

непосредственно с вершинами y . В результате каждая ЭС будет описана более полно с учетом косвенных признаков. Это позволит организовать и более эффективный поиск ЭС, максимально схожей с текущей. Наличие семантически обусловленных ассоциативных связей между признаками $x \in X$ позволяет уточнить описание ЭС $y \in Y$ и, тем самым, способствовать повышению правдоподобности модели ПР.

Запрос, по которому осуществляется вывод (поиск соответствующей ЭС), представляет в общем случае нечеткое подмножество признаков, заданных значениями степеней принадлежности, описывающих текущую ситуацию (поисковый образ), подлежащую классификации.

Логический вывод сводится к определению пути с максимальной оценкой между вершиной z , соответствующей текущей ситуации (поисковому образу) и всеми вершинами-заклЮчениями на графе G_I , образованном присоединением вершины z к графу G , задающему поле знаний [2]. Вершина z соединяется с вершинами-признаками $x \in X$ поля знаний ориентированными ребрами с весами, соответствующими степеням принадлежности соответствующего признака x в описании текущей ситуации (поискового образа). Степень соответствия каждой ЭС посланному запросу (текущей ситуации) определяется оценкой пути, полученного в результате логического вывода.

Предложенная модель определения сходства некоторой ситуации (объекта, документа) с каждой из ЭС также может быть использована и для более гибкого отбора из множества хранящихся в БД документов или иных объектов (описаний схем, ситуаций, методов, алгоритмов и т.п.), когда последние характеризуются взаимозависимыми признаками непосредственно или косвенно.

Предложенная выше структура интеллектуальной системы САПР+ИСППР естественным образом предполагает реализацию в виде распределенной архитектуры с использованием технологии «клиент-сервер». Здесь БД, БЗ, БМ находятся на сервере, а процесс проектирования узлов и блоков сложного устройства или схемы осуществляется в параллельном асинхронном режиме на АРМ-ах конструкторов, подключенных к локальной сети.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Берштейн Л.С., Карелин В.П., Целых А.Н. Модели и методы принятия решений в интегрированных интеллектуальных системах. – Ростов-на-Дону: Изд-во РГУ, 1999.
2. Карелин В.П., Целых А.Н. Методы и модели принятия решений в социотехнических системах. Препринт. – Ростов-на-Дону: Изд-во СКНЦВШ, 1999.
3. Тимаков С.О. Интеллектуальная система поддержки принятия решений. Тезисы докл. 1-й Всероссийской н-т. конф. "Компьютерные технологии в науке, проектировании и производстве". – Нижний Новгород, 1999.

А.А. Лежебоков

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ РАЗМЕЩЕНИЯ С УЧЁТОМ ВРЕМЕННЫХ ЗАДЕРЖЕК*

Введение. Процесс проектирования ЭВА является сложным и многоэтапным. Одной из задач проектирования является размещение модулей ЭВА в монтажном

* Работа выполнена при поддержке РФФИ (гранты № 05-08-18115, № 07-01-00511) и программ развития научного потенциала высшей школы 2006-2008 гг. (РНП.2.1.2.3193, РНП 2.1.2.2238).

пространстве. Высокая плотность размещения элементов ЭВА создает большие трудности при реализации соединений между ними. В этой связи задача размещения элементов на плоскости определяет быстроту и качество трассировки.

В общем виде задача размещения формулируется следующим образом: в монтажном пространстве задана область, которая разбивается на множество позиций (посадочных мест) $P = \{p_1, p_2, \dots, p_q\}$, число которых должно быть не меньше числа размещаемых элементов. Каждый элемент может занимать не более одного посадочного места, расстояние между которыми описывается симметричной матрицей расстояний $D = \|\|d_{i,j}\|\|$. Имеется множество элементов, связанных между собой множеством электрических цепей, которые необходимо разместить на множестве P таким образом, чтобы обеспечивался экстремум целевой функции размещения, при соблюдении наложенных ограничений.

Исходными данными при решении задачи размещения является прямоугольная конструкция (поле для размещения элементов), набор элементов, полученный в результате компоновки (разбиения графа схемы на части), и граф соединений или его матричный (списковый) эквивалент. На прямоугольную конструкцию накладывается Декартова система координат с осями s и t , определяющая граф G_r , представляющий собой координатную решетку. Расстояние $d_{i,j}$ между узлами i и j этого графа описывается выражением $d_{i,j} = |s_i - s_j| + |t_i - t_j|$, при условии, что дорожки ведутся только под прямым углом, т.е. имеют строго вертикальное или строго горизонтальное положение.

Основными известными критериями при размещении [1] являются: минимальная суммарная длина связей, минимальная длина самой длинной связи, минимум числа возможных пересечений, минимум числа изгибов соединений, минимальная площадь кристалла. В настоящее время всё более актуальным становится ещё один критерий решения задачи размещения – суммарная величина временных задержек схемы [2]. В [2] описывается модель временных задержек, которую и предлагается использовать для разработки программного обеспечения решения задачи размещения.

Выбор языка и среды программирования. Язык C++ – это универсальный язык программирования, для которого характерны экономичность выражения, современный поток управления и структуры данных, богатый набор операторов. Кроме того, C++ предоставляет гибкие и эффективные средства определения новых типов. Используя определения новых типов, точно отвечающих концепциям приложения, данная программа разделена на легко поддающиеся контролю части. Такой метод построения программ часто называют абстракцией данных. Информация о типах содержится в некоторых объектах типов, определенных пользователем. Такие объекты просты и надежны в использовании в тех ситуациях, когда их тип нельзя установить на стадии компиляции. Программирование с применением таких объектов часто называют объектно-ориентированным. Этот метод дает более короткие, проще понимаемые и легче контролируемые программы.

В качестве среды программирования был выбран Borland C++ Builder 6.0 фирмы Inprise, так как он является мощной системой разработки, имеет множество функций, расширяющих возможности языка C++, облегчает и убыстряет процесс программирования с помощью визуального подхода, позволяет создавать 32-битные приложения для работы в современных многозадачных операционных системах Windows 9x и Windows XP.

Разработка программы. При разработке программной системы решения задачи размещения ставились следующие требования, которым должно удовлетворять разрабатываемое программное обеспечение (ПО):

- ◆ ПО должно обладать интуитивно понятным интерфейсом, для обеспечения эффективной работы инженера-схемотехника.
- ◆ Ввод схемы должен осуществляться следующими способами:
 - ручной ввод;
 - загрузка из файла;
 - случайно синтезированная схема.
- ◆ Должен быть реализован блок сбора и вывода статистики работы алгоритма для осуществления экспериментальных исследований.

Алгоритм решения задачи размещения описан в [3]. Это модифицированный генетический алгоритм (ГА) [4], учитывающий реальные топологические параметры схемы и реализующий модель временных задержек [2] для подсчёта целевой функции.

Разработанная программная система позволяет работать с различными типами элементов. База данных элементов описывается и хранится отдельно, что делает разработанное ПО довольно гибким и универсальным инструментом решения поставленной задачи.

Окно настроек генетического алгоритма и модели временных задержек представлено на рис.1. Настраиваемыми опциями алгоритма являются:

- ◆ выбор стратегии создания начальной популяции;
- ◆ оператор селекции;
- ◆ оператор кроссинговера и его вероятность;
- ◆ операторы мутации и инверсии и их вероятность;
- ◆ размер популяции и число итераций (количество поколений).

К настройкам модели относятся:

- ◆ ведущее сопротивление и сопротивление ёмкость модуля на единицу длины;
- ◆ ёмкость загрузки контакта и ёмкость модуля на единицу длины;

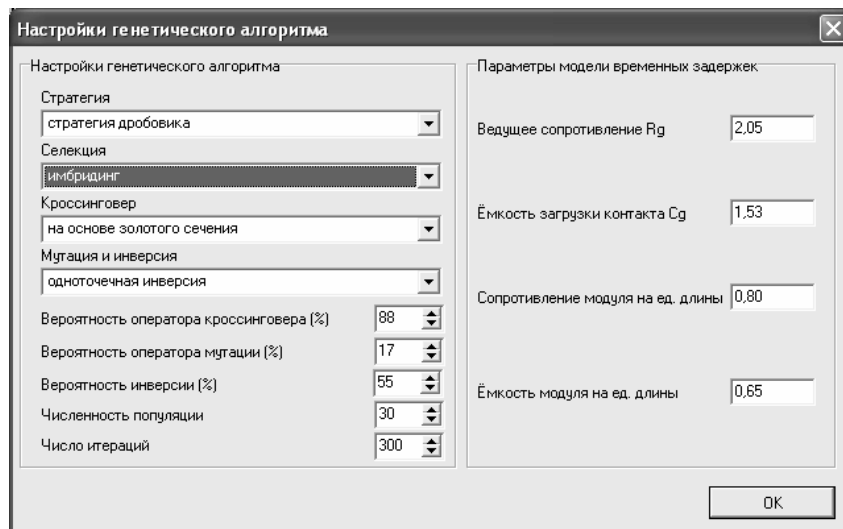


Рис.1. Настройки алгоритма и модели временных задержек

На рис.2 представлено главное окно программы, которое содержит изображение дискретного рабочего поля (ДРП), блок статистики и график улучшения целевой функции.

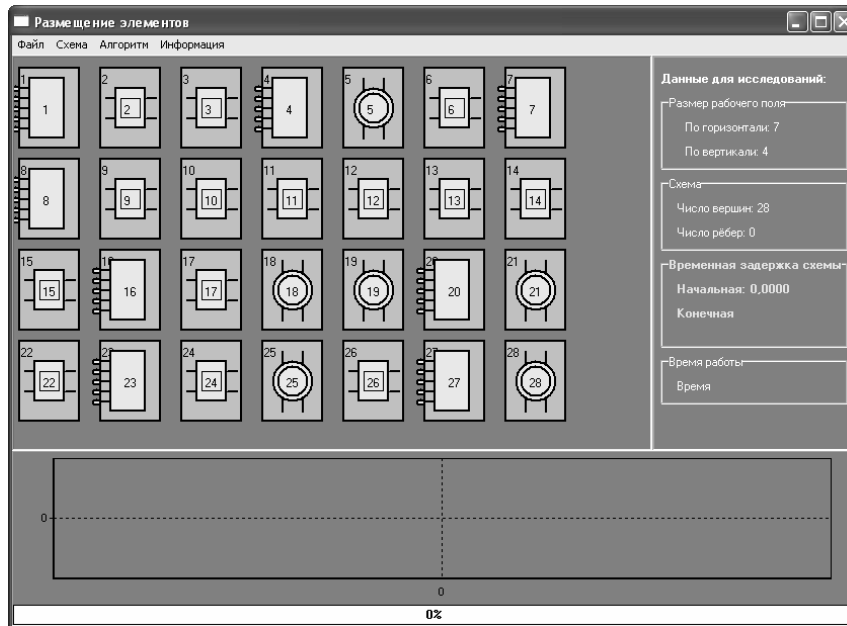


Рис.2. Исходное размещение элементов

После запуска и выполнения алгоритма элементы меняют своё расположение, согласно лучшей хромосоме в популяции. Пример решения задачи представлен на рис.3. По данным из блока статистики и графику улучшения целевой функции имеется возможность провести исследования алгоритма.

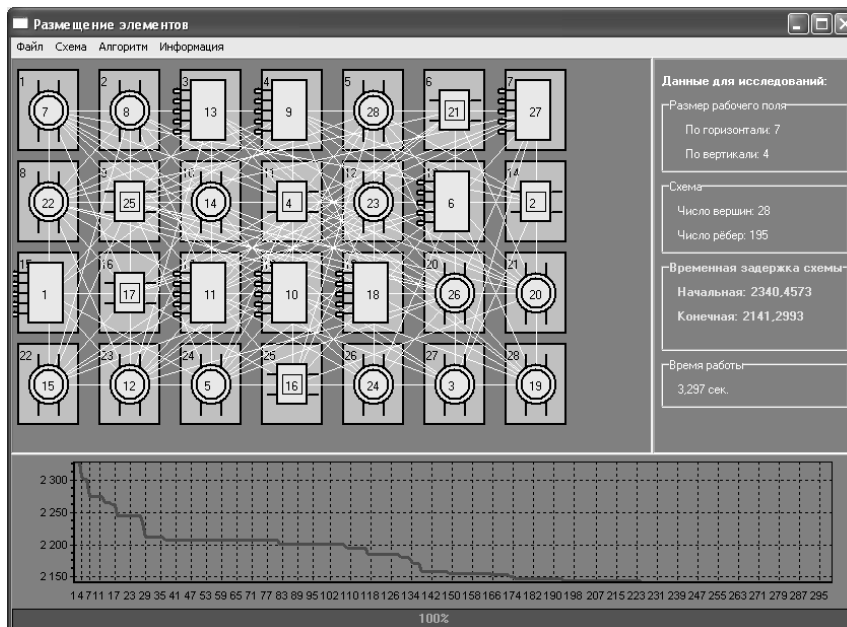


Рис.3. Пример решения задачи размещения

Следует отметить, что в разработанном ПО имеется возможности: ручного редактирования матрицы связности элементов; изменения типов элементов; изменения размеров ДРП; просмотра локальных степеней каждого элемента, для определения максимально и минимально связного элементов.

Экспериментальные исследования. Целью проведения экспериментальных исследований является определение временной сложности алгоритма (ВСА) и сравнение эффективности алгоритма с классическими методами решения задачи.

Временная сложность алгоритма – это зависимость времени работы алгоритма от некоторого управляющего параметра. В данной работе управляющим параметром является размер хромосомы – количество элементов схемы.

Для определения ВСА алгоритма был проведён ряд экспериментов, в которых изменялось число элементов схемы, остальные параметры алгоритма оставались постоянными.

Таблица

Таблица результатов измерений времени работы алгоритмов для разного количества элементов

Элементы	100		200		300		400		500	
Эксп. №1	0,875	0,275	1,89	0,568	3,594	1,002	6,422	1,658	10,344	2,156
Эксп. №2	0,96	0,274	1,859	0,552	3,594	1,11	6,313	1,557	10,203	2,024
Эксп. №3	1,2	0,27	1,891	0,458	3,578	1,125	6,406	1,678	10,407	2,258
Эксп. №4	0,78	0,265	1,859	0,585	3,563	0,988	6,36	1,459	10,25	2,894
Генетический, Среднее значение (сек.)	0,95		1,87		3,58		6,38		10,30	
Последовательный, Среднее значение (сек.)	0,27		0,54		1,06		1,59		2,33	

На рис.4 представлены графики зависимости времени работы от количества элементов схемы для классического последовательного и разработанного генетического алгоритмов.

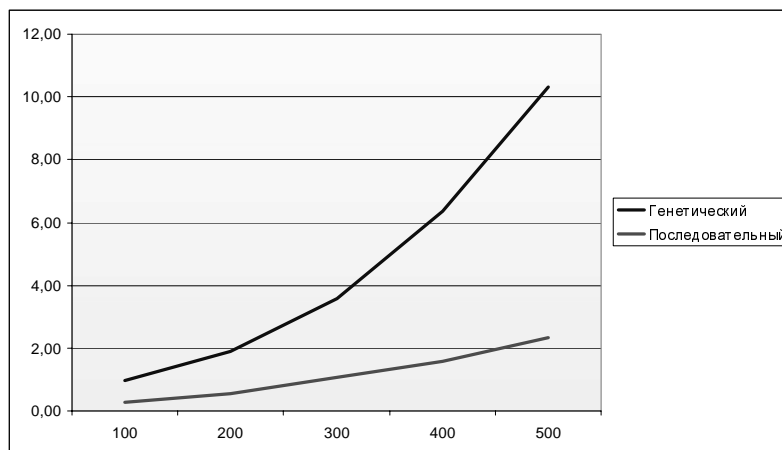


Рис.4. Графики ВСА

Как видно из таблицы и графика, временная сложность генетического алгоритма близка к квадратичной и может быть выражена формулой $O(n^2)$. Для срав-

нения на этом же графике представлена временная сложность последовательного алгоритма, которая имеет линейный характер.

Заключение. Разработанное ПО позволяет эффективно решать задачу размещения элементов СБИС с помощью модифицированного генетического алгоритма. Отдельно хранящееся описание элементов и обширные настройки алгоритма и модели сделали программу достаточно универсальной. Проведённые экспериментальные исследования позволили определить временную сложность алгоритма.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Норенков И.П.* Основы автоматизированного проектирования. – М.: Изд-во МГТУ имени Н.Э.Баумана, 2000. – 360 с.
2. *Лежебоков А.А., Гладков Л.А.* Моделирование временных задержек при решении задачи размещения элементов СБИС // Известия ТРТУ. – Таганрог: Изд-во ТРТУ, № 1, 2007.
3. *Лежебоков А.А., Гладков Л.А.* Генетический алгоритм решения задачи размещения элементов СБИС с учётом временных задержек // Сборник трудов конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте». – Коломна, 2007.
4. *Гладков Л.А., Курейчик В.М., Курейчик В.В.* Генетические алгоритмы. – Ростов-на-Дону: Ростиздат, 2004. – 395 с.
5. *Shervani N.* Algorithms for VLSI physical design automation. – USA, Kluwer Academy Publisher, 1995. – p. 538.
6. *Cong J., He L., Koh C. and Pan Z.* Global interconnect sizing and spacing with consideration of coupling capacitance // In Proc. Int. Conf. on Computer Aided Design. 1997. – p. 570–573.

В.В. Лисяк, Н.К. Лисяк, М.В. Лисяк*

ОБЗОР РАСПРОСТРАНЁННЫХ В РОССИИ САПР ФИРМЫ AUTODESK

Делая выбор в пользу той или иной САПР, следует, прежде всего, убедиться, удовлетворяет ли она следующим требованиям [1]:

- ◆ *Надёжность.* Приобретая систему с нестандартным форматом хранения информации, вы совершаете ошибку.
- ◆ *Доступность.* Если после приобретения САПР вы не сможете пройти необходимого обучения или не на высоте окажется техническая поддержка, то нормальная работа любой САПР маловероятна.
- ◆ *Открытость.* Если система «сама в себе» и её нельзя настроить или доработать под ваши потребности, то выбор такой системы - ошибка! Если смежникам будет тяжело обрабатывать вашу информацию или вы не сможете увеличить мощь системы или круг решаемых задач, то такую систему можно считать «однодневкой».
- ◆ *Стабильность.* Система не должна постоянно менять основные принципы работы. Работа в САПР – это обычаи и своды правил. Поэтому только постоянство стиля САПР позволит вам вовремя и качественно выполнять обязательства.
- ◆ *Масштабируемость.* Хорошая САПР удовлетворит любой кошелёк. Она позволит вам начать с «малого решения» и со временем расширить систему до желаемого уровня.

Программные продукты фирмы Autodesk, Inc. США в значительной мере удовлетворяют перечисленным требованиям и можно сказать, что они являются продуктивными, надёжными и стильными.

* Работа выполнена при поддержке РФФИ (гранты № 05-08-18115, № 06-01-00272) и программ развития научного потенциала высшей школы 2006-2008 гг. (РНП.2.1.2.3193, РНП 2.1.2.2238).