УДК 519.8:004.023

**Ю. И. Валиахметова**

Башкирский государственный аграрный университет
ул. 50-летия Октября, 34, Уфа, 450001, Россия

E-mail: julikas@inbox.ru

# ГИПЕРЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ В ЗАДАЧАХ ПРЯМОУГОЛЬНОГО РАСКРОЯ *

Описываются основные особенности и концепция построения гиперэвристических алгоритмов на примере задач прямоугольного раскроя и упаковки. На основе результатов численного эксперимента, проведенного в ходе внедрения гиперэвристических систем в производственный процесс, получены сведения об их высокой эффективности. Потенциально возможная эффективность гиперэвристик позволяет ожидать бурного их развития в ближайшем будущем.

*Ключевые слова*: гиперэвристика, метаэвристика, оптимизация, поиск, эвристика.

## Введение

Эвристические алгоритмы по праву завоевали всемирную популярность, позволив получать хорошие решения не решаемых точными методами оптимизационных задач. В стремлении получать лучшие решения люди создавали все новые эвристические алгоритмы, исследовали их на различных классах задач и приходили к выводу, что для задач определенной специфики целесообразно использовать эвристики, которые плохо справляются с решением других оптимизационных задач. Среди причин этого явления можно назвать и то, что каждая метаэвристика изначально создается для конкретной задачи и в полной мере учитывает лишь ее особенности, и нельзя назвать эвристический алгоритм, который превосходил бы остальные алгоритмы на всех без исключения классах задач. Было разработано огромное количество эвристик, и применимость каждой из них обосновывалась либо экспериментальными результатами, либо аргументами, основанными на специфике класса задач, для которого создавалась эвристика. Понимая под эвристиками алгоритмы поиска, некоторые авторы порой утверждали абсолютное превосходство одной эвристики над другой. Эта практика стала исчезать, когда в 1995 г. Wolpert и MacReady [1] опубликовали «теорему об эквивалентности алгоритмов поиска» (No Free Lunch Theorem), которая показывает, что в среднем по всем задачам, определенным на некотором заданном конечном пространстве поиска, все алгоритмы поиска имеют одинаковую среднюю эффективность. Все эти обстоятельства послужили толчком к созданию эвристических оптимизационных алгоритмов нового поколения – гиперэвристик. Каждая гиперэвристика по сути представляет собой совокупность нескольких простых эвристик или метаэвристик и набор правил, позволяющих управлять их работой и выбирать для каждой конкретной задачи наиболее эффективные методы из имеющегося перечня. Использование такой структуры дает возможность значительно снижать эффект узко-

* Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект № 12-07-00631-а.

*Валиахметова Ю. И.* Гиперэвристические алгоритмы в задачах прямоугольного раскроя // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2013. Т. 11, вып. 2. С. 36–43.

направленности алгоритмов и создавать более универсальные, применимые для качественного решения различных классов задач.

**Структура гиперэвристических алгоритмов**

Гиперэвристика представляет собой управляющую систему, в подчинении которой имеется определенный перечень эвристических или метаэвристических алгоритмов. Набор подчиненных алгоритмов определяет разработчик исходя из особенностей задачи, выбирая наиболее эффективные для рассматриваемого класса задач.

С помощью гиперэвристик задача может решаться как единственным алгоритмом, так и с применением нескольких алгоритмов, каждый из которых сформирует частичное решение задачи. Гиперэвристика может влиять на очередность запуска подчиненных эвристик, частоту их применения, в зависимости от параметров текущего решения оценивать, какую из эвристик запустить следующей, определяет, какое из найденных решений является лучшим, а также на все без исключения параметры и условия получения решений. Если эвристике отдан приоритет на данном шаге, ее работа продолжается либо заданное количество итераций, либо определенный временной отрезок, либо по некоторому другому правилу, указанному разработчиком. Частным случаем гиперэвристики является метаэвристика, где в качестве подчиненных алгоритмов используются простые эвристики. Однако в общем случае метаэвристика может находиться и на нижнем уровне гиперэвристики. Поэтому область решений, которые могут быть достигнуты с помощью гиперэвристики, гораздо обширнее, чем при работе каждой из метаэвристик или простых эвристик нижнего уровня по отдельности.

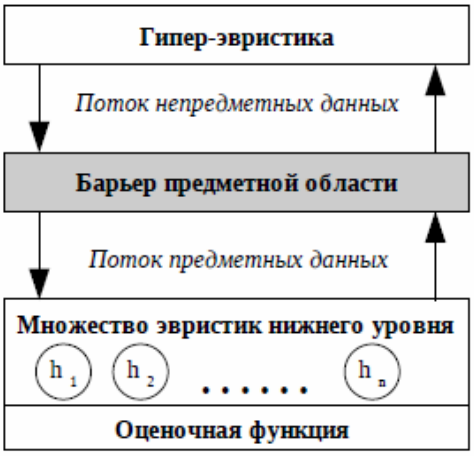



*Рис. 1.* Структура гиперэвристики

Для того чтобы разработанный гиперэвристический алгоритм справлялся с новыми проблемными областями, достаточно заменить множество эвристик нижнего уровня и оценочную функцию, определяющую качество получаемых решений.

Визуализация общей гиперэвристической схемы представлена на рис. 1. Как видно, между эвристиками нижнего уровня и гиперэвристикой находится барьер. Знаниям о предметной области не разрешается пересекать этот барьер. Таким образом, гиперэвристика не имеет информации об области, в которой она работает. Она знает только о том, что имеется $n$ низкоуровневых эвристик, которые можно вызывать и что качество полученного решения определяется оценочной функцией.

Для организации взаимодействия между гиперэвристикой и эвристиками нижнего уровня должен быть хорошо организован интерфейс. Это позволит гиперэвристике обмениваться информацией с эвристиками нижнего уровня, используя стандартный интерфейс; в противном случае гиперэвристике понадобится отдельный интерфейс для взаимодействия с каждой эвристикой нижнего уровня, что, очевидно, нелепо. К тому же это способствует проходу непредметных знаний между эвристиками нижнего уровня и гиперэвристикой (и обратно). Например, интерфейс, разработанный в [2], включает такие компоненты, как результаты оценивающей функции и процессорное время, затраченное эвристикой нижнего уровня, которое может быть вычислено гиперэвристикой. Можно также включить компонент, позволяющий определить для каждой эвристики нижнего уровня, как долго она должна работать. Идея состоит в том, что мы вызываем эвристики по очереди, давая каждой определенное время на работу, и эвристика, показавшая лучшие результаты за предоставленное время, применяется к текущему решению. В совокупности с этим компонентом интерфейс также определяет, должна ли эвристика нижнего уровня соответствующим образом преобразовать текущее решение или она должна только сообщить, каков был бы эффект от этого преобразования. Тогда гиперэвристика может опросить каждую низкоуровневую эвристику, какие

изменения она может произвести с текущим решением, а затем решить, какой эвристике (или набору эвристик) разрешить произвести эти изменения.

Основное предназначение гиперэвристик состоит в создании с их помощью универсальных оптимизационных систем, позволяющих получать хорошие решения для любых задач раскроя-упаковки вне зависимости от их специфики. Реализация гиперэвристических алгоритмов представляется более простой, нежели создание проблемно-ориентированных метаэвристических технологий.

**Некоторые исторические факты**

История гиперэвристик началась около 15 лет назад, когда в нескольких странах независимо друг от друга стали появляться метаэвристичские алгоритмы решения различных оптимизационных задач, имеющие похожую структуру и объединяющие в себе несколько ранее разработанных метаэвристических или эвристических алгоритмов.

Примеры работы гиперэвристики на высшем уровне абстракции можно найти в работах Cowling, Kendall и Soubeiga [2–5]. В [2] гиперэвристический подход был разработан и применен к задаче о саммите по продажам (задаче о сопоставлении поставщиков потенциальным покупателям на семинаре по продажам). Гиперэвристика имела доступ к 10 эвристикам нижнего уровня, которые включали удаление делегата со встречи с определенным поставщиком, разрешение встречи между делегатом и поставщиком и сокращение числа встреч с поставщиком, у которого их слишком много. В [4] гиперэвристика была модифицирована так, что автоматически подбирала некоторые параметры, но опять же в качестве тестовой задачи решалась задача о саммите по продажам.

В [5] гиперэвристика осталась прежней, но была представлена новая задача. На этот раз составлялось расписание презентаций курсовых проектов студентов третьего года обучения одного из университетов Великобритании. Было разработано восемь низкоуровневых эвристик, включая эвристику, предусматривающую замену одного из членов комиссии определенной секции, перенос презентации из одной секции в другую. Только эвристики нижнего уровня и оценочная функция были заменены. Для обеих проблемных областей этот алгоритм позволил получить решения хорошего качества.

В дальнейшем идея была развита в [3], где тот же самый гиперэвристический подход был применен к составлению графика дежурств медицинских сестёр. Единственная информация, к которой гиперэвристика имела доступ в данной схеме, – это данные, общие для всех типов задач, которые она должна сохранить как часть своего внутреннего состояния, например:

- процессорное время, затраченное конкретной эвристикой на ее последнем запуске;
- изменение в оценочной функции при вызове конкретной эвристики;
- время, в течение которого конкретная эвристика не вызывалась.

Важно то, что гиперэвристике не нужно владеть информацией о функциях каждой эвристики и даже о том, какая задача решается.

Следовательно, в обозначенных рамках разработчик гиперэвристики может применить все свое воображение. К примеру, внутреннее состояние гиперэвристики способно хранить информацию о том, насколько хорошо пары эвристик работают вместе. Может возникнуть такая ситуация: если гиперэвристика вызвает одну эвристику за другой, это приводит к ухудшению оценочной функции после первого вызова, но к резкому (или хотя бы плавному) улучшению после вызова второй эвристики. В таком случае одной из гиперэвристических идей может быть хранение данных о парах (тройках и т. д.) эвристик, которые хорошо работают при последовательном вызове, а при отдельных вызовах каждая эвристика в целом показывает плохие результаты.

Используя свое внутреннее состояние, гиперэвристика должна выбрать, какую из низкоуровневых эвристик вызвать следующей, и тут возможны варианты. Например, для того, чтобы лучше обосновать выбор следующей эвристики нижнего уровня (или пары, тройки эвристик и т. д.), можно на очередном шаге отдать приоритет эвристике, которая:

- привела к наибольшему улучшению значения оценочной функции;
- быстрее всех работает;

- давно не вызывалась;
- учесть все эти факторы (возможно, и некоторые другие);
- и т. д.

Внутреннее состояние гиперэвристики – это дело разработчика, как и процесс выбора следующей эвристики, однако хорошая гиперэвристика может показать весьма неплохие результаты на целом спектре задач, а не только на той, для которой изначально реализовывалась.

Гиперэвристическая схема, описанная выше, использовалась в работах [2; 4; 5], где гиперэвристика хранит информацию о внутреннем состоянии; к этой информации имеет доступ функция выбора, решающая, какую из эвристик нижнего уровня вызвать следующей.

Функция выбора – это выражение, включающее информацию об эффективности каждой эвристики нижнего уровня ($f_1$), эффективности пар эвристик ($f_2$) и времени, в течение которого каждая эвристика не вызывалась ($f_3$). Таким образом, мы имеем функцию

$$f(H_k) = \text{б}f_1(H_k) + \text{в}f_2(H_j, H_k) + \text{д}f_3(H_k),$$

где $H_k$ – $k$-я эвристика;

б, в и д – веса, отражающие значимость каждого компонента (именно эти веса адаптивно подбираются в [4]);

$f_1(H_k)$ – показатель эффективности эвристики $H_k$;

$f_2(H_j, H_k)$ – показатель эффективности пары эвристик $H_j$, $H_k$;

$f_3(H_k)$ – время, прошедшее с последнего вызова эвристики $H_k$.

$f_1$ и $f_2$ призваны интенсифицировать поиск, $f_3$ пытается внести элемент диверсификации. Максимальное значение функции $f(H_k)$ определяет эвристику, которая будет вызвана следующей.

Разработка гиперэвристик будет играть ключевую роль среди технологий поиска в ближайшие несколько лет. Потенциал научного прогресса в развитии более универсальных оптимизационных систем для применения в самых разнообразных областях весьма значителен. Важным направлением в разработке метаэвристик станет исследование применения гиперэвристик к широкому кругу задач. В настоящее время работа в этом направлении уже интенсивно ведется.

**Эффективность гиперэвристических алгоритмов**

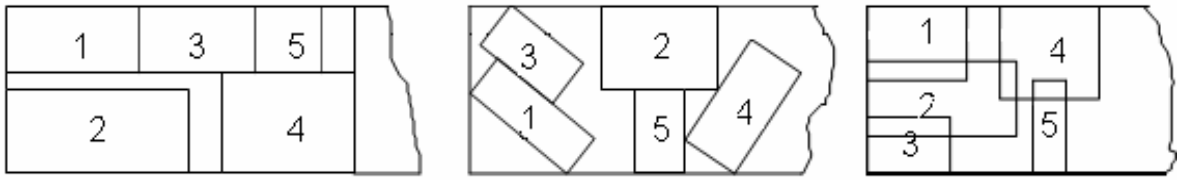
Первые мультиметодные алгоритмы предвосхитили появление гиперэвристик. Так, метод комбинирования эвристик (Heuristics Combination Method, HCM), предложенный И. П. Норенковым [6], стал основой генетического мультиметодного алгоритма (Genetic Multi-Method Algorithm, GMA), который в свою очередь породил гипергенетический алгоритм (Hiper-Genetic Algorithm, HGA) [7].

Все эти алгоритмы были опробированы на решении задачи ортогональной упаковки прямоугольников в полубесконечную полосу.

*Постановка задачи.* Среди задач ортогонального размещения контейнеров (BP) различаются задачи размещения в полубесконечной полосе (1.5DBP) и на прямоугольных листах (2DBP).

Пусть имеются прямоугольная полоса заданной ширины $W$ и неограниченной длины и набор из $m$ прямоугольных предметов заданных размеров $(w_i, l_i)$, $I = 1, \ldots, m$, где $w_i$ – ширина, $l_i$ – длина стороны, параллельной неограниченной грани полосы.

На рис. 2 изображены допустимое и недопустимое размещения. Если длина $L$ занятой части полосы достигает минимума, то полученное размещение называется оптимальным. Часто, особенно в зарубежной литературе, в качестве критерия оптимальности в 1.5DBP используется площадь огибающего прямоугольника. Модель 1.5DBP применяется в основном при теоретических разработках, как подзадача проблемы размещения на листах, или непосредственно при раскрое рулонного материала.




а б в

*Рис. 2.* Размещения прямоугольников: *а* – допустимое ортогональное; *б* – неортогональное; *в* – недопустимое ортогональное

Задача прямоугольного размещения на листах состоит в следующем: кроме ширины $W$ известна и длина $L$ раскраиваемого прямоугольника, в нем требуется разместить прямоугольные детали с размерами $(w_i;\ l_i)$, $i=\overline{1,m}$. Обычно исходная информация о заготовках такова, что одного листа для их размещения оказывается недостаточно и на множестве допустимых размещений требуется найти минимальное количество используемых листов.

Выделяют также задачу размещения прямоугольно-ориентированных предметов на листах или в полубесконечной полосе, которая характеризуется сложной формой укладываемых предметов.

*Математическая модель.* Исходная информация задач представляет набор $\langle W; L; m; w; l; \in \rangle$, где $W$ – ширина полубесконечной полосы или листа; $L$ – длина листа, для полубесконечной полосы $L=\infty$; $m$ – количество прямоугольников (деталей, заготовок); $w=(w_1,...,w_i,...,w_m)$, $w_i$ – ширина прямоугольника $i=\overline{1,m}$; $l=(l_1,...,l_i,...,l_m)$, $l_i$ – длина прямоугольника $i=\overline{1,m}$; $\in$ – флаг разрешения поворота прямоугольников.

Решения рассматриваемых задач могут быть представлены в виде набора $R=\langle X,\ Y,\ S,\ A\rangle$, где $X=(x_1,...,x_i,...,x_m)$, $Y=(y_1,...,y_i,...,y_m)$, $x_i$ и $y_i$ – координаты прямоугольника $i$ по осям $OX$ и $OY$, $i=\overline{1,m}$; $S=(s_1,...,s_i,...,s_m)$, где $s_i$ – номер листа, в который упакован прямоугольник, $i=\overline{1,m}$; в случае раскроя полубесконечной полосы $S=\varnothing$;

$$A=(a_1,...,a_i,...,a_n),\ a_i=\begin{cases}1, \text{ если прямоугольник } i \text{ повернут на } 90^\circ\\ 0 \text{ иначе}\end{cases}\Bigg\}.$$

Набор $RP=\langle X,\ Y,\ S,\ A\rangle$ называется допустимым прямоугольным размещением, если выполняются следующие условия.

1. Грани прямоугольников параллельны граням полосы или листа:

$((i_x=l_i)\wedge(i_y=w_i))\vee((i_x=w_i)\wedge(i_y=l_i)), i=\overline{1,m}$, где $i_x$, $i_y$ – проекции заготовки $i$ на оси координат $OX$ и $OY$.

2. Взаимное неперекрытие прямоугольников: $\forall i\neq j;\ i,j=1,\ ...,\ m$,

$(x_i\geq x_j+l_j)\vee(x_j\geq x_i+l_i)\vee(y_i\geq y_j+w_j)\vee(y_j\geq y_i+w_i)$ или $s_i\neq s_j$, если $S\neq\varnothing$.

3. Неперекрытие прямоугольников с гранями полосы или листов:

$\forall i=1,...,m:(x_i\geq 0)\wedge(y_i\geq 0)\wedge(y_i+w_i\leq W)\wedge(x_i+l_i\leq L)$.

Сформулируем теперь интересующие нас задачи прямоугольной упаковки.

*Задача 1.5DBP.* При исходных данных $\langle W;\ m; w; l; \in\rangle, L=\infty,$ минимизировать длину

$$\Lambda=\max_{i=1,m}(x_i+l_i)$$

занятой части полосы на множестве допустимых размещений $RP=\langle X,\ Y,\ S,\ A\rangle$, $S=\varnothing$, удовлетворяющем условиям 1–3.

Если длина занятой части полосы достигает минимума, то RP (Rectangular Packing, RP) – оптимальная упаковка.

*Задача 2DBP*. При исходных данных $\langle W; L; m; w; l; \in \rangle$ минимизировать количество израсходованных листов

$$N = \max_{i=1,m}(s_i)$$

на множестве допустимых размещений $RP = \langle X,\ Y,\ S,\ A \rangle$, удовлетворяющих условиям 1–3.

Все исследуемые алгоритмы используют перечень простых эвристик и запускают их в некоторой очередности. За один шаг к частичной упаковке добавляется один прямоугольник. Отличие их между собой состоит в том, что в методе комбинирования эвристик пошагово строится одна упаковка, на каждом шаге случайным образом выбирается эвристика, определяющая один прямоугольник для размещения. Генетический мультиметодный алгоритм использует генетическую схему перебора получаемых упаковок с последующим их скрещиванием и отбором лучших, эвристики на каждом шаге по-прежнему выбираются случайным образом. Гипергенетический алгоритм включает дополнительно функцию самообучения, т. е. выбор эвристик на каждом шаге осуществляется не случайно, а с использованием показателей эффективности, которые пересчитываются для каждой эвристики на каждом шаге.

В качестве тестовых задач были использованы примеры из статей Berkley & Wang [8] и Martello & Vigo [9] из библиотеки Or-Library, для которых точным алгоритмом вычислены значения оптимальных решений. Используются 500 задач прямоугольной упаковки в полубесконечную полосу, поделенные на 10 классов. Первые шесть классов были предоставлены Berkley & Wang, для каждого из этих классов указан интервал, на котором равномерно распределены значения длин и высот прямоугольников, а также ширина полубесконечной полосы $W$, являющейся ресурсом в задаче.

Класс 1. [1, 10], $W = 10$.
Класс 2. [1, 10], $W = 30$.
Класс 3. [1, 35], $W = 40$.
Класс 4. [1, 35], $W = 100$.
Класс 5. [1, 100], $W = 100$.
Класс 6. [1, 100], $W = 300$.

В каждом из указанных классов размеры прямоугольников сгенерированы на одинаковых интервалах. Martello & Vigo предложили следующие более близкие к реальности классы, в которых заданы интервалы равномерного распределения ширины прямоугольника $w_j$ и его высоты $h_j$.

Тип 1. $[\frac{2}{3}W; W]$, $[1; \frac{1}{2}W]$.

Тип 2. $[1; \frac{1}{2}W]$, $[\frac{2}{3}W; W]$.

Тип 3. $[\frac{1}{2}W; W]$, $[\frac{1}{2}W; W]$.

Тип 4. $[1; \frac{1}{2}W]$, $[1; \frac{1}{2}W]$.

Высота полосы $W = 100$ для всех классов.

Класс 7. Тип 1 с вероятностью 70 %, типы 2, 3, 4 с вероятностью 10 %.
Класс 8. Тип 2 с вероятностью 70 %, типы 1, 3, 4 с вероятностью 10 %.
Класс 9. Тип 3 с вероятностью 70 %, типы 1, 2, 4 с вероятностью 10 %.
Класс 10. Тип 4 с вероятностью 70 %, типы 1, 2, 3 с вероятностью 10 %.

Количество прямоугольников принимает различные значения: 20, 40, 60, 80, 100. Для каждого класса и количества прямоугольников генерируется 10 задач.

После решения всех 500 задач было найдено среднее значение длины израсходованного ресурса по каждому классу, а также отклонение этой величины от нижней границы (в %). Эти результаты приведены в таблице.

Результаты численного эксперимента

| Класс | Ниж. граница | HCM | | GMA | | HGA | |
|---|---|---|---|---|---|---|---|
| | | длина | м% | длина | м% | длина | м% |
| C1 среднее | 187,18 | 190,46 | 2,01 | 189,34 | 1,43 | 189,24 | 1,21 |
| C2 среднее | 60,52 | 61,86 | 2,69 | 61,40 | 1,75 | 61,10 | 1,24 |
| C3 среднее | 504,12 | 537,06 | 6,72 | 530,60 | 5,40 | 519,70 | 3,23 |
| C4 среднее | 193,50 | 204,12 | 6,11 | 202,08 | 4,89 | 199,54 | 3,61 |
| C5 среднее | 1613,16 | 1724,90 | 6,79 | 1703,48 | 5,49 | 1661,46 | 3,25 |
| C6 среднее | 506,40 | 543,82 | 8,33 | 537,42 | 6,72 | 528,26 | 4,97 |
| C7 среднее | 1577,82 | 1623,64 | 3,02 | 1615,24 | 2,59 | 1597,38 | 1,40 |
| C8 среднее | 1397,92 | 1538,50 | 10,60 | 1516,02 | 8,78 | 1471,40 | 5,54 |
| C9 среднее | 3343,10 | 3358,58 | 0,53 | 3353,82 | 0,39 | 3346,20 | 0,07 |
| C10 среднее | 909,16 | 982,82 | 8,09 | 971,54 | 6,72 | 952,64 | 4,60 |
| Среднее | 1158,905 | 1209,794 | 5,5 | 1200,77 | 4,45 | 1184,23 | 2,91 |

Как видно, разумное управление работой генетического алгоритма дает существенное улучшение качества получаемых решений.

Предложенная методика была внедрена на ряде предприятий в ходе экспериментального тестирования реализации гипергенетического алгоритма.

Полиграфическое предприятие ООО «Европак» (г. Уфа) специализируется на изготовлении картонной тары для различного рода товара. Развертка картонной тары в общем случае представляет собой невыпуклый полигон. Картон, из которого раскраиваются заготовки, является одним из основных ресурсов предприятия, именно поэтому важна организация на предприятии рационального раскроя картонных заготовок.

Разработанный гиперэвристический алгоритм также был внедрен в производственный процесс на предприятии ООО «Матрица-трейд» (г. Уфа), занимающемся доставкой продовольственных товаров по сети магазинов. Для таких предприятий важно получать рациональный план загрузки грузовых отсеков транспортных средств с учетом порядка выгрузки по мере посещения пунктов отгрузки.

В обоих случаях речь идет о решении задачи, сводящейся к задаче прямоугольного раскроя при дополнительных ограничениях.

Экспериментальное внедрение разработанного гиперэвристического алгоритма в производственный процесс показало следующие результаты:

- внедрение на ООО «Европак» (г. Уфа) привело к снижению отходов материала при раскрое картона на 14 %;
- внедрение на ООО «Матрица-трейд» (г. Уфа) привело к увеличению эффективности загрузки грузовых отсеков транспортных средств до 5–7 %.

Основными среди алгоритмических факторов, влияющих на эффективность, следует назвать используемый набор эвристик, вероятности их выбора в операторах мутации и формирования начального поколения, глубину локального поиска, размер макромутаций, характер расположения в хромосоме мутируемых генов.

**Заключение**

Разработка гиперэвристик будет играть ключевую роль среди технологий поиска в ближайшие несколько лет. Потенциал научного прогресса в развитии более универсальных оптимизационных систем для применения в самых разнообразных областях весьма значителен. Важным направлением в разработке метаэвристик станет исследование применения гиперэвристик к широкому кругу задач. В настоящее время работа в этом направлении ведется очень активно. Результаты внедрения гиперэвристических алгоритмов в производственный процесс подтверждает их высокую эффективность и позволяет наметить пути дальнейших исследований.

**Список литературы**

1. *Wolpert D., MacReady W. G.* No Free Lunch Theorems for Optimization // IEEE Transactions on Evolutionary Computation. 1997. Vol. 1 (1). P. 67–82.

2. *Cowling P., Kendall G., Soubeiga E.* A Hyperheuristic Approach to Scheduling a Sales Summit. In LNCS 2079, Practice and Theory of Automated Timetabling III // Third International Conference, PATAT 2000. Konstanz, Germany, August 2000, selected papers / Eds. E. K. Burke, W. Erben. Springer-Verlag, P. 176–190.

3. *Cowling P., Kendall G., Soubeiga E.* Hyperheuristics: A Robust Optimisation Method Applied to Nurse Scheduling // Technical Report NOTTCS-TR-2002-6 (submitted to PPSN 2002 Conference). University of Nottingham, UK, School of Computer Science & IT, 2002.

4. *Cowling P., Kendall G., Soubeiga E*. A Parameter-Free Hyperheuristic for Scheduling a Sales Summit // Proceedings of IV Metahuristics International Conference (MIC 2001). Porto, Portugal, 2001. P. 127–131.

5. *Cowling P., Kendall G., Soubeiga E*. Hyperheuristics: A Tool for Rapid Prototyping in Scheduling and Optimisation // LNCS 2279, Applications of Evolutionary Computing: Proceedings of Evo Workshops 2002 / Eds. S. Cagoni, J. Gottlieb, E. Hart, M. Middendorf, R. Gunther Kinsale, Ireland, 2002. P. 1–10.

6. *Норенков И. П.* Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. 1999. № 1. С. 2–7.

7. *Валиахметова Ю. И., Филиппова А. С.* Мультиметодный генетический алгоритм для решения задач ортогональной упаковки // Информационные технологии. 2007. № 12 (136). С. 50–57.

8. *Berkey J. O., Wang P. Y.* Two Dimensional Finite Bin Packing Algorithms // J. Oper. Res. Soc. 1987. Vol. 38. P. 423–429.

9. *Martello S., Vigo D.* Exact Solution of Two-Dimensional Finite Bin Packing Problem // Management Science. 1997. Vol. 35. P. 64–68.

*Материал поступил в редколлегию 10.01.2012*

**Yu. I. Valiakhmetova**

**THE HYPERHEURISTIC ALGORITHMS ARE IN THE TASKS OF THE RECTANGULAR CUTTING**

The article is devoted description of basic features of hyperheuristic algorithms, applied for the decision of tasks of the rectangular cutting and packing. On the basis of results of numeral experiment got taking about efficiency of hyperheuristic algorithms.

*Keywords*: hyperheuristic, metaheuristic, optimization, search, heuristic.