УДК 681.518.3

ОБЩАЯ ЗАДАЧА УПРАВЛЕНИЯ ДВУХСТАДИЙНЫМ ОБСЛУЖИВАНИЕМ ПОТОКА ОБЪЕКТОВ В ДИСКРЕТНОЙ МОДЕЛИ ПРОИЗВОДСТВЕННО-ТРАНСПОРТНОГО КОМПЛЕКСА

Д.И. Коган
МИРЭА − Российский технологический университет
Россия, 119571, Москва, Проспект Вернадского, д. 86
E-mail: kdi_41@mail.ru

А.С. Митрошина
Волжский государственный университет водного транспорта
Россия, 603950, Нижний Новгород, ул. Нестерова 5
E-mail: fds@vgavt-nn.ru

К.С. Ульянов
МИРЭА − Российский технологический университет
Россия, 119571, Москва, Проспект Вернадского, д. 86
E-mail: kirik516@mail.ru

Ю.С. Федосенко
Волжский государственный университет водного транспорта
Россия, 603950, Нижний Новгород, ул. Нестерова 5
E-mail: fds@vgavt-nn.ru

Ключевые слова: производственно-транспортная логистика, теория расписаний, динамическое программирование, вычислительная сложность.

Аннотация: В дискретной идеализации формулируется математическая модель двухстадийного обслуживания объектов конечного детерминированного потока. На первой стадии каждый объект потока подлежит обслуживанию стационарным процессором-источником, по завершению которого он адресуется для реализации второй стадии обслуживания к тому или иному доступному стационарному процессору-потребителю из конечной пространственно рассредоточенной совокупности. С каждым процессором-потребителем ассоциируется функция штрафа, монотонно возрастающая от момента завершения обслуживания направленного к нему объекта. Модель описывает, в частности, состояние призводственно-транспортного комплекса на момент принятия решений при планировании использования группы судов для перевозки в заданные пункты нерудных строительных материалов, добываемых на русловом месторождении гидромеханизированным способом. В рамках построенной модели ставится оптимизационная задача синтеза стратегии обслуживания и конструируется решающий алгоритм динамического программирования.

1. Введение

Исследуется возникающая в различных приложениях специфическая проблема управлением использованием дискретных ресурсов – оптимизации распределения между исполнителями формируемых пар невзаимнозаменяемых работ. В качестве примера такого приложения укажем прозводственно-транспортный комплекс Камского грузового района [1], в котором выделенная группа неоднотипных грузовых судов (многосекционных судовых составов) [2] используется для перевозки в заданные пункты нерудных строительных материалов (НСМ) [3], загружаемых гидромеханизированным способом землесосным снарядом (ЗС) [4] непосредственно на русловом месторождении.

К моменту завершения сеанса формирования очередного оперативного плана функционирования прозводственно-транспортного комплекса рассматриваемого типа диспетчерской службой должно быть однозначно определено: а) в каком порядке следует подавать под погрузку к ЗС суда выделенной группы; б) какой пункт назначения для выгрузки следует назначить каждому конкретному судну после его загрузки НСМ.

Для формирования оперативных планов функционирования производственно-транспортной системы рассматриваемого типа, эффективных в условиях складывающейся эксплуатационной обстановки, актуальной является разработка и штатное использование специализированной цифровой системы поддержки управления, включающей в себя как модуль математического моделирования технологического процесса, так и алгоритм решения соответствующим образом поставленной экстремальной задачи синтеза расписания подачи к ЗС судов под погрузку и последующего их распределения по пунктам выгрузки НСМ.

2. Модель обслуживания потока объектов

Рассматривается n-элементный поток O_n независимых транспортных средств −объектов (обозначаемых как 1, 2, ... , n), подлежащих однократному одностадийному обслуживанию (загрузкой) стационарным процессором-источником $\boldsymbol{H}$. Считается, что в каждый момент времени процессор не может обслуживать более чем один объект и его непроизводительные простои запрещены. Обслуживание каждого объекта реализуется процессором $\boldsymbol{H}$ без прерываний, необходимости в его переналадках нет. Объекты, прошедшие обслуживание, далее направляются к существенно удаленным от процессора $\boldsymbol{H}$ процессорам-потребителям (далее потребителям), составляющим совокупность $\boldsymbol{P} = \{p_1, p_2, \dots, p_n\}$. В адрес каждого потребителя должен быть направлен ровно один объект, каждому объекту должен быть назначен ровно один потребитель. Считается известной $(n{\times}n)$-матрица $\boldsymbol{E} = \{e_{ij}\}$, где $e_{ij} = 1$, если объект i технически допустим для доставки груза потребителю p_j; в противном случае $e_{ij} = 0$. Предполагается, что матрица $\boldsymbol{E}$ такова, что определяемая ею простейшая задача о назначениях имеет полные решения [5], т.е. каждому потребителю можно взаимно однозначно предписать транспортное средство, способное его обслужить.

Для каждого объекта i считаются заданными: τ_i – норма длительности обслуживания процессором $\boldsymbol{H}$, t_i – момент поступления в очередь на обслуживание; считается, что $0 = t_1 \le t_i \le \dots \le t_n$. Необслуженный объект не может покинуть очередь. Полагается, что процессор $\boldsymbol{H}$ готов к обслуживанию объектов потока O_n начиная от момента времени $t = 0$.

Для каждого потребителю p_j известна функция индивидуального штрафа $\psi_j(t)$; все функции $\psi_j(t)$, $j = \overline{1,n}$ являются монотонно возрастающими (вообще говоря, в нестрогом смысле). Если обслуживание объекта процессором-потребителем $\boldsymbol{U}_j$ (разгрузка прибывшего потребителю p_j транспортного средства) завершается в момент времени t, то $\psi_j(t)$ – величина индивидуального штрафа (потерь) по данному потребителю, $j = \overline{1,n}$. Полагаются известными ($n \times n$)-матрицы $\boldsymbol{V} = \{v(i, j)\}$ и $\boldsymbol{W} = \{w(i, j)\}$, где $v(i, j)$ – продолжительность перехода объекта i от места нахождения процессора $\boldsymbol{H}$ до места расположения потребителя p_j, а $w(i, j)$ – норма длительности обслуживания процессором $\boldsymbol{U}_j$ объекта i, прибывшего к потребителю p_j; в случае $e_{ij} = 0$ считается, что $v(i, j) = w(i, j) = +\infty$). Обслуживание поступившего к потребителю p_j объекта процессором $\boldsymbol{U}_j$, $j = \overline{1,n}$ начинается непосредственно от момента подхода и осуществляется без прерываний.

Время считается дискретным, измеряется в тактах; все относящиеся к сконструированной модели числовые характеристики суть целые числа.

Стратегию обслуживания объектов потока O_n определим как пару $S = [\rho; r(j), j = \overline{1,n}\,]$, где $\rho = (i_1, i_2, \dots, i_n)$ – перестановка элементов множества $\{1, 2, \dots, n\}$, а $r(j)$ – взаимно однозначное отображение множества $\{1, 2, \dots, n\}$ в себя. При реализации стратегии S объект i_k (см. первую компоненту стратегии) процессором $\boldsymbol{H}$ обслуживается k-м по очереди, $k = \overline{1,n}$; $r(j)$ – индекс потребителя, которому объект j предназначается, $j = \overline{1,n}$. Обязательным является выполнение условия $e_{j,r(j)} = 1$, $j = \overline{1,n}$, означающее, что к каждому потребителю направляется транспортное средство, способное его обслужить.

Реализации стратегий считаем компактными [6]. В таком случае при любой фиксированной стратегии S для каждого транспортного средства по перестановке ρ арифметически вычисляется момент завершения его обслуживания процессором $\boldsymbol{H}$. Далее по отображению $r(k)$ для каждого потребителя p_j легко определяется момент завершения обслуживания $t^*(S, j)$, $j = \overline{1,n}$.

Таким образом, с каждой стратегией S ассоциируется величина $\sum_{j=1}^{n} \psi_j(S, j)$ – суммарные потери (суммарный штраф) потребителей и соответствующая оптимизационная записывается в виде

$$\min_S \sum_{j=1}^{n} \psi_j(S, j). \tag{1}$$

Для решения задачи (1) методом динамического программирования [7] будем считать, что при обслуживании объектов потока процессором $\boldsymbol{H}$ управленческие решения принимаются для тех моментов времени, когда процессор свободен; каждое такое решение состоит в определении номера очередного подаваемого к процессору объекта для обслуживания и в адрес какого потребителя будет он направлен по завершению обслуживания.

Текущая ситуация вполне характеризуется тройкой $(t, \boldsymbol{M}, \boldsymbol{N})$, где t – момент принятия решения, в этот момент процессор свободен; $\boldsymbol{M}$ – множество объектов, которые на момент времени t остаются необслуженными; $\boldsymbol{N}$ – множество индексов потребителей, которым еще не направлены предназначенные для них объекты. Определенные выше тройки называем состояниями системы и введем в рассмотрение функцию Беллмана $B(t, \boldsymbol{M}, \boldsymbol{N})$, величина которой при заданных значениях аргументов равна минимально возможному суммарному штрафу в реализациях стратегий, переводящих систему из состояния (t, $\boldsymbol{M}$, $\boldsymbol{N}$) в финальное, т.е. в состояние, в котором множества $\boldsymbol{M}$ и $\boldsymbol{N}$ оказываются пустыми. Очевидно, что

(2) $$B(t, \{i, j\}) = \psi_j(t) \cdot (\max(t, t(i)) + \tau(i) + v(i, j) + w(i, j)).$$

Для неодноэлементных множеств $\boldsymbol{M}$ и $\boldsymbol{N}$ (число элементов в $\boldsymbol{M}$ совпадает с числом элементов в $\boldsymbol{N}$) имеем

(3) $$B(t, \boldsymbol{M}, \boldsymbol{N}) = \min_{(i,\, j) \in MxN} [\, \psi_j(t) \cdot (\max(t, t(i)) + \tau(i) + v(i, j) + w(i, j)) + \\ + B(\max(t, t(i)) ++ \tau(i), + \boldsymbol{M}/\{i\}, \boldsymbol{N}/\{j\})].$$

Формулы (2), (3) – рекуррентные соотношения динамического программирования для решения задачи (1).

При вычислениях вначале по соотношению (3) находим значения функции Беллмана для всех одноэлементных множеств $\boldsymbol{M}$ и $\boldsymbol{N}$ при достаточно больших значениях параметра t; далее вычисления выполняются по формуле (2) в порядке увеличения числа элементов в множествах $\boldsymbol{M}$ и $\boldsymbol{N}$ до тех пор, пока не будет определена величина $B(0, \{1, 2, \ldots, n\}, \{1, 2, \ldots, n\})$ – оптимальное значение критерия в задаче (1).

Вычислительная сложность сконструированного алгоритма определяется числом наборов аргументов, для которых определяются значения функции Беллмана. Как очевидно, пара ($\boldsymbol{M}, \boldsymbol{N}$) принимает $2^n \times 2^n = 4^n$ различных наборов значений, что и определяет вычислительную сложность описанной процедуры.

NP-трудность [8] задачи (1) является следствием *NP*-трудности рассмотренной в [6] задачи диспетчеризации.

Замечание 1. В предположении линейности всех функций индивидуального штрафа отыскание оптимального назначения (первый этап решения задачи) требует не более чем кубично зависящего от n числа элементарных операций, а число элементарных операций, выполняемых при решении канонической задачи диспетчеризации имеет порядок 2^n. Таким образом, переход от общей задачи (1) к ее линейной конкретизации существенно уменьшает вычислительную сложность проблемы, хотя она по-прежнему остается труднорешаемой.

Замечание 2. Если по состоянию на начальный момент времени все объекты потока O_n готовы к обслуживанию процессором $\boldsymbol{H}$, т.е. $t(i) = 0$, $i = \overline{1, n}$, то в случае линейности всех функций индивидуального штрафа задача синтеза оптимального расписания в результате несложных преобразований сводится к известной задаче мастера, алгоритм решения которой тривиален [9].

Для общей задачи и указанных в замечаниях 1 и 2 её частных модификаций в докладе приводятся результаты массовых вычислительных экспериментов по оценке длительности синтеза стратегий обслуживания, полученных сконструированным в работе алгоритмом динамического программирования, а также описываются приближенные стратегии обслуживания, построенные путем реализации известных метаэвристических подходов.

Список литературы

1. Концепция развития внутреннего водного транспорта в Республике Татарстан. http://pandia.ru/text/78/119/109171-2.php
2. Справочник по серийным транспортным судам. Т. 4. М.: Транспорт, 1975. 179 с.
3. Козырев В.К. Грузоведение. М.: Транспорт, 1991. 288 с.
4. Бессонов Е.А. Энциклопедия гидромеханизированных работ. М.: 1989, 2005. 520 с.
5. Алексеев О.Г. Комплексное применение методов дискретной оптимизации. М.: Наука, 1987. 247 с.

6. Коган Д.И. Задача диспетчеризации: анализ вычислительной сложности и полиномиально разрешимые подклассы // Дискретная математика. 1996. Т. 8. №3. С. 135-147.
7. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М.: Наука, 1965. 457 с.
8. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
9. Танаев В.С., Гордон В.С., Шафранский Я.М. Теория расписаний. Одностадийные системы. М.: Наука, 1984. 382 с.