УДК 681.3.016 DOI: 10.31114/2078-7707-2020-3-79-86

Ко-эволюционный алгоритм разбиения СБИС

Б.К. Лебедев, О.Б. Лебедев

Южный федеральный университет, г. Таганрог, lebedev.ob@mail.ru

***Аннотация* — Для повышения эффективности, усиления сходимости алгоритма и способности выхода из локальных оптимумов авторами предложен коалиционный подход к построению алгоритма разбиения. Для решения задачи разбиения авторами разработана модифицированная метаэвристика по аналогии с моделями адаптивного поведения муравьиной колонии. Интерпретацией решения является двудольный подграф $D^k=(X^k \cup W^k, E^k)$, задающий распределение множества вершин X^k по узлам множества W^k. Работа алгоритма фактически заключается в формировании подмножества ребер E^k. Временная сложность процедуры на одной итерации – $O(n^2)$. Разработанные алгоритмы находит решения, не уступающие по качеству, а иногда и превосходящие свои аналогов в среднем на 3-4%.**

***Ключевые слова* — СБИС, разбиение, роевой интеллект, муравьиный алгоритм, адаптивное поведение, субпопуляция, ко-эволюция, оптимизация.**

I. Введение

При решении конструкторских задач, возникающих при проектировании, часто приходится сталкиваться с задачей разбиения интегральных схем на отдельные блоки. При решении подобных задач широко применяется разбиение крупных схем на составные блоки, включающие в себя определенные наборы компонентов. Это связано с тем, что современная СБИС может содержать десятки миллионов транзисторов. Такая схема не может быть спроектирована в целом в связи с ограниченными вычислительными способностями (память, скорость). В результате разбиения формируется множество блоков и межсоединений между блоками. В очень больших схемах также может использоваться иерархическая структура разбиения [1], [2] Для сокращения времени решения задачи разбиения используются различные эвристические методы. В качестве модели схемы используется либо гиперграф $H(X, U)$, где X – множество вершин, $|X|=n$, U – множество ребер, либо двудольный граф $D=(X \cup W,Y)$, где X – множество элементов (первая доля), $|X|=n$, $W=\{w_i|i=1,2,\ldots,n\}$ – множество узлов (вторая доля), формируемых при разбиение множества элементов X. Y – множество ребер, связывающих вершины множества X с вершинами множества W.

При проектировании СБИС используются два подхода к размещению на кристалле сформированных узлов. При первом подходе разбиение схемы на узлы осуществляется без учета возможного размещения узлов на кристалле. Формирование конфигураций узлов и областей их размещения на кристалле осуществляется после решения задачи разбиения. При втором подходе области размещения областей и их размеры определяются по результатам планирования СБИС. Разбиение фактически сводится к распределению множества вершин по областям, размещенным на кристалле СБИС. Поэтому при втором подходе помимо традиционного учета числа связей прогнозируется длина связей.

Существующие алгоритмы разбиения делятся на два основных класса: конструктивные и итеративные [1-3]. Первый класс характеризуется относительным быстродействием, но в тоже время низким качеством решения. Второй класс является более трудоемким, однако позволяет получать более качественные решения [3].

Среди итеративных алгоритмов можно выделить детерминированные и вероятностные. В детерминированных алгоритмах изменение разбиения (решения) реализуется на основе четкой, детерминированной зависимости от изменяемого решения. Недостатком является частое попадание в локальный оптимум («локальную яму»).

В вероятностных алгоритмах переход к новому решению осуществляется случайным образом. Недостатком алгоритмов, реализующих чисто случайный поиск, является значительная трудоемкость [4]. В последние годы интенсивно разрабатывается научное направление с названием «Природные вычисления», объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [4]. К таким методам можно отнести, прежде всего, методы моделирования отжига, метод эволюционного моделирования, генетические алгоритмы, эволюционной адаптации, алгоритмы роевого интеллекта и муравьиные алгоритмы [5-8]. В работе предлагается гибридный подход, способы и методы представления задачи разбиения СБИС, базирующийся на моделировании адаптивного поведения муравьиной колонии (МК).

В гибридных алгоритмах преимущества одного алгоритма могут компенсировать недостатки другого. Интеграция метаэвристик популяционных алгоритмов обеспечивает более широкий обзор пространства поиска и более высокую вероятность локализации глобального экстремума задачи. Такой подход позволяет частично решить проблему преждевременной сходимости, обеспечивает выход из локальных оптимумов и повышает скорость получения

результата. Для решения задачи разбиения разработан ко-эволюционный алгоритм, основанный на методе МК. Суть ко-эволюционного подхода состоит с том, что эволюционирующая популяция делится на субпопуляции, которые эволюционируют параллельно [9], [10]. Периодически агенты перемещаются из одной субпопуляции в другую, передавая свой опыт. Ко-эволюционный подход обеспечивает более широкий обзор пространства решений и более высокую вероятность локализации глобального экстремума задачи.

II. Постановка задачи разбиения

Пусть дан граф $G(X, U)$, где X – множество вершин, $|X|=n$, U – множество ребер. Необходимо разбить множество X на k непустых и непересекающихся подмножеств X_i, $\cup X_i=X$, $X_i \cap X_j=\varnothing$, $X_i \neq \varnothing$. На формируемые узлы (блоки, компоненты) накладываются ограничения: $|X_i|=n_i$, $\sum n_i=n$. Критерий оптимизации – суммарное число связей F между X_i. Цель оптимизации – минимизация критерия F. На рис.1 представлен план, разбитого на области чипа, и модель связей вершины x_i вершинами графа.

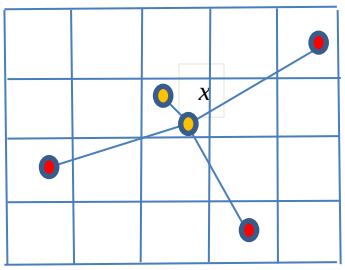

Рис. 1. Модель связей вершины x_i

В качестве координат расположения вершины, расположенной в области, рассматриваются координаты центральной точки области. При подсчете оценки вершины рассматриваются соединения, связывающие вершины, размещенные в разных областях. При заданном размещении областей на чипе и распределении вершин по областям определяется стоимость ξ_i каждой вершины x_i.

$$\xi_i=k_1 d_i+ k_2 s_i,$$

где d_i – суммарное число соединений, инцидентных вершине x_i,

s_i – суммарная длина соединений, инцидентных вершине x_i,

k_1, k_2 – коэффициенты.

Для второго подхода к разбиению оценка решения задачи распределения вершин по областям определится как:

$$\xi=\sum_i (\xi_i).$$

Метаэвристика муравьиного алгоритма (МА) основывается на комбинации двух техник: общая схема строится на базовом методе, в которую включается встроенная процедура. Базовый метод заключается в реализации итерационной процедуры поиска лучшего решения, на основе механизмов адаптивного поведения муравьиной колонии. Встроенная процедура – это конструктивный алгоритм построения агентом некоторой конкретной интерпретации решения [7].

С помощью МА решается задача нахождения кратчайшего маршрута в полном графе. Каждый муравей колонии формирует свой маршрут, являющийся интерпретацией решения некоторой задачи на полном графе поиска решений (ГПР). При движении муравей метит путь феромоном, и эта информация используется другими муравьями для выбора пути.

Авторами разработаны модификации канонической парадигмы муравьиного алгоритма: структура ГПР и конструкция представления (интерпретация) решения на нем; методы построения агентом на ГПР интерпретации решения (разработка конструктивного алгоритма).

Для повышения эффективности, усиления сходимости алгоритма и способности выхода из локальных оптимумов предложен подход к ко-гибридизации [9], [10] алгоритма разбиения на основе модели адаптивного поведения муравьиной колонии. Одновременно в пространстве поиска решения задачи эволюционируют несколько субпопуляций, каждая из которых решает исходную оптимизационную задачу и имеет свою стратегию оптимизации. Решение задачи разбиения осуществляется двумя группами A_1, A_2 агентов. Агенты группы A_1 строят маршруты в соответствии со стратегией C_1. Соответственно: агенты группы A_2 – со стратегией C_2. Стратегии C_1 и C_2 отличаются конструктивными алгоритмами разбиения схемы агентами.

III. Разработка ко-эволюционного алгоритма разбиения на основе моделей адаптивного поведения муравьиной колонии

В качестве графа поиска решений используется полный двудольный граф $D=(X \cup W,E)$, где $X=\{x_i|i=1,2,...,n\}$ – множество вершин (первая доля), соответствующих множеству вершин графа $H(X,U)$, а $W=\{w_v|v=1,2,...,s\}$ – множество вершин (вторая доля), соответствующих множеству узлов $|W|=s$. E – множество ребер (x_i, w_v) связывающих вершины $x_i \in X$ с вершинами множества W. $|E|=ns$.

Задача конструктивного алгоритма сводится к поиску на полном двудольном графе $D=(X \cup W,E)$ двудольного подграфа $D^k=(X \cup W,E^k)$, для которого оценка F имеет минимальное значение. Двудольный подграф D^k используется в качестве представления некоторого k-го решения задачи разбиения – задает распределение множества вершин X по множеству узлов W. Основные ограничения представленного двудольного подграфа $D^k=(X \cup W,E^k)$ заключаются в следующем:

– $E^k=\cup E^k_v$;

– число ребер множества E^k_v, инцидентных вершине w^k_v, равно числу вершин множества X^k_v, $|X_v|=n_v$;

– каждое ребро $(x_i, w_v)=e^k_{iv}\in E^k_v$, с одной стороны, инцидентно вершине $w_v\in W$, с другой стороны – инцидентно одной и только одной вершине $x_i\in X_v$; $\cup X_v=X$;

– локальная степень любой вершины $x_i\in X$ равна единице, $\rho(x_i)=1$;

– локальная степень вершины w_v равна мощности множества X_v, $\rho(w_v)=|X_v|$.

В работе рассматриваются два конструктивных алгоритма разбиения, входящих в состав поискового алгоритма разбиения. Агенты субпопуляции A_1 строят решение используя первый конструктивный алгоритм на множестве вершин X^k. Агенты субпопуляции A_2 - используя второй конструктивный алгоритм на множестве вершин W.

Процесс поиска решений итерационный. Каждая итерация l включает три этапа. На первом этапе каждой итерации каждый агент z_k формирует свое собственное решение, на втором этапе агенты откладывают на ребрах графа поиска решений D феромон, на третьем этапе осуществляется испарение феромона. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромон откладывается агентами на ребрах после полного формирования решения.

Формирование подграфа D^k осуществляется путем последовательного по шагам формирования множества E^k ребер двудольного подграфа $D^k=(X\cup W,E^k)$, на базе множества E ребер полного двудольного графа $D=(X\cup W,E)$ (пошагово).

Множество ребер E^k на базе исходного множества вершин $X\cup W$ образуют двухдольный граф разбиения $D^k=(X\cup W,E^k)$, являющийся решением задачи разбиения.

Рассмотрим первый конструктивный алгоритм разбиения.

Агенты обладают памятью. На каждом шаге t в памяти агента z_k имеется:

– количество феромона $f_{iv}(t)$ отложенного на каждом ребре $e_{iv}=(x_i, w_v)$ графа D;

– список вершин $X_{1k}(t)\in X$, уже включенных в формируемый двудольный подграф $D^k(t)$, и список оставшихся (свободных) вершин $X_{2k}(t)\in X$, $X_{1k}(t)\cup X_{2k}(t)=X$;

– $\rho(w_v(t))$ локальная степень вершины $w_v(t)$ графа $D^k(t)$ на шаге t.

На начальном этапе на всех ребрах графа $D=(X\cup W,E)$ откладывается одинаковое (небольшое) количество феромона $\Theta=\xi/\varepsilon$, где $\varepsilon=|E|$.

Приводится в начальное состояние память агента: $X_{1k}(1)=\varnothing$. $X_{2k}(1)=X$. $E^k(1)=\varnothing$. $\forall v(\rho_v(1)=0$. Отметим, что $\rho_v(t)$ растет с увеличением t. $\hat{W}(1)=W$.

На первом этапе каждой итерации каждый агент z_k формирует свой собственный двудольный подграф D^k. Каждый из агентов z_k выполняет последовательное формирование набора ребер E^k, где k – номер агента.

Процесс формирования E^k включает две стадии, выполняемые на каждом шаге t.

На первой стадии шага t формируется множество вершин $\hat{W}(t)\subset W$ таких, что для каждой вершины $w_v\in\hat{W}(t)$ ее текущая локальная степень $\rho_v(t)<n_v$.

Для каждой вершины $x_i\in X_{2k}(t)$ определяется набор $U_i(t)$ инцидентных ей ребер, связывающих x_i со всеми вершинами w_v множества $\hat{W}(t)$.

На первой стадии шага t для каждой вершины $x_i\in X_{2k}(t)$ определяется набор инцидентных ей ребер $U_i(t)$, связывающих x_i со всеми вершинами w_v множества W. $|U_i(t)|=|W|=s$.

Для каждого ребра $e_{iv}=(x_i,w_v)\in U_i(t)$, инцидентного вершине $x_i\in X_{2k}(t)$, определяется параметр $f_{iv}(t)$ – суммарный уровень феромона на этом ребре e_{iv}.

Для каждой вершины $x_i\in X_{2k}(t)$ определяется среднее количество феромона приходящееся на одно ребро множества $U_i(t)$:

$$f^*(U_i(t))=\sum_v(f_{iv}(t))/s.$$

Величина $\mu_{ik}(t)=f^*(U_i(t))$ объявляется стоимостью вершины x_i. После этого агент z_k с вероятностью $P_{ik}(t)=\mu_{ik}(t)/\sum_i(\mu_{ik}(t))$, пропорциональной $\mu_{ik}(t)$, выбирает вершину $x^*_i\in X_{2k}(t)$.

На второй стадии шага t среди ребер $U_i(t)$, инцидентных выбранной вершине x^*_i, с вероятностью $P_{vk}(t)=f_{iv}(t)/\sum_v(f_{iv}(t))$, выбирается ребро $e^*_i=(x^*_i(t),w_v)$, которое включается в формируемое агентом z_k множество ребер $E^k(t)$.

Далее выполняются постпроцедуры.

Ребро $e^*_{iv}=(x^*_i, w^k_v)$ включается в множество $E^k(t+1)$.

Вершина x^*_i, включается в множество $X_{1k}(t)$ и исключается из множества $X_{2k}(t)$.

Локальная степень вершины $w^k_v\in e^*_{iv}$ в графе $D^k(t)$ увеличивается на 1, $\rho(w_v)(t)=\rho(w_v)(t)+1$.

Переход к следующему шагу.

Процесс формирования агентом zk множества Ek завершается при X2k(t)=∅.

Рассмотрим второй конструктивный алгоритм разбиения на множестве вершин Wk.

Введем обозначения.

Пусть Ŵ(t) список вершин wv, рассматриваемых на шаге t.

ρ(wv)(t) – локальная степень вершины wv∈Ŵ(t) на шаге t.

Для всех $w_v \in \hat{W}(1)$ исходная локальная степень вершины w_v определяется как $\rho_v(1) = n$. Отметим, что $\rho_v(t)$ уменьшается с ростом t.

$Q_v(t)$ – множество ребер $e_{iv}=(x_i, w_v)$, инцидентных вершине $w_v \in \hat{W}(t)$на шаге t.

Приводится в начальное состояние память агента: $\hat{W}(1)=W$.

Для каждой вершины $w_v \in \hat{W}(1)$ формируется начальное множество $Q_v(1) \in E$ инцидентных ей ребер полного двудольного графа D.

$E^k(1)=\varnothing$. $\forall v(\rho_v(1)=n$.

Процесс формирования E^k включает две стадии, выполняемые на каждом шаге t.

На первой стадии шага t в каждом из сформированных на предыдущих шагах множеств $Q_v(t)$ для каждого ребра $e_{iv}=(x_i, w_v) \in Q_v(t)$, определяется параметр f_{iv} – суммарный уровень феромона на этом ребре e_{iv}.

Для каждого множества $Q_v(t)$ определяется среднее количество феромона приходящееся на одно ребро $e_{iv} \in Q_v(t)$:

$$f(Q_v(t)) = \sum_v(f_{iv}) / |Q_v(t)|.$$

Величина $\sigma_{vk}(t)=f(Q_v(t))$ объявляется стоимостью вершины $w_v \in W_{1k}(t)$.

После этого агент z_k с вероятностью $P_{vk}(t)= \sigma_{vk}(t)/\sum_v(\sigma_{vk}(t))$, пропорциональной $\sigma_{vk}(t)$, выбирает вершину $w^*_v \in W_{1k}(t)$ и соответствующее ей множество ребер $Q^*_v(t)$.

На второй стадии шага t среди ребер множества $Q^*_v(t)$, инцидентных выбранной вершине w^*_v, с вероятностью $P_{vk}(t)=f_{iv}(t)/\sum_i(f_{ik}(t))$, выбирается ребро $e^*_{iv}=(x^*_i, w^*_v)$, которое включается в формируемое агентом z_k множество ребер $E^k(t)$.

Далее выполняются постпроцедуры.

Локальная степень $\rho_v(t)$ вершины w^*_v уменьшается на единицу: $\rho_v(t)=\rho_v(t)-1$

– Если локальная степень $\rho_v(t)$ у вершины w^*_v на шаге t становится равной n_v, то подмножество $Q_v(t)$и вершина w^*_v из рассмотрения на дальнейших шагах исключается: вершина w^*_v исключается из $\hat{W}(t)$.

– Определяется вершина x^*_i, инцидентная выбранному ребру $e^*_{iv}=(x^*_i, w^*_v)$.

– Определяется подмножество $U^*_i(t)$ ребер, инцидентных выбранной вершине x^*_i.

– Из всех подмножеств $Q_v(t)$ удаляются ребра подмножества $U^*_i(t)$. Процесс формирования агентом z_k множества E^k завершается, если локальная степень $\rho_v(t)$ у всех вершин w_v на шаге t стала равной n_v и все $Q_v(t)$ исключены из рассмотрения.

Процесс формирования агентом z_k множества E^k завершается, если у всех $Q_v(t)$ локальная степень $\rho(w_v)(t)$ вершины w_v на шаге t стала равной n_v и все $Q_v(t)$ исключены из рассмотрения.

После формирования агентами двудольных подграфов D_k (каждый агент – свой двудольный подграф D_k), на втором этапе итерации, каждый агент откладывает феромон на тех рёбрах полного двудольного графа D, которые соответствуют ребрам построенного двудольного подграфа D_k. После того, как каждый агент сформировал решение и отложил феромон, на третьем этапе происходит общее испарение феромона на ребрах полного двудольного графа R в соответствии с формулой:

$$h_{iv}(t) = h_{iv}(t) \cdot (1-\rho),$$

где $h_{ij}(t)$ – уровень феромона, отложенного агентами на ребре $e_{iv} \in E^k$, ρ – коэффициент обновления.

IV. Ко-эволюционный алгоритм разбиения

1. В соответствии с исходными данными формируется полный двудольный граф поиска решений $D=(X \cup W,Y)$.

2. Задается число агентов N_a в каждой из субпопуляций Z_1 и Z_2.

3. Задаются значения управляющих параметров.

4. Задается число итераций – N_l.

5. На всех ребрах графа $D=(X \cup W,Y)$ откладывается начальное количество феромона. Задаются значения коэффициентов.

6. $l=1$. *(l – номер итерации)*

(1 конструктивный алгоритм)

7. $k=1$. *(k – номер агента)*

8. Формируются значения параметров памяти для агента z_k популяции Z_1. $t=1$. $X_{1k}(1)=\varnothing$. $X_{2k}(1)=X$. $E^k(1)=\varnothing$. $\forall v(\rho_v(1)=0$. $\hat{W}(1)= W$.

9. $t=t+1$. *(t – номер шага)* На первой стадии шага t формируется множество $\hat{W}(t) \subset W$ вершин w_v таких, что для каждой вершины $w_v \in \hat{W}(t)$ ее текущая локальная степень $\rho_v(t)<n_v$.

10. (*Стадия 1*). Для каждой вершины $x_i \in X_{k2}(t)$ определяется набор $U_i(t)$ инцидентных ей ребер $e_{iv}=(x_i,w_v) \in U_i(t)$, связывающих x_i со всеми вершинами w_v множества $\hat{W}(t)$.

11. Для каждого ребра $e_{iv}=(x_i,w_v) \in U_i(t)$ определяется параметр $f_{iv}(t)$ – количество размещенного на e_{iv} феромона.

12. Определяется среднее количество феромона приходящееся на одно ребро множества $U_i(t)$:

$$f^*(U_i(t)) = \sum_v(f_{iv}(t))/s.$$

Величина $\mu_{ik}(t)=f^*(U_i(t))$ объявляется стоимостью вершины $x_i \in X_{2k}(t)$.

13. Среди множества вершин $X_{2k}(t)$ с вероятностью $P_{ik}(t)=\mu_{ik}(t)/\sum_i(\mu_{ik}(t))$, выбирается вершина $x^*_i(t) \in X_{2k}(t)$.

14. (*Стадия 2*). Среди ребер $U^*_i(t)$, инцидентных выбранной вершине $x^*_i(t)$, с вероятностью $P_{vk}(t)=f_{iv}(t)/\sum_v(f_{iv}(t))$, выбирается ребро $e^*_i=(x^*_i(t),w_v)$, которое включается в формируемое агентом z_k множество ребер $E^k(t)$.

15. Выполнение постпроцедур после шага t.

Вершина $x^*_i(t)$, включается в множество $X_{1k}(t)$ и исключается из множества $X_{2k}(t)$. $E^k(t)=E^k(t-1)\cup e^*_i$.

16. Если $X_{2k}(t)=\varnothing$, то переход к 17, иначе переход к 9.

17. Формирование на базе множества ребер $E^k(t)$ графа решения $D^k=(X\cup W,E^k)$ и расчет оценки $F_k(l)$ разбиения, задаваемого графом $D^k=(X\cup W,E^k)$.

18. На ребрах E графа D, соответствующих найденному множеству ребер E^k графа D^k, откладывается феромон в количестве пропорциональном оценке $F_k(l)$:

$$\Delta\tau_l(t)= Q / F_k(l).$$

19. Выполняется процедура испарения феромона на ребрах графа G.

20. Если $k<N_a$, то $k=k+1$ и переход к 8, иначе переход к 21.

(2 конструктивный алгоритм)

21. $k=1$. *(k – номер агента)*

22. Формируются значения параметров памяти для агента z_k популяции Z_2.

$\hat{W}(t)$ список вершин w_v, рассматриваемых на шаге.

$Q_v(t)$ – множество ребер $e_{iv}=(x_i, w_v)$, инцидентных вершине $w_v\in\hat{W}(t)$ на шаге t.

$\rho(w_v)(t)$ – локальная степень вершины $w_v\in\hat{W}(t)$ на шаге t. $\hat{W}(1)=W$.

Для всех $w_v\in\hat{W}(1)$ исходная локальная степень вершины $\rho(w_v)(1)=n$.

Для каждой вершины $w_v\in\hat{W}(1)$ формируется начальное множество $Q_v(1)$ инцидентных ей ребер полного двудольного графа D. $t=1$. $E^k(1)=\varnothing$. $\hat{W}(t)$.

23. $t=t+1$. *(t – номер шага).*

24. *Стадия 1.* В каждом из сформированных на предыдущих шагах множеств $Q_v(t)$ для каждого ребра $e_{iv}=(x_i,w_v)\in Q_v(t)$, определяется суммарный уровень f_{iv} отложенного на нем феромона.

25. Для каждого $Q_v(t)$ определяется среднее количество феромона приходящееся на одно ребро $e_{iv}\in Q_v(t)$:

$$f(Q_v(t))=\sum_v(f_{iv}) / |Q_v(t)|.$$

Величина $\sigma_{vk}(t)=f(Q_v(t))$ объявляется стоимостью вершины $w_v\in\hat{W}(t)$.

26. Агент z_k с вероятностью $P_{vk}(t)=\sigma_{vk}(t)/\sum_v(\sigma_{vk}(t))$, пропорциональной $\sigma_{vk}(t)$, выбирает вершину $w^*_v\in\hat{W}(t)$ и соответствующее ей множество ребер $Q^*_v(t)$.

27. *Стадия 2.* Среди ребер $Q^*_v(t)$, инцидентных выбранной вершине w^*_v, с вероятностью $P_{vk}(t)=f_{iv}(t)/\sum_i(\mu_{ik}(t))$, выбирается ребро $e^*_{iv}=(x^*_i,w^*_v)$, которое включается в формируемое агентом z_k множество ребер $E^k(t)$.

28. Выполнение постпроцедур после шага t.

– Определяется вершина x^*_i, инцидентная выбранному ребру $e^*_{iv}=(x^*_i, w^*_v)$.

– Определяется подмножество $U^*_i(t)$ ребер, инцидентных выбранной вершине x^*_i.

– Из всех подмножеств $Q_v(t)$ удаляются ребра подмножества $U^*_i(t)$.

– Для всех $w_v\in\hat{W}(t)$, инцидентных ребрам подмножества $U^*_i(t)$, пересчитывается $\rho_v(t)=\rho_v(t)+1$.

– Если после удаления $U^*_i(t)$, для некоторых $Q_v(t)$ локальная степень $\rho_v(t)$ вершины w_v на шаге t становится равной n_v, подмножество $Q_v(t)$ из рассмотрения на дальнейших шагах исключается.

29. Если все $Q_v(t)$ исключены из рассмотрения, то переход к 30, иначе переход к 23.

30. Формирование на базе множества ребер $E^k(t)$ графа решения $D^k=(X^k\cup W^k,E^k)$ и расчет оценки $F_k(l)$ разбиения, задаваемого графом $D^k=(X^k\cup W^k,E^k)$.

31. На ребрах E графа D, соответствующих найденному множеству ребер E^k графа D^k, откладывается феромон в количестве пропорциональном оценке $F_k(l)$.

$\Delta\tau_l(t)= Q / F_k(l)$.

32. Выполняется процедура испарения феромона на ребрах графа D.

33. Если $k<N_a$, то $k=k+1$ и переход к 22, иначе переход к 34.

34. Если $l<N_l$, то $l=l+1$ и переход к 7, иначе переход к 35.

35. Конец работы алгоритма.

Временная сложность этого алгоритма на одной итерации определяется как $O(n^2)$. В общем случае время работы этого алгоритма зависит от времени жизни колонии l (число итераций), количества вершин графа n и числа агентов m.

V. Экспериментальные исследования

При создании программы использовался пакет Visual Studio 2005 для 32-разрядных ОС семейства Windows®9x, а отладка, тестирование и эксперименты поводились на IBM® совместимом компьютере с процессором AMD Athlon-64® 3000+ и оперативной памятью 512 MB DIMM PC-3200.

На основе разработанных алгоритмов разбиения создана программа **coevolution partitioning** (CP).

Для проведения экспериментов была использована процедура синтеза контрольных примеров с известным оптимумом $F_{опт}$ по аналогии с известным методом BEKU (Partitioning Examples with Tight Upper Bound of Optimal Solution) [11].

Исследованию подвергались примеры, содержащие до 1000 вершин. Вес вершин принимался равным нулю, а вес всех рёбер – единице. При этом графы «разбивались» на два подграфа с равным количеством вершин в каждом подграфе.

На основе обработки экспериментальных исследований была построена средняя зависимость качества решений от числа итераций (рис. 2), и размера популяции (рис. 3). Оценкой качества служит величина Fопт/F, где F – оценка полученного решения. Установлено, что начальное количество феромона Q должно быть в 14 раз больше среднего количества значения феромона $\tau_k(l)$, откладываемого агентами на каждой итерации. Коэффициент обновления ρ=0,95.

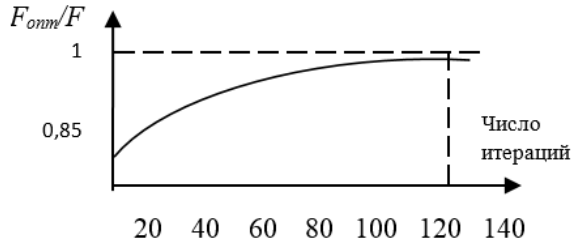

Рис. 2. Зависимость качества решений алгоритма CP от числа итераций

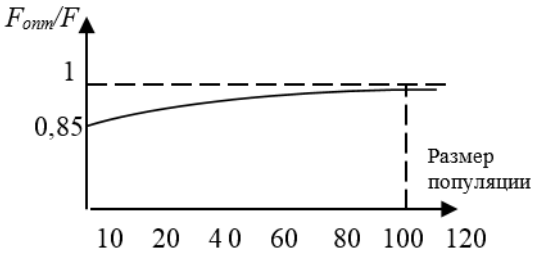

Рис. 3. Зависимость качества решений алгоритма CP от размера популяции

В результате экспериментов установлено, что при объеме популяции M=100 алгоритм сходится в среднем на 120 итерации. При этом у ряда экспериментов число итераций меньше среднего составляло до 10%, а больше с среднего до 35%. Временная сложность процедуры на одной итерации – O(n).

Сравнительный анализ с другими алгоритмами разбиения производился на стандартных тестовых примерах и схемах (бенчмарках), размещенных на: http://www.cad.polito.it/tools/9.html, http://www.cbl.ncsu.edu.

Данные примеры представляют собой стандартные промышленные цепи (блоки, схемы, ИС, БИС, СБИС). Сравнение производилось по следующим параметрам:

– Cut – число цепей, попавших в разрез;

– CPU – процессорное время, затраченное на поиск решения.

Выполнена сравнительная оценка работы разработанных алгоритмов на бенчмарках cordic(881), misex3(1349), X3(1369). c6288(2435) s15850(4321), frisc(4400), elliptic(4711), (в скобках приведено число вентилей) с результатами работы следующих алгоритмов (на этих же схемах): Pure hMetis [12]; Net&Path Based [13]; PPFF [13]; VPR [12].

Сравнительные оценки приведены в табл. 1.

Анализируя результаты экспериментов, можно сделать вывод, что разработанные алгоритмы находят решения, не уступающие по качеству, а иногда и превосходящие своих аналогов в среднем на 3-4%. Ощутимо улучшение результатов **CP** по сравнению к hMetis, который является очень сильным алгоритмом. Например, на схеме Cordic результаты hMetis на 4,8% хуже, чем результаты CP.

Таблица 1

Сравнительная оценка работы алгоритмов

Алгоритм	F	Бенчмарки						
		Cordic	misex3	X3	c6288	s15850	Frisc	Elliptic
Pure hMetis	Cut	327	543	211	182	617	838	508
	CPU	2	5	4	31	28	34	16
Net& Path	Cut	278	614	261	216	671	869	619
	CPU	5	13	6	80	37	61	22
PPFF	Cut	313	540	228	201	615	851	510
	CPU	6	14	8	79	43	66	25
VPR	Cut	319	543	216	204	622	843	513
	CPU	10	18	13	85	47	71	27
CP	Cut	280	610	253	209	651	842	567
	CPU	4	12	5	71	301	46	15

В среднем запуск программы обеспечивает нахождения решения, отличающегося от оптимального менее, чем на 0,5%.

VI. Заключение

Для повышения эффективности, усиления сходимости алгоритма и способности выхода из локальных оптимумов предложен коалиционный подход к построению алгоритма разбиения.

Для решения задачи разбиения авторами разработана модифицированная метаэвристика по аналогии с моделями адаптивного поведения МА.

В работе, в отличие от канонической парадигмы МА интерпретацией решения является подграф Dk=(Xk∪Wk,Ek), задающий распределение множества вершин Xk по узлам множества Wk. Работа алгоритма фактически заключается в формировании подмножества ребер Ek.

Одновременно в пространстве поиска задачи оптимизации эволюционируют две субпопуляции, каждая из которых решает одну и ту же исходную оптимизационную задачу, но имеющие различные области поиска и стратегии оптимизации. Ко-эволюционный подход обеспечивает более широкий

обзор пространства решений и более высокую вероятность локализации глобального экстремума задачи.

Отличительными особенностями взаимодействия между ко-эволюционирующими субпопуляциями является то, что они базируются на использовании общего графа поиска решений, общей эволюционной памяти, формировании единой интерпретации решения в виде двудольного графа.

Поддержка

Работа выполнена при финансовой поддержке гранта РФФИ № 20-07-00260 а.

Литература

[1] Немудров В., Мартин Г. Системы-на-кристалле. Проектирование и развитие. М: Техносфера, 2004. 157 с.

[2] Казеннов Г.Г. Основы проектирования интегральных схем и систем. М.: Бином. Лаборатория знаний, 2005. 295 с.

[3] Alpert C.J., Mehta D.P., Sapatnekar S.S., Handbook of Algorithms for Physical Design Automation. Boston, MA: Auerbach, 2009. 245 p.

[4] Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учебное пособие. М: Издательство МГТУ им. Н.Э. Баумана, 2014. 448 с.

[5] Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Гибридный алгоритм разбиения на основе природных механизмов принятия решений // Искусственный интеллект и принятие решений. М.: Изд-во Институт системного анализа РАН, 2012. С. 3-15.

[6] Лебедев Б.К., Лебедев О.Б. Муравьиные алгоритмы разбиения, использующие представление задачи, отличные от канонического // Вестник Ростовского государственного университета путей сообщения. – Ростов-на-Дону: Изд-во РГУПС, 2016. №2 (62). С. 71-77.

[7] Лебедев О.Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог: Изд-во ЮФУ, 2013. 199 с.

[8] Лебедев Б.К., Лебедев О.Б. Меметический алгоритм разбиения // Вестник Ростовского государственного университета путей сообщения – Ростов-на-Дону: Изд-во РГУПС, 2017. №2 (62). С. 136-145.

[9] Zhou Y., Pei Sh. A hybrid co-evolutionary particle swarm optimization algorithm for solving constrained engineering design problems // Journal of computers (JCP). Academy Publisher, 2010. v.5, №6. pp. 965-972.

[10] Ворбьева Е.Ю., Карпенко А.П., Селиверстов Е.Ю. Ко-гибридизация алгоритмов роя частиц // Наука и жизнь. 2012. №4.

[11] Cong J., Romesis M., Xie M. Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms // Proc. of the International Symposium on Physical Design. Monterey, CA, 2003. pp. 88-94.

[12] Selvakkumaran N., Karypis G. Multi-Objective Hypergraph Partitioning Algorithms for Cut and Maximum Subdomain Degree Minimization // ICCAD, 2003. pp. 234-249.

[13] Ababei C., Selvakkumaran N., Bazargan K., Karypis G. Multi-objective Circuit Partitioning for Cutsize and Path-Based Delay Minimization // ICCAD, 2002. pp. 118-137.

Co-evolution VLSI Partitioning Algorithm

B.K. Lebedev, O.B. Lebedev

Southern Federal University, Taganrog Southern Federal University, lebedev.ob@mail.ru

***Abstract* — To increase the efficiency, enhance the convergence of the algorithm and the ability to exit local optima, the author proposes a co-evolution approach to the construction of the partition algorithm.**

At the same time, two subpopulations evolve in the search space of the optimization problem, each of which solves the same initial optimization problem, but having different search areas and optimization strategies. The co-evolutionary approach provides a wider overview of the solution space and a higher probability of localizing the global extremum of the problem.

To solve the partition problem, the authors developed a modified metaheuristics by analogy with the models of adaptive behavior of ant algorithm (AA).

In the work, in contrast to the canonical paradigm of AA. The interpretation of the solution is the subgraph $D_k=(X_k \cup W_k, E_k)$, which defines the distribution of the set of vertices X_k over the nodes of the set W_k. The operation of the algorithm actually consists in the formation of a subset of edges E_k.

The partition problem is solved by two subpopulations of A_1, A_2 agents on the $D=(X \cup W,Y)$, $D_k \in D$. Agents of subpopulation A_1 construct a solution using the first constructive algorithm on the set of vertices X_k.

Agents of subpopulation A_1 use the second constructive algorithm on the set of vertices X_k. Strategies C_1 and C_2 are distinguished by constructive algorithms for splitting the circuit with agents. Both subpopulations form the same subset of E_k edges.

For the experiments we used the synthesis procedure for control examples with the well-known F_{opt} Optimum by analogy with the well-known BEKU method (Partitioning Examples with Tight Upper Bound of Optimal Solution).

The study was subjected to examples containing up to 1000 vertices. The weight of the vertices was taken equal to zero,

and the weight of all the edges was taken to be unity. In this case, the graphs were "divided" into two subgraphs with an equal number of vertices in each subgraph.

Based on the processing of experimental studies, an average dependence of the quality of solutions on the number of iterations and population size was constructed. The quality estimate is F_{opt}/F, where F is the estimate of the resulting solution. It was found that the initial amount of pheromone Q should be 14 times greater than the average amount of pheromone $\tau_k(l)$ deposited by agents at each iteration. Update rate ρ=0.95. As a result of experiments, it was found that, with a population volume of M=100, the algorithm converges on average at 120 iterations.

Comparative analysis with other partitioning algorithms was performed on standard test examples and diagrams (benchmarks), available on the websites: www.cad.polito.it/tools/9.html, www.cbl.ncsu.edu.

The developed algorithms find solutions that are not inferior in quality, and sometimes even superior to their analogues by an average of 3-4%. On average, launching a program provides solutions that differ from the optimal solution by less than 0.5%.

Keywords **- VLSI, partitioning, swarm intelligence, ant algorithm, adaptive behavior, subpopulation, co-evolution, optimization.**

REFERENCES

[1] Nemudrov V., Martin G. Sistemy-na-kristalle. Proyektirovaniye i razvitiye (Systems-on-a-chip. Design and development). M: Tekhnosfera, 2004.157 c.

[2] Kazennov G.G. Osnovy proyektirovaniya integral'nykh skhem i sistem (Fundamentals of design of integrated circuits and systems). M.: Binom, Laboratoriya znaniy, 2005, 295 c.

[3] Alpert C.J., Mehta D.P., Sapatnekar S.S., Handbook of Algorithms for Physical Design Automation. Boston, MA: Auerbach, 2009. 245 p.

[4] Karpenko A.P. Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: uchebnoye posobiye (Modern search engine optimization algorithms. Algorithms inspired by nature: a tutorial). M: Izdatel'stvo MSTU N.E. Bauman, 2014. 448 p.

[5] Kureichik V.M., Lebedev B.K., Lebedev O.B. Gibridnyy algoritm razbiyeniya na osnove prirodnykh mekhanizmov prinyatiya resheniy (Hybrid partition algorithm based on natural decision-making mechanisms) // Iskusstvennyy intellekt i prinyatiye resheniy. M.: Izd-vo Institut sistemnogo analiza RAN, 2012. pp. 3-15.

[6] Lebedev B.K., Lebedev O.B. Murav'inyye algoritmy razbiyeniya, ispol'zuyushchiye predstavleniye zadachi, otlichnyye ot kanonicheskogo (Ant partitioning algorithms using a task representation other than the canonical) // Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya. Rostov-na-Donu: Izd-vo RGUPS, 2016. №2(62). pp. 71-77.

[7] Lebedev O.B. Modeli adaptivnogo povedeniya murav'inoy kolonii v zadachakh proyektirovaniya (Models of adaptive behavior of an ant colony in design problems). Taganrog: Izd-vo YUFU, 2013. 199 p.

[8] Lebedev B.K., Lebedev O.B. Memeticheskiy algoritm razbiyeniya (The memetic algorithm for splitting) // Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya. Rostov-na-Donu: Izd-vo RGUPS, 2017. №3. pp. 136-145.

[9] Zhou Y., Pei Sh. A hybrid co-evolutionary particle swarm optimization algorithm for solving constrained engineering design problems // Journal of computers (JCP). Academy Publisher, 2010. v.5, №6. pp. 965-972.

[10] Vorobyeva E.Yu., Karpenko A.P., Seliverstov E.Yu. Kogibridizatsiya algoritmov roya chastits (Co-hybridization of particle swarm algorithms) // Nauka i zhizn. 2012. №4.

[11] Cong J., Romesis M., Xie M. Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms // Proc. of the International Symposium on Physical Design. Monterey, CA, 2003. pp. 88-94.

[12] Selvakkumaran N., Karypis G. Multi-Objective Hypergraph Partitioning Algorithms for Cut and Maximum Subdomain Degree Minimization // ICCAD, 2003. pp. 234-249.

[13] Ababei C., Selvakkumaran N., Bazargan K., Karypis G. Multi-objective Circuit Partitioning for Cutsize and Path-Based Delay Minimization // ICCAD, 2002. pp. 118-137.