

**Begliyrov Vadim Valeryevich**

South Russian State University of Economics and Service.

E-mail: TorBWW@yandex.ru.

147, Shevchenko street, Shakhty, 346500, Russia.

Phone: +79081760312

**Bereza Andrew Nikolayevich**

E-mail: anbirch@mail.ru.

Phone: +789281574449.

**Storogenko Andrew Sergeevich**

Volgodonsk Institute of Services.

E-mail: sto-a@yandex.ru.

6, Chernikova street, Volgodonsk, 347375, Russia.

Phone: +79281116807

УДК 621.382.82 (076.5)

**А.В. Ковалев, С.А. Бушин**

**ЭВОЛЮЦИОННЫЙ МЕТОД РАЗМЕЩЕНИЯ РАЗНОГАБАРИТНЫХ БЛОКОВ СБИС**

*Целью данной работы является разработка метода размещения разногабаритных топологических блоков сверхбольших интегральных схем (СБИС), а также проведение оценки эффективности применения отдельных функций (операторов) данного метода. Метод размещения, основанный на принципах моделирования генетической эволюции, при сравнительном анализе на стандартных тестовых задачах показал высокую эффективность. Разработанный метод размещения может быть использован в системах автоматизированного проектирования (САПР) как СБИС, так и печатных плат.*

*Метод размещения блоков; автоматизированное проектирование; СБИС; генетический алгоритм; эволюционное моделирование.*

**A.V. Kovalev, S.A. Bushin**

**THE EVOLUTIONARY METHOD FOR A PLACEMENT OF NON-SLICING BLOCKS IN VLSI**

*The development of the method for non-slicing blocks placement in very large scale integrated circuits (VLSI) is described in this paper. The method for placement based on genetic evolution simulation. The effectiveness analysis showed a high result.*

*The method for non-slicing block placement can be used in CAD systems for VLSI or PCB design.*

*Placement method; computer-aided design; VLSI; genetic algorithm; evolutionary simulation.*

**Введение.** Современные системы-на-кристалле содержат большой набор сложных функциональных (СФ) блоков (IP-ядер), топологические образы которых имеют уникальные габаритные размеры. Такие СФ-блоки с нестандартными формами нельзя скомпоновать в регулярные ряды или матрицы – они требуют разногабаритного, бессеточного размещения [1,2].

Процессом решения задачи размещения блоков СБИС является преобразование электрической схемы, состоящей из модулей с предопределенными входными терминалами и межсоединениями, в топологию с заданными геометрическими позициями ее модулей. При этом критериями эффективности решений могут служить: минимум общей занимаемой площади или минимум суммарной длины про-

трассированных межсоединений. Эти критерии косвенно связаны между собой, т.е. минимизация длины межсоединений эквивалентна минимизации площади.

Разногабаритные блоки при нерегулярном размещении не всегда подходят по размерам друг к другу и часть площади кристалла при этом теряется (остается неиспользованной). Существует некий баланс приоритетов в минимизации определенного параметра. Другие критерии допустимости варианта размещения: модули не должны пересекаться, они должны располагаться в пределах поля кристалла.

Задача размещения блоков СБИС является NP-полной, поэтому получение ее точного решения за полиномиальное время невозможно [3-5].

Для поиска оптимума среди множества возможных решений за приемлемое время используется масса итерационных эвристических алгоритмов, которые помогают получить хороший результат (не всегда наилучший). И часто этого вполне достаточно. Такие эвристические алгоритмы производят итерационное улучшение целевой функции с различными начальными условиями и стремятся получить приемлемое решение в одном из вариантов размещения.

В настоящее время к популярным итерационным алгоритмам относят алгоритмы имитации отжига, генетические и некоторые силовонаправленные методы размещения. Генетические алгоритмы это достаточно мощные оптимизационные алгоритмы, которые моделируют естественный эволюционный процесс как средство достижения оптимальных решений. Впервые идея применения эволюционных процессов для решения задач оптимизации была описана в [6].

Алгоритмы имитации отжига и генетический требуют интенсивных вычислений, однако генетический алгоритм обладает потенциальными возможностями к сокращению необходимого объема вычислительных затрат. Имитация отжига в каждый момент времени производится только с одной конфигурацией, генетический алгоритм, напротив, оперирует большим набором (популяцией) конфигураций, оптимизирующихся одновременно. Это приводит к тому, что метод моделирования отжига, имея для анализа «короткую» историю пробных вариантов, ведет поиск в узкой области, а генетический алгоритм позволяет расширить соответствующее пространство поиска и сделать акцент на «наилучших» областях.

Эффективность генетического алгоритма зависит от правильного выбора генетических операторов и набора различных параметров [7,15]. Наиболее часто используемые генетические операторы – кроссовер, мутация и инверсия, являются аналогами из биологических эволюционных процессов. Главная проблема в разработке генетического алгоритма заключается в выборе и эффективной реализации наиболее подходящих задаче операторов.

**Способ кодирования данных о размещении блоков.** В настоящее время для кодировки расположения блоков на рабочем поле используются различные способы. Например, для описания положения блоков на регулярных сетках применяют бинарные деревья, в которых листьями являются блоки и каждый внутренний узел определяет вертикальную или горизонтальную связь с двумя его потомками. В работе [8] положение разногабаритных блоков описывается отношениями между двумя блоками четырьмя классами и используется метод ветвей и границ для решения задачи размещения. В работах [9,10] применяются последовательные пары и ограниченные регулярные сетки для оперирования с разногабаритными блоками с малой областью решений. Эти два способа используют граф ограничений для трансформаций размещения и сложны для представления графа в топологии.

Результаты современных исследований показывают, что описание размещения на регулярных сетках относительно просто, но описания размещения разногабаритных блоков достаточно сложны и неоднозначны. При проектировании заказных СБИС используются не матричные структуры (полузаказные), а свободные от

сеток разногабаритные блоки. Поэтому для эффективного решения задачи размещения блоков заказных СБИС очень важно разработать простую и удобную для оперирования структуру представления данных о положении блоков на кристалле.

В данной работе предлагается использовать направленное диагональное дерево представления для описания размещения разногабаритных блоков СБИС, в отличие от горизонтального дерева, использованного в работе [11].

Дерево является графом  $G = (V, E)$ , у которого узлы  $V$  – размещаемые блоки, и ребра  $E$  – геометрические отношения между двумя блоками. Направленные ребра  $E$  показывают отношения “родитель-потомок”. В данном графе каждый родитель может иметь не более двух потомков –  $P_L$  (левый) и  $P_R$  (правый).

Корнем дерева является узел-блок, лежащий в левом нижнем углу поля размещения. Кодировочная хромосома (КХ) представляет собой объектное описание иерархии дерева на основе ссылок (рис. 1). Каждый узел дерева может иметь не более двух дочерних узлов (бинарное дерево). Это связано с особенностями процедуры построения топологии, о чем будет сказано ниже.

В данном дереве необходимо явно кодировать обе ветви  $P_L$  и  $P_R$  – левую и правую. Другими словами, при отсутствии одной из ветвей-потомков должно быть ясно, какая из них присутствует – левая или правая, потому что они неэквивалентны. Кодировки бинарных деревьев, применяемые в [12], не позволяют учитывать это обстоятельство.

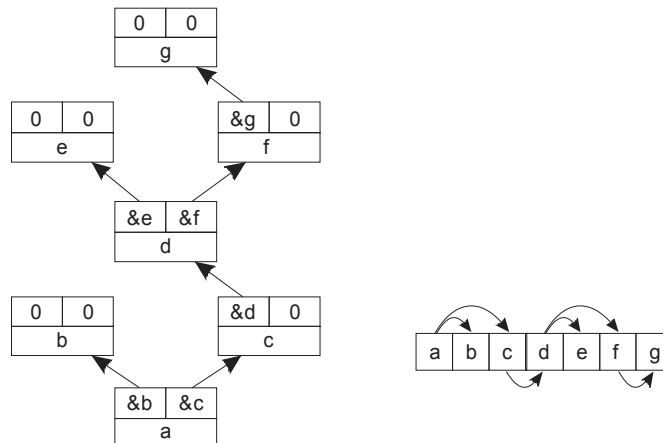


Рис. 1. Пример описания бинарного дерева на основе ссылок

Блоки, размещаемые на рабочем поле, являются прямоугольниками с различными отношениями сторон. Поэтому важно знать, как располагается блок – горизонтально или вертикально. Для описания такого положения блока достаточно одного кодирующего бита: 1 – вертикально, 0 – горизонтально. Однако при расчете значения целевой функции (ЦФ), может учитываться суммарная длина связей, соединяющих не центры блоков, а контакты на границах или внутри блока. В этом случае необходимо кодировать не два возможных положения блока, а четыре (рис. 2), т.е. необходимо два бита.

КХ содержит список узлов-объектов, каждый из которых имеет собственный код (имя) и две адресные ссылки на дочерние узлы-объекты. Каждый узел-объект также должен содержать код ориентации (1 или 2 бита). Сопутствующая узлу-объекту параметрическая информация (идентификатор блока, габаритные размеры и т.п.) в КХ не входит, т.к. не влияет на конфигурацию дерева. Поэтому данная информация хранится в единственном экземпляре.

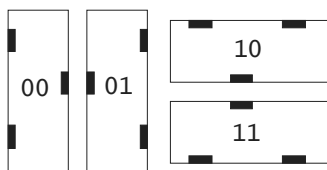


Рис. 2. Варианты ориентации блоков

Таким образом, для кодирования направленного дерева с  $n$  узлами-объектами минимально необходимое количество бит  $L$  можно рассчитать по формуле:

$$L = 3n[\log_2 n] + 2n = n(3[\log_2 n] + 2). \quad (1)$$

Для сравнения, способ представления, описанный в [11], требует для кодировки бинарного дерева  $n([\log_2 n] + 2)$  бит, однако он не учитывает ориентации блоков на рабочем поле и требует больших вычислительных затрат при декодировании. Число возможных кодирующих конфигураций  $O(n!2^{2n-2}/n^{1.5})$  [13].

Предложенный способ кодирования имеет компактную форму представления данных о размещении и позволяет эффективно оперировать ей.

**Декодирование хромосомы и построение топологии.** Для построения топологии рабочего поля необходимо декодировать КХ, т.е. построить дерево и разместить блоки согласно правилам, заложенным в кодировке топологии. В алгоритме построения дерева из КХ движение по ветвям дерева производится согласно принципу поиска в глубину.

Этапы построения топологии из дерева (первым текущим блоком является корневой узел-объект):

- ◆ текущий блок устанавливается в вакантное место. Под вакантным местом понимается LB-компактное (левое нижнее) положение вставляемого блока;
- ◆ строится огибающий контур с учетом текущего размещенного блока;
- ◆ новый огибающий контур будет иметь два вакантных места для потомков  $P_L$  и  $P_R$  вблизи текущего блока (слева сверху и справа снизу);
- ◆ далее в эти вакантные места размещаются дочерние блоки (если существуют);
- ◆ процесс повторяется до тех пор, пока все блоки не будут размещены на поле (текущий блоком становится следующий по порядку следования в глубину).

Пример этапов построения дерева приведен на рис. 3.

Координаты вакантного места определяются не только по контуру, но и по габаритам вставляемого блока.

Для преобразования дерева в топологию описанным способом необходимо линейное время.

В [11] горизонтальное дерево описания размещения декодируется в топологию в несколько этапов. Сначала дерево преобразуется в граф ортогональных ограничений, из которого далее получается вертикальное дерево и затем опять горизонтальное дерево LB-компактной топологии. Для разработанного диагонального дерева таких преобразований не требуется, т.к. LB-компактная топология получается за один этап построения топологии. Это позволяет существенно сократить временные затраты на выполнение итераций алгоритма размещения.

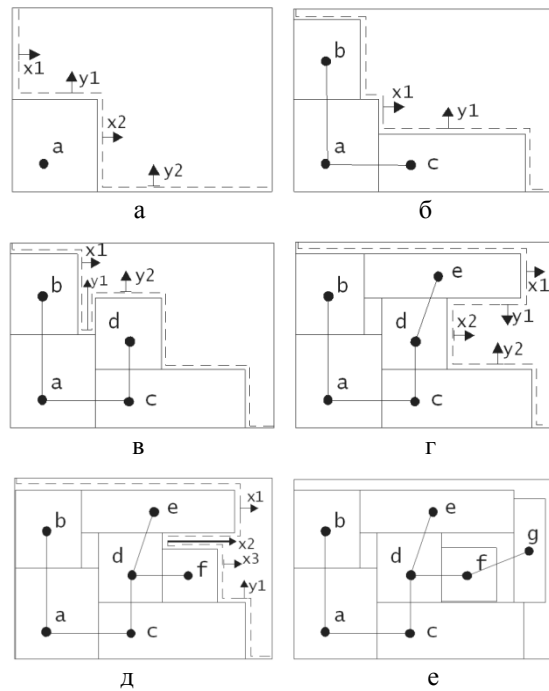


Рис. 3. Демонстрация этапов декодирования КХ («а» – корневой блок)

**Генетические операторы.** Генетические операторы позволяют создавать новое поколение особей путем комбинирования структуры (или локальных подструктур) родителей, выбранных из нынешнего поколения. Из-за стохастического процесса отбора, «хорошие родители», содержащие, например, наиболее компактные локальные подструктуры, произведут больше потомства, нежели «плохие родители» с плохими показателями локальных подструктур. Таким образом, в последующих поколениях, количество вариантов размещений в хорошиими показателями имеет тенденцию к увеличению, а количество неэффективных структур будет снижаться. Другими словами, целевая функция всей популяции будет улучшаться.

Существует два основных вида генетических операторов (ГО) – кроссинговер (перекрест) и мутация [6,7].

Кроссинговер осуществляет скрещивание генов родительских структур. Операторы кроссинговера делятся на несколько типов, которые являются частными случаями универсального кроссинговера. Мутация – производит стойкое изменение генотипа, происходящие под влиянием внешней или внутренней среды. Мутации делятся на спонтанные и индуцированные (наследуемые изменения генома).

Для разработки ГО применительно к конкретной задаче необходимо исследовать ее особенности. Ниже приведены основные свойства разработанных КХ бинарного дерева:

- ◆ первый узел-объект в списке является корнем дерева;
- ◆ вставка нового узла-объекта (“листа”) или поддерева может быть произведена в любую позицию с пустой ссылкой на потомка;
- ◆ допустимы любые перестановки между потомками одного узла-объекта;
- ◆ “листья” дерева имеют две пустые ссылки на потомков;

- ♦ вставка внутреннего узла-объекта может быть произведена в любое место дерева с применением операции копирования ссылок (для сохранения иерархической взаимосвязанности).

Ниже приведен список разработанных ГО, позволяющих оперировать с КХ.

I. ГО инверсии (форма мутации). Применение оператора инверсии фактически даст вращение поддерева относительно узла на любом выбранном уровне. Сначала выбирается осевой узел, относительно которого будет происходить вращение. Затем выделяется поддерево, внутри которого в каждом узле меняются местами ссылки на потомки, включая осевой узел (рис. 4).

II. ГО мутации № 1. Суть оператора состоит в том, что он производит парную перестановку имен узлов по определенным правилам:

- ♦ перестановка имен производится путем обмена именами и свойствами между двумя, определенным образом выбранными, узлами-объектами;
- ♦ пары взаимнообменивающихся узлов могут выбираться в заданных областях (уровнях иерархии). Например, на начальных стадиях жизни популяции члены пары могут принадлежать любым иерархическим уровням, а на последующих – только верхним.

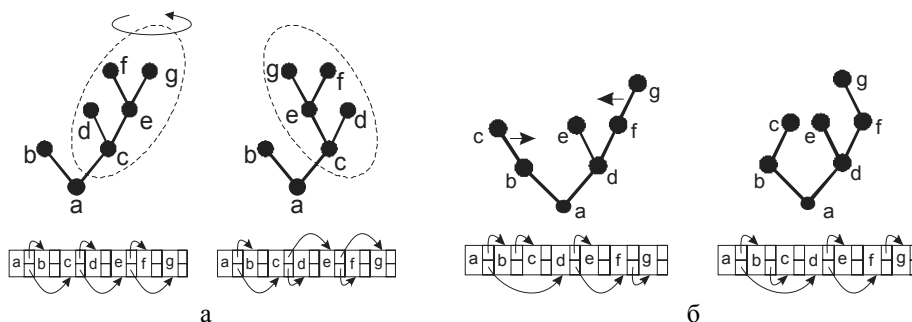


Рис. 4. ГО инверсии: а – поворот поддеревьев; б – поворот листьев

Применение данного ГО не приводит к изменению структуры дерева (рис. 5).

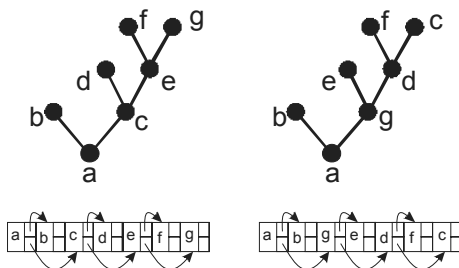


Рис. 5. ГО мутации: парная перестановка имен

III. ГО мутации № 2. Оператор осуществляет изменение ориентации блоков без изменения их позиции и все структуры дерева. Процедура изменения ориентации также может охватывать различные иерархические уровни в зависимости от времени жизни особей в популяции (рис. 6).

IV. ГО упорядоченного кроссинговера. Данный оператор позволяет потомкам наследовать лучшие свойства родительских особей. Суть его состоит в том, что

происходит перекрестный обмен конструкциями одинаковой длины в родительских хромосомах. Сначала в потомок D2 в соответствующую позицию вставляется обменная конструкция из родителя P1, затем добавляются недостающие гены из P2. В потомок 6 в соответствующую позицию вставляется обменная конструкция из родителя P2 и добавляются недостающие гены из P1.

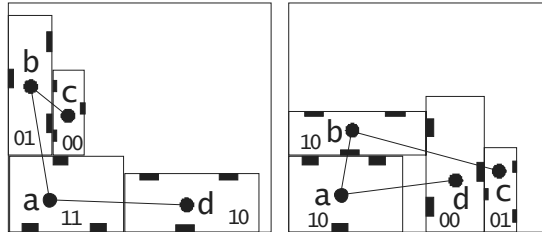


Рис. 6. ГО мутации: изменение ориентации блоков

Применение данного оператора возможно, если в родителях есть конструкции одинаковой длины, иначе они не подходят для скрещивания. Следовательно, применяемая система скрещивания должна учитывать это обстоятельство и производить предварительный отбор совместимых родителей.

Предложенный набор ГО может быть направленным и ненаправленным, что в итоге позволяет улучшить процесс поиска оптимальных вариантов размещения. Ненаправленные ГО воспроизводят особи со случайным изменением их свойств по отношению к предкам. Направленные ГО культивируют заданные свойства особей. Вероятность применения случайных операторов должна убывать со временем, а направленных – увеличиваться. Например, в [14] вероятность применения кроссинговера – (0,01-0,25), а направленной мутации:

$$P_m = P_{m0}(1 - e^{-t}), \quad (2)$$

где  $P_{m0}$  – начальное значение вероятности применения ГО;  $t$  – время эволюции.

Для анализа эффективности применения различных ГО был проведен набор вычислительных экспериментов по тестовому размещению блоков. С целью обеспечения статистической достоверности для каждого типа ГО выполнялись тестовые размещения пяти различных наборов блоков по 10 тыс. экспериментов в каждом (a, b, c, d, e на рис. 7-9). В качестве аргумента каждой функции ГО было принято процентное отношение числа задействованных в особи иерархических уровней от 10 % до 100 %.

Результаты применения ГО оценивались по величине улучшения целевой функции (ось ординат на рис. 7-9), рассчитываемое как отношение между итоговым и начальным ее значением.

Как показали исследования, эффективность применения ГО на 10 % уровней дерева особи практически не отличается от применения ГО на большем проценте уровней (для инверсии на 100 % происходит даже ухудшение). Данное свойство можно использовать для резкого сокращения вычислительных затрат на поиск оптимального решения задачи размещения.

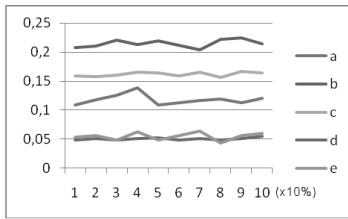


Рис. 7. Эффективность применения парной перестановки имен

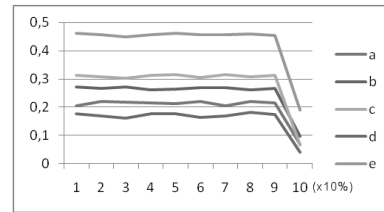


Рис. 8. Эффективность инверсии

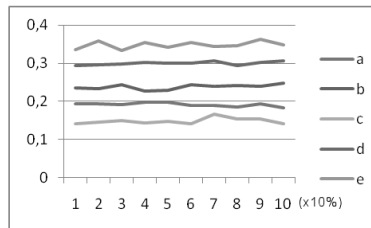


Рис. 9. Эффективность изменения ориентации

**Генетический алгоритм и целевая функция.** Этапы выполнения генетического алгоритма аналогичны описанных в литературе, например в [6]. Учитывая специфику задачи размещения разногабаритных блоков в интегральных схемах, вводится процедура оценки энергоэффективности решений (целевая функция): минимум потребляемой межсоединениями энергии при минимальной задержке сигналов в них. Для определения энергоэффективности вводится интегральный критерий F:

$$F = \frac{1}{1 - \frac{C_{mino}}{\beta C_{mini}}} \left( \left( m RC_{min} \left( 1 + \frac{C_{mino} (\beta - 1)}{C_{mini} + C_{mino}} \right) + \frac{(L_c + L_{sp}) (N + 1) \sqrt{1 + \frac{C_{pin} + C_{mini}}{(L_c + L_{sp}) C_{int}}}}{v} \right) (C_{pin} + \beta^{m-1} C_{mino} + (L_c + L_{sp}) (N + 1) C_{int} + N (C_{pin} + C_{mini})) V_{dd}^2 \right) \quad (3)$$

где  $C_{mino}$  – выходная емкость минимального инвертора;  $C_{mini}$  – входная емкость минимального инвертора;  $\beta$  – коэффициент прогрессии увеличения размеров выходных буферов в цепочке;  $m$  – число буферов в цепочке;  $RC_{min}$  – минимальный коэффициент RC;  $L_c$  – средняя длина стороны принимающего блока;  $L_{sp}$  – расстояние между принимающими блоками;  $N$  – число принимающих блоков;  $C_{pin}$  – емкость вывода;  $C_{int}$  – погонная емкость линии передачи;  $v$  – скорость распространения электрического сигнала;  $V_{dd}$  – напряжение питания.

**Заключение.** Разработанный ГА отличается от существующих способом кодирования и декодирования хромосом, набором адаптированных к особенностям задачи операторов. Предложенный способ кодирования отличается от существующих учетом ориентации блоков, учетом их относительного положения и удобством оперирования хромосомой. Способ декодирования позволяет сократить время построения ЛВ-компактной топологии. Созданные генетические операторы адаптированы к особенностям кодиро-



вания хромосом и позволяют эффективно преодолевать барьеры локальных оптимумов пространства решений задачи размещения разногабаритных блоков.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Muroga S. VLSI System Design // John Wiley, New York, Chap. 9. – 1982. – P. 365-395.
2. Ueda K., Kasai R. and Sudo T. Layout strategy, standardization, and CAD tools // In Layout Design And Verification, Elsevier Science Pub. Co., New York, Chap. 1, 1986.
3. Donath W.E. Complexity theory and design automation // In Proceedings of the 17th Design Automation Conference. – 1980. – P. 412-419.
4. Leighton F.T. Complexity Issues in VLSI // MIT Press, Cambridge, Mass., 1983.
5. Sahni S. and Bhatt A. The complexity of design automation problems // In Proceedings of the 17th Design Automation Conference, 1980 – P. 402-411.
6. Holland J.H. Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, 1975.
7. Курейчик В.М. Генетические алгоритмы. Монография. – Таганрог: Изд-во ТРТУ, 1998. – 240 с.
8. Onodera H., Taniguchi Y., Tamaru K. Branch-and-Bound Placement for Building Block Layout, DAC, 1991. – P. 433-439.
9. Murata H., Fujiyoshi K., Nakatake S., Kajatani Y. Rectangular-Packing-Based Module Placement, ICCAD, 1995. – P. 472-479.
10. Nakatake S., Fujiyoshi K., Murata H., Kajatani Y. Module Placement on BSG-Structure and IC Layout Applications, ICCAD, 1996. – P. 484-491.
11. Guo P.N., Cheng C.K., Yoshimura T. An O-Tree Representation of Non-Slicing Floorplan, DAC 36, 1999.
12. Aktuna M., Carley R.L. Rutenbar Device-Level Early Floorplanning Algorithms for RF Circuits // IEEE Trans. Computer-Aided Design. – 1999. – Vol. 4.
13. Knuth D.E. The Art of Computer Programming, 2nd Ed., Vol. 1, Addison-Wesley Publishing Co., 1973. – P. 385-395.
14. Ведерникова О.Г. Разработка и исследование комбинированного алгоритма генетического поиска и имитации отжига для задачи размещения элементов СБИС: Дис... канд. тех. наук: 05.13.12/ РГА Сельхозмаш. – Ростов-на-Дону, 1999. – 152 с.
15. Курейчик В.В., Курейчик В.М., Родзин С.И. Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-24.

#### **Ковалев Андрей Владимирович**

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

E-mail: andr@fep.tsure.ru.

347928, Таганрог, ул. Шевченко, 2.

Тел.: 88634371603.

#### **Бушин Сергей Алексеевич**

Общество с ограниченной ответственностью «Астор-Трейд».

E-mail: sergey@incotex.ru.

105484, г. Москва, ул. 16-я Парковая, 26.

Тел.: 84954684334.

#### **Kovalev Andrey Vladimirovich**

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

E-mail: andr@fep.tsure.ru.

2, Shevchenko street, Taganrog, 347928, Russia.

Phone: +78634371603.

#### **Bushin Sergey Alekseevich**

“Astor-Trade” Company.

E-mail: sergey@incotex.ru.

26, 16<sup>th</sup> Parkovaya street, Moscow, 105484, Russia.

Phone: +74954684334.