Ant Colony Optimization Algorithm For Protein Folding Problem On Graphics Processing Units

Bekmuratov Tulkun, Bazarov Rustam

Abstract: This article describes the methods of modification and parallelization of ant colony optimization algorithm for the protein folding problem. It describes in detail the software implementation of parallel ant colony optimization algorithm on the graphics processing units.

Index Terms: ant colony optimization algorithm, graphics processing units, protein folding problem.

1. INTRODUCTION

The protein folding problem (PFP), a problem of searching for the tertiary structure of a protein from the primary amino acid sequence, is a fundamental problem in bioinformatics and structural biology. Protein structure prediction is highly important in medicine (for example, in drug-design) and biotechnology. Unfortunately, even with simplified lattice models, in which only hydrophobic interactions are taken into account [1], the PFP is non-deterministic polynomial-time hard (NP -hard) [2]. Such problems are successfully solved only by heuristic methods of global optimization, for example, ant colony optimization algorithm [3]. This paper is devoted to parallel implementation of ant colony algorithm on the graphics processing units.

2 BIOLOGICAL BACKGROUND

A protein is a long linear polypeptide sequence of amino-acid residues. Fig. 1 is showing a polypeptide chain fragment of two amino acids with their side groups R_l and R_{l+1} , where "l" the number of residues in the sequence. The main-chain angles of rotation: (φ,ψ,ω) and that of the side chain (χ) are also presented. Each amino acid consists of a central carbon atom (alfa) and four connecting bonds: a hydrogen atom, a carbonyl group, an NH -group and a side R -chain.

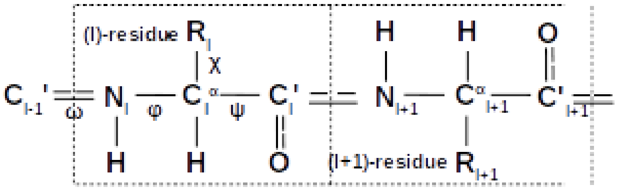


***Figure 1**. Amino acid chain of two residues.*

Amino acids are linked by partial-double planar rigid peptide bonds between C' - and N -atoms. The angle of rotation around this bond is denoted as ω . A flexibility, which implies the ability to fold globules, of the protein chain is provided by rotation around the covalent single bonds. The set (φ,ψ) of rotation angles is called the conformation. The "allowed" and "disallowed" conformations of a residue plotted in the (φ,ψ) coordinates are called Ramachandran plots. Since each residue is supposed to have only three conformational states [4], there are $3^{100}=5\times10^{47}$ conformations for protein with 101 residues. Consideration all of them at a rate of 1 nanosecond (10^{-9} sec.) per configuration would take 3×10^{13} years, that is more than one thousand lifetimes of the Universe. But the real folding time of globular proteins occurs much faster than it can be conceived. Then, how does the protein choose its native structure among zillions of others, asked and answered Levinthal [5]: It seems that there exists a specific folding pathway, and the native fold is simply the end of this pathway rather than the most stable chain fold [6]. So there are two strategies of protein structure prediction: (1) to seek for the structure resulting from the kinetic folding process; and (2) to seek for the most stable (or, equally, the most probable) chain structure. Let us consider the second one, because the first is not yet succeeded [6]. The in-vivo formation of the native (i.e., biologically active) tertiary structure occurs during biosynthesis or immediately after. In a tube (in-vitro) only small (up to 200-300 amino acid residues) water-soluble globular proteins are capable of spontaneous self-organizing. A simplified diagram of that process is shown in (Fig. 2) [6]

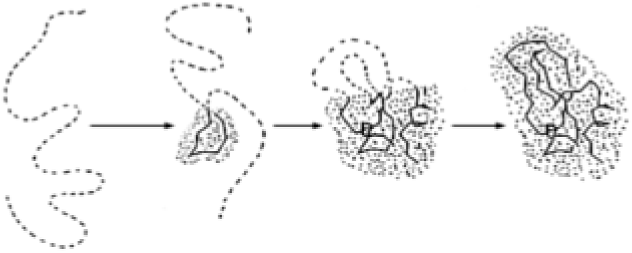


***Figure 2.** The process of protein folding (in-vitro)*

The protein folding in-vitro starts at a position, hereinafter referred to as the folding initiation point. The region highlighted in Fig. 2, corresponds to the part of the globule that has already obtained the final conformation. For simplicity, the side groups of the chain are not shown in the figure. The bonds and interactions between side R -groups of a protein has a significant impact on the folding process. There are five types of such bonds and interactions [7]: disulfide bonds, which are formed by cysteine amino acid residues (Fig. 3a), electrostatic interactions of the charged R-groups of the protein (Fig. 3b), hydrophobic interactions, stipulated by hydrophobicity of some residues (they are not shown in the figure, since they are an instance of cooperative interaction, not pair ones); hydrogen bonds between the residues with a hydroxyl group; van der Waals interactions: all atoms and molecules attract each other at a distance above 2-3 Å [6,7]. In (Fig. 3) two fragments of the chain are shown, which are adjacent in the globule and

- Bekmuratov Tulkun Fayzievich, academic, department of SIC ICT of TUIT named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan. E-mail: bek.tulkun@yandex.com
- Bazarov Rustam Kamilevich, doctoral candidate, department of SIC ICT of TUIT named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan. E-mail: rustam.bazarov@gmail.com

containing four amino acid residues each. Between their side groups the above types of interactions had happened. Distances between the side groups, which are proper for all mentioned types of interactions, are also presented.

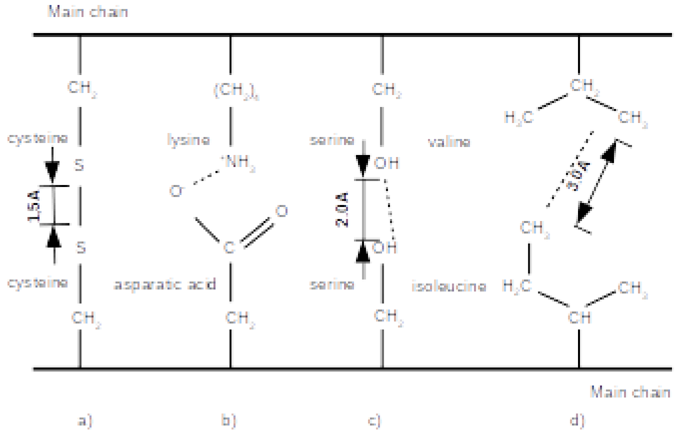


Figure 3. *Interaction types examples between side groups of amino acid residues in a protein chain.*

Protein structures are complex systems, such as they consist of many elements with a dynamically changing nature of their interaction. A system is defined as complex one if it has such properties as [8]:

1) variety of elements and relations between them: 20 residues with pair and cooperative interactions of electrical, quantum and chemical nature between them;

2) multidimensionality: there are $10^{390} = 20^{300}$ different construction alternatives for a typical water-soluble protein of 300 residues. For every 10^{-9} of them only one has a stable structure [9]. Chains with a stable structure are obtained as a result of natural selection.

3) multi-criteria: there are at least two criteria for PFP: a stability of the protein molecule and it's folding time.

4) multi-variance: it is unknown, how many ways there are to get the optimum for these criteria.

5) multiple changes in the composition and/or structure of the system: the polar side chains of residues can form energetically favorable hydrogen bonds with water molecules, that is involving the new elements in the system.

6) multiplicity: there are several models and methods for determining the spatial structure of a protein: semi-empirical methods of quantum chemistry and empirical force fields ones. The quantum chemistry semi-empirical methods are used for the geometry and charge states of residues optimization. Pair interactions between the molecules, considered as classical elastic particles, are estimated by the molecular dynamics methods of empirical force fields. But ab-initio quantum chemistry methods cannot operate with large groups of molecules, and the molecular dynamics methods have low accuracy. The latter ones do not take into account, in particular, the electronic effects (atomic polarizability, electron transfer, the formation and breaking of chemical bonds), and the cooperative hydrophobic interactions cannot be modeled at all. The mathematical complexity of the protein folding simulation consist in necessity to solve both the geometric problem of finding close-packing of equal spheres, the continuous mathematics theme, and the combinatorial one of folds, the discrete mathematics theme. In addition, the discrete mathematics component of the problem use the difficult concept of self-avoiding walk, which exceed even the NP-completeness (#P-completeness). Such optimization problems have no known foundational mathematical theory [10]. Therefore, various simplifying models of protein structure are considered. Amino acid residues of globular proteins can be represented as circles or squares with 5.3 Å on a side (from 4.0 to 6.2Å depending on the R-group's size) [11]. The hydrophobic interactions are dominated in water [12] for small, up to 200 residues, globular proteins, and only they are taken into account according to the lattice HP model [1] proposed by Ken Dill in 1985. Thus, a protein, as a complex system, with such simplifications contains two types of elements: the hydrophobic residues and the hydrophilic ones (Fig. 4). The problem is to find a conformation of the protein with maximum of contacts between hydrophobic residues, adjacent in the folded structure, but not in the chain. The number of these contacts with a minus sign is called an energy of structure. Each residue's degree of freedom in the chain (the number of folding directions) is 3 on the plane and 5 in the space. These are so-called lattice bead models [13,14]. During the process of folding, a core is forming by hydrophobic residues (in Fig. 4 they are indicated in black). The side groups of hydrophilic residues are dangling in water.

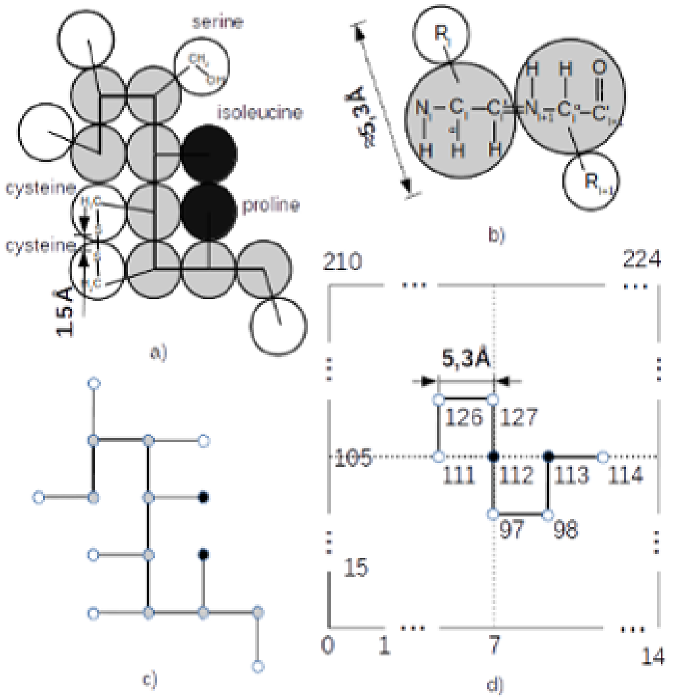


Figure 4. *Non-lattice (a), lattice (b,c) HP-models of the protein structure and conformation space (d) for sequence with 8 residues.*

2 MATHEMATICAL STATEMENT OF THE PROTEIN FOLDING PROBLEM

Bearing all the above assumptions, a mathematical formulation of the PFP is proposed with denotions from [15]: Consider a protein sequence $Q = (q_i \mid i \in [0, |Q|-1], q_i \in \{0,1\})$ of hydrophobic $a=1$ and hydrophilic $a=0$ residues q_i, a space $C^{|X|}=C^{|Q|}=C\times C\times\ldots\times C=[0,4\times|Q|\times(|Q|-1)]$ of all possible conformations $X=(x_i, i\in[0,|Q|-1])$ One of the latters is a vector of components x_i each of them determinates the position of corresponding residue. Here and after the $|Q|$ is the number of components of vector Q.

Let a delta-function is denoted by $\delta(x_1,x_2)=(x_1-x_2\in\Delta x)?1:0$, which is «1» for neighboring amino acids x_1,x_2 and «0» for not. Here $\Delta x=(-1,1,2\times|Q|,-2\times|Q|)$ is an incremental vector for residue location.

For a given sequence Q it is required to find a conformation X^* that maximizes the number of topological contacts:

$$f(X)=\sum_{i=3}^{l}\sum_{j=0}^{i-3}\delta(x_i,x_j)\times q(x_i)\times q(x_j)$$

between hydrophobic residues, are not neighbored in sequence Q. That is we need to solve the maximizing problem of the objective function:

$$f(X):\max_{X\in\Omega\subset C^{l+1}} f(X)=f^*$$

where Ω is a set of all valid conformations for Q.

The sequence Q contains all vectors X from conformational space $C^{|Q|}$ with components are received by the transition:

$$x_i=x_0+\sum_{j=0}^{i-1}\delta x_j, i\in[0,|Q|-1], \delta x_j\in\Delta x$$

from some initial position $x_0=2\times|Q|\times(|Q|-1)$ of the residue q_0.

It should be noted that even with such simplifications, the search for optimal conformation is an NP-complete problem [2]. For a protein chain of length $|Q|=71$, the number of possible conformations on two-dimensional lattice will be 4.2×10^{30} [16].

If it is impossible implementing of complete enumeration, an iterative algorithm may be applied to obtain a suboptimal solution “at random” as a result of series of tests, performed by a given number of subjects (agents) $a_n\in A, n\in[1,|A|], |A|\geq1$ of a certain set (population) A.

The solution of the folding problem in the above formulation is carried out using an iterative evolutionary algorithm [15], the general scheme of which has the form:

1) Setting initial iteration counter values $t=0$ and agent positions $x_n(0), a_n\in A, n\in[1,|A|], |A|\geq1$

2) Applying the migration operator to the current positions $x_n(t)=x_n^t=x_n, n\in[1,|A|]$ of the set of agents that calculates new positions according to the some function $\Phi(\cdot): x_n^{t+1}=\Phi(x_\eta^\tau, f(X_\eta^\tau)), n,\eta\in[1,|A|], \tau\in[0,t], t+1\leq\hat{t}$ in which the new position of the agent is determined by the positions of all agents of the population in the preceding moments as well as the corresponding values of the objective function:

$$f(X_\eta^\tau)=\sum_{t=0}^{\tau}q(x_\eta^t)\times\sum_{\sigma=0}^{t}q(x_\eta^\sigma)$$

where $x_\eta^t=x_\eta^\sigma+\delta x_d$ and

$$x_\eta^\sigma=x_\eta^0+\sum_{t=1}^{\sigma}\delta x_d^t, \delta x_d\in\delta x, d\in D$$

Here X_η^t - is the partial conformation, which had got by agent a_η at the moment $t=\tau$.

The complete conformation $X_\eta=X_\eta^{\hat{t}}$ - is a value of partial conformation at the last time stamp $\hat{t}$.

3) Check the conditions for the algorithm’s completion. For example, $|X^{t+1}-X^t|\leq\varepsilon_X$ or $|f(X^{t+1})-f(X^t)|\leq\varepsilon_f$, where $\varepsilon_X,\varepsilon_f$ are the constants that determine the required accuracy of the solution by the vector of variable parameters and the optimality criterion, respectively.

If the conditions are not met, then go to the next iteration $t=t+1$ and to step 2 of the algorithm. Otherwise, the best of the found positions $\tilde{X}^*$ that was determined from the condition:

$$\tilde{f}^*=f(\tilde{X}^*)=\max_{t\in[0,\hat{t}], n\in[1,|A_m|]} f(X_n^t)$$

is adopted as a suboptimal solution of the problem.

3 ANT OPTIMIZATION ALGORITHM FOR PROTEIN FOLDING

Iterative evolutionary algorithms are mainly differing in the method of calculating the migration operator. Let us consider the ant colony optimization (ACO) algorithm, since it gives the best results in solving the PFP [3]. It should be noted, that the migration operator determines the position of each agent of a population depending on the position of all agents (including this agent) at all previous iterations of the algorithm. In ACO, the impact of some agent's positions at the former moments to the choice of the current position of the ant is called heuristic information. And an impact of other agents of the population is expressed by the so-called pheromone function, the traces, left by ants.

Each of the ant's position triplet $(x_n{}^{\tau-1}, x_n{}^{\tau}, x_n{}^{\tau+1})$ corresponds to a couple coordinate directions. The difference $\Delta d_n^\tau=d_n^{\tau+1}-d_n^\tau$ of them, in turn, corresponds to a pheromone point $\phi_n(t,\tau)$. It's intensity is equal $\Theta_n(t,\tau)$ for given iteration t, where $\tau\in[0,t]$ is the iteration's number, on which this point was delivered. The number of pheromone in pheromone points with an increase in $\Theta_i(t,\tau)=b_\Theta\times\Theta_i(t-1,\tau)$ the number $\Theta_i(t-1,\tau)>\Theta_{\min}$ of iterations decreases (the pheromone evaporates) according with:

, where

Let a position x_n^t of some agent $a_n, n\in[1,|A|]$ at the current moment is denoted x_n and at the next moment is denoted by x'_n. Then complete probability of choosing a folding direction $d\in D$ is defined by components:

The first one $P_{\psi(d)}, P_{\phi(d)}$ is a probability determined by the set of its own positions of the ant (heuristic

$$P_\psi(d)=P_\psi(x'_n=x_n+\delta x_d)=\frac{q(x_n+\delta x_d)\times\sum_{d'\in D}q(x'_n+\delta x_{d'})}{\sum_{d\in D}\sum_{d'\in D}q(x_n+\delta x_d)\times q(x'_n+\delta x_{d'})}$$

information):

and the second is a probability determined by the set of the other ant's positions of the colony (pheromone traces):

$$P_\phi(d)=P_\phi(x'_n=x_n+\delta x_d)=\frac{\phi_{\Delta d}}{\sum_{d\in D}\phi_{\Delta d}}$$

Thus, the complete probability of choosing the direction d by the ant a_n at the moment t will be defined as:

$$P_{\phi\psi}(d)=P_{\phi\psi}(x'_n=x_n+\delta x_d)=\frac{\phi_{\Delta d}\times q(x_n+\delta x_d)\times\sum_{d'\in D}q(x'_n+\delta x_{d'})}{\sum_{d\in D}\sum_{d'\in D}q(x'_n+\delta x_d)\times q(x'_n+\delta x_{d'})\times\phi_{\Delta d}}$$

The last formula is the expression of the migration operator for the ant iterative optimization algorithm.

Heuristic information increases probability $P_\psi(d)$ of choosing a direction with the maximum number of hydrophobic contacts. It should be noted, that this number is not known in advance and determined while the conformation is constructing. ACO modifications for the PFP were considered in [17,18], and convergence issues in [19,20]. A solution for the same problem on a triangular lattice, where amino acids are better packed than in square ones, was proposed in [21]. In that article a novel method for updating the pheromones is introduced. One can find the optimal solution on average in 1.5 times better using it, due to taking into account interactions between distant residues. However, this lattice is less investigated, which limits the possibility of an experimental comparison of the algorithm presented in the article with the known ones. It is possible to reduce the time for obtaining a sub-optimal solution using a parallel and distributed software implementation of the algorithm [22]. An ant algorithm for solving the classification problem, implemented on graphics processors, was described in [23]. There are several methods of ACO parallelization. The first one is a whole colony decomposition into sub-colonies for further handling each of them by corresponding computing element of some cluster [24]. The second is organized by sending the ants themselves to the nodes of some distributed computing system [25]. The third is consisted in concurrent execution of all ant's activities by a supercomputer. This article describes the latter method. A comparison of all the above approaches is given in [26].

So, let's say we have a superpopulation $\bar{A}$ which contains $|\bar{A}|$ ant subpopulations $A_m, m\in[1,|\bar{A}|]$. Each ant $a_{m,n}\in A_m, n\in[1,|A_m|]$ constructs it's conformation $X_{m,n}=X_{m,n}{}^{\hat{t}}=(x_{m,n}{}^t:t\in[1,\hat{t}])$ using the migration operator:

$$P_{\psi\phi}:x_{m,n}^{t+1}=P_{\psi\phi}(x_{m,n}^t).$$

After conformations $X_{m,n}, n\in[1,|A_m|]$ of subpopulations A_m are built, the best one from each subpopulation will be selected:

$$\tilde{X}^*_{\bar{A}}=\{\tilde{X}^*_m:\tilde{f}^*_m=f(\tilde{X}^*_m)=\max_{n\in[1,|A_m|]}f(X_{m,n}),m\in[1,|\bar{A}|]\}$$

Among all the best folds $\tilde{X}^*_{\bar{A}}$ is then selected the best one of the superpopulation:

$$\tilde{X}^*:f(\tilde{X}^*)=\max_{m\in[1,|\bar{A}|]}f(\tilde{X}^*_m)=f^*$$

The software implementation of this algorithm (Fig.5) consists of the following sequential operations: the pheromones table and ants initialization: $\Phi^0, X^0_{m,n}$, calculating heuristic information $\Psi^t_{m,n}$, and then the probability of selecting protein folding directions $P^t_{\Psi,\Phi}$, increasing partial fold by one position $X^{t+1}_{m,n}$, calculating partial fold's energy at this step, updating the pheromone table after conformation's building $\Phi^-, \Phi^+, \Phi^+_{\bar{A}}$, initializing the ants for the next subpopulation.

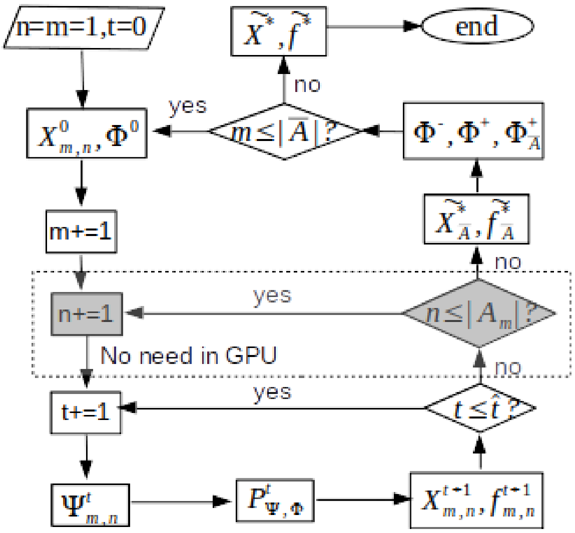


***Figure 5**. Ant colony optimization algorithm for protein folding problem*

The essence of this distribution method is that the ants of some population perform their operations simultaneously, i.e. parallel. To implement such programs, it is advisable to use graphics processing units (GPU).

3 Graphic processing units.

It is convenient to consider the GPU device architecture by the example of a specific NVidia GTX560Ti video card installed on one of the worker nodes of the distributed computing system. [27]. This device (Fig. 1a) consists of two graphic clusters with 4 stream multiprocessors (SM), the computing units, containing 48 scalar processors (SP), 2 blocks for calculating special functions, a command control block and its own memory. Additionally, a block for processing 64-bit floating point numbers is implemented. Multiprocessors communicate through slow, global memory. The interaction of scalar processors is due to fast shared memory, which is common for the processors of a single multiprocessor.

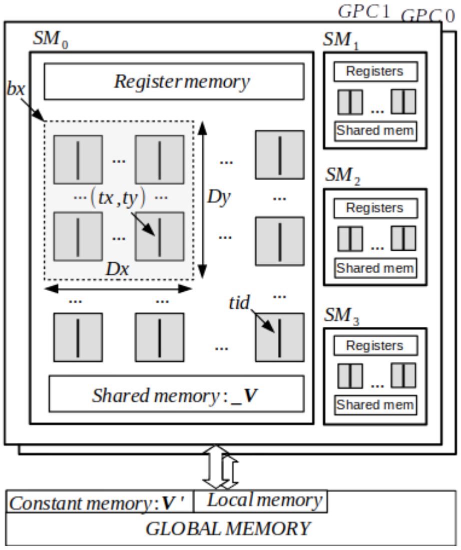


***Figure 6.** Architecture of graphics processing unit NVidia GTX 560Ti.)*

Universal computing on NVidia GPUs is provided by CUDA (Compute Unified Device Architecture) programming technology [28]. The parallel parts of the program are executed in the form of the so-called kernels:

$(results) = KERNEL <<< grid, block >>> (params)$

Here, «results» is a list of vectors, where the result of the kernel execution is written, and params are vectors that are passed to the kernel as parameters. The angle brackets indicate the architecture of the computing system for the kernel: the number of blocks in the grid and the number of threads in each block. The threads run on scalar GPU processors, and the blocks run on multiprocessors. The grid is an abstraction for a graphic cluster. Blocks and grids can be one-, two- or three-dimensional. A small amount of fast local memory is allocated to each thread, a small amount of fast shared memory is allocated to a block, and a very large amount of slow global memory is allocated to the grid. In addition, a fast immutable constant memory is allocated to the kernel, shared by all its threads.

Before describing the software implementation of the ant algorithm on GPU, we introduce the following notation:

v is a vector, $|v|$- the count of it's components; $v', v\_$ - are vectors in constant and shared memory of the gpu, accordingly (Fig.6); (tx, ty, tz) is a block's thread indexes of convenient components; $(bx, by, bz), (Dx, Dy, Dz)$ - indexes and a block sizes of convenient components.

A thread identificator is defined according formula:

$tid = tx + Dx \times ty + Dz \times tx \times ty$

For example, one-dimensional grid, containing two-dimension blocks with $Dx = Dy = |D|$ threads well be created for the kernel: $\Psi = HEURISTICS <<< |A|, (|D|, |D|) >>> (\Psi, X, la, da)$

3 GPU-ACCELERATED ANT COLONY OPTIMIZATION ALGORITHM FOR PROTEIN FOLDING PROBLEM.

Pseudo-code of the program implementation of ACO-HP-PFP-2 (Ant Colony Optimization for Hydrophobic-Polar Model Protein Folding Problem in two-dimensional conformation space) is presented in Fig. 7.

The inputs to the program are: $|\bar{A}|$- the number of ant subpopulations in the superpopulation $\bar{A}$; the number $|A|$ of ants in each subpopulation A ; the number of fold directions $|D|$; coefficients α and β , which are taking into account the influence of pheromones and heuristic information, respectively; the trace evaporation coefficient: ρ ; the initial values of the matrix of pheromones: ϕ_0 a vector for amino acid sequence Q' in which «0» corresponds to the hydrophilic acid, and «1» to hydrophobic one; a heuristic vector with $|\Psi| = |A| \times |D|$; a pheromones vector $|\Phi| = (|Q| - 1) \times |D|$, vectors for probabilities and random values for computing experiment $|P| = |R| = |A| \times |\bar{A}| \times (|Q| - 1)$, a vector of initiating points $|I| = |A| \times |\bar{A}|$, an ant's conformation vector, represented by nodes and directions X, Y nodes augmentations $\Delta X' = (1, 2 \times |Q|, -1, -2 \times |Q|)$, an energy vector F, vectors $|la| = |da| = |A|$ for actual amino acids and their directions at the step t , global optimum energy and amino acid fold storage vector $\tilde{f}$, vectors for the best ants $\tilde{A}_{\bar{A}}$, their energy $\tilde{F}_{\bar{A}}$ and tours $\tilde{X}_{\bar{A}}$ of each subpopulation.

```
Random number generation for the entire experiment
$(R) = RANDOM(R, |A| \times |\bar{A}| \times (|Q| - 1))$
$(I) = RANDOM(I, |A| \times |\bar{A}|)$
Pheromone matrix initialization
$\Phi = INITPHEROMONE <<< |Q| - 1, |D|) >>> (\Phi, \phi_0)$
The main loop across all subpopulations of the multipopulation
$for(m = 0; m \leq |\bar{A}|; m++)\{$
    initiation points initialization
    $\Phi = INITPOINTS <<< |A|, 1) >>> (X, I, m);$
    Folds of all ants from m-th subpopulation
    $for(t = 0; t \leq |Q| - 2; t++)\{$
        current positions calculation
        $(da, la) = ACTUAL <<< |A|, 1) >>> (da, la, I, t, m);$
        heuristric information's calculation
        $\Psi = HEURISTICS <<< |A|, (|D|, |D|) >>> (\Psi, X, la, da);$
        The probabilities of ant's transition calculation
        $P = PROBABILITIES <<< |A|, |D|) >>> (P, \Psi, \Phi, la, \alpha, \beta);$
        Ant's transition by one step
        $(F, X, Y) = ANTSMIGRATION <<< |A|, 1) >>> (\Psi, R, la, da, t, m);$
    $\}//t$
    best ants in subpopulations
    $(\tilde{F}_{\bar{A}}, \tilde{A}_{\bar{A}}) = BESTANTS <<< 1, |A|) >>> (\tilde{F}_{\bar{A}}, \tilde{A}_{\bar{A}}, F, m);$
    and their conformations
    $(\tilde{X}_{\bar{A}}, \tilde{A}_{\bar{A}}) = BESTCONFORMATIONS <<< 1, |Q| >>> (\tilde{X}_{\bar{A}}, \tilde{A}_{\bar{A}}, X, m)$
    Evaporation and deposit of pheromones
    $\Phi = EVAPORATE <<< |Q| - 1, |D|) >>> (\Phi, Y, F, \tilde{F}_{\bar{A}}, m);$
    $\Phi = DEPOSITE <<< |A|, |Q| - 1) >>> (\Phi, Y, F, \tilde{F}_{\bar{A}}, m);$
    Pheromones deposit by the best ant
    $\Phi = DEPOSITEBYBEST <<< 1, |Q| - 1) >>> (\Phi, Y, \tilde{A}_{\bar{A}}, m);$
    Ants initialization
    $(Y, F, X) = INITANTS <<< |A|, |Q|) >>> (X, Y, F, 0);$
$\}//m$
the best ant in the multipopulation
$(\tilde{f}) = BESTOFTHEBESTANT <<< 1, (|\bar{A}|, |D|) >>> (\tilde{f}, \tilde{F}_{\bar{A}});$
The best ant's conformation
$(\tilde{X}) = GLOBALBESTCONFORMATION <<< 1, |Q|) >>> (\tilde{X}, \tilde{X}_{\bar{A}});$
$\}$
```

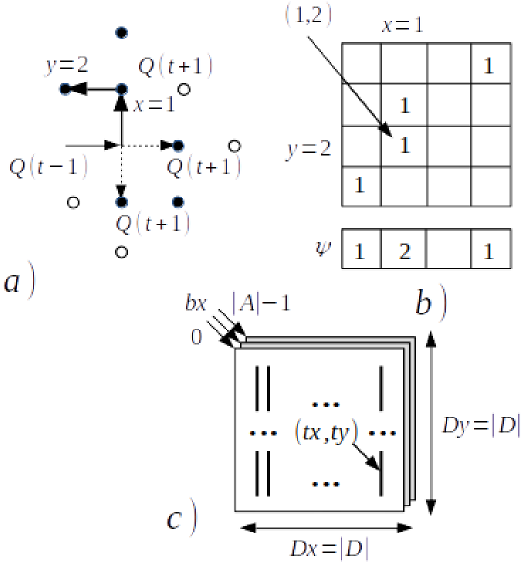
***Figure 7**. Kernel functions for gpu-accelerated ACO-HP-FPF-2*

Let us consider in detail the heuristic information kernel:

$\Psi = HEURISTICS <<< |A|, (|D|, |D|) >>> (\Psi, X, la, da);$

Let us compose a $|D| \times |D|$ table in which the columns and rows are corresponding to the four directions of the folding.

If the choice of direction x increases the number of hydrophobic contacts with the amino acid in direction y , then "1" is set at the intersection (x, y) , if not, then "0". For example, when choosing $x = 1$, the heuristic is increased by 2, as the current hydrophobic amino acid becomes a neighbor for two other hydrophobic amino acids of the same kind, indicated in (Fig. 8) in black.



			1
	1		
	1		
1			

1	2		1

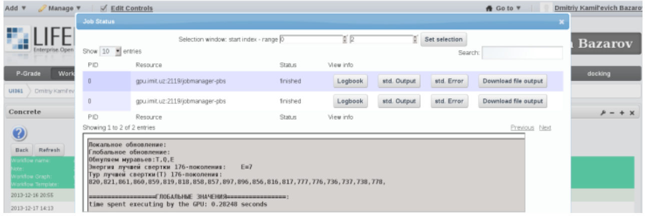
Figure 8. *Calculation of heuristic information.*

The number of tables should coincide with the number of ants in the colony and the architecture for this kernel should be: $<<<|A|,(|D|,|D|)>>>$.

And the source code of this kernel is following:

```
$HEURISTICS(\Psi, X, la, da)\{$
$bs = Dx \times bx; \Psi[ty] = 1;$
$for(i = 0; i \le |Q|-1; i++)\{\_X[i] = X[bx \times |Q| + i];\}$
$for(i = 0; i \le |Q|-1; i++)\{$
$if\ (X[la[bx]] + \Delta X'[tx] + \Delta X'[ty] == \Delta X'[bx \times |Q| + i])$
$\Psi[ty] += Q'[i] \times Q'[la[bx] + da[bx]]$
$\}$
$for(i = 0; i \le |Q|-1; i++)\{$
$if\ (\_X[l[bx]] + \Delta X[ty] == \Delta X[i]) \Psi[ty] = 0;$
$\}$
$\Psi[bx + ty] = \Psi[ty];$
$\}$
```

First, the shared memory of each block is initialized by the initial values of the heuristics (vector $\_\Psi$) and partial tours of each ant (vector $\_X$). Then, in each of the threads, it is checked whether the node has been visited in a given direction. If yes, $\_\Psi[ty]$ increases by $Q'[i] \times Q'[la[bx]+da[bx]]$, and if at least one of the amino acids is hydrophilic (i.e. $Q'[i]=0$ or $Q'[la[bx]+da[bx]]=0$), then the increment of heuristics will be zero. Here $la[bx]$ and $da[bx]$ are the actual amino acids and directions for every ant of given colony, which was found by the kernel at the previous step of the algorithm. Next, if the node in the direction ty has already been visited, then increment of heuristics is also equal to zero. The kernel ends up writing the heuristic information from shared memory to the global one: $\Psi[bx+ty]=\_\Psi[ty]$.The source codes of other kernels of the program that implements the considered algorithm on GPUs are shown in



PID	Resource	Status	View info
0	gpu.imit.uz:2119/jobmanager-pbs	finished	Logbook std. Output std. Error Download file output
0	gpu.imit.uz:2119/jobmanager-pbs	finished	Logbook std. Output std. Error Download file output

Showing 1 to 2 of 2 entries

```
Локальное обновление:
Глобальное обновление:
Обнуляем муравьев:T,Q,E
Энергия лучшей свертки 176-поколения:    E=7
Тур лучшей свертки(T) 176-поколения:
820,821,861,860,859,819,818,858,857,897,896,856,816,817,777,776,736,737,738,778,

=================ГЛОБАЛЬНЫЕ ЗНАЧЕНИЯ================:
time spent executing by the GPU: 0.28248 seconds
```

Figure 9. *The computional experiment's result on web-interface to the computational resource gpu.imit.uz with GTX560Ti video card.*

Fig. 9 shows the result of a computational experiment performed in a grid-infrastructure with support for computing on graphics processors. Energy and the tour of the best fold belonged to 176 subpopulation are shown. The overall time of calculation is 0.28 seconds. A computational experiment results are showing that this method of parallelization is suitable for modeling the process of folding large proteins (from 50 amino acid residues). After all, even a sequential enumeration of such a huge number of colonies (1024 vs 60) compared with the parallel MPI-version of the program [24, 25], for a protein of 20 amino acid residues is carried out in a fraction of a second (Fig. 6). However, this algorithm is not free from the drawbacks of gpu-computing: binding to the architectural features of the video card: for NVidia GTX560Ti (384 cores), valid values are: 1024 colonies, 32 ants each; restarting of the computing experiment leads to the construction of the same optimal ant tour with the same energy value; about 74% of the computing time it takes to transfer data between the device (GPU) and the host (CPU).

CONCLUSION

Thus, we have developed a novel GPU-based ACO algorithm for the protein folding problem, which allows us to obtain suboptimal solutions in a fraction of a second for short protein sequences.

REFERENCES

[1] K.A. Dill, “Theory for the folding and stability of globular Proteins“, J. Biochemistry, vol. 24, pp. 1501–1509, 1985. doi: http://dx.doi.org/10.1021/bi00327a032.

[2] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, M. Yannakakis, “On the complexity of protein folding“, Computation Biology vol. 5 no. 3, pp. 423-465, 1998, doi: http://dx.doi.org/10.1089/cmb.1998.5.423

[3] A. Shmygelska, H.H. Hoos, An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem Bmc Bioinformatics vol. 6 no. 30, 2005, doi: http://dx.doi.org/10.1186/1471-2105-6-30

[4] A.S. Kolaskar, S. Sawant, “Prediction of conformational states of amino acids using a Ramachandran plot“, International Journal of Peptide and Protein Research vol. 47(1-2), pp. 110-116, 1996, doi: http: //dx.doi.org/10.1186/1471-2105-6-30

[5] C. Levinthal. “How to fold graciously“, J. Mossbaun spectroscopy in Biological Systems Proceedings, pp. 22-24, 1969.

[6] A.V. Finkelstein, O.B. Ptitsyn, Protein physics. San Diego: Acadimic Press.,354P, 2002.

[7] A.V. Makeev, The basics of biology [Osnovy biologii]. Moscow: Mir. 235 p, 1997. (in Russian)

[8] A.M. Malyshenko, The mathematical foundations of systems theory [Matematicheskie osnovy teorii system] Tomsk: Polytechnic University Publ., 364 p. (in Russian)

[9] B. Alberts, A. Johnson, Lewis J. Molecular Biology of the Cell. New York Garland Science Available at: https://www.ncbi.nlm.nih.gov/books/NBK26830, 2002.

[10] S. Istrail, F. Lam. Combinatorial algorithms for protein folding in lattice models: a survey of mathematical results Commun. Inf. Syst. 9(4):303–346 doi: http://dx.doi.org/ 2009

[11] F.M. Richards. Areas, volumes, packing and protein structure Annual Review of Biophysics and Bioengineering 6(1):151-176 doi: http://dx.doi.org/10.1146/annurev.bb.06.060177.001055 1977

[12] K.A. Dill. Dominant forces in protein folding. Biochemistry 29(31):7133-7155 doi: http://dx.doi.org/10.1021/bi00483a001, 1990.

[13] M. Mann, S. Will, R. Backofen. CPSP-tools - Exact and complete algorithms for high-throughput 3D lattice protein studies/ Bmc Bioinformatics 9(1):230 doi: http://dx.doi.org/10.1186/1471-2105-9-230, 2008.

[14] C. Thachuk, A. Shmygelska, H.H. Hoos. 2007. A replica exchange Monte Carlo algorithm for protein folding in the HP model Bmc Bioinformatics 8:342 doi: http://dx.doi.org/10.1186/1471-2105-8-342

[15] A.P. Karpenko. Modern search optimization algorithms. Algorithms, inspired by nature [Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennye prirodoy]. Moscow.: Bauman MSTU Publ., 446 p., 2014. (in Russian).

[16] S. Gordon. The self-avoiding walk: a brief survey. Proceednigs of the 33rd SPA Conference. Surveys in Stochastic Processes. Berlin: European Math. Society. 181–199, 2010.

[17] T. Thalheim, D. Merkle, M. Middendorf. Protein folding in the HP-model solved with a hybrid population based ACO Algorithm. IAENG International Journal of Computer Science 35(3):201-300 doi: http://dx.doi.org/ 2008.

[18] Xiao-Min Hu, Jun Zhang, Yun Li. Orthogonal methods based ant colony search for solving continious optimization problems. Journal of Computer Science and Technology 23(1):2-18 doi: http://dx.doi.org/10.1007/s11390-008-9111-5 2008.

[19] Stuzle T., Doringo M. A short convergence proof for a class of ACO algorithms. IEEE Transactions on Evolutionary Computation 6(4):358-365 doi: http://dx.doi.org/10.1109/TEVC.2002.802444 2002.

[20] L. Carvelli , G. Sebastiani. Some issues of aco algorithm convergence Ant Colony Optimization - Methods and Applications 6(4):358-365 doi: http://dx.doi.org/10.5772/14510 2011

[21] D. Chu, M. Till, A. Zomaya. Parallel ant colony optimization for 3D protein structure prediction using the HP lattice model Proceedings 19th IEEE International Parallel and Distributed Processing Symposium vol. 7. Washington: IEEE Computer Society. 193. 2005.

[22] Delisle, P., Krajecki M., Gravel M., Gagne C 2005. Parallel implementation of an ant colony optimization metaheuristic with OpenMP Proceedings of the 3rd European Workshop on OpenMP (EWOMP'01). Barselona: IEEE Computer Society. 193.

[23] Wen-mei W. Hwu. GPU computing gems emerald edition. San Francisco: Morgan Kaufmann Publishers Inc., CA, USA, 886 p. 2011.

[24] T.F. Bekmuratov, D.T. Muhamedieva, R.K. Bazarov, D.D. Ahmedov. Parallel ant colony optimization algorithm [Parallelnyi muravyinyi algoritm optimizacii]. Information and energetics problems [Problemy informatiki i energetiki] 1-2:11–15. doi: http://dx.doi.org/. (In Russian) 2014.

[25] T. F. Bekmuratov, R.K. Bazarov, D.K. Bazarov. The implementation of ant colony optimization algorithm for the study of protein folding problem in distributed systems by the software agents Problems of Computational and Applied Mathematics [Realizaciya muravyinogo algoritma foldinga belkov metodami programmnyh agentov v raspredelennyh systemah]. [Problemy vychislitelnoy i prikladnoy matematiki] 2(8):103–113. 2017doi: http://dx.doi.org/. (In Russian)

[26] R.K. Bazarov. The solving of protein folding problem in distributed computing grid-systems [Reshenie zadachi foldinga belkov v raspredelennyh vychislitelnyh grid-sistemach]. 6th Symposium (International) "New Energy Saving Subsoil Technologies and the Increasing of the Oil and Gas Impact" Proceedings of Republic scientific and technical conference, Tashkent pp. 130-131, mart 10-11, 2016. (In Russian)

[27] V.P. Bruskov, D.K. Bazarov. Grid infrastructure with support for computing on GPUs [Grid-infrastructura s podderjkoy vychisleniy na graficheskih processorah]. "Computer science: problems, methodology, technologies", Proceedings of XVII international scientific and methodological conference, Voronej, pp 190-195, 2017.

[28] CUDA Zone. Available at: https://developer.nvidia.com/cuda-zone