

Б. К. Лебедев, д-р техн. наук, проф., e-mail: lebedev.b.k@gmail.com,

О. Б. Лебедев, канд. техн. наук, доц., e-mail: lebedev.ob@mail.ru,

Южный федеральный университет, Ростов-на-Дону,

О. А. Пурчина, ассистент, e-mail: knagna_olga@inbox.ru,

Донской государственный технический университет, Ростов-на-Дону

Модифицированный роевой алгоритм "муравьиное дерево" в задаче диверсификации трассировочных ресурсов¹

Предлагается модифицированный роевой алгоритм "муравьиное дерево" (trees ant colony optimization (T-ACO)). Отличием от канонической структуры является то, что в качестве графа поиска решений используется дерево решений. Агентом на графе поиска решений создается не маршрут, а дерево, которое по своей структуре совпадает с представлением решения задачи трассировки в надъязыечной области. Это исключает применение дополнительных трансформаций в процессе декодирования решений, допускающих интерпретацию решений в виде деревьев, и позволяет отбросить множество "нелегальных" решений, что приводит к повышению качества полученных решений. Использование процедуры трассировки в надъязыечной области на основе муравьиного алгоритма позволяет разгрузить канал на 15...20 %. Временная сложность алгоритма зависит от числа вершин графа n , числа агентов u , числа итераций l и определяется как $O(n^2lu)$.

Ключевые слова: оптимизация, алгоритм "муравьиное дерево", САПР СБИС, диверсификация трассировочных ресурсов, надъязыечная область, планарный эскиз топологии

Введение

Одной из наиболее важных задач, которые решаются при проектировании топологии СБИС, является канальная трассировка.

Канал представляет собой область, ограниченную двумя линейками контактов: верхней и нижней [1]. На область трассировки наносится опорная ортогональная сеть, по линиям которой проходят трассы. Горизонтальные линии называют магистралями. Вертикальные линии проходят через контакты. При канальной трассировке каждая цепь, связывающая эквипотенциальные выводы, представляется в виде набора горизонтальных и вертикальных фрагментов (участков). В общем случае задача трассировки в канале (ЗКТ) рассматривается как задача размещения множества горизонтальных участков на множестве магистралей с соблюдением ограничений.

Существует множество трассировщиков, некоторые из которых позволяют получать почти оптимальные решения [1–5]. Однако по-прежнему проблема минимизации области канальной трассировки является актуальной. Дефицит трассировочного ресурса часто приводит к тому, что в процессе детальной трас-

сировки проводники прокладываются в обход загруженных областей, что приводит к увеличению задержек в соединениях и уменьшению быстродействия СБИС в целом. Более того, дефицит трассировочного ресурса при недостаточно качественной глобальной трассировке может привести к невозможности 100%-ной детальной трассировки. Оценкой **трассируемости** коммутационной среды (КС) является соотношение между ресурсами КС и ресурсами, требуемыми для реализации соединений. Чем больше протрассированных соединений для КС с заданными размерами, или чем меньше размеры КС, требуемые для 100%-ной трассировки, тем выше трассируемость. Ресурсы КС определяются размерами КС.

Ресурсы, требуемые для реализации соединений, определяются конфигурацией и областью распространения соединений в КС, т. е. геометрическими параметрами соединений (координатами выводов, связываемых цепями).

В работе задача повышения трассируемости решается путем диверсификации трассировочных ресурсов, заключающейся в использовании дополнительных областей, примыкающих к каналам снизу и сверху, с трассировочным слоем, который располагается над ячейками "over-the-cell", как показано на рис. 1. Тогда часть соединений может быть перенаправлена из канала в надъязыечные области (НЯО), что

¹Работа выполнена при финансовой поддержке РФФИ (грант № 18-07-00737а).

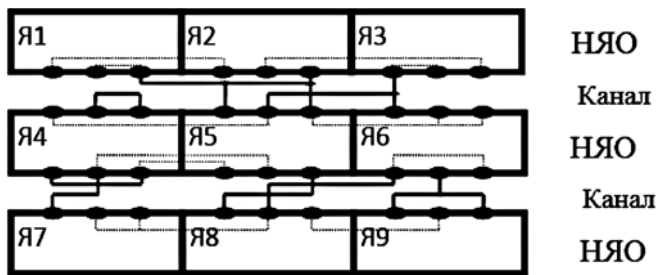


Рис. 1. Структура коммутационного поля

приведет к сокращению числа магистралей, которые используются в канале [4, 5, 7, 8]. Пере-направленные соединения трассируются в одном слое. Целью надъячеечного трассировщика является создание максимального планарного эскиза топологии в НЯО, это должно позволить минимизировать плотность канала [4, 5].

Среди существующих алгоритмов трассировки в надъячеечной области [2] наиболее распространены: генетические алгоритмы (ГА) (авторы: В. М. Goni, Т. Arslan, В. Torton); алгоритм на основе методов динамического программирования (МДП) (авторы: N. Holmes, N. Sherwani, М. Sarrafzadeh). В связи с тем что сложность и размерность решаемых задач в наше время резко возросли, МДП редко используется, поскольку они очень трудоемки. В свою очередь по тем же причинам в ГА резко возросла трудоемкость декодирований и усилились проблемы, связанные с получением "легальных" решений. Помимо этого, возникают проблемы надъячеечной трассировки в областях с ломаными границами, поэтому разработка алгоритма трассировки в НЯО, который мог бы отвечать современным требованиям, является актуальной научной задачей.

Основные процедуры при решении задачи надъячеечной трассировки:

1. Разбиение многовыводных цепей на двухтерминальные соединения.
2. Формирование наборов двухтерминальных соединений (ДТС) — кандидатов для размещения в надъячеечной области.
3. Формирование ориентированного графа поиска решений G в виде дерева, задающего бинарное отношение вложения между каждой парой вершин, соответствующих ДТС.
4. Формирование полного графа R пересечений, отражающего наличие неустраимого пересечения между каждой парой ДТС.
5. Выделение в графе G ориентированного решающего дерева $T_w = (V_w, U_w)$, такого что между каждой парой вершин ДТС отсутствует отношение неустраимого пересечения.

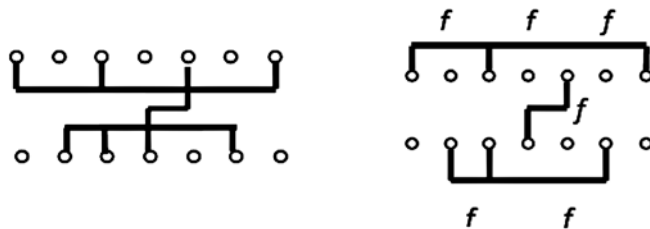


Рис. 2. Разбиение многовыводной цепи на фрагменты

Прежде всего, формируется набор $F = \{f_i | i = 1, 2, \dots, m\}$ двухтерминальных соединений, которые являются кандидатами на перевод из канала в НЯО. Каждая из многовыводных цепей (рис. 2) на каждой из сторон канала разбивается на двухвыводные цепи (фрагменты), которые связывают контакты в линейке.

Пусть l_i обозначает левый край фрагмента f_i , тогда r_i будет правым краем, а $x(l_i)$ и $x(r_i)$ будут координатами краев фрагмента f_i по оси x , которая распространяется вдоль линейки контактов.

Если $x(l_j) < x(l_i) < x(r_j)$, то между фрагментами f_i и f_j существует неустраимое пересечение (рис. 3). Формируется граф неустраимых пересечений $CR = (V, E)$, вершина $v_i \in V$ соответствует фрагменту f_i , т. е. $f_i = G(v_i)$, вершины v_i и v_j связаны ребром $e_k = (v_i, v_j)$, когда f_j пересекается с f_i .

Будем считать, что f_i вкладывается в f_j , в случае если:

- $x(l_j) < x(l_i)$, $x(r_j) > x(r_i)$;
- f_i и f_j размещаются в надъячеечной области без пересечений, т. е. f_j будет выше f_i (рис. 4).

Если $x(r_j) < x(l_i)$, то фрагменты f_i и f_j не пересекаются и могут размещаться в надъячеечной области без взаимопересечения.

Построим ориентированный граф поиска решений $G = (V, U)$, вершина $v_i \in V$ соответствует фрагменту f_i ($f_i = G(v_i)$).

Вершины v_i и v_j связаны ребром, направленным от v_i к v_j только тогда, когда f_j вкладывается в f_i . Вершина v_r , соответствующая верхней границе НЯО, включена в состав V . Вершина (v_r) будет связана с каждой вершиной $V \setminus v_r$, потому что любой из фрагментов вкладывается в НЯО, которая ограничена верхней границей.



Рис. 3. Неустраимое пересечение

Рис. 4. Вложение цепей

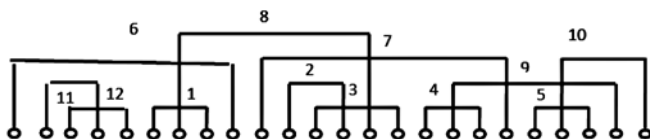


Рис. 5. Пример схемы

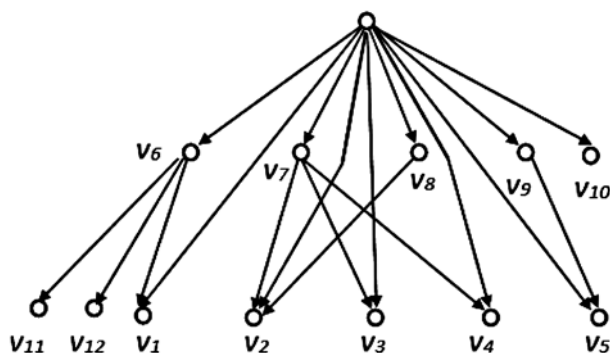


Рис. 6. Граф поиска решений G

На рис. 5 изображена схема, а на рис. 6 соответствующий данной схеме граф поиска решений (ГПР) G .

Среди множества фрагментов F_i , вкладываемых в f_i , существует подмножество $F_{is} \subset F_i$, $G^{-1}(F_{is}) = B_{is} \subset B_i$, такое что никакие два фрагмента $f_k \in F_{is}$ и $f_j \in F_{is}$ не пересекаются. Назовем данное подмножество F_{is} s -м вложением фрагмента f_i , а подмножество B_{is} — s -м вложением вершины v_i . Множеств таких, как F_{is} , может быть несколько, $\cup F_{is} = F_i$.

1. Постановка задачи

Имеется набор $F = \{f_i | i = 1, 2, \dots, m\}$ двухтерминальных соединений (фрагментов), которые являются кандидатами на перевод из канала в НЯО. На базе набора F построен ориентированный граф поиска решений $G = (V, U)$ в виде дерева, отражающий бинарное отношение вложения, и полный граф неустраимых пересечений $CR = (V, E)$. Каждому из фрагментов f_i соответствует стоимость ε_i . Необходимо в ГПР G выделить подграф $T_w = (V_w, U_w)$, $T_w \subset G$, который обладает следующими свойствами [6, 9]:

- T_w представляет собой дерево с корневой вершиной v_r ;

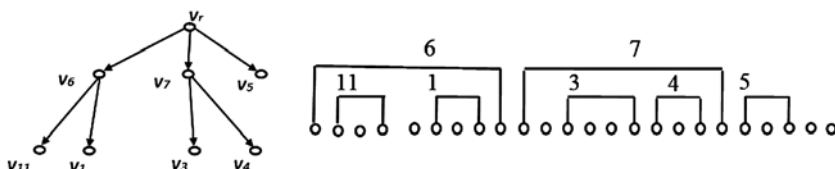


Рис. 7. Пример решающего дерева

- B_{iw} — это множество дочерних вершин вершины $v_i \in V_w$, $B_{iw} \subset V_w$;
- для любой $v_i \in V_w$ множество B_{iw} является вложением вершины v_i , иными словами, фрагменты множества $F_{iw} = G(B_{iw})$ являются непересекающимися и вкладываемыми во фрагмент f_i .

Дерево $T_w \in G$ со свойствами, которые перечислены выше, по факту задает способ планарной укладки в надъязыечной области фрагментов, которые соответствуют вершинам множества $V_w \subset V$. Будем называть данный подграф решающим деревом (РД). На рис. 7 приводится вариант РД T_1 с планарными укладками фрагментов, которые соответствуют вершинам РД.

Допустим, что каждому из фрагментов f_i соответствует стоимость ε_i , тогда РД $T_w = (V_w, U_w)$ соответствует суммарная стоимость:

$$\psi_w = \sum_i (\varepsilon_i), \quad i|v_i \in V_w.$$

Параметр ψ_w принимается в качестве критерия оптимизации. Цель оптимизации состоит в максимизации параметра ψ_w .

Определим рекурсивно уровень α_{wi} фрагмента f_i (вершины v_i), входящего в состав РД T_w [9]. Вершина $v_i \in V_w$, не содержащая вложенных в нее вершин, имеет нулевой уровень ($\alpha_{wi} = 0$). Пусть μ_{wi} — максимальный уровень среди вершин, вложенных в v_i , следовательно, уровень v_i равен $\mu_{wi} + 1$ ($\alpha_{wi} = \mu_{wi} + 1$).

Отметим, что вершина v_i (фрагмент f_i) в разных РД может иметь разный уровень α_i . Назовем блоком t_{wi} поддереву РД T_w , имеющего своим корнем вершину $v_i \in V_w$. Уровень блока t_{wi} равен уровню вершины $v_i \in V_w$, которая входит в РД T_w . Будем считать, что блок t_{wi} входит в состав блока t_{wj} в случае, если в РД T_w корневая вершина v_i блока t_{wi} является дочерней вершиной корневой вершины v_j блока t_{wi} и $\alpha_{wi} \geq \alpha_{wi} + 1$.

Блоки t_{wi} и t_{wj} будем считать непересекающимися лишь в том случае, когда v_i и v_j не пересекаются друг с другом.

Пусть множеству вершин B_{iw} , являющихся вложением вершины v_i , соответствует множество блоков Q_{iw}

Установим стоимость ω_{wi} блока t_{wi} . Стоимость ω_{wi} блока t_{wi} , располагающегося на ненулевом уровне ($\alpha_{wi} \neq 0$), будет эквивалентна суммарной стоимости множества блоков Q_{iw} , входящих в состав блока t_{wi} , и стоимости φ_i вершины v_i :

$$\omega_{wi} = \varepsilon_i + \sum_j j(\omega_{wj}), j|v_j \in B_{wi}.$$

Вершины $v_i \in V_w$, располагающиеся на нулевом уровне, играют роль блока нулевого уровня, его стоимость равна стоимости v_i ; $\omega_{wi} = \varepsilon_i$.

Поэтому $\psi_w = \omega_{wr}$ — стоимость блока t_{wr} с корнем в вершине v_r равна стоимости РД T_w . В свою очередь, t_{wr} и T_w будут соответствовать друг другу. Чтобы установить число магистралей m_w , используемых для укладки РД T_w , воспользуемся значением уровня α_{wr} корневой вершины ($m_w = \alpha_{wr}$).

2. Однослойная трассировка в НЯО на основе комбинаторной оптимизации "муравьиное дерево"

В работах [6—10] исследователями было предложено описание и проведен анализ имеющихся на сегодняшний день алгоритмов однослойной трассировки. Для построения наиболее эффективного современного алгоритма, соответствующего всем требованиям, прибегают к использованию различных инновационных подходов и технологий. Сейчас активно разрабатывается одно из научных направлений, включающее в себя различные математические методики, построение которых осуществляется непосредственно на базе принципов природных механизмов принятия решений [11].

Для создания РД применяется структура комбинаторной оптимизации. Она получила название "*trees ant colony optimization* (Т-АСО)". Эта структура создается на идеях адаптивного поведения муравьиной колонии [12, 13] и предоставляет возможность выполнить синтез дерева [14]. Оптимизационная задача, имеющая вид структуры Т-АСО, заключается в следующем:

- построение ГПР в виде дерева;
- создание на ГПР роem агентов допустимых деревьев решений;
- выбор лучшего решения.

Задача роя агентов заключается в построении на графе $G = (V, U)$ путем последовательного выбора множества вершин $V_w \subset V$ маршрута M_w , обладающего следующими свойствами. Маршрут M_w является упорядоченной последовательностью подмаршрутов $M_w = \{M_{wi} | i = 1, 2, \dots, m\}$, $\cup_i(M_{wi}) = M_w$. Подмаршрут M_{wi} является упорядоченной последовательностью множества взаимно непересекающихся вершин $V_{wi} \subset V_w$, $\cup_i(V_{wi}) = V_w$. Вершины множества V_{wi} соответствуют множеству фрагментов, размещаемых на i -й магистрали НЯО. Число подмаршрутов соответствует числу магистралей в НЯО. Подграф графа $G = (V, U)$, включаю-

щий множество вершин $V_w \subset V$, фактически является РД $T_w = (V_w, U_w)$, задающим способ планарной укладки фрагментов в надъячеечной области.

3. Построение модифицированного графа поиска решений

В качестве графа поиска решений авторы остановили свой выбор на модифицированном ГПР $G = (V, U)$. Основное отличие модернизированного графа поиска решений заключается в том, что в графе отсутствуют ребра, соответствующие замыканию транзитивных маршрутов вложения. Если v_j вкладывается в v_i , то в графе G существует только один маршрут единичной длины, связывающий v_i и v_j ребром, направленным от v_i к v_j . Разработка графа решений начинается от основной вершины v_r . Данная вершина соответствует верхней точке надъячеечной зоны [14]. Изначально с учетом полученных результатов в ходе проведенного анализа схемы проводится построение множества вершин V_1 . Любая из таких вершин может входить в состав только основной вершины v_r , в нашем примере на рис. 8 это $V_1 = \{v_6, v_8, v_7, v_9, v_{10}\}$. v_r связана ребрами со всеми вершинами множества V_1 . Затем формируется множество V_2 . Любая из вершин множества V_2 может вкладываться только в состав некоторых вершин V_1 . Другими словами, любая из вершин множества V_2 может быть дочерней вершиной только у некоторых вершин V_1 . Между парами $v_i \in V_1$ и $v_j \in V_2$ формируются ребра, имеющие направление от v_i к v_j , если f_j вкладывается в состав f_i . На шаге n с учетом полученных результатов в ходе проведенного анализа схемы проводится построение множества вершин V_n . Любая вершина множе-

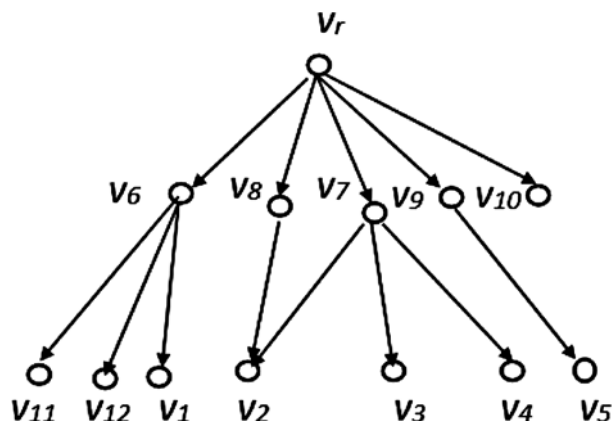


Рис. 8. Граф поиска решений

ства V_n может вкладываться только в некоторые вершины множества V_{n-1} . Такая процедура осуществляется до тех пор, пока в графе G не будут отображены все фрагменты схемы. Граф решений, созданный в качестве примера, отображен на рис. 8. Таким образом, ребра связывают вершины двух соседних уровней.

Результатом работы алгоритма муравьиного дерева (Т-АСО) является РД T_r с наилучшей оценкой на графе решений G . Поиск решения (решающего дерева T_r) сводится к построению маршрута с вышеописанными свойствами на базе графа решений G , который имеет фиксированную структуру, с использованием вспомогательного графа решений GT с трансформирующейся структурой. Исходная структура графа GT идентична структуре графа G . В качестве исходных данных формируется граф пересечений $CR = (V, E)$, между парами вершин $v_i \in E$ и $v_j \in E$ создается ребро, если фрагменты f_i и f_j пересекаются.

Поиск решений задачи осуществляет коллектив агентов $A = \{a_k | k = 1, 2, \dots, n_k\}$. Модели поведения агентов связаны с распределением феромона на вершинах графа G , используемого в качестве хранилища коллективной эволюционной памяти. Первоначально на всех вершинах графа G откладывается равное, небольшое, количество феромона Q/v , здесь $v = |U|$, параметр Q задан априори. Поиск решений является итерационным процессом. На каждой итерации алгоритма муравьиного дерева каждый агент a_k формирует свое решение задачи на трансформирующемся графе решений GT , используя в качестве хранилища коллективной памяти граф G . Интерпретацией решения является РД $T_k \subset G$, которому будет соответствовать эскиз планарной трассировки в НЯО. Итерация l состоит из трех этапов.

4. Построение агентом решения (маршрута M_k и соответствующего решающего дерева $T_k \subset G$)

Первый этап каждой итерации заключается в нахождении каждым агентом популяции своего решения (маршрута M_k и дерева T_k) посредством последовательной трансформации исходного вспомогательного графа решений $GT = (V, U)$. Второй этап — агенты откладывают феромон на вершинах графа поиска решений G . Третий этап состоит в испарении феромона на вершинах графа поиска решений G . В работе применяется циклический "ant-cycle"-метод муравьиной системы, при этом феромон

откладывается агентами на вершинах графа решений G после того, как будут сформированы все решения популяции.

Решение находится в процессе последовательного просмотра вершин дерева решений.

Формирование агентом a_k маршрута M_k и соответствующего РД $T_k \subset G$ осуществляется путем последовательного распределения вершин V графа G по магистралям, начиная с первой, нумерация магистралей — снизу вверх. После выбора очередной магистрали формируется для размещения на ней множество SK вершин-кандидатов.

При заполнении первой магистрали в состав множества кандидатов SK_1 входят только дочерние вершины графа $GT(0)$. Для ГПР (рис. 8) $SK_1 = \{v_{11}, v_{12}, v_1, v_2, v_3, v_4, v_5, v_{10}\}$.

С помощью процедуры *RAZM* агент $a_k \in A$ формирует максимальное по стоимости подмножество $SK_1^* \subset SK_1$ взаимно-непересекающихся вершин множества SK_1 для размещения на первой магистрали: $SK_1^* \subset \{v_{11}, v_1, v_3, v_4, v_5\}$.

На графе пересечений $CR = (V, E)$ определяется множество D_1 вершин, пересекающиеся с вершинами множества SK_1^* . Для нашего примера $D_1 = \{v_{12}, v_8, v_2, v_9, v_{10}\}$.

После этого выполняется трансформация графа $GT(0)$ в граф $GT(1)$ путем удаления из $GT(0)$ подмножеств вершин D_1 и SK_1^* со всеми инцидентными этим вершинам ребрами графа $GT(0)$.

Далее в графе $GT(1)$ формируется множество вершин SK_2 , в состав которого входят только дочерние вершины графа $GT(0)$, и осуществляется переход к заполнению магистрали 2, $\gamma = 2$. Для нашего примера $SK_2 = \{v_6, v_7\}$.

При заполнении магистрали γ в множество кандидатов SK_γ входят только дочерние вершины графа $GT(\gamma - 1)$. С помощью процедуры *RAZM* агент формирует максимальное по стоимости подмножество SK_γ^* взаимно-непересекающихся вершин множества SK_γ для размещения в магистрали γ .

На графе пересечений $CR = (V, E)$ определяется множество вершин D_γ , пересекающиеся с вершинами множества SK_γ^* .

Далее проводится трансформация графа $GT(\gamma - 1)$ в граф $GT(\gamma)$ путем удаления из $GT(\gamma - 1)$ подмножества вершин D_γ и SK_γ^* со всеми инцидентными этим вершинам ребрами графа $GT(\gamma - 1)$. Формируется множество вершин $SK_{\gamma+1}$ и выполняется переход к формированию уровня $\gamma = \gamma + 1$.

Распределение агентом a_k вершин по магистралям прекращается после того, как в ре-

результате трансформаций граф $GT(\gamma)$ станет пустым ($GT(\gamma) = \emptyset$) и, следовательно, $SK_{\gamma+1} = \emptyset$. Обозначим $V_k(l)$ — множество вершин графа G , распределенных агентом a_k по γ магистралям на итерации l . РД $T_k(l) \subset G$ формируется следующим образом. Множество вершин $V_k(l) \subset G$ графа G помечаются. Затем из графа удаляются все непомеченные вершины со всеми инцидентными этим вершинам ребрами графа G . Оставшаяся часть графа G является решающим деревом $T_k(l) \subset G$.

После этого подсчитывается суммарная стоимость $\psi_k(l)$ множества вершин $V_k(l)$, распределенных агентом по магистралям, которая является оценкой решения.

Рассмотрим процедуру *RAZM*. Упорядочим множество вершин SK_γ — кандидатов для размещения на магистрали по возрастанию координат левых концов фрагментов, соответствующих этим вершинам. Построим вектор P . Пусть $P(v_i)$ — порядковый номер v_i в векторе P . Тогда $\forall ij\{x(l_i) < x(l_j)\} \rightarrow \{P(v_i) < P(v_j)\}$.

Упорядочим вершины множества SK_γ в линейку в том порядке, в каком соответствующие им фрагменты расположены в векторе P .

Формирование агентом $a_k \in A$ максимального по стоимости подмножества $SK_\gamma^* \subset SK_\gamma$ взаимно-непересекающихся вершин множества SK_γ для размещения на γ -м уровне осуществляется последовательно (пошагово) путем последовательного просмотра элементов упорядоченного списка SK_γ .

На шаге t в памяти агента a_{tk} хранится номер заполняемой магистрали γ , множество SK_γ , список $SK_\gamma^*(t)$ вершин, уже включенных в формируемый уровень, v_z — последняя вершина (фрагмент f_z), включенная в список $SK_\gamma^*(t)$.

Агент просматривает множество всех свободных на данном шаге вершин $SK_\gamma \setminus SK_\gamma^*(t)$, размещенных в списке SK_γ справа от v_z , и выделяет из него подмножество вершин $Z_\gamma(t) \subset SK_\gamma \setminus SK_\gamma^*(t)$, удовлетворяющих условиям размещения фрагментов в текущей магистрали.

Условия размещения:

а) каждая вершина $v_i \in Z_\gamma(t)$ может быть размещена на текущей заполняемой магистрали без пересечений с уже размещенными вершинами множества $SK_\gamma^*(t)$;

б) если $f_i \in Z_\gamma(t)$ поместить в магистраль, то не существует фрагмента $f_j \in Z_\gamma(t)$, который может быть размещен в оставшейся свободной зоне заполняемой магистрали между фрагментами f_z и f_i без нарушения ограничений и конфликтов (пересечений) с f_i .

Соблюдение рассмотренных условий объясняется следующим образом. Поскольку заполнение магистралей фрагментами проводится последовательно с использованием "безвозвратной" стратегии, нет смысла оставлять на магистрали не заполненными (пустыми) области, в которые могут быть помещены некоторые фрагменты.

Для каждой, отвечающей условиям размещения, вершины $f_i \in Z_\gamma(t)$ рассчитываются три параметра φ_i , λ_i , ε_i :

φ_i — суммарное количество феромона, скопившегося на вершине v_i ;

λ_i — расстояние между окончанием последнего в заполняемой магистрали фрагмента f_z и началом фрагмента f_i , размещенных в одной магистрали. Если f_i располагается в начале магистрали, то d_i — расстояние между началом магистрали и началом f_i ;

ε_i — стоимость фрагмента f_i .

Интегральная стоимость $Q_{iz}(t)$ связи вершины $f_i \in SK_\gamma^*(t)$ с $f_z \in SK_\gamma^*(t)$ определяется при мультипликативной свертке по формуле

$$Q_{iz}(t) = (\varphi_i^\alpha (l-1) \varepsilon_i^\beta) / (\lambda_i^\delta + 1), \quad (1)$$

а при аддитивной свертке — по формуле

$$Q_{iz}(t) = \alpha \varphi_i (l-1) + (\delta / (\lambda_i + 1) + \beta \varepsilon_i), \quad (2)$$

где α , β , δ — управляющие параметры, которые подбираются экспериментально.

Вероятность $P_i(t)$ включения вершины $f_i \in Z_\gamma(t)$ в формируемый маршрут $SK_\gamma^*(t)$ определяется следующим соотношением:

$$P_i(t) = Q_{iz}(t) / \theta, \text{ где } \theta = \sum_i (Q_{iz}(t)), i | f_i \in Z_\gamma(t). \quad (3)$$

Рассмотрим эвристические соображения, положенные в основу формул (1) и (2). Чем больше $\varphi_i(l-1)$, тем больше вероятность того, что вершина f_i входит в состав оптимального маршрута. Чем меньше λ_i , тем выше плотность заполнения магистрали, что способствует уменьшению числа магистралей, необходимого для размещения фрагментов.

Учет в выражениях (1) и (2) параметра ε_i способствует максимизации стоимости подмножества $SK_\gamma^* \subset SK_\gamma$ взаимно-непересекающихся вершин множества SK_γ для размещения на γ -й магистрали, что косвенным образом способствует повышению эффективности поиска и минимизации целевой оценки — числа магистралей.

Агент с вероятностью $P_i(t)$ выбирает одну из вершин $f_i \in Z_\gamma(t)$, которая размещается на γ -й магистрали. После этого множество размещенных вершин уже обозначается как $SK_\gamma^*(t+1)$, а f_i исключается из множества $Z_\gamma(t)$. При $\alpha = 0$ наиболее вероятен выбор ближайшего к f_z фрагмента f_i , т. е. алгоритм становится подобным алгоритму "левого конца". При $\beta = 0$ вероятность выбора практически зависит только от количества феромона на ребре, что приводит к субоптимальным решениям. Соотношение между этими величинами находится экспериментально.

После формирования всеми агентами деревьев решений на вершинах ГПР откладывается феромон. Каждый агент откладывает на вершинах $V_k(l) \subset V$ феромон в количестве $\tau_k(l)$, пропорциональном оценке $\psi_k(l)$ РД:

$$\tau_k(l) = c\psi_k(l), \quad (4)$$

где c — числовой коэффициент.

Обозначим $\phi_i(l)$ — общее количество феромона, скопившегося на вершине $v_i \in V$ в ГПР G после выполнения агентами второго этапа l -й итерации.

На третьем этапе итерации на вершинах графа $R_\tau^*(F, E)$ в соответствии со следующей формулой происходит испарение феромона.

$$\tau_k(l) = \tau_k(l)(1 - \rho), \quad (5)$$

где ρ — коэффициент обновления.

Последние действия на итерации связаны с нахождением лучшего решения, которое запоминается. После этого выполняется следующая итерация. Временная сложность этого алгоритма зависит от числа вершин графа n , числа муравьев u , числа итераций l и определяется как $O(n^2lu)$.

5. Исследование алгоритма трассировки соединений в приканальной надъячеечной области

Целью эксперимента было определение числа итераций муравьиного алгоритма, которое с большой вероятностью (близкой к единице) гарантировало бы нахождение оптимального решения.

Вначале были проведены исследования, направленные на поиск значения управляющих параметров, которые смогли бы обеспечить

наибольшую эффективность работы НЯО-МД. Результаты для муравьиного алгоритма получены при следующих количественных значениях:

- агенты — 100;
- итерации — 120;
- начальное отложение феромона — 1000;
- количество феромона, откладываемого агентом на решении — 100;
- коэффициент испарения феромона — 0,7.

График зависимости трудоемкости алгоритма трассировки в НЯО от числа цепей изображен на рис. 9.

Во всех решенных примерах получены оптимальные результаты. Трудоемкость алгоритма: $O(n^2)$.

В таблице приведены значения для десяти тестовых примеров, полученные трассировщиком без использования НЯО, и результаты, полученные трассировщиком с использованием разработанного муравьиного алгоритма трассировки в НЯО.

Успех работы алгоритма зависит от правильной настройки параметров управления. Источ-

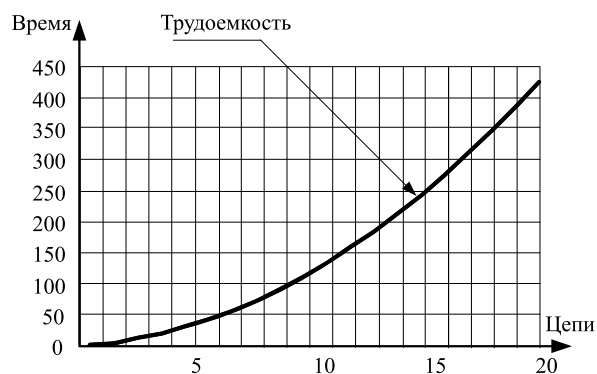


Рис. 9. Зависимость трудоемкости алгоритма трассировки в НЯО от числа цепей

Результаты экспериментальных исследований

№ примера	Без НЯО (число магистралей)	С НЯО (число магистралей)	Процент разгрузки канала
1	14	12	14,3 %
2	12	10	16,7 %
3	14	9	35,7 %
4	20	16	20 %
5	24	21	12,5 %
6	25	22	12 %
7	29	24	17,2 %
8	32	26	20,8 %
9	33	26	21,2 %
10	35	28	25 %

ником усовершенствования работы муравьиного алгоритма может стать использование более сложных модифицированных стратегий выбора альтернатив при прокладке маршрутов.

Проведен синтез ряда примеров с заранее известными оптимальными значениями целевых функций для анализа точности решений, получаемых с использованием данного алгоритма.

При этом вероятность нахождения оптимальных решений составляет 0,93, а общая оценка временной сложности находится в пределах $O(n^2) \dots O(n^3)$.

Заключение

В работе задача повышения трассируемости решается путем диверсификации трассировочных ресурсов, заключающейся в использовании дополнительных областей, примыкающих к каналам снизу и сверху, с трассировочным слоем, который располагается над ячейками — "over-the-cell". Тогда часть соединений может быть перенаправлена из канала в НЯО, что приведет к сокращению числа магистралей, которые используются в канале. Перенаправленные соединения трассируются в одном слое. Целью надъячеечного трассировщика является создание максимального планарного эскиза топологии в НЯО, это должно позволить минимизировать плотность канала.

В целях построения эскиза однослойной трассировки использована структура комбинаторной оптимизации "муравьиное дерево". В качестве графа поиска решений использован упорядоченный модифицированный граф решений $G^* = (V, U)$, его построение осуществляется путем рекурсивного определения возможного вложения фрагментов друг в друга. Отличием от канонической структуры муравьиного алгоритма является то, что муравьем на графе поиска решений создается не маршрут, а дерево, которое по своей структуре совпадает с представлением решения. Это исключает применение дополнительных трансформаций в процессе декодирования решений, допускающих интерпретацию решений в виде деревьев, и позволяет отбросить множество "нелегальных" решений, что приводит к повышению качества полученных решений. Использование процедуры трассировки в НЯО позволяет разгрузить канал на 15...20 %. Временная сложность этого алгоритма зависит от числа вершин графа n , числа агентов u , числа итераций l и определяется как $O(n^2ly)$.

Список литературы

1. Немудров В., Мартин Г. Системы-на-кристалле. Проектирование и развитие. М.: Техносфера, 2004. 157 с.
2. Bozhko A. N., Karpenko A. P. Computer-aided Subassembly Generation // Proceedings of the Vth International workshop "Critical infrastructures: Contingency management, Intelligent, Agent-based, Cloud computing and Cyber security" (IWCI 2018). 2018. DOI:10.2991/iwci-18.2018.2.
3. Божко А. Н., Родионов С. В. Методы искусственного интеллекта в автоматизированном проектировании процессов сборки // Наука и образование. МГТУ им. Н. Э. Баумана. Электрон. журн. Эл № ФС 77-48211. 2016. № 8. DOI: 10.7463/0816.0844719.
4. Карпенко А. П., Волосатова Т. М., Маничев В. Б., Грошев С. В., Жук Д. М., Божко А. Н., Мартынюк В. А., Пивоварова Н. В., Трудоношин В. А., Норенков И. П. Основы автоматизированного проектирования. М.: ООО "Научно-издательский центр ИНФРА-М", 2015. 329 с.
5. Литвинов Е. И., Жихарев Г. Ю., Шагурин И. И. Возможности использования виртуальных платформ для верификации RTL-моделей сложнофункциональных блоков в составе "Систем-на-кристалле" // Проблемы разработки перспективных микро- и нанoeлектронных систем — 2014. Сборник трудов. М.: ИППМ РАН, 2014. Часть 2. С. 51—56.
6. Быков С. О. Автоматизация проектирования сетей-на-кристалле со специализированной топологией // Проблемы разработки перспективных микро- и нанoeлектронных систем — 2014. Сборник трудов. М.: ИППМ РАН. 2014. Часть 2. С. 21—24.
7. Казеннов Г. Г. Основы проектирования интегральных схем и систем. М.: Бином. Лаборатория знаний, 2005. 295 с.
8. Гарбулина Т., Лялинская О. В., Хватов В. М. Повышение эффективности проектирования интегральных схем на ПЛИС с ограниченными трассировочными ресурсами // Проблемы разработки перспективных микро- и нанoeлектронных систем — 2016. Сборник трудов. М.: ИППМ РАН. 2016. № 1. С. 165—171.
9. Лебедев О. Б., Пурчина О. А. Роевой алгоритм трассировки в приканальной надъячеечной области // Известия Южного федерального университета. Технические науки. Ростов-на-Дону: Изд-во ЮФУ, 2015. № 6, С. 125—138.
10. Лебедев Б. К., Лебедев О. Б., Пурчина О. А. Роевой алгоритм повышения плотности топологии СБИС // Сборник научных трудов VIII Международной научно-практической конференции "Интегрированные модели и мягкие вычисления в искусственном интеллекте". Т. 2. М.: Изд-во Физматлит. 2015. С. 624—634.
11. Карпенко А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. М: Издательство МГТУ им. Н. Э. Баумана, 2014. 448 с.
12. Dorigo M., Stützle T. Ant Colony Optimization. MIT Press. Cambridge MA, 2004. 187 p.
13. Лебедев О. Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования. Таганрог: Изд-во ЮФУ. 2013. 199 с.
14. Лебедев Б. К., Лебедев О. Б., Пурчина О. А. Однослойная трассировка в приканальной надъячеечной области на основе алгоритма муравьиного дерева // Труды конгресса по интеллектуальным системам и информационным технологиям "IS—IT'14". Науч. изд. в 4-х томах. М.: ФИЗМАТЛИТ. Т. 1. 2014. С. 58—69.
15. Рыженко Н. В., Быков С. А., Сорокин А. А. Трассировка битовых элементов памяти с автоматическим построением ограничений на границах ячеек // Проблемы разработки перспективных микро- и нанoeлектронных систем — 2016. Сб. тр. М.: ИППМ РАН. 2016. № 1. С. 144—150.

B. K. Lebedev, Doctor of Engineering Sciences, Professor, e-mail: lebedev.b.k@gmail.com,
O. B. Lebedev, Candidate of Engineering Sciences, Associate Professor, e-mail: lebedev.ob@mail.ru,
Southern Federal University, Rostov-on-Don, 344006, Russian Federation,
O. A. Purchina, Assistant, e-mail: knagna_olga@inbox.ru,
Don State Technical University, Rostov-on-Don, 344000, Russian Federation

Modified Swarm Algorithm "Ant Tree" in the Problem of Diversification of Tractor Resources

A modified swarm algorithm (trees ant colony optimization (T-ACO)) is proposed. The difference from the canonical structure is that the decision tree is used as a decision search graph. An agent for the decision search graph does not create a route, but a tree, which in its structure coincides with the representation of the solution to the trace problem in the "over-the-cell" region. This eliminates the use of additional transformations in the decoding process of decisions that allow interpretation of decisions in the form of trees and allows you to discard a lot of "illegal" solutions, which leads to an increase in the quality of the solutions obtained. Using the tracing procedure in "over-the-cell" based on the ant algorithm allows you to unload the channel by 15–20 %. The time complexity of the algorithm depends on the number of vertices n of the decision search graph, the number of agents y , the number of iterations l , and is defined as $O(n^2ly)$.

Keywords: optimization, "ant tree" algorithm, CAD VLSI, diversification of trace resources, "over-the-cell" area, planar sketch of topology

Acknowledgements: This work was supported by the Russian Foundation for Basic Research (Grant № 18-07-00737a).

DOI: 10.17587/it.25.670-678

Reference

1. Nemudrov V., Martin G. System-on-a-chip. Design and development, Moscow, Technosphere 2004, 157 p. (in Russian).
2. Bozhko A. N., Karpenko A. P. Computer-aided Subassembly Generation // *Proceedings of the Vth International workshop "Critical infrastructures: Contingency management, Intelligent, Agent-based, Cloud computing and Cyber security" (IWCI 2018)*, 2018, DOI:10.2991/iwci-18.2018.2.
3. Bozhko A. N., Rodionov S. V. Methods of artificial intelligence in computer-aided design of the assembly process, *Nauka i obrazovaniye. MGTU im. N. E. Bauman. Elektron. zhurn. № FS 77-48211*, 2016, № 8. DOI: 10.7463 / 0816.0844719 (in Russian).
4. Karpenko A. P., Volosatov T. M., Manichev V. B., Groshev S. V., Zhuk D. M., Bozhko A. N., Martynyuk V. A., Pivovarova N. V., Trudonoshin V. A., Norenkov I. P. Fundamentals of computer-aided design: a tutorial, Moscow, Scientific Publishing Center INFRA-M, 2015, 329 p. (in Russian).
5. Litvinov E. I., Zhikharev G. Yu., Shagurin I. I. Possibilities of using virtual platforms for verification of RTL-models of complex functional units in the "Systems on a chip", *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem — 2014, Sbornik trudov*, 2014, no. 2, pp. 51–56 (in Russian).
6. Bykov S. O. Automation of network-on-chip design with a specialized topology, *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem — 2014. Sbornik trudov*, 2014, no. 2, pp. 21–24 (in Russian).
7. Kazennov G. G. Fundamentals of design of integrated circuits and systems, Moscow, Binom, Laboratory of knowledge, 2005, 295 p. (in Russian).
8. Garbulina T., Lyalinskaya O. V., Khvatov V. M. Improving the efficiency of designing integrated circuits on a FPGA with limited trace resources, *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem — 2016, Sb. trudov*, 2016, no. 1, pp. 165–171 (in Russian).
9. Lebedev O. B., Purchina O. A. The swarm tracing algorithm in the near-canal supracellular region, *Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskiye nauki*, 2015, no. 6, pp. 125–138 (in Russian).
10. Lebedev B. K., Lebedev O. B., Purchina O. A. Swarm algorithm for increasing the density of the VLSI topology, *Sbornik nauchnykh trudov VIII Mezhdunarodnoy nauchno-prakticheskoy konferentsii "Integrirovannyye modeli i myagkiye vychisleniya v iskusstvennom intellekte"*, 2015, vol. 2, pp. 624–634 (in Russian).
11. Karpenko A. P. Modern search engine optimization algorithms. Algorithms inspired by nature: a tutorial, Moscow, Publishing House MSTU N. E. Bauman, 2014, 448 p. (in Russian).
12. Dorigo M., Stützle T. Ant Colony Optimization, MIT Press, Cambridge MA, 2004, 187 p.
13. Lebedev O. B. Models of adaptive behavior of an ant colony in design problems, Rostov-on-Don, Publishing House of Southern Federal University, 2013, 199 p. (in Russian).
14. Lebedev B. K., Lebedev O. B., Purchina O. A. Single-layer tracing in the canal supracellular area based on the ant tree algorithm, *Trudy kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam "IS-IT'14"*, 2014, vol. 1, pp. 58–69 (in Russian).
15. Ryzhenko N. V., Bykov S. A., Sorokin A. A. Tracing of bit memory elements with automatic construction of constraints on cell borders, *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem — 2016. Sbornik trudov*, 2016, no.1, pp. 144–150 (in Russian).