

УДК 519.8

ЭВОЛЮЦИОННО-ФРАГМЕНТАРНАЯ МОДЕЛЬ УПАКОВКИ ПЕНТАМИНО

И. В. Козин, С. И. Полюга

Аннотация. Рассматривается задача упаковки пентамино в плоские контейнеры произвольной формы. Введено понятие фрагментарной структуры. Показано, что задача упаковки пентамино может быть представлена как задача на фрагментарной структуре. Предложена эволюционная модель для отыскания оптимальных упаковок пентамино.

Ключевые слова: упаковка пентамино, фрагментарная структура, эволюционно-фрагментарная модель, эволюционный алгоритм.

Введение

Одной из наиболее известных задач дискретной оптимизации является задача упаковки. В общем случае речь идёт об изометрическом вложении подмножеств n -мерного евклидова пространства в заданное подмножество-контейнер так, чтобы внутренности образов не пересекались между собой [10]. Для многих вариантов задачи упаковки доказана NP-трудность [2]. На сегодняшний день не известно алгоритма полиномиальной трудоёмкости для отыскания оптимального решения этих задач. Все точные алгоритмы так или иначе сводятся к полному перебору вариантов упаковки. Поэтому актуальным является исследование приближённых методов поиска оптимальных решений задачи упаковки. Перспективным направлением в этой области является разработка алгоритмов, основанных на метаэвристиках, в частности, эволюционных моделей [3, 8].

1. Невыпуклый целочисленный плоский раскрой. Пентамино

Изыскным примером задачи целочисленной упаковки двумерных объектов является известная игра — головоломка «Пентамино» [1].

«Пентамино» — фигура, составленная из пяти единичных квадратов. Возможны 12 попарно не изометричных фигур пентамино (рис. 1). В игре требуется выложить из всех фигур пентамино определённую фигуру

на плоскости. При этом внутренности укладываемых фигур пентамино не должны пересекаться.

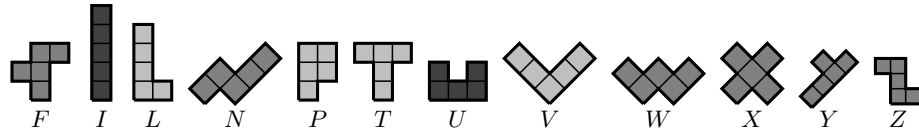


Рис. 1. Фигуры пентамино

Задачу упаковки пентамино рассмотрим в следующей постановке. В качестве контейнера P выбираются плоские фигуры, состоящие из единичных квадратов. Допустимой укладкой фигуры пентамино в контейнер считается её изометричное вложение в координатную плоскость \mathbb{R}^2 такое, что

- (i) любая из сторон составляющих её единичных квадратов параллельна одной из координатных осей;
- (ii) вершины всех единичных квадратов, составляющих образ укладываемой фигуры, совпадают с узлами целочисленной решётки плоскости.

Пусть задана допустимая укладка 12 фигур пентамино на плоскость. Будем считать допустимыми преобразованиями сдвиги образов фигур на целочисленный вектор, вращения образов фигур пентамино на углы, кратные 90° , и отражения относительно координатных осей. Тогда с точностью до сдвига на целочисленный вектор фигуры L , N , P , F и Y могут быть уложены 8 способами каждая; фигуры Z , T , V , U и W могут быть уложены 4 способами каждая; фигура I может быть уложена двумя способами; фигура X может быть уложена единственным способом (рис. 2). Любое другое вложение может быть получено из перечисленных 53 допустимых вложений путём сдвига на целочисленный вектор.

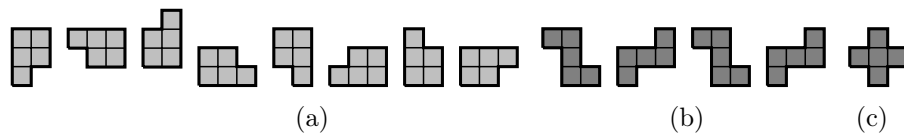


Рис. 2. Различные размещения некоторых фигур пентамино: (a) 8 допустимых укладок; (b) 4 допустимые укладки; (c) 1 допустимая укладка

Упаковкой пентамино будем называть допустимые укладки фигур пентамино в множество-контейнер такие, что образы фигур не выходят за границы контейнера и внутренности укладок отдельных фигур попарно не пересекаются. Критерием упаковки является плотность, т. е.

отношение суммарной площади, занятой уложенными в контейнер фигурами, к площади контейнера. Очевидно, что плотность любой упаковки не превосходит 1. Несмотря на широкую известность и простоту формулировки существующие точные методы поиска решений этой задачи так или иначе сводятся к перебору всех вариантов размещения фигур [14]. В настоящей работе для поиска решений задачи упаковки пентамино предлагается эволюционная модель, основанная на понятии фрагментарной структуры [5, 6].

2. Фрагментарная структура и фрагментарный алгоритм

Определение 1. *Фрагментарной структурой (X, E) на конечном множестве X называется семейство его подмножеств $E = \{E_1, E_2, \dots, E_n\}$ такое, что для любого $E_i \in E \neq \emptyset$ найдётся $e \in E_i : E_i \setminus \{e\} \in E$. Элементы из E будем называть *допустимыми фрагментами*.*

Определение 2. *Одноэлементные множества, являющиеся допустимыми фрагментами, будем называть *элементарными фрагментами*.*

Определение 3. *Фрагмент называется *максимальным*, если он не является подмножеством никакого другого фрагмента.*

Очевидны следующие свойства фрагментов.

Свойство 1. *Пустое множество является допустимым фрагментом: $\emptyset \in E$.*

Свойство 2. *Пусть $M = \max_{i=1, n} |E_i|$. Тогда для любого целого числа m в интервале $0 \leq m \leq M$ найдётся элемент в множестве E , мощность которого равна m .*

Элементы любого конечного множества X можно занумеровать числами $1, 2, \dots, n$. Тем самым исследование фрагментарной структуры на X можно свести к исследованию фрагментарной структуры на конечном множестве натуральных чисел $\{1, 2, \dots, n\}$.

Теорема 1. *Если (X, E) — фрагментарная структура на множестве X , то для любого непустого множества $A \in E$ существует нумерация его элементов $A = \{x_1, x_2, \dots, x_m\}$ такая, что $\{x_1, x_2, \dots, x_k\} \in E$ при любом $k = 1, 2, \dots, m$.*

ДОКАЗАТЕЛЬСТВО. Положим $A_0 = A$. Пусть $A \in E$ и $|A| = n > 0$. Выберем элемент $x_n \in A_0$ так, что $A_1 = A_0 \setminus \{x_n\} \in E$. Повторив эту процедуру $n - 1$ раз по отношению к множествам A_1, A_2, \dots, A_n , получим требуемую нумерацию элементов множества A . Теорема 1 доказана.

Из теоремы вытекает, что всякий допустимый фрагмент можно построить из пустого множества, последовательно добавляя к нему элементы так, что на каждом шаге получаем допустимый фрагмент.

Максимальный фрагмент может быть построен с помощью следующего «жадного» алгоритма:

- (i) элементы множества X линейно упорядочиваются;
- (ii) на начальном шаге выбирается пустое множество $X_0 = \emptyset$;
- (iii) на шаге с номером $k + 1$ выбирается первый по порядку элемент $x \in X \setminus X_k$ такой, что $X_k \cup \{x\} \in E$;
- (iv) алгоритм заканчивает работу, если на очередном шаге не удалось найти элемента $x \in X \setminus X_k$ с требуемым свойством.

Приведённый выше алгоритм построения максимального фрагмента во фрагментарной структуре будем называть *фрагментарным алгоритмом*. Результат применения фрагментарного алгоритма определяется заданным линейным порядком на множестве X . Таким образом, любой максимальный фрагмент может быть задан некоторой перестановкой элементов множества X . Пусть $A \in E$. Условие для элемента $x \in X$, при котором $A \cup \{x\} \in E$, будем называть *условием присоединения элемента x* .

Теорема 2. Если для любых $A \in E$ и $x \in A$ существует алгоритм полиномиальной трудоёмкости по числу элементов множества X для проверки условия присоединения элемента x , то задача построения максимального фрагмента полиномиально разрешима.

ДОКАЗАТЕЛЬСТВО. Пусть трудоёмкость проверки условия присоединения составляет $O(n^k)$, где k — натуральное число. Трудоёмкость упорядочения элементов может быть оценена величиной $O(n \ln(n))$. Тогда для фрагментарного алгоритма построения максимального фрагмента трудоёмкость оценивается величиной $O(n \ln(n) + n^{k+1})$, т. е. $O(n^{k+1})$. Теорема 2 доказана.

Пусть задана фрагментарная структура (X, E) на конечном множестве X . Рангом подмножества $A \subseteq X$ будем называть число $\text{rg}(A)$, равное максимальной из мощностей допустимых фрагментов, содержащихся в A .

Очевидными являются следующие свойства ранга: $0 \leq \text{rg}(A) \leq |A|$ для любого подмножества $A \subseteq X$ и $\text{rg}(A) \leq \text{rg}(B)$ для любых подмножеств $A \subseteq B \subseteq X$.

3. Матроиды, гридоиды, наследственные системы

Существует естественная связь [7] между фрагментарными структурами и такими известными математическими объектами, как матроиды [15], гридоиды [11] и наследственные системы [4].

Определение 4. *Матроидом M называется пара (X, B) , где X — непустое конечное множество, B — непустая совокупность его подмножеств (называемых *базами*), удовлетворяющая условиям [15, 16]:*

- (a) никакая база не содержит в качестве собственного подмножества другую базу;
- (b) если B_1 и B_2 — базы и e — любой элемент из B_1 , то существует элемент f из B_2 такой, что множество $(B_1 \setminus \{e\}) \cup \{f\}$ также является базой.

Применяя достаточное число раз свойство (b), можно показать, что любые две базы матроида M содержат одинаковое число элементов. Это число называется *рангом матроида M* . Любое подмножество элемента базы матроида называется *независимым множеством*. Таким образом, элементы базы матроида — это максимальные по включению независимые множества. Легко заметить, что любой матроид (X, B) является фрагментарной структурой на множестве X , в которой допустимыми фрагментами являются независимые множества матроида. Пусть каждому элементу $x \in X$ приписан неотрицательный вес $\rho(x) : X \rightarrow \mathbb{R}_+$. *Весом множества* будем называть сумму весов его элементов. Пусть (X, B) — матроид. Рассмотрим задачу отыскания множества из базы матроида, вес которого минимален. Эта задача может быть решена следующим жадным алгоритмом:

- (i) элементы множества X линейно упорядочиваются по возрастанию весов;
- (ii) на начальном шаге выбирается пустое множество $X_0 = \emptyset$;
- (iii) на шаге с номером $k + 1$ выбирается первый по порядку элемент $x \in X \setminus X_k$ такой, что $X_k \cup \{x\}$ — независимое множество;
- (iv) алгоритм заканчивает работу, если на очередном шаге не удалось найти элемента $x \in X \setminus X_k$ с требуемым свойством, т. е. построено максимальное по включению независимое множество или элемент базы матроида.

Если мощность элементов базы матроида равна k , то предлагаемый алгоритм сходится ровно за k шагов.

Для матроидов доказано, что предложенный алгоритм точен, т. е. он всегда приводит к элементу базы матроида минимального веса. Бо-

лее того, точность предложенного алгоритма является критерием для матроида в следующем смысле [15].

Теорема 3. Пусть задано семейство непустых подмножеств M конечного множества X , причём для всякого $A \in M$ любое непустое подмножество $B \subseteq A$ также принадлежит семейству M . Семейство множеств M является системой независимых множеств некоторого матроида тогда и только тогда, когда для любой функции весов $\rho : X \rightarrow \mathbb{R}^1$ описанный выше «жадный» алгоритм строит максимальное по включению множество минимального веса.

Рассмотрим матроид $M = (X, B)$ как фрагментарную структуру. Пусть $A \subseteq X$. Из определения ранга подмножества вытекает, что

(а) $0 \leq \text{rg}(A) \leq |A|$ для любого подмножества $A \subseteq X$ (заметим, что равенство $\text{rg}(A) = |A|$ имеет место в том и только том случае, когда множество A независимо);

(б) если $A \subseteq B \subseteq X$, то $\text{rg}(A) \leq \text{rg}(B)$.

Связь между фрагментарными структурами и матроидами определяет

Теорема 4. Для того чтобы фрагментарная структура (X, E) была матроидом, необходимо и достаточно, чтобы

$$\text{rg}(A \cup B) + \text{rg}(A \cap B) \leq \text{rg}(A) + \text{rg}(B)$$

для любых $A, B \subseteq X$.

ДОКАЗАТЕЛЬСТВО. НЕОБХОДИМОСТЬ вытекает из определения и свойств матроида.

ДОСТАТОЧНОСТЬ. Пусть задана фрагментарная структура (X, E) , причём для любых $A, B \subseteq X$ имеет место неравенство

$$\text{rg}(A \cup B) + \text{rg}(A \cap B) \leq \text{rg}(A) + \text{rg}(B).$$

Подмножество $A \subseteq X$ будем называть *независимым*, если $\text{rg}(A) = |A|$. Покажем, что любое подмножество независимого множества независимо. Пусть A — непустое независимое множество и $e \in A$. Обозначим $B = A \setminus \{e\}$. Очевидно, что $\text{rg}(B) \leq |B| = |A| - 1$. С другой стороны, $|A| = \text{rg}(A) = \text{rg}(B \cup \{e\}) \leq \text{rg}(B) + \text{rg}(\{e\}) \leq \text{rg}(B) + 1$, значит, $\text{rg}(B) \geq |A| - 1$. Таким образом, $\text{rg}(B) = |B|$, значит, множество B независимо. Повторяя эти рассуждения нужное число раз, получаем, что любое подмножество независимого множества независимо. Пусть A, B — два независимых множества такие, что $|A| = k$, $|B| = k + 1$. Тогда найдётся

элемент из $e \in B \setminus A$ такой, что множество $A \cup \{e\}$ независимо. Действительно, пусть для любого $e \in B \setminus A$ множество $A \cup \{e\}$ не является независимым, т. е. $\text{rg}(A \cup \{e\}) = k$. Пусть $f \in B \setminus A \setminus \{e\}$. Тогда

$$\text{rg}(A \cup \{e\} \cup \{f\}) \leq \text{rg}(A \cup \{e\}) + \text{rg}(A \cup \{f\}) - \text{rg}(A) = k.$$

Продолжая эту процедуру, приходим к выводу, что $\text{rg}(A \cup B) = k$, что противоречит условию независимости множества B . Таким образом, можно любое независимое множество дополнить до множества максимальной мощности, причём мощности всех максимальных по включению независимых множеств равны. Следовательно, если $A, B \in X$ — максимальные независимые множества, то для любого $e \in A$ найдётся элемент $f \in B$ такой, что $(A \setminus \{e\}) \cup \{f\}$ будет максимальным независимым множеством. Теорема 4 доказана.

Естественным обобщением понятия матроида является гридоид. Одно из определений гридоида имеет следующий вид [11].

Определение 5. *Гридоидом* L называется пара (X, B) , где X — непустое конечное множество, а B — непустая совокупность его подмножеств (эти подмножества называются далее *допустимыми*), удовлетворяющая условиям:

- (i) пустое множество допустимо: $\emptyset \in B$;
- (ii) для всякого непустого допустимого множества $Y \in B$ найдётся такой элемент $y \in Y$, что разность $Y \setminus \{y\}$ — допустимое множество, т. е. $Y \setminus \{y\} \in B$;
- (iii) для любых двух допустимых множеств $Y, Z \in B$ таких, что $|Y| < |Z|$, найдётся элемент $z \in Z \setminus Y$ такой, что $Y \cup \{z\}$ — допустимое множество.

Из определения гридоида следует, что гридоид является фрагментарной структурой. Допустимые множества гридоида суть допустимые фрагменты соответствующей фрагментарной структуры. Если каждому элементу $x \in X$ сопоставлен его вес $\rho(x) : X \rightarrow \mathbb{R}_+$, то гридоид будем называть *взвешенным*. При этом вес любого подмножества в X будем определять аддитивно как сумму весов элементов этого подмножества. Задача отыскания максимального допустимого множества гридоида, вес которого минимален, может быть решена жадным алгоритмом, аналогичным приведённому выше:

- (a) элементы множества X линейно упорядочиваются по возрастанию весов;
- (b) на начальном шаге выбирается пустое множество $X_0 = \emptyset$;

(с) на шаге с номером $k + 1$ выбирается первый по порядку элемент $x \in X \setminus X_k$ такой, что $X_k \cup \{x\}$ — допустимое множество гридоида;

(d) алгоритм заканчивает работу, если на очередном шаге не удалось найти элемента $x \in X \setminus X_k$ с требуемым свойством, т. е. построено максимальное по включению допустимое множество гридоида.

Для гридоида жадный алгоритм поиска максимального допустимого множества минимального веса также является точным.

Очевиден следующий критерий, связывающий гридоиды с фрагментарными структурами.

Для того чтобы фрагментарная структура (X, E) была гридоидом, необходимо и достаточно, чтобы для любых $A, B \in E$ таких, что $|A| < |B|$, существовал элемент $x \in B \setminus A$ такой, что $A \cup \{x\} \in E$.

Пусть U — конечное множество и $A \subset 2^U$ — непустое семейство его подмножеств, удовлетворяющее следующей аксиоме наследственности: $a' \subset a \Rightarrow a' \in A$ для любого $a \in A$. Пара $S = (U, A)$ называется *системой независимости*, или *наследственной системой на U* . Множества семейства A называются *независимыми*, все остальные подмножества U — *зависимыми* [4].

Очевидно, что любая наследственная система является фрагментарной структурой. Допустимыми фрагментами в этой структуре являются независимые множества наследственной системы. В наследственной системе максимальное по включению независимое множество может быть построено жадным алгоритмом, аналогичным приведённым выше. Однако построить максимальное по включению независимое множество минимального веса в наследственной системе с помощью жадного алгоритма в общем случае уже не удаётся.

Связь фрагментарных структур с наследственными системами устанавливается следующим очевидным критерием.

Для того чтобы фрагментарная структура (X, E) была наследственной системой, необходимо и достаточно, чтобы $A \setminus \{x\} \in E$ для любых $A \in E$, $A \neq \emptyset$, и $x \in A$.

Фрагментарную структуру (X, E) будем называть *взвешенной*, если задана неотрицательная функция-вес $\rho(x) : X \rightarrow \mathbb{R}_+$. Вес любого подмножества определяется как сумма весов его элементов. Значение $\rho(A)$ этой функции на множестве A будем называть *весом множества A* . В отличие от матроидов и гридоидов на взвешенных фрагментарных структурах жадный алгоритм, вообще говоря, не позволяет построить

максимального по включению допустимого фрагмента минимального веса. Однако этот алгоритм может рассматриваться как эвристический для отыскания приближённых решений задачи.

4. Эволюционная модель на фрагментарной структуре и ЭВФ-алгоритм

Рассмотрим теперь задачу поиска максимального по весу фрагмента на взвешенной фрагментарной структуре. Такой фрагмент в дальнейшем будем называть *оптимальным*. Как отмечалось выше, любой максимальный фрагмент однозначно определяется некоторой перестановкой элементов множества X . Кроме того, всякая перестановка элементов множества X определяет некоторый максимальный фрагмент. Таким образом, задача поиска оптимального фрагмента — оптимизационная задача на множестве перестановок. Рассмотрим эволюционную модель на перестановках для поиска приближённых решений этой задачи.

Эволюционные (генетические) алгоритмы подробно рассматривались в [3, 9, 12]. Для ряда оптимизационных задач удалось предложить достаточно эффективные процедуры поиска оптимальных решений, основанные на применении алгоритмов такого типа. Чтобы реализовать эволюционный алгоритм, необходимо выделить ряд объектов и процедур, совокупность которых будем называть *эволюционной моделью*. Основные составляющие эволюционной модели на перестановках следующие.

1) Базовое множество решений X — множество перестановок, на котором осуществляется поиск оптимального решения. Конечные непустые подмножества множества X будем называть *популяциями*.

2) Критерий — функция, заданная на базовом множестве X со значениями в \mathbb{R}^1 . Фактически критерий — это правило, которое позволяет по заданной перестановке элементов множества X установить значение целевой функции.

3) Оператор построения начальной популяции — это оператор, который выделяет на множестве X его непустое подмножество $Y_0 \in X$.

4) Оператор селекции — правило выбора, которое позволяет выделить для любой популяции $Y \subseteq X$ множество пар элементов-родителей в этой популяции для последующего выполнения операции кроссовера.

5) Оператор кроссовера (скрещивания) $Kr : X \times X \rightarrow X$, который по двум решениям-родителям строит новое решение-потомок (или несколько таких решений) из множества X .

6) Оператор мутации $M_\alpha : X \rightarrow X$, где $\alpha \in [0, 1]$ — вероятность мутации. Для любой популяции Y множество потомков этой популяции

получается путём последовательного применения правил селекции, кроссовера и мутации.

7) Оператор отбора — правило выбора, позволяющее получить новую популяцию из объединения множеств родителей и потомков. Обычно это правило задаётся с помощью указания размера (числа элементов) популяции и линейного порядка или функции — критерия селекции. Элементы объединения множеств родителей и потомков упорядочиваются и выбираются те, у которых значение критерия наибольшее.

8) Правило остановки.

Приведём краткое описание одного из вариантов эволюционного алгоритма для поиска приближённых решений задачи оптимизации в рамках рассматриваемой эволюционной модели. Шаг эволюционного алгоритма, который будем называть в дальнейшем *шагом эволюции*, состоит в следующем.

Пусть на k -м шаге определена текущая популяция, которую будем называть k -м поколением $Y_k \subseteq X$. На начальном шаге путём применения оператора начальной популяции находится популяция Y_0 . С помощью правила селекции выбирается множество пар-родителей $P_k \subseteq Y_k \times Y_k$ в текущей популяции Y_k . К каждой родительской паре применяется оператор кроссовера и строится множество потомков $C_k = \{z \mid z = Kr(x, y)\}_{x, y \in P_k}$. Ко всем элементам множества C_k применяется правило мутации. А именно, каждый элемент $c \in C_k$ с вероятностью α заменяется элементом $c' = M_\alpha(c)$ или с вероятностью $1 - \alpha$ остаётся без изменений. Таким путём строится множество потомков C'_k .

Далее применяется оператор отбора. Элементы объединения множеств $Y_k \cup C'_k$ упорядочиваются. Первые m (m — размер популяции) из них по значению критерия образуют популяцию Y_{k+1} .

Проверяется условие остановки. Если условие не выполнено, то осуществляется переход к очередному шагу алгоритма. При выполнении условия остановки алгоритм заканчивает работу. Приближённым решением задачи считается лучшее по значению целевой функции решение в последней популяции.

Благодаря свойствам фрагментарных структур можно построить особый класс эволюционных алгоритмов на фрагментарных структурах — ЭВФ-алгоритмы. ЭВФ-алгоритм является комбинацией эволюционного и фрагментарного алгоритмов. Опишем соответствующую эволюционную модель и принцип действия такого алгоритма [6]. В качестве множества допустимых решений рассматривается подмножество максимальных фрагментов на заданной фрагментарной структуре. Каждый фраг-

мент из этого множества определяется как результат работы фрагментарного алгоритма при некоторой заданной перестановке элементарных фрагментов. Таким образом, любому допустимому решению соответствует определённая перестановка чисел $1, 2, \dots, N$, где N — количество элементарных фрагментов. Для каждого допустимого решения определено значение целевой функции. Базовое множество X эволюционной модели — это множество S_N всех перестановок чисел $1, 2, \dots, N$. Оператор построения начальной популяции выделяет произвольное подмножество заданной мощности m из множества X . Правило вычисления критерия селекции устроено следующим образом: по заданной перестановке фрагментов с помощью фрагментарного алгоритма строится максимальный допустимый фрагмент и вычисляется значение целевой функции задачи для этого фрагмента.

Опишем теперь оператор кроссовера. Пусть $U = (u_1, u_2, \dots, u_N)$ и $V = (v_1, v_2, \dots, v_N)$ — две произвольные перестановки. Перестановка-потомок строится следующим образом. Последовательности U и V просматриваются в порядке следования элементов. На k -м шаге выбирается наименьший из первых элементов последовательностей и добавляется в новую перестановку — потомок. Затем этот элемент удаляется из двух последовательностей-родителей. Например,

$$Cr((2, 3, 4, 7, 8, 1, 6, 5), (3, 4, 6, 2, 1, 5, 8, 7)) = (2, 3, 4, 6, 1, 5, 7, 8).$$

Оператор мутации M_α выполняет случайную транспозицию в перестановке. Оператор селекции выбирает случайным образом набор пар из текущей популяции. Оператор отбора упорядочивает элементы промежуточной популяции в последовательность по убыванию значения критерия селекции. В качестве новой текущей популяции выбираются первые m элементов последовательности. Обычное правило остановки — количество поколений достигло предельного значения L . Лучшая по значению критерия селекции перестановка из последней построенной популяции определяет приближённое решение задачи.

Предложенный подход является универсальным и позволяет применять один и тот же эволюционный алгоритм к любым оптимизационным задачам на конечных фрагментарных структурах.

5. Фрагментарная модель задачи упаковки пентамино

Построим фрагментарную модель задачи упаковки пентамино. Множество X содержит всевозможные допустимые укладки 12 фигур пентамино, каждая из которых состоит из 5 единичных квадратов, с точностью до сдвига на целочисленный вектор. Таким образом, элементарным

фрагментом является любой из 53 элементов, каждому из которых соответствует размещение одной из фигур пентамино на плоской целочисленной решётке. Элементарный фрагмент присоединяется в упаковку, если одновременно выполнены следующие условия:

образы вершин укладки очередного элементарного фрагмента совпадают с узлами целочисленной решётки контейнера;

внутренность его не пересекается с внутренностями уже уложенных элементарных фрагментов;

образ очередного фрагмента не может быть получен допустимыми преобразованиями (сдвиги, отражения и повороты) из образа уже уложенного фрагмента.

Укладка очередного фрагмента происходит по правилу top-left, т. е. левый верхний квадрат фрагмента займёт крайнюю из возможных левых верхних позиций в контейнере (рис. 3).

Пусть множество-контейнер состоит из 60 единичных квадратов. Тогда верна

Теорема 5. Если существует оптимальная упаковка 12 различных фигур пентамино с плотностью, равной 1, то эта упаковка может быть получена путём применения фрагментарного алгоритма при некотором упорядочении элементарных фрагментов.

Доказательство. Для доказательства рассмотрим решение с плотностью 1. Перенумеруем фрагменты, входящие в это решение, следуя правилу top-left. Продолжим нумерацию остальных фрагментов (не входящих в решение) в произвольном порядке. Полученная нумерация фрагментов даёт требуемое упорядочение.

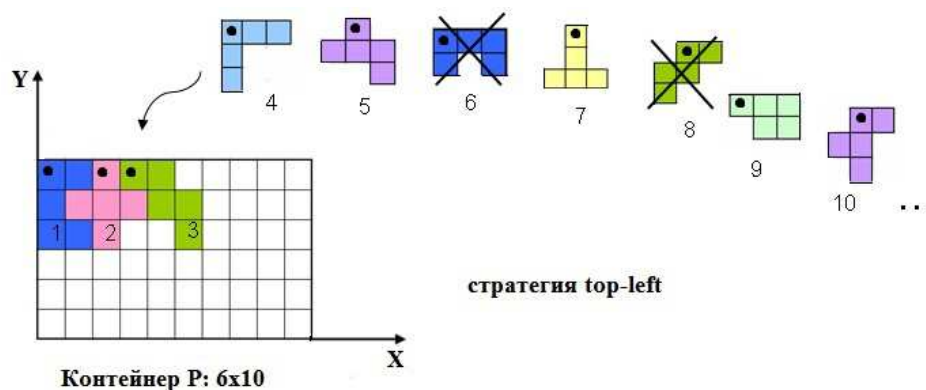


Рис. 3. Работа фрагментарного алгоритма

Конечно, далеко не всякая перестановка элементарных фрагментов приводит к оптимальному решению. Для поиска оптимальной перестановки воспользуемся эволюционной моделью на перестановках, которая приведена выше. Комбинация фрагментарной и эволюционной моделей и есть эволюционно-фрагментарная модель задачи упаковки пентамино. С помощью этой модели легко получаются оптимальные решения (упаковки с плотностью 1) практически любой из известных задач в игре «Пентамино».

Модель легко переносится на более сложные случаи целочисленных упаковок, например, когда ограничены преобразования симметрии фигур, налагаются ограничения на соседство, а также на обобщение игры пентамино — полимино.



Рис. 4 . Примеры контейнеров для упаковки пентамино

С целью апробации результатов исследования в рамках программной системы «Эволюционно-фрагментарное моделирование» был поставлен численный эксперимент по отысканию максимально плотных упаковок пентамино в контейнеры различного вида. Примеры некоторых контейнеров приведены на рис. 4. Было рассмотрено свыше 200 различных задач, для которых получена максимально плотная упаковка пентамино. Для поиска решений использовалась авторская программа, реализующая эволюционную модель на перестановках. Программа написана

на VBA под управлением WINDOWS XP/7. Вычисления проводились на компьютере ACPIx64-based PC с процессором 4x 3400 MHz. Время расчёта для каждой задачи было ограничено 15 минутами. Количество поколений в разных задачах составляло от 2000 до 5000, использовался пропорциональный принцип отбора родителей. Количество эволюций (отдельных запусков алгоритма) менялось от 1 до 20 в зависимости от задачи. Условием остановки являлось достижение плотности упаковки, равной 1, или достижение границы времени расчёта.

Для оценки эффективности алгоритма выполнено сравнение с алгоритмом локального поиска со случайным выбором начальной точки и с жадным алгоритмом со случайным выбором порядка элементов. Было рассмотрено 157 тестовых задач с контейнерами различной формы. Проведено две серии испытаний.

В короткой серии А параметры алгоритмов следующие:

а) локальный поиск со случайным выбором начальной точки: для каждой задачи проводилось 100 запусков алгоритма, выбирался лучший результат;

б) жадный алгоритм со случайным выбором последовательности укладки элементов: количество запусков составляло 2500;

в) эволюционный алгоритм: размер популяции — 500, количество поколений — 100, количество пар скрещивания в каждом поколении — 20, пропорциональный метод отбора пар.

В длинной серии В параметры алгоритмов следующие:

а) локальный поиск со случайным выбором начальной точки: для каждой задачи проводилось 500 запусков алгоритма, выбирался лучший результат;

б) жадный алгоритм со случайным выбором последовательности укладки элементов: количество запусков составляло 11500;

в) эволюционный алгоритм: размер популяции — 1500, количество поколений — 500, количество пар скрещивания в каждом поколении — 20, пропорциональный метод отбора пар.

Результаты работы алгоритмов приведены в табл. 1.

6. Заключение

Предложенный в настоящей работе метод поиска оптимальных решений задачи упаковки пентамино на основе фрагментарно-эволюционной модели даёт хорошие практические результаты в задаче отыскания оптимальных упаковок пентамино в плоские целочисленные контейнеры произвольной формы. Метод легко может быть обобщён на задачу целочисленной упаковки произвольных плоских выпуклых и невыпуклых

фигур в плоские контейнеры достаточно сложной формы.

Т а б л и ц а 1

Результаты сравнения алгоритмов

Серия	Алгоритм	Средний результат (максимум = 1)	Количество найденных точных оптимумов
Серия А	Локальный поиск	0,846	8
	Жадный алгоритм	0,848	45
	ЭВФ-алгоритм	0,896	129
Серия В	Локальный поиск	0,871	16
	Жадный алгоритм	0,882	59
	ЭВФ-алгоритм	0,909	142

ЛИТЕРАТУРА

1. Голомб С. В. Полимино. — М.: Мир, 1975. — 207 с.
2. Гэри М. Р., Джонсон Д. С. Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. — 416 с.
3. Емельянов В. В., Курейчик В. В., Курейчик В. М. Теория и практика эволюционного моделирования. — М.: Физматлит, 2003. — 432 с.
4. Ильев В. П. Задачи на системах независимости, разрешимые жадным алгоритмом // Дискрет. математика. — 2009. — Т. 21, вып. 4. — С. 85–94.
5. Козин И. В. Фрагментарные алгоритмы в системах поддержки принятия решений // Питання прикладної математики і математичного моделювання. — Днепропетровск, 2006. — С. 131–137.
6. Козин И. В., Полюга С. И. О свойствах фрагментарных структур // Вісник Запорізького національного університету. Математичне моделювання і прикладна механіка. — 2012. — № 1. — С. 99–106.
7. Козин И. В., Полюга С. И. Использование ЭВФ-алгоритмов для задачи прямоугольного раскроя // Питання прикладної математики і математичного моделювання. — Днепропетровск, 2009. — С. 199–208.
8. Мухачева А. С., Чиглинцев А. В. Генетический алгоритм поиска минимума в задачах двумерного гильотинного раскроя // Информ. технологии. Машиностроение. — 2001. — № 3. — С. 27–32.
9. Скопцов Ю. О. Основы эволюционных вычислений: Учеб. пособ. — Донецк: ДонНТУ, 2008. — 326 с.
10. Стоян Ю. Г., Яковлев С. В. Математические модели и оптимизационные методы геометрического проектирования. — Киев: Наук. думка, 1986. — 266 с.
11. Bjorner A., Ziegler G. M. Introduction to greedoids. — Cambridge: Cambridge Univ. Press, 1992. — 180 p.
12. Chambers L. Practical handbook of genetic algorithms. — CRC Press, 2001. — Vol. 1. Applications. — 544 p.

13. **Holland J. H.** Adaptation in natural and artificial systems. — Boston: MIT Press, 1992. — 217 p.
14. **Fletcher J. G.** A program to solve the pentomino problem by the recursive use of macros // Commun. ACM. — 1965. Vol. 8, N 10. — P. 621–623.
15. **White N.** Theory of matroids. — Cambridge: Cambridge Univ. Press, 1986. — 153 p. (Encyclopedia of mathematics and its applications; Vol. 26.)
16. **Whitney H.** On the abstract properties of linear dependence // Amer. J. Math. — 1935. — Vol. 57, N 3. — P. 509–533.

Козин Игорь Викторович,
e-mail: ains00@gmail.com
Полюга Светлана Игоревна,
e-mail: veta99@mail.ru

Статья поступила
16 января 2014 г.
Переработанный вариант —
26 июня 2014 г.