ЗАДАЧА ОБ ОПТИМАЛЬНОМ НАПАДЕНИИ В ЛИНЕАРИЗОВАННОЙ ПОСТАНОВКЕ

Ю. В. Шамардин

§ 1. Постановка задачи

Рассматриваемая задача представляет собой модель следующего конфликта. Сторона A атакует сторону B, поражая ее ценностные объекты средствами воздушного нападения. Сторона B препятствует этому, уничтожая атакующие средства с помощью некоторых перехватчиков. Требуется найти такой план нападения, при котором достигает максимума средняя суммарная ценность пораженных объектов с учетом наилучших действий стороны B. Эту ситуацию детализируем предположениями, которые аналогичны условиям из [1].

УСЛОВИЕ 1. Атака стороны A заключается в одновременном применении не более X однородных средств, каждое из которых способно поразить не более одного объекта.

УСЛОВИЕ 2. Сторона B своевременно обнаруживает все нападающие средства и знает их цели.

УСЛОВИЕ 3. Отражение атаки состоит в одновременном применении не более Y однородных перехватчиков, каждый из которых может поразить любое атакующее средство, но не более одного.

Введем обозначения:

I — количество ценностных объектов стороны B, которые занумеруем $1, 2, \ldots, I$; для краткости пишем объект i;

c_i — показатель ценности объекта i;

x_i — количество средств, атакующих объект i;

y_i — количество перехватчиков, прикрывающих объект i;

$P_i(x_i, y_i)$ — вероятность поражения объекта i (функция ущерба);

$x = (x_1, \ldots, x_I)$, $y = (y_1, \ldots, y_I)$ — векторы, описывающие варианты нападения и отражения атаки; они удовлетворяют ограничениям

$$\sum_{i=1}^{I} x_i \leqslant X, \quad x_i = 0, 1, 2, \ldots; \tag{1.1}$$

$$\sum_{i=1}^{I} y_i \leqslant Y, \quad y_i = 0, 1, 2, \ldots; \tag{1.2}$$

$F(x, y) = \sum\limits_{i=1}^{I} c_i P_i(x_i, y_i)$ — средняя суммарная ценность пораженных объектов;

$G(x) = \min_y F(x, y)$ — аналогичная величина при наилучших действиях стороны B против атаки x.

В принятых обозначениях задачу можно сформулировать следующим образом: найти вариант атаки x^* такой, что

$$G(x^*) = \max_x G(x) = \max_x \min_y F(x, y). \tag{1.3}$$

Экстремумы берутся при ограничениях (1.1), (1.2).

Задачи вида (1.3) рассматривались, например, в [1–3], причем в работах [2, 3] без учета целочисленности переменных x_i, y_i. Как в непрерывной, так и в дискретной постановке, задача (1.3) оказывается трудной проблемой. Расчет одного варианта относительно небольших размерностей $I = 15$, $X = 60$, $Y = 54$ методом ветвей и границ [1] занимает до двух минут машинного времени, что может быть неприемлемым в прикладных исследованиях. Изучение же непрерывных постановок сводится к формулировке необходимых условий оптимальности, эффективность численной реализации которых не очевидна.

В данной работе в постановку задачи (1.3) вносятся упрощения с тем, чтобы облегчить ее решение, оставаясь по возможности в рамках той содержательной интерпретации, которая описана выше. Введем дополнительные предположения.

УСЛОВИЕ 4. На каждое атакующее средство направляется не более заданного числа J перехватчиков, и все средства, атакующие объект i, обстреливаются одинаково. Таким образом, величина y_i принимает значения только из множества $\{0, x_i, 2x_i, \dots, Jx_i\}$.

УСЛОВИЕ 5. Функции $\varphi_{ij}(x_i) = c_i P_i(x_i, jx_i)$ кусочно-линейные:

$$\varphi_{ij}(x_i) = c_i \min\{1, (1-q)^j p_i x_i\}; \tag{1.4}$$

здесь q — вероятность поражения атакующего средства одним перехватчиком, p_i — средний ущерб, наносимый объекту i одним средством нападения ($0 < q$, $p_i \leqslant 1$; $(1-q)^0 = 1$ при всех q).

Введем переменные z_{ij} так, что $z_{ij} = 1$, если на каждое средство, атакующее объект i, направляется j перехватчиков, и $z_{ij} = 0$ в противном случае. Сделаем замену

$$y_i = \sum_{j=0}^{J} j x_i z_{ij}.$$

УСЛОВИЕ 6. Суммарная ценность $F(x, z)$ пораженных объектов при нападении x и распределении $z = (z_{ij})$ средств обороны имеет вид

$$F(x, z) = \sum_{i=1}^{I} \sum_{j=0}^{J} \varphi_{ij}(x_i) z_{ij};$$

аналогичная величина при наилучших действиях обороны определяется равенством

$$G(x) = \min_z F(x, z), \tag{1.5}$$

где минимум берется при ограничениях

$$\sum_{i=1}^{I}\sum_{j=0}^{J} jx_i z_{ij} \leqslant Y, \tag{1.6}$$

$$\sum_{j=0}^{J} z_{ij} = 1 \quad (i = 1,\dots,I); \qquad z_{ij} \geqslant 0 \quad (i = 1,\dots,I;\ j = 1,\dots,J) \tag{1.7}$$

без учета целочисленности переменных z_{ij}.

Далее рассматривается задача поиска такого плана атаки x^*, который удовлетворяет ограничениям (1.1) и доставляет максимум функции (1.5):

$$G^* = G(x^*) = \max_x G(x). \tag{1.8}$$

§ 2. Преобразование задачи

Для задачи линейного программирования (1.5)–(1.7) очевидно выполняется соотношение двойственности [4]. Поставим в соответствие неравенству (1.6) переменную λ и введем обозначения:

$$g_i(x_i, \lambda) = \min\{\varphi_{ij}(x_i) + \lambda j x_i \mid j = 0,\dots,J\}, \tag{2.1}$$

$$G(x,\lambda) = \sum_{i=1}^{I} g_i(x_i,\lambda). \tag{2.2}$$

Переходя от задачи (1.5)–(1.7) к двойственной задаче, получим

$$G(x) = \max\{G(x,\lambda) - \lambda Y \mid \lambda \geqslant 0\}.$$

Подставляя последнее выражение в (1.8) и меняя местами экстремумы, приходим к задачам

$$h(\lambda) = \max_x G(x,\lambda), \tag{2.3}$$

$$G^* = G(x^*) = \max\{h(\lambda) - \lambda Y \mid \lambda \geqslant 0\}. \tag{2.4}$$

Таким образом, решение исходной задачи (1.8) распадается на два этапа, причем первый этап включает два шага.

Этап I. Поиск величины λ^*, при которой достигается максимум (2.4).

Шаг 1. Строится конечное множество L точек λ, на котором достигается максимум (2.4).

Шаг 2. С помощью мажоранты $H(\lambda)$ функции $h(\lambda)$ выделяется подмножество $L_0 \subset L$, содержащее λ^*, и алгоритм заключается в переборе точек $\lambda \in L_0$.

Этап II. Решение задачи (2.3) при $\lambda = \lambda^*$.

Подробное изложение решения задачи (1.8) начнем с этапа II. Через $\lfloor a \rfloor$ и $\lceil a \rceil$ обозначим округления числа a снизу и сверху:

$$\lfloor a \rfloor = \max\{n \mid n \leqslant a, \quad n \text{ целое}\},$$
$$\lceil a \rceil = \min\{n \mid n \geqslant a, \quad n \text{ целое}\}.$$

Преобразуя (2.1) с учетом условия (1.4) и обозначения

$$\psi_i(\lambda) = \min\{c_i p_i (1-q)^j + \lambda j \mid j = 0, \dots, J\}, \tag{2.5}$$

получаем

$$g_i(x_i, \lambda) = \min\{c_i,\ \psi_i(\lambda) x_i\}. \tag{2.6}$$

Очевидно, что функция (2.6) вогнута по переменной x_i и, как следует из результатов [5], задача (2.3) решается алгоритмом покоординатного спуска (алгоритм ПС). Опишем этот алгоритм, интерпретируя его как последовательное назначение атакующих средств на объекты — цели и учитывая специфику выражения (2.6).

Ниже используем обозначения:

$x = (x_1, \dots, x_I)$ — целераспределение, сложившееся к очередному шагу алгоритма,

X' — остаток не распределенных средств,

$a := b$ — операция замены значения a на b.

АЛГОРИТМ ПОКООРДИНАТНОГО СПУСКА

Шаг 1. Полагаем $x_i := 0$, $i = 1, \dots, I$, $X' := X$.

Шаг 2. Если $X' = 0$, то вычисления прекращаются. Если $X' > 0$, то вычисляем

$$\Delta =\colon \max\{g_i(x_i + 1, \lambda) - g_i(x_i, \lambda) \mid i = 1, \dots, I\}. \tag{2.7}$$

Если $\Delta = 0$, то алгоритм заканчивает работу. Пусть $\Delta > 0$ и максимум в (2.7) достигается на номере s (если таких номеров несколько, берем минимальный). Тогда переходим к шагу 3.

Шаг 3. Вычисляем размер шага d по соотношениям

$$d = \min\{X', n\}, \qquad n = \begin{cases} \lfloor c_s/\psi_s(\lambda) \rfloor, & \text{если } x_s = 0, \\ 1, & \text{если } x_s > 0, \end{cases}$$

полагаем $x_s := x_s + d$, $X' := X' - d$ и возвращаемся на шаг 2.

Заметим, что $n \geqslant 1$ на шаге 3. Действительно, в силу (2.5)

$$\psi_s(\lambda) \leqslant c_s p_s, \quad c_s/\psi_s(\lambda) \geqslant 1/p_s \geqslant 1.$$

Нетрудно видеть, что алгоритм ПС находит оптимальное решение не более чем за $2I$ итераций, каждая из которых включает шаги 2, 3.

Далее будем считать, что нумерация объектов удовлетворяет условию

$$c_1 p_1 \geqslant c_2 p_2 \geqslant \ldots \geqslant c_I p_I. \tag{2.8}$$

Введем обозначение

$$S_n(\lambda) = \sum_{i=1}^{n} \lfloor c_i/\psi_i(\lambda) \rfloor, \tag{2.9}$$

и пусть $k = k(\lambda)$ — номер, для которого выполняются неравенства

$$S_{k-1}(\lambda) < X \leqslant S_k(\lambda).$$

Если $X \leqslant S_1(\lambda)$ или $X > S_I(\lambda)$, то полагаем $k(\lambda) = 1$ или $k(\lambda) = I + 1$. Следующая лемма позволит уменьшать размер задачи путем исключения «лишних» объектов.

Лемма 2.1. 1. *Если* $k(\lambda) < I$ *при данном* λ, *то найдется оптимальное решение* $x(\lambda) = (x_1(\lambda), \dots, x_I(\lambda))$ *задачи* (2.3) *такое, что* $x_i(\lambda) = 0$, $i = k(\lambda)+1, \dots, I$.

2. *Если* $\lambda' \leqslant \lambda''$, *то* $k(\lambda') \leqslant k(\lambda'')$, *и если* $k(\lambda'') < I$, *то решения* $x(\lambda')$, $x(\lambda'')$ *удовлетворяют соотношениям* $x_i(\lambda') = x_i(\lambda'') = 0$, $i = k(\lambda'')+1, \dots, I$.

Доказательство. 1. Покажем, что решение $x = (x_1, \dots, x_I)$, полученное с помощью алгоритма ПС, обладает требуемым свойством. Введем обозначения

$$n = \max\{i \mid x_i > 0\}, \quad a_i = \lfloor c_i/\psi_i(\lambda) \rfloor,$$
$$\Delta_i = g_i(1,\lambda) - g_i(0,\lambda) = g_i(1,\lambda) = \min\{c_i,\ \psi_i(\lambda)\}$$

и заметим, что $\Delta_i = \psi_i(\lambda)$, $\Delta_i \geqslant \Delta_{i+1}$ в силу (2.5), (2.8). Согласно описанию алгоритма ПС решение x имеет вид

$$x_i = a_i + \delta_i, \quad i = 1, \dots, n-1,$$
$$x_n > 0,$$
$$x_i = 0, \quad i = n+1, \dots, I,$$

где $\delta_i = 0, 1$. Допустим, что $n > k(\lambda)$. Тогда цепочка неравенств

$$X \geqslant \sum_{i=1}^{n} x_i \geqslant \sum_{i=1}^{k(\lambda)} a_i + x_n > \sum_{i=1}^{k(\lambda)} a_i$$

приводит к противоречию с определением $k(\lambda)$. Следовательно, $n \leqslant k(\lambda)$, что и доказывает утверждение 1 леммы 2.1.

2. Покажем, что $k(\lambda') \leqslant k(\lambda'')$ и далее воспользуемся утверждением 1. Учитывая (2.5) и очевидное неравенство

$$c_i p_i (1-q)^j + j\lambda' \leqslant c_i p_i (1-q)^j + j\lambda'',$$

получаем $\psi_i(\lambda') \leqslant \psi_i(\lambda'')$. В силу (2.9) имеем $S_r(\lambda') \geqslant S_r(\lambda'')$, $r = 1, \dots, I$. Обозначим для краткости $n = k(\lambda')$, $m = k(\lambda'')$. Допустим, что $n > m$. Тогда цепочка неравенств

$$X > S_{n-1}(\lambda') \geqslant S_{n-1}(\lambda'') \geqslant S_m(\lambda'') \geqslant X$$

приводит к противоречию. Следовательно, $n \leqslant m$. Лемма 2.1 доказана.

Перейдем к более подробному изучению задачи (2.4). Начнем с расчета участков линейности функции $g_i(x_i, \lambda)$ по переменной λ. Выражение (2.5) перепишем в виде

$$\psi_i(\lambda) = c_i p_i \psi(\lambda/(c_i p_i)), \tag{2.10}$$

$$\psi(\mu) = \min\{(1-q)^j + \mu j \mid j = 0, \dots, J\}, \quad \mu \geqslant 0. \tag{2.11}$$

Ввиду выпуклости функции $(1-q)^j$ по переменной j нетрудно получить представление

$$\psi(\mu) = \begin{cases} 1, & \text{если } \mu_1 \leqslant \mu, \\ (1-q)^j + \mu j, & \text{если } \mu_{j+1} \leqslant \mu \leqslant \mu_j, \quad j = 1, \dots, J-1, \\ (1-q)^J + \mu J, & \text{если } 0 \leqslant \mu \leqslant \mu_J, \end{cases}$$

$$\mu_j = q(1-q)^{j-1}, \quad j = 1, \dots, J.$$

Обозначим через L числовое множество, состоящее из нуля, произведений вида $c_i p_i \mu_j$ и величин λ, удовлетворяющих условиям

$$c_i = \psi_i(\lambda) n, \quad \lambda \leqslant c_i p_i q \tag{2.12}$$

при некоторых $i = 1, \dots, I$ и $n = 1, \dots, X$.

Лемма 2.2. 1. *Множество L конечно, и его мощность $|L|$ оценивается неравенством*

$$|L| \leqslant \begin{cases} I(X+J)+1, & \text{если } q<1, \\ I(X+1)+1, & \text{если } q=1. \end{cases}$$

2. *Функция $h(\lambda)$ из (2.3) не убывает при $\lambda \geqslant 0$ и является выпуклой на любом отрезке $[\lambda', \lambda'']$, $\lambda' \geqslant 0$, не содержащем внутри себя точек из L.*

Доказательство. 1. Различных произведений $\lambda = c_i p_i \mu_j$ не более IJ при $q < 1$ и не более I при $q = 1$. Поскольку функция $\psi_i(\lambda)$ строго возрастает при $\lambda \leqslant c_i p_i q$, что следует из представлений (2.10), (2.11), существует не более одной точки λ, удовлетворяющей соотношениям (2.12) при фиксированных значениях i, n. Следовательно, точек λ, удовлетворяющих (2.12), не более IX. Утверждение 1 леммы 2.2 доказано.

2. Сначала покажем, что функция $g_i(x_i, \lambda)$ представима в виде

$$g_i(x_i, \lambda) = a_i(x_i) + b_i(x_i)\lambda \tag{2.13}$$

при всех $\lambda \in [\lambda', \lambda'']$ и $x_i = 0, \dots, X$. Из условия леммы и равенства (2.6) следует, что справедливы следующие утверждения:

— при фиксированном x_i и любом $\lambda \in [\lambda', \lambda'']$ выполняется $g_i(x_i, \lambda) = c_i$ либо $g_i(x_i, \lambda) = \psi_i(\lambda)x_i$;

— $\psi_i(\lambda) = c_i p_i (1-q)^j + \lambda j$ при всех $\lambda \in [\lambda', \lambda'']$ и некотором j.

Но это и означает справедливость (2.13). Учитывая (2.2), (2.3), получаем

$$h(\lambda) = \max_x \sum_{i=1}^{I} (a_i(x_i) + b_i(x_i)\lambda), \quad \lambda \in [\lambda', \lambda''],$$

откуда следует выпуклость функции $h(\lambda)$. Монотонность вытекает из монотонности функций $\psi_i(\lambda)$ и соотношения (2.6). Лемма 2.2 доказана.

Теорема 2.1. 1. *Максимум в (2.4) достигается на множестве L, т. е.*

$$G^* = \max\{h(\lambda) - \lambda Y \mid \lambda \in L\}.$$

2. *Если $k = k(c_1 p_1 q) < I$, то в исходной задаче (1.8) можно считать $x_i = 0$, $i = k+1, \dots, I$, т. е. исключить объекты с номерами $i > k$.*

Доказательство. 1. Ввиду определения множества L и условия (2.8) максимальный элемент в L равен $c_1 p_1 q$. При $\lambda \geqslant c_1 p_1 q$ и любом i имеем $\psi_i(\lambda) = \psi_i(c_1 p_1 q) = c_i p_i$. Следовательно, $h(\lambda) - \lambda Y = h(c_1 p_1 q) - \lambda Y$, и максимум в (2.4) достигается при $\lambda \leqslant c_1 p_1 q$.

Пусть λ', λ'' — соседние точки из L. В силу леммы 2.2 функция $h(\lambda) - \lambda Y$ выпукла на отрезке $[\lambda', \lambda'']$, а значит, ее максимум на этом отрезке достигается на одном из его концов, что и доказывает утверждение 1 теоремы 2.1.

2. Утверждение 2 следует из леммы 2.1 и того факта, что в силу утверждения 1 теоремы найдется оптимальное значение $\lambda^* \leqslant c_1 p_1 q$. Теорема 2.1 доказана.

Благодаря теореме 2.1 задача (2.4) сводится к перебору точек множества L. Объем перебора можно сократить с помощью мажоранты $H(\lambda) \geqslant$

$h(\lambda)$, к построению которой и переходим. Вернемся к представлению (2.1) функции $g_i(x_i, \lambda)$ и обозначим

$$h_j = \max_x \sum_{i=1}^{I} \varphi_{ij}(x_i), \quad j = 0, \dots, J. \tag{2.14}$$

Эти константы вычисляются алгоритмом ПС, в описании которого следует заменить величины $\psi_i(\lambda)$ на $c_i p_i (1-q)^j$. Учитывая неравенство $g_i(x_i, \lambda) \leqslant \varphi_{ij}(x_i) + j\lambda x_i$ и обозначение (2.14), получаем

$$h(\lambda) \leqslant \min\{h_j + j\lambda X \mid j = 0, \dots, J\} = H(\lambda).$$

Введенная здесь функция $H(\lambda)$ является вогнутой. Мы будем использовать ее как мажоранту функции $h(\lambda)$.

Пусть $\mu \geqslant 0$. Через $[a(\mu), b(\mu)]$ обозначим отрезок, состоящий из точек λ таких, что $H(\lambda) - \lambda Y \geqslant h(\mu) - \mu Y$, $0 \leqslant \lambda \leqslant c_1 p_1 q$, и через λ_0 — оптимальное решение вспомогательной задачи

$$H(\lambda_0) - \lambda_0 Y = \max\{H(\lambda) - \lambda Y \mid 0 \leqslant \lambda \leqslant c_1 p_1 q\}. \tag{2.15}$$

§ 3. Алгоритм решения задачи (2.4) и оценка его трудоемкости в частных случаях

Предлагаемый алгоритм заключается в переборе точек $\lambda_1, \lambda_2, \lambda_3, \dots$ из множества L, занумерованных так, что

$$\lambda_0 \geqslant \lambda_1 > \lambda_3 > \dots > \lambda_{2s-1} > \dots,$$
$$\lambda_0 \leqslant \lambda_2 < \lambda_4 < \dots < \lambda_{2s} < \dots.$$

Перебор точек λ_n с нечетными номерами назовем *движением влево*, а с четными номерами — *движением вправо*. Через l и r обозначим вспомогательные величины, равные единице, если можно продолжать движение влево и вправо, и равные нулю, если движение в соответствующую сторону закончено.

Через λ^* обозначим оптимальное решение задачи (2.4), получаемое в результате работы алгоритма, через $G^* = h(\lambda^*) - \lambda^* Y$ — значение экстремума, через $[A, B]$ — отрезок с концами $A = a(\lambda^*)$, $B = b(\lambda^*)$.

АЛГОРИТМ

Шаг 1. Решая оценочную задачу (2.15), находим λ_0 и полагаем $\lambda^* := \lambda_0$, $G^* := h(\lambda_0) - \lambda_0 Y$, $A := a(\lambda_0)$, $B := b(\lambda_0)$, $l := 1$, $r := 1$. Очевидно, что $\lambda_0 \in [A, B]$.

Шаг 2. Если $l = 1$, то вычисляем очередную точку λ_n с нечетным номером. Если $\lambda_n \leqslant A$, то полагаем $l := 0$ и переходим к следующему шагу. Если $\lambda_n > A$, то вычисляем значение $G = h(\lambda_n) - \lambda_n Y$ и в случае $G > G^*$ меняем рекорд:

$$\lambda^* := \lambda_n, \quad G^* := G, \quad A := a(\lambda_n), \quad B := b(\lambda_n). \tag{3.1}$$

Шаг 3. Если $r = 1$, то вычисляем очередную точку λ_n с четным номером. Если $\lambda_n \geqslant B$, то полагаем $r := 0$ и переходим к следующему шагу. Если $\lambda_n < B$, то вычисляем значение $G = h(\lambda_n) - \lambda_n Y$ и при $G > G^*$ меняем рекорд по формулам (3.1).

Шаг 4. Если $l = r = 0$, то работа алгоритма заканчивается. В противном случае вычисления продолжаются с шага 2.

Количество вычислений величин $h(\lambda_n)$ в ходе работы алгоритма назовем *трудоемкостью* и обозначим через T. В общем случае можно предложить только очевидную оценку $T \leqslant |L|+1$. При дополнительных предположениях можно сделать содержательные выводы. Обозначим

$$P = \sum_{i=1}^{I} \lceil 1/p_i \rceil, \quad C = \sum_{i=1}^{I} c_i.$$

Теорема 3.1. *Пусть $q = 1$, $X > Y > 0$, $X \geqslant P$. Тогда $T \leqslant P + 2$. В частности, если $p_i = 1$ для всех i, то $T \leqslant I + 2$.*

Доказательство. При $q = 1$ выражение (2.5) упрощается:

$$\psi_i(\lambda) = \min\{c_i p_i, \lambda\}. \tag{3.2}$$

Поэтому можно исследовать задачи (2.3) и (2.4) более детально. Введем обозначения

$$d_i(\lambda) = \begin{cases} \lfloor c_i/\lambda \rfloor, & \text{если } \lambda \leqslant c_i p_i, \\ 0, & \text{если } \lambda > c_i p_i, \end{cases} \qquad D(\lambda) = \sum_{i=1}^{I} d_i(\lambda).$$

Нам потребуется следующая

Лемма 3.1. 1. *Если $D(\lambda) \geqslant X$, то $h(\lambda) = \lambda X$.*
2. *Если $D(\lambda) + P \leqslant X$, то $h(\lambda) = C$.*

Доказательство. 1. Поскольку $D(\lambda) \geqslant X$, найдется номер $n \geqslant 1$ такой, что $c_n p_n \geqslant \lambda > c_{n+1} p_{n+1}$ (в частности, $n = I$). Ввиду (3.2) задача (2.3) принимает вид

$$\sum_{i=1}^{n} \min\{c_i, \lambda x_i\} + \sum_{i=n+1}^{I} \min\{c_i, c_i p_i x_i\} \longrightarrow \max_x,$$

где максимум берется при ограничениях (1.1). Применение к этой задаче алгоритма ПС дает следующее оптимальное решение:

$$x_i = \lfloor c_i/\lambda \rfloor, \quad i = 1, \dots, m-1,$$
$$x_m = X - \sum_{i=1}^{m-1} x_i \leqslant \lfloor c_m/\lambda \rfloor,$$
$$x_i = 0, \quad i = m+1, \dots, I,$$

где $m \leqslant n$. Как нетрудно показать,

$$h(\lambda) = \sum_{i=1}^{m} \min\{c_i, \lambda x_i\} = \lambda \sum_{i=1}^{m} x_i = \lambda X.$$

Утверждение 1 леммы 3.1 доказано.

2. Поскольку $g_i(x_i, \lambda) \leqslant c_i$ при всех x_i, λ, имеем

$$h(\lambda) = \max_x \sum_{i=1}^{I} g_i(x_i, \lambda) \leqslant \sum_{i=1}^{I} c_i = C. \tag{3.3}$$

Но если выполняется неравенство $D(\lambda)+P \leqslant X$, то вектор x с компонентами $x_i = d_i(\lambda) + \lceil 1/p_i \rceil$, $i = 1,\dots,I$, является допустимым в задаче (1.1), (2.3). Для этого вектора имеем

$$h(\lambda) \geqslant \sum_{i=1}^{I} g_i(x_i,\lambda) = \sum_{i=1}^{I} c_i = C. \tag{3.4}$$

Сравнение (3.3) и (3.4) завершает доказательство леммы 3.1.

Продолжим доказательство теоремы 3.1. При $q = 1$ мажоранта $H(\lambda)$ задается формулой

$$H(\lambda) = \min\{\lambda X, h_0\},$$

где $h_0 = C$, а положительные точки множества L имеют вид

$$\lambda = \min\{c_i p_i, c_i/n\}, \quad i = 1,\dots,I, \quad n = 1,\dots,X.$$

Определим величины

$$u = \max\{\lambda \mid D(\lambda) \geqslant X, \quad \lambda \in L\}, \tag{3.5}$$

$$v = \begin{cases} c_1 p_1, & \text{если } D(c_1p_1) + P \geqslant X + 1, \\ \min\{\lambda \in L \mid D(\lambda) + P \leqslant X\}, & \text{если } D(c_1p_1) + P \leqslant X. \end{cases} \tag{3.6}$$

Нетрудно видеть, что функция $D(\lambda)$ не возрастает по λ. Поэтому в силу леммы 3.1 справедливы неравенство $u \leqslant v$ и соотношения

$$h(\lambda) - \lambda Y = H(\lambda) - \lambda Y = \lambda(X - Y), \quad \text{если } \lambda \leqslant u, \tag{3.7}$$

$$h(\lambda) - \lambda Y = H(\lambda) - \lambda Y = C - \lambda Y, \quad \text{если } \lambda \geqslant v. \tag{3.8}$$

Рассмотрим теперь алгоритм решения задачи (2.4). Ввиду неравенства $X > Y$ для начальной точки λ_0 (оптимум оценочной задачи (2.15)) имеем $\lambda_0 = C/X$ и $\lambda_0 \in [u,v]$ в силу (3.7), (3.8). Следовательно, алгоритм начинается перебором точек $\lambda \in L$ таких, что $u \leqslant \lambda \leqslant v$. Допустим, что при движении влево алгоритм доходит до точки u. Если оказывается, что $u \leqslant A$ (см. описание алгоритма), то движение влево заканчивается на этом шаге. Если $u > A$, то, как следует из (3.7), происходит смена рекорда и левая граница A становится равной u. Но тогда при просмотре соседней слева от u точки $u' \in L$ будет выполняться неравенство $u' \leqslant A$, и движение влево закончится. Аналогично показывается, что при движении вправо алгоритм не может продолжаться далее точки v.

Если $u = v$, то $T = 1$, и утверждение теоремы верно. Пусть $u < v$. Для точек $\lambda \in L$, лежащих между u и v, введем обозначение

$$u = \nu_1 < \nu_2 < \ldots < \nu_n = v$$

и оценим их количество n. Заметим, что в силу (3.5), (3.6) справедливы неравенства

$$D(\nu_2) \leqslant X - 1, \quad D(\nu_{n-1}) \geqslant X - P + 1.$$

Из определения функции $D(\lambda)$ и ее монотонности получаем

$$D(\nu_m) - D(\nu_{m+1}) \geqslant 1, \quad m = 1, \ldots, n-1.$$

Число n оценим цепочкой неравенств

$$n \leqslant 3 + \sum_{m=2}^{n-2} (D(\nu_m) - D(\nu_{m+1})) = 3 + D(\nu_2) - D(\nu_{n-1})$$

$$\leqslant 3 + (X-1) - (X - P + 1) = P + 1.$$

Учитывая точку λ_0, заключаем $T \leqslant n+1 \leqslant P+2$. Теорема 3.1 доказана.

§ 4. Вычислительный эксперимент. Приближенный алгоритм

Выяснение трудоемкости T алгоритма в общем случае проводилось путем численных расчетов. При фиксированном количестве объектов I решалась серия из $N = 100$ случайным образом выбранных вариантов задач. Вычисления велись на машине IBM PC 386SX; программа составлена на языке Pascal.

Каждый вариант исходных данных формируется следующим образом.

— С помощью равномерного распределения на отрезке $[0.01, 1]$ выбираются вспомогательные величины a_i и затем по формуле

$$c_i = 100 a_i / \sum_{m=1}^{I} a_m, \quad i = 1, \ldots, I,$$

рассчитываются показатели ценности объектов. Их сумма равна 100.

— Вероятности q и p_i для всех i выбираются независимо равномерным распределением на отрезке $[0.1, 1]$. Величина J рассчитывается по формуле

$$J = \begin{cases} 10, & \text{если } q \geqslant 0.5, \\ 20, & \text{если } q < 0.5. \end{cases}$$

— Максимальный объем атаки X выбирается равновероятно из целых чисел отрезка $[P, 10P]$, где $P = \sum_{i=1}^{I} \lfloor 1/p_i \rfloor$.

Условие $X \geqslant P$ гарантирует от заведомо «лишних» объектов (см. теорему 2.1). Количество перехватчиков Y определяется равенством $Y = \lfloor 0.9X/q \rfloor$.

Введем обозначения:

T_n и t_n — трудоемкость (в смысле, определенном выше) и время (в секундах) решения серийной задачи с номером $n = 1, \ldots, N$,

$T_{\text{ср}} = \sum_{n=1}^{N} T_n/N$ и $t_{\text{ср}} = \sum_{n=1}^{N} t_n/N$ — средние значения трудоемкости и времени решения серийной задачи,

$T_{\max} = \max\{T_n \mid n = 1, \ldots, N\}$ и $t_{\max} = \max\{t_n \mid n = 1, \ldots, N\}$ — аналогичные максимальные значения в серии.

В табл. 1 приведены величины $T_{ср}$, $T_{\max}$, $t_{ср}$, $t_{\max}$ в зависимости от количества объектов I.

Таблица 1

I	10	20	30	40	50	60	70
$T_{ср}$	59	113	109	127	147	142	159
$T_{\max}$	316	426	339	579	949	944	971
$t_{ср}$	0.4	1.4	2.3	3.4	5.0	5.3	6.7
$t_{\max}$	1.2	3.1	6.5	10.2	16.7	18.2	22.4

В ходе эксперимента наряду с точным рассчитывалось приближенное решение задачи (2.4) по следующему алгоритму.

Шаг 1. Вычисляем оптимальную точку λ_0 задачи (2.15) и находим отрезок $[a(\lambda_0), b(\lambda_0)]$, который содержит оптимальные решения задачи (2.4) (см. описание точного алгоритма).

Шаг 2. Наилучшую точку среди λ_0 и $\nu_m = a(\lambda_0)+(b(\lambda_0)-a(\lambda_0))m/M$, $m = 0,\dots,M-1$, принимаем за приближенное решение задачи (2.4) (здесь M — фиксированное число).

Обозначим через G_n^*, G_n точное и приближенное значения максимума в задаче (2.4) с номером n, через

$$\delta = \max\{100(G_n^* - G_n)/G_n^* \mid n = 1,\dots,N\}$$

максимум относительного отклонения (в процентах) величины G_n от G_n^* в серии задач. В табл. 2 приведены значения δ в зависимости от количества объектов I и мощности сетки M.

Таблица 2

M	I						
	10	20	30	40	50	60	70
5	5.3	4.3	3.2	2.0	1.4	1.8	1.7
10	1.2	1.5	1.5	0.6	0.6	0.8	1.2
20	0.6	1.0	0.5	0.3	0.6	0.4	0.8
30	0.6	0.7	0.2	0.2	0.4	0.2	0.1

Таким образом, при $M = 20, 30$ приближенный алгоритм дает относительное отклонение δ не более одного процента, что вполне достаточно для приложений. Время решения одного варианта задачи приближенным алгоритмом примерно в $5 \div 10$ раз меньше времени работы точного алгоритма.

ЛИТЕРАТУРА

1. **O'Meara N. T., Soland R. M.** Optimal strategies for problems of simultaneous attack against an area defense with impact-point prediction // Naval Res. Logist. 1992. V. 39, N 1. P. 1–28.
2. **Битюцких В. Т.** Максиминная задача распределения средств нападения и обороны // Техническая кибернетика. 1991. № 1. С. 69–74.
3. **Данскин Дж.** Теория максимина и ее приложения к задачам распределения вооружений. М.: Сов. радио, 1970.
4. **Гольштейн Е. Г.** Выпуклое программирование. Элементы теории. М.: Наука, 1970.
5. **Глебов Н. И.** Об одном классе задач выпуклого целочисленного программирования // Управляемые системы. Новосибирск: Ин-т математики СО АН СССР. 1973. Вып. 11. С. 38–42.