JOURNAL OF INTEGRAL EQUATIONS
AND APPLICATIONS
Volume 19, Number 3, Fall 2007

AN EVOLUTIONARY NEWTON METHOD FOR SHAPE RECONSTRUCTION

HARRY ECKEL AND ROLAND POTTHAST

ABSTRACT. We consider the problem to reconstruct the location and shape of an *unknown number of sound-soft obstacles* from the far field pattern of scattered acoustic waves. First, the *point source method* is used to generate an estimate for the area where the obstacles are located. In its simplest form the point source method will provide the convex hull of the set of obstacles. Then, we investigate a novel evolutionary Newton algorithm which integrates the Newton scheme with elements of an evolutionary approach. The *Newton method* is an efficient local method to find obstacles if the number, approximate location and shape of all obstacles is known. Without this knowledge the method quickly runs into local minima or diverges. The *evolutionary algorithm* is capable of finding the number of obstacles, their location and shape. However, in its typical form the algorithm is not appropriate for inverse scattering problems due to the expensive evaluation of the forward scattering map. Evolutionary algorithms usually use a large number of target function evaluations on a population of solutions and converge slowly. We will employ principles of both algorithms to formulate a novel *evolutionary Newton scheme* which does combine the strength of both evolutionary methods and the Newton scheme and does avoid their particular limitations. This shows that a combination of statistical and deterministical reconstruction methods can be used to significantly extend the range of the algorithms of both areas. In our last part we illustrate the feasibility of the scheme by numerical examples.

1. Introduction. Shape reconstruction problems are important basic problems for many applied sciences, for example in nondestructive testing, medical imaging and geophysical exploration. The mathematical area of inverse scattering problems investigates the reconstruction of the location and shape of objects from the knowledge of the Cauchy values or far field pattern, respectively, of scattered acoustic or electromagnetic waves. For a description of the state-of-the-art of inverse scattering theory we refer to [**1**] and the survey article [**16**].

Received by the editors on November 19, 2006, and in revised form on May 23, 2007.

Evolutionary methods are a popular branch of optimization methods. They are part of the statistical optimization approach and employ the principles of evolution for finding the minimum of some function depending on several variables. As a basic difference of evolutionary methods in contrast to numerical iteration schemes evolutionary methods

• do not only consider one iterate or a sequence of iterates to generate a new approximation to the true solution, but they employ a whole collection of approximate solutions, called a *population*,

• they create a new and 'better' population (called child population) from a previous population (called parent population) using stochastical mutation, splitting and recombination steps. Mutation steps change the parameters of the parent population. Splitting leads to the creation of new individuals with different properties. Recombination combines different features of different individuals into a new individual imitating inheritance observed in biological parentship.

For inverse scattering problems evolutionary methods have not been widely used due to *two crucial disadvantages* of the evolutionary approach. First, since it works with many objects, it needs to solve a large number of forward problems for inversion. If the forward map is time consuming, this leads to large reconstruction times. Second, since the transition from one population to another is driven by 'blind' mutation, the properties of the forward problem (like gradients, derivatives etc.) are not taken into account. This leads to slow convergence and, together with the first disadvantage, to even more time consuming algorithms.

On the other hand, in the deterministic theory of inverse scattering rather efficient iterative algorithms have been developed in recent years. The Newton scheme employs superlinear convergence, compare [**14, 15**]. But usually a good initial guess of the shape is necessary for convergence, and its use is limited to the case where we know the number of unknown domains. To work around this problem, the level set method can be used, see for example, [**3**], which defines a function in the whole space such that the objects are described as level sets of this function. It has the key feature that it is able to split and create domains.

Here, we choose an alternative route. We use the evolutionary approach to introduce splitting and merging of domains and combine it with the classical Newton method to enhance its convergence properties. To this end, we first describe the direct and inverse problem under consideration in Section 2. Then, we use Sections 3 and 4 to describe the evolutionary approach and the classical Newton method. A new evolutionary Newton method will be formulated in Section 5. Finally, numerical examples are given in Section 6.

2. Shape reconstruction in inverse obstacle scattering problems. We will consider inverse scattering by an impenetrable obstacle. The following boundary value problem will serve as a generic problem for the methods under consideration. We will restrict our presentation to scattering of acoustic waves.

We consider scattering of some incident wave u^i by some inhomogeneity $\mathcal{D}$ with support $D \subset \mathbf{R}^m$, $m = 2, 3$. The incident field u^i, which is assumed to satisfy the Helmholtz equation in some set containing the inhomogeneity, generates a scattered field u^s. The total field $u^i + u^s$ is denoted by u. We will assume that the scattering region D is bounded and has boundary of class C^2 such that $\mathbf{R}^m \setminus \overline{D}$ is connected.

Dirichlet scattering problem. Assume that the total field u solves the Helmholtz equation

$$\triangle u + \kappa^2 u = 0 \tag{2.1}$$

with wave number κ (where $\operatorname{Im} \kappa \geq 0$) in the exterior $\mathbf{R}^m \setminus \overline{D}$, the scattered field u^s satisfies the *Sommerfeld radiation condition*

$$r^{(m-1)/2}\left(\frac{\partial}{\partial r} - i\kappa\right)u^s(x) \longrightarrow 0, \quad r = |x| \to \infty \tag{2.2}$$

uniformly for all directions $\hat{x} = x/|x|$ and on the boundary ∂D of D we have the *Dirichlet boundary condition*

$$u|_{\partial D} = 0. \tag{2.3}$$

Then, we say that u solves the *Dirichlet scattering problem*. The solution to the Dirichlet scattering problem in two or three dimensions is well known and can be found for example in the monograph of Colton and Kress [**1**].

The scattered field u^s asymptotically behaves like some outgoing spherical wave multiplied by a factor which is usually known as *far field pattern* or *scattering amplitude*. In particular, we have

$$u^s(x) = \frac{e^{i\kappa|x|}}{|x|^{(m-1)/2}} \left\{ u^\infty(\hat{x}) + O\left(\frac{1}{|x|}\right) \right\}, \tag{2.4}$$

for $|x| \to \infty$ uniformly for all directions $\hat{x} = x/|x|$. We denote the mapping which maps the boundary values u^i of the incident field onto the far field pattern u^∞ by F, i.e., we have

$$u^\infty = Fu^i. \tag{2.5}$$

The mapping F is a compact linear operator. It is a basic result known as *Rellich's lemma* that the scattered field $u^s \in C^2(\mathbf{R}^m \setminus \overline{D})$ is determined by the knowledge of the far field pattern u^∞ on some open subset of the unit circle or sphere, respectively.

For the following presentations we need to consider the scattered field for incident plane waves and for incident point sources. A point source is modeled by the *fundamental solution*

$$\Phi(x,y) := \begin{cases} (i/4)H_0^{(1)}(\kappa|x-y|) & m=2, \\ e^{i\kappa|x-y|}/4\pi|x-y| & m=3, \end{cases} \tag{2.6}$$

of the Helmholtz equation in two or three dimensions. Here, $H_n^{(1,2)}$ denotes the Hankel function of order n of the first or second kind. An incident plane wave is given by $u^i(x,d) := e^{i\kappa x\cdot d}$ with *direction of incidence* $d \in \mathbf{S} := \{x \in \mathbf{R}^m : |x| = 1\}$. When u^i has one argument, we consider it to be an arbitrary incident wave, with two arguments it denotes an incident plane wave.

The *direct* problem under consideration is to calculate u^s or u^∞ from the knowledge of the scatterer $\mathcal{D}$ with its boundary condition and the knowledge of the incident wave u^i.

There are several *inverse* problems which might be of interest, compare [**15**]. We will restrict our main attention to the reconstruction of the support D of $\mathcal{D}$.

3. Evolutionary algorithms. Evolutionary algorithms have been developed in the area of computational optimization. Their main idea is

to employ concepts of biological evolution like recombination, mutation and selection to achieve the approximate solution of optimization problems. As the main difference to conventional optimization methods they use *stochastic components* for iteration and they perform a *parallel search* on a whole population of solutions.

Different main categories of evolutionary algorithms (EA) have been developed independently from each other. The most common form of EA are *genetic algorithms* (GA) developed by Holland [**9**] followed by *evolutionary strategies* (ES) by Rechenberg [**17**] and Schwefel [**18**] and *evolutionary programming* (EP) by Fogel, Owens and Walsh [**5**]. Also *genetic programming* (GP) by Koza [**10**] is considered as a separate concept even if it was developed on the basis of GA. The concepts of these approaches differ by the representation of the solutions, notations and the focus of each technique. Following the general terminology we will speak of the *category* of the evolutionary algorithm when referring to these differences. From a mathematical viewpoint, they all employ the same principles and ideas; the differences are mainly on the level of implementation and software technology.

Our main idea here is to merge classical iteration technique with an evolutionary algorithm. We will first describe the basic elements of evolutionary algorithms, where we focus on the elements which are used in our numerical scheme, compare Figure 1.

3.1. *Principles and notation.* As evolutionary algorithms are inspired by natural principles the notations which are used are taken from their biological equivalent. First an *individual* is a representation of a possible or approximate solution to the given problem. Depending on the category of the EA it is represented in a form which is adequate to the method under consideration.[1] A set of individuals is called a *population*. Evolutionary algorithms, in contrast to iterative schemes, do not only consider one solution at each time but a population of several solutions/individuals. To each individual the algorithm assigns a *fitness value*, which is usually assumed to be constructed such that it leads to a well-ordered set of individuals within some population.

EAs are iterative methods. Starting with an initial solution the same steps are repeated until an individual requires a given criterion. In each iteration the individuals of an existing population are used to generate

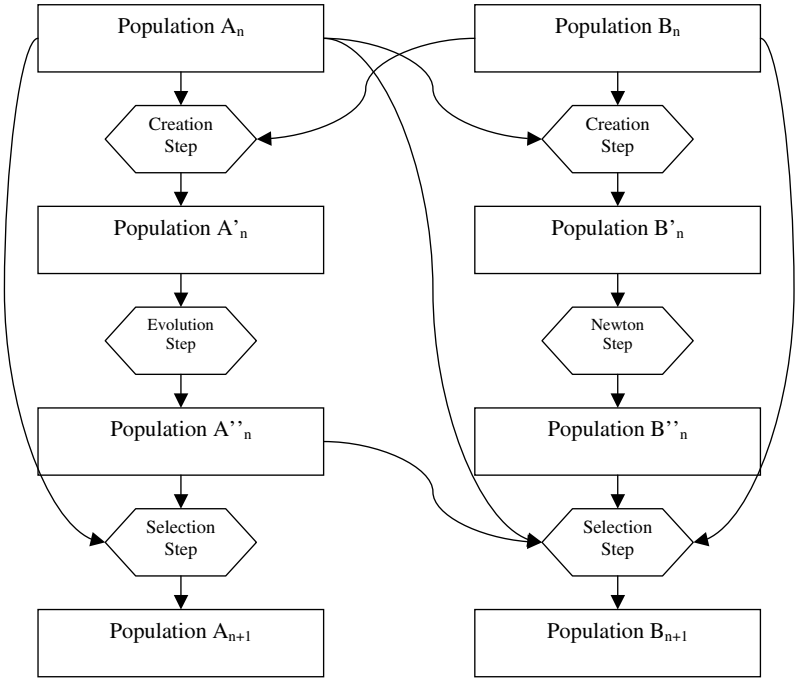

FIGURE 1. Flow diagram for the evolutionary Newton method. We work with two populations A_n and B_n. On population A_n we carry out a classical evolutionary algorithm (left column). Population B_n is called the *Newton population*, here we employ Newton steps for mutation and stochastical steps for the variation of the degrees of freedom for the shape representation (right column). Exchange between these populations is obtained using migration (rolling arrows).

individuals of a new population. The existing population is then called *parents*, the resulting population *children*. The outcome of an iteration step is called *generation*. The typical main steps which are executed in one iteration are *recombination, mutation* and *selection*. The left column of Figure 1 shows the main principle of an EA.

Second, we need to select the stochastic elements of our evolutionary algorithm. It is the goal that the next generation is better than the previous one such that those individuals which have been created by a good recombination and have undergone a good mutation will be

selected for the next generation whereas those with a bad recombination and mutation will not be taken.

In the following we will summarize the realization of recombination, mutation and selection.

3.2. *Recombination.* The creation of a new individual from two or more existing individuals by combination of their properties is called recombination.[3] In nature recombination is realized by the process where a child inherits properties of both of its parents.

Depending on the representation form of an individual one can distinguish between discrete and intermediate recombination. Discrete recombination can be done for any representation form of the individuals. It means that the resulting individual inherits for each variable/property the value of one of the initial individuals. A special form of discrete recombination, which is only used in genetic algorithms, is the so-called 'crossover' which requires a vector representation of the solutions.

Intermediate or continuous recombination can be done only if the properties are represented by real values and means that the resulting value is taken as the weighted mean of the initial values where the weights are randomly chosen such that the sum over all weights has to be 1 for each variable. If the weighting is the same for all variables this is also called 'line recombination.'

3.3. *Mutation.* Mutation means an, at least partially, stochastic change of an individual. It is applied on each newly created individual after recombination or replication. The possible forms of mutation depend on the representation of an individual. In its easiest form,which is mainly used in genetic algorithms, mutation is the addition of relatively small random real numbers to the values of each variable. This number can be totally random or it can be restricted to be a multiple of a given stepsize which corresponds to a certain discretization of the process. In GA the solution is often represented as a binary vector. In this case mutation means the inversion of randomly selected bytes.

In evolutionary strategies the solution is not always represented as a vector of values. Thus, we have to generalize the notion of mutation and understand it just as a change of the solution which involves a stochastic component at at least one point. This also means that in ES the mutation can change the solution significantly and is usually the main operation in the creation of a new generation[4], whereas it is much less important in genetic algorithms.

3.4. *Selection.* In biology selection occurs if there are more individuals than resources which can aliment them. In this case only those individuals which are the best adapted to the environment survive. Also in EAs we use selection to reduce the size of the population and to sort out the individuals with a bad fitness function in order to find a good solution to the given problem. For this reason it is necessary in any form of EA that in each generation new individuals are created such that we have more individuals than before the execution of recombination and mutation. So selection is executed at the end of each generation step to determine the parent population for the next generation.

The most simple case of selection is the deterministic rank-based selection which will be used in our algorithm. We denote the size of the parent population by μ, the size of the children population by λ. Depending on the question whether the parents can be re-selected we speak of a (μ, λ) selection if only the children can be selected for the next generation and a $(\mu, \mu + \lambda)$ selection if also the parents can be selected.[5] The $(\mu, \mu + \lambda)$ selection avoids that the solution can get worse but increases the chance of getting stuck in a local optimum. Note that for the (μ, λ) selection it is necessary that $\lambda > \mu$ as otherwise no individual would be sorted out.[6]

Beyond this simple selection model there exist more sophisticated stochastic models which can also be fitness-proportional instead of rank-based so that in the creation of the next generation the selected individuals with a better fitness function are also more likely to produce offspring. Stochastic selection models are for example roulette-wheel selection, stochastic universal sampling, local selection or tournament selection. For more information see for example [**11**].

Finally note that if there is more than one population selection can also incorporate individuals or properties from other populations, which

is called migration. This concept will be employed for our evolutionary Newton method.

3.5. *Further development of evolutionary algorithms.* The considerable increase of computational power in recent years made it possible to extend the classical concept of one population with a once given definition of recombination, mutation and selection. So several extensional concepts for EA have been developed.

One of these new concepts which will also be applied in our case is the concept of *multiple populations* which have different rules from each other.[7] This might be necessary if for example one has several optimization criteria which are equally important. In this case for each criterion a separate population is created where the fitness function of each individual is defined according to this criterion. Another reason for using this concept can be that one wishes to consider only individuals which fulfill certain restrictions and let them develop independently for a certain time. This is also the motivation in our case.

Obviously multiple populations only make sense if also a certain exchange between them is allowed, at least under certain conditions. Corresponding to biological terms, this exchange is called *migration*.

Another important concept is the so-called *meta evolution*. This means that the parameters of the EA itself[8] are submitted to an optimization process which can itself be again an evolutionary algorithm. The idea has already been pronounced by Rechenberg in 1973 (see [**17**]) as a *learning population* but could not be reasonably realized at that time due to limited computer power. Note that if the method for the meta evolution is an EA we also have to consider several populations which are then considered as individuals of the Meta-EA. We will use the concept of meta evolution to control the degree of polynomials in our domain representations but realize it within one population on the parameter level. We call this *impure meta evolution*.

For the sake of completeness note that there also exists the concept of *co-evolution* following the biological principle of parasitism introduced in [**6**]. This basically means that both the problem instances and the solution concepts are evaluated. A solution concept is good if it solves as many problem instances as possible whereas a problem instance is good if there are many solution concepts that fail to solve it. This competition leads then to more difficult problem instances and more

sophisticated solution concepts. The concept of co-evolution is mainly used in genetic programming.

4. The classical Newton scheme. The Newton method is a well-known classical method for the search of zeros of a function or the solution of operator equations. Newton's method has also been applied to inverse problems, compare the literature in [**1, 4**]. The solution to the nonlinear operator equation

$$L(x) = 0 \tag{4.1}$$

is found by iteratively solving the linearized equation

$$L'(x_n)r_n = -L(x_n) \tag{4.2}$$

with some iterative solution x_n for the update r_n. Starting with some first guess x_0 we define $x_{n+1} = x_n + r_n$. For ill-posed problems the inverse of the operator L is unbounded. In this case, also the Fréchet derivative $L'(x_n)$ is a linear unbounded operator. Thus, the solution of (4.2) needs to be regularized. In the following reasoning the operator L will be given by the operator F, defined in (2.5), applied to some fixed incident field u^i, i.e., we have

$$L : \partial D \longmapsto F(\partial D)u^i. \tag{4.3}$$

Here, we use the setup as described in [**14, 15**]. In a neighborhood of a reference domain ∂D_0 we parametrize a set of C^2-domains by a vector field $r : \partial D_0 \to \mathbf{R}^m$. It is well known, see the literature cited in [**14**], that the Fréchet derivative of the Dirichlet scattering problem can be calculated via

$$L'(\partial D)r = L\left(r_\perp \frac{\partial u}{\partial \nu}\right), \tag{4.4}$$

where $\partial u/\partial \nu$ is the normal derivative of the total field u for the domain ∂D and

$$r_\perp(x) := \nu(x) \cdot r(x), \quad x \in \partial D. \tag{4.5}$$

A classical Newton step can be carried out as follows. We use a basis

$$\left\{ r^{(j)} : \ j = 1, \dots, N \right\} \tag{4.6}$$

of vector fields on the boundary of the iterate ∂D_n. Then we search for the coefficients $\beta_{n,j}$, $j = 1, \dots, N$ of

$$r_n = \sum_{j=1}^{N} \beta_{n,j} r^{(j)}, \tag{4.7}$$

via equation (4.2). We discretize the righthand side using colloquation points $\hat{x}_k$, $k = 1, \dots, N$ for the far field pattern on the unit sphere or unit circle, respectively. This leads to the $N \times N$ discrete linear system

$$\sum_{j=1}^{N} \left(L(\partial D_n) \left(r^{(j)} \frac{\partial u}{\partial \nu} \right) \right) (\hat{x}_k) \beta_{n,j} = -(L(\partial D_n) u^i)(x_k), \quad k = 1, \dots, N, \tag{4.8}$$

which we will abbreviate by

$$A_n \beta_n = f_n. \tag{4.9}$$

A regularized inverse can be obtained for example via Tikhonov regularization

$$\beta_n^{(\alpha)} = (\alpha I + A_n^* A_n)^{-1} A_n^* f_n. \tag{4.10}$$

Alternative realizations have been proposed by Potthast [**14**]. For iteratively regularized Newton schemes we refer to Hohage [**8**]. The convergence of the Newton scheme for exponentially ill-posed problems is still an open problem, where some convergence results can be found in [**14**] or the work of Hohage [**7, 8**].

In the next section, we will use the Newton step described by equation (4.8) or (4.9), respectively, as an ingredient of the evolutionary algorithm and couple it via a multi-population approach with the power of the stochastical update technology.

5. An evolutionary algorithm with Newton updates. The goal of this section is to describe a novel approach to the inverse scattering problem by a combination of the Newton method and the evolutionary approach. The scheme allows the reconstruction of several separate objects from the far field pattern for scattering of one incident time-harmonic wave without any a priori knowledge about the number and location of objects under consideration. The methods are not just taken in turns, but we incorporate the Newton update as update technique into an evolutionary algorithm and further develop this technique by a two-population approach.

For the development of this method, we first tested a standard Newton scheme and a standard evolutionary algorithm. Both have clear disadvantages: the Newton scheme diverges if we do not start with the right number of domains close to the true solution. The evolutionary algorithm is very slow. We also tested first the use of the evolutionary algorithm and then refined the search via the Newton method after a threshold criterion was reached. But it is difficult to define a reasonable threshold which works for a larger set of unknown scatterers. This leads to the natural choice of two parallel populations, one for the EA search and a second where the Newton scheme is used. Exchange between these populations is carried out by a migration strategy. The population with the pure EA search is used to find the number and the approximate location of the obstacles. The Newton population serves to find the exact location and shape of the obstacles. We will denote them by population A and population B.

5.1. *Definition of an individual and of the fitness function.* An individual is always a possible approximation to the given problem. So in this case an individual is a set of domains which lie inside the given search area. In population A an individual is a set of ellipses which are given by the coordinates of their centers (m_1, m_2) and their axes a and b. In population B the numbers (a, b) are replaced by two radial functions

$$\rho_i(t) := \sum_{k=0}^{n_i} a_{ik}\cos(kt) + \sum_{k=1}^{n_i} b_{ik}\sin(kt), \quad i = 1, 2,\ t \in [0; 2\pi] \tag{5.1}$$

with polynomial degrees n_1 and n_2. The *fitness function* of an individual is then defined as the norm of the difference to the measured far

field

$$\mathcal{F}(\mathcal{D}) := ||u^\infty(\mathcal{D}) - u^\infty_{\text{meas}}||. \tag{5.2}$$

Here, we do not add any regularization to the fitness function. For the evolutionary part we obtain regularization by discretization, since we limit our interest to ellipses, i.e., to low-dimensional ansatz spaces. In the Newton part of the algorithm we employ regularization via Tikhonov regularization in equation (4.8) for the Newton updates.

5.2. *Definition of one iteration step.* As described in Section 3, the iteration step describes how to generate the child populations A_{n+1} and B_{n+1} from the parent populations A_n and B_n. Following the classical scheme for an EA we split the iteration into the *creation step*, where populations A'_n and B'_n are created. Then, corresponding to the mutation in the general scheme, we change the individuals in the populations A'_n and B'_n leading to modified populations A''_n and B''_n. For population A'_n this is done by a classical mutation, and we will call this part the *evolution step*, whereas for population B'_n we will incorporate the Newton method, and we will call this part the *Newton step*. Finally, we use a *selection step* to reduce these populations to the child populations A_{n+1} and B_{n+1}. *Migration*, i.e., the exchange of elements between A and B, can occur both in the creation step and in the selection step, where in both cases we will use some asymmetric migration.

5.2.1. *The creation step*: *Replication and recombination.* The individuals in population A'_n are either created

1. by replication of a parent individual from population A_n or by migration of a parent individual from population B_n or

2. by recombination of two parent individuals of population A_n.

For A'_n we generate λ new individuals with $\lambda \geq \mu$. The probability that an individual is replicated from population A_n is set to p_1, for population B_n the migration probability is p_2 with numbers $0 \leq p_1$, $p_2 \leq 1$.

The individuals defined in population B_n are defined by polynomials. Thus, if they are migrated into the population A'_n, then they need to be

reduced to appropriate ellipses. This is carried out by a least-squares approximation.

Second, recombination of individuals is carried out by an arbitrary selection of their connected components. We first select two individuals I_1 and I_2, then for each of them choose a subset of their connected components and create a new individual which has both the selected components from individual I_1 and individual I_2. Note that it is possible that after recombination some domains (components) intersect leading to a nonvalid approximation. In this case we need to carry out a *resolution step* described in subsection 5.2.4.

For population B'_n the creation step is always performed by replication of the parents of B_n and by migration of the parents of A_n, such that we have 2μ individuals.

5.2.2. *Mutation of population* A: *Classical evolution.* The mutation of an individual from population A'_n is carried out either by parameter changes like shifts, dilation or by splits. The following list summarizes the possible mutation steps used in our algorithm.

(M1) Create a random vector $(dm_1, dm_2, da, db) \in \mathbf{R}^4$, and mutate the ellipse

$$D = (m_1, m_2, a, b) \tag{5.3}$$

to the new component

$$D' = (m_1 + dm_1, m_2 + dm_2, a + da, b + db). \tag{5.4}$$

We have experimented with different probability measures on the space $\mathbf{R}^4$, leading to shifts in x or y-direction, dilation and a combination of shifts with dilation.

(M2) Vertical split: the ellipse (5.3) is split vertically into two ellipses defined by

$$\begin{aligned} D'_1 &= \Big(m_1 - \frac{a}{2}, m_2, \frac{a}{2} - \varepsilon, b\Big), \\ D'_2 &= \Big(m_1 + \frac{a}{2}, m_2, \frac{a}{2} - \varepsilon, b\Big) \end{aligned} \tag{5.5}$$

with some small parameter $\varepsilon > 0$.

(M3) Horizontal split: the ellipse (5.3) is split horizontally into

$$D_1' = \Big(m_1, m_2 - \frac{b}{2}, a, \frac{b}{2} - \varepsilon\Big), \tag{5.6}$$
$$D_2' = \Big(m_1, m_2 + \frac{b}{2}, a, \frac{b}{2} - \varepsilon\Big)$$

with some small parameter $\varepsilon > 0$.

The probability distribution for (dm_1, dm_2, da, db) should be according to the principle that, in a purely random search, small changes should be more likely than big ones. Again, it is possible that after these operations some domains (components) intersect, such that we need to carry out again a resolution step described in subsection 5.2.4.

5.2.3. *Mutation of population* B: *Classical Newton and meta evolution.* For the 'Newton-population' B_n we work with mutations based on Newton steps and with meta evolution changing the degree of the polynomials under consideration.

First, for each domain the polynomial degrees of n_1 and n_2 are increased with a probability of $p_3 \in (0, 1)$ independently for n_1 and n_2 or decreased with a probability of p_3.[9] If they are decreased the new coefficients are determined by the least-squares solution.

Experiments with the algorithm show that a large speed-up can be achieved when the probability to increase the polynomial degree is increased with decreasing error functional.

Second, a classical Newton step as described in Section 4 is executed on the whole individual. The coefficients for the new radial functions are then determined by least-squares from the discretized new domain. It is possible that two domains intersect. In this case they are reduced to ellipses and treated like in population A, compare subsection 5.2.4.

5.2.4. *Resolution for domain intersections.* Recombination and mutation of domains can lead to the intersection of some of the components of scatterers. In this case a *resolution step* is necessary to generate an admissible individual in our population A or B. We worked with two types of resolution steps.

Consider two domains D and G which intersect.

1. Union of the two domains. We take the union of the two components and replace it by an appropriate ellipse via a least squares fit.

2. Diminishment or split of a domain. We keep one of the domains fixed, for example G. The other domain is either diminished or split such that we obtain two or three admissible components. A split is carried out if the domain $D \setminus G$ consists of more than one component. Otherwise a diminishment of the domain D is carried out by a least squares fit for an ellipse on $D \setminus G$.

5.2.5. *The selection process.* For the selection process for population A, we employ the $\mu + \lambda$-selection, i.e., we use a union of the parent population A_n with the intermediate population A_n''. Therefore, we obtain a monotonously decreasing error functional. The *deterministic best* μ solutions for population A_{n+1} are chosen using the fitness function 5.2 as *selection criterion*.

To keep the *diversity* of the population, we use the following well-known measure: If the fitness functions of two individuals differ only by a small constant ε, we consider them as equal and choose only one of them and continue with the next solution instead.

For the selection of population B_{n+1}, we compare the parents B_n, the children B_n'', the parents A_n and the λ children A_n''. So the best individual of population B_{n+1} is always the best of both populations. The asymmetric use of migration in the selection process is motivated by the strength of Newton's method to work in a larger space than the stochastic search on ellipses in the evolutionary branch of the algorithm.

5.2.6. *The calculation of a starting solution.* In the introduction we identified the *point source method* introduced in [**12, 13**] as a good scheme to calculate a starting solution. In general, the point source method reconstructs the scattered field in the exterior of some test domain. Then, the unknown shape is found as the zero curve of the sum of the incident and scattered fields. However, it is computationally difficult to construct test domains such that the scattered field can be calculated in regions between different components of a scatterer, i.e., in the interior of the convex hull of the scatterer under consideration,

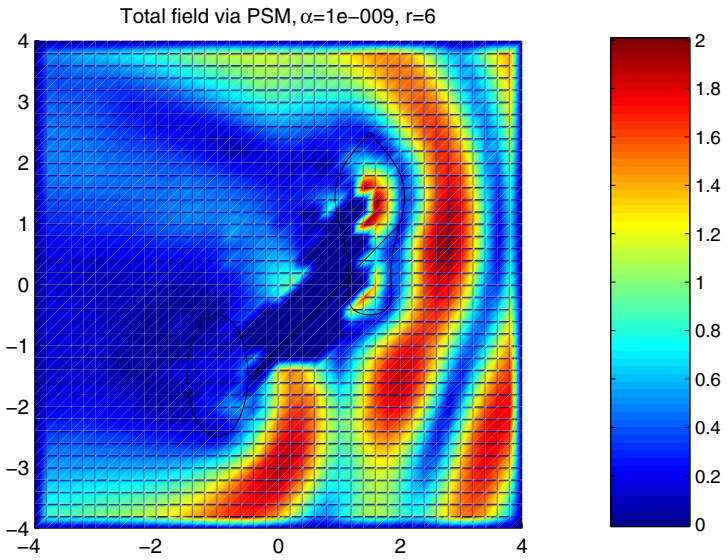

FIGURE 2. Reconstruction of the total field for scattering by two objects via the point source method (PSM). The field is well reconstructed outside of the convex hull of the scatterers. The reconstruction in the region between the scatterers leads to difficult algorithmical problems.

compare Figure 2. Thus, we used the point source method to determine the convex hull of the scatterers in a preprocessing step. The convex hull is then circumscribed with a rectangle which is taken as our *initial search area*.

We divide the initial search area into a grid of $\lceil\sqrt{\mu}\rceil^2$ rectangles of equal size such that the x and y axes of the search area are divided into $\lceil\sqrt{\mu}\rceil$ parts. Into each of these rectangles we put a large ellipse and create an individual which consists of this ellipse as the domain set. This way we get at least μ different individuals which are equally distributed over the search area and we take them as our starting solution. If μ is not square, i.e., we get more than μ individuals, this way we select the best μ of them as our starting solution for both populations.

6. Numerical realization and results. We realized the evolu-

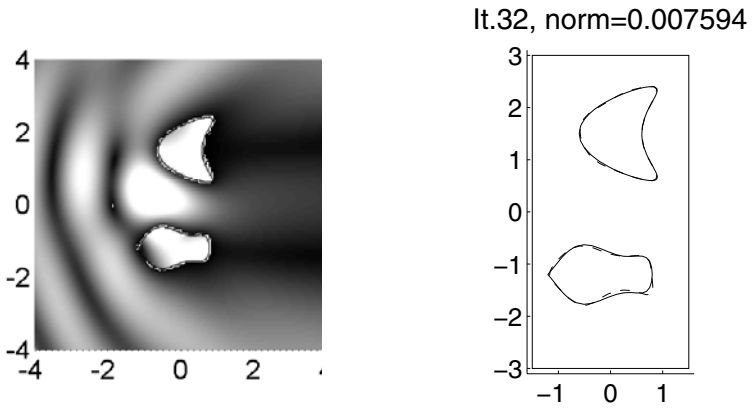

FIGURE 3. Reconstruction of an object with two components for the wave number $\kappa = 2$. The original obstacles and the reconstructions (dotted line) by the evolutionary Newton method are shown in the right image. Here the result was obtained by 32 iterations.

tionary Newton scheme using boundary integral equations. For the numerical solution of the forward Dirichlet scattering problem we used a combined single- and double-layer approach [**2**]

$$u^s(x) = \int_{\partial D} \left\{ \frac{\partial \Phi(x,y)}{\partial \nu(y)} - i\Phi(x,y) \right\} \varphi(y)\, ds(y), \quad x \in \mathbf{R}^2 \setminus \overline{D} \tag{6.1}$$

with density $\varphi \in C(\partial D)$ and the fundamental solution (2.6) to represent the scattered field u^s. This leads to the integral equation

$$(I + K - iS)\varphi = -2u^i \tag{6.2}$$

with the *double-layer boundary integral operator*

$$(K\varphi)(x) := 2 \int_{\partial D} \frac{\partial \Phi(x,y)}{\partial \nu(y)} \varphi(y)\, ds(y), \quad x \in \partial D, \tag{6.3}$$

and the *single-layer boundary integral operator*

$$(S\varphi)(x) := 2 \int_{\partial D} \Phi(x,y) \varphi(y)\, ds(y), \quad x \in \partial D, \tag{6.4}$$

on the boundary ∂D of the scatterer under consideration. This approach is solvable for all wave numbers $\kappa > 0$. We used the Nyström

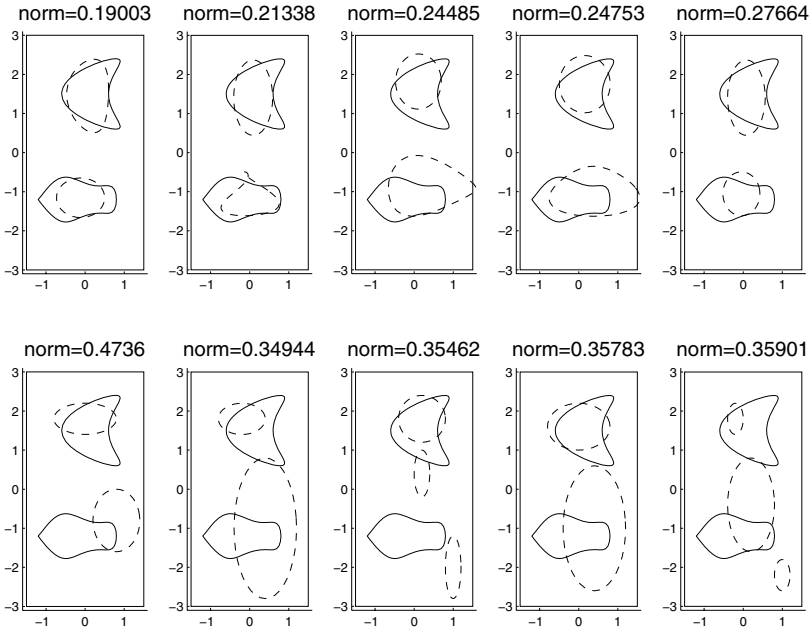

FIGURE 4. Population after 5 iterations. The images in the first row show a selection of population B, the images in the second row are taken from population A.

method described in [**1**] splitting off the singularity. It is exponentially convergent for smooth domains and provides a quick and reliable scheme for the calculation of the scattered field. The far field pattern for the double-layer potential is calculated by

$$(K^\infty\varphi)(\hat{x}) := \frac{e^{i\pi/4}}{\sqrt{8\kappa\pi}} \int_{\partial D} \frac{\partial e^{-i\kappa\hat{x}\cdot y}}{\partial\nu(y)} \varphi(y)\, ds(y), \quad \hat{x} \in \mathbf{S}. \tag{6.5}$$

For the single-layer operator the far field pattern is given by

$$(S^\infty\varphi)(\hat{x}) := \frac{e^{i\pi/4}}{\sqrt{8\kappa\pi}} \int_{\partial D} e^{-i\kappa\hat{x}\cdot y} \varphi(y)\, ds(y), \quad \hat{x} \in \mathbf{S}, \tag{6.6}$$

where the constants in front of the integrals are those for two dimensions.

For the realization of Newton's scheme, we need to calculate the normal derivative $\partial u/\partial\nu$ of the total field at the boundary of the

scatterer. This leads to the representation

$$\frac{\partial u(x)}{\partial \nu(x)} = \frac{1}{2}T - \frac{i}{2}(K' - I)\varphi \tag{6.7}$$

with the adjoint K' of the operator K and the strongly singular operator T given by

$$(T\varphi)(x) := 2\frac{\partial}{\partial \nu(x)} \int_{\partial D} \frac{\partial \Phi(x,y)}{\partial \nu(y)} \varphi(y)\, ds(y), \quad x \in \partial D. \tag{6.8}$$

For simplicity, we approximated the calculation of T by some numerical differentiation of the double-layer potential. Alternatively, we tested a single-layer potential approach which is calculated by solving the first-kind integral equation

$$S\varphi = -2u^i \tag{6.9}$$

on ∂D via Tikhonov regularization. This is reasonable due to the fact that we work with approximate solutions to the inverse problems within a stochastic algorithm and we do not need to calculate a precise normal derivative on the approximation surface. Numerically, for the single-layer approach the normal derivative is approximated by

$$\frac{\partial u(x)}{\partial \nu(x)} \approx \frac{1}{\tau}\left((S\varphi)(x + h \cdot \nu(x)) - (S\varphi)(x + (h + \tau) \cdot \nu(x))\right) \tag{6.10}$$

where $h, \tau > 0$ needs to be chosen appropriately. This turned out to be sufficient for the examples under consideration and can be realized flexibly for any number of scattering components. Then, the solution to the equation (4.8) is calculated with some appropriate set of basis functions. A further alternative is to use a weakly singular integral equation of the second kind like the equation

$$(I + K' - iS)\frac{\partial u}{\partial \nu} = 2\frac{\partial u^i}{\partial \nu} - 2iu^i, \tag{6.11}$$

whose solution density coincides with the searched normal derivative, compare [**1**, page 58].

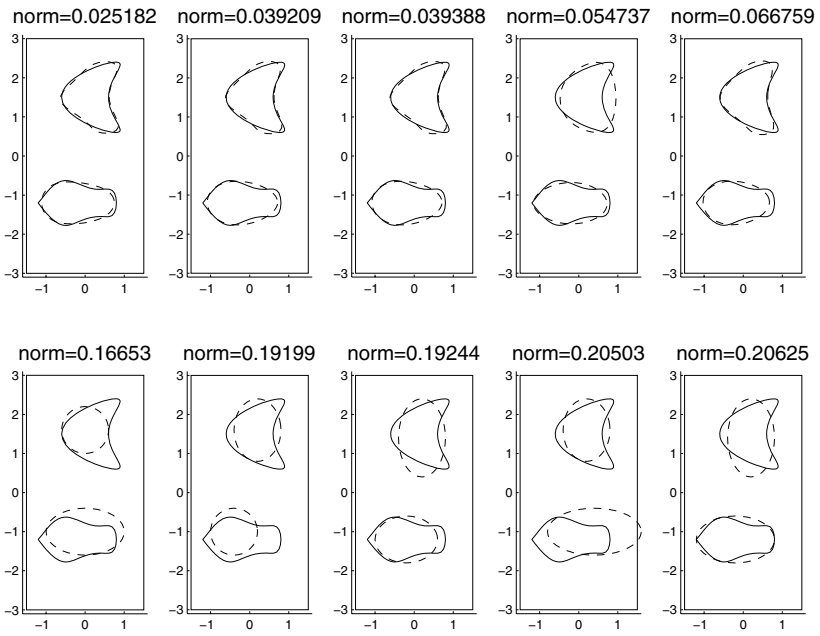

FIGURE 5. Population after 10 iterations. The images in the first row show a selection of population B, the images in the second row are taken from population A.

Parameter choices, control of algorithm and stopping rule. We need to choose all different parameters, splitting and mutation probabilities for the evolutionary algorithm and the Newton scheme. For our realization of the evolutionary search via population A we restricted the parameters of the ellipses to multiples of a given step-size s. In our examples we used $s = 0.2$. This also means that the components of the mutation vector (dm_1, dm_2, da, db) are only allowed to be a multiple of s. As small changes should be more probable than big ones, the probability that the change is $\pm N \cdot s$ is set to 2^{-N} for $N \in \mathbf{N}$.

When splitting an ellipse we set the distance parameter ε between the two new ellipses to $s/2$. The probability p_1 that an individual is replicated from population A_n is set to $1/4$, the probability p_2 that it is migrated from B_n is set to $1/16$. For the mutation of population A the probability for a split is set to $1/4$, otherwise the domain will be randomly mutated.

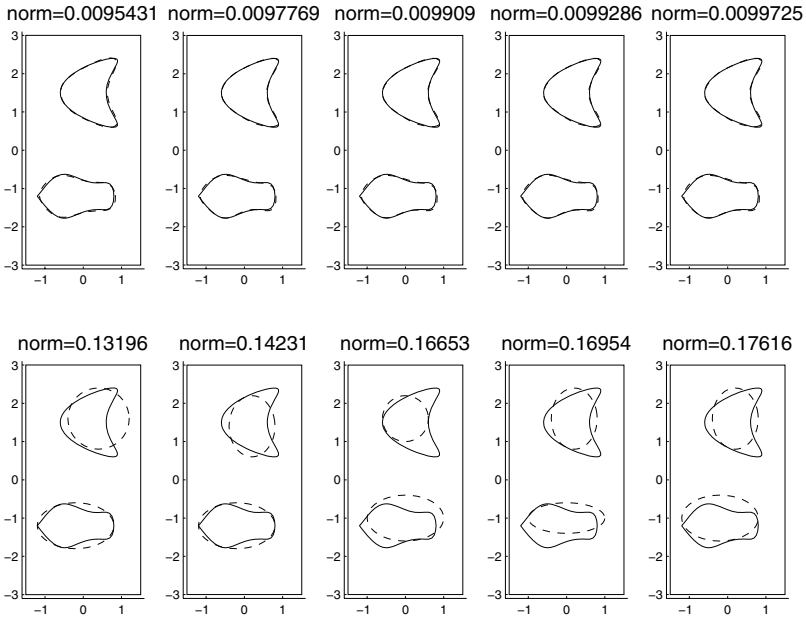

FIGURE 6. Population after 20 iterations. The images in the first row show a selection of population B, the images in the second row are taken from population A.

For the Newton-branch B of the evolutionary Newton algorithm we need to determine the probability p_3 of a change of the polynomial degree. We used the *dynamic choice*

$$p_3 = \frac{1/\mathcal{F}(\mathcal{D})}{10 + (1/\mathcal{F}(\mathcal{D}))} \tag{6.12}$$

i.e., p_3 depends on the fitness function. We limited the maximum polynomial degree to 10. We used five individuals for the parent populations and twenty individuals for the child population A'_n, i.e., we worked with $\mu = 5$ and $\lambda = 20$. For the Newton population B_n, we used $\mu = 5$ and $\lambda = 10$. Here, the child population B'_n is composed by the five parent individuals from population A_n and those from population B_n.

We stopped the iteration if there was no improvement of the best individual in three successive steps (*stopping rule*).

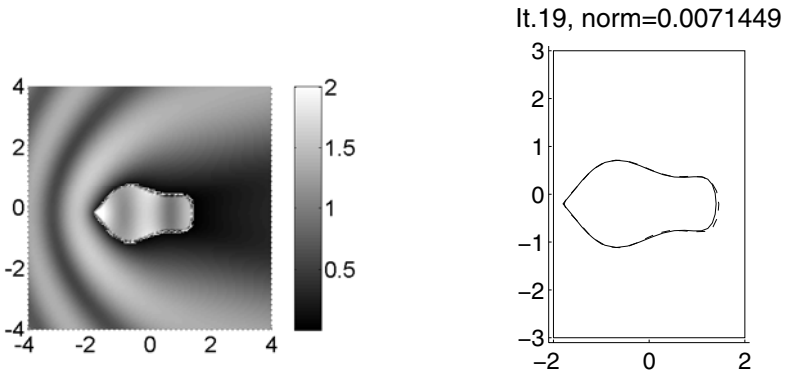

FIGURE 7. Reconstruction of a scatterer with only one component. We show the reconstruction for iteration 19. One component is easily identified and found rather quickly.

Reconstruction time. For the evolutionary Newton method we need to solve the scattering problem for a rather high number of settings. For the two-dimensional problem under consideration, the solution of each scattering problem can be performed within seconds or even quicker. Thus, the evaluation of the fitness function of 20 individuals takes about 20–30 seconds. Each iteration step in the evolutionary Newton

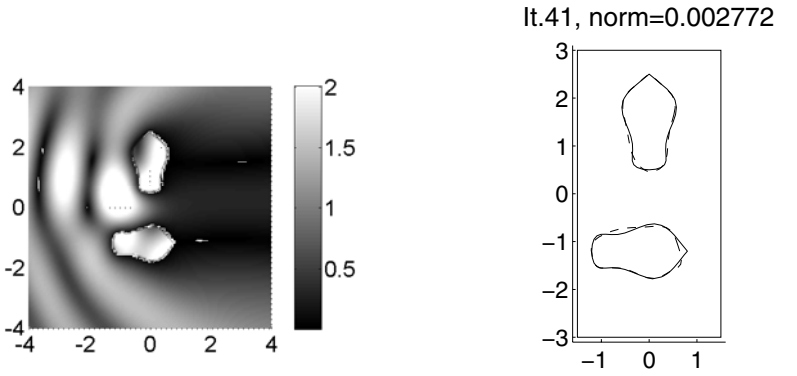

FIGURE 8. Reconstruction of a scatterer with two components. The reconstruction shows the 41st iteration.

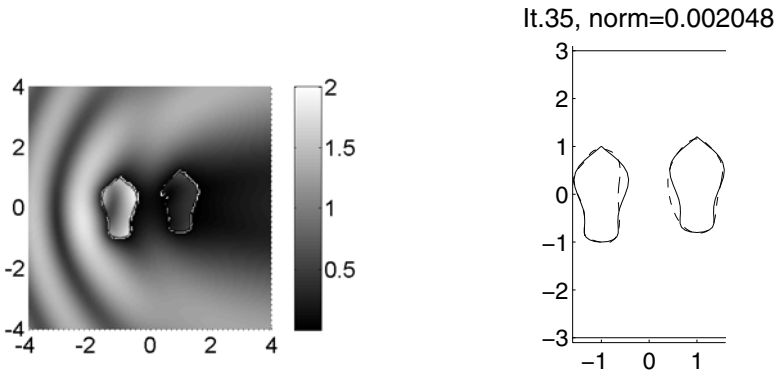

FIGURE 9. Reconstruction of a scatterer with two components, where one component is in the shadow of the second component. Here we needed 35 iterations and clearly see that the shadow regions are more difficult to reconstruct.

scheme needs 30–60 seconds. Twenty iterations correspond to 10–20 minutes. This is much more compared to the reconstruction of the point source method, which takes less than 5 seconds. But the point source method in the basic form only reconstructs the convex hull of the obstacles under consideration. With our evolutionary Newton scheme, we obtain high-quality reconstructions of multiple obstacles without any knowledge of the number of components.

Examples. We first show reconstructions of an object with two components as shown in Figure 3. We demonstrate examples of the populations after 5, 10 and 20 iterations in the Figures 4, 5 and 6. The examples show that the reconstruction works very well. After five iteration steps the algorithm has identified the number of components. After 10 iterations, the shape of the objects is already rather well established. After 20 iterations the reconstruction is nearly perfect. Since we worked with simulated data without additional errors, in principle full reconstructions are possible. However, the true curve is not in the ansatz space for the polynomial shape representation. The polynomial degree of the ansatz for the domain is found by the algorithm itself, which also takes time and effort and explains the role of the iterations 10 to 20.

Further examples with scatterers consisting of one, two or three

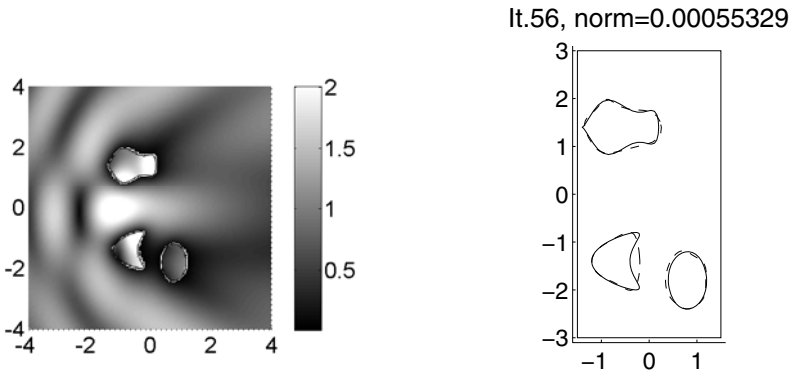

FIGURE 10. Reconstruction of a scatterer with three components. We needed 56 iterations for this example. For three or more components the reconstruction times go up considerably.

components are shown in Figures 7, 8, 9 and 10. Here, we kept the wave number $\kappa = 2$ and tested different numbers of components and different locations of the scatterers. It is more difficult to reconstruct the shadow regions of scatterers, as can be seen in Figures 9 and 10. For each example, we give the iteration after which the algorithm terminated, i.e., there was no improvement in the last three steps. Usually, a good reconstruction is already obtained after less than half of the terminating number of iterations, compare Figure 5.

Further work. Clearly, the algorithm under consideration leaves space for a number of different realizations and improvements. For example, in our examples we did not incorporate the rotation of ellipses in the evolutionary algorithm, which would be a natural further step. Also, the extensive testing with noisy data is of practical importance. Here, we have used the discretization of the ellipses (location of their centers, maximal degree and values of polynomial coefficients for radial functions) as regularizing quantities. This corresponds to *regularization by projection* onto finite dimensional subspaces. The relation and influence of these parameters in comparison to the regularization parameter of Newton's method is an interesting open theme of study. Further investigation is to be carried out in future work.

ENDNOTES

1. This means, e.g., that in GA it has to be represented as a vector of variables or in GP as a parse tree. Other forms like ES do not require a special form.

2. This means that if we have more than one optimization criterion we need to define a priority to get the population well-ordered.

3. If an new individual is created as a copy of another individual this is called replication or cloning.

4. The first ES introduced by Rechenberg did completely omit the recombination.

5. There can be also a mixture as for example a $(\mu, \lambda + 1)$ selection where only the best parent individual can be re-selected.

6. Also for the $(\mu, \mu + \lambda)$ selection it is advised that $\mu > \lambda$ to decrease the probability that certain individuals of the parent population are not reproduced at all. The fact that in the first ES Rechenberg set λ to 1 was more a tribute to limited resources.

7. In biology this corresponds to different environment conditions which require other qualities from an individual.

8. Like size of the population, definition of the selection or the parameter setup.

9. It is possible that only n_1 is changed or only n_2 is modified.

REFERENCES

1. D. Colton and R. Kress, *Inverse acoustic and electromagnetic scattering theory*, second edition, Appl. Math. Sci. **93**, Springer-Verlag, Berlin, 1998.

2. D. Colton and R. Kress, *Integral equation methods in scattering theory*, Pure Appl. Math., John Wiley & Sons Inc., New York, 1983.

3. O. Dorn, E.L. Miller and C.M. Rappaport, *A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets*, Inverse Prob. **16** (2000), 1119–1156.

4. H.W. Engl, M. Hanke and A. Neubauer, *Regularization of inverse problems*, Math. Appl. **375**, Kluwer Academic Publishers, Dordrecht, 1996.

5. L.J. Fogel, A.J. Owens and M.J. Walsh, *Artificial intelligence through simulated evolution*, John Wiley, New York, 1966.

6. W.D. Hillis, *Co-evolving parasites improve simulated evolution as an optimization procedure*, Articial Life II, **10** (1991), 313–324.

7. T. Hohage, *Convergence rates of a regularized Newton method in sound-hard inverse scattering*, SIAM J. Numer. Anal. **36** (1999), 125–142 (electronic).

8. ———, *Iterative methods in inverse obstacle scattering: regularization theory of linear and nonlinear exponentially illposed problems*, PhD thesis, Linz, 1999.

9. J.H. Holland, *Adaptation in natural and artificial systems*, The University of Michigan Press, Ann Arbor, 1975.

10. J.R. Koza, *Genetic programming*: *On the programming of computers by means of natural selection*, MIT Press, Cambridge, 1992.

11. H. Pohlheim, *Evolutionäre Algorithmen–Verfahren, Operatoren, Hinweise aus der Praxis*, Springer, Berlin, 1999.

12. R. Potthast, *A fast new method to solve inverse scattering problems*, Inverse Prob. **12** (1996), 731–742.

13. ———, *A point source method for inverse acoustic and electromagnetic obstacle scattering problems*, IMA J. Appl. Math. **61** (1998), 119–140.

14. ———, *On the convergence of a new Newton-type method in inverse scattering*, Inverse Prob. **17** (2001), 1419–1434.

15. ———, *Point sources and multipoles in inverse scattering theory*, Chapman & Hall/CRC Research Notes Math. **427**, Chapman & Hall/CRC, Boca Raton, Florida, 2001.

16. ———, *A survey on sampling and probe methods for inverse problems*, Inverse Problems, Topical Review, 2006.

17. I. Rechenberg, *Evolutionsstrategie–Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, Stuttgart, 1973.

18. H.-P. Schwefel, *Evolutionsstrategie und numerische Optimierung*, PhD thesis, Technische Universität Berlin, 1975.

Institute für Numerische und Angewandte Mathematik, Universität Göttingen
Email address: eckel@math.uni-goettingen.de

Email address: r.w.e.potthast@reading.ac.uk