

# **АЛГОРИТМ ФОРМИРОВАНИЯ МНОЖЕСТВА ПАРЕТО-ОПТИМАЛЬНЫХ ВАРИАНТОВ ЗНАЧЕНИЙ ПАРАМЕТРОВ ПРОВЕДЕНИЯ ПРОВЕРОК ПРОТИВОПОЖАРНОГО СОСТОЯНИЯ ОБЪЕКТОВ ОРГАНАМИ ГОСУДАРСТВЕННОГО ПОЖАРНОГО НАДЗОРА**

**О.А. Рыбин, доктор технических наук;**

**В.В. Попов, кандидат военных наук, доцент;**

**А.Н. Нестругин.**

**Санкт-Петербургский университет ГПС МЧС России**

Представлено описание алгоритма формирования множества Парето-оптимальных вариантов значений параметров проверок противопожарного состояния надзорных объектов.

*Ключевые слова:* противопожарное состояние, обеспечение безопасности, надзорная деятельность, Парето-оптимальные решения

## **THE ALGORITHM OF FORMATION OF THE SET OF PARETO-OPTIMAL OPTIONS PARAMETER VALUES SPEND OF INSPECTIONS OF FIRE-PREVENTION CONDITION OF OBJECTS BODIES OF THE STATE FIRE SUPERVISION**

O.A. Rybin; V.V. Popov; A.N. Nestrugin.

Saint-Petersburg university of State fire service of EMERCOM of Russia

The article presents the description of algorithm of formation of the set of Pareto-optimal variant of values of parameters of the reviews of fire state Supervisory objects.

*Keywords:* fire fighting condition, security, supervision, Pareto-optimal solutions

Статистика показывает, что, несмотря на общее снижение количества пожаров, величина материального ущерба растёт. Объяснить такую обратную зависимость несложно, можно сделать вывод о том, что чрезвычайные ситуации, связанные с пожарами, случаются «по-крупному», а общую тенденцию, связанную с уменьшением количества пожаров, можно частично связать с недостоверной статистикой либо с повышающейся сознательностью граждан в области пожарной безопасности.

Тем не менее, последние примеры крупных пожаров показывают, что большинство пожаров можно было предупредить, в том числе и проведением целевых проверок. Естественно, нельзя спрогнозировать перечень конкретных объектов, которые имеют большую вероятность возникновения пожара, однако, имея статистику по пожарам и сезонности этих происшествий, можно определить типы объектов надзора и пожароопасные для них периоды, а затем, не учитывая удельную стоимость этих объектов, а также возможный ущерб от пожара, можно сформировать план проверок с учётом ограниченных ресурсов органов государственного пожарного надзора, а также ограничениями по проведению проверок.

Анализ показывает, что задачи подобного рода, а именно выбора значений параметров проведения проверок противопожарного состояния объектов, относятся к классу задач дискретного программирования.

Как правило, задачи комбинаторного программирования сводятся к выбору на конечном множестве альтернатив. Поэтому в принципе эти задачи можно решить методом прямого перебора. Для рассматриваемой задачи без потери общности можно в общем виде считать, что  $B = \{b_1, \dots, b_j, \dots, b_N\}$ ,  $\Lambda_{\{L\}} = \{1, \dots, l, \dots, L\}$ . Совокупность всех последовательностей обозначим через  $X$ . Отображение  $X$  можно тогда отождествить с последовательностью  $\langle x(1), \dots, x(N) \rangle$ , каждый элемент которой принимает значения из множества  $A$ .

Число всех последовательностей  $\{b_1, \dots, b_j, \dots, b_N\}$  с членами из  $L$ -элементного множества  $A$  равно  $card(X) = L^K$ . Алгоритм генерирования подобных комбинаций, обычно называемый алгоритмом генерирования размещений с повторениями, заключается в последовательном увеличении индекса проверяемого объекта  $n$ , который подвергается проверке в начальный период ( $l = 1$ ). При достижении значения  $j = N$  индексу присваивается начальное значение  $n=1$  и осуществляется аналогичное увеличение индекса  $l$ .

Решение оптимизационных задач подобными алгоритмами полного перебора возможных вариантов затруднено в силу вычислительной сложности. По своей трудоёмкости алгоритмы генерирования размещений с повторениями относятся к алгоритмам суперэкспоненциальной трудоёмкости [1], поскольку количество вычислительных операций превышает порог Бремермана-Эшби ( $10^{100}$ ) при достаточно малых  $N$  и  $L$ . Для такой задачи полный перебор может стать неприемлемым даже при  $N > 15$  и  $L > 15$ .

Анализ рассматриваемой задачи показывает, что конструктивные подходы к ее решению лежат на пути учета ее специфических особенностей, основная из которых – представление всего множества распределяемых функций в виде объединения подмножеств однотипных проверяемых объектов  $N_i$ , размещаемых по датам проверки последовательно. Поэтому рассмотрим вначале алгоритм закрепления по датам плана проверок (ПП) элементов множества однотипных объектов.

В соответствии с введенными ограничениями каждый проверяемый объект надзора можно проводить только на один период из множества  $A$ :

$$\sum_{l \in \Lambda} x_{lj} \leq 1, \quad \forall j \in N_i, \quad i \in M. \quad (1)$$

С учетом ограничения (1) максимальное количество анализируемых вариантов определится:

$$Card(X) = \frac{L!}{(L - card(N_i))!}.$$

Опишем алгоритм генерирования размещений в лексикографическом порядке. Для этого на множестве всех последовательностей длиной  $L$  с элементами из множества  $N_i$  определим лексикографический порядок:

$$\langle b_1, \dots, b_2, \dots, b_{N_i} \rangle \prec \langle 1, 2, \dots, L \rangle \Leftrightarrow \exists (i \in \{1, 2, \dots, N_i\}) (b_k \leq l_k \wedge \forall (N_i < L)).$$

Введём в рассмотрение  $N_i$ -мерный вектор-столбец  $\Lambda^T = \langle l[1], \dots, l[b_j], \dots, l[N_i] \rangle$ , компоненты которого принимают значения из множества  $B$ . Таким образом, значение первой компоненты вектора  $\Lambda^T$  означает номер надзорного объекта, который подвергается

ПП в рассматриваемый период и т.д. Процесс генерирования размещений элементов из множества  $N_i \in B$  по элементам из множества  $A$  можно представить процедурами перемещения по корневому дереву. Корнем данного дерева является вершина, которая соответствует закреплению надзорного объекта  $b_j \in B$  за периодом проверки. Очевидно, что число корневых деревьев будет равняться мощности множества  $A$ .

Перемещение в корневом дереве осуществляется по вершинам с первого уровня, соответствующего корневой вершине и до  $L$ -уровня, соответствующего закреплению элемента  $b_j \in N_i$  на элементе из множества  $A$ .

Период начала ПП  $l \in A$ , за которым закрепляется проверяемый объект  $b_j \in N_i$ , будем включать во множество  $\{X\}$ . Ниже даётся пошаговое описание алгоритма генерирования множества размещений.

Шаг 1. Производится инициализация  $N_i$ -мерного вектора – столбца  $\Lambda^T$  путём присвоения всем компонентам данного вектора значений, равных нулю. Множество  $X = \emptyset$ . Задаётся начальное значение индекса элемента  $b_k \in N_i$  (индекс уровня дерева), равное единице,  $i=1$ .

Шаг 2. Проверяется выполнение условий:

1. Начала движения к  $(j+1)$  уровню дерева:

$$l(j) = L. \quad (2)$$

Если условие (2) выполняется, то осуществляется переход к шагу 3. В противном случае – переход к шагу 4.

2. Начала движения по вершинам, принадлежащим последнему уровню дерева ( $N_i$  уровню):

$$j = N_i. \quad (3)$$

При выполнении условия (3) осуществляется переход к шагу 7.

3. Окончания процедуры генерирования размещений, которое наступает при выполнении условия (3) для  $j = N_i$ .

Шаг 3. Реализуется процедура возврата к корневой вершине дерева. Для этого  $j$  компоненте вектора  $\Lambda^T$  присваивается значение  $l_n[j]=0$ . Уменьшается значение  $j$  на единицу, то есть осуществляется переход на  $(j-1)$  уровень дерева. Из множества  $X$  удаляется элемент  $l[j]$ .

Шаг 4. Вершине дерева  $j$ -го уровня ставится в соответствие элемент из множества  $A$  путём увеличения номера периода проверки на единицу:

$$l[j] = l[j] + 1.$$

Шаг 5. Начинается процедура движения к  $(j+1)$  уровню дерева. Признаком начала работы этой процедуры является выполнение условия:

$$l[K] \notin X_n^k. \quad (4)$$

При выполнении условия (4) осуществляется переход к шагу 6, в противном случае – к шагу 2.

Шаг 6. Выполняется процедура движения к  $(j+1)$  уровню дерева. Для этого к множеству  $X_n^k$  добавляется период с номером  $l[j]$ . Затем осуществляется переход к следующему уровню, то есть

$$X_n^k = X_n^k \cup \{l[j]\}, \quad j = j + 1.$$

Шаг 7. Осуществляется движение по вершинам  $N_i$  уровня дерева:

$$l[N_i] = l[N_i] + 1.$$

Шаг 8. Проверяется условие (4). Если оно выполняется, то осуществляется считывание варианта размещения элемента по периодам ПП из  $\Lambda^T$ . В противном случае – переход к шагу 9.

Шаг 9. Проверяется условие:

$$l[N_i] = N.$$

При его выполнении – переход к шагу 2, в противном случае – к шагу 7.

Шаг 10. Конец выполнения процесса генерирования размещений.

В рассмотренном алгоритме можно выделить три основных процедуры движения по корневому дереву.

Процедура 1. Движение по корневому дереву от первого уровня до  $N_i$ -го уровня. Эта процедура реализуется на шагах 2, 4, 5.

Процедура 2. Процедура возврата. Осуществляется движение к корню дерева. Реализуется на шагах 2–4.

Процедура 3. Процедура движения по вершинам одного уровня дерева. Реализуется на шагах 7–9 алгоритма генерации размещений.

Выбор вершины на дереве осуществляется путём последовательного увеличения значений компонент вектора  $\Lambda^T$  с последующей проверкой на принадлежность элемента к множеству  $X$ . Такая процедура выбора позволяет генерировать, не только размещения  $N_i$  проверяемых надзорных объектов по  $L$  периодам начала ПП, но и осуществлять процесс закрепления элемента  $b_j$  по периодам из множества  $A$ . Это дает возможность на первых этапах решения задачи выбора значений параметров сужать область альтернативных вариантов закрепления за счёт уменьшения количества анализируемых вершин, принадлежащих одному уровню. При  $N_i = L$  данный алгоритм является нерекурсивным алгоритмом генерирования  $L$  – факториал перестановок [2].

Можно отметить, что число итераций, необходимых для выбора одной вершины на дереве, является величиной переменной, зависящей от  $L$  и  $N$ . Это число меняется от одной итерации до  $(N+L)/2$ . Поскольку в этом алгоритме реализована концепция «алгоритма с возвратом» (англ. backtracking), то это может оказаться существенным в ситуации, когда с выбором вершины на дереве связаны некоторые вычисления и когда существует возможность использования частичных результатов, полученных при выборе вершин дерева.

### Литература

1. Струченков В.И. Методы оптимизации в прикладных задачах. М.: Солон-Пресс, 2009. 320 с.
2. Подиновский В.В., Ногин В.Д. Парето, оптимальные решения многокритериальных задач. М.: Наука, 1982.