



Using a derivative-free optimization method for multiple solutions of inverse transport problems

Jerawan C. Armstrong¹ · Jeffrey A. Favorite¹

Received: 24 November 2014 / Revised: 29 September 2015 / Accepted: 4 December 2015 /

Published online: 14 January 2016

© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Identifying unknown components of an object that emits radiation is an important problem for national and global security. Radiation signatures measured from an object of interest can be used to infer object parameter values that are not known. This problem is called an inverse transport problem. An inverse transport problem may have multiple solutions and the most widely used approach for its solution is an iterative optimization method. This paper proposes a stochastic derivative-free global optimization algorithm to find multiple solutions of inverse transport problems. The algorithm is an extension of a multilevel single linkage (MLSL) method where a mesh adaptive direct search (MADS) algorithm is incorporated into the local phase. Numerical test cases using uncollided fluxes of discrete gamma-ray lines are presented to show the performance of this new algorithm.

Keywords Inverse transport problem · Derivative-free optimization · Stochastic global optimization · Multilevel single linkage · Mesh adaptive direct search

1 Introduction

Inverse transport problems are problems in which radiation signatures are used to identify unknown components of a radioactive source/shield system. Application areas of inverse transport problems include national security, material safeguards, and radioactive waste management. For a multilayered gamma-ray source/shield system, unknown system parameters may include the number of layers, interface

✉ Jerawan C. Armstrong
jerawan@lanl.gov

Jeffrey A. Favorite
fave@lanl.gov

¹ XCP-3: Monte Carlo Methods, Codes, and Applications Group, Computational Physics Division, Los Alamos National Laboratory, MS F663, Los Alamos, NM 87545, USA

locations, the outer dimension (if it cannot be observed), source isotopic compositions, shield materials, and material mass densities. Inverse transport problems can be solved via an optimization approach whose goal is to find the unknown system parameters that minimize a cost difference between the quantities of interest measured from a system and the quantities of interest calculated using a set of postulated parameters. In applications, inverse transport problems may have multiple solutions, and we want to find *all* solutions so that a better decision can be made. Our definition of inverse problem *solutions* is given in Sect. 2.

A few iterative optimization methods, such as Levenberg–Marquardt and differential evolution (DE) algorithms (Mattingly and Dean 2010; Bledsoe et al. 2011a, b), have been applied to solve inverse transport problems. We have investigated alternative optimization methods. In Armstrong and Favorite (2012), we applied a mesh adaptive direct search (MADS) algorithm to solve the inverse transport problem of material interface location identification in one-dimensional spherical radioactive source/shield systems where all other system parameters are known. The Levenberg–Marquardt, DE, and MADS methods are designed to locate only one minimum. Since our goal is to find all combinations of system parameters for which the difference between the measured and calculated values is minimized, we want to use global optimization methods to search for multiple local minima. We also want to use derivative-free optimization (DFO) methods so that only one transport calculation is performed for each model during iterative schemes.

It is well known that in general solving global optimization problems is very challenging (Dixon and Szego 1975; Stephens and Baritomba 1998; Neumaier 2004). The global optimization problem is to find the lowest value of a function in some domain. It can be described as follows:

$$\begin{aligned} &\text{Given a compact set } X \subset \mathbb{R}^n \quad \text{and} \quad f : X \rightarrow \mathbb{R} \cup \{\infty\}, \\ &\text{find } x^* \in X \text{ such that } f(x^*) \leq f(x) \quad \text{for all } x \in X. \end{aligned} \quad (1)$$

We refer to x^* as a *global minimizer* of f and $f(x^*)$ as the *global minimum*. A function f and a set X are respectively called an objective function and a feasible region. A global optimization problem has only one global minimum but may have multiple different local minima, and a global minimizer is a local minimizer but the reverse is not necessarily true. Global optimization methods can be categorized into two classes, deterministic and stochastic. Typically, stochastic methods start from a sample of points drawn randomly from X while deterministic methods locate the global minimum by an exhaustive search over X or a dense countable subset of X (Boender et al. 1982). Some stochastic methods contain a local phase for finding all local minima in the hope that one of these local minima is the global one. This causes difficulty in solving global optimization problems because there is no constructive method to decide whether or not all local minima are found.

One of the most reliable and efficient stochastic global optimization methods is the multilevel single linkage (MLSL) method, also known as the Boender–Rinnooy Kan–Stougie–Timmer (BRST) algorithm (Boender et al. 1982). The MLSL method was originally developed for bounded constrained optimization problems and was extended to optimization problems with inequality and equality constraints (Sendin

et al. 2009). It is a multistart stochastic global optimization method that relies on a combination of random sampling, clustering, and local search strategies. Its goal is to find all local minima. Each MLSL iteration consists of two search phases, global and local. In the global phase, a finite number of sample points is drawn from a uniform distribution over a feasible region X and an objective function value is evaluated at each sample point. In the local phase, a local optimization method is started from points chosen from a subset of sample points with the lowest objective function values, where the number of local searches is reduced by applying a single linkage clustering technique to sample points of interest. Any local optimization algorithm (including DFO local search algorithms) may be utilized in the MLSL local phase.

DFO is a field of nonlinear optimization where methods that do not use derivatives have been developed and rigorously analyzed (Conn et al. 2009). Directional direct-search algorithms (as opposed to model-based methods) are DFO methods developed to solve optimization problems by not using derivatives but directly making comparisons of objective function values. A recent review of DFO algorithms and comparison of software implementations can be found in Rios and Sahinidis (2013).

The DFO algorithm that we consider in this paper is the class of MADS algorithms. The MADS method is an iterative method introduced and analyzed by Audet and Dennis where nonsmooth calculus is used to prove its convergence properties (Audet and Dennis Jr. 2006; Abramson and Audet 2006). At each iteration, MADS generates a finite number of trial points lying on a spatial discretization called the *mesh*, evaluates an objective function at some trial points with an attempt to find a point with a lower objective function value than the current best point found so far, and then adapts the fineness of the mesh to approach a minimum. A point with a lower objective function value than the current best point found so far is called an improved mesh point. The MADS iteration consists of two steps called SEARCH and POLL. The SEARCH step is optional and any search algorithm can be used to globally explore the mesh, while the POLL step searches near an improved mesh point.

The MADS method is designed for a blackbox optimization problem with general constraints. A blackbox optimization problem is an optimization problem whose objective function and/or functions defining the feasible region are evaluated by computational codes, which may return outputs that cannot be trusted (i.e., that are numerically reliable, as discussed below in this section) even when all parameter values used in calculations are feasible (Audet 2014). The constraints of blackbox optimization problems can be classified as unrelaxable, relaxable, or hidden (Audet et al. 2010). Hidden constraints are defined as feasible points for which computational codes (in the *blackbox*, discussed later in this section) return untrustworthy outputs. Hidden constraints cannot be defined mathematically and can only be evaluated after computational codes return results. Several strategies are developed in the MADS method to handle constraints, and one of these strategies is an extreme barrier approach. This technique rejects an infeasible point or a point that fails a hidden constraint by setting its objective function value to infinity (Audet et al. 2010). The extreme barrier technique for handling constraints is necessary if a

computational code cannot be started or if it returns results that cannot be trusted when a constraint is violated.

We formulate inverse transport problems as blackbox optimization problems. Characteristics of inverse transport optimization problems include: (a) no assumption is made on the functions used to define the objective function, (b) no derivative information is available to be exploited, (c) only objective function values calculated by a computational code are used by an optimization algorithm during an iterative scheme, and (d) a computational code may return outputs that cannot be trusted even when feasible points are used in calculations (i.e., there are hidden constraints). Blackbox optimization problems are common in real optimization applications (Audet 2014).

In this paper, we consider a passive gamma-ray detection problem. Radioactive isotopes decay with the release of gamma rays of specific energies; these are called *lines* in the measured gamma-ray spectrum. Gamma rays arriving at an external (stand-off) detector with their full characteristic line energy can only do so if they do not interact with materials between their source point and the detector. Such gamma rays are said to be *uncollided*. The quantities of interest used in this application are the uncollided fluxes of discrete gamma-ray lines. Details will be given in Sect. 2.

The objective function values can be trusted if the numerical subroutines used to calculate them terminate normally. Feasible points used to calculate the uncollided fluxes where the numerical integration method does not terminate normally define the hidden constraints. We cannot include hidden constraints in the problem definition because these constraints exist due to the failure of the numerical integration method used to calculate the objective function values. Since it is very difficult (or probably impossible) to make the numerical integration subroutine work for all feasible hidden constraint points, we apply the extreme barrier approach to handle the hidden constraints.

Since the inverse transport optimization problem is a blackbox optimization problem, the MADS method is well suited. We want to locate multiple minima, but the MADS method is designed to locate a single minimum. We thus propose to integrate the MADS algorithm into the MLSL method to locate multiple minima in a single run. We incorporate the MADS algorithm with an empty SEARCH step into the local minimization phase of the MLSL method where points that fail hidden constraints in both MLSL global and local phases are handled by the extreme barrier method. We then apply the new combined algorithm to solve for multiple solutions of inverse transport problems using uncollided fluxes of discrete gamma-ray lines.

This paper is organized as follows. Section 2 provides the background of inverse transport problems. Section 3 briefly summarizes the MLSL and MADS methods. Section 4 presents a new variant of the MLSL algorithm that we use to find multiple inverse transport solutions in a single run. Section 5 gives some comments on inverse transport solutions. Numerical test cases are presented in Sect. 6 and conclusions are given in Sect. 7.

2 Background

We consider a gamma-ray source and radiation shield system where both the source and the shield may be multilayered but each layer is assumed to be homogeneous. Information on passive gamma-ray detection and measurements can be found in PANDA (2007). The source emits gamma rays isotropically at discrete energies called *lines*, which can be resolved quite well using a high-purity germanium detector. It is assumed that the detector can resolve energy lines well enough that there is no scattering into the energy lines. Under these conditions, the angular flux of gamma rays of energy E_i within the system is described by the Boltzmann transport equation without scattering,

$$\hat{\Omega} \cdot \nabla \psi^{E_i}(r, \hat{\Omega}) + \Sigma_t^{E_i}(r) \psi^{E_i}(r, \hat{\Omega}) = q^{E_i}(r), \quad i = 1, \dots, N_l, \quad (2)$$

where $\psi^{E_i}(r, \hat{\Omega})$ is the angular flux of gamma rays of energy E_i at position r and angle $\hat{\Omega}$ ($\gamma \cdot \text{cm}^{-2} \cdot \text{s}^{-1}$), $\Sigma_t^{E_i}(r)$ is the total macroscopic photon cross section at energy E_i and position r (cm^{-1}), $q^{E_i}(r)$ is the source rate density of gamma rays of energy E_i at position r ($\gamma \cdot \text{cm}^{-3} \cdot \text{s}^{-1}$), E_i is the energy line at index i , and N_l is the number of energy lines considered.

There are five independent variables in the transport equation (2): three spatial coordinates for position $r \in \mathbb{R}^3$ and two angles $\hat{\Omega} \in [0, 2\pi] \times [0, \pi]$. We assume that the spatial domain of the problem, denoted Υ , has external surface Γ_s and outside of Υ is an infinite region of vacuum. Thus, no gamma rays enter into Υ across Γ_s ; i.e.,

$$\psi^{E_i}(r, \hat{\Omega}) = 0, \quad \hat{n} \cdot \hat{\Omega} < 0, \quad r \in \Gamma_s, \quad i = 1, \dots, N_l, \quad (3)$$

where \hat{n} is the outward unit normal vector at point r on the surface Γ_s .

The derivation and analysis of neutral-particle transport equations can be found in Lewis and Miller (1993) and Anikonov et al. (2002). It is well known that solving transport equations is very challenging for real applications since there is no analytic solution for describing complex geometrical configurations. Advanced analysis and sophisticated numerical methods are typically utilized to calculate the solutions of transport equations. Information on computational methods for neutral-particle transport can be found in Lewis and Miller (1993).

Since uncollided particles do not change direction or energy, (2) can be solved with ray-tracing techniques (Lewis and Miller 1993; Favorite et al. 2009). The ray-tracing technique transforms a differential equation into an integral equation. Applying numerical integration methods to calculate angular fluxes is not an easy task for complex geometrical configurations (for example, a Monte Carlo integration method is proposed in Anikonov et al. 2002).

The source rate density q^{E_i} in (2) is the emission rate per atom multiplied by the atom density. The emission rate per atom is measured nuclear data and is not a parameter to be optimized in this application, though the atom density may be. The total photon macroscopic cross section $\Sigma_t^{E_i}$ in (2) is the product of the microscopic photon cross section, which is measured nuclear data and not a parameter to be

optimized in this application, and the atom density. Both the source rate density and the macroscopic cross section depend on the material.

We assume a measurement of the uncollided flux of each line at a detector external to the source/shield system. The detector response function for the uncollided scalar flux at a point is

$$\Sigma_d^{E_i}(r, \hat{\Omega}) = \delta(r - r_d), \quad (4)$$

where r_d is a spatial coordinate in \mathbb{R}^3 defining the detector location and δ is the Dirac delta function. Integrating the detector response function $\Sigma_d^{E_i}(r, \hat{\Omega})$ with the uncollided angular flux $\psi^{E_i}(r, \hat{\Omega})$ over the entire phase space (volume V and angle $\hat{\Omega}$), the uncollided scalar flux of energy line E_i at point r_d is

$$\begin{aligned} F^{E_i, r_d} &= \int_{V_d} dV \int_{4\pi} d\hat{\Omega} \Sigma_d^{E_i}(r, \hat{\Omega}) \psi^{E_i}(r, \hat{\Omega}) \\ &= \int_{4\pi} d\hat{\Omega} \psi^{E_i}(r_d, \hat{\Omega}), \quad i = 1, \dots, N_l, \end{aligned} \quad (5)$$

where V_d is the detector volume and 4π means 4π in a solid angle. Note that ψ^{E_i} must satisfy (2)–(3). For brevity, we use F_i instead of F^{E_i, r_d} to denote the uncollided scalar flux at point r_d for line i .

The derivation and explicit form of (5) for a multilayered spherical source/shield system is shown in Favorite et al. (2009). We developed a computational code for calculating uncollided scalar fluxes for multilayered spherical source/shield systems, applying the Gauss-Kronrod quadrature formula as implemented in the QUADPACK library (Piessens et al. 1983). The adaptive quadrature subroutines in QUADPACK automatically adapt based on the variation of integrals and return a flag indicating that an extremely bad integrand behavior occurs at some points of the integration interval.

The system parameters of a multilayered spherical source/shield system include the number of layers, outer radii of layers (interface locations and the outermost radius), source isotopic compositions, shield materials, and material mass densities. To calculate F_i , a complete description of the multilayered source/shield system and detector location must be known. This is the *forward transport calculation*.

Suppose there is a source/shield system where some (or all) system parameters are unknown and uncollided scalar flux measurements are available. We want to use the measured uncollided scalar fluxes to infer the values of the unknown system parameters. This problem is called the *inverse transport problem*. The inverse transport problem may not have a unique solution; in fact, in the usual case in which there are more unknown parameters than measured lines N_l , multiple solutions are guaranteed. Our goal is to identify all solutions of the inverse transport problem, since this information can be used to make better decisions in applications. In practice, other system measurements and expert knowledge may be used to infer some of the unknown system parameters.

Let $F_{m,i}$ be the measured value of the uncollided scalar flux of energy E_i , $\sigma_{m,i}$ be the statistical uncertainty associated with measurement $F_{m,i}$, and x be a vector of unknown system parameters. We call x a *solution* of the inverse transport problem if

$$|F_{m,i} - F_i(x)| \leq \sigma_{m,i}, \quad \text{for all } i = 1, \dots, N_l, \quad (6)$$

where $F_i(x)$ is the reliable calculated value of the uncollided scalar flux of energy E_i using x and other known system parameters, and N_l is now the number of measured data points. The value of $F_i(x)$ is numerically reliable if the numerical integration subroutine used to calculate it terminates normally.

We want to locate multiple solutions of an inverse transport problem and an optimization-based inversion technique is the method of choice. We consider a blackbox optimization of the form

$$\min_{x \in \Theta} f(x) := \sum_{i=1}^{N_l} \left(\frac{F_{m,i} - F_i(x)}{\sigma_{m,i}} \right)^2, \quad (7)$$

where $\Theta \subset \mathbb{R}^n$ is a set of feasible solutions and n is the number of unknown system parameters. The set Θ is problem-dependent and the constraints defining Θ are hard or (impossible) to describe using mathematical notation. The physical constraints imposed on unknown system parameters may include the facts that (i) material densities are positive and bounded, (ii) the sum of source weight fractions is 1.0, (iii) outer radii and layer thicknesses are positive, (iv) the outermost radius is less than the distance from the object's center to the detector, (v) shield materials are chosen from a material library. The values of shield materials are non-numeric, of course, but we can arrange them in the library to have some ordering property. We can then treat shield materials as continuous variables in a set of feasible solutions Θ and use a mapping to transform the shield material values in Θ to those in the material library. Another way to solve inverse problems with unknown shield materials is to use mixed variable optimization (MVO) (Abramson et al. 2009), but this approach is inappropriate for our problem because it requires more prior knowledge (to define a discrete neighborhood) than we are willing to apply. Hidden constraints occur when the numerical subroutine used to calculate the uncollided flux F_i does not terminate normally (i.e., it returns a flag indicating that an extremely bad integrand behavior occurs at some points of the integration interval).

The inverse transport optimization problem (7) is actually a nonlinear parameter estimation problem, which can have a large number of local minima, and a global optimization algorithm is a suitable tool for solving a nonlinear parameter estimation problem (Csendes 1988). Since MLSL is a stochastic global optimization algorithm designed for finding all local minima, we want to use MLSL to locate multiple local minimizers of (7) in the hope that some of these minimizers are also solutions of the inverse transport problem (i.e., minimizers that cause (6) to be satisfied). Our definition of a *solution* of an inverse transport problem makes no use of local or global minimizers of f : Any feasible solution x whose values of calculated scalar fluxes are within the uncertainties of the values of measured scalar fluxes is a solution of the inverse transport problem.

3 MLSL and MADS algorithms

3.1 Multilevel single linkage (MLSL) algorithm

The MLSL method was developed by Boender et al. (1982) for a bound constrained global optimization problem of the form

$$\min_{x \in X} f(x), \quad (8)$$

where $X = \{x = (x_1, \dots, x_n) \mid a_i \leq x_i \leq b_i, i = 1, \dots, n\} \subset \mathbb{R}^n$ and $f : X \rightarrow \mathbb{R}$. It is considered to be one of the most reliable and efficient stochastic global optimization methods. The technical details, software implementations, and performance of the MLSL method can be found in Rinnooy Kan and Timmer (1987a, b), Csendes (1988), Csendes et al. (2008), and Sendin et al. (2009). A summary of the MLSL algorithm is presented in this section.

The MLSL method is a multistart stochastic global optimization algorithm designed to find all local minima by iteratively collecting information about the regions of attractions of the local minimizers found so far. The *region of attraction* of a local minimizer x^* is the set of points in X such that, for any $x \in X$, the local optimization starting at x converges to x^* . The MLSL algorithm relies on a combination of random sampling, clustering, and local search strategies. Each MLSL iteration consists of two phases, sampling and minimization. In the sampling

Table 1 High-level structure of the multilevel single linkage (MLSL) algorithm

A0 INITIALIZE	<ol style="list-style-type: none"> 1. Choose termination conditions for MLSL algorithm and local search solver. 2. Choose $N > 0, 0 < \gamma < 1, \alpha > 0$. 3. $C \leftarrow \emptyset; X^* \leftarrow \emptyset; \tilde{X} \leftarrow \emptyset; k \leftarrow 0$.
A1 GLOBAL PHASE	<ol style="list-style-type: none"> 1. $\bar{C} \leftarrow \emptyset; \tilde{C} \leftarrow \emptyset; k \leftarrow k + 1$. 2. Compute r_k. 3. Draw N points with uniform distribution on X, evaluate objective function values at sample points, and add sample points to C. 4. Choose $\lfloor \gamma k N \rfloor$ points in C with lowest objective function values and place them in \bar{C}. 5. Choose all distinct points in \bar{C} and place them in \tilde{C}. 6. If $X^* \neq \emptyset$, apply the single linkage clustering procedure to points in \tilde{C} that have not been assigned to clusters with seed points chosen from $X^* \cup \tilde{X}$. 7. If every point in \tilde{C} has been assigned to a cluster, then go to A3; else ($\tilde{C} \leftarrow \emptyset; i \leftarrow 0$; go to A2).
A2 LOCAL PHASE	<ol style="list-style-type: none"> 1. Choose $\tilde{x} \in \tilde{C} \setminus \tilde{X}$ and add \tilde{x} to \tilde{C}. 2. If \tilde{x} has not been assigned to a cluster, then <ol style="list-style-type: none"> (a) Apply a local optimization algorithm using \tilde{x} as a start point to find a local minimizer x^*. (b) If $x^* \notin X^*$, then ($\tilde{x} \leftarrow x^*$; add x^* to X^*; $i \leftarrow i + 1$); else ($\tilde{x} \leftarrow \tilde{x}$; add \tilde{x} to \tilde{X}). (c) Apply the single linkage clustering procedure to \tilde{C} using \tilde{x} as a seed point. endif 3. If $\text{size}(\tilde{C}) > \text{size}(\tilde{X})$, go to A2. 4. If $i \neq 1$, go to A3. 5. If any MLSL termination criterion is satisfied, then go to A3; else go to A1.
A3 FINALIZE	<ol style="list-style-type: none"> 1. Order points in X^* according to their objective function values. 2. Stop MLSL.

phase, a finite number of sampled random points are drawn from a uniform distribution over X and the objective function value is evaluated at each of these sample points. In the minimization phase, a minimization procedure is performed from a subset of sample points. Several local search solvers have been used within the MLSL minimization phase.

The high-level structure of the MLSL algorithm is described in Table 1, where the notation $\lfloor a \rfloor$ defines the greatest integer that is less than or equal to a . The local minimizers found are in the set X^* . Any local optimization method may be used in the MLSL algorithm. The single linkage clustering technique, which can produce clusters of any geometric shape (Boender et al. 1982), is applied to reduce the number of local searches. At each iteration k , $\tilde{x} \in \tilde{C}$ is selected as a start point for the local optimization method if it has not been used as a start point at a previous iteration (i.e., $\tilde{x} \notin \hat{X}$), and if there is no point $y \in X^* \cup \tilde{C}$ within the critical distance r_k of \tilde{x} with a lower objective function value (i.e., there is no y such that $\|\tilde{x} - y\| \leq r_k$ and $f(y) < f(\tilde{x})$). The critical distance is given by

$$r_k = \pi^{-1/2} \left[\Gamma \left(1 + \frac{n}{2} \right) \mu(X) \alpha \frac{\log kN}{kN} \right]^{\frac{1}{n}}, \quad (9)$$

where μ is the Lebesgue measure (Royden 1988), Γ is the gamma function, α is a positive constant, and N is the sample size per iteration.

3.2 Mesh adaptive direct search (MADS) algorithm

The MADS algorithms generalize the generalized pattern search (GPS) algorithm (Torczon 1997). Both GPS and MADS are directional-base optimization methods. The MADS method was developed and analyzed in 2006 (Audet and Dennis Jr. 2006; Abramson and Audet 2006) for general constrained optimization problems of the form

$$\min_{x \in \mathbb{S}} f(x), \quad (10)$$

where $\mathbb{S} = \{x \in X \mid g_j(x) \leq 0, j \in J\} \subset \mathbb{R}^n$, $f, g_j : X \rightarrow \mathbb{R} \cup \{\infty\}$ for all $j \in J = \{1, 2, \dots, N_c\}$, and $X \subset \mathbb{R}^n$. No differentiability assumptions on either the objective function f or the constraints g_j are required for the MADS method. The domain X represents unrelaxable constraints (i.e., constraints that necessarily need to be satisfied in order for the functions to be evaluated). The unrelaxable constraints often include bounds constraints $l \leq x \leq u$ with l and u in $(\mathbb{R} \cup \{\pm\infty\})^n$ and Boolean constraints that indicate if they are satisfied or not. The functions g_j represent the other constraints and are referred to as relaxable constraints. The objective function f and the constraints g_j 's defining the feasible domain \mathbb{S} are usually provided as blackboxes in the sense that the way to obtain a function value from a given $x \in \mathbb{R}^n$ is not provided in an analytical way, or it may be time-consuming or expensive to evaluate. The blackboxes may also fail to return function values at some feasible points, and these points are referred to as hidden constraint points. Blackbox functions are typically evaluated by running computational codes

or simulations where advanced numerical methods are applied to solve complex problems. We summarize the principle behind the MADS algorithms in this section. The technical details and software implementation of MADS can be found in Le Digabel (2011) and references therein.

The MADS method can handle constraints by three different strategies: the extreme barrier (EB), the progressive barrier (PB), and the progressive-to-extreme barrier (PEB) (Audet et al. 2010). In the EB approach, infeasible trial points and feasible points that fail hidden constraints are rejected by setting their objective function value to infinity. In the PB approach, a constraint violation function and a threshold which is progressively reduced are introduced, and trial points whose constraint violation values exceed the threshold are rejected from consideration. The PEB approach is a hybrid of the PB and EB approaches.

MADS is an iterative method that attempts to locate a minimum over the feasible region by directly making comparisons of the objective function value at some trial points lying on a mesh over the domain space. At each iteration, the algorithm generates a finite number of trial points on the mesh, evaluates the functions at some (or all) generated trial points with an attempt to find a point with a lower objective function value than the current best iterate point found so far, and then adapts the fineness of the mesh to approach an optimum. The high-level structure of the MADS algorithm is given in Table 2.

Each iteration k of the MADS algorithm is characterized by the SEARCH and POLL steps where trial points are generated and functions are evaluated. These trial points must lie on the mesh M_k defined by

$$M_k = \bigcup_{x \in V_k} \{x + \Delta_k^m \Phi z \mid z \in \mathbb{N}^{n_\Phi}\}, \tag{11}$$

where $\Delta_k^m > 0$ is the mesh size parameter, $V_k \subset \mathbb{R}^n$ is the set of all previously evaluated trial points at iteration k , V_0 is the set of initial trial points, and Φ is an $n \times n_\Phi$ matrix representing a fixed finite set of n_Φ directions in \mathbb{R}^n .

Table 2 High-level structure of the mesh adaptive direct search (MADS) algorithm

B0 INITIALIZE	<ol style="list-style-type: none"> 1. Choose $\Delta_0^m > 0, \Delta_0^p > 0$, and termination condition(s). 2. Choose $x_0 \in X$ and evaluate the functions at x_0. 3. $k \leftarrow 0$.
B1 SEARCH	<ol style="list-style-type: none"> 1. If the SEARCH step is empty, go to B2. 2. Evaluate the functions at a finite number of points of M_k. 3. If a point with a lower objective function value than that of x_k is found, then go to B3; else go to B2.
B2 POLL	<ol style="list-style-type: none"> 1. Evaluate the functions on P_k until a point with a lower objective function value than that of x_k is found or all points of P_k are evaluated. 2. Go to B3.
B3 UPDATE	<ol style="list-style-type: none"> 1. If a point \hat{x} was found in B1 or B2 such that $f(\hat{x}) < f(x_k)$, then $x_{k+1} \leftarrow \hat{x}$; else $x_{k+1} \leftarrow x_k$. 2. If any termination condition is satisfied, stop MADS. 3. Update Δ_{k+1}^m and Δ_{k+1}^p. 4. $k \leftarrow k + 1$; go to B1.

The columns of matrix Φ constitute the set of mesh directions. Φ is constructed so that $\Phi = HZ$, where H is a nonsingular $n \times n$ matrix and Z is an $n \times n_\Phi$ integer matrix. Since M_k is defined to be the union of sets over V_k , all previously evaluated trial points before iteration k lie on M_k and all new trial points can be constructed around these previously evaluated points using the directions in Φ .

The SEARCH step is optional. If trial points are not generated and functions are not evaluated in the SEARCH step, it is referred to as an empty SEARCH step. In the SEARCH step, function evaluations are allowed at any trial point on the mesh and any algorithm can be used. The SEARCH step can be used to globally explore the feasible region. A few algorithms have been incorporated in the SEARCH step, such as a particle swarm search algorithm with GPS (Vaz and Vicente 2007) and a variable neighborhood search (VNS) algorithm with MADS (Audet et al. 2008).

The POLL step evaluates the objective function at local trial points in a poll trial set defined by

$$P_k = \{x_k + \Delta_k^m \phi \mid \phi \in \Phi_k\} \subset M_k, \quad (12)$$

where x_k and Φ_k are the poll center and the set of poll directions at iteration k , respectively. The poll size parameter Δ_k^p is introduced in the POLL step and points of P_k must be generated so that their distances to the poll center x_k are bounded below by Δ_k^p . The POLL step includes evaluating the constraint functions as well.

The MADS method constructs the set of directions Φ_k in the POLL step. Φ_k must be a positive spanning set and trial points generated by Φ_k must lie on the mesh M_k . Two approaches for generating Φ_k are LTMADS and ORTHOMADS. LTMADS uses a random lower triangular matrix to generate directions (Audet and Dennis Jr 2006), while ORTHOMADS uses a Householder matrix and Halton sequences to deterministically generate orthogonal directions (Abramson et al. 2009). The convergence properties of LTMADS and ORTHOMADS are proved in Audet and Dennis Jr (2006) and Abramson et al. (2009), respectively.

When a trial point with a lower objective function value is found or all generated trial points are evaluated, the next poll center x_{k+1} will be selected and the mesh and poll size parameters will be updated. If a trial point with a lower objective function value is found at iteration k , the mesh size parameter at iteration $k + 1$ must be greater than or equal to the one at iteration k . Otherwise, it must be reduced. The poll size parameter must also be updated in this step (using rules appropriate to the algorithm used to generate Φ_k). The formula used to update the mesh and poll sizes can be found in Audet and Dennis Jr. (2006) and Abramson et al. (2009).

A software package called Nonlinear Optimization by Mesh Adaptive Direct Search (NOMAD) is a C++ implementation of MADS algorithms (Le Digabel 2011; Abramson et al. 2014). Three algorithms implemented in NOMAD to generate the directions are ORTHOMADS, LTMADS, and GPS. There is another MADS instance called QRMADS which uses an equal-area partition of the unit sphere and QR decomposition to generate the directions (Van Dyke and Asaki 2015), but QRMADS is not currently implemented in NOMAD.

4 A new variant of the multilevel single linkage algorithm

We propose to combine the MADS algorithm with the MLSL method. This new combined algorithm is designed to locate multiple local minima of blackbox optimization problems. We consider an optimization problem of the form

$$\min_{x \in \Theta} f(x), \quad (13)$$

where $\Theta \subset \mathbb{R}^n$, $f: \Theta \rightarrow \mathbb{R} \cup \{\infty\}$, the functions defining f and Θ are computed by computational codes that may return invalid outputs even when all inputs are valid, and the constraints defining Θ may be evaluated before or after running the computational codes.

Each MLSL iteration consists of two phases, global and local. The sample points are drawn from a uniform distribution over the feasible space and the function evaluation is performed at sample points in the global phase, and the local search solver is applied from some of these sample points in the local phase. Only some of the sample points are used as the start points for the local search procedure because it is costly to perform the local searches from all sample points. The strategy used to reduce the number of local searches is that a subset of sample points with the lowest objective function values are selected and the single linkage clustering procedure is then applied to cluster these sample points so that only one local search procedure initializes from each cluster.

Since MADS algorithms are designed for blackbox optimization, we want to utilize MADS algorithms with empty SEARCH steps in the MLSL local phase. Any MADS algorithm (such as LTMADS or ORTHOMADS) used to generate the directions in the POLL step of the MADS method can be used in the MLSL local phase, and all strategies introduced in MADS to handle constraints are applicable in this MLSL local phase. All strategies used by MADS can be used in the MLSL global phase since only objective function values are used in the MLSL global phase. Any sample point whose objective function value is finite can be used as a candidate of a start point of a local optimization method. Unlike in the MLSL local phase, infeasible points that satisfy the PB constraints can be used without switching to EB, because these points may be used as starting points of the MADS algorithm in the local phase. The choice of MADS algorithms (LTMADS or ORTHOMADS) and constraints handling techniques used in the combined MLSL–MADS algorithm depends on applications. For example, if no information is available to help handle infeasible points, then the EB approach is necessary; if uniform searching in the local phase is desired, then ORTHOMADS should be applied in the local phase.

For blackbox optimization problems, objective function values are calculated by a computational code and hidden constraint points may exist. Traditionally, the optimization procedure is paused to provide information about hidden constraint points so that the algorithmic parameters of the numerical subroutines used to calculate the objective function values may be tuned to work for such points. Since tuning algorithmic parameters to work for all hidden constraint points is not an easy task, we deal with these hidden constraint points by setting their objective function

values to infinity. The feasible points that fail hidden constraints are not used as start points for the MADS algorithm integrated into the MLSL method.

The new combined MLSL–MADS algorithm is presented in Table 3 (the notation “ $\tilde{C} \setminus \hat{C}$ ” is for $\{x \mid x \in \tilde{C} \text{ and } x \notin \hat{C}\}$). The critical distance used in this combined MLSL–MADS algorithm is

$$r_k = \pi^{-1/2} \left[\Gamma \left(1 + \frac{n}{2} \right) \mu(\Theta) \left(1 - \alpha^{1/(N_k-1)} \right) \right]^{\frac{1}{n}}, \quad (14)$$

where μ is the Lebesgue measure, Γ is the gamma function, α is constant in $(0, 1)$, and N_k is the sample size at iteration k . The critical distance r_k was introduced in Boender et al. (1982) and its theoretical properties were proved in Rinnooy Kan and Timmer (1987a). Based on the theoretical results, using r_k defined in (14) within the MLSL method results in the least cost in terms of the number of local searches.

Table 3 High-level structure of the modified multilevel single linkage (MLSL–MADS) algorithm

C0 INITIALIZE

1. Choose termination conditions for MLSL algorithm and MADS algorithm.
2. Choose $\tilde{N} > 0$, $\hat{N} > 0$, $0 < \gamma < 1$, $\alpha > 0$.
3. $C \leftarrow \emptyset$; $X^* \leftarrow \emptyset$; $\tilde{X} \leftarrow \emptyset$; $N_0 \leftarrow 0$; $k \leftarrow 0$.

C1 GLOBAL PHASE

C1-0 Setting

- (a) $\tilde{C} \leftarrow \emptyset$; $k \leftarrow k + 1$; $j \leftarrow 0$.
- (b) go to **C1-1**.

C1-1 Sampling

- (a) $\tilde{C} \leftarrow \emptyset$; $j \leftarrow j + 1$.
- (b) Draw \tilde{N} points with uniform distribution on Θ , evaluate objective function values at sample points, and add sample points to C .
- (c) Choose $\lfloor \gamma k \tilde{N} \rfloor$ points in C with lowest objective function values and place them in \tilde{C} .
- (d) If all points in \tilde{C} have finite objective function values, then go **C1-2**.
- (e) If $j < \tilde{N}$, then go to **C1-1**; else go to **C1-2**.

C1-2 Processing

- (a) Remove all points whose objective function values are infinity from \tilde{C} .
- (b) If $\tilde{C} = \emptyset$, go to **C3**.
- (c) Choose all distinct points in \tilde{C} and place them in \tilde{C} .
- (d) $N_k \leftarrow N_{k-1} + j \tilde{N}$; compute r_k .
- (e) If $X^* \neq \emptyset$, apply the single linkage clustering procedure to points in \tilde{C} that have not been assigned to clusters with seed points chosen from $X^* \cup \tilde{X}$.
- (f) If every point in \tilde{C} has been assigned to a cluster, then go to **C3**; else ($\hat{C} \leftarrow \emptyset$; $i \leftarrow 0$; go to **C2**).

C2 LOCAL PHASE

1. Choose $\tilde{x} \in \tilde{C} \setminus \hat{C}$ and add \tilde{x} to \hat{C} .
2. If \tilde{x} has not been assigned to a cluster, then
 - (a) Apply a MADS algorithm with empty SEARCH step using \tilde{x} as a start point to find a local minimizer x^* .
 - (b) If $x^* \notin X^*$, then ($\hat{x} \leftarrow x^*$; add x^* to X^* ; $i \leftarrow 1$); else ($\hat{x} \leftarrow \tilde{x}$; add \tilde{x} to \hat{X}).
 - (c) Apply the single linkage clustering procedure to \tilde{C} using \tilde{x} as a seed point.
- endif
3. If $\text{size}(\tilde{C}) > \text{size}(\hat{C})$, go to **C2**.
4. If $i \neq 1$, go to **C3**.
5. If any MLSL termination criterion is satisfied, then go to **C3**; else go to **C1**.

C3 FINALIZE

1. If $X^* \neq \emptyset$, order points in X^* according to their objective function values.
 2. Stop MLSL–MADS.
-

The stopping criteria of the combined MLSL–MADS can be set using different information. The MLSL stopping criteria may include a maximum number of found minimizers (size of X^*), a maximum number of iterations k , and/or a maximum number of random sampling points (size of C). Any MADS stopping criterion that is available in NOMAD can be used in the MLSL–MADS algorithm. The stopping criteria of MADS may include a maximum number of iterations, mesh or poll size tolerances, and/or a convergence criterion.

5 On multiple solutions and different families of solutions

We expect that inverse transport problems of the type that we discuss in this paper will have many unknown parameters and comparatively fewer measured data points. Even for the case of more data points than unknown parameters, we expect that many different system models could produce the same measurement. In other words, we do not expect the inverse transport problem to have a unique solution. Our goal in this work is to find all possible solutions. However, we wish to distinguish between solutions that are related in some parametric way and those that are not. For example, consider the simple case in which the mass and density of a homogeneous cylinder are known, but the height and radius are to be found. There are an infinite number of solutions to this inverse problem, but they are all related in a parametric way. We call such solutions *equivalent*. However, if the shape of the object is not known, then we may also seek solutions in cuboids or spheres. There are an infinite number of solutions in the family of cuboids, but only one (or fewer) solution in the family of spheres. We use *family* to distinguish among solutions that are not related in an obvious parametric way. We are seeking all equivalent solutions in all possible families. Presently, our implementation of MLSL–MADS generally requires the user to manually initiate searches in different families (spheres or cylinders, for example), but, as shown in Sect. 6, there are problems for which solutions in different families can be found automatically.

6 Numerical test cases

Two numerical test cases illustrate the performance of the combined MLSL–MADS algorithm for solving inverse transport problems. We consider the spherical test geometry shown in Fig. 1. It is a solid spherical high-enriched uranium (HEU) gamma-ray source containing 94.73 % ^{235}U and 5.27 % ^{238}U (by weight); the HEU sphere has a radius of 8.741 cm. This spherical source is surrounded by a spherical shield that is separated from the source by a void region. The shield is made of layers of pure lead and pure aluminum. The lead layer has an inner radius of 12.4 cm and a thickness of 0.5 cm, and the aluminum layer has an inner radius of 12.9 cm and a thickness of 0.3 cm. The HEU, lead, and aluminum mass densities are 18.74, 11.4, and 2.7 g/cm³, respectively.

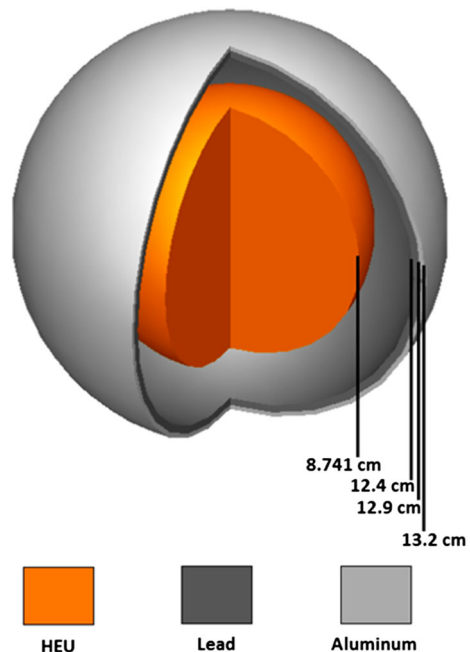
We use the uncollided flux of the 766-keV gamma-ray line emitted from ^{238}U as the quantity of interest. The emission rate of this line is $1.525 \times 10^4 \text{ } \gamma/\text{s}/(\text{mole } ^{238}\text{U})$. Using the material parameters of the HEU given above, the source rate density q^{E_i} of the 766-keV line in HEU is $38.1 \text{ } \gamma/\text{s}/\text{cm}^3$. (Actually, the 766-keV line comes from $^{234\text{m}}\text{Pa}$, a daughter of ^{238}U ; the two are almost always in secular equilibrium, and this line is often used for uranium diagnostics (PANDA 2007).)

The detector is modeled as a point located 100 cm from the center of the radiating sphere. Multiplying the uncollided flux measured at this detector by $4\pi(100 \text{ cm})^2$ gives an estimate of the total leakage rate of the 766-keV line. Using the ray-tracing method of Favorite et al. (2009) with the 15-point Gauss-Kronrod adaptive quadrature formula implemented in QUADPACK (Piessens et al. 1983), the uncollided flux of the 766-keV line at the detector point is $1.99642 \times 10^{-2} \text{ } \gamma/\text{s}/\text{cm}^2$ and the total estimated leakage rate is $2.50877 \times 10^3 \text{ } \gamma/\text{s}$. We used this value as the reference $F_{m,i}$ in (7) and assumed a relative uncertainty $\sigma_{m,i}/F_{m,i}$ of 0.01 %.

The uncollided fluxes used during the MLSL–MADS iterations are also calculated by the ray-tracing method, and the algorithmic parameters of the QUADPACK subroutine are the same ones used in calculating the reference values. We input enough digits from the reference result into the MLSL–MADS algorithm such that the objective function value $f(x)$ of (7) is zero for the model shown in Fig. 1.

We consider two test cases. The unknown parameters of case I are the HEU source radius, the radius of the inner surface of the lead layer, and the mass density

Fig. 1 Test geometry
(rendering provided by K.
C. Bledsoe, Oak Ridge National
Laboratory)



of the HEU source. The unknown parameters of case II are the radius of the interface between the two regions of the shield (lead and aluminum in Fig. 1) and the material contained in those two regions of the shield. With three unknown system parameters and one data point, these problems are underdetermined, and multiple solutions are expected.

Typically, information and knowledge related to an optimization problem should be used to define the correct feasible region, and information associated with the feasible region may be used by the optimization algorithm to locate trial points. In applications, other measurements and/or information may be available to help define the correct feasible region, but we do not use measurements or information to define the feasible region that is related to the true test geometry in this paper. We thus impose only a few constraints on the unknown system parameters in the test problems. Each layer's thickness is at least 1.0×10^{-6} cm, and an unknown interface location (radius) is not allowed to cross a neighboring known interface location. The mass densities are in $[0.0001, 25.0]$ g/cm³. Unknown shield materials are constrained to be one of those present in a library of 119 materials that we developed for this purpose. The library defines each material's composition (isotopes and weight fractions), mass density, and macroscopic photon cross section $\Sigma_i^{E_i}$. The materials in the library are ordered by mass density.

We implemented the MLSL–MADS algorithm using a mixed Fortran/C++ programming paradigm where the MLSL algorithm and the subroutines for computing the objective function values are written in Fortran and the algorithms implemented in the NOMAD software package (Abramson et al. 2014), written in C++, are used. The MLSL algorithm in our implementation is modified from the GLOBALm software package (Csendes et al. 2008). The stopping criteria for the global phase are $k > 100$, $size(X^*) > 100$ or $size(\hat{X}) > 100$ and the stopping criteria for the local phase are that the maximum number of iterations is 50,000 or the convergence of objective function values is within 1.0×10^{-6} . We used the EB approach to handle constraints (including hidden constraints).

We ran each test case using the combined MADS–MLSL algorithms for 20 independent runs (i.e., using a different random number seed for each run).

For case I, each run of the MLSL–MADS algorithm found 94–96 local minimizers. Every local minimizer was also a solution in the sense that we defined *solution* in (6) of Sect. 2: The calculated line leakage was within one standard deviation of the reference line leakage. All local minimizers from all 20 runs are plotted in Fig. 2, where “r1”, “r2”, and “d1” represent the HEU source radius, the radius of the inner surface of the lead layer, and the mass density of the HEU source, respectively. The surface shown in Fig. 2 is the space of all local minimizers of this transport problem. It is also the space of all solutions. Because these solutions all lie on a surface—i.e., the three unknown parameters are always related by some parameter—we say that they are all *equivalent* solutions rather than multiple solutions (as discussed in Sect. 5).

The blue point on Fig. 2 is the exact solution shown in Fig. 1, which is also the global minimizer. None of the 20 runs of the optimizer found the global minimum. However, we are not merely seeking the global minimum but rather all possible

solutions. A method that finds the global minimum but ignores the local minimizers will not report the huge space of viable solutions. In other words, for our purposes, finding the surface shown in Fig. 2 is much more important than finding the single global minimum.

Also shown for comparison in Fig. 2 are the results of 20 independent runs of a differential evolution implementation (Bledsoe et al. 2011a), shown in green. The differential evolution algorithm reports a single minimizer with each run. None of the 20 runs found the global minimum, but they all found points on the surface. Each of the 20 runs of the differential evolution algorithm took an average of 4.6 s to run, while each of the 20 runs of the MLSL–MADS algorithm took an average of 44.5 s to run. Thus, the MLSL–MADS algorithm took approximately 0.46 s to find each solution, while the differential evolution algorithm took approximately 4.6 s to find each solution. This timing comparison was for sequential implementations of the algorithms and can change if either algorithm is implemented differently, but it shows the efficiency of the MLSL–MADS algorithm.

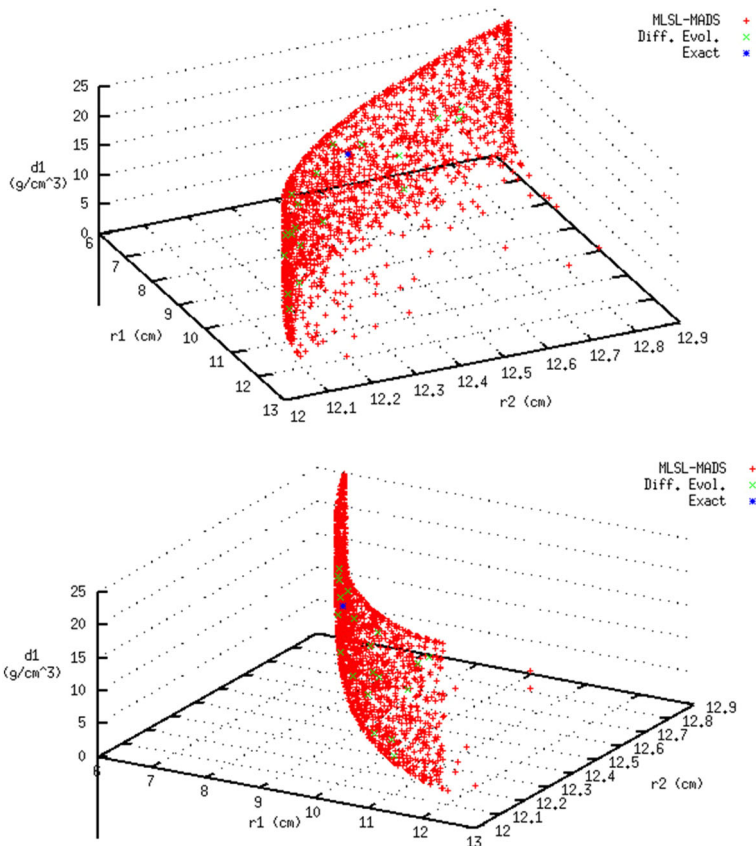


Fig. 2 Two views of the surface formed of local minimizers for case I

For case II, each run of the MLSL–MADS algorithm found 90–95 local minimizers, except that one run found only 31. As for case I, every local minimizer was also a solution in the sense that we defined *solution* in Sect. 2 with (6). All local minimizers from all 20 runs are plotted in Fig. 3, where “xs2”, “xs3”, and “r3” represent the macroscopic photon cross section for the inner shield layer, the macroscopic photon cross section for the outer shield layer, and the radius of the interface between the shield layers, respectively. The macroscopic cross sections were collected after the optimization and are plotted for convenience; it is the shield materials, with their unknown composition and density, that are the unknown parameters. Four of the 20 runs found the global minimum, indicated by the blue point on Fig. 3. However, as for case I, it is not merely the global minimum that we seek, but the entire space of solutions.

For this problem, the MLSL–MADS algorithm found two different families of solutions, each indicated by a surface on Fig. 3. One family of solutions has materials with cross section greater than 0.708 cm^{-1} (corresponding to pure silver at

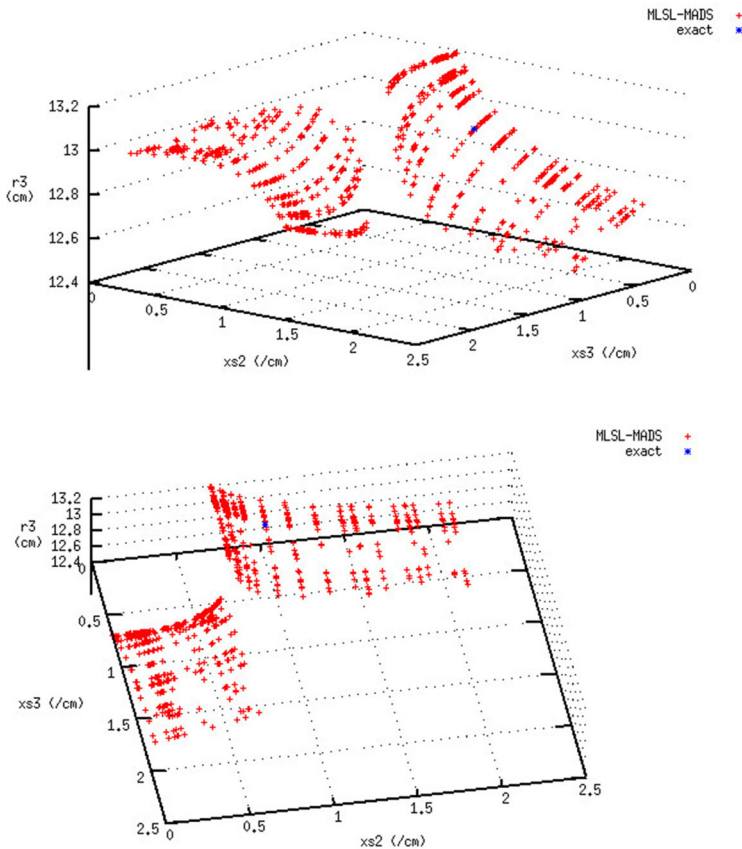


Fig. 3 Two views of the surface formed of local minimizers for case II

a mass density of 10.5 g/cm^3) as the inner shield, and the other has materials with cross section greater than 0.708 cm^{-1} as the outer shield. As discussed in Sect. 5, we call these multiple *families* of solutions because there is no clear parameterization that connects the two surfaces. Each surface is composed of “stripes.” The macroscopic photon cross section for lead at 766 keV is 1.002 cm^{-1} . The stripe containing the exact solution in Fig. 3 ($xs2 \approx 1 \text{ cm}^{-1}$) shows all solutions with lead as the inner shield material and different values of the interface location for different values of the outer shield material. There is another stripe on the other surface ($xs3 \approx 1 \text{ cm}^{-1}$) with lead as the outer shield material and different values of the interface location for different values of the inner shield material. When pure iron at a mass density of 7.896 g/cm^3 is the inner shield material and lead is the outer shield material, the interface location is 12.909 cm, only 0.009 cm greater than the true interface location. For our purposes, this solution is just as valid as the exact solution or global minimum. (Our implementation of the differential evolution algorithm was not run on this problem because it treats shield materials differently, so the comparison with MLSL–MADS would not be straightforward).

We ran these problems using both LTMADS and ORTHOMADS algorithms. For case I, there are very small (round-off sized) differences in most of the solutions found when using LTMADS versus ORTHOMADS, but essentially the same set of solutions were found. The plots (Fig. 2) look exactly the same. For case II, there are very small differences in a few of the solutions found when using LTMADS versus ORTHOMADS, but essentially the same set of solutions were found. The plots (Fig. 3) look exactly the same.

The ability to analyze MLSL–MADS results automatically, rather than visually, would help enormously in realistic applications.

7 Conclusions

The inverse transport problem is a constrained nonlinear parameter estimation problem. It is known that this problem often has a large number of local minima, and the MLSL method is a reliable and efficient method designed for finding all local minima that are potentially global. The inverse transport optimization problem considered in this paper is a blackbox optimization problem in the sense that the objective function value is computed by a computer code in which advanced numerical methods are applied and hidden constraint points may be present. Since MADS is a DFO method designed for blackbox optimization with general constraints, we integrated the MADS algorithm into the local phase of the MLSL method where the techniques used to handle constraints in the local phase can be extended to the global phase. The new combined algorithm is implemented and applied to solve inverse transport optimization problems of a multilayered spherical system where the unknown system parameters are mixed types (interface locations, shield materials, and mass densities).

Numerical results show that the combined MLSL–MADS algorithm locates multiple local minimizers that are solutions of inverse transport problems (using our definition of *solution*). We can determine whether or not the solutions are drawn

from different families by visually examining surface plots, and we are seeking computational methods for determining whether the solutions represent different families or the same family.

Since the MLSL–MADS algorithm is designed for blackbox optimization problems, our future work is to apply this algorithm to solve other related blackbox optimization problems in particle transport applications.

Acknowledgments The authors would like to acknowledge Sakura Kawano (University of Wisconsin) for her help in analyzing the test cases.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Abramson MA, Audet C (2006) Convergence of mesh adaptive direct search to second-order stationary points. *SIAM J Optim* 17(2):606–619
- Abramson MA, Audet C, Chrissis JW (2009) Mesh adaptive direct search for mixed variable optimization. *Optim Lett* 3(1):35–47
- Abramson MA, Audet C, Couture G, Dennis JE Jr., Le Digabel S, Tribes C (2014) The NOMAD project. Software available at <http://www.gerad.ca/nomad>
- Abramson MA, Audet C, Dennis JE Jr, Le Digabel S (2009) ORTHOMADS: a deterministic MADS instance with orthogonal directions. *SIAM J Optim* 20(2):948–966
- Anikonov DS, Kovtanyuk A, Prokhorov IV (2002) Transport equation and tomography. Inverse and Ill-posed problems. VSP, Zeist
- Armstrong JC, Favorite JA (2012) Identification of unknown interface locations in a source/shield system using the mesh adaptive direct search method. *Trans Am Nucl Soc* 106:375–377
- Audet C (2014) A survey on direct search methods for blackbox optimization and their applications, chap. 2. Mathematics without boundaries: surveys in interdisciplinary research. Springer, New York
- Audet C, Bechard V, Le Digabel S (2008) Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. *J Glob Optim* 41(2):299–318
- Audet C, Dennis JE Jr (2006) Mesh adaptive direct search algorithms for constrained optimization. *SIAM J Optim* 17(1):188–217
- Audet C, Dennis JE Jr, Le Digabel S (2010) Globalization strategies for mesh adaptive direct search. *Comput Optim Appl* 46(2):193–215
- Bledsoe KC, Favorite JA, Aldemir T (2011) Applications of the differential evolution methods to solve inverse transport problems. *Nucl Sci Eng* 169(2):208–221
- Bledsoe KC, Favorite JA, Aldemir T (2011) Using the Levenberg–Marquardt method for solutions of inverse transport problems in one-and-two dimensional geometries. *Nucl Technol* 176(1):106–126
- Boender CGE, Rinnooy Kan AHG, Stougie L, Timmer GT (1982) A stochastic method for global optimization. *Math Program* 22:125–140
- Conn AR, Scheinberg K, Vicente LN (2009) Introduction to derivative-free optimization. *SIAM Series on Optimization*, SIAM, Philadelphia
- Csendes T (1988) Nonlinear parameter estimation by global optimization—efficiency and reliability. *Acta Cybern* 8:361–370
- Csendes T, Pal L, Sendin JH (2008) The global optimization method revisited. *Optim Lett* 2:445–454
- Dixon LC, Szego GP (Eds.) (1975, 1978) Towards global optimization, vol. 1, 2. North-Holland, Amsterdam, Netherlands
- Favorite JA, Bledsoe KC, Ketcheson DI (2009) Surface and volume integrals of uncollided adjoint fluxes and forward-adjoint flux products. *Nucl Sci Eng* 163:73–84

- Le Digabel S (2011) Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm. *ACM Trans Math Softw* 37(4):1–15
- Lewis EE, Miller WF (1993) *Computational methods of neutron transport*. American Nuclear Society, Inc., La Grange Park
- Mattingly J, Mitchell D (2010) A framework for the solution of inverse radiation transport problems. *IEEE Trans Nucl Sci* 57(6):3734–3743
- Neumaier A (2004) *Complete search in continuous global optimization and constraint satisfaction*, chap. 4. Cambridge University Press, Cambridge
- Piessens R, Doncker-Kapenger E, Ueberhuber C, Kahaner D (1983) *QUADPACK*, a subroutine package for automatic integration. Springer Verlag, New York
- Kan Rinnooy AHG, Timmer GT (1987) Stochastic global optimization methods, part I: clustering methods. *Math Program* 39:27–56
- Kan Rinnooy AHG, Timmer GT (1987) Stochastic global optimization methods, part II: Multi level methods. *Math Program* 39:57–78
- Passive nondestructive assay of nuclear materials—PANDA (2007) Los Alamos National Laboratory. <http://www.lanl.gov/orgs/n/n1/panda/>
- Rios LM, Sahinidis NV (2013) Derivative-free optimization: a review of algorithms and comparison of software implementations. *J Glob Optim* 56:1247–1293
- Royden HL (1988) *Real analysis*, 3rd edn. Macmillan Publishing, New York
- Sendin JOH, Banga JR, Csendes T (2009) Extension of a multistart clustering algorithm for constrained global optimization problems. *Ind Eng Chem Res* 48:3014–3023
- Stephens CP, Baritomba W (1998) Global optimization requires global information. *J Optim Theory Appl* 96(3):575–588
- Torczon V (1997) On the convergence of pattern search algorithms. *SIAM J Optim* 7(1):1–25
- Van Dyke B, Asaki TJ (2015) Using QR decomposition to obtain a new instance of mesh adaptive direct search with uniformly distributed polling directions. *J Optim Theory Appl* 159:805–821
- Vaz AIF, Vicente LN (2007) A particle swarm pattern search method for bound constrained global optimization. *J Glob Optim* 39:197–219