

Open source platform for block polymer formulation design using particle swarm optimization

Logan J. Case¹, Kris T. Delaney², Glenn H. Fredrickson², Frank S. Bates¹, and Kevin D. Dorfman^{1a}

¹ Department of Chemical Engineering and Materials Science, University of Minnesota - Twin Cities, 421 Washington Avenue SE, Minneapolis, MN 55455, USA

² Department of Chemical Engineering and Materials Research Laboratory, University of California, Santa Barbara, Santa Barbara, CA 93106, USA

Received: date / Revised version: August 18, 2021

Abstract. Facile exploration of large design spaces is critical to the development of new functional soft materials, including self-assembling block polymers, and computational inverse design methodologies are a promising route to initialize this task. We present here an open-source software package coupling Particle Swarm Optimization (PSO) with an existing open-source Self-Consistent Field Theory (SCFT) software for the inverse design of self-assembling block polymers to target bulk morphologies. To lower the barrier to use of the software, and facilitate exploration of novel design spaces, the underlying SCFT calculations are seeded with algorithmically generated initial fields for four typical morphologies: lamellae, network phases, cylindrical phases, and spherical phases. In addition to its utility within PSO, the initial guess tool also finds generic applicability for stand-alone SCFT calculations. The robustness of the software is demonstrated with two searches for classical phases in the conformationally symmetric diblock system, as well as one search for the Frank-Kasper σ phase in conformationally asymmetric diblocks. The source code for both the initial guess generation and the PSO wrapper are publicly available.

^a e-mail: dorfman@umn.edu

1 Introduction

The ability of block polymers to self-assemble into a range of nanostructured morphologies has been the focus of substantial theoretical and experimental work, motivated by the potential to leverage this self-assembly to tune the microstructure of these materials, thereby enabling them to possess unique physical or chemical properties [1]. The linear AB diblock copolymer melt is the simplest such system, characterized at the mean-field level by a trio of dimensionless parameters: the volume fraction of block A, f_A , defining chain composition; conformational asymmetry, $\epsilon = b_A/b_B$ where b_i is the statistical segment length of chemistry i ; and the segregation strength, $\chi_{AB}N$, consisting of the temperature-dependent Flory-Huggins interaction parameter, χ_{AB} , and degree of polymerization, N . Even this relatively simple molecule produces a wide array of nanostructured morphologies within its three-dimensional design space. In addition to the classical phases of body-centered cubic spheres, hexagonally-packed cylinders and lamellae [2], diblock copolymers exhibit two network phases, double-gyroid [3,4] and O^{70} [5,6], as well as four Frank-Kasper phases, σ [7,8], A15 [9–11] C14 [12,13] and C15 [12], that emerge with increasing conformational asymmetry.

Going beyond the capabilities of the diblock copolymer melt requires adding more blocks, which quickly increases the complexity of the problem [14]. Simply adding one additional block to a triblock polymer increases the number of possible morphologies, even for something as simple as the particle-forming phases [15], while also increasing the dimensionality of the design space. Adding yet another block using the same palette of three monomers produces a tetrablock terpolymer, a system that has proven to be a particularly fruitful avenue for materials discovery [7,16–22]. Here, each of the nine available block sequences (ABAC, ABCA, ACAB, BABC, etc.) offers an 8-dimensional design space, with three segregation strengths, three independent block fractions, and two statistical segment length ratios. Adding more blocks and chemistries leads to factorial growth in the number of block sequences of linear polymers in addition to new block fractions, segregation strengths, and segment lengths [14]. Still more degrees of freedom become available when considering branching architectures or multi-component blends [1,23–27]. Thorough experimental exploration of such vast design spaces is infeasible.

A more efficient paradigm computationally screens large design spaces to identify regions where a desired morphology is likely to occur, and then uses these promising regions to guide experimental study or a more thorough computational analysis of the region. Indeed, theory has often played a central role in driving and rationalizing the discovery of new phases in diblock copolymer melts cited at the outset [2,4,5,9]. The question then becomes: how should this computational screening be done? The exhaustive grid searches used in the traditional, forward approach to these design problems become prohibitively expensive for high-dimensionality systems, forcing researchers to constrain the design spaces. Examples of such constrictions can be seen in previous cases where theory played a key role in

driving experiments [5, 16, 18, 28–30]. A particularly relevant example of such theory-driven collaborations is the work of Chanpuriya *et al.* [16]. Leveraging prior experimental findings, they focused on poly(styrene)-*b*-poly(isoprene)-*b*-poly(styrene)’-*b*-poly(ethylene oxide) (SIS’O) at fixed temperature [16], thus fixing the block sequence, Flory-Huggins parameters and statistical segment lengths (eliminating six degrees of freedom). They further constrained their search by fixing the degree of polymerization of the SIS’ parent triblock as well as the ratio of total styrene (in both blocks) to isoprene in that parent triblock [16]. Ultimately, their search considered only two design parameters from the nine possible degrees of freedom: distribution of styrene between the S and S’ blocks, and the block fraction of ethylene oxide, the latter of which was constrained further to low volume fractions to focus on the sphere-forming region. Their calculations predicted the emergence of the Frank-Kasper σ phase; accompanying experiments not only confirmed the existence of σ , but identified even more morphologies including an A15 phase [16]. In another example, Liu *et al.* [18] presented a theoretical study of binary blends of AB diblock copolymer, and predicted the formation of Frank-Kasper phases, including σ and A15. A blend of two AB diblock copolymers offers six degrees of freedom, including two block fractions, segregation strength, conformational asymmetry, a blend fraction, and a length ratio between the two chains. To make their analysis feasible, they were forced to constrain their system to conformationally symmetric diblocks with fixed segregation strength and one fixed block fraction. Of the remaining three variables, their grid searches varied only the blend fraction and block length ratio, while restricting the second block fractions to two cases: equivalent majority blocks in both chains, or fixed length of the second minority block [18]. Recent experiments, inspired by these calculations, have confirmed the emergence of σ and A15, among other morphologies, in similar binary diblock blends [31, 32]. These works offer valuable examples of theory guiding experimental discovery, but also highlight the compromises required to make exhaustive searches computationally feasible.

An alternative to a constrained grid search is inverse design. Here, a guided search of the design space is performed using optimization algorithms set to target a particular property [33]. Various methods have been proposed for such inverse design protocols in block polymers [34–37], including Monte-Carlo [38, 39] and evolutionary algorithms [40–42]. Most of this work, however, focuses on directed self-assembly in thin films, rather than the less-constrained problem of bulk morphologies.

The approach we choose to pursue here applies particle swarm optimization (PSO) in conjunction with self-consistent field theory (SCFT) calculations to the problem of block polymer design. SCFT is one of the most powerful and widespread theoretical frameworks to study the equilibrium behavior of block copolymers, and has been well-described in other works [43–45]. SCFT serves as the compute engine to determine the fitness of a given morphology, and the motion through the design space is controlled by PSO, which emerged from the study of biological swarming

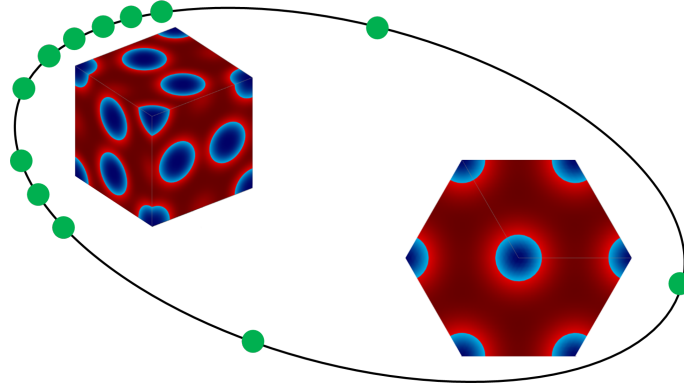


Fig. 1: Visualization of PSO algorithm, showing the swarm’s agents on a ring communication graph, skewing toward the target phase, the Frank-Kasper A15 phase, and away from a competitor, hexagonal cylinders.

behavior seen in, *e.g.*, birds, insects, and fish [46,47] and has been applied to a diverse range of problems including predictions of crystal structures [48], mechanical structure design [49,50], scheduling [51], and robotics [52]. Similar to the biological case, the PSO algorithm relies on communication among several individuals (“particles” or “agents”) to direct the movement of the group (“swarm”) as a whole towards a region of high fitness [46,47], which will correspond in our case to a low free energy for the desired block polymer morphology when compared to all competing candidate phases. As each agent moves through hyperspace, they gain more knowledge about the likely location of the global optimum, and communicate this information with the rest of the swarm until the group reaches a consensus [46,47]. This idea is illustrated in fig. 1.

PSO has been used previously in two studies of block polymer phase behavior. Paradiso *et al.* [53] first demonstrated the ability of the PSO algorithm to search block polymer parameters for those that favored particular morphologies in thin films. This work leveraged the reduced dimensionality of thin-film systems to simplify the forward problem of determining an equilibrium morphology for a given set of polymer parameters. Khadilkar *et al.* [54] introduced the additional complexity of metastability in bulk morphologies. In this case, they leveraged free energy differences among a set of predefined competing morphologies to demonstrate PSO’s capacity to sort among these metastable competitors to identify regions of stability for bulk morphologies. These works demonstrate the potential for PSO to be deployed as a materials discovery engine for block polymer design. However, they were developed in the context of a custom software code that poses an obstacle to more widespread use.

To make PSO more broadly accessible to the block polymer community, the present work adapts the previous PSO approach for bulk morphologies [54] into an open-source software package for the design of block polymers. The work described here diverges from that of Khadilkar *et al.* [54] in two important ways. First, our program, called

pscfInverse [55], is distributed freely as an open-source wrapper around the open-source Polymer Self-Consistent Field (PSCF) software [56], with which users may already be familiar. Second, we attempt to lower the barrier to use by non-computational research groups by seeding SCFT calculations with algorithmically generated fields, rather than a pre-computed library. By operating in this way, we reduce the set-up work required of the user to prepare for the calculation. This new tool for algorithmic initial guess generation, which is called pscfFieldGen [55], should also find utility in stand-alone SCFT calculations where previous solutions are unavailable, e.g., when analyzing a promising new morphology. The robustness of the PSO tool is demonstrated through the solution of two simple two-dimensional search problems for conformationally symmetric diblock copolymers, and a more challenging and computationally intense three-dimensional search for the Frank-Kasper σ phase for a conformationally asymmetric diblock polymer.

2 Algorithm and Implementation

Particle Swarm Optimization (PSO) is a heuristic algorithm which leverages communication among a network of semi-independent particles or agents to solve a global optimization problem [46, 47]. By communicating relevant data about the individual experiences in the search space, the swarm builds a consensus among agents about the locations of the best positions in the search space [46, 47, 57]. Best positions, in this case, are those with the highest value for the objective function, Ω , which we define below. For a D-dimensional search space, each agent tracks its position $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, its velocity $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, and the “personal” best position visited in its individual history $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$. Each agent is also able to communicate with a fixed set of neighbors, as determined by the swarm’s communication graph.

At the start of the search, each agent is spawned at a random position within the search space, is assigned a random velocity, and calculates its initial fitness. At each step, the agent first determines the “global” best position visited by any of its neighbors $\mathbf{g}_i = (g_{i1}, g_{i2}, \dots, g_{iD})$ and then updates its position and velocity according to [57]

$$\mathbf{v}_i^{k+1} = \gamma (\mathbf{v}_i^k + c_1 \boldsymbol{\epsilon}_1 \odot (\mathbf{p}_i - \mathbf{x}_i^k) + c_2 \boldsymbol{\epsilon}_2 \odot (\mathbf{g}_i - \mathbf{x}_i^k)) \quad (1)$$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \quad (2)$$

where $\odot$ indicates the Hadamard product of two vectors; superscripts have been used to indicate the old (k) and new ($k+1$) values; $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are random vectors generated uniquely for each agent at each step with components on $(0, 1]$; γ is a constriction factor in the range $(0, 1]$; c_1 and c_2 are weighting factors. The standard construction of PSO [57] uses a ring communication graph with $\gamma = 0.72984$ and $c_1 = c_2 = 2.05$ based on results of a stability analysis of the algorithm [58].

At each point in the search space, SCFT is used to obtain the intensive free energy, F , of the target phase, T , and each phase in the set of competitors, C . The objective function is defined as [54]

$$\Omega = \min_{m \in C} (F_m) - F_T \quad (3)$$

which acts as a measure of the stability of the target phase by comparing target phase free energy, F_T , to the most stable competitor’s free energy, $F_{m \in C}$. Using this sign convention, the target phase is thermodynamically stable when Ω is positive and becomes more stable with increasing Ω ; thus, PSO seeks to maximize Ω .

Any optimization over polymer system parameters requires some mapping between the polymer parameters required for a SCFT calculation and the variables that define the PSO search space. The most obvious such mapping is a direct 1:1 map from the polymer parameters to the search variables. While simple, this mapping constrains the user’s control over the search space by providing no expression of relationships between parameters. For example, had Chanpuriya *et al.* [16] used an inverse design approach in their study of SIS’O, a direct map alone would have provided no mechanism to vary the distribution of styrene between the S and S’ blocks in this tetrablock terpolymer. Instead, pscfInverse allows users to define this variable-to-parameter mapping with a set of linearizable “parameter relationships” which can be used either as PSO search variables or as fixed constraints to fully define the system. When PSO variables are updated, their new values are used to construct linear equations of polymer parameters defining each relationship. These are collected from each variable, along with similar equations from each constraint, and the resulting system of linear equations is solved to obtain the new polymer system parameters.

Five parameter relationships are available in pscfInverse and used in the case studies described in the present contribution. Polymer chain composition is controlled through two parameter relationships. The first is the total block length, L_T , which defines the sum of the degree of polymerization, or length, of a set of blocks i such that $L_T = \sum L_i$. The second is the log of a ratio of sums of block lengths, $L_R = \ln(\sum L_n / \sum L_d)$, for blocks n and blocks d , specified by the user to be accounted, respectively, in the numerator or denominator of the ratio. For example, in the calculations of Chanpuriya *et al.* [16], the distribution of styrene was captured in their τ parameter. In our PSO code, this would require that the length of the S block be accounted in both the numerator and denominator, while accounting the S’ block only in the denominator. Here, logarithmic scaling was chosen to offer balanced numerical representation in the PSO search range for ratios and their reciprocal. This representation requires no knowledge of polymer architecture at the program level, and thus enforces no restrictions about such onto the user, making for a simple but flexible interface. Block fractions, rather than block lengths, can be considered by simply constraining the total length of a polymer to unity. Statistical segment length is also described by two relationships: first as a logarithmic ratio between single values, $\ln(b_1/b_2)$, where logarithmic scaling was again chosen to balance numerical representation of a ratio and

its reciprocal; second, as a direct value map. Flory-Huggins parameters are presently treated only with direct value maps.

2.1 SCFT Details

SCFT calculations required for fitness updates use the Fortran version of the Polymer Self-Consistent Field (PSCF) software [56]. PSCF implements SCFT unit-cell calculations for linear block copolymers and small-molecule solvents using the incompressible Gaussian chain polymer model [56]. The implementation uses either a quasi-Newton method [56] or Anderson mixing algorithm [59–63] to optimize the field while searching for the optimal unit cell dimensions [64]. PSCF also enforces symmetry constraints by using a symmetry-adapted Fourier expansion consistent with the desired space group to represent the field during the calculation. All calculations shown here make use of symmetry constraints in the SCFT calculations and employ the simultaneous unit cell optimization capability [56] to ensure accurate free energies for each phase. All calculations are converged to an error less than 10^{-5} as defined in Ref. [63].

2.2 Initial Field Generation

Starting an SCFT calculation requires an initial guess for the chemical exchange field and the pressure field [44]. These initial guesses are then evolved as part of the calculation to achieve self-consistency and, following convergence, used to compute the intensive free energy and thus establish the fitness in the PSO calculation. As a result, it is essential that the initial guesses be within the basin of attraction for the fields governing a candidate phase to ensure consistent and repeatable fitness values at each point in the search space. To accomplish this goal, prior PSO work initialized SCFT calculations from a curated library of defect-free fields obtained from test calculations in the phase’s known stability window [54]. In many cases, a SCFT solution at a new point in phase space can be efficiently obtained from an existing solution at an old point by incrementally updating the polymer parameters in a parameter sweep between the points [56]. Thus, the library approach should be efficient if the library is reasonably representative of the search space.

For well-explored systems with known phase diagrams over part of the parameter space, generating fields to place in this library is time-intensive, but straightforward. However, the power of an inverse design software is its ability to automate the exploration of large, complex, and novel design spaces. Building a library for a novel design space would first require manual exploratory calculations to find appropriate regions from which to collect the seed fields. Depending on the complexity of the space, the reliability of the user’s intuition guiding the search, and the size of the

regions in which each competitor will converge, this step could demand a substantial search on its own; in the extreme case, this setup itself becomes an exhaustive grid search and obviates the need for design software altogether. After the seed fields have been converged, the need to organize and index the fields for a database, and define querying protocols to look up the ‘best’ seed field to use at a given point in phase space creates another barrier to use of the method outside dedicated computational groups. Automation of the indexing and query definition could lower this barrier, but implementing an automation prescriptive enough to be accessible to a user unfamiliar with database systems yet flexible enough to handle the nearly limitless variety of polymer architectures is a nontrivial problem. Finally, even when the library is built, complex design spaces could contain multiple stability windows for a given morphology [54]. If one stability window were separated from the library entries by a region where the morphology is difficult to converge, it could be obscured by failed parameter sweeps.

Our approach instead relies on algorithmic generation of seed fields to provide initial guesses. The generation algorithms, which are also available as a stand-alone field generation tool called `pscfFieldGen` [55], support lamellae, network phases (such as double gyroid or O^{70}), and particle phases (cylinders or spheres). In each case, the field generator determines the overall volume fraction of each monomer type, $\bar{\phi}_i$ for monomer i , from the PSCF parameter file. For simple systems, such as a neat diblock copolymer melt, $\bar{\phi}_i$ is the volume fraction f_i of the block. However, in some multiblock systems (e.g., tetrablock terpolymers) and blends, where the same chemistry may appear in multiple blocks or in different molecules, $\bar{\phi}_i$ accounts for all instances of a particular monomer type. A core or reference monomer is used to properly size the defining structures of each phase, and is chosen from among all monomers, or from a user-definable subset of monomer options, to be the monomer with the lowest $\bar{\phi}_i$. This selection criteria for core monomer offers some capacity to accommodate structural inversions that occur, for example, between low and high block fractions in diblock polymers. When such an approximation is deemed insufficiently accurate, separate candidates or targets can be assigned to the same morphology, each with a different core monomer. Presently, initial guesses are generated for each candidate using fixed unit cell dimensions specified by the user; from these fixed initial unit cell dimensions, PSCF’s simultaneous unit cell optimization [56] searches for the optimal unit cell parameters. Once the density fields have been initialized for each species, a previously established algorithm in PSCF is used to convert them to the initial guess for the fields appearing within the SCFT equations [56].

In the simplest of these algorithms, lamellae, the reference monomer, k , is assigned a single-period sinusoidal density profile across the unit cell with amplitude $\bar{\phi}_k$ and $\phi_k(r/d = 0.5) = 0$ for unit cell length d . To ensure that the chosen amplitude does not cause non-physical volume fractions in the initial guess for any monomer, the lamellar algorithm does not accept user-defined reference options.

Network phases are initialized with the level-set method [56,65,66]. In this method, the core monomer, k , is taken to be the monomer within the network structure. A real-space density profile for monomer k is generated from a small set of non-zero symmetry-adapted basis functions of the space group using PSCF. From this density profile, a level-set value, c , is chosen such that the associated level surface encloses a volume $V_{\text{network}} = V_{\text{unitcell}} \bar{\phi}_k$. The field is then generated on the real-space grid such that points inside the level surface contain pure core monomer, $\phi_k(\mathbf{r}) = 1$, while non-core monomers are uniformly distributed through points outside the level surface.

Finally, particle phases are initialized using the form-factor method [56,67]. In this method, the user first specifies the positions $\mathbf{R}_n$ of each of the N_p particles in the structure's unit cell. After assigning a form-factor $f_n(\mathbf{q})$ to each particle based on the particle's geometry (presently assumed to be spheres in 3D and cylinders in 2D), the monomer density profile is calculated as Fourier amplitudes for each wave-vector $\mathbf{q}$ with

$$\phi_k(\mathbf{q}) = \frac{1}{V} \sum_{n=1}^{N_p} \exp(i\mathbf{q} \cdot \mathbf{R}_n) f_n(\mathbf{q}) f_s(\mathbf{q}) \quad (4)$$

for the core monomer. $f_s(\mathbf{q})$ is a Gaussian smearing factor applied to soften particle interfaces according to

$$f_s(\mathbf{q}) = \exp(-\psi^2 |\mathbf{q}|^2 / 2)$$

using an interface width, ψ , estimated from Helfand-Tagami theory [68] with

$$\psi = 2\sqrt{b_{\text{avg}}^2 / (6\chi_{\text{avg}})}$$

where b_{avg} is the geometric mean of statistical segment lengths and χ_{avg} is the geometric mean of interaction parameters between the core monomer and each non-core monomer. Amplitudes for non-core monomers are then given by

$$\phi_{j \neq k}(\mathbf{q}) = -\phi_k(\mathbf{q}) \frac{\bar{\phi}_j}{1 - \bar{\phi}_k}$$

which enforces incompressibility for the system overall, but treats the matrix in a multiblock system as well-mixed.

Although the available generation algorithms easily handle the classical phases as well more complex network and particle phases, more exotic morphologies, such as the Janus-like structures recently investigated by Xie *et al.* [22] are not supported by the current guess generation tools, which only support a single core monomer. A planned extension of the tool to support separate core-monomer definitions for each particle would allow approximation of several of these structures. Others, particularly the helical supercylinder and supersphere structures [22], would require more substantial extensions of the generation algorithms.

2.3 SCFT Robustness

Fitness values at each point in the PSO search rely on the numerical solution of the highly nonlinear SCFT equations, making it susceptible to a range of numerical inaccuracies that could cause erroneous fitness values. Thus, because the calculations are automated and largely unmonitored, mechanisms must be in place to screen SCFT calculation results for reliability. If any potential errors are detected, the calculation is flagged as unsuccessful and the associated phase is assigned an arbitrarily high energy (`numpy.inf`). For competitor phases, this effectively removes the phase from consideration for fitness calculations. When the failed calculation involves the target phase, this results in $\Omega = -\infty$ at the position regardless of results from competitor phase calculations, and causes the swarm to always favor positions where the target phase calculation succeeded in any future steps, thus minimizing the position’s impact on the search. The most common cause of an unsuccessful calculation is a failure to converge to a solution, as indicated by PSCF taking more than the maximum allowed iterations to return a result. In extremely rare instances, numerical instabilities in the field optimization can cause non-numeric or infinite energies to be returned. Although unlikely, the program checks for this outcome, and flags those calculations as unsuccessful. Finally, even with symmetry constraints it is possible that the field will converge to something other than the intended mesophase: convergence to either disorder or a supergroup morphology is possible [54, 69]. Similar to Khadilkar *et al.* [54], we screen for this error by checking the inner product of the normalized Fourier modes of the initial and final fields. If the initial and final field are identical, the inner product would equal unity; orthogonal fields would produce an inner product of 0. Thus, if the program detects an inner product below a preset threshold, the calculation is also flagged as unsuccessful with the assumption that it returned an incorrect field. Our testing suggests that this methodology is sufficient in most cases to detect both convergence to disorder as well as convergence to a supergroup, and in examples that follow, a threshold of 0.7 has been used. The effectiveness of this similarity test may vary between phases [54]; the lamellar phase seems to be one example of this which we discuss further in sect. 3. For defect detection and supergroup convergence, manual inspection of completed SCFT calculations during and after a PSO run would be beneficial as a supplement to the similarity check. For convergence to disorder, `pscfInverse` can optionally check the variation in monomer concentrations across the field and reject fields in which the difference between minimum and maximum concentrations for each monomer is below a user-defined threshold.

3 Results

To verify the operation of the software, we focus our case studies on the single-component diblock copolymer system. We begin with two-dimensional searches of conformationally symmetric ($\epsilon = 1$) diblocks varying composition and segregation strength. In each case, we constrain our search to the region $-2.5 \leq \ln(f_A/f_b) \leq 2.5$ (approximately $0.0759 \leq f_A \leq 0.9241$), and $10 \leq \chi N \leq 30$. In these small initial trials, we consider only four candidate phases: lamellae, hexagonally packed cylinders, double gyroid, and BCC spheres. We use a swarm of five agents and run 100 PSO steps. We then present a three-dimensional search in which we introduce conformational asymmetry, ϵ , as an additional design parameter, as well as introducing complex Frank-Kasper phases to the search while targeting σ . Composition is similarly constrained, while segregation strength is constrained to $10 \leq \chi N \leq 20$, and ϵ is allowed to vary from 1 to 2. Here, the bounds for χN were reduced to restrict the size of the σ -stability window [8], making for a more challenging search. The set of candidate phases for this last example is expanded to include FCC spheres, as well as the Frank-Kasper σ and A15 phases. In the larger design space, we expand to a swarm of 10 agents, and again run 100 PSO steps.

Figure 2 shows the position and fitness histories from a search for the BCC phase in a conformationally symmetric diblock copolymer melt. Initially, the swarm has no positions within the stability window for spheres, leading the swarm to identify points in the lamellar region as the best seen positions. Within three steps, however, one agent identifies a point in the A-minority BCC window, near the coexistence line with cylinders. From here, an additional 27 steps are required before every agent has individually found a point where BCC is stable, during which time, positions were identified in both the A-minority and B-minority sphere regions. Identification of both windows leads the swarm to explore both stable regions rather than converging on just one side of the phase diagram. This behavior is most clearly illustrated by the presence of dense clusters in both upper corners of the search space in fig. 2a, with a high number of late-step positions directly between those clusters as agents fly between the two regions. The positional diversity within the swarm enforced by these competing stability basins can be seen in fig. 2c, where the range of current fitness values remains wide throughout the search. At the same time, the swarm's search emphasizes ever more promising regions, indicated by the reduced frequency of failed target-phase SCFT calculations (indicated by points along the bottom axis), which are more likely farther from the target's stability window. Khadilkar *et al.* [54] presented a similar search for BCC in their demonstration of this method. In contrast to the results shown in fig. 2, their search, which used four agents and ran 200 PSO steps, identified only the A-minority stability basin, and clustered primarily around that region. This difference illustrates both the randomness of the PSO search, which is affected by the initial

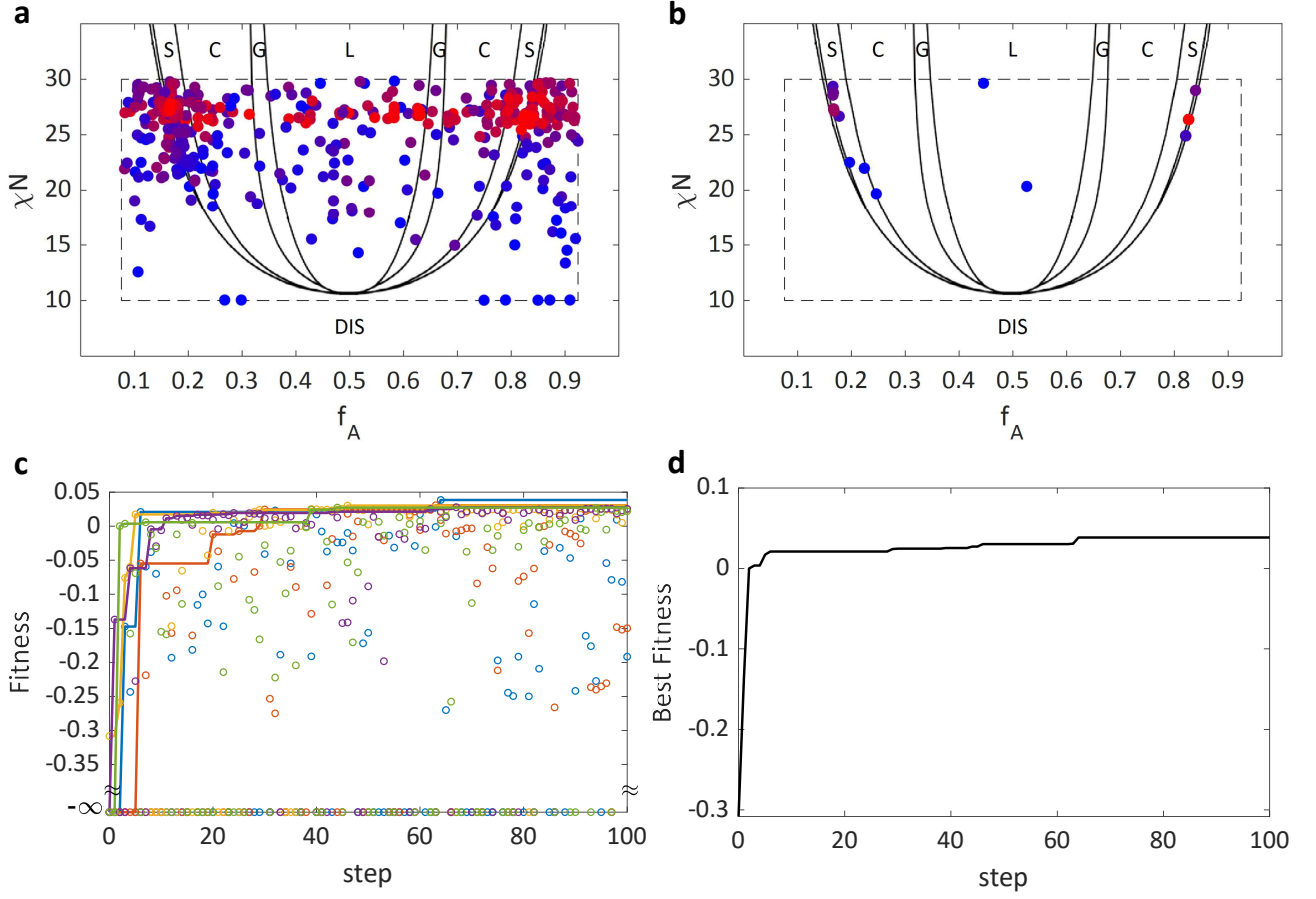


Fig. 2: Results of a 2D search for BCC phase in a conformationally symmetric diblock copolymer melt showing (a) all positions visited during the search and (b) all positions that, at one time in the search, were the best found by the swarm, (c) current fitness (circles) and personal best seen fitness (lines) at each step for each agent and (d) the best fitness found by the swarm through each step. (a) and (b) are both plotted over the known diblock phase diagram (modified from [43]) with the dotted line marking the search bounds and with point color varying from blue (early steps) to red (late steps).

positions and the number of agents, and the robustness, since both our search and the prior work [54] successfully found the BCC phase.

Changing the target to a lamellar phase, shown in fig. 3, emphasizes the effect that competing stability basins can have on the PSO searches. Here, random initialization of agent positions serendipitously placed one of the agents at a point ($f_A \approx 0.45$, $\chi N \approx 29.64$) very close to the global optimum for the lamellar phase within the search window ($f_A = 0.5$, $\chi N = 30$). Although the swarm maintains some diversity in agent position early in the search, this fails to

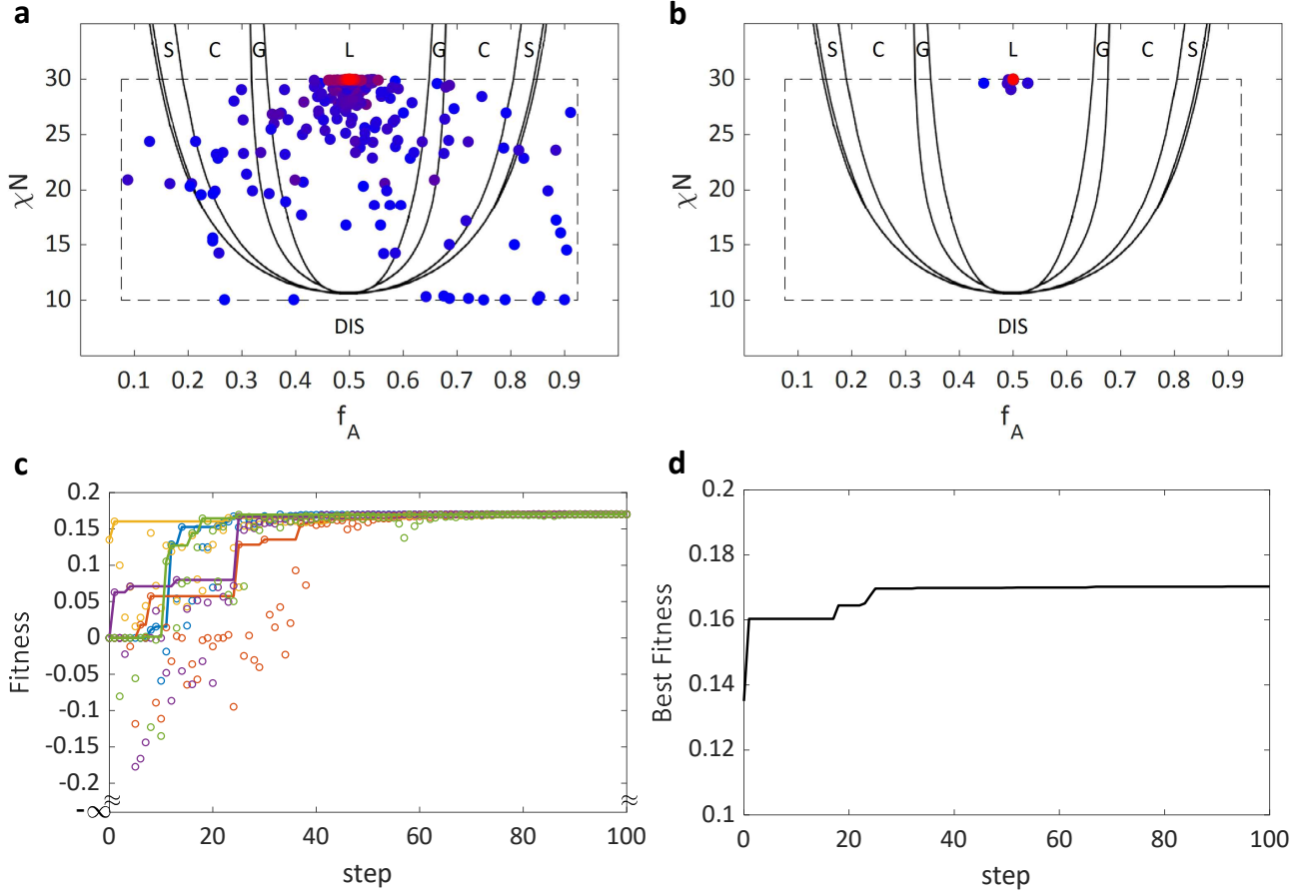


Fig. 3: Results of a 2D search for the lamellar phase in a conformationally symmetric diblock copolymer melt showing (a) all positions visited during the search and (b) all positions that, at one time in the search, were the best found by the swarm, (c) current fitness (circles) and personal best seen fitness (lines) at each step for each agent and (d) the best fitness found by the swarm through each step. (a) and (b) are both plotted over the known diblock phase diagram (modified from [43]) with the dotted line marking the search bounds and with point color varying from blue (early steps) to red (late steps).

identify any reasonably competitive regions. Knowledge of the near-optimum from one agent's initial position quickly permeates the swarm and drives the search toward that point. Without any significant competing stability basins, each agent quickly finds positions high in the lamellar window. Once the personal best positions collapse to a small region, the swarm quickly loses momentum and converges around the global optimum by step 40. Tight clustering in fig. 3a shows this quick convergence, but the importance of diverse personal best positions for continued exploration can be seen in fig. 3c. Around step 25, all but one agent has found a position very close to the global optimum; after

that step, the only agent to not have found a near-optimum position is also the only one to stray noticeably from the rest of the swarm. Once that agent finds a near-optimum position, the whole swarm quickly settles at the optimum.

Comparing figs. 2c and 3c reveals a sharp contrast in the number of “failed” target phase calculations, with none appearing in fig. 3c. The small size and low dimensionality of the one-dimensional lamellar SCFT problem means that the calculation always converges within the maximum allowed number of iterations (set at 200). We observe that, rather than failing to converge as other phases often do in highly unfavorable regions of phase space, the lamellar calculations tend to collapse to disorder. At the same time, the simplicity of the initialized field (using only two basis functions) makes the similarity check ineffective. In this calculation, the concentration variation test was not used to check the final real-space concentration field, causing many calculations which converged to disorder to be erroneously accepted by the program. This can be seen in the clustering of points along $\Omega = 0$ in early steps. Despite the erroneous acceptance of these SCFT calculations, the PSO algorithm proved robust enough to tolerate the errors and yield a correct result. We repeated this calculation using the concentration range test; results can be found in fig. S1.

Frank-Kasper phases offer a valuable route to increase the complexity of design problems in diblock polymer space, and serve as a more stringent test of the PSO algorithm. Not only do Frank-Kasper phases require expansion of the search into a third dimension to access conformationally asymmetric diblock polymers, but the near-degeneracy of the free energies of most Frank-Kasper phases [9, 11, 12] combined with the higher complexity of the Frank-Kasper structures themselves make the robustness of the automated SCFT protocols all the more critical. Figure 4 shows results of a 3D search for the Frank-Kasper σ phase. With this particular set of random initial positions, the swarm initially has no agents at positions where σ is the stable phase, and most agents do not even successfully converge the target phase due to its relatively small volume in the phase space and the challenges in converging metastable σ phase solutions. By assigning failed convergence of σ to have infinite energy, rather than discarding the calculation, we force the swarm to move away from these numerically challenging regions of the state space, which tend to be regions far from the σ stability window. Early in the search, the swarm’s best seen positions generally fall in or near the A-minority BCC stability windows at low ϵ . Early improvements in the swarm’s best seen position do, however, trend toward higher χN and ϵ , in the direction of the known σ phase stability window. Rather early in the search an agent identifies a position where σ is stable, after which the swarm gradually coalesces around the high- χN and high- ϵ σ phase stability window. Following the identification of a stable point, the swarm periodically identifies positions further within the stability basin, approaching a global optimum within the search space. During this time, reduced frequency of infinite fitness values in fig. 4c as well as clustering in fig. 4a indicate that the swarm narrows its search to areas where σ calculations readily converge.

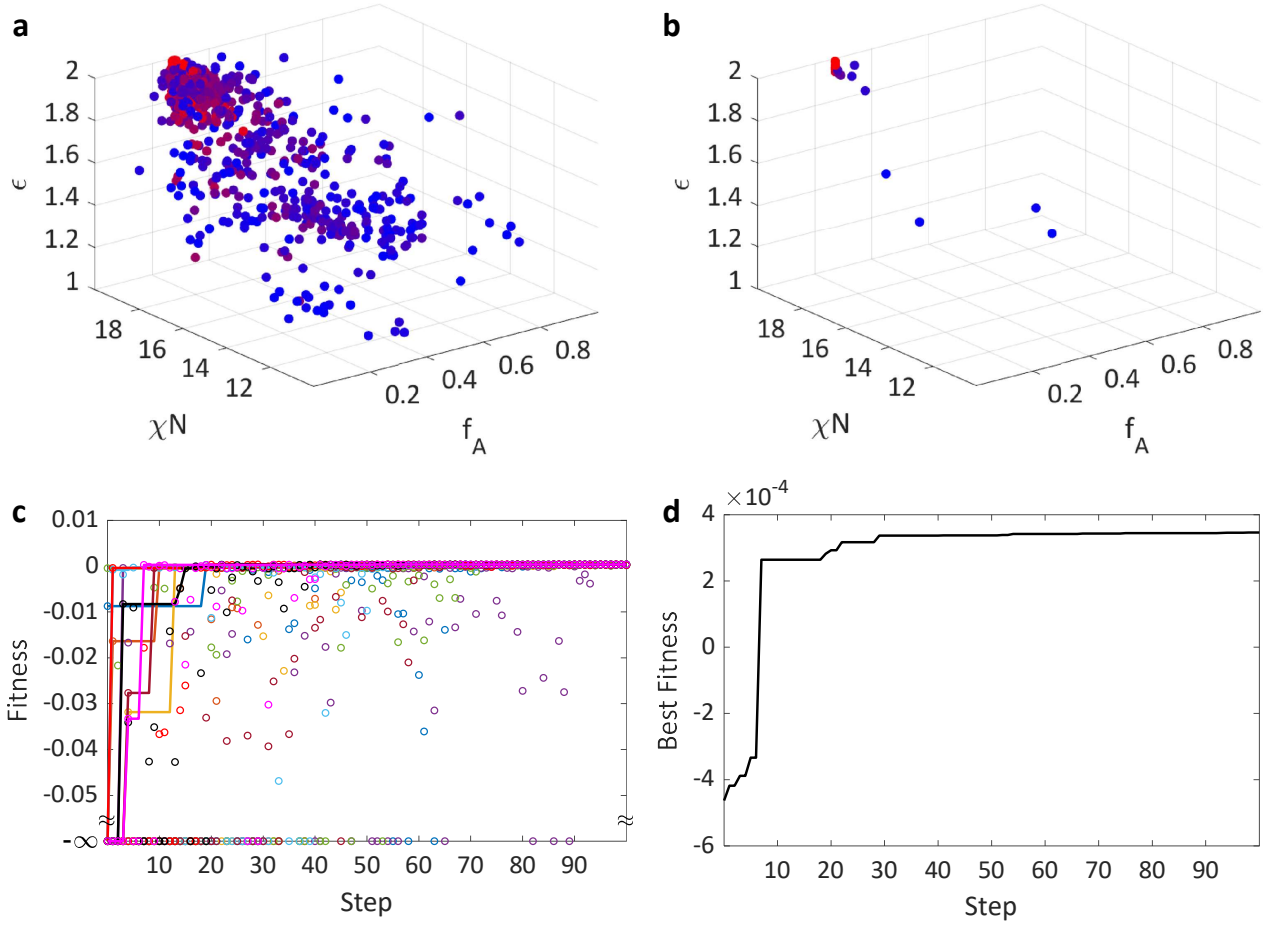


Fig. 4: Results of a 3D search for sigma phase in a neat diblock copolymer melt showing (a) all positions visited during the search and (b) all positions that, at one time in the search, were the best found by the swarm, (c) current fitness (circles) and personal best seen fitness (lines) at each step for each agent and (d) the best fitness found by the swarm through each step. (a) and (b) also show search progression by varying point color from blue (early steps) to red (late steps).

4 Conclusions

This work introduces an open-source software package for the inverse design of linear block polymers to target bulk morphologies using PSO and SCFT. Unlike in prior applications of PSO to such design problems [54], the present software package is made more accessible through the use of algorithmic generation of initial guess fields for SCFT calculations using broadly accessible software. Software functionality was demonstrated with design problems in the diblock polymer parameter space. The software successfully identified stability windows for BCC and lamellae in $(f_A, \chi N)$ space, as well as σ in $(f_A, \chi N, \epsilon)$.

We have demonstrated a successful implementation of PSO for block polymer design using the open-source PSCF program, but several limitations of the existing tool should be noted. The present implementation uses the Fortran version of PSCF [56] for SCFT calculations, which restricts its application to linear polymers at this time. The `pscfInverse` program was designed to allow different SCFT solvers to be wrapped by the PSO code with relative ease, and extension to interface with the new C++ version of PSCF [70] would enable design with the branched architectures available in the nascent C++ code. Likewise, it should be relatively easy for users who already have their own SCFT code to implement this PSO wrapper with some modest modifications to provide compatibility with input/output files in their code. Moving toward consideration of branched architectures, further improvements to the PSO wrapper will be required to access the full polymer design space. For example, the nature of the PSO algorithm restricts searches to continuous variables, limiting its ability to search discretized design parameters such as the number or connectivity of blocks. Creative use of discrete ranges on a continuous variable could enable access to such design parameters, but the program does not presently support this functionality. Other, more generic, improvements could also be implemented. The guess generation algorithms, as previously discussed, place limitations on the variety of phases that can be considered. Improvements to the guess-generation algorithms would further broaden the potential of the program. Finally, the method is unable to consider any morphology not in the set of candidate phases. Identification of unanticipated stable morphologies [71–74] remains a significant open question in block polymer material design [14].

Testing of this tool has also uncovered a subtle difficulty in the application of this method. Fundamentally, this method relies on the ability of the SCFT solver to obtain converged fields for each phase in all regions of the search space where the phase is reasonably competitive and thus of interest in the search. We attempted a search for the A15 phase in the three-dimensional diblock parameter space, but were unable to obtain accurate results. Analysis of the results showed abnormally high failure rates for σ phase at high conformational asymmetry ($\epsilon > 2.5$) where A15 is known to become stable [11]. Since σ should be the closest competitor to A15 over much of this range, and should provide the lower bound to the A15 stability window in the f_A dimension, failure of these calculations proved significant and caused the swarm to converge on points in the σ phase window, rather than that of A15. The cause of these convergence problems in PSCF is presently unknown, but was only seen in the course of this work to impact σ at high conformational asymmetry. In general, understanding the convergence of SCFT calculations is a challenging issue [61] due to the nonlinear, non-local nature of the problem and the need to combine multiple numerical methods to achieve the solution. Investigation and correction of this or similar convergence difficulties in PSCF or any other SCFT solver interfaced with this tool can enable application of the approach to more problems.

Although already less computationally expensive than typical brute-force grid searches [54], there may be ways to further improve the efficiency of the PSO methodology. Acceleration of the underlying SCFT calculations, for example on a Graphics Processing Unit [70, 75], could reduce the overall run-time of the program. Adoption of a hybrid approach to initial field seeding, perhaps by combining a library and generation approach, could also reduce the cost of underlying SCFT calculations by minimizing the repetition of more expensive generated-guess calculations at neighboring or identical points in phase space. Nevertheless, the open-source platform presented here represents a significant step toward the broader adoption of inverse design methodologies by providing a robust new tool for block polymer materials discovery via PSO to the research community.

This work was supported by the DMREF Program of the National Science Foundation under awards DMR-1725272 at the University of Minnesota and DMR-1725414 at the University of California, Santa Barbara. Computational resources were provided by the Minnesota Supercomputing Institute.

Author contribution statement

LJC and KDD designed the research; LJC wrote the initial guess generation code; LJC wrote the PSO code based partially on sample code provided by KTD and GHF; KDD and FSB supervised the research; LJC analyzed the results; LJC and KDD wrote the manuscript with input from KTD, GHF, and FSB.

References

1. C.M. Bates, F.S. Bates, *Macromolecules* **50**(1), 3 (2017)
2. L. Leibler, *Macromolecules* **13**(6), 1602 (1980)
3. D.A. Hajduk, P.E. Harper, S.M. Gruner, C.C. Honeker, G. Kim, E.L. Thomas, L.J. Fetters, *Macromolecules* **27**(15), 4063 (1994)
4. M.W. Matsen, M. Schick, *Phys. Rev. Lett.* **72**(16), 2660 (1994)
5. C.A. Tyler, D.C. Morse, *Phys. Rev. Lett.* **94**(20), 208302 (2005)
6. M. Takenaka, T. Wakada, S. Akasaka, S. Nishitsuji, K. Saijo, H. Shimizu, M.I. Kim, H. Hasegawa, *Macromolecules* **40**(13), 4399 (2007)
7. S. Lee, M.J. Bluemle, F.S. Bates, *Science* **330**, 349 (2010)
8. N. Xie, W. Li, F. Qiu, A.C. Shi, *ACS Macro Lett.* **3**(9), 906 (2014)
9. G.M. Grason, B.A. DiDonna, R.D. Kamien, *Phys. Rev. Lett.* **91**(5), 1 (2003)

10. G.M. Grason, *Phys. Rep.* **433**(1), 1 (2006)
11. M.W. Bates, J. Lequeieu, S.M. Barbon, R.M. Lewis, K.T. Delaney, A. Anastasaki, C.J. Hawker, G.H. Fredrickson, C.M. Bates, *Proc. Natl. Acad. Sci. USA* **116**(27), 13194 (2019)
12. K. Kim, M.W. Schulze, A. Arora, R.M. Lewis, M.A. Hillmyer, K.D. Dorfman, F.S. Bates, *Science* **356**(6337), 520 (2017)
13. S. Jeon, T. Jun, S. Jo, H. Ahn, S. Lee, B. Lee, D.Y. Ryu, *Macromol. Rapid Commun.* **40**(19), 1900259 (2019)
14. F.S. Bates, M.A. Hillmyer, T.P. Lodge, C.M. Bates, K.T. Delaney, G.H. Fredrickson, *Science* **336**(6080), 434 (2012)
15. N. Xie, M. Liu, H. Deng, W. Li, F. Qiu, A.C. Shi, *J. Am. Chem. Soc.* **136**(8), 2974 (2014)
16. S. Chanpuriya, K. Kim, J. Zhang, S. Lee, A. Arora, K.D. Dorfman, K.T. Delaney, G.H. Fredrickson, F.S. Bates, *ACS Nano* **10**(5), 4961 (2016)
17. Y. Miyamori, J. Suzuki, A. Takano, Y. Matsushita, *ACS Macro Lett.* **9**, 32 (2020)
18. M. Liu, Y. Qiang, W. Li, F. Qiu, A.C. Shi, *ACS Macro Lett.* **5**(10), 1167 (2016)
19. M. Kaga, T. Ohta, *J. Phys. Soc. Japan* **75**, 043002 (2006)
20. D. Liu, Y.Y. Wang, Y.C. Sun, Y.Y. Han, J. Cui, W. Jiang, *Chin. J. Polym. Sci.* **36**, 888 (2018)
21. A. Takano, K. Soga, J. Suzuki, Y. Matsushita, *Macromolecules* **36**, 9288 (2003)
22. Q. Xie, Y. Qiang, G. Zhang, W. Li, *Macromolecules* **53**(17), 7380 (2020)
23. G. Polymeropoulos, G. Zapsas, K. Ntetsikas, P. Bilalis, Y. Gnanou, N. Hadjichristidis, *Macromolecules* **50**(4), 1253 (2017)
24. Y. Gao, H. Deng, W. Li, F. Qiu, A.C. Shi, *Phys. Rev. Lett.* **116**(6), 068304 (2016)
25. K. Orfanou, H. Iatrou, D.J. Lohse, N. Hadjichristidis, *Macromolecules* **39**(13), 4361 (2006)
26. Y. Xu, W. Li, F. Qiu, Z. Lin, *Nanoscale* **6**(12), 6844 (2014)
27. M. Zhao, W. Li, *Macromolecules* **52**(4), 1832 (2019)
28. B. Zhao, W. Jiang, L. Chen, W. Li, F. Qiu, A.C. Shi, *ACS Macro Lett.* **7**(1), 95 (2018)
29. S.T. Milner, *Macromolecules* **27**(8), 2333 (1994)
30. A.B. Chang, C.M. Bates, B. Lee, C.M. Garland, S.C. Jones, R.K.W. Spencer, M.W. Matsen, R.H. Grubbs, *Proc. Natl. Acad. Sci. USA* **114**(25), 6462 (2017)
31. A.P. Lindsay, R.M. Lewis, B. Lee, A.J. Peterson, T.P. Lodge, F.S. Bates, *ACS Macro Lett.* **9**(2), 197 (2020)
32. A.P. Lindsay, G.K. Cheong, A.J. Peterson, S. Weigand, K.D. Dorfman, T.P. Lodge, F.S. Bates, *Macromolecules* **54**(15), 7088 (2021)
33. Z.M. Sherman, M.P. Howard, B.A. Lindquist, R.B. Jadrich, T.M. Truskett, *J. Phys. Chem.* **152**, 140902 (2020)
34. K.R. Gadelrab, A.F. Hannon, C.A. Ross, A. Alexander-Katz, *Mol. Syst. Des. Eng.* **2**, 539 (2017)
35. M.Z. Miskin, G. Khaira, J.J. de Pablo, H.M. Jaeger, *Proc. Natl. Acad. Sci. USA* **113**, 34 (2016)
36. D. Xu, H. Liu, Y.L. Zhu, Z.Y. Lu, *Nanoscale* **8**, 5235 (2016)
37. R. Zhang, L. Zhang, J. Lin, S. Lin, *Phys. Chem. Chem. Phys.* **21**, 7781 (2019)

38. A.F. Hannon, K.W. Gotrik, C.A. Ross, A. Alexander-Katz, *ACS Macro Lett.* **2**(3), 251 (2013)
39. A.F. Hannon, Y. Ding, W. Bai, C.A. Ross, A. Alexander-Katz, *Nano Lett.* **14**(1), 318 (2014)
40. J. Qin, G.S. Khaira, Y. Su, G.P. Garner, M. Miskin, H.M. Jaeger, J.J. de Pablo, *Soft Matter* **9**(48), 11467 (2013)
41. G.S. Khaira, J. Qin, G.P. Garner, S. Xiong, L. Wan, R. Ruiz, H.M. Jaeger, P.F. Nealey, J.J. de Pablo, *ACS Macro Lett.* **3**(8), 747 (2014)
42. H.M. Jaeger, J.J. de Pablo, *APL Materials* **4**(5), 053209 (2016)
43. M.W. Matsen, *J. Phys.: Condens. Matter* **14**(2), R21 (2002)
44. G.H. Fredrickson, *The Equilibrium Theory of Inhomogeneous Polymers* (Oxford University Press, Oxford, United Kingdom, 2006)
45. A.C. Shi, *Adv. Theory Simulations* p. 1800188 (2019)
46. J. Kennedy, R. Eberhart, *Particle swarm optimization*, in *Proceedings of ICNN'95 - International Conference on Neural Networks* (IEEE, 1995), Vol. 4, pp. 1942–1948
47. R. Eberhart, James Kennedy, *A New Optimizer Using Particle Swarm Theory*, in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (Nagoya, Japan, 1995), pp. 39–43
48. Y. Wang, J. Lv, L. Zhu, Y. Ma, *Phys. Rev. B* **82**, 094116 (2010)
49. P.L. Fernández-Cabán, F.J. Masters, *Comput. Struct.* **202**, 1 (2018)
50. L. Li, Z. Huang, F. Liu, Q. Wu, *Comput. Struct.* **85**(7-8), 340 (2007)
51. S. Pandey, L. Wu, S.M. Guru, R. Buyya, *A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments*, in *2010 24th IEEE International Conference on Advanced Information Networking and Applications* (IEEE, 2010), pp. 400–407
52. J. Pugh, A. Martinoli, *Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization*, in *2007 IEEE Swarm Intelligence Symposium* (IEEE, 2007), Sis, pp. 332–339
53. S.P. Paradiso, K.T. Delaney, G.H. Fredrickson, *ACS Macro Lett.* **5**(8), 972 (2016)
54. M.R. Khadilkar, S.P. Paradiso, K.T. Delaney, G.H. Fredrickson, *Macromolecules* **50**(17), 6702 (2017)
55. <https://pscf.cems.umn.edu>
56. A. Arora, J. Qin, D.C. Morse, K.T. Delaney, G.H. Fredrickson, F.S. Bates, K.D. Dorfman, *Macromolecules* **49**(13), 4675 (2016)
57. D. Bratton, J. Kennedy, *Defining a standard for particle swarm optimization*, in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium, SIS 2007* (2007), pp. 120–127
58. M. Clerc, J. Kennedy, *IEEE Transactions on Evolutionary Computation* **6**(1), 58 (2002)
59. R.B. Thompson, K. Rasmussen, T. Lookman, *J. Chem. Phys.* **120**(1), 31 (2004)
60. M.W. Matsen, *Eur. Phys. J. E* **30**(4), 361 (2009)

61. P. Stasiak, M.W. Matsen, Eur. Phys. J. E **34**(10), 110 (2011)
62. D.G. Anderson, J. Assoc. Comp. Mach. **12**(4), 547 (1965)
63. A. Arora, F.S. Bates, D.C. Morse, K.D. Dorfman, J. Phys. Chem. **146**, 244902 (2017)
64. C.A. Tyler, D.C. Morse, Macromolecules **36**(21), 8184 (2003)
65. E.W. Cochran, Ph.D. thesis, University of Minnesota (2004)
66. M. Wohlgemuth, N. Yufa, J. Hoffman, E.L. Thomas, Macromolecules **34**(17), 6083 (2001)
67. W. Xu, K. Jiang, P. Zhang, A.C. Shi, J. Phys. Chem. B **117**(17), 5296 (2013)
68. E. Helfand, Y. Tagami, J. Chem. Phys. **56**(7), 3592 (1972)
69. C.A. Tyler, J. Qin, F.S. Bates, D.C. Morse, Macromolecules **40**(13), 4654 (2007)
70. G.K. Cheong, A. Chawla, D.C. Morse, K.D. Dorfman, Eur. Phys. J. E **43**(2), 15 (2020)
71. F. Drolet, G.H. Fredrickson, Phys. Rev. Lett. **83**(21), 4317 (1999)
72. Y. Bohbot-Raviv, Z.G. Wang, Phys. Rev. Lett. **85**(16), 3428 (2000)
73. Z. Guo, G. Zhang, F. Qiu, H. Zhang, Y. Yang, A.C. Shi, Phys. Rev. Lett. **101**(2), 028301 (2008)
74. C.L. Tsai, K.T. Delaney, G.H. Fredrickson, Macromolecules **49**(17), 6558 (2016)
75. K.T. Delaney, G.H. Fredrickson, Comp. Phys. Comm. **184**(9), 2102 (2013)

Supplementary Information for “Open source platform for block polymer formulation design using particle swarm optimization”

Logan J. Case,¹ Kris T. Delaney,² Glenn H. Fredrickson,² Frank S. Bates,¹ and
Kevin D. Dorfman¹

¹ Department of Chemical Engineering and Materials Science, University of Minnesota -
Twin Cities, 421 Washington Avenue SE, Minneapolis, MN 55455, USA

² Department of Chemical Engineering and Materials Research Laboratory, University
of California, Santa Barbara, Santa Barbara, CA 93106, USA

S1 Lamellar Search with Concentration Range Check

Lamellar search results in the main text included no explicit check for convergence to disorder. As discussed, this resulted in several lamellar calculations that had converged to disorder being accepted as successful. This calculation has been redone with a check on the concentration ranges observed in the final field for lamellar calculations (the test is omitted for other phases, as these disordered convergences are typically captured in the standard similarity checks). With this test active, an SCFT calculation is accepted as successful only when the difference between the minimum and maximum concentrations of each monomer on the real-space grid is below the chosen threshold of $\Delta\rho = 0.001$. The results of this calculation are shown in fig. S1.

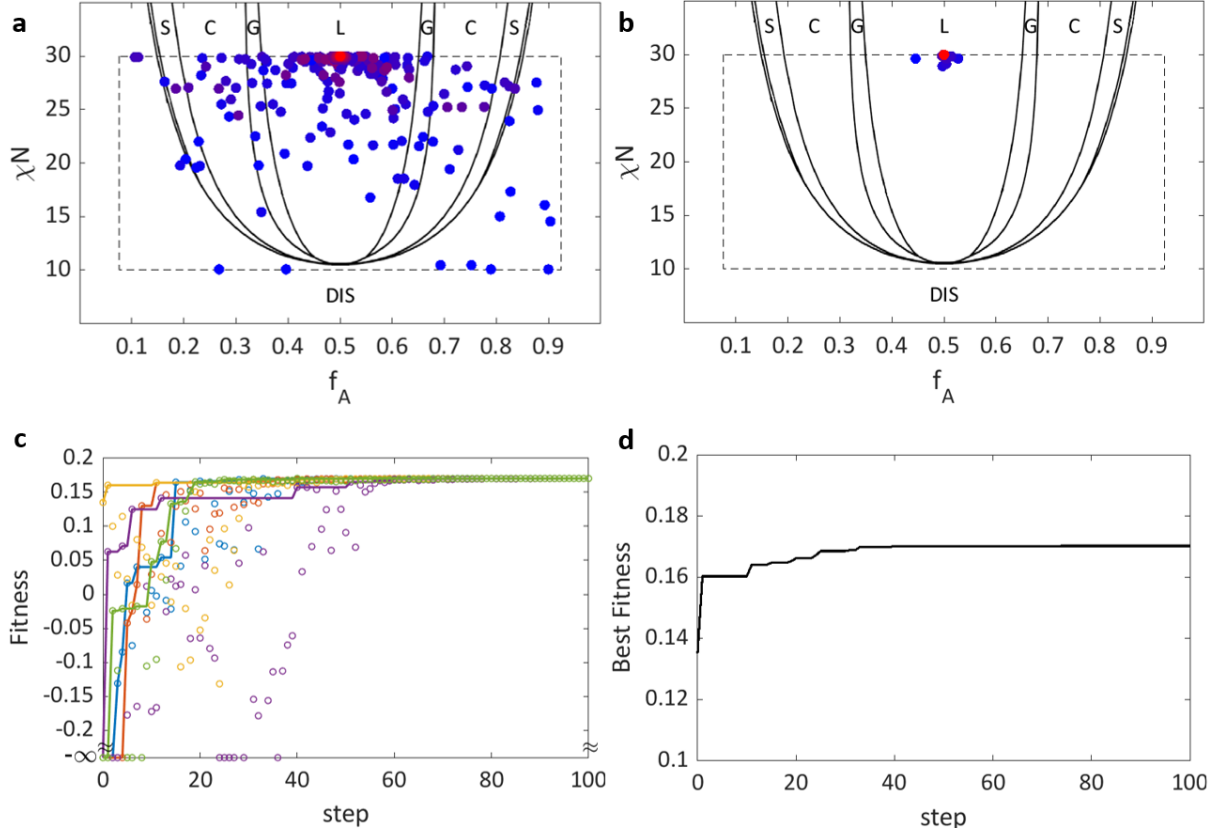


Figure S1: Results of a 2D search for lamellar phase showing (a) all positions visited during the search and (b) all positions that, at one time in the search, were the best found by the swarm, (c) current fitness (circles) and personal best seen fitness (lines) at each step for each agent and (d) the best fitness found by the swarm through each step. (a) and (b) are both plotted over the known diblock phase diagram (modified from [1]) with the dotted line marking the search bounds and with point color varying from blue (early steps) to red (late steps). Lamellar calculations include explicit check for convergence to disorder.

Comparing the results in fig. S1 to those in the main text fig. 3, we see that in several early steps, agent positions that once had fitness $\Omega = 0$ now have fitness $\Omega = -\infty$. This reflects the calculations now correctly identified as having converged to disorder, and rejected as unsuccessful. Besides the presence of some newly-rejected SCFT results, the swarm behavior and search results are largely unaffected.

References

- [1] M.W. Matsen, J. Phys.: Condens. Matter **14**(2), R21 (2002)