# A Hybrid Approach to the Maximum Clique Problem in the Domain of Information Management

**Demidovskij Alexander, Babkin Eduard and Tatiana Babkina**

**Abstract** In this paper we observe the opportunity to offer new methods of solving NP-hard problems which frequently arise in the domain of information management, including design of database structures and big data processing. In our research we are focusing on the Maximum Clique Problem (MCP) and propose a new approach to solving that problem. The approach combines the artificial neuro-network paradigm and genetic programming. For boosting the convergence of the Hopfield Neural Network (HNN) we propose the genetic algorithm as the selection mechanism for terms of energy function. As a result, we demonstrate the proposed approach on experimental graphs and formulate two hypotheses for further research.

**Keywords** Information management · Maximum clique problem · Hopfield neural network · Genetic algorithm

## 1 Introduction

This paper is dedicated to the study of one of classical NP-hard problems, proposed by Karp in 1972, which is the Maximum Clique Problem (MCP) [5, 6]. Special emphasis should be laid on the variety of practical implications of the MCP to solving significant real-life problems in the domain of information management. In that domain not only the problem of big data processing is actual, but also the problem of engineering the structures and algorithms for comparison of large amounts of data. In other words, the process of revealing common elements and general regularity in

D. Alexander · B. Eduard (✉) · T. Babkina
Higher School of Economics, Bolshaya Pecherskaya 25, Nizhny Novgorod, Russia
e-mail: eababkin@hse.ru

D. Alexander
e-mail: ademidovskiy@hse.ru

T. Babkina
e-mail: taty-bab@yandex.ru
URL:http://www.hse.ru

V. Ravi et al. (eds.), *Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO - 2015)*, Advances in Intelligent Systems and Computing 415, DOI 10.1007/978-3-319-27212-2_25

various objects of the environment becomes the serious task for modern researchers. It is obvious that these computations are going to be more complex as data amount is rocketing.

To begin with, we propose to consider the opportunity of application of the MCP in issues connected with databases and, in particular, their design. There is no doubt that any information obtained should be stored somewhere at least for reasons of being available for further investigations. Due to the fact that amounts of information are rocketing the problem of the proper design of the database appears to be the crucial one in modern circumstances. On the other hand, besides the availability, the other problem arises the problem of efficiency of the database structure e.g. from the glance of convenient retrieving data or getting rid of extra dependencies and duplication. These reasons make the problem of designing databases structure of the great importance. Artifacts gained after experiments can vary in their content and at the same time they can have very similar features. For instance, when analyzing activities of sportsmen we know that all of them have the frequency of heart beat, frequency of their breathing etc. However, for swimmers time would mean something quite different from time of the runner. This brings to the idea of creating such relational databases that would recognize such similarities and differences of the given data and store it according to these features.

In this field fundamental principles of MCP application were proposed by multiple researchers [2, 3, 13]. In particular, Beeri et al. [2] proposed the idea of using maximum cliques to reveal common properties of the given data and this idea was applied to the special class of database schemes, called acyclic. In their later research Beeri et al. [3] developed some techniques of manipulating existing dependencies and again, after casting the database scheme to the split-free normal form, it can be referred to as the graph, where the clique denotes the combination of closely related features to be joined for storing appropriate data. The amazing breakthrough is that such a method can be already applied to relational (cyclic or acyclic) databases.

In order to emphasize the relevance of our research topic, in addition to the problems mentioned above we should draw attention to plenty of other tasks, which can be easily interpreted in terms of the MCP, where entities denote vertices and edges represent interactions between these entities. To be concise, such tasks are: financial analysis, diseases detection and classification, data mining, expert finding and so on. This problem has various practical implications like template recognition, computer vision, comparison of biological structures, producing of complex medicaments, revealing mutations in human genome etc.

From the very beginning, multiple attempts have been made to solve the MCP problem by different methods. Some of them will be observed further in this paper as, for instance, the modifications of the Branch-and-Bound algorithm [15, 18], which uses special heuristics of graph colouring. The most important drawback of these studies is the loss of universality because of focusing on several special types of graphs even though these approaches over-perform more traditional and universal solutions.

The general goal of our work is to elaborate the neuro-network approach to solving the MCP in the arbitrary graph. After correct formulation of mathematical statement of the task neural networks are considered to be a universal tool, which has a great potential from the point of application of parallel computing technologies. In present time the neural network approach is often considered as the most efficient and winning one both in case of speed and precision of results [11, 17, 19–21]. Despite of that, a neural network is able to recognize even noisy data and provide correct results. Moreover, it is essential to mention that neural networks are in most cases resistant to failure. In particular, one of the most valuable features of the Hopfield Neural Networks (HNN) [10] is a relatively simple parallelization, which reveals the additional potential of the usage of the HNN.

In the current paper we propose the novel approach to solving the MCP. Lets refer to this approach hereinafter to as the hybrid approach. It is based on two basic milestones and combines the flexibility and ease of parallelization of the HNN and the adoption ability of genetic algorithms. The proposed method provides the higher speed of a neural network convergence and as a result it is expected to significantly decrease time of solving the MCP on each instance of known graphs from the large DIMACS[1] (Center for Discrete Mathematics and Theoretical Computer Science) dataset. Therefore, our novel technique to solving the MCP should be considered as the efficient and cost-effective tool.

This paper is organized as follows. In Sect. 2 the MCP is formulated and we observe approaches which were already applied to maximum clique search, especially we focus on the structure of the HNN. Furthermore (in Sect. 3), we explain the novel design of the HNN for this problem, which is based on the usage of the genetic algorithm as the selection mechanism for coefficients of the energy function, then we propose the general technique of solving the MCP on the sample graph, and, finally, we describe the structure of the distributed framework which we use as the experimental platform for validation of the proposed in this paper approach. In Sect. 4 we demonstrate results of experiments on a low-dimensional graph from the Second DIMACS dataset. The final analysis and conclusions are given in Sect. 5.

## 2 Background of the Study

First of all, several terms and definitions should be given for further formulating the MCP.

**Definition 1** (*Graph*) A graph G consists of the set of vertices V and the collection (not only the set) of not ordered pairs of vertices, being referred to as edges. The graph is denoted by G = (V, E).

**Definition 2** (*Complete graph*) A complete graph $K_n$ is the graph, which consists of n vertices and for every pair of vertices exists the edge between them.

**Definition 3** (*Subgraph*) A graph H = (W, F) is called a subgraph of G = (V, E) if W is a subset of V, F is a subset of E.

[1] http://dimacs.rutgers.edu/Challenges/.

**Definition 4** (*Clique*) The clique in graph G is the complete subgraph of graph G.

**Definition 5** (*Adjacency matrix*) An adjacency matrix of graph is the matrix $A_{nn} = [a_{ij}]$, where the non diagonal element $a_{ij}$ denotes the existence of the edge between vertices $i$ and $j$, the diagonal element $a_{ii}$ denotes the existence of cycles in the vertice $i$. The adjacency matrix of graph is symmetric: $a_{ij} = a_{ji}$ for each $i$ and $j$. The adjacency matrix in the simple graph is binary and every diagonal element is equal to 0. For the complete graph $K_n$ the diagonal element is equal to 1.

**Definition 6** (*Maximum clique*) A maximum clique is the clique which size (the number of vertices) is maximum.

Also let us introduce here some definitions required for basic understanding of the proposed approach in the case of the genetic algorithm as the selection tool for the coefficients of the terms of the energy function:

**Definition 7** (*Chromosome*) Cromosome is the bit string which represents the individual. It represents the genetic information.

**Definition 8** (*Individual*) Individual is the entity, which stores the chromosome.

**Definition 9** (*Population*) Population is number of individuals.

**Definition 10** (*Crossover*) Crossover is the operation of creation of new individual by mixing chromosomes of two individuals in the population.

**Definition 11** (*Mutation*) Mutation is an operation of arbitrary changing the chromosome of the single individual in the population.

Having defined these definitions it is now possible to represent the mathematical formulation of the problem. The MCP is all about the search of the maximum clique in the graph. Lets denote the size of the clique for graph G as $w$(G). Due to the fact that we are looking for the maximum clique we have to maximize the number of vertices in the clique. Mathematically it can be posed as follows (1):

$$\begin{gathered} max(\sum_{i=0}^{n} x_i), \\ where\ x_i + x_j \leq 1\ \forall(i,j) \subseteq E \\ x_i \subseteq \{0,1\}\ i = 1,..,n \\ x_i = \begin{cases} 0, \text{if vertice } i \text{ is in clique} \\ 1, \text{otherwise} \end{cases} \end{gathered} \tag{1}$$

Various attempts have been made to elaborate an efficient approach and apply it to the MCP: enumeration algorithms, genetic algorithms, tabu-search, the branch-and-bound algorithm etc. [5, 6, 18] In one of the recent studies ([15]) the MCP was applied to the analysis of large real-life networks for retrieving overlapping communities. More importantly, not only was the community analysis approach proposed, but also a rather competitive advanced Branch-and-Bound algorithm, where the pruning routine was improved to make the algorithm be applicable to massive graphs. The results were reported not only for standard DIMACS dataset, but also for real-world graphs like those from the Stanford Large Network Dataset Collection. Later, in this paper we will use these results as benchmark ones.

The special emphasis should be laid on the neuro-network paradigm due to the opportunity of parallelization and learning (self-learning). As the MCP is the rather complex problem it is especially crucial for the approach to be able to adapt to the input data and produce qualitative results. First attempts to solve this problem using neural networks are practically not comparable with modern studies due to the fact that researches did not include experimental results in papers and reports. Thus, the earliest approach, which already included results of experiments, was the research of Lin and Lee [14], which used 0–1 formulation, as the basis for the heuristic that they applied to the MCP. The second important contribution to the investigation of this problem was made by Grossman [8], who proposed the discrete HNN with an additional threshold parameter, which defined the degree of stability of the neural network. For manipulation of that parameter it was proposed to use simulation annealing. However, the reported results of experiments on DIMACS graph were over-performed by more modern approaches.

The special attention should be paid to the approach, proposed by Jagota et al. [11]. The new contemporary trend, proposed by these scientists, had immediate consequences for the general direction of research. The idea is to decrease dynamics stochastically as much as possible and combine it with mean field annealing imitation. The core problem with this idea is lack of performance on small graphs when comparing to simpler methods, but on high dimension graphs the algorithm outperformed traditional approaches. More importantly, the subject inspired a lot of substantive debate among professionals not only due to performance, but because of non-optimal final solution as well. Jagota’ s research has called into question the widespread trend to develop approaches to find the maximum clique, while it is even more important to solve the generalized problem: finding the clique of a certain size. The annealing imitation idea receives strong support as it provides an opportunity to solve the general problem. Due to this fact the idea can be found in numerous publications dedicated to the hierarchical cluster division in pattern recognition operating with the methods of the graph theory. Stochastic networks slightly overperform other similar approaches in case of producing more precise results when applying simulation annealing in the HNN for escaping from a local minimum. The similar approach is used in Boltzmann machines [9]. However the process of modelling these networks is the complex task even for low-dimensional problems.

Therefore, in the framework of current study we apply the deterministic HNN [10, 11, 19]. In 1982 Hopfield proved that such a network can be used for approximation of optimization problems because the network converges to the state which denotes the minimum of energy function if following conditions are satisfied: the weight matrix W should be symmetric ($w_{ij} = w_{ji}$) and the weight of self-connection of every neuron should be equal to 0: $w_{ii} = 0$. The procedure of the neuron update can be observed as the search of the minimum in the solution space of the energy function.

## 3 The Novel Design of the HNN Dynamics for Solving the MCP

Due to the fact that the dynamics of the HNN is completely described by the energy function, it is important to adopt the general energy function according to the specific conditions of the MCP. The energy function consists of several summands which are usually referred to as terms of the energy function. Traditionally, when being applied to solving various problems, the HNN is working according to statically chosen and once assigned coefficients of these terms. There is no doubt that these coefficients define the dynamics of the HNN, which in other words means that they have an impact on the speed of convergence of the neural network. That is why it is important to choose their values according to each instance of problem to which the HNN is applied in order to obtain best results. This brings to the lack of flexibility of the proposed approaches and tools. In this paper we propose an innovative technique of applying the genetic paradigm as the selection tool for coefficients of the energy function. In other words we introduce a flexible mechanism of adopting the HNN to every graph, in which we try to find the maximum clique. In the rest part of this section we will examine the proposed genetic approach to solving the MCP.

To begin with, we have to formulate the energy function which fully describes the dynamics of the HNN. The example of formulating the energy function can be found in [19], however, we propose the novel energy function. To ensure that we are able to solve the MCP with the HNN we need the energy function reflect the the MCP. We have already formulated the MCP earlier (1) and we use it to build the respective energy function (2):

$$-1/2 * A * \sum_{i=1}^{N} \sum_{j \neq i}^{N} d_{ij} x_i x_j + B * \sum_{i=1}^{N} x_i, \tag{2}$$

where the output of $x_j$ neuron $i$ :

$$x_j = \begin{cases} 1, & \mathit{if\ neuron\ j\ is\ in\ maximum\ clique} \\ 0, & \mathit{otherwise} \end{cases} \tag{3}$$

$$d_{ij} = \begin{cases} 1, & \mathit{if\ vertices\ i\ and\ j\ are\ connected\ with\ an\ edge} \\ 0, & \mathit{otherwize} \end{cases} \tag{4}$$

A, B—constants. However, we should modify the energy function to the canonical energy function form of the HNN. For this reason we have modified the weight assignment rule proposed in [20] as follows:

$$w_{ij} = d_{ij}(1 - \delta ij) \tag{5}$$

$$h_i = B \tag{6}$$

$$\delta ij = \begin{cases} 1, & \textit{if i is equal to j} \\ 0, & \textit{otherwise} \end{cases} \tag{7}$$

Now the energy function has the canonical form: (3):

$$-1/2 * A * \sum_{i=1}^{N} \sum_{j \neq i}^{N} w_{ij} x_i x_j + \sum_{i=1}^{N} h_i x_i,$$

as for the activation function—we use the classical sigmoid function.

As we can can see in (3) it has two terms. So, we need to find optimal values of these two coefficients. In order to solve the problem of finding coefficients it is exceedingly important to map it in terms of the genetic paradigm. Therefore, we denote the set of coefficients as the chromosome of the size equals two. Next, we should define the size of the population, the maximum number of generations etc. Moreover, we should define if the crossover and mutation mechanism are used, ranges of probabilities for entities to be involved in crossover or mutation. Finally, the finishing condition should be defined as the number of generations, where the minimum suitability remains stable. Having defined these parameters, we can proceed to the exact algorithm of finding optimal coefficients.

Firstly, we initialize the list of entities, according to the defined earlier rules and generate the initial population. Secondly, we estimate the suitability of the initial population and get that one with the highest estimation. After that we pass parameters obtained to the HNN, which cycles until it does not converge. If it does not converge, then we define the suitability of the corresponding chromosome as the worst one. After that, if there is no optimal solution in initial population, the probabilities of entities to be involved in crossover and mutation are then estimated and, according to these probabilities, we prepare and perform the crossover and mutation algorithms. In particular, when applying crossover it is essential to make it only for those entities, whose probabilities are from the corresponding range, discussed earlier. Then, chromosomes, which passed this filtering, are translated to the Gray code, grouped in pairs, each one split into two halves and, finally, they are joined in such a way, that two halves of one chromosome can not organize the new entity. After that, chromosomes are again translated back, checked for the uniqueness and the suitability of every entity is estimated.

The second step is the mutation procedure. Here we use the random selection mechanism for getting entities for mutation, then we apply the transformation of chromosomes to the Gray code and randomly change bits of the chromosome code.

Again, after finishing the procedure we check the uniqueness of the generated chromosomes and calculate their suitability. Finally, all entities are combined, sorted by their suitability and then the selection procedure is performed. As the stop condition we observe the number of cycles of the genetic algorithm while which the minimum suitability among all entities in population remains stable.

To provide deeper comprehension of this specific design of the HNN we introduce the simple graph which is represented in Fig. 1. When speaking in terms of database design, as we discussed it earlier in the Sect. 1, nodes can represent features that we try to store in the database and the clique then represents that these features, which respective nodes are in the clique, are close to each other and should be joined to store appropriate data.

It is obvious that the maximum clique exists in this graph and it consists of vertices $\{2, 3, 5\}$. As it was already mentioned above, while constructing the neural network we map every edge in the neuron, where the weight between neurons denotes the following: the weight of connection is stronger if and only if these two neurons will be in the final solution. In terms of the HNN this means that the energy function decreases when comparing with its current state.

Firstly, we compose the adjacency matrix of this graph (Table 1):

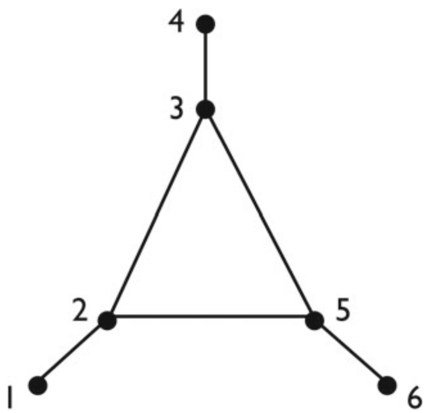

**Fig. 1** A sample graph (the maximum clique is {2,3,5})

**Table 1** Adjacency matrix of the graph represented in Fig. 1

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 |

Next, we make mapping from the 6-vertices graph to the HNN with one layer (Fig. 2).

After the HNN converges we obtain the values at the axons of neurons, which correspond to the output of the neural network. Then we apply the filter to this output and get neurons which output value is extremely close to 1. Having mapped neurons back to the graph structure, we get the resulting set of vertices which stands for the maximum clique. For further investigation and study of the neuro-network approach features in this paper we use the distributed framework described by Karpunina [1]. Besides the HNN the genetic algorithm was also implemented in this framework, which we use in each iteration as the selection mechanism for terms of the energy function. It was implemented in the C++ programming language in the Object-Oriented paradigm. The general principle of the framework functioning is represented in Fig. 3:

In this distributed framework the MPI technology (Message Passing Interface) [4] was used as the mechanism for mass parallelization, which enables us to make computations easily distributed on several clusters, while not losing any computation power and being very reliable. This framework, due to the flexible entity of the HNN, can qualitatively solve the problem if it is formulated correctly. More importantly, we can expect the boost up to 11 times [4] in computing the final solution when comparing with the more traditional approach.

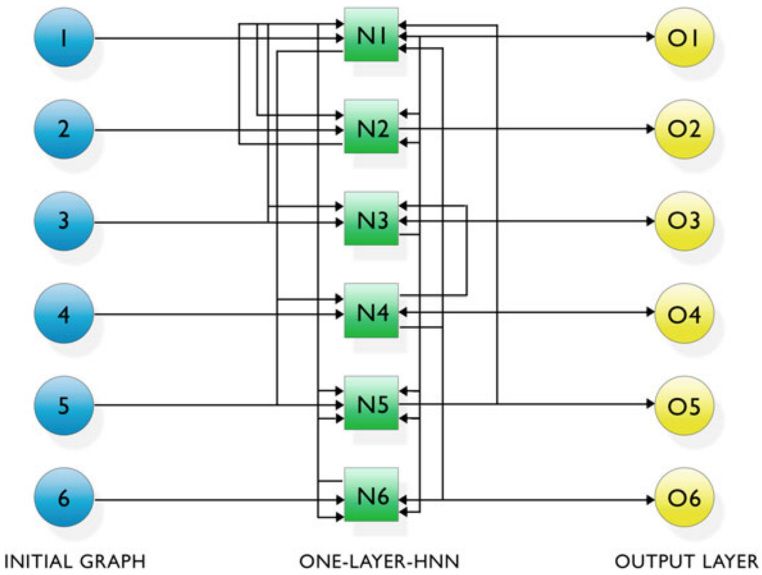

**Fig. 2** General technique of solving the MCP on the example of small graph from Fig. 1

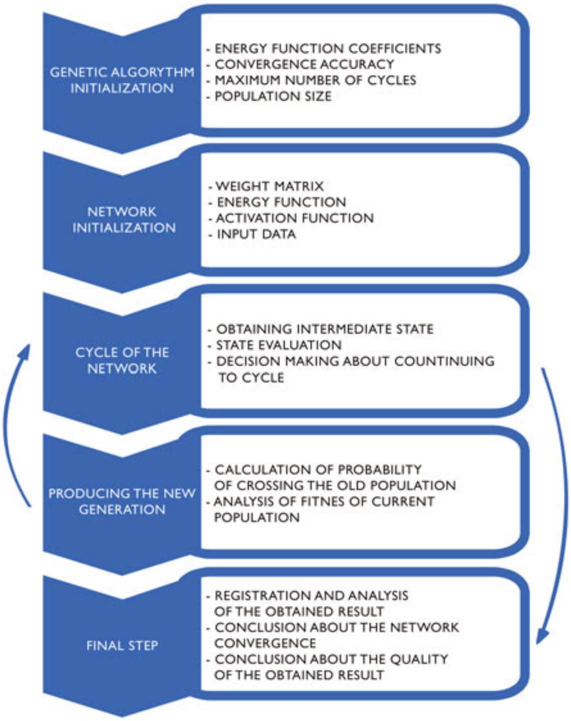
GENETIC ALGORYTHM INITIALIZATION
- ENERGY FUNCTION COEFFICIENTS
- CONVERGENCE ACCURACY
- MAXIMUM NUMBER OF CYCLES
- POPULATION SIZE
NETWORK INITIALIZATION
- WEIGHT MATRIX
- ENERGY FUNCTION
- ACTIVATION FUNCTION
- INPUT DATA
CYCLE OF THE NETWORK
- OBTAINING INTERMEDIATE STATE
- STATE EVALUATION
- DECISION MAKING ABOUT COUNTINUING TO CYCLE
PRODUCING THE NEW GENERATION
- CALCULATION OF PROBABILITY OF CROSSING THE OLD POPULATION
- ANALYSIS OF FITNES OF CURRENT POPULATION
FINAL STEP
- REGISTRATION AND ANALYSIS OF THE OBTAINED RESULT
- CONCLUSION ABOUT THE NETWORK CONVERGENCE
- CONCLUSION ABOUT THE QUALITY OF THE OBTAINED RESULT

**Fig. 3** The structure of the distributed framework

# 4 Experiments

In this section we introduce results of the preliminary analysis of the proposed approach. The data consists of the general information about the analysed graph and the values of the size of the clique and time spent when applying our innovative technique.

These experiments we carried out on the Intel i5 vPro processor, 4GB CPU, host OS—Windows OS.

Results obtained are demonstrated in Table 4, where:
|V|—number of vertices,
|E|—number of edges,
|W|—actual size of the clique,
$|W_{HNN\&GA}|$—size of the clique obtained by the proposed approach,
$t_{HNN\&GA}$—average time spent on obtaining the result using the proposed approach,
$|W_{BnB[13]}|$—size of the clique obtained by the Branch-and-Bound algorithm [18]
$t_{BnB[13]}$—average time spent on obtaining the result using the Branch-and-Bound algorithm [18]

As we can see from the table we have used some graph instances from the DIMACS benchmark dataset. It contains graphs of various density, size and struc-

**Table 2** Results obtained on the several graphs from the DIMACS dataset

| Name | $\|V\|$ | $\|E\|$ | $\|W\|$ | $\|W_{BnB[13]}\|$ | $t_{BnB[13]}$ | $\|W_{HNN\&GA}\|$ | $t_{HNN\&GA}$ |
|---|---|---|---|---|---|---|---|
| C125.9 | 125 | 6963 | 34 | 34 | no information | 34 | 0.015 |
| C250.9 | 250 | 27984 | 44 | 44 | no information | 44 | 0.02 |
| C500.9 | 500 | 112332 | 57 | 57 | no information | 56 | 0.1 |
| brock200_2 | 200 | 9876 | 13 | 13 | 0.016 | 13 | 0.05 |
| brock200_4 | 200 | 13089 | 17 | 17 | 0.32 | 16 | 0.4 |

ture. Most of these graphs are divided into groups often referred to as families of graphs, which represent real-life structures and systems. Although his dataset is not the unique one and there are multiple sets of graphs, but historically DIMACS dataset is used in most of papers and, therefore, we also reference to its graphs for benchmarking our proposed solution (Table 2).

From the table we can see that the obtained results are optimal in most of cases, which proves the validity of the proposed approach. However, there is a slight drawback in the size of the maximum clique obtained when comparing with the traditional Branch-and-Bound algorithm [18]. This brings us to the conclusion that our algorithm is rather precise because it produces near-optimal results. Therefore, the usage of the rather computationally complicated state transition mechanism such as the genetic algorithm is a rather efficient technique. Moreover, it is expected to provide much better results on the graphs of the large density and dimension.

However, efficiency of the algorithm is determined not only by the precision of results but also by the cost effectiveness, which means that the efficient approach is a trade-off between the quality and speed of finding the solution. More importantly the proposed approach should be compared to Branch-and-Bound approaches in terms of costs and, in particular, in terms of computational time. To maintain the consistency of the results reported we included information about computational time for one of existing and well-known approaches observed before ([18]). As for the proposed approach we should state that we have got quite similar results, which ensures that the novel design of the dynamics of the HNN provides us with the competitive solution. However, there is the drawback in the minor overhead which stays for the genetic algorithm needs. In general, there is the opportunity of optimizing its primitives to overcome this obstacle in time losses.

## 5 Conclusions and Directions for Further Research

To sum up, in the framework of this study the broad investigation of the MCP has been conducted, which revealed the actuality of this problem for the modern methods of information management (for example, design of complex data base structures).

The special emphasis should be laid on the neuro-network approach and generic formulations of the problem to allow solving this problem on the arbitrary graph. As we have stated it at the beginning of this paper we refer to the novel proposed algorithm as to the *hybrid* one. The nature of this algorithm combines two paradigms: neural-networks and genetic algorithms, where each one serves the exact aim: basic calculations were performed by the HNN and the selection mechanism of coefficients of the energy function was implemented with the help of the genetic algorithm. The former provides ease of implementation and parallelization while the second considerably cuts the number of cycles of the HNN and improves the flexibility of solving the MCP with the HNN.

In our particular research we offer a special selection mechanism for the coefficients of the energy function of the Hopfield Neural Network (HNN), which provides the flexibility and universalism of the proposed solution. The novelty of our contribution is the efficient combination of the genetic paradigm and the dynamics of the HNN to solving the MCP. To prove the validity of the proposed approach we have implemented it on the basis of the distributed neuro-network framework. Finally, having used the software prototype we compared the quality of results with modifications of another widely used approach - the Branch-and-Bound algorithm [15, 18]. Achieved results form a generic formal framework which may be specialized for particular engineering problems of information management. Specification and analysis of such particular cases are included to our research agenda.

Another valuable result of this paper is that we have identified key hypotheses that can help to build a more powerful and competitive approach to the MCP. They are the following:

1. the application of advanced heuristics can provide much better results. This hypothesis is based on the fact that the HNN is very likely to locate the local optimum instead of the global one, therefore the application of new heuristics can produce better solutions;
2. the application of the distributed implementation of the proposed approach is likely to provide significant speedups in solving the MCP;

Formulated above hypotheses can be considered as potential improvements to the proposed approach and further experiments can prove its validity. These hypotheses are expected to become crucial directions of the further research. In particular, it would be logical to use the advanced distributed Tabu-Machine [1] that is also implemented in the observed above framework. As we have seen in [15] massive parallelization can considerably boost the algorithm and cut computation time.

Finally, we are convinced, that it is important to lay special emphasis on the application of the MCP to the analysis of social networks. Due to the growing interest, inherent complexity and noise in the analysis of social networks we consider that the MCP approach can be efficiently applied to this task and it should be evaluated as the considerably essential tool for such analysis.

More importantly, as millions of people are using Web as the collaboration and work platform, it reveals great opportunities for collecting data about peoples preferences and interests. According to various parameters people can be divided into different groups and subgroups, which are often referred to as *user communities*. Such a division opens new perspectives for business because it enables precise targeting and providing customer-oriented products and services. Such an attempt to reveal user communities was proposed in [16]. The core idea proposed in this research is that it is possible to build a weighted graph according to features specific for users. After applying the threshold filter we get the unweighted graph, where all maximal cliques are found. Not only shows it how people can be grouped, but also helps to obtain some behavioral patterns. From the point of the social network analysis [12] such revealed communities can describe the society analyzed in terms of speed of information exchange, probability of appearance of conflicts and so on. For instance, all these conclusions can be made on the basis of the level of overlapping of these communities. One of considerable examples of analyzing real-life problems is [7], where authors analyzed the social network Ipernity.com as the directed labeled graph.

For the problem of social networks analysis, we propose to imagine that the graph we have analyzed (1) is the small network, for instance, it is the project group. As we have already discussed, we can model such a network as the undirected graph and try to find the maximum clique. Having obtained this clique we can easily detect the core part of the project team, understand the way interactions are performed there and, as a result, it can help the manager or the product owner to better organize the work of this small community. This small and rather artificial example should be considered as the example of application of the MCP to the real-world cases.

## References

1. Babkin, E., Karpunina, M.: Towards application of neural networks for optimal synthesis of distributed database systems. In: Proceedings of the 12th IEEE International Conference on Electronics, Circuits, and Systems, Satellite Workshop Modeling, Computation and Services (2005)
2. Beeri, C., Fagin, R., Maier, D., Yannakakis, M.: On the desirability of acyclic database schemes. J. ACM (JACM) **30**, 479–513 (1983)
3. Beeri, C., Kifer, M.: An integrated approach to logical design of relational database schemes. ACM Trans. Database Syst. (TODS) **11**, 134–158 (1986)
4. Dahl, G., McAvinney, A., Newhall, T., et al.: Parallelizing neural network training for cluster systems. In: Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks, pp. 220–225. ACTA Press (2008)
5. Du, D.-Z., Pardalos, P.M.: Handbook of Combinatorial Optimization, Second Edn. Springer, Berlin (2013)
6. Du, D.-Z., Pardalos, P.M.: Handbook of Combinatorial Optimization. Springer, New York (1999)
7. Erétéo, G., Buffa, M., Gandon, F., Corby, O.: Analysis of a Real Online Social Network Using Semantic Web Frameworks. Springer, Berlin (2009)
8. Grossman, T.: Applying the INN model to the maximum clique problem. Cliques Color. Satisf. **26**, 125–146 (1996)
9. Hinton, G.E., Sejnowski, T.J.: Learning and relearning in Boltzmann machines. MIT Press, Cambridge, Mass. **1**, 282–317 (1986)
10. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. **79**, 2554–2558 (1982)

11. Jagota, A.: Approximating maximum clique with a Hopfield network. IEEE Trans. Neural Netw. **6**, 724–735 (1995)
12. Jamali, M., Abolhassani, H.: Different aspects of social network analysis. In: IEEE/WIC/ACM International Conference on Web Intelligence, pp. 66–72. Springer (2006)
13. Kwok, Y.-K., Karlapalem, K., Ahmad, I., Pun, N.M.: Design and evaluation of data allocation algorithms for distributed multimedia database systems. IEEE J. Sel. Areas Commun. **14**, 1332–1348 (1996)
14. Lin, F.: A parallel computation network for the maximum clique problem. In: ISCAS'93, 1993 IEEE International Symposium on Circuits and Systems, pp. 2549–2552 (1993)
15. Pattabiraman, B., Patwary, Md., Ali, M., Gebremedhin, A.H., Liao, W.-K., Choudhary, A.: Fast Algorithms for the Maximum Clique Problem on Massive Graphs with Applications to Overlapping Community Detection (2014)
16. Pierrakos, D., Paliouras, G., Papatheodorou, C., Karkaletsis, V., Dikaiakos, M.D.: Web community directories: a new approach to web personalization. In: Berendt, B., Hotho, A., Mladenič, D., van Someren, M., Spiliopoulou, M., Stumme, G. (eds.) EWMF 2003. LNCS (LNAI), vol. 3209, pp. 113–129. Springer, Heidelberg (2004)
17. Smith, K.A.: Neural networks for combinatorial optimization: a review of more than a decade of research. INFORMS J. Comput. **11**, 15–34 (1999)
18. Tomita, E., Seki, T.: An efficient branch-and-bound algorithm for finding a maximum clique. Discrete Mathematics and Theoretical Computer Science, pp. 278–289. Springer, Berlin (2003)
19. Wang, R.L., Tang, Z., Cao, Q.P.: An efficient approximation algorithm for finding a maximum clique using Hopfield network learning. Neural Comput. **15**, 1605–1619 (2003)
20. Wang, Y.-F., Cui, G.-Z., Zhang, X.-C., Huang, B.-Y.: An improved hopfield neural network algorithm for computational codeword design. J. Comput. Theoret. Nanosci. **4**, 1257–1262 (2007)
21. Yang, G., Yang, N., Yi, J., Tang, Z.: An improved competitive Hopfield network with inhibitive competitive activation mechanism for maximum clique problem. Neural Comput. **130**, 28–35 (2014)