

УДК 681.3

АЛГОРИТМ РАЗБИЕНИЯ СВЕРХБОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ
С АДАПТИВНЫМ УПРАВЛЯЮЩИМ ПАРАМЕТРОМ

П.Н. Филиппенко

Рассматривается задача разбиения сверхбольших интегральных схем. В качестве математической модели описания интегральных схем выбран граф. Предложена модификация муравьиного алгоритма оптимизации, используемая для решения задачи разбиения графа. Муравьиный алгоритм – это оптимизационный метод, основанный на принципах самоорганизации и других полезных особенностях поведения муравьев. Предлагается методика, основанная на муравьином алгоритме, с улучшенным методом начального распределения и динамической корректировкой управляющих параметров. Экспериментальные исследования и сравнение с известными алгоритмами показали, что предложенная методика обеспечивает улучшение решений задачи разбиения сверхбольших интегральных схем. Ключевые слова: разбиение, муравьиный алгоритм, управляющий параметр.

Введение

В настоящее время сверхбольшие интегральные схемы (СБИС) могут состоять из миллионов транзисторов. Учитывая огромное количество элементов, из которых состоит схема, ее топология не может быть спроектирована целиком. Это обусловлено ограничениями возможностей вычислительных мощностей компьютеров и недостаточными объемами памяти для одновременной обработки всей схемы [1]. В связи с этим необходимо разбиение схемы с целью получения блоков меньших размеров, работа с которыми возможна и эффективна.

Для математического описания СБИС используем аппарат теории графов. Пусть $G=(V, E)$ – неориентированный граф, где $V = \{v_i | i = 1, 2, \dots, n\}$ – множество вершин графа; $E = \{e_u | u = 1, 2, \dots, n\}$ – множество ребер графа [2]. Для решения задачи разбиения воспользуемся методом перехода от схемы к графу $G=(V, E)$, при котором элементам схемы соответствуют вершины графа V , а электрические цепи представляются ребрами E . В таком случае для решения задачи необходимо разбить V на k непустых частей, $V = V_1 \cup V_2 \cup \dots \cup V_k, \forall t \neq j, V_t \cap V_j = \emptyset$. Критерием оптимизации выступает суммарное число ребер F , находящихся между частями разбиения V_1, V_2, \dots, V_k . Цель оптимизации – минимизация F .

Муравьиный алгоритм – это оптимизационный метод, основанный на принципах самоорганизации и других полезных особенностях поведения муравьев [3]. Поскольку в основе муравьиного алгоритма лежит моделирование перемещения муравьев по заданным путям, этот подход является эффективным способом поиска решений оптимизационных задач [4], допускающих представление в виде графовой модели [2]. Рассмотрим обобщенно традиционный муравьиный алгоритм. Он представляет собой подход к решению комбинаторных задач оптимизации, в частности, задачи нахождения оптимального разбиения множества вершин V графа G на части в соответствии с какой-либо функцией качества [2]. Функцией качества может выступать величина минимального количества ребер, попадающих в разрез между кусками графа разбиения [5]. Муравьиный алгоритм рассматривает компоненты решения S и таблицу феромонов T и построен на принципах самоорганизации и коллективной адаптации [2, 4, 5]. Муравей является агентом, производящим строительство решения S . Решение находится последовательным объединением вершин в подмножества разбиения [6]. Во время избирательного движения агента-муравья от вершины к вершине он принимает решения о добавлении их в соответствующий блок разбиения [5, 7]. Каждая вершина характеризуется значением феромона, отложенного при инициализации или на предыдущих итерациях работы алгоритма агентами, уже проходившими через эту вершину [6]. Вероятность перемещения к вершине и добавления ее в соответствующий блок разбиения определяется агентом и зависит от количества находящегося на вершине феромона и управляющей информации, заложенной в системе изначально [8, 9]. Вероятность вхождения в формируемый кусок для каждой вершины $v_i \in V$ определяется [10] как

$$\text{Pr}(v_i) \equiv \frac{\tau_i^\alpha \eta(v_i)^\beta}{\sum_{v_j \in V} \tau_j^\alpha \eta(v_j)^\beta}, \quad (1)$$

где $\tau_i \in T$ означает, что феромон соответствует вершине v_i , а $\eta(v_i)^\beta$ представляет собой дополнительные априорные знания или управляющую информацию, с помощью которой муравей

оценивает привлекательность вершины v_i [7]. Параметры $\alpha \geq 0$ и $\beta \geq 0$ определяют соотношение между информацией, заключенной в феромонах, и управляющей информацией соответственно [6].

В итоговой части муравьиного подхода осуществляется обновление феромона, и удачные решения помечаются более чем остальные, что повышает вероятность их использования в решениях, найденных на следующих итерациях. Также в этой процедуре феромоны частично испаряются с течением времени [2]. Это делается во избежание застревания в локальных оптимумах во время поиска колониями новых решений.

Приведем способ решения задачи оптимизации с использованием муравьиного алгоритма [5–10]:

1. собрать решение-кандидата (кусочек разбиения) из множества вершин на основе вероятностей, руководствуясь величиной оставленного феромона на вершине и управляющей информацией.
2. обновить модель феромонов, основываясь на качестве решения-кандидата.

Повторять первые два шага, пока результат не будет удовлетворять критериям либо не будет достигнут лимит итераций работы алгоритма [6].

Предложенный метод начального распределения муравьев

В начале работы муравьиного алгоритма через некоторые пути проходит множество муравьев, но есть пути, которые не посещаются. Локальная эвристика, основанная на видимости, поощряет муравьев выбрать вершины, находящиеся ближе. В некоторых вершинах может быть много муравьев, в то время как в других их не будет вообще. Поскольку количество феромона на каждом пути первоначально идентично, муравей, главным образом, использует расстояние между двумя вершинами как эвристический фактор, когда выбирает направление к следующей вершине [10]. В то время как муравьи находятся в относительно большем количестве в определенной вершине, плотность феромона на близлежащих ребрах пути будет усилена из-за относительно большего числа муравьев, путешествующих вдоль пути. Этот путь не обязательно является самым коротким путем, но при этом локально оптимальное решение найдено, и муравьи создают ситуацию попадания алгоритма в локальную яму.

Чтобы решить эту проблему, алгоритм строится таким образом, чтобы муравьи были распределены равномерно, т.е. размещает m муравьев в n вершинах и удостоверяется, что по крайней мере один муравей находится в каждой вершине. Таким образом, производительность поиска и вероятность получения лучшего результата увеличены.

Стратегия изменения управляющего параметра

В муравьином алгоритме управляющая информация важна для генерации высококачественных маршрутов. Поскольку ценность следов феромона не дает достаточной информации в ранней стадии накопления опыта системой и не может направлять муравьев к построению хороших маршрутов, для управляющих параметров могут быть установлены значения большей величины [6, 9]. С другой стороны, на более поздних стадиях управляющему параметру могут быть более полезны небольшие величины значений, потому что следы феромонов к этому моменту накапливают достаточно информации для требуемого поведения, и управляющая информация может вводить в заблуждение, сбивая с правильного пути поиска и направляя к локальным оптимумам. Управляющий параметр устанавливается в качестве константы в традиционном муравьином алгоритме [5, 7]. Этот параметр подбирают экспериментальным путем или аналитически [8]. В настоящей работе в качестве управляющего параметра выступает параметр β (1). Если ценность управляющего параметра является большой, муравей будет пытаться найти другой маршрут (отличный от уже найденного, имеющего большую концентрацию феромона и вызывающего преждевременную сходимость). Очевидно, что маленькая величина управляющего параметра может привести к плохой работе на ранней стадии накопления опыта муравьями. Однако небольшие значения ценности управляющего параметра могут быть эффективны в случае достаточно продолжительного поиска [10, 11]. Исходя из этого, используется интуитивный адаптивный управляющий параметр для муравьиного алгоритма. Предлагается способ назначения адаптивного управляющего параметра, приводящий к улучшению производительности поиска.

Когда муравьиный алгоритм начинает работу, он располагает одинаковой информацией о каждом из маршрутов, в этот момент информационная энтропия системы максимальна. С увеличением количества феромона на путях энтропия будет постепенно уменьшаться. Если в текущий момент энтропией не управлять, она, в конечном счете, уменьшится до нулевого значения. Феромон при этом будет сосредоточен только на одном пути и максимален. В такой ситуации будет получено ошибочное решение, приводящее к преждевременной сходимости [10, 12]. Для того чтобы преодолеть преждевременную сходимость, вызывающую дефекты решения сложных комбинаторных оптимизационных задач с помощью алгоритма, основанного на симуляции муравьиной колонии, необходима его модификация. Предлагается муравьиный алгоритм, основанный на анализе информационной энтропии и использующий значения управляющего параметра, задающегося энтропией. Каждый след – дискретная случайная переменная в матрице распределения феромонов. Энтропия случайной переменной определена как

$$E(X) = -\sum_{i=1}^r P_i \log P_i, \tag{2}$$

где P_i представляет вероятность возникновения каждого следа в матрице распределения феромонов. Для симметрии n вершин матрицы феромонов: $n(n-1)/2$ отмеченных следов феромона, $r = n(n-1)/2$. Максимальная энтропия задается как

$$E_{\max} = -\sum_{i=1}^r P_i \log P_i = -\sum_{i=1}^r \frac{1}{r} \log \frac{1}{r} = \log r. \tag{3}$$

Предлагается использовать величину энтропии в качестве показателя, определяющего степень накопления информации, содержащейся в феромонах на всех участках пути. Управляющий параметр обновляется в соответствии со следующим правилом:

$$\beta = \begin{cases} 5 & A < E' \leq 1 \\ 4 & B < E' \leq A \\ 3 & C < E' \leq B \\ 2 & 0 < E' \leq C \end{cases}, \text{ когда} \tag{4}$$

$$E' = 1 - \frac{E_{\max} - E}{E_{\max}}, \tag{5}$$

где E' является величиной энтропии для текущей матрицы распределения феромонов и рассчитывается по формуле (3), а A, B, C – границы, обусловленные величиной накопления феромонов. Значение границы A установлено в пределах $0,81 \sim 0,89$, границы $B - 0,74 \sim 0,54$, а границу C решено сделать эвристически основанной на величине B .



Рисунок. Упрощенная схема модифицированного муравьиного алгоритма

Предложенный алгоритм объединяет в себе метод начального распределения положения муравьев, повышающий качество поиска на начальных этапах работы, и стратегию изменения управляющего параметра, предотвращающую преждевременную сходимость. Работу предложенного модифицированного алгоритма можно условно описать следующими шагами.

Шаг 1. Ввод исходных данных и инициализация параметров алгоритма.

Шаг 2. Расчет максимальной энтропии по формуле (4).

Шаг 3. Размещение муравьев в стартовых позициях согласно предложенной стратегии распределения.

Шаг 4. Поиск решений муравьиным алгоритмом, описанным выше.

Шаг 5. Перерасчет системы, переопределение и запоминание лучшего результата, полученного в результате работы алгоритма.

Шаг 6. Возможно, улучшение результатов с помощью локального поиска, осуществляемого другим алгоритмом на базе решения, полученного на текущем этапе работы муравьиного алгоритма.

Шаг 7. Проверка выполнения необходимого числа итераций алгоритма. Если не выполнено, то переход к шагу 4, если выполнено, то переход к шагу 8.

Шаг 8. Вычисление значений величин энтропии на ребрах графа по формуле (2).

Шаг 9. Расчет нового управляющего параметра поиска по формуле (5). Для выполнения следующей итерации алгоритма переход к шагу 4.

Шаг 10. Формирование окончательного решения.

Шаг 11. Конец работы алгоритма.

Упрощенная блок-схема алгоритма приведена на рисунке.

Алгоритм разбиения блоков был реализован на языке C++. При создании программы использовался пакет Visual Studio 2008 для 32-разрядных операционных систем семейства Windows, тестирование и эксперименты проводились на IBM-совместимом компьютере с процессором Intel(R) Duo Core(TM) Duo 2Ghz и оперативной памятью 512 MB DIMM PC-3200.

Эффективность разработанного алгоритма разбиения оценивалась путем проведения серии экспериментов на специальных схемах (бенчмарках) PrimGA1. Выполнялось сравнение с муравьиными алгоритмами [6, 12], не использующими предложенные в настоящей работе модификации. В качестве параметров, используемых для сравнения, выступали σ – стабильность алгоритма и κ – коэффициент, определяющий степень улучшения решений.

Стабильность алгоритма определяется как число лучших значений целевой функции на определенном интервале итераций исследуемого алгоритма. Чем больше значение σ , тем более нестабилен алгоритм. Стабильность σ вычисляется следующим образом:

$$\sigma = \left| \frac{f_{cp} - f}{f_{cp}} \right|,$$

где f_{cp} – усредненное значение лучшей целевой функции на определенном интервале итераций; f – значение лучшей целевой функции исследуемого алгоритма.

Коэффициент κ , определяющий степень улучшения решений, вычисляется по формуле

$$\kappa = \left(100 \frac{f}{f'} \right) - 100 [\%],$$

где f' – значение лучшей целевой функции алгоритма, с которым сравнивается исследуемый алгоритм; f – значение лучшей целевой функции исследуемого алгоритма.

В результате сравнения с другими муравьиными алгоритмами [6, 12] удалось установить, что представленный в работе модифицированный муравьиный алгоритм находит оптимальные решения за меньшее число итераций и эффективней в среднем на 6%.

Заключение

Наилучшие результаты муравьиные алгоритмы показывают для задач с большими размерностями областей поиска, к которым относится задача разбиения на блоки. Алгоритмические усовершенствования простого алгоритма приводят к улучшению качества решений. Поскольку от выбора управляющих параметров сильно зависит поведение муравьиных алгоритмов, именно к этой проблеме обращено наибольшее внимание. Предложенный алгоритм объединяет в себе метод начального распределения положения муравьев, повышающий качество поиска на начальных этапах работы, и стратегию изменения управляющего параметра, предотвращающую преждевременную сходимость.

В качестве направления для развития дальнейших исследований можно отметить распараллеливание алгоритма с целью улучшения получаемых решений [13]. Взаимодействия, происходящие на уровне самоорганизации внутри колонии, следует дополнить взаимодействием информацией с другими обособ-

ленными колониями, сосуществующими в едином пространстве поиска. При этом просчет взаимодействий внутри каждой колонии целесообразно реализовывать на отдельных вычислительных машинах, а взаимодействие на межколонииальном уровне – через сеть. В случае продолжения работы в указанном направлении анализ способа выбора управляющих параметров способен повысить качество полученных решений.

Литература

1. Балюк Л.В., Курейчик В.В., Сороколетов П.В. Перспективная технология интегрированного поиска в САПР // Известия ЮФУ. Технические науки. Тематический выпуск «Интеллектуальные САПР». – 2007. – № 2 (77). – С. 18–25.
2. Dorigo M., Stützle T. Ant Colony Optimization. – London: A Bradford, 2004. – 305 p.
3. Курейчик В.М., Кажаров А.А. Муравьиные алгоритмы для решения транспортных задач // Известия РАН. Теория и системы управления. – 2010. – № 1. – С. 32–45.
4. Гладков Л.А., Курейчик В.В., Курейчик В.М. Биоинспирированные алгоритмы в оптимизации. – М.: ФИЗМАТЛИТ, 2009. – 384 с.
5. Курейчик В.М., Кажаров А.А. О некоторых модификациях муравьиного алгоритма // Известия ЮФУ. Технические науки. Тематический выпуск «Интеллектуальные САПР». – 2008. – № 4 (81). – С. 7–12.
6. Лебедев О.Б. Муравьиный алгоритм разбиения // Известия ЮФУ. Технические науки. Тематический выпуск «Интеллектуальные САПР». – 2009. – № 12 (101). – С. 53–58.
7. Курейчик В.В., Курейчик В.М., Родзин С.И. Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. Тематический выпуск «Интеллектуальные САПР». – 2009. – № 4 (93). – С. 16–24.
8. Штовба С.Д. Муравьиные алгоритмы // Математика в приложениях. – 2004. – № 4 (4). – С. 70–75.
9. Лебедев Б.К., Лебедев О.Б. Эволюционные процедуры разбиения СБИС // Известия ЮФУ. Технические науки. – 2008. – № 4 (81). – С. 20–26.
10. Hlaing Z., Khine M. An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem // International Proceedings of Computer Science and Information Technology. – 2011. – V. 16. – P. 54–59.
11. Hinne M., Marchiori. E. Cutting Graphs Using Competing Ant Colonies and an Edge Clustering Heuristic // Proceedings of the 11th European conference on Evolutionary computation in combinatorial optimization. – 2011. – V. 6622. – P. 60–71.
12. Randall M., Lewis A. Modifications and additions to ant colony optimization to solve the set partitioning problem // Proceedings of the 2010 6th IEEE International Conference on e-Science Workshops. – 2010. – P. 110–115.
13. Филиппенко П.Н., Шашелов А.А., Сеитова С.В. Построение систем, обрабатывающих большие вычисления: проблемы и тенденции // Известия ЮФУ. Технические науки. Тематический выпуск «Интеллектуальные САПР». – 2010. – № 12 (113). – С. 196–203.

Филиппенко Петр Николаевич

– Южный федеральный университет, аспирант, peterpain@mail.ru