NANOSYSTEMS: PHYSICS, CHEMISTRY, MATHEMATICS, 2013, 4 (4), P. 512–523

GENETIC ALGORITHM FOR CONSTRUCTING GRAPHENE NANORIBBON WITH GIVEN ELECTRONIC TRANSPORT PROPERTIES

I. S. Lobanov, A. I. Trifanov, E. S. Trifanova

St. Petersburg National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, Russia

lobanov.igor@gmail.com, alextrifanov@gmail.com, etrifanova@gmail.com

PACS 72.80.Vp, 73.21.Hb, 73.22.Dj, 73.22.Pr, 73.63.Nm

Electronic transport in carbon nanoribbon is studied in a quantum graph model. A numerical method for current-voltage curve calculation is proposed. Various optimizations of a parallelization scheme are discussed. A parallel genetic algorithm to solve an inverse transport problem is invented.

Keywords: quantum graph, electronic properties, genetic algorithm, graphene nanoribbon.

1. Introduction

In last decade, carbon nanoribbons have proven to be one of the most promising nanosystems for microelectronics [10]. There is a huge number of works devoted to the construction of diodes, transistors, antennas and other devices based on nanoribbons, see e.g. recent publications [7–9]. Unfortunately, the theoretical prediction of properties of non-trivial geometry nanoribbons a is very complicated task, which will probably not be solved for decades. For that reason, nanoribbons are commonly simulated numerically. The most precise methods belong to the family of ab-initio methods and are applicable to the simulation of systems with dozens of atoms. Methods more appropriate for practical usage are based on density functional theory, which give results quite satisfactorily with experimental data, and are appropriate for use with systems not larger than few hundreds of atoms [6]. The only method suitable for the simulation of systems with many thousands of atoms is based on the tight-binding model [12]. However, even the tight-binding model requires sophisticated implementation to simulate millions atoms [11, 13].

In the present work, we describe a high-performance simulator of carbon nanoribbons based on a quantum graph model, which is a more precise model than the tight-binding approximation. We describe the parallelization of the proposed algorithm, which is capable of analyzing systems of millions atoms using existing supercomputers. A variant of genetic algorithm for solution of inverse transport problem is provided. Crossover and mutation operations are modified to fit the underlying physical problem.

2. Theoretical background

For convenience, we recall the basics of the quantum graph model, see for details [1–5]. By definition, a quantum graph is a collection of line segments glued at the ends together with Schrödinger operator on them. Almost everywhere, the quantum graph is a one dimensional manifold with the exception of gluing points. Hence, the motion of a

spinless nonrelativistic particle is described by the convenient one-dimensional Schrödinger operator on every edge:

$$-\phi_e'' + q_e\phi_e = E\phi_e, \tag{1}$$

where ϕ_e and q_e are restrictions of the wave function ϕ and the scalar potential q to the edge e, and E is energy. At the gluing point, the behavior of the wave function ϕ is determined by boundary conditions, which are in most cases chosen to be Kirchhoff conditions:

$$\phi_{e_1}(v) = \ldots = \phi_{e_k}(v), \quad \sum_{n=1}^{k} \phi'_{e_n}(v) = 0,$$

where $e_1, \ldots, e_k$ are all edges having common end v, $\phi_e(v)$ is the value of the wave function at the vertex v, and $\phi'_e(v)$ is the derivative of the wave function ϕ at the vertex v along external normal to the edge e.

Due to the existence and uniqueness theorem for the solution of ordinary differential equations. Equation (1) can be written as system of two algebraic energy dependent equations:

$$D_e(E)\begin{pmatrix}\phi_e(0)\\ \phi_e(l)\end{pmatrix} = N_e(E)\begin{pmatrix}-\phi'_e(0)\\ \phi'_e(l)\end{pmatrix},$$

where 0 and l denote the ends of the edge e, and the square matrices D_e and N_e define Dirichlet-to-Neumann (DN) mapping (also called Weyl function or Krein Q-function). As proven in the theory of boundary triples, the DN mapping can be chosen satisfying the following conditions:

(1) D_e and N_e are entire functions of the variable E;
(2) $N_e^{-1}D_e$ and $D_e^{-1}N_e$ are meromorphic functions of E with simple poles at the Neumann and Dirichlet spectra, respectively;
(3) eigenvalues of $N_e^{-1}D_e$ and $D_e^{-1}N_e$ are monotonic functions of E on every real interval of continuity.

For a vanishing scalar potential, the DN mapping is known explicitly:

$$D(E) = \sqrt{-E}\begin{pmatrix}\alpha & -\frac{1}{\alpha}\\ -\frac{1}{\alpha} & \alpha\end{pmatrix}, \quad N(E) = \begin{pmatrix}\alpha & \frac{1}{\alpha}\\ \frac{1}{\alpha} & \alpha\end{pmatrix}, \quad \alpha = \exp\left(\frac{i}{2}\sqrt{-E}\right).$$

We denote by F vector of values of wave function ϕ at all ends of all segments of the quantum graph, and denote by F' the vector of the external derivatives. Then, in terms of DN mapping, the Schrödinger equation can be written as

$$D(E)F = N(E)F', \tag{2}$$

and boundary conditions can be written as

$$BF = CF'. \tag{3}$$

It is worth noting that all matrices B and C satisfying $BC^* = CB^*$, such that the matrix $(B|C)$ has maximal rank, define self-adjoint boundary conditions, and all self-adjoint boundary conditions can be written in the form (3). Moreover, without loss of generality, B and C can be chosen to be self-adjoint and B can be non-negatively defined. To solve the Schrödinger equation for the energy E, it is convenient to consider matrix

$$R(E) = CN^{-1}(E)D(E) - B,$$

since E is an eigenvalue of the quantum graph, if and only if $R(E)$ is degenerate or E belongs to the Dirichlet spectrum. The eigenvalues of R are real and monotonous functions of E on every segment not containing points of Dirichlet spectrum, hence computation

of zeros of the eigenvalues as functions of E is a relatively simple problem. However, computation of the eigenvalues of $R(E)$ is a slow operation of complexity $O(n^3)$, where n is the number of edges in the quantum graph. The complexity can be significantly reduced in some cases, since matrices $D(E), N(E)$ and often B and C are sparse; the corresponding divide-and-conquer algorithm is outlined below.

Consider a subgraph of the quantum graph. We call a lead every end of a segment glued to an edge not belonging to the subgraph. Restriction of a solution of the stationary Schrödinger equation on the quantum graph to the subgraph should satisfy Equations (2) and (3) for the appropriate matrices B and C, but in this case, the matrices are not of the maximal rank, since we have no boundary conditions at leads [1]. Solving the system of equations with respect to the values of the wave function and derivatives of the wave function at vertices having attached boundary conditions, we obtain smaller system again of the form (2), but vectors F and F' contain in this case only values at leads. The functions D and N in the newly obtained system define Dirichlet-to-Neumann mapping for the subgraph and have all mentioned above properties of DN mappings. The complexity of the DN mapping calculation by Gaussian elimination is $O(m^2(m-n))$, where m is the number of all ends of all edges in the subgraph, and n is the number of leads.

The proposed divide-and-conquer algorithm for the DN mapping calculation for a subgraph is based on splitting the large subgraph graph into smaller subgraphs, which are then divided into even smaller subgraphs and so on; the DN mapping for smallest subgraphs (edges and vertices) are known exactly, the DN mappings for larger subgraphs are calculated using the above-mentioned elimination process using the DN mappings for smaller subgraphs. Clearly, the complexity of the algorithm depends on the way the splitting is done. We denote by δ the supremum of logarithm of ratio of the number of ends lying on a sphere in the quantum graph to the number of ends lying in the ball bounded by the sphere. Assuming that we divide every subgraph into two parts with halved diameter, the complexity of the algorithm is estimated as $O(m^{3\delta}) + O(m \ln m)$, where m is the number of edges in the largest subgraph [1]. For example, a $\mathbb{Z}$-periodic graph with afinite fundamental domain, the complexity is $O(m \ln m)$, for a $\mathbb{Z}^2$-periodic graph with finite fundamental domain, the complexity is $O(m^{3/2})$.

Now, we consider a transport problem for a quantum graph consisting of a compact part and attached semi-infinite segments with constant potentials. As is well-known, the scattering matrix of the system is given by [1]

$$S(E) = (D(E) - T(E)N(E))^{-1}(D(E) + T(E)N(E)), \quad T(E) = \mathrm{diag}\sqrt{V - E}, \qquad (4)$$

where V is the vector of scalar potentials on semi-infinite segments, $\mathrm{diag}V$ denotes the diagonal matrix with the diagonal V. Provided $D(E)$ and $N(E)$ are already calculated, the complexity of the scattering matrix calculation is equal to sum of complexities of matrix inversion and multiplication, which are less than $O(n^3)$, where n is the number of all leads.

3. Quantum graph model of graphene nanoribbon in elecric field

The calculation of transport properties for a quantum mechanical system is generally a complex theoretical and numerical problem, therefore, simplified models are generally used. Current state of artdensity functional theory is used to model systems with hundreds of atoms, while the tight binding approximation is used to model systems with millions of atoms. In the present work, we use the quantum graph model, which is slightly slower, but it takes into account the continuous character of electron motion. The tight binding approximation can be considered as a special case of the quantum graph model.

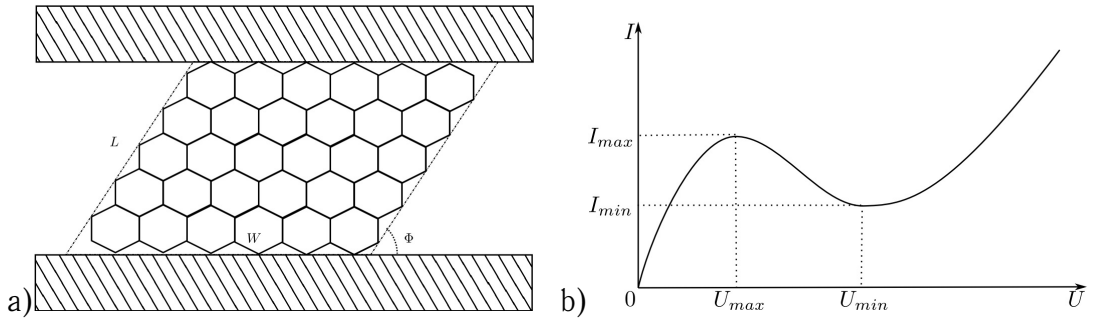

FIG. 1. (a) Graphene nanoribbon. (b) Typical current-voltage characteristics of parallelogram shaped nanoribbon.

One of the most challenging transport problems is transport in the presence of an external electric field, which often breaks the symmetry of the problem. Howeve,r for the quantum graph model, consideration of the electric field is relatively simple, since:

(1) The electric field is an explicit parameter of the model. Variation of the field does not imply recalculation of auxiliary parameters, introducing additional assumptions and so on.
(2) Calculation of the scattering matrix does not require one to solve the spectral problem. In fact, the transport problem is easier to solve than the spectral problem.

Since all transport properties can be expressed in terms of the scattering matrix, we focus on the calculation of the scattering matrix, and as an example, we consider computation of current-voltage characteristics.

Consider a parallelogram shaped graphene nanoribbon with zigzag sides attached to two electrodes, see Figure 1 (a). We denote by W the number of benzene rings lying on one electrode, by L the distance between electrodes in benzene rings and by Φ the cutting angle; let $T = \tan\Phi$. In the quantum graph model, the electron moves along segments representing chemical bonds, and the segments are glued at the carbon atoms. Boundary conditions are assumed to be Kirchhoff conditions. We model the electrodes by semi-infinite segments. To avoid unrealistic interface states, we attach separate semi-infinite segments to every atom lying on an electrode.

Typical behavior of the current-voltage characteristics (IV curve) is shown in Figure 1 (b). In applications, only key features of IV curve are of importance, e.g. in the considered case the first local maximum (U_{max}, I_{max}) and the first local minimum (U_{min}, I_{min}) should be taken into account.

For the quantum graph model we use the convenient Landauer formula to calculate conductance:

$$\frac{e^2}{\pi\hbar}\sum_{s,d}|S_{ds}|^2 - |S_{sd}|^2,$$

where s runs over all indices of the semi-infinite segments corresponding to one electrode, and d, to another. The crucial part of the computation is calculation of the scattering matrix by the formula (4), which requires one to compute the Dirichlet-to-Neumann mapping. As mentioned before, the complexity of the DN mapping computation highly depends on how the quantum graph is split into subgraphs. Below, we provide explicit splitting which is best suited for long narrow ribbons.

The considered nanoribbon can be divided into primitive parts as shown in Figures 2 and 3. The nanoribbon is divided to collection of chains, where all the elements of every

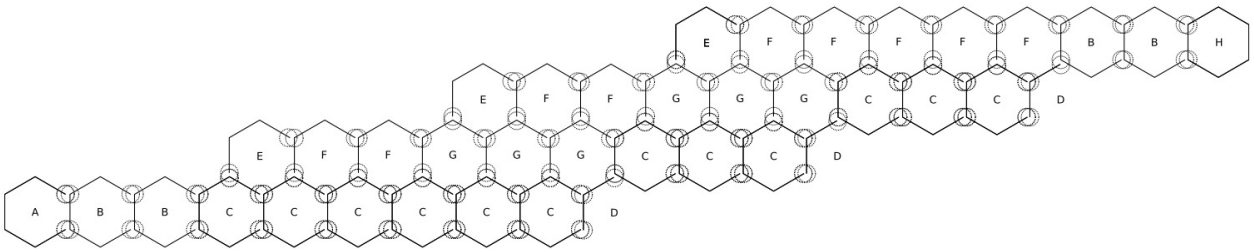

FIG. 2. Splitting of the nanoribbon to primitive pieces.

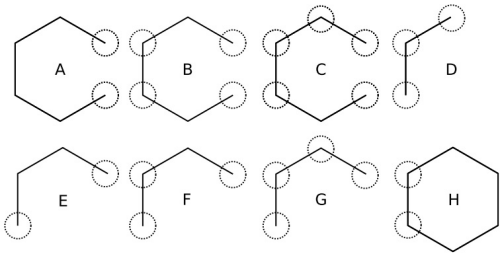

FIG. 3. List of primitive pieces of graphene nanoribbon.

chain are located a fixed distance from electrodes and therefore, have the same potential. To calculate the DN mapping for one chain, we first calculate the DN mapping for two segments having distinct angles with electric field, then compute the DN mappings for all primitive pieces, compute the DN mappings for the chains containing pieces of one type, and finally compute the DN mapping for the whole chain. The computation of the DN mapping for the chain of equal elements has a complexity $O(\ln m)$, where m is the length of the chain, since all subchains of equal length have equal DN mappings. Since the number of leads of the primitive pieces and the number of the primitive pieces are fixed, the overall complexity of computing the DN mapping for one chain is $O(\ln W)$. Every chain has more than W ends, and no more than W ends are leads (more precisely $2(W - [T])$), hence the complexity of gluing two chains together is less than $O(W^3)$. Since the total number of chains is L, the complexity of the calculation of the DN mapping for the whole nanoribbon is $O(LW^3)$. It is worth noting that the proposed splitting gives linear complexity with respect to length L of the ribbon, which is better than the above-given theoretical estimate $O(L^{3/2}W^{3/2})$. Hence, the provided splitting is best suited for very long ribbons.

For the case of a short and wide nanoribbon, one can split the quantum graph into narrow chains of length L and then glue them. Such splitting gives us complexity $O(WL^3)$. The worst case is a ribbon with equal W and L, where two provided partitions leads to the algorithm of complexity $O(L^4)$.

The computation time of current for a given voltage for the nanoribbon using the method described in the previous section is several hours for nanoribbon with W ca. 100 and L ca. 1000. Hence, for the simulation of large graphs and the use of genetic algorithms, one should speed up computations, which can be done using both sophisticated algorithms for matrix operations and parallel computations. Optimization of matrix operations is a well-studied subject, see e.g. [14], documentation on LAPACK and so on. Here, we discuss opportunities for parallelization. The main observation is the independence of the Dirichlet-to-Nuemann mapping calculations for distinct subgraphs, hence computations for such subgraphs can be performed concurrently. But, one should take into account that to

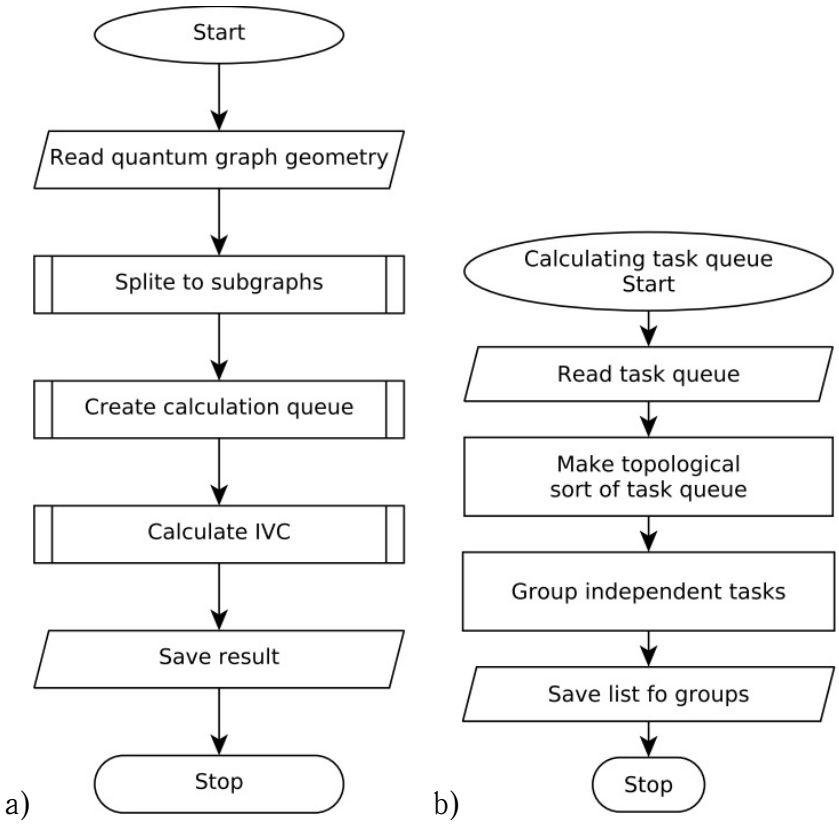

FIG. 4. (a) Flowchart for the I-V curve calculation. (b) Flowchart for the queue generation

compute the DN mapping for a subgraph, one should preliminarily compute the mappings for all parts of the subgraph, hence subgraphs form an hierarchy, where computations on lower levels must be done prior to higher ones. Finally, to perform parallel a computation of the DN mapping for the whole quantum graph, one should divide the graph into subgraphs, calculate the dependence of the graphs, collect subgraphs to groups which can be analyzed simultaneously and execute all groups in correct order. It is worth noting that the memory requirement should be taken into consideration while forming computation groups, since the storage of all temporary buffers in memory simultaneously requires much more resources than available on modern computers. That means that some kind of dynamic memory allocation should be done. Unfortunately, contemporary parallel memory managers are to slow for such tasks. To overcome this difficulty, we have implemented a memory manager that preallocates all the memory by using overlapping regions, which however will never be accessed simultaneously under the given order of DN mappings calculation. Further opportunities for parallelization are the parallel matrix operations and parallel calculation for different energy and electric field values. Flowcharts for the parallel calculation of I-V curve for the nanoribbon are shown on Figures 4-8.

4. Inverse transport problem

The inverse transport problem is to recover system geometry based on given transport properties. We are going to consider the current-voltage curve, which is one of the most important transport properties. Solution of the inverse problem is extremely complex

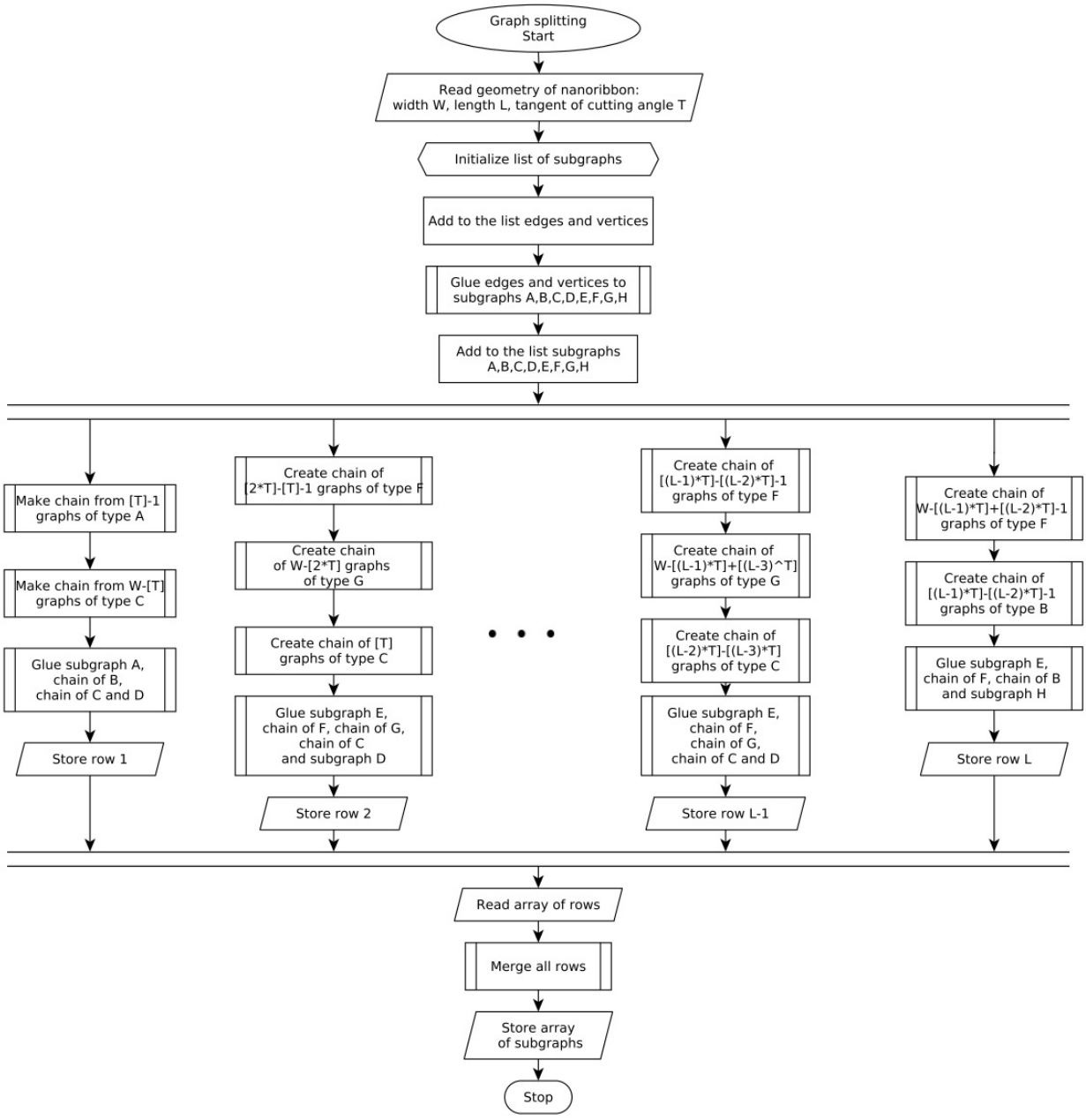

FIG. 5. Flowchart for the nanoribbon splitting

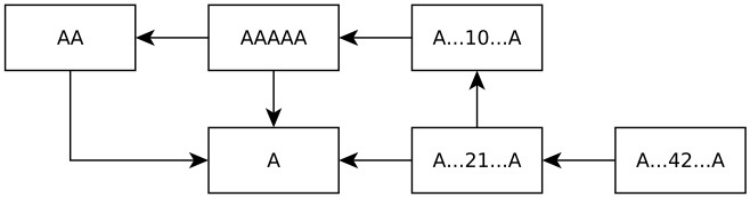

FIG. 6. Demonstration of optimal splitting of chain of 42 graphs of type A to subgraphs

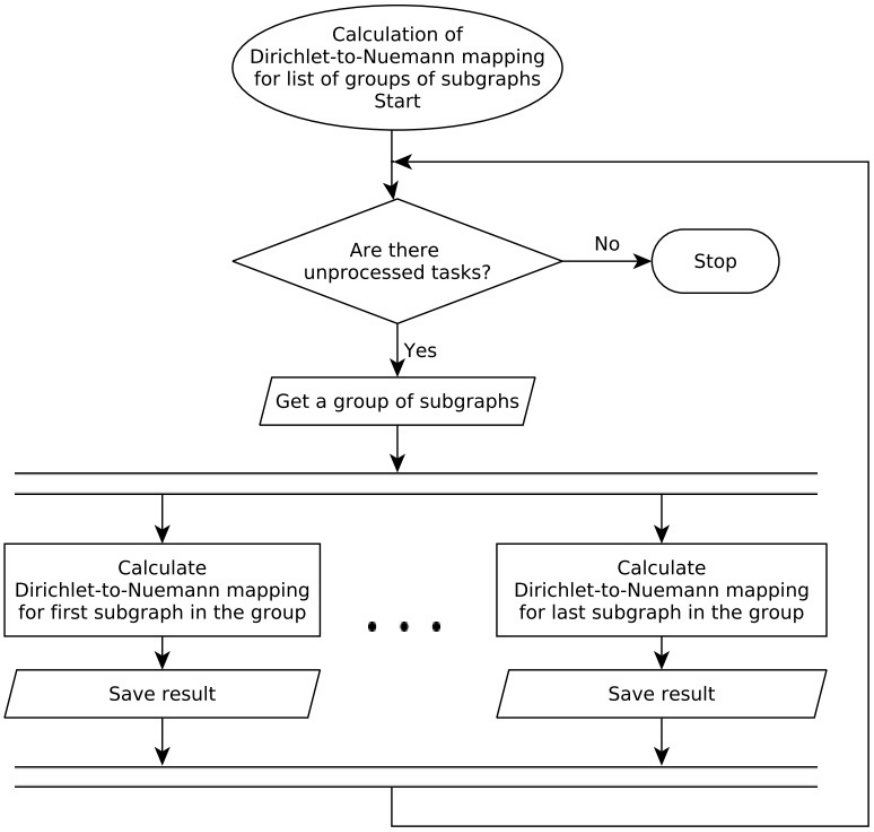

FIG. 7. Flowchart for the execution of enqueued calculations

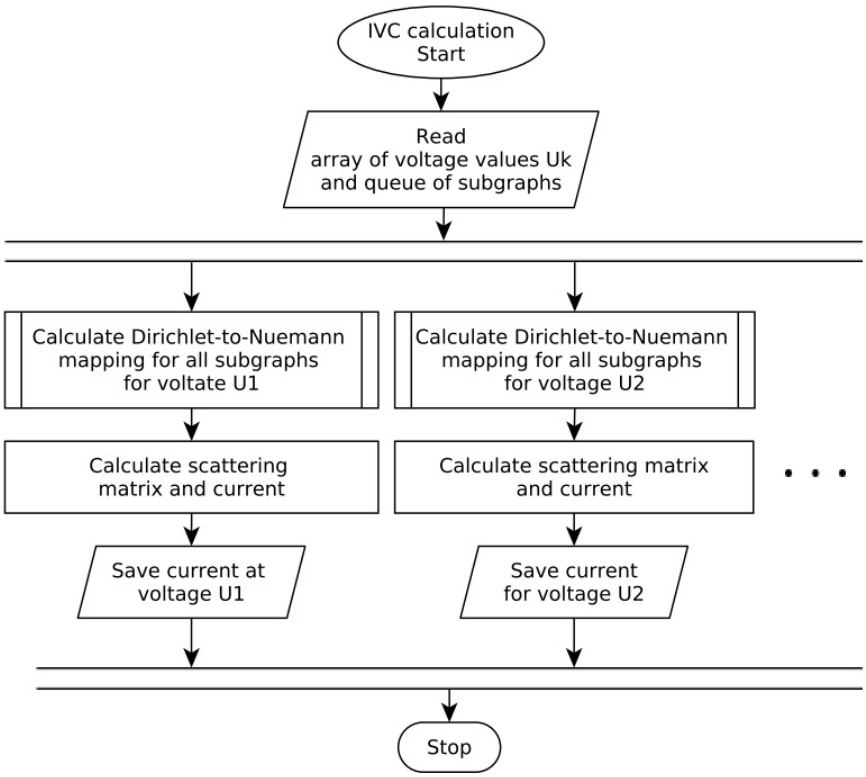

FIG. 8. Flowchart for the current-voltage characteristics calculation

and has no general solution at the moment. However, in the present work we propose a genetic algorithm to construct a nanoribbon with properties close to the given I-V curve.

Due to the high complexity of calculating the fitness function in the problem, only parallel algorithms deserve consideration. At the moment, there are four main classes of parallel genetic algorithms: global master-slave, global fine-grained, coarse grained and hierarchical. The fine-grained algorithm was implemented, since it has good scalability as we show below. We divide the whole population to subpopulations, which are processed concurrently on different processing nodes. To avoid degeneracy and to balance the load, individuals are exchanged between subpopulations. Depending on the timing of when individuals are exchanged, the algorithm can be synchronous or asynchronous. The island model was chosen for implementation, since it can be adapted to arbitrary topology.

We use processes of two types: the first type computes mutation, crossover and selection, while the second type assists ones of the first type by computing the fitness function (the I-V curve). Clearly, the number of processes in an ideal situation should coincide with the number of individuals. One ”master” process is selected, which should send stop signal to other nodes as soon as the desired fitness is achieved.

Every assisting node waits until the ribbon geometry is received. Upon receipt of the gometry, the I-V curve is computed as described in the previous section, then the fitness is calculated as the distance between the obtained curve and the desired one. The fitness and the I-V curve are sent back to the evolution computing nodes. If the stop signal is received, then the process terminates, otherwise the process again waits for input.

The stop conditions are only checked by the designated node. If desired fitness value is achieved or the maximum number of iterations is done, the designated node sends a stop signal to all nodes and prints the individual that has the geometry closest to the desired current-voltage curve displayed.

The input parameters of the algorithm are positions of the first local maximum (V_{max}, I_{max}) and the first local minimum (V_{min}, I_{min}) of the I-V curve. The results of the computation are the width W, the length L and the cutting angle Φ of the nanoribbon. The set of the parameters can be extended to a wider class of geometries, e.g. one can append intrusions.

The genetic algorithm starts with the generation of a subpopulation on every island. Every individual is a nanoribbon geometry, which in our case is described by the values W, L and Φ. On initialization, the geometric parameters are generated randomly using a uniform distribution, where the maximum values of W and L are bounded by available processing power and physical motivations.

Further randomly chosen individuals mutate. The number of mutating individuals has a constant ratio to the size of subpopulation; the ratio is a parameter of the algorithm. The mutation operation has the following steps: geometric parameters subjected to mutation are chosen randomly; a value is appended to each of the parameters normally distributed; the obtained parameters are checked for correctness; if the geometry is invalid, the individual is eliminated.

In the island model, crossover is done only inside subpopulations. To improve algorithm convergence, the crossover (inside) population makes use of a simplified model. Roughly speaking, the nanoribbon can be considered as an electrical circuit. The I-V curves for parallel and series circuits are well known and can be used for a fast, rough estimate of the results for gluing nanoribbons together. Hence, during crossover, the inside subpopulation I-V curves for all individual pairs in the subpopulation are estimated

using classical methods, and from these, the most promising pairs are chosen to produce offspring.

As well known genetic algorithms for small populations tend to converge to a local maximum of the fitness function, which for complex fitness functions, is most likely not the global maximum. To avoid this trap, the population size must be increased, or in the island model, individuals must be exchanged between islands. Experiments show that the best convergence is obtained When the best individuals exchange islands. Since the fitness computation time varies from several seconds to several hours, depending on the geometry of the nanoribbon, an iteration of genetic algorithm on one island may be several times slower than the iteration on another island. Therefore, the individual exchange must be asynchronous. Every island, after the fitness calculation, sends a fixed number of its best individuals to a different, randomly chosen island. Before the crossover, every island checks, if there are migrants waiting, which are appended to the subpopulation.

Finally, if the size of a subpopulation becomes larger than the predefined value, the worst individuals are eliminated to decrease the subpopulation size. Hence, the size of every subpopulation varies, but cannot become larger than the fixed value determined by the available computation power.

5. Parallel genetic logarithm convergence

The inverse transport problem is one of complex optimization, with lots of local optima. Separation of the population into subpopulations improves the convergence to an optimum, but the obtained solution may be far from a global optimum. If a subpopulation comes close to a local optimum, the subpopulation starts to degenerate, which leads to a waste of computational resources. To eliminate such a situation, we mix the population by exchanging the best individuals between islands. In the present section, we estimate the convergence rate of such an algorithm.

Individuals exchanged in a real situation depends on topology, but for simplicity, we assume that the connection graph is complete.

Let f^0 be the mean value of the fitness function in the initial population. We assume that after one iteration, the mean value of the fitness function is improved p times, that is

$$f^{k+1} = p \cdot f^k = p^k \cdot f^0.$$

Hence, to obtain the desired value F for the fitness function, we should make $n = \log_p \frac{F}{f^0}$ iterations.

Now, we consider several populations exchanged by individuals. We assume that for every subpopulation the estimate above is valid, Hence, without exchange, an increase in the population size gives no speed up in the convergence. We now consider the improvement of the fitness function after the exchange. We denote by f_j^k the mean value of the fitness function in a subpopulation j on an iteration k. We estimate the convergence by a geometric progression, that is, we assume $f_j^{k+1} = s_j f^k$ for some parameter $s_j = p \cdot q_j$. Let M be the number of all possible distinct individuals, N be the number of individuals in every subpopulation. Taking into account the quality of individuals to exchange, we get the estimate $q_k = p^\beta$, $0 < \beta < 1$, for non overlapping populations. Taking into account the existence of identical species in distinct subpopulations, we get estimate

$$s_k = p^{\beta \sum_{j=1}^m P_j}, \quad P_j = \prod_{i=1}^{J} \prod_{j=0}^{N-1} \frac{M - i \cdot N - j}{M}.$$

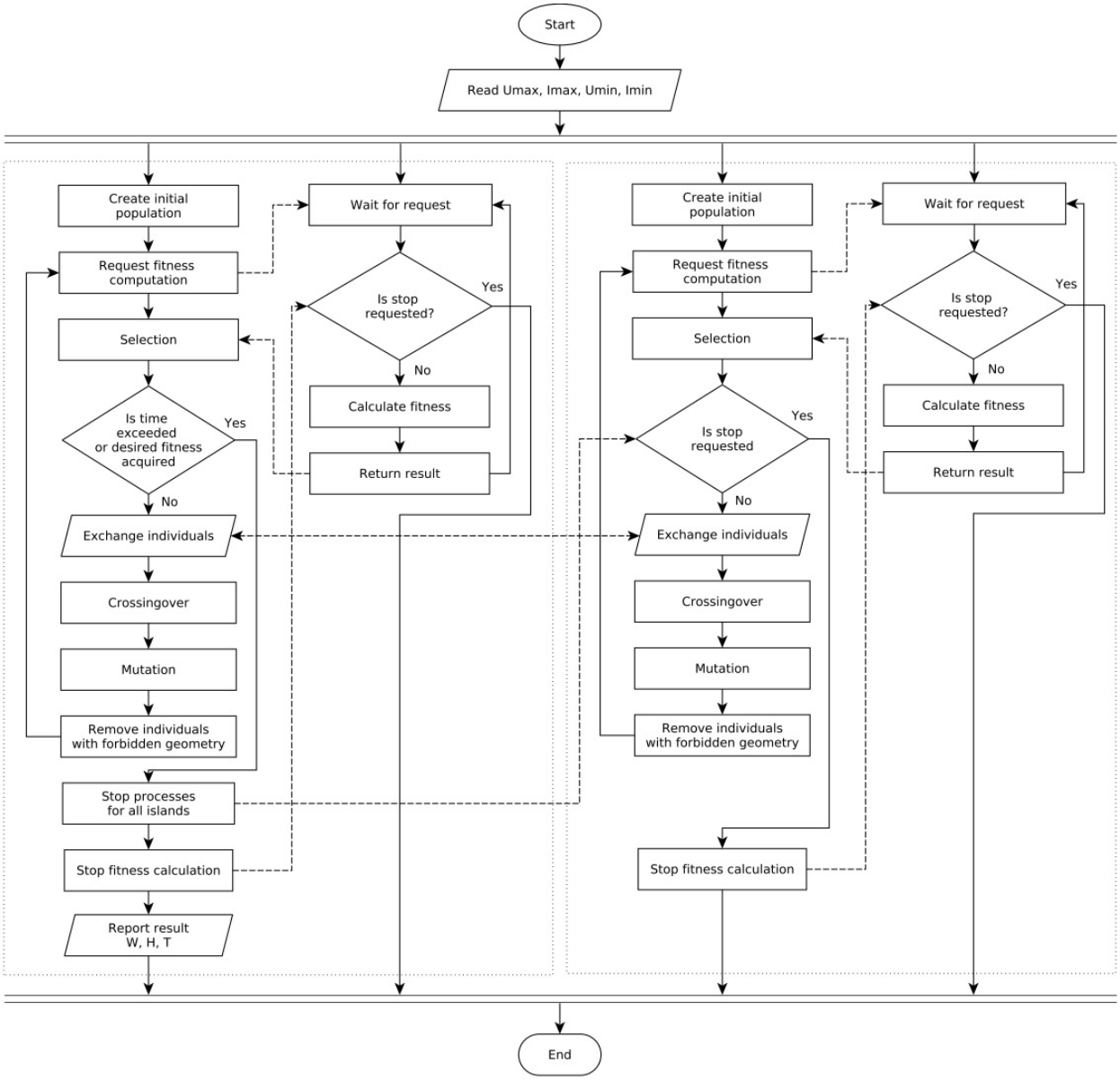

FIG. 9. Flowchart of parallel genetic algorithm

Using latter estimate, we obtain the number of steps required to achieve the desired fitness F using k subpopulations:

$$n_j = \frac{\log_p \frac{F}{f_j^0}}{\beta \cdot \sum_{k=1}^{m} P_k}.$$

Hence, the parallel algorithm gives the following for rate enhancement:

$$S(m) = \frac{\min n_j}{n}.$$

Estimates of the rate enhancement are shown on Figure 10 for parameters $N = 1000$, $\frac{F}{f_j^0} = 10^3$, $p = 1.1$, $\beta = 0.058$.

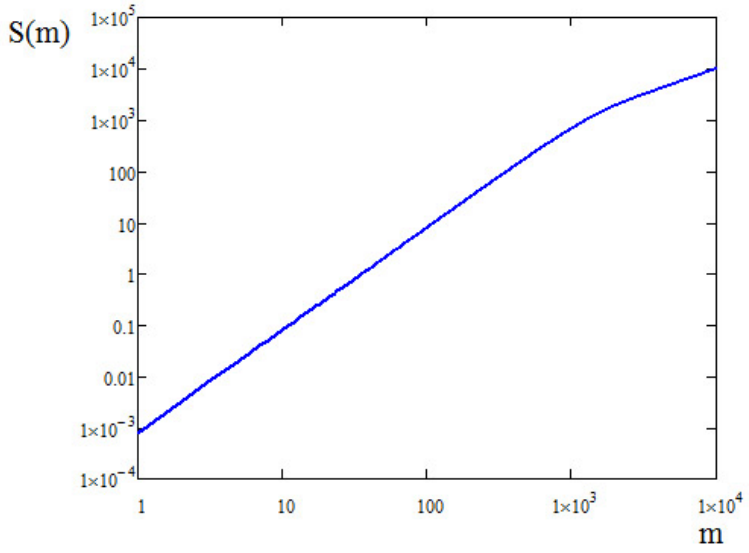

FIG. 10. Speed up of parallel genetic algorithm as function of number of processors

Acknowledment

The work is supported by state contract 07.514.11.4146.

References

[1] I.S. Lobanov, E.S. Trifanova. Direct and inverse problems for quantum graph model. *Nanosystems: Physics, Chemistry, Mathematics*, **3** (5), P. 6–32 (2012).
[2] P. Kuchment. Graph models of wave propagation in thin structures. *Waves in Random Media*, **12**, (4), P. R1–R24 (2002).
[3] U. Smilansky. Discrete graphs — a paradigm model for quantum chaos. *Seminaire Poincare XIV*, P. 89–114 (2010).
[4] *Analysis on Graphs and its Applications*, Ed by. Exner P. et al. Proc. Symp. Pure Math., AMS, 2008.
[5] *Quantum Graphs and Their Applications*, Ed. By Berkolaiko G. et al. Contemp. Math., 2006, V. 415.
[6] P. Jadaun, B.R. Sahu, L.F. Register, S.K. Banerjee. Density functional theory studies of interactions of graphene with its environment: Substrate, gate dielectric and edge effects. *Solid State Communications*, **152** (15), P. 1497–1502 (2012).
[7] G. Deligeorgis, F. Coccetti, G. Konstantinidis, R. Plana. Radio frequency signal detection by ballistic transport in Y-shaped graphene nanoribbons. *Appl. Phys. Lett.*, **101**, P. 013502 (2012).
[8] C. Cocchi, D. Prezzi, et al. Optical Excitations and Field Enhancement in Short Graphene Nanoribbons. *The Journal of Physical Chemistry Letters*, **3** (7), P. 924–929 (2012).
[9] W.S. Hwang, K. Tahy, et al. Transport properties of graphene nanoribbon transistors on chemical-vapor-deposition grown wafer-scale graphene. *Appl. Phys. Lett.*, **100**, P. 203107 (2012).
[10] O.V. Yazyev. A Guide to the Design of Electronic Properties of Graphene Nanoribbons. *Acc. Chem. Res.*, Article ASAP (2013).
[11] Miao Gao, Gui-Ping Zhang, Zhong-Yi Lu. Electronic transport of a large scale system studied by renormalized transfer matrix method: application to armchair graphene nanoribbons between quantum wires. `arXiv:1305.6682`.
[12] Y. Wu, P.A. Childs. Conductance of Graphene Nanoribbon Junctions and the Tight Binding Model. *Nanoscale Res. Lett.*, **6**, P. 62 (2011).
[13] S. Ihnatsenka. Computation of electron quantum transport in graphene nanoribbons using GPU. *Computer Physics Communications*, **183** (3), P. 543–546 (2012).
[14] W.H. Press, S.A. Teukolsky et.al., *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge university press, 2007, 1235 p.