

На правах рукописи

**ШИРГАЗИН Рамиль Ришатович**

**ЭВОЛЮЦИОННЫЕ МЕТОДЫ  
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
ДЛЯ РЕШЕНИЯ ЗАДАЧ  
ОРТОГОНАЛЬНОЙ УПАКОВКИ  
НА БАЗЕ БЛОЧНЫХ СТРУКТУР**

**05.13.18 – Математическое моделирование, численные методы и  
комплексы программ**

**АВТОРЕФЕРАТ  
диссертации на соискание ученой степени  
кандидата технических наук**

Уфа 2006



ШИРГАЗИН Рамиль Ришатович

**ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ ПОИСКА ОПТИМАЛЬНОГО  
РЕШЕНИЯ В ЗАДАЧАХ ОРТОГОНАЛЬНОЙ УПАКОВКИ НА БАЗЕ  
БЛОЧНЫХ СТРУКТУР**

05.13.18 - Математическое моделирование,  
численные методы и комплексы программ

**АВТОРЕФЕРАТ**  
диссертации на соискание ученой степени  
кандидата технических наук

Подписано в печать 00.00.2005. Формат 60×84 <sup>1</sup>/<sub>16</sub>  
Бумага офсетная. Печать плоская. Гарнитура Times New Roman Cyr.  
Усл. печ. л. 1,5. Усл. кр. отт. 1,4. Уч.-изд. л. 1,0  
Тираж 100 экз. Заказ №

Уфимский государственный авиационный технический университет  
Центр оперативной полиграфии УГАТУ  
450000, Уфа-центр, ул. К.Маркса, 12

**Актуальность проблемы.** Задачи раскроя-упаковки представляют собой важный прикладной раздел дискретной оптимизации. Актуальность проблемы создания эффективных алгоритмов для решения задачи раскроя-упаковки двумерного прямоугольного ресурса обусловлена как широким практическим применением задач в различных отраслях производства, так и трудностью создания адекватных математических моделей и методов их решения.

От качества полученного решения зависит:

- эффективность использования ресурсов;
- продуктивность использования оборудования;
- время проектирования и производительность труда.

Задачи раскроя – упаковки представляют большой интерес для производства. В условиях массового производства изделий даже незначительная экономия сырья на одно изделие даст после приведения к годовому объему продукции многие тонны сэкономленного материала. Этим объясняется внимание, которое уделяется совершенствованию методов расчета раскроя.

Сложность решения задачи раскроя-упаковки обусловлена ее принадлежностью к классу NP-трудных задач комбинаторной оптимизации. Исследуемая задача является NP-трудной в сильном смысле, так как содержит в качестве подзадачи также NP-трудную задачу. Во многих случаях применение точных методов для ее решения невозможно из-за больших затрат вычислительного времени. В связи с этим большое значение приобретает разработка и исследование эвристических методов оптимизации, в том числе метаэвристик. В их числе широкое применение получили генетические алгоритмы. Известна асимптотическая сходимость таких методов. На практике метаэвристики очень хорошо себя зарекомендовали. Качество полученного решения зависит не только от выбранного метода расчета раскроя-упаковки. Важную роль выполняют и способы кодирования и декодирования упаковок.

Исходя из вышеизложенного, представляет интерес разработка и применение новых методов, в том числе эволюционных алгоритмов для решения задачи раскроя-упаковки прямоугольных предметов на базе эффективных принципов кодирования упаковок. Важным является создание программной реализации, которая позволила бы получать качественные решения производственных задач за приемлемое время. Разработанное алгоритмическое и программное обеспечение становится конкурентоспособным в ряду точных и эвристических подходов. В этом состоит актуальность данной разработки.

**Целью работы** является разработка и реализация эффективных численных методов, алгоритмов и программного обеспечения для решения задач прямоугольной упаковки на базе блочных технологий.

Для ее достижения были поставлены и решены следующие задачи:

1. Рассмотреть постановки задач прямоугольной упаковки и выделить часто встречающиеся в промышленности модели и методы решения задач упаковки и раскроя.

2. Провести анализ и модифицировать простые алгоритмы конструирования упаковок (декодеры, формирующие упаковку на основе заданного кода), с целью повышения их эффективности.

3. Разработать новый эффективный декодер на базе блочных структур, совмещающий высокую производительность простых эвристик с оптимальным использованием материала за счет применения «жадных» стратегий.

4. Разработать «наивный» эволюционный алгоритм на базе простых эвристик и «наивного оператора мутации»

5. Модифицировать эволюционные алгоритмы с применением жадного декодера замещения для поиска и исследования эффективных методов решения задачи упаковки.

6. Создать программное обеспечение, реализующее модифицированные и новые разработанные методы.

7. Провести анализ эффективности и характеристик разработанных алгоритмов на основе результатов численных экспериментов в сравнении с другими методами, описанными в литературе.

**На защиту выносятся:**

1. Эволюционный «наивный» алгоритм на базе простых эвристик типа «подходящий» и «наивного оператора мутации».

2. Декодер «жадного» замещения Greedy Sub на базе блочной структуры и его модификации.

3. Генетический алгоритм на базе блочных структур для решения задачи прямоугольной упаковки полубесконечной полосы и на листы.

4. Гибридизация генетического алгоритма с блочным декодером «жадного» замещения.

5. Программное обеспечение, реализующее разработанные алгоритмы и предназначенное для проведения численных экспериментов.

6. Результаты исследования эффективности предложенных методов на основе проведенного численного эксперимента.

**Научная новизна работы.** Новыми в работе являются:

1. «Блочный декодер жадного замещения» – создан новый эффективный метод конструирования двухмерной упаковки, который реализует жадную стратегию одновременно в разных направлениях, оперируя несколькими приоритетными списками прямоугольников; декодер может применяться в различных метаэвристиках.

2. Новый метод локального поиска оптимальной упаковки, представляющий собой генетический алгоритм с жадным блочным декодером. Показано, что такой подход дает лучшие решения и большую экономию материала по сравнению со многими известными в литературе реализациями.

3. Модификация метода последовательного уточнения оценок на случай прямоугольной упаковки блочной структуры.

4. Программное обеспечение, реализующее новые методы расчета упаковок, ориентированное на численные эксперименты и промышленные расчеты.

### **Практическая значимость работы**

Программная реализация генетического алгоритма и жадного блочного декодера показала значительные преимущества перед известными классическими способами применения эвристических методов к задачам упаковки на достаточно широком классе задач. Это позволяет использовать разработанные алгоритмы при практических расчетах и включать комплекс программ в автоматизированные системы проектирования и управления. Экономический эффект составляет от 1 до 10% экономии материала.

**Связь исследования научными проектами:** Работа выполнялась при частичной поддержке РФФИ, проект 01-01-00510.

**Апробация работы.** Результаты работы и отдельные ее разделы докладывались и обсуждались на следующих конференциях и семинарах:

- Международная конференция «Ресурсосберегающие технологии». Санкт-Петербург 2001.
- Байкальская международная школа-семинар «Математическое программирование», Иркутск, 2002.
- Международная конференция «Математическое программирование и приложения», Екатеринбург 2003.
- Международная конференция ESICUP (Европейская специальная группа по раскрою и упаковке) (Germany, 2004г.);
- Семинары института вычислительной математики Дрезденского технологического университета, 2005г.
- Семинары кафедры вычислительной математики и кибернетики Уфимского государственного авиационного технического университета.
- Конференция региональной зимней школы-семинара аспирантов и молодых ученых. «Интеллектуальные системы обработки информации и управления», Уфа, 2006г.

В 2004–2005 учебном году прошел по линии ДААД научную стажировку в институте вычислительной математики Дрезденского технического университета. Результаты работа внедрены в учебный процесс УГАТУ и в Центр обработки информации ОАО «Башнефтегеофизика», г.Уфа.

По теме диссертации опубликовано 9 работ: 5 статей (2 из них в изданиях рекомендованных ВАК на 13 машинописных страницах), 3 статьи в материалах конференций. Правовая сторона программного обеспечения защищена в Роспатент свидетельством об официальной регистрации программ для ЭВМ № 2006612649.

Выражаю глубокую благодарность научному руководителю д-ру техн. наук, проф. Элите Александровне Мухачевой и канд. физ.-мат. наук, доц. Анне Сергеевне Мухачевой–Филипповой.

### **Содержание диссертации**

**Во введении** к диссертации обоснована актуальность решаемой научной проблемы; сформулирована цель и задачи исследования; приведены результаты, выносимые на защиту; отмечена их научная новизна и

практическая значимость. Приведены сведения об апробации работы и публикациях.

**В первой** главе проведен аналитический обзор моделей и методов решения задач прямоугольной упаковки и раскроя.

На основе проведенного аналитического обзора были сформулированы следующие выводы:

- Задачи раскроя-упаковки относятся к проблемам комбинаторной оптимизации. Простейшая из них, задача линейного раскроя, является NP-трудной проблемой. Поэтому для нее и других более сложных задач разрабатываются наряду с точными, многочисленные эвристические и приближенные методы.

- Для решения задач раскроя-упаковки перспективным направлением является применение метаэвристик, в том числе эволюционных и среди них генетических алгоритмов. Анализ литературы по метаэвристикам позволил высказать следующие предположения: эффективность алгоритма зависит от выбора конкретной метаэвристики, конструирования окрестностей и в значительной мере от способа кодирования упаковки и от алгоритма-декодера, осуществляющего построение упаковки. На основе этого анализа проведены исследования в следующих главах диссертации.

**Во второй** главе приводятся различные варианты кодирования – декодирования упаковок. Рассмотрена модель блок – структуры упаковки и описаны различные варианты блочных декодеров. Предложен новый блочный декодер «жадного» замещения.

#### Декодер замещения Sub (NF), следующий подходящий

Декодер **Sub(NF)** представляет собой блочную модификацию простой эвристики **NF** (следующий подходящий). Он размещает прямоугольники последовательно по алгоритму NF, согласно приоритетному списку  $\pi$ , замещая также пустые участки между прямоугольниками в блоках. Если очередной прямоугольник не входит в указанное свободное место в блоке, то он помещается либо выше текущего свободного места, либо в следующий блок.

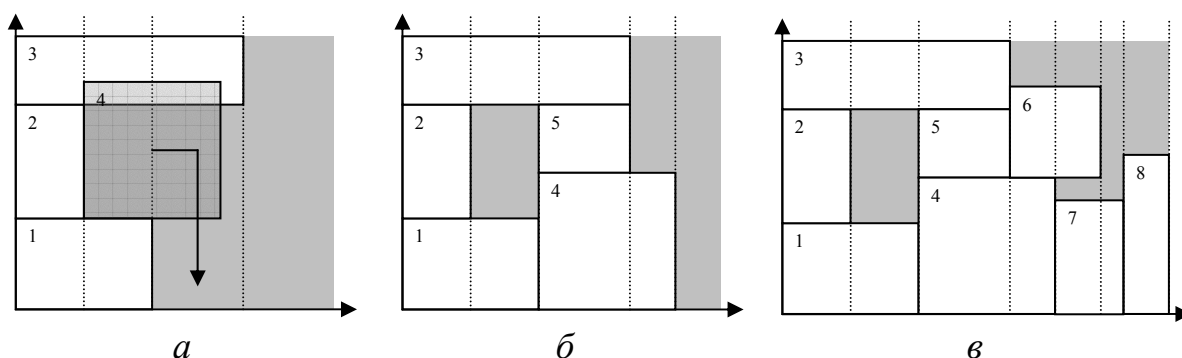


Рис. 1. Иллюстрация работы Sub(NF),  $\pi=(1, 2, 3, 4, 5, 6, 7, 8)$ :

*a* – размещены прямоугольники №1, №2, №3;

*б* – размещены прямоугольники №4 и №5;

*в* – полное размещение

Процесс добавления идет с нижней границы очередного блока. А первый блок образуется после добавления в левый нижний угол пустой полосы 1-го прямоугольника из списка  $\pi$ . Блоки заполняются последовательно, включая пустые участки, до тех пор пока не будет размещен последний прямоугольник.

Сложность декодера **Sub(NF)**  $\leq O(m^2)$

### Декодер замещения **Sub (FF)**, первый подходящий

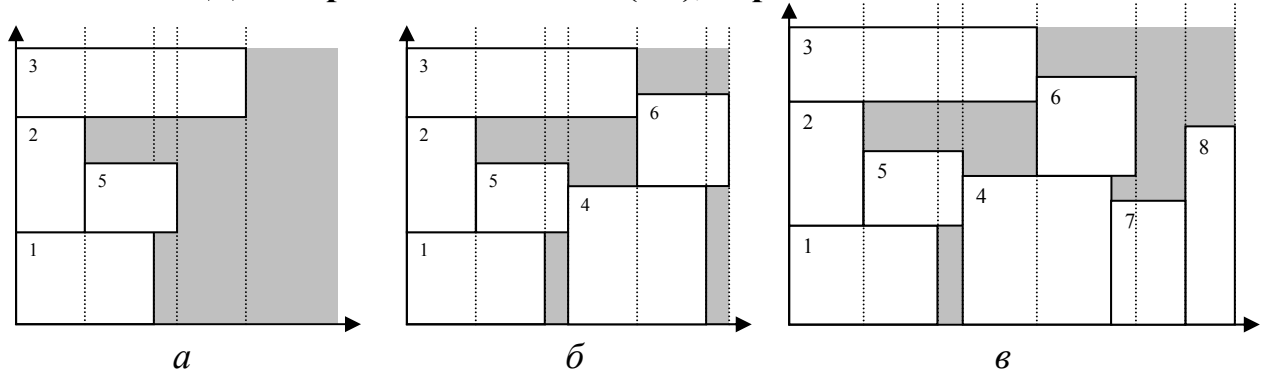


Рис. 2. Иллюстрация работы **Sub(FF)**:

*a* – размещение прямоугольников №1, №2, №3, №5;

*б* – размещение прямоугольников №4, №6;

*в* – полное размещение

Декодер размещает прямоугольники последовательно по алгоритму FF. В отличие от **Sub(NF)**, где мы для очередного прямоугольника находим подходящий блок, куда он входит по высоте, **Sub(FF)** находит первый из  $\pi$  подходящий прямоугольник для заданного пустого участка в блоке. И только в случае, если ни один прямоугольник не подошел, **Sub(FF)** переходит к следующему пустому участку.

### Декодер жадного замещения **Greedy Sub**

Декодер **Greedy Sub** представляет собой эвристический метод упаковки на основе блочного декодера замещения. Данный метод использует 2 приоритетных списка. Первый список  $\pi$  подается на вход декодера, второй список  $\pi_{\text{sort}}$  формируется путем упорядочивания прямоугольников по невозрастанию высот. Обычно в жадных конструктивных алгоритмах используется только второй список. В случае, если разрешены повороты прямоугольников, декодер использует еще один дополнительный приоритетный список  $\pi_{\text{sort}}^*$ . Он формируется путем упорядочивания прямоугольников по невозрастанию длин.

При формировании блок структуры прямоугольник вводится согласно приоритетному списку  $\pi$  в том случае, если он добавляется выше всех остальных прямоугольников в блоке, т.е. после добавления он окажется верхним в блоке. А в случае заполнения пустот между другими уже упакованными прямоугольниками используются приоритетные списки  $\pi_{\text{sort}}$  и  $\pi_{\text{sort}}^*$ . В списке  $\pi_{\text{sort}}$  декодер находит первый прямоугольник, назовем его  $r_1$ , который может заполнить заданное пустое пространство. В списке  $\pi_{\text{sort}}^*$  декодер также находит первый подходящий прямоугольник, назовем его  $r_2$ ,



предварительно перевернув его (данное действие выполняется при разрешенных поворотах). Затем из  $r_1$  и  $r_2$  выбирается прямоугольник с наибольшей высотой. Таким образом для каждого пустого участка между прямоугольниками будет найден прямоугольник с максимальной возможной высотой, заполняющей пустоту. В этом заключается принцип жадности данного декодера.

Пример работы декодера GreedySub (без поворотов) показан на рис.3.

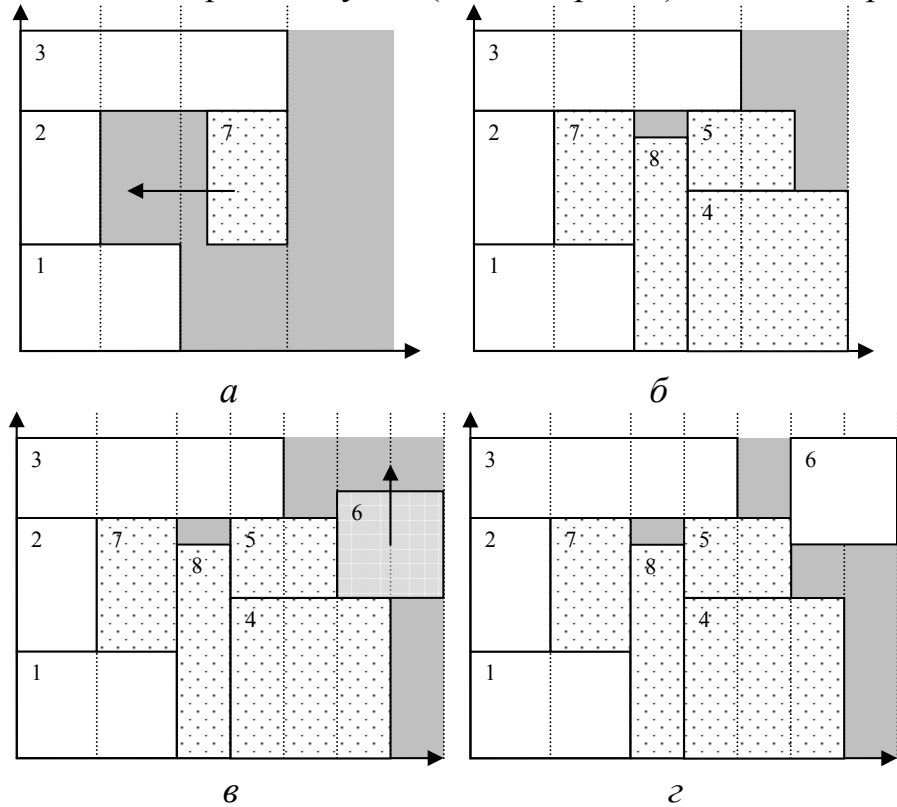


Рис. 3. Иллюстрация работы декодера Greedy Sub:

Приоритетный  $\pi=(1, 2, 3, 4, 5, 6, 7, 8)$ ;  $\pi_{\text{sort}}=(8, 4, 2, 7, 1, 6, 3, 5)$

$a$  – размещение прямоугольников №1,2,3;

$б$  – №7,8,4,5;

$в$  – сдвиг прямоугольника №6;

$г$  – полное размещение.

Также выполняется еще одна эвристическая процедура, которая немного повышает эффективность метода: после формирования очередного блока самый верхний прямоугольник сдвигается к верхней границе полосы. Это сделано с той целью, чтобы в центре полосы освободить большее пространство для размещения, см. рис. 3,в и 3,г.

Прямоугольники №1, №2 и №3 последовательно добавляются в 1-й вертикальный блок. Далее во 2-й блок подходит прямоугольник №7 из приоритетного списка  $\pi_{\text{sort}}$ . Прямоугольники, упакованные по списку  $\pi_{\text{sort}}$  обозначены штриховкой. Затем остаток 3-го блока по высоте лучше всего замещает прямоугольник №8. Далее в 4 блок – прямоугольники №4 и №5. Последним упаковывается прямоугольник №6, он взят из приоритетного списка  $\pi$ , т.к. пустой остаток, куда он упаковывается является самым верхним в блоке.

Затем прямоугольник №6 сдвигается к верхней границе полосы, образуя тем самым более широкий остаток снизу.

Рассмотрим еще один пример, где будет более наглядно видно преимущество такого сдвига (рис 4,а и 4,б).

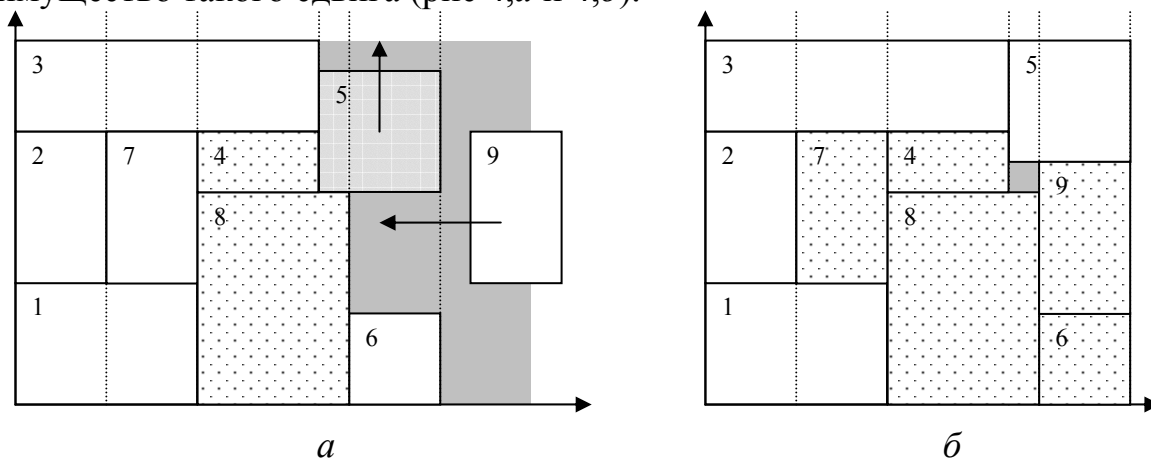


Рис. 4. Пример работы Greedy Sub:

*a* – сдвиг прямоугольника №5 освобождает место для прямоугольника №9;  
*б* – получена лучшая упаковка

#### Схема алгоритма:

##### 1. (Инициализация)

Входные данные  $\langle L, W, m, l, w, \pi \rangle$

Текущий вертикальный блок  $C=1$

##### 2. (Итерации)

Пока не упакованы все прямоугольники выполнять 2.1 – 2.3

2.1. Цикл по пустым участкам в вертикальном блоке  $C$  (снизу вверх)

2.1.1. Если пустой участок (его высота  $H$ ) самый верхний в блоке, то

Находим в списке  $\pi$  по порядку прямоугольник  $R$ , удовл. условию  $w_R \leq H$

Иначе

а) Находим в списке  $\pi$ sort по порядку прямоугольник  $R$ , удовл. условию  $w_R \leq H$

б) Если разрешены повороты, то находим в списке  $\pi$ sort по порядку прямоугольник  $R_2$ , удовл. условию  $l_{R_2} \leq H$ ;

в) Если  $l_{R_2} > w_R$ , то  $R=R_2$

2.1.2. Если нашли, то

Вставляем прямоугольник  $R$

Вычисляем длину  $l_{min}$  самого короткого прямоугольника в

блоке  $C$ .

Модифицируем блочную структуру, так что длина блока  $C$  равна  $l_{min}$

2.2. Конец цикла

2.3.  $C=C+1$

##### 3. Вывод найденного решения

Сложность декодера GreedySub  $\leq O(m^3)$ .

**Третья глава** посвящена эволюционным методам решения задач упаковки. Предложена модификация генетического алгоритма с использованием разработанного блочного декодера Greedy Sub.

### **Эволюционный алгоритм (1+1)**

Эволюционные алгоритмы (ЭА) – это обобщенный термин, применяемый для описания вычислительных методов, использующих модели эволюционных процессов. Основой всех эволюционных алгоритмов является имитация эволюции индивидуальных структур через процессы отбора, мутации и воспроизведения. ЭА рассматривают популяции структур, развивающиеся путем мутаций и воспроизведения согласно правилам отбора. Отбор происходит по критериям соответствия структуры условиям внешней среды. ЭА достаточно широко используются при решении задач многопараметрической оптимизации.

Генетические методы являются одной из разновидностей эволюционных алгоритмов. Они максимально приближены к реальным эволюционным процессам живой природы. Но существуют и более простые ЭА. В частности, к таким относится и эволюционный алгоритм (1+1). Схема его работы строится на следующем подходе:

Имеется одно закодированное решение (хромосома), которое является родителем. На каждой итерации с помощью модификации этой родительской хромосомы получается потомок. Модификация представляет собой мутацию. В случае, если целевая функция потомка достигает лучшего значения, он заменяет родителя. Доказано, что, если оператор мутации обладает свойством монотонности, то алгоритм (1+1)-EA является оптимальной стратегией поиска среди различных ЭА. Для операторов мутации применяемых в среде задач раскроя-упаковки, свойство монотонности мутации пока не доказано.

Отличия ЭА (1+1) от генетического алгоритма:

1. В (1+1)-EA отсутствует понятие популяции. Если в генетических алгоритмах все операции выполняются над множеством закодированных решений (хромосом), то в предложенном эволюционном алгоритме имеется только одна особь.

2. В (1+1) не применяется оператор скрещивания. На каждой итерации алгоритм работает только с одной особью. И единственным оператором, который к ней применим, является мутация, работающая с приоритетным списком.

Как и для генетического алгоритма, в ЭА (1+1) в качестве хромосом выступают приоритетные списки и мутации подвергаются именно они.

Мутация будет представлять собой случайную перестановку элементов внутри приоритетного списка. Но если в классическом генетическом алгоритме мутация заключалась в случайном выборе одной пары генов для обмена, то для ЭА (1+1) предлагается в процессе мутации выбирать несколько таких пар.

### **«Жадный» генетический алгоритм с декодером Greedy Sub**

Работа генетического алгоритма заключается в случайном формировании множества хромосом (перестановок прямоугольников) и выбора из них

наилучшего, т.е. того списка, который находится по заданному критерию ближе к оптимальному решению. Основными процедурами работы генетического алгоритма являются процедуры скрещивания, отбора и мутации.

На этапе скрещивания происходит формирование новых особей (перестановок прямоугольников) на основе уже существующих. При этом в новом особе-потомке находят свое отражение черты обоих родителей.

Существенную роль в общем процессе нахождения решения играет декодер, получающий карту размещения по перестановке прямоугольников: от того, насколько плотно будут расположены прямоугольники декодером, зависит значение целевой функции. В данном случае используется декодер Greedy Sub

### **Схема алгоритма Genetic Greedy Sub:**

#### **1. (Инициализация)**

Ввод данных  $\langle L, W, m, l, w, h \rangle, K$ , где  $K$  – число итераций  $k=1$ , где  $k$  – номер итерации

Генерирование первоначальной хромосомы

Запуск для декодера Greedy Sub

#### **2. (Итерации)**

Пока  $k < K$  выполнять 2.1 – 2.4

##### **2.1. Если $(k \bmod 3) = 0$ , тогда**

Генерирование случайной хромосомы PL

Иначе

Выполняем операцию скрещивания над популяцией

Берем очередную хромосому из списка потомков PL

##### **2.2. Запуск декодера Greedy Sub для выбранной хромосомы PL**

##### **2.3. Вычисление значения целевой функции (коэффициента раскрытия,**

СС)

для полученной упаковки.

##### **2.4. Если $CC(PL)$ удовлетворяет попаданию в популяцию, то добавляем PL в популяцию**

удаляем хромосому с наименьшим значением СС из популяции

##### **2.5. $k=k+1$**

#### **3. Вывод решения, соответствующего хромосоме с наибольшим СС**

##### **Решение задачи прямоугольной упаковки на листы**

Решение задачи упаковки на прямоугольные листы в данной реализации отличается от задачи упаковки в прямоугольную полосу лишь введением одного дополнительного ограничения, это ограничение – ширина листа  $L$ , в случае упаковки в полубесконечную полосу  $L = \infty$ . Длина листа  $L$  учитывается в декодере, когда тот добавляет очередной прямоугольник в блок. Если при добавлении прямоугольника суммарная длина блоков больше  $L$ , то такой прямоугольник не добавляется. В случае если ни один прямоугольник невозможно добавить в текущий лист, учитывая ограничение по длине  $L$ , то декодер заканчивает свою работу. Создается новый пустой лист и декодер

запускается заново для нового листа, куда упаковываются оставшиеся прямоугольники.

В четвертой главе приведено описание программного обеспечения и приведены результаты численных экспериментов.

### Решение задач размещения на полосу на примерах А.Бортфельда (библиотека OR-library)

Сравнение результатов работы проводилось для следующих алгоритмов и декодеров:

- Генетический алгоритм Бортфельда;
- Генетический алгоритм с декодером замещения (Смагина М.А.) GSM;
- Генетический блочный алгоритм GBA (Мухачева А.С., Чиглинцев А.В.);
- Наивный эволюционный алгоритм с декодером Greedy Sub;
- «Жадный» генетический алгоритм Genetic Greedy Sub.

Отклонение от нижней границы считалось по формуле:

$$GAP = \frac{L - L_0}{L} \cdot 100\%, \text{ где } L - \text{длина упаковки, } L_0 - \text{нижняя граница.}$$

В эксперименте указаны не только алгоритмы, но и декодеры, чтобы показать насколько эффективным в алгоритме является именно выбор декодера. Декодер **Greedy Sub** показал результаты лучше чем у большинства других алгоритмов.

Как видно из таблицы, наименьшее среднее отклонение от нижней границы по всем классам дает «жадный» генетический, Genetic Greedy Sub. 6 классов задач из 10 решены алгоритмом GGSUB лучше других.

Таблица 1.

Результаты эксперимента. Отклонения от нижней границы GAP.  
Средние значения для каждого класса.

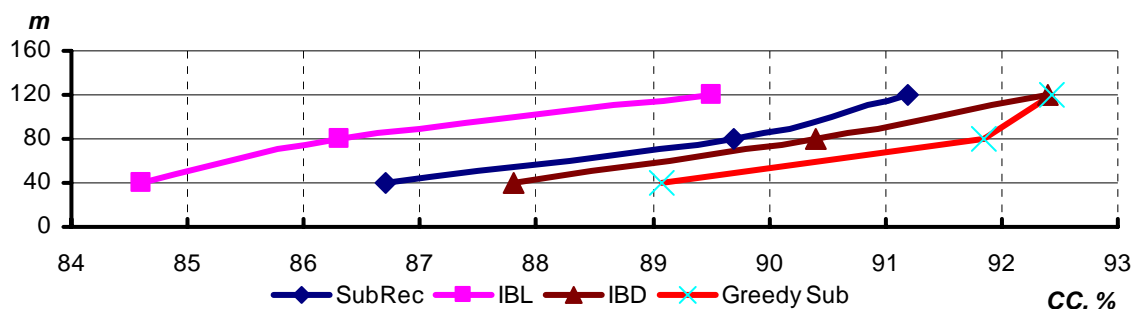
Класс	Bortfeld	GSM	GBA	GSUB	GGSUB
C1	0,75	0,96	0,89	0,65	0,59
C2	0,84	0,79	1,21	1,35	1,23
C3	2,52	1,92	2,75	2,08	1,99
C4	3,08	4,68	3,17	3,49	3,22
C5	2,53	1,94	2,77	2,13	2,08
C6	4,67	6,05	4,56	4,52	4,44
<b>среднее C1-C6</b>	<b>2,40</b>	<b>2,72</b>	<b>2,56</b>	<b>2,37</b>	<b>2,26</b>
C7	1,22	0,95	1,35	1,13	1,13
C8	3,66	5,80	4,81	3,97	3,60
C9	0,12	0,09	0,07	0,07	0,07
C10	3,04	4,20	3,92	3,66	3,19
<b>среднее C7-C10</b>	<b>2,01</b>	<b>2,76</b>	<b>2,54</b>	<b>2,21</b>	<b>2,00</b>
<b>среднее C1-C10</b>	<b>2,21</b>	<b>2,74</b>	<b>2,55</b>	<b>2,29</b>	<b>2,13</b>

### Исследование эффективности алгоритмов декодеров при использовании генетических алгоритмов

Целью эксперимента является исследование эффективности предложенных в работе жадного блочного декодера и декодеров (в том числе блочных) других авторов. Для проведения эксперимента использовалась классическая реализация генетического алгоритма, в которой на стадии декодирования упаковки из приоритетного списка использовались декодеры:

- **IBL** – улучшенный нижний левый, D.Liu & H.Teng, 1999
- **SubRec** – декодер замещения с перестройкой, А.С. Мухачева, 2004
- **IBD** – улучшенный блочный декодер, А.В. Чиглинцев, 2000
- **Greedy Sub** – «жадный» блочный декодер

Сравнение производилось на классе задач с набором предметов различных размеров от малых 5% до 95% ширины полосы. Количество прямоугольников 40, 80, 120. Примеры были сгенерированы в соответствии с методикой профессора Г.Вэшера.



Верхнее и нижнее ограничение по ширине прямоугольников равны  $v_1=0,05$  и  $v_2=0,95$  соответственно. Верхнее и нижнее ограничение по длине прямоугольников равны  $w_1=0,30$  и  $w_2=0,70$  соответственно. В каждом классе задач было сгенерировано и просчитано по 10 задач.

В качестве критерия эффективности упаковки применяется «коэффициент раскроя» (Cutting Coefficient, CC) равный отношению полезной площади ко всей затраченной. Для безотходной упаковки  $CC=100\%$ . Наилучшие результаты у «жадного» блочного декодера **GreedySub**, за ним с небольшим отставанием следует **IBD**.

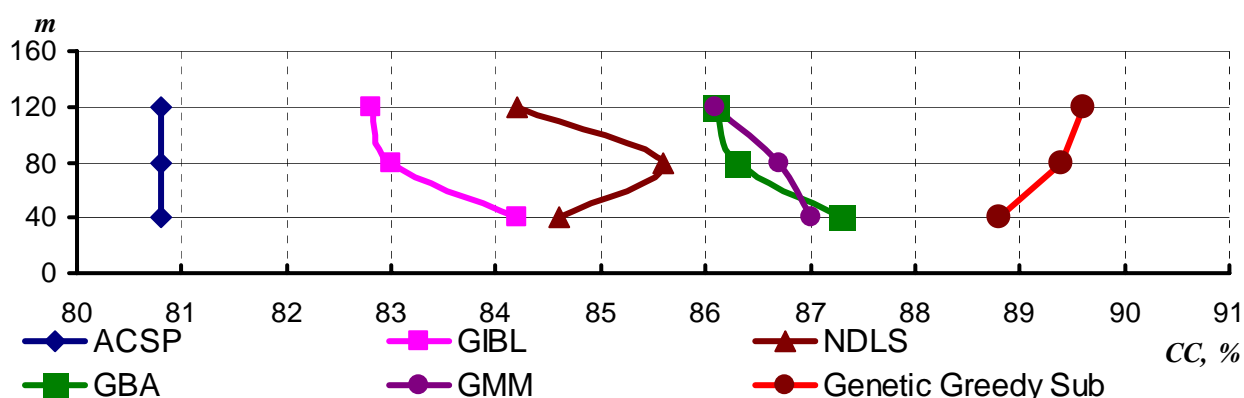
### Исследование эффективности генетического гибридного алгоритма Genetic Greedy Sub. Сравнительный эксперимент с метаэвристическими алгоритмами

Целью эксперимента является исследование эффективности разработанного генетического гибридного алгоритма с декодером «жадного замещения». Для сравнения были выбраны пять алгоритмов, реализации которых были разработаны на кафедре ВМиК УГАТУ:

- **Genetic Greedy Sub** – генетический алгоритм с декодером «жадного замещения»;
- **GVA** – генетический блочный алгоритм, А.С.Мухачева, А.В.Чиглинцев, 2000;

- **GIBL** - классический генетический алгоритм, D.Liu & H.Teng, 1999 (программная реализация М.А. Смагина);
- **GMM** – генетический мультиметодный алгоритм, И.П. Норенков, 2000 (программная реализация Л. Ильясова);
- **ACSP** – алгоритм муравьиной колонии, А.Ф. Валеева, М.Н. Аглиуллин, 2003 (программная реализация М.Н. Аглиуллин);
- **NDLS** – алгоритм наивного поиска двойственной блок-структуры, А.С. Мухачева, Р.Р. Ширгазин, 2002;

Сравнение производилось на классе сложных задач (триплеты)  $v_1=0,3$ ;  $v_2=0,4$ ;  $w_1=0,35$ ;  $w_2=0,45$ . Ранее было замечено, что характеристики по длине мало влияют на результативность, при решении задачи на полосу с запретом поворотов, поэтому решение задач выполнялось с поворотами.



Среди генетических алгоритмов лидирует Genetic Greedy Sub, далее идут «блочный генетический» и «мультиметодный генетический» алгоритмы.

### Решение задач размещения на листы на примерах С.П. Фекете и Я.Шеперс

Данный эксперимент выполнялся для оценки эффективности различных алгоритмов решения задач размещения на листы. Использовались примеры задач S.P. Fekete и J. Schepers. Критерием сравнения результатов также являлось отклонение от нижней границы, рассчитанной алгоритмом этих же авторов. Примеры тестовых задач взяты из библиотеки OR-library.

Площадь листа во всех задачах равна  $100 \times 100$ . Для каждой задачи размерности  $n=40,50,100,150,250,500,1000$  было решено по 10 примеров.

Таблица 2.

Результаты эксперимента с примерами С.П.Фекете и Я. Шеперс.

Класс задач	Отклонение от нижней границы, %		
	Наивный перебор (1+1) – EA	Алгоритм муравьиной колонии (ACP)	Наивный перебор (Greedy Sub)
1	2,93	2,88	1,87
2	2,33	2,23	1,37
3	1,73	1,59	0,88

Отклонение от нижней границы результатов полученных декодером Greedy Sub на всех классах задач меньше по сравнению с результатами других авторов. Кроме хороших результатов, полученных декодером Greedy Sub, он еще оставляет возможность улучшения своих показателей, путем встраивания его в другие эвристические алгоритмы раскроя-упаковки, как это было сделано в «жадном» генетическом алгоритме Genetic Greedy Sub, реализованном в рамках данной диссертации.

### **ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ И ВЫВОДЫ**

1. Представлена обобщенная модель задач прямоугольной упаковки, которая позволила реализовать новые эффективные методы решения задачи прямоугольного раскроя и упаковки

2. Модифицированы однопроходные эвристические методы конструирования упаковок: следующий подходящий, NF; первый подходящий FF; лучший подходящий, BF; с использованием блок-структур; на базе этих алгоритмов разработаны высокопроизводительные варианты метода замещения, обеспечивающие эффективное конструирование рациональных упаковок.

3. Разработан новый декодер «жадного» замещения Greedy Sub на базе блочной структуры и его модификации. Впервые в работе декодера стратегия жадности реализована в разных направлениях, используя несколько приоритетных списков прямоугольников. Его эффективность превосходит аналогичные показатели других алгоритмов.

4. Разработан «наивный» эволюционный алгоритм (1+1)-EA с различными операторами мутации и декодерами. Показана его высокая качественная и временная эффективность с декодером «первый подходящий» и «наивным» оператором мутации.

5. Разработан генетический алгоритм на базе блочных структур для решения задачи прямоугольной упаковки полубесконечной полосы и на листы. Это позволило осуществить гибридизацию алгоритма с другими эвристиками блочной структуры и выбрать наиболее подходящую из них – жадный декодер замещения

6. Создано программное обеспечение, реализующее разработанные алгоритмы и декодеры и предназначенное для проведения вычислительных экспериментов.

7. Проведены численные эксперименты. Результаты (коэффициенты упаковки и временные показатели) разработанных алгоритмов, в сравнении с другими методами, оказались лучше на 5 – 10% на различных классах задач. Эти результаты показывают преимущество разработанного декодера и генетического алгоритма и позволяют сделать заключение об эффективности предложенных методов для решения задач ортогональной упаковки в полубесконечную полосу и на листы.



## СПИСОК ПУБЛИКАЦИЙ

### *Публикации в изданиях рекомендованных ВАК*

1. Методы локального поиска оптимума прямоугольной упаковки с использованием двойственной схемы / А.С. Мухачева, Р.Р. Ширгазин, М.А. Смагин, С.Х. Куреленков // Информационные технологии. М. : Машиностроение, 2002. №10. С. 26–31.

2. Задачи упаковки прямоугольников: рандомизированная эвристика на базе двойственной схемы локального поиска оптимума / А.С. Мухачева, Р.Р.Ширгазин // Информационные технологии. М. : Машиностроение, 2003. №5. С.18–23.

### *Другие публикации*

3. Двойственный метод локального поиска оптимума в задаче прямоугольной упаковки / А.С.Мухачева, Р.Р.Ширгазин // ОПТИМ–2001. Ресурсосберегающие технологии: математическое обеспечение оптимизационных задач в системах автоматизированного проектирования: материалы конференции. СПб.: ЦНИИ технолог. судостроения, 2001. С.113–118.

4. Рандомизированный алгоритм на базе двойственной схемы решения задачи прямоугольной упаковки / А.С.Мухачева, Р.Р.Ширгазин, Д.С.Санников // Информационный бюллетень ассоциации математического программирования. Екатеринбург: УрО РАН, 2003. №1. С. 186–188.

5. Задача двумерной упаковки в полосу:: Численный эксперимент на безотходных примерах с использованием алгоритмов блок-структуры / Г.Белов, А.В.Чиглинцев, А.С.Филиппова, Э.А.Мухачева, Г.Шейтхауэр, Р.Р.Ширгазин. // МАТН-НМ-01. Дрезден: Техн. ун. Дрездена, 2005. С. 17–25. (на англ. языке).

6. Двойственная схема прямоугольной упаковки: конструирование быстрых алгоритмов / А.С.Мухачева, Р.Р.Ширгазин, С.Х.Куреленков // Принятие решений в условиях неопределенности: сб.статей. Уфа: УГАТУ, 2002. С.131–138.

7. Упаковка прямоугольников в полубесконечную полосу: декодеры пары последовательностей и замещения / Р.Р.Ширгазин, К.В.Ильина // Принятие решений в условиях неопределенности: сб.статей. Уфа: УГАТУ, 2005. С. 82–88.

8. Проектирование ортогональных упаковок с применением гибридного алгоритма блочной технологии / Р.Р.Ширгазин // Интеллектуальные системы обработки информации и управления: сб. статей регион. зимн. шк.-сем. аспирантов и молодых ученых. Уфа: Технологии, 2006. С. 149–154.

9. Свид. об офиц. рег. программы для ЭВМ № 2006612649. Программа расчета прямоугольных упаковок (2DPack) / Р.Р. Ширгазин, А.С. Филиппова. М: Роспатент, 2006.