In silico optimization of critical currents in superconductors

Gregory Kimmel, Ivan A. Sadovskyy, and Andreas Glatz

Phys. Rev. E **96**, 013318 — Published 27 July 2017

DOI: 10.1103/PhysRevE.96.013318

In silico optimization of critical currents in superconductors

Gregory Kimmel,[1,2] Ivan A. Sadovskyy,[1,3] and Andreas Glatz[1,4]

[1] *Materials Science Division, Argonne National Laboratory, 9700 S. Cass Av., Argonne, IL 60637, USA*
[2] *Department of Engineering Sciences and Applied Mathematics,*
Northwestern University, 633 Clark St, Evanston, IL 60208, USA
[3] *Computation Institute, University of Chicago, 5735 S. Ellis Av., Chicago, IL 60637, USA*
[4] *Department of Physics, Northern Illinois University, DeKalb, IL 60115, USA*

(Dated: July 12, 2017)

For many technological applications of superconductors the performance of a material is determined by the highest current it can carry losslessly — the critical current. In turn, the critical current can be controlled by adding non-superconducting defects in the superconductor matrix. Here we report on systematic comparison of different local and global optimization strategies to predict *optimal* structures of pinning centers leading to the highest possible critical currents. We demonstrate performance of these methods for a superconductor with randomly placed spherical, elliptical, and columnar defects.

Keywords:

1. INTRODUCTION

The most important feature of superconductors for high-performance applications[1] is its ability to carry large currents with almost no dissipative loss. At the same time, recent advances have made it possible to manufacture type-II-superconductor-based cables on industrial scales making large-scale application possible.[2,3] The main dissipative mechanism in these type-II superconductors stems from the motion of magnetic vortices within the superconducting matrix. These are normal elastic filaments, carrying quantized magnetic fluxes,[4] which appear in these superconductors in magnetic fields larger than the first critical field — in contrast to type-I superconductors, which lose their superconducting properties once the magnetic field enters the material. The highest amount of current that can be passed through a superconductor without dissipation is known as the critical current, which strongly depends on the vortex dynamics and its interaction with non-superconducting defects. Strictly speaking, the critical current in the presence of vortices is always zero at finite temperatures due to thermal creep, but it is conventionally defined as the current at which the dissipation/voltage reaches some threshold value. At sufficiently large currents the dissipation and associated heating of the superconductor will ultimately lead to the loss of superconductivity.

Despite all technological advances, typical critical currents are still way below the theoretical limit and from an economic point of view, too low to be competitive with conventional cables for large-distance energy transport. An obvious way to improve the critical current in existing superconductors is to impede the vortex motion more effectively. This can be accomplished by placing defects, which can "pin" magnetic vortices and thus prevent their motion,[5,6] in a sophisticated way. The entirety of all defects is also called the pinning landscape (or *pinscape*), which in conjunction with intrinsic material inhomogeneities and the sample geometry define the critical current of a given sample.

This defines the task, which we consider in this work: the optimization of the pinscape for highest possible critical currents. More precisely, we concentrate on defects or pinning centers, which can be controlled during the fabrication of a sample, like self-assembled inclusions[7,8] or irradiation defects.[9–11] The efficiency of the pinscape strongly depends on the shape, size, and arrangement of individual pinning centers. Indeed, bigger defects ensure a larger pinning force, but, at the same time, reduce the effective cross-section of the superconductor needed for current flow. The optimal pinscape also depends on the intended application, particularly on the type of superconductor and on the value and the direction of external magnetic fields. The question we address in this paper is, how one can find the best pinscape most effectively. This systematic prediction of optimal pinscapes for a given application aims at replacing the traditional trial-and-error approach.[9]

Here, we test several optimization strategies allowing for a systematic improvement of the critical current in superconductors. We compare the efficiency of a global method (particle swarm optimization) and three local methods (Nelder-Mead method, pattern search, and adaptive pattern search) for a typical critical current optimization problem. The critical current for a given pinscape is calculated using a GPU-based iterative solver for the time-dependent Ginzburg-Landau (TDGL) equation describing type-II superconductors.[12] This model correctly captures the vortex dynamics[5,6,13] in superconductors in the vicinity of the critical temperature and is capable of reproducing experimental critical currents for a given pinscape.[8,9,14–16] In addition, we provide a detailed analysis of these methods applied to several benchmark functions for comparison.

The article is organized as follows. In Sec. 2 we formulate the general optimization problem and describe the optimization methods studied here. In Sec. 3 we present a detailed comparison of the efficiency of the chosen optimization strategies on benchmark functions and discuss results in Sec. 3.4. In Sec. 4 we briefly describe the TDGL model for superconductors and define three physically relevant optimization problems in Sec. 5 with discussion in Sec. 6. Finally, we summarize our results in Sec. 7.

2. OPTIMIZATION METHODS AND PROBLEM FORMULATION

Optimization methods are divided into two classes: local and global methods. Examples of global search methods include particle swarm optimization (PSO), cuckoo search, simulated annealing, etc. We chose to focus on PSO because of its straightforward parallelizability. Simulated annealing and cuckoo search were rejected because it is computationally prohibitive in higher dimensional spaces as it does not use any information about prior steps taken. However, PSO uses global and individual information to make directed movements in parameter space. We compare this global PSO method to three local methods: pattern search (aka coordinate descent), adaptive pattern search, and the Nelder-Mead method (aka downhill simplex method). The pattern search and Nelder-Mead methods are standard local methods and their analysis, convergence properties, and pitfalls have been widely studied.[17–23] The adaptive pattern search method is a recent improvement on the traditional pattern search.[24] These methods can be used in conjunction with some more sophisticated methods, or routines, such as multi-level single linkage, which begins with many different initial starting points, and collects them into multiple sets, which depend on whether they are sufficiently close to a previously found local optimum. If they are not close to any previously found optimum, a local search is started.[25,26] The primary use behind this type of routine is to terminate local searches, which are falling into the basin of attraction of an optimum point already found, which reduces the computation time and number of function evaluations. As an alternative to this method, one can create a global/local hybrid, where first a global method (e.g. PSO) is used to search through parameter space and when a sufficient number of particles are within a specified distance, we switch to a local method at the points with greatest particle swarm densities.

The general optimization problem of a function f can be formulated as follows

$$\mathbf{x}_{\mathrm{opt}} = \arg\min_{\mathbf{x}\in\Omega}\{f(\mathbf{x})\}, \quad f_{\mathrm{opt}} = \min_{\mathbf{x}\in\Omega}\{f(\mathbf{x})\}, \qquad (1)$$

where $\mathbf{x} \in \Omega$ is a set of parameters in parameter space Ω. $\mathbf{x}$ defines all variable properties in the system, which in turn, defines the objective function $f(\mathbf{x})$. In the case of a superconductor, the parameter set $\mathbf{x}$ can, e.g., determine the pinscape of the system and the corresponding objective function is $f(\mathbf{x}) = -J_{\mathrm{c}}(\mathbf{x})$, where $J_{\mathrm{c}}(\mathbf{x})$ is the critical current of the superconductor with pinscape $\mathbf{x}$. Note, that the minimization of this $f(\mathbf{x})$ is equivalent to the maximization of $J_{\mathrm{c}}(\mathbf{x})$.

2.1. Particle swarm optimization

The PSO algorithm is a meta-heuristic global optimization algorithm.[27,28] Its convergence properties have been studied in a simplified form, where a single particle was used and the randomness in the algorithm was replaced by its averaged value.[29] It performs well on all test problems, but is typically outshone by local methods when there was only a single minimum. The utility of this method is best seen on functions with multiple local optima, see Fig. 1(c). This test function is relevant for problems having multiple local extrema, such as the problem of periodically arranged pinning centers in superconductors. In that case, many local maxima of J_{c} exist at integer values of the ratio of number of pinning centers to number of vortices.[16]

The PSO has four main control parameters given by $\mathbf{q} = \{S,\ \omega,\ \phi_{\mathrm{p}},\ \phi_{\mathrm{g}}\}$, where S is the swarm size, ω the inertia of the individual particle (its tendency to move in its current direction), ϕ_{p} and ϕ_{g} are the weights for the particle to move towards its individual and the global optimum in parameter space, respectively. The pseudocode for PSO is presented in Listing 1.

Listing 1 Particle swarm optimization

```
input: Lower and upper limits, $\mathbf{L}$ and $\mathbf{U}$
input: Parameters $S$, $\omega$, $\phi_{\mathrm{p}}$, and $\phi_{\mathrm{g}}$
input: Parameters $K_{\mathrm{exit}}$ from $[1, S]$ and $D_{\mathrm{exit}}$
for $i = 1, \ldots, S$ do
    Uniformly distributed particle position $\mathbf{x}_i \leftarrow U_n(\mathbf{L}, \mathbf{U})$
    Particle velocity $\mathbf{v}_i \leftarrow U_n(\mathbf{L}-\mathbf{U}, \mathbf{U}-\mathbf{L})$
    Particle best known position $\mathbf{p}_i \leftarrow \mathbf{x}_i$
end for
Best global position $\mathbf{g} = \arg\min_i f(\mathbf{p}_i)$
repeat
    for $i = 1, \ldots, S$ do
        $\mathbf{r}_{\mathrm{p}} \leftarrow U_n(0,1)$,    $\mathbf{r}_{\mathrm{g}} \leftarrow U_n(0,1)$
        $\mathbf{v}_i \leftarrow \omega\mathbf{v}_i + \phi_{\mathrm{p}}\mathbf{r}_{\mathrm{p}}(\mathbf{p}_i - \mathbf{x}_i) + \phi_{\mathrm{g}}\mathbf{r}_{\mathrm{g}}(\mathbf{g} - \mathbf{x}_i)$
        $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$
        if $f(\mathbf{x}_i) < f(\mathbf{p}_i)$ then
            $\mathbf{p}_i \leftarrow \mathbf{x}_i$
            if $f(\mathbf{p}_i) < f(\mathbf{g})$ then
                $\mathbf{g} \leftarrow \mathbf{p}_i$
            end if
        end if
        Distance from global best position $d_i \leftarrow \left\| \frac{\mathbf{x}_i - \mathbf{g}}{\mathbf{U} - \mathbf{L}} \right\|_2$
    end for
    Sort $d_1, d_2, \ldots, d_S$ in ascending order
    $\bar{d} \leftarrow \frac{1}{K_{\mathrm{exit}}} \sum_{i=1}^{K_{\mathrm{exit}}} d_i$
until $\bar{d} < D_{\mathrm{exit}}$
output: $\mathbf{g}$
```

The objective function is updated independently for each particle, which makes PSO parallelizable. In this way, a large architecture can make this method highly efficient for multimodal surfaces, where a large swarm size can be way more efficient in converging towards a global solution than local methods.

The biggest challenge is in determining an appropriate exit criterion for the routine. We use

(i) the change, $|f(\mathbf{g}_{\text{best}}) - \langle f(\mathbf{g})\rangle_M|$, in the best found objective function value, $f(\mathbf{g}_{\text{best}})$, with $f(\mathbf{g})$ averaged over the last M iterations of PSO, $\langle f(\mathbf{g})\rangle_M$; or

(ii) the average distance of the particle from the swarm global best position $K_{\text{exit}}^{-1}\sum_{i=1}^{K_{\text{exit}}}\|(\mathbf{x}_i - \mathbf{g})/(\mathbf{U} - \mathbf{L})\|_2$, where the sum is taken over K_{exit} particles with the best objective values, see lines 21, 24, and 25 in Listing 1. A small variant would be to use $\mathbf{p}_i$ instead $\mathbf{x}_i$ of each respective particle, which would be less sensitive to the exploratory nature of PSO (through the inertial term $\omega\mathbf{v}_i$).

The first exit criterion avoids taking an unnecessary amount of evaluations if there is no sufficient improvement (the method could then be terminated, switched to a local method, or restarted). The second was done to avoid cases where there may be many local optimum and so a few particles get trapped among them. This avoid premature stopping if many particles are spread out in parameter space and the improvement in $\mathbf{g}$ is slow. We stop when a certain proportion of the swarm has been attracted to the best found point. A higher proportion means more evaluations but also a higher probability that we converged to the correct optimum. Therefore, there is a balance between speed and accuracy. In the testing on the benchmark functions, we found no significant difference in accuracy for $K_{\text{exit}} > S/2$, but a difference in speed. In the simulations presented below we chose $K_{\text{exit}} = 0.7S$.

2.2. Pattern search

Pattern search is a straightforward method, which starting from a random point, evaluates $2n + 1$ points (including the initial point) where n is the dimension of the parameter space, by moving a distance along each dimension in the search space. It then moves to the point which improves the function the most. The method then evaluates $2n - 1$ new points (it does not need to re-evaluate the point its on or the point it came from). If no improvement is made, the step size is reduced. Once the step size is below some threshold, it exits out of the loop. Along with its simplicity comes its ability to converge to non-stationary problems on some relatively simple problems.[30] It has a particularly difficult time with functions with coordinate systems which are highly correlated, e.g., roughly speaking, the function gradient is not along any of the main coordinate axes, such as the Rosenbrock function shown in Fig. 1(b).

2.3. Adaptive pattern search

The adaptive pattern search method is a recent modification of pattern search. The algorithm works similar to pattern search, however, while searching the parameter space, it adapts the coordinate system to achieve faster convergence, see Listing 2 adapted from Ref. 24. The presented adaptive encoding is similar to principal component analysis (PCA). The covariance matrix $\mathbf{C}$ and the transformation matrix $\mathbf{B}$ are updated using the most successful μ points. The transformation matrix modifies the coordinates to make them as uncorrelated as possible. Unlike PCA which typically looks to reduce dimensionality by retaining only eigenvectors corresponding to the largest eigenvalues, adaptive encoding retains all components.

Listing 2 Adaptive encoding

```
input: Parameters $\mu$, $\sigma$, $k_\text{s}$, and $k_\text{u}$
input: $\mu$ best points $\mathbf{x}_1, \ldots, \mathbf{x}_\mu$
if Initialize then
    $w_i \leftarrow \frac{1}{\mu}$,  $c_p \leftarrow \frac{1}{\sqrt{n}}$,  $c_1 \leftarrow \frac{1}{2n}$,  and $c_\mu \leftarrow \frac{1}{2n}$
    $\mathbf{p} \leftarrow 0$
    $\mathbf{C} \leftarrow \mathbf{I}$;   $\mathbf{B} \leftarrow \mathbf{I}$
    $m \leftarrow \sum_{i=1}^{\mu} x_i w_i$
else
    $m^- \leftarrow m$
    $m \leftarrow \sum_{i=1}^{\mu} x_i w_i$
    $z_0 \leftarrow \frac{\sqrt{n}}{\|\mathbf{B}^{-1}(m - m^-)\|}(m - m^-)$
    for $i = 1, \ldots, \mu$ do
        $z_i \leftarrow \frac{\sqrt{n}}{\|\mathbf{B}^{-1}(x_i - m^-)\|}(x_i - m^-)$
    end for
    $\mathbf{p} \leftarrow (1 - c_p)\mathbf{p} + \sqrt{c_p(2 - c_p)}z_0$
    $\mathbf{C}_\mu \leftarrow \sum_{i=1}^{\mu} w_i z_i z_i^\text{T}$
    $\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\mathbf{C} + c_1\mathbf{p}\mathbf{p}^\text{T} + c_\mu\mathbf{C}_\mu$
    $\mathbf{B}^\text{H}\mathbf{D}\mathbf{D}\mathbf{B}^\text{H} \leftarrow \texttt{eigendecomposition}(\mathbf{C})$
    $\mathbf{B} \leftarrow \mathbf{B}^\text{H}\mathbf{D}$
end if
output: $\mathbf{p}$, $\mathbf{B}$, $\mathbf{C}$
```

Apart from the adaptive encoding, adaptive pattern search is very similar to pattern search. Suppose, we have an n-dimensional optimization problem, it initially searches in each coordinate direction, independently recording the new optimal as it progresses (in contrast to the generic pattern search we employed, which is not done sequentially). Then it keeps the best μ points, $\mu < 2n + 1$, and Listing 2 is called. This transformation is updated after each sweep of all n dimensions ($2n - 1$ points) and applied in the next iteration. There are four parameters $\mathbf{q} = \{\mu, \sigma, k_\text{s}, k_\text{u}\}$ which control the success and efficiency of this method: μ is the number of points used in the adaptive encoding function call, σ is the initial step size, and k_s (k_u) are the increase (decrease) of the step size upon successful (unsuccessful) improvement of the function value. The convergence rate was observed to be most sensitive to the parameter μ. The utility of this method is best seen when applied to the Rosenbrock function, where pattern search needs over 20,000 iterations to converge in a two-dimensional parameter space, while the adaptive pattern search method typically converged in less than 1,000 iterations.

The key for the method's improved performance on Rosenbrock-type functions, is the adaptive encoding part of the algorithm. At this point we remark that the adaptive pattern search is typically a more efficient/performing method compared to the generic pattern search, but comparison to the latter offers a useful way to measure the overall shape of the optimal solution. We can logically deduce from the ratio of performance between these two methods, that if adaptive pattern search is orders of magnitude better than pattern search, then this implies that the parameters of interest are highly correlated.

2.4. Nelder-Mead method

The Nelder-Mead method (or downhill simplex) was chosen for its relative simple structure and its independence on the choice of the chosen coordinate system (it does not move along each dimension sequentially). This is most easily seen in comparing the method against the Rosenbrock function. This method utilizes simplexes, which are polytopes with $n+1$ points (or vertices) in n dimensions (i.e., triangles for $n=2$, or tetrahedrons in $n=3$).

Over time, many variants of Nelder-Mead have been conceived.[31–33] Following Ref. 33, Listing 3 describes the general Nelder-Mead method. Here we used standard values for parameters $\alpha=1$, $\gamma=2$, $\rho=1/2$, and $\sigma=1/2$. The exit criteria in line 25 is satisfied when the points are within a certain distance and is defined by the "volume"

$$\mathcal{V}(\mathbf{x}_1,\,\ldots,\,\mathbf{x}_{n+1}) = \frac{|\det(\mathbf{V})|}{n!\,\Delta^n},$$

where $\Delta = \min\|\mathbf{x}_i - \mathbf{x}_j\|$ and matrix $\mathbf{V} = (\dot{\mathbf{x}}_1,\,\ldots,\,\dot{\mathbf{x}}_n)$ consists of n vectors $\dot{\mathbf{x}}_i = \mathbf{x}_{i+1} - \mathbf{x}_1$. This choice of the exit criteria was justified in Ref. 33.

Listing 3 Nelder-Mead method

```
 1: input: Lower and upper limits, $\mathbf{L}$ and $\mathbf{U}$
 2: input: Parameters $\alpha$, $\gamma$, $\rho$, and $\sigma$
 3: Vertices $\mathbf{x}_1,\ldots\mathbf{x}_{n+1} \leftarrow U_{n+1}(\mathbf{L},\mathbf{U})$
 4: repeat
 5:     Order the vertices $f(\mathbf{x}_1) \leqslant f(\mathbf{x}_2) \leqslant \ldots \leqslant f(\mathbf{x}_{n+1})$
 6:     Centroid of $n$ best points $\mathbf{x}_0 \leftarrow \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$
 7:     Reflected point $\mathbf{x}_\mathrm{r} \leftarrow \mathbf{x}_0 + \alpha(\mathbf{x}_0 - \mathbf{x}_{n+1})$
 8:     if $f(\mathbf{x}_1) \leqslant f(\mathbf{x}_\mathrm{r}) < f(\mathbf{x}_n)$ then
 9:         $\mathbf{x}_{n+1} \leftarrow \mathbf{x}_\mathrm{r}$
10:     else if $f(\mathbf{x}_\mathrm{r}) < f(\mathbf{x}_1)$, then
11:         Expanded point $\mathbf{x}_\mathrm{e} \leftarrow \mathbf{x}_0 + \gamma(\mathbf{x}_\mathrm{r} - \mathbf{x}_0)$
12:         if $f(\mathbf{x}_\mathrm{e}) < f(\mathbf{x}_\mathrm{r})$ then
13:             $\mathbf{x}_{n+1} \leftarrow \mathbf{x}_\mathrm{e}$
14:         else
15:             $\mathbf{x}_{n+1} \leftarrow \mathbf{x}_\mathrm{r}$
16:         end if
17:     else
18:         Contracted point $\mathbf{x}_\mathrm{c} \leftarrow \mathbf{x}_0 + \rho(\mathbf{x}_{n+1} - \mathbf{x}_0)$
19:         if $f(\mathbf{x}_\mathrm{c}) < f(\mathbf{x}_{n+1})$ then
20:             $\mathbf{x}_{n+1} \leftarrow \mathbf{x}_\mathrm{c}$
21:         else
22:             $\mathbf{x}_i \leftarrow \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1)$ for all $i > 1$
23:         end if
24:     end if
25: until $\mathcal{V}(\mathbf{x}_1,\,\ldots,\,\mathbf{x}_{n+1}) < \epsilon$
26: output: $\mathbf{x}_1$
```

3. TESTING ON BENCHMARK FUNCTIONS

The above mentioned optimization methods are tested on the three benchmark functions shown in Fig. 1: the sphere function, the Rosenbrock function, and the Rastrigin function. These were chosen for their similarity to previously obtained critical-current surfaces in superconductors for low-dimensional sets of parameters. We expect scenarios where either a single optimum or multiple local with one global extrema exists. As a particular and practical example, the behavior of the solvers on the Rosenbrock function is related to the optimization of a pinscape consisting of spherical defects with two parameters being the number and diameter of the defects. In this case, it turns out that the Rosenbrock-type J_c surface can be removed by replacing the number of defects by the volume fraction. In general, however, one cannot be sure that an appropriate transformation exists or can even be found.

The study was broken up as follows: optimal parameters for the PSO and adaptive pattern search are obtained using the three benchmark functions. These were found by overlaying each respective optimization routine with PSO. For example, adaptive pattern search has tuning parameters $\mathbf{q} = \{\mu,\,k_\mathrm{s},\,k_\mathrm{u},\,\sigma\}$, see Sec. 2.3. The nested PSO algorithm then searches through this parameter space in an attempt to find the optimal parameters

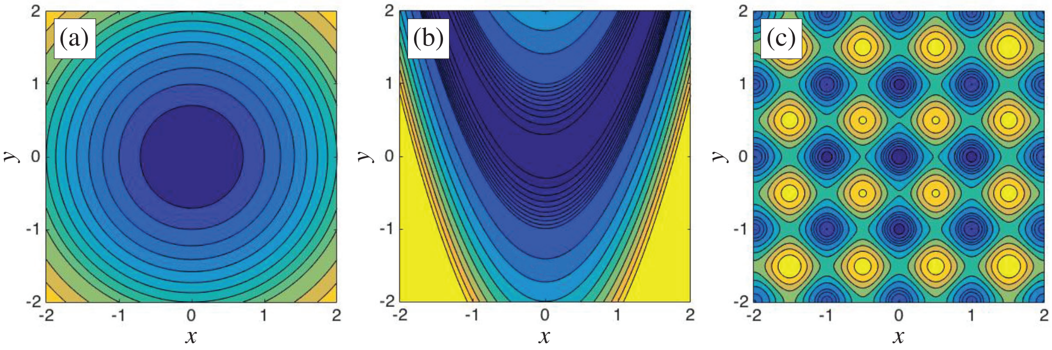

Figure 1: Examples of different test function with one global minimum. (a) Sphere function, a simple multidimensional parabola. (b) Rosenbrock function, a quartic multidimensional polynomial with shallow valley not along one of the coordinate axes. (c) Rastrigin function, a periodic function with many local extrema.

for the algorithm. We initially considered the function

$$\bar{E}_f(\mathbf{q}) = \frac{1}{M}\sum_{i=1}^{M} E_{f,i}(\mathbf{q}), \tag{2}$$

where $E_{f,i}$ is the number of function evaluations required to find the global optimum for a function f, $\mathbf{q}$ are the parameters used for the optimization routine, and M are the total number of simulations which successfully found the global optimum. However, this is not the most useful measure as it does not take into account the rate at which the algorithm successfully finds the global optimum. Indeed, defining $r_f(\mathbf{q})$ as the rate at which a correct solution is found (to within a specified tolerance) for a set of the method parameters $\mathbf{q}$. Then, it may turn out that $\bar{E}_f(\mathbf{q}_1) < \bar{E}_f(\mathbf{q}_2)$, but $r_f(\mathbf{q}_1) \ll r_f(\mathbf{q}_2)$ for some certain $\mathbf{q}_1$ and $\mathbf{q}_2$. It then may happen that we would require many more runs for $\mathbf{q}_1$ so that $\mathbf{q}_2$ was actually the better choice. Therefore, we make a refinement to Eq. (2), and obtain the optimal parameters $\mathbf{q}_{\mathrm{opt}}$ for each optimization method by solving the following auxiliary optimization problem:

$$\mathbf{q}_{\mathrm{opt}} = \arg\min_{\mathbf{q}}\{F_{f,\alpha}(\mathbf{q})\}, \tag{3a}$$

$$F_{f,\alpha}(\mathbf{q}) = N_{f,\alpha}(\mathbf{q})\bar{E}_f(\mathbf{q}) \tag{3b}$$

$$N_{f,\alpha}(\mathbf{q}) = \frac{\log(1-\alpha)}{\log[1-r_f(\mathbf{q})]}, \tag{3c}$$

where $N_{f,\alpha}$ is the number of iterations needed to be at least α sure that we have found the global solution and $\bar{E}_f(\mathbf{q})$ was defined by Eq. (2). In this work, we used $\alpha = 0.99$. The dimensionality in the function was absorbed into the optimization problem, and an analysis of the problem dimensionality and number of iterations was tested.

We sampled 10^3 different starting configurations $\mathbf{x}$, where $x_i \in [-10, 10]$ for the algorithms and then ran the nested PSO algorithm 10 times for each dimension and each benchmark function. The best parameters were recorded. Once these were obtained, we tested all the algorithms mentioned using the same starting points (when possible) and compared the performance. The algorithms each had a maximum iteration number of $10^3 n^2$ where $n = |\Omega|$ (the exception being PSO, where we used $\max(10^3 n^2, 100S)$ to ensure at least 100 possible iterations of PSO) is the dimension of the original optimization problem (1), and would exit out of the loop with a tolerance of $10^{-3} n^2$.

3.1. Sphere function

The sphere function shown in Fig. 1(a) is defined by

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2. \tag{4}$$

Tables I and II show a tabulated view of the effectiveness of the chosen methods to find the extremum of this function in the given number of iterations. The function is very simple and the coordinates are uncorrelated. Thus, a wide range of parameters actually turned out to be similarly effective. A comparison of the method's performances is presented in Fig. 2 by using 100 random initial starting configurations and employing Eq. (3).

n	$\{S$	ω	ϕ_{p}	$\phi_{\mathrm{g}}\}$	$F_{f,\alpha}$
2	5	0.22	0.93	1.93	84.7
3	5	0.36	1.35	1.68	131.3
4	8	0.23	0.80	1.96	183.7
5	10	0.18	0.99	1.96	235.5
6	9	0.36	1.55	1.55	290.5
7	12	0.27	1.18	1.75	348.4

Table I: Optimal PSO parameters $\mathbf{q}_{\mathrm{opt}} = \{S, \omega, \phi_{\mathrm{p}}, \phi_{\mathrm{g}}\}$ and $F_{f,\alpha}(\mathbf{q}_{\mathrm{opt}})$ for the sphere function.

n	$\{\mu$	σ	k_{s}	$k_{\mathrm{u}}\}$	$F_{f,\alpha}$
2	1	0.21	1.00	0.24	44.3
3	2	0.22	1.00	0.33	64.0
4	4	0.12	1.00	0.29	82.0
5	2	0.36	1.00	0.29	99.5
6	3	0.21	1.00	0.33	114.9
7	13	0.25	1.00	0.33	128.7

Table II: Optimal adaptive pattern search parameters $\mathbf{q}_{\mathrm{opt}} = \{\mu,\ \sigma,\ k_{\mathrm{s}},\ k_{\mathrm{u}}\}$ and $F_{f,\alpha}(\mathbf{q}_{\mathrm{opt}})$ for the sphere function.

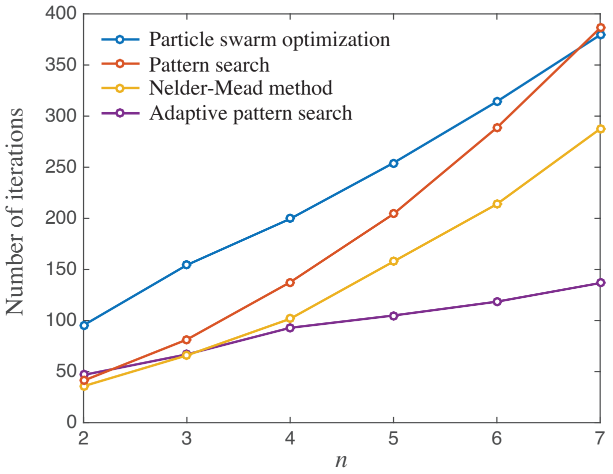

Figure 2: A comparison of each methods efficiency as a function of dimension for the sphere function.

3.2. Rosenbrock function

The Rosenbrock function shown in Fig. 1(b) is a standard test for optimization methods and is given by

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right]. \qquad (5)$$

Optimization of this function demonstrates the utility of coordinate independent local search methods (like Nelder-Mead or adaptive pattern search). This is due to the fact that the minimum is contained inside a parabolic valley which requires constant shrinking of the step size for pattern search to make progress. Consider the 2D case

$$f(x,y) = 100(y - x^2)^2 + (1 - x)^2.$$

A change of variables to the (u,v) plane given by $u = 1 - x$ and $v = y - x^2$ leads to elliptical level sets $g(u,v) = u^2 + 100v^2$, which is much more favorable to coordinate-dependent methods. In practice it is usually difficult or impossible to find the appropriate transform converting to elliptical level sets. Nevertheless, this gives a useful test of the morphology of the surface of J_{c} by comparing iterations between coordinate-dependent and coordinate-independent methods.

Tables III and IV show the effectiveness of the PSO and adaptive pattern search methods to minimize the Rosenbrock function in a given number of iterations. As we can see from the table, the PSO method is much slower than Nelder-Mead and adaptive pattern search. The optimization of parameters for PSO have revealed that the dimensionality and swarm size are (perhaps not surprisingly) correlated. As the dimensionality increases, the optimal swarm size (holding other parameters fixed) increases. To verify this, we sampled 100 random starting points for swarm sizes between 10–200; the results are shown in Fig. 4.

n	$\{S$	ω	ϕ_{p}	$\phi_{\mathrm{g}}\}$	$F_{f,\alpha}$	r_f	$N_{f,\alpha}$
2	28	0.25	−0.21	1.58	823	0.99	1
3	34	0.29	−0.15	1.73	2942	0.95	2
4	40	0.35	−0.17	1.59	9352	0.73	4
5	49	0.28	−0.21	1.67	15650	0.61	5
6	61	0.30	−0.20	1.70	22339	0.56	6
7	73	0.27	−0.19	1.68	39277	0.53	7

Table III: Optimal PSO parameters $\mathbf{q}_{\mathrm{opt}} = \{S,\ \omega,\ \phi_{\mathrm{p}},\ \phi_{\mathrm{g}}\}$, $F_{f,\alpha}(\mathbf{q}_{\mathrm{opt}})$, $r_f(\mathbf{q}_{\mathrm{opt}})$, and $N_{f,\alpha}(\mathbf{q}_{\mathrm{opt}})$ for the Rosenbrock function.

n	$\{\mu$	σ	k_{s}	$k_{\mathrm{u}}\}$	$F_{f,\alpha}$	r_f	$N_{f,\alpha}$
2	2	0.28	1.67	0.43	227	1.00	1
3	3	0.15	1.48	0.46	392	1.00	1
4	4	0.22	1.81	0.38	781	0.94	2
5	5	0.14	1.71	0.45	1384	0.92	2
6	6	0.37	1.96	0.36	1775	0.93	2
7	7	0.10	2.77	0.34	2993	0.92	2

Table IV: Optimal adaptive pattern search parameters $\mathbf{q}_{\mathrm{opt}} = \{\mu,\ \sigma,\ k_{\mathrm{s}},\ k_{\mathrm{u}}\}$, $F_{f,\alpha}(\mathbf{q}_{\mathrm{opt}})$, $r_f(\mathbf{q}_{\mathrm{opt}})$, and $N_{f,\alpha}(\mathbf{q}_{\mathrm{opt}})$ for the Rosenbrock function.

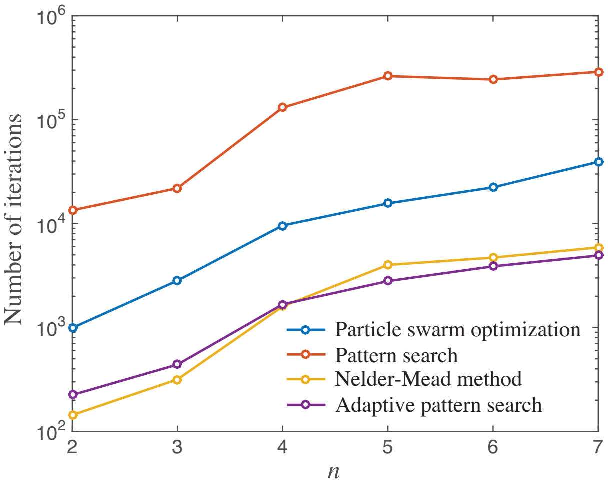

Figure 3: A comparison of each methods efficiency as a function of dimension for the Rosenbrock function.

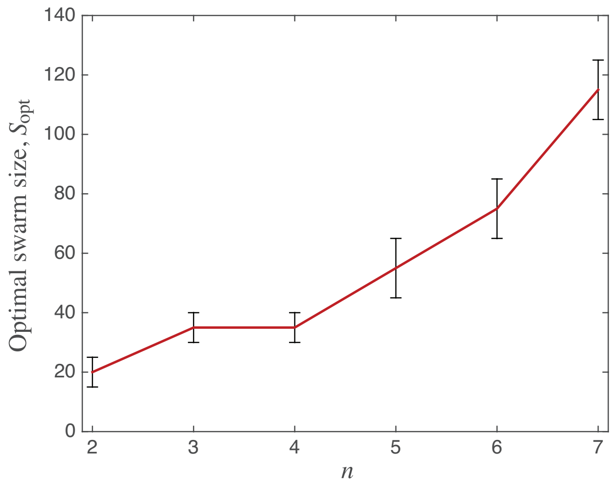

Figure 4: The optimal swarm size, S, of the particle swam optimization on the Rosenbrock function as a function of dimension, n while holding the other parameters fixed $\{\omega, \phi_{\rm p}, \phi_{\rm g}\} = \{0.3, -0.2, 1.7\}$.

3.3. Rastrigin function

The third test function we consider here, is the Rastrigin function [see Fig. 1(c)], which is the sum of a sphere function and periodic oscillations, having multiple minima. This type of objective function is similar to the objective function of superconductors having periodic pinscapes. It is defined by

$$f(\mathbf{x}) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i)\right]. \qquad (6)$$

This is where PSO performs particularly well compared to the local search methods. Table V shows its effectiveness to solve the Rastrigin function optimization problem in the given number of iterations. The local methods do a comparatively poor job as can be seen in Table VI and Fig. 5. The optimal swarm size also grows rapidly with increasing dimensionality and we sampled 100 random starting points for swarm sizes between 50 and 1,000; the results are shown in Fig. 6.

n	$\{S$	ω	$\phi_{\rm p}$	$\phi_{\rm g}\}$	$F_{f,\alpha}$	r_f	$N_{f,\alpha}$
2	50	0.25	2.00	1.00	934	1.00	1
3	145	0.58	2.08	0.89	4514	1.00	1
4	265	0.58	2.21	0.75	10953	1.00	1
5	365	0.48	2.22	0.77	20410	1.00	1
6	650	0.5	2.25	0.67	43939	0.998	1
7	945	0.51	2.29	0.65	81779	0.993	1

Table V: Optimal PSO parameters $\mathbf{q}_{\rm opt} = \{S, \omega, \phi_{\rm p}, \phi_{\rm g}\}$, $F_{f,\alpha}(\mathbf{q}_{\rm opt})$, $r_f(\mathbf{q}_{\rm opt})$, and $N_{f,\alpha}(\mathbf{q}_{\rm opt})$ for Rastrigin function.

n	$\{\mu$	σ	$k_{\rm s}$	$k_{\rm u}\}$	$F_{f,\alpha}$	r_f	$N_{f,\alpha}$
2	3	0.25	1.79	0.33	1051	0.22	19
3	1	0.37	1.94	0.31	7972	0.04	113
4	1	0.39	1.69	0.29	27017	0.01	327
5	2	0.62	1.00	0.45	132134	0.004	1149
6	6	0.3	6.59	0.29	197052	0.003	1532
7	7	0.41	5.93	0.11	584564	0.001	4603

Table VI: Optimal adaptive search parameters $\mathbf{q}_{\rm opt} = \{S, \omega, \phi_{\rm p}, \phi_{\rm g}\}$, $F_{f,\alpha}(\mathbf{q}_{\rm opt})$, $r_f(\mathbf{q}_{\rm opt})$, and $N_{f,\alpha}(\mathbf{q}_{\rm opt})$ for Rastrigin function.

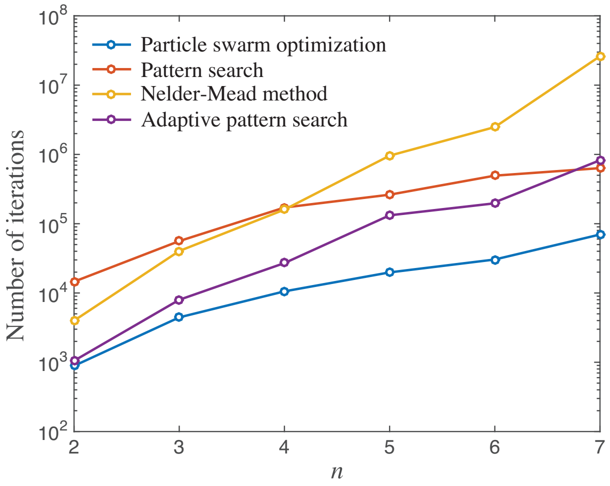

Figure 5: A comparison of each methods efficiency as a function of dimension for the Rastrigin function.

3.4. Results of benchmark function optimizations

The optimization of parameters for adaptive pattern search and PSO has led to some interesting results. First, the nested PSO algorithm finds many different "good" choices for the parameters needed. This is to be expected, since each problem is unique, and it would be surprising if multiple sets of parameters did not yield similar results. What was surprising, is the sensitivity of the parameters to small deviations from optima that were obtained on Rastrigin's function. A good example would be in a particular optimization run of adaptive pattern search's parameters for $n = 6$. We found that the set $\{\mu, \sigma, k_{\rm s}, k_{\rm u}\} = \{6, 0.3, 6.59, 0.29\}$ was the best for that particular run found with $2 \cdot 10^5$ function evaluations needed. Just changing one of these parameters slightly to $\{6, 0.3, 6.59, 0.39\}$, needs $4.25 \cdot 10^5$ function evaluations.

This sensitivity was not observed in the other two benchmark functions, meaning that the sensitivity of these parameters for the optimization routines are highly dependent on the type of function, being particularly sensitive to functions with multiple optima. At the other end of this spectrum, it was observed that many different sets of parameter values led to a minimal amount of evaluations. Essentially, there was not a large difference between the best observed set, which is reported in

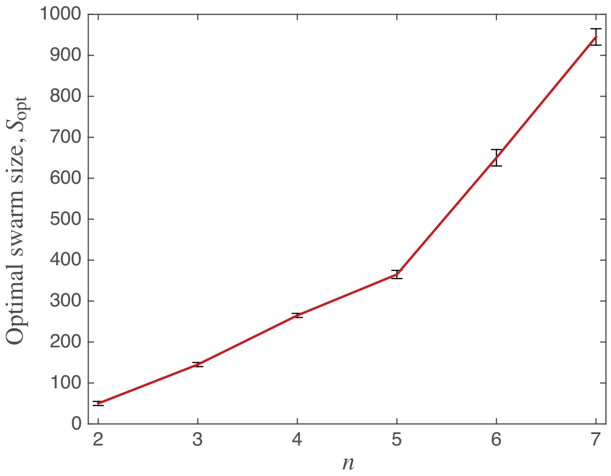

Figure 6: The optimal swarm size, S, of the particle swam optimization on the Rastrigin function as a function of dimension, n while holding the other parameters fixed (ω, ϕ_{p}, ϕ_{g}) = (0.4, 2.0, 1.0).

Tables I and II, and other choices. From this we can conclude that surfaces with uncorrelated dimensional space and with small number of optima are more likely to be insensitive to the choice of parameters and hence do not need to be tuned for the particular problem.

The Rastrigin function is particularly useful in analyzing the effect multiple optima have on local methods. One can find the number of extrema as a function of dimension size n. The derivative of the Rastrigin function in each respective coordinate is set to 0 and is given by $x_i + 10\pi \sin(2\pi x_i) = 0$. We recall that the bounds for our benchmark functions are $x_i \in [-10, 10]$. It is easy to see that due to symmetry, we only need to consider $x_i > 0$. We further note that solutions are not possible for $x_i > 10\pi$ and for each oscillation period, one obtains two extrema. Since the period is 1, this implies 10 minima (20 extremum which by symmetry has 10 maxima and 10 minima) within the above bounds. Furthermore, symmetry implies 10 more in the negative direction plus the obvious $x_i = 0$ (global) minimum, which results in 21 minima total. It then follows that the number of minima in our problem for n dimensions is $20^n + 1 \approx 20^n$.

Another useful metric of an optimization routine is the probability that a found optimum is the global optimum. For this we can consider any local method which is run R times, and suppose the method has an equal probability to converge to any of the m minima (in general this need not be true). We also assume $m \gg 1$, and we want to be $\alpha = 0.99$ sure we have found the global solution. This would require $R \approx 4.6m$ [or $R = O(m)$]. Thus the number of local runs should scale approximately linearly with increasing minima. With just $n = 2$ for the Rastrigin function for example would imply that a local method would need to be run $R \approx 2000$ times to be 99% sure we have converged to the global solution. Again, it is important to note that this number assumed each extremum was equally likely, that is, each of their respective basins of attractions were the same size. This assumption in general is not true, but it provides an insight into how many runs of a local method would be needed to be confident that the routine has found the global optimum. For example, suppose a local method converges in 50 evaluations on average to a local extremum. In two dimensions this would require on average close to 100000 total evaluations of the objective function. Compare this with the number needed for PSO. In reality this turns out to be an overestimation, the total number of evaluations for local methods are far lower, but this analysis provides some insight as to why they suffer at this level and how the number of evaluations will scale with increasing number of optimums.

In contrast, PSO and other global methods are more robust to the number of local optima as can be seen by comparing Figs. 3 and 5, the number of function evaluations is on the same order for both Rosenbrock and Rastrigin functions. A possible reason why PSO behaves similarly for Rosenbrock and Rastrigin is because there is only a ”1D path” towards the minimum for Rosenbrock, while the sphere function has a 50% chance to lower the objective in each direction. Therefore, particles are as likely to become stuck in the Rosenbrock valley as they are in a Rastrigin local minimum.

The PSO optimal parameters also provide some insight into the nature of the function. It helps to first interpret what each parameter does. Increasing S obviously leads to a greater chance of finding the best solution, the cost is an increase in function evaluations. The inertial weight $\omega < 1$ is not a coincidence. As the particles approach the global best, $\mathbf{v}_i \leftarrow \omega \mathbf{v}_i$ and this has a fixed point at $\mathbf{v}_i = 0$. The ϕ_{p} and ϕ_{g} are quantities which affect the exploratory nature of the algorithm and the weight associated with the global information that the swarm provides to each individual particle. As $\phi_{\mathrm{g}} \to 0$ each of the particles behave more independently and their movement becomes more individualized. As $\phi_{\mathrm{p}} \to 0$, one essentially tells each particle to only trust the group on the correct direction to go.

Armed with this interpretation, one can understand some of the results from the tables in Sec. 3. The sphere function and Rosenbrock both have few minima (sphere has only one, and the Rosenbrock has no more than two for $n \leqslant 7$). Therefore we expect $\phi_{\mathrm{g}} \geqslant \phi_{\mathrm{p}}$. However, the results are different in Table V, which shows ϕ_{p} increasing and ϕ_{g} decreasing as n (number of optima) increases. This makes sense for multiple optima, since the swarm should behave more independently and not get caught among the many local optima. On the other hand for $\phi_{\mathrm{g}} = 0$ it would take very long to converge (e.g. simulated annealing).

4. MODEL FOR THE CRITICAL CURRENT IN SUPERCONDUCTORS

Here we define the objective function for the optimization problem in superconductors mentioned before. For the description of the vortex dynamics in strong type-II superconductors, we use the TDGL equation for the superconducting order parameter $\psi = \psi(\mathbf{r}, t)$,

$$(\partial_t + i\mu)\psi = \epsilon(\mathbf{r})\psi - |\psi|^2\psi + (\nabla - i\mathbf{A})^2\psi + \zeta(\mathbf{r}, t), \quad (7)$$

which is solved numerically. Here $\mu = \mu(\mathbf{r}, t)$ is the chemical potential and $\mathbf{A}$ is the vector potential associated with an external magnetic field $\mathbf{B}$ as $\mathbf{B} = \nabla \times \mathbf{A}$. The temperature-induced noise is simulated using a δ-correlated Langevin term $\zeta(\mathbf{r}, t)$ (in this work we use noise of approximately 1 K). The unit of length is given by the superconducting coherence length ξ, the unit of magnetic field by the upper critical field H_{c2}, and the corresponding unit of current is J_0 (in this unit, the depairing current is given by $J_{\mathrm{dp}} = 2/\sqrt{27}$). See Ref. 12 for the details of TDGL model implementation and physical units. The current density is given by the expression

$$\mathbf{J} = \mathrm{Im}\big[\psi^*(\nabla - i\mathbf{A})\psi\big] - \nabla\mu. \quad (8)$$

To determine the critical current value — the maximal current, which can flow through the superconductor without dissipation, — we use a finite-electrical-field criterion. Specifically, we choose a certain small external electric field, $E_{\mathrm{c}} = 10^{-4}$, which measures the dissipation and adjust the applied external current, J, to reach this electrical-field/dissipation level on average during the simulation. The time-averaged value of external current over a steady state gives the critical current, $J_{\mathrm{c}} = \langle J \rangle$.

The critical current in the presence of an external magnetic field is mostly defined by the pattern of non-superconducting defects, including their sizes, shapes, and spatial distribution, which prevents vortices from moving under the influence of the Lorentz force, $\mathbf{f}_{\mathrm{L}} = \mathbf{J} \times \mathbf{B}$.

The pinscape is characterized by a set of parameters $\mathbf{x}$, which corresponds to the objective function

$$f(\mathbf{x}) = -J_{\mathrm{c}}(\mathbf{x}) \quad (9)$$

used in optimization problem (1). Each element $\mathbf{x}$ of the parameter space Ω describes the pinscape in the superconductor, e.g. the shape of each defect and their spatial distribution. The optimal configuration of the defects $\mathbf{x}_{\mathrm{opt}}$ corresponds to the minimization of the objective function, $f_{\mathrm{opt}} = f(\mathbf{x}_{\mathrm{opt}})$.

Knowledge of the shape and behavior of the function $J_{\mathrm{c}}(\mathbf{x})$ is not known *a priori*. In addition if we consider, for example, a random placement of defects in the domain, each realization can yield slightly different values for J_{c} — for the same $\mathbf{x}$ (if it does not describe the positions of all defects explicitly), such that averaging is required. In that case one can expect, as the number of random simulations tends to infinity, J_{c} approaches a “true” value due to self-averaging. For a finite number of trials it leads, however, to a noisy J_{c} surface. This can create difficulties for local methods to converge to the global solution. A modification of the local methods to multi-level starting points, helps to overcome the noise in these types of problems.

To obtain the critical current $J_{\mathrm{c}}(\mathbf{x})$, we solve the TDGL equation in the domain of interest with a specified parameter set $\mathbf{x}$. Each evaluation of J_{c} is relatively expensive (typically, a few GPU-hours), but can be performed independently for different $\mathbf{x}$. In order to reduce noise in J_{c}, one can (i) average it over several realizations of random positions of defects and/or (ii) increase the system size. Both techniques naturally increase the computation time of J_{c}. The final value defines the objective function for the optimization problem.

5. OPTIMIZATION OF THE CRITICAL CURRENT

The objective function in Eq. (9) is then used to solve the general optimization problem for the pinscape defined in Eq. (1). Typical pinscapes with like defects are described by parameter spaces Ω having $n = 2$ to 8 dimensions. Here we consider three particular and important cases for $n = 2$, 3, 4 to analyze the described optimization strategies. For practical applications also the *robustness* of an optimum is important, since e.g. the size and shape of defects cannot be controlled exactly in an experiment. Therefore, the found optimal critical current should be relatively insensitive to small perturbations in the optimal pinscape $\mathbf{x}_{\mathrm{opt}}$. In the examples studied here, this condition is fulfilled.

For the simulations needed to obtain the critical current for a given pinscape, we use a three-dimensional superconducting sample with dimensions $64 \times 64 \times 64$ in units of coherence length ξ with (quasi-)periodic boundary conditions. The external current J is applied in the x-direction, perpendicular to the magnetic field B in the z-direction. All defects are modeled by local variation of the critical temperature, such that defects are regions being in the normal state. To this end the system is discretized by a regular cubic grid with grid resolution of half a coherence length. As demonstrated in earlier works,[8,9] this resolution is sufficient to capture the involved physical processes correctly. A pattern generator then creates the pinscape according to the defining parameter set $\mathbf{x}$, by assigning each grid point either to the superconductor or to the defect, distinguished by a high or low local T_{c} value. Defect regions with low T_{c} values typically consist of many connected grid point.

First, we consider randomly placed spherical defects (typically self-assembled inclusions in real sample) in a three-dimensional system. All inclusions have the same diameter, such that the parameter space for this prob-

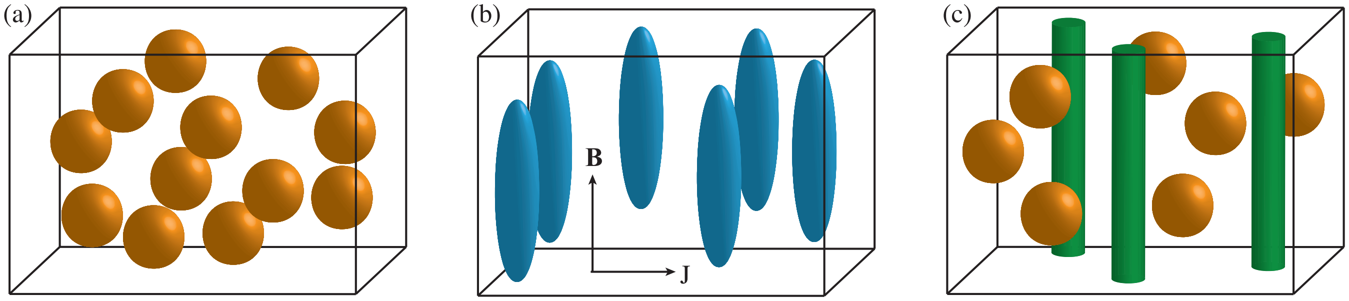

Figure 7: Pinscapes for optimization. (a) Randomly placed spherical inclusions for problem (10a). (b) Spheroidal inclusions for problem (10b). (c) Mixture of spherical and columnar inclusions for problem (10c).

lem is defined by the defect density ρ_s within the sample and their diameter D_s. As mentioned before, this set of parameters is not unique, such that the same pinscape could be described by a different $\mathbf{x}$ (e.g., the diameter and total number of defects). This two-parameter pinscape problem was considered recently in Ref. 34 by sampling of the parameter space, which was possible in that case.

Another example of a two-parameter problem would be a system with two kinds of defects having fixed shape, e.g., a superconductor having columnar defects of fixed diameter imprinted (typically introduced by high-energy heavy-ion irradiation in a real sample), which already had intrinsic (usually chemically grown) nano-rod pinning centers.[9] In this case the parameter space will be defined by the concentration ρ_c of columnar defects and the concentration ρ_n of the intrinsic nano-rod defects, i.e., $\mathbf{x} = (\rho_\mathrm{c}, \rho_\mathrm{n})$.

For the first problem we consider here, the pinscape is defined by

$$\mathbf{x} = (\rho_\mathrm{s}, D_\mathrm{s}), \qquad N_\mathrm{s} = \left\lfloor \frac{6 L_x L_y L_z \rho_\mathrm{s}}{\pi D_\mathrm{s}^3} \right\rceil, \tag{10a}$$

where N_s is the number of identical spherical inclusions to be placed randomly [see Fig. 7(a)] and $\lfloor\cdot\rceil$ denotes the integer rounding function. In the second, three-parameter (3D) optimization problem shown in Fig. 7(b) we replace the spherical defects by spheroidal defects, which have a different diameter in z-direction than in the xy-plane, i.e., it is described by two different diameters — D_{xy} in the xy plane and D_z in z direction. In this case the pinscape is defined by

$$\mathbf{x} = (\rho_\mathrm{e}, D_{xy}, D_z), \qquad N_\mathrm{e} = \left\lfloor \frac{4 L_x L_y L_z \rho_\mathrm{e}}{\pi \kappa D_{xy}^2 \tilde{D}_z} \right\rceil, \tag{10b}$$

where $\tilde{D}_z = \min\{D_z, L_z\}$ and $\kappa = 1 - \tilde{D}_z^2/3D_z^2$ — the former definitions take into account periodic boundary conditions in z-direction, when the z-diameter grows beyond the simulation cuboid. In the final, four-parameter (4D) problem, we consider two types of defects described by a single parameter and their respective volume fractions, namely spherical and columnar (cylindrical) inclusions, see Fig. 7(c). The respective pinscape control parameters are then ρ_s, D_s for spheres and ρ_c, D_c for columns, i.e.,

$$\mathbf{x} = (\rho_\mathrm{s}, D_\mathrm{s}, \rho_\mathrm{c}, D_\mathrm{c}), \qquad N_\mathrm{c} = \left\lfloor \frac{4 L_x L_y \rho_\mathrm{s}}{\pi D_{xy}^2} \right\rceil, \tag{10c}$$

where N_s was defined in Eq. (10a). In the following we discuss the results for these three optimization problems as well as the benchmark function optimizations.

6. RESULTS OF CRITICAL CURRENT OPTIMIZATIONS

Figures 8, 9, and 10 show the optimization "paths" for all four optimization methods for the 2D [Eq. (10a) with parameter space dimension $n = 2$], 3D [Eq. (10b) with $n = 3$], and 4D [Eq. (10c) with $n = 4$] problems, respectively. These plots show the values of the currently best critical current and the associated parameter set $\mathbf{x}$ as function of number of evaluations of the objective function (solid lines). If a new optimal value is found it is marked by circles, while all other evaluations (with objective values above the current minimum) are marked by gray dots. The optimal values are shown by dashed horizontal lines. Note, that the pattern search algorithm is not sequential, meaning that a minimum is only calculated after a certain number of evaluations (up to $2n+1$), which defines an iteration of the method. The iterations are marked by vertical dotted lines in the plots. If the number of evaluations per iteration is less that $2n+1$, the remaining objective values were calculated before and are just looked up in a database. Iterations are shaded according to the step sizes in parameter space. The optimization results are summarized in Table VII. In the 2D case for randomly placed spherical inclusions, the optimal values reproduce the ones of Ref. 34 at magnetic field $B = 0.1 H_{\mathrm{c}2}$. In the 3D case for spheroidal inclusions, the optimal pinscape corresponds to an infinite size in z direction (larger then the system size), i.e., when the spheroids have "evolved" to columnar inclusion, and the optimal value of the critical current is increased by $\sim 50\%$. In the 4D case, for a combination of the columnar

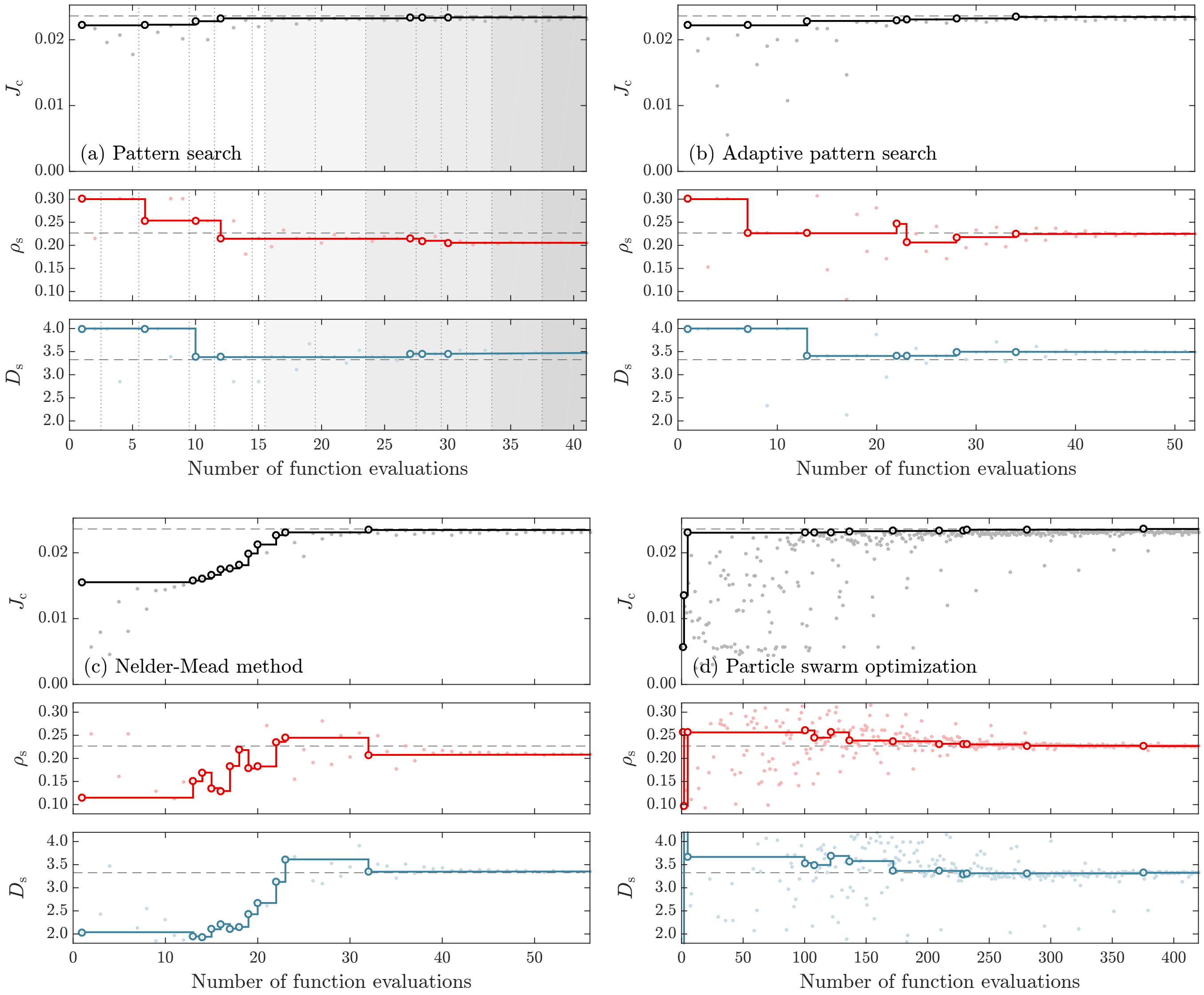

Figure 8: Optimization procedure for the two-parameter (2D) problem (10a) with monodisperse spherical defects characterized by the volume fraction ρ_s occupied by them and their diameter D_s. Critical current J_c and optimization parameters are shown as a function of number of objective function evaluations for (a) pattern search, (b) adaptive pattern search, (c) Nelder-Mead method, and (d) particle swarm optimization. The 2D optimization problem has an objective function close to a sphere-type (4).[34] Thus one can expect that all methods converge to the same optimum marked by horizontal dashed line. However, due to sample fluctuations there are still visible differences in the optimal parameters which need much longer times to resolve. Vertical dotted lines in panel (a) separate different iterations; darker background color corresponds to smaller step size. PSO exit criterion causes over 300 additional evaluations for marginal improvement.

and spherical defects, the 3D result is confirmed, since the volume fraction for the spheres vanish and the best pinscape turns out to be consisting of columnar defects in z direction only. The diameter and volume fraction of the best columnar defects in the 3D and 4D cases is accordingly the same.

As mentioned before, in order to obtain a good critical current value for randomly placed inclusions, either a large number of random realizations or much larger systems are needed. While the latter is easier to run, we chose the former for performance reasons, such that a single objective function evaluation requires averaging the critical current over many pinscapes, described by the same $\mathbf{x}$. However, even with a large number of realizations, the critical current as a function of $\mathbf{x}$ can still be quite noisy. In the worst case this leads to local methods failing to reach the optimal solution. In contrast to that, global methods like PSO are much more resistant to noise in the objective function.

First, let us analyze the two parameter optimization more closely (see Fig. 8). All optimization methods converge to the same optimal pinscape. However, the opti-

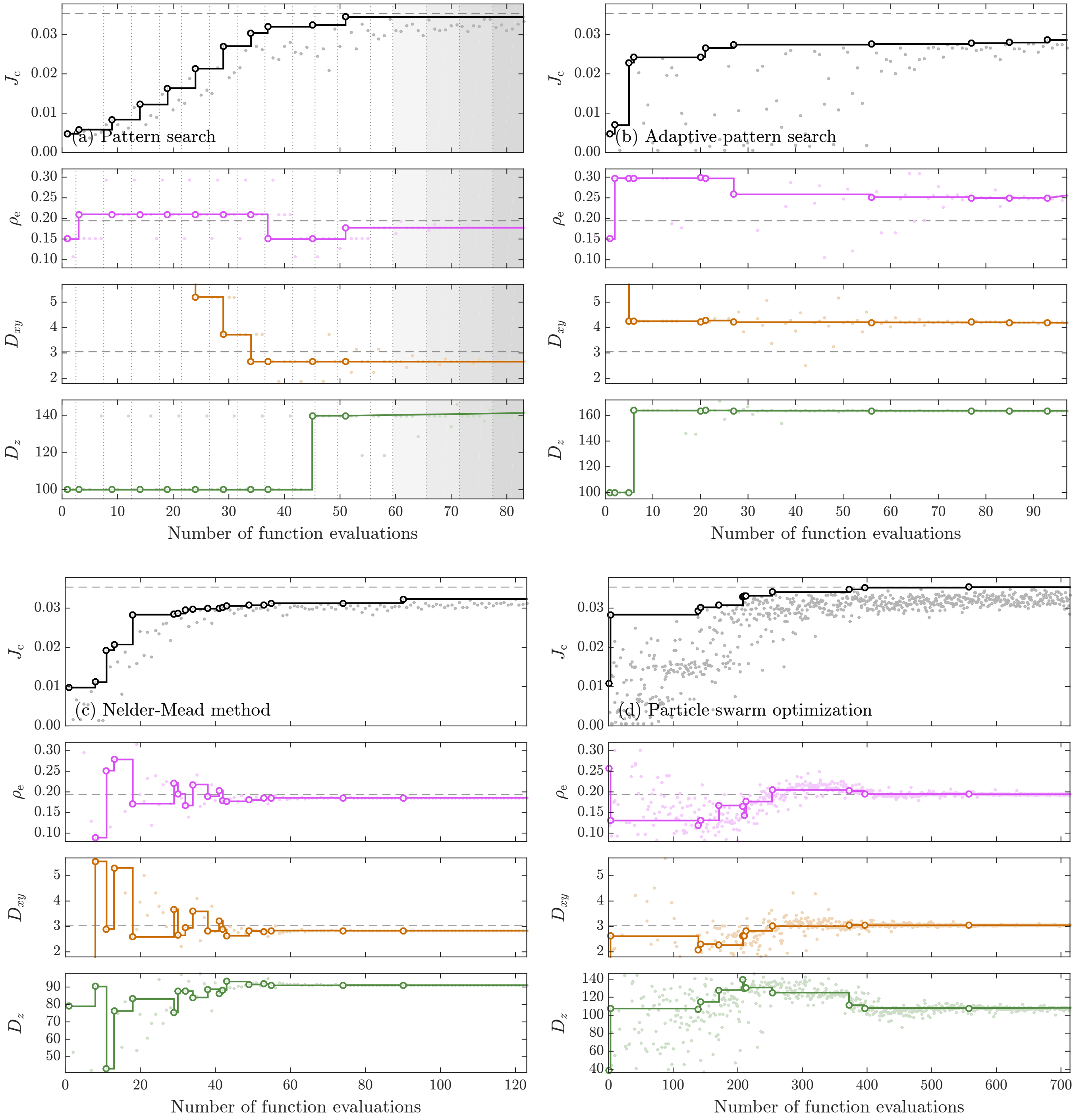

Figure 9: Optimization procedure for the three-parameter (3D) problem (10b) with spheroidal defects characterized by the occupied volume fraction ρ_e as well as defect diameter in D_{xy} in xy plane along applied current and diameter in D_z in z direction along applied magnetic field. Critical current J_c and optimization parameters are shown as a function of number of objective function evaluations for (a) pattern search, (b) adaptive pattern search, (c) Nelder-Mead method, and (d) particle swarm optimization. Optimal diameter in z direction is larger then system size $L_z = 64$ for all methods, which indicates that in the bulk sample optimal z-diameter D_z is infinite and spheroidal defects transform into columnar defects along z-axis.

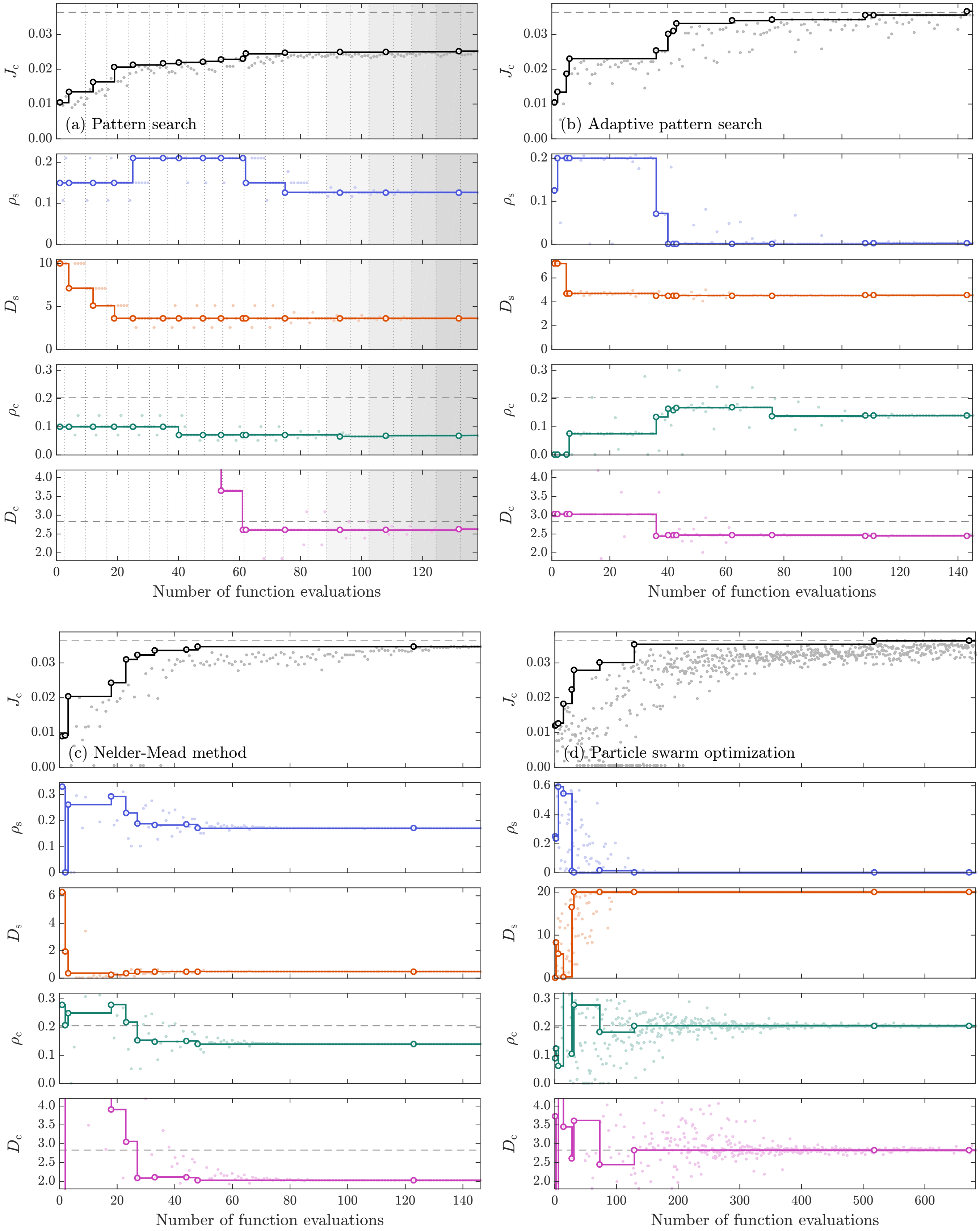

Figure 10: Optimization procedure for the four-parameter (4D) problem (10c) with mixed pinscape containing spherical particles (characterized by volume fraction ρ_s and diameter in D_s) columnar defects along z axis (volume fraction ρ_c and diameter in D_c). Critical current J_c and optimization parameters are shown as a function of number of objective function evaluations for (a) pattern search, (b) adaptive pattern search, (c) Nelder-Mead method, and (d) particle swarm optimization. PSO and adaptive pattern search converge to the pinscape containing columnar defects only.

n	$\mathbf{x} = \{\rho_\mathrm{s}$	$D_\mathrm{s}\}$	$J_\mathrm{c}\,[J_0]$	$J_\mathrm{c}\,[J_\mathrm{dp}]$
2	0.22	3.5	0.0235	0.061

n	$\mathbf{x} = \{\rho$	D_{xy}	$D_z\}$	$J_\mathrm{c}\,[J_0]$	$J_\mathrm{c}\,[J_\mathrm{dp}]$
3	0.20	3.0	∞	0.035	0.091

n	$\mathbf{x} = \{\rho_\mathrm{s}$	D_s	ρ_c	$D_\mathrm{c}\}$	$J_\mathrm{c}\,[J_0]$	$J_\mathrm{c}\,[J_\mathrm{dp}]$
4	0.00	—	0.20	3.0	0.035	0.091

Table VII: Optimal parameters of the pinscape for maximal critical current in the superconductor.

mization "paths" are quite different as one could expect: although all methods reach a critical current value close to the optimum after about 10–20 evaluations — indicating a robust, relatively flat peak in parameter space, the actual optimal configuration takes quite a while to be reached: It takes about twice as many evaluations for the local methods and an order of magnitude more for the global PSO method. Note, that defining an appropriate exit criterion for the optimization can be challenging, since the improvement in the critical current can be marginal, while the configuration can still change noticeably, as is clearly seen in the Nelder-Mead path.

Increasing the parameter space to three dimensions, (Fig. 9), shows that all methods need more evaluations as expected (scaling roughly linear with the dimension) and again the local methods reach a close to optimal critical current significantly faster than PSO. However, one can see that the local methods already start to have convergence problems, which is most likely a result of the noise in the objective function. The figure shows that the adaptive pattern search is a bit more efficient dealing with this noise than simple coordinate decent. However, in this case all methods still reach a configuration close to the optimal one.

An obvious observation is that increasing the parameter space, cannot lower the optimal value of the objective function. Indeed, the chosen examples are in a sense supersets of each other with higher dimensionality of Ω, such that the best J_c value either increases or remains the same. When going over from the 3D to 4D optimization problem, we kept the result for the optimal z-diameter of the defects, i.e., that the optimal defects are cylinders, but added back spherical defects to the system. In the 4D case one could then imagine three different scenarios for the global optimal configuration: coexistence of both defect types, only cylindrical defects remain, or only spherical are optimal. Due to the results of the 3D problem, the latter is rather unlikely, since vortices and columnar defects are well aligned and the pinning potential of the defects are therefore "optimal."

As seen from Fig. 10, the local methods converge to different optimal solutions. The coexistence scenario appears to be a good pinscape. However, the largest J_c value is still below the cylinder-only solution. One can expect that the optimal configuration is dependent on the angle between applied magnetic field and the main axis of the cylindrical defects. If they deviate from the parallel setup we studied here, the coexistence of spherical and columnar defects might be optimal, but it is clear that at a 90° angle between field and columns only spherical defects would "survive" in an optimization problem (for isotropic superconductors).

When we compare our results for the J_c optimization to the studied test functions before, we can conclude, that the J_c surface for the chosen parameters is roughly uncorrelated, since the number of evaluations required for adaptive pattern search and pattern search to converge are nearly the same.

Overall, we see that the PSO algorithm works reliable in all cases, at the cost of about 10 times more evaluations needed to reach the optimum, where often many evaluations, on the order 10^2–10^3, are yielding only a marginal gain in the objective function ($< 1\%$). The local methods start to fail in higher dimensional parameter spaces, most likely due to noise in the objective function. As mentioned before, combining the local searches with a global method can potentially reduce the overall number of evaluations. Another way to mitigate this problem is to use more realizations or larger system sizes, which should smooth out the error from the random placement of defects.

Another benefit of using a global method such as PSO, is that it consistently finds better solution than the local methods. In higher dimensional spaces $n \gtrsim 4$, taking a larger swarm size S becomes necessary to ensure convergence. The local methods, however, converge faster. But in any case there is no guarantee that the found configuration is the optimum. One could only estimate a probability that the found result is close to the optimal one.

Finally, we remark that larger oscillations in the optimization paths for the local methods in higher dimensions suggest a flatter global maximum of J_c or a more robust critical current when one deviates from the optimal configuring. This is particularly important for practical applications.

7. CONCLUSIONS

We have tested various methods for optimization problems and determined that particle swarm optimization and adaptive pattern search performed the best on the benchmark functions as well as in the physical optimization problem presented. Particle swarm optimization in particular does a relatively good job at handling the noisy surface of the objective function, but takes a long time to converge, where a significant amount of time is used for marginal improvement. This is a result of the difficulty to define an exit criterion. On the other side, adaptive pattern search can get caught in local extrema, which are either physical or a result of the noise. However, the number of evaluations are often much less (on the order of 7–10 times). As mentioned before, a multi-level local method could be used to mitigate this, which should make adaptive pattern search competitive with particle swarm optimization. Although we presented only some results for the pinscape optimization in superconductors, our studies of different optimization strategies can be important for a variety of different physical systems where the evaluation of the objective function is expensive, e.g., requiring first principle calculations or molecular dynamics simulations.

Acknowledgments

This work was supported by the Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Basic Energy Science, Division of Materials Science and Engineering. For the computational work we used resources of the Argonne Leadership Computing Facility (DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357) and the GPU cluster GAEA at Northern Illinois University.

[1] W.-K. Kwok, U. Welp, A. Glatz, A. E. Koshelev, K. J. Kihlstrom, and G. W. Crabtree, *Vortices in high-performance high-temperature superconductors*, Rep. Prog. Phys. **79**, 116501 (2016).

[2] T. G. Holesinger, L. Civale, B. Maiorov, D. M. Feldmann, J. Y. Coulter, D. J. Miller, V. A. Maroni, Z. Chen, D. C. Larbalestier, R. Feenstra, *et al.*, *Progress in nanoengineered microstructures for tunable high-current, high-temperature superconducting wires*, Adv. Mat. **20**, 391 (2008).

[3] A. Malozemoff, *Second-generation high-temperature superconductor wires for the electric power grid*, Annu. Rev. Mater. Res. **42**, 373 (2012).

[4] A. A. Abrikosov, *The magnetic properties of superconducting alloys*, J. Phys. Chem. Solids **2**, 199 (1957).

[5] G. Blatter, M. V. Feigel'man, V. B. Geshkenbein, A. I. Larkin, and V. M. Vinokur, *Vortices in high-temperature superconductors*, Rev. Mod. Phys. **66**, 1125 (1994).

[6] G. Blatter and V. Geshkenbein, *Vortex matter*, in The physics of superconductors, edited by K. Bennemann and J. Ketterson (Springer Berlin Heidelberg, 2003) pp. 725–936.

[7] V. Ortalan, M. Herrera, M. W. Rupich, and N. D. Browning, *Three dimensional analyses of flux pinning centers in Dy-doped $YBa_2Cu_3O_{7-x}$ coated superconductors by STEM tomography*, Phys. C **469**, 2052 (2009).

[8] I. A. Sadovskyy, A. E. Koshelev, A. Glatz, V. Ortalan, M. W. Rupich, and M. Leroux, *Simulation of the vortex dynamics in a real pinning landscape of $YBa_2Cu_3O_{7-\delta}$ coated conductors*, Phys. Rev. Applied **5**, 014011 (2016).

[9] I. A. Sadovskyy, Y. Jia, M. Leroux, J. Kwon, H. Hu, L. Fang, C. Chaparro, S. Zhu, U. Welp, J.-M. Zuo, Y. Zhang, R. Nakasaki, V. Selvamanickam, G. W. Crabtree, A. E. Koshelev, A. Glatz, and W.-K. Kwok, *Toward superconducting critical current by design*, Adv. Mat. **28**, 4593 (2016).

[10] M. Leroux, K. J. Kihlstrom, S. Holleis, M. W. Rupich, S. Sathyamurthy, S. Fleshler, H. Sheng, D. J. Miller, S. Eley, L. Civale, A. Kayani, P. M. Niraula, U. Welp, and W.-K. Kwok, *Rapid doubling of the critical current of $YBa_2Cu_3O_{7-\delta}$ coated conductors for viable high-speed industrial processing*, Appl. Phys. Lett. **107**, 192601 (2015).

[11] S. Eley, M. Leroux, M. W. Rupich, D. J. Miller, H. Sheng, P. M. Niraula, A. Kayani, U. Welp, W.-K. Kwok, and L. Civale, *Decoupling and tuning competing effects of different types of defects on flux creep in irradiated $YBa_2Cu_3O_{7-\delta}$ coated conductors*, Supercond. Sci. Tech. **30**, 015010 (2017).

[12] I. A. Sadovskyy, A. E. Koshelev, C. L. Phillips, D. A. Karpeyev, and A. Glatz, *Stable large-scale solver for Ginzburg-Landau equations for superconductors*, J. Comp. Phys. **294**, 639 (2015).

[13] D. Y. Vodolazov, *Vortex-induced negative magnetoresistance and peak effect in narrow superconducting films*, Phys. Rev. B **88**, 014525 (2013).

[14] G. R. Berdiyorov, M. V. Milošević, and F. M. Peeters, *Novel commensurability effects in superconducting films with antidot arrays*, Phys. Rev. Lett. **96**, 207001 (2006).

[15] G. R. Berdiyorov, M. V. Milošević, and F. M. Peeters, *Superconducting films with antidot arrays — novel behavior of the critical current*, Europhys. Lett. **74**, 493 (2006).

[16] I. A. Sadovskyy, Y. L. Wang, Z.-L. Xiao, W.-K. Kwok, and A. Glatz, *Effect of hexagonal patterned arrays and defect geometry on the critical current of superconducting films*, Phys. Rev. B **95**, 075303 (2017).

[17] R. Hooke and T. A. Jeeves, *"Direct search" solution of numerical and statistical problems*, J. ACM **8**, 212 (1961).

[18] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim. **7**, 1 (1997).

[19] C. Audet and J. E. Dennis Jr, *Analysis of generalized pattern searches*, SIAM J. Optim. **13**, 889 (2002).

[20] E. D. Dolan, R. M. Lewis, and V. Torczon, *On the local convergence of pattern search*, SIAM J. Optim. **14**, 567 (2003).

[21] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, *Convergence properties of the Nelder-Mead simplex method in low dimensions*, SIAM J. Optim. **9**, 112 (1998).

[22] K. I. McKinnon, *Convergence of the Nelder-Mead simplex method to a nonstationary point*, SIAM J. Optim. **9**, 148 (1998).

[23] L. M. Rios and N. V. Sahinidis, *Derivative-free optimiza-*

tion: a review of algorithms and comparison of software implementations, J. Global Optim. **56**, 1247 (2013).

[24] I. Loshchilov, M. Schoenauer, and M. Sebag, *Adaptive coordinate descent*, in *Proc. of the 13th Annual Conference on Genetic and Evolutionary Computation* (ACM, 2011) pp. 885–892.

[25] A. H. G. Rinnooy Kan and G. T. Timmer, *Stochastic global optimization methods. Part I: Clustering methods*, Math. Prog. **39**, 27 (1987).

[26] A. H. G. Rinnooy Kan and G. T. Timmer, *Stochastic global optimization methods. Part II: Multi level methods*, Math. Prog. **39**, 57 (1987).

[27] J. Kennedy, *Particle swarm optimization*, in *Encyclopedia of Machine Learning* (Springer, 2011) pp. 760–766.

[28] R. Poli, J. Kennedy, and T. Blackwell, *Particle swarm optimization*, Swarm Intell. **1**, 33 (2007).

[29] I. C. Trelea, *The particle swarm optimization algorithm: convergence analysis and parameter selection*, Inform. Process. Lett. **85**, 317 (2003).

[30] M. J. D. Powell, *On search directions for minimization algorithms*, Math. Prog. **4**, 193 (1973).

[31] C. Price, I. Coope, and D. Byatt, *A convergent variant of the Nelder-Mead algorithm*, J. Optim. Theory Appl. **113**, 5 (2002).

[32] L. Nazareth and P. Tseng, *Gilding the Lily: a variant of the Nelder-Mead algorithm based on golden-section search*, Comp. Optim. Appl. **22**, 133 (2002).

[33] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Simplicial direct-search methods*, in *Introduction to derivative-free optimization* (SIAM, 2009) Chap. 8, pp. 141–162.

[34] A. E. Koshelev, I. A. Sadovskyy, C. L. Phillips, and A. Glatz, *Optimization of vortex pinning by nanoparticles using simulations of the time-dependent Ginzburg-Landau model*, Phys. Rev. B **93**, 060508 (2016).