[3] Zitzler E. (1999). Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph. D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland. TIK-Schriftenreihe Nr. 30, Diss ETH No. 13398, Shaker Verlag, Aachen, Germany.
[4] Куимова А.С., Федосенко Ю.С. Модификация концепции d-расписаний для бикритериальной задачи обслуживания бинарного потока объектов в стационарной однопроцессорной системе с накопительным элементом (в настоящем сборнике).

А.С. Куимова, Ю.С. Федосенко
ФБОУ ВПО «ВГАВТ»

МОДИФИКАЦИЯ КОНЦЕПЦИИ D-РАСПИСАНИЙ ДЛЯ БИКРИТЕРИАЛЬНОЙ ЗАДАЧИ ОБСЛУЖИВАНИЯ БИНАРНОГО ПОТОКА ОБЪЕКТОВ В СТАЦИОНАРНОЙ ОДНОПРОЦЕССОРНОЙ СИСТЕМЕ С НАКОПИТЕЛЬНЫМ ЭЛЕМЕНТОМ

Вводится модель обслуживания детерминированного потока объектов на стационарном терминале с накопительным элементом. Формулируется бикритериальная задача синтеза стратегий обслуживания. Показано, что учет ограничений на структуру стратегий обслуживания выводит задачу из класса NP-трудных и позволяет построить полиномиальный алгоритм синтеза стратегий обслуживания.

Введение. Рассматривается математическая модель обслуживания конечного детерминированного потока объектов стационарным процессором с накопительным компонентом при наличии двух независимых критериев оценки качества управления [1]. На содержательном уровне модель, в частности, описывает технологию северного завоза дизельного топлива потребителям, дислоцированным вблизи малых рек приполярного региона полуострова Ямал, через речной порт г. Салехард. В рамках модели ставится оптимизационная задача и описывается алгоритм синтеза стратегий обслуживания, реализующий в рамках концепции Парето идеологию динамического программирования в его бикритериальном расширении [2].

В силу оценок вычислительной сложности [3], задача синтеза стратегий обслуживания для рассматриваемой модели является *NP*-трудной. Поэтому актуальной является проблема построения модификаций бикритериальной модели, которые, сохраняя адекватность описания прикладной специфике, порождают полиноминально разрешимые подклассы. В статье рассматривается модификация модели обслуживания и алгоритм решения соответствующей оптимизационной задачи, которое основано на введенном в работе [3] понятии d-расписаний, адаптированном для рассматриваемой модели обслуживания бинарного потока объектов на стационарном терминале с накопительным компонентом.

Математическая модель и постановка задачи. Пусть $O_n = \{o_1, o_2, \ldots, o_n\}$ − поток объектов, подлежащих однофазному обслуживанию стационарным процессором с накопительным компонентом. Для каждого объекта o_i определены целочисленные параметры: t_i – момент поступления в очередь на обслуживание ($0 \le t_1 \le \ldots \le t_n$), τ_i– норма длительности обслуживания, a_i– штраф за единицу времени простоя в ожидании обслуживания, η_i– мягкий директивный срок завершения обслуживания ($\eta_i \ge t_i + \tau_i$), v_i – объемная характеристика. Поток O_n обладает свойством бинарности и состоит из двух подпотоков: входящего O^+ и исходящего O^- таких, что $O^+ \cup O^- = O_n$, $O^+ \cap O^- = \varnothing$. Принадлежность объекта o_i ($i = \overline{1,n}$) тому или иному подпотоку определяется значением параметра w_i ($w_i = +1$, если $o_i \in O^+$, и $w_i = -1$, если $o_i \in O^-$). Накопительный компонент представляет собой резервуар объемом V^*, с заполнением равным V_0 в начальный момент времени $t = 0$. В результате обслуживания объекта o_i из подпотока O^+ (O^-) заполнение резервуара увеличивается (уменьшается) на величину v_i, отсюда следуют следующие объёмные ограничения на очередность обслуживания объектов потока. Обслуживание любого объекта $o_i \in O^+$ считается возможным, если в результате его реализации заполнение резервуара $V \leq V^*$. Аналогично обслуживание любого объекта $o_i \in O^-$ считается возможным, если имеющееся к его началу заполнение резервуара $V \geq v_i$.

Стратегия обслуживания объектов S представляет собой произвольную перестановку $S = \{i_1, i_2, \ldots, i_n\}$ совокупности индексов $N = \{1, 2, \ldots, n\}$; при её реализации объект с индексом i_k обслуживается k-м по очереди ($k = \overline{1,n}$). Стратегию S именуем допустимой, если удовлетворяются отмеченные выше объёмные ограничения на очередность обслуживания объектов o_i ($i = \overline{1,n}$), множество всех допустимых стратегий обозначим Ω. Необходимым и достаточным условием непустоты множества допустимых стратегий Ω является выполнение неравенств $0 \leq V_0 + \sum_{i:o_i \in O^+} v_i - \sum_{i:o_i \in O^-} v_i \leq V^*$ с учетом всегда выполняемого на практике ограничения $2 \cdot v_i \leq V^*$ $(i = \overline{1,n})$.

Качество той или иной стратегии S будем оценивать по значениям двух минимизируемых критериев $K_1(S)$ и $K_2(S)$. Первый критерий представляет собой суммарный штраф за простои объектов в ожидании обслуживания, а второй оценивает максимальное по продолжительности нарушение директивного срока завершения обслуживания среди всех объектов потока O_n. Формулы для вычисления значений критериев: $K_1(S) = \sum_{k=1}^{n} a_{i_k} (t^*(i_k, S) - t_{i_k})$, $K_2(S) = \max_{1 \leq k \leq n}(\bar{t}(i_k, S) - \eta_{i_k}, 0)$, где $t^*(i_k, S)$ и $\bar{t}(i_k, S)$ моменты начала и завершения обслуживания объекта с индексом i_k.

Требуется в плоскости ($K_1(S)$, $K_2(S)$), где $S \in \Omega$, выделить полную совокупность эффективных по Парето точек с возможностью после-дующего синтеза стратегии обслуживания для выбранной ЛПР точке:

$$\{\min_{S \in \Omega}(K_1(S)), \min_{S \in \Omega}(K_2(S))\}. \qquad (1)$$

Модификация математической модели и алгоритм решения задачи. С целью сокращения времени решения оптимизационной задачи предлагается модификация модели обслуживания, которая включает в себя следующее ограничение, задаваемое лицом принимающим решения (ЛПР).

Считаем, что при реализации стратегии $S \in \Omega$ объект o_β опережает в обслуживании объект o_α, если $\alpha < \beta$, но объект с индексом β обслуживается раньше, чем объект с индексом α; разность $\mathrm{d} = \beta - \alpha$ назовем величиной опережения. Полагаем, что величина опережения для объектов потока O_n не может превышать задаваемого ЛПР значения параметра d. Стратегию, удовлетворяющую данному ограничению, будем именовать d-стратегией, множество таких стратегий будем обозначать как $S^0(2\mathrm{d})$ ($S^0(2\mathrm{d}) \subseteq \Omega$).

Введение такого ограничения на поток объектов O_n в рассматриваемой модели приводит к случаям, когда множество допустимых стратегий обслуживания $S^0(2\mathrm{d})$ пусто и задача не имеет решений. Это связано с тем, что объёмные ограничения на очередность обслуживания объектов потока пересекаются с ограничением на величину опережения. Таким образом, в ходе решения задачи возникают такие ситуации, когда текущее заполнение резервуара не позволяет обслужить текущий объект из входящего или исходящего подпотока и, при этом, ограничение на величину опережения не позволяет текущему объекту пропустить вперед себя следующий объект.

Поэтому была рассмотрена следующая модификация концепции d-расписаний [3], которая позволяет исключить случаи, когда задача не имеет решения. Для этого понятие величины опережения вводится отдельно для каждого подподока: величина опережения для объекта подпотока O^+ не может превышать значения параметра d^+, а для объекта подпотока O^- – значения параметра d^-. Стратегию, удовлетворяющую данному ограничению, назовём 2d-стратегией; множество таких стратегий будем обозначать как $S^0(2\text{d})$ ($S^0(2\text{d}) \subseteq \Omega$).

С учетом введенного ограничения на величину обслуживания алгоритм решения задачи (1) в классе 2d-стратегий, также может быть описан в виде рекуррентных соотношений динамического программирования. Для этого воспользуемся введенными в [1] обозначениями: $\otimes$ – двуместная операция между произвольным двумерным вектором и множеством векторов, $eff[M]$ – максимальное по включению подмножество недоминируемых в множестве M двумерных векторов; $D^+(t, \Delta)$ ($D^-(t, \Delta)$) – совокупность индексов объектов подпотока O^+ (O^-), поступающих в систему обслуживания на интервале времени $[t, t + \Delta]$, $\Delta \geq 1$; $A_t(B_t)$ – множество индексов объектов подпотока O^+ (O^-) ожидающих обслуживания в момент времени t; $E_{2\text{d}}(t, j, A_t, l, B_t)$ – совокупность эффективных по Парето двумерных оценок, сгенерированная за период времени от момента t и до окончания процесса обслуживания всех объектов потока O_n; $E_{2\text{d}}(0, \min_{j\in A_0} j, A_0, \min_{l\in B_0} l, B_0)$ – полная совокупность эффективных оценок задачи (1).

Величина заполнения емкости накопительного компонента процессора $V(t, j, A_t, l, B_t)$ в каждом состоянии системы обслуживания однозначно определяется тройкой значений t, A_t, B_t. В текущем состоянии (t, j, A_t, l, B_t) объект с индексом $\alpha \in A_t$ допустим к обслуживанию при условии выполнения неравенств $V(t, j, A_t, l, B_t) + v_\alpha \leq V^*$, $\alpha - j \leq \text{d}^+$. Объект с индексом $\beta \in B_t$ может быть обслужен, если $V(t, j, A_t, l, B_t) \geq v_\beta$, $\beta - l \leq \text{d}^-$.

Введем также обозначения: A_t^{j*} ($A_t^{j*} \subseteq A_t$) (B_t^{j*} ($B_t^{j*} \subseteq B_t$)) для множества индексов объектов подпотока O^+ (O^-), допустимых к обслуживанию в состоянии (t, j, A_t, l, B_t); Q_t – совокупность индексов всех необслуженных к моменту времени t объектов;$\xi[Q_t]$ – минимальный из положительных индексов объектов множества Q_t. С учетом введенных выше обозначений решающие задачу (1) рекуррентные соотношения записываются в виде:

$$E_{2\text{d}}(t_n + \theta, \alpha, \{\alpha\}, \varnothing, \varnothing) = \{(a_\alpha(t_n + \theta - t_\alpha), \max(\bar{t}(\alpha, S) - \eta_\alpha, 0))\},$$

$$E_{2\text{d}}(t_n + \theta, \varnothing, \varnothing, \beta, \{\beta\}) = \{(a_\beta(t_n + \theta - t_\beta), \max(\bar{t}(\beta, S) - \eta_\beta, 0))\},$$

$$E_{2\text{d}}(t, j, A_t, l, B_t) = eff(\underset{\alpha \in A_t^{j*}}{eff}((a_\alpha(t - t_\alpha), \max(\bar{t}(\alpha, S) - \eta_\alpha, 0)) \otimes$$

$$\otimes E_{2\text{d}}(t + \tau_\alpha, \xi^+[Q_t \setminus \{\alpha\}], (A_t \setminus \{\alpha\}) \bigcup D^+(t, \tau_\alpha), \xi^-[Q_t \setminus \{\alpha\}],$$

$$B_t \bigcup D^-(t, \tau_\alpha))) \bigcup \underset{\beta \in B_t^{j*}}{eff}((a_\beta(t - t_\beta), \max(\bar{t}(\beta, S) - \eta_\beta, 0)) \otimes E_{2\text{d}}(t + \tau_\beta,$$

$$\xi^+[Q_t \setminus \{\beta\}], A_t \bigcup D^+(t, \tau_\beta), \xi^-[Q_t \setminus \{\beta\}], (B_t \setminus \{\beta\}) \bigcup D^-(t, \tau_\beta)))).$$

Вычислительная сложность предлагаемого алгоритма оценивается как $O(n^3 2^d)$. Ниже в таблице приведены результаты экспериментов, иллюстрирующие, в частности, что для обычно используемого на практике ограничения $d \leq 4$ разработанный алгоритм позволяет сократить время синтеза стратегий обслуживания на два порядка.

Таблица 1

Среднее время решения частных задач, с

d / n	1	2	3	4	5	6	7	d = n – 1
15	0,000	0,004	0,062	0,627	4,139	21,359	81,015	291,582

Список литературы:

[1] Куимова А.С., Минаев Д.В., Федосенко Ю.С. Управление однопроцессорным обслуживанием бинарного потока объектов в системе с накопительным компонентом // Информационно-измерительные и управляющие системы, 2011. Т. 9. № 3. С. 33–37.
[2] Коган Д.И. Динамическое программирование и дискретная многокритериальная оптимизация. Н. Новгород: Изд-во ННГУ, 2005. 260 с.
[3] Коган Д.И., Федосенко Ю.С. Задача диспетчеризации: анализ вычислительной сложности и полиномиально разрешимые подклассы // Дискретная математика. 1996. Т. 8. Вып. 3. С. 135–147.

Д.Е. Копасов
ФБОУ ВПО «ВГАВТ»

ИСПОЛЬЗОВАНИЕ ПЛАТФОРМЫ 1С 8.2 ДЛЯ УЧЕТА КОНТИНГЕНТА СТУДЕНТОВ И РАСЧЕТОВ С НИМИ

В данной статье рассматривается использование платформы 1С: Предприятие 8.2 и конфигурации «Деканат». Предложенная конфигурация не является типовой и предназначена для использования в учебном заведении. Здесь изложены все возможности данной конфигурации, все ее преимущества и возможные направления развития.

Для эффективного управления делами студентов, денежными расчетами, учета численности и контингента студентов необходима полная автоматизация всего документооборота, относящегося к студентам. На данный момент для решения этой задачи в ФБОУ ВПО «ВГАВТ» используется АСУ «Деканат» на платформе 1С: Предприятие 8.2. На сегодняшний день данная версия платформы является новейшей и отвечает всем требованиям вуза.

Конфигурация «Деканат» не является типовой и была специально разработана для академии. Условно база данных (БД) была разделена на следующие подсистемы:

– Успеваемость студента
– Расчеты со студентами
– Движение студентов

Данная конфигурация включает в себя следующих пользователей:

– Деканаты
– Учебный отдел
– Отдел кадров
– Бухгалтерия
– Отдел магистратуры
– Прочие (военно-учетный стол, профком студентов, приемная комиссия, бюро пропусков).

Каждая группа пользователей решает свой круг задач, в связи с чем для каждой группы был разработан различный функционал для обработки БД и предусмотрены