An Evolutionary Algorithm for the Surface Structure Problem

J. Martínez , M.F. López , J.A. Martín-Gago , and V. Martín

Abstract. Many macroscopic properties: hardness, corrosion, catalytic activity, etc. are directly related to the surface structure, that is, to the position and chemical identity of the outermost atoms of the material. Current experimental techniques for its determination produce a "signature" from which the structure must be inferred by solving an inverse problem: a solution is proposed, its corresponding signature computed and then compared to the experiment. This is a challenging optimization problem where the search space and the number of local minima grows exponentially with the number of atoms, hence its solution cannot be achieved for arbitrarily large structures. Nowadays, it is solved by using a mixture of human knowledge and local search techniques: an expert proposes a solution that is refined using a local minimizer. If the outcome does not fit the experiment, a new solution must be proposed again. Solving a small surface can take from days to weeks of this trial and error method. Here we describe our ongoing work in its solution. We use an hybrid algorithm that mixes evolutionary techniques with trusted region methods and reuses knowledge gained during the execution to avoid repeated search of structures. Its parallelization produces good results even when not requiring the gathering of the full population, hence it can be used in loosely coupled environments such as grids. With this algorithm, the solution of test cases that previously took weeks of expert time can be automatically solved in a day or two of uniprocessor time.

1 Statement and Background of the Problem

Current trends in nanotechnology require of new methods to derive the structure of materials at the nanoscale. Here we face the problem of finding the surface structure of a material: the precise position and type of atoms lying in the basic surface cell –the smallest set of atoms that, by repetition, tessellate the whole of the surface. The standard way of finding out the surface structure involves an experiment using one among several techniques (SXRD: Surface X-Ray Diffraction, LEED: Low Energy Electron Diffraction, etc). The data obtained does not directly determine the structure; it is just a signature that has to be computationally matched by solving an inverse problem. A set of atomic positions, types and other relevant parameters are proposed as a candidate solution and

then the signature is computed and compared to the experimental data. When a match is found, the structure has been determined. We are interested in SXRD data [1]: a beam of highly energetic and focused X-rays impinge on a surface and its diffraction pattern recorded at different angles. This constitutes the experimental signature to be matched by minimizing a suitable fitness parameter. Diffraction patterns obtained by adding the contributions of the many atoms that form the surface are extremely sensitive, but also lead to very complex fitness hipersurfaces. The only methods available today to solve the inverse problem for SXRD data are based on either Levenberg-Marquardt or simulated annealing [2]. None of these produce satisfactory results for complex surfaces. Other approaches have been tried, but using other experimental techniques. The most studied one is LEED, where Genetic Algorithms [3] and Pattern Search Methods [4] have been used. No research has been done to apply more sophisticated algorithms to SXRD data. In this case, the calculation of the signature is reduced to the sum of a few thousand terms, each one easily computed. This allows for a fast exploration of the search space through a population based algorithm. Since previous work with this kind of algorithms for the geometric optimization of clusters [5], akin to the present problem, were successful, evolutionary algorithms looked a promising path to improve the performance of the currently available methods.

2 Description of the Algorithm, Results and Future

The algorithm generates candidate solutions for the specific problem and tests them for matching the given experimental data. The significance of the matching is measured using a χ^2 test, standard in this field, that is also used to define the fitness. A randomly generated population is the usual starting point, although seeding is also possible as well as to impose symmetries on the individuals: an important step, since it reduces the search space. An individual encodes a candidate solution to the problem, with all the continuous and categorical variables. In its more basic form, its chromosome is a real coded set of $3n$ coordinates for the n atoms in the unit cell, their type and fractional occupancy. Individuals are then ranked through its fitness to regulate the recombination probability. Mutation and recombination steps are then applied. Mutation is either completely random, such that a given atom can be sent anywhere within the unit cell space (physically motivated to simulate mass transport) or, like in typical evolution strategies, moving the chosen coordinate using a gaussian distribution probability centered around the atom's position and with a width that is also evolving with the population [6]. Crossover operators have also several choices, the standard one and two points crossovers can be used, but the one with the best performance so far is patterned by realizing that surfaces are usually arranged in layers. Hence, a crossover operator that physically divides the surface in two and mixes the halves is normally used. Its performance can be fine tuned by allowing more than one layer. However, this raises some practical issues and usually the default two layers or one point crossover are the wiser selections.

In order to reuse the knowledge gained during the run, whenever a new individual is produced, it is indexed using an algorithm that divides the search space into n-dimensional boxes. Within the resolution of the box, a given index identifies uniquely an individual that is assumed to belong to the same attraction basin, hence no two individuals with the same index are allowed to belong to the population. This avoids revisiting places already searched while maintaining diversity. The size of the box diminishes during the execution, allowing to explore in more detail the most promising parts of the search space.

The algorithm is designed to run in a distributed environment. In its simplest way, a starting population, with a size dependent on the size of the problem to be solved, is generated. This population evolves by applying the above mentioned steps till its diversity is exhausted. The resulting successful individual is stored, and the process repeated with a new population till a full elite population of different successful individuals is obtained. This step can be done in parallel using several populations and just a set of networked computers are needed. It is a loosely coupled computation in which communications are only required after each population has been processed. When the elite population is complete, the master processor continues the algorithm. This time, after a predefined number of steps, a local search is performed using as input the current individual. Local searches are not iterated till convergence, but only a fixed number of times. Its purpose is to increase the fitness of the individuals and also to eliminate from the population those that, although having different indexes, really belong to the same attraction basin. In the meantime, the rest of the available processors continue producing elite individuals that are introduced in the population of the master processor whenever they differ in index from those being processed. Since communications from the master to the slaves is only done during the initial set up step, and then only when the parameters for the boxed classifier varies, a bottleneck cannot be produced in this direction. Communication from the slave to the master is done only when a new elite individual has been found. To produce a bottleneck in this direction would require thousands of very fast slaves. The process is asynchronous and there are no dependencies, thus scaling very well to a high number of processors.

Results have been obtained using known surfaces with low, medium and high complexity. Low and medium complexity cases —those that take of the order of days using the standard trial and error method— were usually solved in a matter of minutes to hours in one processor. For medium complexity cases other, unexpected solutions were also found. These were identified as a symmetry transformation of the correct one, being actually identical and thus demonstrating that the algorithm effectively explored well the search space. For high complexity cases the algorithm takes of the order of days. Using the standard method, they were solved using weeks of computer time and needed the guide of an expert and some extra assumptions to limit its complexity. The required guidance and the fact that only a few high complexity cases have been solved by the standard method, makes direct comparison with the present, fully automatic method, of limited value from an algorithmic perspective. However, to give some indicative

numbers of its performance, it has been applied to solve a complex problem with 45 degrees of freedom (15 atoms) in a volume of $150\mathring{A}^3$, finding the solution in an average time of around a day in one processor. A brute force approach should try all the combinations of 15 atoms in cells of, at most, $0.1\mathring{A}$ per side. A modern processor would need of the order of 10^{55} years. The standard method has not been applied to this problem but similar sized surfaces has been reported to take months of work by their authors and this after reducing the problem to optimize only the position of the outermost layer. These results are very promising and would allow to tackle problems that are currently out of reach.

Future work include, on the physics side, improving the handling of surface symmetries, domains and extending the method to use molecules. On the algorithmic side, we plan on introducing more heuristic and already have some changes underway: while studying the evolution of the algorithm, we found out that the number of times that the index of an elite individual appears can be used as a new ranking to implement a league-like [7] step. This can be used to accelerate the algorithm through a new recombination operator that would be incorporating information obtained during the run. Further improvement can be obtained by using more robust algorithms for the local minimization, in particular, a Generating Set Search Method [8] is planned, which also has the advantage of being directly applicable to the categorical variables.

In conclusion, we have presented an evolutionary algorithm tailored to the surface structure problem that attempts to find the solution without the need of an expert, as required in the approach used nowadays. The algorithm has been designed to be efficiently implemented in distributed systems. It has low communication overhead and synchronization requirements and thus can be scaled to a high number of processors. The tests performed indicate that it easily outperforms the current method, saving from days to weeks of work to the user.

The authors acknowledge the computer resources and assistance provided by Centro de Supercomputación y Visualización de Madrid (CeSViMa).

References

1. Feidenhans'l, R.: Surface structure determination by X-ray diffraction. Surface Science Reports 10, 105–188 (1989)
2. Vlieg, E.: ROD: A program for surface X-ray crystallography. J. of Appl. Cryst. 33, 401–405 (2000)
3. Döll, R., van Hove, M.A.: Global optimization in LEED structure determination using genetic algorithms. Surface Science 355, L393 (1996)
4. Zhao, Z., Meza, J., Van Hove, M.A.: Pattern Search Methods for Surface Structure Determination of Nanomaterials. J. Phys. Cond. Matter 18, 8693–8706 (2006)
5. Wales, D.J., Scheraga, H.A.: Global Optimization of Clusters, Crystals, and Biomolecules. Science 285, 1368–1372 (1999)
6. De Jong, K.: Evolutionary Computation: A Unified Approach. MIT Press, Cambridge (2006)
7. Rehr, J.J.: Nanostructures in a new league. Nature 440, 618–619 (2006)
8. Kolda, T., Lewis, R., Torczon, V.: Optimization by direct search: New perspectives on some classical and modern methods. SIAM Review 45, 385–482 (2003)