©2003г. П.А. Борисовский,
А.В. Еремеев, канд. физ.–мат. наук
(Омский филиал Института математики СО РАН)

О сравнении некоторых эволюционных алгоритмов

В работе проводится теоретическое сравнение одного из наиболее простых эволюционных алгоритмов с другими методами этого класса в ситуации, когда оператор перехода к новым решениям удовлетворяет, так называемому, условию монотонности. Показано, что при условии монотонности рассматриваемый алгоритм является оптимальным в классе эволюционных алгоритмов.

1 Введение

Эволюционные алгоритмы, к числу которых относятся генетические алгоритмы [1, 2], эволюционные стратегии [3], алгоритмы генетического программирования [4] и др., получают все более широкое применение в решении задач оптимизации. Характерной особенностью этих алгоритмов является имитация процесса эволюции биологической популяции, где особи соответствуют пробным точкам в пространстве решений задачи оптимизации, а приспособленность особей к условиям окружающей среды определяется значениями целевой функции в этих точках. Как правило, процесс поиска оптимального (или близкого к нему) решения направляется исключительно полученными значениями целевой функции в предыдущих пробных точках пространства решений. При этом никак не используются предположения о таких свойствах целевой функции и ограничений, как выпуклость, дифференцируемость, выполнение соотношения Беллмана и т.д.

Принципы наследственности, изменчивости и отбора, сформулированные Ч.Дарвином [5], в эволюционных алгоритмах реализуются в построении новых решений-потомков посредством случайных операторов, модифицирующих полученные ранее пробные точки. Отбор таких пробных точек для модификации производится с учетом значений целевой функции в них (чем выше приспособленность – тем больше шансы точки на отбор в качестве "родительской" особи).

Ввиду простоты адаптации вычислительных схем эволюционных алгоритмов к различным задачам, эти методы активно применяются как в дискретной, так и в непрерывной оптимизации. Имеется большое число экспериментальных работ, свидетельствующих о применимости эволюционных алгоритмов при решении задач распознавания образов, предсказания, проектирования сложных систем, систем управления и т.д. В то же время, теоретические исследования не позволяют пока полностью объяснить причины успешного (или безуспешного в некоторых случаях) использования эволюционных алгоритмов. Обзоры теоретических результатов, полученных в этом направлении, могут быть найдены, например, в [6, 7, 8].

Цель настоящей работы состоит в обосновании некоторых интуитивно ясных свойств эволюционных алгоритмов, которые, тем не менее, долгое время не имели строгой математической формулировки. Основной полученный здесь результат позволяет выделить класс задач, на которых один из наиболее простых эволюционных алгоритмов, обозначаемый далее через (1+1)-EA, является оптимальной стратегией поиска. Алгоритм (1+1)-EA является простейшим вариантом эволюционного алгоритма, где популяция состоит из одной особи и может рассматриваться как (1+1)-эволюционная стратегия [3], локальный алгоритм случайного поиска с пересчетом при неудачном шаге [9], или как частный случай алгоритма имитации отжига с нулевой температурой (см., например, [10]). Данный результат представляет собой обобщение полученного ранее в [11], где предполагалась дискретность множества значений целевой функции.

2 Постановка задачи

Пусть задача оптимизации состоит в максимизации (или минимизации) целевой функции $f : X \to \mathbf{R}$, где X – *множество допустимых решений* и $\mathbf{R}$ – множество действительных чисел. Далее в этой работе будем полагать, что целевая функция ограничена, и рассматривается задача максимизации (на задачи минимизации все последующие рассуждения переносятся очевидным образом вычитанием целевой функции из достаточно большой константы).

Опишем схему рассматриваемого здесь алгоритма (1+1)-EA. Предполагается, что начальное решение $x^{(0)}$ задано. Для построения нового решения к текущему $x^{(t)}$ применяем вероятностную процедуру *мутации* $x' = \mathcal{M}(x^{(t)})$, результат которой случайным образом распределен на множестве X, и полагаем $x^{(t+1)} = x'$, если $f(x^{(t)}) < f(x')$, иначе $x^{(t+1)} = x^{(t)}$. Алгоритм заканчивает работу при достижении заданного количества итераций или по какому-либо другому критерию остановки.

Оператор мутации здесь понимается, как вероятностный алгоритм (см., например, [12]), который принимает на вход родительское решение x и случайным образом производит потомка $\mathcal{M}(x) \in X$. При этом вероятностное распределение потомка зависит от входного решения и данных задачи (можно считать, что данные задачи также подаются на вход оператора мутации).

Также нас будет интересовать обобщенный вариант мутации с произвольным числом родителей и потомков – оператор *воспроизведения*, определение которого будет дано в следующем разделе.

Одним из наиболее часто используемых операторов мутации в случае $X = \{0,1\}^n$ является, так называемая, *стандартная мутация* [1], которая применяется к бинарным строкам, и под ее действием каждый бит строки независимо от других меняет свое значение с фиксированной вероятностью $p_{\mathcal{M}}$. Другой пример – оператор *точечной мутации*, в котором с равномерным распределением выбирается номер i и i-я компонента решения заменяется новым значением (см., например, [13]). В общем случае, в зависимости от свойств решаемой задачи, оператор мутации (воспроизведения) может включать в себя некоторые дополнительные процедуры, например "исправление" недопустимого решения, или корректировку методом локальной оптимизации (см., например, [2, 14]). Следующее определение описывает ситуацию, когда выбор более приспособленной родительской особи гарантирует неубывание шансов получения потомков с достаточно высокой приспособленностью.

Определение 1. *Будем называть оператор $\mathcal{M}$ монотонным, если для любых особей x, y таких, что $f(x) \le f(y)$, для всех $\phi \in \mathbf{R}$ выполняется:*

$$P\{f(\mathcal{M}(x)) \ge \phi\} \le P\{f(\mathcal{M}(y)) \ge \phi\}.$$

Рассмотрим простейший пример монотонного оператора мутации. Пусть задача состоит в максимизации числа единиц в булевой строке без каких-либо ограничений, т.е. $f(x) = x_1 + ... + x_n$ и $X = \{0,1\}^n$. Тогда стандартный оператор мутации при $p_{\mathcal{M}} \le 1/2$ является монотонным [15]. Другой пример монотонности имеем в задаче максимизации функции $e^{-||x||}$, $x \in X = \mathbf{R}^n$, если результат на выходе оператора мутации $\mathcal{M}(x)$ распределен нормально с математическим ожиданием x и матрицей ковариаций $\mathbf{P} = \sigma\mathbf{I}$, где $\sigma \in \mathbf{R}_+$ и $\mathbf{I}$ - единичная матрица.

В случае монотонного оператора мутации, в частности, достигаются полученные ранее оценки для доли решений заданного качества в популяции генетического алгоритма, основанного на операторах мутации и селекции [15]. Также, в равенства обращаются оценки [11] математического ожидания времени достижения оптимума и вероятности получения решений требуемого качества после заданного числа итераций в (1+1)-EA.

3 Сравнение алгоритма (1+1)-EA с другими эволюционными алгоритмами

Определим *оператор воспроизведения* $\mathcal{R}(x^1, ..., x^r)$ как вероятностный алгоритм, который получает на вход r родительских решений и случайным образом строит s решений-потомков, распределение которых зависит от входных решений и данных задачи.

Рассмотрим общую схему эволюционного алгоритма A, основанного на операторе $\mathcal{R}$. Дано множество начальных решений $a^{(0,1)}, ..., a^{(0,N)}$, и на каждой итерации t набор решений $a^{(t,1)}, ..., a^{(t,s)}$ строится применением $\mathcal{R}(x^1, ..., x^r)$, где каждое $x^k \in X, k = 1, ..., r$ найдено на каком-то из предыдущих шагов. Иначе говоря $x^k \in \mathcal{A}^{(t-1)}$, где $\mathcal{A}^{(t-1)} = \{a^{(0,l)} : l = 1..., N\} \cup \{a^{(\tau,j)} : \tau = 1, ..., t-1, j = 1, ..., s\}$. Легко показать, что большинство эволюционных алгоритмов, таких как генетические алгоритмы [1], методы генетического программирования [4], эволюционные стратегии (μ, λ)-ES и $(\mu + \lambda)$-ES [3], алгоритмы, предложенные в [16], удовлетворяют этой схеме.

Рассмотрим обобщение понятия монотонности на случай оператора воспроизведения. Пусть случайное решение $\mathcal{R}_{max}(x^1, ..., x^r)$ представляет собой лучший из s потомков, полученных применением $\mathcal{R}$ к набору $x^1, ..., x^r$.

Определение 2. *Оператор $\mathcal{R}$ называется монотонным, если для любых наборов решений $x^1, ..., x^r$ и $y^1, ..., y^r$ таких, что*

$$f(x^1) \le f(y^1), ..., f(x^r) \le f(y^r), \tag{3.1}$$

следующее условие выполняется для всех $\phi \in \mathbf{R}$:

$$P\left\{f(\mathcal{R}_{max}(x^1, ..., x^r)) \ge \phi\right\} \le P\left\{f(\mathcal{R}_{max}(y^1, ..., y^r)) \ge \phi\right\}. \tag{3.2}$$

Заметим, что из этого определения следует, что если все условия (3.1) выполнены как равенства, то распределения случайных величин $f(\mathcal{R}_{max}(x^1, ..., x^r))$ и $f(\mathcal{R}_{max}(y^1, ..., y^r))$ должны совпадать. Очевидно, что монотонный оператор мутации – частный случай монотонного оператора воспроизведения при $r = s = 1$.

Снова рассмотрим для примера простейшую задачу максимизации числа единиц в булевой строке длины n. Определим оператор воспроизведения $\mathcal{R}(x, y)$ следующим образом. Сначала выполняется случайная перестановка элементов векторов x и y, а именно, генерируются векторы $x' = (x_{\sigma(1)}, ..., x_{\sigma(n)})$ и $y' = (y_{\hat\sigma(1)}, ..., y_{\hat\sigma(n)})$, где перестановки σ и $\hat\sigma$ выбираются независимо с равномерным распределением. Затем два потомка x'' и y'' строятся применением, так называемого, *равномерного кроссинговера* (см., например, [2]), при котором для каждого i с вероятностью $1/2$ полагаем $x''_i := x'_i$, $y''_i := y'_i$, иначе полагаем $x''_i := y'_i$, $y''_i := x'_i$. Нетрудно показать, что такой оператор воспроизведения является монотонным.

Условие монотонности может быть ослаблено следующим образом: назовем $\mathcal{R}$ *слабо монотонным*, если неравенство (3.2) выполняется хотя бы для всех $\phi > \max\{f(y^k) : k = 1, ..., r\}$.

Для примера рассмотрим семейство задач о наименьшем покрытии множества специальной структуры, предложенное Балашем [17]. В общем виде задача о наименьшем покрытии множества формулируется следующим образом: дано множество $M = \{1, ..., m\}$ и набор его подмножеств $M_j \subseteq M$, $j \in N = \{1, \dots, n\}$. Подмножество индексов $J \subseteq N$ называется *покрытием*, если $\mathbf{U}_{j \in J} M_j = M$. Требуется найти покрытие наименьшей мощности.

Введем обозначение $N_i = \{j \in N : i \in M_j\}$, $i = 1, ..., m$. В семействе задач Балаша $B(n, p)$ предполагается, что $m = C_n^{p-1}$ и множество $\{N_1, N_2, ..., N_m\}$ состоит из всех $n - p + 1$-элементных подмножеств N. Тогда $J \subseteq N$ – оптимально, если и только если $|J| = p$.

Пусть решения представлены бинарными строками, т.е. $x \in \{0, 1\}^n$ соответствует множеству $C(x) = \{j \in N : x_j = 1\} \subseteq N$. Если $C(x)$ оказывается покрытием, то полагаем $f(x) = n - |C(x)|$, иначе $f(x) = 0$ (т.о. переходим к задаче максимизации). Нетрудно показать, что точечный оператор мутации в этом случае будет слабо монотонным [11].

Оператору $\mathcal{R}$ сопоставим оператор мутации $\mathcal{M}_\mathcal{R}(x)$, действующий следующим образом: сначала оператор $\mathcal{R}$ применяется к набору, состоящему из r копий решения x, и в качестве результата выдается $x' = \mathcal{R}_{max}(x, ..., x)$, если $f(x') \geq f(x)$. Иначе полагаем $\mathcal{M}_\mathcal{R}(x) = x$. Очевидно, что если $\mathcal{R}$ является монотонным то и $\mathcal{M}_\mathcal{R}$ тоже. Для монотонного оператора $\mathcal{R}$ введем функцию $\Phi_\mathcal{R}(\psi, z) = P\{f(\mathcal{M}_\mathcal{R}(x)) \geq \psi\}$, где x - некоторое решение такое, что $f(x) = z$. Благодаря монотонности $\mathcal{M}_\mathcal{R}$ эта функция не зависит от выбора конкретного решения x (см. замечание после определения 2) и является неубывающей по z. В общем случае $\Phi_\mathcal{R}(\psi, z)$ может не являться непрерывной,[1] далее будем предполагать, что при каждом ψ число точек разрыва $\Phi_\mathcal{R}(\psi, z)$ как функции от z не более чем конечно. Для тех значений z, при которых не существует такого x, что $f(x) = z$, функция $\Phi_\mathcal{R}(\psi, z)$ доопределяется так, чтобы она оставалась монотонной по z при каждом ψ, например, $\Phi_\mathcal{R}(\psi, z) = \inf\{\Phi_\mathcal{R}(\psi, z') \mid z' > z, \exists x : f(x) = z'\}$, и $\Phi_\mathcal{R}(\psi, z) = 1$ при $z \geq \sup_X f$.

Обозначим через $\tilde{a}^{(t)}$ лучшее решение из множества $\mathcal{A}^{(t)}$ (также напомним, что через $x^{(t)}$ обозначается текущее решение на итерации t алгоритма (1+1)-EA). Сформулируем основной результат данной работы:

Теорема 1. *Пусть монотонный оператор $\mathcal{R}$ используется в алгоритме A, соответствующий ему оператор $\mathcal{M}_\mathcal{R}$ применяется в алгоритме (1+1)-EA, и $\Phi_\mathcal{R}(\psi, z)$ при каждом ψ имеет конечное число точек разрыва как функция от z. Если в (1+1)-EA за начальное решение принято $x^{(0)} \in argmax\{f(a^{(0,1)}), ..., f(a^{(0,N)})\}$, то для всех $t \geq 0$ и всех $\phi \in \mathbf{R}$*

$$P\{f(\tilde{a}^{(t)}) \geq \phi\} \leq P\{f(x^{(t)}) \geq \phi\}. \tag{3.3}$$

Легко показать, что $\mathcal{M}_\mathcal{R}$ остается монотонным, даже если $\mathcal{R}$ – слабо монотонный, откуда вытекает следующее утверждение:

Следствие 1. *Условие монотонности $\mathcal{R}$ в теореме 1 может быть заменено условием слабой монотонности.*

4 Заключение

Вопрос о том какой из алгоритмов предпочтительнее для некоторого класса задач, является одним из ключевых в теории случайного поиска. В данной работе рассмотрен класс эволюционных алгоритмов в ситуации, когда оператор воспроизведения обладает свойством монотонности. Основной результат показывает, что в этих условиях алгоритм (1+1)-EA является наилучшей стратегией поиска.

[1] Рассмотрим тривиальный пример максимизации функции $f(x) = x$ на отрезке $[0, 1]$. Пусть $\mathcal{R}(x) = 1/2$, при $x < 1/2$, и $\mathcal{R}(x) = 1$, при $x \geq 1/2$. Тогда функция $\Phi_\mathcal{R}(1, z)$ является ступенчатой с разрывом в точке $z = 1/2$.

Установлено, что свойство монотонности имеет место при решении ряда модельных задач. Заметим, что хотя эти задачи не представляют самостоятельной практической ценности, тем не менее, некоторые из них используются в теоретическом анализе алгоритмов многими авторами (см., например, [11, 13, 15, 17]).

При решении практических задач, точное выполнение условия монотонности представляется маловероятным, однако, в ряде случав операторы воспроизведения можно считать монотонными лишь с небольшим отклонением. Результат такого рода был получен в работе [18], где рассматривалось одно семейство задач о наибольшем независимом множестве графа. Представленный там же вычислительный эксперимент показал убедительное преимущество алгоритма (1+1)-EA над стандартным генетическим алгоритмом, в условиях, когда свойство монотонности нарушено незначительно. В связи с этим, можно предположить, что дальнейшим развитием данного исследования будет формулировка и изучение понятия монотонности в приближенном смысле.

Лемма 1. *Пусть оператор $\mathcal{M}_{\mathcal{R}}$ – монотонный, и $\Phi_{\mathcal{R}}(\psi, z)$ при каждом ψ имеет конечное число точек разрыва как функция от z. Если x и y – случайные решения такие, что для всех $\phi \in \mathbf{R}$ выполнено $P\{f(x) \ge \phi\} \le P\{f(y) \ge \phi\}$, тогда для всех $\psi \in \mathbf{R}$*

$$P\{f(\mathcal{M}_{\mathcal{R}}(x)) \ge \psi\} \le P\{f(\mathcal{M}_{\mathcal{R}}(y)) \ge \psi\}.$$

Доказательство леммы 1.
Для случайного решения x обозначим через $F_x(z) = P\{f(x) < z\}$ функцию распределения величины $f(x)$, аналогично введем $F_y(z)$. Зафиксируем произвольное ψ. Пусть $a = \inf_X f$ и $b = \sup_X f + 1$. По формуле полной вероятности получаем:

$$P\{f(\mathcal{M}_{\mathcal{R}}(x)) \ge \psi\} = \int_{[a,b)} P\{f(\mathcal{M}_{\mathcal{R}}(x)) \ge \psi | f(x) = z\} dF_x(z) = \int_{[a,b)} \Phi_{\mathcal{R}}(\psi, z) dF_x(z),$$

где интеграл понимается в смысле Лебега-Стилтьеса. Разобьем множество $[a, b)$ семейством Π полуинтервалов $[a_i, b_i), i = 1, ..., k$ так, что $a_1 = a, b_k = b, b_i = a_{i+1},\ i = 1, ..., k-1$, и чтобы каждая точка разрыва z' функции $\Phi_{\mathcal{R}}(\psi, z)$ являлась концом одного из полуинтервалов, т.е. существует такое i, что $z' = a_i$ (предполагается, что k достаточно велико). Определим ступенчатую функцию $G_\Pi(z)$ равную $\Phi_{\mathcal{R}}(\psi, a_i)$, если $z \in [a_i, b_i)$. Пусть $a_i \to b_i, i = 1, ..., k$ при $k \to \infty$. Очевидно, $G_\Pi(z)$ монотонна, и $G_\Pi(z) \uparrow \Phi_{\mathcal{R}}(\psi, z)$ при $k \to \infty$. Тогда

$$\int_{[a,b)} G_\Pi(z) dF_x(z) = \sum_{i=1}^{n} G_\Pi(a_i)(F_x(b_i) - F_x(a_i)) =$$

$$G_\Pi(a_n)F_x(b_n) - G_\Pi(a_1)F_x(a_1) - \sum_{i=2}^{n}(G_\Pi(a_i) - G_\Pi(a_{i-1}))F_x(a_i).$$

Ввиду того, что $G_\Pi(a_{i+1}) - G_\Pi(a_i) \geq 0$ и $F_x(a_i) \geq F_y(a_i)$, получаем:

$$\int_{[a,b)} G_\Pi(z) dF_x(z) \leq \int_{[a,b)} G_\Pi(z) dF_y(z),$$

откуда утверждение леммы следует в силу теоремы Лебега о предельном переходе под знаком интеграла.

Доказательство теоремы 1.
Для $t = 0$ неравенство (3.3) очевидно. По индукции по t предположим, что $P\{f(\tilde{a}^{(t-1)}) \geq \psi\} \leq P\{f(\tilde{x}^{(t-1)}) \geq \psi\}$ для всех ψ. Пусть на итерации t алгоритма A оператор $\mathcal{R}$ применяется к набору $x^1, ..., x^r$ и $a^{(t)} = \mathcal{R}_{max}(x^1, ..., x^r)$. Тогда $\tilde{a}^{(t)} \in argmax\{f(a^{(t)}), f(\tilde{a}^{(t-1)})\}$, откуда следует, что

$$P\{f(\tilde{a}^{(t)}) \geq \phi\} = P\{f(\tilde{a}^{(t-1)}) \geq \phi\} + P\{f(a^{(t)}) \geq \phi, f(\tilde{a}^{(t-1)}) < \phi\}. \qquad (\Pi.1)$$

Второе слагаемое может быть представлено в виде:

$$P\{f(a^{(t)}) \geq \phi, f(\tilde{a}^{(t-1)}) < \phi\} = \int_{[\inf_X f, \phi)} P\{f(a^{(t)}) \geq \phi | f(\tilde{a}^{(t-1)}) = z\} dF_{\tilde{a}^{(t-1)}}(z). \quad (\Pi.2)$$

Применим оператор $\mathcal{M}_\mathcal{R}$ к решению $\tilde{a}^{(t-1)}$ и вычислим аналогичным образом вероятность $P\{f(\mathcal{M}_\mathcal{R}(\tilde{a}^{(t-1)})) \geq \phi)\}$.

$$P\{f(\mathcal{M}_\mathcal{R}(\tilde{a}^{(t-1)})) \geq \phi)\} = \qquad (\Pi.3)$$

$$= P\{f(\tilde{a}^{(t-1)}) \geq \phi\} + \int_{[\inf_X f, \phi)} P\{f(\mathcal{M}_\mathcal{R}(\tilde{a}^{(t-1)})) \geq \phi | f(\tilde{a}^{(t-1)}) = z\} dF_{\tilde{a}^{(t-1)}}(z).$$

Далее, заметим, что из монотонности $\mathcal{R}$ и определения $\mathcal{M}_\mathcal{R}$ следует неравенство

$$P\{f(a^{(t)}) \geq \phi | f(\tilde{a}^{(t-1)}) = z\} \leq P\{f(\mathcal{M}_\mathcal{R}(\tilde{a}^{(t-1)})) \geq \phi | f(\tilde{a}^{(t-1)}) = z\},$$

подставив которое в (П.1) и (П.3) с использованием (П.2), получаем:

$$P\{f(\tilde{a}^{(t)}) \geq \phi\} \leq P\{f(\mathcal{M}_\mathcal{R}(\tilde{a}^{(t-1)})) \geq \phi)\}.$$

Используя предположение индукции и утверждение леммы 1, заключаем, что

$$P\{f(\mathcal{M}_\mathcal{R}(\tilde{a}^{(t-1)})) \geq \phi)\} \leq P\{f(\mathcal{M}_\mathcal{R}(x^{(t-1)})) \geq \phi)\}.$$

СПИСОК ЛИТЕРАТУРЫ

1. *Goldberg D.E.* Genetic Algorithms in Search, Optimization and Machine Learning. Reading: Addison Wesley. 1989.

2. *Reeves C.R.* Genetic Algorithms for the Operations Researcher // INFORMS Journal on Computing. 1997. V. 9. No. 3. P. 231-250.

3. *Rechenberg I.* Evolutionsstrategie: Optimerung Technischer Systeme nach Prinzipen der Biologischen Evolution. Stuttgart: Formann-Holzboog Verlag. 1973.

4. *Koza J.R.* Genetic Programming II : Automatic Discovery of Reusable Programs (Complex Adaptive Systems). MIT Press. 1994.

5. *Darvin Ch.* On the Origin of Spicies. London: Cloves and Sons. 1860.

6. *H.-G. Beyer.* The Theory of Evolution Strategies. Natural computing series. Springer, Heidelberg. 2001.

7. *Reeves C.R., Rowe J.E.* Genetic Algorithms: Principles and Perspectives. Norwell, MA: Kluwer. 2002.

8. *Vose M.D.* The Simple Genetic Algorithm: Foundations and Theory. Cambridge, MA: MIT Press. 1999.

9. *Растригин Л.А.* Статистические методы поиска. М.: Наука. 1968.

10. *Johnson D.S., Aragon C.R., McGeoch L.A., Schevon C.* Optimization by Simulated Annealing: an Experimental Evaluation; part I, graph partitioning // Oper. Res. 1989. V. 37. No. 6. P. 865–892.

11. *Borisovsky P.A., Eremeev A.V.* A Study on Performance of the (1+1)-Evolutionary Algorithm // In K. De Jong, R. Poli, and J. Rowe, eds. Foundations of Genetic Algorithms 7. San Francisco: Morgan Kaufmann. 2003. P. 271–287.

12. *Motwani R., Raghavan P.* Randomized Algorithms. Cambridge University Press. 1995.

13. *Rudolph G.* Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon // Fundamenta Informaticae. 1998. V. 35. No. 1-4. P. 67–89.

14. *Еремеев А.В.* Генетический алгоритм для задачи о покрытии // Дискрет. анализ и исслед. операций. Сер. 2. 2000. Т. 7. No 1. С. 47-60.

15. *Eremeev A.V.* Modeling and Analysis of Genetic Algorithm with Tournament Selection // In C. Fonlupt et al eds. Proc. of Artificial Evolution Conference (AE'99), Dunkerque. Springer Verlag. 2000. LNCS, V. 1829. P. 84–95.

16. *Aldous D., Vazirani U.U.*"Go with the Winners" Algorithms // Proc. IEEE Symposium on Foundations of Computer Science. 1994. P. 492–501.

17. *Balas E.* A Sharp Bound on the Ratio Between Optimal Integer and Fractional Covers // Math. Oper. Res. 1984. V. 9. No. 1. P. 1-5.

18. *Borisovsky P.A., Zavolovskaya M.S.* Experimental Comparison of Two Evolutionary Algorithms for Independent Set Problem // In S.Cagnoni et al. (Eds.) Applications of evolutionary computing: Proc. of EvoWorkshops 2003, Essex. Springer Verlag. 2003. LNCS, V. 2611. P. 154-164.