Bibliography

[1] Sergienko I.V., Kaspshitskaya M.F. Models and methods solving combinatorial optimization problems on computers. Kyiv, Naukova Dumka, 1981. 287 p. (In Russian)

[2] Stoyan Yu.G., Emets O.O. Theory and methods of Euclidean combinatorial optimization. Kyiv, Institute of System Research in Education, 1993.–188 p. (In Ukraine)

[3] Lebedeva T.T., Semenova N.V., Sergieko T.I. Conditions for optimality and solvability in the problems of linear vector optimization with a convex feasible set. In: Reports of the National Academy of Sciences of Ukraine, 2003, №10, p. 80-85 (In Ukrainian).

[4] Emets O.O., Kolechkina L.M. Combinatorial optimization problems with linear fractional objective functions. Kyiv, Naukova Dumka, 2005. 118p. (In Ukrainian)

[5] C. Malivert, N. Boissard. Structure of efficient sets for strictly quasiconvex objectives. J. of Convex Analysis, 1994, vol. 1, 2, p.143-150.

[6] Lebedeva T.T., Semenova N.V., Sergieko T.I. Stability of vector integer optimization problems: Connection with stability of optimal and nonoptimal solution sets. Kibernetika i Sistemny Analiz, 2005, №4, p. 90-100 (In Russian)

[7] Podinovsky V.V., Nogin V.D. Pareto-optimal solution to multiobjective problems. – M.: Nauka, 1982. 256p. (In Rassian)

[8] T.R. Gulati, M.A. Islam. Efficiency in Linear Fractional Vector Maximization Problems with Nonlinear Constrains // Optimization. – 1989. Vol. 20. - №4. – P.477-482.

Information about authors

Natalia V. Semenova – V.M. Glushkov Institute of Cybernetics, National Academy of Science of Ukraine, Candidate of Physical & Mathematical Sciences, Senior Scientific Researcher, 40, Prospect Akademika Glushkova, 03187, Kyiv, Ukraine; e-mail: nvsemenova@meta.ua

Lyudmyla M. Kolechkina – V.M. Glushkov Institute of Cybernetics, National Academy of Science of Ukraine, Candidate of Physical & Mathematical Sciences, Doctoral Student, 40, Prospect Akademika Glushkova, 03187, Kyiv, Ukraine; e-mail: ludapl@ukr.net

Alla M. Nagirna – Ukrainian State University of Finances and Foreign Trade, Senior Teacher, Higher Mathematics and Informatics Department, 3, Chigorin Str., Kyiv; e-mail: vpn2006@rambler.ru

ONE CLASS OF STOCHASTIC LOCAL SEARCH ALGORITHMS

Leonid Hulianytskyi, Alexander Turchin

Abstract: Accelerated probabilistic modeling algorithms, presenting stochastic local search (SLS) technique, are considered. General algorithm scheme and specific combinatorial optimization method, using "golden section" rule (GS-method), are given. Convergence rates using Markov chains are received. An overview of current combinatorial optimization techniques is presented.

Keywords: combinatorial optimization, stochastic local search, simulated annealing, Markov chains

Introduction

Approximate algorithms are well-known to solve different combinatorial problems. This based on some facts: firstly, mostly each problem is NP-hard; secondly, result functions have lots of local extremes; and finally, real data often are given with some inaccuracy and this makes serious calculations unnecessary. Also, it must be noticed that key ideas of these algorithms (metaheuristics) allow us to design algorithms, which can solve not one but some classes of optimization problems.

The most known definition of combinatorial optimization problem (according to Papadimitriou and Steiglitz [1,2]) is following: find $x_* \in X$

$$x_* = arg \min_{x \in D \subseteq X} f(x), \tag{1}$$

where *X* – finite (or possibly countably infinite) solution space, *D* – subspace defined by problem constraints, *f: X→R¹* – objective function.

This definition works perfectly on finite sets, but in case of infinite sets some classification problems nay appear – classification according to structure of solution space may be difficult. To get over these complications, Berge [3] introduced *combinatorial configuration*. Let *m,n* – natural numbers, *U*={1, … , *m*}, *V*={v₁, … , *vₙ* } – some sets, and some order is given for *V*: v₁ <… < *vₙ* (in other words, *V* is a chain).

Definition 1. *Combinatorial configuration* is a reflection $\varphi : U \to V$, satisfying some contingencies Λ.

From this definition, if m,n – some fixed numbers, the number of combinatorial configurations is finite. Usage of contingencies Λ allows describing different combinatorial configurations.

Berge's definition can be generalized as follows: Let *Y*={1, … , *m*}, *Z* – some discrete (may be finite) space called "forming space", φ – homomorphism, $\varphi : Y \to Z$, satisfying some contingencies system Ω. It must be noted, that discrete space is the space of isolated points.

Definition 2. *Combinatorial object* is $\kappa = (\varphi, \tilde{X}, \Omega)$, where $\tilde{X}$ - basic space.

Definition 3. *1st order combinatorial object* is such combinatorial object, where basic space is the forming space:

$$\kappa = (\varphi, X_{(1)}, \Omega),$$

Here $X_{(1)} \equiv Z$. It is easy to see that in case of finite Z these combinatorial objects are the same with Berge's combinatorial configurations [3].

Definition 4. *k-order combinatorial object* is such combinatorial object that:

$$\kappa = (\varphi, X_{(k)}, \Omega),$$

where $X_{(k)} \subseteq X_{(k-1)} \cup X^k$.

After everything stated above, another definition of optimization problem (1) can be stated as follows:

Definition 5. Optimization problem (1) will be *combinatorial optimization problem (COP)*, if solution space *X* is the space of combinatorial objects.

Also, it must be noted that received results are valid for the case of discrete countable X.

Simulated Annealing (SA) method is well-known effective search technique solving COP [2]. SA algorithms allow finding high accuracy results and can be easily realized on multi-processor systems.

Transaction probability in SA scheme depends on result function changes and realized with parameter *T* (temperature). Probabilistic transactions (*temperature schedule*) in neighborhoods of worse solutions should decrease with the number of iterations and should tent to zero. Changes of parameter *T* are stated with *equilibrium conditions*. Such probabilistic mechanism is a key feature of SA scheme and allows avoiding local extremes during first iterations and concentrates on global extremes in time.

At the same time, different researches showed great dependence of received solutions from values of parameters. This fact leads to necessity of probabilistic mechanism generalization. As a result, a new class of *accelerated probabilistic modeling algorithms* (also called *G-algorithms*) was introduced [4], and among them – algorithm with specific probabilistic transaction scheme [5].

G-algorithms

The scheme of accelerated probabilistic modeling algorithms concludes one key idea: algorithm build next element from current solution neighborhood, and if result function of this element is better – we accept it as updated current solution, if result function is worse – element can be accepted with some probability. But, unlike

SA, in *G*-algorithm these “thresholds” are calculated in similar way during entire computational process, but algorithm changes the parameter that defines elimination of worse solution.

To build *G*-algorithm, we should define $\{\mu_t\}, 0 \le \mu_0 < \mu_1 < \ldots \le 1$ – some strictly monotonous sequence of real numbers, that can be describe as similarity of temperature schedule in SA. If x^h is a current solution on iteration *h*, and $L(x)$ is a neighborhood of $x \in X$, then next element $y \in L(x^h)$ with *f(y)* > *f(x^h)* (in case of minimization problem) can be accepted as x^{h+1} with some probability $p(x^h,y)$, that depends on current value of μ_t.

Let *F(x,y),* 0≤ *F*(*x,y*) ≤1– some functional dependent on result function. Transaction probability from *x* to *y* (*x,y*∈*X*) can be defined in the next way:

$$p \equiv p(x,y) = (1-\mu_t)\cdot F(x,y)$$

The general scheme of G-algorithms presented on Picture 1.

```
      procedure  G_Search(x)
      begin
             $x^0$ := some initial solution from X;
             $\mu_0$ : = 0;  h := 0;  t : = 0;
             $x_{rec}$:= $x^0$;  $f_{rec}$ := $f(x^0)$;
             while neighborhood of current solution $L(x^h)$ is not checked totally do
                 begin
                     while equilibrium condition is not met  do
                        begin
                             y := GenerateNextNeighbor $L(x^h)$;
                             Calculate $F(x^h,y)$ ;
                             p := (1- $\mu_t$ ) $F(x^h,y)$;
                             $\xi$  := random[0,1];
                             if  p ≥ $\xi$  then
      h:=h+1;  $x^h$ := y;
      if  $f_{rec}$ > $f(x^h)$  then
          $x_{rec}$ := $x^h$;  $f_{rec}$ := $f(x^h)$
      end if
                                          end if;
                        end;
                       CalculateNextValue $\mu_{t+1}$;
                       t := t + 1;
                  end;
                return   x= $x_{rec}$;
        end
```

Picture 1. Accelerated probabilistic modeling algorithm (*G*-algorithm)

Here *random*[0,1] – random generator of values from [0,1].

To build *G*-algorithm, these elements must be defined:

- functional *F*(*x,y*);
- *{μ_t}* building mechanism;
- equilibrium conditions;
- stopping rule.

Functional F(*x,y*) must use monotonous by result function consequences, that met such conditions:

a) *F*(*x,y*)→1, if *f(y)*→*f(x)*.

b) *F*(*x,y*)→0, if *f(y)*→∞.

For example, next functional can be used as *F* (*x,y*):

$$F(x,y)=\begin{cases}\left[\dfrac{f(x)}{f(y)}\right]^{\beta}, if \ \ f(y)\geq f(x),\\ 1, \qquad otherwise.\end{cases}$$

where β > 0 – some real, or natural number particularly.

Usually, it is easy to find the upper bound of result function f_{max} in terms of some COP, because of finite *X*: $f(x)\leq f_{\max}$, $\forall x\in X$. In this case functional may be as follows:

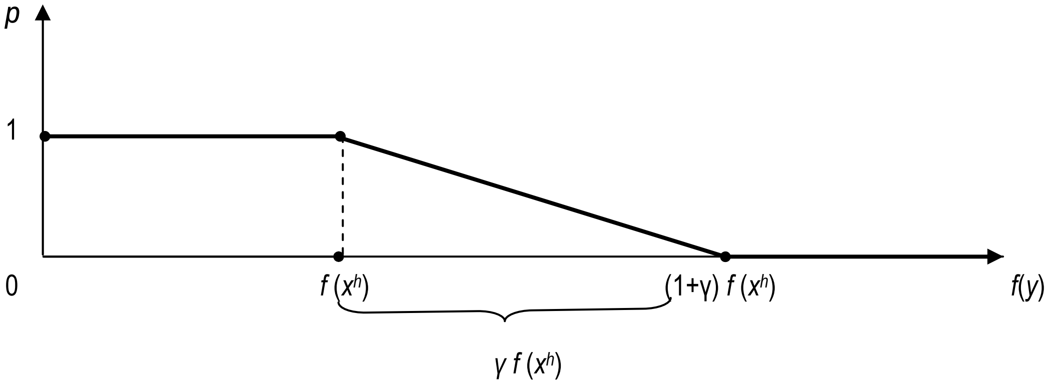
$$F(x,y)=1-\left[\frac{f(y)-f(x)}{f_{\max}-f(x)}\right]^{\beta}.$$

In general case, common G-algorithm uses next piecewise linear functional [5]:

$$F(x,y)=\begin{cases}\min\{1,1-\dfrac{f(y)-f(x)}{\gamma f(x)}\}, \ if \ \ f(y)\geq f(x),\\ 0, \ if \ \ f(y)< f(x),\end{cases}$$

where parameter $\gamma,\gamma>0$ is a real number.

In this case, for current solution x^h transaction probability $p(x^h,y)$ from x^h to $y\in L(x^h)$ represented with piecewise linear functional, that consists from three main parts: it is 1 if $f(y) < f(x^h)$, it decreases from 1 to 0 if $f(x^h) \leq f(y) \leq (1+ \gamma) f(x^h)$, and it is 0 when $f(y) > (1+ \gamma) f(x^h)$. This can be illustrated in next way (Picture 2).



Picture 2. Transaction probability in the neighborhood

Thus, parameter γ defines the middle interval $\gamma \cdot f(x^h)$, and if *f(y)* gets into this interval, *y* may be chosen as updated solution x^{h+1}. This means that in case of essentially worse f(y) the next neighborhood element *y* will be far away from acceptance. Also, it should be noted that the length of interval decreases if $f(x^h)$ gets closer to global optimum, and increases otherwise.

Moreover, usage of relative values allows avoiding the dependence from absolute values. Thereby, computational process adapts to dynamics of result function changes.

There are two main approaches to build consequence $\{\mu_t\}$: using some strictly monotonous function called "*G*-function" (that is why algorithms are called *G*-glgorithms [4,5]), and using "golden section" rule [6].

Let *G*: [0,1] → [0,1] – some strictly monotonous function, then $\mu_{t+1} = G(\mu_t)$. In common cases, G-functions are like this example:

$$G_k(x)=\min\{1,(x^{1/k}+H)^k\}, \ \ x\geq 0. \tag{2}$$

where $k \in\{1, 2, 3\}$, a *H*, 0 < *H* < 1 – some small value defining the convergence rate $G_k(x)\to 1$ if $x\to 1$.

Otherwise, values μ_t may be calculated through division of [0,1] interval with some step and sequential selection of constructed points. Apparently, when consequence reaches 1, proposed algorithm will work like common determinate local search algorithm.

Usage of "golden section" rule became an effective approach to build algorithms for COP with 1-argument continuous result function. In this case μ_t is the left "golden section" point of [μ_{t-1},1]; μ_0 =0, and μ_{t+1}=*G(u*$_t$*)*. Thus, "golden section" rule defines the speed of left border approaching to 1, and corresponding μ_t are the arguments for *G(x)* (for example (2)). Algorithm using "golden section" rule is called GS-algorithm.

Setting up the equilibrium condition may be effective with usage of SA experience, and analogy of "temperature" in SA and μ, in G-algorithms. Particularly, possible equilibrium condition can be defined as follows: let ν is some natural number and $\varepsilon > 0$ is real; and realization of ν transactions is called a "run" [7]. If, for current temperature algorithm made *k* runs and some consequence $f_1,...,f_k$ is received, equilibrium condition is met if

$$| f_{k+1} - f_i | \le \varepsilon$$

for some $i \in \{1,..., k\}$. In this case f_i may be the average result function value or the best one, received on current run.

Algorithm stops when stopping rule is met. This can be: total check of neighborhood with no transactions done, algorithm computational time limitations, reaching some denoted solution accuracy (in case of known fesult function lower bound) etc. Also, another possible stopping rule may base on comparison of min and max result function values – and min and max result function values with some value of μ: if this correspondence tends to 1, algorithm stops.

In case of some contingencies computational scheme of G-algorithms can easy take these contingencies into account when next point of neighborhood is generated. This approach allows solving some COP classes, and its contingencies can be changed even during algorithm work.

Similarly to Iterated Local Search (ILS) [2], after finishing general GS-algorithm work the received solution can be modified and used as initial solution for built-in GA-algorithm. In other words, we get some metaheuristic method that can be called "Iterated GS-algorithm".

Convergence analysis

Convergence analysis will be made on essential class of COP – permutation tasks with transposition metric. Let $Gr = (V, H)$ will be the complete weighted graph with $V = \{1,...,n\}$ - set of nods and *H* - set of ribs. The path between two point from solution space *X* means the consequence of ribs: each nod relevant to $x \in X$, will be connected with nods relevant to $L(x)$ neighborhood elements. Let's consider neighborhoods of minimal radius 1: $L(x) = L_1(x)$. The correspondence between neighborhood $L(x)$ and set $N(v), v \in V$ will be built upon next rule: if two point in space *X* differs with one transposition, their nods-images on graph Gr will be connected with a rib.

The search process in this case will be imitated with transactions on graph Gr. The degree *d* of graph is $d = \frac{n(n-1)}{2}$, and diameter $D = n - 1$.

Search process in this case is being modeling with Markov chain because next state depends only on current state. GS-algorithm convergence if corresponding Markov chain will have at least one value corresponding to global optimum.

Transaction probability is limited above with value μ^s. Let $\hat{d} = \left(\frac{1}{d}\right)^{-D}, \hat{\mu} = \prod_{i=1}^{D} \mu^i$.

Lemma. Let $v \in V$ - some nod corresponding to some *x* from solution space. Then expected number of steps to reach the nod, corresponding to the global optimum, not more than $\frac{\hat{d}}{\hat{\mu}}$.

Proof. Let $\underline{x}$ – global optimum of COP (1), and let nod $\underline{v}$ in graph $Gr = (V, H)$ corresponds to $\underline{x}$. There is a path from v to $\underline{v}$ with length $q < D$. In other words, there is a consequence of nods $v_1, v_2, ...v_q$.

If each element of neighborhood can be selected for checking with the same probability, transaction probability of visiting $\underline{v}$ in *q* steps will be at least $\left(\frac{1}{d}\right)^q \times \prod_{i=1}^{q} \mu^i$, and will be valid

$$\left(\frac{1}{d}\right)^q \times \prod_{i=1}^{q} \mu^i \ge \left(\frac{1}{d}\right)^D \prod_{i=1}^{D} \mu^i = \frac{\hat{\mu}}{\hat{d}}.$$

Therefore, probability of visiting $\underline{v}$ starting from some v will be at least $\frac{\hat{\mu}}{\hat{d}}$. This means that expected number of algorithm steps to reach global optimum will be not more than $\frac{\hat{d}}{\hat{\mu}}$.

Lemma is proved.

Theorem. GS-algorithm convergence to the global optimum with probability more than $(1 - \frac{1}{C^k})$ during the number of steps not more than $Ck \cdot \frac{\hat{d}}{\hat{\mu}}$ and this estimation does not depends on initial solution ($C = const > 1$).

Proof. Mathematical induction approach will be used. Let $Q = C\left[\frac{\hat{\mu}}{\hat{d}}\right]$, and let's prove that probability of missing nod $\underline{v}$ in kQ steps is not more than $1/C^k$.

For basic case $k = 1$ and some initial state *x*, according to the Lemma the expected number of steps to reach $\underline{v}$ is not more than $\frac{\hat{d}}{\hat{\mu}}$. From Markov's inequality

$$p(\frac{\hat{d}}{\hat{\mu}} > Q) = \frac{M(\frac{\hat{d}}{\hat{\mu}})}{Q} = \frac{\frac{\hat{d}}{\hat{\mu}}}{C\frac{\hat{d}}{\hat{\mu}}} = \frac{1}{C}.$$

Let's estimate that theorem is valid for all $k \le K - 1$ and prove it for $k = K$. Let $x_Q, x_{2Q}, ...x_{(K-1)Q}$ - Markov chain built on steps $Q, 2Q, ..., (K-1)Q$ correspondently. Consider two events:

- Event H1: nod $\underline{v}$ is not reached during first Q steps;
- Event H2: nod $\underline{v}$ is not reached during next $(K-1)Q$ steps;

According to these events, probability to miss $\underline{v}$ during KM steps is $p = p(H2 \mid H1) \times P(H1)$. From Markov chain theory, the probability to reach global optimum during some number of steps depends on this number of steps and does not depends on states visited previously. Using this fact, probability $P(H2 \mid H1)$ depends on state on iteration Q and $(K-1)Q$ steps:

$$P = P(H1) \sum_{i \in V} P(H2 \mid x_Q = i) \times P(x_Q = i).$$

Probability $P(H1)$ is not more than $\frac{1}{C}$ according to the basic case, probability $P(H2 \mid x_Q = i)$ is not more than $\frac{1}{C^{K-1}}$ for each $i \in V$ according to the estimation. Finally:

$$P \le \frac{1}{C} \times \frac{1}{C^{K-1}} = \frac{1}{C^K}.$$

Probability to reach global optimum is opposite to miss it. Therefore, with probability more than $(1 - \frac{1}{C^k})$ will be reached during the number of steps not more than $Ck \cdot \frac{\hat{d}}{\hat{\mu}}$, and this estimation does not depends on initial solution.

Theorem is proved.

Corollary. If $k \to \infty$, there is probability convergence of received solutions consequence to the global optimum. This fact is obvious according to the probability convergence definition.

The key idea of received convergence rates is in generalization of rates received previously for SA [8].

Also, received convergence rates can be used in convergence analysis of other accelerated probabilistic modeling algorithms and other similar search techniques.

Practical application

Proposed combinatorial optimization algorithms can be applied to solve different classes of problems because of its general requests for problem formulation. The complex analysis of algorithms solving popular COP (like TSP and QAP), alongside with specific problems (like location problem, warehousing problem etc) was made.

TSP was the very first application of *G*-algorithms because this problem is a well-known proving ground for new algorithms construction. Paper [9] presents some received results, and these results congruent with best known solutions received on supercomputing systems. Also, some essential experience in combining algorithms construction (based on genetic algorithm and *G*-algorithm hybridization) was accumulated.

QAP (Quadratic Assignment Problem) is also very popular in different branches like economics, production planning etc. Despite of the fact that this problems is well-known and been solved during years, it is hard to find the exact solution for problems with dimension *n* > 15-20, and even ε-close solution finding is NP-hard problem [1]. Performed results allowed choosing most effective calculation schemes, whose were used to build different algorithms like *GS*-algorithm (using “golden section” rule) and metaheuristics used for TSP and QAP solving [5,9].

In paper [10] some specific problem of optimal cutting was presented. This is a problem of optimal bar layout on semi-infinite tape, when bars have same height and different width. Some algorithms based on local optimization schemes, SA and *G*-algorithms were proposed. Received results proved that *G*-algorithm performed better results comparing with other algorithms.

The warehousing problem is a very important strategic problem in case of different types of products and forecasts about possible needs [11]. Presented *G*-algorithm was compared with three LS and SA algorithms. Received results showed that *G*-algorithm can find high accuracy solutions in less time than other rivals.

Problem of optimal network channel capacity is one of the most important problems in telecommunication network construction with ATM technology. In this case network consists of commutators connected with fixed length network channels. Some requirements about information traffic are given for each pair of commutators, and general information flows are given for each channel. Network channel capacity for some channel depends on basic channel capacity. Problem is to find such number of basic channels that the cost of network will be the smallest and some requirements about service quality will be met. Some algorithms presenting Local Search, Iterated Local Search, Simulated Annealing, *G*-algorithms and Genetic Algorithms were developed. Practical application showed that for different traffic types best solutions in most cases were found be *G*-algorithm and ILS [12].

Conclusion

Some SLS algorithms are presented in this paper, alongside with own modification called *GS-Algorithm*, using “golden section” rule. The combination of probabilistic calculation mechanisms with LS procedure allowed successfully solve different classes of combinatorial optimization problems. An upper bound of global optimum convergence rate is received with usage of Markov chain theory.

Some examples of algorithm applications for some COP classes solving are given. According to received results, proposed algorithm can be used for applied problems solving allow and used as built-in procedure in metaheuristics (first of all, population-based metaheuristics [2]).

Bibliography

1. *Papadimitriu C.H., Steigliz K.* Combinatorial Optimization: Algorithms and Complexiti. – Engiewood Cliffs: Pretice-Hall, 1982.
2. *Hoos H.H., Stützle T.* Stochastic Local Search: Foundations and Applications. – San Francisco: Morgan Kaufmann Publ., 2005.
3. *Berge C.* Principes de combinatoire. – Paris: Dunod, 1968.
4. *Гуляницкий Л.Ф.* Модифицированные алгоритмы вероятностного моделирования в комбинаторной оптимизации //Технология и методы решения задач прикладной математики. – Киев: Ин-т кибернетики им.В.М.Глушкова АН Украины, 1991. – С. 10–14.
5. *Гуляницкий Л.Ф.* Решение задач комбинаторной оптимизации алгоритмами ускоренного вероятностного моделирования // Компьютерная математика. – Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 2004. – №1. – С. 64–72.
6. *Гуляницький Л.Ф., Турчин О.Я.* Использование правила "золотого сечения" в алгоритмах вероятностного моделирования // Журн. вычисл. и прикладной математики. – 1998. – № 1(83). – С.26–29 (укр.).
7. *Golden B.L., Skiscim C.C.* Using simulated annealing to solve routing and location problem // Naval Research Logistics Quarterly. – 1986. – 33, № 6. – P. 273–277.
8. *Rajasekaran S.* Simulated annealing and nested annealing // J.of Global Optimization.– 2000.–16. –P. 43-56.
9. *Hulianitsky L., Turchin A.* On approach to combining genetic algorithms and accelerated probabilistic modeling algorithms /The Experience of Designing and Application of CAD Systems in Microelectronics: Proc. of the VII Int. Conf. CADMS 2003 (18-22 February 2003, Lviv-Slavske, Ukraine). –Lviv: Publ. House of Lviv Polytechnic National University, 2003. – P. 218–219.
10. *Гуляницкий Л.Ф., Гобов Д.А.* О сравнении эффективности алгоритмов решения одного класса задач размещения прямоугольников на полубесконечной ленте //Компьютерная математика. – 2003. – №1. – С. 88-97.
11. *Гуляницкий Л.Ф.* О решении на ПЭВМ одной задачи о складировании продукции // Теория и программная реализация методов дискретной оптимизации. -Киев: Ин-т кибернетики им.В.М.Глушкова АН УССР, 1989. – С.20-25.
12. *Hulianytskyi L., Baklan A.* Optimization of ATM telecommunication Networks // Int. J. "Information theories & applications". – 2005. – 12, N 4. – P. 328–335.

Authors’ Information

***Leonid F. Hulianytskyi** – V.M.Glushkov Institute of Cybernetics of NAS of Ukraine, Glushkova ave. 40, Kyiv, 03680, Ukraine; e-mail: lh_dar@hotmail.com*

***Alexander Y. Turchin** - V.M.Glushkov Institute of Cybernetics of NAS of Ukraine, Glushkova ave. 40, Kyiv, 03680, Ukraine; e-mail: turchin@ua.fm*