ИНФОРМАТИКА

2006 **апрель-июнь** **№ 2**

УДК 004.722.25

Ю.И. Воротницкий, Д.А. Стрикелев

ПРОЕКТИРОВАНИЕ СЕТЕЙ НА ОСНОВЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ, МОДИФИЦИРУЮЩИХ СТРУКТУРУ ХРОМОСОМ

Рассматриваются специфика решения задач оптимального проектирования сетей средствами генетических алгоритмов и причины, снижающие эффективность их применения. Предлагается модифицированная версия генетического алгоритма для синтеза плотносвязных образующих блоков и показывается ее эффективность для размещения в сети реплицируемого ресурса.

Введение

Задачи проектирования информационных сетей и оптимального размещения ресурсов в них привлекают внимание исследователей на протяжении последних лет [1–4]. Существенной проблемой при решении подобных задач является высокая размерность пространства решений получаемых моделей, что затрудняет применение классических методов дискретной оптимизации и вынуждает вести разработку новых подходов к решению. Одним из таких подходов является использование генетических алгоритмов (ГА) [5].

1. Решение оптимизационных задач средствами генетических алгоритмов

ГА являются поисковыми процедурами, основанными на механизмах естественного отбора и генетики. В процессе вычислений ГА оперируют не одним решением, а целым их множеством – *популяцией.* Каждое решение, называемое *представителем* или *индивидом,* может быть любой комбинацией значений переменных задачи, заданных в структуре данных, которая известна также как *хромосома.* Мера оптимальности каждого из решений (фитнес) является скалярной величиной и оценивается специально определенной целевой функцией.

Хромосома состоит из последовательности генов (представляющих переменные задачи), которые могут как вносить вклад в целевую функцию (в этом случае ген является активным), так и не влиять на нее (в этом случае ген является пассивным). Каждый ген может принимать значения только из фиксированного множества (аллеля), при этом выбранное значение называется *выраженным.* Комбинация некоторого количества генов, положительно влияющая на целевую функцию при условии, что любая комбинация меньшего числа генов не влияет или влияет отрицательно, называется *образующим блоком* (ОБ). ОБ характеризуется такими свойствами, как размер (количество входящих в него генов) и диаметр (расстояние между крайними генами), при этом если диаметр сопоставим с размером, то ОБ называют компактным, в противном случае – разреженным.

В пространстве решений осуществляется поиск с целью нахождения глобального экстремального значения целевой функции. Процесс поиска, называемый также *эволюцией,* производится на основе выбранных правил (операторов): оператора бридинга (ОПБ), отбирающего индивидов для скрещивания; оператора кроссовера (ОПК), обеспечивающего обмен фрагментами хромосом среди выбранных членов популяции; оператора мутации (ОПМ), поддерживающего разнородность значений генов и удерживающего популяцию от досрочного схождения к локальному экстремуму; оператора селекции (ОПС), отвечающего за поддержание размера популяции на постоянном уровне и не позволяющего ей бесконтрольно разрастаться.

Тем не менее, несмотря на то, что в целом при решении задач оптимизации ГА зарекомендовали себя успешно, эффективность их применения в значительной степени снижается из-за большого размера (до 10 % длины хромосомы) и диаметра (сопоставимого с длиной хромосомы) ОБ [5], а также нестабильной сходимости процессов их обнаружения и выделения.

В случае применения базового ГА (Simple Genetic Algorithm – SGA), т. е. при использовании стандартной структуры хромосомы с фиксированным размещением генов (рис. 1) и N-точечного ОПК, разреженные ОБ с высокой вероятностью разбиваются на части точкой разреза

в процессе скрещивания и не наследуются потомками, результатом чего является ухудшение фитнеса популяции в процессе эволюции. Отсутствие на этапе разработки структуры хромосомы информации о связности генов также обусловливает замедленность и неустойчивость синтеза ОБ из разрозненных фрагментов. В результате, если ни один из индивидов начальной популяции не обладал целым ОБ, то вероятность его образования в процессе эволюции невысока.

Рис. 1. Структура бинарной хромосомы с позиционным выражением генов (гены идентифицируются по своему положению в хромосоме, для каждого гена присутствует только одно значение из всего аллеля – 0 или 1, которое и является выраженным)

Для решения задач с неизвестным порядком следования генов в общем случае успешно применяется модификация ГА, изучающая связность элементов ОБ в ходе своей работы (Linkage Learning Genetic Algorithm – LLGA) [6]. Тем не менее, случаев применения LLGA к задачам оптимального проектирования сетей авторам неизвестно. Таким образом, целью данного исследования является изучение применимости LLGA для решения задач оптимизации компьютерных сетей.

2. Специфика задач оптимизации компьютерных сетей

Модели реальных сетей и информационных ресурсов характеризуются тесными связями между переменными модели. Например, реализация решения задачи о размещении СУБД в локальной сети увеличит сетевой трафик за счет запросов, поступающих на обработку. Результатом станет увеличение времени доступа к другим ресурсам, находящимся в этой же сети и не связанным с СУБД. В свою очередь, это может потребовать перемещения ряда ресурсов в новые места и изменения схем движения сетевого трафика. Другими словами, ОБ, формирующие оптимальные решения реальных задач, в подавляющем большинстве случаев состоят из генов, расположенных в различных частях хромосомы, и одним из наиболее действенных способов повысить эффективность применения ГА является обнаружение связей между генами и перегруппировка их в плотные формации, т. е. построение ОБ малого размера и диаметра.

3. Модифицированный генетический алгоритм, повышающий связность образующих блоков

При использовании LLGA вместо фиксированного позиционирования генов применяется механизм вероятностного выражения (probabilistic expression – PE), хромосома имеет кольцевую форму и в ней всегда присутствуют все возможные значения аллеля гена (рис. 2). Из различных значений аллеля, присутствующих в кольцевой хромосоме, выраженным считается то, которое встречается первым при обходе хромосомы по кругу. Для каждого гена допускается присутствие только одной копии выраженного значения аллеля и N копий каждого из невыраженных значений. Подобная схема контроля количества генов и выражения их значений определяется как PE-N. Очевидно, что именно выбор стартовой точки отсчета (точки интерпретации), от которой начинается обход хромосомы, является основополагающим.

Для изучения связности генов в структуру хромосомы внедрены *интроны*, т. е. пассивные гены, не влияющие на фитнес, но участвующие в формировании ее структуры. Роль интронов заключается в том, что они резервируют свободные места в хромосоме, которые затем используются при перестановке генов.

При скрещивании двух индивидов у донора случайным образом выбираются две точки, называемые точками экстракции, а у акцептора случайно выбирается точка инжекции. Участок хромосомы донора, находящийся между точками экстракции, вставляется в хромосому акцептора начиная с точки инжекции, которая при этом становится точкой интерпретации акцептора. Так как внедрение дополнительного генетического материала может привести к нарушению правила PE-N, после каждого скрещивания необходимо проверять количество выраженных и невыраженных генов, а также количество интронов.

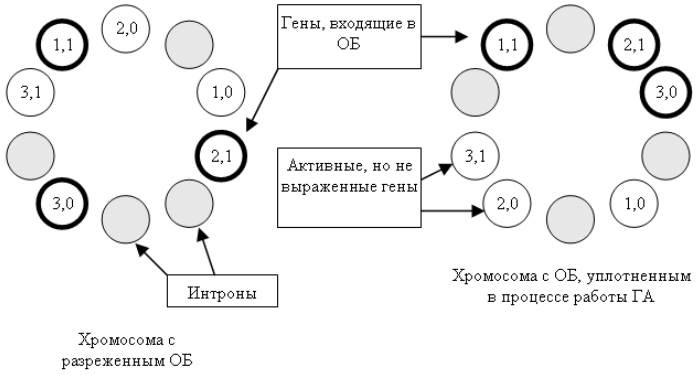

Рис. 2. Структура бинарной хромосомы с вероятностным выражением генов (гены идентифицируются по дополнительному признаку – первому числу пары, в хромосоме представлен полный набор значений аллеля – второе число пары, выраженным является первое по порядку значение от точки интерпретации хромосомы)

Задачи проектирования сетей характеризуются большим количеством переменных (до нескольких тысяч), поэтому для обеспечения возможности одновременного формирования нескольких ОБ в различных частях хромосомы необходимо внедрение значительного числа интронов (соотношение между числом генов и интронов 1:2 и более). В этом случае в целях экономии оперативной памяти применяется компрессия хромосом [7]. Так как значения интронов не оказывают влияния на фитнес индивида, то хранение их значений не имеет смысла, интерес представляет только количество интронов в непрерывной цепочке. Это позволяет производить сжатие, представляя непрерывную цепочку интронов в виде счетчика повторений. Следует заметить, что использование сжатых интронов экономит оперативную память, но приводит к дополнительным вычислительным затратам на распаковку-упаковку в процессе скрещивания.

Согласно работе [5] цикл SGA выглядит следующим образом:

1. Сформировать начальную популяцию.
2. В цикле на N поколений:
    - 2.1. Отобрать особей, подлежащих скрещиванию.
    - 2.2. Сформировать среди отобранных особей пары.
    - 2.3. Произвести скрещивание:
        - 2.3.1. Случайным образом выбрать точки разрезания хромосом предков.
        - 2.3.2. Произвести обмен фрагментами хромосом предков.
    - 2.4. Произвести мутацию популяции.
    - 2.5. Произвести селекцию внутри разросшейся популяции.
3. Завершить эволюцию.

Использование LLGA, в отличие от SGA, требует контроля правильности индивидов по схеме PE-N после каждого скрещивания, а также устраняет необходимость мутации как отдельного этапа, так как приведение хромосомы к виду PE-N связано с ее случайным изменением и может рассматриваться как механизм скрытой мутации. Как отмечается в работе [7], в подавляющем большинстве случаев генетический материал донора внедряется в потомка неизменным, а избыточные гены и интроны удаляются из хромосомы акцептора. Это означает, что проверке по схеме PE-N подлежит только остаточный фрагмент хромосомы индивида-акцептора. Поэтому весь цикл LLGA можно представить следующим образом:

1. Сформировать начальную популяцию.
2. В цикле на N поколений:
   2.1. Отобрать особей, подлежащих скрещиванию.
   2.2. Сформировать среди отобранных особей пары.
   2.3. Произвести скрещивание:
      2.3.1. В каждом из членов пары случайным образом выбрать точки инжекции и экстракции, которые обязательно должны прийтись на интрон.
      2.3.2. Скопировать индивиды родителей, получив два индивида потомков.
      2.3.3. Рассматривая одного родителя как донора, инжектировать нетронутым его фрагмент хромосомы в потомка другого родителя (акцептора).
      2.3.4. Проверить правильность акцептора по схеме PE-N, при необходимости удалить избыточные гены и интроны.
      2.3.5. Повторить процедуру, рассматривая в качестве донора другого родителя, а в качестве акцептора – другого потомка.
   2.4. Произвести селекцию внутри разросшейся популяции.
3. Завершить эволюцию.

Несмотря на различия в алгоритме эволюции, LLGA, как и SGA, оперирует линейной структурой данных – хромосомой. Сетевая топология по своему определению является связной и иерархической структурой. Это означает, что для успешного применения ГА к задачам сетевой оптимизации необходимо сохранять сведения о сетевой топологии в хромосоме.

4. Задание иерархической сетевой топологии в линейной структуре хромосомы

Основным типом сетевой топологии в настоящее время является иерархическая «звезда», поэтому для указания местоположения и взаимосвязей узлов в сети возможна их маркировка по схеме виртуального дерева Хаффмана (рис. 3) [8]. Все вершины этого дерева имеют метки, представляющие собой полный путь от корня (обозначенного меткой λ). В пределах одного уровня иерархии метки вершин назначаются по порядку слева направо (1, 2, 3, …).

В полном дереве глубины $\lfloor p/2 \rfloor$ (целая часть от $p/2$) каждый узел имеет $p - 1$ дочерних узлов. Общее количество вершин дерева определяется как

$$\frac{(p-1)^{\lfloor p/2 \rfloor}-1}{p-2}.$$

На этом дереве любой узел может быть легко идентифицирован по своей метке. Так как длина метки равна глубине узла, обозначенного данной меткой, наибольшая метка будет содержать $\lfloor p/2 \rfloor$ символов. Для того чтобы расставить метки в таком дереве, необходимо задействовать $p - 1$ различных букв. Для дерева, изображенного на рис. 3, топология «звезда» задается набором меток $\{\lambda, 1, 2, 3, 4\}$. Топология «шина» задается набором $\{\lambda, 1, 11, 2, 23\}$.

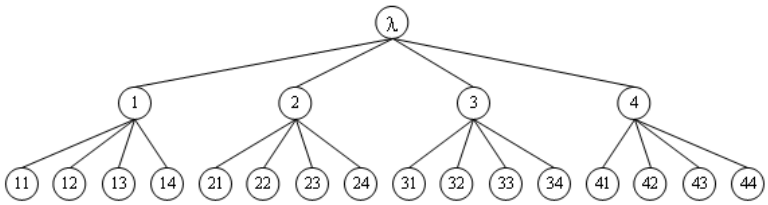

Рис. 3. Виртуальное дерево Хаффмана для p=5

ГА работают с хромосомами одинаковой длины, что накладывает требование равенства длин меток узлов, выполняемое заданием дополнительного «пустого» символа. Таким образом, используя виртуальное дерево Хаффмана, можно получить любую из возможных конфигураций иерархической локальной сети, метка каждого узла при этом будет являться путем к нему

от корня дерева. Для сети на рис. 3 хромосому можно записать как (00 λ), (01 λ), (02 λ), (03 λ), (04 λ), (11 λ), (12 λ), (13 λ), (14 λ), …, (44 λ).

Представление иерархической топологии сети в виде линейного списка позволяет перечислять узлы в произвольном порядке и группировать их в ОБ не согласно иерархическому соподчинению, а согласно интенсивности взаимодействия друг с другом. В случае применения SGA внутренняя структура хромосомы не анализировалась, поэтому возможность изменения расположения генов, описывающих содержимое сетевого узла, оставалась невостребованной. В случае же LLGA перестановка генов в хромосоме позволяет сгруппировать вместе те узлы, которые активно участвуют во взаимодействии друг с другом, таким образом вероятность их разделения ОПК уменьшается. Например, если узлы 11 и 41 интенсивно обмениваются данными, то хромосома в процессе работы LLGA изменится на схожую со следующей: (01 λ), (11 λ), (00 λ), (41 λ), (04 λ), … В то же время использование меток Хаффмана позволяет в любой момент восстановить иерархические связи между узлами, что необходимо на этапе вычисления фитнеса индивидов, так как взаимодействие между различными уровнями иерархии может характеризоваться различными количественными и качественными характеристиками.

5. Получение оптимальных по Парето решений многокритериальных задач

Большинство задач сетевого проектирования и оптимизации являются многокритериальными, поэтому соответствующим образом должна быть изменена и форма целевой функции. ГА позволяют эффективно учесть несколько критериев с помощью коррекции фитнеса индивидов по мере их сходства. Согласно этому подходу все множество возможных индивидов разбивается на несколько классов (ниш), при этом в каждый класс попадают те индивиды, значения критериев которых сходны (находятся в пределах некоторой области пространства решений). После группировки по нишам объективный фитнес индивида корректируется (снижается) в зависимости от степени наполнения ниши, тем самым стимулируя заполнение пустых ниш и поддержание разнообразия популяции. Данный подход позволяет избежать схождения решения к единственной точке и получить одновременно целое множество оптимальных по Парето решений [9]. Следует заметить, что целевая функция для каждой задачи оптимизации формулируется отдельно и невозможна ее запись в обобщенном виде.

6. Вычислительный эксперимент

Для эксперимента была выбрана задача размещения копий реплицируемого ресурса в иерархически организованной сети с целью минимизации стоимости поддержания копий ресурса в узлах и стоимости использования каналов передачи данных сети (рис. 4). Каналы передачи данных используются для обработки пользовательских запросов и синхронизации вторичных копий с центральной. Стоимость поддержания копии ресурса зависит от расположения узла.

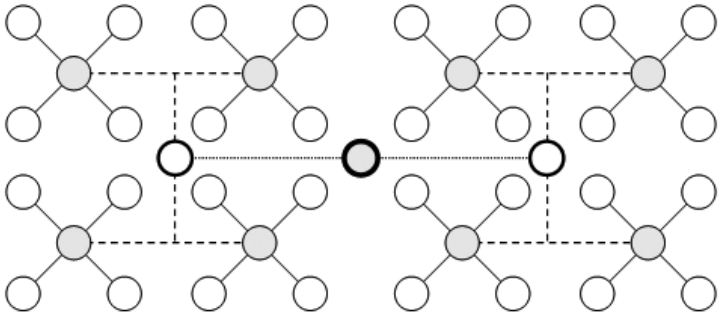

Рис. 4. Структура сети для оценки эффективности LLGA: серые кружки – центральные узлы рабочих групп, светлые кружки с обводкой – центральные узлы отделов, серый кружок с обводкой – корневой узел сети, сплошные линии имеют вес C_1, штриховые – C_2, пунктирные – C_3

В сети выделены два уровня иерархии – рабочие группы и отделы, узлы которых связаны тремя типами соединений:

рабочая станция – центральный узел рабочей группы, вес C1;

центральный узел рабочей группы – центральный узел отдела, вес C2;

центральный узел отдела – центральный узел сети, вес C3.

Выбранная структура сети удобна не только тем, что представляет наиболее часто используемую схему построения локальных сетей, но и тем, что путем изменения количества рабочих станций в группе (N_1), групп в отделе (N_2), отделов в сети (N_3) можно легко изменять общее количество узлов сети, а следовательно, размерность задачи.

При формулировке модели приняты следующие допущения:

1. К ресурсу должны иметь доступ все узлы сети, кроме центральных узлов рабочих групп и отделов, так как в их роли чаще всего выступает коммутационное оборудование.

2. Запрос на обслуживание отправляется ближайшей копии ресурса.

3. Все копии должны синхронизироваться с центральной для поддержания актуальности данных, «стоимость» синхронизации превышает стоимость обработки запроса в K_{sync} раз.

4. Стоимость передачи данных по каналам внутри рабочей группы мала, но возрастает при переходе на более высокие уровни (связи между рабочими группами и отделами), что отражает увеличивающуюся загрузку каналов по мере подъема от уровня рабочих станций к уровню центральных коммутаторов.

5. Интенсивности пользовательских запросов не рассматриваются, т. е. мощность вычислительного оборудования в узлах считается достаточной и в перенаправлении запросов к соседним узлам нет необходимости.

В общем случае задача размещения копий ресурса по узлам сети записывается в следующем виде:

$$\begin{cases} \sum\limits_{i \in V} C_i^{Support} \to \min; \\ \sum\limits_{j \in E} (C_j^{Transfer} + C_j^{Sync}) \to \min, \end{cases}$$

где V – множество узлов сети; $C_i^{Support}$ – стоимость эксплуатации i-го узла; E – множество ребер сети; $C_j^{Transfer}$ и C_j^{Sync} – стоимость эксплуатации j-го ребра сети с учетом количества запросов клиентов и синхронизации, проходящих через него.

Для проведения численного эксперимента на языке PHP были реализованы два вида ГА: SGA и LLGA, эффективность применения которых затем оценивалась в случае различных размеров сети и параметров ГА.

Для лучшего контроля элитизма популяции (управления тем, насколько быстро и агрессивно лучшие индивиды вытесняют менее приспособленные) [10] отбор особей для скрещивания и селекции был реализован турнирами.

В случае SGA использовались турнирный ОПБ с размером пула 2, одноточечный ОПК с вероятностью кроссовера 100 %, случайная мутация с вероятностью 2 %, турнирный ОПС с размером пула 4.

В случае LLGA соотношение активных генов к интронам было выбрано как 1:2, использовались турнирный ОПБ с размером пула 2, двухточечный ОПК с постоянным приведением хромосом к виду, стандартному по PE-2, элитный ОПС [6].

В ходе экспериментов рассчитывалось размещение копий ресурса в двух разных сетях (56 и 188 узлов соответственно). Расчеты производились при фиксированном количестве членов популяции (изменяющемся количестве поколений) и поколений (изменяющемся количестве членов популяции). Так как ГА являются вероятностными алгоритмами, то в каждом случае проводилось по 15 независимых расчетов модели, получаемые значения усреднялись. Вычисления проводились на ЭВМ Intel Celeron 1.7 GHz, RAM 256 MB, результаты приведены в таблице.

Таблица

Сравнительная характеристика качества решения и времени работы SGA и LLGA

Параметры алгоритма	SGA			LLGA		
	Время работы, с	Средний фитнес	Лучший фитнес	Время работы, с	Средний фитнес	Лучший фитнес
Сеть из 56 узлов						
30 особей, 35 поколений	16,0	282,2	252,0	191,4	246,5	220,0
50 особей, 35 поколений	25,1	257,4	228,0	318,4	230,0	214,0
80 особей, 35 поколений	39,9	234,4	214,0	577,6	224,8	214,0
50 особей, 20 поколений	14,5	255,4	220,0	188,4	255,6	226,0
50 особей, 35 поколений	25,1	257,4	228,0	318,4	230,0	214,0
50 особей, 50 поколений	38,6	258,5	218,0	457,8	223,2	210,0
Сеть из 188 узлов						
90 особей, 35 поколений	469,7	1215,0	1184,0	2476,0	1097,0	1080,0
150 особей, 35 поколений	-	-	-	4961,0	1082,0	1053,0
240 особей, 35 поколений	-	-	-	6312,0	1046,0	1024,0
150 особей, 20 поколений	-	-	-	2253,0	1130,0	1096,0
150 особей, 35 поколений	-	-	-	4961,0	1082,0	1053,0
150 особей, 50 поколений	-	-	-	5492,0	1039,0	1028,0

Априорно правильным решением задачи является размещение копий ресурса во всех коммутационных узлах (корневом узле сети, центральных узлах отделов и рабочих групп). Решения, найденные LLGA, всегда содержат копии ресурса в корневом узле сети, центральных узлах отделов, а также более чем в 90 % центральных узлов рабочих групп. Отсутствие копии в какой-либо из рабочих групп является отклонением от глобального оптимума и объясняет колебания оптимального значения целевой функции. Причиной наличия таких ошибочных фрагментов хромосомы является то, что для получения глобального оптимума необходимо синтезировать плотносвязный ОБ длиной 28 генов и, в ряде случаев, отведенного количества поколений недостаточно для его формирования. SGA, в свою очередь, допускает ошибки в размещении копий ресурса и в центральных узлах отделов, что и объясняет худшее качество получаемых решений (или отсутствие решения вообще).

Как видно из таблицы, в задачах малой размерности LLGA дает решения, которые в среднем на 10–15 % превосходят по качеству решения, представленные SGA. С другой стороны, недостатком LLGA является значительно более высокая вычислительная сложность (временные затраты выше на порядок), что показывает нецелесообразность его применения в этом случае. Однако при увеличении размерности модели SGA не смог найти решение вообще, в то время как LLGA достиг успеха. Эта особенность обусловлена тем, что при малой размерности задачи большинство ОБ уже представлено в начальном решении и ГА необходимо их только обнаружить и перенести в следующие поколения. При увеличении размерности вероятность нахождения в начальном решении готовых ОБ мала и ГА необходимо предварительно синтезировать их из фрагментов, что не всегда по силам SGA.

Заключение

Разновидность генетического алгоритма LLGA, основанная на модификации внутренней структуры хромосом, может эффективно применяться для решения тех задач сетевого проектирования, которые ранее были недоступны для SGA из-за сложной структуры образующих блоков.

Список литературы

1. Цвиркун, А.Д. Оптимизация распределения вычислительных работ и баз данных в сети ЭВМ / А.Д. Цвиркун, К.С. Тониев. – М.: Наука, 1983. – 152 с.

2. Листопад, Н.И. Теоретические основы моделирования корпоративных информационно-телекоммуникационных систем / Н.И. Листопад. – Минск, 2001. – 161 с.

3. Воротницкий, Ю.И. О задаче размещения распределенного информационного сервера в интернет в условиях переменной нагрузки / Ю.И. Воротницкий, Д.А. Стрикелев // Управление информационными ресурсами: материалы II науч.-практ. конф., Минск, 16 марта 2004 г. – Минск, 2004. – С. 44–46.

4. Воротницкий, Ю.И. Декомпозиционный подход к размещению информационных ресурсов в мультисервисных сетях / Ю.И. Воротницкий, Д.А. Стрикелев // Информационные системы и технологии (IST'2004): материалы II междунар. науч.-практ. конф., Минск, 8–10 ноября 2004 г.: в 2 ч. – Минск, 2004. – Ч. 2. – С. 18–21.

5. Goldberg, D.E. Genetic Algorithms in Search, Optimization and Machine Learning / D.E. Goldberg. – NY: Addison-Wesley Publishing Company, 1989. – 432 p.

6. Harik, G.R. Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms / G.R. Harik // Illinois Genetic Algorithm Laboratory [Electronic resource]. – 1997. – Mode of access: ftp://ftp-illigal.ge.uiuc.edu/pub/papers/IlliGALs/97005.ps.Z. – Date of access: 23.05.2006.

7. Compressed Introns in a Linkage Learning Genetic Algorithm / F.G. Lobo [et al.] // Illinois Genetic Algorithm Laboratory [Electronic resource]. – 1997. – Mode of access: ftp://ftp-illigal.ge.uiuc.edu/pub/papers/IlliGALs/97010.ps.Z. – Date of access: 23.05.2006.

8. Even, S. Graph Algorithms / S. Even. – Rockville: Computer Science Press, 1979. – 249 p.

9. Miller, B.L. Genetic algorithms with dynamic niche sharing for multimodal function optimization / B.L.Miller, M.J.Shaw // 1st IEEE Conference on Evolutionary Computation: materials of IEEE World Congress on Computational Intelligence. – NJ, 1996. – Vol. 1. – P. 786–791.

10. Miller, B.L. Genetic Algorithms, Tournament Selection, and the Effects of Noise / B.L. Miller, D.E. Goldberg // Illinois Genetic Algorithm Laboratory [Electronic resource]. – 1995. – Mode of access: ftp://ftp-illigal.ge.uiuc.edu/pub/papers/IlliGALs/95006.ps.Z. – Date of access: 23.05.2006.

Поступила 28.12.05

Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: dstrikelev@art-mission.com

Y.I. Varatnitsky, D.A. Strykeleu

NETWORK DESIGN USING LINKAGE LEARNING GENETIC ALGORITHMS

A specificity of optimal network design is examined, and factors leading to decreasing the efficiency of genetic algorithm application are revealed. A modified version of genetic algorithm is proposed, which reorders the genes in circular chromosomes to achieve better linkage and tighter building blocks. Efficiency of modified algorithm for allocation of replicated resource is shown.