

**А.А. ОВАНЕСЯН,**

аспирант, Институт программных систем им. А.К. Айламазяна РАН, Россия,  
e-mail: ovanesyana@interin.ru

**А.В. ЛЕВИЧЕВ,**

аспирант, Институт программных систем им. А.К. Айламазяна РАН, Россия,  
e-mail: levichev@interin.ru

**Д.В. БЕЛЫШЕВ,**

к.т.н., Институт программных систем им. А.К. Айламазяна РАН, Россия,  
e-mail: belyshev@interin.ru

**А.М. ЦИРЛИН,**

д.т.н., профессор, Институт программных систем им. А.К. Айламазяна РАН, Россия,  
e-mail: tsirlin@sarc.botik.ru

## ЗАДАЧИ РАСПРЕДЕЛЕНИЯ МЕДИЦИНСКИХ НАПРАВЛЕНИЙ

УДК 61:007 (Медицинская кибернетика)

Ованесян А.А., Левичев А.В., Бельшев Д.В., Цирлин А.М. *Задачи распределения медицинских направлений* (Институт программных систем им. А.К. Айламазяна РАН, Россия)

**Аннотация.** Обсуждаются постановки и алгоритмы решения задач о формировании маршрутов медицинских обследований и лечения. Рассмотрены задачи об оптимальном маршруте для конкретного пациента с учетом расписания работы врачей и о распределении направлений для группы пациентов с одинаковыми и различающимися требованиями к медицинскому обслуживанию.

**Ключевые слова:** медицинское обслуживание, выбор последовательности направлений, пункты обслуживания, оптимальный маршрут, обслуживание группы.

UDC 61:007 (Medical Cybernetics)

Ovanesyana A.A., Levichev A.V., Belyshev D.V., Tsirlin A.M. *Tasks of distribution of medical appointments* (Ailamazyan Program Systems Institute of RAS, Russia)

**Abstract.** The statements and algorithms for solving problems on the formation of routes for medical examinations and treatment are discussed. The problems of the optimal route for an individual patient are considered taking into account the doctors' work schedule and the distribution of appointments for a group of patients with the same and different medical care requirements.

**Keywords:** medical care, choice of sequence of appointments, service points, optimal route, group service.

### ВВЕДЕНИЕ

Одной из важных задач обработки информации в медицинских информационных системах является задача о составлении расписаний обслуживания пациентов и распределении врачебных направлений.

Решение таких задач без привлечения информационных технологий может привести к неравномерной загрузке обслуживающего персонала и потерям времени пациентов на перемещение от одного пункта обслуживания (ПО) к другому и ожиданию приема.

Подобные задачи возникают в разных областях и наиболее близки к задачам составления расписаний, рассмотренными в [1, 2], коммивояжера (см. [3, 4]). Однако задачи распределения медицинских направлений имеют свою специфику, им посвящены исследования [8, 9].

В классической задаче коммивояжера требуется найти замкнутый маршрут между  $n$  пунктами, для каждой пары которых задана продолжительность перемещения  $\delta_{ij}$ . При этом общая продолжительность маршрута  $T(v^*)$  должна быть минимальна. Алгоритмы решения такой задачи детально отработаны.



В медицинской практике необходимо учесть не только продолжительность перемещения пациента, но и продолжительность ожидания приема, связанную с расписанием работы ПО. При этом продолжительность ожидания зависит от календарного времени прихода пациента в каждый  $i$ -й ПО, а значит, от его предыдущего маршрута.

ПО иногда частично упорядочены – бывает, что пациент не может пройти одни обследования, не пройдя предварительно какие-то другие, или нельзя выполнять некие процедуры сразу после других.

Второй важной особенностью задач формирования маршрутов обслуживания является то, что оптимальный маршрут часто нужно формировать не для одного пациента, а для группы или постоянного потока пациентов, поступающих в медицинское учреждение. Здесь нужно стремиться к минимальным потерям времени на обслуживание в среднем для группы пациентов. При этом сама группа может быть неоднородной, так как у одних пациентов список необходимых ПО может быть отличен от аналогичного списка для других.

В упомянутых выше работах применяется стохастическое моделирование, временные характеристики, универсальные методы планирования и эвристические алгоритмы. В данном исследовании рассмотрены две задачи: выбор маршрута обслуживания конкретного пациента, составление маршрутов для группы пациентов. В решении используется один алгоритм с различными исходными данными, что для информационных технологий является плюсом.

На рынке медицинских информационных систем предлагаются решения, которые предоставляют возможность обеспечить информационную поддержку процессам составления расписания обслуживания пациентов, автоматизацию бизнес-процесса: направление – диспетчеризация – исполнение. К ним относятся такие системы как «Здравница» [5], «Кинт: Управление санаторием» [6], «Санаториум» [7] и ряд других. Алгоритмы распределения медицинских направлений этих систем в открытом доступе недоступны, поэтому оценить их эффективность не представляется возможным. Однако интерес производителей показывает актуальность данной задачи.

### Задача выбора маршрута обслуживания конкретного пациента

Пусть заданы:

$j = 1, \dots, n$  – ПО, которые должен пройти пациент;

$\tau_j$  – продолжительность обслуживания в каждом  $j$ -ом пункте;

$t_j$  – моменты (расписание приема для каждого ПО), в которые  $j$ -ый ПО может принять очередного пациента;

$t_0$  – момент начала движения пациента;

$\delta_{ij}$  – продолжительность перемещения пациента из  $i$ -го в  $j$ -ый пункт. Если посещение  $j$ -го пункта невозможно (нельзя пройти  $j$ -й ПО после  $i$ -го, не проходя некоторого другого), то эта продолжительность сколь угодно велика;  $\delta_{ii} = \infty$ ;  $\delta_{ij} \neq \delta_{ji}$ .

Обозначим через  $v$  – индекс маршрута, то есть последовательности прохождения ПО. Число возможных маршрутов равно  $n!$  и уже для  $n = 8$  составляет 40320.

Пусть  $T_{jv}$  – продолжительность достижения пациентом  $j$ -го пункта из заданного начального при выборе  $v$ -го маршрута, а  $T_{ov}$  – продолжительность ожидания приема в  $j$ -м ПО.

В качестве критерия оптимальности выберем общую продолжительность обслуживания пациента, зависящую от выбранного маршрута,  $T_v$ :

$$T(v^*) = \min_v T_v. \quad (1)$$

Управлением является вектор  $U$  с составляющими  $u_{ij}$ , который определяет в какой  $j$ -ый ПО должен направиться пациент, прошедший  $i$ -ый ПО. Эти составляющие принимают значение ноль, если пациент не направлен в  $j$ -ый ПО, и единицу, если он направлен. Поскольку для любого пройденного ПО пациент направляется только в один пункт, то:

$$\sum_i u_{ij} = 1 \forall j, \sum_j u_{ij} = 1 \forall i. \quad (2)$$

Величина  $T_v$  состоит из трех слагаемых:

1. Продолжительности перемещения пациента по всем заданным ПО при выбранном  $v$ -ом маршруте:

$$T_{dv} = \sum_{i,j=1}^n \delta_{ij} u_{ij}^v. \quad (3)$$

2. Суммарной продолжительности обслуживания во всех пунктах  $T_0 = \sum_{j=1}^n \tau_j$ , которая не зависит от выбора маршрута.

3. Суммарной продолжительности ожидания приема.

Продолжительность ожидания приема в  $j$ -м ПО  $v$ -го маршрута равна разнице между ближайшим моментом приема в этом ПО и моментом готовности пациента к обслуживанию (предполагается



отсутствие очереди). Календарный момент готовности к обслуживанию равен сумме момента начала обслуживания  $t_0$ , суммарной продолжительности перемещения и обслуживания на предыдущих ПО  $T_{j-1,v}$  и суммарной продолжительности ожидания на предыдущих ПО. Так что продолжительность ожидания приема удовлетворяет рекуррентному соотношению:

$$T_{0j}^v = \min_{t_j} \left[ t_j - t_0 - T_{j-1,v} - \sum_{i=0}^{j-1} T_{oi}^v \right], T_{00}^v = 0 \quad \forall v. \quad (4)$$

Здесь величина

$$T_{j-1,v} = \sum_{i=1}^j \delta_{ij} u_{ij}^v + \tau_i. \quad (5)$$

Минимум в (4) ищется на множестве тех моментов  $t_j$ , для которых значение  $T_{0j}^v$  не отрицательно. Это множество зависит от выбранного маршрута. Множество возможных моментов приема включает только те моменты, которые не выписаны предыдущим пациентам, что исключает образование очереди.

Будем называть сумму продолжительностей перемещения и ожидания дополнительными затратами времени. Оптимальный маршрут предполагает достижение минимума этих дополнительных затрат.

Оценку снизу для минимального времени обслуживания  $T(v^*)$  нетрудно получить, предположив, что время ожидания во всех ПО близко к нулю, и найдя маршрут  $v_0$ , соответствующий минимуму суммарного времени перемещения между ПО. Расчет такого маршрута сводится к решению известной задачи коммивояжера [3, 4].

Задача коммивояжера допускает матричную интерпретацию. Пусть по столбцам и строкам матрицы направлений отложены ПО. На пересечении каждого столбца с каждой строкой содержится «расстояние»  $\delta_{ij}$  между соответствующими ПО. Выделение элемента матрицы соответствует направлению пациента в  $j$ -ый ПО, если он прошел  $i$ -ый. Так как пациенту не нужно возвращаться в пройденный ПО, то  $\delta_{ii} = \infty$ .

Задача заключается в том, чтобы выделить в каждой строке матрицы по одному элементу, так, чтобы выделенные элементы находились в разных столбцах, и при этом условии сумма расстояний по всем выделенным элементам была минимальна. Выбор вектора управления оптимизирует матрицу направлений.

В качестве «расстояния» могут фигурировать продолжительность перемещения пациента, стоимость перемещения и прочее. В нашей задаче

расстояние соответствует затратам времени на переход в очередной ПО. В задаче с разомкнутым маршрутом, в которой задан начальный ПО (регистратура), выделенные элементы в матрице переходов должны располагаться во всех столбцах, кроме того, который соответствует начальному ПО, и во всех строках, кроме той, которая окажется соответствующей конечному ПО.

Алгоритмы решения задачи коммивояжера тщательно изучены и мы не будем здесь на них останавливаться. Кратко опишем лишь алгоритм «ветвей-границ», поскольку он применен для задач рассмотренных ниже.

Основные этапы алгоритма для задачи с заданным начальным ПО и разомкнутым маршрутом:

1. Строят матрицу переходов, исключив столбец, соответствующий начальному ПО.
2. В каждой строке находят минимальное расстояние  $d_i$  и вычитают этот минимум из всех элементов строки.
3. Получают редуцированную матрицу, в каждой строке которой стоят приросты расстояния по отношению к минимальному. Каждая строка содержит не менее одного нуля.
4. В каждом столбце редуцированной матрицы приростов находят минимальный элемент  $d_i$  и вычитают его из каждого элемента соответствующего столбца.
5. Получают вторую редуцированную матрицу, в каждой строке и в каждом столбце которой имеется не менее одного нуля.
6. Оценивают каждый из нулей. Оценкой нуля является сумма минимальных элементов второй редуцированной матрицы вычисленных по столбцу и по строке, в пересечении которых стоит оцениваемый нуль. Эта сумма является нижней оценкой прироста суммарного расстояния в том случае, если клетка, соответствующая положению оцениваемого нуля, не будет выбрана в качестве перехода.
7. Выбирают в качестве перехода ту клетку матрицы переходов, которой соответствует во второй редуцированной матрице нуль с максимальной оценкой.
8. Исключают из матрицы переходов строку и столбец, для которых выбран переход, а в клетке, соответствующей обратному переходу для выбранных ПО, ставят бесконечность.
9. Возвращаются к пункту 2.

Заметим, что величину минимального суммарного расстояния для разомкнутого маршрута легко



оценить снизу. Эта оценка равна сумме  $d_i$  по всем строкам матрицы, кроме той, которая окажется соответствующей последнему на оптимальном маршруте ПО:

$$L^* = \sum_{i=1}^n d_i - d_f. \quad (6)$$

Поскольку до решения задачи мы не знаем какой ПО окажется последним, то для получения несколько более грубой нижней оценки можно в правой части равенства (6) вместо  $d_f$  вычесть максимальное по  $i$  значение  $d_i$ .

Маршрут, найденный при допущении того, что время ожидания пренебрежимо мало, будем называть «идеальным». Продолжительность обслуживания на идеальном маршруте равна сумме минимальной продолжительности перемещения и  $T_0$ .

Отметим, что приведенный алгоритм оптимизации матрицы направлений является общим и может быть использован и в том случае, когда эта матрица прямоугольная, а не квадратная.

Равенства (3), (4) позволяют для любого выбранного маршрута найти продолжительность обслуживания. Одним из таких маршрутов может быть «близорукий» маршрут  $v_b$ . При формировании этого маршрута в момент  $t_0$  из всех потребных ПО выбирают в качестве первого тот, для которого дополнительные затраты времени на перемещение и ожидание из начального пункта (регистратуры) минимальны.

Вычисляют момент готовности пациента к дальнейшему обслуживанию  $t_i$  с учетом прохождения первого ПО и выбирают следующий пункт вновь по минимуму дополнительных затрат времени. Число перебираемых вариантов на первом шаге равно  $n$ , на втором  $n-1$  и так далее. Для восьми ПО общее число расчетов дополнительных затрат времени составит 35. Продолжительность обслуживания на «близоруком» маршруте (суммарные дополнительные затраты плюс  $T_0$ ) является оценкой сверху для искомой минимальной продолжительности обслуживания. Так что:

$$T(v_b) \geq T(v^*) \geq T(v_0). \quad (7)$$

Другим способом получения верхней оценки критерия оптимальности является расчет по формуле (4) суммарной продолжительности ожидания приема для выбранного «идеального» маршрута. Пусть она равна  $\Delta T(v_0)$ . Тогда неравенства (7) примут форму:

$$T(v_0 + \Delta T(v_0)) \geq T(v^*) \geq T(v_0). \quad (8)$$

Если в каком-либо из неравенств (7) или (8) оценки сверху и снизу отличаются не более, чем, например, на 5%, то маршрут, доставляющий более низкую оценку сверху, «близорукий» либо «идеальный», но с реальным ожиданием приема, может быть принят в качестве оптимального.

В противном случае может быть использован алгоритм поиска маршрута, уменьшающего суммарное время обслуживания. Для этого введем в «близорукий» алгоритм процедуру учета времени ожидания приема. А именно, на первом шаге для выбора первого пункта находим  $l < n$  ПО, для которых дополнительные затраты времени минимальны. Число таких «перспективных» ПО может составлять выбранную долю от  $n$ . Каждому из выбранных ПО с учетом продолжительности обслуживания соответствует свой момент готовности пациента к новому обслуживанию  $t_{ij}, j = 1, \dots, l$ . Для каждого такого момента можно найти группу  $l$  «перспективных» ПО и среднюю для этой группы величину дополнительных затрат времени (вероятные дополнительные затраты).

Выбор первого пункта осуществляем так, чтобы сумма дополнительных затрат на первом шаге и вероятных дополнительных затрат, зависящих от выбора первого ПО, была минимальна. То есть в алгоритме с учетом времени ожидания приема выбирают ПО на первом шаге с учетом вероятной величины дополнительных затрат на втором шаге.

На следующем шаге расчет повторяют по тому же правилу, исключив выбранный на первом шаге пункт. При этом группа «перспективных ПО» уже найдена, так что требуется только для каждого из них найти вероятные дополнительные затраты на перемещение и ожидание приема на третьем шаге.

Прогнозируют дополнительные затраты на два шага, но реализуют только первый. Число перебираемых вариантов даже в случае, когда все пункты на каждом шаге рассматривают как «перспективные», для восьми ПО равно 168.

### Задача распределения направлений при обслуживании группы пациентов

Результатом предыдущей задачи является маршрут для одного пациента. Во многих случаях необходимо составить расписание обслуживания для



группы пациентов. Например, для прохождения профосмотров, когда предприятие направляет своих сотрудников в лечебное учреждение. В этом случае все пациенты равноправны между собой и надо сократить средние дополнительные затраты времени для группы. Не является оптимальным решение, для которого первоначально выбирают оптимальный маршрут первого пациента с использованием изложенного выше алгоритма, затем делают то же для следующего пациента, исключив моменты приема, которые заняты первым, и так далее, так как выбор для первых влияет на возможности выбора маршрутов для последующих пациентов.

Естественным критерием для группы пациентов является минимум среднего времени обслуживания группы, чему соответствует максимум пропускной способности системы ПО. Эта пропускная способность ограничена «узким местом», то есть числом пациентов, которое может пропустить ПО, имеющий максимальное значение продолжительности обслуживания  $\tau^* = \max_j \tau_j$ . Назовем этот пункт «критическим». Если продолжительность работы ПО  $T_R$  часов в сутки, то пропускная способность каждого ПО равна:

$$C_j = \frac{T_R}{\tau_j}$$

пациентов в день. Для «критического» ПО пропускная способность  $C^0$  минимальна. Если число пациентов в группе больше, чем  $C^0$ , то обслуживание должно продолжаться больше одного дня. Если оно меньше, то нужно обеспечить такую выдачу направлений, чтобы «критический» ПО успел принять всех пациентов. Таким образом нужно, чтобы дополнительные суммарные затраты времени пациентов были минимальны и при этом «критический» ПО был полностью загружен.

Введем определения:

1. Состояние системы «Пациенты-ПО» характеризуется для каждого  $i$ -го пациента ( $i=1, \dots, m$ ) моментом готовности к обслуживанию  $t_i$ , не пройденными им ПО с их моментами приема  $t_j$ , ближайшими и большими, чем  $t_i$ . Время ожидания приема  $T_{oij} = t_j - t_i$ . Еще одной составляющей вектора состояния является момент календарного времени начала стадии  $t_k$ . Таким образом, вектор состояния  $X_k$  имеет размерность  $n_k + m_k + 1$  с неотрицательными составляющими.

2. Назовем  $k$ -ой стадией процесса обслуживания переход системы «Пациенты-ПО» из одного состояния в другое, при котором хотя бы один из

пациентов прошел один ПО. Начальное состояние задано.

3. Будем называть «расстоянием»  $\delta_{ij}^k(t_k, X_k)$  между пациентом и непройденным пунктом суммарные дополнительные затраты времени на перемещение в ПО и ожидание приема в этом пункте. Расстояние зависит от состояния системы и меняется от стадии к стадии. Меняется и число пациентов  $m_k$ , не прошедших все ПО, и число ПО  $n_k$ , через которые не прошли все пациенты. Причем эти числа монотонно уменьшаются. Матрицу, в которой «расстояния» стоят на пересечении строки пациента и столбца ПО, называют матрицей направлений.

4. Вектором управления  $U_k$  на каждой стадии является вектор, определяющий в какой из ПО направлен на этой стадии каждый из пациентов. Он имеет размерность  $m_k n_k$ , его составляющие  $u_{ijk}$  принимают значение 1, если пациент направлен в данный ПО и ноль, если он не направлен. Так как каждый пациент на данной стадии направляется только в один ПО, то для вектора управления выполнены условия:

$$\sum_i u_{ijk} = 1 \forall j, k \quad \sum_j u_{ijk} = 1 \forall i, k. \quad (9)$$

Критерием оптимальности задачи является минимум суммарных затрат времени всеми пациентами на всех стадиях:

$$T = \sum_{k,i,j} \delta_{ij}^k(t_k, X_k) u_{ijk} \rightarrow \min / X_{k+1} = F(X_k, U_k). \quad (10)$$

Начальное состояние  $X_0$  известно. Связь  $F$  между состояниями на соседних стадиях и вектором управления задана алгоритмически через матрицу направлений. При этом:

$$t_{i,k+1} = t_{ik} + \sum_j (\delta_{ij} + \tau_j) u_{ijk}, \quad t_{k+1} = \min_i t_{i,k+1}, \quad (11)$$

где продолжительность обслуживания  $\tau_j$  предполагается одинаковой для всех пациентов. Моменты начала приема ближайшие и большие  $t_{i,k+1}$  изменяются с изменением моментов готовности.

Если  $\delta_{ij}$  бесконечно велико, а  $u_{ijk} = 0$ , то их произведение будем предполагать равным нулю.

### **Алгоритм распределения направлений**

Упорядочим ПО по продолжительности обслуживания, так что критическому ПО соответствует последний столбец. На пересечении  $i$ -ой строки



и  $j$ -го столбца в матрице направлений стоит «расстояние» между соответствующим пациентом и ПО, равное сумме продолжительности перехода пациента из пройденного последнего ПО в  $j$ -ый и продолжительности ожидания в нем. Чтобы вычислить расстояние и построить матрицу переходов нужно знать пройденный пациентом последний ПО, все непройденные и момент готовности  $t_i$ . Для пройденных ПО или для тех, которые нельзя пройти, не пройдя предварительно некоторого другого, расстояние  $\delta_{ij}$ , соответствующее элементу матрицы переходов, считается бесконечным.

Матрица направлений отличается от матрицы, использованной в предыдущем разделе, тем, что она не является квадратной и меняется от стадии к стадии. Некоторые направления могут быть заранее фиксированными по тем или иным причинам. В этом случае строку и столбец, соответствующие фиксированному направлению, исключают из матрицы, а расстояние, стоящее в фиксированном элементе матрицы, добавляют к минимальному расстоянию, найденному в результате решения.

Чтобы критический пункт был полностью загружен, алгоритм предполагает, что в любом состоянии первоначально выделяют минимальный по строкам элемент последнего столбца и соответствующий пациент получает направление в критический ПО. После этого число пациентов и число ПО уменьшается на единицу, за счет исключения из матрицы направлений строки, соответствующей выбранному пациенту и последнего столбца.

В терминах получившейся модифицированной матрицы задача заключается в том, чтобы в каждой ее строке выбрать один элемент, находящийся в столбце, не содержащем других выбранных элементов, так, чтобы сумма расстояний для всех выбранных элементов матрицы переходов была минимальной.

Такой выбор позволяет найти следующее состояние системы «Пациенты-ПО». При этом в новой матрице число элементов с бесконечным расстоянием возрастает, столбец или строку, в которых все элементы равны бесконечности, исключают, уменьшая размерность матрицы. Расчет заканчивается, когда все пациенты пройдут все предназначенные для каждого из них ПО.

Задача о выборе направлений для модифицированной матрицы совпадает с оптимизацией матрицы направлений в задаче выбора индивидуального маршрута обслуживания и может быть решена

с использованием тех же алгоритмов, в частности, изложенного выше алгоритма «ветвей-границ». В данном случае эту задачу приходится решать для каждого перехода от стадии к стадии. Справедлива и оценка  $\delta$ ), без вычитания  $d_j$ , для добавочных затрат времени на каждой стадии, а следовательно, и оценка затрат времени для всего процесса как суммы этих затрат по всем стадиям.

Результатом решения задачи является  $n$  взаимосвязанных маршрутов, для которых средняя продолжительность обслуживания пациента близка к минимуму.

Данный алгоритм не гарантирует получения оптимального решения, так как он игнорирует связь между состоянием на предыдущей и состоянием на текущей стадии, выбор управления на каждой стадии осуществляется только по минимуму суммарных дополнительных затрат на этой стадии. Поэтому полученное решение следует проверить на оптимальность, используя, как это было сделано для индивидуального маршрута, сравнение с нижней оценкой или варьирование траектории на двух соседних стадиях.

### Пример решения задачи распределения направлений при обслуживании группы пациентов

Рассмотрим ситуацию, когда количество пациентов  $m$  равно количеству пунктов  $n$ . Данные о ПО представлены в *таблице 1*. Продолжительность передвижения между ПО указана в *таблице 2*.

Любой ответ на начальной стадии будет локально оптимальным, так как «расстояние» у всех пациентов одинаковое. Начальное (нулевое) состояние системы «Пациенты-ПО»:

1. 1-й пациент:  $P_5$  (8:00–8:20);
2. 2-й пациент:  $P_4$  (8:00–8:05);
3. 3-й пациент:  $P_1$  (8:00–8:10);
4. 4-й пациент:  $P_3$  (8:00–8:10);
5. 5-й пациент:  $P_2$  (8:00–8:15).

Построим матрицу направлений для определения второго состояния системы «Пациенты-ПО» (*в таблице 3*). Для пройденных пациентами ПО расстояние считается бесконечным.

Оценка снизу по  $\delta$ ) при  $d_j = 0$  равна 25 минутам.

Чтобы критический пункт (у него самая большая продолжительность обслуживания) был загружен, выделяем минимальный элемент по строкам в столбце критического ПО. В данном случае он будет у 3-го пациента.



Таблица 1

**ПО**

ПО $P_j$	$\tau_j$	$t_j$
1. $P_1$ Врач-терапевт	10	(8:00, 8:10, 8:20, 8:30, 8:40, 8:50, 9:00, 9:10, 9:20, 9:30, 9:40, 9:50, 10:00, 10:10, 10:20)
2. $P_2$ Врач-оториноларинголог	15	(8:00, 8:15, 8:30, 8:45, 9:00, 9:15, 9:30, 9:45, 10:00, 10:15, 10:30, 10:45, 11:00, 11:15, 11:30)
3. $P_3$ Врач-психиатр	10	(8:00, 8:10, 8:20, 8:30, 8:40, 8:50, 9:00, 9:10, 9:20, 9:30, 9:40, 9:50, 10:00, 10:10, 10:20)
4. $P_4$ Врач-нарколог	5	(8:00, 8:05, 8:10, 8:15, 8:20, 8:25, 8:30, 8:35, 8:40, 8:45, 8:50, 8:55, 9:00, 9:05, 9:10, 9:15, 9:20, 9:25, 9:30, 9:35, 9:40, 9:45, 9:50, 9:55, 10:00, 10:05, 10:10, 10:15, 10:20, 10:25)
5. $P_5$ Врач-невролог	20	(8:00, 8:20, 8:40, 9:00, 9:20, 9:40, 10:00, 10:20, 10:40, 11:00, 11:20, 11:40, 12:00, 12:20, 12:40)

Таблица 2

**Продолжительность передвижения между ПО**

ПО	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$
$P_1$	$\infty$	5	3	4	2
$P_2$	5	$\infty$	5	5	5
$P_3$	3	5	$\infty$	2	4
$P_4$	4	4	2	$\infty$	4
$P_5$	3	6	5	4	$\infty$

Таблица 3

**Матрица направлений**

ПО \ Пациенты	$P_4$	$P_1$	$P_3$	$P_2$	$P_5$
1-й пациент	5	10	10	10	$\infty$
2-й пациент	$\infty$	5	5	10	15
3-й пациент	5	$\infty$	10	5	10
4-й пациент	5	10	$\infty$	5	10
5-й пациент	5	5	5	$\infty$	15

Таблица 4

**Редуцированная матрица направлений**

ПО \ Пациенты	$P_4$	$P_1$	$P_3$	$P_2$
1-й пациент	5	10	10	10
2-й пациент	$\infty$	5	5	10
4-й пациент	5	10	$\infty$	5
5-й пациент	5	5	5	$\infty$

Таблица 5

**Поиск минимального расстояния по строке**

ПО \ Пациенты	$P_4$	$P_1$	$P_3$	$P_2$	$d_i$
1-й пациент	5	10	10	10	5
2-й пациент	$\infty$	5	5	10	5
4-й пациент	5	10	$\infty$	5	5
5-й пациент	5	5	5	$\infty$	5



Убираем строку и столбец выбранного элемента. Для оставшихся элементов матрицы решаем задачу выбора направлений (в таблице 4), используя алгоритм ветвей-границ.

1. Находим минимальное расстояние  $d_i$  в строке и вычитаем этот минимум из всех элементов строки (в таблицах 5, 6).

2. Находим минимальное расстояние  $d_i$  в каждом столбце и вычитаем этот минимум из всех элементов столбца (в таблице 7).

3. Оцениваем нули второй редуцированной матрицы (в таблице 8).

4. Выбираем в качестве перехода ту клетку матрицы переходов, которая соответствует во второй редуцированной матрице нулю с максимальной оценкой (у 1-го пациента).

5. Исключаем из матрицы направлений строку и столбец, соответствующие клетке перехода. Повторяем действия с 1-го пункта. В таблице 9 представлена матрица направлений после исключения строки и столбца выбранного элемента.

Таблица 6

**Редуцированная по строкам матрица**

Пациенты \ ПО	$P_4$	$P_1$	$P_3$	$P_2$	$d_i$
1-й пациент	0	5	5	5	5
2-й пациент	$\infty$	0	0	5	5
4-й пациент	0	5	$\infty$	0	5
5-й пациент	0	0	0	$\infty$	5

Таблица 7

**Редуцированная по столбцам матрица**

Пациенты \ ПО	$P_4$	$P_1$	$P_3$	$P_2$	$d_i$
1-й пациент	0	5	5	5	5
2-й пациент	$\infty$	0	0	5	5
4-й пациент	0	5	$\infty$	0	5
5-й пациент	0	0	0	$\infty$	5
$d_j$	0	0	0	0	

Таблица 8

**Оценка нулей**

Пациенты \ ПО	$P_4$	$P_1$	$P_3$	$P_2$	$d_i$
1-й пациент	0 (5)	5	5	5	5
2-й пациент	$\infty$	0 (0)	0 (0)	5	5
4-й пациент	0 (0)	5	$\infty$	0 (5)	5
5-й пациент	0 (0)	0 (0)	0 (0)	$\infty$	5
$d_j$	0	0	0	0	

Таблица 9

**Матрица направлений для поиска третьего элемента**

Пациенты \ ПО	$P_1$	$P_3$	$P_2$
2-й пациент	5	5	10
4-й пациент	10	$\infty$	5
5-й пациент	5	5	$\infty$



В итоге после выбора всех элементов в матрице направлений первое состояние системы «Пациенты-ПО»:

1. 1-й пациент:  $P_5 (8:00-8:20) \rightarrow P_4 (8:25-8:30)$ ;
2. 2-й пациент:  $P_4 (8:00-8:05) \rightarrow P_1 (8:10-8:20)$ ;
3. 3-й пациент:  $P_1 (8:00-8:10) \rightarrow P_5 (8:20-8:40)$ ;
4. 4-й пациент:  $P_3 (8:00-8:10) \rightarrow P_2 (8:15-8:30)$ ;
5. 5-й пациент:  $P_2 (8:00-8:15) \rightarrow P_3 (8:20-8:30)$ .

Время переходов равно 30 минут, что на 5 минут больше оценки снизу.

По аналогии с предыдущими шагами определяем следующие состояния системы.

Второе состояние системы «Пациенты-ПО»:

Оценка снизу равна 35 минут.

1. 1-й пациент:  $P_5 (8:00-8:20) \rightarrow P_4 (8:25-8:30) \rightarrow P_1 (8:40-8:50)$ ;
2. 2-й пациент:  $P_4 (8:00-8:05) \rightarrow P_1 (8:10-8:20) \rightarrow P_3 (8:30-8:40)$ ;
3. 3-й пациент:  $P_1 (8:00-8:10) \rightarrow P_5 (8:20-8:40) \rightarrow P_2 (8:45-9:00)$ ;
4. 4-й пациент:  $P_3 (8:00-8:10) \rightarrow P_2 (8:15-8:30) \rightarrow P_5 (8:40-9:00)$ ;
5. 5-й пациент:  $P_2 (8:00-8:15) \rightarrow P_3 (8:20-8:30) \rightarrow P_4 (8:35-8:40)$ .

Время переходов равно 40 минут, что на 5 минут больше оценки снизу.

Третье состояние системы «Пациенты-ПО»:

Оценка снизу равна 50 минут.

1. 1-й пациент:  $P_5 (8:00-8:20) \rightarrow P_4 (8:25-8:30) \rightarrow P_1 (8:40-8:50) \rightarrow P_3 (9:00-9:10)$ ;
2. 2-й пациент:  $P_4 (8:00-8:05) \rightarrow P_1 (8:10-8:20) \rightarrow P_3 (8:30-8:40) \rightarrow P_5 (9:00-9:20)$ ;
3. 3-й пациент:  $P_1 (8:00-8:10) \rightarrow P_5 (8:20-8:40) \rightarrow P_2 (8:45-9:00) \rightarrow P_4 (9:05-9:10)$ ;
4. 4-й пациент:  $P_3 (8:00-8:10) \rightarrow P_2 (8:15-8:30) \rightarrow P_5 (8:40-9:00) \rightarrow P_4 (9:10-9:15)$ ;
5. 5-й пациент:  $P_2 (8:00-8:15) \rightarrow P_3 (8:20-8:30) \rightarrow P_4 (8:35-8:40) \rightarrow P_1 (8:50-9:00)$ .

Время переходов равно 55 минут, что на 5 минут больше оценки снизу.

Четвертое состояние системы «Пациенты-ПО»:

Оценка снизу равна 50 минут.

1. 1-й пациент:  $P_5 (8:00-8:20) \rightarrow P_4 (8:25-8:30) \rightarrow P_1 (8:40-8:50) \rightarrow P_3 (9:00-9:10) \rightarrow P_2 (9:15-9:30)$ ;
2. 2-й пациент:  $P_4 (8:00-8:05) \rightarrow P_1 (8:10-8:20) \rightarrow P_3 (8:30-8:40) \rightarrow P_5 (9:00-9:20) \rightarrow P_2 (9:30-9:45)$ ;
3. 3-й пациент:  $P_1 (8:00-8:10) \rightarrow P_5 (8:20-8:40) \rightarrow P_2 (8:45-9:00) \rightarrow P_4 (9:05-9:10) \rightarrow P_3 (9:20-9:30)$ ;
4. 4-й пациент:  $P_3 (8:00-8:10) \rightarrow P_2 (8:15-8:30) \rightarrow$

- $P_5 (8:40-9:00) \rightarrow P_4 (9:10-9:15) \rightarrow P_1 (9:20-9:30)$ ;
5. 5-й пациент:  $P_2 (8:00-8:15) \rightarrow P_3 (8:20-8:30) \rightarrow P_4 (8:35-8:40) \rightarrow P_1 (8:50-9:00) \rightarrow P_5 (9:20-9:40)$ .

Время переходов равно 50 минут, что соответствует оценке снизу.

Все маршруты найдены. Минимальное суммарное время переходов и ожидания для группы равно 175 минут, что на 15 минут больше суммарной оценки снизу.

### **Сравнение результатов при обслуживании группы пациентов**

Ранее утверждалось, что для обслуживания группы пациентов неправомерно использовать алгоритм, при котором последовательно находят оптимальные маршруты для каждого пациента. Покажем это на примере.

С помощью алгоритма решения задачи выбора маршрута обслуживания конкретного пациента по таблице 1 найдем последовательно 5 маршрутов: определяем оптимальный маршрут для пациента, найденные моменты приема убираем из множества моментов  $t_j$  определяем следующий маршрут и так далее.

В итоге, найдены маршруты:

- 1-й пациент:  $P_2 (8:00-8:15) \rightarrow P_3 (8:20-8:30) \rightarrow P_5 (8:40-9:00) \rightarrow P_4 (9:05-9:10) \rightarrow P_1 (9:20-9:30)$ ;
- 2-й пациент:  $P_5 (8:00-8:20) \rightarrow P_3 (8:30-8:40) \rightarrow P_2 (8:45-9:00) \rightarrow P_4 (9:10-9:15) \rightarrow P_1 (9:30-9:40)$ ;
- 3-й пациент:  $P_2 (8:15-8:30) \rightarrow P_3 (8:40-8:50) \rightarrow P_5 (9:00-9:20) \rightarrow P_4 (9:25-9:30) \rightarrow P_1 (9:40-9:50)$ ;
- 4-й пациент:  $P_5 (8:20-8:40) \rightarrow P_3 (8:50-9:00) \rightarrow P_1 (9:10-9:20) \rightarrow P_4 (9:35-9:40) \rightarrow P_2 (9:45-10:00)$ ;
- 5-й пациент:  $P_2 (8:30-8:45) \rightarrow P_3 (9:00-9:10) \rightarrow P_5 (9:20-9:40) \rightarrow P_4 (9:45-9:50) \rightarrow P_1 (10:00-10:10)$ .

Суммарное время переходов и ожидания равно 185 минут, что на 10 минут больше, чем при решении задачи обслуживания группы с помощью предложенного выше алгоритма.

### **Выводы**

Приведены постановки и алгоритмы решения задач о формировании маршрутов обслуживания пациентов в медицинских информационных системах для конкретного пациента и для группы. Показано, что во многих случаях эти задачи могут быть решены с использованием алгоритмов близких к алгоритмам решения задачи коммивояжера и ее модификаций.



## ЛИТЕРАТУРА



1. *Войтенков С.С., Денисов Е.С.* Применение теории расписаний в грузовых автомобильных перевозках // Развитие теории и практики автомобильных перевозок, транспортной логистики. – 2017. – С. 325–334.
2. *Коган Д.И., Сигал И.Х.* Учет временных характеристик для одного класса задач построения расписаний работы перемещающегося процессора // Автоматика и телемеханика. – 2015. – № 12. – С. 121–134.
3. *Жукова Г.Н., Ульянов М.В., Фомичев М.И.* Эффективный по времени точный комбинированный алгоритм для асимметричной задачи коммивояжера // Бизнес-информатика. – 2018. – № 3(45) – С. 20–28.
4. *Подшивалов С.Ф., Подшивалова К.С., Левицкая Л.В., Дунаев С.В.* Совершенствование алгоритма метода ветвей и границ в задаче коммивояжера для автомобильного транспортного графа // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 1 (21). – С. 169–180.
5. Электронное расписание процедур. Сетевой программный комплекс «Здравница» (<http://resortsoft.ru/dispatcherzaciya.html>).
6. Документ «Назначение услуг». КинтВики ([http://wiki.kint.ru/index.php/Документ\\_«Назначение\\_услуг»\\_\(КУС\)](http://wiki.kint.ru/index.php/Документ_«Назначение_услуг»_(КУС))).
7. Санаториум (<http://sanatorium-is.ru/>).
8. *Zacharias C., Armony M.* Joint Panel Sizing and Appointment Scheduling in Outpatient Care // Management Science. – 2016. – № 11.
9. *Liu N.* Optimal Choice for Appointment Scheduling Window under Patient No-Show Behavior // Production and Operations Management. – 2016. – № 1.

## Новости отрасли



## В ПОИСКЕ НОВЫХ ГОРИЗОНТОВ ПРИМЕНЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В МЕДИЦИНЕ

Секция, посвященная ИИ в медицине, состоялась 21 октября в рамках проходящего в технопарке «Сколково» VIII Международного форума инновационного развития «Открытые инновации». В заседании приняли участие эксперты Ассоциации разработчиков и пользователей систем искусственного интеллекта в медицине «Национальная база медицинских знаний» (НБМЗ).

Доклад директора центра развития комплексных проектов НТИ Андрея Алмазова был посвящен мета-проектной деятельности как закономерному этапу эволюции управления системными проектами. При этом происходит переход от управления ресурсами и временем к управлению культурой и ценностями, от изолированных продуктов проектов – к программам продуктов и консорциумам ответственности. А целью является воспроизводящееся сообщество лидеров рынка, закрепление и удержание позиций.

О том, что такой подход работает эффективно, рассказали руководители компаний-участников «НБМЗ» Александр Гусев и Сергей Сорокин. «Наш стартап увидел огромную пользу в объединении», – отметил эксперт компании «К-МИС» Александр Гусев, – «мы объединяемся с другими участниками и делаем новые сервисы». В то же время, по мнению Александра, поднятия волны ИИ-разработок в медицине России до уровня «цунами» еще далеко: «Для этого, по крайней мере, волна цунами должна стать выше существующих барьеров».

Впрочем, несовершенство законодательства не мешает достигать успехов в отдельных областях, как, например, в радиологии, о чем рассказал руководитель компании «Интеллоджик» Сергей Сорокин. Вдохновившись успехами региональных пилотов (прежде всего на Ямале), компания уже готовит для рынка новый продукт.

«Пора искать новые горизонты применения ИИ в медицине», – резюмировал итоги дискуссии на секции её ведущий Сергей Воинов, директор по акселерации, руководитель направления Digital Health фонда «Сколково».

Источник: ассоциация «Национальная база медицинских знаний»,  
<http://nbmz.ru/2019/10/21/v-poiske-novyh-gorizontov-primenenija-ii-v-medicine/>