УДК 004.312+004.023

Теорема о сходимости генетического алгоритма с двухуровневым кодированием к точному решению задачи реконфигурации ПЛИС

А. Ю. Городилов
Пермский государственный национальный исследовательский университет
Россия, 614068, Пермь, ул. Букирева, 15
gora830@yandex.ru

Е. Ю. Данилова
Пермский государственный национальный исследовательский университет
Россия, 614068, Пермь, ул. Букирева, 15
ket-eref@yandex.ru

Приведена краткая постановка задачи реконфигурации ПЛИС, описаны базовый и генетический алгоритмы реконфигурации. Генетический алгоритм использует двухуровневое кодирование решений с промежуточным представлением в виде матрицы смежности полного двудольного графа. Сформулирована и доказана теорема о сходимости разработанного генетического алгоритма к точному решению задачи реконфигурации ПЛИС.

Ключевые слова: *генетический алгоритм; ПЛИС; FPGA; реконфигурация; сходимость; оптимизация*.

Введение

Для решения задачи реконфигурации ПЛИС необходимо совершить несколько шагов. Первый из них – проведение диагностики, по результатам которой становится известно состояние каждого логического элемента (ЛЭ). Таких состояний может быть два: 0 – неработоспособный и 1 – работоспособный.

Эти данные можно представить в виде таблицы, называемой матрицей состояний и обозначаемой A. Для реконфигурации выбирается наиболее предпочтительный набор работоспособных элементов, расположенных максимально компактно. Множество используемых ЛЭ в системах конфигурации ПЛИС описывается набором прямоугольников (координатами их левого верхнего угла и размерами). Прямоугольники не должны пересекаться и не должны содержать неработоспособных ЛЭ. Чем меньше количество прямоугольников в описании, тем быстрее будет выполнена реконфигурация.

Пусть $P_{x,y}(h,l)$ – матрица размерности $h\times l$, полученная из матрицы A удалением строк с номерами $1, 2, \ldots, x\text{-}1, x+h, \ldots, n$ и столбцов с номерами $1, 2, \ldots, y\text{-}1, y+l, \ldots, m$. То есть

$$P_{x,y}(h,l)=\begin{pmatrix} a_{x,y} & \cdots & a_{x,y+l-1} \\ \vdots & \ddots & \vdots \\ a_{x+h-1,y} & \cdots & a_{x+h-1,y+l-1} \end{pmatrix}. \quad (1)$$

Элемент $a_{x,y}$ будем называть *якорем* матрицы $P_{x,y}$. Если все элементы матрицы $P_{x,y}(h,l)$ равны 1, то это *покрывающий прямоугольник*. Количество элементов в нем назовем *площадью покрывающего прямоугольника* и обозначим $S(P)$. Множество R покрывающих прямоугольников $P_1,P_2,\ldots,P_k$ назовем *покрытием*, если никакая пара покрывающих прямоугольников из этого множества не пересекается. Сумму площадей покрывающих прямоугольников, входящих в покрытие, назовем *площадью покрытия* и обозначим $S(R)$.

Пусть t – количество ЛЭ, необходимых для реализации схемы устройства. Тогда задачу можно сформулировать так: дана матрица состояний A и число t. Найти покрытие площади $\geq t$ минимальной мощности. Данная задача является NP-трудной [1], поэтому на больших логических схемах точные алгоритмы будут работать неприемлемо долго. Следовательно, необходима разработка приближенных, эвристических алгоритмов.

Базовый приближенный алгоритм

Покрывающий прямоугольник $P_{x,y}(h_0,l_0)$ назовем *локально-максимальным*, если не существует покрывающего прямоугольника большей площади с тем же якорем.

На основе этого определения можно предложить следующий алгоритм решения задачи:

1. $R \leftarrow \varnothing$.
2. Установить порядок просмотра элементов матрицы A.
3. Выбрать очередной элемент $a_{x,y}$ из матрицы A в соответствии с установленным порядком.
4. Найти локально-максимальный покрывающий прямоугольник $P_{x,y}(h_0,l_0)$ с якорем $a_{x,y}$.
5. $R \leftarrow R \cup P_{x,y}(h_0,l_0)$.
6. Обнулить все элементы матрицы A, входящие в $P_{x,y}(h_0,l_0)$: $a_{ij} \leftarrow 0$, $i=x..x+h_0-1$, $j=y..y+l_0-1$.
7. Вычислить, сколько «единиц» еще нужно покрыть: $t \leftarrow t - h_0 \cdot l_0$.
8. Если $t>0$, перейти к шагу 3.
9. Покрытие R является решением задачи.

Данный алгоритм работает быстро, но не всегда находит точное решение, то есть не всегда найденное количество покрывающих прямоугольников оказывается минимальным. Очевидно, что результат работы алгоритма может зависеть от установленного в шаге 2 порядка просмотра элементов.

Генетический алгоритм

Для разработки генетического алгоритма (ГА) необходимо в первую очередь определить структуру хромосом и функцию приспособленности. Принимая во внимание алгоритм α, решение может быть найдено в виде последовательности, задающей порядок просмотра элементов матрицы состояний. Неработоспособные элементы можно сразу исключить из рассмотрения. Таким образом, длина каждой хромосомы L будет фиксирована и равна количеству "1" в таблице состояний. Но сама такая последовательность s (перестановка на множестве работоспособных ЛЭ) является только особью, но не может кодироваться в виде хромосомы, поскольку гены (элементы перестановки) зависят друг от друга. Чтобы обеспечить независимость генов, закодируем перестановку с помощью промежуточного объекта. В качестве такого объекта предлагается совершенное паросочетание в полном двудольном графе. Хромосомой при этом будет сам граф, заданный в виде матрицы смежности. Такое представление обладает двумя основными свойствами: независимость генов (элементов матрицы) и существование эффективной процедуры преобразования в перестановку, для этого достаточно упорядочить все ребра паросочетания по вершинам первой доли (перестановку образуют вершины второй доли). Процедуру преобразования будем обозначать *Present*. Для поиска самого паросочетания в графе используется эффективный "венгерский алгоритм", который находит правильный ответ за полиномиальное время [2].

Подробное описание операторов двухуровневого ГА и промежуточного представления можно найти в [3, 4, 5].

Приспособленность особи определяется после запуска алгоритма α как величина, обратная к мощности множества R.

Обоснование сходимости генетического алгоритма к точному решению

ГА относятся к числу эвристических методов, поэтому строгое доказательство сходимости получить невозможно. Но использование стандартных классических операторов в предложенном ГА позволяет принять во внимание все существующие теоретические обоснования и экспериментальные подтверждения эффективности генетических алгоритмов.

Единственное, что отличает предложенный алгоритм от классических – это оригинальный способ кодирования и, соответственно, вычисления приспособленности особей. Следовательно, необходимо проверить три условия, которым должна удовлетворять фитнесс-функция в классических ГА.

1. Потомки, получающиеся в результате скрещивания, должны быть «похожими» на своих родителей.

Это означает, что в результате скрещивания двух перестановок должны получаться перестановки, «похожие» на родительские особи. Термин «похожими» в данном случае формализовать невозможно. Будем исходить из такой трактовки этого термина: перестановки должны совпадать значениями отдельных элементов в некоторых позициях. Пусть особи μ_3 и μ_4 являются потомками особей μ_1 и μ_2,

$$s_1=Present(\mu_1)=(i'_1 \quad i'_2 \quad \dots \quad i'_L),$$
$$s_2=Present(\mu_2)=(i''_1 \quad i''_2 \quad \dots \quad i''_L),$$
$$s_3=Present(\mu_3)=(j'_1 \quad j'_2 \quad \dots \quad j'_L),$$
$$s_4=Present(\mu_4)=(j''_1 \quad j''_2 \quad \dots \quad j''_L).$$

Нас интересуют позиции k, в которых значение элемента потомка j'_k совпадает со значением соответствующего элемента в одном из родителей, то есть с i'_k или i''_k. Обозначим через Sum_1 количество таких номеров k для первого потомка, а через Sum_2 – для второго. То есть,

$$Sum_1=\sum_{k=1}^{L} I\left(\left(j'_k=i'_k\right)\vee\left(j'_k=i''_k\right)\right),$$

$$Sum_2=\sum_{k=1}^{L} I\left(\left(j''_k=i'_k\right)\vee\left(j''_k=i''_k\right)\right).$$

Общее количество позиций, в которых может быть совпадение элементов перестановок, очевидно, равно $2L$. Таким образом, нас интересует отношение $Q=\dfrac{Sum_1+Sum_2}{2L}$, которое имеет смысл доли генов в потомстве, унаследованных от родителей, и выражает степень «похожести» потомков на предков.

К сожалению, теоретически нельзя гарантировать, что параметр Q будет достаточно большим, поэтому значение Q проверялось эмпирически. Для проведения эксперимента было сгенерировано 35 матриц размерности 11, заполненных случайными числами. Далее 80 раз был запущен оператор скрещивания, и для каждого запуска определялось значение параметра Q. По полученным данным можно сделать следующие выводы:

- во всех случаях параметр Q превысил значение 0,4;
- более чем в 90 % случаев параметр Q превысил значение 0,5;
- среднее значение параметра Q составило 0,732.

То есть в абсолютном большинстве случаев потомками от родителей наследуется по меньшей мере половина значений, что можно признать вполне приемлемым результатом.

2. Чем больше значение приспособленности особи, тем лучше должно быть решение, кодируемое данной особью.

Выполнение данного условия гарантируется непосредственно правилом вычисления приспособленности: чем больше величина 1/|R|, тем меньше |R|, т. е. меньше количество задействованных прямоугольников и, следовательно, элементы располагаются более компактно.

3. Существует особь с максимальной приспособленностью, кодирующая точное решение задачи.

Для проверки данного условия докажем следующую теорему.

Теорема. Существует последовательность просмотра элементов $a_{x_1,y_1}, a_{x_2,y_2}, \dots, a_{x_k,y_k}$ такая, что выполнение алгоритма α дает точное решение задачи.

Доказательство. Обозначим точное решение задачи R^*, а решение, полученное алгоритмом α, обозначим R. Докажем, что для любых матрицы A и числа t существует такая последовательность просмотра элементов, что $|R|=|R^*|$. Доказательство выполним по индукции по мощности покрытия R^* для пары (A, t).

<u>База индукции</u> $|R^*|=1$. То есть $R^* = \{P^*_{x,y}(h_1,l_1)\}$. Тогда очевидно, что последовательность, состоящая из одного элемента $(a_{x,y})$, является искомой. Действительно, в локально-максимальном покрывающем прямоугольнике $P_{x,y}(h_0,l_0)$: $h_0 \cdot l_0 \ge h_1 \cdot l_1 \ge t$, т. е. $R = \{P_{x,y}(h_0,l_0)\}$ и $|R|=|R^*|=1$.

<u>Индукционный переход</u>. Пусть утверждение справедливо для всех пар (A, t), для которых $|R^*|=k$-1. Докажем, что оно будет справедливо и для пар, в которых $|R^*|=k$.

Лемма 1. В оптимальном покрытии R^* найдется покрывающий прямоугольник $P^*_{u,v}(h_1,l_1)$ такой, что локально-максимальный покрывающий прямоугольник с тем же якорем $P_{u,v}(h_0,l_0)$ попарно не пересекается ни с одним из остальных покрывающих прямоугольников из R^*, то есть

$$\exists P^*_{u,v}(h_1,l_1):\forall P^*\in R^*, P^*\neq P^*_{u,v}(h_1,l_1): \quad P_{u,v}(h_0,l_0)\cap P^*=\varnothing. \tag{2}$$

Доказательство

Предположим противное:

$$\forall P^*_{u,v}(h_1,l_1):\exists P^*\in R^*, P^*\neq P^*_{u,v}(h_1,l_1): \quad P_{u,v}(h_0,l_0)\cap P^*\neq\varnothing. \tag{3}$$

Пусть $R^*=\left\{P^*_{x_1,y_1},P^*_{x_2,y_2},\ldots,P^*_{x_k,y_k}\right\}$. Построим ориентированный граф $G=(V,E)$. Отметим, что множество вершин соответствует множеству якорей покрывающих прямоугольников из R^*:

$$V=\{v_1,v_2,\ldots,v_k\}, v_i=a_{x_i,y_i}, i=\overline{1,k}.$$

Ребро

$$(v_i,v_j)\in E \Leftrightarrow P_{x_i,y_i}(h_0,l_0)\cap P^*_{x_j,y_j}(h_j,l_j)\neq\varnothing.$$

То есть ребро идет из вершины i в вершину j, если локально-максимальный покрывающий прямоугольник с якорем в i-той вершине пересекается с покрывающим прямоугольником с якорем в j-той вершине.

Лемма 1.1

$$P_{x',y'}(h',l')\cap P_{x'',y''}(h'',l'')\neq\varnothing \Leftrightarrow x'+h'>x''\wedge x'<x''+h''\wedge y'+l'>y''\wedge y'<y''+l''.$$

Доказательство

$\Rightarrow$ Пусть $a_{x,y}\in P_{x',y'}(h',l')\cap P_{x'',y''}(h'',l'')$. Тогда

$$\left.\begin{array}{l} a_{x,y}\in P_{x',y'}(h',l')\Rightarrow x\geq x' \\ a_{x,y}\in P_{x'',y''}(h'',l'')\Rightarrow x<x''+h'' \end{array}\right\}\Rightarrow x'<x''+h''.$$

Остальные неравенства доказываются аналогично.

$\Leftarrow$. Возьмем $x=\max(x',x'')$. Очевидно, $x\geq x', x\geq x''$. Поскольку $x'<x'+h'$ и $x''<x'+h'$, то $x<x'+h'$. Аналогично доказывается, что $x<x''+h''$, $y<y'+l'$ и $y<y''+l''$. Таким образом,

$$a_{x,y}\in P_{x',y'}(h',l'), a_{x,y}\in P_{x'',y''}(h'',l''),$$

что и требовалось доказать.

Следствие 1.1

$$(v_i,v_j)\in E\Rightarrow x_i+h_0>x_j\wedge x_i<x_j+h_j\wedge y_i+l_0>y_j\wedge y_i<y_j+l_j. \tag{4}$$

Лемма 1.2

$$(v_i,v_j)\in E\Rightarrow y_j>y_i\vee x_j>x_i.$$

Доказательство

Предположим противное, и $y_j\leq y_i\wedge x_j\leq x_i$. По следствию 1.1 $x_i<x_j+h_j$ и $y_i<y_j+l_j$. Следовательно,

$$x_j\leq x_i<x_j+h_j\wedge y_j\leq y_i<y_j+l_j\Rightarrow a_{x_i,y_i}\in P^*_{x_j,y_j}(h_j,l_j).$$

Кроме того, очевидно, $a_{x_i,y_i}\in P^*_{x_i,y_i}(h_i,l_i)$. Но два покрывающих прямоугольника из покрытия R^* не могут иметь общих элементов. Возникает противоречие. Лемма 1.2 доказана.

В соответствии с (3)

$$\forall i\ \exists j,\ j\neq i: P_{x_i,y_i}(h_0,l_0)\cap P^*_{x_j,y_j}(h_j,l_j)\neq\varnothing.$$

То есть, из любой вершины графа G есть хотя бы одно исходящее ребро. Следовательно, в графе G есть ориентированный цикл. Пусть его образуют вершины $v_1,v_2,\ldots,v_w$. Далее будем рассматривать только вершины и ребра, составляющие этот цикл.

На основе графа $G=(V,E)$ (точнее, вершин и ребер, образующих ориентированный цикл) построим ориентированный ортогональный граф $H=(U,E_H)$ на плоскости по следующим правилам. Вершины:

$$v_i=a_{x_i,y_i}\in V\rightarrow u_i=(x_i,y_i)\in U.$$

Ребра. В соответствии с леммой 1.2 возможны три типа ребер в графе G (рис. 1).

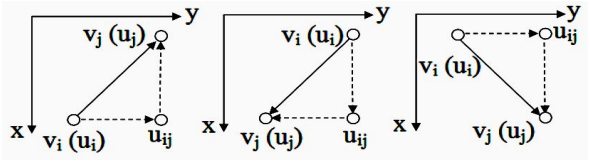

Рис. 1. *Возможные ребра в графе G*

Тип 1. $y_j>y_i, x_j\leq x_i$. Добавим в граф H вершину $u_{ij}=(x_i,y_j)$ и пару ребер: горизонтальное (u_i,u_{ij}) и вертикальное (u_{ij},u_j) (в случае $x_j=x_i$ вершина u_{ij} совпадает с вершиной u_j и вертикальное ребро отсутствует). Заметим, что если точка (x, y) лежит на ребре (u_{ij}, u_j), то $y=y_j$, $x_j\leq x\leq x_i$.

По следствию 1.1, $x_i<x_j+h_j$.

Из указанных соотношений получаем, что

$$y=y_j,\ x_j\leq x\leq x_i<x_j+h_j\Rightarrow a_{x,y}\in P^*_{x_j,y_j}. \tag{5}$$

Тип 2. $y_j\leq y_i, x_j>x_i$. Действуем аналогично, добавляя вершину $u_{ij}=(x_j,y_i)$ и пару ребер: (u_i,u_{ij}) и (u_{ij},u_j). Если точка (x, y) лежит на ребре (u_{ij}, u_j), то

$$x=x_j,\ y_j\leq y\leq y_i<y_j+l_j\Rightarrow a_{x,y}\in P^*_{x_j,y_j}. \tag{6}$$

Тип 3. $y_j > y_i, x_j > x_i$. Добавим в граф H вершину $u_{ij} = (x_i, y_j)$ и пару ребер: горизонтальное (u_i, u_{ij}) и вертикальное (u_{ij}, u_j).

По построению граф *H* представляет собой граф-цикл.

Пусть $y_r = \max(y_1, y_2, \ldots, y_w)$. Рассмотрим соответствующую вершину v_r (если для нескольких вершин $y=y_r$, возьмем вершину с минимальным значением *x*). Пусть вершине v_r инцидентны ребра $(v_i,v_r),(v_r,v_j)$. Поскольку $y_r \geq y_j$, ребро (v_r,v_j) относится к типу 2. В случае $y_r=y_i$, согласно лемме 1.2, $x_r>x_i$, но в случае равенства мы выбирали в качестве v_r вершину с минимальным значением *x*. Значит, $y_r>y_i$ и ребро (v_i,v_r) относится к типу 1 или к типу 3. Разберем эти случаи.

Случай 1. Ребро (v_i,v_r) относится к типу 1 (рис. 2).

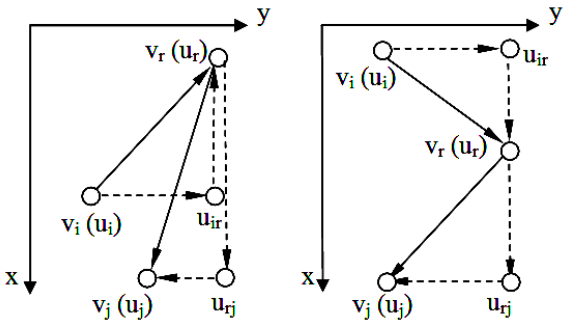

Рис. 2. *Варианты расположения ребер, инцидентных вершине* v_r.

В графе *H* получим последовательность ребер $(u_i,u_{ir}),(u_{ir},u_r),(u_r,u_{rj}),(u_{rj},u_j)$, где

$$u_{ir}=(x_i,y_r),u_{rj}=(x_j,y_r),y_r>y_i,x_r\leq x_i,y_j\leq y_r,x_j>x_r. \quad (7)$$

Последовательность двух вертикальных ребер $(u_{ir},u_r),(u_r,u_{rj})$ заменим одним ребром (u_{ir},u_{rj}). Определим его направление.

$P^*_{x_r,y_r}(h_r,l_r) \cap P^*_{x_j,y_j}(h_j,l_j) = \varnothing$, поэтому по лемме 1.1:

$$x_r + h_r \leq x_j \vee x_r \geq x_j + h_j \vee \\ y_r + l_r \leq y_j \vee y_r \geq y_j + l_j. \quad (8)$$

Из (7) получаем $x_r<x_j<x_j+h_j$, $y_j\leq y_r<y_r+l_r$, следовательно, в выражении (8) второе и третье неравенства не выполняются.

$(v_r, v_j) \in E$, поэтому по следствию 1.1: $y_r < y_j + l_j$, то есть в выражении (8) четвертое неравенство тоже не выполняется. Значит, должно выполняться первое неравенство: $x_r + h_r \leq x_j$. $(v_i, v_r) \in E$, поэтому по следствию 1.1: $x_i < x_r + h_r$.

Получаем $x_i < x_r + h_r \leq x_j$. Таким образом, ребро (u_{ir},u_{rj}) будет направлено вниз. Вся фигура, ограниченная графом-циклом *H*, располагается в правой полуплоскости относительно ребра (u_{ir},u_{rj}), т. е. цикл *H* имеет ориентацию по часовой стрелке.

Случай 2. Ребро (v_i,v_r) относится к типу 3 (рис. 2).

В графе *H* получим последовательность ребер $(u_i,u_{ir}),(u_{ir},u_r),(u_r,u_{rj}),(u_{rj},u_j)$, где

$$u_{ir}=(x_i,y_r),u_{rj}=(x_j,y_r),y_r>y_i,x_r>x_i,y_j\leq y_r,x_j>x_r. \quad (9)$$

Последовательность двух вертикальных ребер $(u_{ir},u_r),(u_r,u_{rj})$ заменим одним ребром (u_{ir},u_{rj}). В соответствии с (9): $x_j>x_r>x_i$.

Таким образом, ребро (u_{ir},u_{rj}) будет направлено вниз. Вся фигура, ограниченная графом-циклом *H*, располагается в правой полуплоскости относительно ребра (u_{ir},u_{rj}), то есть цикл *H* имеет ориентацию по часовой стрелке.

В обоих случаях мы получили, что цикл *H* имеет ориентацию по часовой стрелке. Точки самопересечения (если они есть) разбивают цикл *H* на несколько отдельных замкнутых частей без самопересечений. Рассмотрим ту часть, которая включает в себя ребро (u_{ir}, u_{rj}). По доказанному выше, этот цикл имеет ориентацию по часовой стрелке. Рассмотрим самые левые точки этого цикла и выберем из них самую нижнюю. Пусть это будет точка (x,y). Учитывая выбор точки (x, y) и тот факт, что все ребра цикла строго вертикальны либо горизонтальны, точка (x, y) находится на пересечении горизонтального ребра (u', u'') и вертикального ребра (u''', u'''') (возможно, в вершинах этих ребер). Учитывая ориентацию цикла (части цикла), ребро (u', u'') ориентировано справа налево (цикл расположен выше), а ребро (u''', u'''') – снизу вверх (цикл расположен справа). Анализируя все возможные типы ребер, перечисленные выше, получаем, что ребро (u', u'') получается только из ребра типа 2 и является ребром вида (u_{ij}, u_j), а ребро (u''', u'''') – из ребра типа 1 и является ребром типа (u_{rt}, u_t). При рассмотрении ребер типа 2 было установлено, что, согласно (6), если точка (x,y) лежит на ребре (u_{ij}, u_j), то $a_{x,y} \in P^*_{x_j,y_j}$. При рассмотрении ребер типа 1 было установлено, что, согласно (5), если точка (x,y) лежит на ребре (u_{rt}, u_t), то $a_{x,y} \in P^*_{x_t,y_t}$. Но тогда получается, что

$P^*_{x_j,y_j}(h_j,l_j) \cap P^*_{x_t,y_t}(h_t,l_t) \neq \varnothing$, что возможно только при $j=t$, $i=r$. Но тогда ребро $(v_i,v_j)=(v_r,v_t)$ относится одновременно и к типу 1, и к типу 2, что невозможно. Получили противоречие.

Доказательство леммы 1 завершено.

Вернемся к доказательству теоремы. Рассмотрим оптимальное покрытие R^*. Согласно доказанной лемме 1, в нем найдется покрывающий прямоугольник $P^*_{u,v}(h_1,l_1)$ такой, что локально-максимальный покрывающий прямоугольник с тем же якорем $P_{u,v}(h_0,l_0)$ попарно не пересекается ни с одним из остальных покрывающих прямоугольников из R^*. Обнулим элементы матрицы A, входящие в $P_{u,v}(h_0,l_0)$. Получим матрицу $A'=(a'_{i,j})$:

$$a'_{i,j} = \begin{cases} a_{i,j}, i \in [1;u-1] \cup [u+h_0;n] \vee \\ \quad j \in [1;v-1] \cup [v+l_0;m] \\ 0, i \in [u;u+h_0-1] \wedge \\ \quad j \in [v;v+l_0-1]. \end{cases} \qquad (10)$$

Обозначим

$$t' = t - h_0 \cdot l_0, R'^* = R^* \setminus P^*_{u,v}(h_1,l_1).$$

$$\left|R'^*\right| = \left|R^* \setminus P^*_{u,v}(h_1,l_1)\right| = \left|R^*\right| - 1 = k-1.$$

Лемма 2. Покрытие R'^* является точным оптимальным решением рассматриваемой задачи для пары (A',t').

Доказательство. Во-первых, докажем, что R'^* действительно является покрытием в матрице A'. Рассмотрим произвольный покрывающий прямоугольник $P' \in R'^*$ и произвольный элемент $a'_{i,j} \in P'$. Поскольку $P' \in R'^* \subset R^*$ являлся покрывающим прямоугольником в матрице A,

$$a_{i,j} = 1. \qquad (11)$$

С другой стороны,

$$P' \in R^*, P' \neq P^*_{u,v}(h_1,l_1).$$

Следовательно, по лемме 1, согласно (2): $P_{u,v}(h_0,l_0) \cap P' = \varnothing$. То есть, поскольку $a'_{i,j} \in P'$,

$$a'_{i,j} \notin P_{u,v}(h_0,l_0) \Rightarrow i \in [1;u-1] \cup [u+h_0;n] \vee$$
$$j \in [1;v-1] \cup [v+l_0;m]$$

и, в соответствии с (10) и (11): $a'_{i,j} = a_{i,j} = 1$.

Таким образом, все элементы прямоугольника P' в матрице A' равны 1, то есть он является покрывающим, а R'^* является покрытием.

Во-вторых,

$$S(R'^*) = S(R^*) - S(P^*_{u,v}(h_1,l_1)) =$$
$$S(R^*) - h_1 \cdot l_1 \geq t - h_0 \cdot l_0 = t'.$$

То, что мощность R'^* минимальна, легко доказать от противного.

Доказательство леммы 2 завершено.

Возвращаемся к доказательству теоремы. Итак, покрытие R'^* является точным оптимальным решением рассматриваемой задачи для пары (A',t') и $\left|R'^*\right| = k-1$. Значит, по предположению индукции, существует последовательность просмотра элементов $a_{x_1,y_1}, a_{x_2,y_2}, \ldots, a_{x_{k-1},y_{k-1}}$ такая, что выполнение алгоритма α дает точное решение задачи для пары (A',t'). Обозначим это решение R'. $\left|R'\right| = \left|R'^*\right| = k-1$. Тогда при выполнении алгоритма α для пары (A, t) с последовательностью $a_{u,v}, a_{x_1,y_1}, a_{x_2,y_2}, \ldots, a_{x_{k-1},y_{k-1}}$ при первой итерации на шаге 4 будет выбран прямоугольник $P_{u,v}(h_0,l_0)$, на шаге 6 будет построена матрица A', а на шаге 7 вычислено $t' = t - h_0 \cdot l_0$. После первой итерации мы получаем задачу для пары (A',t') и последовательность $a_{x_1,y_1}, a_{x_2,y_2}, \ldots, a_{x_{k-1},y_{k-1}}$. Дальнейшее выполнение алгоритма даст покрытие R'. Результатом алгоритма α для пары (A, t) будет покрытие $R = P_{u,v}(h_0,l_0) \cup R'$, которое является точным решением задачи.

Действительно, R по построению является покрытием с площадью $\geq t$ и

$$\left|R\right| = \left|P_{u,v}(h_0,l_0) \cup R'\right| = 1 + k - 1 = k = \left|R^*\right|.$$

Теорема доказана.

Таким образом, действительно существует особь генетического алгоритма, кодирующая точное решение задачи. В силу справедливости условия 2, приспособленность такой особи, очевидно, максимальна.

Заключение

Одним из спорных моментов любого эвристического алгоритма, к которым относятся и генетические алгоритмы, является сходимость. В данной статье приведена и доказана теорема о сходимости генетического алгоритма с двухуровневым кодированием для задачи реконфигурации ПЛИС. Были

проверены условия сходимости для фитнесс-функции ГА, подробно доказано наиболее интересное условие – существование особи с максимальной приспособленностью, кодирующей точное решение задачи реконфигурации. Кроме того, был подробно описан базовый приближенный алгоритм поиска покрытия, который используется для вычисления фитнесс-функции.

Список литературы

1. *Еремеев А.В., Заозерская Л.А., Колоколов А.А*. Задача о покрытии множества: сложность, алгоритмы, экспериментальные исследования. Дискретный анализ и исследование операций. Сер. 2. 2000. Т. 7, № 2. С.22–46.
2. *Асанов М. О., Баранский В. А., Расин В. В.* Дискретная математика: графы, матроиды, алгоритмы. НИЦ "Регулярная и хаотическая динамика": Ижевск, 2001.
3. *Данилова Е. Ю., Городилов А. Ю.* Сравнение генетических алгоритмов на примере задачи коммивояжера // Вестник Пермского университета. Математика. Механика. Информатика. Пермь: ПГУ, 2009. Вып. 3 (29). С. 49–53.
4. *Городилов А. Ю.* Двухуровневый генетический алгоритм для решения задач выделения компактных групп объектов // Междисциплинарные исследования: сб. матер. науч.-практ. конф. Перм. гос. нац. исслед. ун-т. Пермь, 2013. Т. 1. С. 190–193.
5. *Городилов А.Ю.* Двухуровневый генетический алгоритм реконфигурации программируемых логических интегральных схем // Information Technologies and Knowledge. 2014. Vol. 8, № 2. P. 131–140.

A theorem on the convergence of two-level coding genetic algorithm to the exact solution of the FPGA reconfiguration task

A. Yu. Gorodilov
Perm State University, Russia, 614068, Perm, Bukirev st., 15
gora830@yandex.ru

E. Yu. Danilova
Perm State University, Russia, 614068, Perm, Bukirev st., 15
ket-eref@yandex.ru

The article gives a brief statement of the FPGA reconfiguration problem and describes basic and genetic reconfiguration algorithms. The genetic algorithm uses a two-level solution coding and intermediate representation in the form of the complete bipartite graph adjacency matrix. A theorem on the convergence of developed genetic algorithm to the exact solution of the FPGA reconfiguration task is formulated and proved.

Key words: *genetic algorithm; FPGA; reconfiguration; convergence; optimization.*