

Полуян Анна Юрьевна

Старший преподаватель кафедры «Вычислительные системы и информационная безопасность».

Chernyshev Jury Olegovich

The Don State Technical University.

E-mail: orfiki@rambler.ru.

1, Strana Sovetov Street, Rostov-on-Don, 344023, Russia.

Phone: +78632589136.

Dr. of Eng. Sc., Professor the Department «Computing Systems and Information Security».

Poluyan Anna Urievna

Head Lecturer of Chairs «Computing Systems and Information Security».

УДК 519.712.2

Л.А. Гладков

ИНТЕГРИРОВАННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧ РАЗМЕЩЕНИЯ И ТРАССИРОВКИ НА ОСНОВЕ НЕЧЕТКИХ ГЕНЕТИЧЕСКИХ МЕТОДОВ*

Предлагается интегрированный подход к решению задач размещения и трассировки СБИС основанный на их совместном решении с использованием нечетких генетических методов. Приводится описание рассматриваемой задачи и краткий анализ существующих подходов к ее решению. Дается описание структуры предложенного алгоритма, а также его основных этапов. Приводится описание разработанных эвристик, операторов и стратегий поиска оптимальных решений. В заключении представлено краткое описание проведенных вычислительных экспериментов, подтверждающих эффективность предложенного метода.

Размещение; трассировка; эволюционные вычисления; нечеткие генетические алгоритмы; нечеткий логический контроллер.

L.A. Gladkov

THE INTEGRATED ALGORITHM OF THE DECISION OF PROBLEMS OF PLACEMENT AND ROUTING ON THE BASIS OF FUZZY GENETIC METHODS

In work integrated approach to the decision of problems of placement and routing VLSI based on their joint decision with use of fuzzy genetic methods is offered. The description of a considered problem and the short analysis of existing approaches is resulted in its decision. The description of structure of the offered algorithm, and also its basic stages is given. The description developed heuristics, operators and strategy of search of optimum decisions is resulted. In the conclusion the short description of the spent computing experiments confirming efficiency of the offered method is presented.

Placement; routing; evolutionary calculations; fuzzy genetic algorithms; the indistinct fuzzy controller.

При создании новой электронной аппаратуры огромное значение имеют методы автоматизированного проектирования, которые позволяют создавать высоконадежные сверх- и ультрабольшие интегральные схемы (СБИС) в короткие сроки и при сравнительно низких затратах. Тенденция к росту степени интеграции СБИС приводит к существенному увеличению трудоемкости их проектирования, что вызывается ростом размерности решаемых при проектировании задач.

* Работа выполнена при частичной поддержке РФФИ (проект № 11-01-00122).

Проектирование СБИС представляет собой многоуровневый процесс, в котором каждый уровень характеризуется своим математическим и программным обеспечением. Одним из важных этапов процесса проектирования СБИС является этап конструкторского проектирования. Задачи конструкторского проектирования, как правило, характеризуются большой вычислительной сложностью обусловленной необходимостью перебора огромного числа различных вариантов решений. Причем, для получения точного решения требуется выполнить полный перебор, что не представляется возможным [1].

В процессе проектирования, как правило, используют ориентировочные значения данных, в то время, как их истинные значения становятся известными только после выполнения последующих процедур. Это обуславливает итерационный характер процесса проектирования с возвратами от последующих этапов к предыдущим, что, естественно, существенно увеличивает затраты на проектирование. Поэтому продолжается поиск методов сокращения числа итераций в цикле проектирования СБИС [2].

На этапе конструкторского проектирования СБИС особую роль занимают задачи размещения и трассировки. Традиционно они решаются на разных этапах различными методами, что ведет к увеличению затрат временных и вычислительных ресурсов. Поэтому представляется целесообразной разработка интегрированных методов решения задач размещения и трассировки, позволяющих выполнять эти задачи в одном цикле с взаимным учетом имеющихся ограничений и текущих результатов.

Одним из направлений повышения эффективности решения задач автоматизированного проектирования сложных технических систем, в том числе, СБИС, являются эволюционные и генетические алгоритмы [3]. Они успешно применяются для решения задач структурной и параметрической оптимизации, возникающих в ходе проектирования сложных технических систем [4].

Новым этапом развития теории генетических алгоритмов стали гибридные системы, основанные на совмещении различных научных направлений. Существуют различные способы гибридизации. Один из них – нечеткие генетические алгоритмы, в которых методы нечеткой математики используются для настройки параметров ГА. Также используют нечеткие операторы и нечеткие правила для создания генетических операторов с различными свойствами; системы нечеткого логического контроля параметров ГА в соответствии принятыми критериями; нечеткие критерии остановки процесса генетического поиска. Математический аппарат теории нечетких систем используется в данном случае для кодирования, подбора оптимальных параметров генетических алгоритмов, значений вероятности генетических операторов, выбора функции пригодности и критерия останова, создания нечетких генетических операторов [5].

1. Анализ методов решения задач размещения и трассировки. Основная сложность в постановке задачи размещения заключается в выборе целевой функции. Связано это с тем, что одной из главных целей размещения является создание наилучших условий для дальнейшей трассировки соединений, что невозможно проверить без проведения самой трассировки. Следовательно, конечный результат может быть найден только при совместном решении задач размещения, выбора очередности проведения соединений и трассировки, что практически невозможно вследствие огромных затрат машинного времени.

Поэтому все применяемые в настоящее время алгоритмы размещения используют промежуточные критерии, которые лишь косвенно способствуют решению основной задачи: получению оптимальной трассировки соединений. К числу

критериев, которые могут использоваться для оценки качества размещения и трассировки, можно отнести следующие:

- 1) суммарная взвешенная длина соединений;
- 2) число соединений, длина которых больше заданной;
- 3) количество пересечений проводников;
- 4) число соединений между элементами, находящимися в соседних позициях либо в позициях, указанных разработчиком.

Наибольшее распространение в алгоритмах размещения получил первый критерий, что объясняется следующими причинами: уменьшение длины связей позволяет улучшить электрические характеристики схемы, упрощает процесс трассировки; кроме того, он сравнительно прост в реализации.

Трассировка соединений является, как правило, заключительным этапом конструкторского проектирования ЭВА и состоит в определении линий, соединяющих эквипотенциальные контакты элементов, составляющих проектируемое устройство [6].

В процессе трассировки современных сверх и ультрабольших интегральных схем принято выделять следующие два этапа: глобальная (назначение цепей в определенных регионах) и детальная (прокладка соединений внутри областей) трассировки. В свою очередь, детальная трассировка разделяется на трассировку коммутационных блоков и трассировку каналов [7].

Большинство программ глобальной трассировки основано на алгоритмах Ли или Дейкстры. Как правило, волновой алгоритм Ли позволяет находить минимально возможное соединение между двумя элементами.

2. Постановка задачи. Любая оптимизационная задача может быть описана кортежем длины следующего вида: $\langle X, D, Q \rangle$. При совместном решении задач размещения и трассировки данный кортеж можно интерпретировать следующим образом:

X – множество всех хромосом из всех популяций. Пусть P_t – некоторая популяция на шаге t , $P_t = \{h_1, h_2 \dots h_s\}$, где h_s – хромосома из этой популяции, $t = [1, N]$; $s = [1, M]$ (N – число популяций, M – число хромосом в популяции) Тогда множество всех решений задается выражением:

$$X = \{P_t; t = 1, 2, \dots, N\} = \{h_{st}; t = 1, 2, \dots, N; s = 1, 2, \dots, M\}$$

Целевая функция Q – представляет собой нормированный аддитивный критерий, который включает оценку количества непротрассированных соединений и суммарную длину цепей:

$$Q = k_1 Q_1 + k_2 Q_2,$$

где k_1, k_2 – весовые коэффициенты локальных критериев, с помощью которых учитывается важность того или иного критерия на качество решения; Q_1 – критерий оценки количества непротрассированных соединений; Q_2 – критерий оценки суммарной длины цепей.

Оптимизация задачи сводится к минимизации значения критерия Q , т.е. $Q(X) \rightarrow \min Q(h_{\text{opt}}) = \min Q(h_{ij})$, где $h_{ij} \subset S$.

3. Описание алгоритма. В результате проведенного анализа существующих методов и алгоритмов решения задач размещения и трассировки была предложена следующая процедура совместного решения задачи размещения и трассировки:

1. Ввод исходных данных (число элементов каждого типа, число цепей, число контактов и др.).
2. Задание начального размещения элементов.

3. Построение начального множества решений (популяции).
4. Задание начальных значений управляющих параметров генетического алгоритма.
5. Процедура улучшения качества размещения за счет использования нечетких генетических операторов:
 - a. выбор решений для выполнения операторов;
 - b. выбор стратегии поиска оптимальных решений;
 - c. применение генетических операторов;
 - d. отбор лучших решений;
 - e. формирование нового множества решений.
6. Выбор и применение заданного алгоритма трассировки.
7. Расчет полученного значения целевой функции.
8. Проверка соблюдения критерия остановки.
9. Работа нечеткого логического контроллера, изменение управляющих параметров алгоритма.
10. Возврат к шагу 5 для улучшения качества текущего размещения, либо завершение работы алгоритма.

После улучшения размещения элементов схемы в результате выполнения генетических процедур информация о размещении сохраняется в буфере. Затем проводится трассировка соединений с помощью одного из заданных алгоритмов трассировки (например, лучевого алгоритма), оценка качества полученного решения и, при необходимости, возврат на этап размещения.

Для улучшения качества размещения используется разработанный нечёткий генетический алгоритм с вещественным кодированием решений [8]. В этом случае хромосома представляет собой вектор вещественных чисел, соответствующих координатам (по осям x и y) посадочных позиций рабочего поля в которых размещаются элементы схемы, например:

$$\begin{aligned}x|0.63222|0.16464|0.9325.... \\y|0.13334|0.46561|0.7568....\end{aligned}$$

Длина хромосомы совпадает с длиной вектора-решения оптимизационной задачи, каждому гену соответствует один элемент.

Выбор позиции для размещения очередного элемента производится таким образом, чтобы приращение фактической длины было минимально.

С целью повышения эффективности поиска предложены две модифицированные модели поиска и формирования новой популяции.

Модель формирования популяции на основе *минимального разрыва между поколениями* (ММП). При выполнении оператора кроссинговера пары родителей выбираются по принципу одно решение «элитное», одно случайно. Полученные после выполнения кроссинговера решения и исходные решения образуют подпопуляцию, из которой выбираются лучшие индивидуумы. В процессе случайного выбора может использоваться метод «колеса рулетки». В данном случае оператор мутации не применяется.

Модель формирования популяции на основе *обобщения поколений* (ОП). Из исходной популяции выбирается заданное множество решений. Эти решения образуют подпопуляцию к которой применяются генетические операторы. Модель ОП сохраняет полученные решения с предыдущей итерации.

Схема процедуры создания новой популяции представлена на рис. 2.

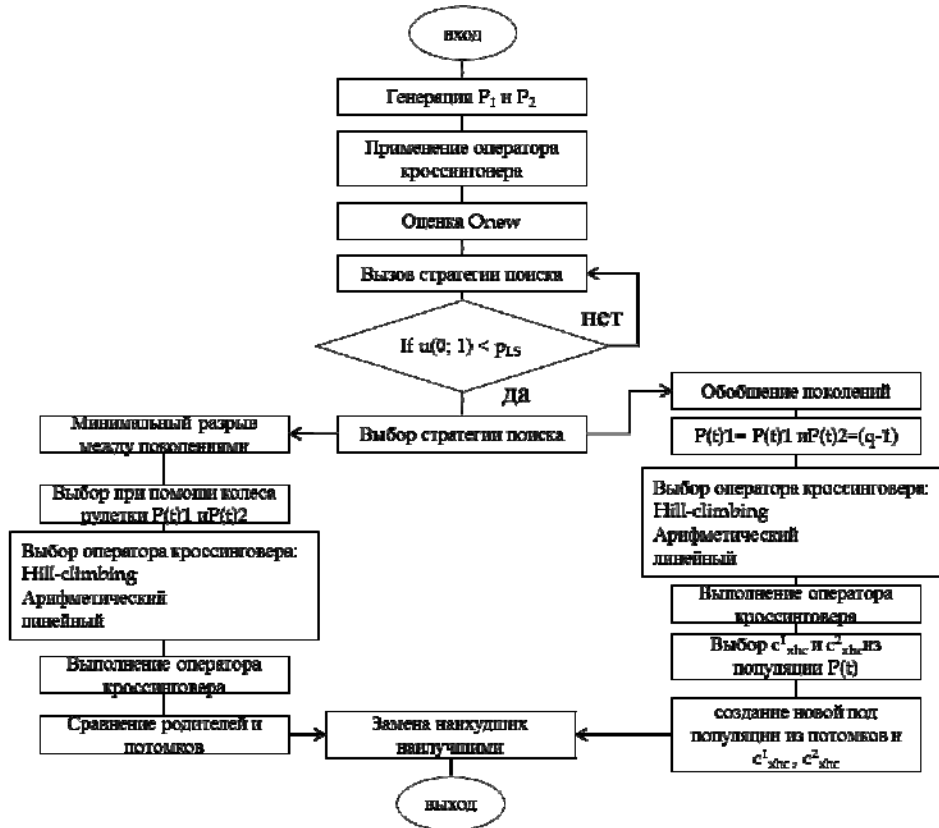


Рис. 2. Обобщенная схема работы процедуры создания новой популяции

Важную роль в выполнении нечеткого генетического алгоритма играют модифицированные генетические операторы [9]. Основным механизмом получения новых решений является оператор кроссинговера. Алгоритм использует модифицированный оператор «hill-climbing» («восхождения на холм») на основе методов градиентных локального поиска [10].

Предлагается модифицированный оператор кроссинговера – оператор поиска экстремума «hill-climbing».

Пусть $X = (x_1, \dots, x_n)$ и $Y = (y_1, \dots, y_n)$ – хромосомы с вещественным кодированием, выбранные на основе некой стратегии для выполнения оператора кроссинговера. В результате получаем потомков:

$$Z_1 = (z_1^1 \dots z_n^1) \text{ или } Z_2 = (z_1^2 \dots z_n^2),$$

где z_i^1 – произвольным образом выбранное число на интервале $[l_i^1, u_i^1]$, причём,

$$l_i^1 = \max\{a_i, x_i - I^* \alpha\} \text{ и } u_i^1 = \min\{b_i, x_i + I^* \alpha\},$$

а z_i^2 выбирается на интервале $[l_i^2, u_i^2]$ и

$$l_i^2 = \max\{a_i, y_i - I^* \alpha\} \text{ и } u_i^2 = \min\{b_i, y_i + I^* \alpha\},$$

где $I = [x_i - y_i]$.

Основными преимуществами оператора являются учет разнообразия популяции решений и степени «близости» вновь полученных решений к родительским.

Также в процессе выполнения алгоритма могут использоваться еще два модифицированных оператора кроссинговера: арифметический и линейный [11].

Пусть $C1=(c_1^1, c_2^1, \dots, c_n^1)$ и $C2=(c_1^2, c_2^2, \dots, c_n^2)$ – две хромосомы, выбранные для выполнения генетического оператора. При этом предположим, что выполняются условия $c_k^1 \leq c_k^2$ и $f(C_1) \geq f(C_2)$.

В результате применения *арифметического оператора кроссингвера* создаются три потомка:

$H_1=(h_1^1, \dots, h_n^1)$, $H_2=(h_1^2, \dots, h_n^2)$, $H_3=(h_1^3, \dots, h_n^3)$, где $h_k^1=wc_k^1+(1-w)c_k^1$, $h_k^2=wc_k^2+(1-w)c_k^2$, $h_k^3=wc_k^3+(1-w)c_k^3$, $k=1, \dots, n$, w – константа, заданная на интервале $[0; 1]$.

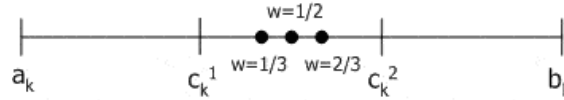


Рис. 3. Графическое представление арифметического кроссингвера

После выполнения *линейного кроссингвера* также создаются три потомка $H_q = (h_1^q, \dots, h_k^q, \dots, h_n^q)$, $q = 1, 2, 3$, где $h_k^1 = 0.5c_k^1 + 0.5c_k^2$, $h_k^2 = 1.5c_k^1 - 0.5c_k^2$, $h_k^3 = -0.5c_k^1 + 1.5c_k^2$. Графически результат выполнения линейного кроссингвера можно представить следующим образом (рис. 4).

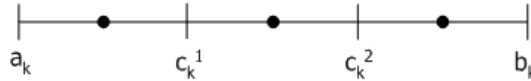


Рис. 4. Пример построения решения при арифметическом кроссингвере

Также предложен модифицированный оператор *неоднородной мутации*. Механизм выполнения данного оператора можно представить следующим образом.

При изменении значения гена y_i новое значение y'_i случайным образом генерируется на отрезке $[min_i, max_i]$,

$$y'_i = \begin{cases} y_i + (max_i - y_i)(1 - r(1 - \frac{1}{T})^t), & \text{если } q = 0 \\ y_i - (y_i - min_i)(1 - r(1 - \frac{1}{T})^t), & \text{если } q = 1 \end{cases}$$

где q случайным образом принимает значения 0 или 1;

r – случайное число, принимающее значение в диапазоне $[0; 1]$;

t – номер поколения;

T – максимальное число поколений;

b – параметр, обусловленный условиями задачи;

min_i и max_i – верхняя и нижняя границы для величины y_i .

После применения генетических операторов рассчитывается значение целевой функции. Модификация размещения элементов происходит до тех пор, пока не будут выполнены заданные критерии останова.

Для ускорения процесса поиска в алгоритме используется нечеткий логический контроллер (НЛК). Используя набор хромосом и их целевые функции, мы можем задать некоторый диапазон значений целевой функции, который нам необходим. Используя лингвистические переменные, задаем изменение управляющих параметров алгоритма (рис. 5).

На основе предложенных алгоритмов, методов и процедур был разработан программный комплекс для интегрированного решения задач размещения и трассировки. Поиск оптимального решения проводится по двум вариантам стратегии поиска, предусмотрена возможность распараллеливания серий выполняемых вычислительных экспериментов, причем для каждого варианта можно выбрать свои параметры. Результаты проведения серий экспериментов отображаются на экране (рис. 6).

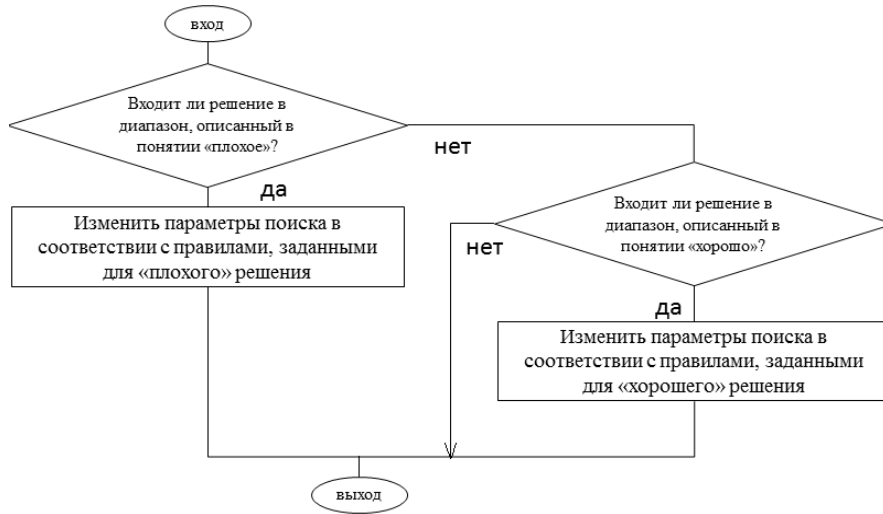


Рис. 5. Обобщенная схема алгоритма работы нечеткого контроллера

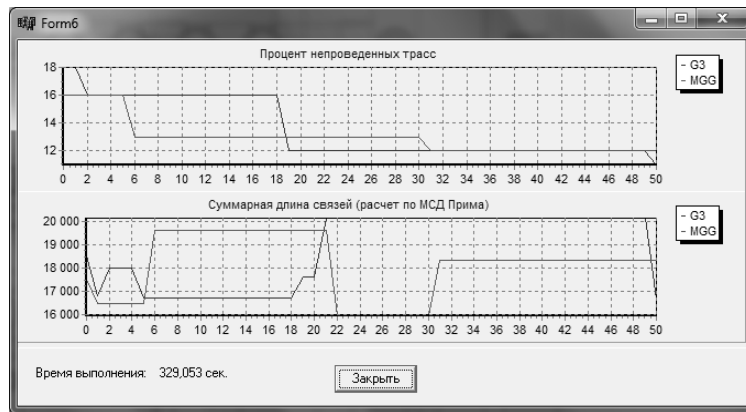


Рис. 6. Окно отображения результатов экспериментов

По завершению работы программы, результаты отображаются в виде графиков (рис. 7).



Рис. 7. Окно отображения результатов работы программы

При сравнении результатов работы программы с использованием НЛК и без него, очевидно, что в первом случае целевая функция сходится быстрее (рис. 8).

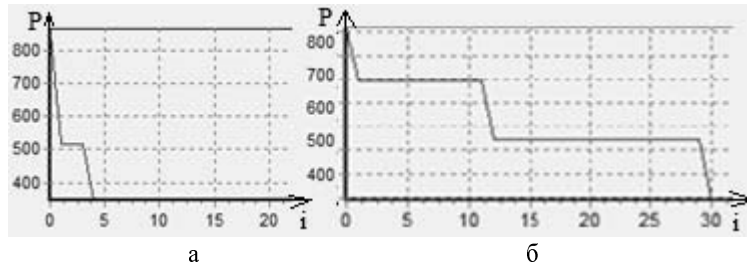


Рис. 8. Результаты работы программы с использованием НЛК а) и без него б)

При анализе результатов работы программы также проводилось сравнение полученных результатов с существующими тестовыми задачами (бенчмарками). Полученные результаты позволяют утверждать, что предложенный метод не уступает в быстродействии существующим образцам, но при этом позволяет выполнять совместное решение задач размещения и трассировки. В то же время, необходимо продолжение исследований в данном направлении.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Норенков И.П.* Основы автоматизированного проектирования. – М.: Изд-во МГТУ им. Баумана, 2006.
2. *Казеннов Г.Г.* Основы проектирования интегральных схем и систем. – М.: БИНОМ. Лаборатория знаний, 2005.
3. *Cohon J.P., Karro J., Lienig J.* Evolutionary Algorithms for the Physical Design of VLSI Circuits. Advances in Evolutionary Computing: Theory and Applications, Ghosh, A., Tsutsui, S. (eds.) Springer Verlag, London, 2003. – P. 683-712.
4. *Курейчик В.М., Курейчик В.В., Родзин С.И.* Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-25.
5. *Гладков Л.А., Гладкова Н.В.* Новые подходы к построению систем анализа и извлечения знаний на основе гибридных методов // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 146-154.
6. *Курейчик В.М.* Математическое обеспечение конструкторского и технологического проектирования с применением САПР. – М.: Радио и Связь, 1990.
7. *Shervani N.* Algorithms for VLSI physical design automation. – USA, Kluwer Academy Publisher, 1995. – 538 p.
8. *Deb K., Joshi D., Anand A.* Real-Coded Evolutionary Algorithms with Parent-Centric Recombination. Kanpur Genetic Algorithms Laboratory (KanGAL), Kanpur, PIN 208 016, India. KanGAL Report No. 2001003.
9. *Курейчик В.М.* Модифицированные генетические операторы // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 7-15.
10. *Herrera F., Lozano M.* Fuzzy Adaptive Genetic Algorithms: design, taxonomy, and future directions // Soft Computing 7(2003), Springer-Verlag, 2003. – P. 545-562.
11. *Гладков Л.А., Гладкова Н.В.* Особенности использования нечетких генетических алгоритмов для решения задач оптимизации и управления // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 130-136.

Статью рекомендовал к опубликованию д.т.н., профессор Н.И. Витиска.

Гладков Леонид Анатольевич

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: leo@tsure.ru.

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 88634371625.

Кафедра системного анализа и телекоммуникаций; доцент.

Gladkov Leonid Anatol'evich

Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: leo@tsure.ru.

44, Nekrasovsky, Taganrog, 347928, Russia.

Phone: +78634371743.

The Department of System Analysis and Telecommunications; Associate Professor.

УДК 681.3

В.М. Курейчик, А.А. Кажаров

**ИСПОЛЬЗОВАНИЕ РОЕВОГО ИНТЕЛЛЕКТА В РЕШЕНИИ
NP-ТРУДНЫХ ЗАДАЧ***

Роевой интеллект описывает коллективное поведение децентрализованной самоорганизующейся системы. Рассматривается в теории искусственного интеллекта как метод оптимизации. Термин был введен Херардо Бени и Ван Цзином в 1989 году. Системы роевого интеллекта, как правило, состоят из множества агентов локально взаимодействующих между собой и с окружающей средой. Сами агенты обычно довольно просты, но все вместе, локально взаимодействуя, создают, так называемый, роевой интеллект. Статья посвящена исследованию алгоритмов роевого интеллекта для решения NP-полных задач. Основная идея этих алгоритмов – моделирование поведения роя. В ходе данной работы были исследованы муравьиный, пчелиный и метод роя частиц. С использованием этих алгоритмов решены следующие задачи: задача коммивояжера, маршрутизации транспорта, разбиения графа, размещение элементов СБИС. Данные задачи без каких-либо изменений в ее интерпретации решаются для проектирования СБИС, логистических расчетов. В ходе проделанной работы была разработана программа на ЭВМ, реализующая описанные алгоритмы. Полученные результаты позволяют судить об оптимальном выборе параметров алгоритмов. Экспериментальные исследования доказали эффективность роевых алгоритмов по сравнению со стандартными итерационными, эвристическими и генетическими алгоритмами.

Роевой интеллект; муравьиные алгоритмы; МА; пчелиные алгоритмы; роевой алгоритм; задача о коммивояжере; ЗК; NP-задача; СБИС; генетические алгоритмы.

V.M. Kureichik, A.A. Kazharov

SWARM INTELLIGENCE USING FOR NP-TASKS SOLVING

Swarm intelligence (SI) is the collective behaviour of decentralized, self-organized systems, natural or artificial. The concept is employed in work on artificial intelligence. The expression was introduced by Gerardo Beni and Jing Wang in 1989. SI systems are typically made up of a population of simple agents or boids interacting locally with one another and with their environment. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions

* Работа выполнена при частичной поддержке РФФИ (проекты: № 10-01-90017, № 09-01-00492).