

Раздел V. Эволюционное моделирование, генетические и бионические алгоритмы

УДК 321.3

В.В. Курейчик, Вл.Вл. Курейчик

БИОИНСПИРИРОВАННЫЙ ПОИСК ПРИ ПРОЕКТИРОВАНИИ И УПРАВЛЕНИИ*

Описывается поиск при проектировании на основе методов, инспирированных природными системами. Описаны эволюционный, генетический, муравьиный и пчелиный алгоритмы. Предложены новые и модифицированные архитектуры биоинспирированного поиска, использующие многоуровневую эволюцию. Это позволяет распараллеливать процесс решения и частично устранять проблему предварительной сходимости алгоритмов. Принципиальным отличием предложенных методов является разделение процесса поиска на два этапа и применение на каждом из этих этапов различных алгоритмов. Проведенные серии тестов и экспериментов позволили уточнить теоретические оценки временной сложности алгоритмов проектирования и их поведение для схем различной структуры. В лучшем случае временная сложность алгоритмов $\approx O(n \log n)$, в худшем случае – $O(n^3)$.

Гибридный поиск; схематехническое проектирование; управление; биоинспирированный алгоритм; генетический алгоритм; муравьиный алгоритм; пчелиный алгоритм.

V.V. Kureichik, Vl.Vl. Kureichik

SEARCH INSPIRED BY NATURAL SYSTEMS, FOR THE DESIGN AND MANAGEMENT

This paper describes the design of search-based methods, inspired by natural systems. We describe an evolutionary, genetic, ant and bee algorithms. Proposed new and modified search architecture, inspired by natural systems that use multi-level evolution. This allows you to parallelize the process of solving the problem and partially eliminate the pre-convergence algorithms. The principal difference between the proposed methods is to divide the search process into two stages and the use of each of these stages, different algorithms. Conducted a series of tests and experiments provide a more accurate theoretical estimates of the time complexity of the design of algorithms and their behavior patterns to different structure. In the best case time complexity of algorithms $\approx O(n \log n)$, in the worst case – $O(n^3)$.

Hybrid search; schematic design; management; algorithm; inspired by natural systems; genetic algorithm; ant algorithm; bee algorithm.

Введение. В связи с развитием нанометровых технологий в области производства возникает проблема перехода на новые методы проектирования. Это требует развития новых методов и алгоритмов. Одним из таких подходов является биоинспирированный поиск, использующий стратегии эволюционного моделирования и принципы природных механизмов принятия решений [1, 2]. Биоинспирированный поиск включают в себя генетические алгоритмы (ГА), эволюционные алгоритмы (ЭА), алгоритмы, моделирующие механизмы принятия решений природными системами. Это методы роевого интеллекта (муравьиные (МА), пчели-

* Работа выполнена при частичной поддержке РФФИ (проекты: № 10-01-00115, № 12-01-31356).

ные (ПА) и др.), основанные на принципах коллективного поведения децентрализованной самоорганизующейся системы. Данный класс алгоритмов анализирует различные области пространства решений одновременно, и они более приспособлены к нахождению новых областей с оптимальными значениями целевой функции (ЦФ) при решении задач проектирования и управления.

Генетические и эволюционные алгоритмы. Генетические алгоритмы осуществляют поиск баланса между эффективностью и качеством решений за счет «выживания сильнейших альтернативных решений», в неопределенных и нечетких условиях. Приведем формальное определение ГА [1–4]:

$$GA = (P_{i0}, N, P_iT, k, T, L_j, A, (ЦФ, ОГР, ГУ), ГО, ОМГ, t),$$

где P_{i0} – исходная популяция хромосом (альтернативных решений), $P_{i0} = (P_{i01}, P_{i02}, \dots, P_{i0n})$, P_i – хромосома (альтернативное решение); N – мощность популяции; $T = 0, 1, 2, \dots$ – номер поколения, проходящего популяцией во время эволюции; L_j – длина i -ой хромосомы, т.е. число генов (элементов, входящих в закодированное решение, представленное в заданном алфавите); (ЦФ, ОГР, ГУ) – целевая функция, ограничения и граничные условия, которые определяются на основе заданной модели исходной решаемой задачи; ГО – генетические операторы, ОМГ – оператор миграции, t – критерий окончания работы ГА [1–4].

При решении практических задач с использованием ГА, обычно, выполняют четыре предварительных этапа: выбор способа представления решения; разработка операторов случайных изменений; определение способов «выживания» решений; создание начальной популяции альтернативных решений [3].

Приведем модифицированный генетический алгоритм. Сначала создается начальная популяция решений, далее определяется ЦФ для каждой хромосомы и средняя ЦФ всей популяции, затем выбираются родительские решения и выполняются различные генетические операторы до получения удовлетворительного решения или выполнения критерия останова [1, 3].

Качество конкретного генетического алгоритма можно оценить расстоянием $D = (f_1 - f_2) / f_1$, где f_1 – глобальный оптимум, полученный экспериментально (если он известен), а f_2 – лучшее значение, найденное этим алгоритмом.

Если значение глобального оптимума известно, то условием окончания работы алгоритма можно считать нахождение значения ЦФ, превышающей глобальное значение на заданную величину ϵ в случае минимизации ($ЦФ \geq ЦФ_{гл} + \epsilon$, здесь ЦФ – текущее значение, а ЦФ_{гл} – известное глобальное значение целевой функции). Если значение глобального оптимума неизвестно, то приоритетом является время работы или количество заданных генераций алгоритма. Отметим, что эволюционный алгоритм является частным случаем генетического алгоритма. В эволюционном алгоритме применяется только оператор мутации [1].

Муравьиные методы и алгоритмы. Одним из биоинспирированных подходов является метод роевого интеллекта включающий в себя муравьиные алгоритмы, пчелиные алгоритмы, метод роя частиц, алгоритм каплей воды и др.

Роевой интеллект описывает коллективное поведение децентрализованной самоорганизующейся системы, которая состоит из множества агентов, локально взаимодействующих между собой и с окружающей средой. Агенты обычно довольно просты, но, локально взаимодействуя, вместе создают, так называемый, роевой интеллект [2, 5, 6].

Муравьи относятся к социальным насекомым, живущим внутри некоторого коллектива – колонии. Основу «социального» поведения муравьев составляет самоорганизация – множество динамических механизмов, обеспечивающих достижение системой глобальной цели в результате низкоуровневого взаимодействия ее элементов. Принципиальной особенностью такого взаимодействия является ис-

пользование элементами системы только локальной информации. При этом исключается любое централизованное управление и обращение к глобальному образу, репрезентирующему систему во внешнем мире. Самоорганизация является результатом взаимодействия следующих четырех компонентов: случайность; многократность; положительная обратная связь; отрицательная обратная связь.

Муравьиные алгоритмы представляют собой вероятностную жадную эвристику, где вероятности устанавливаются, исходя из информации о качестве решения, полученного из предыдущих решений [5–7].

В сравнении с ГА муравьиные алгоритмы имеют некоторые преимущества: опираются на память всей колонии вместо памяти только о предыдущем поколении и меньше подвержены неоптимальным начальным решениям (из-за случайного выбора пути и памяти колонии).

Ряд экспериментов показывает, что эффективность муравьиных алгоритмов растёт с ростом размерности решаемых задач оптимизации. Хорошие результаты получаются для нестационарных систем с изменяемыми во времени параметрами.

Важным свойством муравьиных алгоритмов является неконвергентность: даже после большого числа итераций одновременно исследуется множество вариантов решения, вследствие чего, не происходит длительных временных задержек в локальных экстремумах. Перспективными путями улучшения муравьиных алгоритмов являются on-line адаптация параметров с помощью базы нечетких правил, а также их гибридизация с другими методами природных вычислений, например генетическими алгоритмами. Гибридизация может осуществляться по островной схеме, когда различные алгоритмы решают задачу параллельно и автономно, или по принципу «мастер-подмастерье», когда основной алгоритм – «мастер» передает решение типовых подзадач «подмастерью».

Пчелиные методы и алгоритмы. Пчелиный алгоритм – это оптимизационный алгоритм, в основе которого лежит поведение пчёл в живой природе [5, 8, 9].

Применительно к задаче оптимизации в пчелином алгоритме каждое решение представляется в виде пчелы, которая знает (хранит) расположение (координаты или параметры многомерной функции) какого-то участка. Выделим два варианта поведения.

В первом варианте две пчелы нашли два разных пересекающихся участка, и оба этих участка следует отметить как лучшие или выбранные. Во втором варианте будем считать, что это один участок, центр которого находится в точке, соответствующем большому значению целевой функции.

Второй вариант поведения менее подвержен попаданию в локальные оптимумы за счет просмотра перспективных мест и их окрестностей. Причем, на каждой итерации алгоритма область просмотра уменьшается.

Основная идея пчелиного алгоритма заключается в том, что все пчелы на каждом шаге будут выбирать как элитные участки для исследования, так и участки в окрестности элитных, что позволит, во-первых, разнообразить популяцию решений на последующих итерациях, во-вторых, увеличить вероятность обнаружения, близких к оптимальным решения. После чего в окрестности остальных участков ($m-e$), в зависимости от значения их ЦФ, отправляются рабочие пчелы ($l = N-n$).

Таким образом, работа алгоритма зависит от следующих основных параметров: общего числа пчёл-разведчиков (N); общего числа участков (m); числа элитных участков (e); числа пчёл-разведчиков на элитных участках (n); числа пчёл (l) на остальных ($m-e$) участках; начального размера участков, т.е. размера участков вместе с их окрестностями (S); максимального числа итераций (I).

Приведём словесное описание алгоритма пчёл [9]. Сначала создается популяция пчёл и производится оценка их ЦФ, затем выбор участков для поиска в их окрестностях и отправка пчёл-разведчиков, далее производится выбор пчёл с луч-

шими значениями ЦФ с каждого участка и отправка рабочих пчёл, осуществляющих случайный поиск с оценкой их ЦФ. Затем формируется новая популяция и производится проверка условия окончания работы алгоритма.

Таким образом, ключевой операцией алгоритма пчёл является совместное исследование перспективных областей и их окрестностей. В конце работы алгоритма популяция решений будет состоять из двух частей: пчёлы с лучшими значениями ЦФ элитных участков, а также группы рабочих пчёл со случайными значениями ЦФ. Отличительной особенностью алгоритма является способность динамически разбивать поисковое пространство на области, что уменьшает время работы алгоритма. Данный алгоритм иллюстрирует стратегию поиска «Разделяй и властвуй». Главным преимуществом является тот факт, что резко снижается вероятность попадания в локальный оптимум, а за счёт распараллеливания уменьшается время.

Пчелиный алгоритм легко распределяется на несколько параллельных процессов, за счёт чего значительно увеличится его скорость. По сравнению с генетическим алгоритмом, операторы которого могут быть реализованы различным образом, имеет лишь один оператор, что делает его более простым в использовании. Концепция этих методов основывается на двух совершенно разных природных процессах: пчелиный алгоритм основывается на социальном поведении роя, а генетический алгоритм имитирует процесс эволюции и естественного отбора. За счёт этого есть возможность объединения этих методов.

Модифицированные архитектуры поиска. Основной трудностью решения задач проектирования с большим количеством локальных оптимумов является предварительная сходимость алгоритмов. Другими словами, попадание решения в один, далеко не самый лучший, локальный оптимум [1]. Для эффективного решения этой проблемы предлагаются параллельные методы и архитектуры [1–4, 9]. Это позволит сократить количество компьютерных ресурсов, время поиска и позволит получать оптимальные и квазиоптимальные результаты за полиномиальное время.

Предлагается идея многоуровневой эволюции, когда биоинспирированный поиск используется на двух уровнях. Архитектура такого поиска приведена на рис. 1.

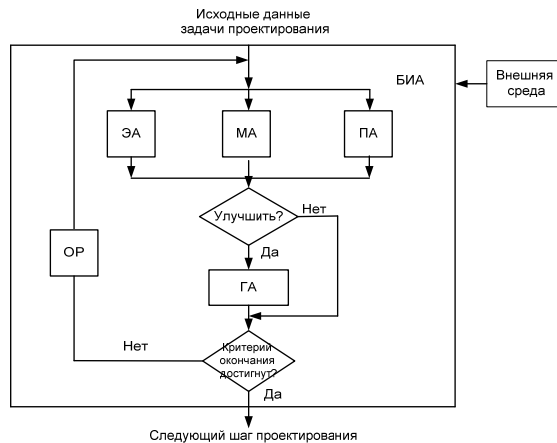


Рис. 1. Архитектура поиска на основе многоуровневой эволюции

В первом случае на первом уровне выполняется параллельная реализация таких алгоритмов, как ПА, МА, ЭА, пока они не начнут сходиться. Затем в наилучших решениях (хромосомах) каждой популяции определяются ближайшие друг к другу объекты, которые образуют кластеры. На основе кластеров создается новая хромосома как объект для второго уровня выполнения ГА. Затем, если решение,

улучшенное генетическим алгоритмом, оптимально, осуществляется переход к следующему шагу конструкторского проектирования. Иначе применяется оператор репродукции и поиск продолжается итерационно до получения эффективных решений. Внешняя среда оказывает влияние на осуществление поиска и настройки используемых алгоритмов.

Во втором случае на первом уровне предлагается использовать ПА, позволяющий быстро разбивать поисковое пространство на области высоких значений ЦФ (модель островов). Архитектура такого поиска приведена на рис. 2.

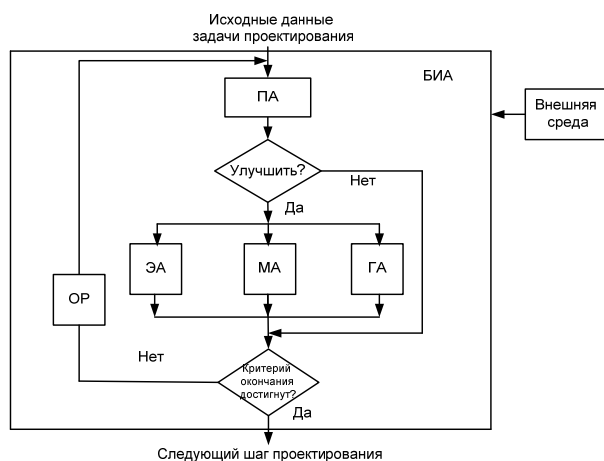


Рис. 2. Архитектура поиска на основе островов

В таком алгоритме введен адаптивный фильтр, отсекающий решения с низким значением ЦФ. Причем нижняя граница ЦФ адаптивна в смысле зависимости от решений на каждой генерации. Здесь популяции в ГА, МА, ЭА эволюционируют в течение нескольких заданных генераций независимо друг от друга на определенных «островах», а затем производится обмен генетическим материалом и снова независимая эволюция. Этот процесс происходит в цикле разнообразными способами.

В третьем случае на первом уровне выполняется генетический алгоритм. Здесь хромосомы одной популяции моделируются как точки в некотором пространстве. Вводится отношение близости, и генетические операторы применяются к хромосомам, расположенным близко друг от друга. Это позволяет быстрее получать локальные оптимумы. Далее выполняется параллельная реализация таких алгоритмов как ПА, МА, ЭА, для получения квазиоптимальных решений.

Отметим, что эти структуры могут быть модифицированы на любое число уровней в зависимости от памяти и временных ограничений на получение результата.

Заключение. Приведенные методы и алгоритмы позволяют распараллеливать процесс поиска, эффективно им управлять и получать оптимальные и квазиоптимальные результаты за полиномиальное время. Принципиальным отличием предложенных методов является разделение процесса поиска на два этапа и применение на каждом из этих этапов различных алгоритмов. Это позволяет частично устранить преждевременную сходимость алгоритмов. Разработана программная среда на языке C++. Проведен вычислительный эксперимент. Проведенные серии тестов и экспериментов позволили уточнить теоретические оценки временной сложности алгоритмов проектирования и их поведение для схем различной структуры. В лучшем случае временная сложность алгоритмов $\approx O(n \log n)$, в худшем случае – $O(n^3)$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Курейчик В.В., Курейчик В.М., Гладков Л.А., Сороколетов П.В. Бионспирированные методы в оптимизации. – М.: Физматлит, 2009.
2. Курейчик В.В., Курейчик В.М., Родзин С.И. Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-27.
3. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2010.
4. Курейчик В.М., Кныш Д.С. Проблемы, обзор и параллельные генетические алгоритмы: состояние // Известия РАН. Теория и системы управления. – 2010. – № 4. – С. 72-82.
5. Abraham A., Grosan G., Ramos V. Swarm Intelligence in Data Mining. – Berlin - Heidelberg: Springer Verlag, 2006.
6. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Разбиение на основе моделирования адаптивного поведения биологических систем // Нейрокомпьютеры. Разработка применение. – 2010. – № 2. – С. 28-33.
7. Dorigo M., Maniezzo V., Colomi A. The Ant System: Optimization by a colony of cooperating objects // IEEE Trans. on Systems, Man, and Cybernetics. – 1996. – Part B. – № 26 (1). – P. 29-41.
8. Karaboga D. An idea based on honey bee swarm for numerical optimization // Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
9. Курейчик В.В., Запорожец Д.Ю. Роевой алгоритм в задачах оптимизации // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 28-32.

Статью рекомендовала к опубликованию д.т.н., профессор Л.С. Лисицына.

Курейчик Владимир Викторович – Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге; e-mail: vkur@tsure.ru; 347928, г. Таганрог, пер. Некрасовский, 44, ГСП 17А; тел.: 88634371651; кафедра систем автоматизированного проектирования; зав. кафедрой; д.т.н.; профессор.

Курейчик Владимир Владимирович – e-mail: kureichik@yandex.ru; кафедра систем автоматизированного проектирования; студент.

Kureichik Vladimir Victorovich – Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: vkur@tsure.ru; GSP 17A, 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; head of department; dr. of eng. sc.; professor.

Kureichik Vladimir Vladimirovich – e-mail: kureichik@yandex.ru; the department of computer aided design; student.

УДК 621.3.049:004.896

Д.Ю. Запорожец, Д.В. Заруба, А.А. Лежебоков

**ОБ ОДНОМ СПОСОБЕ КОДИРОВАНИЯ РЕШЕНИЯ ДЛЯ ЗАДАЧИ
РАЗМЕЩЕНИЯ***

Рассматривается современное состояние проблемы размещения компонентов сверх-больших интегральных схем. Предложен новый способ кодирования решения в генетическом алгоритме, используемом для решения задачи размещения. Приводится пример хромосомы, рассматривается процедура получения координат для каждого фрагмента схемы и дальнейшего построения дерева размещения на основе обратной польской записи. Приведены результаты экспериментальных исследований, показавшие эффективность модифицированного генетического алгоритма на основе предложенного способа кодирования решений.

Размещение; генетический алгоритм; кодирование и декодирование решений; обратная польская запись; дерево решений.

* Работа выполнена при частичной поддержке РФФИ (проект № 12-01-31356).