**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Кафедра дискретной математики и алгоритмики

**ПИЛЯК КСЕНИЯ ЮРЬЕВНА**

**ОПТИМАЛЬНОЕ НАЗНАЧЕНИЕ ПАЦИЕНТОВ В ОПЕРАЦИОННЫЕ КОМНАТЫ: МАТЕМАТИЧЕСКИЕ МОДЕЛИ И ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ**

Отчёт по производственной практике по специальности:

1-31 81 09 «Алгоритмы и системы обработки больших объемов информации»

Руководитель практики от кафедры

Ковалев Михаил Яковлевич,

член-корреспондент НАН РБ,

доктор физико-математических наук

Руководитель практики от организации

Ковалев Михаил Яковлевич,

член-корреспондент НАН РБ,

доктор физико-математических наук

Минск, 2018

Задача MaxWeight об оптимальном назначении пациентов в операционные комнаты

Формулировка

Задача исследования формулируется следующим образом. Есть *n* пациентов из множества *N*. Каждый из пациентов ожидает операцию в одной из нескольких операционных комнат в один из интервалов времени [*At*, *Bt*], где *At < Bt*, *t* = 1, …, *T*. Время окончания операции *Bt* не фиксировано и может быть увеличено, причем стоимость единицы времени увеличения составляет *ct*. Кроме того, увеличение *Bt* не должно превзойти порог δt, а суммарная стоимость увеличения всех *Bt* ≤ *C*.

Каждый интервал времени связан с операционной комнатой. Интервалы с одинаковыми моментами начала и завершения связаны с разными операционными комнатами, таким образом, набор интервалов {[*At*, *Bt*], *t* = 1, …, *T*} является мультимножеством. Интервалы времени различных операционных комнат также могут перекрываться, в то время как интервалы одной и той же комнаты перекрываться не могут. Для удобства множество интервалов {1, …, *T*} разбито на подмножества *Lv*, |*Lv*| ≥ 1, *v* = 1, …, *V*, *V* ≤ *T*. Каждое подмножество *Lv* содержит индексы интервалов времени с одинаковыми моментами начала и окончания, а различные *Lv* содержат индексы неперекрывающихся интервалов.

Каждому пациенту *j* соответствуют следующие известные числовые параметры: время готовности *rj*, до которого операция не может быть начата; директивный срок *dj*, к которому желательно провести операцию; вес *wj*, обозначающий важность операции или доход от нее; плановая длительность операции *pj*; множество индексов недопустимых интервалов *Mj* ⊂ {1, …, *T*}, в которых операция не может быть проведена из-за отсутствия необходимых ресурсов. При этом *rj* ∈ {*At*| *t* = 1, …, *T*}, *dj* ∈ {*Bt*| *t* = 1, …, *T*} для всех *j* ∈ *N*. Все параметры предполагаются неотрицательными рациональными числами. Предполагается, что госпиталь применяет такую политику планирования операций, что *ri< rj* влечет *di≤ dj* для любых пациентов *i* и *j*. В дальнейшем индекс пациента и индекс соответствующей ему операции не различаются.

Необходимо определить значения увеличения моментов *Bt* окончания операций, выбрать подмножество пациентов и назначить их в интервалы [*At*, *Bt*] так, чтобы были соблюдены сроки готовности и директивные сроки проведения операций, стоимость увеличения не превосходила порога *C*, а суммарный вес пациентов был максимален. Назовем такую задачу MaxWeight [1].

В работе рассматривается два варианта задачи: детерминированная и стохастическая с неопределенными продолжительностями операций.

Классификация задач планирования операционных комнат

В работе [2] приведена классификация задач планирования операционных комнат с помощью нескольких полей. Каждое поле описывает набор характеристик для конкретной задачи с помощью параметров.

***Поле* α*: характеристики пациента***. Пациенты могут быть выборные (*elective*, *el*) и невыборные (*non-elective*, *nel*). Первую категорию представляют пациенты, для которых операции могут быть заранее спланированы. Ко второй категории относятся экстренные пациенты, для которых операции выполняются в срочном порядке.

***Поле* β*: уточнение решения***. Введение второго поля указывает, какие решения должны приниматься в планировании операционных комнат. Это поле состоит из трех параметров:

**β1*: объект решения***. Параметр указывает, к кому конкретно применяется решение. Это могут быть медицинские дисциплины (*medical* *disciplines*, *disc*), хирурги (*surgeons*, *surg*), пациенты или типы пациентов (*patients*, *pat*) и другие объекты (*other*), такие как госпитали.

**β2*: тип решения***. Какое решение должно быть принято. Это может быть решение, связанное с назначением даты (*date*), времени (*time*), операционной комнаты (*room*), продолжительности операции (*capacity*, *cap*) и другие (*other*).

**β3*: степень интеграции***. Есть ли интеграция между операционной комнатой и другими подразделениями госпиталя. Таким образом, в задаче могут рассматриваться как изолированные операционные комнаты (*isolated*, *iso*), так и в объединении с другими подразделениями (*integrated*, *int*). В последнем случае такими подразделениями могут быть реанимация (*postanesthesia* *care unit*, *PACU*), отделение интенсивной терапии (*intensive* *care* *unit*, *ICU*), больничные палаты (*ward*) и другие (*other*).

***Поле* γ*: степень неопределенности***. Задача может быть либо детерминированной (*deterministic*, *det*), либо стохастической (*stochastic*, *stock*). В случае стохастической задачи могут быть неопределенности поступления пациентов (*arrival*, *arr*), продолжительности операций (*duration*, *dur*) и другие виды неопределенностей (*other*), например, количество ресурсов.

***Поле* δ*: показатели эффективности***. Поле состоит из двух параметров:

**δ1*: целевые функции***. Задача может быть однокритериальной (*single*) и многокритериальной (*multi*).

**δ2*: показатели эффективности в задаче***. Различают критерии эффективности, связанные со временем ожидания (*wait*), пропускной способностью (*through*), утилизацией (*util*), сверхурочной работой (*otime*), ограничением по времени (*utime*), финансовыми проблемами (*fin*) и другие (*other*).

Согласно данной классификации рассматриваемая в настоящей работе задача относится к категории *el | pat; time, room; iso | dur | single; through*.

Практическая ситуация и ее сведение к задаче MaxWeight

Рассматривается следующая практическая ситуация. В госпитале есть список пациентов, ожидающих операцию. Этот список обновляется каждое утро. Операции в списке разбиты на *G*0 групп *g* = 1, …, *G*0. Операции группы g имеют одинаковые сроки готовности *r*(*g*) и директивные сроки *d*(*g*), которые согласовываются с пациентом и госпиталем, и связаны с состоянием здоровья пациента, предоперационными процедурами и прочими условиями. При этом предполагается, что сроки согласованы так, что *r*(1)≤ … ≤ и *d*(1)≤ … ≤ . В большинстве случаев для любого заданного *g* значение *r*(*g*) совпадает с началом рабочего дня, а *d*(*g*) – с концом того же дня. Все другие характеристики операций одной и той же группы, кроме вышеназванных сроков, могут быть различными. Время пред- и послеоперационных процедур включается во время операции и не зависит от комнаты.

Госпиталю необходимо провести как можно больше операций. При этом любая операция может быть назначена в подходящую для нее комнату в подходящее время. Рабочий интервал – это рабочая смена или полусмена. Если операция продолжается после окончания смены или полусмены, то это влечет дополнительную стоимость операции (например, стоимость персонала и других ресурсов больницы).

Необходимо выбрать допустимые значения увеличения рабочих интервалов так, чтобы суммарная стоимость такого увеличения не превзошла *C*. Также необходимо из первых *G* групп 1, …, *G*, *G* ≤ *G*0, выбрать операции и составить расписание для них на некоторый календарный период (день, месяц, год). При этом расписание должно удовлетворять всем допустимым срокам операций. Задача состоит в поясняемой ниже *лексикографической максимизации* числа операций, выбранных из первых *G* групп. Число операций, выбранных из группы 1, должно быть глобально максимизировано. Пусть это значение равно . Число операций из группы 2 должно быть максимизировано при условии, что в первой группе выбрано  операций. Пусть это значение равно . Число операций из группы 3 должно быть максимизировано при условии, что в первой группе выбрано  операций, а во второй – операций. Условия повторяются до группы *G*.

Расписание для невыбранных операций составляется путем повторного решения задачи для новых входных данных, учитывающих текущее расписание.

Задача лексикографической максимизации сводится к задаче MaxWeight следующим образом. Пусть *ng* – количество пациентов в группе *g*. Определим веса операций группы g так, что они все одинаковы и равны *w(g)* (*g* = 1, …, *G*):

|  |  |
| --- | --- |
| *w*(*G*) = 1 и *w*(*g*) = *w*(*g*+1)*ng*+1 + *w*(*g*+2)*ng*+2 + … + *w*(*G*)*nG* + 1, *g* = *G* – 1, *G* – 2, …, 1. | (2.1) |

Пусть , …,  – оптимальное число операций, выбранных в группах 1, …, *G* соответственно, для задачи MaxWeight, которая максимизирует функцию *W*(*x*) = *w*(1)*x*1 + … + *w*(*G*)*xG*. Пусть , …,  – число операций в оптимальном решении *Q*\* задачи лексикографической максимизации. Предположим, что  < . Отметим, что решение с вектором количества выбранных операций *x*(1) = (, 0, …, 0), в котором выбраны лишь  операций из первой группы в *Q*\* и значения увеличения рабочих интервалов определены в *Q*\*, допустимо для обеих задач. Также

 ≤ *ng*, *g* = 1, …, *G*.

Тогда

*W*(*x*(1)) – *W*(*x*\*) = *w*(1)( – ) – (*w*(2)  + … + *w*(*G*) ) ≥ *w*(1) –  
– (*w*(2)*n*2 + … + *w*(*G*)*nG*) = 1 > 0.

Таким образом, *W*(*x*(1)) > *W*(*x*\*). Значит, *x*\* не является оптимальным решением задачи MaxWeight, что является противоречием. Следовательно, ≥  но и по определению  ≤ , откуда следует =. Рассуждая аналогично для (, ), (, ), …, приходим к выводу, что *x*\* = *y*\*. Это значит, что множества оптимальных решений задачи лексикографической максимизации и задачи MaxWeight с весами из формулы (2.1) совпадают и можно воспользоваться алгоритмом оптимального решения второй задачи для оптимального решения первой.

Если ресурсы больницы заранее не распределены, то любая операция может быть назначена в любой рабочий интервал в любую операционную комнату, а затем осуществлено распределение ресурсов. Если происходит сбой в графике операций (недоступность ресурсов, внеплановые операции и др.), вводятся новые данные и задача решается для новых данных таким образом, что уже начатые операции должны быть завершены. При этом операции первой необходимости должны быть проведены как можно скорее. Для этого можно поставить их в группу 1, сделав сроки готовности нулевыми. Чтобы увеличить шансы на то, что в случае поступления экстренного пациента хотя бы одна подходящая операционная комната была свободна, можно добавить ко времени каждой операции некоторое время задержки (простоя), в котором можно начать незапланированную экстренную операцию в случае необходимости [1].

Моделирование детерминированной задачи оптимального назначения пациентов

Задача Det

Пусть *zt*– переменные, обозначающие увеличение *Bt*, *t* = 1, …, *T*. Положим *xjt* = 1, если операция пациента *j* назначена в интервал [*At*, *Bt*], *t* = 1, …, *T*, и 0 в противном случае, при этом , *j* = 1, …, *n*. Обозначим *z* = (*z*1, …, *zT*), и *x* как матрицу размером *n* × *T* с элементами *xjt*.

Кроме того, введем для каждой операции *j* множество *Ij* индексов интервалов, в которые операция не может быть проведена из-за того, что этот интервал является недопустимым, либо из-за того, что этот интервал не совпадает с допустимыми сроками проведения операции:

|  |  |
| --- | --- |
| *Ij* = *Mj* ∪ {*t* | *rj* + *pj* > *Bt*, *t* = 1, …, *T*} ∪ {*t* | *dj* – *pj* < *At*, *t* = 1, …, *T*}, *j* = 1, …, *n*. | (3.1) |

Введем порядок операций, называемый Earliest Ready Time – Earliest Due Date (ERT–EDD), в котором операции перенумерованы так, что *r*1 ≤ … ≤ *rn* и *d*1 ≤ … ≤ *dn*. Это значит, что сроки операции должны быть согласованы таким образом, что *ri* < *rj* ⇒ *di* ≤ *dj*.

Утверждается, что существует оптимальное решение детерминированной задачи, в которой операции назначены на один и тот же интервал и выполняются в ERD–EDD порядке. Рассмотрим классическую задачу расписаний: минимизацию взвешенного количества запаздывающих работ для одного прибора с согласованными моментами готовности и директивными сроками, т.е. 1 | *ri* < *rj* ⇒ *di* ≤ *dj* | . В работе [3] предложен алгоритм, который находит оптимальное решение такой задачи за время *O*(*n*2). Другой вариант решения с временной сложностью *O*(*n* log *n*) приведен в работе [4]. Данная задача эквивалента задаче MaxWeight в случае одного интервала времени, следовательно, оптимальное решение для последней существует.

Кроме увеличения моментов окончания интервалов *Bt* введем увеличение директивных сроков операций *dj*. Поскольку *dj* ∈ {*Bt*| *t* = 1, …, *T*} для всех *j* ∈ *N*, то в случае если *dj* ≤ *Bt* интервал [*At*, *Bt*] нельзя будет увеличить. Увеличение директивного срока фактически означает увеличение интервала времени с моментом окончания, равному этому директивном сроку. Заметим, что для ERT–EDD последовательности операций *i1,…,ih*, назначенных в интервал [*At*, *Bt*] одной и той же операционной комнаты*,* сроки готовности, конечные сроки и границы рабочих интервалов удовлетворены тогда и только тогда, когда max{*At*, } + ≤ min{*Bt*, } + *z*τ, 1 ≤ *k* ≤ *l* ≤ *h*. Здесь *z*τ – это увеличение интервала [*At*, *Bt*], если *Bt* < , в противном случае *z*τ является увеличением директивного срока. Теперь можно сформулировать задачу смешанного целочисленного линейного программирования, соответствующую исходной задаче, которую будем также обозначать Det:

max ;

max{*At*, *rk*} + ≤ min{*Bt*, *dl*} + *z*τ,

где *t* ∉ *Ik*, 1 ≤ *k* ≤ *l* ≤ *n*, *t* = 1, …, *T*,

τ = ;

*zt* ≤ δ*t*;

≤ *С*;

≤ 1, где *j* = 1, …, *n*;

*xjt* = 0, где *t* ∈ *Ij*, *j* = 1, …, *n*, *t* = 1, …, *T*;

*zt* ≥ 0; *xjt* ∈ {0, 1}, где *j* = 1, …, *n*, *t* = 1, ..., *T*.

Задача Det-Alt

Введем дополнительные переменные *yj*, которые принимают значение 1, если операция *j* назначена на какой-либо интервал времени, и 0 в противном случае. Приведем альтернативную формулировку задачи, называемую Det-Alt:

max ;

|  |  |
| --- | --- |
| max{*At*, *rk*} + ≤ min{*Bt*, *dl*} + *z*τ,  где *t* ∉ *Ik*, 1 ≤ *k* ≤ *l* ≤ *n*, *t* = 1, …, *T*,  τ = ; | (3.2) |
| *zt* ≤ δ*t*; | (3.3) |
| ≤ *С*; | (3.4) |
| = 0, где *j* = 1, …, *n*; | (3.5) |
| , где *t* ∈ *Lv*, *t* + 1 ∈ *Lv*, | *Lv* | ≥ 2, *v* = 1, …, *V*; | (3.6) |
| *xjt* = 0, где *t* ∈ *Ij*, *j* = 1, …, *n*, *t* = 1, …, *T*; | (3.7) |
| *zt* ≥ 0; *xjt* ∈ {0, 1}, где *j* = 1, …, *n*, *t* = 1, ..., *T*. | (3.8) |

Формула (3.6) означает, что при существовании комнат, которым соответствуют интервалы времени с одинаковым началом и концом, предпочтение при назначении пациентов отдается комнатам с меньшими номерами.

Использование более ранних интервалов времени

Заметим, что может существовать несколько оптимальных решений для максимизации суммарного веса  в задаче Det. Наиболее подходящим является решение, в котором пациенты назначаются на более ранние интервалы времени, чтобы операционные комнаты стали как можно раньше доступными для последующих пациентов. Для этого нужно рассмотреть второй критерий. В качестве такого критерия можно взять минимизацию функции . Значение функции есть сумма моментов начала интервалов времени, на которые назначены операции. Таким образом, целевая функция примет вид:

max *M* –  = max .

Здесь множитель *M* – достаточно большое число. Он взят для того, чтобы никакое уменьшение  не компенсировало уменьшение . Наиболее оптимально взять *M* = *nA*max, где *A*max = . Назовем задачу с новой целевой функцией Ext-Det, где Ext обозначает *extended*.

Аналогично видоизменяется целевая функция задачи Det-Alt:

max *M* – .

Задачу с такой целевой функцией назовем Ext-Det-Alt [1].

Эвристические и рандомизированные алгоритмы

В случае, когда нахождение оптимального решения точными методами смешанного целочисленного линейного программирования затруднено, например, при больших объемах входных данных время выполнения алгоритма неоправданно велико, имеет место использование эвристических и рандомизированных алгоритмов.

Пусть *Nt* – множество операций, подходящих для проведения в интервал времени *t*: *Nt* = {*j* ∈ *N* | *t* ∉ *Mj*, [*At*, *Bt*] ∩ [*rj*, *dj*] = ∅} для всех *t* = 1, …, *T*.

Один из эвристических алгоритмов нахождения решения задачи Ext-Det приведен ниже.

1. Отсортировать интервалы времени в порядке неубывания их начала.
2. Рассмотреть текущий интервал [*At*, *Bt*]. Составить список кандидатов *CLt* операций из *Nt*, которые могут быть назначены в данный интервал при условии, что в нем еще есть свободное место для их проведения. Выбрать операцию *j* из *CLt* с наибольшей плановой длительностью, назначить ее на интервал и повторить пункт 2. Если *CLt* = ∅ и есть, по крайней мере, одна не назначенная операция, рассмотреть другой интервал времени и повторить пункт 2. В противном случае перейти к пункту 3.
3. Если все операции назначены, перейти к пункту 6, иначе перейти к пункту 4.
4. Отсортировать не назначенные операции в порядке невозрастания их весов.
5. Рассмотреть текущую операцию *j*. Найти интервал времени *t* ∈ *T* \ *Mj*, для которого стоимость увеличения при назначении операции *j* будет наименьшей. Назначить операцию *j* на интервал времени *t*, если суммарная стоимость увеличения не превосходит *C*, выбрать другую операцию и повторить пункт 5. В противном случае перейти к пункту 6.
6. Возможное решение найдено.

Еще один вариант эвристического алгоритма отличается от вышеприведенного тем, что вместо отдельных интервалов времени в пункте 2 следует рассматривать подмножества *Lv*, *v* = 1, …, *V*. В пределах одного подмножества *Lv* интервалы рассматриваются по порядку. Как только на рассматриваемый интервал будет назначена операция, переходим к другому интервалу. Если все интервалы рассмотрены, но еще остались не назначенные операции, вновь рассматриваются эти же интервалы с учетом ранее назначенных операций. Так необходимо повторять до тех пор, пока все операции не будут назначены или не найдется ни одного интервала из *Lv*, куда их можно назначить.

Рандомизированный алгоритм отличается от эвристического тем, что операция j ∈ CLt выбирается случайным образом с вероятностью . Алгоритм можно повторить несколько раз для увеличения вероятности получения большего значения целевой функции.

Литература

1. Статья ковалев
2. Статья кардоен классификация
3. Kise ibaraki mine
4. Lawer 1982B

Док-во kise ibaraki mine