

УДК 519.8 + 681.3

БИКРИТЕРИАЛЬНАЯ МОДЕЛЬ И АЛГОРИТМЫ СИНТЕЗА ОПТИМАЛЬНО-КОМПРОМИССНЫХ СТРАТЕГИЙ ОДНОПРОЦЕССОРНОГО ОБСЛУЖИВАНИЯ ЛИНЕЙНОЙ ГРУППИРОВКИ СТАЦИОНАРНЫХ ОБЪЕКТОВ

© 2012 г.

Н.А. Дуничкина

Волжская государственная академия водного транспорта, Нижний Новгород

nadezhda.dunichkina@gmail.com

Поступила в редакцию 16.06.2011

Рассматривается дискретная модель однофазного обслуживания группировки стационарных объектов, рассредоточенных вдоль одномерной рабочей зоны mobile-процессора. С каждым объектом ассоциирована пара монотонно возрастающих функций индивидуального штрафа. Формулируются задачи синтеза в плоскости критериев полной совокупности эффективных оценок и соответствующих им Парето-оптимальных стратегий обслуживания. Выводятся решающие соотношения динамического программирования, позволяющие построить всю Парето-область. Отдельное внимание уделяется алгоритмам синтеза субоптимальных стратегий обслуживания.

Ключевые слова: дискретная модель обслуживания, Парето-оптимальные стратегии, динамическое программирование, метаэвристические алгоритмы.

Введение

Рассматриваемая в статье проблема возникла в связи с созданием компьютерных средств поддержки оперативного управления снабжением дизельным топливом плавучих дизель-электрических комплексов, осуществляющих русловую добычу нерудных строительных материалов (НСМ) в крупномасштабных русловых районах внутренних водных путей РФ. Примерами таких полигонов, протяженностью до нескольких сотен километров, являются Камский грузовой район и 4-й грузовой район на реке Белой. В навигационный период в каждом таком районе работает группировка из 15–20 единиц плавучих добывающих комплексов (ПДК). В непрерывном технологическом цикле добывающий комплекс осуществляет выемку, обезвоживание и погрузку НСМ в речные суда и составы для последующей транспортировки их к местам складирования и реализации предприятиям-потребителям [1].

Снабжение ПДК дизельным топливом осуществляется закрепленными за полигоном специализированными танкерами-заправщиками.

В условиях высоких расходов на эксплуатацию ПДК и цен на дизельное топливо, сокращения возможностей предприятий по созданию его технически значимых оперативных запасов основная задача диспетчера группировки (лица, принимающего решения – ЛПР) заключается в

выработке (и последующем обеспечении реализации) такой стратегии снабжения группировки ПДК дизельным топливом, при которой минимизируются экономические потери, связанные с их непроизводительными простоями.

В статье рассматривается математическая модель одной из типовых технологических схем снабжения линейно рассредоточенной вдоль полигона группировки ПДК, согласно которой доставка дизельного топлива осуществляется одним танкером-заправщиком в процессе выполнения кругового рейса – от базовой, «начальной» точки полигона к его дальней граничной точке и обратно. При этом качество плана снабжения группировки ПДК дизельным топливом оценивается по двум независимым критериям, отражающим те или иные имеющие экономический смысл потери в связи с её реализацией. Выбор конкретных пар критериев оценки плана снабжения зависит от складывающейся на горизонте оперативного планирования эксплуатационной ситуации.

В целях разработки инструментария для штатного решения вышеуказанных бикритериальных задач в работе предлагаются алгоритмы синтеза стратегий обслуживания, реализующие в рамках концепции Парето [2] идеологию динамического программирования [3–5], а также алгоритмы синтеза субоптимальных стратегий обслуживания. Приводятся примеры вычислительной реализации разработанных алгоритмов.

**1. Математическая модель
и постановка оптимизационной задачи**

Считается заданной группировка $O_n = \{o_1, o_2, \dots, o_n\}$ стационарных объектов, расположенных в рабочей зоне Ξ обслуживающего процессора P . Зона Ξ представляет собой отрезок L , в фиксированных точках которого l_1, l_2, \dots, l_n ($l_1 \leq l_2 \leq \dots \leq l_n$) расположены соответственно объекты o_1, o_2, \dots, o_n . Граничная точка A отрезка L является базовой для процессора. Другая граничная точка B является местом расположения объекта o_n . Из точки A , начиная с момента времени $t = 0$, процессор P перемещается к граничной точке B отрезка L (прямой рейс $-\lambda_+$), а затем, достигнув ее, возвращается в точку A (обратный рейс $-\lambda_-$). При реализации цикла $\lambda_+\lambda_-$ процессор должен выполнить однократное, без прерываний, обслуживание каждого объекта группировки O_n ; при этом обслуживание части объектов реализуется в рейсе λ_+ , а обслуживание оставшихся объектов – в рейсе λ_- . Не связанные с обслуживанием объектов простои процессора и одновременное обслуживание процессором двух и более объектов недопустимы.

С каждым объектом o_j ассоциируется функция индивидуального штрафа $\varphi_j(t)$, выражающая зависящую от момента завершения его обслуживания величину потерь ($j = \overline{1, n}$). Если в связи со специфическими условиями реализации обслуживания удобно говорить о штрафе по объекту o_j ($j = \overline{1, n}$) за некоторый временной интервал $[p, q]$, то в этом случае величина индивидуального штрафа за данный временной отрезок равна $\varphi_j(q) - \varphi_j(p)$. Для произвольного подмножества объектов штраф за отрезок времени $[p, q]$ определяется как сумма индивидуальных штрафов за этот промежуток времени по всем объектам подмножества.

Для используемых ниже норм времени примем обозначения: τ_j – длительность обслуживания объекта o_j ; $\gamma_{j-1,j}$ и $\gamma_{j,j-1}$ – продолжительность перемещения процессора P между точками расположения соседних объектов в рейсах λ_+ и λ_- соответственно ($j = \overline{1, n}$); при этом $\gamma_{0,1}$ и $\gamma_{1,0}$ – продолжительности перемещения процессора между базовой точкой A и точкой l_1 в рейсах λ_+ и λ_- . Параметры $\tau_j, \gamma_{j-1,j}, \gamma_{j,j-1}$ далее будем считать принимающими целочисленные значения.

Стратегией обслуживания будем называть произвольное подмножество элементов V из совокупности индексов $N = \{1, 2, \dots, n\}$, содержащее элемент n . Объекты $o_j, j \in V$, в реализации стратегии V обслуживаются процессором

P в рейсе λ_+ , а все остальные объекты группировки O_n – в рейсе λ_- . Для определенности полагаем, что объект o_n обслуживается при завершении процессором рейса λ_+ . Таким образом, число различных стратегий обслуживания равно 2^{n-1} . Любая стратегия однозначно определяет моменты начала и завершения обслуживания каждого из объектов. Для объекта o_j через $t_j^*(V)$ обозначим момент завершения его обслуживания при реализации стратегии $V, j = \overline{1, n}$. Если объект o_j обслуживается в рейсе λ_+ ($j \in V$), то

$$t_j^*(V) = \sum_{k=1}^j \gamma_{k-1,k} + \sum_{k \in V(j)} \tau_k,$$

где $V(j)$ – совокупность не превосходящих j элементов из V . Для

$$\text{суммы } \sum_{k=1}^j \gamma_{k-1,k} + \sum_{k \in V(j)} \tau_k \text{ введем обозначение } \delta_j.$$

Если $j \in M \setminus V$, то обслуживание объекта o_j реализуется позже, поскольку процессор должен предварительно выполнить перемещения от точки l_j до точки B и обратно, обслужив при этом объекты $o_{j+1}, o_{j+2}, \dots, o_n$.

Обозначим через $M(j)$ задержку по объекту o_j при переносе его обслуживания с рейса λ_+ на рейс λ_- . Ясно, что $M(j)$ – сумма трех слагаемых:

$$\sum_{k=j+1}^j \gamma_{k-1,k} \text{ (затраты времени на перемещение$$

$$\text{процессора от точки } l_j \text{ к точке } B), \sum_{k=j+1}^j \gamma_{k,k-1} \text{ (за}$$

$$\text{траты времени на перемещение процессора от}$$

$$\text{точки } B \text{ к точке } l_j), \sum_{k=j+1}^n \tau_k \text{ (суммарная продол}$$

жительность обслуживания объектов $o_{j+1}, o_{j+2}, \dots, o_n$). В итоге для $j = \overline{1, n}$ имеем общую формулу

$$t_j^*(V) = \begin{cases} \delta_j, & \text{если } j \in V, \\ \delta_j + \tau_j + M(j), & \text{если } j \notin V. \end{cases}$$

Существенно, что величины $M(j)$ ($j = \overline{1, n-1}$) не зависят от того, какие объекты обслуживаются в прямом, а какие – в обратном рейсе процессора P .

С позиций повышения эффективности диспетчерского управления возникает задача построения полной совокупности Парето-оптимальных стратегий обслуживания для бикритериальной проблемы

$$\min_V \left(\sum_{j=1}^n \varphi_j(t_j^*(V)), \min_V \sum_{j=1}^n \psi_j(t_j^*(V)) \right) \quad (1)$$

минимизации суммы индивидуальных штрафов φ_j и ψ_j по всем объектам группировки O_n .

2. Алгоритм синтеза совокупности Парето-оптимальных стратегий обслуживания

Для синтеза совокупности Парето-оптимальных стратегий обслуживания введем следующие операции.

Записью вида $eff(M)$ будем обозначать максимальное по включению подмножество недоминируемых в M векторов, где M – произвольное множество двумерных векторов-оценок.

Выражением вида $Y \oplus x$ будем обозначать совокупность всех векторов v , представимых в виде $v = y + x$, где $y \in Y$.

Введем в рассмотрение ряд частных задач $Z(i, D)$, отличающихся от задачи (1) следующим: функции индивидуального штрафа по объектам $o_{i+1}, o_{i+2}, \dots, o_n$ тождественно нулевые, а общее время, затрачиваемое процессором P на обслуживание в рейсе λ_+ объектов o_1, o_2, \dots, o_i , равно D . Через $E(i, D)$ обозначим полную совокупность эффективных оценок в задаче $Z(i, D)$. Пары (i, D) далее будем именовать ситуациями. Не все ситуации реализуемы в процессах обслуживания. Так, для $i = 1$ реализуемы только ситуации $(1, \tau_1)$ и $(1, 0)$. Удобно считать, что для всех нереализуемых ситуаций (i, D) имеет место соотношение $E(i, D) = \{(+\infty, +\infty)\}$. Отметим также, что для всех значений параметра D , не принадлежащих множеству $\{0, 1, 2, \dots, \sum_{k=1}^n \tau_k\}$, ситуации (i, D) нереализуемы. В частности, $E(i, D) = \{(+\infty, +\infty)\}$ при всех отрицательных значениях D .

Как очевидно, ситуация $(1, 0)$ соответствует обслуживанию объекта o_1 в рейсе λ_- , которое завершается в момент времени $\gamma_{0,1} + M(1) + \tau_1$. Напомним, что $M(j)$ – задержка по объекту o_j при переносе его обслуживания с прямого рейса на обратный ($j = \overline{1, n-1}$) – определяется как сумма трех слагаемых: $\sum_{k=j+1}^j \gamma_{k-1,k}$ (затраты времени на перемещение процессора P от точки l_j к точке B), $\sum_{k=j+1}^j \gamma_{k,k-1}$ (затраты времени на перемещение процессора P от точки B к точке l_j), $\sum_{k=j+1}^n \tau_k$ (суммарная продолжительность обслуживания объектов $o_{j+1}, o_{j+2}, \dots, o_n$). Ситуация $(1, \tau_1)$ описывает обслуживание объекта o_1 в рейсе λ_+ , завершающееся в момент $\gamma_{0,1} + \tau_1$. При $D \notin \{0, \tau_1\}$ величины $E(1, D)$ смысла не имеют. Поэтому для задачи (1) имеют место соотношения

$$E(1, 0) = \{\varphi_1(\gamma_{0,1} + M(1) + \tau_1), \psi_1(\gamma_{0,1} + M(1) + \tau_1)\}, \quad (2)$$

$$E(1, \tau_1) = \{\varphi_1(\gamma_{0,1} + \tau_1), \psi_1(\gamma_{0,1} + \tau_1)\}, \quad (3)$$

$$E(1, D) = \{(+\infty, +\infty)\} \text{ при } D \notin \{0, \tau_1\}. \quad (4)$$

Пусть все значения $E(i, D)$ для некоторого $i \in \{1, 2, \dots, n-2\}$ уже найдены. При отыскании значений $E(i+1, D)$ следует учитывать следующие две возможности:

1) объект o_{i+1} обслуживается в рейсе λ_+ , и тогда рассматриваемой ситуации $(i+1, D)$ непосредственно предшествует ситуация $(i, D - \tau_{i+1})$;

2) объект o_{i+1} обслуживается в рейсе λ_- , и тогда ситуации $(i+1, D)$ непосредственно предшествует ситуация (i, D) .

С учетом указанных возможностей для $i \in \{1, 2, \dots, n-2\}$ получаем соотношение

$$E(i+1, D) = eff(K_1, K_2), \quad (5)$$

где

$$K_1 = E(i, D - \tau_{i+1}) \oplus (\varphi_{i+1}(\sum_{j=0}^i \gamma_{j,j+1} + D), \psi_{i+1}(\sum_{j=0}^i \gamma_{j,j+1} + D)), \quad (6)$$

$$K_2 = E(i, D) \oplus (\varphi_{i+1}(\sum_{j=n}^{i+1} \gamma_{j+1,j} + \tau_{i+1} + D + M(i+1)), \psi_{i+1}(\sum_{j=n}^{i+1} \gamma_{j+1,j} + \tau_{i+1} + D + M(i+1))). \quad (7)$$

Величины K_1 и K_2 ниже именуем соответственно первой и второй компонентами формулы (5), вычисляемое по которой значение $E(i+1, D)$ оказывается равным $\{(+\infty, +\infty)\}$ в том и только том случае, когда ситуация $(i+1, D)$ нереализуема.

При определении значений $E(n, D)$ следует иметь в виду, что объект o_n обслуживается в рейсе λ_+ . Поэтому для задачи (1) имеем соотношение

$$E(n, D) = E(n-1, D - \tau_n) \oplus (\varphi_{n+1}(\sum_{j=0}^{n-1} \gamma_{j,j+1} + D), \psi_{n+1}(\sum_{j=0}^{n-1} \gamma_{j,j+1} + D)), \quad (8)$$

и полная совокупность эффективных оценок в задаче (1) определяется по формуле

$$E = eff(\bigcup_{D=0}^{\sum_{k=1}^n \tau_k} E(n, D)). \quad (9)$$

Формулы (2)–(4), (8) суть рекуррентные соотношения динамического программирования, позволяющие совместно с (5)–(7), (9) получить решение задачи (1).

3. Алгоритмы синтеза субоптимальных стратегий обслуживания

Наряду с синтезом точных решений задачи (1), требующим, вообще говоря, значительных и не всегда допустимых в производственных условиях временных затрат, интерес представляют быстро работающие алгоритмы построения приближенных решений с приемлемыми для приложений погрешностями. Ниже приводится описание трех таких алгоритмов (TS, SA и GA), сконструированных с использованием метаэвристических концепций поиска с запретами (taboo search) [6], имитации отжига (simulated annealing) [7] и эволюционно-генетического моделирования (genetic algorithm) [8]. Их применение для синтеза субоптимальных стратегий обслуживания является особенно значимым при больших размерностях задачи (1) ввиду ее NP-трудности [9].

Введем операцию *добавления без нарушения эффективности* (ДБНЭ). Пусть имеется оценка (W, Q) и множество оценок M , такое, что ни одна из оценок этого множества не доминирует другую оценку этого множества или, другими словами, $M = \text{eff}(M)$. В результате применения операции ДБНЭ к оценке (W, Q) и к множеству M получим множество $\text{Eff}(M \cup \{(W, Q)\})$.

Если в множестве M существуют оценки, доминируемые оценкой (W, Q) , то эти оценки удаляются из множества M , а оценка (W, Q) добавляется. Будем называть эту ситуацию успешной.

Если же оценка (W, Q) доминируется некоторой оценкой множества M , то она не добавляется. Будем называть эту ситуацию неуспешной.

В качестве текущего решения в алгоритмах SA, TS или особи для алгоритма GA будем полагать булев вектор $i = (i_1, i_2, \dots, i_{n-1})$, такой, что $i = 1$ тогда и только тогда, когда объект o_i обслуживается в рейсе λ_+ . Такой вектор однозначно определяет стратегию $V(i)$ обслуживания группировки объектов, поскольку объект o_n всегда обслуживается в прямом рейсе. Данной стратегии соответствуют значения критериев

$$\begin{aligned} (W_{V(i)}, Q_{V(i)}) &= (W_i, Q_i) = \\ &= \left(\sum_{j=1}^n \varphi_j(t_j^*(V(i))), \sum_{j=1}^n \psi_j(t_j^*(V(i))) \right). \end{aligned}$$

Под $\text{NB}(i, j)$ ($1 \leq j \leq n-1$) будем понимать булев вектор $(i'_1, i'_2, \dots, i'_{n-1})$: $i'_j = 1 - i_j$, $i'_l = i_l$ для $\forall l \in \{1, \dots, n-1\}$, $l \neq j$, соответствующий стратегии, полученной из текущей стратегии перестановкой одного произвольного объекта o_j с обслуживанием в рейсе λ_+ (λ_-) на обслуживание в рейсе λ_- (λ_+).

3.1. Алгоритм TS. В основе этого алгоритма лежит схема, позволяющая осуществлять поиск глобального оптимума, не останавливаясь в локальном оптимуме; при этом основной идеей является формирование в процессе поиска так называемого списка запретов, который блокирует зондирование части или всей окрестности текущего решения.

Для учета в схеме поиска с запретами специфики задачи (1) будем считать множество векторов $\{\text{NB}(i, j) : j = \overline{1, n-1}\}$ окрестностью точки $i = (i_1, i_2, \dots, i_{n-1})$. Число векторов в такой окрестности равно $n-1$.

В процессе поиска с запретами нам потребуется сравнивать некоторую оценку (W, Q) с эталоном – другой оценкой (W', Q') . Будем говорить, что оценка (W, Q) проходит операцию *сравнения с эталоном*, если $(W - W')^2 + (Q - Q')^2 \leq \varepsilon$, где ε – наперед заданное оценочное число. Иначе будем считать, что оценка не проходит операцию сравнения с эталоном.

С учетом сделанных пояснений алгоритм TS решения задачи (1) записывается в виде следующей пошаговой последовательности действий.

1. Выбрать в качестве начального решение $U = (1, 1, \dots, 1)$, соответствующее обслуживанию всех объектов в рейсе λ_+ .

2. Очистить список запретов, а множество оценок M считать состоящим из точки (W_U, Q_U) , объявляемой на данном шаге эталоном для сравнения.

3. Для всех точек U' из окрестности решения U , не входящих в список запретов, вычислить значения функций штрафов $(W_{U'}, Q_{U'})$.

4. Применить операцию ДБНЭ к оценке $(W_{U'}, Q_{U'})$ и множеству M .

5. Если результат операции ДБНЭ на шаге 4 является неуспешным и оценка $(W_{U'}, Q_{U'})$ не проходит сравнение с эталоном, то добавить решение U' в список запретов; перейти к шагу 7.

6. Если результат операции, выполненной на шаге 4, является успешным, то считать оценку $(W_{U'}, Q_{U'})$ эталоном для сравнения, а соответствующее ей решение U' добавить в список запретов.

7. Если в результате выполнения шагов 3–6 эталон изменился, то положить текущее решение U равным решению, соответствующему эталону; иначе в качестве текущего выбрать решение, соответствующее случайно выбранной оценке из M .

8. Перейти к шагу 3, если не выполнено условие останова; иначе завершение работы алгоритма; M – множество субоптимальных оценок.

Стоит отметить, что длина списка запретов считается заданной, и из него по принципу очереди начинают удаляться элементы в порядке поступления в список.

3.2. Алгоритм SA. Функционирование алгоритма основано на моделировании физического процесса управляемого охлаждения и применительно к задаче (1) может быть описано следующей последовательностью действий.

1. Зафиксировать значение начальной температуры $T = T_{\max}$ и коэффициента охлаждения $c = c_0$.

2. Выбрать в качестве начального решение $U = (1, 1, \dots, 1)$, соответствующее обслуживанию всех объектов в прямом рейсе; множество субоптимальных оценок считать состоящим из одного элемента $M = \{W(U), Q(U)\}$.

3. Определить точку U' как $NB(U, j)$, где j – случайно выбранное число в интервале от 1 до $n - 1$.

4. Вычислить изменения функций штрафов при переходе к точке U' :

$$\Delta_1 = W(U') - W(U), \quad \Delta_2 = Q(U') - Q(U).$$

5. Если выполняется хотя бы одно из следующих условий:

$$\Delta_1 < 0 \text{ или } \Delta_2 < 0 \text{ или } \Delta_1 = 0 \text{ и } \Delta_2 = 0, \quad (10)$$

применить операцию ДБНЭ к точке $(W(U'), Q(U'))$ и множеству M ; текущую точку U заменить на U' .

6. Если ни одно из условий (10) не выполняется, переход в точку $U = U'$ осуществляется тогда и только тогда, когда выполняется условие

$e^{-\left(\frac{\Delta_1^2 + \Delta_2^2}{T^2}\right)} > R$, где R – случайно выбранное число из диапазона $(0, 1)$; в противном случае текущая точка не изменяется.

7. Значение температуры уменьшается согласно закону охлаждения $T = c \cdot T$.

8. Циклически повторяются шаги 3–7 до тех пор, пока не будет произведено заданное количество итераций или совокупность оценок M не будет изменяться на протяжении заданного количества шагов.

По завершению последовательности действий (1)–(8) множество M будет представлять собой совокупность субоптимальных оценок.

3.3. Алгоритм GA. Общими процедурами в различных реализациях эволюционно-генетических алгоритмов являются: а) создание начальной популяции; б) вычисление значений функций приспособленности для всех особей популяции; в) циклический процесс до момента выполнения критерия останова: селекция, скрещивание, мутация, вычисление приспособленности для всех особей, формирование нового поколения.

Применительно к задаче (1) реализация решающего алгоритма может быть описана следующим образом.

Начальная популяция population создается из N особей, сгенерированных случайным образом с использованием датчика случайных чисел. Численное значение N считается заданным.

Функцией приспособленности считается вектор-функция $(-W_V, -Q_V)$, компоненты которой вычисляются как значения критериев на соответствующей особи V , взятые со знаком минус.

На этапе *селекции* производится отбор тех особей, значения функции приспособленности которых входят в множество $Eff(\{(-W_V, -Q_V): V \in \text{population}\})$.

По итогам селекции из N особей популяции отбираются H особей, которые в дальнейшем перейдут в следующую популяцию. Число H может быть различным для разных поколений.

Пусть этап отбора не прошли $(N - H)$ особей. Тем не менее в целях сохранения разнообразия они будут использованы на этапе скрещивания и мутации.

Считаем заданным число $k, k \in [0, 1]$, которое определяет, какую долю из $(N - H)$ особей займут в следующем поколении особи, полученные путем мутации. Остальные «места» займут особи, полученные с помощью скрещивания. Таким образом, $(N - H) \cdot k$ особей появятся в следующем поколении из-за мутации, а $(N - H) \cdot (1 - k)$ особей получатся с помощью скрещивания.

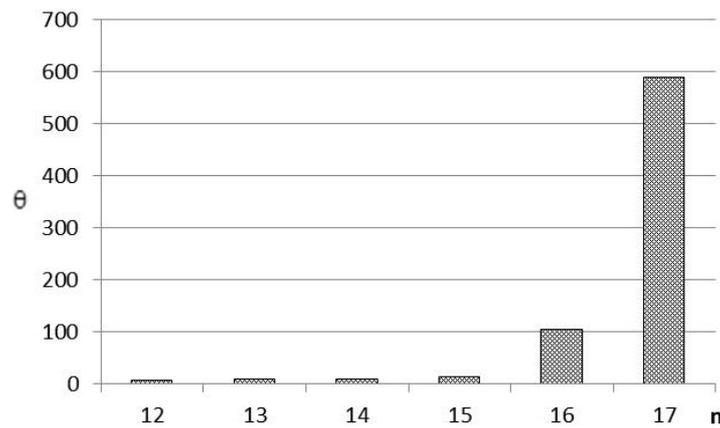
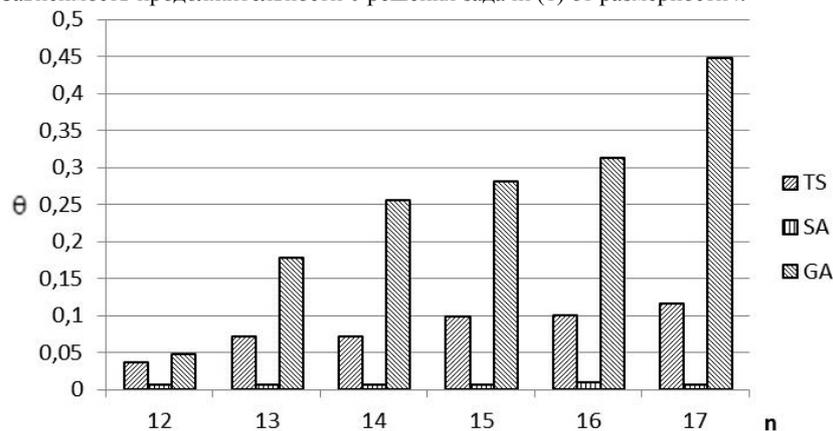
В генетических алгоритмах рассматривают, как правило, половое *размножение*. Это значит, что для получения одного потомка необходимо два родителя. Поэтому для получения $(N - H) \cdot (1 - k)$ потомков надо иметь $2(N - H) \cdot (1 - k)$ родителей. Первого родителя выбираем случайным образом из числа H особей, отобранных на этапе селекции. Вторым родителем может быть любая из N особей. Пусть $\mu = (\mu_1, \mu_2, \dots, \mu_{n-1})$, $\eta = (\eta_1, \eta_2, \dots, \eta_{n-1})$ – два родителя. Тогда в результате скрещивания получается особь $\rho = (\rho_1, \rho_2, \dots, \rho_{n-1})$, такая, что $\rho = \mu \mid \eta$.

Для проведения *мутации* выбирается $(N - H) \cdot k$ особь из числа неотобранных на этапе селекции; это делается для предотвращения застоя в одной из эффективных оценок или локальном минимуме. Для каждой такой особи U случайным образом выбирается особь $U' = NB(U, j)$. Если $W_{U'} < W_U$ или $Q_{U'} < Q_U$, то в новое поколение переходит особь U' . В противном случае в новое поколение переходит особь U .

Вычисление функции приспособленности для особей целесообразно проводить в момент конструирования новых особей.

Таким образом, в *новое поколение* переходят H особей, отобранных из предыдущего поколения $(N - H) \cdot (1 - k)$ особей, полученных на этапе скрещивания, и $(N - H) \cdot k$ мутировавших особей (или особей из предыдущего поколения, если мутация не дала увеличения приспособленности).

Критерием останова алгоритма может являться заданное число итераций или неизменность особей, выбранных на этапе селекции, на протяжении определенного числа итераций.

Рис. 1. Зависимость продолжительности θ решения задачи (1) от размерности n Рис. 2. Зависимость длительности θ обработки алгоритмов TS, SA, GA от размерности n

При выполнении критерия останова необходимо произвести отбор в последнем поколении. Отобранные особи будут соответствовать субоптимальным стратегиям обслуживания, а значения их функции приспособленности – значениям индивидуального штрафа, взятого со знаком минус.

3.4. Оценка точности субоптимальных решений. Для оценки отклонения Δ совокупностей субоптимальных решений от точной совокупности применялась формула

$$\Delta = \frac{\text{avg}_{(W', Q') \in M'} \left(\min_{(W, Q) \in M} \rho((W, Q); (W', Q')) \right)}{\text{avg}_{(W, Q) \in M} (\rho((W, Q); (0, 0)))}, \quad (11)$$

где M – точное множество эффективных оценок, M' – множество оценок, найденное метаэвристическим алгоритмом, avg – операция вычисления среднего значения, $\rho((W, Q); (W', Q'))$ – расстояние между двумя оценками, вычисляемое как $\max(|W - W'|, |Q - Q'|)$. Соответственно $\rho((W, Q); (0, 0))$ – расстояние между оценкой (W, Q) и оценкой $(0, 0)$.

4. Результаты вычислительных экспериментов

В процессе проведения серии вычислительных экспериментов установлено, что среднее

время θ обработки алгоритма построения полной совокупности Парето-оптимальных стратегий обслуживания для бикритериальной проблемы (1) составляет 0.1 микросекунды при $n = 12$ и возрастает до 10 минут при $n = 17$. Соответствующая диаграмма представлена на рис. 1.

Среднее время θ обработки алгоритмов TS, SA, GA для $n \leq 18$ не превосходит 0.45 секунды, и его зависимость от размерности n представлена на рис. 2.

Среднее значение отклонения Δ совокупностей субоптимальных решений от точных совокупностей составило менее 2% и в интегрированном виде проиллюстрировано на рис. 3.

Список литературы

1. Синий А.В., Федосенко Ю.С. Базовые математические модели снабжения топливом земснарядов в крупномасштабных районах русловой добычи нерудных строительных материалов // Международный научно-промышленный форум «Великие реки – 2004». Генеральные доклады, тезисы докладов. Н.Новгород.: ННГАСУ, 2004. С. 68–470.
2. Подиновский В.В., Ногин В.Д. Парето-оптимальные решения многокритериальных задач. М.: Наука, 1982. 255 с.
3. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М.: Наука, 1965. 457 с.

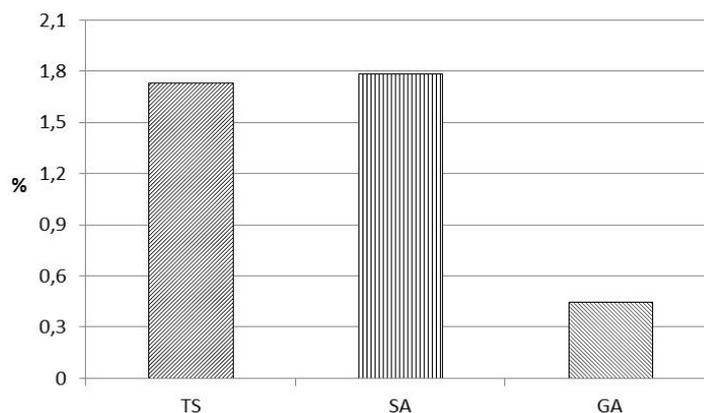


Рис. 3. Среднее значение отклонения Δ , соответствующего алгоритмам TS, SA, GA

4. Klamroth K., Wiecek M. Dynamic Programming Approaches to the Multiple Criteria Knapsack Problem // Technical Report #666. Dept. of Math. Sc., Clemson University, Clemson, SC, 1998.

5. Коган Д.И. Динамическое программирование и дискретная многокритериальная оптимизация. Н. Новгород: Изд-во Нижегородского госуниверситета, 2005. 260 с.

6. Glover F., Laguna M. Tabu Search // Modern Heuristic Techniques for Combinatorial Problems. 1993. P. 70–150.

7. Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by Simulated Annealing // Science. 1983. Vol. 220. No 4598. P. 671–680.

8. Holland J.H. Adaptation in Natural and Artificial Systems. The University of Michigan Press, 1975.

9. Коган Д.И., Федосенко Ю.С., Дуничкина Н.А. Задачи обслуживания линейно рассредоточенных стационарных объектов перемещающимися процессорами II // VI Московская международная конференция по исследованию операций (ORM 2010), Москва, 19–23 октября 2010 г.: Труды. М.: МАКС Пресс, 2010. С. 298–299.

BICRITERIA MODEL AND SYNTHESIS ALGORITHMS OF OPTIMAL COMPROMISE STRATEGIES FOR SINGLE PROCESSOR SERVICING OF A LINEAR GROUP OF STATIONARY OBJECTS

N.A. Dunichkina

A discrete model of single-stage servicing of a group of stationary objects is considered. The objects are located along a one-dimensional working zone of a mobile processor. Each object is associated with a pair of monotonically increasing penalty functions. Synthesis problems are formulated in the criteria plane of a full set of effective estimates and corresponding Pareto-optimal service strategies. Recurrence relations of dynamic programming allowing us to build the entire Pareto domain are derived. Special attention is paid to the synthesis algorithms of suboptimal servicing strategies.

Keywords: discrete model of servicing, Pareto-optimal strategies, dynamic programming, metaheuristic algorithms.