19. *Kuliev E.V. Dukkardt A.N, Kureychik V.V. Legebokov A.A.* Neighborhood Research Approach in Swarm Intelligence for Solving the Optimization Problems, *Proceedings of IEEE East-West Design & Test Symposium – (EWDTS'2014) Kiev, Ukraine, September 26–29, 2014*, pp. 112-115.
20. *Garey M.R.,Graham R.L., Johnson D.S., Yao A.C.* Resource constrained scheduling as generalized bin packing, *J. CombinatorialTheory*. Ser. A21, pp. 257-298.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.А. Гатчин.

Орлов Антон Николаевич – Южный федеральный университет; e-mail: ky92@mail.ru; 347924, г. Таганрог, ул. Воскова, 111/а, кв. 2; тел.: 88634371651; кафедра систем автоматизированного проектирования; аспирант.

Курейчик Владимир Викторович – e-mail: vkur@sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; зав. кафедрой; д.т.н.; профессор.

Глущенко Александр Евгеньевич – e-mail: alex-14-93@mail.ru; 347939, г. Таганрог, ул. Сызранова, 10, кв. 97; тел.: 89612941882; кафедра систем автоматизированного проектированияж аспирант.

Orlov Anton Nikolaevich – Southern Federal University; e-mail: ky92@mail.ru; 111/a, Voskova street, ap. 2, Taganrog 347924, Russia; phone: +78634371651; the department of computer aided design; postgraduate student.

Kureichik VladimirVictorovich – e-mail: vkur@sfedu.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; head of department; dr. of eng. sc.; professor.

Glushchenko Alexander Evgenyevich – e-mail: alex-14-93@mail.ru; 10, Syzranova street, ap. 97, Taganrog, 347939, Russia; phone: +79612941882; the department of computer aided design; postgraduate student.

УДК 004.896

Д.В. Заруба, Д.Ю. Запорожец

ГЕНЕРАЦИЯ БИОИНСПИРИРОВАННЫХ ПОИСКОВЫХ ПРОЦЕДУР ДЛЯ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ*

Рассматриваются проблемы использования принципов поведения объектов живой природы для решения NP-полных оптимизационных задач. В качестве наиболее перспективных рассматриваются методы и алгоритмы на основе роевого интеллекта. Для решения проблемы баланса между скоростью сходимости и широтой поиска применяются механизмы адаптации. Методы адаптации осуществляют закономерное изменение значений настроечных параметров алгоритмов таким образом, что обеспечивается последовательный переход от диверсификации на начальных этапах работы биоинспирированного алгоритма к интенсификации на заключительных итерациях. В статье предложена подсистема генерации новых гибридных алгоритмов на основе методов биоинспирированного поиска. Предложена архитектура подсистемы, включающая в себя блоки управления данными об алгоритме, решаемой задачи, а так же модуля автоматизации процесса генерации новых поисковых процедур. В статье предлагается новый оптимизационный подход, основанный на механизме гибридизации различных атомарных поисковых процедур. Данный модуль основан на механизмах генетического поиска и генетического программирования. Для обеспечения работоспособности модуля генерации новых решений автором построены новые механизмы кодирования и декодирования стандартизированного представ-

* Исследование выполнено за счет гранта Российского научного фонда (проект № 14-11-00242) в Южном федеральном университете.

ления алгоритма оптимизации. Они позволяют представлять стандартные описания в виде альтернативных решений (хромосом). В статье предлагается новый подход к описанию структуры данных, основанной на XML описании. В качестве структуры данных в хромосоме предлагается использовать обратную польскую нотацию (форма записи выражений, в которой операнды расположены перед знаками операторов). Результатом работы данного модуля и подсистемы в целом является исполняемый алгоритм решения оптимизационных задач.

Роевые методы оптимизации; атомарная поисковая процедура; биоинспирированный поиск; XML; обратная польская нотация.

D.V. Zaruba, D.Yu. Zaporozgets

GENERATION OF BIOINSPIRED SEARCH PROCEDURES FOR OPTIMIZATION PROBLEMS

The paper deals with the problem of using the principles of behavior of wildlife objects to solve NP-complete optimization problems. The most promising deals with methods and algorithms based on swarm intelligence. To solve the problem of the balance between the rate of convergence and the breadth of the search mechanisms of adaptation are applied. adaptation techniques, the regular change in the algorithm tuning parameters values in a way that ensures a gradual transition from the diversification in the early stages of bioinspired algorithm to intensify in the final iteration. The paper proposes a new hybrid generation subsystem of algorithms based on bioinspired search methods. The architecture of a subsystem, including a data control about the algorithm of the problem, as well as the module automate the process of generating new search procedures. The paper proposes a new optimization approach based on the mechanism of hybridization of different atomic search procedures. This module is based on the mechanisms of genetic research and genetic programming. To ensure the efficiency of the module generating new solutions by the author constructed new mechanisms of encoding and decoding of a standardized presentation of the optimization algorithm. They allow you to provide standard descriptions in the form of alternative solutions (chromosomes). The paper proposes a new approach to the description of the data structure based on the XML description. As a data structure in the chromosome are encouraged to use reverse Polish notation (form of the expression, in which the operands are located in front of characters operators). The result of this module, and subsystems as a whole is an executable algorithm for solving optimization problems.

Swarm optimization methods; atomic search procedure; bioinspired search; XML; reverse Polish notation.

Введение. Для эффективного решения задач глобальной оптимизации в 1980-х гг. начали интенсивно разрабатывать класс стохастических поисковых алгоритмов оптимизации, которые называют биоинспирированными алгоритмами. К ним можно отнести алгоритм роя частиц, колонии муравьев, роя пчел, роя светлячков и др. В условиях быстрого прогресса в области информационных технологий классические оптимизационные алгоритмы не позволяют находить эффективные решения или требуют большое количество процессорного времени для их поиска [1]. Для повышения эффективности и качества решения задач глобальной оптимизации целесообразна разработка подсистемы, позволяющей генерировать биоинспирированные поисковые процедуры на основе генетического и пчелиного алгоритмов.

Биоинспирированные алгоритмы предполагают параллельную проработку сразу нескольких альтернативных решений задачи оптимизации и представляют собой альтернативу классическим поисковым алгоритмам, которые работают только с одним решением поставленной оптимизационной задачи. Все биоинспирированные алгоритмы относятся к классу эвристических алгоритмов, то есть алгоритмов, для которых теоретически не доказана сходимость к глобальному оптимуму, но эмпирически установлено, что вероятность получения оптимального или квазиоптимального решения высока [2].

1. Общие положения теории биоинспирированного поиска. В качестве общего названия элементов популяции предлагается использовать термин агент. В различных биоинспирированных алгоритмах агенты называются хромосомами, муравьями, бактериями, пчелами и т.д. Совокупность агентов называется популяцией.

Для получения начальной популяции могут быть применены детерминированные или случайные алгоритмы. При формировании начальной популяции агентами покрывается по возможности вся область поиска.

Одним из основных преимуществ биоинспирированных алгоритмов является их модульная структура. Такой поход позволяет быстро получить большое число новых вариантов алгоритма за счет разработки новых и модификации имеющихся правил инициализации и генерации новых агентов.

Одной из основных проблем создания биоинспирированных алгоритмов является баланс между скоростью сходимости алгоритма (интенсификацией) и диверсификацией поиска (разнообразием агентов). Высокая скорость сходимости требует уменьшение разнообразия агентов в популяции, а диверсификация, наоборот, отвечает за более широкий набор агентов, покрывающих все пространство поиска, а значит и высокую вероятность нахождения глобального решения задачи [1, 2]. Принцип диверсификации должен обеспечивать сохранность разнообразия агентов популяции на протяжении как можно большего количества итераций биоинспирированного алгоритма.

Одним из подходов для решения проблемы баланса между скоростью сходимости и широтой поиска являются механизмы адаптации. Методы адаптации осуществляют закономерное изменение значений настроечных параметров алгоритмов таким образом, что обеспечивается последовательный переход от диверсификации на начальных этапах работы биоинспирированного алгоритма к интенсификации на заключительных итерациях.

На сегодняшний день перспективным средством создания биоинспирированных алгоритмов является гибридизация бионических алгоритмов с целью получения новых структур, повышающих вероятность нахождения глобального оптимума за счет достижения баланса между интенсификации и диверсификации на разных этапах гибридного поиска [1, 3–5].

2 Архитектура подсистемы генерации биоинспирированных поисковых процедур. На рис. 1 представлена обобщенная архитектура подсистемы генерации биоинспирированных поисковых процедур.

В блоке 1 ЛПР вводит стандартизированные описания начальных иерархических поисковых механизмов, а также атомарные поисковые процедуры (АПП) в виде, необходимом для подсистемы. Данные об АПП сохраняются в базу данных АПП (БДАПП), а стандартизированные описания передаются непосредственно на модуль генерации иерархических поисковых процедур.

На втором этапе (Блок 2) подсистема на основании входных данных генерирует новые иерархические процедуры. Используется весь спектр АПП, хранящийся в БДАПП. На третьем этапе (Блок 3) происходит сборка полученной иерархической поисковой процедуры.

В качестве входных данных выступают результаты работы блока 2 в виде стандартизированного описания, а так же данные, полученные в блоке 5. В блоке 5 ЛПР предлагается определить функции создания начального закодированного решения, а также функцию расчета целевой функции calculate() [1, 4]. Обе эти функции необходимо реализовывать с учетом поставленной задачи. Функция генерации начального закодированного решения в общем случае состоит из двух этапов: поиск решения и его кодирования [5]. Очевидно, что также перед ЛПР стоит задача определения функций кодирования и декодирования. Модуль сборки преобразует записанную в XML-файле информацию в конечную последовательность действий, которая выполняется в Блоке 4, а результатом ее работы является исполняемый алгоритм оптимизации.

Блок 4 представляет собой модуль тестирования и получения результатов.

Блок 6 отвечает за вывод данных конечному пользователю. Используя данный модуль, ЛПР может получить всю информацию о процессе генерации алгоритма, стандартизированное описание полученного иерархического механизма, исполняемый алгоритм в виде подключаемой библиотеки и информацию о значениях настроечных параметров, использовавшихся в ходе получения новых иерархических механизмов.

Проведя анализ полученной архитектуры можно сделать вывод, что наиболее трудоемким для ЛПР являются этапы ввода исходной информации. На этом этапе ЛПР необходимо разработать оптимизационную иерархию, вычислить теоретические оценки эффективности, переложить алгоритм на предложенное стандартизированное описание. Данная информация вводится единожды и может быть использована и модифицирована в последующем в рамках разрабатываемой подсистемы.

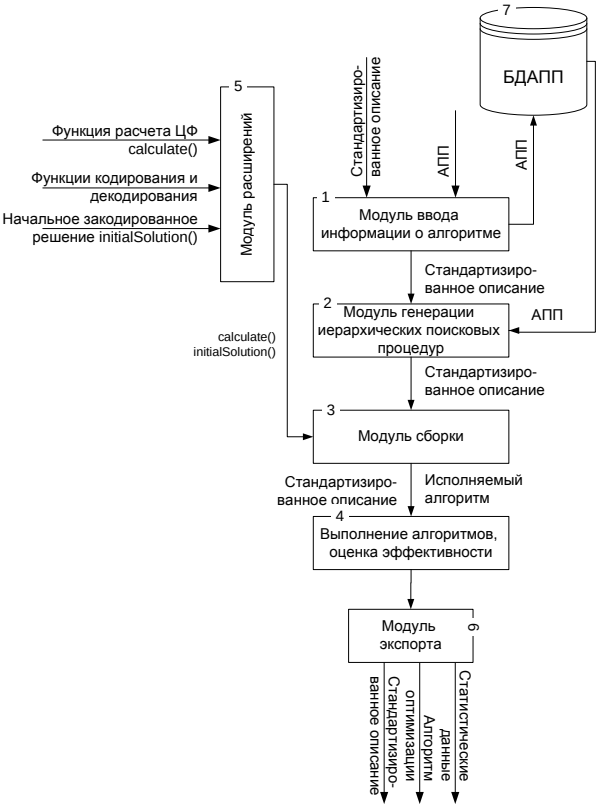

Рис. 1. Обобщенная архитектура подсистемы генерации биоинспирированных поисковых процедур

В статье предлагает новый оптимизационный подход, основанный на механизме гибридизации различных АПП. Под АПП будем понимать алгоритм, имеющий унифицированное описание (интерфейс), принимающий на вход закодированное решение и возвращающий на выходе одно или несколько оптимальных или квазиоптимальных закодированных решений. Интерфейс АПП ($АПП_{инт}$) выглядит следующим образом:

$$АПП_{инт} = \langle id, solution, objectiveFunction, attributesList \rangle,$$

где id – идентификатор алгоритма; solution – начальное закодированное решение; objectiveFunction – целевая функция; attributeList – список атрибутов, необходимых для работы алгоритма (например, в качестве атрибута может выступать количество итераций).

С целью создания эффективной иерархической архитектуры на основе АПП, автор вводит язык описания поисковой иерархии на основе стандартного языка расширяемой разметки XML (2008) [6].

Для того чтобы при создании иерархической поисковой архитектуры имелась возможность создавать последовательные и параллельные поисковые процедуры предлагается ввести следующие сущности: flow–сущность, описывающая глобальную процедуру поиска, и subflow–сущность, описывающая различные процессы, происходящие в рамках глобальной поисковой процедуры. Для обеспечения взаимодействие автор предлагает использовать различные виды отношений. Опишем возможные отношения между введенными сущностями. Отношение parent – указывает на сущность, которая является родительской для текущей сущности. Родительская сущность – сущность, которая инициирует работу текущей сущности после завершения своей работы. Соответственно отношение child – указывает на сущность, которая является дочерней для текущей сущности.

Сущность foreach вводится для того, чтобы появилась возможность динамического запуска нескольких процессов.

Рассмотрим наглядный пример, демонстрирующий возможности предложенного описания иерархических оптимизационных процедур. На первом этапе поиска оптимизация происходит с использованием пчелиного алгоритма (ПА), далее полученные квазиоптимальные решения оптимизируются с помощью генетического алгоритма (ГА). Введем функцию генерации начального закодированного решения initialSolution() и функцию расчета целевой функции – calculate(). Необходимыми параметрами работы пчелиного алгоритма являются количество итераций – iteration = 100, количество блоков разбиения block = 10 и количество пчел-фуражиров – bees = 500. Параметрами работы генетического алгоритма являются так же количество итераций – iteration = 50, размер популяции – population = 100. Приведенные параметры и их значения приведены в качестве примера. Формализуем алгоритмы в соответствии с предложенным интерфейсом АПП и введенными сущностями XML. Пчелиный алгоритм примет вид:

<search id=”beeAlgorithm” solution=”initialSolution()” objectiveFunction=”calculate()” attributesList=”iteration=100,block=10,bees=500” />.

Описание генетического алгоритма примет следующий вид:

<search id=”geneticAlgorith” solution=”result(beeAlgorith)” objectiveFunction=”calculate()” attributesList=”iteration=50,population=100”/>,

где result(beeAlgorithm) – результат, полученный с помощью пчелиного алгоритма. Тогда XML-описание предложенного алгоритма, представленное в табл. 1, примет вид, приведенный ниже и на рис. 3.

В данном случае сущностью, содержащей атрибут parent="InitialPoint" является сущность

<flowid="mainFlow" parent="InitialPoint" child = "beeAlgorithmFlow">

(строка 3, табл. 1), где mainFlow – идентификатор глобального поискового процесса.

Данная сущность запускает процесс с идентификатором beeAlgorithmFlow, о чем говорит атрибут child="beeAlgorithmFlow". В рамках данного процесса запускается непосредственно АПП beeAlgorithm с описанными ранее параметрами. После окончания работы АПП beeAlgorithm запускается работа генетическо-

го алгоритма <subflow id = "geneticAlgorithmFlow" child = "geneticAlgorithm" parent = "beeAlgorithmFlow"> (строка 7, табл. 1), так как соответствующий процесс имеет атрибут parent ="beeAlgorithmFlow".

Таблица 1

Пример XML-описания

1	*<algorithm>*
2	*<start id="InitialPoint"/>*
3	*<flow id="* *mainFlow" parent="InitialPoint" child="beeAlgorithmFlow">*
4	*<subflow id="beeAlgorithmFlow" child="beeAlgorithm">*
5	*<search id="beeAlgorithm" solution="initialSolution()"* *objectiveFunction="calculate()"* *attributesList="iteration=100,block=10,bees=500"/>*
6	*</subflow>*
7	*<subflow id="geneticAlgorithmFlow" child="geneticAlgorithm" par-* *ent="beeAlgorithmFlow">*
8	*<search id="geneticAlgorith" solution="result(beeAlgorith)"* *objectiveFunction="calculate()" attributesList="iteration=50,population=100"/>*
9	*</subflow>*
10	*</flow>*
11	*<end id="finishPoint" parent="mainFlow"/>*
12	*</algorithm>*

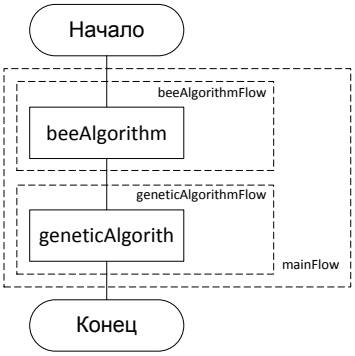

Рис. 2. Представление XML-описания в виде блок схемы

По окончании работы генетического алгоритма заканчивается работа и всего flow с идентификатором mainFlow в целом. Данное событие вызывает завершение работы алгоритма, <end id="finishPoint" parent="mainFlow"/> (строка 11, табл. 1), так как метка end id="finishPoint" ссылается на родительскую сущность mainFlow (parent="mainFlow").

3. Автоматизация процесса генерации стандартизированного представления алгоритмов оптимизации. В общем случае процесс автоматизации процесса генерации стандартизированного представления алгоритма оптимизации (СПАО) состоит из следующих этапов:

1. Определение множества АПП(a_i). На данном этапе ЛПР определяет множество всех атомарных поисковых процедур, которые могут быть включены в конечный алгоритм оптимизации.

2. Определение функции оценки результирующего алгоритма оптимизации $\Theta(\psi(F))$, где F – результирующий алгоритм оптимизации, ψ – функция оценки (критерий) с известным глобальным или квазиоптимальным значением.

$$\Theta = \psi_{best} - \psi_{result},$$

где ψ_{best} – известное глобальное или квазиоптимальное решение, ψ_{result} – лучше решение, полученное разрабатываемым генетическим алгоритмом. Очевидно, что для задачи минимизации при $\Theta < 0$ результирующий алгоритм (СПАО) является не эффективным, по сравнению с тем, с помощью которого было получено значение ψ_{best}. В качестве оценок для сравнения целесообразно применять бэнчмарки.

3. Ввод ограничений $C(c_i)$. К ограничениям можно отнести временные ограничения, а также ограничения, специфичные для решаемой задачи.

4. Вид решения VIEW, с которым будет работать алгоритм оптимизации. Другими словами, необходимо определить структуру данных, которая будет подаваться на вход алгоритма и приниматься на выходе.

В данной статье для упрощения процесса генерации стандартизированных описаний алгоритма предлагается использовать следующие допущения:

1. В качестве функции оценки $\psi(F)$ использовать скалярные математические функции, например функцию Экли, Растригина, Швефеля и др. Данные функции широко применяются в международном научном сообществе для оценки качества получаемых оптимизационных механизмов.

2. В качестве поискового механизма для получения новых алгоритмов оптимизации применять генетический алгоритм с модифицированными генетическими операторами. Выбор генетического алгоритма обусловлен тем, что стандартизированное описание алгоритмов оптимизации легко записать в виде хромосомы; количество АПП в искомом алгоритме не превышает 10–15 элементов, что делает вероятность нахождения глобального оптимума высокой; временная сложность генетических алгоритмов невысока для малого числа входных данных; в результате работы алгоритма получается множество альтернативных решений, что дает возможность ЛПР выбрать лучший, с его точки зрения, алгоритм.

3. Вид решений VIEW будет представлен в виде числовой, бинарной или векторной хромосомы (в зависимости от количества аргументов функции $\psi(F)$.

Для обеспечения работоспособности модуля генерации новых решений автором построены новые механизмы кодирования и декодирования СПАО. Они позволяют представлять стандартные описания в виде альтернативных решений (хромосом). Разрабатываемая структура кодирования или декодирования должна соответствовать следующим требованиям [7–9]:

- скорость кодирования и последующего декодирования. Временная сложность алгоритмов кодирования / декодирования не должна превышать nlog (n). Данный факт обусловлен тем, что генетические алгоритмы работают с большим набором альтернативных решений, каждое из которых должно быть закодировано и декодировано как минимум один раз целью его оценки;
- целостность данных. Алгоритмы кодирования и декодирования должны быть реализованы таким образом, чтобы после последовательного применения методов кодирования и декодирования исходное решение было идентично результирующему;
- простота кодированного представления. СПАО должно быть закодировано в таком виде, чтобы легко было автоматически вносить случайные изменения (оператор мутации) и объединять два СПАО в один (оператор скрещивания).

Для того чтобы создать эффективный механизм кодирования и декодирования СПАО предлагается использовать элементы теории генетического программирования (ГП) [10–12]. Традиционно, любой алгоритм легко представить в виде

дерева. В древовидном кодировании каждый узел дерева содержит функцию, а каждый лист – операнд. Выражение, представленное в виде дерева, может быть легко рекурсивно посчитано [13, 14].

Применительно к решаемой задаче узлы дерева интерпретируются как связи между АПП, а листья – как сами АПП. Для уменьшения количества кодируемой информации и упрощения работы генетических операторов автором предлагается использовать двухуровневый механизм кодирования [13,14]. На первом уровне выполняется процедура упрощения, на второй кодирование упрощенного СПАО.

Процедура упрощения заключается в следующем. Стандарт XML-описания обязывает применять дополнительные сущности для структурирования СПАО. Такие структуры, как algorithm, start, end, flow и subflow, обязательны для XML документа, но имеют избыточность при работе алгоритма генерации СПАО. Поэтому в данной работе предлагается удалить избыточные данные при кодировании. Рассмотрим пример, приведенный в табл. 1. После упрощения описание примет вид представленный в табл. 2.

Таблица 2

Пример XML-описания

1.	*<search id="beeAlgorithm" solution="initialSolution()" objectiveFunction="calculate()" attributesList="iteration=100,block=10,bees=500"/>*
2.	*<search id="geneticAlgorith" solution="result(beeAlgorith)" objectiveFunction="calculate()" attributesList="iteration=50,population=100"/>*

В соответствии с данными табл. 2 представление СПАО в виде дерева примет вид, представленный на рис. 3.

На рис. 3 вершина S указывает на отношения между алгоритмами. Введем два типа отношений S – последовательный и P – параллельный. Также отметим, что отношения не являются коммутативными, из чего следует, что порядок операндов имеет значение. Касательно примера, изображенного на рис. 3, справедливо утверждение, что результат работы beeAlgoritm является исходными данными для geneticAlgorithm.

В качестве структуры данных в хромосоме предлагается использовать обратную польскую нотацию (ОПН). ОПН — форма записи выражений, в которой операнды расположены перед знаками операторов. Под операндами автор понимает АПП, а под операторами – отношения между АПП [14].

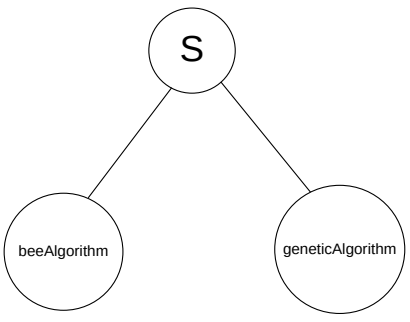

Рис. 3. Представление упрощенного СПАО в виде дерева

4. Разработка генетического алгоритма для генерации иерархических поисковых процедур. Для решения задачи автоматизированной генерации иерархического алгоритма оптимизации предлагается использовать модифицированный генетический алгоритм [15–17]. В качестве входных данных используются уже разработанные иерархические структуры. Целью автоматизированной генерации является автоматическое создание новых многоуровневых поисковых механизмов путем скрещивания имеющихся результатов с последующей оценкой времени их работы и качества решений, благодаря запуску новых алгоритмов для нахождения глобального оптимума для скалярных функций [3, 5, 9].

На первом этапе ЛПР производит выбор начальных данных (поисковых механизмов) для последующей работы генетического алгоритма автоматизированной генерации иерархических поисковых процедур. Данные процедуры представлены в виде стандартизированного описания, предложенного выше, и могут быть легко кодированы и декодированы в виде, необходимом для работы генетического алгоритма. Выбор поисковых механизмов является важным шагом, поскольку некоторые свойства начальных данных будут переданы потомкам в результате работы генетических операторов [5, 7].

На следующем шаге система генерации создает список используемых АПП, производит поиск значений необходимых настроечных параметров в БДАПП. После ЛПР предлагается уточнить значения найденных параметров и ввести значения недостающих. К данным параметрам относятся вероятности использования генетических операторов, количество итераций и т.д. [4, 8].

При получении всей необходимой информации система генерации производит операцию расчета значений целевых функции (оценок) для каждого иерархического оптимизационного алгоритма с последующим кодированием. В соответствии с рассчитанными значениями ЦФ выполняются модифицированные операторы кроссинговера и мутации [9].

Для новых решений производится операция декодирования и расчета оценок. Данные оценки являются основными данными при работе оператора селекции на основе колеса рулетки [2, 19, 20]. Выбор данного типа селекции обусловлен тем фактом, что в конечную выборку могут попасть неэффективные решения. Данные неэффективные решения благоприятно сказываются на общем генотипе популяции, не позволяя алгоритму попасть в локальный оптимум. После этого процесс продолжается итерационно по достижению заданного количества итераций.

На последнем шаге производится выбор лучшего решения по результатам работы всего генетического поиска в целом. Лучшее решение декодируется и предоставляется ЛПР в виде стандартизированного описания алгоритма оптимизации. Результат может быть изменен ЛПР или принят в исходном виде.

5. Экспериментальные исследования. Серия экспериментальных исследований были проведены на основе алгоритма размещения фрагментов схемы СБИС при фиксированных значениях параметров гибридного поиска и методов, включенных в его состав. Размер популяции пчелиного и генетического алгоритмов $|Pi| = 50$. Для пчелиного алгоритма количество итераций $Ib = 50$, количество агентов $Ab = 500$. Для генетиченского алгоритма количество итераций $Ig = 100$, вероятность мутации $Pg = 0{,}2$.

Результаты определения оценок затрат процессорного времени и объема памяти вычислительной системы при выполнении алгоритма приведены на рис. 4 и 5.

Рис. 4. Время выполнения биоинспирированного алгоритма размещения

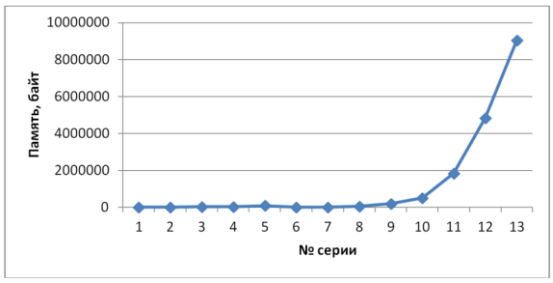

Рис. 5. Затраты памяти вычислительной системы

Временная сложность алгоритма составила $O(n^2)$, где n – число входных данных.

Основным преимуществом разработанного алгоритма является снижение вероятности попадания в локальный оптимум, а за счет распараллеливания уменьшается время.

Заключение. В данной работе описаны основные положения теории биоинспирированного поиска, а также выделены фундаментальные механизмы создания алгоритмов, основанных на моделировании биологических систем. Разработана новая подсистема генерации новых поисковых механизмов. Выделены основные компоненты подсистемы и разработана общая стандартизированная структура данных, основанная на XML-описании. Для эффективного управления информацией о гибридных поисковых механизмах разработана новая структура данных, основанная на XML описании. Представлены примеры стандартизированного описания. Для генерации новых иерархических поисковых процедур был разработан модифицированный генетический алгоритм, Авторами были интегрированы модифицированные алгоритмы кодирования и декодирования решений, а также модифицированные генетические операторы кроссинговера, позволяющие сократить время выполнения алгоритма в целом за счет генерации только валидных решений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 446 с.
2. *Бельков В.Н., Ланшаков В.Л.* Автоматизированное проектирование технических систем: учеб. пособие. – М.: Изд-во "Академия естествознания", 2009. – 143 с.
3. *Zaporozhets D., Zaruba D.V., Kureichik V.V.* Hybrid bionic algorithms for solving problems of parametric optimization // World Applied Sciences Journal. – 2013. – No. 23 (8). – P. 1032-1036.
4. *Курейчик В.В., Заруба Д.В., Запорожец Д.Ю.* Биоинспирированный алгоритм компоновки блоков ЭВА на основе модифицированной раскраски графа // Известия ЮФУ. Технические науки. – 2015. – № 4 (165). – С. 6-14.
5. *Клаг У., Каммингс М.* Основы генетики. – М.: Техносфера, 2007. – 896 с.

6. Спецификация языка описания схем XML (XML Schema Definition – XSD). http://www.w3.org/2001XMLShcema.
7. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы. – М.: Физматлит, 2010. – 368 с.
8. *Гладков Л.А., Курейчик В.В., Курейчик В.М., Сороколетов П.В.* Биоинспирированные методы в оптимизации. – М.: Физматлит, 2009. – 384 с.
9. *Курейчик В.М., Полковникова Н.А.* Многокритериальная оптимизация на основе эволюционных алгоритмов // Известия ЮФУ. Технические науки. – 2015. – № 2 (163). – С. 149-162.
10. *Курейчик В.М., Курейчик В.В.* Генетические алгоритмы в комбинаторно-логических задачах искусственного интеллекта // Известия ТРТУ. – 1999. – № 3 (13). – С. 126-128.
11. *Курейчик В.В., Курейчик В.М., Родзин С.И.* Теория эволюционных вычислений. Научное издание / под ред. В.М. Курейчика. – М.: Физматлит, 2012. – 260 с.
12. *Запорожец Д.Ю., Заруба Д.В., Лежебоков А.А.* Об одном способе кодирования решения для задачи размещения // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 183-188.
13. *Курейчик В.В., Заруба Д.В., Запорожец Д.Ю.* Иерархический подход при размещении компонентов СБИС // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 75-84.
14. *Zaporozhets D.U., Zaruba, D.V., Kureichik, V.V.* Representation of solutions in genetic VLSI placement algorithms, IEEE East-West Design & Test Symposium – (EWDTS'2014) Kiev, Ukraine, 2014. – P. 1-4.
15. *Курейчик В.В., Заруба Д.В., Запорожец Д.Ю.* Алгоритм параметрической оптимизации на основе модели поведения роя светлячков // Известия ЮФУ. Технические науки. – 2015. – № 6 (167). – С. 6-15.
16. *Kureichik, V.V., Kureichik, V.V. Jr., Zaruba, D.V.* Partitioning of ECE schemes components based on modified graph coloring algorithm. IEEE.
17. *Alpert C.J., Dinesh P.M., Sachin S.S.* Handbook of Algorithms for Physical design Automation, Auerbach Publications Taylor & Francis Group, USA, 2009. East-West Design & Test Symposium – (EWDTS'2014) Kiev, Ukraine, 2014. – P. 1-4.
18. *Запорожец Д.Ю., Кудаев А.Ю., Лежебоков А.А.* Многоуровневый алгоритм решения задачи параметрической оптимизации на основе биоинспирированных эвристик // Известия Кабардино-Балкарского научного центра РАН. – 2013. – № 4 (54). – С. 21-28.
19. *Rastrigin L.A.* Random Search in Evolutionary Computations // Proceedings 1st International conf., Evolutionary Computation and Its Application, EvCA' 96. – Moscow, 1996. – P. 135-143.
20. *Кулиев Э.В., Лежебоков А.А., Дуккардт А.Н.* Подход к исследованию окрестностей в роевых алгоритмах для решения оптимизационных задач // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 15-26.

REFERENCES

1. *Karpenko A.P.* Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennye prirodoy: ucheb. posobie [Modern algorithms of search engine optimization. Algorithms inspired by nature: a training manual]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2014, 446 p.
2. *Bel'kov V.N., Lanshakov V.L.* Avtomatizirovannoe proektirovanie tekhnicheskikh sistem: ucheb. posobie [Automated design of technical systems: textbook]. Moscow: Izd-vo "Akademiya estestvoznaniya", 2009, 143 p.
3. *Zaporozhets D., Zaruba D.V., Kureichik V.V.* Hybrid bionic algorithms for solving problems of parametric optimization, *World Applied Sciences Journal*, 2013, No. 23 (8), pp. 1032-1036.
4. *Kureychik V.V., Zaruba D.V., Zaporozhets D.Yu.* Bioinspirirovannyy algoritm komponovki blokov EVA na osnove modifitsirovannoy raskraski grafa [Bioinspired approach to partitioning of ece schemes components problem based on the modified graph coloring], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 4 (165), pp. 6-14.
5. *Klag U., Kammings M.* Osnovy genetiki [The basics of genetics]. Moscow: Tekhnosfera, 2007, 896 p.
6. Spetsifikatsiya yazyka opisaniya skhem XML (XML Schema Definition – XSD) [Specification description language XML schema (XML Schema Definition – XSD)]. Available at: http://www.w3.org/2001XMLShcema.
7. *Gladkov L.A., Kureychik V.V., Kureychik V.M.* Geneticheskie algoritmy [Genetic algorithms]. Moscow: Fizmatlit, 2010, 368 p.

8. *Gladkov L.A., Kureychik V.V., Kureychik V.M., Sorokoletov P.V.* Bioinspirirovannye metody v optimizatsii [Bioinspired methods in optimization]. Moscow: Fizmatlit, 2009, 384 p.
9. *Kureychik V.M., Polkovnikova N.A.* Mnogokriterial'naya optimizatsiya na osnove evolyutsionnykh algoritmov [Multiobjective optimization on the base of evolutionary algorithms], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Science], 2015, No. 2 (163), pp. 149-162.
10. *Kureychik V.M., Kureychik V.V.* Geneticheskie algoritmy v kombinatorno-logicheskikh zadachakh iskusstvennogo intellekta [Genetic algorithms in combinatorial-logical problems of the artificial intelligence], *Izvestiya TRTU* [Izvestiya TSURE], 1999, No. 3 (13), pp. 126-128.
11. *Kureychik V.V., Kureychik V.M., Rodzin S.I.* Teoriya evolyutsionnykh vychisleniy. Nauchnoe izdanie [The theory of evolutionary computing. Scientific publication], under ed. V.M. Kureychika. Moscow: Fizmatlit, 2012, 260 p.
12. *Zaporozhets D.Yu., Zaruba D.V., Lezhebokov A.A.* Ob odnom sposobe kodirovaniya resheniya dlya zadachi razmeshcheniya [A method of coding solutions for solving problems placement], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Science], 2012, No. 11 (136), pp. 183-188.
13. *Kureychik V.V., Zaruba D.V., Zaporozhets D.Yu.* Ierarkhicheskiy podkhod pri razmeshchenii komponentov SBIS [Hierarchical approach for VLSI components placement], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Science], 2014, No. 7 (156), pp. 75-84.
14. *Zaporozhets D.U., Zaruba, D.V., Kureichik, V.V.* Representation of solutions in genetic VLSI placement algorithms, IEEE East-West Design & Test Symposium – (EWDTS'2014) Kiev, Ukraine, 2014, pp. 1-4.
15. *Kureychik V.V., Zaruba D.V., Zaporozhets D.Yu.* Algoritm parametricheskoy optimizatsii na osnove modeli povedeniya roya svetlyachkov [Parametric optimization algorithm based on the model of glowworm swarm behavior], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Science], 2015, № 6 (167), pp. 6-15.
16. *Kureichik, V.V., Kureichik, V.V. Jr., Zaruba, D.V.* Partitioning of ECE schemes components based on modified graph coloring algorithm. IEEE.
17. *Alpert C.J., Dinesh P.M., Sachin S.S.* Handbook of Algorithms for Physical design Automation, Auerbach Publications Taylor & Francis Group, USA, 2009. East-West Design & Test Symposium – (EWDTS'2014) Kiev, Ukraine, 2014, pp. 1-4.
18. *Zaporozhets D.Yu., Kudaev A.Yu., Lezhebokov A.A.* Mnogourovnevyy algoritm resheniya zadachi parametricheskoy optimizatsii na osnove bioinspirirovannykh evristik [A multilevel algorithm for solving the problem of parametric optimization based on bio-inspired heuristics], *Izvestiya Kabardino-Balkarskogo nauchnogo tsentra RAN* [Izvestiya of Kabardino-Balkar scientific centre of the RAS], 2013, No. 4 (54), pp. 21-28.
19. *Rastrigin L.A.* Random Search in Evolutionary Computations, *Proceedings 1st International conf., Evolutionary Computation and Its Application, EvCA' 96*. Moscow, 1996, pp. 135-143.
20. *Kuliev E.V., Lezhebokov A.A., Dukkardt A.N.* Podkhod k issledovaniyu okrestnostey v roevykh algoritmakh dlya resheniya optimizatsionnykh zadach [Neural network technologies, fuzzy clustering and genetic algorithms in expert system], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Science], 2014, No. 7 (156), pp. 15-26.

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

Заруба Дарья Викторовна – Южный федеральный университет; e-mail: daria.zaruba@gmail.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; аспирант.

Запорожец Дмитрий Юрьевич – e-mail: elpilasgsm@gmail.com; кафедра систем автоматизированного проектирования; ассистент.

Zaruba Daria Viktorovna – Southern Federal University; e-mail: daria.zaruba@gmail.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; postgraduate student.

Zaporoghetz Dmitri Yurievich – e-mail: elpilasgsm@gmail.com; the department of computer aided design; teaching assistant.