

корректную реализацию данных элементов. Дальнейшее связывание этой библиотеки с кодом пользовательского приложения обеспечивает получение готовой управляющей среды, часть из которой *заведомо корректна*. Кроме того, есть возможность значительно улучшить важный фактор – скорость разработки и, как следствие, затраты на неё.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Документы института системного программирования <http://www.ispras.ru/>.
2. *Фонин Ю., Грассман С.* Архитектура и принципы построения операционной среды «миниОС». Институт Системного Программирования РАН, 2004.
3. *Буч Г.* Объектно-ориентированный анализ и проектирование, 1999.
4. Документация по существующим операционным средам: AvrX (<http://www.larrybarello.org>), TinyOS (<http://www.tinyOS.org>), FreeRTOS (<http://www.freeRTOS.org>).
5. Разработка ОС реального времени для цифрового сигнального процессора <http://www.citforum.ru/>.

О.Н. Родзина, Л.С. Родзина

ЭВРИСТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ И ОПТИМИЗАЦИЯ, ОСНОВАННАЯ НА ПРИРОДНЫХ АНАЛОГИЯХ

Модели эволюционной оптимизации являются далеко не единственными, построенными на природных аналогиях. Другими эвристическими процедурами являются метод отжига, пороговые и потоковые алгоритмы, метод рекордных оценок, конкурирующие с эволюционными вычислениями при решении задач практической оптимизации. Конкуренция происходит по критериям универсальности (пригодность для решения широкого класса задач оптимизации) и вычислительной мощности (качество полученного решения в сочетании с трудоемкостью его получения). Попробуем экспериментально установить сходство и различие этих методов с эволюционными алгоритмами.

Метод отжига был предложен С. Киркпатриком в начале 80-х годов для решения комбинаторных задач оптимизации. В качестве природного аналога был взят процесс кристаллизации эмалевой субстанции. В ходе ее охлаждения степень свободного движения молекул постепенно становится ограниченной, достигая точки равновесия с минимальным энергетическим уровнем в кристаллической структуре. Скорость ее охлаждения подчиняется распределению Больцмана. Аналогия с физическим процессом здесь заключается в том, что решения оптимизационной задачи соответствуют состояниям системы в процессе охлаждения физической субстанции; целевая функция соответствует ее энергетическому уровню; процедура поиска оптимального решения аналогична поиску состояния системы с минимальным энергетическим уровнем, а температура является параметром для управления процедурой оптимизации. Проведенное программное модели-

рование позволяет говорить о значительном влиянии на результативность метода отжига «плана охлаждения», который устанавливает динамику изменения температуры. Оказалось, что метод отжига демонстрирует толерантность к неоптимальным решениям: вероятность «выживания» решения тем выше, чем выше температура и чем меньше разница между сравниваемыми решениями. Медленное «охлаждение», в отличие от быстрого, приводит к лучшим результатам, т.е. выбирать необходимо между скоростью сходимости и результативностью метода. На наш взгляд, именно этими обстоятельствами можно объяснить тот факт, что растет число гибридных систем, в которых метод отжига комбинируется с другими стратегиями поиска, например, для решения задачи о коммивояжере, разбиения графа и других трудных комбинаторных проблем.

Пороговый алгоритм базируется на методе отжига, а основные отличия состоят в стремлении упростить реализацию алгоритма, сократить время поиска и повысить качество получаемых решений. В частности, пороговый алгоритм предусматривает иное, нежели метод отжига, правило принятия решения. По методу отжига, если новое решение приводит к ухудшению целевой функции, то рассчитывается вероятность его использования в ходе дальнейшего поиска. Согласно пороговому алгоритму каждое новое решение должно быть «не намного хуже, чем предыдущее», причем мера ухудшения устанавливается в качестве некоторого порогового значения, которое в процессе работы алгоритма уменьшается до нуля. Экспериментально на примере классической NP-трудной задачи размещения машин были установлены параметры алгоритма, при которых снижается пороговое значение и возрастает эффективность поиска лучшего решения.

Потоковый алгоритм является дальнейшим упрощением порогового алгоритма. Метафорой потокового алгоритма является поведение путешественника при наводнении в гористой местности. При медленно растущем уровне воды новое решение принимается, если значение целевой функции для этого решения выше, нежели предыдущее. От скорости подъема потока воды зависит, сумеет или нет путешественник найти лучшее решение оптимизационной задачи. В этом состоит отличие от порогового алгоритма. Эксперименты по сравнению потокового и порогового алгоритмов показали, что, немного уступая в качестве найденных решений, потоковый алгоритм работает быстрее. На наш взгляд, это объясняется двумя обстоятельствами. Во-первых, практические оптимизационные задачи имеют зачастую много различных хороших решений, во-вторых, с ростом размерности пространства поиска имеется больше возможностей выйти из локальной «ямы».

Метод рекордных оценок аналогичен потоковому алгоритму, а основной областью его применения являются задачи планирования выпуска продукции, транспортная и распределительная задачи, размещение элементов микросхем на кристалле, целочисленное программирование и т.д.

В заключение отметим, что рассмотренные методы оптимизации, основанные на природных аналогиях, практически не имеют ограничений на вид целевой функции. Однако между ними и эволюционными алгоритмами имеется существенное отличие: понятие «размер популяции решений» в

рассмотренных алгоритмах сводится к единице или двум, поиск ведется локально от точки к точке, селекция проводится по специальным правилам, а «память» алгоритмов ограничивается лишь последними лучшими решениями.

С другой стороны, используя такие механизмы, как «план охлаждения» в методе отжига, или аналогичные эвристики в пороговом и потоковом алгоритмах, методе рекордных оценок можно более эффективно управлять скоростью сходимости процесса поиска. Однако эти процедуры не обладают свойством параллелизма, который, наоборот, является отличительной чертой эволюционных алгоритмов.

Данный анализ эвристических программ для решения оптимизационных задач проводился в рамках разработки образовательного контента по дисциплине «Методы оптимизации».

Е.С. Синютин, Ю.И. Волочков

ПРИМЕНЕНИЕ ОПЕРАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ ПРОЕКТИРОВАНИЯ МИКРОПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОНОМНЫХ КАРДИОГРАФОВ И МОБИЛЬНЫХ ПОЛИГРАФОВ

Автономные кардиографы, такие как, например, монитор Холтера, раньше имели только одно назначение – длительная регистрация одного, двух или трех каналов ЭКГ и запись на твердотельный или магнитный носитель. От постановки до считывания пользователь мог иметь только одну возможность взаимодействия с прибором – нажатие на тревожную кнопку. Факт и время нажатия регистрировались и сохранялись на носителе. Нажатие на тревожную кнопку могло означать все что угодно – это зависело от договоренности с врачом-диагностом. Чаще всего монитор даже не обладал индикацией валидности наложения электродов, что часто приводило к большому числу артефактов в получаемой записи или вся запись могла оказаться артефактной [1].

С развитием вычислительной техники появилась возможность расширить базовую функцию автономного кардиографа. Самым простым, но очень эффективным нововведением является встраивание индицирующего элемента. Это может быть жидкокристаллический индикатор, на который выводятся самые важные сведения о состоянии прибора, ходе записи и, возможно, о состоянии самого пациента. Для расширения опций по настройке записи прибор может быть оснащен пленочной клавиатурой. Эти два улучшения приводят к значительному усложнению алгоритма программы микроконтроллера, который является основой устройства. В табл.1 приведены три типа автономных устройства для съема, записи и хранения ЭКГ с различным набором функций.