

УДК 004.023

**ОПТИМИЗАЦИЯ ОБСЛУЖИВАЮЩИХ РАБОТ НА ОСНОВЕ  
МЕТАЭВРИСТИЧЕСКИХ ТЕХНОЛОГИЙ****Гавенко С.С., Светличная В.А.**Донецкий национальный технический университет  
кафедра Автоматизированных систем управления  
E-mail: [scorpius17@bk.ru](mailto:scorpius17@bk.ru)**Аннотация**

*Гавенко С.С., Светличная В.А. Оптимизация обслуживающих работ на основе метаэвристических технологий. Проведен анализ текущего положения процесса сопровождения программного обеспечения. Приведено описание и обоснование актуальности проблемы. Рассмотрено применение муравьиного алгоритма для оптимального распределения обслуживающих работ в условиях ООО «Парус-Донецк» путем построения оптимальных маршрутов для каждого сотрудника.*

**Общая постановка проблемы**

Поддержка клиентов в настоящее время занимает высокие позиции среди целей корпораций, в том числе для привлечения новых клиентов. Современные технологии позволяют повысить эффективность решения данной задачи. В настоящее время есть возможность общения с клиентами по средствам не только телефонии, но и видео-телефонии (Skype, Google Talk, QIP и др.). Несмотря на это, существуют проблемы, решить которые возможно только выездом сотрудника организации к клиенту. Примером обслуживающей организации будет служить корпорация «Парус-Донецк». Основной вид деятельности компании: разработка, продвижение и внедрение программного обеспечения (ПО) для автоматизации управления предприятием. «Парус-Донецк» – это представительство корпорации «Парус-Украина» в Донбассе. Чтобы обеспечивать поддержку организациям, количество которых превышает 1500, сотрудникам донецкого филиала необходимо оптимально распределять свое время и выбирать наилучший маршрут для продуктивной работы.

**Постановка задачи исследования**

Штат корпорации «Парус-Донецк» насчитывает  $V = 43$  сотрудника, занимающихся обслуживанием организаций-клиентов. В списке клиентов находится 434 организации, расположенных в Донецке и 1136 в пределах области. Всего их  $O' = 1570$ .

Каждый месяц выходит обновленный инсталлятор. Для обновления ПО, в идеале, клиент не нуждается в помощи, но на деле получаем такую статистику: примерно 70% организаций-клиентов не могут провести обновление самостоятельно и обращаются за помощью в «Парус-Донецк». 70% от  $O'$  организаций – это  $O \approx 1100$  единиц фирм. Учитывая штат представительства в Донецке, получаем отношение «организаций на сотрудника»:  $p = O / V = 1100 / 43 \approx 26$ .

Каждые 3 месяца в обновлении нуждается программный продукт «Парус-Консолидация». Процесс обновления этой системы не должен длиться больше недели. Получается, что 1100 организаций необходимо обновить 43 сотрудникам «Парус-Донецк» за  $d = 5$  рабочих дней. Для более ясного представления, подсчитаем количество организаций, которые должен обслужить каждый сотрудник корпорации по автоматизации управления производством. Учитывая то, что на сотрудника приходится 26 организаций, получим:  $r = p / d = 26 / 5 \approx 5$  (организаций в день). Процесс обновления ПО у одного клиента занимает в среднем  $tw_i = 1$  час, в то время как на дорогу уходит  $tr_i = 2 \div 3$  часа (в области). Если

рассчитать количество времени, затрачиваемое сотрудником «Парус-Донецк» на 1 организацию в области, получим:  $t_i = 1 + 3 = 4$  (часа) максимум). В сумме:  $T_f = 4 * 5 = 20$  (часов) на 5 организаций – норма для сотрудника Парус-Донецк). Учитывая, что рабочий день составляет  $T_p = 8$  часов, сталкиваемся с проблемой нехватки времени, а точнее – сотрудник не успеет обслужить половину организаций при максимальной нагрузке.

Число клиентов «Парус-Донецк» растет с каждым днем, а значит и нагрузка на персонал, обслуживающий их возрастает. Целью сотрудника в период обновления ПО является успеть обслужить всех клиентов по плану. При этом у него есть ограничения по времени. Достижению цели могут способствовать большие затраты на проезд. Т.е. можно сменить вид транспортного средства (например, вместо автобуса воспользоваться услугами такси). Данный вариант нас не устраивает, потому что нашей задачей является минимизация затрат при максимальной продуктивности. Стоимость проезда – понятие относительное. Нельзя сказать что суммарная стоимость проезда к двум организациям в Донецке может быть равна суммарной стоимости проезда к двум организациям за его чертой (так как стоимость зависит от длины пути). Поэтому, будем минимизировать не саму стоимость, а отношение стоимости ко времени.

### Решение задачи

Задача оптимизации распределения сотрудников может быть сформулирована как минимизация стоимости всех замкнутых маршрутов с учетом выполнения ограничений.

Основная целевая функция:

$$\sum_{i=1}^V \frac{S_i}{T_{f_i}} \rightarrow \min \quad (1)$$

где  $S_i$  – сумма стоимости проезда по  $O_{f_i}$  организациям,  $T_{f_i}$  – суммарное количество времени на  $O_{f_i}$  организаций.

Ограничение по объектам:

$$O_{f_i} = O_{p_i} \quad (2)$$

где  $O_{f_i}$  – фактическое количество организаций на  $i$ -го сотрудника,  $O_{p_i}$  – плановое количество организаций на  $i$ -го сотрудника.

Ограничения по времени:

$$T_{f_i} \leq T_p \quad (3)$$

где  $T_p$  – рабочее время сотрудника за день, равное норме (8 часов).

$$S_i = s_1 + \sum_{j=2}^{O_{f_i}} s_j \quad (4)$$

Где  $s_1$  – затраты на проезд от офиса к первой организации,  $s_j$  – затраты на проезд от  $(j-1)$ -ой к  $j$ -ой организации  $i$ -му сотруднику,  $O_{f_i}$  – количество организаций, которые должен посетить  $i$ -ый сотрудник за день.

$$T_{f_i} = t_{w_1} + \sum_{j=2}^{O_{f_i}} (t_{r_j} + t_{w_j}) \quad (5)$$

где  $t_{w_1}$  – время на дорогу от офиса к первой организации,  $t_{r_j}$  – время на дорогу от  $(j-1)$ -ой к  $j$ -ой организации для  $i$ -го сотрудника,  $t_{w_j}$  – время на обслуживающие работы  $j$ -ой организации для  $i$ -го сотрудника.

$$O_{p_i} = \frac{O}{V * \tau}, \quad (6)$$

где  $O$  – общее число организаций,  $V$  – общее число сотрудников,  $\tau$  – период обновления.

Формула (1) является основной целевой функцией при ограничениях (2) и (3), а формулы (4) – (6) описывают каждый параметр. Гранулярность целевой функции объясняется тем, что рассматриваемая задача состоит из нескольких подзадач и имеет определенные ограничения для каждой из них. Так, например, (4) ограничена количеством возможных путей следования и дальностью поездки, а (5) зависит от количества контрольных точек (организаций). Число организаций по плану для каждого сотрудника (6) является переменным и зависит от периода обновления ПО и количеством клиентов, которых необходимо обслужить.

Решать данную задачу можно теми же методами, что и задачу коммивояжера, но при этом необходимо учитывать не только время в дорогу ( $t_{rj}$ ), но и время обслуживания ( $t_{wj}$ ). К таким методам относятся:

- полный перебор;
- случайный перебор;
- жадные алгоритмы;
- генетические алгоритмы;
- метод ветвей и границ;
- метод эластичной сети;
- муравьиный алгоритм.

Для решения был выбран последний метод. В основе алгоритма лежит поведение муравьиной колонии — маркировка более удачных путей большим количеством феромона. Работа начинается с размещения муравьёв в вершинах графа (рис. 1), затем начинается движение муравьёв — направление определяется вероятностным методом (7). Решение не является точным и даже может быть одним из худших, однако, в силу вероятности решения, повторение алгоритма может выдавать достаточно точный результат.

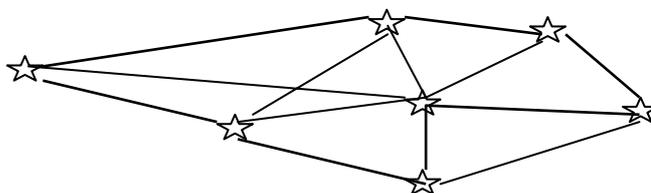


Рисунок 1 – Граф возможных путей следования сотрудника

В условиях данной задачи в роли муравья выступает сотрудник, который начиная движение с офиса должен пройти по всем указанным точкам, ребром является маршрут между организациями.

Для адаптации классической задачи к нашей теме введем следующие обозначения:

- муравей → сотрудник;
- феромон → рейтинг;
- ребро графа (тропа) → путь;
- вершина графа → клиент.

Моделирование движения сотрудника связано с распределением рейтинга на пути маршрута. При этом вероятность (7) включения пути в маршрут отдельного сотрудника пропорциональна количеству рейтинга на этом пути, а количество инкрементируемого рейтинга пропорционально длине маршрута. Чем короче маршрут, тем больше рейтинга будет инкрементировано на его пути (8), следовательно, большее количество сотрудников будет включать его в синтез собственных маршрутов. Моделирование такого подхода, использующего только положительную обратную связь, приводит к преждевременной сходимости - большинство сотрудников движется по локально оптимальному маршруту. Избежать этого можно, моделируя отрицательную обратную связь в виде декрементации рейтинга. При этом если рейтинг декрементирует быстро, то это приводит к «потере памяти»

сотрудников и забыванию хороших решений, с другой стороны, большое время декрементации может привести к получению устойчивого локального оптимального решения.

$$P_{ij}(t) = \frac{w_{ij}(t)^\alpha \left(\frac{1}{d_{ij}}\right)^\beta}{\sum w_{ij}(t)^\alpha \left(\frac{1}{d_{ij}}\right)^\beta} \quad (7)$$

где  $w_{ij}(t)$  – уровень рейтинга,  $d_{ij}$  – эвристическое расстояние, а  $\beta$ ,  $\alpha$  – константные параметры. При  $\alpha = 0$  выбор ближайшего клиента наиболее вероятен, то есть алгоритм становится жадным. При  $\beta = 0$  выбор происходит только на основании рейтинга, что приводит к субоптимальным решениям.

$$w_{ij}(t+1) = (1 - \rho)w_{ij}(t) + \sum \frac{Q}{L_k} \quad (8)$$

где  $\rho$  – интенсивность декрементации,  $L_k(t)$  – цена текущего решения для  $k$ -ого сотрудника, а  $Q$  – параметр, имеющий значение порядка цены оптимального решения, то есть  $\frac{Q}{L_k(t)}$  – рейтинг, инкрементируемый  $k$ -ым сотрудником, использующим ребро  $(i,j)$ .

В начале алгоритма количества рейтинга на пути принимается равным небольшому положительному числу (9). Общее количество сотрудников остаётся постоянным, каждый сотрудник начинает маршрут из офиса.

$$w = \frac{K}{M} \quad (9)$$

где  $K$  – количество пересадок между транспортом,  $M$  – расстояние между точками.

Локальные правила поведения сотрудников при выборе пути выглядят следующим образом:

1. Сотрудники имеют собственную «память». Поскольку каждый клиент может быть посещён только один раз, то у каждого сотрудника есть список уже посещённых клиентов – список запретов. Обозначим через  $J_{i,k}$  список клиентов, которые необходимо посетить сотруднику  $k$ , находящемуся у клиента  $i$ .

2. Сотрудники обладают «зрением» – видимость есть эвристическое желание посетить клиента  $j$ , если сотрудник находится у клиента  $i$ . Будем считать, что видимость обратно пропорциональна расстоянию между клиентами  $\eta_{ij}=1/D_{ij}$ .

3. Сотрудники обладают «интуицией» – они могут отслеживать рейтинг, подтверждающий желание посетить клиента  $j$  от клиента  $i$  на основании опыта других сотрудников. Число рейтинга на ребре  $(i,j)$  в момент времени  $t$  обозначим через  $\phi_{ij}(t)$ .

4. На этом основании мы можем сформулировать вероятностно-пропорциональное правило (8), определяющее вероятность перехода  $k$ -ого сотрудника от клиента  $i$  к клиенту  $j$ :

$$\left\{ \begin{array}{l} P_{ij}(t) = \frac{w_{ij}(t)^\alpha \eta_{ij}^\beta}{\sum_{l \in J_{i,k}} w_{il}(t)^\alpha \eta_{il}^\beta}, j \in J_{i,k}; \\ P_{ij,k}(t) = 0, j \notin J_{i,k}; \end{array} \right. \quad (10)$$

где  $\alpha$ ,  $\beta$  – параметры, задающие веса рейтинга. При  $\alpha=0$  алгоритм вырождается до жадного алгоритма (будет выбран ближайший клиент). Заметим, что выбор клиента является вероятностным, правило (10) лишь определяет ширину зоны клиента  $j$ ; в общую зону всех клиентов бросается случайное число, которое и определяет выбор сотрудника. Правило (10)

не изменяется в ходе алгоритма, но у двух разных сотрудников значение вероятности перехода будут отличаться, т.к. они имеют разный список разрешённых клиентов.

5. Пройдя ребро  $(i,j)$ , сотрудник инкрементирует ему некоторое количество рейтинга, которое должно быть связано с оптимальностью сделанного выбора. Пусть  $M_k(t)$  есть маршрут, пройденный сотрудником  $k$  к моменту времени  $t$ ,  $D_k(t)$  – длина этого маршрута, а  $Q$  – параметр, имеющий значение порядка длины оптимального пути. Тогда инкрементируемое число рейтинга может быть задано в виде:

$$\Delta w_{ij,k}(t) = \begin{cases} \frac{Q}{D_k(t)}, & (i,j) \in M_k(t); \\ 0, & (i,j) \notin M_k(t); \end{cases} \quad (11)$$

Правила внешней среды определяют, в первую очередь, декрементацию рейтинга. Пусть  $\rho \in [0,1]$  это коэффициент декрементации, тогда правило декрементации имеет вид:

$$w_{ij}(t+1) = (1 - \rho)w_{ij}(t) + \Delta w_{ij}(t); \Delta w_{ij}(t) = \sum_{k=1}^V \Delta w_{ij,k}(t), \quad (12)$$

где  $V$  – количество сотрудников в корпорации.

Псевдокод реализации алгоритма может выглядеть следующим образом:

- 1) Ввод матрицы расстояний  $D_m$ ;
- 2) Инициализация параметров алгоритма –  $\alpha, \beta, \epsilon, Q$ ;
- 3) Инициализация рёбер - присвоение видимости  $\eta_{ij}$  и начальному значению рейтинга;
- 4) Размещение сотрудников по случайно выбранным клиентам без совпадений;
- 5) Выбор начального кратчайшего маршрута и определение  $D^*$ ;
- 6) Цикл по времени обновления  $t=1, t_{\max}$ ;
- 7) Цикл по всем сотрудникам  $k=1, m$ ;
- 8) Построить маршрут  $M_k(t)$  по правилу (10) и рассчитать длину  $D_k(t)$ ;
- 9) конец цикла по сотрудникам;
- 10) Проверка всех  $D_k(t)$  на лучшее решение по сравнению с  $D^*$ ;
- 11) В случае если решение  $D_k(t)$  лучше, обновить  $D^*$  и  $M^*$ ;
- 12) Цикл по всем рёбрам графа;
- 13) Обновить значение рейтинга на ребре по правилам (11) и (12);
- 14) конец цикла по рёбрам;
- 15) конец цикла по времени;
- 16) Вывести кратчайший маршрут  $M^*$  и его длину  $D^*$ ;

### Выводы

Анализ, проведенный в работе, показал, что задача распределения напрямую зависит от задачи оптимизации маршрута, которая может быть формализована как частный случай задачи коммивояжера с дополнительными ограничениями по времени. Предложен способ решения задачи с применением муравьиного алгоритма. Дальнейшая работа будет направлена на реализацию данного метода и его экспериментальное исследование в рамках поставленной задачи.

### Список литературы

1. Чураков М., Якушев А. «Муравьиные алгоритмы», 2006 г.
2. Штовба С.Д. Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях, 2003, №4, с.70-75.
3. МакКоннелл Дж. Основы современных алгоритмов. — М.: Техносфера, 2004.
4. Thompson, Jonathan. Ant Colony Optimization.
5. Barker T. and Von Haartman M. Ant Colony Optimization.
6. А.А. Кажаров, В.М.Курейчик «Об одном «муравьином» алгоритме».