

## ОДИН КЛАСС АЛГОРИТМОВ СТОХАСТИЧЕСКОГО ЛОКАЛЬНОГО ПОИСКА

Леонид Гуляницкий, Александр Турчин

**Аннотация:** Рассматриваются алгоритмы ускоренного вероятностного моделирования, которые относятся к классу методов стохастического локального поиска. Приводится общая вычислительная схема алгоритмов, на основе которой с использованием принципа «золотого сечения» предлагается алгоритм комбинаторной оптимизации, названный GS-алгоритмом. Исследуются вопросы сходимости предложенного алгоритма на основе анализа цепей Маркова. Приведена оценка числа операций, необходимых для достижения глобального решения. Дан обзор применения рассмотренных алгоритмов для решения разных типов задач комбинаторной оптимизации.

**Ключевые слова:** combinatorial optimization, stochastic local search, simulated annealing, Markov chains.

---

### Введение

---

Для решения задач комбинаторной оптимизации (КО) широкое применение нашли приближенные алгоритмы. Это объясняется рядом обстоятельств, среди которых выделим следующие: во-первых, все практически важные задачи относятся к числу  $NP$ -трудных, так что точное их решение весьма проблематично даже с использованием современных и перспективных компьютеров; во-вторых, целевые функции имеют, как правило, большое количество локальных экстремумов; в-третьих, во многих прикладных проблемах данные задаются с определенными погрешностями, что делает нецелесообразными те существенные вычислительные затраты, которые необходимы для нахождения их точного решения; в-четвертых, положенные в основу разработки приближенных вычислительных схем идеи (метаэвристики) позволяют создавать алгоритмы, которые могут решать не одну, а целый класс близких по формулировке оптимизационных задач.

В дальнейшем под задачей КО будем понимать проблему поиска хотя бы одного аргумента экстремума заданной целевой функции:

$$x_* = \arg \min_{x \in D_\pi \subseteq X} f(x) \quad (1)$$

где  $X$  – комбинаторное (конечное) пространство,  $D_\pi$  – его подмножество, элементы которого удовлетворяют ограничивающим условиям задачи,  $f: X \rightarrow \mathbb{R}^1$  – целевая функция задачи минимизации.

Отметим, что большинство полученных результатов без труда переносится и на тот случай, когда  $X$  (или только  $D_\pi$ ) – дискретное счетное пространство.

Одним из распространенных и эффективных алгоритмов КО является метод имитационного отжига (МИО) [1]. Опыт применения алгоритмов МИО показал, что получаемые результаты имеют высокую точность, эффективно реализуются на многопроцессорных ЭВМ.

В МИО вероятность принятия очередного решения зависит от степени изменения целевой функции, абсолютная величина которой масштабируется с помощью параметра  $T$  (в МИО он называется температурой). Его изменение (температурное расписание) задают так, чтобы по ходу применения алгоритма в исследуемой окрестности вероятности перехода к худшим вариантам решения (т.е. к

вариантам с бóльшим значением целевой функции) убывали бы от итерации к итерации, стремясь к нулю. Условия, которые определяют необходимость перехода к новому значению параметра  $T$ , в МИО называются условиями равновесия. Наличие такого вероятностного механизма создает предпосылки для выхода в начале поиска алгоритмов МИО из локальных экстремумов, а в конце вычислений способствует концентрации поиска вокруг улучшенных локальных решений.

В тот же время, проведенные многими исследователями вычислительные эксперименты выявили существенные затраты компьютерного времени на поиск решений, а также значительную изменчивость точности получаемых вариантов решений задачи в зависимости от задания значений варьируемых параметров. Необходимость в обобщении вероятностных механизмов и в повышении эффективности алгоритмов дискретной оптимизации привела к разработке алгоритмов ускоренного вероятностного моделирования [2], в которых применена иная вероятностная модель.

## G-алгоритмы

В алгоритмах ускоренного вероятностного моделирования, которые получили еще название G-алгоритмы, на каждой итерации также осуществляется построение точек из окрестности текущего варианта и улучшающие варианты всегда принимаются в качестве очередного приближения, а варианты, соответствующие ухудшению (возрастанию – в случае минимизации) целевой функции, также могут быть выбраны с некоторой вероятностью. Однако, в отличие от МИО, значения этих вероятностей рассчитываются аналогично на протяжении всего вычислительного процесса, но изменяется весовое значение, которое определяет условия отсева ухудшающих вариантов.

Для построения вычислительного процесса формируется строго монотонно возрастающая последовательность действительных чисел  $\{\mu_t\}$ ,  $0 \leq \mu_0 < \mu_1 < \dots \leq 1$ , которые играют роль, в каком-то смысле соответствующую роли параметра  $T$  в МИО. Если  $x^h$  – это найденный текущий вариант на шаге  $h$ , а  $L(x)$  – заданная (метрическая) окрестность произвольной точки  $x \in X$ , то исследуемый вариант  $y \in L(x^h)$ , для которого  $f(y) > f(x^h)$ , может быть принят в качестве  $x^{h+1}$  с вероятностью перехода  $p(x^h, y)$ , зависящей от текущих значений величины  $\mu_t$ .

Пусть  $\Phi(x, y)$ ,  $0 \leq \Phi(x, y) \leq 1$ , – некоторый функционал, зависящий от значений целевой функции задачи. Тогда определим вероятность перехода от точки  $x$  к точке  $y$ ,  $x, y \in X$ , так:

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

Предлагается следующая обобщенная схема G-алгоритмов для решения задач вида (1), представленная на рис.1.

### procedure G\_Search(x)

**begin**

$x^0$  := некоторый начальный допустимый вариант решения из  $X$ ;

$\mu_0$  := 0;  $h$  := 0;  $t$  := 0;

$x_{rec}$  :=  $x^0$ ;  $f_{rec}$  :=  $f(x^0)$ ;

**while** окрестность текущего решения  $L(x^h)$  не просмотрена полностью **do**

**begin**

```

while не выполнено условие равновесия do
  begin
     $y := \text{ГенерированиеСледующейТочкиОкрестности } L(x^h);$ 
    Вычисление  $\Phi(x^h, y);$ 
     $\rho := (1 - \mu_t) \Phi(x^h, y);$ 
     $\xi := \text{random}[0,1];$ 
    if  $\rho \geq \xi$  then
       $h := h + 1; x^h := y;$ 
      if  $f_{\text{rec}} > f(x^h)$  then
         $x_{\text{rec}} := x^h; f_{\text{rec}} := f(x^h)$ 
      end if
    end if;
  end;
  ФормированиеОчередногоЗначения  $\mu_{t+1};$ 
   $t := t + 1;$ 
end;
return  $x = x_{\text{rec}};$ 
end

```

Рис. 1. Алгоритм ускоренного вероятностного моделирования (G-алгоритм)

Здесь  $\text{random}[0,1]$  – датчик случайных чисел из отрезка  $[0,1]$ , при помощи которого моделируются вероятностные переходы.

Построение конкретного алгоритма осуществляется путем конкретизации таких основных аспектов:

- задание функционала  $\Phi(x, y)$ ;
- механизм построения последовательности  $\{\mu_t\}$ ;
- условия равновесия при данном значении величины  $\mu_t$ ;
- правило останова.

При задании функционала  $\Phi(x, y)$  можно использовать монотонные по значениям целевой функции зависимости, удовлетворяющие двум условиям:

- а)  $\Phi(x, y) \rightarrow 1$ , если  $f(y) \rightarrow f(x)$ .
- б)  $\Phi(x, y) \rightarrow 0$ , если  $f(y) \rightarrow \infty$ .

Примером функционала такого рода может быть функционал:

$$\Phi(x, y) = \begin{cases} \left[ \frac{f(x)}{f(y)} \right]^\beta, & \text{если } f(y) \geq f(x), \\ 1, & \text{в противном случае.} \end{cases}$$

где параметр алгоритма  $\beta > 0$  – действительное, в частности, натуральное число.

Во многих практических задачах КО не составляет особых затруднений нахождение верхней границы целевой функции (учитывая, что пространство  $X$  во многих случаях конечно), т.е. такой величины  $f_{\max}$ , что  $f(x) \leq f_{\max}, \forall x \in X$ .

В этом случае можно положить

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

В обычных G-алгоритмах (см. [6] и цитированную там литературу) в качестве функционала  $\Phi(x, y)$  выбирается кусочно-линейный функционал

$$\varphi(x, y) = \begin{cases} \min\{1, 1 - \frac{f(y) - f(x)}{\gamma f(x)}\}, & \text{если } f(y) \geq f(x), \\ 0, & \text{если } f(y) < f(x), \end{cases}$$

где параметр  $\gamma, \gamma > 0$ , – действительное число.

В этом случае при текущем варианте  $x^h$  вероятность перехода  $\rho(x^h, y)$  к точке  $y \in L(x^h)$  представляется кусочно-линейным функционалом, состоящим из трех фрагментов: при  $f(y) < f(x^h)$  его значение равно 1, при  $f(x^h) \leq f(y) \leq (1 + \gamma) f(x^h)$  он убывает от 1 до 0, а при  $f(y) > (1 + \gamma) f(x^h)$  – принимает нулевое значение (рис. 2).

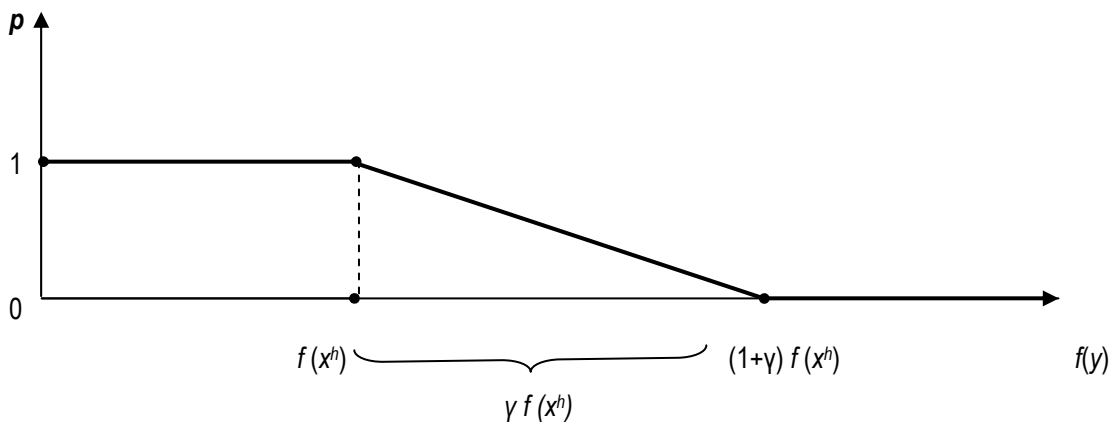


Рис. 2. Вероятность перехода в новую точку окрестности

Таким образом, значение параметра  $\gamma$  определяет величину среднего интервала  $\gamma \cdot f(x^h)$ , при попадании значений  $f(y)$  в который точка  $y$  еще может быть выбрана в качестве  $x^{h+1}$ . Соответственно, при более значительном ухудшении (увеличении в случае поиска минимума) значений  $f(y)$  эта точка всегда удаляется из рассмотрения. Важно отметить, что величина этого интервала автоматически уменьшается при уменьшении  $f(x^h)$ , что происходит при приближении к более точным вариантам решения задачи, и увеличивается в противном случае.

Кроме этого, использование относительных величин позволяет избежать зависимости от абсолютных значений целевой функции. Тем самым, происходит адаптация вычислительного процесса к динамике изменения значений (ландшафта) целевой функции.

В настоящее время наиболее часто применяются два способа порождения последовательности  $\{\mu_t\}$ : через определение и использование некоторой строго монотонной функции, обозначаемой  $G$  (отсюда и сокращенное название –  $G$ -алгоритмы [2]), и на основе использования принципа “золотого сечения” [3].

Пусть  $G: [0,1] \rightarrow [0,1]$  – некоторая заданная строго монотонная действительнoзначная функция, тогда в  $G$ -алгоритмах полагают  $\mu_{t+1} = G(\mu_t)$ . В качестве функции  $G$  на практике обычно выбирают функции вида

$$G_k(x) = \min\{1, (x^{1/k} + H)^k\}, \quad x \geq 0. \quad (3)$$

где  $k \in \{1, 2, 3\}$ , а  $H, 0 < H < 1$ , – некоторая малая величина, выбором значения которой достигается скорость следования  $G_k(x) \rightarrow 1$  при  $x \rightarrow 1$ .

Альтернативно, значения  $\mu_t$  могут формироваться путем деления отрезка  $[0,1]$  с определенным шагом и последовательным выбором построенных точек. Следует заметить, что в случае конечности упомянутой последовательности после достижения максимального значения (равного единице) алгоритм трансформируется в детерминированный локальный поиск.

Эффективным подходом стало использование правила «золотого сечения», которое (хотя и совсем иным способом) используется и при построении алгоритмов поиска экстремумов непрерывных функций одного аргумента. В нашем случае определяется величина  $\mu_t$  как меньшая (левая) из двух точек, которые реализовывают «золотое сечение» отрезка  $[\mu_t, 1]$  (при условии, что  $\mu_0 = 0$ , а  $\mu_{t+1} = G(\mu_t)$ ). Таким образом, правило «золотого сечения» задает скорость приближения левой границы этого отрезка к 1, а соответствующие величины  $\mu_t$  выступают аргументами выбранной функции  $G(x)$ , например, из (3). Алгоритм, использующий правило «золотого сечения», назовем  $GS$ -алгоритмом.

При разработке условий равновесия, учитывая аналогию между параметром “температура” в МИО и  $\mu$ , целесообразно использовать опыт, накопленный при создании алгоритмов МИО. В частности, равновесие может определяться так: задаются некоторый (натуральный) параметр  $\nu$  и действительное число  $\varepsilon > 0$ , а выполнение  $\nu$  переходов называют прогоном [4]. Если при данной температуре выполнено  $k$  прогонов и получены значения  $f_1, \dots, f_k$ , то считают, что достигнуто равновесие, если в  $(k+1)$ -м прогоне выполняется неравенство

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

для некоторого  $i \in \{1, \dots, k\}$ . Величина  $f_i$  для прогона может быть, например, средним значением целевой функции или наилучшим среди тех, которые составляют прогон.

Правилом останова может служить: окончание перебора всех точек в окрестности без реализованного перехода в новую точку; ограничение по продолжительности работы алгоритма; достижение требуемой точности – при известной нижней границе целевой функции. Еще одно из распространенных правил – сравнение разности максимального и минимального значений целевой функции с максимальным значением изменения этой функции при данном значении  $\mu$ : если это соотношение стремится к единице, то вычисления завершаются.

Следует отметить, что в вычислительной схеме  $G$ -алгоритмов достаточно просто учесть многие типы ограничительных условий, которые часто встречаются на практике: соответствующие условия проверки на допустимость могут быть учтены при порождении точек из окрестности текущего варианта. Это позволяет решать не одну, а целый подкласс задач, причем ограничения и некоторые данные задачи могут меняться по ходу ее решения.

По аналогии с алгоритмом повторяющегося локального поиска [1], после завершения работы общей схемы  $GS$ -алгоритма можно использовать процедуру возмущения найденного локального решения, а

полученный в результате возмущения вариант использовать в качестве нового начального приближения для встроеного GS-алгоритма, порождая тем самым метаэвристический метод, который можно назвать повторяющимся GS-алгоритмом.

### Исследование условий сходимости

Изучим вопросы сходимости GS-алгоритма на примере важного класса задач КО – задач на перестановках с транспозиционной метрикой. Для получения оценок скорости сходимости траекторию поиска представим в виде графа. Для этого построим полный взвешенный граф  $Gr = (V, H)$ , где  $V = \{1, \dots, n\}$  является множеством вершин, а  $H$  – множеством ребер графа. Пути между двумя точками пространства решений  $X$  оптимизационной задачи (1) будет соответствовать последовательность ребер этого графа: каждая из вершин графа, соответствующая  $x \in X$ , будет инцидентна тем вершинам, которые соответствуют точкам окрестности  $L(x)$ . Не ограничивая общности, будем рассматривать окрестности минимального радиуса – в большинстве случаев комбинаторных пространств он равен единице:  $L(x) = L_1(x)$ . Поставим в соответствие окрестности  $L(x)$  множество  $N(v), v \in V$  следующим образом: если точка  $x'$  находилась в окрестности  $L(x)$  точки  $x \in X$ , то вершина, соответствующая  $x'$  в графе  $Gr$ , будет связана ребром с вершиной-образом точки  $x$ . Иными словами, если точки в пространстве  $X$  отличались одной транспозицией – то на графе  $Gr$  их вершины-образы будут соединены ребром.

В этом случае случайному процессу поиска решения задачи соответствует процесс переходов от одной вершины графа к другой с использованием определенных вероятностных механизмов. В нашем случае

степень  $d$  графа  $Gr = (V, H)$  равна  $d = \frac{n(n-1)}{2}$ , а диаметр  $D = n - 1$ .

Тогда при выполнении GS-алгоритма процесс поиска моделируется цепью Маркова, поскольку определение состояния, в которое состоится переход, зависит только от текущего состояния системы. Будем считать, что GS-алгоритм сходится, если соответствующая цепь Маркова как минимум раз будет содержать состояние, соответствующее глобальному оптимуму.

Вероятность перехода в следующее состояние при условии нахождения приближения решения с не меньшим значением целевой функции ограничена снизу величиной  $\mu^s$ . Обозначим

$$\hat{d} = \left(\frac{1}{d}\right)^{-D}, \hat{\mu} = \prod_{i=1}^D \mu^i.$$

**Лемма.** Пусть  $v \in V$  – вершина, соответствующая некоторому состоянию  $x$ . Тогда ожидаемое число шагов, за которое будет достигнута вершина, соответствующая глобальному оптимуму,

не превышает  $\frac{\hat{d}}{\hat{\mu}}$ .

*Доказательство.* Пусть  $\underline{x}$  – глобальный оптимум задачи (1), и пусть в графе  $Gr = (V, H)$  этому оптимуму соответствует вершина  $\underline{v}$ . Существует путь от вершины  $v$  к вершине  $\underline{v}$  длины  $q < D$ . Значит, существует последовательность вершин  $v_1, v_2, \dots, v_q$ , которые определяют маршрут.

Если допустить, что любой элемент окрестности  $L(x)$  может быть с одинаковой вероятностью выбран в качестве следующего приближения решения, то можно утверждать, что за  $q$  шагов вероятность перехода в состояние  $\underline{v}$  будет, как минимум, равной  $\left(\frac{1}{d}\right)^q \times \prod_{i=1}^q \mu^i$  и будет выполняться неравенство

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

Следовательно, вероятность попадания в состояние  $\underline{v}$ , начиная с определенного состояния  $v$ , будет не меньше, чем  $\frac{\hat{\mu}}{\hat{d}}$ . А значит, можно утверждать, что ожидаемое число шагов алгоритма, необходимое для

попадания в глобальный оптимум, не будет превышать  $\frac{\hat{d}}{\hat{\mu}}$ .

*Лемма доказана.*

**Теорема.** GS-алгоритм достигает глобального оптимума целевой функции с вероятностью большей, чем  $(1 - \frac{1}{C^k})$ , за число шагов, не превышающее  $Ck \cdot \frac{\hat{d}}{\hat{\mu}}$ , и эта оценка не зависит от начального приближения решения ( $C = const > 1$ ).

*Доказательство.* Для доказательства используем метод математической индукции по  $k$ . Положим  $Q = C \left[ \frac{\hat{\mu}}{\hat{d}} \right]$ . Докажем, что вероятность непопадания за  $kQ$  шагов в вершину  $\underline{v}$ , соответствующую глобальному оптимуму, не превышает  $1/C^k$ .

Базовый случай соответствует  $k=1$  и определенному состоянию  $x$ . В соответствии с леммой, необходимое число итераций для достижения вершины  $\underline{v}$  не будет превышать  $\frac{\hat{d}}{\hat{\mu}}$ . Используя неравенство Маркова, получаем:

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

Допустим, что утверждение справедливо для всех  $k \leq K-1$ . Докажем, что утверждение будет справедливым и для  $k=K$ . Пусть  $x_Q, x_{2Q}, \dots, x_{(K-1)Q}$  – последовательность элементов цепи Маркова, построенная за число шагов  $Q, 2Q, \dots, (K-1)Q$  соответственно.

Рассмотрим два события:

Событие  $H1$ : вершина  $\underline{v}$  не достигнута за первые  $Q$  шагов;

Событие  $H2$ : вершина  $\underline{v}$  не достигнута за следующие  $(K-1)Q$  шагов.

В соответствии с таким определением событий вероятность того, что вершина  $\underline{v}$  не буде достигнута за  $KM$  шагов, равна  $p = p(H2 | H1) \times P(H1)$ .

Из теории цепей Маркова известно, что вероятность достижения глобального оптимума за определенное число шагов, начиная с некоторого состояния, зависит только от количества шагов и текущего состояния и не зависит от состояний, посещенных раньше.

Используя этот факт можно утверждать, что вероятность  $P(H2 | H1)$  зависит только того состояния, в котором будет находиться цепь Маркова на итерации  $Q$ , и от шагов  $(K - 1)Q$ . Значит,

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

Вероятность  $P(H1)$  не превышает  $\frac{1}{C}$  в соответствии с базовым случаем, а вероятность

$P(H2 | x_Q = i)$  не превышает  $\frac{1}{C^{K-1}}$  для каждого  $i \in V$ , исходя из гипотезы математической индукции. Имеем:

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

Вероятность попадания в глобальный оптимум будет обратной к вероятности непадания.

Следовательно, можно утверждать, что с вероятностью, большей чем  $(1 - \frac{1}{C^k})$ , будет достигнут

глобальный оптимум целевой функции за число шагов, не превышающее  $Ck \cdot \frac{\hat{d}}{\hat{\mu}}$ , и эта оценка не зависит

от начального приближения решения.

*Теорема доказана.*

**Следствие.** При  $k \rightarrow \infty$  последовательность построенных приближений решения сходится к глобальному оптимуму по вероятности.

Доказательство этого факта можно получить, исходя из определения сходимости по вероятности.

Особенность полученных оценок состоит в том, что они обобщают известные оценки сходимости алгоритмов МИО (см., например, [5]).

Полученные результаты теоретического исследования сходимости GS-алгоритма могут использоваться и при исследовании иных алгоритмов ускоренного вероятностного моделирования и близких к ним техник поиска.

## Практические применения

Разработанные алгоритмы КО могут применяться для решения задач из разных классов, поскольку они предъявляют весьма общие требования к формулировкам этих задач. Были исследованы алгоритмы решения как известных проблем (задач коммивояжера, квадратичных задач о назначениях), так и более специальных задач (задач размещения, складирования продукции, проектирования технических систем, настройки параметров модели при макроэкономическом прогнозировании и др.).



Одним из первых применений  $G$ -алгоритмов было решение задачи коммивояжера – наверное, наиболее известной из проблем КО как в плане приложений, так и как полигона для исследования предлагаемых исследователями алгоритмов. В [6] приведены решения ряда известных задач, причем результаты оказались сравнимыми с решениями, полученными с использованием супер-ЭВМ. За последнее время накоплен опыт решения комбинированными алгоритмами, основанными на синтезе генетических алгоритмов и  $G$ -алгоритмов.

Широкое применение в различных сферах находит и квадратичная задача о назначениях: разработка радиоэлектронной и цифровой аппаратуры, экономика, планирование и размещение производства, проектирование сложных технических систем и др. Несмотря на то, что эта задача исследуется давно, точное решение даже при использовании супер-ЭВМ вызывает затруднение в общем случае уже для  $n > 15-20$ , причем поиск  $\epsilon$ -приближенного решения является также  $NP$ -полной проблемой [1]. Полученные результаты позволили отобрать наиболее эффективные варианты вычислительных схем, на базе которых в дальнейшем были созданы алгоритмы, использующие правило «золотого сечения» [3], а также метаэвристические алгоритмы (они применялись, в частности, для решения задач коммивояжера и квадратичных задач о назначениях [7,8]).

В работе [9] рассматривался один из классов задач гильотинного раскроя – проблемы оптимального размещения прямоугольников фиксированной высоты и разной длины на полубесконечной ленте, состоящей из заданного числа полос той же высоты при наличии ограничений на размещение [10]. Предложен ряд алгоритмов размещения, которые основаны на схемах локальной оптимизации, имитационного отжига и  $G$ -алгоритмов. Результаты проведенных вычислительных экспериментов показали, что для рассмотренного типа задач  $G$ -алгоритмы позволяют получать лучшие результаты как по точности, так и по трудоемкости по сравнению с другими рассмотренными алгоритмами.

При выборе стратегических аспектов управления запасами на предприятиях и в компаниях возникает проблема оптимального размещения нескольких видов продукции на складе ограниченной вместимости с учетом информации о вероятных потребностях в каждом виде и убытках от неудовлетворенного спроса [11]. Разработанный  $G$ -алгоритм сравнивался с тремя модификациям алгоритма локального поиска и имитационного отжига. Полученные результаты обширного вычислительного эксперимента продемонстрировали стабильность и высокую точность решений, полученных  $G$ -алгоритмом при меньших по сравнению с имитационным отжигом затратах времени.

Одной из важных задач проектирования и оптимизации телекоммуникационных сетей с технологией АТМ является задача оптимального выбора пропускных способностей каналов связи. В ней задана структура сети, состоящая из коммутаторов, соединенных каналами связи определенной длины. Заданы также требования в передаче трафика для каждой упорядоченной пара коммутаторов, а для каждого канала связи известны величины общих потоков. Пропускная способность любого канала связи пропорциональна пропускной способности базового канала. Необходимо для всех каналов связи выбрать такое количество базовых каналов, при котором стоимость функционирования сети будет минимальной при выполнении ограничений на показатели качества обслуживания. Для решения названных задач были разработаны и реализованы алгоритмы локального поиска и повторяющегося локального поиска, МИВ,  $G$ -алгоритм, а также генетический алгоритм. Анализ результатов вычислительных экспериментов показал, что для разных видов трафика наилучшие по точности решения в большинстве случаев находились  $G$ -алгоритмом, сравнимые с ними результаты для задач с отдельными видами трафика показал алгоритм повторяющегося локального поиска [12].

---

## Заклучение

---

В работе описаны алгоритмы, принадлежащие к классу методов стохастического локального поиска. Предложена их модификация, названная GS-алгоритмом, которая использует правило "золотого сечения". Применение оригинальных вероятностных механизмов организации вычислений в соединении с процедурой локального поиска позволяет эффективно решать задачи КО из разных классов. На основе использования цепей Маркова получена вероятностная оценка верхней границы для числа итераций, необходимых для нахождения глобального минимума. Приведены примеры применения алгоритмов рассмотренного типа для решения ряда задач КО. Полученные результаты позволяют рекомендовать разработанные алгоритмы как для решения прикладных задач КО, так и в качестве встроенной процедуры в метаэвристических алгоритмах, прежде всего в алгоритмах, основанных на популяциях [1].

---

## Список литературы

---

1. Hoos H.H., Stützle T. Stochastic Local Search: Foundations and Applications. – San Francisco: Morgan Kaufmann Publ., 2005. – 658 p.
2. 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.
3. Гуляницкий 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.).
4. 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.
5. Rajasekaran S. Simulated annealing and nested annealing // J.of Global Optimization.– 2000.–16. –P. 43-56.
6. 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.
7. Гуляницкий Л.Ф., Турчин О.Я. Об одном подходе к использованию вероятностного моделирования в схеме генетического алгоритма //Тр. Междун. конф. по индуктивному моделированию, Львов 20-25 мая 2002. –Львов: ГНИИ информационной инфраструктуры, 2002. Т 2. – С. 275-281 (укр.).
8. 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.
9. 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.
10. Грицюк Ю.И. Регулярное размещение прямоугольных объектов вдоль полос односторонне ограниченной ленты. – Львов: Изд. дом «Панорама», 2002. – 220 с. (укр.).
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.

---

**Об авторах**

---

**Леонид Гуляницкий** (Hulianytskyi) – д.т.н., Институт кибернетики им. В.М.Глушкова НАН Украины, пр-т Глушкова, 40, Киев, 03680, Украина. e-mail: [lh\\_dar@hotmail.com](mailto:lh_dar@hotmail.com)

**Александр Турчин** – Институт кибернетики им. В.М.Глушкова НАН Украины, пр-т Глушкова, 40, Киев, 03680, Украина. e-mail: [turchin@ua.fm](mailto:turchin@ua.fm)