

N-body parallel model of tumor proliferation

Rafał Wcisło¹, Paweł Gosztyła¹,
and
Witold Dzwiniel^{1,2}

¹AGH University of Science and Technology, Institute of Computer Science, Kraków, Poland

²WSEiP School of Economics and Law, Department of Computer Science, Kielce, Poland
{wcislo, dzwiniel}@agh.edu.pl

Keywords: complex automata, particle model, tumor proliferation, angiogenesis, parallel implementation

Abstract.

We present a novel parallel 3-D model of tumor proliferation. To simulate the growth dynamics of normal, cancerous and vascular tissues we use a hybrid method integrating cellular automata (CA) with N-body off-grid paradigm, so called, complex automata model (CxA). The interacting particles with dynamically evolving attributes may represent a single cell (cancerous or normal) or a fragment of blood vessel. Therefore, to mimic realistic tumor masses, huge ensembles of particles have to be simulated on multi-core processor systems. There exist many methods widely used for parallelization of classical N-body dynamics. However, they cannot be applied directly in our CxA model, because the evolution of tumor system is controlled by additional processes such as: cell life-cycle, nutrients and TAFs (tumor angiogenesis factors) diffusion and blood flow. These processes influence physical states of particles, e.g., their type, size and ability for proliferation/annihilation. We show that despite these difficulties, particle model can be efficiently implemented on small multi-core processor systems achieving almost linear speedup for as many as 8 threads and speedup of about 30 on 64 threads UltraSPARC T2 processor. We show that this model framework allows for simulating up to 1 million particles in a reasonable time using modest computer resources.

1. INTRODUCTION

Complex automata (CxA) paradigm is a generalization of cellular automata (CA) and is originally defined [1,2] as a scalable hierarchical aggregation of CA and agent-based models. The agents represent sub-systems operating on their typical spatial and temporal scales. This CxA can be an interesting framework for the development of the multiple scale models.

In particular, CxA can be represented by interacting particles whose dynamics are governed by both the Newtonian laws of motion and the CA-like evolution of particle states. Such the hybridization of N-body simulation with CA principles integrates the advantages of both

simulation paradigms: the strong generalization ability (CA) on the one hand, and reconstruction of realistic dynamics (N-body) on the other.

As shown in [3,4], by using N-body methods one can model - depending on the definition of the particle - the evolution of complex phenomena occurring in multiple scales: from atoms to crowd dynamics. By integrating various particle methods such as molecular dynamics, dissipative particle dynamics (or its other clones) and mass-spring systems, one can simulate the realistic evolution of multi-component complex systems from, e.g., complex fluids [5], microscopic blood flow [6] to scenes from computer games [7]. On the other hand, the CA rules can mimic the processes representing other types of local “interactions” than purely mechanical ones, e.g., involving some environmental factors or specific individual properties of particles. In response, the particles on their own can change their attributes, proliferate, grow, decay or annihilate. The feedback between particle attributes and particle dynamics may result in complex emergent behavior of the entire multi-scale CxA system. The flexibility and simplicity of this framework allow us for attacking the most complex problems including the problem of cancer proliferation. Typically, growth of a solid tumor consists of three stages: avascular growth, angiogenesis, and vascular growth (e.g., [8]). In the earliest stage, the tumor develops due to nutrients (e.g. oxygen) diffusion through its surface. As shown in Figure 1a, in angiogenic phase, some of the cells of avascular tumor mass produce and release proteins and other chemical species called tumor angiogenic factors (TAFs). The tumor angiogenic factors diffuse throughout the tissue, and, upon arrival to the blood vessels, they trigger a cascade of events which stimulates the growth of vasculature towards the tumor. In vascular growth stage the tumor has access to unlimited resources of oxygen and other nutrients, which allow to its fast growth. Moreover, through the blood vasculature, the tumor secretes cancerogenic material forming metastases. Thus, whereas in the avascular phase tumors are basically harmless, once they become vascular they are potentially fatal. Computer modeling could allow for answering many principal questions concerning the effects of prescribed chemotherapy or testing new drugs to control the process of tumor growth in all its phases [9,10].

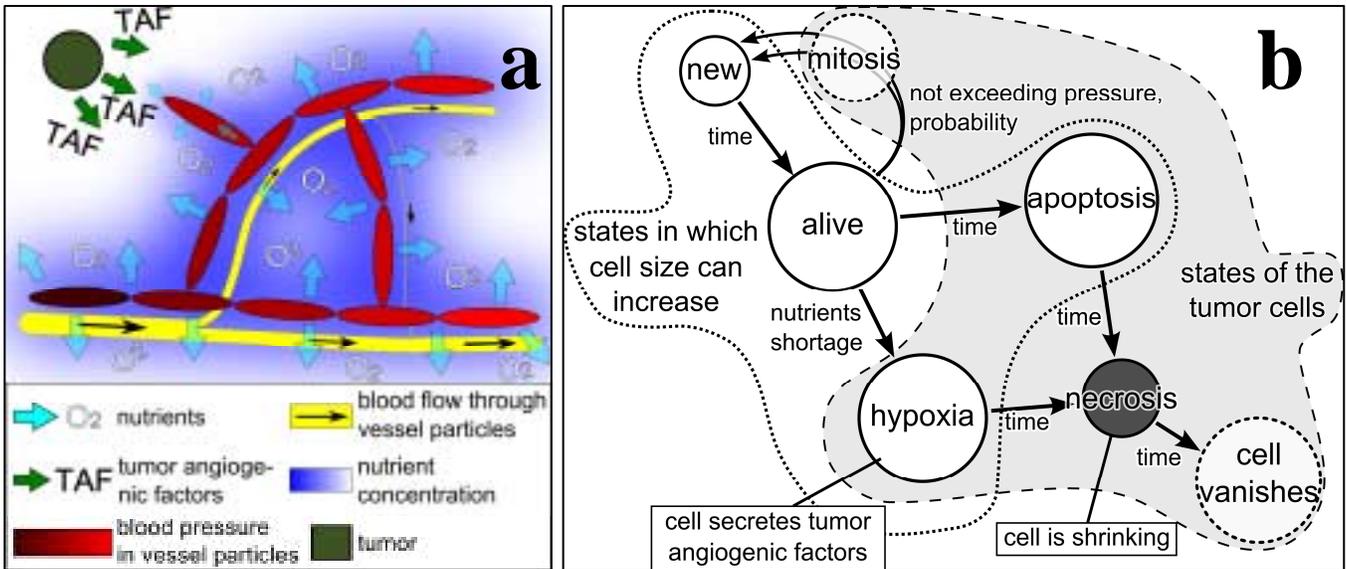


Figure 1 The diagrams of a) the model of tumor induced angiogenesis b) the cell life-cycle.

There exist many mathematical models of tumor progression [11-14]. These models fall into four categories: (a) continuum models that treat the endothelial cell (EC) and chemical species densities as continuous variables that evolve according to a reaction-diffusion system, (b) mechano-chemical models that incorporate some of the mechanical effects of EC-ECM (extracellular matrix) interactions (c) discrete, cellular automata or agent based models in which cells are treated as units, which grow and divide according to prescribed rules (d) hybrid multiscale models involving processes from micro-to-macroscale. Neglecting all microscopic and mesoscopic biological and biophysical processes, tumor growth is purely mechanical phenomenon. It involves dissipative interactions between the main actors: normal, cancerous tissues and vascular network. Due to the effect of tumor directional progression, the surrounded tissue, vasculature and tumor on its own undergo continuous process of remodeling. This kind of tumor dynamics could not be reconstructed in the framework of existing models while just tumor remodeling is responsible for its heterogeneity, which influences the drug dosage/rate in chemotherapy.

In [15] we present the concept of a novel modeling framework, which is based on particle dynamics and the model of complex automata. In this model a particle represents a single cell in ECM envelope. However, this assumption becomes very computationally demanding for modeling tumors of realistic sizes. Tumor of 1 mm in diameter consists of 10^5 - 10^6 cells depending on the size of tumor cell. To model tumor of that size together with its closest environment (normal cells), the dynamics of at least 10^6 - 10^7 particles have to be simulated. Notwithstanding, to

simulate realistic tumor sizes the efficient parallel models exploiting multiprocessor systems are badly needed.

The particle system representing growing tumor is very different than standard N-body systems such as molecular dynamics (MD). Due to cell life-cycle (see Figure 1b) the particles (cells) can proliferate, change their sizes and annihilate. Additionally they have attributes, which evolve according to the rules of cellular automata, and are coupled with particle dynamics. Moreover, the values of attributes depend on concentration fields of oxygen and tumor angiogenic factors. Thus, the process of parallelization is expected to be more complicated than in standard particle codes. In this paper we present the framework of parallel 3-D model of tumor growth, which bases on particle dynamics. The model is optimized for multi-core CPUs rather than for massively parallel systems consisting of many CPUs.

The paper is structured as follows. In the following section we present a brief description of the particle tumor model. In section no.3 we describe its parallel implementation, computer timings and preliminary results. Finally, we discuss the prospects of the model and summarize the conclusions.

2. COMPLEX AUTOMATA MODEL

Here we give only a brief description of our complex automata model of tumor progression based on particle dynamics. More details can be found in [15].

We assume that a fragment of tissue, is made of a set of particles $\Lambda_N = \{O_i: O(\mathbf{r}_i, \mathbf{v}_i, \mathbf{a}_i), i=1, \dots, N\}$ where: i - particle index; N - the number of particles, $\mathbf{r}_i, \mathbf{v}_i, \mathbf{a}_i$ - particle position, velocity and attributes, respectively. Each particle represents

a single cell with a fragment of ECM matrix. The vector of attributes \mathbf{a}_i is defined by the particle type $\{tumor\ cell\ (TC),\ normal\ cell\ (NC),\ endothelial\ cell\ (EC)\}$, cell life-cycle state (see Figure 1b) $\{newly\ formed,\ mature,\ in\ hypoxia,\ after\ hypoxia,\ apoptosis,\ necrosis\}$, cell size, cell age, *hypoxia* time, concentrations of $k=TAF, O_2$ (and others) and total pressure exerted on particle i from its closest neighbors. The particle system is confined in the cubical computational box with a constant external pressure. For the sake of simplicity the vessel is constructed of tube-like “particles” – EC-tubes – made of two particles connected by a rigid spring (see Figure 2a). We define three types of interactions: particle-particle, particle-tube, and tube-tube. The forces between particles mimic both mechanical repulsion from squashed cells and attraction due to cell adhesiveness and depletion interactions cause by both ECM matrix and the cell. We postulate the heuristics - particle interaction potential $\Omega(d_{ij})$ (Figure 2b) - in the following form:

$$\Omega(d_{ij}) = \begin{cases} a_1 d_{ij}^2, & \text{for } d_{ij} < 0 \\ a_2 d_{ij}^2, & \text{for } 0 < d_{ij} < d_{cut} \\ a_2 d_{cut}^2, & \text{for } d_{ij} \geq d_{cut} \end{cases} \quad \text{where } a_1 > a_2$$

$$d_{ij} = |\mathbf{r}_{ij}| - (r_i + r_j) \quad (1)$$

where $|\mathbf{r}_{ij}|$ is the distance between particles while r_i and r_j are their radiuses.

We assume that the interactions between spherical particles and EC-tube particles have similar character. However, as shown in Figure 3a, additional rules have to be introduced to enable appropriate growth of the vascular network. The particle dynamics is governed by the Newtonian laws:

$$m_i \frac{d\mathbf{v}_i}{dt} = -a \cdot \nabla \Omega(d_{ij}) - \lambda \cdot \mathbf{v}_i \quad \frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i \quad (2)$$

$$\mathbf{r}_{ij} = (\mathbf{r}_i - \mathbf{r}_j) \cdot (\mathbf{r}_i - \mathbf{r}_j)^T$$

where m_i is the mass of particle i while λ is a friction coefficient.

As shown in Figure 1b, both normal and tumor cells change their states from *new* to *apoptotic* (or *necrotic*). After *mitosis*, well oxygenated cells of certain age and size split into two daughter cells with d_{MIN} diameters. The cell diameter increases proportionally to oxygen concentration up to d_{MAX} . Finally, after given time period, the particles die due to programmed cell death (*apoptosis*). For oxygen concentration smaller than a given threshold, the living cell changes its state to *hypoxia* being the source of TAFs. The cells die and become *necrotic* if they remain in *hypoxia* state too long. We assume that at the beginning, the diameter of *necrotic* cell decreases twice and, after some time, the cell vanishes. This is contrary to apoptotic cells, which are

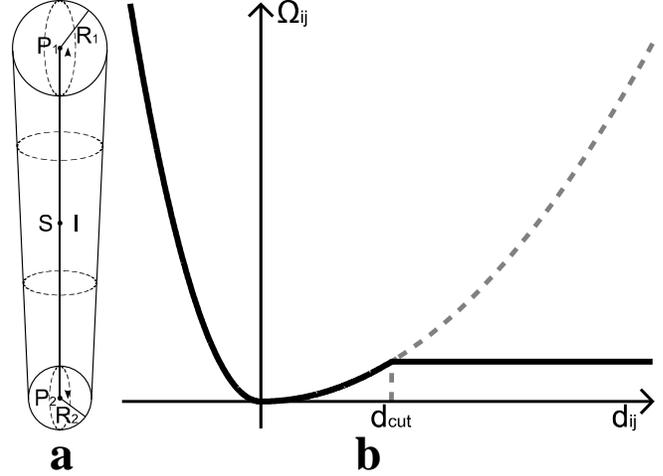


Figure 2 a) Tube-like particle made of two spherical “vessel particles”. b) $\Omega(d_{ij})$ potential form.

rapidly digested by their neighbors or by macrophages. Both normal and tumor cells differ considerably in duration of the life cycle phases and, especially, in the period of time they can live in *hypoxia*. The *hypoxic* cancerous cells can stay alive a few orders of magnitude longer than normal cells.

The life-cycle for EC-tubes is different. They can grow both in length and in diameter. Reduced blood flow, the lack of VEGF (*vascular endothelial growth factor*), dilation, perfusion and solid stress exerted by the tumor can cause their rapid collapse. Because the EC-tube is a cluster of EC cells, its division onto two adjoined tubes does not represent the process of *mitosis* but is a computational metaphor of vessel growth. Unlike normal and tumor cells, the tubes can appear as tips of newly created capillaries sprouting from existing vessels. The new sprout is formed when the TAFs concentration exceeds a given threshold and its growth is directed to its local gradient.

The distribution of hematocrit is the source of oxygen, while the distribution of tumor cells in *hypoxia* is the source of TAFs. We assume that the cells of any type consume oxygen with the rate depending on both cell type and its current state, while TAFs are absorbed by EC-tubes only. TAFs are washed out from the system due to blood flow.

Because diffusion of oxygen and TAFs through the tissue is many orders of magnitude faster than the process of tumor growth, we assume that both the concentrations and hydrodynamic quantities are in steady state in the time-scale defined by the time-step of numerical integration of equations of motion. On the other hand, the blood circulation is slower than diffusion but still faster than *mitosis* cycle. These facts allow for employing fast approximation procedures for both calculation of blood flow rates in capillaries and solving reaction-diffusion equation (see [15]). The main procedures invoked in a single time

step are shown in Figure 3b. After initialization phase, in subsequent time-steps we calculate forces acting on particles, new particle positions, the diffusion of active substances (nutrients, TAFs, pericytes), the intensity of blood flow in the vessels and the states of individual cells triggered by previous three factors and constrained by time clocks of individual cells. All of these modifications of cell states may result in cell *mitosis* or its death. They can also change some cell functions (e.g. those under *hypoxia*) their size and environmental properties (e.g., cancerous cells can secrete acid to eliminate neighboring normal cells).

3. PARALLEL IMPLEMENTATION

3.1. Algorithms And Data Structures

Classical N -body codes, such as molecular dynamics (MD), simulate evolution of a particle system confined in a periodic cube by integrating numerically Newtonian equations [16]. The single time-step consists of two principal procedures: computation of forces acting on each particle and shifting them according to the total momentum calculated from Eqs.2. From the point of view of efficient computations, the linear computational complexity $O(N)$ of the algorithm is the most important requirement. Fortunately, both the particle interactions and approximation kernel are short-ranged. However, both the calculation of forces and approximate procedure used for solving diffusion equation, require finding all the pairs of particles in the nearest neighborhood. For short-range interactions, e.g., given by Eq.1, the forces can be computed using fast $O(N)$ method exploiting alternately Hockney or Verlet algorithms [16]. As shown in Figure 4a, the

computational box is divided into cubic sub-boxes with edges equal to the interaction range. As demonstrated in Figure 4a, the particle located in a given sub-box interacts with other particles located in this sub-box and in adjacent sub-boxes. Of course, many other approaches are possible both for serial and parallel implementations [17].

For modeling tumors of realistic sizes the dynamics of 10^5 - 10^7 particles (normal, cancerous and EC-tube cells) have to be simulated by exploiting the power of nowadays multi-core CPUs, multi-processor systems and by using optimized N -body parallel codes. Unfortunately, the particle system representing our tumor model is very different than standard MD ensembles. Consequently, the process of code parallelization is expected to be more complicated [17-19]. The particles (cells) can proliferate, resize or annihilate. Moreover, they have additional attributes, which evolve according to the rules of cellular automata and influence the cells' dynamics. The attributes, in turn, depend on O_2 and TAF concentration fields. This requires both solving reaction-diffusion equation and calculating the intensity of blood flow in capillary vessels.

As shown in Figure 6, the computation of EC-tube particles interactions is the most critical component influencing computational efficiency. The length of EC-tube is considerably greater than its width, what involves considerably larger sizes of Hockney cells (sub-boxes) than those used for spherical particles. Moreover, the tubes can grow covering even 5 sub-boxes used for forces calculation between spherical cells. To overcome this problem we propose using instead of one, two separate data structures \mathbf{P} and \mathbf{V} , for storing spherical cells and EC-tubes, respectively.

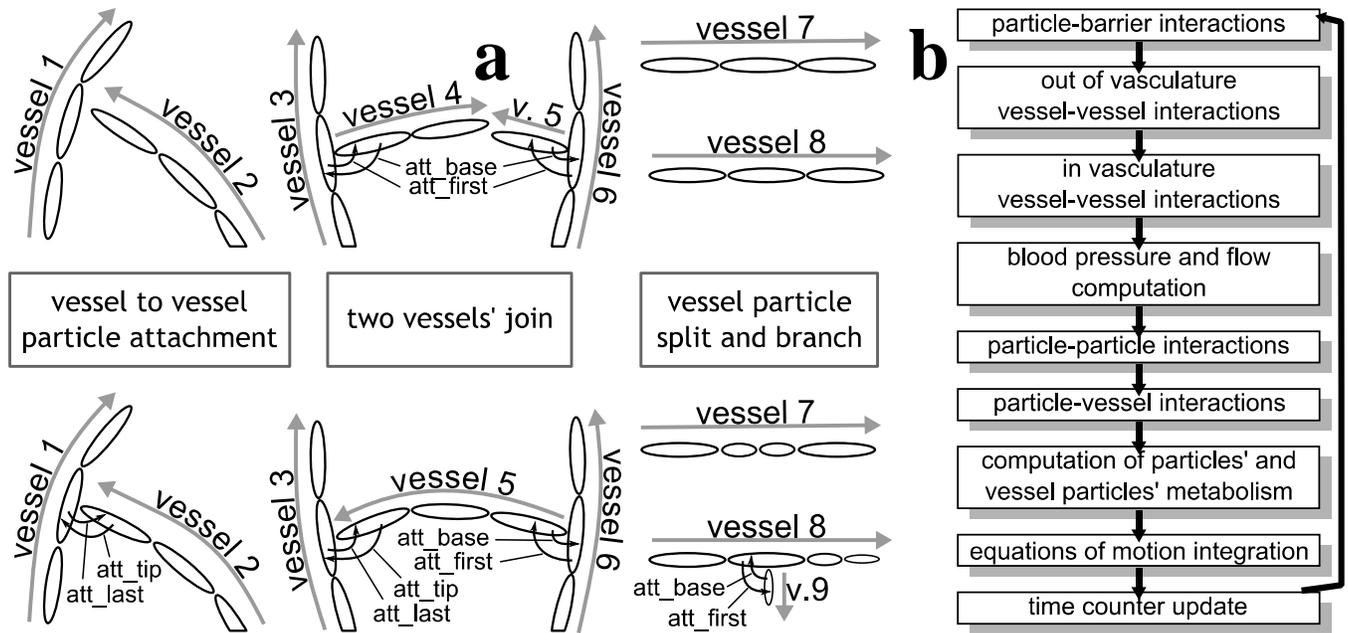


Figure 3 a) Vessel-vessel interactions and vessel growth rules. b) The main procedures invoked in a single time-step.

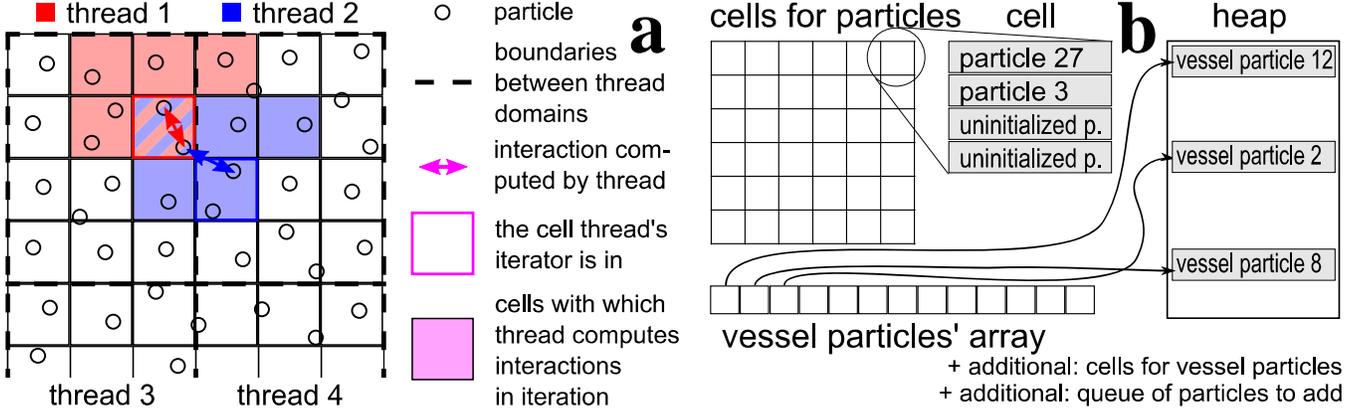


Figure 4 a) Domain decomposition used for forces calculation. b) Data structures storing spherical and vessel particles.

The \mathbf{P} data structure is represented by 3-D array of Hockney cells with TC and NC particles, while \mathbf{V} is a data structure consisting of the array of pointers to records representing EC-tubes and the additional 3-D array of cells used to compute particle-tube and tube-tube interactions. The cells in this array correspond to respective cells in \mathbf{P} . Because vessel particle is long enough to cross several cells, it cannot be assigned to a single cell, as it is in Hockney algorithm. Instead, EC-tube is placed in a minimal cuboid composed of all the cells it crosses. This cuboid is enlarged then by one cell margin in each direction, covering the vessel particle together with its cut-off radius. We assume that the vessel particle belongs to all the cells forming this final cuboid (see Figure 5).

Calculation of forces between particles is shared between three separate algorithms calculating: particle-particle, particle-vessel, vessel-vessel interactions. For particle-particle forces we use standard Hockney algorithm. In case of particle-vessel computation, for each corresponding pair of cells cp , cv from \mathbf{P} and \mathbf{V} , respectively, particles from cp are tested against vessel particles from cv . If the distance between the pair is shorter than the cut-off radius, their interaction is calculated.

The algorithm used for vessel-vessel computation exploits the fact that if two EC-tubes lie in a distance shorter than cut-off radius, there exists at least one cell in \mathbf{V} containing both particles. This way, all interacting pairs can be found by iterating throughout all cells and making all-to-all test for distance. However, a pair of particles representing two interacting EC-tubes can be found in many cells whereas it should be taken only once. To address this problem, we introduce ternary relation R (see Figure 5), which eliminates redundant interactions:

$$Ec \times Ec \times C \supseteq R = \begin{cases} (e_1, e_2, c) : e_1, e_2 \in Ec; c \in C; \\ \max(e_1.mx, e_2.mx) = c.x, \\ \max(e_1.my, e_2.my) = c.y, \\ \max(e_1.mz, e_2.mz) = c.z \end{cases} \quad (3)$$

where:

Ec is the set of EC-tubes, C is the set of cells in mesh, $c.x$, $c.y$, $c.z$ are coordinates of cell c in 3-D array and
 $e.mx = \min(\text{cell}(e.p1).x, \text{cell}(e.p2).x)$,
 $e.my = \min(\text{cell}(e.p1).y, \text{cell}(e.p2).y)$,
 $e.mz = \min(\text{cell}(e.p1).z, \text{cell}(e.p2).z)$,

where: $e.p1$ and $e.p2$ are two ends of EC-tube e , $\text{cell}(p)$ is the cell to which point p belongs to.

To reduce the number of cache misses, the cells in \mathbf{P} do not contain pointers to particle records but whole records instead. Each cell is represented then by an array of fixed number of objects (see Figure 4b). All the TC and NC particles are allocated directly inside corresponding cells. This ensures that particles are always properly ordered in memory according to their positions for the price of greater memory consumption. This is because the cells have various numbers of particles and many records are empty. As particles move, they change the cells they belong to. Therefore, the arrays \mathbf{P} and \mathbf{V} are updated after each time-step. In case of \mathbf{V} , the cells are built from the beginning by using the array of EC-tube pointers located in \mathbf{V} . Whereas for \mathbf{P} , because particles are allocated inside the cells, changing location from one cell to the other means that the whole particle record must be moved to a different memory location.

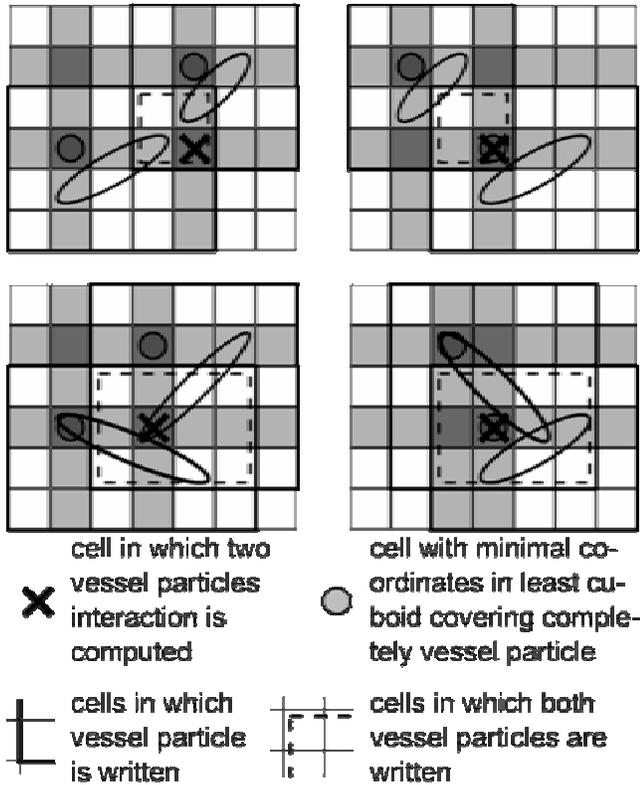


Figure 5 Graphical interpretation of relation (3).

This takes longer time in comparison to pointers operation in the standard approach. However, because of steady nature of particles dynamics in our model, such the situation does not occur too often. In fact, in our simulations the process of reordering particles requires less time than standard linked-list procedure. The reason is, that in the former, the particles which do not change their cells need only “read” operation of their coordinates from the memory, while in the latter, for all the particles there is an additional “write” operation.

During simulation, the number of spherical and EC-tube particles can both increase due to *mitosis* and decrease as the result of *apoptosis* and *necrosis*. Information of newly formed and dead particles must be added and removed from the data structures. As doing this directly could cause problems with synchronization, three intermediate data structures are employed: for newborn particles in **P**, for new vessel particles in **V** and for indexes of dying vessel particles in **V**. Removing objects from **P** is done directly as it is cell-local operation, which does not impair other threads operation and never cause data structures to be rebuilt. Moving object from intermediate structures to **P** and **V** and removing object from **V** is done sequentially between separate time-steps.

3.2. Speedups And Exemplary Results

The tests were carried out on the SGI Altix XE 1300 cluster consisting of 256 SGI Altix XE 300 nodes and SPARC Enterprise T5120. Each Altix node consists of two four-core processors Intel Xeon 2.66 with 16 GB of RAM allowing for maximum 8 threads executed in parallel. The SPARC computer consists of eight-core 1.2 GHz UltraSPARC T2 CPU capable of running in parallel eight threads per single core. It gives in total 64 threads per node executed concurrently on 32 GB of RAM. Our parallel algorithm is constructed for a single shared memory node and is implemented in C++ with OpenMP interface.

We used domain decomposition approach both along one side of the computational box (each box slice was handled by one thread) and dividing the box onto sub-boxes of equal sizes (for 8 threads we have 2x2x2 grid of sub-boxes, while for 64 4x4x4 grid of sub-boxes).

To exploit the full power of multiprocessor system, the second level parallelism should be introduced based on message-passing MPI interface. It would make the code extremely complicated and rigid for improvements. This would also extend the time for implementation and tests. Moreover, running the code on the large number of CPUs is usually restricted by system administrators and consumes much time and money. So, having in mind the shift in modern chip technology towards production of multiple-

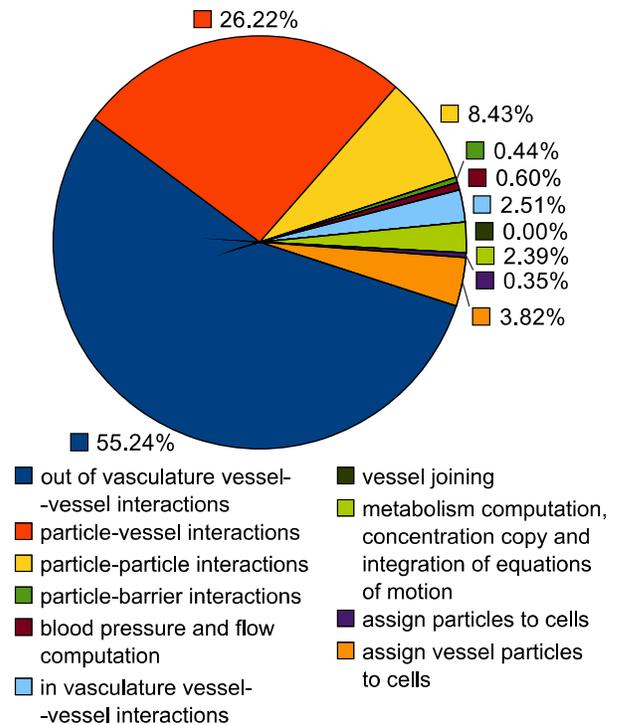


Figure 6 The diagram showing the shares of computational time used by various procedures (the evolution of 10^6 particles was simulated).

core CPUs (empowered by GPU) we decided to meet this trend implementing the code open for both future improvements in the model and technological progress.

As shown in Figure 7, the preliminary timings obtained for our parallel code are very encouraging. We got speed-up of about 7 on 8 threads CPU and about 30 on 64 threads CPU simulating 10^6 particles. The timings could be better for more realistic vessel densities much lower than those considered in the test runs.

The snapshots from simulations of tumor and its vasculature progression obtained for timing tests are shown in Figure 8. The tests were made for various particle ensembles. The initial scene consists of two straight parallel vessels, the cells representing normal tissue and a few cancerous cells located between the vessels. Because of increasing TAFs concentration, secreted by the tumor cells in *hypoxia*, we can observe newborn capillaries sprouting out from the source vessels. The vasculature expands and is continually remodeled due to tumor growth dynamics. The sprouts can bifurcate and merge creating anastomoses. The blood flow is stimulated by pressure difference in anastomosing vessels. Only productive vessels have a chance to survive if the TAFs concentration is sufficiently high. Unproductive vessels disappear after some time. Well oxygenated cells are colored blue (dark gray) while the cells in hypoxia are marked by shades of green (light gray).

Necrotic cells are black.

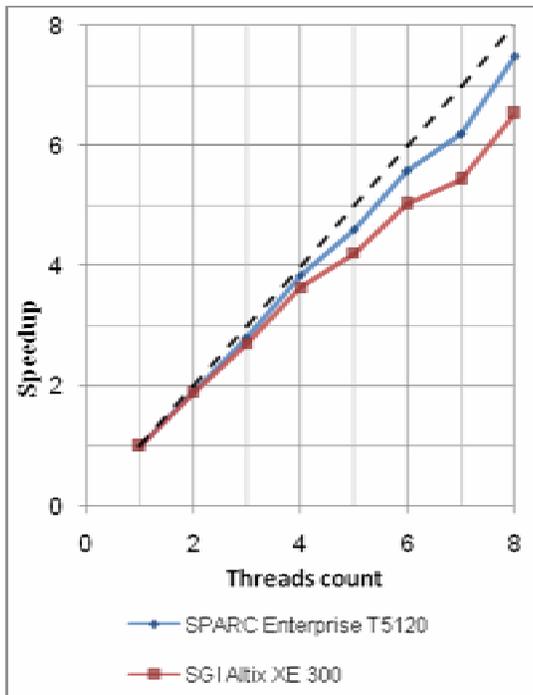
4. CONCLUSIONS

The complex automata paradigm employing both particle dynamics and cellular automata rules can be used as a robust modeling framework, e.g., for developing realistic models of tumor growth as a result of emergent behavior of many interacting cells. Particularly, this framework targets principal problems involved with mechanical interactions between growing tumor, normal tissue, and expanding vascular network. The particle model of tumor growth can be easily extended by implementing more precise sub-models of all the processes – known and unknown – responsible for tumor creation and proliferation.

However, to simulate realistic tumor sizes, the number of particles should be greater than 1 million what require considerable computational resources to obtain the results in a reasonable time. Moreover, well known parallel algorithms, employed for simulating N-body system, cannot be implemented in a straightforward way in our model.

We have shown here that by using data parallel paradigm and shared memory computer systems, relatively large particle ensembles can be modeled by using nowadays multi-core processors and, possibly, future multiple-processor CPUs. We expect that just technological progress

SGI Altix XE 300 & SPARC Enterprise T5120 comparison



SPARC Enterprise T5120

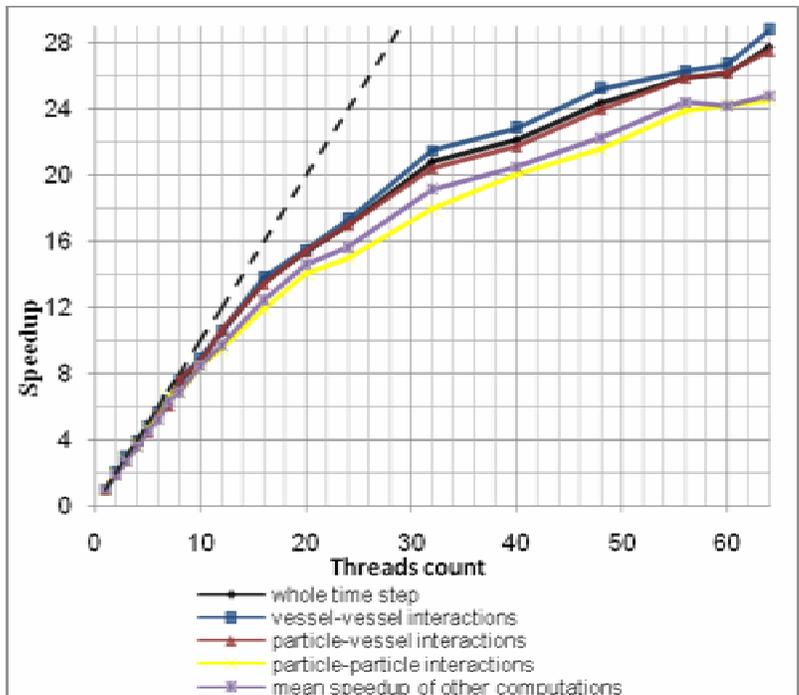


Figure 7 The speed-ups obtained for the main procedures of the tumor model during simulation of 10^6 particle ensembles on two test machines.

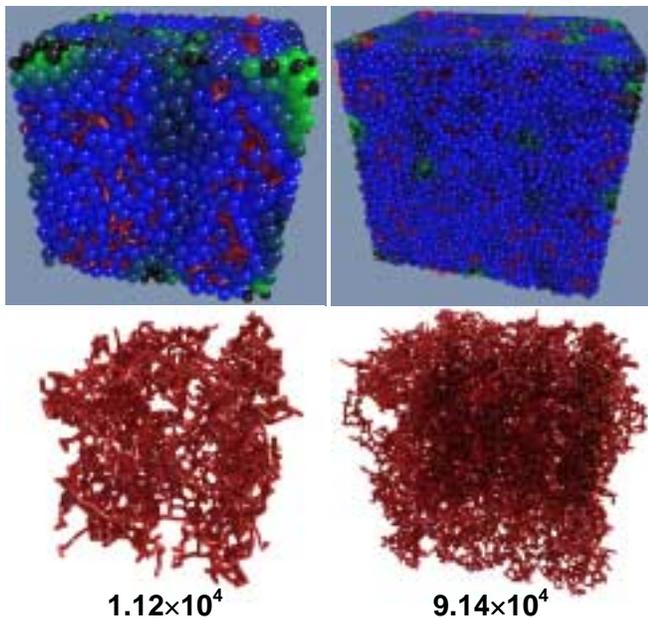


Figure 8 The snapshots from evolution of particle systems (upper part of the picture) for two ensembles of particles. The vascular networks are shown in the bottom.

simultaneously with the model improvements will allow for increasing of both the sizes of tumors simulated and the precision of the results obtained.

We are working now on creating an efficient parallel platform integrating, apart from multi-core CPUs, the modern GPU processors, to provide flexible and fast simulation tool for oncologists, which could be used on small but strong stand-alone workstations.

Acknowledgements This research is financed by the Polish Ministry of Higher Education and Science, project N N519 579338 and partially by AGH grant No. 11.11.120.777. The movies from simulations are collected on the www.icsr.agh.edu.pl/~wcislo/Angiogeneza/index.html.

References

1. Hoekstra, A.G., Lorenz, E., Falcone, L.C., Chopard, B., 2007, "Towards a complex automata framework for multi-scale modeling: Formalism and the scale separation map". *Lect Notes Comput Sci*, 4487, 922-930.
2. Sloot, P.M.A., Kroc, J., 2009, "Complex Systems Modeling by Cellular Automata", *Encyclopedia of Artificial Intelligence*, Ed. Rabunal JR, Rabunal Dopico JR, Dorado J, Sierra AP, *Informatio SCI*, Harshey-New York, 353-360.
3. Dzwinel, W., Alda, W., Yuen, D.A., 1999, "Cross-Scale Numerical Simulations Using Discrete-Particle Models", *Molecular Simulation*, 22, 397-418.
4. Dzwinel, W., Alda, W., Kitowski, J., Yuen, D.A., 2000, "Using discrete particles as a natural solver in simulating multiple-scale phenomena", *Molecular Simulation*, 20, no.6, 361-384.
5. Dzwinel, W., Yuen, D.A., Boryczko, K., 2006, "Bridging diverse physical scales with the discrete-particle paradigm in modeling colloidal dynamics with mesoscopic features", *Chemical Engineering Sci.*, 61, 2169-2185.
6. Boryczko, K., Dzwinel, W., Yuen, D.A., 2003, "Dynamical clustering of red blood cells in capillary vessels", *J Mol. Modeling*, 9, no.16-33.
7. Eberly, D.H., 2004. *Game Physics*, Morgan Kaufman – Elsevier. NY.
8. Folkman, J., 1971, "Tumor angiogenesis, Therapeutic implications", *N Engl J Med*, 285, 1182-1186.
9. Castorina, P., Carcò, D., Guiot, C., Deisboeck, T.S., 2009, "Tumor Growth Instability and Its Implications for Chemotherapy", *Cancer Res*, 69, no.21.
10. Bellomo, N., de Angelis, E., Preziosi, L., 2003, "Multiscale Modeling and Mathematical Problems Related to Tumor Evolution and Medical Therapy", *J Theor Med.* 5, no.2, 111–136.
11. Chaplain, M.A.J., 2000, "Mathematical modelling of angiogenesis", *J Neuro-Oncol*, 50, 37–51.
12. Preziosi, L. (ed), 2003, *Cancer modelling and simulation*. Chapman & Hall/CRC *Mathematical Biology & Medicine*.
13. Mantzaris, N., Webb, S., Othmer, H.G., 2004, "Mathematical Modeling of Tumor-induced Angiogenesis", *J Math Biol.*, 49, no.2, 1432-1416.
14. Lowengrub, J.S., Frieboes, H.B., Jin, F., Chuang, Y-L., Li, X., Macklin, P., Wise, S.M., Cristini, V., 2010: "Nonlinear modelling of cancer: bridging the gap between cells and tumours", *Nonlinearity*, 23, R1-R91.
15. Weislo, R., Dzwinel, W., Yuen, D., Dudek, A.Z., 2009, "A new model of tumor progression based on the concept of complex automata driven by particle dynamics", *J. Mol. Mod*, 15, no.12, 1517–1539.
16. Haile, P.M., 1992, *Molecular Dynamics Simulation*. Wiley&Sons, NY.
17. Kadau, K., Germann, T.C., Lomdahl, P.S., 2004, "Large-scale molecular-dynamics simulation of 19 billion particles", *Int. J. Mod. Phys. C*, 15, no.1, 193-201.
18. Boryczko, K., Dzwinel, W., Yuen, D.A., 2002, "Parallel Implementation of the Fluid Particle Model for Simulating Complex Fluids in the Mesoscale", *Concurrency and Computation: Practice and Experience*, 14, 1-25.
19. Boryczko, K., Dzwinel, W., Yuen, D.A., 2005, "Modeling Heterogeneous Mesoscopic Fluids in Irregular Geometries using Shared Memory Systems", *Mol Simul*, 31, no.1, 45-56.