

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.b.k@gmail.com; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: +79282897933; the department of computer aided design; professor.

Lebedev Oleg Borisovich – e-mail: lebedev.ob@mail.ru; phone: +79085135512; the department of computer aided design; associate professor.

Lebedeva Elena Mikhailovna – e-mail: lebedev.ob@mail.ru; phone: +79081702418; the department of computer aided design; graduate student.

УДК 004.896

Э.В. Кулиев, С.Н. Щеглов, Е.А. Пантелюк, Н.В. Кулиева

АДАПТИВНЫЙ АЛГОРИТМ СТАИ СЕРЫХ ВОЛКОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ ПРОЕКТИРОВАНИЯ*

Данная статья связана с решением одной из ключевых задач этапа автоматизированного конструкторского проектирования – размещения компонентов сверхбольших интегральных схем. В последнее время началось исследование возможностей применения и разработка алгоритмов, инспирированных природными системами, для эффективного принятия решения в задачах САПР. При этом постоянно возникает конфликт между сложностью САПР и требованиями принятия эффективных решений в реальном масштабе времени. Данные проблемы не могут быть полностью решены распараллеливанием процесса принятия решений, увеличением числа операторов, пользователей и т.д. Одним из возможных подходов к решению этой проблемы является использование новых технологий на стыке информатики, бионики и автоматизации проектирования. В этой связи разработка новых принципов и подходов принятия эффективных решений в задачах проектирования и управления имеет важное экономико-социальное значение и является, в настоящее время, актуальной и важной. В статье описывается алгоритм живой природы, который основывается на примере стаи серых волков. Приведена постановка задачи размещения элементов схем ЭВА на множестве заданных позиций дискретного рабочего поля. Представлена модифицированная технология разработки инспирированных природой алгоритмов. Показаны основные шаги работы алгоритма поведения стаи серых волков применительно к задаче размещения. Приведены сравнительные результаты вычислительных экспериментов. Основной целью исследования является оценка возможности применения интегрированных методов, инспирированных природными системами, для решения задач конструкторского проектирования САПР на примере использования алгоритма поведения стаи серых волков в живой природе.

Новой алгоритм; генетический алгоритм; целевая функция; окрестность; стая серых волков.

E. V. Kuliev, S. N. Sheglov, E. A. Pantelyuk, N. V. Kulieva

ADAPTIVE ALGORITHM OF THE PACK OF GREY WOLVES FOR SOLVING DESIGN OBJECTIVES

This article is related to the solution of one of the key tasks of the automated design stage – a placement of components of super-large integrated circuits. Recently, started have been a study of application possibilities and the development of algorithms inspired by natural systems for effective decision making in CAD tasks. At the same time, there is a constant conflict between the

* Работа выполнена при финансовой поддержке РФФИ (проект № 15–07–06415) и внутреннего гранта ЮФУ по теме №2.6432.2017/БЧ.

complexity of CAD and the requirements for making effective decisions in real time. These problems can not be completely solved by parallelizing the decision-making process, increasing the number of operators, users, etc. One of the possible approaches to solving this problem is the use of new technologies at the junction of informatics, bionics and design automation. In this regard, the development of new principles and approaches for making effective decisions in design and management tasks is of great economic and social importance and is, at present, relevant and important. The article describes the algorithm of living nature, which is based on the example of a pack of gray wolves. The formulation of the problem of placing elements of ECE schemes on the set of given positions of a discrete working field is given. A modified technology for the development of nature-inspired algorithms is presented. The main steps of the algorithm for the behavior of a pack of gray wolves in relation to the allocation problem are shown. Comparative results of computational experiments are presented. The main purpose of the study is to assess the feasibility of using integrated methods inspired by natural systems to solve CAD design problems using the example of the behavior of the gray wolf pack in wildlife.

Swarm algorithm; genetic algorithm; objective function; neighborhood; a flock of gray wolves.

Введение. Проектирование сложных объектов требует больших затрат как по времени, так и человеческих ресурсов, поэтому принято использовать различные средства автоматизированного проектирования. В данном случае ЭВМ является как инструментом проектирования, так и самим объектом. Автоматизация проектирования зависит от совершенства вычислительной техники и её элементной базы, то есть сверхбольшие и сверхскоростные интегральные схемы. Для автоматизированного проектирования потребуются следующие виды обеспечения [1–6, 8, 11]:

1. Математическое обеспечение – это переход от описания объекта к его математической модели и математическому проектированию, т.е. методам и алгоритмам решения этих задач.
2. Программное обеспечение – совокупность программных решений, которые реализуют операции и процедуры, которые необходимы для получения описания разрабатываемого объекта.
3. Информационное обеспечение – структурированные данные, которые используются для справочного и проектного материала.
4. Лингвистическое обеспечение – вся терминология, то есть языки программирования и описания объекта.

Одной из самых распространённых задач, связанных с конструкторским проектированием, является задача планирования СБИС. Задача планирования СБИС заключается в размещении на поле кристалла блоков, полученных на этапе разбиения, имеющих заданную площадь и не имеющих фиксированных размеров. При размещении решают сразу две задачи: размещение элементов и фиксируются размеры каждого блока. Задача размещения заключается в определении места на поле кристалла для каждого элемента и блока. При размещении идет учет таких характеристик, как: длина связи, распределение блоков по кристаллу, временные задержки, площадь кристалла его и размеры. Главная цель размещения – это создание наилучших условий для последующей трассировки. Для этого вводят критерии и оценки. Их оптимизация приводит к наилучшим условиям размещения.

Постановка задачи. Проблема размещения компонентов СБИС может быть сформулирована следующим образом. Дано множество элементов с расположенными на них контактами (выводами). Заданы элементарные цепи, связывающие контакты элементов. Задано коммутационное поле, на котором могут размещаться элементы. Необходимо разместить элементы на коммутационном поле с оптимизацией заданных критериев качества. Главной целью задачи размещения является минимизация общей площади кристалла, создание благоприятных условий для трассировки, а также минимизация общей суммарной длины соединений ($L(G) \rightarrow \min$)

[3, 7, 9, 10]. В общем виде задача размещения может быть сформулирована следующим образом: в монтажном пространстве задана область, которая разбивается на множество позиций, которых должно быть не меньше числа размещаемых элементов. Очевидно, что каждый элемент может занимать не более одного посадочного места, расстояние между которыми описывается симметричной матрицей расстояний. Имеющееся множество элементов, связанных между собой множеством электрических цепей необходимо таким образом отобразить на множестве P , чтобы обеспечивался экстремум целевой функции качества размещения. Задача размещения сводится теперь к отображению заданного графа схемы в решетку таким образом, чтобы вершины множества размещались в узлах решетки и, например, суммарная длина были наименьшими для возможных способов отождествления вершин графа и узлов решетки. При этом на исходную плоскость наносится, декартова система координат с осями s и t . В каждую ячейку плоскости может быть размещен элемент коммутационной схемы.

Алгоритм серых волков. В последнее время в САПР для задач размещения и проектирования применяются решения с использованием биоинспирированных методов. Биоинспирированные методы основываются на использовании алгоритмов, описывающих природные эволюционные процессы, а также к ним относят генетические алгоритмы. Данный метод позволяет получать оптимальные и квази-оптимальные решения за полиномиальное время.

В данной работе будет рассмотрен – алгоритм серых волков. Данный алгоритм был предложен в 2014 году. Он основывается на механизме охоты и иерархии серых волков в природе. Выделяют четыре типа серых волков: Альфа, Бета, Дельта и Омега, которые служат для моделирования иерархии лидерства. Также выделяют три основных этапа охоты: поиска добычи, окружение и нападение на жертву, реализуемые для оптимизации. Серые волки принадлежат семейству собачьих и стоят на вершине иерархии хищников, что значит, что они находятся на вершине пищевой цепи. Серые волки живут в основном стаями, причём размер стаи в среднем 5–12 особей. Особый интерес заключается в том, что у них очень строгая социальная иерархия доминирования.

Лидерами являются самцы и самки, называемые Альфа. Альфа принимает решения об охоте, ночлеге, времени подъёма стаи и так далее. Тем не менее, какое-то демократическое поведение также наблюдается, в котором Альфа следует за другими волками в стае. На собрании волков Альфу признаёт вся стая путём удерживания хвостов внизу. Альфа-волкам допускается спариваться только в стае. Интересно, что Альфа-это не обязательно самый сильный член стаи, но лучший в плане управления стаей. Это показывает, что организация и дисциплина в стае гораздо важнее, чем его сила.

На втором месте в иерархии стаи стоит Бета-волк. Бета-волки находятся в подчинении у Альфы и помогают им в принятии решений или другой деятельности стаи. Бета-волк может быть любого пола, и он является лучшим кандидатом, чтобы быть альфой в случае смерти или старости одного из Альфа-волков. Бета-волк должен уважать Альфа, и командовать другими волками более низкого уровня. Он играет роль советника Альфы и дисциплинирует стаю. Бета усиливает Альфу командами по всей стае и дает обратную связь Альфе [12, 15].

Самая низкая иерархическое положение у серого волка типа Омега. Омега-волки всегда должны подчиняться всем доминирующим волкам. Может показаться что данный тип волков не нужен, но было замечено, что все столкновения и проблемы внутри стаи возникают в случае их отсутствия. Они помогают с выплеском агрессии и насилия остальных волков на омегу, что позволяет сохранить структуру иерархии.

Также выделяют ещё один тип волков, который называется Дельта. Дельта-волки подчиняются Альфа и Бета волкам, но стоят выше Омега-волков. Разведчики, дозорные, старейшины, охотники и смотрители относятся к этой категории. Разведчики отвечают за просмотром границ и предупреждают стаю в случаи опасности. Дозорные защищают стаю. Старейшинами являются опытные волки (Альфа и Бета в прошлом). Охотники помогают Альфам и Бетам при охоте (рис. 1).

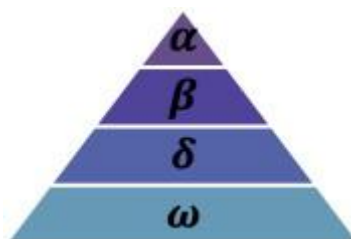


Рис. 1. Социальная иерархия серых волков

Для того, чтобы математически смоделировать социальную иерархию волков при проектировании, считается наиболее приспособленным решением Альфа (α). Следовательно, второе и третье лучшие решения называются бета (β) и Дельта (δ) соответственно. Остальные кандидаты решения считаются Омега (ω). В АСВ охотой (оптимизацией) руководствуется α , β и δ .

Изначально данный алгоритм в большей степени предназначен для решения векторных задач, поэтому необходимо адаптировать основные принципы его работы под решение задачи размещения. Для этого необходимо представить начальную популяцию в виде хромосом и осуществить их кодирование/декодирование в соответствии со следующим принципом: номер ген в хромосомах – это посадочные места, а значения в этих генах – номера элементов, которые будут на этих посадочных местах.

Как уже говорилось выше, серые волки окружают добычу во время охоты. Для того, чтобы математически смоделировать поведения окружающих волков, вводятся следующие уравнения [3, 13]:

$$D = |\vec{C} \times \vec{X}_p(t) - \vec{X}(t)|; \quad (1)$$

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} \times \vec{D}, \quad (2)$$

где t обозначает текущую итерацию, \vec{A} и \vec{C} являются векторы-коэффициенты, \vec{X}_p – позиция вектора «добыча», и \vec{X} показывает положение вектора «серый волк».

Векторы \vec{A} и \vec{C} рассчитываются следующим образом:

$$\vec{A} = 2\vec{a} \times \vec{r}_1 - \vec{a}; \quad (3)$$

$$\vec{C} = 2 \times \vec{r}_2, \quad (4)$$

где компонент \vec{a} линейно снижается с 2 до 0 в течение итераций и r_1, r_2 являются случайными векторами в $[0,1]$. В нашей задаче в роли параметра a выступает одноточечная мутация между всеми особями, уменьшение которой равномерно распределено на все итерации. Случайные векторы r_1, r_2 разрешают волкам добраться в любую позицию между двух конкретных точек, при этом пространство может быть с N-измерениями, что означает, что серые волки будут перемещаться в гипер-кубе вокруг наилучшего решения (рис. 2).

С учетом представленных выше уравнений 3 и 4, серый волк в положение (X, Y) может обновить свою позицию в зависимости от положения добычи (X^*, Y^*) . Различные места вокруг лучшего агента могут быть достигнуты по отношению к текущей позиции, регулируя значение векторов \vec{A} и \vec{C} . Например, (X^*-X, Y^*) может быть достигнуто путем установки $\vec{A} = (1,0)$ и $\vec{C} = (1,1)$.

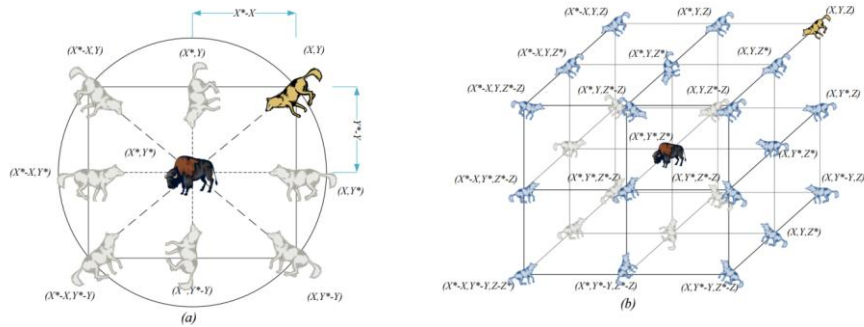


Рис. 2. Обновление позиции волков

Серый волк имеет способность распознавать местонахождение добычи и окружать ее. Охотой, как правило, занимаются Альфа. Бета и Дельта иногда также могут участвовать в охоте. Однако, в абстрактном пространстве поиска мы понятия не имеем, о месте нахождения оптимума (добычи). Для того, чтобы математически смоделировать охотничье поведение "серые волки", предположим, что Альфа (лучший кандидат решения) Бета и Дельта имеют более четкое представление о потенциальном месте добычи. Таким образом, мы сохраняем первые три лучших решения, полученных до этого момента и обязываем других поисковых агентов (в том числе Омега-волков) обновить свои позиции в зависимости от позиции лучших поисковых агентов [13, 15–19].

Серые волки заканчивают охоту, нападая на добычу, когда она перестает двигаться. Для того, чтобы математически смоделировать атаку добычи, мы уменьшаем значение \vec{a} . Обратите внимание, что диапазон колебания \vec{A} также снизился на \vec{a} . Другими словами \vec{A} это случайное значение в интервале $[-2a, 2a]$, где уменьшается от 2 до 0 в течение итераций. Когда случайные значения \vec{A} находятся в интервале $[-1, 1]$, следующая позиция поискового агента может находиться в любом положении между своей текущей позицией и позицией жертвы.

Серые волки в основном осуществляют поиск в соответствии с Положением Альфа, Бета и Дельта волков. Они расходятся друг от друга в поисках добычи и сходятся, чтобы атаковать жертву. Для того, чтобы математически показать расхождение, мы используем \vec{A} с случайные значения больше 1 или меньше -1 и обязываем поискового агента расходиться с добычей. Это делает разведку и позволяет АСВ осуществить глобальный поиск. $|A| > 1$ заставляет серых волков расходиться с добычей, надеясь найти след добычи. Еще одним компонентом АСВ, что способствует разведке, является \vec{C} , который содержит случайные значения в диапазоне $[0, 2]$. Этот компонент обеспечивает случайными весами добычу в целях стохастического усиления ($C > 1$) или ослабления ($C < 1$) влияния добычи в определении расстояния [4, 7, 20, 22].

Процесс поиска начинается с создания случайных популяций серых Волков (вариантов решения) в АСВ. В течение итераций, Альфа, Бета и Дельта Волки оценивают вероятную позицию жертвы. Каждый кандидат решения обновляет свою дистанцию до добычи. Параметр a уменьшается от 2 до 0 для того, чтобы подчеркнуть разведку и атаку, соответственно. Кандидаты решений расходятся с добычей, когда $|A| > 1$ и сходятся в направлении добычи, когда $|A| < 1$ (рис. 3). Алгоритм завершается достижением конечного критерия.

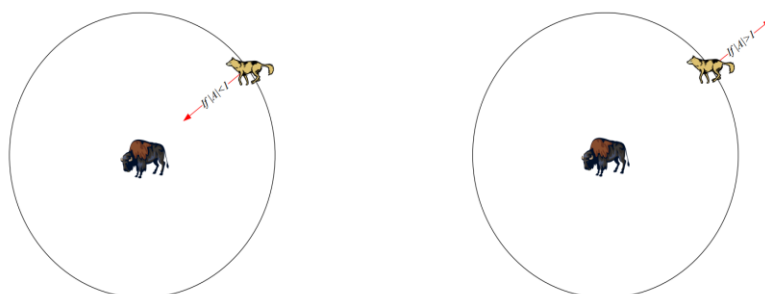


Рис. 3. Разведка

Алгоритм АСВ:

- 1) Инициализация начальной популяции хромосом X_i ($i = 1, 2, \dots, n$)
- 2) Инициализация параметра a . На данном этапе задается количество мутаций над хромосомами, что обеспечивает эффективное выполнение таких этапов работы алгоритма, как атака и разведка.
- 3) Вычисление приспособленности (целевой функции) каждого поискового агента.
- 4) $X_\alpha, X_\beta, X_\delta$ – наиболее перспективные поисковые агенты в данный момент времени. Предполагается, что $X_\alpha, X_\beta, X_\delta$ волки наиболее эффективным образом осуществляют поиск жертвы и ее окружения. Для этого необходимо отсортировать все хромосомы по возрастанию ЦФ.
 $X_\alpha, X_\beta, X_\delta$ будут являться первые три хромосомы.
- 5) Число итераций задается в исходных данных. По достижении этого параметра фиксируется значение X_α .
- 6) Происходит обновление позиций волков X_{t+1} вокруг жертвы относительно $X_\alpha, X_\beta, X_\delta$. Для этого необходимо выполнить упорядочивающий кроссинговер по следующему принципу: $X_\alpha * X_\beta \rightarrow X^*$; $X^* * X_\delta \rightarrow X_{t+1}$. Таким образом, X_{t+1} получит свойства $X_\alpha, X_\beta, X_\delta$.
- 7) Обновление параметра a , который уменьшается в ходе итераций, сокращая, таким образом, область поиска. Другими словами, параметра a , как уже упоминалось, имитирует атаку.
- 8) В каждой итерации необходимо выполнять вычисление ЦФ всех поисковых агентов для дальнейшей фиксации лучшего решения.
- 9) Обновление X_α, X_β и X_δ как наиболее перспективных решений.
- 10) Если критерий «Кол-во итераций» достигнут, то следует фиксация X_α , как самого лучшего решения.

Экспериментальные исследования. Экспериментальные исследования проводились на различных графах. Одна из основных задач при использовании биоинспирированных алгоритмов – это поиск оптимальных параметров, при которых алгоритм показывает наиболее эффективные решения.

Определим оптимальное значение параметра a для АСВ. Пусть размер популяции будет равно 100, а число итераций 100. Число размещаемых элементов равно 50. Результаты экспериментов приведены на рис. 4.

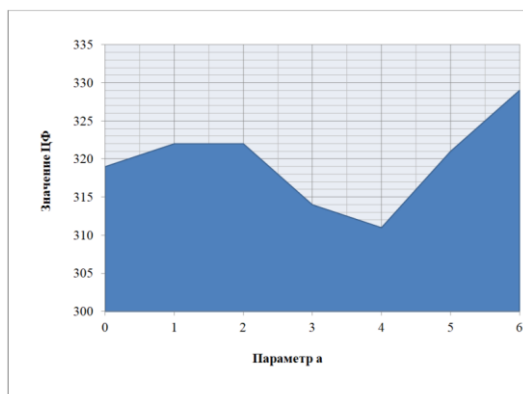


Рис. 4. Зависимость значения ЦФ от a

Как видно из рис. 4, наиболее оптимальным значением параметра a является 4.

Для определения ВСА алгоритма был проведён ряд экспериментов, в которых были получены результаты времени работы алгоритма в среднем случае, отражённые соответственно в табл. 1 и на рис. 5.

Таблица 1

Сравнение временной сложности алгоритмов

Алгоритм	Количество элементов								
	100000	200000	300000	400000	500000	600000	700000	800000	900000
Генетический (с)	11,05	21,57	33,78	54,01	72,15	94,01	125,6	169,5	185,7
Муравьиный (с)	21,4	52,12	71,82	110,1	180,5	205,9	298,8	359,6	404,9
Стаи волков (с)	35,58	84,45	157,56	248,78	351,7	409,9	558,9	694,8	857,7

График зависимости времени работы от количества элементов подтверждает сделанные предположения о временной сложности разработанного алгоритма основанного на поведении стаи волков, которая имеет квадратичный характер и может быть выражена формулой $O(\alpha \cdot n^2)$.

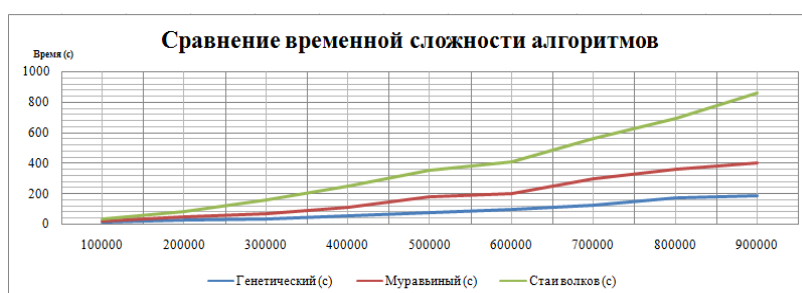


Рис. 5. График зависимости времени работы от количества элементов

Для определения эффективности разработанного алгоритма были проведены исследования качества решения на нескольких тестовых примерах. Под эффективностью алгоритма будем понимать качество решения, получаемого с его помощью (табл. 2). На рис. 6 приведена гистограмма сравнения качества решения, основываясь на данных из табл. 2.

Таблица 2

Сравнение эффективности алгоритмов

Алгоритм (мс)	Число элементов схемы				
	100000	250000	500000	750000	1000000
Генетический	87,7	128,4	142	171,4	215,7
Муравьиный	75,1	110,1	133	154,9	203,8
Стаи волков	66,5	98,3	101,7	139,1	177,4

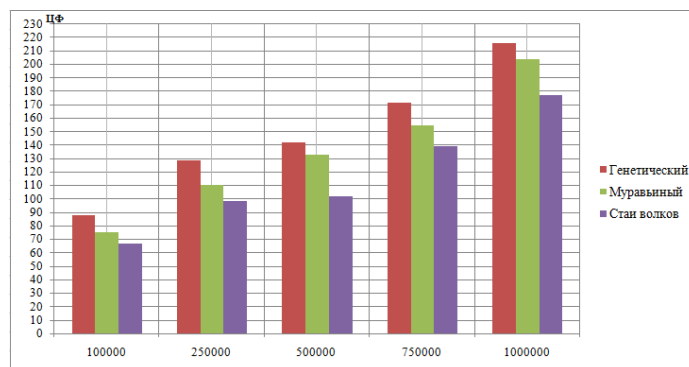


Рис. 6. Гистограмма сравнения качества решения

Из результатов эксперимента видно, что наиболее эффективным является разработанный алгоритм, основанный на поведении стаи волков. Он в среднем на 12 % эффективнее генетического алгоритма и на 8 % эффективнее муравьиного алгоритма.

Заключение. В статье рассмотрен алгоритм серых волков. Данный алгоритм в целом является достаточно эффективным, однако, как и любой другой популяционный алгоритм, он не является универсальным методом решения всех оптимизационных задач.

Выделяют четыре типа серых волков: Альфа, Бета, Дельта и Омега, которые служат для моделирования иерархии лидерства. Также выделяют три основных этапа охоты: поиска добычи, окружение и нападение на жертву, реализуемые для оптимизации. Серые волки принадлежат семейству собачьих и стоят на вершине иерархии хищников, что значит, что они находятся на вершине пищевой цепи. Серые волки живут в основном стаями, причём размер стаи в среднем 5–12 особей.

Старейшинами являются опытные волки (Альфа и Бета в прошлом). Охотники помогают Альфам и Бетам при охоте. Изначально данный алгоритм в большей степени предназначен для решения векторных задач, поэтому авторами адаптированы основные принципы его работы под решение задачи размещения. Для этого представлена начальная популяция в виде хромосом и осуществлено их кодирование/декодирование в соответствии со следующим принципом: номер ген в хромосомах – это посадочные места, а значения в этих генах – номера элементов, которые будут на этих посадочных местах.

В статье описано математическое представление поведения окружающих волков. Для того, чтобы математически смоделировать охотничье поведение "серые волки", предположено, что Альфа (лучший кандидат решения) Бета и Дельта имеют более четкое представление о потенциальном месте добычи. Таким образом, сохраняются первые три лучших решения, полученных до этого момента и обязывают других поисковых агентов (в том числе Омега-волков) обновить свои позиции в зависимости от позиции лучших поисковых агентов.

Основываясь на представленном алгоритме, была разработана программный продукт на языке программирования C++. Основной идеей экспериментальных исследований было нахождение набора параметров поставленной задачи, применение которых обеспечивает процесс нахождения квазиоптимальных решений за полиномиальное время. Экспериментальные исследования проводились на различных графах. Временная сложность разработанного алгоритма основанного на поведении стаи волков имеет квадратичный характер и может быть выражена формулой $O(n^2)$. Для сравнения эффективности разработанного алгоритма были проведены экспериментальные исследования на тестовых примерах (бенчмарках) муравьиного и генетического алгоритмов. Разработанный алгоритм, основанный на поведении стаи волков эффективнее в среднем на 12 % генетического алгоритма и на 8 % эффективнее муравьиного алгоритма.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Норенков И.П., Арутюнян Н.М.* Эволюционные методы в задачах выбора проектных решений // Электронный журнал «Наука и образование». – 2007. – № 9.
2. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им.Н.Э. Баумана, 2014. – 448 с.
3. *Ахмедова Ш.А.* Об эффективности «стаиногo» алгоритма оптимизации // Труды XLIII Краевой научной студенческой конференции по математике и компьютерным наукам. – Красноярск: СФУ, 2010. – С. 9-12.
4. *Курейчик В.В., Запорожец Д.Ю.* Современные проблемы при размещении элементов СБИС // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 68-73.
5. *Курейчик В.В., Бова В.В., Курейчик В.В.* Комбинированный поиск при проектировании // Образовательные ресурсы и технологии. – 2014. – № 2 (5). – С. 90-94.
6. *Zaporozhets D.U., Zaruba D.V., Kureichik V.V.* Representation of solutions in genetic VLSI placement algorithms – 2014 // Proceedings of IEEE East-West Design and Test Symposium, EWDTS 2014.
7. *Zaporozhets D.Yu., Zaruba D.V., Kureichik V.V.* Hybrid bionic algorithms for solving problems of parametric optimization // World Applied Sciences Journal. – 2013. – Vol. 23 (8). – P. 1032-1036.
8. *Kuliev E.V., Dukhardt 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 and Test Symposium, EWDTS 2014.
9. *Курейчик В.В., Запорожец Д.Ю.* Роевой алгоритм в задачах оптимизации // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 28-32.
10. *Bova V.V., Lezhebokov A.A., Gladkov L.A.* Problem-oriented algorithms of solutions search based on the methods of swarm intelligence // World Applied Sciences Journal. – 2013. – Vol. 27 (9). – P. 1201-1205.
11. *Zaruba D., Zaporozhets D., Kureichik V.* VLSI placement problem based on ant colony optimization algorithm // Advances in Intelligent Systems and Computing. – 2016. – Vol. 464. – P. 127-133.
12. *Kureichik V., Kureichik V., Bova V.* Placement of VLSI fragments based on a multilayered approach // Advances in Intelligent Systems and Computing. – 2016. – Vol. 464. – P. 181-190.
13. *Kureichik V.V., Zaruba D.V.* The bioinspired algorithm of electronic computing equipment schemes elements placement // Advances in Intelligent Systems and Computing. – 2015. – Vol. 347. – P. 51-58.
14. *Zaporozhets D., Zaruba D.V., Kureichik V.V.* Hierarchical approach for VLSI components placement // Advances in Intelligent Systems and Computing. – 2015. – Vol. 347. – P. 79-87.
15. *Zaporozhets D.U., Zaruba D.V., Kureichik V.V.* Representation of solutions in genetic VLSI placement algorithms // Proceedings of IEEE East-West Design and Test Symposium, EWDTS 2014.
16. *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.

17. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Решение задачи размещения на основе эволюционного моделирования // Известия РАН. Теория и системы управления. – 2007. – № 4. – С. 78-91.
18. Курейчик В.В., Курейчик Вл.Вл. Бионический поиск при проектировании и управлении // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 178-183.
19. Курейчик В.В., Курейчик В.М., Родзин С.И. Теория эволюционных вычислений. – М.: Физматлит, 2012. – 260 с.
20. Кулиев Э.В., Лежебоков А.А. О гибридном алгоритме размещения компонентов СБИС // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 188-192.
21. Кулиев Э.В., Лежебоков А.А. Исследование характеристик гибридного алгоритма размещения // Известия ЮФУ. Технические науки. – 2013. – № 3 (140). – С. 255- 261.

REFERENCES

1. Norenkov I.P. Arutyunyan N.M. Evolyutsionnye metody v zadachakh vybora proektnykh resheniy [Evolutionary techniques in the problems of design choices], *Elektronnyy zhurnal «Nauka i obrazovanie»* [Electronic magazine "Science and education"], 2007, No. 9.
2. Karpenko A.P. Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern algorithms of search engine optimization. Algorithms inspired by nature: a training manual]. Мпцсшцц: Izd-vo MGTU im.N.E. Bauman, 2014, 448 p.
3. Akhmedova Sh.A. Ob effektivnosti «staynogo» algoritma optimizatsii [The effectiveness of the "schooling" of the optimization algorithm], *Trudy XLIII Kraevoy nauchnoy studencheskoy konferentsii po matematike i komp'yuternym naukam* [Proceedings of XLIII the Regional student scientific conference on mathematics and computer science]. Krasnoyarsk: SFU, 2010, pp. 9-12.
4. Kureychik V.V., Zaporozhets D.Yu. Sovremennyye problemy pri razmeshchenii elementov SBIS [Modern placement's problems of VLSI], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2011, No. 7 (120), pp. 68-73.
5. Kureychik V.V., Bova V.V., Kureychik V.V. Kombinirovannyi poisk pri proektirovanii [Combined search in the design], *Obrazovatel'nye resursy i tekhnologii* [Educational resources and technology], 2014, No. 2 (5), pp. 90-94.
6. Zaporozhets D.U., Zaruba D.V., Kureichik V.V. Representation of solutions in genetic VLSI placement algorithms – 2014, *Proceedings of IEEE East-West Design and Test Symposium, EWDTS 2014*.
7. Zaporozhets D.Yu., Zaruba D.V., Kureichik V.V. Hybrid bionic algorithms for solving problems of parametric optimization, *World Applied Sciences Journal*, 2013, Vol. 23 (8), pp. 1032-1036.
8. 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 and Test Symposium, EWDTS 2014*.
9. Kureychik V.V., Zaporozhets D.Yu. Roeffvoy algoritm v zadachakh optimizatsii [Swarm algorithm in optimisation problems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 7 (108), pp. 28-32.
10. Bova V.V., Lezhebokov A.A., Gladkov L.A. Problem-oriented algorithms of solutions search based on the methods of swarm intelligence, *World Applied Sciences Journal*, 2013, Vol. 27 (9), pp. 1201-1205.
11. Zaruba D., Zaporozhets D., Kureichik V. VLSI placement problem based on ant colony optimization algorithm, *Advances in Intelligent Systems and Computing*, 2016, Vol. 464, pp. 127-133.
12. Kureichik V., Kureichik V., Bova V. Placement of VLSI fragments based on a multilayered approach, *Advances in Intelligent Systems and Computing*, 2016, Vol. 464, pp. 181-190.
13. Kureichik V.V., Zaruba D.V. The bioinspired algorithm of electronic computing equipment schemes elements placement, *Advances in Intelligent Systems and Computing*, 2015, Vol. 347, pp. 51-58.
14. Zaporozhets D., Zaruba D.V., Kureichik V.V. Hierarchical approach for VLSI components placement, *Advances in Intelligent Systems and Computing*, 2015, Vol. 347, p. 79-87.
15. Zaporozhets D.U., Zaruba D.V., Kureichik V.V. Representation of solutions in genetic VLSI placement algorithms, *Proceedings of IEEE East-West Design and Test Symposium, EWDTS 2014*.
16. Kuliev E.V., Lezhebokov A.A., Dukkardt A.N. Podkhod k issledovaniyu okrestnostey v roevykh algoritmakh dlya resheniya optimizatsionnykh zadach [Approach to research environs in swarms algorithm for solution of optimizing problems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 7 (156), pp. 15-26.

17. Kureychik V.M., Lebedev B.K., Lebedev O.B. Reshenie zadachi razmeshcheniya na osnove evolyutsionnogo modelirovaniya [The solution of the location problem based on evolutionary modeling], *Izvestiya RAN. Teoriya i sistemy upravleniya* [Journal of Computer and Systems Sciences International], 2007, No. 4, pp. 78-91.
18. Kureychik V.V., Kureychik V.I. Bionicheskiy poisk pri proektirovanii i upravlenii [Search inspired by natural systems, for the design and management], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 178-183.
19. Kureychik V.V., Kureychik V.M., Rodzin S.I. Teoriya evolyutsionnykh vychisleniy [The theory of evolutionary computing]. M.: Fizmatlit, 2012, 260 p.
20. Kuliev E.V., Lezhebokov A.A. O gibridnom algoritme razmeshcheniya komponentov SBIS [On the hybrid algorithm of component placement VLSI], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 188-192.
21. Kuliev E.V., Lezhebokov A.A. Issledovanie kharakteristik gibridnogo algoritma razmeshcheniya [Research parameters of hybrid algorithm for placement] *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 3 (140), pp. 255-261.

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

Кулиев Эльмар Валерьевич – Южный федеральный университет; e-mail: ekuliev@sfnu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; доцент

Щеглов Сергей Николаевич – e-mail: sch@sfnu.ru; кафедра систем автоматизированного проектирования; доцент.

Пантелюк Елизавета Александровна – e-mail: pantelyk@sfnu.ru; кафедра систем автоматизированного проектирования; магистр.

Кулиева Нина Владимировна – e-mail: xolopova@sfnu.ru; кафедра систем автоматизированного проектирования; аспирант.

Kuliev Elmar Valerievich – Southern Federal University; e-mail: ekuliev@sfnu.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; assistant professor.

Sheglov Sergey Nikolaevich – e-mail: sch@sfnu.ru; the department of computer aided design; assistant professor.

Pantelyuk Elizaveta Alexandrovna – e-mail: pantelyk@sfnu.ru; the department of computer aided design; master's degree.

Kulieva Nina Vladimirovna – e-mail: xolopova@sfnu.ru; the department of computer aided design; graduate student.

УДК 004.896

Б.К. Лебедев, О.Б. Лебедев, Е.О. Лебедева

МОДИФИЦИРОВАННЫЙ МУРАВЬИНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ МИНИМАЛЬНОГО ДЕРЕВА ШТЕЙНЕРА*

Предлагаются новые технологии, принципы и механизмы решения задачи построения дерева Штейнера, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. Задача построения дерева Штейнера представляется в виде адаптивной системы, на основе интеграции принципов самоорганизации и муравьиного подхода к поиску решения. Известная проблема Штейнера состоит в следующем. Дано множество P точек на плоскости. Формируется ортогональная сетка пу-

* Работа выполнена при финансовой поддержке РФФИ (проект № 15–01–05297).