we concentrate on the evaluation of two aspects: *effectiveness*, i.e. the ability to find optimal or near-optimal solutions, and *efficiency*, i.e. finding (near-) optimal solutions with the lowest number of function evaluations.

### 4.1. Experimental setup

We employed ten benchmarking functions commonly used for the assessment of (global) optimisation algorithms and representative of a wide class of response surfaces (see Table 3 and Appendix B). Diverse functions were selected to avoid biased results towards a given set of response surfaces, and to assess the performance of the algorithms when the optimum was not located at the origin of the coordinate system. Functions were analysed in a 10-, 20-, and 30-dimensional space.

Each benchmark function was optimised using six algorithms, namely, Gauss-Marquardt-Levenberg (GML) (Marquardt, 1963) (implemented in PEST 12.3.1 by Doherty, 2010), Shuffled Complex Evolution (SCE_UA) (Duan et al., 1992) (implemented in the *hydromad* R package 0.9–15 by Andrews et al., 2011), Differential Evolution (DE) (Storn and Price, 1997) (implemented in the *DEoptim* R package 2.3.1 by Mullen et al., 2011), Differential

**Table 2**
Main features for customising the *hydroPSO* calibration engine.

| Feature | *hydroPSO* argument | Description | References |
|---|---|---|---|
| 1. PSO Variants | method = "spso2011" | 1.1. Standard PSO 2011 | Clerc (2012) |
| | method = "spso2007" | 1.2. Standard PSO 2007 | Clerc (2012) |
| | method = "fips" | 1.3. Fully Informed Particle Swarm | Mendes et al. (2004) |
| | method = "wfips" | 1.4. Weighted FIPS | Mendes et al. (2004) |
| | method = "ipso" | 1.5. Improved PSO (IPSO) | Zhao (2006) |
| | method = "canonical" | 1.6. Canonical PSO | Clerc (2007) |
| 2. Topologies | topology = "random" | 2.1. Random ($k = 3$) | Clerc (2012) |
| | topology = "lbest" | 2.2. *local best* | Kennedy and Mendes (2002) |
| | topology = "vonNeumann" | 2.3. von Neumann | Kennedy and Mendes (2002) |
| | topology = "gbest" | 2.4. *global best* | Kennedy and Mendes (2002) |
| 3. Boundary conditions | boundary.wall = "absorbing2011" | 3.1. Absorbing following SPSO-2011 | Clerc (2012) |
| | boundary.wall = "absorbing2007" | 3.2. Absorbing following SPSO-2007 | Clerc (2012) |
| | boundary.wall = "reflecting" | 3.3. Reflecting | Robinson and Rahmat-Samii (2004) |
| | boundary.wall = "invisible" | 3.4. Invisible | Robinson and Rahmat-Samii (2004) |
| | boundary.wall = "damping" | 3.5. Damping | Huang and Mohan (2005) |
| 4. Definition of $X_{ini}$ | Xini.type = "random" | 4.1. Random | Eberhart and Kennedy (1995) |
| | Xini.type = "lhs" | 4.2. Latin Hypercube Sampling (LHS) | *hydroPSO* |
| 5. Definition of $V_{ini}$ | Vini.type = "random2011" | 5.1. Random following SPSO-2011 | Clerc (2012) |
| | Vini.type = "lhs2011" | 5.2. LHS following SPSO-2011 | Clerc (2012) |
| | Vini.type = "random2007" | 5.3. Random following SPSO-2007 | Clerc (2012) |
| | Vini.type = "lhs2007" | 5.4. LHS following SPSO-2007 | Clerc (2012) |
| | Vini.type = "zero" | 5.5. Zero velocity | *hydroPSO* |
| 6. Definition of inertia weight $\omega$ | use.IW = TRUE, IW.w = 1/(2*log(2)) | 6.1. Linear variation | |
| | IW.type = "linear" | $\omega_{iter} = \left[\frac{iter_{max} - iter}{iter_{max}}\right](\omega_{ini} - \omega_{fin}) + \omega_{fin}$ [a] | Shi and Eberhart (1998a) |
| | IW.type = "non-linear" | 6.2. Non-linear variation | |
| | | $\omega_{iter} = \left[\frac{iter_{max} - iter}{iter_{max}}\right]^{n}(\omega_{ini} - \omega_{fin}) + \omega_{fin}$ [a] | Chatterjee and Siarry (2006) |
| | IW.type = "aiwf" | 6.3. Adaptive Inertia Weight Factor (AIWF) | |
| | | $\omega_{iter} = \begin{cases} \left[\frac{(\omega_{max} - \omega_{min})(f - f_{min})}{f_{avg} - f_{min}}\right] + \omega_{min} & ,f \le f_{avg} \\ \omega_{max} & ,f > f_{avg} \end{cases}$ [b] | Liu et al. (2005) |
| | IW.type = "GLratio" | 6.4. Global-Local best ratio inertia weight | |
| | | $\omega_{iter} = 1.1 - \frac{G_d^t}{P_{id}^t}$ [c] | Arumugam and Rao (2008) |
| | IW.type = "runif" | 6.5. Random inertia weight | |
| | | $\omega_{iter} = 0.5 + \frac{rnd(\omega_{ini}, \omega_{fin})}{2}$ [d] | Eberhart and Shi (2001) |
| 7. Time-varying acceleration coefficients | use.TVc1 = FALSE, c1 = 0.5log(2) | 7.1.1. Linear variation | |
| | TVc1.type = "linear" | $c1_{iter} = \left[\frac{iter}{iter_{max}}(c1_{fin} - c1_{ini})\right] + c1_{ini}$ [e] | Ratnaweera et al. (2004) |
| | TVc1.type = "non-linear" | 7.1.2. Non-linear variation | |
| | | $c1_{iter} = \left[\frac{iter}{iter_{max}}(c1_{fin} - c1_{ini})\right]^{n} + c1_{ini}$ [e] | *hydroPSO* |
| | TVc1.type = "GLratio" | 7.1.3. Global–Local best ratio | |
| | | $c1_{iter} = 1.0 + \frac{G_d^t}{P_{id}^t}$ | Arumugam and Rao (2008) |
| | use.TVc2 = FALSE, c2 = 0.5log(2) | 7.2.1. Linear variation | |
| | TVc2.type = "linear" | $c2_{iter} = \left[\frac{iter}{iter_{max}}(c2_{fin} - c2_{ini})\right] + c2_{ini}$ [e] | Ratnaweera et al. (2004) |
| | TVc2.type = "non-linear" | 7.2.2. Non-linear variation | |
| | | $c2_{iter} = \left[\frac{iter}{iter_{max}}(c2_{fin} - c2_{ini})\right]^{n} + c2_{ini}$ [e] | Ratnaweera et al. (2004) |
| 8. Time-varying maximum velocity | use.TVlambda = FALSE | $\vec{V}_{max}^{iter} = \lambda^{iter}\vec{X}_{max}$ | |
| | TVlambda.type = "linear" | 8.1. Linear variation | |
| | | $\lambda^{iter} = \left[\frac{iter}{iter_{max}}(\lambda^{fin} - \lambda^{ini})\right] + \lambda^{ini}$ [g] | Chatterjee and Siarry (2006) |
| | TVlambda.type = "non-linear" | 8.2. Non-linear variation | |
| | | $\lambda^{iter} = \left[\frac{iter}{iter_{max}}(\lambda^{fin} - \lambda^{ini})\right]^{n} + \lambda^{ini}$ [g] | *hydroPSO* |
| 9. Regrouping | use.RG = FALSE | 9.1. swarm radius $= \delta^t = \text{median}\|\vec{X}_i^t - \vec{G}^t\|$ [f] | Evers and Ghalia (2009) |
| | RG.thr = $\epsilon$ | normalised swarm radius $= \delta_{norm}^t = \frac{\delta^t}{diam(\Psi)} < \epsilon$ | |
| | | $diam(\Psi) = \|range_d(\Psi)\|$ | |
| | | $range_d(\Psi) = X_d^U - X_d^L$ | |
| | | $range_d(\Psi^r) = \min\{range_d(\Psi^0), \rho \max\|X_{id}^{r-1} - G_d^{r-1}\|\}$ | |
| | RG.r = $\rho$ | $\vec{X}_i = G^{r-1} + \left[\vec{r_3} \cdot range_i(\Psi^r) - \frac{1}{2} range_i(\Psi^r)\right]$ | |
| | RG.miniter = $r$ | $V_d^{max} = \lambda\, range_d(\Psi^r)$ | |
| 10. Update of positions and velocities | best.update = "sync" | 10.1. Synchronous | Mussi et al. (2009) |
| | best.update = "async" | 10.2. Asynchronous | Mussi et al. (2009) |
| 11. Normalisation | normalise = FALSE | 11.1. Standard PSO 2011 | Clerc (2012) |

**Table 2** (*continued*)

| Feature | *hydroPSO* argument | Description | References |
|---|---|---|---|
| 12. Parallelisation | parallel = "none" | Type of parallelisation | *hydroPSO* |
| | par.nnodes = NA | number of cores/nodes | *hydroPSO* |
| | par.pkgs = c() | packages required in each core/node | *hydroPSO* |

[a] $\omega_{iter}$ is the current inertia weight, $\omega_{ini}$ and $\omega_{fin}$ are the initial and final inertia weights, and $n$ is the non-linear modulation index.
[b] $f$ is the current objective value of the particle, $f_{avg}$ is the average objective value for all the particles, $f_{min}$ is the minimum objective value of all the particles, $\omega_{max}$ and $\omega_{min}$ are user-defined maximum and minimum values for $\omega$.
[c] It sets a minimisation problem where $G_d^t$ and $\overline{P_{id}^t}$ are the local best and the average of all personal bests during iteration $t$, respectively.
[d] $rnd()$ is a random number in $[\omega_{ini}, \omega_{fin}]$.
[e] $c1_{ini}$, $c1_{fin}$, $c2_{ini}$, $c2_{fin}$ are constants.
[f] $\vec{X}_i^t$ are particles' positions, $\vec{G}^t$ is local best location, $\epsilon$ is the stagnation threshold, $\text{diam}(\Psi)$ is the diameter of the search space, $\text{range}_d(\Psi)$ is the length of the search space along dimension $d$, $X_d^U$ and $X_d^L$ are the upper and lower bounds of the search space, $\Psi^r$ is the regrouped search space, $r$ is the swarm regrouping index, $\Psi^0$ is the original search space, $\rho$ is the regrouping factor, $\vec{r}_3$ is a uniform random vector, and $\lambda$ is a percentage to limit $\vec{V}_{max}$.
[g] $\lambda^{iter}$ is a percentage to limit $\vec{V}_{max}$, and $\lambda^{ini}$ and $\lambda^{fin}$ are the percentage to limit $\vec{V}_{max}$ at the start and end of a given run, respectively.

Evolution Adaptive Metropolis (DREAM) (Vrugt et al., 2009) (implemented in the *dream* R package 0.4–2 by Guillaume and Andrews, 2012 -although DREAM is not intended as an optimisation algorithm it was included upon suggestion from a reviewer-), SPSO-2011 (Clerc, 2012) (implemented in the *hydroPSO* R package 0.3-0, method = "spso2011"), and *hydroPSO*. We configured the PSO-based optimisation engine of *hydroPSO* as follows: LH initialisation of particle positions and velocities, random topology with 11 informants, acceleration coefficients $c_1$ and $c_2$ equal to 2.05, linearly decreasing clamping factor ($\lambda$) for $\vec{V}_{max}$ in the range [1.0,0.5], and use of the Clerc's constriction factor instead of the inertia weight $\omega$.

As suggested by Thyer et al. (1999), in a comparison exercise care must be taken in selecting proper controlling parameters of the algorithms to provide the *fairest* basis for the assessment. In our case, population size, maximum number of function evaluations, and tolerance error were the most relevant controlling parameters to assess the algorithms' performance. In this comparison exercise, for example, tolerance errors (i.e. relative changes in the objective function) were modified to allow each of the six algorithms reach a fixed number of 20,000 function evaluations. In addition, for the population size of each algorithm we used values recommended in the literature, which build upon wealth of experience from the users. For example, SPSO-2011 uses a swarm with 40 particles for all the dimensions (Clerc, 2012); DE uses a population size equal to $10D$ (Mullen et al., 2011); DREAM uses a number of sequences/chains equal to $D$ (Vrugt et al., 2009); whereas SCE_UA uses a population equal to $2D + 1$ for each complex (Andrews et al., 2011). For this last algorithm, we followed the suggestion of a reviewer who recommended setting the number of complexes equal to the dimension of the problem, resulting in a population size for SCE_UA equal to $D(2D + 1)$. For the GML algorithm, we followed values recommended in the PEST manual (Doherty, 2010).

**Table 3**
Benchmark functions used to assess the efficiency and effectiveness of different optimisation algorithms.

| Function | Search space | Global optimum | Acceptance threshold |
|---|---|---|---|
| Sphere (F1) | $[-100, 100]^D$ | 0 | 1e − 02 |
| Rosenbrock (F2) | $[-30, 30]^D$ | 0 | 1e + 02 |
| Rastrigin (F3) | $[-5.12, 5.12]^D$ | 0 | 1e + 02 |
| Griewank (F4) | $[-600, 600]^D$ | 0 | 5e − 02 |
| Ackley (F5) | $[-32, 32]^D$ | 0 | 1e − 02 |
| SchafferF6 (F6) | $[-100, 100]^D$ | 0 | 1e − 05 |
| Shifted Sphere (F7) | $[-100, 100]^D$ | −450 | 1e − 02 |
| Shifted Griewank (F8) | $[-600, 600]^D$ | −180 | 1e − 02 |
| Shifted Ackley (F9) | $[-32, 32]^D$ | −140 | 1e − 02 |
| Shifted Rastrigin (F10) | $[-5.12, 5.12]^D$ | −330 | 1e + 02 |

In summary, we did our best to provide the fairest basis for comparison by adapting the population size based on previous accepted knowledge, fixing the maximum number of function evaluations to obtain a computationally tractable problem, and modifying the tolerance error when required to meet the maximum number of function evaluations given the population size.

Given the stochastic nature of the optimisation problem and to minimise the influence of the initial population on the optimisation results, each algorithm was executed in a loop of 500 runs for each function, starting from randomly selected points in the search domain defined in Table 3. In total, 1.8e + 09 function evaluations were calculated (10 functions × 3 dimensions × 6 algorithms × 20,000 function evaluations × 500 runs).

To assess the *effectiveness* of the algorithms we defined the Success Rate (SR) as the fraction of successful runs out of the total runs (500). An optimisation run was considered successful when the optimum found was below an *acceptance threshold* defined after Kennedy et al. (2001) (see Table 3). At the same time, *efficiency* was assessed by the number of function evaluations required to reach the acceptance threshold over the 500 runs.

### 4.2. Comparison results

We report basic summary statistics for all the functions and dimensions evaluated as well as the success rate (SR) (see Figures in Supplementary material). It is clear from these results that none of the algorithms can be considered as the "*best*" for all the functions and all the dimensions. This is in full agreement with the "No Free Lunch" theorem by Wolpert and Macready (1997).

Fig. 3 shows boxplots summarising the number of function evaluations required to obtain a successful run (efficiency) and the success rate (SR, effectiveness) for $D = 20$, which can be considered as an intermediate case for the three dimensions analysed. Here we see that *hydroPSO* proved to be more efficient than all the other global algorithms (SCE_UA, DE, DREAM, SPSO-2011), requiring a median number of function evaluations lower than 10e + 03 (or 5e + 03 for some cases) to meet the acceptance threshold. It is important to notice that the configuration proposed in this work for *hydroPSO* (see Section 4.1) is more efficient than the SPSO-2011 in all benchmark functions, in particular, for the shifted versions (F7, F8, F9, F10). We believe the increase in efficiency is due to the higher number of informants ($K = 11$) used in the proposed configuration. *hydroPSO* also showed to be equally or more effective than the other global algorithms in 9 out of 10 benchmark functions. Notwithstanding some algorithms show a high SR value given an acceptance threshold, the final optimal value may be located far away from the known optimum. Fig. 4 illustrates this point clearly. With the exception of the Rastrigin and Schaffer F6

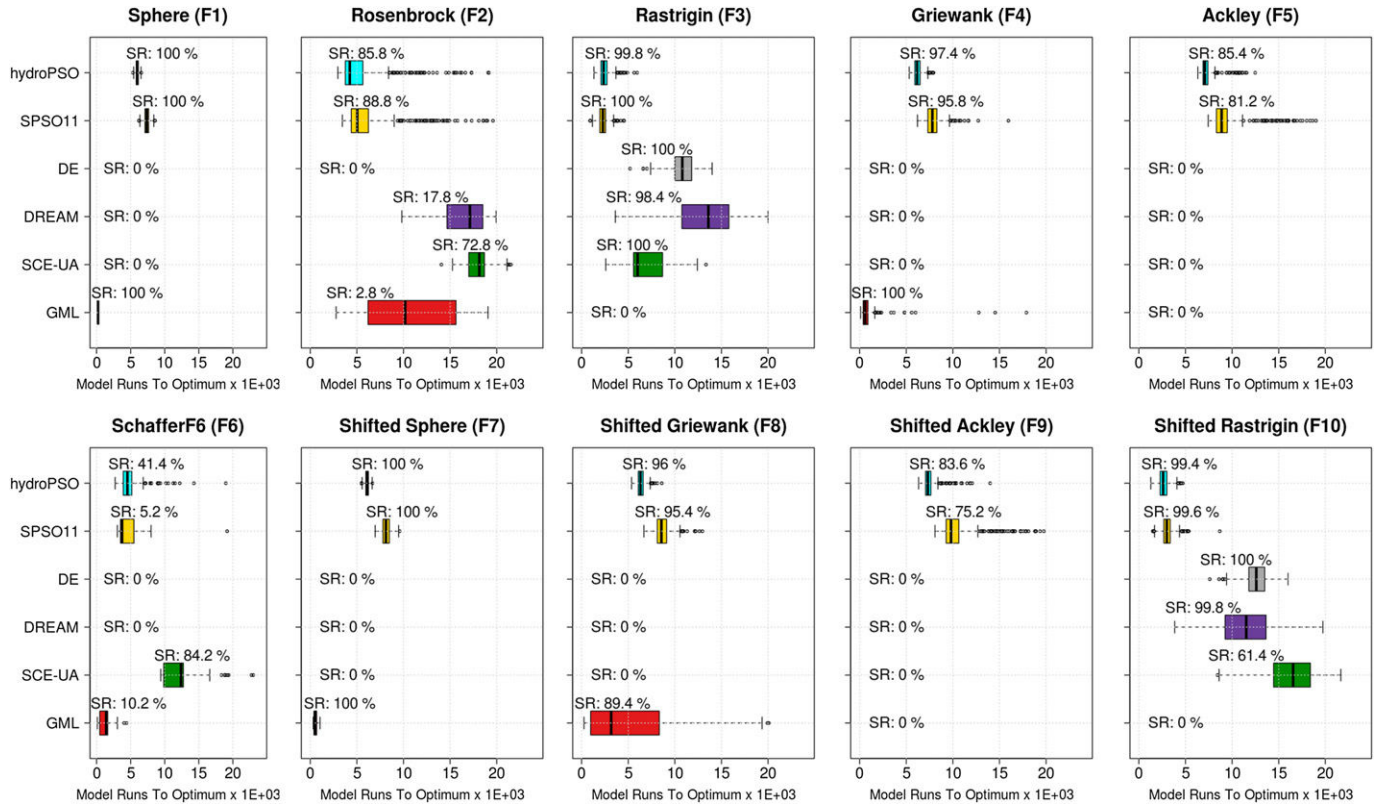


**Fig. 3.** Box-plots summarising the success rate (SR) and the number of function evaluations required to achieve the acceptance thresholds defined in Table 3 for $D = 20$.

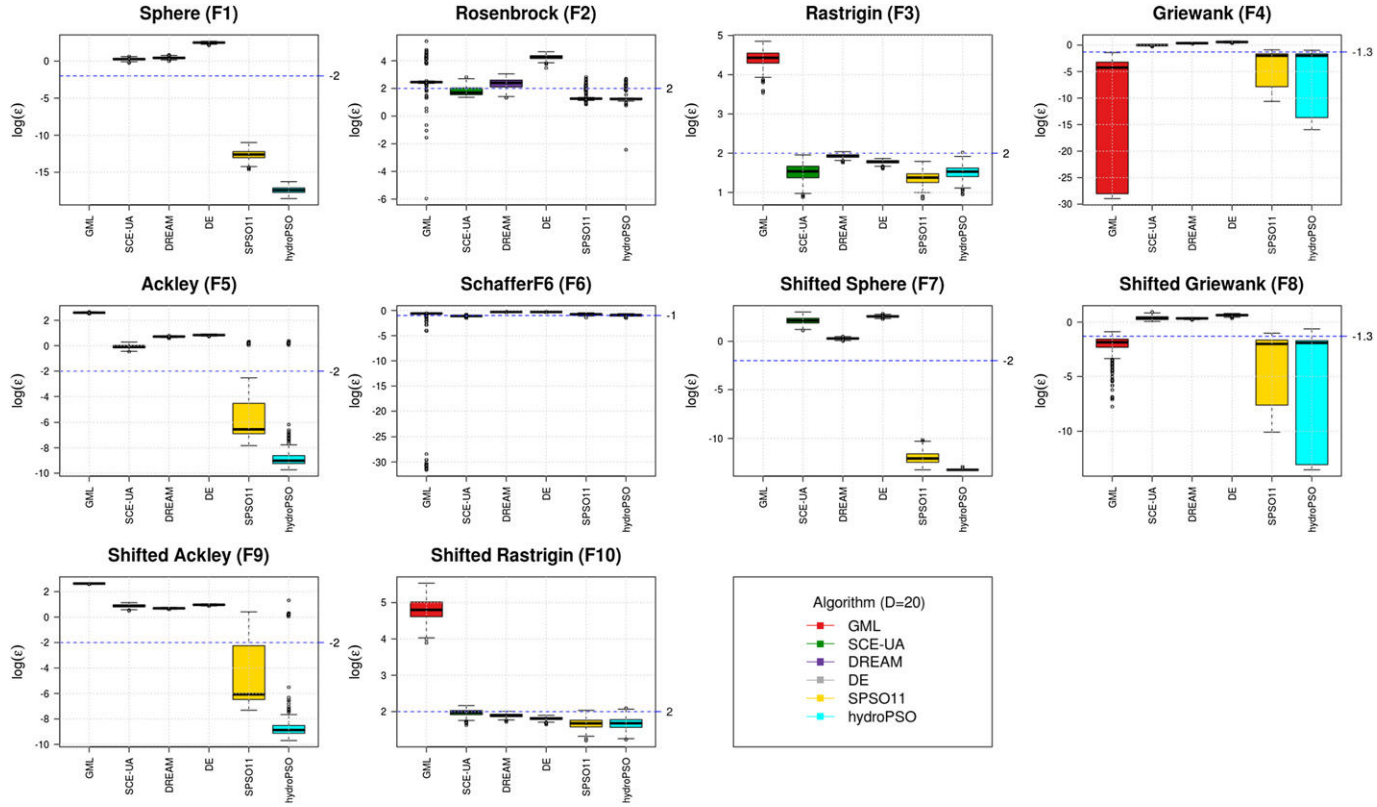


**Fig. 4.** Box-plots summarising the final optimum value found by the algorithms defined in Table 3 for $D = 20$. $\epsilon$ is the difference between the known global optimum and the final optimised value (for $\epsilon < 1e-100$ box-plot is not shown for benefit of the visual inspection).

function (F3, F6 and F10), *hydroPSO* shows a higher proportion of final optimised values closer to the known optimum compared to the other algorithms.

It is worth noting that GML outperforms the other algorithms for the Sphere function (F1 and F7) for all the dimensions. The latter is expected for gradient-based techniques optimising unimodal, continuous, convex, and derivable functions. At the same time, a counter-intuitive result was the performance of GML for the multimodal Griewank function (F4 and F8). This is explained by the fact that this function becomes easier to optimise for an increasing number of dimensions, even though the number of local minima increases exponentially with $D$ (see Locatelli, 2003). In most of the multimodal functions and for all the dimensions, however, GML presented a SR close to zero, highlighting the limitations of "local search" algorithms for solving complex multimodal problems.

One important finding of this exercise was the "*scalable*" behaviour of *hydroPSO* (and to a lesser extent SPSO-2011), i.e. keeping a similar SR across all the three dimensions (see, e.g., SR values for specific functions across different dimensions in the Supplementary material). Most of the other algorithms showed an important decrease in SR when moving from 10 to 20 dimensions. Although not shown here, the evolution of the optimum value along the function evaluations indicates that these algorithms would require a larger number of function evaluations to achieve the acceptance threshold compared to *hydroPSO*. This highlights the high efficiency of *hydroPSO* for high dimensional problems. An unexpected finding for *hydroPSO* (and for SPSO-2011) was the lower SR for the multimodal Griewank function for $D = 10$. We attribute this result to the excessive number of particles (40) used for optimising a 10-dimensional problem.

In summary, in terms of efficiency *hydroPSO* was always superior compared to the other global algorithms (SCE_UA, DE, DREAM, SPSO-2011) across all three dimensions analysed. In addition, in terms of effectiveness *hydroPSO* was superior in 6, 9, and 9 out of 10 benchmark functions for 10, 20 and 30 dimensions, respectively.

## 5. Case studies and *hydroPSO* implementation

This section presents the application of *hydroPSO* to calibrate a surface hydrological model implemented in SWAT-2005 (Neitsch et al., 2005), and a steady-state groundwater flow model implemented in MODFLOW-2005 (Harbaugh, 2005). Distinctive features of these two applications are summarised in Table 4. For implementation details of *hydroPSO* we refer the reader to the accompanying tutorial developed for the R package, which is available from http://www.rforge.net/hydroPSO/.

It should be noted that the *hydroPSO* application to calibrate MODFLOW-2005 differs from the SWAT-2005 application in two main aspects. Firstly, a user-defined Gaussian likelihood is used as a goodness-of-fit-measure. Secondly, the sensitivity analysis using the lhoat() function is skipped. These decisions are taken to allow a comparative assessment between *hydroPSO* and an extensive Markov Chain Monte Carlo (MCMC) simulation performed for the same groundwater model by Rojas et al. (2010).

### 5.1. Calibration of a semi-distributed surface hydrological model for the Ega River Basin, Spain

#### 5.1.1. Study area

The Ega River, a tributary of the Ebro River, flows through the province of Navarra (see Fig. 5). It has an area of 1445 km$^2$ and elevations ranging from 300 up to 1400 m above sea level (a.s.l.) (CHE, 2000). For the implementation of *hydroPSO* we concentrate on the upper Ega catchment, which has an area of 808 km$^2$, mean annual precipitation of ca. 818 mm year$^{-1}$, and mean daily discharge equal to 12.5 m$^3$ s$^{-1}$ measured in Ega en Estella for the period 1961–1990 (Q071 in Fig. 5).

The dominant soils in the catchment comprise marlstones, argillaceous marlstones, and breccia, whereas the dominant land uses are forest (57%), pasture (39%), agriculture (3%), rocks (<1%) and urban areas (<1%). Daily precipitation estimates, obtained from interpolation of data from four rain gauges (P9175, P9176, P9095, P9177U), and daily (max/min) temperatures are used as main meteorological drivers for the hydrological simulations (see Fig. 5).

#### 5.1.2. Hydrological modelling

Simulated daily discharges are obtained with the Soil and Water Assessment Tool (SWAT) 2005 (Neitsch et al., 2005). We set up SWAT-2005 with the modified SCS curve number to obtain surface runoff, the Priestley-Taylor method for computing evapotranspiration, and the Muskingum procedure for the channel routing (see Neitsch et al., 2005). Information on weather, topography, soil properties, and land use was pre-processed in order to prepare the input files required by SWAT-2005.

#### 5.1.3. Sensitivity analysis

Table 5 shows a subset of 22 SWAT-2005 parameters selected as relevant for the hydrological simulations implemented here. The sensitivity analysis is performed on these parameters by using the lhoat() function (see Section 3.3). lhoat() is implemented for the period Jan/1962–Dec/1970 with the Nash-Sutcliffe efficiency (NSE) (Nash and Sutcliffe, 1970) as measure of model performance. NSE is defined by

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{n} \left(Q_i^s - Q_i^o\right)^2}{\sum_{i=1}^{n} \left(Q_i^o - \overline{Q^o}\right)^2} \tag{4}$$

where $n$ is the number of observations, $Q_i^o$ and $Q_i^s$ are the observed and simulated equivalent discharges values at day $i$, respectively, and $\overline{Q^o}$ is the arithmetic mean of the observed discharges. NSE measures the fraction of the variance in the observed flows

**Table 4**
Distinctive features of the *hydroPSO* applications presented in Section 5.

| Application feature | SWAT-2005 | MODFLOW-2005 |
|---|---|---|
| Operating system | GNU/Linux | Windows 7 |
| Type of model | Semi-distributed, surface hydrology | Fully-distributed, groundwater |
| *hydroPSO*-model interface | Basic – ASCII files | Advanced – R functions |
| Executable model code | Single file (swat2005.out) | Sequential batch file (*.bat) |
| Simulated model outputs | Transient (1962–1970), single observation | Steady-state (1960), multiple observations |
| Goodness-of-fit measure | Pre-defined, Nash-Sutcliffe Efficiency | User-defined, Gaussian likelihood |
| Single forward running time | 1.0 s | 0.8–2.7 s |
| Optimisation running time | | |
| Serial (parallel) | 24.1 (7.4) h | 100.3 (87.6) h |
| Number of cores/nodes available | 6 | 6 |

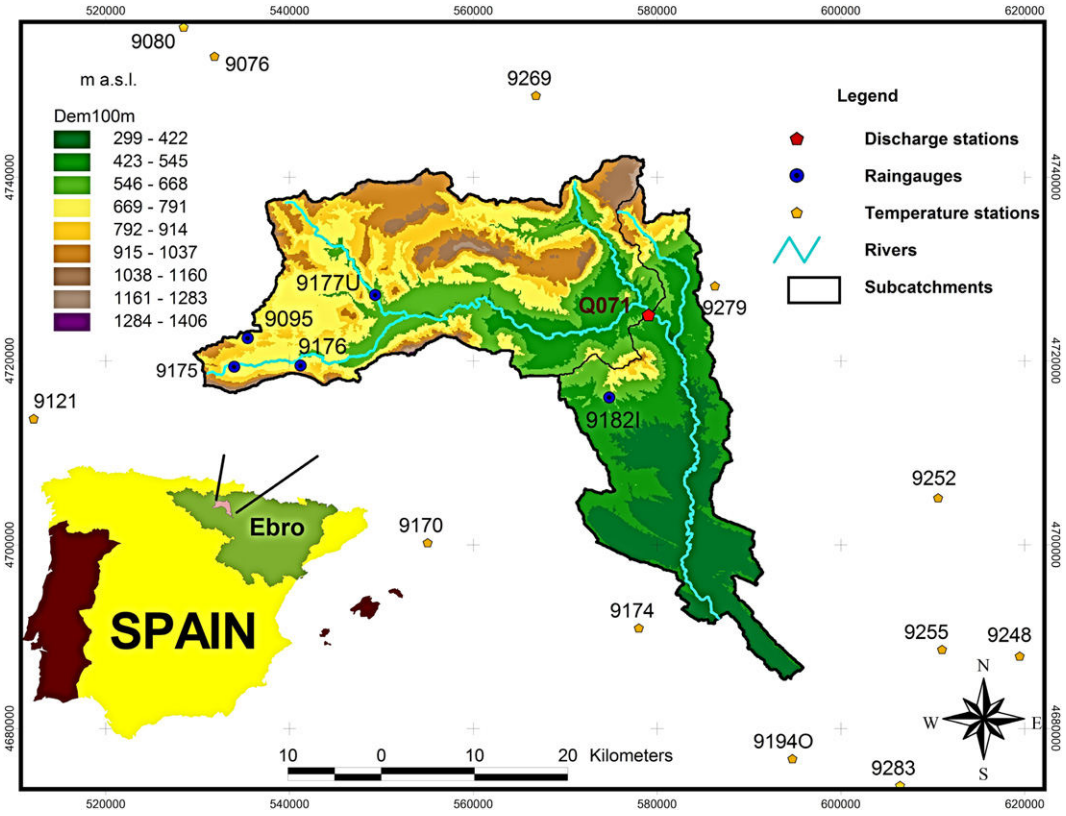


**Fig. 5.** Location of the Ega River basin, meteorological stations, and discharge station used for the calibration of the upper catchment.

**Table 5**
Parameters of the SWAT-2005 model relevant for hydrological simulation. Range and sensitivity ranking constitute the basis for the implementation of the *hydroPSO* package to calibrate SWAT-2005 in the upper Ega catchment.

| Parameter | Parameter name | SWAT-2005 file | Range | | Default value | Sensitivity ranking[a] |
|---|---|---|---|---|---|---|
| | | | Min | Max | | |
| Base flow alpha factor [days] | ALPHA_BF | *.gw | 1.00e − 02 | 9.90e − 01 | 4.80e − 02 | 1 |
| Manning's "*n*" value for the main channel [−] | CH_N2 | *.rte | 1.60e − 02 | 1.50e − 01 | 1.40e − 02 | 2 |
| Initial SCS CN II value [−] | CN2 | *.mgt | 4.00e + 01 | 9.50e + 01 | 5.21e + 01[b] | 3 |
| Saturated hydraulic conductivity [mm/h] | SOL_K | *.sol | 1.00e − 03 | 1.00e + 03 | 4.28e + 00[b] | 4 |
| Available water capacity [mm $H_2O$/mm soil] | SOL_AWC | *.sol | 1.00e − 02 | 3.50e − 01 | 1.20e − 01[b] | 5 |
| Effective hydraulic conductivity in main channel alluvium [mm/h] | CH_K2 | *.rte | 0.00e + 00 | 2.00e + 02 | 0.00e + 00 | 6 |
| Soil evaporation compensation factor [−] | ESCO | *.hru | 1.00e − 02 | 1.00e + 00 | 9.50e − 01 | 7 |
| Surface runoff lag time [days] | SURLAG | *.bsn | 1.00e + 00 | 1.20e + 01 | 4.00e + 00 | 8 |
| Snowfall temperature [°C] | SFTMP | *.bsn | −5.00e + 00 | 5.00e + 00 | 1.00e + 00 | 9 |
| Snowmelt base temperature [°C] | SMTMP | *.bsn | −5.00e + 00 | 5.00e + 00 | 5.00e − 01 | 10[c] |
| Minimum melt factor for snow [°C] | SMFMN | *.bsn | 1.40e + 00 | 6.90e + 00 | 4.50e + 00 | 11[c] |
| Snowpack temperature lag factor [−] | TIMP | *.bsn | 1.00e − 02 | 1.00e + 00 | 1.00e + 00 | 12[c] |
| Maximum melt factor for snow [°C] | SMFMX | *.bsn | 1.40e + 00 | 6.90e + 00 | 4.50e + 00 | 13[c] |
| Manning's "*n*" value for overland flow [−] | OV_N | *.hru | 8.00e − 03 | 6.00e − 01 | 1.00e − 01 | 14[c] |
| Deep aquifer percolation factor [−] | RCHRG_DP | *.gw | 0.00e + 00 | 1.00e + 00 | 5.00e − 02 | 15[c] |
| Threshold water depth in the shallow aquifer for flow [mm] | GWQMN | *.gw | 0.00e + 00 | 5.00e + 03 | 0.00e + 00 | 16[c] |
| Groundwater "*revap*" coefficient [−] | GW_REVAP | *.gw | 0.00e + 00 | 2.00e − 01 | 2.00e − 02 | 17[c] |
| Groundwater delay time [days] | GW_DELAY | *.gw | 1.00e + 00 | 1.00e + 02 | 3.10e + 01 | 18[c] |
| Moist soil albedo | SOL_ALB | *.sol | 0.00e + 00 | 1.00e − 01 | 1.00e − 02[b] | 19[c] |
| Threshold water depth in the shallow aquifer for "*revap*" [mm] | REVAVMN | *.gw | 1.00e + 00 | 5.00e + 02 | 1.00e + 00 | 20[c] |
| Plant uptake compensation factor [−] | EPCO | *.bsn | 1.00e − 02 | 1.00e + 00 | 1.00e + 00 | 22[c] |
| Maximum canopy storage [mm $H_2O$] | CANMX | *.hru | 0.00e + 00 | 1.00e + 01 | 0.00e + 00 | 22[c] |

[a] Sensitivity analysis based on LH-OAT (see Section 5.1).
[b] Default values based on the study area information.
[c] Insensitive parameters obtained from the LH-OAT analysis.

explained by the model and it has an optimal value of 1. In particular, NSE (and a series of alternative goodness-of-fit measures) is available in a complementary R package (*hydroGOF*) developed by one of the authors of *hydroPSO* (Zambrano-Bigiarini, 2012a).

The sensitivity ranking obtained from the lhoat() function is included in the last column of Table 5. Parameters related to base flow (ALPHA_BF) and surface runoff (CH_N2 and CN2) are identified as the most sensitive. A second group of sensitive parameters relates to physical soil properties, in particular, hydraulic conductivities (SOL_K and CH_K) and water capacity (SOL_AWC). A third group relates to different processes such as soil evaporation (ESCO), surface runoff (SURLAG), and snowfall (SFTMP). Although parameters SMTMP (10th) and SMFMN (11th) also relate to snowfall, several trials of the lhoat() function have shown that these parameters are mildly sensitive, showing a low "relative importance" compared to the 9th parameter (SFTMP). In general, this ranking is fully consistent with previous research on the sensitivity of SWAT-2005 parameters (see e.g., van Griensven et al., 2006; Kannan et al., 2007), and with the characteristics of the study area (mountainous with rainfall-dominated stream flow). Based on this, we selected 9 sensitive parameters for the study area, which constitute the basis for the calibration of the SWAT-2005 model.

We should point out that this ranking is not absolute. Using an alternative goodness-of fit measure that concentrates on a particular spectrum of the flow regime (e.g., low- or high-flows), will most likely result in a different ranking. An analysis of this type, however, is beyond the scope of this article.

### 5.1.4. SWAT-2005 calibration

The calibration period was 9 years (Jan/1962–Dec/1970) with year 1961 used as a warming-up period to minimise the influence of the initial states. This calibration period was selected in order to obtain sufficient variability in the stream flow signal (see Yapo et al., 1996). Similar to the sensitivity analysis, the goodness-of-fit measure corresponds to NSE.

Results from Section 4 and preliminary trials led to the following *hydroPSO* configuration: a swarm of 40 particles, maximum number of iterations equal to 1000 (i.e. 40,000 model runs), random topology ($k = 11$ informants), $X_{ini}$ and $V_{ini}$ initialised using LH sampling on $D = 9$, Clerc's constriction factor active, cognitive and social acceleration coefficients $c_1 = c_2 = 2.05$, linearly decreasing $\lambda$ ($\vec{V}_{max}$ clamping factor) in the range [1.0;0.5], absorbing boundary condition, and synchronous update of the local best (see Table 2 for details on the options). In addition, the parameter space was normalised to the range [0;1] (normalise = TRUE) as recommended by (Clerc, 2012).

### 5.1.5. Calibration results

Fig. 6 shows the evolution of parameter values as a function of the number of model runs. We see that most parameters stabilise around 6000 model evaluations, i.e. around 150 iterations. Although not shown here, several trials including different numbers of iterations, swarm sizes and boundary conditions indicate that these attraction zones are most likely located in the region of the (unknown) optimum value. Table 6 shows the optimum

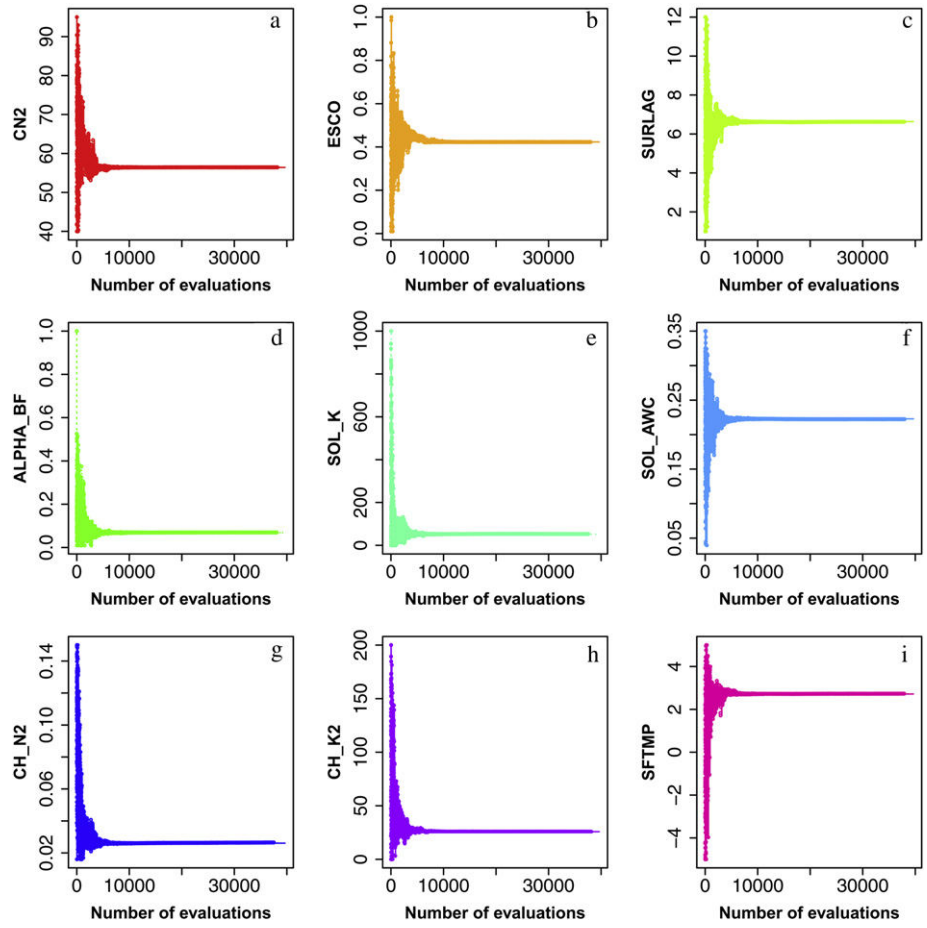


**Fig. 6.** Evolution of parameter values (NSE > 0.5) as a function of the number of model evaluations for the calibration of SWAT-2005 in the upper Ega catchment. Parameter names are described in Table 5.

**Table 6**
Optimised values and 95% confidence intervals obtained from *hydroPSO* for the parametes of the SWAT-2005 model for the upper Ega catchment. Confidence intervals are presented for the full retained sample (40000) and for a "behavioural" subset with NSE > 0.5.

| Parameter | Optimised value | 95% CI full sample | 95% CI "behavioural" sample |
|---|---|---|---|
| ALPHA_BF | 6.86e − 02 | [6.62e − 02; 1.20e − 01] | [6.64e − 02; 1.07e − 01] |
| CH_N2 | 2.61e − 02 | [2.60e − 02; 4.28e − 02] | [2.60e − 02; 3.75e − 02] |
| CN2 | 5.64e + 01 | [5.60e + 01; 6.37e + 01] | [5.60e + 01; 6.31e + 01] |
| SOL_K | 5.20e + 01 | [4.80e + 01; 7.38e + 01] | [4.89e + 01; 6.60e + 01] |
| SOL_AWC | 2.23e − 01 | [2.12e − 01; 2.27e − 01] | [2.15e − 01; 2.25e − 01] |
| CH_K2 | 2.58e − 01 | [2.57e + 01; 4.32e + 01] | [2.58e + 01; 4.03e + 01] |
| ESCO | 4.23e − 01 | [4.18e − 01; 4.87e − 01] | [4.23e − 01; 4.82e − 01] |
| SURLAG | 6.64e + 00 | [5.88e + 00; 6.66e + 00] | [5.97e + 00; 6.65e + 00] |
| SFTMP | 2.72e + 00 | [2.20e + 00; 2.75e + 00] | [2.32e + 00; 2.75e + 00] |

parameter values obtained from *hydroPSO* with the associated 95% confidence intervals obtained from the full parameter samples and from a subset of "behavioural" samples with NSE > 0.5 (see Beven, 2006). This threshold to define behavioural samples is defined on the basis of Moriasi et al. (2007).

Although not shown here, parameter histograms show that the optimum parameters found by *hydroPSO* are located in the attraction zones, indicating a proper identification of the optimum parameter values. From the 9 parameters selected for calibration, ESCO (and to a lesser extent SURLAG) exhibits some spread around the corresponding optimum value, thus indicating a moderate uncertainty around the optimum. In Fig. 7, parameter interactions are further summarised by projecting the NSE response surface onto the parameter space. In this figure we observe clear attraction zones for all parameters around the highest performance values as well as complex interactions amongst parameters such as SOL_K vs. CN_2, and ALPHA_BF vs. SOL_K.

Fig. 8 shows a comparison between (daily and monthly) observed and "best" simulated discharges obtained using the "optimum" parameter set from *hydroPSO* (Table 6). In general, both hydrographs show a good correspondence with the observed time series, with a slight underestimation of recession periods and peak-flows. Relatively low values for the mean absolute error (MAE = 4.7 $m^3 s^{-1}$), root mean square error (RMSE = 8.7 $m^3 s^{-1}$), and percentage bias (PBIAS = −2.6%) together with relatively high values for the model performance (NSE = 0.78), index of agreement ($d$ = 0.93), and correlation coefficient ($r$ = 0.88) indicate that the simulated discharges are acceptable for this case study (see, e.g., Moriasi et al., 2007).

*hydroPSO* also allows the modeller to produce Empirical Cumulative Density Functions (ECDFs), either for parameters or model outputs, at user-defined quantiles. Fig. 9 shows the assessment of three different stream flow spectra: low-, medium- and high-flow, by using the 5th, 50th, and 95th percentiles, respectively. ECDFs are created using a subset of "behavioural" simulations with a NSE > 0.5 (see Moriasi et al., 2007). Simulated stream flow

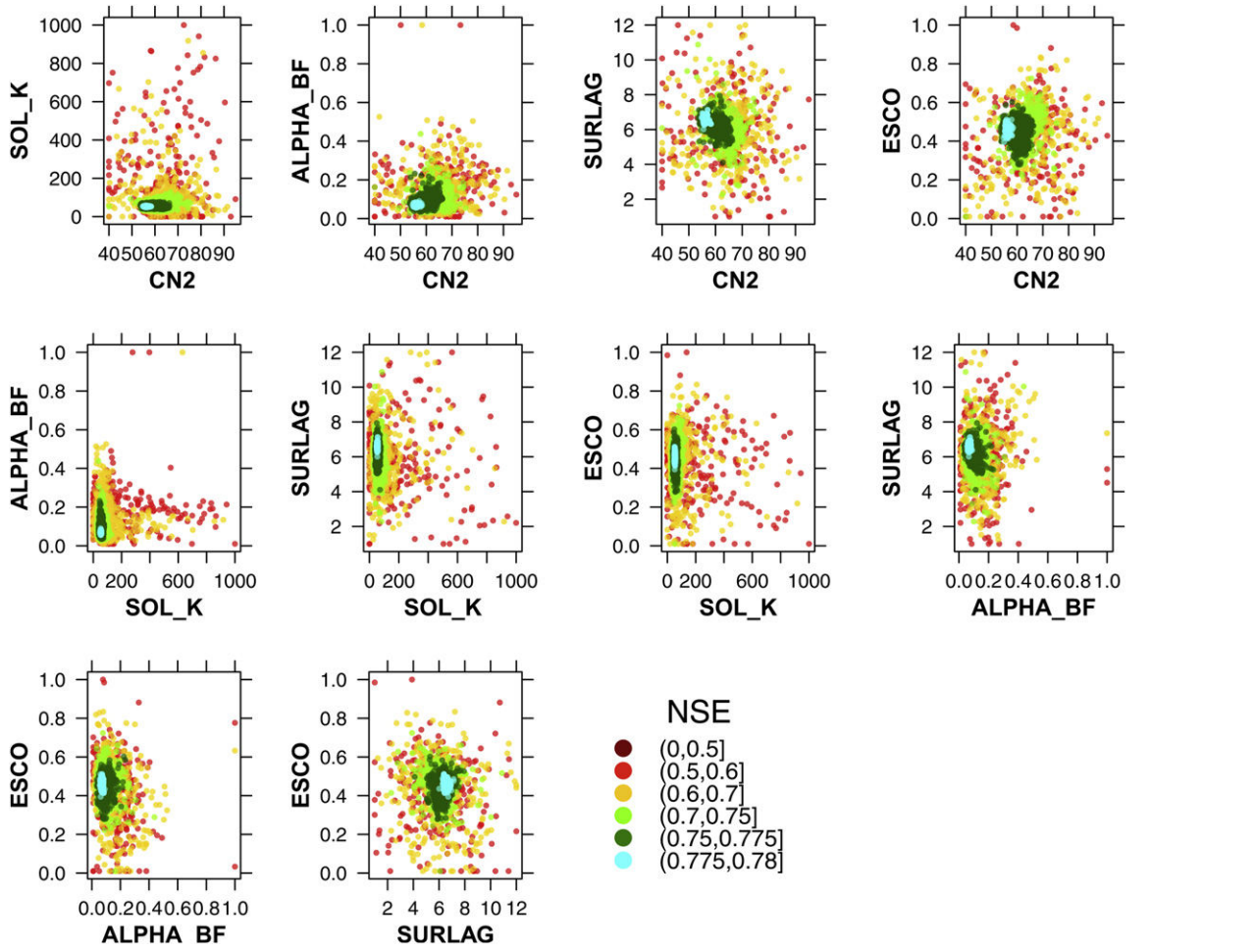


**Fig. 7.** Model performance (NSE) projected onto the parameter space. Panels show "behavioural" samples with NSE > 0.5 for a subset of parameters. Parameter names are described in Table 5.

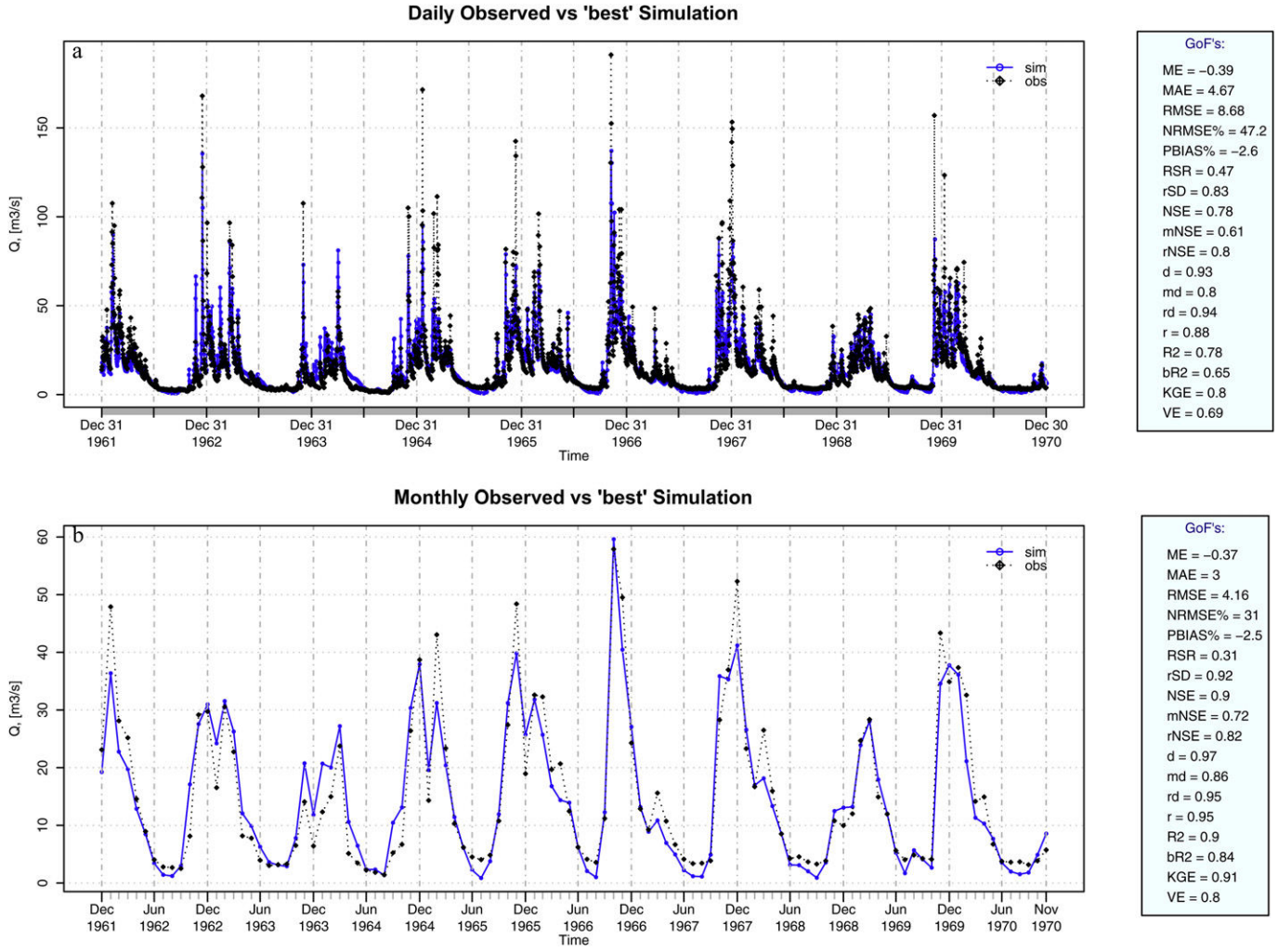


**Fig. 8.** Daily (upper panel) and monthly (lower panel) hydrographs between observed and simulated equivalent discharges obtained using the "optimum" parameter set found by *hydroPSO* during the calibration period Jan/1962–Dec/1970 for the upper Ega catchment.

discharges tend to underestimate low- (−57%) and high-flows (ca. −8%), whereas medium-flows are overestimated (+23%). Slight underestimation of high-flows results from using NSE as the objective function for the optimisation, which has previously been recognised as underestimating flow variability (see Gupta et al., 2009). Underestimation of low-flows, in turn, might be linked to the fact that only one parameter related to the slow-flow component was selected as sensitive by the LH-OAT (ALPHA_BF), or related to flaws in the model structure and/or input data.

### 5.2. Calibration of a steady-state groundwater flow model for the Pampa del Tamarugal Aquifer (PTA), Chile

#### 5.2.1. Study area

The Pampa del Tamarugal Aquifer (PTA) is an unconfined system located in the Atacama Desert covering an area of ca. 5000 km$^2$ (Fig. 10). The aquifer basin is a complex asymmetric graben bounded to the west and east by North-South regional fault systems. The system is recharged by groundwater flows produced from seven eastern sub-basins, which receive precipitation at high altitudes (>3500 m asl) (Rojas et al., 2010) (see Fig. 10a).

In the PTA arid conditions are extreme, which translates as direct evaporation of groundwater from well-defined areas known as "salares" (playas) (see Fig. 10b). In addition, forested areas comprising trees that are highly-adapted to arid and saline conditions promote direct discharge of groundwater through plant transpiration (Rojas et al., 2010). The PTA system exchanges groundwater with contiguous aquifers at the western and southern boundaries, whereas a groundwater divide is shown at the northern boundary. Fig. 10b shows the groundwater elevation map for the year 1960 (steady-state), which is the basis for the model calibration.

#### 5.2.2. Groundwater modelling

Simulation of groundwater flow and balance for the year 1960 were obtained with MODFLOW-2005 (Harbaugh, 2005) and ZONEBUDGET (Harbaugh, 1990), respectively. The groundwater system is conceptualised as a one-layer aquifer with a varying thickness of between ca. 70 m and 300 m. Northern and southern interactions with external forcing are expressed as constant-head boundaries, whereas recharge mechanisms are expressed by lateral fluxes and spatially-distributed recharge rates. Evaporation from "playas" is implemented using the EVT package, whereas transpiration from the forested areas is expressed as negative recharge rates. In particular, the groundwater model implemented in this work corresponds to the model "M2b" used in the MCMC analysis by Rojas et al. (2010).

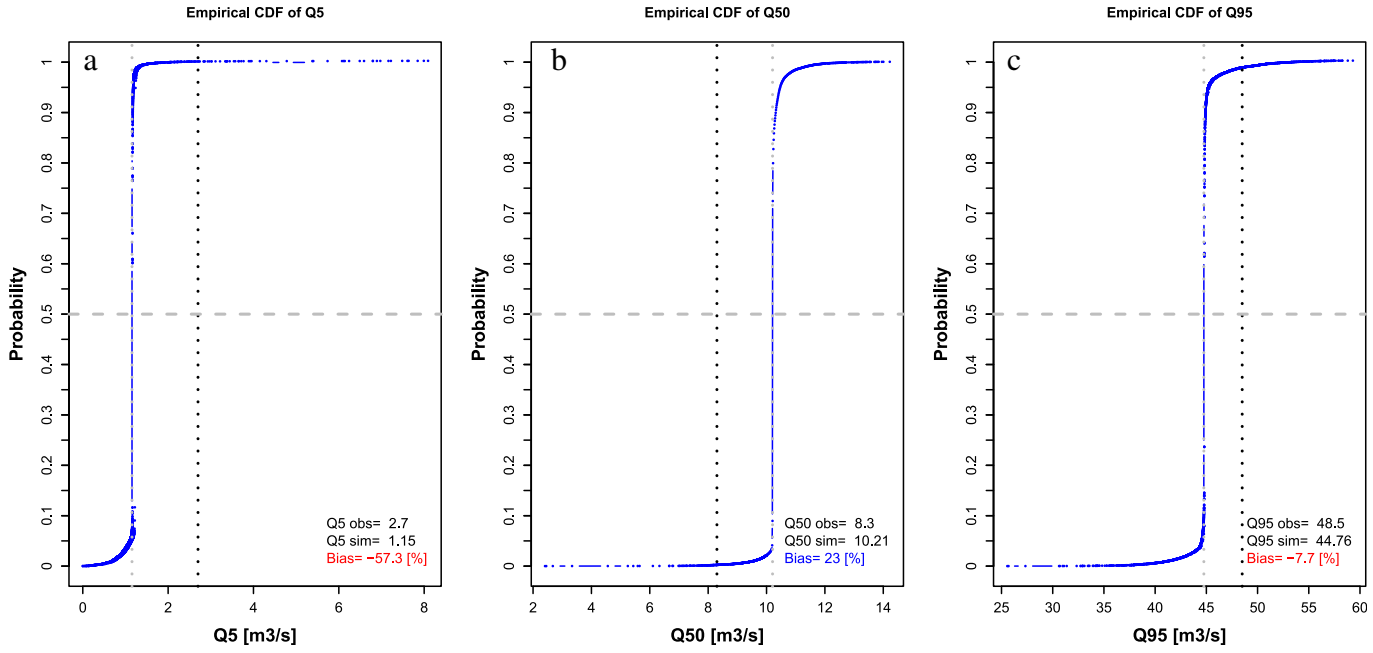


**Fig. 9.** Empirical CDFs (blue lines) for 5th, 50th, and 95th percentiles for the simulated discharges obtained from SWAT-2005 for the upper Ega catchment (Jan/1962–Dec/1970). Vertical black-dotted lines represent the observed percentiles, while vertical grey-dotted lines indicate the (median of the) simulated quantiles.

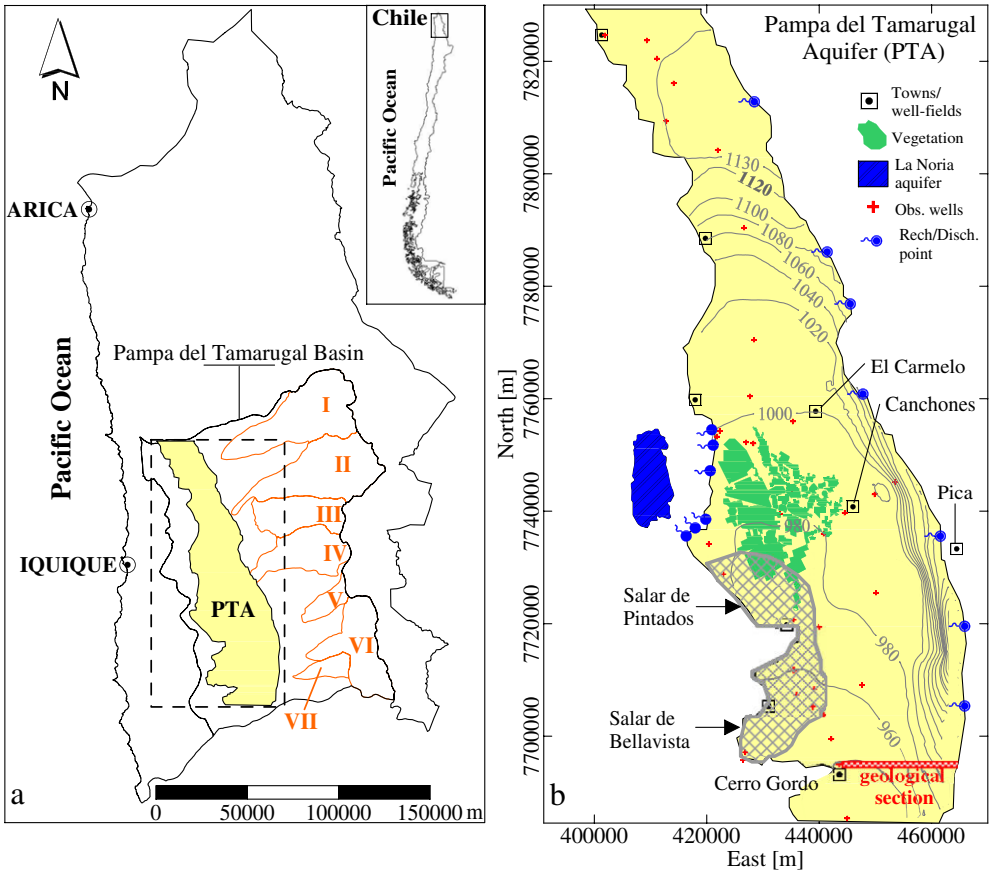


**Fig. 10.** (a) Location of the Pampa del Tamarugal Aquifer (PTA) and seven eastern sub-basins recharging the system, and (b) evaporation zone, transpiration zone, main recharge and discharge points, and groundwater heads (m a.s.l.) for steady-state conditions defined in year 1960 (after Rojas et al., 2010).

**Table 7**
Parameter ranges used to implement *hydroPSO* to calibrate the MODFLOW-2005 groundwater flow model for the PTA.

| Parameter | Parameter name | MF2005 file | Range | |
|---|---|---|---|---|
| | | | Min | Max |
| Recharge [$m^3$ $d^{-1}$] | R_RECH | *.rch | 0.00e + 00 | 3.46e + 05 |
| Recharge from basement rocks [$m^3$ $d^{-1}$] | R_BSMNT | *.rch | 0.00e + 00 | 1.73e + 05 |
| Transpiration forested areas [$m^3$ $d^{-1}$] | R_TRANSP | *.rch | 0.00e + 00 | 1.73e + 05 |
| Discharge to La Noria aquifer [$m^3$ $d^{-1}$] | R_NOR | *.wel | 0.00e + 00 | 8.64e + 04 |
| Evaporation rate [m $d^{-1}$] | R_EVTR | *.evt | 0.00e + 00 | 1.00e − 02 |
| Extinction depth [m] | R_EXTD | *.evt | 0.00e + 00 | 2.00e + 01 |
| Elevation constant head north [m] | R_CHN | *.ba6 | 1.08e + 03 | 1.12e + 03 |
| Elevation constant head south [m] | R_CHS | *.ba6 | 8.75e + 02 | 9.20e + 02 |
| Hydraulic conductivity ($K$) for 22 zones [m $d^{-1}$] | HK1…HK22 | *.lpf | 0.00e + 00 | 1.00e + 02 |

#### 5.2.3. MODFLOW-2005 calibration

Thirty parameters related to recharge, evaporation, transpiration, (north and south) boundary conditions, and hydraulic conductivities were calibrated for the PTA groundwater flow model (see Table 7).

Model performance (goodness-of-fit measure) is assessed through a Gaussian likelihood expressed by

$$L = (2\pi)^{-l/2}|C|^{-1/2}\exp\left(-\frac{1}{2}(h_{sim}-h_{obs})^{T}C^{-1}(h_{sim}-h_{obs})\right) \quad (5)$$

where $C$ is the covariance matrix of the observed system variables, $l$ is the number of observations, and $h_{obs}$ and $h_{sim}$ are the observed and simulated equivalent groundwater heads, respectively. A set comprising 42 observation wells was used to calculate this likelihood, assuming no measurement error or correlation among observations (i.e. off-diagonal terms in $C$ equal to 0). The likelihood for the corresponding *hydroPSO* model evaluation (i.e. parameter set) was obtained using a product inference function as in Rojas et al. (2008, 2010). Following the product inference function, all parameter sets producing at least one simulated equivalent equal to zero (or resulting in a "dry" cell) in an observation well, were rejected. This results in a strict criterion to obtain sampled parameters, which requires more model evaluations but which is fully consistent with the physics dominating the aquifer system.

A swarm of 70 particles and 3000 iterations, i.e. 210,000 model evaluations, were defined to implement *hydroPSO*. Other fine-tuning options were identical to the SWAT-2005 application (see Section 5.1.4). We should note that this number of model evaluations was selected to allow a direct comparison with the MCMC samples obtained by Rojas et al. (2010).

#### 5.2.4. Calibration results

Fig. 11 shows the evolution of the goodness-of-fit measure (Gaussian likelihood) as a function of the number of iterations. An initial phase of exploration of approximately 500 iterations (i.e. 3500 model evaluations) with significant increases in the objective function is observed. Then, marginal improvements in the objective function accompanied by a stabilisation in the normalised swarm radius ($\delta_{norm}$) are observed. The search space covers an attraction zone sized between 46% and less than 0.02% of the full search space. Although there is no guarantee that this attraction zone contains the (unknown) true global optimum, the best likelihood found by *hydroPSO* (3.94e − 02) is superior compared to the highest likelihood obtained in the MCMC analysis by Rojas et al. (2010) (3.29e − 02).

Fig. 12, in turn, shows the observed groundwater heads against the simulated equivalents obtained by using the optimum parameter set found by *hydroPSO*. In general, we observe an excellent correspondence between observed and simulated groundwater heads resulting in MAE of 1.2 m and RMSE of 1.6 m. These results represent a significant improvement in terms of quality of the calibration compared to the MCMC simulation by Rojas et al. (2010), for which the error terms reached MAE = 4.7 m and RMSE = 6.2 m for the simulation showing the highest likelihood.

Effective groundwater flows obtained from ZONEBUDGET are shown in Fig. 13. This figure shows scatter plots of the samples obtained from *hydroPSO*. In general, well-defined attraction zones are observed for the recharge inflows from the eastern sub-basins (Fig. 13a), evaporation from salares (Fig. 13b), and groundwater outflows in the southern aquifer cross section (Fig. 13c). For the PTA outflows to the western La Noria aquifer, a more diffuse likelihood response surface is obtained. Fig. 13 also includes results from an extensive MCMC simulation. In general, MCMC-based likelihood response surfaces show a wider spread, a somewhat different attraction zone (panels a and c), and lower likelihood values than the *hydroPSO*-based samples.

Empirical CDFs (ECDFs) of groundwater heads for a subset of observation wells are shown in Fig. 14. *hydroPSO*-based ECDFs effectively capture the observed groundwater heads, with a few exceptions showing the observation at the tail of the ECDF. Comparatively, *hydroPSO*-based ECDFs are more accurate than their MCMC-based counterpart for this particular case.

## 6. Discussion

Good modelling practice has long been a topic of debate in environmental sciences (see, e.g., Refsgaard and Henriksen, 2004; Argent, 2004; Jakeman et al., 2006). As highlighted by Matott et al. (2009), proper model evaluation activities complementing

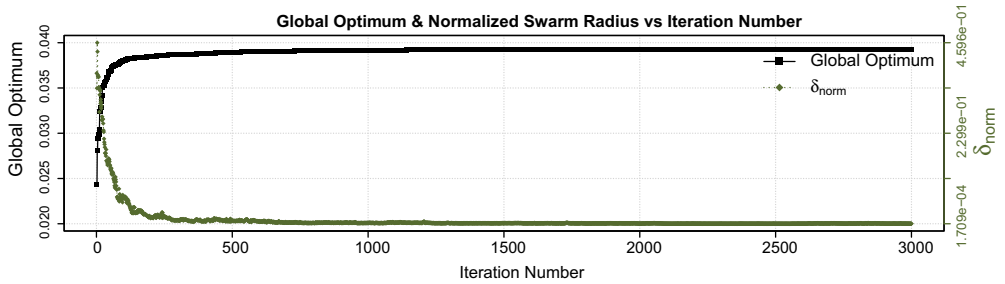


**Fig. 11.** Evolution of the global optimum (Gaussian likelihood) and the normalised swarm radius ($\delta_{norm}$) as a function of the iteration number for the calibration of the PTA.

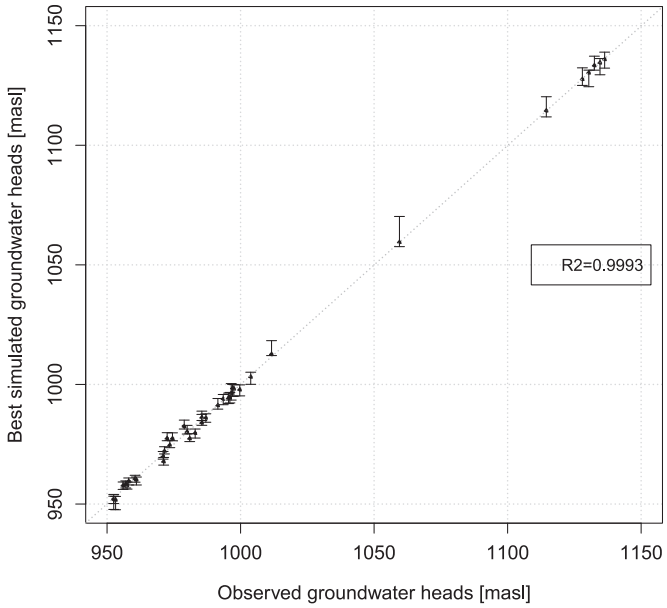


**Fig. 12.** Observed groundwater heads versus simulated equivalents obtained using the optimum parameter set found by *hydroPSO*. Vertical error bars represent the 95% confidence interval obtained from a "behavioural" sample with $L > 3.8e-02$.

parameter optimisation should (ideally) include: input data analysis, sensitivity analysis, and uncertainty analysis. A stand-alone application of *hydroPSO* allows the modeller to implement: sensitivity analysis, parameter calibration through global optimisation, uncertainty assessment using a GLUE-based analysis (see Beven, 2006), and assessment of calibration results. Despite the fact that GLUE-based analysis has recently come under scrutiny in the literature (see, e.g., Mantovan and Todini, 2006; Stedinger et al., 2008), GLUE has become a widely-used alternative to more "formal" Bayesian approaches underpinned in strong assumptions about the functional form of the residuals (see e.g., Beven et al., 2011; Montanari and Koutsoyiannis, 2012). We do acknowledge the limitations of GLUE, e.g., subjective decisions involved in the assessment process and the impossibility to explicitly disentangle the contributing sources of modelling error, however, we believe its widespread use justifies its inclusion in *hydroPSO*.

*hydroPSO* is the "natural" follow-up to two existing R packages, which implement time series management and analysis (*hydroTSM* Zambrano-Bigiarini, 2012b), and alternative goodness-of-fit measures for model evaluation (*hydroGOF* Zambrano-Bigiarini, 2012a). Used together, these packages would allow the modeller to implement a robust model calibration and a proper evaluation of results following the guidelines highlighted by Matott et al. (2009).

In the context of current R packages developed for similar purposes (e.g., *hydromad* by Andrews et al. (2011) and *R-SWAT-FME* by Wu and Liu (2012)), and other general-purpose optimisation packages (e.g., *DEoptim* by Mullen et al. (2011) and *pso* by Bendtsen

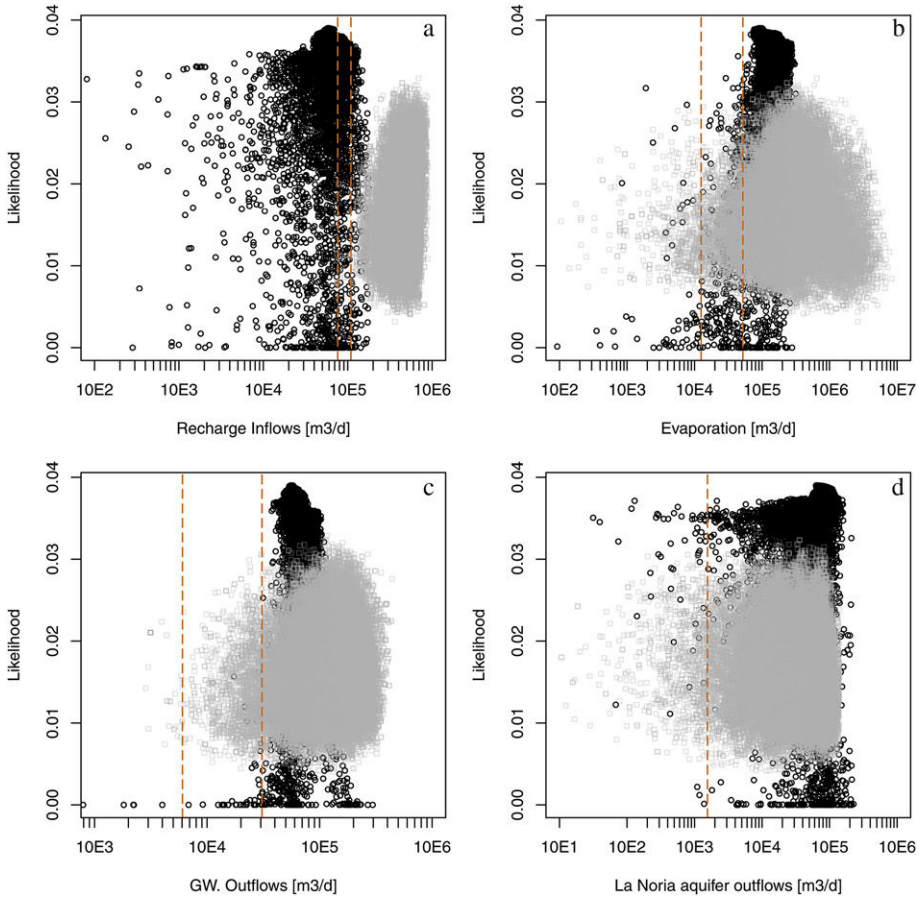


**Fig. 13.** Scatter plots of effective groundwater flows estimated using ZONEBUDGET for the retained samples obtained from *hydroPSO*. Light-grey scatter plots show results from the MCMC analysis performed by Rojas et al. (2010). Vertical dash-lines show estimations from previous studies (see, e.g., Rojas and Dassargues, 2007).

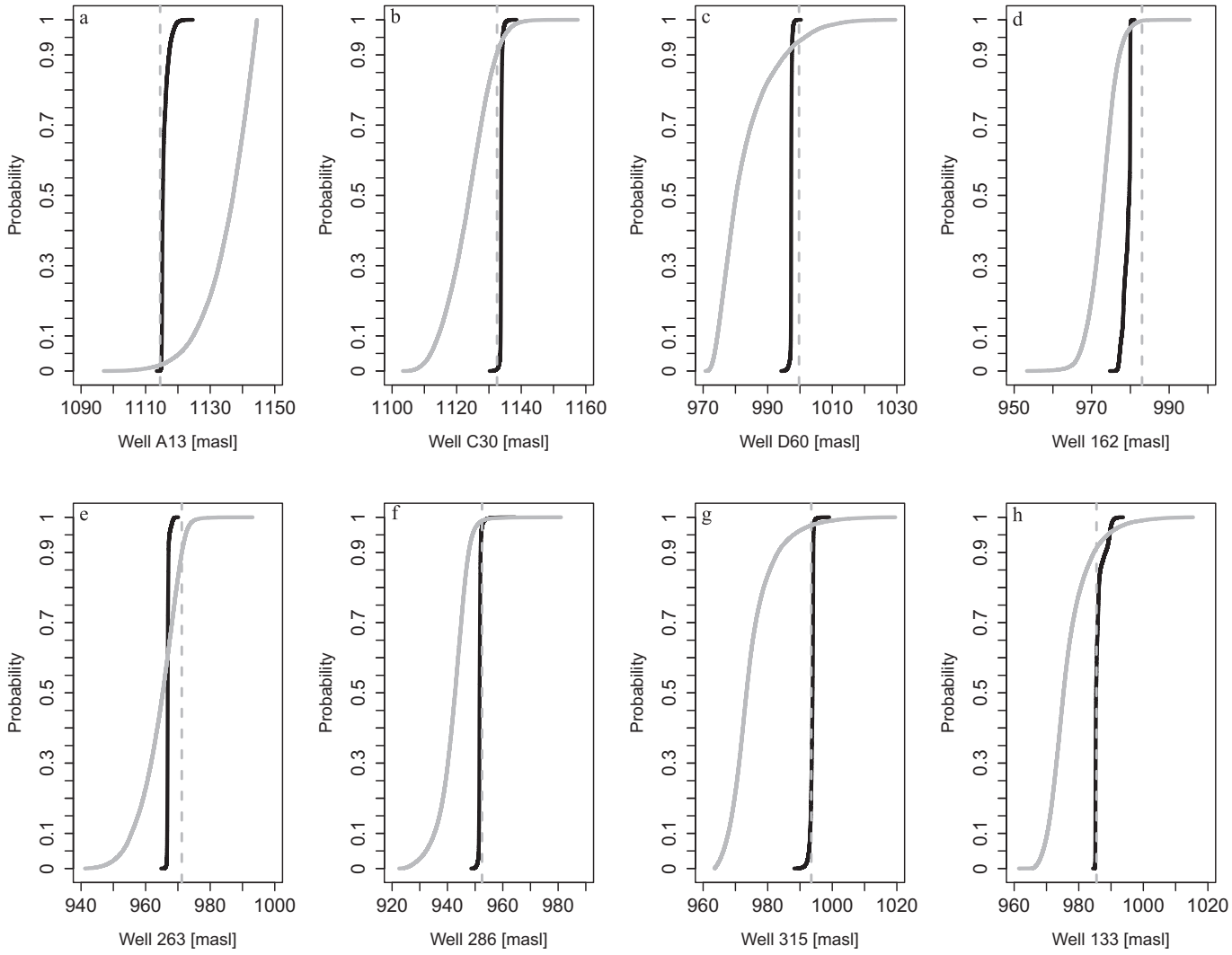


**Fig. 14.** Empirical CDFs (ECDFs) for a subset of observation wells used for the calibration of the PTA. Black lines show *hydroPSO*-based ECDFs, whereas grey lines show MCMC-based ECDFs obtained from Rojas et al. (2010). Vertical dash-lines indicate the observed values for steady-state conditions (1960). *hydroPSO*-based ECDFs are constructed from a "behavioural" sample with $L > 3.8e-02$.

(2012)), *hydroPSO* shows several innovative aspects. For example, the possibility of performing sensitivity analysis using the same package and the automation of plotting for analysis of results. More important, however, *hydroPSO* is not restricted to a limited number of hard-coded hydrological models; does not require re-programming the native model code into an R function; interfaces any (R-external) model with the PSO-based calibration engine; and uses multi-core machines or network clusters with frugal user intervention. One of the key features of *hydroPSO* compared to the aforementioned packages is its flexibility in implementing parameter optimisation for different models (model-independence). This flexibility has been demonstrated by our two case studies, which illustrate two different applications, and rests on the unique role of the hydromod function. hydromod links the external model and *hydroPSO* through the definition of ASCII files and/or R functions. Although "model-independent" tools should not require the user to write any additional code, in practice, many of these tools require some form of interfacing between the calibration engine and the model code (Matott et al., 2009). In the accompanying *hydroPSO* tutorial we provide examples of advanced interfacing between MF2005 and *hydroPSO* through the definition of I/O R functions, which we believe comes to a "low programming cost" for the modeller. We acknowledge that it is not impossible that other modellers (perhaps unfamiliar with R) may disagree on this claim, which, of course, rests on our past experience with R.

Although not shown here, *hydroPSO* does not require that the model outputs be stored in plain text files since it can take advantage of ca. 4000 contributed R packages to read several spatial and spatio-temporal file formats as well as other foreign formats. This constitutes a real advantage compared to model-independent tools restricted to read PEST-like (input/output) templates stored in plain text files.

In terms of computational implementation, *hydroPSO* constitutes the first R package implementing the latest theoretical developments of the standard PSO (SPSO-2011), including parallel capabilities, and allowing PSO-users to explore a wide range of PSO variants. In addition, it is a pioneering R package adopting a PEST-based approach for model calibration with no need of requiring the source code.

Through a comparison exercise against standard optimisation algorithms we demonstrated that *hydroPSO* is efficient and effective to solve a wide range of well-known optimisation functions. The latter highlights the functionality of *hydroPSO* to handle

different response surfaces. In addition, *hydroPSO* maintained a high success rate when increasing the dimensions of the problem without increasing the population size. This "*scalable*" behaviour is a desirable property when optimising complex models requiring long computational time.

To the best of our knowledge, this comparison exercise presents the first and most comprehensive assessment of the six algorithms mentioned in Section 4.1, involving ca. 1.8e + 09 function evaluations. For that reason, some comments need to be made. First, SCE_UA showed a decreasing SR when the true optimum was not located in the origin of the coordinate system (see functions F4, F5, F8 and F9 for $D = 10$). Second, DREAM showed a high SR only for 3 functions (F2, F3, F10 for $D = 10$ and $D = 20$), while the SR was zero for all the remaining functions in all the 3 dimensions. This behaviour may be explained by the fact that DREAM is not developed to optimise, but to accelerate the convergence of Markov Chain Monte Carlo methods (see Vrugt et al., 2009). We are aware that an "*expert*" user of any of the algorithms used in this exercise may tweak control parameters to obtain a better performance for a particular test function. Nevertheless, we did our best to provide the fairest basis for comparison across all the benchmarks.

It should be noted that the configuration proposed in this work for *hydroPSO* is "one of many" possible configurations based on empirical experience with PSO and its variants. Given that specific problems may benefit from a specific configuration of *hydroPSO*, specific recommendations on fine-tuning parameters are difficult to be made. Nevertheless, the *hydroPSO* configuration used in Sections 4 and 5 seems to address different response surfaces without compromising on effectiveness and efficiency. Based on our past experience on both benchmark functions and real-world case studies we recommend: i) using the normalise option when the feasible search space is not symmetrical, ii) try at least two different boundary.wall options, iii) try different number of particles to assess the impact on the performance of *hydroPSO*, especially for low-dimensional problems, iv) try different number of informants for the random topology, and v) use the default options in absence of previous PSO insight. Other PSO users are invited to take advantage of the full suite of fine-tuning options implemented in *hydroPSO* to assess the performance of a new PSO variant.

Calibration of the SWAT-2005 model for the upper Ega catchment showed a "typical" *hydroPSO* application using a basic interfacing between model and calibration engine. *hydroPSO* showed excellent results in terms of parameter identifiability and model performance. Few parameters related to the slow response of the catchment were not identified as sensitive by the LH-OAT procedure. This resulted in a reduced ability of the model to capture the low-flow component and in a relevant under-estimation of daily low-flows during calibration. Moreover, a slight under-estimation of daily high-flows was also observed. We ascribe these two issues mainly to the use of NSE for both sensitivity analysis and model calibration, and (possibly) to flaws in the model structure/input data. In NSE, squared differences between observed and simulated values give less emphasis to differences in low-flows (see, e.g., Legates and McCabe, 1999; Krause et al., 2005), thus, resulting in a modest contribution to NSE. For high-flows, in turn, it is recognised that NSE tends to underestimate the variability in the flows, especially, for the high-end spectrum (Gupta et al., 2009). The implementation of an alternative goodness-of-fit measure (e.g., accounting for the slow response of the catchment) could solve this issue leading to better identification of the sensitive parameters. For example, using the complementary *hydroGOF* R package would allow the modeller to assess the impact of using alternative goodness-of-fit measures for LH-OAT analysis and posterior calibration. An analysis of this type, however, was beyond the scope of this article.

Calibration of MODFLOW-2005 for the PTA showed a different implementation of the *hydroPSO* package. Advanced interfacing between the model and the calibration engine (batch application), together with a customised goodness-of-fit measure (Gaussian likelihood) were the two main features. *hydroPSO* succeeded in finding an optimum likelihood value, which outperformed the highest likelihood obtained from an extensive MCMC simulation performed for the same aquifer by Rojas et al. (2010). More importantly, this drastic improvement in the optimum likelihood came at a lower computational cost (2.1e + 05 vs. 1e + 06 model evaluations). The customised goodness-of-fit measure was implemented such that model evaluations in *hydroPSO* complied with the known behaviour of the aquifer, i.e. parameter sets leading to at least one "dry" or "zero head" cell for the observation wells were discarded. This stringent criterion added discontinuities when evaluating certain parameter sets, thus resulting in weaker convergence of the particles' velocities (displacements). Moreover, additional discontinuities were also obtained when the numerical solver of the groundwater flow equation in MODFLOW-2005 did not converge. Despite these two issues, the optimum likelihood value reported by *hydroPSO* was superior to previous results obtained for the same aquifer (Rojas and Dassargues, 2007; Rojas et al., 2010). It should be emphasised that it was not the purpose of this case study to explicitly compare both techniques (*hydroPSO* vs. MCMC), as they are designed to solve different problems. Instead, we used the MCMC results to assess the relative quality of the *hydroPSO* results. Nevertheless, optimised parameter values from *hydroPSO* could be used as starting points for intensive MCMC to obtain parameter distributions (see, e.g., Thyer et al., 2002).

## 7. Conclusions

A new multi-OS and model-independent software tool termed *hydroPSO* was presented. *hydroPSO* is developed as an R package and provides a flexible and ready-to-use global optimisation engine for model calibration. *hydroPSO* includes several state-of-the-art enhancements to the PSO algorithm, which allow the calibration engine to be customised to specific user requirements. A standalone application of *hydroPSO* allows the modeller to perform a typical modelling work flow including: sensitivity analysis, model calibration, GLUE-based uncertainty analysis, and assessment of results through advanced plotting functionalities.

For a series of benchmarking functions, *hydroPSO* showed to be more efficient and (equally or) more effective than standard optimisation algorithms (SCE_UA, DE, DREAM, SPSO-2011, and GML). A desirable property of "scalability", i.e. maintaining a high success rate in finding the optimum when increasing the dimensionality of the problem, was observed. This could bring considerable benefits when calibrating complex models demanding high computational time. For specific optimisation problems, a customised configuration of *hydroPSO* may have a significant impact on the optimisation results. In particular, by tailoring the calibration engine complex response surfaces can be effectively tackled.

Case study results highlight the importance of selecting a proper goodness-of-fit measure to drive the parameter optimisation, as well as the efficiency and effectiveness of *hydroPSO* to find optimal solutions compared to intensive Markov Chain Monte Carlo-based techniques. More important, these case studies illustrate the outstanding flexibility of *hydroPSO* to handle several issues that are commonly faced by the modelling community such as: working on different operating systems, single or batch model execution, transient- or steady-state modelling conditions, and use of pre- or user-defined goodness-of-fit measures to drive parameter optimisation. All these aspects, together with the frugal user intervention required to interface *hydroPSO* with external models and to

take advantage of multi-core/network clusters, highlight the functionality and ease of use of this package. The latter is also reflected by the recent support we have provided to *hydroPSO* users interested in calibrating models such as SWAT-2005, MODFLOW-2005, LISFLOOD, AGNPS, and HBV-96.

The *hydroPSO* version used in this article works only with single-objective functions. The implementation of multi-objective (MO) optimisation and global sensitivity analysis (GSA) are the subject of ongoing development.

Applications discussed here relate only to hydrological models. However, based on the flexibility of this software tool and the benefits added by programming it in R, we believe *hydroPSO* can be applied to a wider class of models requiring some form of parameter optimisation.

## Acknowledgements

The authors would like to thank the Dirección General de Aguas (DGA), Chile, and Confederación Hidrográfica del Ebro (CHE), Spain, for providing the data of the Pampa del Tamarugal Aquifer and the Ega River Basin, respectively. Authors are also grateful to the active R community for unselfish and prompt support, Alberto Bellin for introducing PSO to the first author, Luc Feyen for reviewing a preliminary version of the manuscript, Maurice Clerc for comments on SPSO-2011, and Till Francke for an inspiring discussion at EGU2009. The authors thank the associate editor and four anonymous reviewers whose constructive comments significantly improved the original manuscript.

## Appendix A. Supplementary data

Supplementary data related to this article can be found at http://dx.doi.org/10.1016/j.envsoft.2013.01.004.

## Appendix B. Test functions

Test functions used as benchmarks for a given $n$-dimensional problem are:

1. **F1** Sphere function

$$f(x) = \sum_{i=1}^{n} x_i^2; \quad -100 \le x_i \le 100; \quad i = 1,2,\ldots,n. \tag{A.1}$$

2. **F2** Rosenbrock function

$$f(x) = \sum_{i=1}^{n-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (1-x_i)^2\right]; \quad -30 \le x_i \le 30; \quad i = 1,2,\ldots,n. \tag{A.2}$$

3. **F3** Rastrigin function

$$f(x) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i)\right]; \quad -5.1 \le x_i \le 5.1; \quad i = 1,2,\ldots,n. \tag{A.3}$$

4. **F4** Griewank function

$$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1; \quad -600 \le x_i \le 600; \quad i = 1,2,\ldots,n. \tag{A.4}$$

5. **F5** Ackley function

$$f(x) = 20 + \exp(1) - 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right); \quad -32 \le x_i \le 32; \quad i = 1,2,\ldots,n. \tag{A.5}$$

6. **F6** Schaffer F6 function

$$f(x) = 0.5 + \frac{\sin^2\sqrt{\sum_{i=1}^{n} x_i^2} - 0.5}{\left(1 + 0.001\sum_{i=1}^{n} x_i^2\right)^2}; \quad -100 \le x_i \le 100; \quad i = 1,2,\ldots,n. \tag{A.6}$$

7. **F7** Shifted Sphere function ($f\_bias = -450$)

$$f(z) = \sum_{i=1}^{n} z_i^2 + f\_bias, \quad z = x - o; \quad i = 1,2,\ldots,n. \tag{A.7}$$

8. **F8** Shifted Griewank function ($f\_bias = -180$)

$$f(z) = \frac{1}{4000}\sum_{i=1}^{n} z_i^2 - \prod_{i=1}^{n}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f\_bias, \quad z = x - o; \quad i = 1,2,\ldots,n. \tag{A.8}$$

9. **F9** Shifted Ackley function ($f\_bias = -140$)

$$f(z) = 20 + \exp(1) - 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)\right) + f\_bias, \quad z = x - o; \quad i = 1,2,\ldots,n. \tag{A.9}$$

10. **F10** Shifted Rastrigin function ($f\_bias = -330$)

$$f(z) = 10n + \sum_{i=1}^{n}\left[z_i^2 - 10\cos(2\pi z_i)\right] + f\_bias, \quad z = x - o; \quad i = 1,2,\ldots,n. \tag{A.10}$$

## References

Andrews, F., Croke, B., Jakeman, A., 2011. An open software environment for hydrological model assessment and development. Environmental Modelling & Software 26, 1171–1185.

Argent, R., 2004. An overview of model integration for environmental applications – components, frameworks and semantics. Environmental Modelling & Software 19, 219–234.

Arumugam, M., Rao, M., 2008. On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems. Applied Soft Computing 8, 324–336.

Bendtsen, C., 2012. Particle Swarm Optimization (PSO) Package 1.0.3. Technical Report. R Foundation for Statistical Computing.

Beven, K., 2006. A manifesto for the equifinality thesis. Journal of Hydrology 320, 18–36.

Beven, K., Binley, A., 1992. The future of distributed models: model calibration and uncertainty prediction. Hydrological Processes 6, 279–283.

Beven, K., Smith, P., Wood, A., 2011. On the colour and spin of epistemic error (and what we might do about it). Hydrology and Earth System Sciences 15, 3123–3133.
Chatterjee, A., Siarry, P., 2006. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Computers & Operations Research 33, 859–871.
CHE, 2000. Los Aprovechamientos en la cuenca del Ebro: Afección en el régimen hydrológico fluvial -2000-PH-24I [The Exploitation in the Ebro Basin: Alterations to the Hydrological Regime of the River – 2000-PH-24I]. Technical Report. Confederación Hidrográfica del Ebro. Available on: http://oph.chebro.es/DOCUMENTACION/Hidrologicos/RegimenHidrologico.htm (last accessed Oct-2011).
Clerc, M., 2007. Back to Random Topology. Technical Report. Particle Swarm Central.
Clerc, M., 2009. A Method to Improve Standard PSO. Technical Report MC2009-03-13. Particle Swarm Central.
Clerc, M., 2010. From theory to practice in Particle Swarm Optimization. In: Panigrahi, B.K., Shi, Y., Lim, M.H., Hiot, L.M., Ong, Y.S. (Eds.), Handbook of Swarm Intelligence. Adaptation, Learning, and Optimization, vol. 8. Springer Berlin Heidelberg, pp. 3–36.
Clerc, M., 2012. Standard Particle Swarm Optimisation. Technical Report. Particle Swarm Central (Online; last accessed 24.09.12.).
Cooren, Y., Clerc, M., Siarry, P., 2009. Performance evaluation of TRIBES, an adaptive particle swarm optimisation algorithm. Swarm Intelligence 3, 149–178.
Doherty, J., 2010. PEST: Model-independent Parameter Estimation. User Manual, fifth ed. Watermark Numerical Computing.
Dorigo, M., Stutzle, T., 2004. Ant Colony Optimization. MIT Press, Cambridge (MA).
Duan, Q., Sorooshian, S., Gupta, H., 1992. Effective and efficient global optimization for conceptual ranfall–runoff models. Water Resources Research 28, 1015–1031.
Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: Micro Machine and Human Science, 1995. MHS '95. Proceedings of the Sixth International Symposium, pp. 39–43.
Eberhart, R., Shi, Y., 2000. Comparing inertia weights and constriction factors in particle swarm optimization. In: Evolutionary Computation, 2000. Proceedings of the 2000 Congress, pp. 84–88.
Eberhart, R., Shi, Y., 2001. Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation, pp. 81–86.
Evers, G., Ghalia, M., 2009. Regrouping particle swarm optimization: a new global optimization algorithm with improved performance consistency across benchmarks. In: IEEE International Conference on Systems, Man and Cybernetics. SMC, pp. 3901–3908.
Fox, J., 2009. Aspects of the social organization and trajectory of the R project. The R Journal 1, 5–13.
Goldberg, D., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addisson-Wesley, Reading (MA).
Guillaume, J., Andrews, F., 2012. dream: DiffeRential Evolution Adaptive Metropolis. R Package Version 0.4-2. https://r-forge.r-project.org/projects/dream/.
Gupta, H., Kling, H., Yilmaz, K., Martinez, G., 2009. Decomposition of the mean squared error and NSE performance criteria: implications for improving hydrological modelling. Journal of Hydrology 377, 80–91.
Harbaugh, A., 1990. A Computer Program for Calculating Subregional Water Budgets Using Results from the U.S. Geological Survey Modular Three-dimensional Ground-water Flow Model. Open File Report 90–392. United States Geological Survey, Reston, Virginia, USA.
Harbaugh, A., 2005. MODFLOW–2005, the U.S. Geological Survey Modular Groundwater Model–the Ground–water Flow Process. Techniques and Methods 6–A16. United States Geological Survey, Reston, Virginia, USA.
Hill, M., Tiedeman, C., 2007. Effective Groundwater Model Calibration: With Analysis of Data, Sensitivities, Predictions and Uncertainty, first ed. John Wiley & Sons, Inc., New Jersey.
Huang, T., Mohan, A., 2005. A hybrid boundary condition for robust particle swarm optimization. Antennas and Wireless Propagation Letters 4, 112–117.
Ihaka, R., Gentleman, R., 1996. R: a language for data analysis and graphics. Journal of Computational and Graphical Statistics 5, 299–314.
Ince, D., Hatton, L., Graham-Cumming, J., 2012. The case of open computer programs – editorial. Nature 482, 485–488.
Jakeman, A., Letcher, R., Norton, J., 2006. Ten iterative steps in development and evaluation of environmental models. Environmental Modelling & Software 21, 602–614.
Kannan, N., White, S., Worrall, F., Whelan, M., 2007. Sensitivity analysis and identification of the best evapotranspiration and runoff options for hydrological modelling in SWAT-2000. Journal of Hydrology 332, 456–466.
Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings IEEE International Conference on Neural Networks, pp. 1942–1948.
Kennedy, J., Eberhart, R., Shi, Y., 2001. Swarm Intelligence. In: Chapter Variations and Comparisons. The Morgan Kaufmann Series in Evolutionary Computation. Morgan Kaufmann Publishers Inc., San Francisco, CA 94104–3205, USA.
Kennedy, J., Mendes, R., 2002. Population structure and particle swarm performance. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02, pp. 1671–1676.
Kirckpatrick, S., Gelatt, C., Vecchi, M., 1983. Optimization by simulated annealing. Science 220, 671–680.
Krause, P., Boyle, D., Bäse, F., 2005. Comparison of different efficiency criteria for hydrological model assessment. Advances in Geosciences 5, 89–97.
Legates, D., McCabe Jr., G., 1999. Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. Water Resources Research 35, 233–241.
Liu, B., Wang, L., Jin, Y., Tang, F., Huang, D., 2005. Improved particle swarm optimization combined with chaos. Chaos, Solitons & Fractals 25, 1261.
Locatelli, M., 2003. A note on the Griewank test function. Journal of Global Optimization 25, 169–174.
Mantovan, P., Todini, E., 2006. Hydrological forecasting uncertainty assessment: incoherence of the GLUE methodology. Journal of Hydrology 330, 368–381.
Marquardt, D., 1963. An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics 11, 431–441.
Matott, L., 2005. Ostrich: An Optimization Software Tool, Documentation and Users Guide, Version 1.6. Department of Civil, Structural and Environmental Engineering, University at Buffalo, Buffalo, NY.
Matott, L., Babendreier, J., Purucker, S., 2009. Evaluating uncertainty in integrated environmental models: a review of concepts and tools. Water Resources Research 45, W06421.
Mendes, R., 2004. Population topologies and their influence in particle swarm performance. Ph.D. thesis. Departamento de Informática, Escola de Engenharia, Universidade do Minho, Minho, Portugal.
Mendes, R., Kennedy, J., Neves, J., 2004. The fully informed particle swarm: simpler, maybe better. IEEE Transactions on Evolutionary Computation 8, 204–210.
Montanari, A., Koutsoyiannis, D., 2012. A blueprint for process-based modeling of uncertain hydrological systems. Water Resources Research 48, W09555.
Moriasi, D., Arnold, J., Van Liew, M., Binger, R., Harmel, R., Veith, T., 2007. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. Transactions of the ASABE 50, 885–900.
Mullen, K., Ardia, D., Gil, D., Windover, D., Cline, J., 2011. Deoptim: an r package for global optimization by differential evolution. Journal of Statistical Software 40, 1–26.
Mussi, L., Cagnoni, S., Daolio, F., 2009. Empirical assessment of the effects of update synchronization in particle swarm optimization. In: Proceedings of the 2009 AIIA Workshop on Complexity, Evolution and Emergent Intelligence, ISBN 978-88-903581-1-1, pp. 1–10. Published on CD.
Nash, J., Sutcliffe, J., 1970. River flow forecasting through conceptual models. Part I–A discussion of principles. Journal of Hydrology 10, 282–290.
Neitsch, S., Arnold, J., Kiniry, J., Srinivasan, R., Williams, J., 2005. Soil and Water Assessment Tool Theoretical Documentation Version 2005. Grassland, Soil and Water Research Laboratory; Agricultural Research Service/Blackland Research & Extension Center Texas Agricultural Experiment Station, 808 East Blackland Road; Temple, Texas 76502/720 East Blackland Road Temple, Texas 76502, USA.
Poeter, E., Hill, M., Banta, E., Mehl, S., Christensen, S., 2005. UCODE–2005 and Six Other Computer Codes for Universal Sensitivity Analysis, Calibration, and Uncertainty Evaluation. Technical Methods 6–A11. United States Geological Survey, Reston, Virginia, USA.
Poli, R., 2008. Analysis of the publications on the applications of particle swarm optimisation. Journal of Artificial Evolution and Applications, 1–10.
Poli, R., Kennedy, J., Blackwell, T., 2007. Particle swarm optimization. Swarm Intelligence 1, 33–57.
R Development Core Team, 2011. R: a Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.
Ratnaweera, A., Halgamuge, S., Watson, H., 2004. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Transactions on Evolutionary Computation 8, 240–255.
Refsgaard, J., Henriksen, H., 2004. Modelling guidelines – terminology and guiding principles. Advances in Water Resources 27, 71–82.
Robinson, J., Rahmat-Samii, Y., 2004. Particle swarm optimization in electromagnetics. IEEE Transactions on Antennas and Propagation 52, 397–407.
Rojas, R., Batelaan, O., Feyen, L., Dassargues, A., 2010. Assessment of conceptual model uncertainty for the regional aquifer Pampa del Tamarugal–North Chile. Hydrology and Earth System Sciences 14, 171–192.
Rojas, R., Dassargues, A., 2007. Groundwater flow modelling of the regional aquifer of the Pampa del Tamarugal, northern Chile. Hydrogeology Journal 15, 537–551.
Rojas, R., Feyen, L., Dassargues, A., 2008. Conceptual model uncertainty in groundwater modeling: combining generalized likelihood uncertainty estimation and Bayesian model averaging. Water Resources Research 44, W12418.
Saltelli, A., Tarantola, S., Campolango, F., 2000. Sensitivity analysis as an ingredient of modeling. Statistical Science 15, 377–395.
Shi, Y., Eberhart, R., 1998a. A modified particle swarm optimizer. In: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference, pp. 69–73.
Shi, Y., Eberhart, R., 1998b. Parameter selection in particle swarm optimization. In: EP '98 Proceedings of the 7th International Conference on Evolutionary Programming VII.
Stedinger, J., Vogel, R., Lee, S., Batchelder, R., 2008. Appraisal of the generalized likelihood uncertainty estimation (GLUE) method. Water Resources Research 44, W00B06.
Storn, R., Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359.
Thyer, M., Kuczera, G., Bates, B., 1999. Probabilistic optimization for conceptual rainfall-runoff models: a comparison of the shuffled complex evolution and simulated annealing algorithms. Water Resources Research 35, 767–773.

Thyer, M., Kuczera, G., Wang, Q., 2002. Quantifying parameter uncertainty in stochastic models using the boxcox transformation. Journal of Hydrology 265, 246–257.

Uhlenbrook, S., Seibert, J., Leibundgut, C., Rodhe, A., 1999. Prediction uncertainty of conceptual rainfall–runoff models caused by problems in identifying model parameters and structure. Hydrological Sciences Journal 44, 779–797.

van Griensven, A., Meixner, T., Grunwald, S., Bishop, T., Diluzio, M., Srinivasan, R., 2006. A global sensitivity analysis tool for the parameters of multi-variable catchment models. Journal of Hydrology 324, 10–23.

Vesselinov, V., Harp, D., 2012. Model analysis and decision support (MADS) for comple physics models. In: XIX International Conference on Water Resources – CMWR 2012. University of Illinois Urbana-Champaign, Illinois, USA.

Vrugt, J., Gupta, H., Bouten, W., Sorooshian, S., 2003. A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrological model parameters. Water Resources Research 39, 1201.

Vrugt, J., Robinson, B., 2007. Improved evolutionary optimization from genetically adaptive multimethod search. Proceedings of The National Academy of Sciences of The United States of America 104, 708–711.

Vrugt, J., ter Braak, C., Clark, M., Hyman, J., Robinson, B., 2008. Treatment of input uncertainty in hydrologic modeling: doing hydrology backward with Markov Chain Monte Carlo simulation. Water Resources Research 44, W00B09.

Vrugt, J., ter Braak, C., Diks, C., Robinson, B., Hyman, J., Higdon, D., 2009. Accelerating Markov Chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. International Journal of Nonlinear Sciences and Numerical Simulation 10, 273–290.

Wagener, T., Wheather, H., Gupta, H., 2004. Rainfall–Runoff Modelling in Gauged and Ungauged Catchments. Imperial College Press, London UK.

Wolpert, D., Macready, W., 1997. No free lunch theorems for optimization. Evolutionary Computation, IEEE Transactions on 1, 67–82.

Wu, Y., Liu, S., 2012. Automating calibration, sensitivity and uncertainty analysis of complex models using the r package flexible modeling environment (fme): Swat as an example. Environmental Modelling & Software 31, 99–109.

Yapo, P., Gupta, H., Sorooshian, S., 1996. Automatic calibration of conceptual rainfall–runoff models: sensitivity to calibration data. Journal of Hydrology 181, 23–48.

Zambrano-Bigiarini, M., 2012a. hydroGOF: Goodness-of-fit Functions for Comparison of Simulated and Observed Hydrological Time Series. R package version 0.3-2-2.

Zambrano-Bigiarini, M., 2012b. hydroTSM: Time Series Management, Analysis and Interpolation for Hydrological Modelling. R package version 0.3-3.

Zhao, B., 2006. An improved particle swarm optimization algorithm for global numerical optimization. In: Alexandrov, V., van Albada, G., Sloot, P., Dongarra, J. (Eds.), Computational Science – ICCS 2006. Lecture Notes in Computer Science, vol. 3991. Springer-Verlag, Berlin Heidelberg, pp. 657–664.