

ведения соединительных связей, расчет оценки производительности схемы, проведение межэлементных соединений, а также ускорить процессы глобальной и локальной маршрутизации.

Анализ получаемых решений показал уменьшение задержки Elmore в среднем на 22%-32% при незначительном увеличении площади кристалла на 3%-6%. Исследования проводились на задачах размерностью от 100 до 200000 элементов. Единственным недостатком алгоритма оказались повышенные требования к памяти ПК.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Jason Cong and David Pan*: Interconnect Estimation and Planning for Deep Submicron Designs, - Department of Computer Science University of California.
2. *Chan-Ping Chen and Yao Ping Chen*: Optimal Wire Sizing Formula Under the Elmore delay Model, - Department of Computer Science University of California University of Texas.

УДК 004.89

М.Д. Сеченов, Ю.А. Киселев

МЕСТО ХАОСА В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

Хаос можно рассматривать как последовательность действий, которая не может быть определена, т.е. уровень энтропии системы очень велик.

Рассмотрим данную проблему на примере иерархической системы с обратными связями. В этой системе существует ряд блоков, решающих поставленные перед ними конкретные задачи. Если в общей задаче существуют подзадачи, которые определенный блок не может решить, то он транспортирует ее блоку, имеющему необходимые функции для решения. Поэтому любая сложная система состоит из блоков, каждый из которых наделен правами и имеет определенный функциональный набор, необходимый для решения поставленных перед ним задач. В тот момент, когда на вход поступает задача, она разбивается каждым вышестоящим (родительским) блоком на подзадачи и передается своим дочерним блокам. Так происходит до тех пор, пока подзадача не становится элементарной и может быть решена блоком, принявшим ее. Теперь рассмотрим, что же происходит с решением подзадачи. Блок, принявший ее, имеет определенные права на манипуляцию составляющими его функциями. В этом случае нет необходимости подавать запрос своему родителю на конкретную последовательность действий. Это означает, что родитель не может знать всех тонкостей решения данной дочерней для него подзадачи. А, следовательно, не может знать и заранее определить действий дочернего блока. В этом случае можно говорить о возникновении хаоса в этих системах. При увеличении числа подзадач уровень хаоса возрастает. Для нахождения оптимального алгоритма решения поставленной задачи последовательность операций у дочерних блоков может меняться, что говорит о хаотизме внутри организованного объекта. Хаос может исчезнуть лишь тогда, когда будут известны все без исключения параметры и алгоритмы действий изучаемого объекта. При изучении объекта степень хаотических действий говорит нам о его изученности. Можно сказать, что если выводы, описывающие поведение объекта, не уменьшают его степень хаотичности, то они скорее могут считаться ложными, нежели истинными.

Для программиста, который создал законченный программный продукт, алгоритм действий программы никогда не будет хаотичным, так как данный человек сам описал последовательность действий. Но при имитации действий интеллектуального объекта всегда должен присутствовать процент энтропии, иначе данный объект просто станет высокоорганизованной программой, ничем не отличающейся от сегодняшних компьютерных программ. При создании интеллектуальных систем, одной из главных целей является воссоздание неопределенности, т.е. действий, ведущих к самостоятельному улучшению или же ухудшению параметров системы. Этого можно добиться лишь при создании системой новых объектов со своими алгоритмами действий.

УДК 681.3

О.В. Коновалов

УНИВЕРСАЛЬНЫЙ ПРОГРАММНЫЙ КОМПЛЕКС АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ НЕЙРОННЫХ СЕТЕЙ*

Для автоматизации процессов принятия решений в САПР в настоящее время используется множество различных методов, зачастую основанных на принципах построения машинного или искусственного интеллекта. Интеллектуальные технологии, применяемые для повышения точности и скорости принятия решений – один из последних этапов развития аналитических подходов. Последние десять лет идет активное развитие аналитических систем нового типа. В их основе лежат имитации природных процессов таких, как деятельность нейронов мозга или процесс естественного отбора.

Для моделирования поведения проектируемых нейронных сетей и алгоритмов их обучения был разработан программный комплекс, реализующий функции графического проектирования и генерации нейросетей с применением стандартных алгоритмических модулей, komponуемых в зависимости от исследуемого типа нейронной модели или потребностей решаемой задачи. Комплекс реализован на языке Java, что обеспечивает его переносимость на различные платформы, включая Unix- и Win32-совместимые. Показатели производительности комплекса, полученные в процессе моделирования, сравнимы с аналогичными показателями для специализированных программ, традиционно реализующих один или несколько статических алгоритмов для некоторых нейросетевых моделей. Модульность программного комплекса предполагает гибкость проведения исследований, когда любая часть, функциональная или алгоритмическая, может быть заменена нестандартной, служащей для реализации необходимого пользователю действия. В системе поддерживаются комплексные эволюционные вычисления для гибридных нейрогенетических и кооперативно-соревновательных моделей. Реализованы функции типичных и нетипичных нейронных сетей [1,3-6]: радиального базиса (RBFNN), Кохонена, Хопфилда, Хэмминга, Розенблатта и многих других. Существует возможность генерировать нейронные сети в автоматическом режиме, когда заданы только начальные условия, порядки межслойных и межнейронных связей, с учетом будущего алгоритма обучения/обслуживания сети.

* Работа выполнена при поддержке Мин. образования, грант № E02-2.0-44