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

Velimir V. Vesselinov *, Dylan R. Harp

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

ARTICLE INFO

ABSTRACT

A new adaptive hybrid optimization strategy, entitled *squads*, is proposed for complex inverse analysis of computationally intensive physics-based 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. *Squads* is designed to be computationally efficient and robust in identification of the global optimum (i.e. maximum or minimum value of an objective function). It integrates global and local optimization using Adaptive Particle Swarm Optimization (APSO) and Levenberg–Marquardt (LM) optimization using adaptive rules based on runtime performance. The global strategy (APSO) optimizes the location of a set of solutions (particles) in the parameter space. The local strategy (LM) 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 APSO. 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; i.e. TRIBES), and an existing hybrid optimization strategy (hPSO). All the strategies are tested on 2D, 5D and 10D Rosenbrock and Griewank polynomial test functions and a synthetic hydrogeologic application to identify the source of a contaminant plume in an aquifer. Tests are performed using a series of runs with random initial guesses for the estimated parameters. The performance of the strategies are compared based on their robustness, defined as the percentage of runs that identify the global optimum, and their efficiency, quantified by a statistical representation of the number of function evaluations performed prior to identification of the global optimum. *Squads* is observed to have better performance than the other strategies for the test functions and the hydrogeologic application when both robustness and efficiency are taken into consideration.

Published by Elsevier Ltd.

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 $f(\theta)$ for a given set of model parameters θ 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 $f(\theta)$ and the system observations. This discrepancy metric is also called the objective function (OF; $\Phi(\theta)$), and is a function of model parameters θ. The metric is represented by a multi-dimensional response hyper-surface; a two-dimensional surface in a three-dimensional space for the case of two model parameters. The OF typically has a complex shape due to multiple minima (representing multiple plausible solutions) and flat regions (representing insensitivity of the OF to the model parameters). An 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

* Corresponding author. Tel. +1 505 665 1458; fax: +1 505 665 8737.
E-mail addresses: vvv@lanl.gov (V.V. Vesselinov), dharp@lanl.gov (D.R. Harp).

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 takes 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.

There are various optimization strategies that have been developed to solve inverse problems. Different optimization techniques have their own strengths that are beneficial when applied to different types of problems. In general, existing optimization strategies can be classified as global and local (Nocedal and Wright, 1999). Global optimization excels at robust exploration of the OF, identifying multiple areas of attraction; however, global strategies are inefficient at locating the parameter set producing an optimal solution within an area of attraction. As a result, in the case of real world model inversions, the application of global optimization may be unfeasible when the model evaluations take a substantial amount of computational time (Keating et al., 2010). Typically, the global optimization strategies do not require smooth objective functions. Local optimization excels at efficiently identifying the optimal model parameters within an area of attraction; however, local optimization is not designed for robust exploration of the OF space outside of an area of attraction. Local optimization is efficient within an area of attraction because it utilizes local information about the gradient and curvature of the OF. This requires estimation of the first and second order derivatives of the OF in the parameter space. As many real world model problems have OF with multiple minima, the use of local optimization alone is not always robust. One of the most commonly used local strategies is Levenberg–Marquardt (LM), which has been applied in many inverse analysis and parameter estimation codes in the geosciences such as UCODE (Poeter and Hill, 1999; Doherty, 2005).

Global and local optimizations are complimentary; where one excels, the other struggles, and vice versa. The benefits of hybrid global/local optimization have been demonstrated previously using particle swarms (Noel and Jannett, 2004; Leontitsis, 2004; Zhang et al., 2007; Ghaffari-Miab et al., 2007). Memetic algorithms (Moscato and Cotta, 2003; Digalakis and Margaritis, 2004; Hart et al., 2005; Krasnogor, 2005; Moscato et al., 2007; Goh et al., 2009), which draw on the concept of *Universal Darwinism* developed by Dawkins (2006), and are hybrid strategies typically associated with combining local optimization with genetic algorithms. The global optimization code AMALGAM, developed by Vrugt and Robinson (2007), has many similarities to these approaches, adaptively shifting preference among a set of evolutionary algorithms. Here, we introduce a new development in hybrid optimization, coupling recent developments in Adaptive Particle Swarm Optimization (APSO) and Levenberg–Marquardt (LM) techniques, producing a novel adaptive hybrid strategy entitled *squads*. In essence, *squads* is a global strategy utilizing local optimization speedup. *Squads* applies LM to a subset of particles at different stages of APSO based on adaptive rules. After the LM update of the particle position, the particle is passed back to APSO and continues to evolve based on APSO rules.

Squads is substantially different than AMALGAM and other memetic algorithms. In AMALGAM, all the optimization strategies are simultaneously applied to tackle the inverse problem, and the AMALGAM algorithm provides a set of adaptive rules to exchange information between optimization strategies and to shift preference among the optimization strategies based on their performance. In *squads*, the APSO strategy drives the inverse problem from the start till the end of the optimization process; the APSO strategy identifies the initial parameter guesses and terminates the optimization process. The LM strategy is utilized by APSO as a local speedup of the optimization process. No information is exchanged between the individual LM runs about the properties of the OF space, they are completely independent, and their performance does not terminate the *squads* run. The performances of *squads* and AMALGAM (as well as other memetic algorithms) are difficult to compare because they are designed to tackle different types of inverse problems. *Squads* is only suitable for problems that have a continuous OF space with well defined derivatives; AMALGAM is more suitable for stochastic inverse problems with very complex and discontinuous parameter space. Both types of inverse problems are commonly solved in practice.

Squads is specifically designed to be robust and computationally efficient, capable of identifying the global minimum with a relatively small number of model evaluations in complex inverse problems with continuous OF space. The name *squads* refers to the hierarchical structure of the population of solutions in the algorithm into small groups, or squads, similar to the APSO algorithm TRIBES (Clerc, 2004). While previous hybrid strategies combining PSO and LM have been introduced (Katare et al., 2004; Chau, 2005), *squads* provides advancements in hybrid optimization as it (1) is adaptive, (2) couples TRIBES (an adaptive multi-swarm PSO algorithm as opposed to standard PSO) with LM, and (3) fully integrates APSO and LM as opposed to a staged optimization approach.

The relative performance of *squads* is compared to other optimization strategies using the Rosenbrock (1960) and Griewank (1981) polynomial test functions. While many other test functions are available exhibiting more complex characteristics, for the purposes of comparing the relative performance of strategies, the Rosenbrock and Griewank functions are deemed appropriate. These test functions are frequently used for comparisons and testing of global and local optimization algorithms (cf. Pan et al., 2010; Rao et al., 2012a, 2012b). Many real world inverse problems are characterized with OF space complexities similar to the complexities represented by these test functions. *Squads* is also applied to solve a hydrogeological problem related to identification of the source of a contaminant plume in an aquifer; this problem is frequently solved in applications related to protection and remediation of groundwater resources (Bagtzoglou et al., 1991; Snodgrass and Kitanidis, 1997; Atmadji and Bagtzoglou, 2001; Sun et al., 2006; Dokou and Pinder, 2009). In order to demonstrate the relative benefits of the hybrid strategy of *squads*, its performance is compared to open-source distributions of LM (Lourakis, 2004), PSO (Particle Swarm Central, 2006), and APSO (Clerc, 2004). Additionally, *squads* is compared to hPSO (Leontitsis, 2004), an open-source hybrid strategy that combines PSO and the Nelder–Mead (1965) downhill simplex strategy implemented in the MATLAB® (The MathWorks Inc, 2003) computing environment.

Our intention in this research is to (1) demonstrate the relative benefit of the adaptive coupling of PSO and LM in *squads* by direct comparisons with PSO, LM, and TRIBES, (2) compare the performance of *squads* against a freely available alternative hybrid approach (hPSO; Leontitsis, 2004). Based on this research, we conclude that the adaptive, hybrid strategy proposed here and implemented in *squads* is an efficient and robust optimizer worthy of further consideration. Further research is required to provide comparisons of *squads* with other algorithms and on other test problems and applications.

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 algorithms 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 PSO 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 $S = 10 + 2\sqrt{D}$ (Particle 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, where the best location of the K informants is denoted as $\mathbf{g}_i = [g_{i,1}, g_{i,2}, \ldots, g_{i,D}]$. A standard value for K is 3 (Particle Swarm Central, 2006). These networks of informers are reinitialized after iterations with no improvement to the global best particle of the swarm. The velocity of the ith particle in the jth dimension is updated from iteration k to $k+1$ as

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

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 the 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 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) has 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 PSO parameters. PSO 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 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 (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 optimization. Clerc (1999) introduced a PSO strategy that adjusts particle locations based on approximations of the gradient of the OF utilizing 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 PSO with the Nelder–Mead simplex strategy (Lagarias et al., 1998, hPSO), 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 APSO with Levenberg–Marquardt (LM). The following provides a detailed description of a coupling of APSO and LM based on adaptive rules, where LM 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 LM is performed using the LevMar library (Lourakis, 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 LM strategy is optimized to work well on many problems using default and internally estimated optimization parameters (Lourakis, 2004). The adaptive rules implemented in *squads* to control the performance of LM speedups during APSO are also designed to be general and capable to tackle problems with different complexity.

A flow diagram of *squads* is presented in Fig. 1. 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 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 shamans, 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 iteration. This is performed stochastically, by comparing the fraction of particles in the tribe that improved their location in the last move
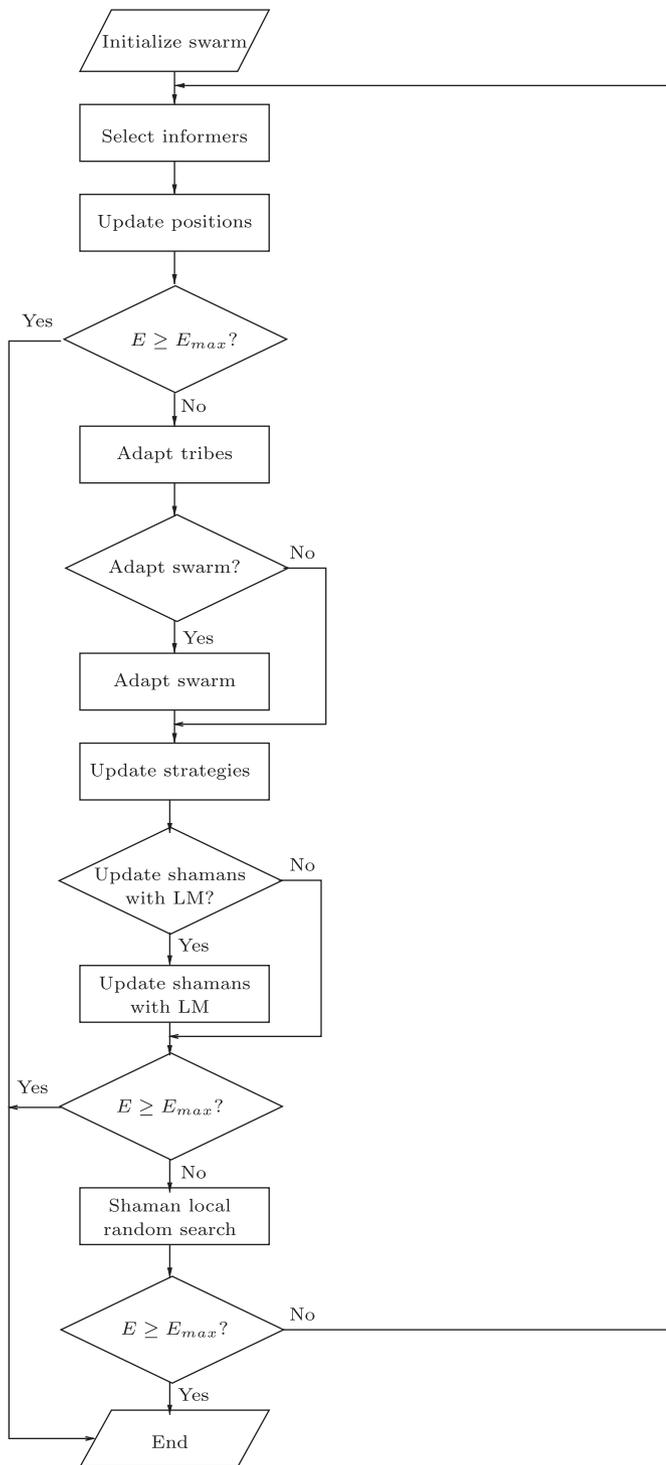
Fig. 1. Flow diagram of *squads*. E is the current number of model evaluations and E_{max} is the allowable number of model evaluations. Decisions to *adapt swarm* or *update shamans with LM* are determined by adaptive rules.

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. Therefore, the larger the fraction of improved particles in a tribe, the greater probability that the tribe will be considered a *good 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, while removing the worst particle from a *good tribe* is intended to intensify the search focusing on the particles at the best locations. This may seem counterintuitive to some who feel that *bad tribes* are being rewarded and *good tribes* punished, however, it must be noted that these are strategies for promoting exploration and intensification.

The swarm adaptation occurs either every $N_t*(N_t-1)/4$ 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 each of 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) the particle's 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 APSO. This postpones LM until APSO 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.

LM 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 LM to estimate the local gradient and curvature of the OF. These calculations utilize numerical derivatives of the OF components in Eq. (3) with respect to the model parameters (also called the Jacobian matrix). Based on the Jacobian matrix, LM 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 OF. LM 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 Eq. (3) is represented by the discrepancy between model simulated values $f_i(\boldsymbol{\theta})$ and corresponding observations o_i, where $i=1,\ldots,N$, and N is now the number of observations. For example, frequently the OF is computed as

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

Table 1
Particle initialization rules and their selection criteria.

Particle initialization rules:
1. User specified
2. Randomly chosen position within parameter space:
$p_{new_j} = U(p_{min_j}, p_{max_j})$, $j = 1, \ldots, D$
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:
$r_j = |p_{best_j} - p_{tribe\ best_j}|$ $j = 1, \ldots, D$
$p_{new_j} = U(p_{best_j} - r_j, p_{best_j} + r_j)$ $j = 1, \ldots, D$
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 $p_{new_j} = p_{max_j}$ $j = 1, \ldots, D$
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, 3, 4, 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. $N(\mu, \sigma)$ is a normal distribution with a mean μ and standard deviation σ, $U(a, b)$ is a uniform distribution with minimum a and maximum b, $f(-)$ is the value of the OF, $\mathbf{g} = [g_1, g_2, \ldots, g_D]$ is the location of the particle's 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. Refer to the text for a description of the particle status notation.

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

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

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

The following criteria are defined by default in LevMar to terminate the LM optimization (Lourakis, 2004), and applied in the LM updating of *squads* as well: (1) the change in any parameter is less than 10^{-10} times the current parameter value; (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 stand-alone LM is performed; eight in *squads*). Refer to Lourakis (2004) or Vesselinov (2011) for additional details on the implementation of these criteria. 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 user-specified cutoff value.

The final step of each iteration is to perform a random local search in the empty space around each shaman (Clerc, 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. According to Clerc (2004), this often results in an improved shaman position at little additional expense.

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 LM 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 θ 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. APSO is performed in the transformed parameter space bounded within $[-\pi/2; \pi/2]$ in all dimensions, 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 APSO 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 APSO, 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. As we have implemented this transformation for LM in cases without prior information, we have not explored its use in cases with prior information or for estimators based on the maximum likelihood principle.

4. Test functions

The performance of *squads* is compared to other optimization strategies by optimizing the Rosenbrock and Griewank test functions. The Rosenbrock and Griewank functions present difficult optimization problems exhibiting frequently observed complexities in OF topology in real world problems (e.g. Rosenbrock, 1960; Griewank, 1981; Clerc, 2006; Cooren et al., 2009). While many other test functions are available, for the comparison of relative optimization strategy performance conducted here, the Rosenbrock and Griewank functions are deemed appropriate.

The response function defined by the Rosenbrock function composed 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 Fig. 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 OF by LM requires the test function to be represented as a summation of parts as in Eq. (3). The summation components of $\Phi_r(x_1, \ldots, x_D)$ can be expressed as

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

and

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

producing $2(D-1)$ OF components where Eqs. (9) and (10) define the odd and even numbered components, respectively; therefore, the number of components (also called *observations* in the case of inverse problems) in the 2D, 5D, and 10D cases are 2, 8, and 18, respectively. LM 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 OF. In most real world problems, the analytical
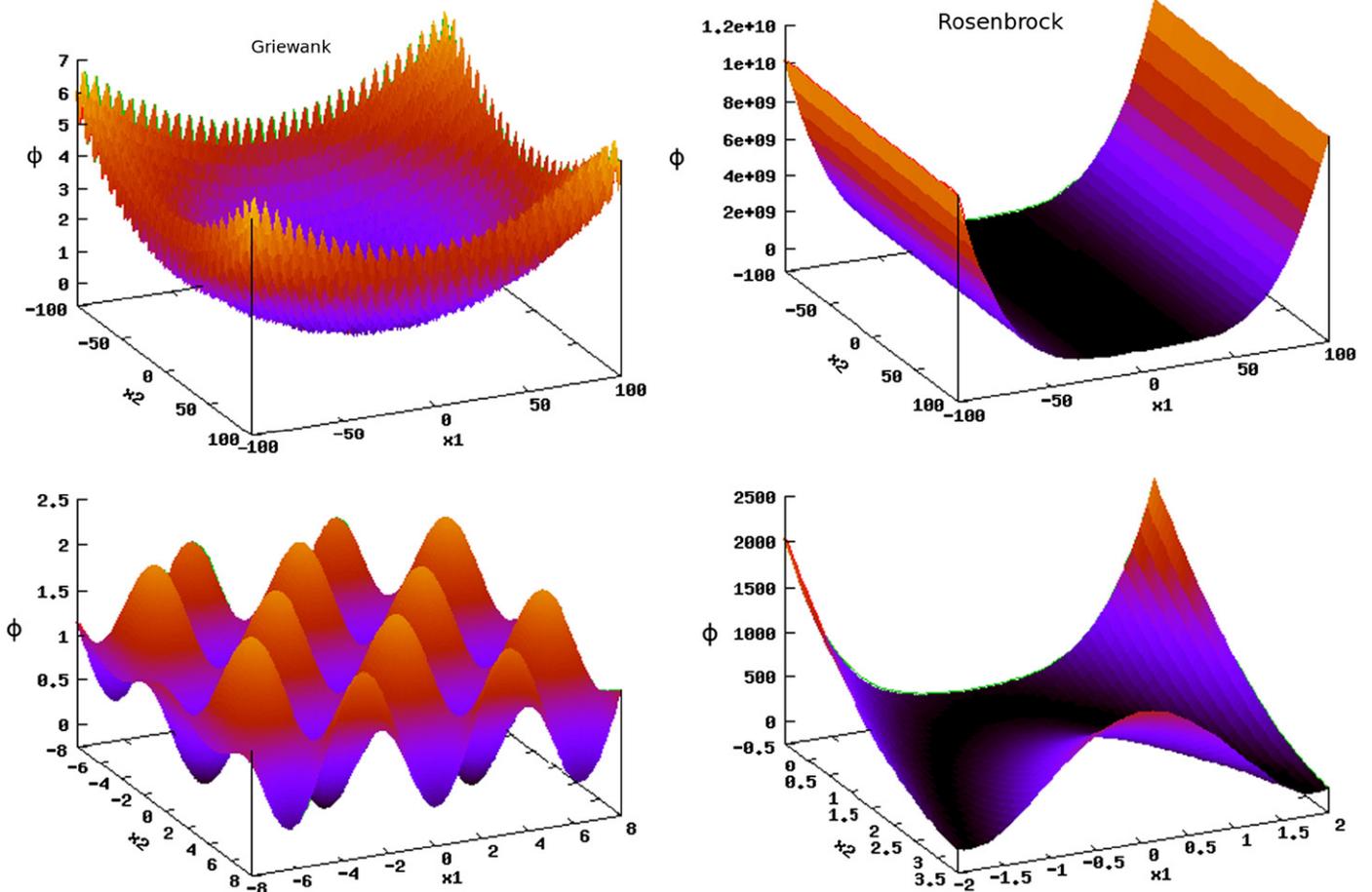

Fig. 2. Rosenbrock and Griewank polynomial test functions with global minima at (1,1) and (0,0), respectively. Note the different parameter ranges on the top and bottom rows. The top row shows the parameter space explored by the optimization strategies. The bottom row focuses on the parameter space near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

Table 3
Well coordinates, screen top (z_{top}) and bottom (z_{bot}) depths below the water table, and year and value of observed contaminant concentrations.

Well	x (m)	y (m)	z_{top} (m)	z_{bot} (m)	t (a)	c (ppb)
w01	1503	1954	5.57	12.55	49	0
w02	2113	1479	36.73	55.14	49	0
w03	418	950	0	15.04	49	0
w04	1377	1534	13.15	20.41	44	350
					49	432
w05	3268	1074	26.73	33.71	49	0
w06	2112	2294	69.01	83.98	49	0
w07	2086	2284	11.15	18.19	49	0
w08	2770	2119	4.86	11.87	49	0
w09	975	1450	3.66	10.09	49	981
w10	723	1599	3.32	9.63	49	1.1
			23.2	26.24	49	0.1
w11	1850	1368	4.94	7.99	49	22
			32.46	35.48	49	0.3
w12	1761	1636	3.59	6.64	49	15
			32.51	38.61	49	0.17
w13	1485	1149	3	6	50	72
			36	42	50	0.26
w14	972	869	3	6	50	0
w15	940	1160	3	6	50	38

Table 4
'True', minimum, and maximum parameter values for the contaminant transport test case.

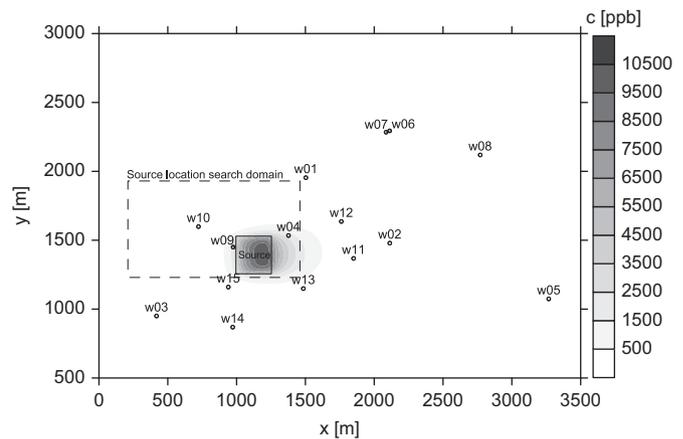| Estimate	x_s (m)	y_s (m)	x_d (m)	y_d (m)
'true'	1124	1393	258	273
min	210	1230	1	1
max	1460	1930	500	500

Fig. 3. Map of monitoring well locations. The *true* source is shown as a solid rectangle. The search domain for x_s and y_s is shown as a dotted rectangle. The contaminant concentration plume at $t=49$ years is represented by the color map. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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

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

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

The summation components of $\Phi_g(x_1,\ldots,x_D)$ can be defined as

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

Therefore, the number of components (*observations*) equals the number of model parameters.

The multidimensional Griewank function is important for testing 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 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. Contaminant source identification test case

Optimization is commonly employed to calibrate physics-based models to available observations. We demonstrate the

optimization strategies on a hydrogeologic application to identify the center (x_s, y_s) and dimensions (x_d, y_d) of a parallelepiped contaminant source in an aquifer using observations of contaminant concentrations from monitoring wells near the expected source location. The synthetic groundwater flow and transport problem is three-dimensional and semi-infinite, the top model boundary aligns with the top of the aquifer, and the model extends to infinity laterally and with depth. The locations of the monitoring wells, depths below the aquifer top boundary of the top and bottom of the screens, times of observation after the contaminant release, and observed contaminant concentrations are presented in Table 3. The parameter values used to generate the *true* concentrations at the monitoring wells and the minimum and maximum parameter values allowed in optimization runs are presented in Table 4. The *true* location of the source, the location of monitoring wells, and contaminant concentrations at $t=49$ years since the contaminant was released are presented in Fig. 3. A similar model is presented in Harp and Vesselinov (2011) with additional details.

The OF for the contaminant transport test case is expressed as a *sum of squared residuals* (SSR) as

$$\Phi(\boldsymbol{\theta}) = \sum_{i=1}^{N}(\hat{c}_i(\boldsymbol{\theta}) - c_i)^2, \quad (13)$$

where $\hat{c}_i(\boldsymbol{\theta})$ is the ith simulated concentration resulting from $\boldsymbol{\theta}$, c_i is the ith observed concentration, and N is the number of observations. In summary, there are four unknown model parameters constrained by 20 observations.

The simulated contaminant concentrations ($\hat{c}$) are produced from an analytical contaminant transport model encoded in MADS (Wexler, 1992; Wang and Wu, 2009; Vesselinov, 2011) (refer to Harp and Vesselinov, 2011 for additional simulation details). Due to the rounding of the observed concentrations, a value of $\Phi = 0.55$ is obtained from the true parameter values.
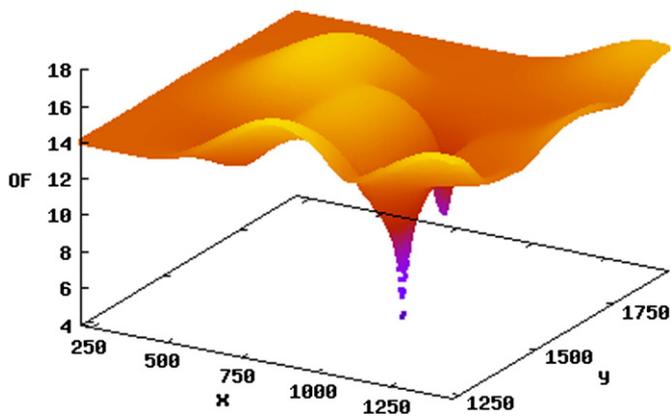
Fig. 4. Contaminant source identification OF for contaminant source locations defined by x_s and y_s. The minimum OF value for each combination of x_s and y_s are plotted considering allowable ranges for source lateral dimensions, x_d and y_d.

The OF resulting from Eq. (13) plotted as a function of x_s and y_s is presented in Fig. 4. The values plotted in Fig. 4 are the lowest values for the OF at each combination of x_s and y_s considering allowable ranges for x_d and y_d. In other words, this is the OF that the strategies would traverse if they knew the optimal values for x_d and y_d for each combination of x_s and y_s. Note that the actual 4D OF is more complicated than this 2D representation. Features from both the Griewank and Rosenbrock test functions can be seen in this representation of the OF with multiple areas of attraction (three suboptimal minima and one global minimum), regions of parameter insensitivity (flat regions), and narrow, curved valleys.

Even though the contaminant transport model is analytical, the computational time is substantially higher than for the test functions. Within MADS, the number of function evaluation per second is $\sim 40,000$ for the test functions compared to ~ 400 for the contaminant transport model. For hPSO, the number of function evaluations per second is $\sim 11,000$ for the test functions compared to ~ 2 for the contaminant transport model; the substantial increase in hPSO computation time between the test functions and the hydrogeologic application is due to external coupling between the Matlab computing environment (applied to execute hPSO) and the external C based transport simulator.

6. 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. LM is an implementation of LevMar (Lourakis, 2004) (the same LM included in *squads*), PSO is an implementation of Standard PSO 2006 (Particle Swarm Central, 2006), TRIBES is an APSO strategy implemented in the code described in Clerc (2006), hPSO is a freely available hybrid optimization code from Leontitsis (2004). LM, PSO, TRIBES and *squads* are built into the code MADS (Vesselinov, 2011), 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 optimization parameters for PSO and hPSO are set to values that have been demonstrated to perform well in many test cases (Particle Swarm Central, 2006) as $w=0.72$, $c_1=1.2$ and $c_2=1.2$ (refer to Eq. (1)).

The strategies are tested on both functions by performing 1000 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 (Fig. 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 OF. The ability of LM to identify the global minimum depends on whether the minimum encountered by LM is local or global.

Figs. 5 and 6 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 boxes represent the 25th–75th percentile ranges, the bars inside of the boxes represent the median values, and the whiskers represent the minimum and maximum values. The fraction of successful runs out of the attempted runs are presented above the boxes. Note that the statistical definitions of the boxplots are not accurate for the cases where the number of successful runs does not present a statistically significant sample. Robustness is defined as the percentage of successful runs (i.e. fraction of successful runs * 100) achieved within a given number of function evaluations. Efficiency is summarized by the statistics presented in the boxplots.

For the Rosenbrock function (Fig. 5), the robustness of LM decreases from the 2D case to the 10D case from 36% to 0%. The robustness of PSO and TRIBES 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. The 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 similar for the 2D and 5D cases (Fig. 5). 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 (Fig. 6), 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 derivative-based strategy (Locatelli, 2003), the robustness is only 3%. Since LM is local, it is not surprising that LM frequently converges at non-optimal 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) 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 derivative-based (LM) strategy becomes more robust, while
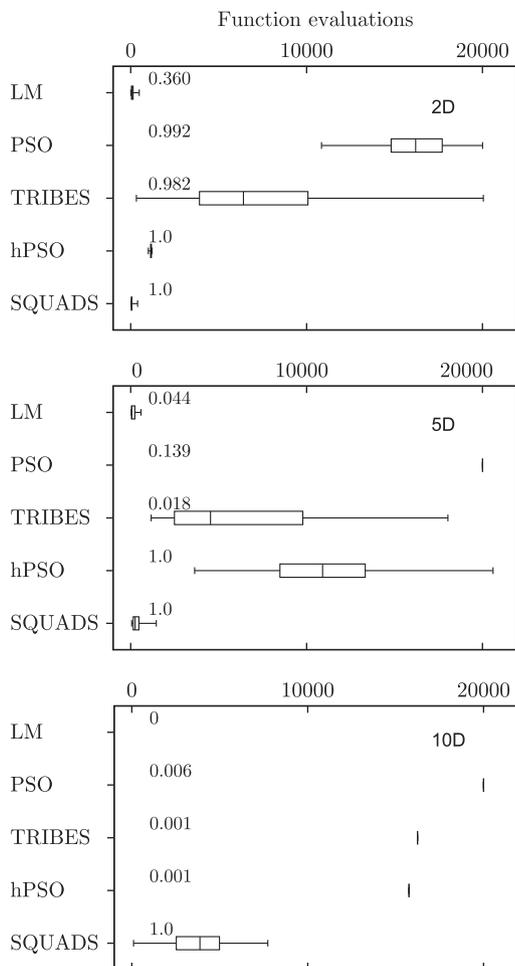
Fig. 5. Boxplots of number of function evaluations to reach the global minimum for the 2D, 5D, and 10D Rosenbrock function. The boxes represent the 25th to 75th percentile ranges, the bars inside of the boxes represent the median values, and the whiskers represent the min and max values. Note that the statistical definitions are not accurate when the number of successful runs does not present a statistically significant sample and some boxplots have been reduced to single vertical lines in these cases. The fraction 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.
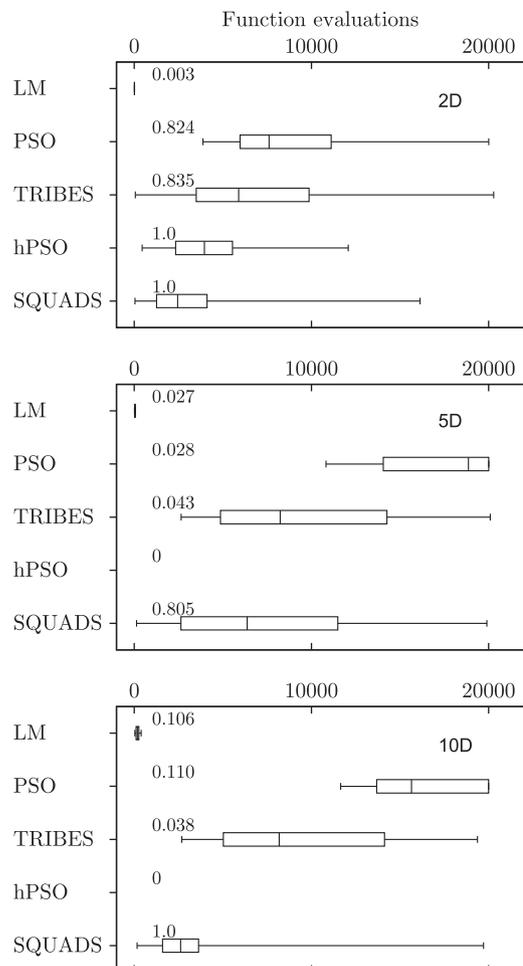
Fig. 6. Boxplots of number of function evaluations to reach the global minimum for the 2D, 5D, 10D Griewank function. The boxes represent the 25th–75th percentile ranges, the bars inside of the boxes represent the median values, and the whiskers represent the min and max values. Note that the statistical definitions are not accurate when the number of successful runs does not present a statistically significant sample and some boxplots have been reduced to single vertical lines in these cases. The fraction 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.
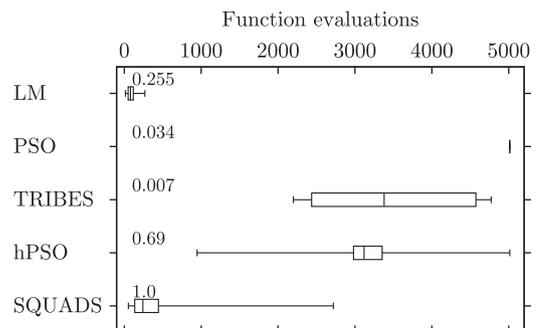
the global (APSO) strategy becomes less robust. At $D=2$, *squads* is both more robust and efficient than the other global methods (Fig. 6). At $D=10$, *squads* benefits from the local derivative-based search, which performs better at higher dimensions. The 5D Griewank function is observed to be the 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 70,000 (results are not shown here). In summary for the Griewank cases, *squads* is observed to have the best performance when both robustness and efficiency are taken into consideration than the other strategies (Fig. 6).

The performance of the strategies is demonstrated on the hydrogeologic application in contaminant source identification presented in Section 5. A boxplot of the necessary function evaluations for successful runs is presented in Fig. 7. As with the test functions, 1000 runs are performed for each strategy, except for hPSO, where only 100 runs are performed due to the computational expense of evaluating the contaminant transport model from hPSO (~ 2.2 function evaluations per second). The

Fig. 7. Boxplots of number of function evaluations to reach the global minimum for the hydrogeologic application presented in Section 5. The boxes represent the 25th–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 when the number of successful runs does not present a statistically significant sample and some boxplots have been reduced to single vertical lines in these cases. The fraction of successful runs (out of 1000 for LM, PSO, TRIBES, and *squads*; out of 100 for hPSO) is stated above the box. The maximum allowable function evaluations for each run is 5000.

maximum number of function evaluations allowed for each optimization run is limited to 5000. An optimization run is considered successful if the OF is reduced below a value of 1. This ensures that the solution has reached the area of attraction around the global minimum as the suboptimal minima of the OF are all greater than 1. As with the test functions, it is observed that LM is efficient on the hydrogeologic application, but only approximately 26% robust. PSO and TRIBES are observed to be inefficient and not robust in this case requiring high numbers of function evaluations with approximately 3% and 0.7% robustness, respectively. The hPSO demonstrates some robustness at 69%, but with low efficiency in general with a large variability in the necessary number of function evaluations. *squads* demonstrates high robustness at 100% with higher efficiency than PSO, TRIBES, and hPSO. While the efficiency of LM is better than *squads* in this case, this is with a significantly lower robustness.

It is important to emphasize that in all test cases, *squads* converges at a lower number of model evaluations than PSO, TRIBES and hPSO. This is manifested by the minimum values of the boxplots in Figs. 5–7. 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 (Figs. 5–7). This demonstrates that more frequently *squads* may converge with lower number of functional evaluations.

The application of *squads* is performed using the code MADS (Vesselinov, 2011). 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.

7. Conclusions

A new adaptive global hybrid optimization strategy called *squads* is developed for solving computationally intensive inverse problems. Comparisons of robustness and efficiency between *squads* and LM, PSO, TRIBES, and hPSO are performed using 2D, 5D, and 10D Griewank and Rosenbrock test functions and a 4D synthetic hydrogeologic application. Based on this research, the following conclusions can be drawn:

1. Based on comparisons of robustness with a local strategy (LM), *squads* is robust in avoiding becoming stuck in local minima during the optimization as typically observed in the case of local derivative-based strategies.
2. Based on comparisons of strategy efficiency with a local strategy (LM), *squads* has nearly the same efficiency as local optimization for the Rosenbrock function, but substantially lower efficiency for the Griewank and hydrogeologic application.
3. Based on comparisons of strategy efficiency with global strategies (PSO and TRIBES), *squads* reduces the number of model runs typically required of global strategies by efficiently exploring local areas of attraction.
4. Based on comparisons of strategy robustness and efficiency with an alternative hybrid strategy (hPSO), *squads* locates the global minimum at a greater frequency and with less function evaluations.
5. When both robustness and efficiency are taken into consideration to evaluate strategy performance, *squads* outperforms LM, PSO, TRIBES, and hPSO on the Rosenbrock and Griewank functions and on the hydrogeologic application evaluated here.
6. The results of this study indicate that *squads* is a promising alternative for calibration and parameter estimation of physics-based models.
7. Additional comparisons to other strategies utilizing other test functions and applications are required to further evaluate *squads* performance.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org.10.1016/j.cageo.2012.05.027.

References

Atmadji, J., Bagtzoglou, A.C., 2001. State of the art report on mathematical methods for groundwater pollution source identification. Environmental Forensics 2, 205–214.
Bagtzoglou, A., Tompson, A., Dougherty, D., 1991. Probabilistic simulation for reliable solute source identification in heterogeneous porous media. In: Water Resources Engineering Risk Assessment. NATO ASI Series, vol. G29, pp. 189–201.
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.
Chau, K., 2005. A split-step PSO algorithm in prediction of water quality pollution. Advances in Neural Networks (2005) 1034–1039.
Clerc, M., 1999. The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. IEEE, pp. 1951–1957.
Clerc, M., 2006. Particle Swarm Optimization. ISTE, London.
Clerc, M., July 2004. Tribes, an adaptive particle swarm optimization algorithm in C. ⟨http://clerc.maurice.free.fr/pso/Tribes/TRIBES-D.zip⟩, (Accessed on 4 November 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.
Dawkins, R., 2006. The Selfish Gene. Oxford University Press, USA.
Digalakis, J., Margaritis, K., 2004. Performance comparison of memetic algorithms. Applied Mathematics and Computation 158 (1), 237–252.
Doherty, J., 2005. PEST model-independent parameter estimation: User manual.
Dokou, Z., Pinder, G.F., 2009. Optimal search strategy for the definition of a DNAPL source. Journal of Hydrology 376, 542–556.
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.
Goh, C., Ong, Y., Tan, K., 2009. Multi-Objective Memetic Algorithms, vol. 171. Springer Verlag.
Griewank, A.O., 1981. Generalized descent for global optimization. Journal of Optimization Theory and Applications 34, 11–39.
Harp, D.R., Vesselinov, V.V., 2011. An agent-based approach to global uncertainty and sensitivity analysis. Computers & Geosciences 40, 19–27.
Hart, W., Krasnogor, N., Smith, J., 2005. Recent advances in memetic algorithms. vol. 166. Springer Verlag.
Jessell, M., 2001. Three-dimensional geological modelling of potential-field data. Computers & Geosciences 27, 455–465.
Katare, S., Kalos, A., West, D., 2004. A hybrid swarm optimizer for efficient parameter estimation. In: Congress on Evolutionary Computation, 2004. CEC2004, vol. 1. IEEE, pp. 309–315.
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.
Krasnogor, N., 2005. Towards robust memetic algorithms. Recent Advances in Memetic Algorithms, 185–207.
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. Quarterly of Applied Mathematics 2, 164–168.
Locatelli, M., 2003. A note on the Griewank test function. Journal of Global Optimization 25, 169–174.
Lourakis, M., July 2004. levmar: Levenberg–Marquardt nonlinear least squares algorithms in C/C++. ⟨http://www.ics.forth.gr/~lourakis/levmar/⟩ (Accessed on 4 November 2010).
Marquardt, D., 1963. An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics 11, 431–441.
Meek, D., 2001. A semiparametric method for estimating the scale of fluctuation. Computers & Geosciences 27, 1243–1249.
Moscato, P., Berretta, R., Cotta, C., 2007. Memetic Algorithms. Wiley Encyclopedia of Operations Research and Management Science.
Moscato, P., Cotta, C., 2003. A gentle introduction to memetic algorithms. Handbook of Metaheuristics, 105–144.

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

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

Noel, M.M., Jannett, T.C., 2004. Simulation of a New Hybrid Particle Swarm Optimization Algorithm. IEEE, pp. 150–153.

Pan, Q.-K., Suganthan, P., Tasgetiren, M.F., Liang, J., 2010. A self-adaptive global best harmony search algorithm for continuous optimization problems. Applied Mathematics and Computation 216 (3), 830–848.

Particle 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., 1995. Reducing uncertainty associated with ground-water flow and transport predictions. Ground Water 33 (November (6)), 899–904.

Rao, R., Savsani, V., Vakharia, D., 2012a. Teaching learning-based optimization: an optimization method for continuous non-linear large scale problems. Information Sciences 183 (1), 1–15.

Rao, R.V., Savsani, V.J., Rao, R.V., Savsani, V.J., 2012b. Applications of hybrid optimization algorithms to the unconstrained and constrained problems. In: Mechanical Design Optimization Using Advanced Optimization Techniques. Springer Series in Advanced Manufacturing. Springer London, pp. 123–131.

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.

Snodgrass, M.F., Kitanidis, P.K., 1997. A geostatistical approach to contaminant source identification. Water Resources Research 33 (April (4)), 537–546.

Sun, A.Y., Painter, S.L., Wittmeyer, G.W., 2006. A robust approach for iterative contaminant source location and release history recovery. Journal of Contaminant Hydrology 88, 181–196.

The MathWorks Inc., 2003.

Vesselinov, V.V., 2011. MADS, Modeling Analysis and Decision Support toolkit in C. ⟨http://mads.lanl.gov⟩, (Accessed on 2 December 2011).

Vesselinov, V.V., Neuman, S.P., Illman, W.A., 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 (December (12)), 3019–3041.

Vrugt, J., Robinson, B., 2007. Improved evolutionary optimization from genetically adaptive multimethod search. Proceedings of the National Academy of Sciences USA 104 (3), 708.

Wang, H., Wu, H., 2009. Analytical solutions of three-dimensional contaminant transport in uniform flow field in porous media: a library. Frontiers in Environmental Science and Engineering in China 3 (1), 112–128.

Wexler, E.J., 1992. Chapter B7 Analytical solutions for one-, two-, and three-dimensional solute transport in ground-water systems with uniform flow, Techniques of Water-Resources Investigations of the United States Geological Survey. USGS, pp. 80–133.

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.