

УДК 681.3.001.63

DOI: 10.15827/2311-6749.17.4.7

ПОСТРОЕНИЕ ГИБРИДНОЙ БИОИНСПИРИРОВАННОЙ МНОГОАГЕНТНОЙ СИСТЕМЫ НА ПРИМЕРЕ ЗАДАЧИ РАЗБИЕНИЯ

Б.К. Лебедев, д.т.н., профессор, lebedev.b.k@gmail.com; О.Б. Лебедев, к.т.н., доцент, lebedev.ob@mail.ru; В.А. Ванидовский, аспирант, vanidovskiy.v.a@gmail.ru
(Институт компьютерных технологий и информационной безопасности
Южного федерального университета, ул. Энгельса, 1, г. Таганрог, 347928, Россия)

В работе рассматривается подход к построению гибридной биоинспирированной многоагентной системы для решения задачи разбиения методом поисковой адаптации. В основе работы адаптивной многоагентной системы лежит ситуационная кооперация между агентами, когда решение о совместных действиях принимаются агентами на каждом такте работы на основе анализа состояния среды функционирования и наличия у агентов условий для кооперации. Двухуровневое представление объекта оптимизации и идеи коллективной адаптации, позволяющие сочетать эволюционную и альтернативную адаптации, использованы с учетом специфики задачи разбиения. Описаны механизмы альтернативной коллективной адаптации, моделируемой вероятностными обучающимися автоматами адаптации, структуры вероятностных обучающихся автоматов адаптации, механизмы переходов, выработки откликов среды, реализации альтернатив.

Ключевые слова: многоагентная система, гибридизация, биоинспирированный алгоритм, кооперация агентов, коллективная адаптация, автомат адаптации.

Современная СБИС может содержать десятки миллионов транзисторов, поэтому в связи с ограниченными возможностями вычислительных средств (память, скорость) не может быть спроектирована топология всей схемы в целом. Нормальным является разбиение схемы группированием компонентов в блоки. В результате разбиения формируется множество блоков и соединений между ними. В очень больших схемах используется иерархическая структура разбиения [1, 2].

Задачи разбиения можно разделить на два класса [3]:

– задачи, в которых осуществляется разбиение схемы на блоки с учетом конструктивных факторов, таких как размер блоков, число блоков и соединений между блоками, число внешних выводов у блоков и т.д.;

– задачи, в которых, помимо конструктивных факторов, существенны и функциональные характеристики блоков.

В данной работе рассматривается метод решения наиболее распространенных задач первого класса.

Существующие алгоритмы разбиения делятся на два класса: конструктивные и итеративные [4, 5].

Входными данными для конструктивных алгоритмов являются компоненты и связи между ними, выходными – разбиение схемы на блоки, полученное, как правило, путем последовательного распределения компонентов по блокам. Алгоритмы этого класса отличаются, с одной стороны, относительным быстродействием, а с другой – низким качеством решений.

В качестве исходных данных итеративные алгоритмы дополнительно используют некоторое начальное разбиение схемы. На каждой итерации осуществляется переход к новому разбиению (решению). Изменение разбиения осуществляется с помощью групповых или парных перестановок между блоками. Суть итерационного алгоритма – поиск в пространстве решений решения (разбиения) с лучшими показателями качества. Отметим, что в качестве начального разбиения итеративный алгоритм может использовать либо случайное разбиение, либо разбиение, полученное в результате работы конструктивного алгоритма.

В свою очередь, итеративные алгоритмы делятся на детерминированные и вероятностные [3, 4].

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.

Дальнейшим совершенствованием алгоритмов случайного поиска является разработка методов, основанных на моделировании естественных процессов. К ним относятся методы моделирования отжига, метод эволюционного моделирования, генетической адаптации, биоинспирированные алгоритмы [6–11].

Все эти методы относятся к классу методов случайного направленного поиска и имеют существенные отличия. К недостаткам относится то, что они все же в большей степени реализуют стратегии «слепого» поиска, что увеличивает объем просматриваемого пространства решений.

Особый интерес представляет поисковая адаптация, основанная на использовании обучающихся автоматов, моделирующих поведение объекта адаптации в среде. Трудности использования такого подхода связаны прежде всего с проблемой представления исходной формулировки задачи в виде адаптивной системы. Достоинствами являются повышенная целенаправленность и сходимости алгоритма [12].

В работе используется представление задачи разбиения в виде адаптивной системы, основными компонентами которой являются среда (объект оптимизации) и вероятностный автомат, помещенный в эту среду и реализующий алгоритм поиска [12, 13].

Постановка задачи разбиения

Большинство разработанных к настоящему времени алгоритмов разбиения используют в качестве модели схемы граф или гиперграф [1, 2]. Существенным недостатком при использовании графа в качестве модели схемы является наличие неоднозначности, так как одной цепи соответствует набор деревьев, построенных на связываемых этой цепью вершинах.

Задача разбиения гиперграфа с взвешенными вершинами и ребрами формулируется следующим образом.

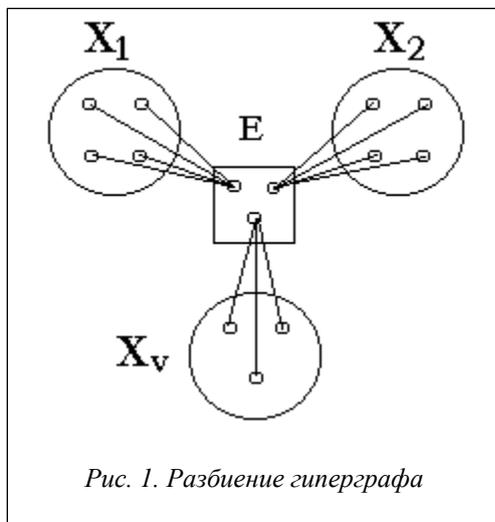


Рис. 1. Разбиение гиперграфа

Дан гиперграф $H = (X, E)$, где $X = \{x_i / i=1, 2, \dots, n\}$ – множество вершин, а $E = \{e_j / e_j \subset X, j=1, 2, \dots, m\}$ – множество ребер (каждое ребро – подмножество связываемых им вершин). Вес вершин задается множеством $\Phi = \{\phi_i / i=1, 2, \dots, n\}$, а вес ребер – множеством $\Psi = \{\psi_j / j=1, 2, \dots, m\}$. Необходимо сформировать K узлов, то есть множество X разбить на K непустых и непересекающихся подмножеств $X_v, X = \cup X_v, (\forall i, j) [X_i \cap X_j = \emptyset], X_v \neq \emptyset$ (рис. 1). На формируемые узлы накладываются ограничения. С помощью вектора $P = \{p_v / v=1, 2, \dots, k\}$ задается максимально допустимый суммарный вес вершин, назначенных в v -й узел, а с помощью вектора $N = \{n_v / v=1, 2, \dots, k\}$ – максимально допустимое число вершин, назначенных в v -й узел.

Ограничения на вместимость имеют вид:

$$\sum_{i \in I} \phi_i \leq p_v, I = \{i / x_i \in X_v\}, v = 1, 2, \dots, k, \tag{1}$$

$$|X_v| \leq n_v, v=1, 2, \dots, k. \tag{2}$$

Выражение (1) является ограничением на максимальный

вес узла, а выражение (2) – на максимальное число вершин в узле.

Иногда задано допустимое число выводов γ_{\max} для узлов. Ограничение для узлов на число выводов γ_v имеет вид:

$$\gamma_v \leq \gamma_{\max}, v=1, 2, \dots, k, \tag{3}$$

$$\gamma_v = |E_v|, E_v = \{e_j / (e_j \cap X_v \neq \emptyset) \& (e_j \cap X_v \neq e_j)\},$$

E_v – множество ребер, связывающих множество вершин X_v с вершинами остальных узлов.

Основным критерием является F_1 – суммарная стоимость ребер в разрезе.

$$F_1 = \sum_{j \in J} \psi_j, J = \{j / e_j \in C\}, \tag{4}$$

$C = \{e_j / (\forall v) [e_j \cap X_v \neq e_j]\}$ – множество ребер в разрезе.

Вторым часто используемым критерием является F_2 – суммарное число выводов.

$$F_2 = \sum_{v=1}^k \gamma_v. \tag{5}$$

Возможно использование критерия F , являющегося аддитивной сверткой критериев F_1 и F_2 :

$$F = k_1 \cdot F_1 + k_2 \cdot F_2.$$

Задачу разбиения гиперграфа $H = (X, E)$ со взвешенными вершинами и ребрами сведем к задаче о назначении множества гиперребер E в K узлов при выполнении следующих условий: каждое гиперребро может быть назначено только в один узел, а стоимость вершин, назначенных в узел z_v , не должна превышать наперед заданную величину p_v . Будем считать, что гиперребро e_j назначено в узел z_v , если все множество составляющих его вершин назначено в этот узел.

Введем булевы переменные:

$$y_{iv} = \begin{cases} 1, & \text{если вершина } x_i \text{ назначена в узел } z_v, \\ 0 & \text{в противном случае.} \end{cases}$$

$$h_{jv} = \begin{cases} 1, & \text{если гиперребро } e_j \text{ назначено в узел } z_v, \\ 0 & \text{в противном случае.} \end{cases}$$

Согласно описанным условиям, должны выполняться следующие ограничения:

$$\sum_{i=1}^n y_{iv} \phi_i \leq p_v; \quad v = 1, 2, \dots, k;$$

$$\sum_{v=1}^k y_{iv} = 1; \quad i = 1, 2, \dots, n.$$

Цель оптимизации – минимизация стоимости связей между узлами, или, что одно и то же, максимизация стоимости гиперребер, полностью размещенных в узлах. Целевая функция примет вид:

максимизировать

$$F_2 = \sum_{v=1}^k \sum_{j=1}^m h_{jv} \Psi_j.$$

Установим связь между переменными h_{jv} и y_{iv} с помощью следующей группы ограничений:

$$\sum_{i \in I_j} y_{iv} \geq h_{jv} \Theta(e_j); \quad j=1, 2, \dots, m; \quad v=1, 2, \dots, k;$$

где $\Theta(e_j)$ – мощность множества e_j , то есть $\Theta(e_j) = |e_j|$, а $I_j = \{i \mid x_i \in e_j\}$.

Действительно, при некотором опорном плане, то есть пробном назначении гиперребер в узлы, в соответствии с этой группой ограничений все составляющие эти ребра вершины будут также назначены в те же узлы. Таким образом, задача разбиения гиперграфа со взвешенными вершинами и ребрами по критерию минимальной стоимости связей между узлами формулируется следующим образом:

$$\text{максимизировать } F_2 = \sum_{v=1}^k \sum_{j=1}^m h_{jv} \Psi_j,$$

при ограничениях

$$\sum_{i=1}^n y_{iv} \phi_i \leq p_v; \quad v=1, 2, \dots, k;$$

$$\sum_{v=1}^k y_{iv} = 1, \quad i=1, 2, \dots, n;$$

$$\sum_{i \in I_j} y_{iv} \geq h_{jv} \Theta(e_j), \quad j=1, 2, \dots, m; \quad v=1, 2, \dots, k.$$

Число ограничений $D = (m + 1) \cdot k + n$.

Для построения математической модели задачи разбиения гиперграфа с взвешенными вершинами и ребрами по критерию минимума числа выводов формируемых узлов введем булеву переменную q_{jv} :

$$q_{jv} = \begin{cases} 1, & \text{если } e_j \cap x_v \neq \emptyset; \\ 0, & \text{если } e_j \cap x_v = \emptyset. \end{cases}$$

Целью задачи является отыскание минимума функции

$$F_3 = \sum_{j=1}^m \sum_{v=1}^k q_{jv}.$$

Установим связь между переменными q_{jv} и y_{iv} с помощью следующей группы ограничений:

$$\sum_{i \in I_j} y_{iv} \leq q_{jv} \Theta(e_j); \quad j=1, 2, \dots, m; \quad v=1, 2, \dots, k;$$

где $\Theta(e_j)$ – мощность множества e_j , то есть $\Theta(e_j) = |e_j|$, а $I_j = \{i \mid x_i \in e_j\}$.

Действительно, при некотором опорном плане в соответствии с этой группой ограничений $q_{jv} = 0$ только в том случае, когда ни одна из вершин, входящих в состав гиперребра e_j , не назначена в узел z_v .

Задача разбиения гиперграфа с взвешенными вершинами и ребрами по критерию минимума числа выводов формируемых узлов формулируется следующим образом:

$$\text{Минимизировать } F_3 = \sum_{v=1}^k \sum_{j=1}^m q_{jv},$$

при ограничениях

$$\sum_{i=1}^n y_{iv} \phi_i \leq p_v; \quad v=1, 2, \dots, k;$$

$$\sum_{v=1}^k y_{iv} = 1, i=1, 2, \dots, n;$$

$$\sum_{i \in I_j} y_{iv} \leq q_{jv} \Theta(e_j), j=1, 2, \dots, m; v=1, 2, \dots, k.$$

Число ограничений $D = (m + 1) \cdot k + n$.

Механизмы кооперативного поведения агентов для решения задачи разбиения

Исходная формулировка задачи представляется в виде адаптивной многоагентной системы, работа которой основана на идеях коллективного поведения объектов адаптации.

В качестве агента (элементарного объекта адаптации) рассматривается вершина $x_i \in X$. Коллектив агентов (их совокупность) соответствует объекту оптимизации (ОО). Кооперативное поведение агентов для решения общей задачи предполагает согласование действий в процессе решения задачи.

Работу адаптивной многоагентной системы можно представить как функционирование некоторого роя вероятностных автоматов адаптации (АА), действующего в случайной среде функционирования [13, 14]. Под средой функционирования понимается объект управления (ОО). Функции управляющего устройства агента x_i реализуются АА a_i , состояние которого A_i изменяется в соответствии с откликом среды функционирования Q .

В рассматриваемом случае в качестве ОО (среды функционирования) рассматривается гиперграф, разбитый на K узлов. Состояние среды функционирования оценивается вектором $S = \{s_i / i = 1, 2, \dots, n\}$, где s_i – номер узла, в который помещена вершина x_i . В основе работы адаптивной многоагентной системы лежит ситуационная кооперация между агентами, когда решение о совместных действиях принимается агентами на каждом такте работы на основе анализа состояния S среды функционирования и наличия у агентов условий для кооперации. Компонента управления (АА a_i) агентом x_i определяет элементарные действия агента для различных состояний среды функционирования.

На каждом такте работы адаптивной многоагентной системы в соответствии с состояниями A роя автоматов адаптации (РАА) формируется управляющее воздействие U , приводящее к изменению состояния S среды функционирования и показателя $F(S)$ (рис. 2).

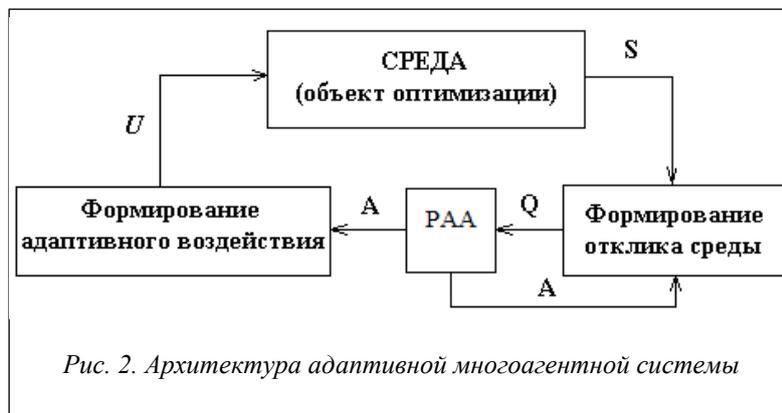


Рис. 2. Архитектура адаптивной многоагентной системы

Q является откликом среды на реализацию управляющего воздействия U . Под действием Q автоматы переходят в новое состояние и вырабатывают новые значения A .

Пусть $P = \{S_i / i = 1, 2, \dots\}$ – пространство возможных состояний среды функционирования (возможных решений задачи разбиения).

Предлагаемый алгоритм разбиения является алгоритмом случайного поиска в пространстве состояний – P . Алгоритм случайного поиска представляется в виде коллектива вероятностных автоматов, адаптация которых производится путем введения самообучения в процессе их функционирования [14].

Пусть объект оптимизации (среда функционирования) находится в некотором состоянии S_j , то есть заданы разбиение X и, соответственно, распределение вершин x_i по узлам.

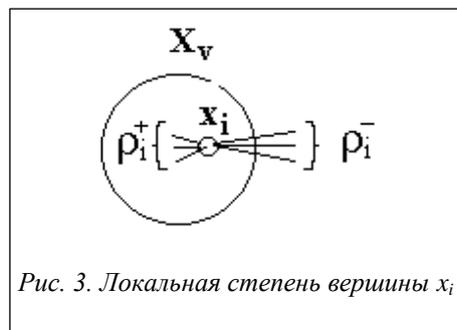


Рис. 3. Локальная степень вершины x_i

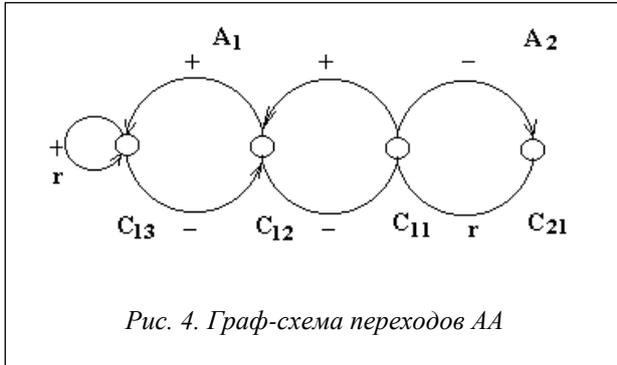
Пусть ρ_i^+ – число гиперребер, связывающих $x_i \in X_v$ с вершинами $x_j \in X_v, x_i \neq x_j$, а ρ_i^- – число гиперребер, связывающих $x_i \in X_v$ с вершинами $x_j \notin X_v$ (рис. 3). $\rho_i \leq \rho_i^+ + \rho_i^-$, ρ_i – локальная степень вершины x_i .

Локальная цель агента (объекта адаптации x_i) – достижение такого состояния (то есть такого распределения x_i), при котором его оценка $\rho_i^- = 0$. Другими словами, в процессе адаптации минимизируется ρ_i^- .

Локальная цель агента (объекта адаптации x_i) – достижение такого состояния (то есть такого распределения x_i), при котором его оценка $\rho_i^- = 0$. Другими словами, в процессе адаптации минимизируется ρ_i^- .

Глобальная цель многоагентной системы (коллектива объектов адаптации) заключается в достижении такого состояния S (то есть такого распределения вершин по узлам), при котором $F(S) \rightarrow \min$.

Для реализации механизма адаптации каждому агенту $x_i \in X$ сопоставляется АА a_i , моделирующий поведение объекта адаптации в среде функционирования [14]. АА имеет две группы состояний: $C_1 = \{c_{1i} / i = 1, 2, \dots, g\}$ и $C_2 = \{c_{21}\}$, соответствующие альтернативам A_1 и A_2 поведения агента в среде функционирования. A_1 – остаться в том же узле, A_2 – выход из узла и перераспределение. Таким образом, выходной алфавит автомата – $A = \{A_1, A_2\}$. Входной алфавит – $Q = \{+, -\}$ включает возможные отклики среды функционирования – «поощрение» (+) и «наказание» (-).



Граф-схема переходов АА показана на рисунке 4.

Отклик среды для АА a_i в соответствии с состоянием среды и объекта адаптации формируется следующим образом.

Если $\rho_i > \rho_i^+$, то всегда вырабатывается сигнал «наказание» (-).

Если $\rho_i^- \leq \rho_i^+$, то с вероятностью $P_n = \frac{\rho_i^-}{\rho_i^- + \rho_i^+}$ вырабатывается сигнал «наказание» (-); а с вероятностью $P_n = 1 - P_n = \frac{\rho_i^+}{\rho_i^- + \rho_i^+}$ вырабатывается

сигнал «поощрение» (+).

На каждой итерации работы адаптивной системы процесс коллективной адаптации осуществляется за четыре такта (рис. 5).



На первом такте в соответствии с состоянием среды для каждого объекта адаптации рассчитываются параметры ρ_i^+ и ρ_i^- .

На втором такте для каждого автомата адаптации $a_i = \Gamma(x_i)$ вырабатывается отклик среды q_i : сигнал «поощрение» (+) или «наказание» (-).

На третьем такте в каждом автомате адаптации a_i под действием подаваемого на его вход отклика q_i осуществляется переход в новое состояние.

На четвертом такте реализуется альтернатива в соответствии с выходами АА. Объект оптимизации после реализации альтернатив переходит в новое состояние, для которого рассчитывается показатель $F(S)$. Состояние с лучшим значением $F(S)$ запоминается.

Рассмотрим процесс реализации для объектов адаптации альтернативы A_2 – выход из узла и перераспределение. Из всех подмножеств X_v удаляются вершины x_i , для которых реализуется альтернатива A_2 . Все такие вершины объединяются в множество $R = \{R_v / v = 1, 2, \dots, k\}$, где R_v – множество вершин, удаленных из X_v .

Процесс переназначения вершин осуществляется последовательно. На каждом шаге случайным образом (равновероятно) выбирается вершина $x_i \in R$. Для нее определяется множество доступных узлов. Узел X_v является доступным, если после назначения в него x_i допустимый вес p_v или допустимое число вершин n_v не будут перекрыты.

Обозначим через ρ_i^v число гиперребер, связывающих выбранную x_i с доступным узлом X_v .

$$\bar{\rho}_i = \sum_v \rho_i^v - \text{число связей } x_i \text{ с доступными узлами.}$$

Пусть w – число доступных узлов.

Назначение x_i в один из доступных узлов осуществляется вероятностным способом.

Вероятность назначения x_i в недоступный узел равна нулю.

Вероятность попадания x_i в один из доступных узлов X_v пропорциональна числу связей x_i с X_v и определяется как

$$p_i^v = \frac{\rho_i^v + \delta}{\rho_i + w \cdot \delta}; \sum_v p_i^v = 1; P_i = \{\rho_i^v | v = 1, 2, \dots, k\},$$

где P_i – распределение вероятности назначения для x_i ; P_i – рассчитывается только для выбранной для переназначения x_i . С помощью параметра δ осуществляется управление распределением вероятностей для x_i .

В работе используется подход, имеющий сходство с методом моделирования отжига.

Параметр δ вычисляется по формуле

$\delta = \alpha_n - t \cdot \Delta\alpha$, $t=1, 2, \dots, T_0$, t – номер итерации.

$$T_0 = \frac{\alpha_n - \alpha_0}{\Delta\alpha},$$

где α_n , α_0 , $\Delta\alpha$ – управляющие параметры, задаваемые заранее (выбираемые из эвристических соображений). δ имеет максимальное значение на первой итерации работы адаптивной системы. Последовательно уменьшаясь, δ приобретает минимальное значение на итерации T_0 и после этого не меняется. В процессе переназначения множества R на итерации t параметр δ не меняется. Чем больше δ , тем более близким (равновероятным) становятся значения $\rho_i^v \in P_i$. Другими словами, на первых итерациях переназначаемые элементы имеют большую степень свободы и меньшую степень зависимости от ρ_i^v и поэтому элемент x_i может попасть в узел, с которым он не связан. После выполнения T итераций, если при переназначении x_i имеет связи с доступными узлами, то вероятность попадания x_i в доступный узел, с которым нет связей, близка к нулю.

Отметим, что δ не может быть равной нулю (а следовательно, и α_0), ибо возможна ситуация в процессе переназначения R , когда будет выбрана такая вершина x_i , что для всех доступных узлов $\rho_i^v = 0$, $\bar{\rho}_i = 0$.

Такой подход к расчету распределения вероятности назначения вершины создает условия для выхода из локальных оптимумов.

На рисунке 6 представлен псевдокод адаптивного алгоритма разбиения.

Процедурой *ИСХОД_ДАННЫЕ* формируется массив *граф*, в котором хранятся исходные данные, описывающие задачу. Массив *управление*, сформированный процедурой *НАСТРОЙКА*, хранит значения управляющих параметров: число итераций, число набросов, α_n , α_0 , $\Delta\alpha$, g – глубина памяти АА и т.п.

Процедурой *ИСХОД_РАЗБИЕНИЕ(граф)* формируется случайным образом начальное состояние среды S , то есть начальное разбиение гиперграфа – *решение*. Это *решение* на данный момент считается лучшим и заносится в массив *лучш_решение*.

С помощью процедуры *НАЧ_СОСТОЯНИЕ* все АА переводятся в состояние S_n . Состояние АА заносится в массив *состоян_aa*. Затем выполняется заданное число итераций. С помощью процедуры *РАСЧЕТ(решение)* рассчитываются параметры ρ_i^+ и ρ_i^- объектов адаптации, которые заносятся в массив *параметры_oa*.

Algorithm Адаптивное разбиение

```

begin
  граф=ИСХОД_ДАННЫЕ;
  управление=НАСТРОЙКА;
  решение=ИСХОД_РАЗБИЕНИЕ(граф);
  лучш_решение=решение;
  состоян_aa=НАЧ_СОСТОЯНИЕ;
  итерация=ЧИСЛО_ИТЕРАЦИЙ;
  while (итерация > 0) do
    {параметры_oa=РАСЧЕТ(решение);
     отклик=РЕАКЦИЯ(параметры_oa);
     состоян_aa=ПЕРЕХОД(состоян_aa, отклик);
     вершина=ВЫХОД(граф, состоян_aa);
     решение=УДАЛИТЬ(решение, вершина);
     while (вершины ≠ 0) do
       {вершина=ВЫБОР_ВЕРШ(вершины);
        распределение=ВЕРОЯТНО(вершина, решение, управление);
        узел=ВЫБОР_УЗЛА(вершина, распределение);
        решение=ДОБАВИТЬ(решение, вершина, узел);
       }
     лучш_решение=СРАВНЕНИЕ(решение, лучш_решение);
     итерация= итерация-1;
    }
  end

```

Рис. 6. Псевдокод адаптивного алгоритма разбиения

Процедурой *РЕАКЦИЯ*(*параметры_оа*) формируется массив *отклик* со значениями откликов среды для каждого автомата адаптации a_i .

Процедурой *ПЕРЕХОД*(*состоян_aa*, *отклик*) автоматы адаптации переходят в новые состояния, которые фиксируются в массиве *состоян_aa*.

Затем процедурой *ВЫХОД*(*граф*, *состоян_aa*) формируется массив *вершины*, в который включаются вершины, для которых реализуется альтернатива A_2 .

Процедурой *УДАЛИТЬ*(*решение*, *вершина*) полное решение задачи заменяется частичным путем удаления из узлов X , перераспределяемых вершин R .

После этого выполняется этап переназначения множества вершин R .

Процедурой *ВЫБОР_ВЕРШ*(*вершины*) выбирается случайным образом *вершина*. Для выбранной вершины процедурой *ВЕРОЯТНО*(*вершина*, *решение*, *управление*) формируется массив распределения вероятности переназначения – *распределение*.

Процедура *ВЫБОР_УЗЛА*(*вершина*, *распределение*) выбирает *узел* для назначения вершины.

Процедурой *ДОБАВИТЬ*(*решение*, *вершина*, *узел*) частичное решение уточняется путем размещения выбранной вершины в выбранный *узел*. После назначения всех вершин множества R массив *решение* вновь будет хранить полное решение задачи разбиения.

Если полученное решение имеет лучшую оценку $F(S)$, то процедурой *СПРАВНЕНИЕ*(*решение*, *лучш_решение*), оно заносится в массив *лучш_решение*.

Отметим, что δ не может быть равной нулю (а следовательно, и α_0), ибо возможна ситуация, когда для некоторой $x_i \in R$ при выборе узла окажется, что для всех доступных узлов значения $\rho_i^v = 0$, $\bar{\rho}_i = 0$, а отсюда вытекает неопределенность значения p_i^v .

Экспериментальные исследования

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

Для проведения экспериментов была использована процедура синтеза контрольных примеров с известным оптимумом F_{opt} по аналогии с известным методом BEKU (Partitioning Examples with Tight Upper Bound of Optimal Solution) [15].

Исследованию подвергались примеры, содержащие до 1 000 вершин. Эксперименты показали, что временная сложность алгоритма на одной итерации имеет оценку $O(n)$, где n – число вершин в гиперграфе.

Наилучшие результаты адаптивный алгоритм показал при следующих значениях управляющих параметров: g (глубина памяти) – 2; T (число итераций) – 300; $\alpha_n=100,01$; $\alpha_0=0,01$; $\Delta\alpha=1$.

Тестирование производилось на бенчмарках 19s, PrimGA1, PrimGA2. По сравнению с существующими алгоритмами достигнуто улучшение результатов на 5–9 %.

В среднем запуск программы со случайными начальными разбиениями обеспечивает нахождение решения, отличающегося от оптимального менее чем на 0,05 %, пяти запусков программы со случайными начальными разбиениями было достаточно для нахождения глобального оптимума.

Работа выполнена при финансовой поддержке гранта РФФИ № 17 - 07 – 00997.

Литература

1. Немудров В., Мартин Г. Системы-на-кристалле. Проектирование и развитие. М.: Техносфера, 2004. 216 с.
2. Малышев И.В., Немудров В.Г. Состояние и перспективы отечественных разработок СБИС типа «система на кристалле» // Системы и средства связи, телевидения и радиовещания. 2003. № 1, 2. С. 56–57.
3. Норенков И.П. Основы автоматизированного проектирования. М.: Изд-во МГТУ им. Н.Э. Баумана, 2006. 446 с.
4. Дендобренко Б.П., Малика А.С. Автоматизация проектирования радиоэлектронной аппаратуры. М.: Высш. шк., 2002. 384 с.
5. Mehta D., Sapatnekar S., Alpert C. Handbook of Algorithms for Physical Design Automation. NY, CRC Press, 2009, 1042 p.
6. Chatterjee A. and Hartley R. A new simultaneous circuit partitioning and chip placement approach based on simulated annealing. Proc. of Design Automation Conf., 1990, pp. 36–39.
7. Анализ и исследование эволюционных методов решения задач разбиения СБИС; [под ред. В.В. Курейчика]. Таганрог: Изд-во ТТИ ЮФУ, 2010. 84 с.

8. Лебедев Б.К., Лебедев О.Б. Разбиение на основе поисковой адаптации // Интеллектуальные системы. Коллективная монография; [под ред. В.М. Курейчика]. М.: Физматлит, 2005. С. 234–256.
9. Alpert C. and Kahng A. A hybrid multilevel/genetic approach for circuit partitioning. Proc. 5th ACM/SIGDA Physical Design Workshop, 2002, pp. 100–105.
10. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. 448 с.
11. Лебедев О.Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования. Таганрог: Изд-во ЮФУ, 2013. 199 с.
12. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация: теория и практика. М.: Физматлит, 2006. 272 с.
13. Цетлин М.Л. Исследования по теории автоматов и моделированию биологических систем. М.: Наука, 1969. 316 с.
14. Лебедев Б.К. Адаптация в САПР. Таганрог: Изд-во ТРТУ, 1999. 160 с.
15. Cong J., Romesis M., and Xie M. Optimality, scalability and stability study of partitioning and placement algorithms. Proc. Intern. Sympos. Physical Design, Monterey, CA. April 2003, pp. 88–94.