

УДК 681.5

Г.О. Иванов, Н.Е. Опутин, П.Н. Игнатьев, А.А. Утин
Пермский национальный исследовательский
политехнический университет, г. Пермь

АВТОМАТИЗАЦИЯ РАСПРЕДЕЛЕНИЯ ЗАЯВОК ВЫЗОВА АВТОМОБИЛЕЙ СКОРОЙ ПОМОЩИ

Метод автоматизации распределения автомобилей скорой помощи на заявки пациентов, с учетом возможности перераспределения автомобилей в пути, позволяющий закрыть все заявки за минимальное итоговое время.

Ключевые слова: автоматизация; заявки, скорая помощь; пациенты; GRIN; Excel.

G.O. Ivanov, N.E. Oputin, P.N. Ignatiev, A.A. Utin
Perm national research polytechnic university, Perm

AUTOMATION OF THE DISTRIBUTION OF AMBULANCE ORDER APPLICATIONS

A method for automating the distribution of ambulances for patients' requests, taking into account the possibility of redistributing vehicles on the way, which allows you to close all requests in the minimum total time.

Key words: automation; applications, ambulance; the patients; GRIN; Excel.

В период пандемии, огромную нагрузку испытывают все социальные службы. Но наибольший удар пришелся на систему здравоохранения, которая не справляется с огромным числом поступающих заявок. В данной статье разбирается способ проведения автоматизации, направленный на распределение ресурсов таким образом, чтобы, в силу их ограниченности, использовать эти ресурсы наиболее эффективно.

Call-центры скорой помощи занимаются обработкой звонков от пациентов. После получения всей необходимой информации от звонившего, специалистами принимается решение о необходимости отправки экипажа скорой помощи. Если такая необходимость присутствует, выделение конкретного экипажа происходит вручную специалистом, основываясь на оперативной информации (адрес звонившего, местоположение свободных автомобилей скорой помощи), отображаемой на карте. После того как на заявку будет выделен экипаж, он становится недоступным до завершения текущей заявки.

Предлагаемое решение автоматизации несет в себе два ключевых аспекта: автомобили скорой помощи будут выделяться программой, основываясь на расчете наименьшей по времени продолжительности пути до пациента и автомобили могут перераспределять на другие, более экстренные, ближайшие вызовы. Далее, в статье будет описана реализация предлагаемой автоматизации.

Первым этапом необходимо получить оперативную информацию, на основе которой будет осуществляться работа программы. В качестве примера, был рассмотрен сектор города Перми, в пределах которого будут находиться как пациенты, так и автомобили скорой помощи.

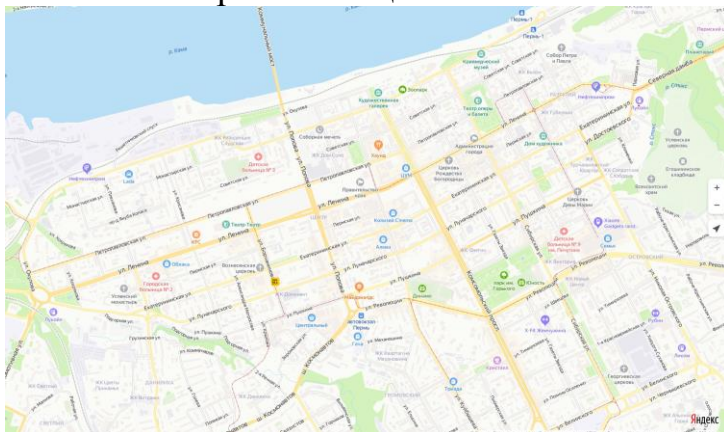


Рисунок 1 – Выделенный сектор города Перми (1:200)

На выбранном секторе для поиска необходимых величин будут играть роль дороги, по которым могут передвигаться автомобили скорой помощи и точки их нахождения/прибытия.



Рисунок 2 – Выделенные дороги для передвижения

Для упрощения расчетов, в качестве искомым точек будут братья перекрестки, где происходит пересечение нескольких дорог. Поскольку, вызов может поступить из участка, который не будет лежать ни на одной точке (справедливо и для скорой, которая не будет лежать на взятых точках), положение такого участка мы будем накладывать на ближайшую к нему точку, тем самым допуская незначительную погрешность при расчете фактического расстояние от скорой до пациента.

Таким образом, мы можем построить граф описывающий передвижение скорой помощи по выделенному сектору города. Обозначив условную координату нулевой точки в системе $[x; y]$, определим для каждой точки координату.

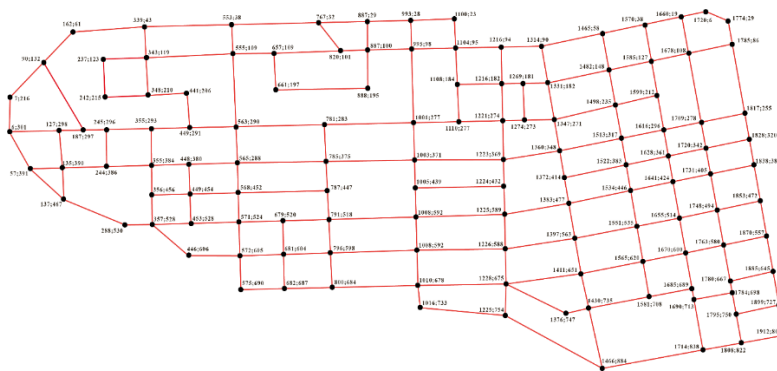


Рисунок 3 – Граф маршрута скорой помощи и значения вершин

Величину ребер графа, соответствующих дорогам, будем определять статически, беря в расчет время, которое потребуется автомобилю скорой помощи на преодоление участка дороги от одного перекрестка к другому. Мы рассматриваем ситуацию, при которой скорая помощь использует включенные сигнальные огни, что позволяет ей игнорировать заторы на дорогах. Для получения более точных расчетов, стоит определять величину ребер графа динамически, беря в расчет текущую ситуацию на дороге и движение скорой без сигнальных огней. В рамках данной статьи мы ограничимся единичным определением величин ребер, используя для этого найденные координаты вершин и среднюю скорость движения автомобиля скорой – 55 км/ч. [1]

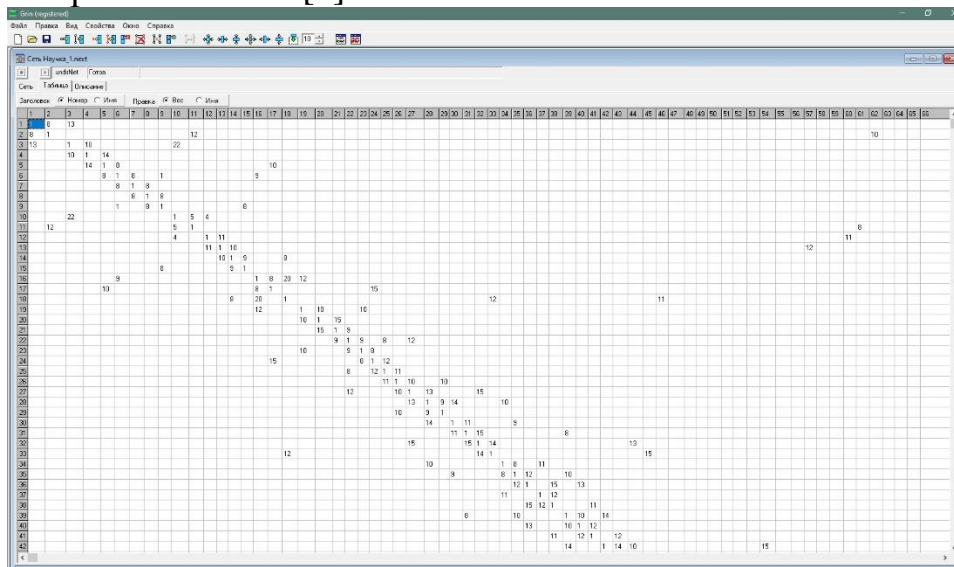


Рисунок 4 – Значение ребер графа маршрута

Используя программу “Grin”, построим получившийся граф, указав найденные значения величин ребер. Используя функцию “Кратчайший путь”, получаем значение, соответствующее минимальному времени, которое потребуется выбранной скорой для прибытия к пациенту. Выполняя данную функцию для каждого автомобиля скорой, мы получим множество значений времен, за которое каждый из автомобилей доберется до пациента. [2]

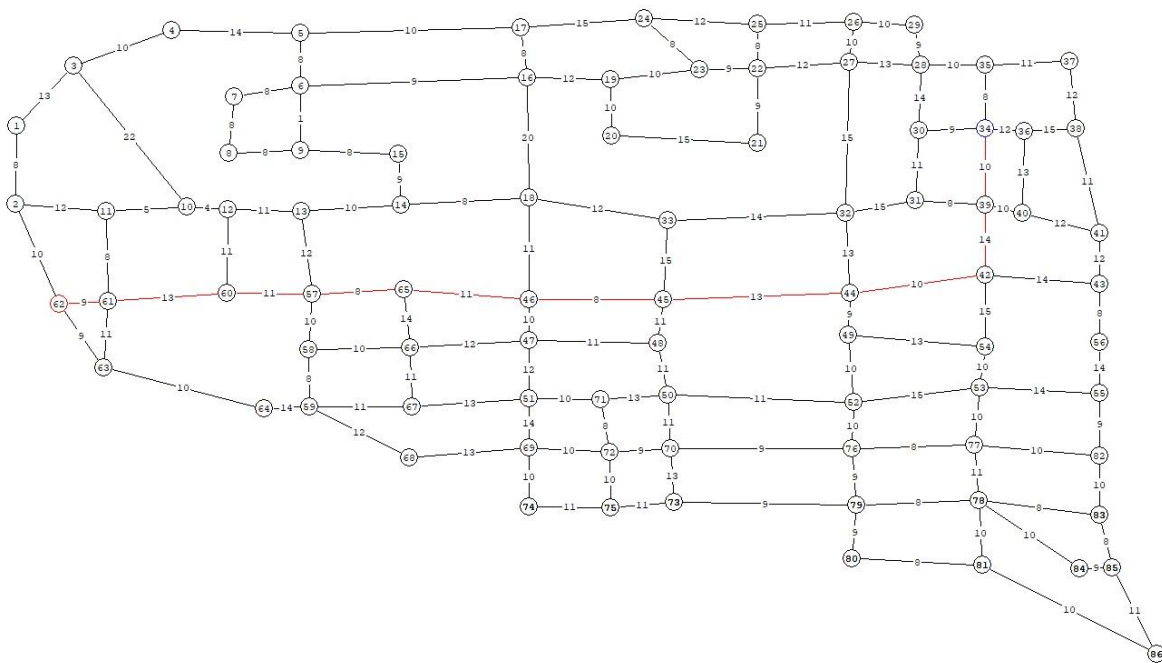


Рисунок 5 – Граф маршрута скорой помощи (значение ребер в секундах)

Определить состояние скорой помощи и наличие заявки на вызов от пациента мы сможем с помощью 2 файлов: “Скорая.txt” и “Люди.txt”. Первый файл содержит текущее состояние автомобиля скорой помощи, свободен он или нет. Во втором файле отображается состояние заявки, закрыта она или нет. Занесением скорых и пациентов занимаются сотрудники Call-центра, а их состояние будет обновлять уже автоматически.

Скорая.txt – Блокнот

Файл	Правка	Формат	Вид	Справка
Скорая	1	1		
Скорая	2	1		
Скорая	3	1		
Скорая	4	1		
Скорая	5	1		

Люди.txt – Блокнот

Файл	Правка	Формат	Вид	Справка
Пациент	1	1		
Пациент	2	1		
Пациент	3	1		
Пациент	4	1		
Пациент	5	1		

Рисунок 6 – Структура файлов “Скорая.txt” и “Люди.txt”

Для дальнейшего расчета используется среда построения электронных таблиц “Excel”. В ячейке A1 находится кнопка, при нажатии которой выполняется макрос. Макрос заполняется столбец A данными из файла “Скорая.txt”. Если у скорой стоит статус 1, то она заносится зеленым цветом, если 0, то красным. В строку 1 заносятся данные из файла “Люди.txt”. Если у пациента стоит статус 0, то он не будет занесен в таблицу. На пересечения строк и столбцов вносятся значение времен, получаемые при расчете программой “GRIN”.


	A	B	C	D	E	F
1		Пациент 1	Пациент 2	Пациент 3	Пациент 4	Пациент 5
2	Скорая 1	151	160	319	246	232
3	Скорая 2	330	343	165	154	345
4	Скорая 3	132	136	138	173	137
5	Скорая 4	256	280	196	200	335
6	Скорая 5	243	269	251	293	249

Рисунок 7 – Заполнение таблицы в Excel

Для того чтобы иметь возможность перераспределить автомобили скорой, мы установим частоту обновления = 1 мин. Это означает, что каждую 1 минут будут проверяться файлы “Скорая.txt” и “Люди.txt” на наличие новых данных, а также будет обновляться текущее положение автомобилей скорой помощи на графе, что позволит осуществить перерасчет затрат времени обновленных длин путей с помощью “GRIN”. Также будет обновляться и таблица в “Excel”.

Чтобы исключить ситуацию, когда автомобилю скорой помощи, находящемуся уже у пациента, придется ехать на новый адрес, установим пороговое значение, которое будет менять статус автомобиля на 0 (автомобиль не доступен). Порогом будет значение времени в 2 минут, через которое скорая прибудет к пациенту. Первый автомобиль, который перейдет установленное пороговое значение, будет закреплен за соответствующим пациентом. В таблице это будет означать, что при появлении в ячейке значения времени от 0 до 5, строка скорой заполняется значением -1, Скорая в столбце А окрашивается в красный цвет, столбец с пациентом удаляется, а в отчетном файле “Gotovo.txt” появляется запись о том, какая скорая везет какого пациента. В файлы “Скорая.txt” и “Люди.txt” так же вносятся изменения в статус скорой и удаляется закрытый пациент.


	A	B	C	D
1		Пациент 1	Пациент 3	Пациент 4
2	Скорая 1	174	253	163
3	Скорая 2	-1	-1	-1
4	Скорая 3	279	192	221
5	Скорая 4	-1	-1	-1
6	Скорая 5	293	229	222

Рисунок 8 – Изменение в таблице решений

```
*Gotovo.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
Скорая 2 везет Пациент 2
Скорая 4 везет Пациент 5
```

Рисунок 9 – Появление записи в файле “Gotovo.txt”

Люди.txt – Блокнот			Скорая.txt – Блокнот				
Файл	Правка	Формат	Файл	Правка	Формат	Вид	Справка
Пациент	1	1	Скорая	1	1		
Пациент	3	1	Скорая	2	0		
Пациент	4	1	Скорая	3	1		
			Скорая	4	0		
			Скорая	5	1		

Рисунок 10 – Изменение записей в файлах “Скорая.txt” и “Люди.txt”

По полученной таблице данных, с частотой обновления 1 мин, каждый раз будет проводиться поиск решения как для транспортной задачи с приоритетом по времени, то есть будет определяться оптимальное распределение автомобилей скорой помощи для каждого пациента с минимальной затратой общего времени. Для этого, таблица должна иметь одинаковое количество столбцов и строк. Изначально, число автомобилей скорой помощи у нас фиксировано. Если для равенства нам не хватает пациентов, то мы будем добавлять столбцы с нулевыми пациентами, для которых выделять автомобиль не нужно. Если пациентов больше, чем автомобилей, то каждого нового пациента мы пробуем поставить в таблицу вместо каждого из старых. Если для новой таблицы будет найдено решение лучше, то оставляем ее в новом виде, иначе, не меняем. И так поступаем для каждого нового пациента, добавленного сверх количества автомобилей скорой помощи.

Оптимальнее всего осуществлять поиск решения будет с помощью линейного симплекс метода, реализация которого приводится в ранее опубликованной статье “Транспортная задача. Реализация по критерию времени в программе MS Excel” [3].

Таким образом, найденное решение будет соответствовать оптимальному распределению автомобилей скорой помощи, чтобы как можно быстрее по времени закрыть все заявки. Выполнение будет продолжаться до тех пор, пока все заявки не будут закрыты.

Список литературы

1. The Village. Как устроен «Яндекс.Навигатор». Как алгоритм строит маршрут. - [Электронный ресурс]. – URL: <https://www.the-village.ru/business/how/278022-naviga>. (дата обращения: 10.12.2020). Бесекиерский В.А. Теория систем автоматического управления. – СПб.: Профессия, 2003. – 752 с.
2. Печенкин В. Программное обеспечение Grin. О программе - [Электронный ресурс]. – URL: www.geocities.com/pechv_ru (дата обращения: 11.12.2020).
3. Научная статья Г.О. Иванова – “Транспортная задача. Реализация по критерию времени в программе MS Excel” из научного журнала “Вестник Пермского университета” за 2020 г.

Сведения об авторах

Иванов Глеб Олегович – магистрант Пермского национального исследовательского политехнического университета, Пермь, email: gleb_molodoi5@mail.ru

Опутин Никита Егорович – магистрант Пермского национального исследовательского политехнического университета, Пермь, email: oputin1906@mail.ru

Игнатьев Павел Николаевич – магистрант Пермского национального исследовательского политехнического университета, Пермь, email: homakig@gmail.com

Утин Антон Александрович – магистрант Пермского национального исследовательского политехнического университета, Пермь, email: Poickovizok@yandex.ru

About the authors

Ivanov Gleb Olegovich – Master's student of the Perm National Research Polytechnic University, Perm, email: gleb_molodoi5@mail.ru

Oputin Nikita Egorovich – Master's student of the Perm National Research Polytechnic University, Perm, email: oputin1906@mail.ru

Ignatiev Pavel Nikolaevich – Master's student of the Perm National Research Polytechnic University, Perm, email: homakig@gmail.com

Utin Anton Aleksandrovich – Master's student of the Perm National Research Polytechnic University, Perm, email: Poickovizok@yandex.ru