Science & Education of the Bauman MSTU
Electronic journal
ISSN 1994-0408

Science and Education of the Bauman MSTU, 2015, no. 10, pp. 438–452.

DOI: **10.7463/1015.0814435**

Received:	02.09.2015
Revised:	17.09.2015

Multi-memetic Mind Evolutionary Computation Algorithm for Loosely Coupled Systems of Desktop Computers

M.K. Sakharov[1,*], A.P. Karpenko[1], Ya.I. Velisevich[1]

[*]max.sfn90@gmail.com

[1]Bauman Moscow State Technical University, Moscow, Russia

This paper deals with the development and software implementation of the hybrid multi-memetic algorithm for distributed computing systems. The main algorithm is based on the modification of MEC algorithm proposed by the authors. The multi-memetic algorithm utilizes three various local optimization methods. Software implementation was developed using MPI for Python and tested on a grid network made of twenty desktop computers. Performance of the proposed algorithm and its software implementation was investigated using multi-dimensional multi-modal benchmark functions from CEC'14.

Keywords: global optimization, memetic algorithms, hybrid algorithms, distributed computing, evolutionary computation, mind evolutionary computation

1. Introduction

Solving complex optimization problems in the field of engineering design is quite often coupled with large computational expenses. Nowadays, one of the most promising approaches to increasing the efficiency of methods for dealing with this type of problems are distributed computations. Several desktop computers connected into a parallel computing system are often used as a part of such framework. The advantage of this type of networks is that their capacity can be increased literally with no limit by means of scaling [1, 2].

However, in order to make such calculations possible one needs a special software implementation which would be based on a data exchange interface. In this work MPI was utilized as one of the most widely used interfaces, designed for loosely coupled systems. At the same time, it wouldn't be enough just to increase the amount of computational resources for obtaining high-quality solutions to optimization problems. A development of specialized algorithms which would be based on the features of particular parallel systems is required in order to increase the efficiency of methods.

Nowadays, population algorithms are used most frequently for solving global optimization problems. The main idea of such algorithms is based on the modeling of a collective behavior of self-organizing living and nonliving systems. Population algorithms allow one to process several solutions independently in parallel. Subsequently, the main advantage of this type of algorithms is their decentralization and elegancy of parallelization [3, 4].

Among main parallelization models for population algorithms one can highlight a global model, an island model, a diffusion model and other hybrid models. In this paper an island model was utilized as it takes into account a low carrying capacity of communication network of desktop computers. In the meantime, if one speaks about loosely coupled systems where communication between subpopulations is absent it's appropriate to use a special case of an island model called a model of noncommuting population [5].

In order to increase performance of the specified global optimization methods researchers use either hybridization or meta-optimization. A development of hybrid algorithms implies a combination of various or same methods with different values of free parameters in such a way that advantages of one method would overcome disadvantages of another one. Meta-optimization, on the other hand, implies the adjustment of free parameters' values that would provide the maximum efficiency of an algorithm being investigated.

In the recent times, so called memetic algorithms, MAs became widely popular. They represent hybrid population-based meta-heuristic global optimization algorithms based on the concept of a meme. In this context, a meme represents any local optimization method, which improves a current solution at the particular stages of the main algorithm. Generally, memetic algorithms are hybridization of a population method and one or several local optimization methods [6, 7].

There are many various hybridization methods for optimization algorithms and, in particular, for population-based methods. One of the most widely used classification of such methods is a one-level classification proposed by Wang [3]. In accordance with this classification there are three groups of hybrid algorithms: embedded algorithms, preprocessor/postprocessor algorithms and coalgorithms. The first category can be also divided into two subgroups: high-level and low-level hybridization. In this work a high-level embedded hybridization was utilized; this suggests a weak connection between algorithms being combined.

Memetic algorithms are a very good example of such hybridization scheme, their distinct feature is a significant autonomy of utilized algorithms. A structure of MAs provides researchers with a lot of different opportunities for developing their modifications which could differ from one another, for instance, by the frequency of local search appliance, its termination criteria and other parameters.

Modification of MAs that are most frequently used in practice implies a simultaneous usage of various memes and called multi-memetic algorithms [8]. The goal of this work is development and software implementation of a parallel multi-memetic algorithm for loosely coupled computing systems as well as the investigation of its performance with a use of several benchmark optimization functions.

2 Problem statement

In this paper a multi-dimensional global constrained minimization is considered [3]

$$\min_{X \in D \subseteq R^{|X|}} F(X) = F(X^*) = F^*, \tag{1}$$

where set D is determined with inequality constraints

$$D = \{X |\, x_i^- \leq x_i \leq x_i^+, i \in [1; |X|]\}. \tag{2}$$

Here $F(X)$ – the objective function being minimized and defined in every point of search domain D, $F(X^*) = F^*$ – the desired minimum value of the objective function $F(X)$, X – a vector of variables, X^* – the desired vector of variables, wherein the objective function takes up its minimal value, $|X|$ – the dimension of vector X: $X = (x_1, x_2, \dots, x_{|X|})$.

3 Utilized Algorithms

3.1 Base Algorithm

In this paper Mind Evolutionary Computation, MEC was selected as a base algorithm for the considered hybridization scheme. Its concept was firstly proposed in 1998 [9, 10]. This choice is justified, first of all, by the commitment to loosely coupled computing systems. MEC is capable of providing the minimal number of connections between subpopulations which evolve on the separate computational nodes. Such feature is required for a high efficiency of the method.

MEC simulates some aspects of human behavior in the society; every individual is an intellectual agent which operates within a group of other individuals. In order to achieve a high position within its group, an individual has to study from the most successful individuals in this group. And groups themselves should follow the same principle to stay alive in the intergroup competition [10].

In accordance with the algorithm a multi-population consists of some leading groups $S^b = \left(S_1^b, S_2^b, \dots, S_{|S^b|}^b\right)$ and some lagging groups $S^w = \left(S_1^w, S_2^w, \dots, S_{|S^w|}^w\right)$, which include $|\boldsymbol{S^b}|$ and $|\boldsymbol{S^w}|$ subpopulations correspondingly. A number of individuals in every of the mentioned subpopulations is set to be the same and equals $|S|$.

Each of subpopulations S_i^b, S_j^w has its own communication environment named a local blackboard and denoted as C_i^b, C_j^w correspondingly. A multi-population as a whole $\boldsymbol{S} = \{\boldsymbol{S^b}, \boldsymbol{S^w}\}$ has a common global blackboard C^g.

Canonical MEC is composed of three main stages: initialization of groups, similar-taxis and dissimilation. Operations of similar-taxis and dissimilation are repeated iteratively while the best obtained value of an objective function is changing. When the best obtained values stops changing, the winner of the best group from a set of leading ones is selected as a solution to the optimization problem.

3.2 Hybrid MEC

There are several modifications of the canonical MEC algorithm proposed by different authors, for instance, Extended MEC, Improved MEC, Chaotic MEC and so forth [10]. This paper utilizes a modification proposed by the authors in one of their previous works [1, 2] and named HMEC. The distinct feature of that algorithm is an addition of the decomposition stage and modification of the similar-taxis stage. The choice is made after a certain number of iteration what meme out of a set of available memes is the most suitable for a given search subdomain. This choice is independent for each group [8].

In this particular modification a greedy hyperheuristic was used for determining the best meme in the groups, however other hyperheuristics are applicable as well [8]. The greedy strategy suggests that the best meme is selected at each iteration in accordance with the local improvement it has demonstrated. The value of an objective function after the improvement was taken as a choice criterion.

A general scheme of that algorithm can be described as follows [2].

1. Initialization of groups within the search domain D.
   a. Divide domain D into subdomains $D_1, D_2, \dots, D_n$ by means of decomposing interval $[x_\alpha^{min}; x_\alpha^{max}]$ into η equal subintervals. Here $\alpha \in [1:|X|]$, η – the algorithm's free parameters.
   b. Generate a given number γ of groups $S_{k,i}, i \in [1:\gamma]$ in each subdomain $D_k, k \in [1:\eta]$, where γ – another free parameter of the algorithm;
   $$\cup_{k=1}^{\eta} \cup_{i=1}^{\gamma} S_{k,i} = S.$$
   c. Generate a random vector $X_{k,i,1}$, whose components are distributed uniformly within the corresponding search subdomain in each group $S_{k,i}$. Identify this vector with the individual $s_{k,i,1}$ of the group $S_{k,i}$.
   d. Determine the initial coordinates of the rest of individuals in the group $s_{k,i,j}$, $j \in [2:|S|]$ following the formula
   $$X_{k,i,j} = X_{k,i,1} + N_{|X|}(0, \sigma), \quad (3)$$
   in other words they are placed randomly around the main individual $s_{k,i,1}$ in accordance with $|X|$-dimensional normal distribution law $N_{|X|}(0,\sigma)$, with aero mathematical expectation along all $|X|$ coordinates and standard deviation σ.
   e. Calculate the scores of all individuals in the population S and put them on the corresponding local blackboards.
   f. Create leading $\boldsymbol{S}^b$ and lagging $\boldsymbol{S}^w$ groups on the basis of obtained information.
2. Similar-taxis operation is performed in every group.
   a. Take information on the current best individual $s_{k,i,j^*}, j^* \in [1:|S|]$ of the group $S_{k,i}$ from the blackboard $C_{k,i}$.
   b. Launch a meme chosen for a particular group sequantially from the current positions of each individual except for the main one.

   c. Determine a winner i.e. new main individual from the results of local improvement in every group $s_{k,i,l^*}, l^* \in [1:|S|]$.
   d. Create leading and lagging groups on the basis of obtained information.
   e. Put information on the new winners in every group of the population on the corresponding local and global blackboards.
3. Dissimilation operation.
   a. Read the scores of all groups $f_i^b, f_j^b,\ i \in [1:|S^b|], j \in [1:|S^w|]$ from the global blackboard C^g (scores of the best individuals in the groups).
   b. Compare those scores. If a score of any leading group S_i^b appeared to be less than a score of any lagging group S_j^w, than the latter becomes a leading group and the first group becomes a lagging one. If a score of a lagging group S_k^w is lower than scores of all leading groups, then it's removed from the population.
   c. Using the initialization operation and formula (3) each removed group is replaced with a new one.
   d. Evaluate the termination criteria. If either a number of stagnation iterations λ_{stag} or maximum allowed number of iterations λ_{stop} are exceeded then the iterative process should be stopped, otherwise it continues and goes to point 2.

3.3 Local search methods

At the stage of local optimization HMEC chooses the best meme (in accordance with selected hyperheuristic) for each search subdomain from a set of local unconstrained optimization algorithms, which in this work consists of the methods of Nelder–Mead, Hooke-Jeeves and Monte-Carlo [11, 12].

The first method belongs to the class of zero-order deterministic optimization methods, i.e. based only on the objective function's values. The advantage of this method is that the shape of a deformable polyhedron conforms to the topography of the objective function due to the expansion and reduction operations. The disadvantage of the Nelder-Mead method is in possible degeneration ("flattering") of simplex for strongly ravine functions.

The Hooke-Jeeves method also belongs to the class of zero-order deterministic optimization methods. The advantage of this method is its relative simplicity of implementation and hybridization with global search methods. The disadvantage of this method is that it can't ensure the convergence to a minimum point in case of highly elongated, curved or sharp-edge contour lines of the objective functions.

The Monte-Carlo method represents the class of zero-order stochastic optimization methods.

4 Software Implementation and Testing

A loosely coupled computational system that consists of 20 desktop computers running on Windows 8 OS (8 GB RAM, Intel Core i5 processor with a CPU clock 3.20 GHz) was used for implementation and performance investigation of HMEC algorithm. Python programming language and Python compiler v. 2.7.3 were chosen as development tools. Data exchange interface MPI with Python implementation was also selected to support the distributed computing.

HMEC algorithm was implemented as an executable program and a set of modules. It supports a “multi-start” mode and takes the following input data: the number of iterations; the dimension of the search domain; the number of leading and lagging groups; the number of individuals in groups; boundaries of the search domain; the objective function.

The program returns best obtained function value and the vector of variables that provided that best value of the objective function.

Based on the data obtained from the calculations in the “multi-start” mode the probability estimation of global minimum localization and the average number of trials for each benchmarks function were determined.

Proposed algorithm and its program implementation were tested using a spherical function that is continuous, convex and unimodal throughout its definition domain. This function has the global minimum in the point $x_i = 0$ and $F(X) = 0$, $i = [1{:}\,|X|]$.

All tests were carried out under the same conditions: the number of iteration – 1000, the number of individuals in groups – 20, the number of leading groups – 10, the number of lagging groups – 10. The following results were received during the tests: best-obtained value for a canonical MEC algorithm is 3.5379e-05, for the Nelder-Mead method – 1.3372e-07, for the Hooke-Jeeves method – 1.3567e-05, for the Monte-Carlo method – 5.5689e-06. All tests were performed in the “multi-start” mode.

According to the results, we can conclude that all implemented algorithms ensure the convergence of the objective function to its global minimum for a given search domain with the specified values of the free parameters and coefficients.

5 Performance Investigation of HMEC

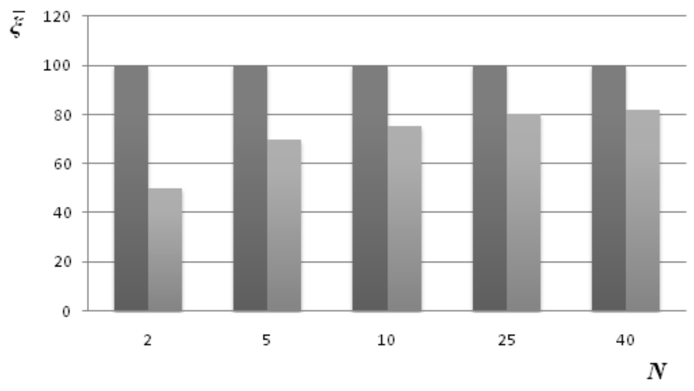
It's assumed that the iterative process went into a state of stagnation if there are no changes with a given tolerance $e = 10^{-6}$ in the best obtained value of the objective function for 30 consecutive iterations.

A speed of convergence was defined as a number of iterations before a state of stagnation takes place. The average speed of convergence $\bar{t}$ – the average number of iterations in the “multi-start: mode before an iterative computational process goes into a state of stagnation. The average number of wins for every meme was denoted as $\bar{S}$. The corresponding average number of trials was denoted as $\bar{N}$. Also for evaluating the efficiency of the software implementation a probability estimate of global minimum localization $\bar{\xi}$ was used.

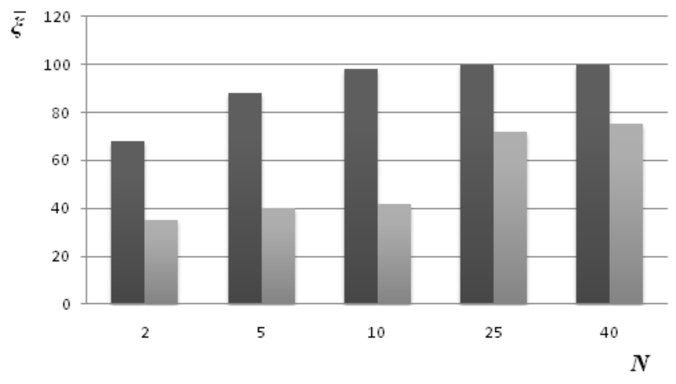
For all benchmark optimization functions a search domain is constrained as follows $[-100.0, 100.0]$; that domain was divided into a number of subgroups that is equal to the number of utilized CPUs N. A distributed computational network, which consists of 2, 5, 10, 25 and 40 CPU cores, was used for carrying out the experiments. Due to the large computational expenses the dimension of vector X was limited to $|X| = \{2, 10\}$.

For performance investigation of the software implementation the following benchmark multi-modal optimization functions were utilized [13]: shifted and rotated Rosenbrock function; shifted and rotated Weierstrass function; shifted and rotated Griewank function; shifted Schwefel function.

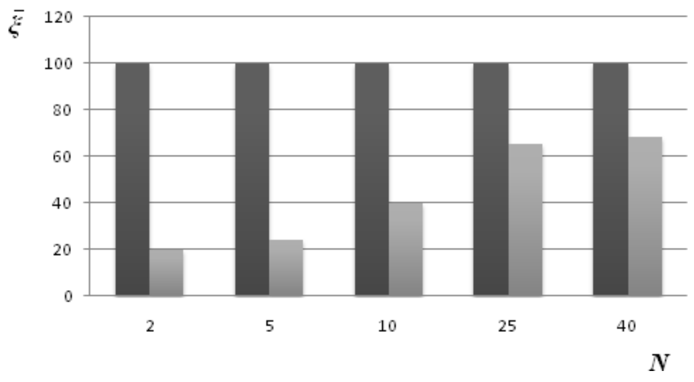
The dependency of an estimation of the probability of global minimum localization on the number of CPUs is presented on the figure 1. It follows from an analysis of these data that increase in the number of processors and subsequently the number of decomposition areas leads to a significant growth of the localization probability.



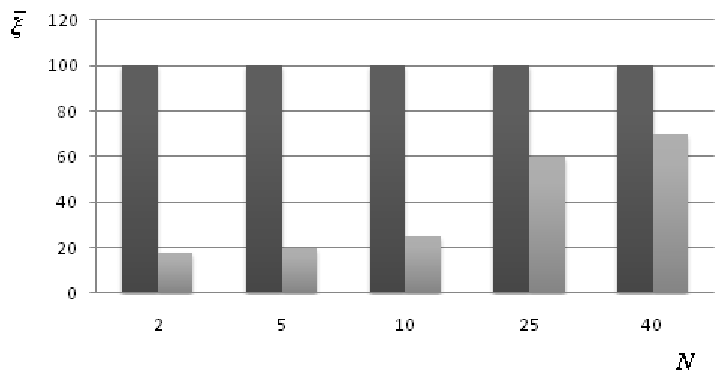
Rosenbrock function



Weierstrass function



Griewank function

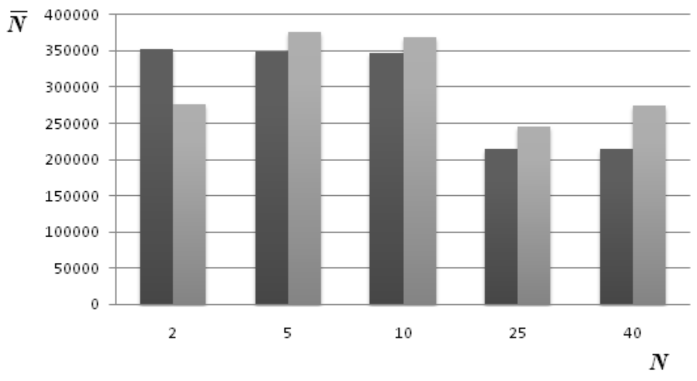


Schwefel function

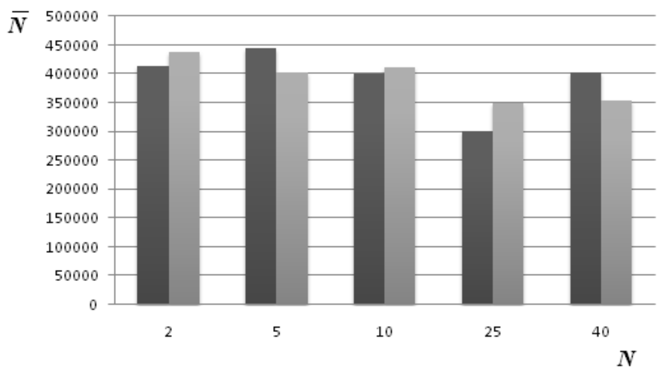
Fig. 1. – Estimation of the probability of global minimum localization $\bar{\xi}$ depending on the number of CPUs N for parallel HMEC: ■ – $|X| = 2$, ▨ – $|X| = 10$

Increase in the dimension of benchmark functions leads to the decrease in quality of obtained solution, but the parallel algorithm with initial decomposition of the search domain provides better chances of finding a high-quality solution even to complex multi-dimensional functions.

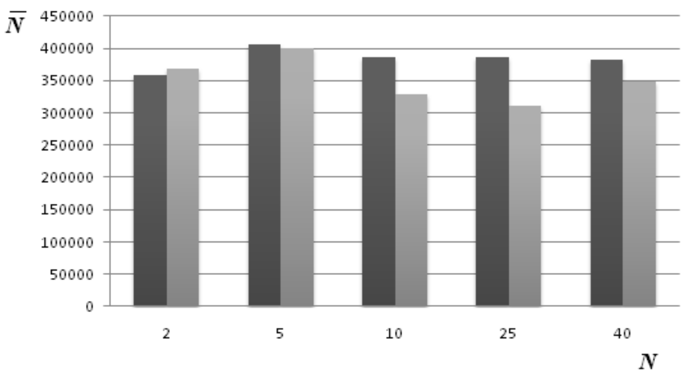
The dependency of the averaged maximum number of trials on the number of CPUs is presented on the figure 2. From these diagrams it's clear that on average the dimension or the number of CPUs have no explicit influence on the maximum number of trials.



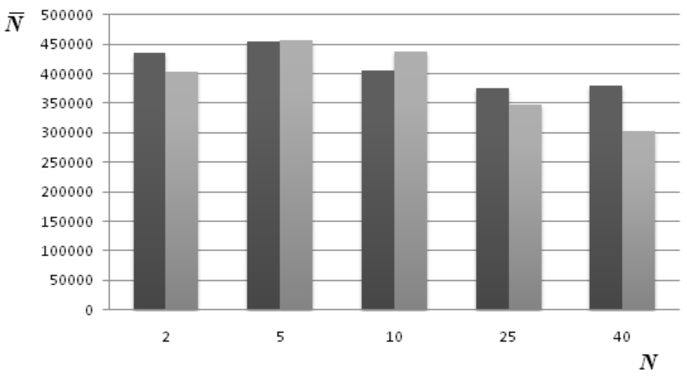
Rosenbrock function



Weierstrass function



Griewank function

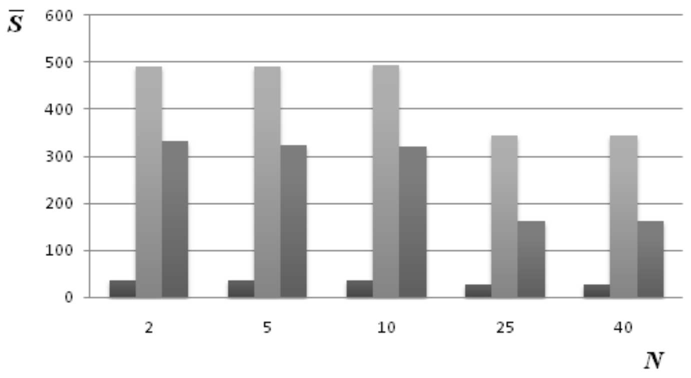


Schwefel function

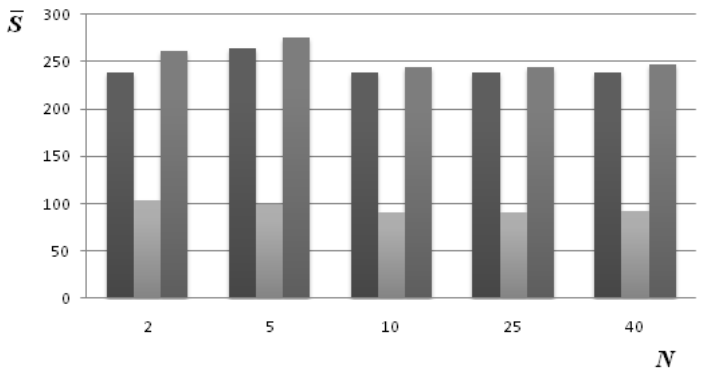
Fig. 2. – Average number of trials depending on the number of CPUs N for a parallel implementation of HMEC: ■ – $|X| = 2$, ▨ – $|X| = 10$

Obtained results are caused by the fact that a success of minimization and a number of trials are connected to the initial values of the components of vector X in the search domain.

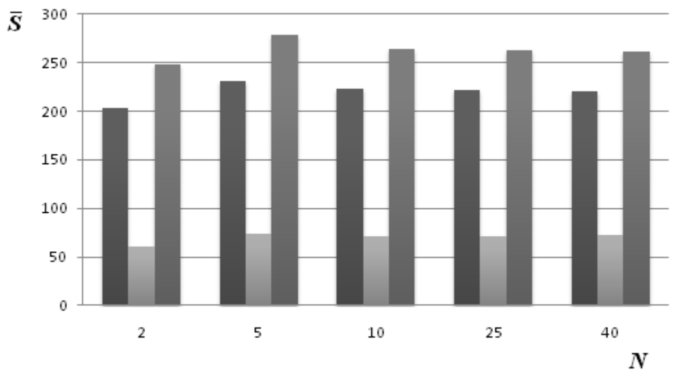
On the figure 3 the dependency of the average number of wins S for every meme with a limit of 50 iterations on the number of CPUs for each benchmark function with $|X| = 2$ is presented. It's seen from received results that successful memes don't depend on the number of parallel process and subsequently the number of decomposition domains. It's important to notice that for every function there is a more successful meme that is selected more frequently.



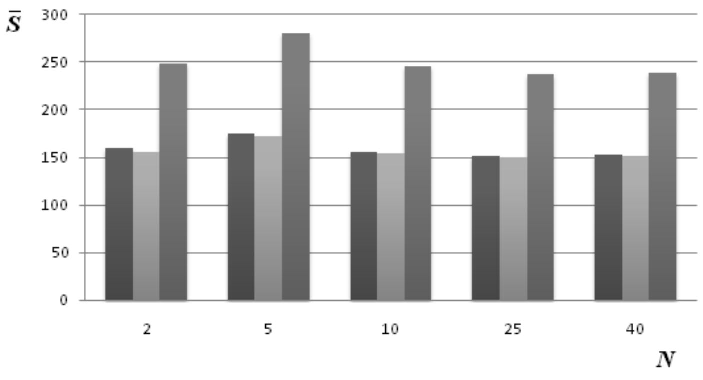
Rosenbrock function



Weierstrass function



Griewank function



Schwefel function

Fig. 3. – Average number of victories for every meme depending on the number of CPUs N: ■ – Nelder-Mead, ■ – Hooke-Jeeves, ■ – Monte-Carlo

The Hooke-Jeeves method appeared to be the most suitable for Rosenbrock function, however for the rest of the functions the Monte-Carlo method was able to perform best local improvements. It’s also worth noticing that for Weierstrass and Griewank functions the Nelder-Mead method was almost equivalent to the Monte-Carlo method. It may be concluded from obtained results that HMEC can successfully adapt to various complex functions and take their special features into account.

Further expansion of a set of utilized meme would make this method even more adaptive and, hence, suitable for various functions and real-world problems. The average number of iterations $\bar{t}$ depends on the dimension of the objective function and increases along with the latter.

It's seen from analysis of obtained data that a parallel implementation of the hybrid multi-memetic algorithm allows one to increase not only the quality of a solution but also the probability of global minimum localization and avoid the premature convergence in the neighborhood of some local optimum.

6 Conclusion

A new multi-memetic algorithm for solving global optimization problems named HMEC was proposed in this paper. The distinct feature of this algorithm consists in using a set of different memes, which allows the algorithm to adapt to various objective functions. It also includes an operation of decomposing the search domain in order to avoid the premature convergence.

Software parallel implementation of that algorithm was developed using data exchange interface MPI to make distributed computation possible on a loosely coupled system made of desktop computers.

Extensive performance investigation of the algorithm was carried out with a use of modified benchmark optimization functions. The efficiency of HMEC was estimated by the probability of global optimum localization, the average number of iterations and the average number of trials. The number of victories for each meme during local competitions in the "multi-start" mode was also measured and analyzed. Results showed the parallel implementation of HMEC is more capable of finding a high-quality minimum of an objective function (in terms of both probability and accuracy) then the sequential one. Modifications of the hybrid algorithm as well as the whole multi-memetic approach proved to be promising and are worth of further investigation.

This work was supported by the Ministry of Education and Science of the Russian Federation (2014-14-579-0144)

References

1. Karpenko A., Posypkin M., Rubtsov A., Sakharov M. Multi-memetic Global Optimization based on the Mind Evolutionary Computation. *Proceedings of the IV International Conference on Optimization Methods and Application "Optimization and Applications" (OPTIMA-2013)*. Moscow, Dorodnicyn Computing Centre of RAS, 2013, pp. 83-84.
2. Karpenko A.P. *Sovremennye algoritmy poiskovoi optimizatsii. Algoritmy, vdokhnovlennye prirodoi* [Modern algorithms of search engine optimization. Nature-inspired optimization algorithms]. Moscow, Bauman MSTU Publ., 2014. 446 p. (in Russian).
3. Karpenko A.P., Sakharov M.K. Multi-Memes Global Optimization Based on the Algorithm of Mind Evolutionary Computation. *Informacionnye Tehnologii = Information Technologies*, 2014, no. 7, pp. 23-30. (in Russian).

4. Weise T. *Global Optimization Algorithms. Theory and Application*. University of Kassel, 2008. 758 p.
5. Talbi E. A Taxonomy of Hybrid Metaheuristics. *Journal of Heuristics*, 2002, vol. 8, iss. 5, pp. 541-564. DOI: 10.1023/A:1016540724870
6. Dawkins R. *The Selfish Gene*. Oxford University Press, 1976. 384 p.
7. Nguyen Q.H., Ong Y.S., Krasnogor N. A Study on the Design Issues of Memetic Algorithm. *IEEE Congress on Evolutionary Computation (CEC 2007).* IEEE Publ., 2007, pp. 2390-2397. DOI: 10.1109/CEC.2007.4424770
8. Ong Y.S., Lim M.H., Zhu N., Wong K.W. Classification of adaptive memetic algorithms: A comparative study. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2006, vol. 36, iss. 1, pp. 141-152. DOI: 10.1109/TSMCB.2005.856143
9. Chengyi S., Yan S., Wanzhen W. A Survey of MEC: 1998-2001. *2002 IEEE International Conference on Systems, Man and Cybernetics. Vol. 6*. IEEE Publ., 2002, pp. 445-453. DOI: 10.1109/ICSMC.2002.1175629
10. Jie J., Zeng J. Improved Mind Evolutionary Computation for Optimizations. *Proceedings of 5th World Congress on Intelligent Control and Automation. Vol. 3*. IEEE Publ., 2004, pp. 2200-2204. DOI: 10.1109/WCICA.2004.1341978
11. Floudas A.A., Pardalos P.M., Adjiman C., Esposito W.R., Gümüs Z.H., Harding S.T., Klepeis J.L., Meyer C.A., Schweiger C.A. *Handbook of Test Problems in Local and Global Optimization*. Kluwer, Dordrecht, 1999. 441 p.
12. Nelder J.A., Meade R. A Simplex Method for Function Minimization. *Computer Journal*, 1965, vol. 7, iss. 4, pp. 308-313. DOI: 10.1093/comjnl/7.4.308
13. Liang J.J., Qu B.Y., Suganthan P.N. *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*. Technical Report 201311. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China; Technical Report. Nanyang Technological University, Singapore, 2013. 32 p.

Наука и Образование
МГТУ им. Н.Э. Баумана
Сетевое научное издание
ISSN 1994-0408

Наука и Образование. МГТУ им. Н.Э. Баумана.
Электрон. журн. 2015. № 10. С. 438–452.

DOI: **10.7463/1015.0814435**

Представлена в редакцию: 02.09.2015
Исправлена: 17.09.2015

УДК 519.6

Мультимемеевый алгоритм эволюции разума для слабосвязанных систем на основе персональных компьютеров

Сахаров М. К.[1,*],
профессор, д.ф.-м.н. Карпенко А. П.[1],
Велисевич Я. И.[1]

[*] max.sfn90@gmail.com

[1] МГТУ им. Н.Э. Баумана, Москва, Россия

Ключевые слова: глобальная оптимизация, меметические алгоритмы, гибридные алгоритмы, распределенные вычисления, эволюционные вычисления, алгоритм эволюции разума

Решение современных задач оптимизации проектных решений, зачастую, сопряжено с большими затратами вычислительных ресурсов. В настоящее время одним из наиболее перспективных методов повышения эффективности работы алгоритмов для решения подобных задач являются распределенные параллельные вычисления.

В тоже время, простого увеличения вычислительных мощностей недостаточно для решения практически значимых задач. Необходима разработка специализированных алгоритмов, ориентированных на конкретную параллельную систему, что позволило бы повысить эффективность метода, благодаря использованию особенностей архитектуры вычислительной системы.

В данной работе предложен параллельный гибридный алгоритм эволюции разума *HMEC*, который относится к классу меметических алгоритмов глобальной оптимизации. Отличительной особенностью алгоритма является использование ряда различных мемов, позволяющих алгоритму адаптироваться к разного рода целевым функциям. Под мемом понимают какой-либо алгоритм локальной оптимизации, уточняющий текущее решение исходной задачи оптимизации на определенных шагах выполнения основного алгоритма. В широком смысле, меметические алгоритмы представляют собой гибридизацию одного из популяционных алгоритмов глобального поиска и одного или нескольких алгоритмов локальной оптимизации.

Алгоритм ориентирован на использование в слабосвязанных вычислительных системах, состоящих из персональных компьютеров.

В рамках работы выполнена программная реализация данного алгоритма с использованием декомпозиции области поиска на стадии первичной инициализации популяции. В качестве инструментов реализации использовались язык программирования *Python* и интерфейс обмена сообщениями *MPI*, как один из наиболее распространенных и ориентированных на слабосвязанные системы интерфейсов.

Широкое исследование эффективности алгоритма проведено на наборе тестовых модифицированных функций Розенброка, Вейерштрасса, Griванка и Швефеля. Эффективность алгоритма оценена с помощью таких показателей, как оценка вероятности локализации глобального экстремума, среднее число итераций и среднее число испытаний. Также определено и проанализировано число побед мемов в ходе локальных состязаний в процессе мультистарта для каждой тестовой функции. Результаты исследований показывают, что параллельный меметический алгоритм *HMEC* позволяет во многих случаях с высокой вероятностью и точностью локализовать глобальный экстремум для заданной целевой функции чем последовательная версия алгоритма.

Результаты исследования показывают перспективность дальнейших модификаций и развития параллельных мультимемеевых алгоритмов глобальной оптимизации.

Список литературы

1. Karpenko A., Posypkin M., Rubtsov A., Sakharov M. Multi-memetic Global Optimization based on the Mind Evolutionary Computation // Proceedings of the IV International Conference on Optimization Methods and Application “Optimization and Applications” (OPTIMA-2013). Moscow: Dorodnicyn Computing Centre of RAS, 2013. P. 83-84.
2. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. 446.
3. Карпенко А.П., Сахаров М.К. Мультимемеевая глобальная оптимизация на основе алгоритма эволюции разума // Информационные технологии. 2014. № 7. С. 23-30.
4. Weise T. Global Optimization Algorithms. Theory and Application. University of Kassel, 2008. 758 p.
5. Talbi E. A Taxonomy of Hybrid Metaheuristics // Journal of Heuristics. 2002. Vol. 8, iss. 5. P. 541-564. DOI: 10.1023/A:1016540724870
6. Dawkins R. The Selfish Gene. Oxford University Press, 1976. 384 p.
7. Nguyen Q.H., Ong Y.S., Krasnogor N. A Study on the Design Issues of Memetic Algorithm // IEEE Congress on Evolutionary Computation (CEC 2007). IEEE Publ., 2007. P. 2390-2397. DOI: 10.1109/CEC.2007.4424770
8. Ong Y.S., Lim M.H., Zhu N., Wong K.W. Classification of adaptive memetic algorithms: A comparative study // IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. 2006. Vol. 36, iss. 1. P. 141-152. DOI: 10.1109/TSMCB.2005.856143
9. Chengyi S., Yan S., Wanzhen W. A Survey of MEC: 1998-2001 // 2002 IEEE International Conference on Systems, Man and Cybernetics. Vol. 6. IEEE Publ., 2002. P. 445-453. DOI: 10.1109/ICSMC.2002.1175629

10. Jie J., Zeng J. Improved Mind Evolutionary Computation for Optimizations // Proceedings of 5th World Congress on Intelligent Control and Automation. Vol. 3. IEEE Publ., 2004. P. 2200-2204. DOI: 10.1109/WCICA.2004.1341978
11. Floudas A.A., Pardalos P.M., Adjiman C., Esposito W.R., Gümüs Z.H., Harding S.T., Klepeis J.L., Meyer C.A., Schweiger C.A. Handbook of Test Problems in Local and Global Optimization. Kluwer, Dordrecht, 1999. 441 p.
12. Nelder J.A., Meade R. A Simplex Method for Function Minimization // Computer Journal. 1965. Vol. 7, iss. 4. P. 308-313. DOI: 10.1093/comjnl/7.4.308
13. Liang J.J., Qu B.Y., Suganthan P.N. Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical Report 201311. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China; Technical Report. Nanyang Technological University, Singapore, 2013. 32 p.