NANO LETTERS

xxxx
Vol. 0, No. 0
A−F

A Genetic Algorithm Approach to Probing the Evolution of Self-Organized Nanostructured Systems

Peter Siepmann,[†] Christopher P. Martin,[‡] Ioan Vancea,[§] Philip J. Moriarty,[‡] and Natalio Krasnogor*,[†]

School of Computer Science & IT, The University of Nottingham, Nottingham NG7 2RD, U.K., The School of Physics & Astronomy, The University of Nottingham, Nottingham NG7 2RD, U.K., and Max-Planck-Institut für Physik Komplexer Systeme, Nöthnitzer Strasse 38, 01187 Dresden, Germany

Received April 2, 2007; Revised Manuscript Received May 11, 2007

ABSTRACT

We present a new methodology, based on a combination of genetic algorithms and image morphometry, for matching the outcome of a Monte Carlo simulation to experimental observations of a far-from-equilibrium nanosystem. The Monte Carlo model used simulates a colloidal solution of nanoparticles drying on a solid substrate and has previously been shown to produce patterns very similar to those observed experimentally. Our approach enables the broad parameter space associated with simulated nanoparticle self-organization to be searched effectively for a given experimental target morphology.

Complex systems in chemistry, physics, biology, ecology, economics, computer science, and beyond have often been simulated using cellular automata[1,2] and the closely related lattice gas model technique.[3] Both approaches are appealing modeling paradigms not only because they allow for a piecemeal specification of the laws that govern a given system's dynamics but also because they are intrinsically distributed tools amenable to computational parallelization. However, due to the complex nature of the processes that are simulated with these methods, it is not always possible to analytically derive specific values for the many model parameters that control their time−space evolution. This problem gets more insidious when the intention is for the simulation to quantitatively match observations made in the laboratory of experiments where the underlying physics is not wholly understood. Importantly, however, identifying regions of parameter space which produce good agreement with experiment can provide significant insight into the key physicochemical processes underlying the self-organization of the system.

In this Letter we describe how the combination of a Monte Carlo model[4,5] with a genetic algorithm (GA)[6] can be used to tune the evolution of a simulated self-organizing nanoscale system toward a predefined nonequilibrium morphology. The prototype system we have chosen—a colloidal solution of Au nanoparticles adsorbed on a substrate—not only produces a striking array of complex nonequilibrium patterns but has previously been shown[4,5] to be remarkably well-described by a relatively simple Monte Carlo code. Image morphom-

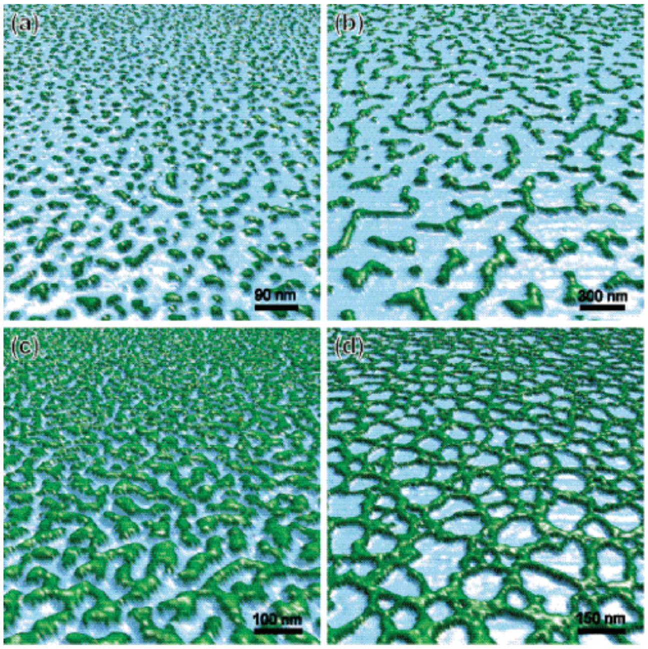
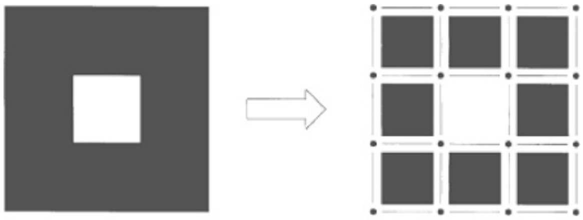


Figure 1. Three-dimensionally-rendered atomic force microscope images showing four of the morphologies that are commonly observed in our experiments. These are formed by spin-casting solutions of ~2 nm diameter thiol-passivated gold nanoparticles onto silicon substrates. With increasing solution concentration from (a−d), we observe (a) isolated droplets, (b) "wormlike" domains, (c) interconnected labyrinthine patterns, and (d) cellular networks.

[†] School of Computer Science & IT, The University of Nottingham.
[‡] The School of Physics & Astronomy, The University of Nottingham.
[§] Max-Planck-Institut für Physik Komplexer Systeme.

10.1021/nl070773m CCC: $37.00 PAGE EST: 5.1
Published on Web 00/00/0000



Number of squares, $n_s = 8$
Number of edges, $n_e = 24$,
Number of vertices, $n_v = 16$

Area (A) $= n_s = 8$,
Perimeter (U) $= -4n_s + 2n_e = 16$,
Euler (χ) $= n_s - n_e + n_v = 0$

Figure 2. Calculation of the 2D Minkowksi functionals that form the basis of the fitness function in the genetic algorithm.[7]

```
while (stopping condition not fulfilled)
  parents = select parents from population
  with a defined probabilty
    'mate' the parents to form (usually) two children
  else
    children = parents
  with a defined probability
    mutate children
  insert children into population
  evaluate and cull population
```

(a) Pseudo-code

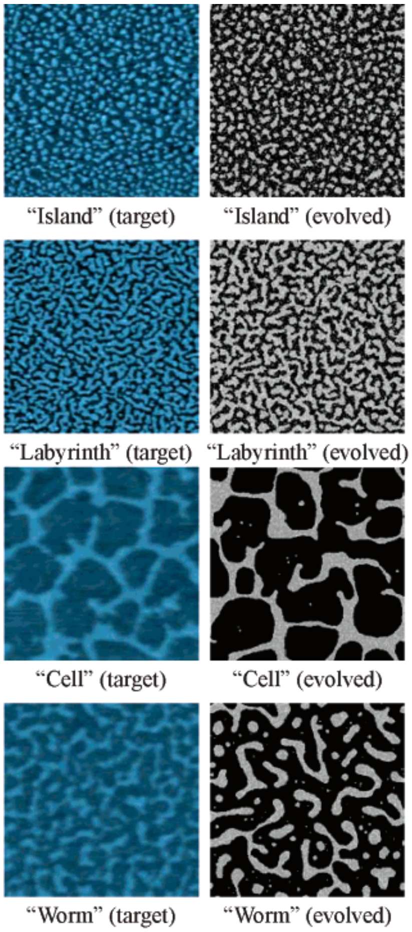
Parent selection:	roulette wheel
Crossover operator:	uniform
Probability of crossover:	0.7
Mutation operator:	BCG[26]
Mutation rate:	0.3
Stopping condition:	after 100 generations
Replacement strategy:	$(\mu+\lambda)$, with $\mu = 16$ and $\lambda = 8$

(b) GA system parameters (where μ represents the population size and λ the number of offspring produced in each generation

Figure 3. Genetic algorithm details.

etry—specifically, Minkowski functional analysis[7]—is used as the basis of the fitness function for the GA. Evolved simulation parameters produce simulated nanoparticle patterns which closely match the target images taken from experimental data and replicate a number of morphological families. Our results provide an important bridge between simulation and experiment in the study of self-organizing nanostructured systems and, moreover, bring us closer to the concept of software control of matter.[8]

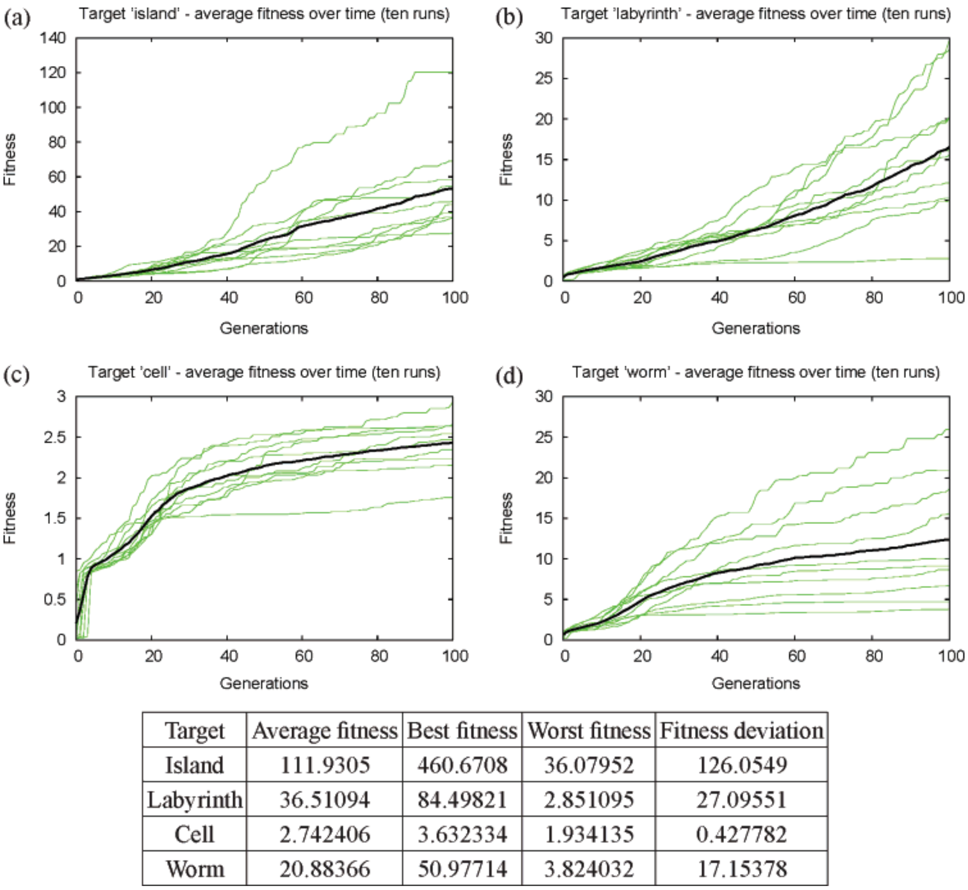
When deposited onto a solid substrate, colloidal nanoparticles self-organize into a variety of complex patterns[4,5,9−13] driven in many cases by the evaporative dewetting of the solvent. The system of interest in this Letter, namely, Au nanoparticles in toluene deposited onto a native oxide-terminated Si(111) substrate, has been described at length in a number of earlier papers[5,10,13,14] and here we therefore include only a brief description of the patterns formed. Figure 1 shows a subset of the different morphologies obtained. These depend on a number of factors including nanoparticle concentration, the nature of the solvent and substrate (e.g., wettability), and the length of the thiol groups used to passivate the gold particles. Understanding the physical processes that govern the self-organization of patterns like those shown in Figure 1 is an area of intense research where the interplay of simulation and experiment plays a pivotal role.



Target	A_{target}	$A_{evolved}$	A_{error}	U_{target}	$U_{evolved}$	U_{error}	χ_{target}	$\chi_{evolved}$	χ_{error}
Island	304862	308170	1.05%	72512	72198	0.43%	632	596	5.70%
Labyrinth	516669	508958	1.49%	77984	77502	0.62%	114	147	28.95%
Cell	305642	258304	15.50%	18588	24050	29.38%	5	2	60%
Worm	301378	302338	0.32%	32198	34610	7.49%	88	110	25%

Figure 4. Evolved patterns using the Minkowski functional-based fitness function. The left column shows the target, i.e., experimental, images. The right column shows self-organized patterns mimicking the experimental data. These patterns were evolved using the evolutionary algorithm described in the main text. The table shows the specific Minkowski values for the area (A), perimeter (U), and Euler characteristic (χ) for both the experimental target and evolved images as well as the discrepancy, i.e., % error, between the two.

Our simulations[5,15] are based on a two-dimensional Monte Carlo (Metropolis algorithm) model introduced by Rabani et al.[4] The solvent is represented as an array of cells on a square grid, each of which represents 1 nm^2, and can have a value of either 1 or 0 to represent liquid or vapor, respectively. Each gold nanoparticle occupies an area of 3 × 3 cells, and liquid is excluded from the sites where a particle is present. The simulation proceeds by two processes: the evaporation (and recondensation) of solvent and



Target	Average fitness	Best fitness	Worst fitness	Fitness deviation
Island	111.9305	460.6708	36.07952	126.0549
Labyrinth	36.51094	84.49821	2.851095	27.09551
Cell	2.742406	3.632334	1.934135	0.427782
Worm	20.88366	50.97714	3.824032	17.15378

Figure 5. Population dynamics of the genetic algorithm. Each experimental image was used as a target in ten independent runs of the GA. Parts a−d show the average population fitness as a function of time ("generations") of each run as well as the average evolution (dark line). The table shows, for each experimental target, details of the fitness achieved by the winning individual in each of the ten runs.

the random walks of nanoparticles. The Metropolis algorithm is governed by the following equations

$$P_{\text{accept}} = \min\left(1, \exp\left(\frac{-\Delta H}{k_{\text{B}}T}\right)\right) \quad (1)$$

$$H = -\epsilon_{\text{l}} \sum_{\langle i,j\rangle} l_i l_j - \epsilon_{\text{n}} \sum_{\langle i,j\rangle} n_i n_j - \epsilon_{\text{nl}} \sum_{\langle i,j\rangle} n_i l_j - \mu \sum_{\langle i\rangle} l_i \quad (2)$$

where p_{accept} represents the probability of acceptance of an event, ϵ_{l}, ϵ_{n}, and ϵ_{nl} determine the liquid−liquid, nanoparticle−nanoparticle, and nanoparticle−liquid interactions, respectively, and μ is the chemical potential of the liquid, which defines its equilibrium state.[15] These parameters determine the nature of the pattern formed as output.

In order to program the simulated self-organized patterns to match as closely as possible those observed experimentally, we couple the simulator to a genetic algorithm that will tune these parameters. GAs are the mainstay of evolutionary computation and one of the most powerful and widely used methods in the optimization and machine-learning toolbox. They are particularly suited to optimization problems involving very large search spaces and/or complex objective functions which are not amenable to traditional numerical analysis. First proposed in the 1970s by John Holland,[16] GAs have earned great popularity both for their conceptual simplicity and power, as exemplified in a great many practical applications,[17−22] and for their theoretical foundations.[23−25] A genetic algorithm maintains a set of vectors, called a population of individuals, where each vector represents a particular set of input parameters for the simulator. Each vector is passed onto the simulator and the resulting self-organized pattern compared against the experimental target, evaluated, and assigned a "fitness" value. Fit individuals "breed" preferentially. Thus good traits (parameters) present in specific vectors accumulate and, over time, the average quality of the population increases.

In order to coerce the Monte Carlo simulator into producing a particular morphology, a method of measuring similarity between self-organized patterns must be used. In this paper we employ Minkowski functionals.[7] These characterize a binary pattern in terms of area, perimeter, and Euler characteristic (a measure of connectivity) (see Figure 2). The objective function that the GA is set to minimize is derived by taking the root mean squared error (RMSE) between the target Minkowski values and those derived from the evolved patterns. Hence the fitness of an individual can be seen as the reciprocal of this RMSE value (as plotted in Figure 5). As the simulation is intrinsically stochastic, each individual, i.e., parameter vector, must be evaluated a number of times, hence the use of mean errors. Also, as each Minkowski functional can take values over widely different intervals,

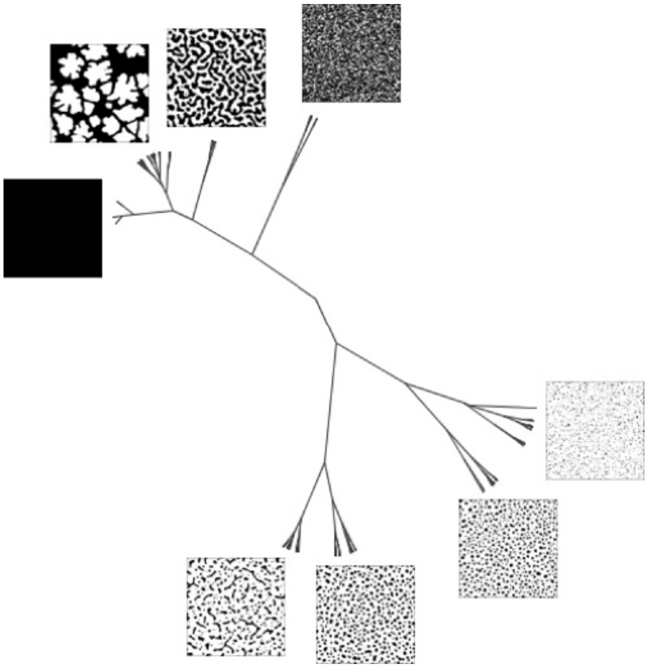


Figure 6. A partial depiction of the logarithmic cluster tree for the 256-piece dataset.

we scale each functional to the [0,1] interval so as to give each of them equal weighting within the fitness function. The GA is initialized with a randomly generated population, i.e., multiset, of vectors and we let the evolutionary process take its course for a number of generations. A generic GA pseudocode, along with parameters from our system, is shown in Figure 3.

To test the methodology decribed above, we defined a set of four patterns, each demonstrating different morphological families, taken from *experimental images* (see Figure 4). These four patterns were the "targets" that the GA needed to reverse engineer by finding a suitable set of parameters for the MC simulator. For any of the given targets, the GA was run for 100 generations using a population of 20 individuals. Each individual comprised a candidate parameter set for the MC simulator. On each target pattern we run the GA ten times.

In every case, the simulator was run for 1000 Monte Carlo cycles. Figure 4 shows representative results from the GA runs that are characterized by the striking similarity to their respective targets; the results for the island and labyrinth targets are particularly good and taking into account that both the experimental and simulated patterns arise from a stochastic process (i.e., for a given parameter set two distinct runs will produce similar yet not identical behaviors) the cell and worm patterns are also remarkably close to their experimental objective.

As shown in the evolution graphs and the statistics shown in Figure 5, each run followed a similar evolutionary trajectory. A good (i.e., visually acceptable) result was

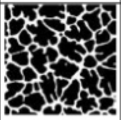
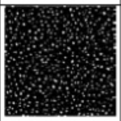
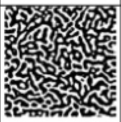
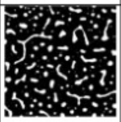
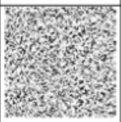
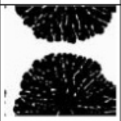
Family name	Characterstics	Example	Number of samples
Cell	Highly connected, large length scale		17
Island	Large number of unconnected, small, regular clusters		71
Labyrinth	Highly connected, small length scale		24
Worm	Disconnected, larger non-regular clusters		14
Indiscernible	No spatially correlated pattern visible		60
No pattern	Completely black (i.e., solvent saturated)		63
Unusual	Other novel patterns, including fingering morphologies[30]		7

Figure 7. Table illustrating the size of the different morphological families found in the dataset. Families containing a larger number of representative patterns are deemed more designable as it is easier for the GA to find a parameter set realizing the pattern.

obtained in each of the ten runs performed for each target, despite the often large standard deviations. Indeed, even for the "worst" runs, although the numerical fitness is substantially lower than average, the result was still visually acceptable (though not as convincing, of course, as the pattern evolved in the "best" run). This surprising feature can be best explained by performing a detailed analysis of the Minkowski-based fitness function as we recently proposed in ref 20.

We defined a dataset comprising 256 sample images representing a cross section of the entire range of simulation parameters. Clustering this dataset using the Minkowksi-based fitness function taken as a similarity measure results in a hierarchical tree that organizes simulation results based on their topological distances. Figure 6 shows the tree we obtain for our dataset. Note how each of the 14 main clusters represents a particular type of morphology. There are some clusters, such as 1 and 4 that look visually very similar, yet are quite far apart in the tree; this draws attention to the fact that the Minkowski functionals are often more sensitive than human vision, i.e., two images that look similiar can have quite different Minkowksi values. This result provides an explanation for the observation made above that even for results with a numerical fitness substantially lower than average, the result was still visually acceptable.

Particularly interesting to note is that the cluster analysis shows that the search space can be partitioned into a number of "families" of morphological likeness. A simple manual (visual) classification of these into morphological families in Figure 7 shows the relative size of each class. We note that in general, those targets scoring higher fitness tend to be members of the larger families, that is, these patterns are more *designable*.[27] This supports the observation from our results above that the evolution of, e.g., the "island" target, achieved much higher fitnesses than the other three targets, while the "cell" target produced relatively low values. Interestingly, it has been argued that designability plays a key role in the evolution of proteins.[28,29] An analogy can be made: as it is the case for proteins where a complex sequence → structure → function mapping exists and is molded by natural selection, the self-organized nanostructures studied in this paper also present a similar mapping albeit "implemented" in a different way. That is, the nanosystems studied here can be thought as obeying the following mapping sequence: experimental conditions/MC parameters → structure: self-organized pattern → function. We argue that future implementations of intelligent self-organized surfaces could use a process of artificial selection such as that presented in this paper in order to evolve toward target *function*, rather than *structure* as done in this paper, if the desired functions were to be embodied into the more designable structures.

This work has presented evolutionary computation as a method for designing target morphologies of self-organizing nanostructured systems. We have used Minkowski functionals to direct the evolution in search of simulated patterns that closely mimic those observed experimentally. The simulation is also able to produce a number of patterns that are more uncommon in experiments, such as branched structures that are reminiscent of viscous fingering.[31,32] The obvious, albeit extremely challenging, next step is to couple the GA directly to an experiment rather than a simulator, in a fashion similar to the research currently being explored by the CHELLnet project.[33]

Acknowledgment. The authors gratefully acknowledge the support of Marie Curie Actions through their funding of Grant MRTN-CT-2004005728 and the EPSRC through the funding of Grants EP/D021847/1 and EP/E017215/1. C.P.M. was supported by an EPSRC DTA award.

References

(1) Toffoli, T.; Margolus, N. *Cellular automata machines - a new environment for modelling*; MIT Press: Cambridge, MA, 1987.

(2) Chopard, B.; Droz, M. *Cellular automata modeling of physical systems*; Cambrige University Press: Cambridge, 1998.

(3) *Santa Fe studies in the science of Complexity*; Dooler, G., Ed.; Addison Wesley Longman Publishers: Reading, MA, 1990.

(4) Rabani, E.; Reichman, D. R.; Geissler, P. L.; Brus, L. E. *Nature* **2003**, *426*, 271−274.

(5) Martin, C. P.; Blunt, M. O.; Moriarty, P. *Nano Lett.* **2004**, *4*, 2389−2392.

(6) Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, 1989.

(7) Michielsen, K.; de Raedt, H. *Phys. Rep. 347*, **2001**, 461−538.

(8) Pollack, J. B.; Lipson, H.; Ficici, S.; Funes, P.; Hornby, G.; Watson, R. In *Evolvable Systems: from biology to hardware; proceedings of the third international conference (ICES 2000)*; Miller, J., et al., Eds.; Lecture Notes in Computer Science; Springer: Berlin, 2000; pp 175−186.

(9) Ge, G.; Brus, L. *J. Phys. Chem. B* **2000**, *104*, 9573−9575.

(10) Moriarty, P.; Taylor, M. D. R.; Brust, M. *Phys. Rev. Lett.* **2002**, *89*.

(11) Narayanan, S.; Wang, J.; Lin, X.-M. *Phys. Rev. Lett.* **2004**, *93*.

(12) Bigioni, T. P.; Lin, X.-M.; Nguyen, T. T.; Corwin, E. I.; Witten, T. A.; Jaeger, H. M. *Nat. Mater.* **2006**, *5*.

(13) Blunt, M. O.; Martin, C. P.; Ahola-Tuomi, M.; Pauliac-Vaujour, E.; Sharp, P.; Nativo, P.; Brust, M.; Moriarty, P. J. *Nat. Nanotechnol.* **2007**, *2*, 167.

(14) Blunt, M. O.; Suvakov, M.; Pulizzi, F.; Martin, C. P.; Pauliac-Vaujour, E.; Stannard, A.; Rushforth, A. W.; Tadic, B.; Moriarty, P. J. *Nano Lett.* **2007**, *7*, 855.

(15) Martin, C. P.; Blunt, M. O.; Pauliac-Vaujour, E.; Vancea, I.; Thiele, U.; Moriarty, P. Unpublished. We have recently found that by using a simple modification to the chemical potential term in eq 2, it is possible to simulate classes of patterns observed experimentally but not reproduced by the standard Rabani et al. algorithm. Here we use only the original Rabani et al. algorithm[4] modified as described by Martin et al.[5] to include next-nearest-neighbor interactions.

(16) Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; University of Michigan Press: Ann Arbor, MI, 1975.

(17) Harding, S.; Miller, J.; Rietman, E. *IEEE Trans. Nanotechnol.* **2005**.

(18) Thompson. In *Proceedings of the First International Conference on Evolvable Systems*, 1996.

(19) Mitchell, M.; Crutchfield, J.; Das, R. In *Proceedings of the First International Conference on Evolutionary Computation and its Applications*, 1996.

(20) Krasnogor, N.; Siepmann, P.; Terrazas, G. In *Proceedings of the Seventh International Conference of Adaptive Computing in Design and Manufacture*, 2006.

(21) Kruska, J. B. *Proceedings of the American Mathematical Society*, (1), pp 48−50.

(22) Horn, J. In *Proceedings of IEEE Congress on Evolutionary Computation*, 2005.

(23) Shapiro, J. L. *Theoretical Aspects of Evolutionary Computing*; Springer: Berlin, 2001; pp 87−108.

(24) Poli, R.; McPhee, N. F.; Rowe, J. E. *Genetic Programming and Evolvable Machines*, **2004**, *5* (1), 31−70.

(25) Krasnogor, N.; Smith, J. E. *J. Mathematical Modelling Algorithms*, in press.

(26) Lozano, M.; Herrera, F.; Krasnogor, N.; Molina, D.

(27) Hogg, T. *Nanotechnology* **1999**, *10* (3), 300−307(8).

(28) Li, H.; Helling, R.; Tang, C.; Wingreen, N. Emergence of preferred structures in a simple model of protein folding. *Science* **1996**, *273* (5275), 666−669.

(29) Wong, P.; Frishman, D. Fold designability, distribution, and disease. *PLoS Comput. Biol.* **2006**, *2* (5).

(30) Yosef, G.; Rabani, E. *J. Phys. Chem B* **2006**, *110*, 20965−20972.

(31) Hele-Shaw, H. S. *Nature* **1898**, *58*.

(32) Pauliac-Vaujour, E.; et al. in preparation.

(33) Cronin, L.; Krasnogor, N.; Davis, B. G.; Alexander, C.; Robertson, N.; Steinke, J. H. G.; Schroeder, S. L. M.; Khlobystov, A. N.; Cooper, G.; Gardner, P.; Siepmann, P. A.; Whitaker, B. J.; Marsh, D. *Nat. Biotechnol.* October **2006**, *24* (10).

NL070773M