

УДК 004.8.023 + 004.81.85

С.И. Родзин

**ОРГАНИЗАЦИЯ ПАРАЛЛЕЛЬНЫХ ЭВОЛЮЦИОННЫХ ВЫЧИСЛЕНИЙ\***

*Рассматриваются возможности распараллеливания эволюционных вычислений как процесса математических преобразований, позволяющих трансформировать входной поток информации в выходной по правилам, основанным на имитации механизмов эволюционного синтеза, статистическом подходе к исследованию ситуаций и итерационном приближении к искомому решению. Анализируются операции базового цикла алгоритмов эволюционных вычислений. Предлагаются различные модели организации параллелизма: глобальный параллелизм, миграционная и диффузионная модели. Приводятся рекомендации по распараллеливанию операторов базового цикла эволюционных вычислений. Отмечается важное значение дальнейших исследований в области организации параллельных эволюционных вычислений в гетерогенных компьютерах сетях, на параллельных высокопроизводительных кластерных вычислителях может. Это обеспечит значительные выгоды с точки зрения производительности и масштабируемости вычислений.*

*Генотип; фенотип; эволюционные вычисления, параллелизм.*

S.I. Rodzin

**ORGANIZATION OF PARALLEL EVOLUTIONARY COMPUTATION**

*The possibility of parallel evolutionary computation as a process of mathematical transformations which allow to transform the input stream to an output of rules based on imitation mechanisms of the evolutionary synthesis, a statistical approach to the study of situations and iterative approximation to the desired solution. Analyzes the operation of basic cycle of evolutionary algorithms for computing. Various models for the organization of parallelism: a global concurrency, migration and diffusion model. Provides guidelines for operators paralleling the basic cycle of evolutionary algorithms. Indicates the importance of further research in the field of parallel evolutionary algorithms in heterogeneous computer networks, parallel high-performance cluster calculators can. This will provide significant benefits in terms of performance and scalability of algorithms.*

*Genotype, phenotype; evolutionary computation; parallelism.*

**Введение.** Для решения разнообразных задач функциональной оптимизации, поддержки принятия решений все чаще стали использоваться алгоритмы и компьютерные программы эволюционных вычислений (ЭВ). В течение последних 50 лет были предложены разнообразные алгоритмы ЭВ: генетический алгоритм, генетическое программирование, эволюционные стратегии, эволюционное программирование, модели коллективного поведения (колонии муравьев, стаи птиц, пчелиного роя), модели отжига, табуированный поиск, меметика и другие эвристики, инспирированные природными системами. Разработана общая теория ЭВ [1].

Однако в области ЭВ имеются теоретические и практические проблемы, которые предстоит решать и которые в определенной мере затронуты в данной работе – это необходимость повышения точности и сокращения затрат вычислительных ресурсов на решение задач с помощью эволюционных алгоритмов.

**Эволюционные вычисления.** Эволюционные вычисления – это процесс математических преобразований, позволяющих трансформировать входной поток информации в выходной по правилам, основанным на имитации механизмов эволюционного синтеза, статистическом подходе к исследованию ситуаций и итерационном

\* Работа выполнена при частичной поддержке РФФИ (проекты: № 10-01-90017-Бел, № 11-07-00094, № 09-01-00492).

приближении к искомому решению. Основными элементами этого процесса является композиция эволюционных операторов «генерирующих» постоянную активность поисковых переменных в процессе эволюции; селективный отбор результатов и вычисление целевой функции «полезности»; наличие «памяти», реализующей свойство адаптивности поискового алгоритма; наличие гранул параллелизма [1,3].

Из этих четырех элементов эволюционных вычислений Ч. Дарвин гениально «угадал» первые два: ввел фундаментальное понятие «изменчивости» (поисковая активность алгоритма через композицию эволюционных операторов), а также «естественного отбора» (селекция) и «полезности», «выгодности», «приспособленности» (прообраз целевой функции).

Формально процесс ЭВ выглядит следующим образом. Фиксируется множество объектов  $X$  (популяция решений), обладающих некоторыми параметрами и связанных друг с другом посредством определенной структуры. Среди этих объектов необходимо выбрать наилучшие в смысле некоторого критерия качества (оптимальности)  $F$ . Критерий оптимальности формируется на основе свойств объектов и не обязательно существует в виде аналитических выражений. Важно, что существует отображение  $F: X \rightarrow R$ , где  $R$  – множество действительных чисел и каждому объекту  $x \in X$  из множества  $X$  сопоставляется значение критерия  $F(x)$ .

Фенотипическая природа исследуемого множества объектов произвольна, поэтому необходимо построить кодированное представление исходного множества объектов в другом, конечном множестве, обладающем структурой, например, векторного пространства  $S$  (генотип). Отображение вида  $\varphi: X \rightarrow S$  описывает связь между исследуемыми объектами, которые выступают в качестве потенциальных решений задачи поиска и объектами, управление и манипулирование которыми осуществляет алгоритм ЭВ. Существует обратное отображение вида  $\varphi^{-1}: S \rightarrow X$  и каждому вновь сгенерированному элементу представления  $s \in S$  соответствует элемент во множестве  $X$ .

Тогда, например процесс оптимизации с помощью алгоритма ЭВ, состоит в построении множества объектов-решений  $X_{opt} \in X$ , для которых выполняются следующие условия:  $X_{opt} = \operatorname{argmax} F[\varphi^{-1}(s)], s \in S$ .

Таким образом, в процессе оптимизации множество  $X$  развивается и эволюционирует к оптимальному состоянию, изменяя свой состав и параметры входящих в него объектов. Способ построения множества объектов  $s \in S$  определяется алгоритмом ЭВ.

Особенностью моделей ЭВ является то, что в качестве множества  $S$  строится так называемое кодовое пространство – множество представлений объектов  $x$  в виде кодов (хромосом). Эволюция множества  $X$  задается эволюцией представления  $S$ . На множестве  $S$  определяется подмножество  $P_0$  – случайная начальная популяция. Решение на каждом шаге эволюции определяется следующей разностной вычислительной схемой  $P_{t+1} = A(P_t)$ , где  $A$  – композиция различных эволюционных операторов. Критерий оптимальности вычисляется на каждом шаге вместе с критерием отбора решений, реализуемым в композиции операторов  $A$ .

**Базовый цикл эволюционных вычислений.** Одним из следствий общей теории ЭВ является наличие *базового цикла* алгоритма ЭВ, который включает следующую последовательность шагов: вычисление целевой функции, оценку качества решений, селективный отбор решений для репродукции и репродукцию, т.е. создание новых решений. Условиями остановки алгоритма ЭВ обычно является нахождение глобального, либо субоптимального решения, выход на «плато», исчерпание числа поколений, отпущенных на эволюцию или исчерпание времени, отпущенного на эволюцию.

Важными элементами эволюционных алгоритмов являются представление (кодирование) решений, а также правила, по которым эволюционирует реализуемая алгоритмом модель. Вначале фиксируется популяция  $P_0$  решений (подмножество объектов  $X$ ), обладающих некоторыми фенотипическими свойствами и связанных друг с другом посредством определенной структуры. На основе свойств объектов формируется критерий  $F$  (не обязательно аналитическое выражение), согласно которому среди объектов выбираются наилучшие:

$$X_{opt} = \operatorname{argmax} F [\varphi^{-1}(s)], s \in S,$$

где в качестве множества  $S$  строится так называемое генотипическое кодовое пространство – множество представлений объектов  $x$  в виде кодов (хромосом)  $s$ . Эволюция множества  $X$  задается эволюцией представления  $S$ . Таким образом, представление генотипа  $S$  и отображение генотип – фенотип  $\varphi^{-1}: S \rightarrow X$  имеет важное значение для организации работы ЭА. Поясним это положение теории эволюционных вычислений.

Предполагается, что целью задачи поиска является нахождение объекта с некоторыми свойствами. Как правило, поиск производится среди конечного (иногда бесконечного) множества объектов (потенциальных решений). Первый шаг при решении задачи поиска состоит в том, чтобы определиться относительно пространства исследуемых объектов множества  $X$ . В качестве этого пространства могут фигурировать пространство  $n$ -мерных векторов вещественных чисел, множество позиций в игре или множество вариантов размещения и трассировки микросхемы.

Второй шаг, который должен предшествовать процедуре поиска состоит в выборе некоторого представления объектов из пространства  $X$ . Представление определяется множеством  $S$  – пространством представлений. Множество  $S$  выбирается с таким расчетом, что алгоритму поиска будет легче манипулировать с элементами множества  $S$ , нежели  $X$ .

В отличие от пространства объектов  $X$ , пространство представлений  $S$  обязательно конечно. Таким образом, отображение  $\varphi^{-1}: X \rightarrow S$  есть функция кодирования, которая описывает связь между исследуемыми объектами, выступающими в качестве потенциальных решений задачи поиска, и объектами, управлением и манипулированием которых занимается ЭА. Обратное отношение (функция декодирования)  $\varphi^{-1}: S \rightarrow X$  используется тогда, когда по новому представлению, полученному в результате поиска, требуется восстановить соответствующее ему решение из  $X$ .

Использование представлений решения позволяет осуществлять поиск практически при минимуме информации о характере и свойствах пространства объектов. Достаточно только той информации, которая позволяет описать ландшафт критериальной функции в пространстве представлений.

Для реализации алгоритма поиска в пространстве представлений вводится целевая функция оценки представлений  $F^{-1}$ , аналогичная функции  $F$ , определенная на элементах из множества  $S$ . Иногда, в зависимости от конкретных операторов алгоритма поиска, бывает необходимо, чтобы целевая функция принимала положительные значения.

Таким образом, прежде чем построить поисковый алгоритм, включающий композицию генетических операторов  $A(P_i)$ , необходимо определиться с символической моделью задачи. Она включает в себя пространство потенциальных решений  $X$ , пространство представлений  $S$ , функции кодирования  $\varphi$  и декодирования  $\varphi^{-1}$ , функцию качества (пригодности, приспособленности)  $F$  и целевую функцию оценки представлений  $F^{-1}$ .

Эволюционные алгоритмы в процессе своей работы требуют немалых вычислительных ресурсов, поэтому актуальной является проблема переноса ЭВ в распределенную вычислительную среду. Для этого имеются веские основания:

алгоритмы ЭВ потенциально обладают свойством массового параллелизма при обработке информации, что допускает их эффективную параллельную реализацию [2]. Организация распараллеливания ЭВ на базе распределенной вычислительной системы позволяет существенно сократить время, затрачиваемое на решение задачи, как за счет параллельного выполнения вычислений, так и за счет применения более эффективных, чем в последовательном случае, способов реализации алгоритмов.

**Параллельные эволюционные вычисления.** Создание многоядерных структур процессоров, недорогих кластерных решений на основе стандартных компонентов, массовое использование распределенных вычислений отражают тенденцию к параллелизму в различных его проявлениях. Необходимость в параллелизме объясняется существованием задач, требующих большого количества ресурсов памяти, и процессорного времени, а также тем обстоятельством, что индустрия упёрлась в технологический барьер и дальнейшее увеличение тактовой частоты затруднено.

Структура базового цикла ЭВ обладает параллелизмом, как на уровне организации работы алгоритма (эволюция в природе «обсчитывается» на гигантской параллельной машине, состоящей из всех природных объектов), так и на уровне его компьютерной реализации.

Параллелизм на уровне компьютерной реализации означает вычисление на параллельных системах или процессорах значений целевой функции для разных решений, параллельное выполнение операторов ЭВ, что пропорционально повышает скорость работы алгоритма. Параллелизм на уровне организации работы алгоритма ЭВ может достигаться структурированием популяции решений. Оба подхода могут эффективно реализоваться на параллельных вычислителях, причем способы организации параллелизма могут быть различными: глобальный параллелизм, миграционная модель, диффузионная модель [1].

Рассмотрим модель глобального параллелизма эволюционных вычислений, представленную на рис. 1.

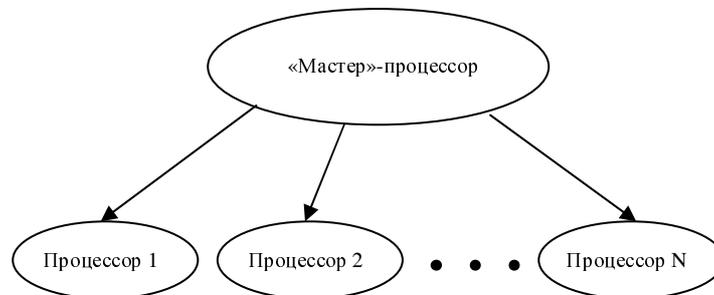


Рис. 1. Модель глобального параллелизма

В этой модели основной гранулой параллелизма или участком эволюционного алгоритма, выполняющимся на отдельном процессоре и не требующим при своём исполнении данных от других процессоров, являются такие операции базового цикла эволюционных вычислений как расчёт целевой функции и параллельное выполнение операторов репродукции новых решений. Целевая функция отдельного индивидуума в популяции, как правило, не зависит от других индивидуумов. Поэтому вычисление целевых функций допускает распараллеливание. При этом популяция хранится в общей памяти, отдельный процессор считывает код решения (например, хромосому в генетическом алгоритме) из памяти и возвращает результат после вычисления целевой функции. Процедура синхронизации здесь про-

водится лишь при формировании новой популяции. В случае если вычислительная система имеет распределенную память, то популяция и значения функции качества отдельных индивидуумов хранятся в «мастер»-процессоре, а расчеты производятся на вспомогательных процессорах.

При использовании модели глобального параллелизма трудность для распараллеливания при реализации базового цикла эволюционных вычислений представляет процедура селективного отбора решений. Здесь преимущество в смысле распараллеливания получают те формы селекции, которые не используют глобальную статистику о популяции, например «соревновательные» способы селекции [1]. Напротив, легко распараллеливаемыми являются такие наиболее широко применяемые операторы эволюции, как мутация и кроссинговер, так как они выполняются независимо друг от друга и в основном случайным образом.

Если между отдельными подпопуляциями установить правила миграции отдельных индивидуумов, то получится миграционная модель, показанная на рис. 2.

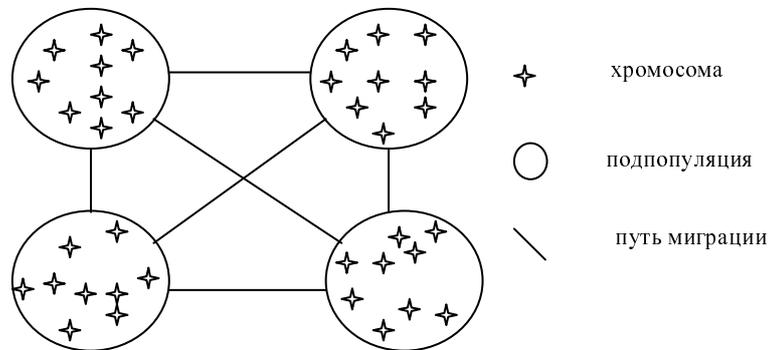


Рис. 2. Миграционная модель параллелизма

В этой модели основной гранулой параллелизма являются все операции базового цикла эволюционных вычислений, выполняемые внутри подпопуляции. Для построения миграционной модели требуется задать количество и размеры подпопуляций, топологию связей между ними, частоту и интервал миграции, а также стратегию эмиграции и иммиграции. Размеры и число подпопуляций определяются числом и быстродействием используемых процессоров или транзьютеров. Что касается топологии связей в миграционной модели, то можно применять различные варианты: «островную» модель, предусматривающую миграцию между любыми подпопуляциями; модель, предусматривающую обмен решениями между соседними подпопуляциями и др. Применение того или иного варианта миграционной модели зависит от архитектуры параллельного вычислителя (сетевая, кольцевая или др.). Частота миграции определяется размерами подпопуляций. При высокой частоте подпопуляция может быстро потерять гетерогенность. Наоборот, при слабой интенсивности миграции и большом числе подпопуляций возрастает вероятность получения нескольких хороших решений, однако сходимость алгоритма замедляется.

Для обеспечения выбора особей для миграции предлагается использовать общий буфер хромосом. Буфер заполняется хромосомами нескольких популяций. При наступлении определенной ситуации отдельная популяция обращается к буферу и считывает оттуда часть или все хромосомы, добавляя в буфер часть своих хромосом. В ходе ЭВ необходимо лишь определить механизм регулирования размера буфера. Работа параллельного эволюционного алгоритма при этом происходит следующим образом. Каждая популяция эволюционирует отдельно от других.

На каждой итерации проверяется условие необходимости миграции (через заданный интервал, вырождение популяции и т.п.). Если условие наступило, происходит миграция хромосом через общий буфер. Затем проверяется размер буфера. Если размер больше заданного значения, то выполняется процедура отбора.

Отметим, что условия миграции могут быть разными для разных популяций. Также различными могут быть композиции генетических операторов для разных популяций.

Миграционная модель эффективно реализуется в вычислительных системах с MIMD-архитектурой (множественный поток команд и множественный поток данных) путем разделения общей популяции на отдельные подпопуляции, эволюционное моделирование которых осуществляется на отдельных процессорах.

Эволюционные вычисления стохастичны, поэтому при разных запусках эволюционного алгоритма популяция может сходиться к разным субоптимальным решениям. Данная модель позволяет запустить алгоритм сразу несколько раз и совместить «достижения» разных «островов» для получения наилучшего решения.

Если большую популяцию разделить на множество немногочисленных подпопуляций, а композицию эволюционных операторов применять лишь в ограниченной области, определяемой отношением соседства, то получится диффузионная модель эволюционных вычислений, представленная на рис. 3.

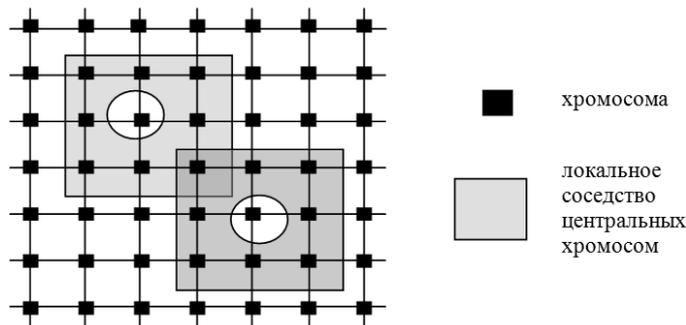


Рис. 3. Диффузионная модель параллелизма

В этой модели основной гранулой параллелизма являются все операции базового цикла эволюционных вычислений, выполняемые внутри небольших по размеру подпопуляций. Это приводит к тому, что хорошие решения медленно распространяются в популяции (диффузия) и проблема попадания в «локальную яму» становится менее острой.

Популяция в природе, как правило, состоит из отдельных, более или менее изолированных локальных подпопуляций, допускающих диффузию отдельных индивидуумов. Большинство существующих эволюционных алгоритмов это обстоятельство игнорируется, алгоритмы работают с одной большой неструктурированной популяцией.

Другое преимущество диффузионной модели состоит в том, что она также допускает эффективную реализацию эволюционных вычислений на массиве параллельных вычислителей с различной коммутационной структурой.

Однако наиболее эффективно диффузионная модель реализуется в вычислительных системах с SIMD-архитектурой (одиночный поток команд, множественный поток данных). Эффективность того или иного способа организации распределенных ЭВ зависит от архитектуры вычислительной системы, используемой для реализации ЭВ.

Из анализа базового цикла алгоритма ЭВ вытекают следующие рекомендации по распараллеливанию, входящих в него операций:

- ◆ две составляющие базового цикла алгоритма ЭВ потенциально могут значительно повлиять на параллелизм – оценка качества решения и репродукция новых решений;
- ◆ процесс вычисления целевой функции для каждого решения может осуществляться параллельно и независимо от остальных решений;
- ◆ оценка качества решений может потребовать сложных расчетов или моделирования и является одним из наиболее трудоемких этапов ЭВ. В процессе оценки качества требуется сравнить значения функции качества каждого решения с остальными и обновить структуры данных. Это затрудняет процесс распараллеливания;
- ◆ выполнение фазы селекции в базовом цикле алгоритма ЭВ может потребовать доступа к определенным подмножествам решений в популяции или обновления данных. Поэтому возможности ее распараллеливания зависят от выбора схемы селекции;
- ◆ выполнение фазы репродукции в базовом цикле алгоритма ЭВ нетрудно распараллелить. Репродукция, являясь независимой задачей, может включать либо копирование решения, либо выполнение унарных, бинарных или n-арных эволюционных операторов.

Эксперименты, проведенные в различных вычислительных средах показывают следующее. Поскольку объём популяции решений может быть значительно больше, чем число имеющихся процессоров, то один общий поток заданий распределяется по процессорам, что приводит к образованию очереди заданий и необходимости организации процессов их обслуживания. Такой подход по эффективности может даже уступать обработке заданий на однопроцессорной машине.

Распределенный алгоритм будет эффективным только в том случае, если задержки, вызванные передачей и обменом необходимыми данными, будут незначительными.

Наиболее простым подходом к организации распределенных ЭВ представляется глобальный параллелизм и клиент-серверная схема его реализации. Роль клиента выполняет мастер-процессор, реализующий выполнение таких шагов базового цикла ЭВ как оценка качества решений и селекция. Серверы выполняют вычисление целевой функции и репродукцию и могут включать несколько ядер. Однако экспериментальное решение некоторых проектных задач оптимизации и структурного синтеза большой размерности показали, что клиент-серверная архитектура может оказаться не оптимальной. Это происходит в следующих случаях:

- ◆ в процессе решения задачи время, необходимое для передачи и обмена данными между клиентом и серверами значительно превышает время, необходимое серверу для выполнения своих задач;
- ◆ представление решений и объем популяции требует значительной памяти, поэтому организовывать работу распределенного алгоритма ЭВ желательно на основе миграционной или диффузионной модели параллелизма.

Используя различные алгоритмы ЭВ можно моделировать эволюцию в «островных популяциях» на различных узлах в сети компьютеров, каждый из которых представляет один «остров». Связь между узлами понадобится лишь для миграции между «островами». Эта связь может осуществляться асинхронно через сеть самостоятельно работающих алгоритмов ЭВ. Правила миграции могут быть выбраны таким образом, чтобы оптимизировать сетевой трафик. Память, требуемая для организации ЭВ на одной машине, уменьшается, каждый узел работает независимо.

Однако простота идеи программирования параллельных эволюционных вычислений на практике требует выполнения некоторых условий. В частности, в многопроцессорной системе разбиение на слишком крупные гранулы не позволяет равномерно загрузить процессоры и добиться минимального времени вычислений, а излишне мелкая «нарезка» означает рост непроизводительных расходов на связь и синхронизацию. Параллелизму присуща зависимость структуры алгоритма от топологии вычислительной платформы. Процесс создания максимально эффективного алгоритма практически не автоматизируется и связан с большими трудозатратами на поиск специфической структуры алгоритма, оптимальной для конкретной топологии используемой многопроцессорной системы. Найденная структура обеспечит минимальное время получения результата, но, скорее всего, будет неэффективна для другой конфигурации.

**Заключение.** Организация параллельных ЭВ в гетерогенных компьютерах сетях или на параллельных высокопроизводительных кластерных вычислителях может обеспечить значительные выгоды с точки зрения производительности и масштабируемости. Для этого имеются веские основания. Все разновидности эволюционных алгоритмов обладают свойством параллелизма и предоставляют полноценные возможности для нахождения оптимальных решений. Такие свойства у них появляются вследствие наличия механизмов поддержания разнообразия в популяции на протяжении всего поиска в распределенном пространстве решений. Практическая реализация параллелизма ЭВ позволит всесторонне исследовать пространство решений, что, в частности, является ключевым моментом для решения многих трудных проблем из различных прикладных областей

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Курейчик В.М., Курейчик В.В., Родзин С.И. Основы теории эволюционных вычислений. – Ростов-на-Дону: Изд-во ЮФУ, 2010. – 224 с.
2. Родзин С.И. Эволюционные вычисления: единая теория и организация параллелизма // Нечеткие системы и мягкие вычисления: сб. ст. Третьей Всероссийской научной конференции: В 2-х т. Т. I. – Волгоград: Изд-во ВолгГТУ, 2009. – С. 30-40.
3. Курейчик В.В., Курейчик В.М., Родзин С.И. Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-25.

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

#### **Родзин Сергей Иванович**

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

E-mail: rodzin@mopevm.tsure.ru.

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

Тел.: 88634371673.

Кафедра математического обеспечения и применения ЭВМ; профессор.

#### **Rodzin Sergey Ivanovich**

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

E-mail: rodzin@mopevm.tsure.ru.

44, Nekrasovskiy, Taganrog, 347928, Russia.

Phone: +78634371673.

The Department of Software Engineering; Professor.