

УДК 681.31.001.8

В.В. Гудилов

РЕКОНФИГУРИРУЕМЫЕ ВЕКТОРНЫЕ ЭВОЛЮЦИОННЫЕ ПРОЦЕССОРЫ

Представлены методы построения архитектуры реконфигурируемых векторных процессоров, предназначенных для моделирования процессов эволюции на основе различных стратегий, в том числе открытой эволюции. Рассмотрены общие принципы построения векторных генетических процессоров, а также организация эволюционного процессора, применяемого для построения многопроцессорных эволюционных систем. Приводится описание выполнения эволюционного процесса с организацией четырехуровневого параллельного взаимодействия генетического материала.

Реконфигурируемые эволюционные процессоры; генетические процессоры; эволюционные системы; векторные эволюционные процессоры; аппаратная реализация генетических алгоритмов; параллельные генетические алгоритмы; динамические эволюционные алгоритмы.

V.V. Gudilov

RECONFIGURED VECTORIAL EVOLUTIONARY PROCESSORS

Methods of creation of architecture of the reconfigured vector processors appropriated for modeling of processes of evolution on the basis of various strategies are presented, including an open evolution. It is considered the general principles of creation of vectorial genetic processors, and also the organization of the evolutionary processor applied to creation of the multiprocessor evolutionary systems. The description of performance of evolutionary process with the organization of four-level parallel interaction of a genetic material is resulted.

Reconfigured evolutionary processors; genetic processors; evolutionary systems; vectorial evolutionary processors; hardware implementation of genetic algorithms; parallel genetic algorithms; dynamic evolutionary algorithms.

Введение. Моделирование эволюции для решения задач поиска или оптимизации с применением эволюционных алгоритмов (ЭА) сталкивается с временными ограничениями, накладываемыми структурой эволюционного процесса, естественное развитие которого в природе занимает миллионы лет. Программная реализация ЭА с применением современных персональных компьютеров позволяет ускорить эволюционный процесс до секунд, но приблизить эволюцию к масштабам реального времени возможно только с применением аппаратных акселераторов или специализированных процессоров. Как было показано [1], применение подобных решений может выступать не только в качестве метода повышения быстродействия, но и предоставляет возможность использовать ЭА в автономных системах и комплексах. При разработке методов эволюционного моделирования открытой эволюции, в которых структура представления решения и ЭА является динамической и может изменяться в процессе эволюционного поиска, применение только методов программной реализации затруднительно уже на этапе разработки и исследования. Подобные классы алгоритмов оперируют разнообразным генетическим материалом, одновременно представленном во множестве пространств поиска, и нуждаются в значительных вычислительных мощностях, ориентированных на возможность динамической реконфигурации, как собственной архитектуры, так и ее элементов.

В данной работе будут рассмотрены методы построения аппаратной архитектуры реконфигурируемых векторных эволюционных процессоров, которые предлагается использовать для решения задач с применением методов искусственного

моделирования эволюции, а также методов, инспирированных природными системами [2] и при проектировании аппаратно-ориентированных ЭА в автономных системах и инструментальных комплексах. В основу работы положен опыт разработки и аппаратной реализации генетических алгоритмов, а также предложенные ранее методы аппаратного распараллеливания вычислений, хорошо зарекомендовавшие себя при проектировании аппаратных структур генетических алгоритмов.

Структура эволюционных процессоров. При разработке высокопроизводительных многопроцессорных систем большое значение имеет топология межсоединений процессоров. Действительно, с одной стороны, система должна быть расширяемой, с другой, топология должна максимально полно соответствовать структуре решаемой задачи. Известно [3], что ЭА в качестве экземпляра решения оперируют хромосомами (особями), закодированными в виде последовательностей (векторов) данных, как правило, имеющих большую размерность, при этом совокупность решений определяет популяцию, которую можно представить в виде двумерной матрицы, элементами которой являются гены хромосом. Многие операции над генетическим материалом могут быть сведены к выполнению операций над матрицами [1]. Следовательно, структура межпроцессорных соединений может иметь вид прямоугольной решетки, что значительно упростит механизмы адресации и позволит легко наращивать архитектуру.

Необходимо заметить, что структура хромосомы представляет собой набор ген (образующих вектор), закодированных в выбранной системе счисления или специальным образом, и при развитии эволюционного процесса выполняется оперирование генетическим материалом (выполнение операторов мутации, кроссинговера и др.) на основе набора ген, иногда различной длины и структуры. При этом, порядок обработки ген хромосомы обычно не определен, и в лучшем случае, обработка всех ген хромосомы должна выполняться параллельно. Для выполнения данного условия наиболее оптимальным является применение процессоров векторного типа, обладающих свойством массового параллелизма посредством параллельной обработки всех элементов вектора на наборе элементарных арифметико-логических устройств. Таким образом, можно сделать вывод, что разрабатываемая архитектура должна иметь вид прямоугольной решетки, в узлах которой находятся специализированные векторные процессоры.

Если взглянуть на задачу шире, то несложно видеть, что многопроцессорную систему целесообразно строить по принципу хост-контроллера (или микроконтроллера [4]), при котором в системе имеется базовый процессор (БП), и набор генетических процессоров (ГП) – реализующих заданный набор генетических операторов. Задача базового процессора – обеспечение коммутации и управление генетическими процессорами таким образом, чтобы посредством их функционирования реализовать заданный эволюционный процесс.

Т.к. для различных задач максимальная длина хромосомы различна, то для достижения эффекта наибольшей пропускной способности необходимо параллельное объединение генетических процессоров в единый вычислительный блок. Посредством соответствующего объединения вычислительных блоков возможно создание конвейера, когда текущий блок получает данные от предыдущего, выполняет требуемое генетическое преобразование и передает данные последующему блоку. Для обеспечения универсальности, необходимо, чтобы каждый ГП включал в себя набор всех необходимых генетических операторов. Несложно видеть, что для аппаратной реализации эволюционного процесса необходимо иметь БП, и как минимум один ГП. Тогда увеличение числа генетических процессоров необходимо только для повышения быстродействия.

При подобной организации взаимодействия процессоров, наиболее эффективным будет построение структуры в виде 3-х мерного гиперкуба, грани которого задают коммутационные каналы связи, в вершинах расположены генетические процессоры, а в центре расположен базовый процессор (рис. 1,а.). Каждый из генетических процессоров имеет по четыре программируемых канала связи – три для передачи (или получения) данных соседним (или от соседних) ГП и один к БП. Соответственно базовый процессор имеет восемь прямых каналов связи с окружающими его ГП и дополнительный канал внешнего управления (подсистему ввода-вывода). Данную структуру (состоящую из 8 генетических и одного базового процессоров) далее будем называть эволюционным процессором (ЭП).

Структура ЭП определяет максимальное количество генетических процессоров, равное восьми. При необходимости, возможно наращивание производительности посредством увеличения числа ЭП в системе (объединение выполняется на основе подсистемы ввода-вывода), соединенных по схеме, приведенной на рис. 1,б, где в вершинах куба находятся ЭП, тем самым, объединяя 64 генетических процессора в единую эволюционную систему. Применение подобной эволюционной системы возможно для параллельной организации эволюционного поиска на основе различных эволюционных стратегий [2]. Количество генетических процессоров может варьироваться в зависимости от потребностей решаемой задачи. В силу увеличения сложности алгоритма управления эволюционной системой, для организации эволюционного процесса в данном случае потребуется внешнее управление.

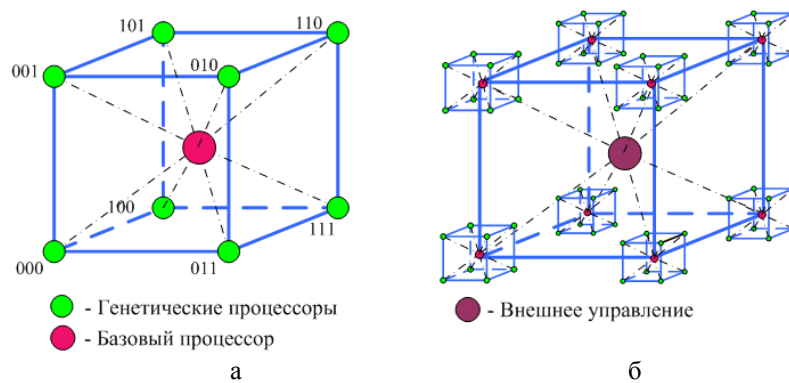
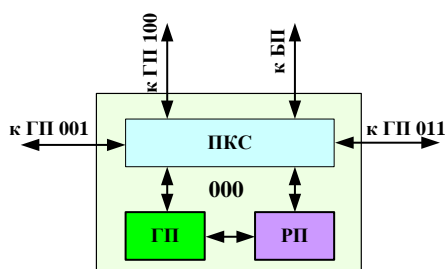


Рис. 1. Организация эволюционного процессора: а – в виде 3-х мерного гиперкуба в составе 8-ми генетических процессоров; б – эволюционная система из 64 генетических процессоров

Архитектура эволюционного процессора. Каждый генетический процессор имеет прямой доступ к своей собственной памяти (распределенная память) и при работе взаимодействует только с данным банком памяти (рис. 2). Добавление в систему нового генетического процессора осуществляется совместно с добавлением распределенной памяти данного процессора. Связка: ГП + РП при этом получает единую нумерацию в наборе генетических процессоров, т.е. адрес процессора отождествляется с его распределенной памятью. Тогда для базового процессора совокупность банков памяти генетических процессоров представлена в виде чередуемой памяти с единым адресным пространством.



ПКС – программируемая коммутационная структура
 ГП – генетический процессор, РП – распределенная память

Рис. 2. Представление вершины 3-мерного гиперкуба – блок генетического процессора

Со стороны базового процессора запросы к различным банкам памяти генетических процессоров могут выполняться одновременно, поэтому базовый процессор может выполнять чтение (или запись) из любого банка памяти. Кроме того, у БП имеется своя память (основная память), выступающая в роли центрального хранилища, доступ к которой имеется только у базового процессора. Подобная архитектура взаимодействия позволяет выполнять параллельную обработку хромосом на различных ГП, а также обрабатывать хромосомы по частям на отдельных генетических процессорах, в случае, если длина хромосомы превышает разрядность генетического процессора. Промежуточные результаты сохраняются в распределенной памяти ГП, с последующей сборкой хромосом в памяти базового процессора. Другим положительным эффектом является то, что генетические процессоры в процессе работы взаимодействуют только с распределенной памятью, что исключает необходимость ожидания данных или решения вопросов совместного использования памяти, при одновременном обращении нескольких генетических процессоров к одному адресному пространству памяти. Схема взаимодействия процессоров упрощается, что приводит к уменьшению нагрузки на коммутационную систему и упрощает алгоритм управления системой в целом. Недостатком является то, что задача подготовки исходных данных и сбора результатов из распределенной памяти генетических процессоров полностью возлагается на базовый процессор, что в некоторой степени усложняет алгоритм его функционирования.

Как было упомянуто выше, ГП имеет возможность прямой коммутации с четырьмя внешними блоками. Для обеспечения данной возможности используется коммутационная структура на основе программируемых каналов связи. Программируемая коммутационная структура (ПКС) позволяет выполнять одновременную коммутацию шести каналов связи, обеспечивая тем самым максимальную гибкость и универсальность при обеспечении функции передачи данных, и попарно объединяя шесть периферийных устройств. Например, возможно одновременное соединение генетического процессора с двумя соседними ГП и организация канала связи между распределенной памятью данного ГП и базовым процессором. Различные схемы соединений, реализуемые на ПКС, позволяют выполнить настройку структуры ЭП на организацию любого необходимого межпроцессорного взаимодействия. В том числе, возможна коммутация соседних узлов в обход данного ГП, что позволяет коммутировать несмежные вершины гиперкуба напрямую, не используя при этом ГП. Возможность перепрограммирования ПКС в процессе функционирования обеспечивает функцию реконфигурации ЭП на структурном уровне. Подробное описание архитектур применяемых коммутационных структур будет

рассмотрено в статье «Методы реконфигурации структуры эволюционных процессоров», в данном изложении ограничимся лишь указанием общих функциональных характеристик.

В рамках структуры ЭП можно выделить несколько каналов связи: упомянутые программируемые каналы связи, прямые каналы связи и каналы управления, по типу коммутации, относящиеся к прямым каналам связи. Объединение узлов ЭП (ГП и БП) с помощью каналов связи определяет архитектуру ЭП, представленную на рис. 3.

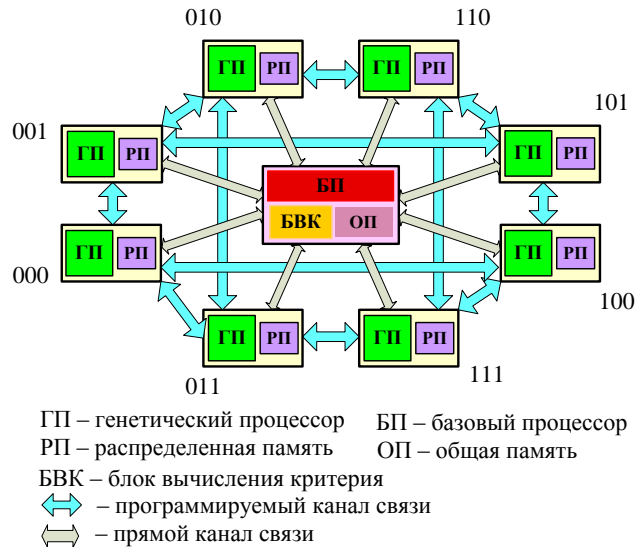


Рис. 3. Архитектура эволюционного процессора

Как видно из рисунка (см. рис. 3), блок базового процессора содержит общую память и блок вычисления критерия (БВК), который функционально представляет собой отдельный векторный процессор, аппаратно выполняющий оценку критерия пригодности хромосом по заданному алгоритму (или набору алгоритмов) вычисления. В силу того, что специфика задач, для решения которых используются алгоритмы эволюционного поиска, накладывает отпечаток на алгоритмы оценки решений, разработка БВК должна выполняться для каждого класса задач (а зачастую и каждой задачи) в отдельности [5]. Возможна разработка и применение универсальных БВК для набора эволюционных стратегий, при аппаратной реализации которых необходимо применять методы распараллеливания, конвейеризации и векторных вычислений.

Архитектура генетического процессора. Построение архитектуры генетического процессора рассмотрим с позиции взаимодействия генетических операторов в моделируемых ЭА, а так же принципа взаимодействия ГП с внешними устройствами. Отдельным вопросом выступает определение разрядности генетического процессора. Блок-схема генетического процессора представлена на рис. 4. Генетический процессор включает набор основных генетических операторов, программируемую коммутационную структуру, набор двунаправленных регистров, внутреннюю кэш-память и блок микропрограммного управления (БМПУ). Применение программируемой коммутационной структуры позволяет выполнять произвольную коммутацию блоков, с организацией циклов и их параллельного функционирования. Для хранения промежуточных результатов вычисления использу-

ется внутренняя кэш память, для управления адресацией которой задействован блок адресного указателя. Управление устройством выполняется с помощью блока микропрограммного управления [6], построенного по принципу генерации параллельных микрокоманд управления для всех блоков процессора и обработки приоритетных аппаратных прерываний. Данный блок содержит внутреннюю память микрокоманд, в которой хранятся микрокоманды настройки и управления как отдельных блоков, так и их совокупности, для организации параллельного функционирования блоков. Выбор режима управления внутренними блоками ГП производится на основе внешнего управления, при котором БМПУ получает команды инициализации от базового процессора, после чего БМПУ выполняет инициализацию блоков генетического процессора и далее происходит отработка соответствующего режима функционирования. Процесс инициализации включает этап коммутации блоков, на основе которого выполняется реконфигурация структуры процессора с целью получения требуемой архитектуры. По завершению отработки текущей операции, полученные данные эволюционных преобразований сохраняются в кэше или распределенной памяти ГП, далее возможна очередная реконфигурация ГП для выполнения следующей операции.

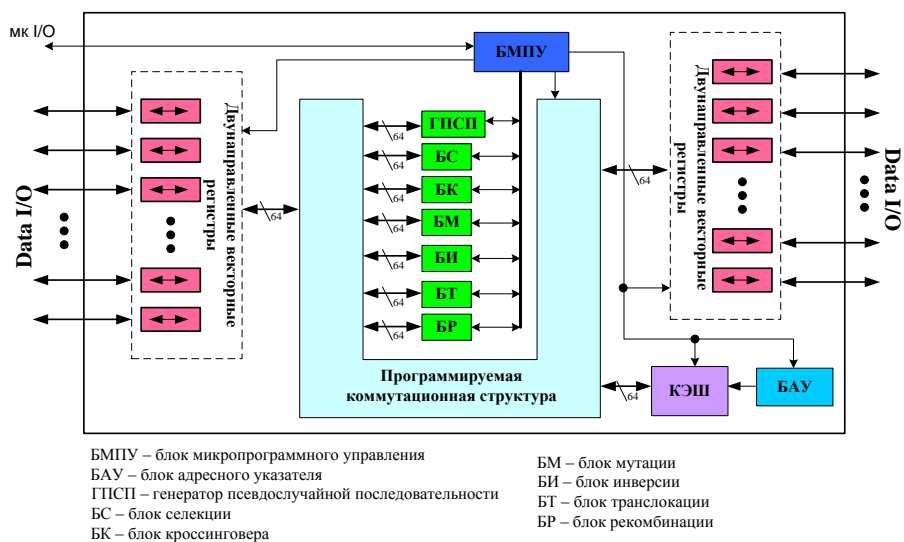


Рис. 4. Блок-схема генетического процессора

Взаимодействие ГП с внешними устройствами определено в виде схемы из регистра в регистр, что позволяет исключить необходимость постоянного обращения к внешней памяти. Данные загружаются во входные двунаправленные векторные регистры, после чего выполняется эволюционная обработка, и результат помещается либо в кэш, либо в один из выходных регистров для передачи следующему устройству (в зависимости от настройки схемы взаимодействия ГП в эволюционном процессоре). Разрядность данного типа ГП равна 64 битам, что позволяет эффективно использовать многокаскадные сети MCRB [6], как для построения программируемой коммутационной структуры, так и для реализации некоторых генетических операторов. Возможно наращивание разрядности ГП до требуемой величины как посредством параллельного использования нескольких однотипных ГП, так и посредством разработки векторного генетического процессора требуемой разрядности. В последнем случае, необходимо учесть значительное повыше-

ние требований к емкости кристалла ПЛИС (FPGA) [7], в рамках которого будет выполняться разработка векторного ГП, а также доступное количество контактов кристалла, необходимых для осуществления коммутации ГП в структуру эволюционного процессора.

Пример разработки и аппаратной реализации 32-разрядного векторного генетического процессора представлен автором в работах [1, 6, 8], принципы организации векторных вычислений для реализации генетических операторов показаны в [9–11]. Практическим результатом, достигнутым при аппаратной реализации генетического процессора, было достижение быстродействия, превосходящего программную реализацию более чем на три порядка, а так же возможность использования генетических алгоритмов в автономных структурах. Повышение быстродействия было достигнуто как за счет векторной обработки данных, так и организации многоуровневого конвейера на основе параллельного выполнения генетических операторов.

Организация эволюционного процесса. Моделирование процесса эволюции для различных эволюционных стратегий производится посредством соответствующей настройки генетических процессоров и их взаимодействия. Граф алгоритма эволюционного процесса разбивается на подграфы, вершины которого определяют генетические операторы, дуги – взаимодействие генетического материала между операторами. Далее выполняется отождествление полученных подграфов структуре ЭП, в котором выполняется настройка ГП на выполнение заданных генетических операторов и настраивается программируемая коммутационная структура на организацию взаимодействия между ГП. Если длина хромосомы превышает разрядность генетического процессора, возможно разбиение хромосомы на части, кратные разрядности ГП, с последующей поэтапной обработкой, или объединение нескольких ГП в параллельную структуру, где выполняется параллельная обработка всех ген хромосомы.

Параллельное моделирование нескольких эволюционных стратегий на заданном наборе ГП возможно посредством распределения ЭА между различными группами ГП, или совместного использования ГП посредством реконфигурации их архитектуры и перенастройки коммутационной структуры.

Выводы. В работе рассмотрены методы построения реконфигурируемой архитектуры ЭП, чего ранее никто не предлагал. Показано взаимодействие генетических процессоров для организации эволюционного процесса, и раскрыты общие принципы построения векторных генетических процессоров. Отличительной чертой представленной архитектуры ЭП является обеспечение четырех уровней параллельных вычислений:

- ◆ параллелизм на уровне распределения заданий, когда каждый генетический процессор выполняет независимую от других ГП задачу;
- ◆ параллелизм на уровне подпрограмм, когда основная программа разбивается на части, и каждый ГП выполняет свою часть программы;
- ◆ параллелизм на уровне организации конвейерного взаимодействия ГП, когда генетические процессоры посредством настройки на выполнение соответствующих генетических операторов и организации межпроцессорного взаимодействия образуют конвейер, в рамках которого протекает эволюционный процесс;
- ◆ параллелизм на векторном уровне, когда на ГП выполняется параллельная обработка всех элементов вектора входных данных.

Недостаток предложенного решения заключается в том, что решение при всей своей универсальности, имеет ограничения по архитектурной составляющей. Необходимость внесения изменений в архитектуру эволюционного процессора

(например, переход на принципиально новые ГП [2, 12]) приводит к необходимости доработки существующего решения под требуемые изменения. Основная нагрузка при замене ГП ляжет на базовый процессор, т.к. в его работе должны будут учтены все необходимые настройки и алгоритмы работы новых генетических процессоров. Преимуществом предложенной архитектуры является не только возможность общего повышения быстродействия ЭА и их использование в автономных системах, но и возможность исследований в области динамического эволюционного моделирования, где посредством динамических изменений структуры и состава хромосомы возможно образование новых видов (представления решений), что требует существенной вычислительной мощности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Гудилов В.В.* Разработка и исследование алгоритмов эволюционного синтеза комбинационных схем. Диссертация. – Таганрог, 2007.
2. *Курейчик В.В.* Биоинспирированный поиск с использованием сценарного подхода // Известия ЮФУ. – 2010. – № 7 (108). – С. 7-13.
3. *Курейчик В.В., Курейчик В.М., Родзин С.И.* Теория эволюционных вычислений. – МПА: Физмалит, 2012. – 260 с.
4. *Угрюмов Е.* Цифровая схемотехника. – СПб.: БХВ-Петербург, 2000.
5. *Takashi I., Barry S.J., Etsuko O., Mitsuhiro Y.* Fitness function circuit/Patent number: US6185547. Application number: US19970909830 19970812.
6. *Гудилов В.В., Зинченко Л.А., Курейчик В.В., Курейчик В.М.* Устройство аппаратной реализации вероятностных генетических алгоритмов. Патент Российской Федерации на изобретение №2294561, заявка № 2005108760 от 28.03.2005.
7. *Stephen D. Brown, Robert J. Francis, Jonathat Rose, Zvonko G. Vranesic.* Field-Programmable Gate Arrays // Kluwer Academic Publishers, USA. – 1992. – 206 p.
8. *Гудилов В.В., Зинченко Л.А.* Аппаратная реализация вероятностных генетических алгоритмов с параллельным формированием хромосомы // Перспективные информационные технологии и интеллектуальные системы. – 2003. – № 4. – С. 34-38.
9. *Гудилов В.В., Зинченко Л.А.* Аппаратная реализация вероятностных алгоритмов эволюционного поиска // Известия ТРТУ. – 2003. – № 2 (31) – С. 209-215.
10. *Гудилов В.В., Курейчик В.М.* Методы повышения эффективности вероятностных алгоритмов, адаптированные к возможности динамических изменений параметров функционирования на примере аппаратной реализации генетического алгоритма UMDA // Перспективные информационные технологии и интеллектуальные системы. – 2005. – № 1. – С. 43-55.
11. *Гудилов В.В.* Аппаратная реализация генетических алгоритмов. КИИ – 2006. Десятая национальная конференция по искусственному интеллекту с международным участием. – М.: Физматлит, 2006. Т. 3. – С. 1013-1021.
12. *Зайцев А.А., Курейчик В.В., Полупанов А.А.* Обзор эволюционных методов оптимизации на основе роевого интеллекта // Известия ЮФУ. – 2010. – № 12 (113). – С. 7-12.

Статью рекомендовал к опубликованию д.т.н., профессор В.В. Курейчик.

Гудилов Виталий Витальевич – ООО "Ямал", г. Ростов-на-Дону; e-mail: vgudilov@mail.ru; 344000, г. Ростов-на-Дону, ул. Шаумяна 48, оф. 2; тел.: +79198781600; к.т.н.; руководитель отдела информационных технологий.

Gudilov Vitaly Vital'evich – Ltd. "Yamal", Rostov-on-Don; e-mail: vgudilov@mail.ru; 48, Shaumyana street, of. 2, Rostov-on-Don, 344000, Russia; phone: +79198781600; cand. of eng. sc.; head of information technology.