

МАРШРУТИЗАЦИЯ ДОСТАВКИ РОЗНИЧНОЙ ПРОДУКЦИИ В ГОРОДСКОЙ ДОРОЖНОЙ СЕТИ НА ОСНОВЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

В.П. Григорьев, К.А. Киселев

Томский политехнический университет

E-mail: jam@am.tpu.ru

Разработана модель, имитирующая динамические процессы в городской транспортной сети. На основе генетического поиска создана методика построения маршрутов обслуживания. Спроектирована система автоматизированной маршрутизации транспорта, выполняющего доставку розничной продукции.

Введение

В современных условиях рыночной экономики залогом стабильного и успешного развития любого предприятия является наличие конкурентного преимущества перед другими участниками некоторого рыночного сектора. Достичь выгодных условий производства можно за счет широкого спектра действий. Качественная организация процесса сбыта продукции является для предприятия конкурентным преимуществом. Основные направления оптимизации: оперативное оформление заявок и стабильное выполнение заказов в полном объеме и ассортименте. Для чего требуется оптимизированная схема движения транспорта. Маршруты должны удовлетворять ряду специфических требований:

- используемые в доставке автомобили имеют ограниченную грузоподъемность;
- ограниченное количество автомобилей определяет возможность их неоднократного использования;
- каждый маршрут начинается и заканчивается на складе готовой продукции;
- распределение клиентов по автомобилям и последовательность их объезда должны минимизировать расходы на доставку;
- маршруты должны отражать меняющуюся в течение суток дорожную ситуацию и отвечать существующим правилам дорожного движения.

Проблема маршрутизации транспорта (VRP – Vehicle Routing Problem) широко обсуждается в литературе. Первые исследования по проблеме появились еще в начале 70-х гг. прошлого века [1, 2]. В настоящее время также осуществляются небезуспешные попытки решения задачи и её расширений как точными, так и приближенными методами. Большинство существующих на сегодняшний день методик решают статический вариант VRP. Это упрощенная и не всегда адекватная реальным условиям ситуация. Для городских дорожных сетей (ГДС) характерны колебания в течение суток пропускной способности на многих участках движения, что неизбежно сказывается на качестве работы всевозможных служб доставки. Кроме того, практически каждая реальная ГДС изобилует ули-

цами с односторонним движением, постоянно или периодически запрещенными поворотами. Время преодоления маршрута может существенно зависеть от его конфигурации, поскольку ломаные схемы объезда снижают среднюю скорость движения. Проблема особенно актуальна для розничных продуктовых магазинов, где требуются регулярные поставки относительно небольших партий товаров. Создавать складские помещения в таких условиях обычно не имеет смысла: реализуемая продукция по большей части имеет ограниченный срок годности, дополнительные площади влекут дополнительные расходы на их содержание.

Специфика VRP на любом графе состоит в том, что на суммарный пройденный всем используемым транспортом путь, на время выполнения всех заказов влияют как качество разбиения множества клиентов по маршрутам, так и качество найденных циклов объезда клиентов транспортными единицами. Эта специфика особенно важна при моделировании сетей с большими колебаниями пропускной способности, что характерно для современных городов.

В данной работе учитывается указанная специфика VRP. Предлагаемая методика решения задачи маршрутизации состоит из нескольких этапов: преобразование исходного графа ГДС к графу определенного вида применением специально модифицированного алгоритма Дейкстры; построение маршрутов при помощи разработанной метаэвристики, реализованной на основе генетического алгоритма (ГА). В составе ГА предлагается комплекс мер, позволяющий эффективно находить схемы движения транспорта, минимизирующие стоимость доставки. Выбор метода обоснован тем, что задача маршрутизации является NP сложной, т. е. размер пространства поиска имеет экспоненциальную зависимость от количества переменных. В предложенном расширении она имеет большое и сложное пространство решений даже при сравнительно небольшом количестве клиентов. Существует ряд источников, например [3], где на тестах показана эффективность применения на графовых задачах метаэвристик, особенно ГА. С ростом размерности задачи качество получаемых в ГА решений, по сравнению с другими методами, растет.

Задача построения планов доставки для розничной клиентской сети

В качестве модели ГДС использован динамический граф, который имитирует изменения пропускной способности улиц, существование участков с односторонним движением, постоянно и периодически запрещенные повороты, увеличение времени преодоления маршрута при использовании поворотов большой крутизны, связанное с необходимостью тормозить и снова набирать скорость. В условиях динамического графа целевым критерием принят баланс протяженности и продолжительности искоемых маршрутов.

Представим городскую дорожную сеть в виде ориентированного связанного графа $G=\langle D, U \rangle$, где $D=\{1, 2, \dots, n_D\}$ – множество вершин графа мощностью n_D , U – множество дуг, соединяющих вершины. Узлами графа считаем как различные точки реализации продукции, например магазины, столовые, больницы или детские сады, так и простые перекрестки автомобильных дорог. Выделим целевое подмножество вершин $\hat{P}=\{1, 2, \dots, \hat{n}\}$, $\hat{P} \subset D$, которое составляют только обслуживаемые клиенты.

На рис. 1 множество вершин D – все вершины графа, $\hat{P}$ – черные вершины, $u_{ij} \in U$, $i, j \in D$ – дуги между любыми соседними вершинами, причем $u_{ij} \neq u_{ji}$. В левом верхнем углу обозначен склад готовой продукции.

Предприятие выпускает продукцию циклически. Поэтому все множество клиентов $\hat{P}$, при поступлении заявок, разносится по циклам производства. Тогда нашей задачей является планирование доставки для клиентов некоторого цикла производства. В качестве подмножества клиентов некоторого производственного цикла примем P , $P \subset \hat{P}$. Тогда n – количество клиентов, соответствующее некоторому циклу, которое необходимо обслужить.

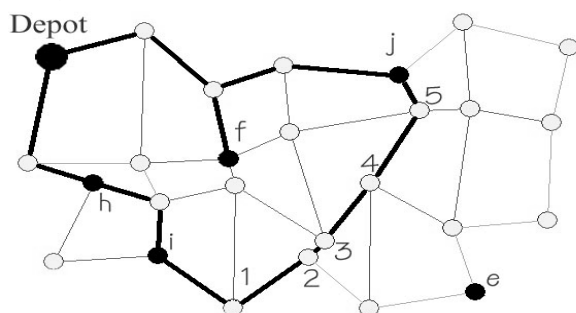


Рис. 1. Граф G – модель городской дорожной сети

Пропускная способность улиц является функцией времени и почти циклически изменяется каждые сутки. Поэтому весь суточный диапазон условно разобьем на T отрезков, внутри которых показатель постоянен. Тогда каждая дуга $u_{ij} \in U$ графа G , соединяющая вершины $i, j \in D$ и представляющая собой некоторую часть улицы, обладает протяженностью $l(u_{ij})$, км, и средней скоростью $w(u_{ij}, \tau)$, $\tau=1 \dots T$, км/ч, в момент времени τ , где τ – дискрет-

ное время, определяющее номер суточного интервала. Номер интервала τ определяется как функция времени $\tau=\tau(t)$, где t – текущее модельное время.

В распоряжении транспортного цеха находится q автомобилей. Каждый автомобиль характеризуется грузоподъемностью – K , кг(м³). Заказ i -го клиента имеет вид: $d_i=(d_{iv})$, $i \in P$, $v \in S$, где d_{iv} – объем d товара v , который должен быть доставлен клиенту i , S – множество производимой на заводе продукции.

Множество клиентов P требуется разбить по автомобилям так, чтобы суммарный объем заказов подмножества клиентов, назначенных конкретному автомобилю для перевозки, не превышал грузоподъемности этого автомобиля. Обозначим через $u \in Y$ разбиение множества P на непересекающиеся подмножества $p_1 \cup \dots \cup p_k \cup \dots \cup p_\alpha$ так, что $p_k \cap p_i = \emptyset$, где α – общее количество рейсов, Y – множество всех допустимых разбиений множества P . Например, на рис. 1, $p_1=\{h, f, j, i\}$, $p_2=\{e\}$. Тогда загрузка автомобилей для перевозки товаров назначенным заказчикам определяется неравенством

$$\sum_{i \in p_k} d_i \leq K_k, \quad k=1 \dots \alpha. \quad (1)$$

Для каждого используемого автомобиля необходимо вычислить оптимальный маршрут объезда вершин назначенного подмножества p_k разбиения u . Обозначим перестановкой $z_k=(i_1 \dots i_m \dots i_{g_k-1}, i_{g_k})$ альтернативный порядок объезда автомобилем k вершин $i_m \in p_k$, $m=1 \dots g_k$ из множества всех возможных перестановок Z_k . Здесь i_1 – отправная вершина, обозначающая начало маршрута, т. е. склад готовой продукции; $g_k-1=|p_k|$ – количество целевых вершин на маршруте. На рис. 1 $z_1=(i_1, h, i, j, f, i_6)$. Завершение объезда происходит в исходной вершине, т. е. $i_{g_k}=i_1$. Маршруты должны охватывать вершины множества p_k в порядке, обеспечивающем минимальную суммарную стоимость доставки.

Основными факторами, определяющими ее стоимость, являются продолжительность и протяженность маршрутов. Поэтому в качестве целевого критерия рассматривается их линейная комбинация по всем транспортным единицам, что ведет к оптимальному использованию транспортных ресурсов и, следовательно, к оптимизации затрат. Целевая функция имеет вид:

$$\sum_{k=1 \dots \alpha} [\lambda l(z_k) + (1-\lambda)\theta(z_k)] \rightarrow \min, \quad (2)$$

где $l(z_k)$ и $\theta(z_k)$ – протяженность и продолжительность маршрута z_k соответственно, λ – весовой коэффициент.

Протяженность $l(z_k)$ маршрута z_k складывается из длин путей $\tilde{u}_{i_m i_{m+1}}(\tau) \in \tilde{U}(\tau)$, соединяющих следующие друг за другом вершины $i_m, i_{m+1} \in p_k$, $m=1 \dots (g_k-1)$. На рис. 1 $\tilde{u}_{i_m i_{m+1}}(\tau)=(i, 1, 2, 3, 4, 5, j)$. Путь $\tilde{u}_{i_m i_{m+1}}(\tau)$ является функцией дискретного времени τ , поскольку с течением суток оптимальный маршрут, соединяющий вершины i_m, i_{m+1} , может существенно меняться, $\tau=\tau(t_{i_{m-1}})$, где $t_{i_{m-1}}$ – время отъезда

из вершины i_{m-1} . Подробней алгоритм построения множества $\tilde{U}(\tau)$ описан ниже.

Длительность $\theta(z_k)$ маршрута z_k определяется следующим образом.

Пусть время отъезда из текущей вершины i_m рассчитывается по формуле

$$t_{i_m} = \begin{cases} t_{i_m}^{\partial g} + t_{i_m}^{pa3c} & | m = 2..g_k \\ t_k^{cmapm} & | m = 1 \end{cases}, \quad (3)$$

где $t_{i_m}^{ob}$ — момент прибытия в вершину i_m из вершины i_{m-1} ; t_k^{start} — время отправления со склада автомобиля k ; t_m^{pass} — интервал времени, необходимый для разгрузки заказа клиента $i_m \in p_k$, $k=1... \alpha$.

Величина $t_{i_n}^{de}$ складывается из времени отъезда из предыдущей вершины $t_{i_{n-1}}$ и времени на преодоление расстояния $l(\tilde{u}_{i_{n-1}+1}(\tau))$, $\tau = \tau(t_{i_{n-1}})$ со средней скоростью $w(\tilde{u}_{i_{n-1}+1}(\tau))$, τ возвращает номер суточного интервала, соответствующий текущему значению $t_{i_{n-1}}$.

$$t_{i_m}^{\partial g} = t_{i_{m-1}} + \frac{l(\tilde{u}_{i_{m-1}i_m}(\tau))}{w(\tilde{u}_{i_{m-1}i_m}(\tau))}, \quad i_m \in p_k. \quad (4)$$

Время разгрузки $t_{l_m}^{раз}$ заказа определяется по формуле

$$t_{i_m}^{pa3c} = \tilde{t} \cdot \sum_{v \in S} d_{i_m, v} \quad , \quad i_m \in p_k \quad , \quad (5)$$

где $\tilde{t}$ – среднее количество времени для разгрузки 1 ед. продукции.

Тогда время маршрута отдельно взятого автомобиля $\theta(z_k)$ равно величине $t_{i_k}^{os}$, определяющей время возвращения автомобиля k на склад — вершина i_{sk} минус время старта $t_{i_{sk}}^{start}$

$$t_{i_{gk}}^{\partial \Theta} = t_{i_{gk-1}} + \frac{l(\tilde{u}_{i_{gk-1}^{j_{gk}}}(\tau))}{w(\tilde{u}_{i_{gk-1}^{j_{gk}}}(\tau))}, \tau = \tau(t_{i_{gk-1}}), i_{gk} \in p_k, k=1 \dots \alpha.$$

Метод решения

ГДС вместе с розничной сетью может насчитывать более тысячи точек. Следовательно, возможно большое количество вариантов соединения клиентов между собой. Разделим задачу организации маршрутов доставки на два основных этапа: упрощение исходного графа городской дорожной сети (построение сокращенного пространства поиска) и непосредственно нахождение маршрутов.

Упрощение исходного графа городской дорожной сети.

Для решения VRP, в указанной постановке, строим новый граф $G = \langle \bar{P}, \bar{U}(\tau) \rangle$, $\bar{P} \subset D$, $\tau = 1 \dots T$ на основании исходного G , где $\bar{P}$ – множество всех клиентов, $\bar{U}(\tau)$ – множество путей, соединяющих клиентов оптимальным для суточного интервала образом.

Для построения множеств дуг $\bar{U}(\tau)$, $\tau=1...T$ применяем алгоритм Дейкстры [4], который специально модифицируем для этих целей. В модифицированной форме алгоритм учитывает: протяженность

$l(u_{ij})$, $u_{ij} \in U$, $i, j \in \hat{P}$ исходных дуг, среднюю скорость движения $w(u_{ij}, t)$ в текущем интервале, запрещенные на момент τ повороты, дороги с односторонним движением, а также проводит штрафование за совершение поворотов большой крутизны. Алгоритм работает за линейное время.

Применяя алгоритм $\hat{n}(\hat{n}-1)T$ раз, строим упрощенный граф для каждого τ . В итоге совокупность множеств $\tilde{U}(\tau)$, $\tau=1...T$ образует полностью допустимое, оптимально сокращенное пространство поиска, учитывающее специфику ГДС, где в каждый момент времени точно известно, каким маршрутом следует двигаться из одной целевой вершины в другую.

Готовый граф $\tilde{G}$ можно хранить в отдельной базе данных и проводить пересчет только при изменении списка постоянных клиентов — появлении новых или удалении пассивных на протяжении определенного срока клиентов.

Поиск маршрутов. Решение задачи маршрутизации транспортных единиц для обслуживания клиентов реализовано в виде ГА.

Архитектура генетического алгоритма

Генетический алгоритм представляет собой методику случайного поиска [5, 6].

Метод кодирования решений подбирается под конкретный тип исследуемой задачи и должен обеспечивать адекватное представление любой точки пространства решений. В качестве альтернативного решения рассмотрим последовательность обслуживаемых клиентов, в которой выполняется доставка продукции. Например, первая альтернатива в популяции решений вида

$$\begin{aligned} p_1 & : 1-2-5-11-7-4-6 \\ p_2 & : 6-11-7-5-2-1-4 \\ & \dots\dots\dots \\ p_r & : 6-7-4-2-11-1-5 \end{aligned}, \quad (6)$$

говорит о том, что, если в первый автомобиль можно загрузить только трёх первых клиентов, не нарушая условие (1), то его маршрут будет следующим: 0–1–2–5–0; если все оставшиеся клиенты помещаются в следующий автомобиль, его маршрут будет: 0–11–7–4–6–0, где ноль является складом готовой продукции.

Для запуска поискового процесса необходимо **создать стартовую популяцию** решений-хромосом. В нашем случае применена смешанная схема инициализации. Часть популяции создана при помощи широко известного метода Ближайшего Соседа (*Nearest Neighbour*) [7]. При создании оставшихся особей используется генератор случайных чисел, выдающий допустимые последовательности на конечном множестве клиентов P . Подобная схема позволяет существенно улучшить поиск при выраженной кластеризованности клиентов.

Приспособленность альтернативных хромосом (качество решений) оценивается специально разработанной эвристической функцией, легко реализуемой на любом языке программирования. На рис. 2 представлена функциональная схема алгоритма.

1. Цикл оценки альтернатив, где m – размер популяции.
2. Цикл перебора доступного транспорта.
3. Установка времени начала выполнения доставки для привязки к конкретному временному интервалу τ .
4. Цикл загрузки заказов клиентов j в текущий автомобиль в порядке, определяемом текущей хромосомой.
5. Проверка наличия места в автомобиле k для загрузки заказа клиента j .
6. Загрузка очередного заказа.

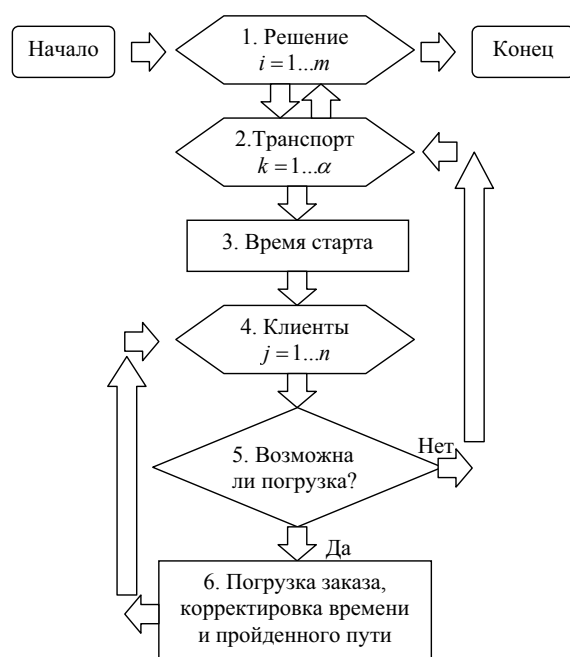


Рис. 2. Функциональная схема алгоритма оценки

При погрузке заказа корректируется свободное место в автомобиле на соответствующую заказу величину. При этом увеличиваются счетчики пройденного пути и затраченного на дорогу и разгрузку заказа времени на величины соответствующих характеристик дуг $\tilde{t}(\tau)$, соединяющих соседних в решении клиентов (3–4). Также учитываем и дуги, соединяющие первого и последнего клиента со складом. Корректировка текущего времени t по окончании разгрузки приводит к смене τ , если текущее время вышло за границы интервала. Что позволяет выбрать оптимальный маршрут движения к следующему клиенту.

Потенциальное количество заказов, которое можно загрузить, зависит от их объема, следовательно, количество вершин в одном автомобиле

может варьироваться в широких пределах. Поэтому предложена система представления решений, кодирующая не транспортные единицы со списками обслуживаемых клиентов, а простую их последовательность. Это в сочетании с описанным алгоритмом оценки позволяет эффективно использовать выделенную память, максимально упростив при этом структуру кодируемых решений (6), и исключить отдельный блок контроля допустимости получаемых во время эволюции альтернатив. Недопустимые с точки зрения перегруза транспорта решения просто исключаются.

С каждой оценкой на альтернативную последовательность клиентов накладываются в строго фиксированном для всей эволюции порядке емкости доступных автомобилей, то есть выполняется строго последовательная динамическая загрузка транспорта клиентами. Поэтому с течением эволюционного процесса лучшими решениями будут те альтернативы, где сравнительно близкие клиенты будут группироваться в соответствии с емкостями автомобилей и в последовательности, обеспечивающей минимум целевой функции (1).

Сочетание в данном ГА системы представления решений и оценочной процедуры обладает ключевыми для задачи свойствами: в эволюционном процессе происходит параллельный поиск оптимального разбиения вершин по маршрутам и оптимального порядка объезда клиентов внутри каждого подмножества.

Здесь использована **стратегия отбора** родителей по принципу колеса рулетки и стратегия элитизма. Таким образом, исключается потеря хороших решений при применении генетических операторов. При помощи **генетических операторов** производится создание новых, ранее не присутствовавших в популяции особей. Сканирование поискового пространства осуществляется операторами **скрещивания** (кроссинговера), **мутации**, **инверсии**. Применен частично – соответствующий оператор скрещивания. Выбор продиктован таким условием задачи, что любой клиент может быть обслужен только однажды и одним автомобилем: $p_k \cap p_s = \emptyset$, $k \neq s$, $k, s = 1 \dots \alpha$. То есть потомок должен иметь тот же генный состав, что и исходные особи. Модификации проявляются лишь в порядке следования вершин. Существенно улучшить качество решений позволяет применение жадных мутаций – **стратегий локального поиска**, модифицирующих исходное решение на основе анализа целевой функции или ее некоторых параметров. Описание методов локального поиска и различных генетических операторов широко представлено в литературе и сети Интернет, например, в [5, 6].

Условия останова поиска:

- алгоритм выполнил заданное количество итераций $AbsN$;
- на протяжении $LocN$ итераций лучшее решение не изменяется.

Сравнительные результаты и эффективность подхода

Пробные расчеты на статических тестовых задачах, опубликованных в научной литературе и доступных в сети Интернет, показывают хорошие результаты, сопоставимые с лучшими достижениями, опубликованными в литературе [8]. По многим задачам отклонения достигают меньше 2 %. Стандартных задач на динамических графах не найдено. Следует отметить, что качество решений, в том числе их стабильность, существенным образом зависят от отведенного на поиск времени. Например, для задачи с 200 клиентами при $LocN = 200$ среднее время поиска составляет 40 с, а разброс результатов может достигать 5...7 % от лучшего найденного алгоритмом решения. При $LocN = 2000$ среднее время поиска уже равно 7 мин., и разброс составляет 2...3 %.

Предложенный выше подход подразумевает такую стратегию поиска, что найденные на динамическом графе маршруты обеспечивают лучшие результаты при их использовании в реальных условиях, по сравнению с маршрутами, найденными на статическом графе. Предложена следующая методика оценки эффективности поиска на динамическом графе. Пусть P_1 – решение задачи, найденное на статическом графе, P_2 – на динамическом. Качество решения определенное по весам статического графа обозначим через $L(P)$, где P – некоторое решение. Оценка решения P на динамическом графе – $H(P)$. За оценку снизу примем $L(P_1)$, поскольку решение найдено в легких дорожных условиях. Очевидно, худшей оценкой является $H(P_1)$,

т. к. решение P_1 не учитывает особенностей динамического графа. Через $\Delta_1 = H(P_1) - L(P_1)$ обозначим погрешность моделирования. Тогда эффективность метода решения определим формулой

$$\Delta_2 = \frac{H(P_1) - H(P_2)}{\Delta_1} \quad \text{в предположении, что}$$

$L(P_1) \leq H(P_2) \leq H(P_1)$. Чем ближе Δ_2 к единице, тем больше эффект от моделирования на динамическом графе. Расчеты показали, что $\Delta_2 \approx 0,6$, т. е. моделирование на динамическом графе позволяет сократить примерно на 60 % ошибку моделирования Δ_1 , которая составляет 30 % по отношению к $H(P_1)$. Причем параметр существенно зависит от степени и структуры загрузки графа, количества и разброса клиентов.

Заключение

Создана модель городской дорожной системы, имитирующая сложные, меняющиеся с течением времени, характеристики, присущие реальным дорожным сетям современных городов. На архитектуре генетического поиска реализована методика поиска маршрутов движения обслуживающего транспорта, учитывающая особенности реальной дорожной сети. Разработан программный продукт, представляющий систему маршрутизации транспортных средств. Система выполняет работу по сбору, хранению и подготовке информации о клиентах, текущих заказах, состоянии ГДС, имеющегося транспорте, а так же непосредственное решение VRP одним из реализованных методов.

СПИСОК ЛИТЕРАТУРЫ

1. Christofides N., Eilon S. An algorithm for the vehicle-dispatching problem // Operational Research Quarterly. – 1969. – № 20. – P. 309–318.
2. Christofides N., Mingozzi A., Toth P. The vehicle routing problem. In: Christofides N., Mingozzi A., Toth P., Sandi C., editors. Combinatorial optimization. – London: Wiley, 1979. – 220 p.
3. Минаков И.А. Сравнительный анализ некоторых методов случайного поиска и оптимизации // Известия Самарского НЦ РАН. – 1999. – № 2. – С. 286–293.
4. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 470 с.
5. Емельянов В.В., Курейчик В.М., Курейчик В.В. Теория и практика эволюционного моделирования. – М.: Физматлит, 2003. – 595 с.
6. Материалы сайта <http://qai.narod.ru/GA/index.htm>
7. Материалы сайта www.citforum.ru
8. Материалы сайта <http://neo.lcc.uma.es/radi-aeb/WebVRP/main.html>

Поступила 13.11.2006 г.