

ОПТИМИЗАЦИЯ ОБРАБОТКИ ПОТОКА ЗАКАЗОВ КЛИЕНТОВ АВТОТРАНСПОРТНОГО ПРЕДПРИЯТИЯ

Войцев А.В., Юрков Д.А., Герасимов Е.И.

Восточноукраинский национальный университет им. В. Даля, г. Луганск
кафедра «Автоматика и системы управления»

E-Mail: voice-ua@yandex.ru; dma@cci.lg.ua; evgeniygerasimov@gmail.com

Abstract

Voytsev A.V., Yurkov D.A., Gerasimov E.I. Optimization of treatment of stream of orders of clients of a motor transport enterprise. A variant of solution of the task of service of clients, that are located remotely from each other, on the basis of the genetic algorithms is offered. The role of the parameters of the genetic algorithm on the effectiveness of solution search has been investigated.

Анализ состояния вопроса. В условиях современной конкурентной борьбы повышение эффективности производственного процесса для любой организации является практически единственной возможностью сохранить те позиции, которые она занимает на рынке услуг или производства. Одним из распространенных способов снижения доли затрат, а следовательно и повышения эффективности, является планирование последовательности выполнения потока производственных заданий.

Особенно актуально это в области транспортных перевозок, где задачи маршрутизации транспортных средств являются ключевыми, так как доставка товара иногда добавляет к его стоимости сумму, сравнимую со стоимостью самого товара. Однако разработка алгоритмов, позволяющих найти оптимальную последовательность выполнения потока клиентских заказов и распределение ресурсов для их обработки (исполнители, автотранспортные средства), является с вычислительной точки зрения крайне трудоемкой. Фактически, вычислительная сложность задачи зависит от размера входных данных (количества заявок, числа исполнителей и т.п.) экспоненциально, что означает невозможность обнаружения оптимального решения за приемлемое время [1]. В связи с этим при решении подобных задач требуются особые подходы, позволяющие обнаружить если не оптимальное решение, то близкое к нему.

Большинство подходов для подобных сложных случаев основано на эвристических алгоритмах. В течение последних десяти лет такие алгоритмы интенсивно исследовались и приводили к некоторым эффективным методам решения, среди которых можно указать моделируемую нормализацию (Simulated Annealing - SA), детерминированную нормализацию (Deterministic Annealing - DA), поиск запретов (Tabu Search - TS), метод муравьиной колонии (Ant Systems - AS) и конечно же генетические алгоритмы (Genetic Algorithms - GA) [2].

Особенно широкое распространение получили генетические алгоритмы благодаря особенности сравнительно простой адаптации к условиям решаемой задачи и сопутствующим ограничениям.

Постановка задачи. Рассмотрим задачу доставки грузов территориально распределенным клиентам (по их требованиям) при ограниченном наборе транспортных средств и исполнителей различной квалификации (водителей). Основным критерием является минимум финансовых издержек предприятия на транспортировку, чего можно достичь планированием последовательности обработки заказов с учетом объективных ограничений (времени выполнения, качества, грузоподъемности транспортных средств,

допуска водителей к определенному виду работ и т.п.). В общем виде это можно сформулировать как задачу минимизации функциональной зависимости:

$$f_0 = \sum_{i=1}^n \varphi_i(\bar{z}_i), \quad (1)$$

где φ_i - стоимость обслуживания i -го маршрута, n -количество маршрутов, необходимых для выполнения партии заказов, \bar{z}_i - вектор параметров, описывающий затраты на транспортировку грузов. Стоимость обслуживания каждого маршрута зависит от расстояний между пунктами транспортировки грузов и условий перевозки, что можно сформулировать как

$$\varphi_i(\bar{z}) = S(\bar{z}_i)_{m,0} + \sum_{j=0}^m S_{j,j+1}(\bar{z}_i), \quad (2)$$

где $S_{j,j+1}(\bar{z}_i)$ - стоимость перемещения грузов между j -м и $j+1$ -м пунктами маршрута, m - количество пунктов на маршруте. Так как транспортному средству необходимо вернуться в исходный пункт, то необходимо учитывать затраты и на этот отрезок пути - $S(\bar{z}_i)_{m,0}$.

Минимизацию функциональной зависимости (1) необходимо производить при следующих естественных ограничениях:

1. каждый клиент должен быть посещен один, и только один раз;
2. на каждого клиента отведено определенное время (период времени, в течение которого мы должны посетить клиента и перерывы);
3. грузоподъемность транспортных средств;
4. ограничения на маршрут движения для некоторых типов автотранспорта (невозможность движения некоторых автомобилей по некоторым дорогам или в туннелях)
5. квалификация и физическое состояние водительского персонала;
6. возможность возврата на склад клиентами попутных грузов (некоторые товары, тара и т.п.).

Анализ характера функциональной зависимости (1), ограничений, которые необходимо учитывать и размерности задачи (сотни клиентов и десятки транспортных средств) определил выбор метода решения – генетические алгоритмы.

Адаптация генетического алгоритма. Идея генетических алгоритмов основана на некоторых формализованных принципах естественного эволюционного отбора, основные положения которого описаны в [3] и [5].

Использование различных форм представления решений и специфические генетические операторы позволяет адаптировать эволюционный алгоритм к конкретной задаче так, что результат получается с приемлемым качеством за разумное время. Операторы могут быть легко приспособлены для решения различных проблем, но требования к скорости и качеству решения требует их тщательной проектировки с целью получения эффективного алгоритма.

Использование генетических алгоритмов предполагает при практической реализации определенное представление генетической информации применительно к решению конкретной задачи. Рассмотрим особенности использования генетического алгоритма для оптимизации обработки потока клиентских заказов.

Как показал опыт решения рассматриваемой задачи, при наличии большого числа ограничений в процессе скрещивания возможно появление значительного количества особей, не удовлетворяющих вышеуказанным ограничениям, но имеющие хорошее значение целевой функции. Фактически, с одной стороны происходит генерация решений, не

отвечающих ограничениям, но с другой стороны происходит расширение области поиска, что в конечном итоге приводит к улучшению результатов.

В этом случае контроль не отвечающих ограничениям решений в популяции можно производить, добавив «штрафную» составляющую к целевой функции. Причем добавочная составляющая зависит от того, насколько сильно отклоняется решение от заданных параметров и от итерации (номера поколения), на которой это решение сгенерировано.

Таким образом, на ранних итерациях штраф устанавливается меньше, для того чтобы расширить область поиска, а на конечных увеличивается, чтобы можно было избавиться от невыполнимых решений. Обобщая вышесказанное, целевую функцию можно записать в виде

$$f = f_0 + f_p, \tag{3}$$

$$f_p = \frac{p}{T} \sum_{R \in S} [\max(0, V_Z - V_T)]^2, \tag{4}$$

где p - номер текущей итерации, T - общее число итераций, V_Z - суммарный вес всех заказов на маршруте R в решении S .

Существенное влияние на качество работы алгоритма оказывает оператор кроссинговера (оператор скрещивания). Рассмотрим унифицированный оператор [4], схема которого для нашей задачи показана на рис. 1.

Для ясности показан пример доставки грузов для четырнадцати клиентов и двух родительских решений со стоимостью их маршрутов. При этом учитываются ограничения по вместимости и грузоподъемности каждого из транспортных средств.

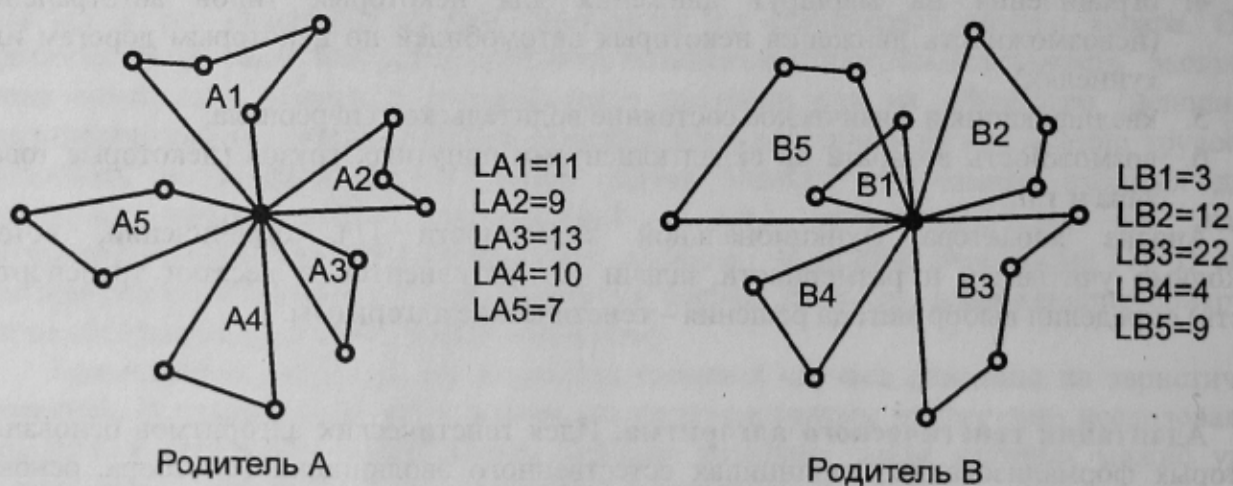


Рисунок 1 - Пример двух родительских решений и стоимость их маршрутов. Точками указаны пункты доставки грузов.

Основная идея, заложенная в унифицированном операторе скрещивания, состоит в том, что нужно перенести как можно больше нетронутых маршрутов из родительских решений в потомка. Однако невозможно перенести два маршрута из разных родителей, если они конфликтуют друг с другом, т.е. имеют общих клиентов. Каждый маршрут в одном родителе будет конфликтовать как минимум с одним маршрутом в другом родителе.

Для того, чтобы выбрать один из них, используется коэффициент R_{SR} , равный отношению стоимости маршрута к количеству обслуживаемых в нем клиентов. Таким образом, коэффициент отражает стоимость маршрута, приходящуюся на каждого клиента.

$$R_{SR} = \frac{C_{S,R}}{M_{S,R}}, \tag{5}$$

где $C_{S,R}$ - стоимость маршрута R в решении S (с учетом «штрафной» составляющей), $M_{S,R}$ - количество клиентов в маршруте R в решении S . Для нашего примера коэффициенты R_{SR} приведены в таблице 1.

Таблица 1. Коэффициент R_{SR} для всех маршрутов

Родитель A	Родитель B
$R_{A1} = 2,75$	$R_{B1} = 1,5$
$R_{A2} = 3$	$R_{B2} = 4$
$R_{A3} = 6,5$	$R_{B3} = 5,5$
$R_{A4} = 5$	$R_{B4} = 2$
$R_{A5} = 2,33$	$R_{B5} = 3$

Процедура выглядит следующим образом. Первый маршрут с наименьшей стоимостью выбирается из родителя A , и все маршруты родителя B , конфликтующие с ним, исключаются. Затем среди не исключенных маршрутов родителя B выбирается маршрут с наименьшей стоимостью, и исключаются все маршруты родителя A , конфликтующие с ним.

Процесс продолжается до тех пор, пока все маршруты не будут включены или исключены. В идеале, потомок должен содержать всех клиентов из родительских решений. Но обычно получается так, что часть клиентов в него не попадает. На Рисунок 2 показано, как маршруты в решениях на Рисунок 1 конфликтуют друг с другом

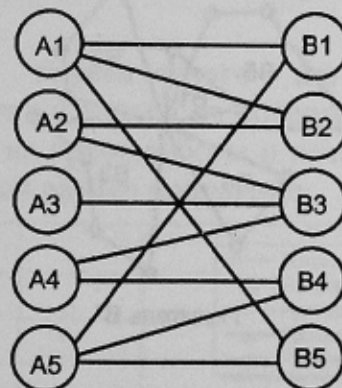


Рисунок 2 - Конфликт маршрутов при выполнении оператора скрещивания

Сначала маршрут $A5$ выбирается из решения A , так как он имеет наименьший коэффициент R_{SR} (по Таблица 1), при этом исключаются маршруты $B1, B2, B5$, конфликтующие с ним. Затем выбирается маршрут $B2$, исключая при этом маршруты $A2$ и $A1$. Затем выбирается маршрут $A4$, исключая последний оставшийся маршрут $B3$ родителя B . И наконец, выбирается маршрут $A3$. В результате работы оператора скрещивания, 10 точек из родительских решений попали в потомка, а четыре не попали, что показано на рис. 3. Этим клиентам необходимо добавить в решение искусственно.

Несмотря на достаточно хорошую работу, этот алгоритм имеет явный недостаток. Потомок наследует от родителей маршруты целиком, несмотря на то, что часть маршрута может быть качественной, а часть – нет. Лучшим решением было бы наследование не маршрутов, а фрагментов маршрута. Это проиллюстрировано на рис 4.

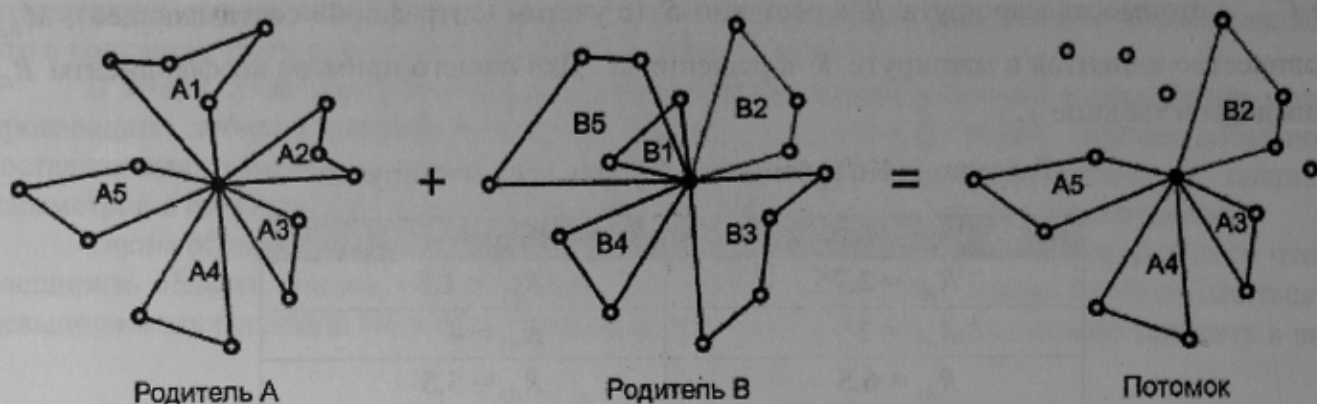


Рисунок 3 - Схема работы унифицированного оператора скрещивания

Однако, реализовать такой алгоритм довольно сложная задача, особенно если на решение накладывается ряд ограничений. В процессе кроссинговера нужно учесть все эти ограничения, чтобы потомок удовлетворял всем критериям. Компромиссным решением, включающим в себя и преимущества частичного кроссинговера и не слишком сложную практическую реализацию можно считать использование «полумаршрутов» и «половинчатый» кроссинговер.

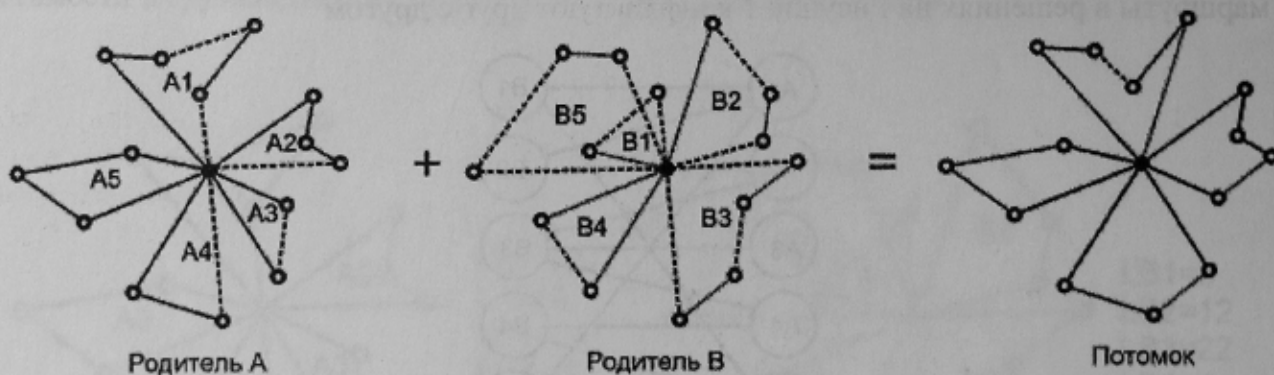


Рисунок 4 - Схема работы оператора скрещивания с делением маршрута на части

Алгоритм работы половинчатого кроссинговера похож на алгоритм унифицированного кроссинговера, с тем отличием, что все маршруты предварительно «разрезаются», что показано на рис. 5. То есть, выбирается отрезок маршрута, который будет исключен из решения.

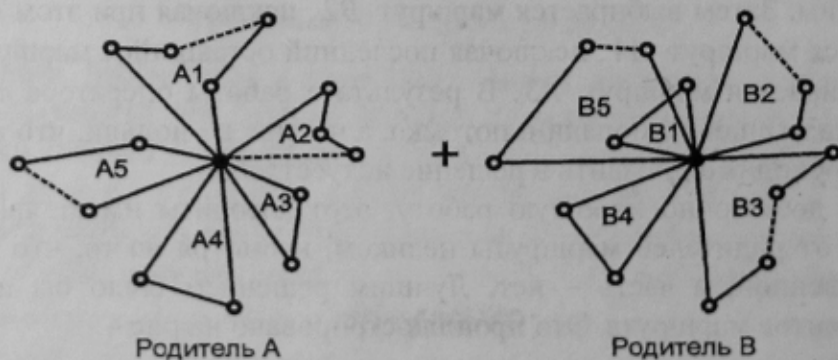


Рисунок 5 - Схема работы «половинчатого» кроссинговера

Таким образом, маршрут распадается на две части, а все дальнейшие действия выполняются над этими «половинками», а не над полными маршрутами. При этом маршруты, имеющие менее трех клиентов, не разрезаются.

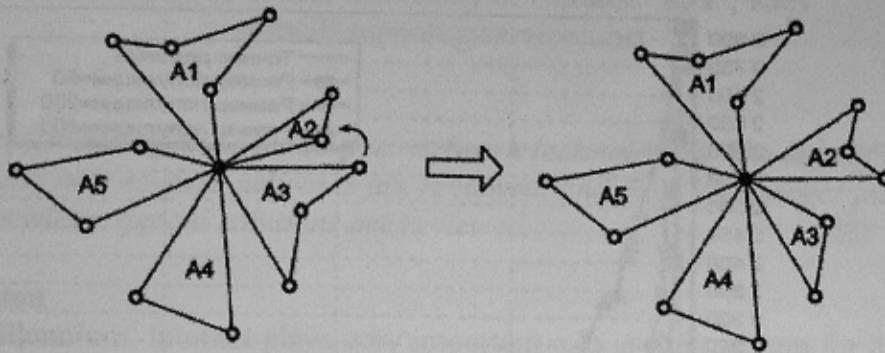


Рисунок 6 - Схема работы оператора мутации

Одним из условий успешного поиска решений является поддержка разнообразия популяции, для чего используется оператор мутации, который к тому же вносит случайность в решение. Эти изменения могут добавить новые характеристики в популяцию, и при этом не могут быть получены путем кроссинговера. Для конкретной задачи важно не использовать мутацию очень часто, чтобы не свести алгоритм к случайному поиску. Пример действия оператора мутации показан на рис. 6.

Исследование работы генетического алгоритма. Эффективность работы генетического алгоритма зависит не только от содержания операторов кроссинговера или мутации. Значительное влияние на время работы и качество работы алгоритма оказывает и такой параметр, как максимальный размер популяции.

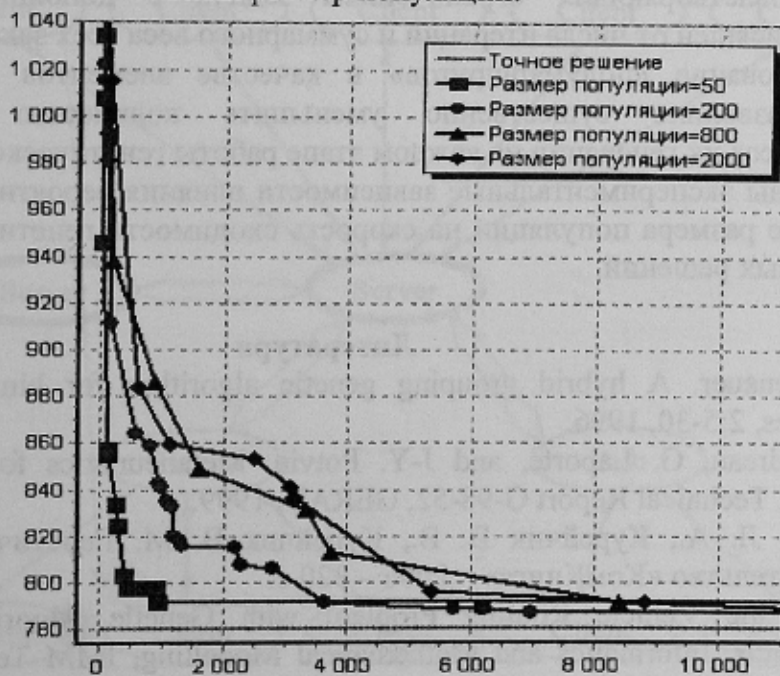


Рисунок 7 - Зависимость сходимости решения от максимального размера популяции при обслуживании 32-х клиентов

Например, на рис. 7 показана зависимость скорости сходимости решения для задачи с 32-мя клиентами. Из графика видно, что с ростом размера популяции скорость схождения популяции уменьшается, но улучшается конечный результат, хотя и незначительно.

Значительное уменьшение скорости сходимости алгоритма получается с ростом количества клиентских запросов. На рис. 8 показана та же зависимость для случая, когда необходимо обслужить 80 клиентских требований.

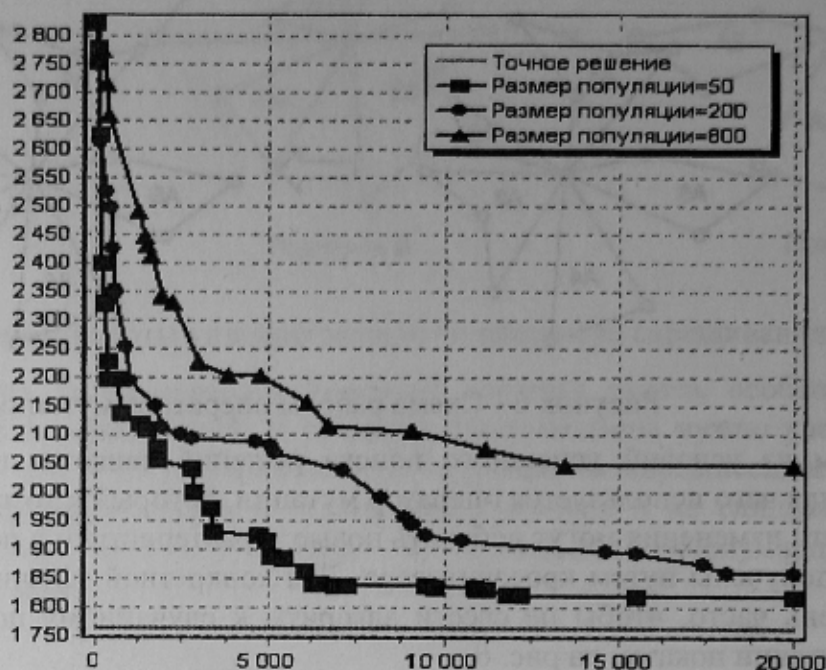


Рисунок 8 - Зависимость сходимости решения от максимального размера популяции при обслуживании 80-ти клиентов

Выводы.

1. Для расширения области поиска в каждой популяции предложено использование решений, не удовлетворяющих ограничениям задачи с дополнительной штрафной составляющей, зависящей от числа итераций и суммарного веса всех заказов на маршрутах.
2. Использование «полумаршрутов» в качестве элементов модифицированного кроссингвера позволило существенно уменьшить количество «конфликтующих» маршрутов в процессе их генерации на каждом этапе работы генетического алгоритма.
3. Получены экспериментальные зависимости влияния вероятностей кроссингвера и мутации, а также размера популяции на скорость сходимости генетического алгоритма и точность получаемых решений.

Литература

1. E. Falkenauer. A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2:5-30, 1996.
2. M. Gendreau, G. Laporte, and J-Y. Potvin. Metaheuristics for the vehicle routing problem. Technical Report G-98-52, GERAD, 1999.
3. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы. – М.:Издательство «КомКнига» – 2006 – 320 с.
4. Solving the Vehicle Routing Problem with Genetic Algorithms, Áslaug Sóley Bjarnadóttir, Informatics and Mathematical Modelling, IMM Technical University of Denmark, DTU, 2004
5. Вороновский Г.К. и др. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. / Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. – Х.: Основа, 1997. – 112с.