Y_1 (113) = {12110111101} с момента t =113c по 118c – переходный процесс (информация отсеивается системой сбора и обработки информации)
Y_1 (114) = {11110111111}
Y_1 (115) = {11111111111}
Y_1 (116) = {11111111111}
Y_1 (117) = {10111111111}
Y_1 (118) = {10111111111}
Y_1 (119) = {11111111111}

Таким образом, для получения кодов ситуаций необходимо за время выхода вторых производных состояния всех контролируемых элементов системы из допустимой зоны сделать не менее 5–7 опросов датчиков, чтобы выделить основной код. Временя выхода и оценка минимального времени (t_{min}) может быть определена только опытным путем и t_{min} различено для различных систем.

А.М. Пушкин
Московский государственный университет информационных технологий, радиотехники и электроники

О МОДЕЛЯХ И АЛГОРИТМЕ СИНТЕЗА СТРАТЕГИЙ ОБЛУЖИВАНИЯ СИСТЕМЫ СТАЦИОНАРНЫХ ОБЪЕКТОВ

Вводится модель обслуживания пространственно рассроченных в одномерной рабочей зоне стационарных объектов перемещающимся процессором при выполнении им рейса между крайними точками рабочей зоны. Процессору доступны ограниченные возвраты к ранее оставшимся необслуженными объектам. С каждым объектом ассоциированы параметры: ранний срок начала обслуживания, время обслуживания, функция индивидуального штрафа. Формулируется бикритериальная задача синтеза оптимальных стратегий обслуживания и в рамках концепции Парето строится решающий алгоритм.

Введение. Рассматриваемая модель предназначена для описания систем, в которых перемещающийся процессор должен обслужить совокупность стационарных объектов, линейно рассредоточенных по пунктам одномерной рабочей зоны. Считается, что процессор выполняет однократный рейс из начального пункта в конечный, в процессе которого допускаются возвраты на заданное условиями задачи количество пунктов назад. С каждым объектом ассоциированы ранний срок начала обслуживания, время обслуживания и индивидуальный штраф, являющийся монотонно возрастающей функцией от момента завершения обслуживания объекта. В качестве минимизируемых критериев выступают общее время работы процессора на рабочей зоне и суммарный по всем объектам штраф.

Подобная модель была рассмотрена в [Список литературы:
], где также было наложено ограничение на перемещения процессора – выполнялся единственный рейс из начального пункта в конечный, при этом разрешались возвраты не далее одного пункта назад. Следует также отметить работы [**Ошибка! Источник ссылки не найден.**, **Ошибка! Источник ссылки не найден.**], где бикритериальные задачи синтеза расписаний рассматривались в предположении, что обслуживание выполняется при реализации процессором двух рейсов – прямого и обратного – без возможности возвратов.

Для формулируемой задачи принимается концепция Парето, предусматривающая синтез полной совокупности эффективных оценок [**Ошибка! Источник ссылки не найден.**, **Ошибка! Источник ссылки не найден.**]. Из этой совокупности ЛПР (лицо, принимающее решения) осуществляет свой выбор. Благодаря технологии выполненного счета, по выбранной оценке, легко определяется оптимальное расписание обслуживания.

Математическая модель. Задана совокупность $O_n = \{o_1, o_2, \ldots, o_n\}$ стационарных объектов, расположенных соответственно в точках $1, 2, \ldots, n$ одномерной рабочей зоны *L* обслуживающего процессора *P*. Начальная точка *A* является базой для процессора. Объекты пронумерованы в порядке возрастания их расстояний от точки *A*. В конечной точке *B* зоны *L* расположен объект o_n.

Примем следующие обозначения: r_j – ранний срок начала обслуживания объекта o_j (до этого срока обслуживание объекта o_j либо невозможно, либо не имеет смысла); τ_j – требуемое время обслуживания процессором *P* объекта o_j; $\varphi_j(t)$ – монотонно возрастающая в нестрогом смысле функция индивидуального штрафа, выражающая зависящую от момента завершения обслуживания o_j величину экономических потерь; $\gamma_{j-1,j}$ и $\gamma_{j,j-1}$ – затраты времени на перемещения процессора от точки $j-1$ до точки j и от точки j до точки $j-1$ соответственно. Через $\gamma_{p,q}$ будем обозначать затраты времени на перемещение процессора от точки *p* до точки *q*. Если $p < q$, то $\gamma_{p,q} = \sum_{j=p}^{q-1} \gamma_{j,j+1}$; если $p > q$, то $\gamma_{p,q} = \sum_{j=q}^{p-1} \gamma_{j+1,j}$. Все числа r_j, τ_j, $\gamma_{j-1,j}$ и $\gamma_{j,j-1}$ считаем целыми положительными.

В момент времени $t = 0$ из точки *A* процессор *P* начинает движение к точке *B*. При реализации рейса *AB* процессор выполняет однократное без прерываний обслуживание каждого объекта группы O_n. При необходимости процессор может перенести обслуживание объектов $o_j, o_{j+1}, ..., o_{j+k-1}$ на более поздний срок, назначая их обслуживание непосредственно после объекта o_{j+k}, т.е. разрешаются возвраты не далее k пунктов назад. Не связанные с обслуживанием объектов и наступлением ранних сроков промежуточные простои процессора запрещены.

Постановка оптимизационной задачи. Стратегией именуем допустимую последовательность $\rho = (l_1, l_2, \ldots, l_n)$, являющаяся перестановкой совокупности $N = \{1, 2, ..., n\}$ индексов объектов, определяющая порядок их обслуживания. Согласно вышеотмеченному, условие допустимости стратегии записывается следующим образом: $|l_u - u| \le k$, $u = \overline{1, n}$.

Минимизируемыми критериями являются следующие показатели: 1) $K_1(\rho)$ – общее время работы процессора; 2) $K_2(\rho)$ – суммарный по всем объектам штраф. С позиций повышения эффективности диспетчерского управления возникает бикритериальная задача:

$$\min_{\rho} \{K_1(\rho), K_2(\rho)\}. \tag{1}$$

При решении бикритериальной задачи **Ошибка! Источник ссылки не найден.** принимается концепция Парето [**Ошибка! Источник ссылки не найден.**].

Алгоритм решения. Алгоритм основан на принципе динамического программирования. Задачу **Ошибка! Источник ссылки не найден.** обозначим символом Z. Введём совокупность частных задач $Z(i,t)$. В каждой задаче $Z(i,t)$ рассматривается множество объектов с индексами от 1 по i, которые на момент времени t должны быть обслужены процессором с допустимыми возвратами или без таковых; сам процессор в момент времени t должен находиться в точке i . Пары (i,t) далее будем называть состояниями системы обслуживания. Через $Q(i,t)$ обозначим суммарный штраф в задаче $Z(i,t)$.

В рассматриваемой модели допускаются возвраты не далее k пунктов назад, поэтому для состояния (i,t) существует ряд вариантов движения процессора:

1) процессор перемещается в точку $i+1$ и далее обслуживает объект o_{i+1};

2) для каждого фиксированного параметра m, $m=\overline{2,k+1}$, $i+m \le n$, процессор поэтапно: а) перемещается в точку $i+m$, где обслуживает объект o_{i+m}; б) возвращается в точку $i+1$, где обслуживает объект o_{i+1}; в) движется в точку $i+m$, последовательно обслуживая по пути объекты $o_{i+2}, o_{i+3}, \ldots, o_{i+m-1}$.

Для процессора, выполнившего обслуживание всех объектов, кроме последнего, и находящегося в точке ($n-1$) возможно только перемещение в точку n с дальнейшим обслуживанием объекта o_n.

Опишем вычислительную процедуру последовательно, в порядке увеличения значений аргументов. Процесс будет поддерживаться с помощью массива *M*, в котором указываются записи вида $\{i,\, t,\, Q(i,t),\, \rho(i,t)\}$, где параметры i и t описывают состояние (i,t) системы обслуживания, $Q(i,t)$ – суммарный штраф для состояния (i,t), $\rho(i,t)$ – расписание обслуживания, которое приводит систему в состояние (i,t) при суммарном штрафе $Q(i,t)$. Назовем запись $\{i,\, t,\, Q(i,t),\, \rho(i,t)\}$ из M эффективной, если при одинаковом параметре i для пары значений $\{t, Q(i,t)\}$ этой записи не найдется пары $\{t', Q'(i,t')\}$ другой записи из M такой, что $t' \le t$ и $Q'(i,t') \le Q(i,t)$, причем, по меньшей мере, одно из записанных неравенств выполняется как строгое неравенство.

В пошаговом режиме алгоритм решения задачи **Ошибка! Источник ссылки не найден.** функционирует следующим образом.

1. В M добавляется начальная запись $\{0,0,0,(-)\}$; состояние $(0,0)$ выбирается для дальнейшего раскрытия.

2. Раскрывается выбранное состояние системы (i,t), т.е. вводятся новые состояния системы и вычисляются величины суммарного штрафа, соответствующие этим состояниям; для i, $i=\overline{0,n-2}$:

– $(i+1,t_1)$, для которого $Q(i+1,t_1)=Q(i,t)+\varphi_{i+1}(t_1)$;

– $\left(i+m,t_m^e\right)$, для которого $Q\left(i+m,t_m^e\right)=Q(i,t)+\varphi_{i+m}(t_m)+\Phi_m^e$ (для каждого фиксированного параметра m, $m=\overline{2,k+1}$, $i+m \le n$).

Здесь указаны следующие параметры:

а) $t_1 = \max\left(t+\gamma_{i,i+1},\, r_{i+1}\right)+\tau_{i+1}$ – время, которое необходимо процессору для перемещения и обслуживания объекта o_{i+1};

б) $t_m = \max\left(t + \gamma_{i,i+m},\, r_{i+m}\right) + \tau_{i+m}$ – время, которое процессору необходимо для перемещения в точку $i+m$ и обслуживания объекта o_{i+m}, минуя объекты $o_{i+1}, o_{i+2}, ..., o_{i+m-1}$;

в) t_m^e – время, необходимое процессору для возвращения из точки $i+m$ в точку $i+1$ и дальнейшего последовательного обслуживания объектов $o_{i+1}, o_{i+2}, ..., o_{i+m-1}$, и перемещения после этого в точку $i+m$;

г) Φ_m^e – штраф, получаемый процессором, после возвращения из точки $i+m$ в точку $i+1$ и последующего последовательного обслуживания объектов $o_{i+1}, o_{i+2}, ..., o_{i+m-1}$.

Момент времени t_e и штраф Φ_m^e определяются следующим образом:

а) полагаются $g := 1$, $T_g := \max\left(t + \gamma_{i+m,i+1}, r_{i+1}\right) + \tau_{i+1}$, $F_g := \varphi_{i+1}\left(T_g\right)$;

б) итеративно для $p = i+1, i+2, ..., i+m-2$ (где $i+1 \le i+m-2$) выполняется ряд действий: $T_{g+1} := \max\left(T_g + \gamma_{p,p+1}, r_{p+1}\right) + \tau_{p+1}$, $F_{g+1} := \varphi_{p+1}\left(T_{g+1}\right)$, $g := g+1$;

в) полагаются $T_{g+1} := T_g + \gamma_{i+m-1,i+m}$, $t_m^e := T_{g+1}$, $\Phi_m^e := F_g$.

Для $i = n-1$ вводится единственное состояние $\left(i+1, t_1\right)$.

3. Состояние (i,t) считается раскрытым, оно изымается из M; полученные состояния системы добавляются в M.

4. Среди записей из M выделяются эффективные; неэффективные записи из M изымаются.

5. Дополняются расписания в новых записях из M; для состояния $\left(i+1, t_1\right)$ расписание расширяется переходом в точку $i+1$; для каждого состояния $\left(i+m, t_m^e\right)$, $m = \overline{2, k+1}$, $i+m \le n$ – подпоследовательностью $\left(i+m, i+1, i+2, ..., i+m-1\right)$.

6. Из M выбирается состояние системы с минимальным параметром i; если таких состояний несколько, то выбирается состояние с наименьшим параметром t. Если в некоторой записи параметр i равен n, то соответствующее состояние считается финальным и далее не рассматривается.

7. Для выбранного состояния повторяются шаги 2–7; если выбрать новое состояние не удается (все состояния в M финальны), то алгоритм прекращает свою работу.

После работы алгоритма в M останется одна или несколько эффективных записей; этим записям соответствуют финальные состояний системы. Выбор среди таких записей осуществляет ЛПР. Выбранная запись содержит необходимую ЛПР стратегию обслуживания рассредоточенных стационарных объектов.

Заключение. В работе описана модель одностадийного обслуживания группы пространственно рассредоточенных стационарных объектов, расположенных вдоль одномерной рабочей зоны перемещающегося процессора. Сформулирована задача с практически значимыми критериями, предложен алгоритм синтеза полной совокупности эффективных записей, описывающих для конкретных состояний время работы процессора, суммарный по всем объектам штраф и стратегию обслуживания, приводящая систему в данное состояние.

Список литературы:

[1] Пушкин А.М. Модель обслуживания стационарных объектов перемещающимся процессором с возможностью возвратов / А.М. Пушкин // Научно-технический вестник Поволжья. №5. 2014 г. – С. 288-292.
[2] Коган Д.И. Задачи синтеза оптимальных стратегий обслуживания стационарных объектов в одномерной рабочей зоне процессора / Д.И. Коган, Ю.С. Федосенко // Автоматика и телемеханика. – 2010. – №10. – С. 50–62.
[3] Коган Д.И. Бикритериальные задачи обслуживания стационарных объектов в одномерной рабочей зоне процессора / Д.И. Коган, Ю.С. Федосенко, Н.А. Дуничкина // Автоматика и телемеханика. 2012. №10. – С. 93–110.
[4] Подиновский В.В. Парето-оптимальные решения многокритериальных задач / В.В. Подиновский, В.Д. Ногин. – М.: Наука. Главная редакция физико-математической литературы, 1982. – 256 с.
[5] Штойер Р. Многокритериальная оптимизация. Теория, вычисления и приложения: Пер. с англ. / Р. Штойер – М.: Радио и связь, 1992. – 504 с.: ил.

А.В. Романов
ФГБОУ ВО «ВГУВТ»

ОБ АВТОМАТИЧЕСКОЙ КОРРЕКЦИИ МОДЕЛИ ДИНАМИКИ ВОДОИЗМЕЩАЮЩЕГО СУДНА В ПРОЦЕССЕ ЕГО ДВИЖЕНИЯ

Определение коэффициентов по натурным данным в штатном режиме движения судна осложняется недостаточной точностью или отсутствием на судах датчика угла дрейфа. Чувствительность датчика угла дрейфа не превышает 0,5–1 градуса и, в процессе удержания на заданном курсе угол дрейфа, как правило, неизмерим. При погрешности измерения координат состояния судна более 2–3% от среднего значения коэффициентов модели рассчитываются неоднозначно. Автоматический выбор комбинаций коэффициентов занимает значительное время и не всегда дает значения соответствующие истинной динамике судна.

Так как коэффициенты модели динамики судна более всего зависят от относительной осадки *T/H*, в данной работе по графикам зависимостей гидродинамических характеристик транспортных судов от относительной осадки, полученных В.И. Коганом [1], были выведены поправочные множители коэффициентов для произвольной глубины судового хода.

Большинство динамических эффектов, наблюдаемых при движении судов, можно описать с помощью модели, предложенной А.М. Басиным [1]:

$$\frac{d\beta}{dt} = -q_{21}\beta - r_{21}\omega - s_{21}\alpha - h|\beta|\beta, \qquad \frac{d\omega}{dt} = -q_{31}\beta - r_{31}\omega - s_{31}\alpha, \qquad \frac{d\varphi}{dt} = \omega, \tag{1}$$

где ω – угловая скорость судна в горизонтальной плоскости,
β – угол дрейфа,
α – угол перекладки руля,
φ – курсовой угол,
t – время.

Коэффициенты уравнений (1) обычно рассчитываются теоретически по конструктивным параметрам судна с использованием результатов натурных экспериментов на физических моделях и на реальных судах при выполнении определенных маневров.

Предположим, что гидродинамические коэффициенты q_{31}, r_{31}, s_{31}, q_{21}, r_{21}, s_{21}, h рассчитаны по данным, измеренным при энергичном маневре (циркуляции) для