EDID 287709

ENTERED

LA-UR-11-11755

Approved for public release; distribution is unlimited.

Title: Adaptive hybrid optimization strategy for calibration and parameter estimation of physical models

Author(s): Vesselinov, Velimir V.

Intended for: Computers & Geosciences

Los Alamos
NATIONAL LABORATORY
EST. 1943

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer,is operated by the Los Alamos National Security, LLC for the National NuclearSecurity Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Departmentof Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

35502

Adaptive hybrid optimization strategy for calibration and parameter estimation of physical models

Velimir V. Vesselinov*, Dylan R. Harp

Computational Earth Sciences Group, Earth and Environmental Science Division, Los Alamos National Laboratory, Los Alamos, USA.

Abstract

A new adaptive hybrid optimization strategy, entitled *squads*, is proposed for complex inverse analysis of computationally intensive physical models. Typically, models are calibrated and model parameters are estimated by minimization of the discrepancy between model simulations characterizing the system and existing observations requiring a substantial number of model evaluations. The new strategy is designed to be computationally efficient and robust in identification of the global optimum (e.g maximum or minimum value of a performance metric). It integrates a global Adaptive Particle Swarm Optimization (APSO) strategy with a local Levenberg-Marquardt (LM) optimization strategy using adaptive rules based on runtime performance. The global strategy optimizes the location of a set of solutions (particles) in the parameter space. The LM strategy is applied only to a subset of the particles at different stages of the optimization based on the adaptive rules. After the LM adjustment of the subset of particle positions, the updated particles are returned to the APSO strategy. Therefore, *squads* is a global strategy that utilizes a local optimization speedup. The advantages of coupling APSO and LM in the manner implemented in *squads* is demonstrated by comparisons of *squads* performance against Levenberg-Marquardt (LM), Particle Swarm Optimization (PSO), Adaptive Particle Swarm Optimization (APSO; the TRIBES strategy),

*vvv@lanl.gov; ph. +1 505 665 1458; fax +1 505 665 8737

Email addresses: `vvv@lanl.gov` (Velimir V. Vesselinov), `dharp@lanl.gov` (Dylan R. Harp)

and an existing hybrid optimization strategy (hPSO). All the strategies are tested on 2D, 5D and 10D Rosenbrock and Griewank polynomial test functions for a set of 1000 runs with random initial guesses for the function (model) parameters and with total number of function (model) evaluations less than 20,000. The performance of the strategies on the test functions are compared based on their robustness, defined as the percentage of runs that identify the global optimum given random initial function parameter guesses, and their efficiency, quantified by a statistical representation (boxplots) of the number of function evaluations performed prior to identification of the global optimum for successful runs. *Squads* is observed to have the best performance in terms of robustness and efficiency than the other strategies for all test functions.
Keywords: optimization, Rosenbrock, Griewank, hybrid strategy, adaptive strategy, Particle Swarm Optimization

---

1. Introduction

Models are often used in the geosciences to indirectly estimate unknown (not observable) physical properties of a system based on observable quantities representing system behavior (Carrera and Neuman, 1986; Dahlin, 2001; Jessell, 2001; Meek, 2001; Poeter and McKenna, 1995). In this process, the mathematical model is designed to simulate the system behavior $f(\boldsymbol{\theta})$ for a given set of model parameters $\boldsymbol{\theta}$ representing the actual physical properties of the system. The more accurately the model matches the observations, the more representative the model parameters are assumed to be. The process of making inferences about model parameters, commonly referred to as inverse modeling, regularly results in difficult optimization problems where a set of model parameters capable of acceptable representation of system behavior is sought. The optimization process is based on a metric representing the discrepancy between the model simulations $f(\boldsymbol{\theta})$ and the system observations. The discrepancy metric is also called the objective function (OF; $\Phi(\boldsymbol{\theta})$), and is a function of model parameters $\boldsymbol{\theta}$. In the parameter space, the metric is represented by a multi-dimensional

response hyper-surface; a three-dimensional surface for the case of two model parameters. The response surface typically has a complex shape due to multiple minima (representing multiple plausible solutions) and flat regions (representing insensitivity of model parameters to the OF) in the response surface. The optimization process is based on a series of guided model evaluations for different model parameter sets. The challenges in the optimization process come from complications in identifying the global minima and from requirements to execute a substantial number of model evaluations. Frequently, the number of model evaluations needed for optimization can vary from about 10^2 to more than 10^6 depending on the complexity of the inverse model. As a result, the optimization process can be especially difficult in real-world applications using physical models where a single forward model simulation is performed from several minutes to more than an hour. In these situations, even efficient parallel techniques (e.g. Vesselinov et al. (2001)) can cause substantial computational burden. Therefore it is important to develop computationally efficient and robust strategies that can identify the global minimum with a relatively small number of model evaluations.

Optimization strategies can be classified as global and local strategies (Nocedal and Wright, 1999). Global strategies excel at robust exploring the response surface, identifying multiple areas of attraction; however, global strategies are inefficient at locating the parameter set producing an optimal solution within a global or local area of attraction. As a result, in the case of real world model inversions, the application of global strategies may be infeasible when the model evaluations take a substantial amount of computational time (Keating et al., 2010). Local strategies excel at efficiently identifying the optimal model parameters within an area of attraction; however, local strategies are not designed for robust exploration of a response surface outside of an area of attraction. The local strategies are efficient within an area of attraction because they utilize local information about the gradient and curvature of the response surface. This requires estimation of the first and second order derivatives of the discrepancy metric in the parameter space. As many real world model

problems are characterized with multiple minima, the use of local strategies alone is not always robust. One of the most commonly used local strategies is the Levenberg-Marquardt (LM) strategy which has been applied in many frequently used inverse analysis and parameter estimation codes in the geosciences such as UCODE (Poeter and Hill, 1999) and PEST (Doherty, 2005).

Global and local strategies are complimentary; where one excels, the other struggles, and vice versa. The benefits of hybrid global/local strategies have been demonstrated previously (Noel and Jannett, 2004; Leontitsis, 2004; Zhang et al., 2007; Ghaffari-Miab et al., 2007). We introduce a new development in hybrid optimization coupling recent developments in Adaptive Particle Swarm Optimization (APSO) and a Levenberg-Marquardt (LM) strategy producing a novel adaptive hybrid strategy entitled *squads*. The strategy applies an LM strategy to a subset of particles at different stages of an APSO strategy based on adaptive rules. After the LM update of the particle position, the particle is passed back to the APSO strategy and continues to evolve based on APSO rules. In essence, *squads* is a global strategy utilizing local optimization speedup. *Squads* is specifically designed to be a robust and computationally efficient strategy capable of identify the global minimum with a relatively small number of model evaluations in complex inverse problems. The name *squads* refers to the hierarchical structure of the population of solutions in the algorithm, similar to the APSO algorithm TRIBES (Clerc, Jul. 2004).

The *squads* strategy is tested using the Rosenbrock (Rosenbrock, 1960) and Griewank (Griewank, 1981) polynomial test functions. In order to demonstrate the relative benefits of the hybrid strategy of *squads*, its performance is compared to open-source distributions of the LM (Lourakis, Jul. 2004), PSO (Paricle Swarm Central, 2006), and APSO (Clerc, Jul. 2004) strategies. Additionally, *squads* is compared to hPSO (Leontitsis, 2004), an open-source hybrid strategy that combines PSO and the Nelder-Mead downhill simplex strategy (Nelder and Mead, 1965) implemented in the MATLAB® (The MathWorks Inc., 2003) computing environment.

2. Particle swarm optimization

Sociobiologists have theorized that individuals within a population can benefit from the previous knowledge and experience of other members of the population while searching for sporadically distributed food sources (Wilson, 1975). The ubiquity of schooling and flocking tendencies common among many species suggests that this is an efficient, cost-effective strategy for the survival of individuals. It is easy to recognize the analogy of organisms searching for food sources and mathematical strategies searching for optimal solutions. This recognition led to the development of PSO by Kennedy and Eberhart (1995), building on previous research intended to graphically simulate the flocking behavior of birds. Certain aspects of the flocking behavior of this early research has been eliminated in order to improve the strategy's performance in global optimization, leading to the use of the term "swarm" to describe the graphical behavior of PSO.

The development of PSO has produced a parsimonious optimization strategy modeling a population of randomly selected initial solutions (particles) by their position and velocity (Clerc, 2006). In a D-dimensional parameter space, the position and velocity of the ith particle can be represented as $\mathbf{p}_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,D}]$ and $\mathbf{v}_i = [v_{i,1}, v_{i,2}, \ldots, v_{i,D}]$, respectively. An empirical formula for determining the swarm size S has been suggested as $S = 10 + 2\sqrt{D}$ (Paricle Swarm Central, 2006). Particles retain a record of the best location they have visited so far denoted as $\mathbf{b}_i = [b_{i,1}, b_{i,2}, \ldots, b_{i,D}]$. Particles are also informed of the best location that K other randomly chosen particles have visited, denoted as $\mathbf{g}_i = [g_{i,1}, g_{i,2}, \ldots, g_{i,D}]$. A standard value for K is 3 (Paricle Swarm Central, 2006). These networks of informers are reinitialized after iterations with no improvement to the global best particles of the swarm. The velocity of the ith particle in the jth dimension is updated from strategy iteration k to $k+1$ as

$$v_{i,j}(k+1) = wv_{i,j}(k) + c_1 r_1 (b_{i,j} - p_{i,j}(k)) + c_2 r_2 (g_{i,j} - p_{i,j}(k)), \quad k = \{1, \ldots, D\}, \tag{1}$$

where w is a constant referred to as the inertia weight, c_1 and c_2 are constants referred to as acceleration coefficients, r_1 and r_2 are independent uniform random numbers in $[0, 1]$. The parameter w controls the level of influence between its previous and current particle displacement, c_1 and c_2 scale the random influence of (1) the particle memory (past particle locations in the parameter space), and (2) the current network of particle informers (current informer locations in the parameter space), respectively. A limitation on the magnitude of the velocity V_{max} is commonly employed. The particle position is updated during each strategy iteration as

$$p_{i,j}(k+1) = p_{i,j}(k) + v_{i,j}(k+1), \quad k = \{1, \ldots, D\}. \tag{2}$$

It has been recognized that the selection of w, c_1, c_2, and V_{max} tune the performance of PSO, modifying the balance between exploration (spreading the particles throughout the parameter space) and intensification (focusing the particles within an area of attraction). Manual tuning of PSO's parameters can be a delicate task. Adaptive PSO (APSO) strategies have emerged in order to reduce or eliminate the often difficult and time-consuming process of parameter tuning of PSO (Cooren et al., 2009).

One of the algorithmic variants of APSO is TRIBES (Clerc, 2006) (TRIBES is not an acronym, but we follow the convention of all capital letters as proposed by its designer), which eliminates the tuning of the PSO strategy parameters. The strategy has been proven competitive with well-known strategies on a suite of test problems (Cooren et al., 2009). As the name suggests, TRIBES partitions the particles into groups, referred to as *tribes*, intended to facilitate the exploration of multiple areas of attraction. In this way, a hierarchical structure is established where the swarm is composed of a network of tribes, and each tribe is a network of particles. The intent is to eliminate parameter tuning

as the swarm evolves from an initial set of tribes, and the tribes evolve from single particles based on rules governing the evolution of the swarm topology and rules for generation and elimination of entire tribes and individual particles within the tribes. The particle within a tribe with the lowest/highest OF for minimization/maximization is considered the shaman of the tribe. Information is shared only between the particles within a given tribe. Information between the tribes is shared only through the shamans. In this way, the displacements of non-shaman particles are influenced by the shaman of their tribe, while the displacements of the shamans are influenced by the *best* shaman in the swarm. The source code for TRIBES is available from Clerc (Jul. 2004).

3. *Squads* adaptive hybrid optimization

Various approaches have been introduced to couple the global search capabilities of PSO with the efficiency of first and second-order local strategies. Clerc (1999) introduced a PSO strategy that adjusts particle locations based on approximations of the gradient of the OF utilizing the OF values of the current particle locations. Noel and Jannett (2004) developed a hybrid PSO strategy incorporating gradient information directly in the calculation of particle velocity. Leontitsis (2004) coupled a PSO strategy with the Nelder-Mead simplex strategy (hPSO, Lagarias et al. (1998)), Zhang et al. (2007) coupled PSO and back-propagation to train neural networks. Ghaffari-Miab et al. (2007) developed a hybrid strategy, iterating between PSO and BFGS quasi-Newton optimization. We present a hybrid strategy called *squads* that couples an APSO strategy with a Levenberg-Marquardt (LM) strategy. The following provides a detailed description of a fine-tuned coupling of APSO and LM based on adaptive rules, where the LM optimization is applied to improve the locations of a subset of selected particles (the shamans) in the course of the optimization process. The current APSO strategy implemented in *squads* is TRIBES (Clerc, 2006), and the LM optimization is performed using the LevMar library (Lourakis, Jul. 2004).

Much of the time-consuming and difficult tuning required of many optimization strategies is reduced in *squads* utilizing adaptive rules. The APSO strategy does not require the specification of optimization parameters (Clerc, 2006), and the applied LM strategy is optimized to work well on many problems using default and internally estimated optimization parameters (Lourakis, Jul. 2004). The adaptive rules implemented in *squads* to control the performance of LM speedups during the APSO optimization are also designed to be general and capable to tackle problems with different complexity.

A flow diagram of the *squads* strategy is presented in Figure 6. Tables 1 and 2 describe the particle initialization and displacement rules and their selection within *squads*. For consistency with other global strategies discussed here, *squads* is initialized with $N_t = S = 10 + 2\sqrt{D}$ mono-particle tribes, where N_t is the number of tribes in the swarm and S is the number of particles. However, *squads* can also be initiated with a single mono-particle tribe and allow the swarm to develop based on the built-in adaptive rules. If provided, one of the initial particles is set to predefined values (rule 1 in Table 1), while the remaining positions of the initial particles are determined according to rule 5 in Table 1.

Each iteration of the strategy is initiated by determining the informers for all the particles. For non-shaman particles, this will be the shaman of their tribe. A shaman is the particle with the best (e.g. lowest for minimization) OF value within the tribe. For shaman's, this will be the shaman with the best OF value within the swarm, referred to as the *best shaman*. Particle positions are then updated according to the rules described in Table 2. Particles are initialized to use displacement rule 1. After informers are determined, particle positions are updated based on their currently selected displacement rule.

The decision to adapt a tribe is based on whether the tribe has demonstrated sufficient improvement during the previous strategy iteration. This is performed stochastically, by comparing the fraction of particles in the tribe that improved their location in the last move with a random number between 0 and 1. If the fraction is greater than the random number, the tribe is considered a *good tribe*,

and the worst particle is removed from the tribe. This eliminates unnecessary model evaluations, focusing the attention of the tribe on the *good particles*. Otherwise, the tribe is considered a *bad tribe*, and a particle is added to the tribe (refer to Table 1 for details on particle initialization rule selection) and a randomly selected dimension of a randomly selected particle in the tribe (other than the shaman) is reinitialized randomly. Adding a particle to a *bad tribe* is intended to increase the exploration of the parameter space by the tribe.

The swarm adaptation occurs either every $N_t * (N_t - 1)/4$ strategy iterations or if the swarm is labeled by LM as a *bad swarm*. A swarm is considered a *bad swarm* if LM speedup was performed in the previous iteration, and the OF was not reduced by at least 2/3 for all the LM updated shamans. A mono-particle tribe is added to the swarm if it is considered *bad* according to rule 5 in Table 1. The tribe led by a shaman with the worst OF in the swarm is removed if the swarm is considered *good*.

Next, particle displacement rule selections are updated. Particle displacement rule selections are modified based on whether or not (1) their position has improved in the last move and (2) their best overall position has improved in the last move. Following the convention of Clerc (2006), we use a (+) to indicate improvement, (=) the same OF value, and (–) a worse position. The particles performance can then be denoted as one of the following: (–=), (==), (+=), and (++), where the first symbol indicates if the particle improved its position in the last move, and second symbol indicates if the overall best position of the particle improved in the last move. Note that the best overall performance can only stay the same or improve, and an improvement in the overall performance indicates an improvement over the last position. Table 2 lists the displacement rule selection based on particle performance.

After the swarm adaptation, *squads* checks whether or not to update the shamans using LM. LM updating is turned off in *squads* if none of the shamans reduces the OF of the previous shamans by more than 2/3 during the last LM updating. LM updating will be restarted when the best OF of the previously obtained OF during LM has been reduced by an order of magnitude by the APSO

strategy. This postpones LM until the global APSO strategy has identified a position with a significant improvement, which will perhaps be a previously unidentified area of attraction. After the LM optimization, the new shaman location is used in the APSO strategy.

In contrast with the APSO strategy, the LM strategy requires that the OF be represented as a summation of components at least equal to the number of parameters as

$$\Phi(\boldsymbol{\theta}) = \sum_{i=1}^{N} \Phi_i(\boldsymbol{\theta}), \tag{3}$$

where $\boldsymbol{\theta}$ is a vector of model parameters and N is equal or larger than the number of model parameters. This allows the LM strategy to estimate the local gradient and curvature of the response surface in the parameter space. These calculations utilize numerical derivatives of the OF components in equation 3 with respect to the model parameters (also called the Jacobian matrix). Based on the Jacobian matrix, the LM strategy also estimates the second-order derivatives of the OF components with respect to model parameters (also called a Hessian matrix). The second-order derivatives approximate the local curvature of the response surface. The LM strategy searches for the local optimum by adaptive adjustment between first and second-order optimization techniques (Levenberg, 1944; Marquardt, 1963). Frequently in the case of model inversion problems, the OF in equation 3 is represented by the discrepancy between model simulated values $f_i(\boldsymbol{\theta})$ and corresponding observations o_i, where $i = 1, ..., N$, and N is now the number of observations. For example, frequently $\Phi(\boldsymbol{\theta})$ is computed as

$$\Phi(\boldsymbol{\theta}) = \sum_{i=1}^{N} \Phi_i(\boldsymbol{\theta}) = \sum_{i=1}^{N} (f_i(\boldsymbol{\theta}) - o_i)^2. \tag{4}$$

Squads estimates the first-order derivatives using a finite difference approach applied in the LevMar library (Lourakis, Jul. 2004).

The following criteria are defined by default in LevMar to terminate the LM optimization (Lourakis, Jul. 2004), and applied in the LM updating of *squads* as well: (1) the maximum change in any parameter is less than 10^{-5}; (2) the

relative change in the L2 norm of the change in the parameter values is less than 10^{-5} of the L2 norm of the parameter values; (3) the OF reaches a value of zero; (4) the Jacobian matrix is close to singular, and (5) the maximum number of LM iterations (i.e. derivative approximation and Marquardt parameter value exploration) is achieved (50 when standalone LM is performed; 8 in *squads*). The criteria are designed to terminate LM once it successfully identifies a local optimum. Typically, criteria 1, 2, and 5 terminate the LM updating in *squads* (the termination criteria of the LM updating within *squads* do not terminate the *squads* run). *Squads* is terminated when either one of the following conditions are met: (1) E_{max}, the number of allowable model evaluations, is exceeded or (2) the OF reaches below a predefined cutoff value.

The final step of each iteration is to perform a random local search in the empty space around each shaman (Clerc, Jul. 2004). In this step, a random position within the largest hyperparallelepid centered on the tribe's shaman, void of other particles, is evaluated. If the position is an improvement over the current shaman position, the shaman is moved to this location. Otherwise, the position is forgotten.

Global strategies in general, including APSO, are designed to operate on a bounded parameter space. The parameter ranges are typically predefined depending on the physical constraints or prior knowledge about the parameter distributions. However, the LM optimization by default works in an unbounded parameter space. There are various techniques to constrain an LM strategy within a parameter space, but these techniques typically have a negative impact on LM performance. To avoid this, *squads* operates in a transformed parameter space. For example, an element of the parameter vector $\boldsymbol{\theta}$ is transformed as

$$\hat{\theta} = \arcsin\left(\frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}} \cdot 2 - 1\right), \tag{5}$$

where $\hat{\theta}$ is the transformed parameter, and θ_{max} and θ_{min} are the upper and lower bounds for parameter θ, respectively. The APSO strategy is performed in the transformed parameter space bounded within $[-\pi/2; \pi/2]$ in all dimen-

sions, while the LM updating is performed unconstrained in the transformed parameter space. Model (function) evaluations are performed on de-transformed parameters by

$$\theta = \theta_{min} + \left(\frac{\sin(\hat{\theta}) + 1}{2} \right) (\theta_{max} - \theta_{min}). \tag{6}$$

In this way, the LM updating is unaware of parameter boundaries and is unaffected by performance issues associated with calculating numerical derivatives near boundaries. It should be noted that in the process of the LM updating, the transformed parameters can be moved outside of the $[-\pi/2; \pi/2]$ range; however, the transformed parameters are returned to equivalent values within $[-\pi/2; \pi/2]$ before being passed back to the APSO strategy by

$$\hat{\theta}_{APSO} = \arcsin(\sin(\hat{\theta}_{LM})). \tag{7}$$

where $\hat{\theta}_{LM}$ represents the unconstrained transformed parameters resulting from LM updating and $\hat{\theta}_{APSO}$ represents the constrained transformed parameters passed back to the APSO strategy, thereby ensuring that APSO receives parameters within its explicitly defined, bounded parameter space. It is important to note that $\hat{\theta}_{APSO}$ and $\hat{\theta}_{LM}$ are equivalent in the non-transformed parameter space.

4. Test functions

The *squads* strategy is tested by optimizing the Rosenbrock and Griewank test functions. The Rosenbrock and Griewank functions present difficult optimization problems exhibiting frequently observed complexities in response surface topology in real world problems (e.g. Rosenbrock (1960); Griewank (1981); Clerc (2006); Cooren et al. (2009)).

The response function defined by the Rosenbrock function is comprised of a large valley with an ill-defined, shallow global minimum. For $D \leq 3$, the function is unimodal with a global minimum at $\mathbf{x} = \mathbf{1}$ (where $\mathbf{1} = [1, \ldots, 1]$).

For $4 \leq D \leq 7$, a local minimum exists at $(x_1, x_2, \ldots, x_D) = (-1, 1, \ldots, 1)$ in addition to the global minimum, while for $D > 7$, multiple suboptimal local minima exist (Shang and Qiu, 2006). In the case of two model parameters, the shape of the Rosenbrock function is presented in Figure 2. The Rosenbrock function generalized to any number of dimensions greater than or equal to two can be expressed as

$$\Phi_r(x_1, \ldots, x_D) = \sum_{i=1}^{D-1} (1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2. \tag{8}$$

The estimation of the local gradient and curvature of the response surface by LM requires the test function to be represented as a summation of parts as in equation 3. The summation components of $\Phi_r(x_1, \ldots, x_D)$ can be expressed as

$$\Phi_{r,2i-1}(x_i) = (1 - x_i)^2 \;\; i < D \tag{9}$$

and

$$\Phi_{r,2i}(x_i, x_{i+1}) = 100(x_{i+1} - x_i^2)^2 \;\; i < D \tag{10}$$

producing $2(D-1)$ OF components where equation 9 and 10 define the odd and even numbered components, respectively. The LM strategy uses the derivatives of $\Phi_{r,i}(x_1, \ldots, x_D)$ with respect to model parameters to evaluate the local gradient and curvature of the response surface. In most real world problems, the analytical computation of derivatives is not feasible. Therefore, in all the examples presented below, the derivatives are computed numerically using a finite difference approach, even though the analytical derivation in this case is trivial. Other alternative representations of Φ_r as a sum of components are also possible.

The D-dimensional Griewank function is defined as

$$\Phi_g(x_1, \ldots, x_D) = 1 + \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right). \tag{11}$$

The Griewank function has numerous local areas of attraction, but a single global minimum of zero at $\mathbf{x} = \mathbf{0}$. In the two-dimensional case, the function has the shape of an "egg carton" that is depressed in the center, as depicted in Figure 2).

The summation components can be defined as

$$\Phi_{g,i}(x_1, \ldots, x_D) = \frac{1}{D} + \frac{x_i^2}{4000} - \frac{1}{D} \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) \tag{12}$$

The multidimensional Griewank function is important for testing of hybrid optimization strategies because it becomes more difficult to minimize for global strategies as its dimensionality increases (Locatelli, 2003). However, although counterintuitive, the Griewank function becomes easier to minimize for local strategies as the dimensionality increases. Therefore, with the increase in dimensionality, it is expected that LM performance will improve while the PSO, TRIBES and hPSO performance will decrease. For different parameter-space dimensionality, the performance of hybrid strategies will depend on how efficiently they adaptively balance between the local and global strategies. At low dimensionality ($D = 2$), the hybrid strategies should benefit from the global strategy; at high dimensionality, the hybrid strategies should benefit from the local strategy.

5. Results and discussion

The performance of *squads* on the Rosenbrock and Griewank functions is compared with (1) LM, (2) PSO, (3) TRIBES, and (4) hPSO. The LM strategy is an implementation of LevMar (Lourakis, Jul. 2004) (the same strategy as applied in *squads*), PSO is an implementation of Standard PSO 2006 (Paricle Swarm Central, 2006), TRIBES is an implementation of the code described in Clerc (2006), hPSO is freely available hybrid optimization code from Leontitsis (2004). LM, PSO, TRIBES and *squads* are built into the code MADS (Vesselinov, 2010), which is utilized for all analyses except hPSO. The hPSO analysis is performed using MATLAB version 7.8.0.347 (R2009a) (The MathWorks Inc.,

2003). The strategy parameters (i.e. optimization parameters) for PSO and hPSO are set to values that have been demonstrated to perform well in many test cases (Paricle Swarm Central, 2006) as $w = 0.72$, $c_1 = 1.2$ and $c_2 = 1.2$ (refer to equation 1).

The strategies are tested on both functions by performing 1,000 independent optimizations runs with random initial guesses distributed in the searchable parameter space bounded by [-100:100] for all dimensions. In the case of LM, the searchable parameter space is not bounded. This did not influence its performance as the OFs of both functions have generally increasing trends towards the boundaries (Figure 2)). Optimization success is defined as identifying a solution with all parameters values within 0.1 of the global minimum parameter values ($\mathbf{x} = \mathbf{1}$ for the Rosenbrock function and $\mathbf{x} = \mathbf{0}$ for the Griewank function). The maximum number of function (model) evaluations (E_{max}) for the strategies is set to 20,000. However, in performed analyses, LM runs terminate at fewer function evaluations as the convergence criteria of LM are designed to terminate its run once it identifies a minimum in the response surface. The robustness of LM depends on whether the identified minimum is local or global.

Figures 3 and 4 present boxplots for the number of function evaluations for successful runs for 2D, 5D, and 10D Rosenbrock and Griewank functions, respectively. In the figures, the box represent the 25th to 75th percentile ranges, the bars inside of the boxes represent the median values, and the whiskers represent the min and max values. The number of successful runs out of the 1000 attempted runs are presented above the boxes. Note that the statistical definitions are not accurate for the cases where the number of successful runs does not present a statistically significant sample. The robustness of the strategies is defined as the percentage of successful runs (i.e. number of successful runs / 1000 * 100). The efficiency of the strategies is summarized by the statistics presented in the boxplots.

For the Rosenbrock function (Figure 3), the robustness of LM decreases from the 2D case to the 10D case from 36% to 0%. The robustness of PSO and TRIBE is comparable in the 2D case, albeit with TRIBES exhibiting higher efficiency in

general. In the 5D case, PSO has a higher robustness than TRIBES, however, at lower efficiency. hPSO achieves 100% robustness in the 2D and 5D cases, with a significant decrease in efficiency from the 2D to 5D case. The robustness of hPSO decreases significantly in the 10D case with only a single success out of 1000 (0.1%). In the 10D case, LM, PSO, TRIBES, and hPSO exhibit low robustness. *Squads* is 100% robust in all cases. The efficiency is observed to decrease from the 2D case to the 10D case for *squads*; however, the efficiency of *squads* is greater than PSO, TRIBES, and hPSO in all cases. The efficiency of *squads* and LM are on the same order of magnitude for the 2D and 5D cases (Figure 3). However, in these two cases, the robustness of *squads* is 100% which is considerably better than the robustness of LM (36% for 2D and 4% for 5D). In the 10D case, LM did not produce a single successful run while *squads* is still 100% robust.

For the Griewank function (Figure 4), as expected (see Locatelli (2003)), the robustness of LM increases as the dimensionality of the problem increases. In the 2D case, which is the most difficult for a local gradient-based strategy (Locatelli, 2003), the robustness is only 3%. Since LM is a local strategy, it is not surprising that LM frequently converges at non-global minima. As expected for the 2D case, the global strategies (PSO, TRIBES, hPSO and *squads*) are substantially more robust than LM. The robustness of PSO and TRIBES (both purely global strategies) decrease significantly from the 2D to the 5D case, while decreasing only slightly from the 5D to the 10D case (the efficiency of PSO decreases also). hPSO is 100% robust for the 2D case; however, is unable to locate the global minimum in the 5D and 10D cases. *Squads* is 100% robust in the 2D and 10D cases and 80% robust in the 5D case.

As already discussed, the multidimensional Griewank function is important for testing of hybrid strategies such as *squads*. For different parameter-space dimensionality, the performance of *squads* is influenced by the ability of the adaptive rules in the optimization algorithm to balance between the local (LM) and global (APSO) strategies. With the increase of dimensionality, the local (LM) strategy becomes more robust, while the global (APSO) strategy becomes less

robust. At $D = 2$, *squads* is both more robust and efficient than the other global methods (Figure 4). At $D = 10$, *squads* benefits from the local gradient-based search strategy which performs better at higher dimensions. The 5D Griewank function is observed to be the most difficult test problem for *squads* as both the local and global strategies struggle in this dimensionality of the Griewank function. Nevertheless, for the 5D case, *squads* produces the highest robustness (80%) and efficiency (excluding LM) of all the tested strategies; *squads* is 100% robust if the maximum number of function evaluations is increased to 70000 (results are not shown here). In summary for the Griewank cases, *squads* is observed to have the best performance in terms of robustness and efficiency than the other strategies (Figure 4).

It is important to emphasize in all test cases, *squads* can converge at relatively low number of model evaluations when compared to PSO, TRIBES and hPSO. This is manifested by the minimum values of the boxplots in Figures 3 and 4. Furthermore, the statistical distributions of the number of model evaluations required to achieve the global minimum for *squads* are skewed to the left in all cases (Figures 3 and 4). This demonstrates that more frequently *squads* may converge with lower number of functional evaluations.

6. Conclusions

A new adaptive global hybrid optimization strategy called *squads* is developed for solving computationally intensive inverse problems involving models representing the behavior of complex systems; such models are commonly applied in the geosciences. *Squads* utilizes a (1) global strategy for robust exploration of the parameter space to identify multiple areas of attraction and (2) a local gradient-based search strategy to efficiently locate the optimum of areas of attraction. In essence, *squads* is a global strategy that implements a local optimization speedup. *Squads* is sufficiently robust in avoiding becoming stuck in local minima during the optimization. The new strategy reduces the number of model runs typically required of other frequently used global strategies such

as Particle Swarm Optimization (PSO) by efficiently exploring local areas of attraction.

The strategy is tested on 2D, 5D, and 10D variations of two commonly used polynomial test functions: the Rosenbrock and Griewank functions. The robustness of a strategy is defined as the percentage of runs that identify the global minimum of a test function. The efficiency of a strategy is evaluated through a statistical representation of the number of function evaluations necessary to identify the global optimum. In all cases, *squads* is as robust or more robust than the other tested strategies: LM, PSO, TRIBES, and hPSO. *Squads* is more efficient in general than PSO, TRIBES, and hPSO in all cases. For the Rosenbrock function, *squads* has comparable efficiency to LM, however, in these cases, the robustness of *squads* (100%) is considerably better than the robustness of LM (less than 36%). For other optimization problems, *squads* may converge for the same number of functional evaluations as LM (Figure 4)

The application of the *squads* strategy is performed using the code MADS (Vesselinov, 2010). MADS and other files needed to execute the synthetic problems presented in this paper are available at http://www.ees.lanl.gov/staff/monty/codes/mads.html

References

Carrera, J., Neuman, S. P., 1986. Estimation of aquifer parameters under transient and steady state conditions: 1. Maximum likelihood method incorporating prior information. Water Resources Research 22 (2), 199–210.

Clerc, M., 1999. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. IEEE, 1951–1957.

Clerc, M., 2006. Particle Swarm Optimization. ISTE, London.

Clerc, M., Jul. 2004. Tribes, an adaptive particle swarm optimization algorithm in C. [web page] http://clerc.maurice.free.fr/pso/Tribes/TRIBES-D.zip, [Accessed on 4 Nov. 2010.].

Cooren, Y., Clerc, M., Siarry, P., 2009. Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm. Swarm Intelligence 3, 149–178.

Dahlin, T., 2001. The development of DC resistivity imaging techniques. Computers & Geosciences 27, 1019–1029.

Doherty, J., 2005. PEST model-independent parameter estimation: User manual.

Ghaffari-Miab, M., Farmahini-Farahani, A., Faraji-Dana, R., Lucas, C., 2007. An efficient hybrid swarm intelligence-gradient optimization method for complex time Grees's functions of multilayer media. Progress in Electromagnetics Research 77, 181–192.

Griewank, A. O., 1981. Generalized descent for global optimization. Journal of Optimization Theory and Applications 34, 11–39.

Jessell, M., 2001. Three-dimensional geological modelling of potential-field data. Computers & Geosciences 27, 455–465.

Keating, E. H., Doherty, J., Vrugt, J. A., Kang, Q., 2010. Optimization and uncertainty assessment of strongly nonlinear groundwater models with high parameter dimensionality. Water Resources Research 46.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks. IEEE Press, Piscataway, pp. 1942–2948.

Lagarias, J., Reeds, J., Wright, M., Wright, P., 1998. Convergence properties of the Nelder-Mead simplex method in low dimensions. SIAM Journal of Optimization 9 (1), 112–147.

Leontitsis, A., 2004. Hybrid Particle Swarm Optimization. http://www.mathworks.com/matlabcentral/fileexchange/6497-hybrid-particle-swarm-optimization.

Levenberg, K., 1944. A method for the solution of certain nonlinear problems in least squares. Q. Appl. Math. 2, 164–168.

Locatelli, M., 2003. A note on the griewank test function. Journal of Global Optimization 25, 169–174.

Lourakis, M., Jul. 2004. levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++. [web page] http://www.ics.forth.gr/ lourakis/levmar/, [Accessed on 4 Nov. 2010.].

Marquardt, D., 1963. An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. 11, 431–441.

Meek, D., 2001. A semiparametric method for estimating the scale of fluctuation. Computers & Geosciences 27, 1243–1249.

Nelder, J. A., Mead, R., 1965. A simplex method for function minimization. Computer Journal 7, 308–313.

Nocedal, J., Wright, S. J., 1999. Numberical Optimization. Springer.

Noel, M. M., Jannett, T. C., 2004. Simulation of a new hybrid particle swarm optimization algorithm. IEEE, 150–153.

Paricle Swarm Central, 2006. http://www.particleswarm.info/Standard_PSO_2006.c.

Poeter, E. P., Hill, M. C., 1999. UCODE, a computer code for universal inverse modeling. Computers & Geosciences 25, 457–462.

Poeter, E. P., McKenna, S. A., November 1995. Reducing uncertainty associated with ground-water flow and transport predictions. Ground Water 33 (6), 899–904.

Rosenbrock, H. H., 1960. An automated method for finding the greatest or least value of a function. Computer Journal 3 (3), 175–184.

Shang, Y., Qiu, Y., 2006. A note on the extended Rosenbrock function. Evolutionary Computing 14 (1), 119–126.

The MathWorks Inc., 2003.

Vesselinov, V. V., 2010. MADS, Modeling Analysis and Decision Support toolkit in C. [web page] http://www.ees.lanl.gov/staff/monty/codes/mads.html, [Accessed on 15 Nov. 2010.].

Vesselinov, V. V., Neuman, S. P., Illman, W. A., December 2001. Three-dimensional numerical inversion of pneumatic cross-hole tests in unsaturated fractured tuff 2. Equivalent parameters, high-resolution stochastic imaging and scale effects. Water Resource Research 37 (12), 3019–3041.

Wilson, E., 1975. Sociobiology: The new synthesis. Belknap Press, Cambridge, MA.

Zhang, J.-R., Zhang, J., Lok, T.-M., Lyu, M. R., 2007. A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training. Applied Mathematics and Computation 185, 1026–1037.

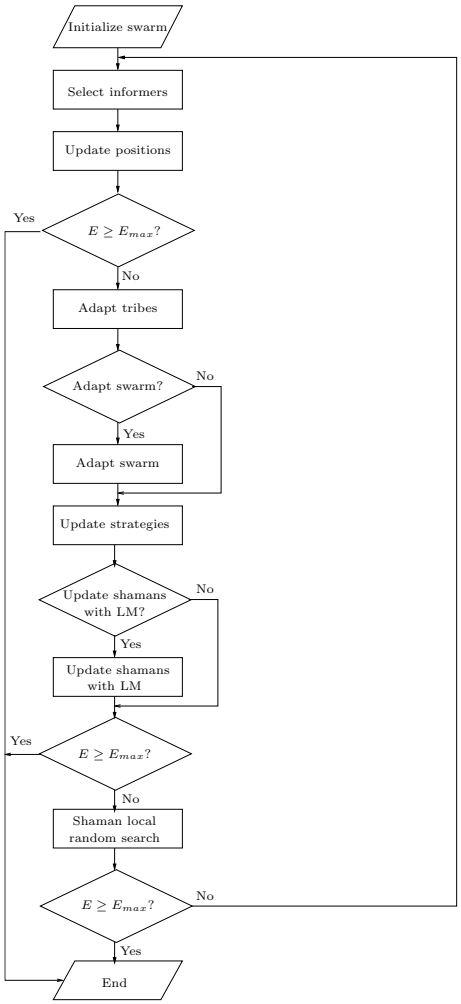

Figure 1: Flow diagram of *squads* strategy. E is the current number of model evaluations and E_{max} is the allowable number of model evaluations. Decisions to "adapt swarm" or "update shamans with LM" are determined by adaptive rules.

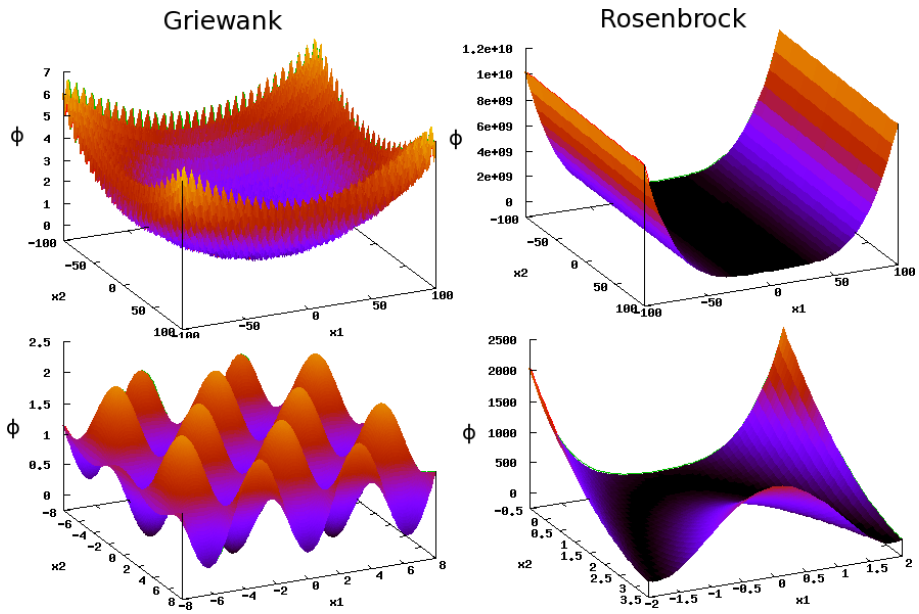

Figure 2: Rosenbrock and Griewank polynomial test functions with global minima at (1,1) and (0,0), respectively. Note the different parameter ranges on the top and bottom rows. The top row shows the parameter space explored by the optimization strategies. The bottom row focuses on the parameter space near the respective global minima.

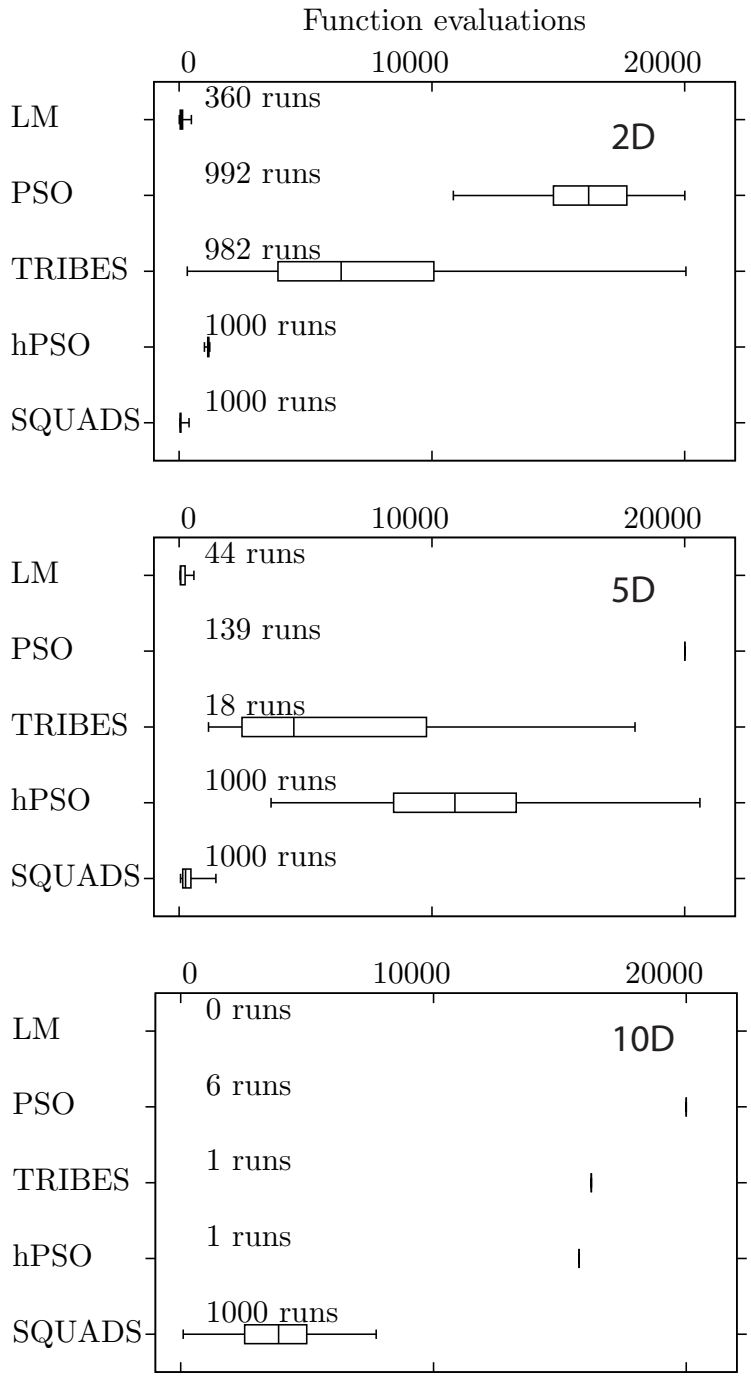

Figure 3: Boxplots of number of function evaluations to reach the global minimum for the 2D, 5D, and 10D Rosenbrock function. The boxes represent the 25th to 75th percentile ranges, the bars inside of the boxes represent the median values, and the whiskers represent the min and max values. Note that the statistical definitions are not accurate for TRIBES and hPSO in the 10D case where the number of successful runs does not present a statistically significant sample. The number of successful runs out of 1000 for each strategy is stated above the boxes. The maximum allowable function evaluations for each run is 20,000.

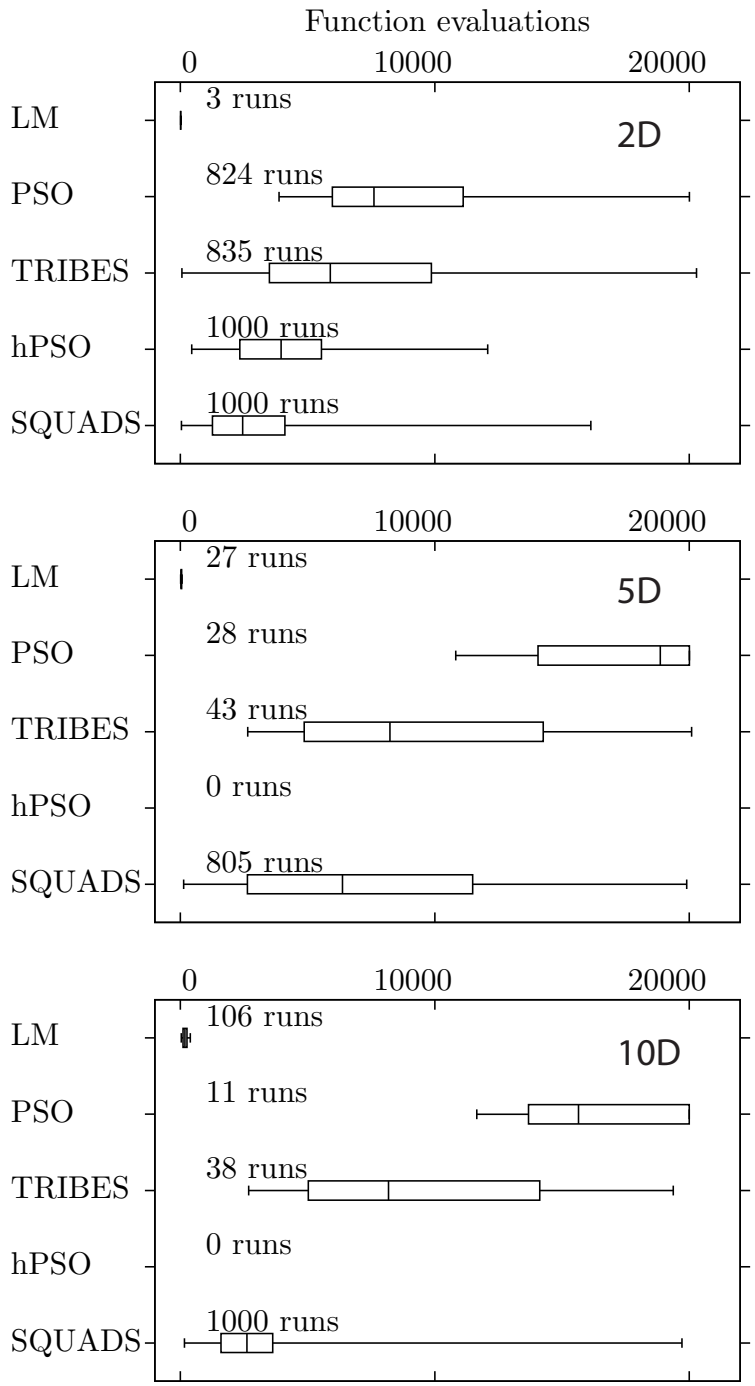

Figure 4: Boxplots of number of function evaluations to reach the global minimum for the 2D, 5D, 10D Griewank function. The boxes represent the 25th to 75th percentile ranges, the bars inside of the boxes represent the median values, and the whiskers represent the min and max values. The number of successful runs out of 1000 for each strategy is stated above the boxes. The maximum allowable function evaluations for each run is 20,000.

Table 1: Particle initialization rules and their selection criteria.

Particle initialization rules:

1. User specified
2. Randomly chosen position within parameter space:
   p_{new_j} =U(p_{min_j}, p_{max_j}), $\quad j = 1, \ldots, D$
3. Randomly chosen within hyperparallelepid surrounding the best position of the swarm with dimensions $(2 \cdot r_j)$ determined by Euclidean distance between the swarm's and tribe's best position:
   $r_j = |p_{best_j} - p_{tribe\ best_j}| \quad j = 1, \ldots, D$
   p_{new_j} =U$(p_{best_j} - r_j, p_{best_j} + r_j) j = 1, \ldots, D$
4. On one of the vertices of the parameter space with equal probability of being the max or min of each dimension:
   if (U$(0, 1) < 0.5$) then $p_{new_j} = p_{min_j}$, else $p_{new_j} = p_{max_j} \quad j = 1, \ldots, D$
5. Randomly chosen within the largest empty hyperparallelepid of the parameter space

Criteria	Initialization rule selection
First particle of the strategy	1
If initial population is greater than 1, other initial particles	5
Particle added to "bad" tribe (tribe adaptation)	randomly chosen between 2 and 5
Mono-particle tribe added (swarm adaptation)	5
LM unable to reduce OF of shaman by 2/3	5

Table 2: Particle displacement rules and their selection criteria based on the status of the particle. $\mathrm{N}(\mu, \sigma)$ is a normal distribution with a mean μ and standard deviation σ, $\mathrm{U}(a, b)$ is a uniform distribution with minimum a and maximum b, $f(-)$ is the value of the objective function, $\mathbf{g} = [g_1, g_2, \ldots, g_D]$ is the location of the particles designated informer, $\mathbf{b} = [b_1, b_2, \ldots, b_D]$ the particle's current best location, and min_j and max_j are the minimum and maximum values for the jth dimension, respectively.

Particle displacement rules:

1. $p_j = \mathrm{U}(min_j, max_j) \quad j = 1, \ldots, D$, change displacement rule to 2 for next time
2. $p_j = \mathrm{N}(g_j, 0.74 \cdot |b_j - g_j|)$
   or, if no informer
   $p_j = \mathrm{N}(b_j, \max(b_j - min_j, max_j - b_j))$
3. $p_j = \frac{f(\mathbf{g})}{f(\mathbf{g})+f(\mathbf{b})} \cdot \mathrm{U}(b_j - |b_j - g_j|, b_j + |b_j - g_j|) + \frac{f(\mathbf{b})}{f(\mathbf{g})+f(\mathbf{b})} \cdot \mathrm{U}(g_j - |b_j - g_j|, g_j + |b_j - g_j|)$
   or, if no informer
   $p_j = \mathrm{N}(b_j, 3 \cdot \max(b_j - min_j, max_j - b_j))$

Particle status	Displacement rule selection
(−=)	randomly choose any rule other than current one
(==)	randomly choose between rule 2 and 3
(+=) or (++)	change to rule 1 with 50% probability