УДК 004.056: 519.832.2: 519.854 DOI: 10.18698/2541-8009-2019-04-471

АЛГОРИТМЫ ПОИСКА ДОПУСТИМЫХ РЕШЕНИЙ ДЛЯ ПОСТРОЕНИЯ МАТРИЦЫ ИГРЫ В ЗАДАЧЕ ВЫБОРА ОБЪЕКТОВ ЗАЩИТЫ ПРИ ОГРАНИЧЕНИЯХ НА РЕСУРСЫ

А.А. Кошман

anastasiay.andreevnaa@gmail.com
SPIN-код: 9147-2445

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Рассмотрена задача выбора объектов защиты при ограничениях на ресурсы. Задача является игрой с нулевой суммой, где каждый игрок решает свою задачу булева программирования при фиксированном решении другого игрока. Описаны два алгоритма поиска допустимых решений защитника и нападающего с максимальным числом единиц в игровой постановке задачи. Эти алгоритмы могут быть использованы для построения матрицы игры. Алгоритмы основаны на неполном переборе решений начиная с единичного и нулевого решений. Проверена работоспособность алгоритмов, доказано, что они позволяют получать один и тот же результат. Приведен пример решения задачи.

Ключевые слова

Защита информации, теория игр, булево программирование, матрица игры, оптимальная стратегия, объект защиты, ограничение ресурсов, неполный перебор, дерево решений

Поступила в редакцию 20.03.2019

Введение. Подавляющее большинство социально-экономических решений приходится принимать с учетом противоречивых интересов, относящихся либо к различным лицам или организациям, либо к различным аспектам рассматриваемого явления, либо к тому и другому. В таких случаях невозможно применять традиционные методы оптимизации. В обычных экстремальных задачах речь идет о выборе решения одним лицом, при этом результат решения зависит от этого выбора, т. е. определяется действиями только одного лица. В такую схему не укладываются ситуации, где решения, оптимальные для одной стороны, совсем не оптимальны для другой, и результат решения зависит от всех конфликтующих сторон.

Конфликтный характер таких задач не предполагает вражды между участниками, а свидетельствует о различных интересах. При анализе подобных ситуаций применяют математический аппарат — теорию игр.

Теория игр представляет собой часть обширной теории, изучающей процессы принятия оптимальных решений. Она дает формальный язык для описания процессов принятия сознательных, целенаправленных решений с участием одного или нескольких лиц в условиях неопределенности и конфликта, вызываемого столкновением интересов конфликтующих сторон. Неопределенность может быть вызвана не только стремлением противников скрыть свои действия в игре, но и дефицитом информации и данных о рассматриваемом явлении. В этом случае можно говорить о конфликте человека с природой.

При решении задач защиты информации (например, при выборе объектов для защиты в условиях ограниченных ресурсов) часто используется подход на основе теории игр. Как правило, рассматриваются два игрока — игрок стороны защиты и игрок стороны нападения. Каждый игрок решает свою оптимизационную задачу — задачу математического программирования при фиксированном решении другого игрока. Постановка задачи может быть как непрерывной, так и дискретной, игра может быть как игра с нулевой суммой, так и без нулевой суммы. Некоторые примеры подобных задач представлены в [1–5].

Ниже рассмотрим игровую задачу выбора объектов для защитников, при этом нападающий выбирает эти же объекты для проведения атак. Считается, что ресурсы защитника и нападающего ограничены, подобная задача в дискретной форме описана в [2]. Игра является игрой с нулевой суммой, каждый игрок решает свою задачу булева программирования при фиксированном решении другого игрока. Задача может быть сведена к матричной игре большой размерности, далее может быть найдена седловая точка в чистых или смешанных стратегиях. Для построения подобной матрицы ниже представлены два алгоритма поиска всех допустимых решений, при этом используются идеи неполного перебора на основе подходов, предложенных в методе Балаша [6, 7].

Постановка задачи выбора игроками объектов для атаки и для защиты

Исходные данные

Базисные множества:

1) $Z=\{z_1,z_2,\ldots,z_m\}$ — множество защищаемых объектов, $M=\{1,2,\ldots,m\}$ — множество индексов этих объектов;

2) $R=\{r_1,r_2,\ldots,r_l\}$ — множество ограниченных ресурсов стороны защиты, $L=\{1,2,\ldots,l\}$ — множество индексов этих ресурсов;

3) $N=\{n_1,n_2,\ldots,n_s\}$ — множество ограниченных ресурсов стороны нападения, $S=\{1,2,\ldots,s\}$ — множество индексов этих ресурсов.

Параметры элементов множеств и отношений между ними задаются вещественными значениями:

1) $w_i \ge 0\ \forall i \in M$ — возможный ущерб при нарушении безопасности i-го защищаемого объекта (стоимость объекта);

2) $p_{\text{пр}\,i} \in [0,1]\ \forall i \in M$ — вероятность (или возможность) предотвращения атаки на i-й объект при его защите;

3) $a_{ki} \ge 0\ \forall k \in L,\ i \in M$ — значение k-го ограниченного ресурса, используемого для обеспечения защиты i-го объекта;

4) $b_k \ge 0\ \forall k \in L$ — максимальное значение k-го ограниченного ресурса, выделенное на защиту;

5) $c_{ki} \ge 0\ \forall k \in S,\ i \in M$ — значение k-го ограниченного ресурса стороны нападения, используемого для атаки на i-й объект;

6) $d_k \geq 0 \ \forall k \in S$ — максимальное значение k-го ограниченного ресурса стороны нападения, выделенного на проведение атак.

Искомые параметры. Введем переменную $x_i \in \{0,1\} \ \forall i \in M$, $x_i = 1$, если i-й объект защищается защитником, $x_i = 0$ — в противном случае. Переменные образуют вектор $X = \left[x_1, x_2, ..., x_m\right]^{\text{т}}$. Для стороны нападения введем переменную $y_i \in \{0,1\} \ \forall i \in M$, $y_i = 1$, если i-й объект подвергается атаке нападающего, $y_i = 0$ — в противном случае. Переменные образуют вектор $Y = \left[y_1, y_2, ..., y_m\right]^{\text{т}}$.

Показатели игроков. Для игры с нулевой суммой показатели качества двух игроков определяются ущербом стороны защиты. Средний ущерб можно определить так:

$$U(X,Y) = U_{\max}(Y) - U_{\text{пр}}(X,Y) = \sum_{i \in M} w_i y_i - \sum_{i \in M} p_{\text{пр}} w_i x_i y_i, \tag{1}$$

где $U_{\max}(Y) = \sum_{i \in M} w_i y_i$ — максимальный ущерб, который может быть нанесен стороной нападения при отсутствии защиты; $U_{\text{пр}}(X,Y) = \sum_{i \in M} p_{\text{пр}\,i} w_i \, x_i y_i$ — предотвращенный ущерб стороной защиты.

Сторона защиты желает этот показатель минимизировать, а сторона нападения — максимизировать.

Ограничения. Система ограничений на использование ограниченных ресурсов стороной защиты, задающая множество допустимых альтернатив $\Delta_{\text{доп}}^{(X)}$, имеет вид

$$\Delta_{\text{доп}}^{(X)} = \sum_{i \in M} a_{ki} x_i \leq b_k \forall k \in L \tag{2}$$

Система ограничений на использование ограниченных ресурсов стороной нападения, задающая множество допустимых альтернатив $\Delta_{\text{доп}}^{(Y)}$, имеет вид

$$\Delta_{\text{доп}}^{(Y)} = \sum_{i \in M} c_{ki} y_i \leq d_k \forall k \in S \tag{3}$$

Будем полагать, что системы ограничений (2) и (3) не позволяют защитнику и нападающему выбрать решения, состоящие из всех единиц (полная защита или полное нападение), поскольку такие решения являются оптимальными и задача становится тривиальной. Таким образом, каждый из игроков решает задачу линейного булевого программирования при фиксированном решении другого игрока.

Алгоритмы решения задачи на основе построения матрицы игры. Можно построить матрицу игры в явном виде и далее искать седловую точку в чистых или смешанных стратегиях. Для этого необходимо перебрать все допустимые решения защитника, удовлетворяющие ограничениям (2), и все допустимые решения нападающего, удовлетворяющие ограничениям (3). Пусть решения защитника в матрице задают столбцы, а решения нападающего — строки. Элементами матрицы являются значения показателя (1) в условных единицах при заданных решениях защитника и нападающего. Например, для случая, когда размерности векторов X и Y равны пяти, возможный вариант матрицы представлен в табл. 1.

Таблица 1

Возможный вариант представления матрицы игры

Решение нападающего	Решение защитника					
	$[1,1,0,0,0]$	$[1,0,1,0,0]$	$[1,0,0,0,1]$	$[0,1,1,0,0]$	$[0,1,0,0,1]$	$[0,0,0,1,1]$
$[1,0,0,0,1]$	1660	1660	593	3190	2120	2120
$[0,1,0,0,0]$	385	3740	3740	385	385	3740
$[0,0,1,1,0]$	5940	2990	5940	2990	5940	4730
$[0,0,0,1,1]$	3330	3330	2260	3330	2260	1060

При этом для сокращения матрицы игры можно сразу исключить доминируемые строки и столбцы. Например, если решение нападающего $[0,0,0,1,1]$ является допустимым, то допустимыми являются решения нападающего, получаемые из исходного решения заменой единиц нулями, например, решения $[0,0,0,0,1]$, $[0,0,0,1,0]$, $[0,0,0,0,0]$. Но эти решения не дадут выигрыш нападающему, больший, чем решение $[0,0,0,1,1]$ при любых решениях защитника, т. е. эти решения доминируются решением $[0,0,0,1,1]$ и их можно исключить из матрицы. Аналогично можно поступить для защитника. Таким образом, в табл. 1 присутствуют решения с максимальным числом единиц: в этих решения при замене любого нуля единицей получаем недопустимое по ограничениям (2) или (3) решение.

Рассмотрим два алгоритма поиска всех допустимых решений, удовлетворяющих ограничениям (2) или (3), содержащих максимальное число единиц. Алгоритмы основаны на идеях неполного перебора за счет исключения из перебора заведомо недопустимых решений. Первый алгоритм основан на переборе начиная с решения, содержащего все единицы (единичное решение), второй алгоритм основан на переборе начиная с решения, содержащего все 0 (нулевое решение).

Алгоритм поиска допустимых решений с максимальным числом единиц начиная с единичного решения. Рассмотрим особенности алгоритма поиска всех допустимых решений, удовлетворяющих ограничениям (2), для ограничений (3) алгоритмы работают аналогично. Все возможные решения можно представить в виде дерева решений, вид дерева для случая $m = 4$ показан на рис. 1.

Вершина дерева — решение, состоящее из всех единиц, предполагается, что оно недопустимое. Следующий уровень — решения, содержащие на одну единицу меньше, и т. д. Самый нижний уровень содержит решение из всех нулей. Решения уровня, который не является самым верхним, можно получить из решений уровня выше путем замены одной единицы нулем. На рис. 1 показаны связи между вершинами уровней. Решение предшествует решению более нижнего уровня или доминирует над ним, если из него можно получить решение более нижнего уровня путем одной замены на нуль. Очевидно, что если на каком-то уровне найдено решение допустимое по ограничениям (2), то получать из этого решения более низких уровней путем замены единиц на нули не имеет смысла, так как получаемые решения будут содержать меньшее число единиц, чем исходное допустимое решение.

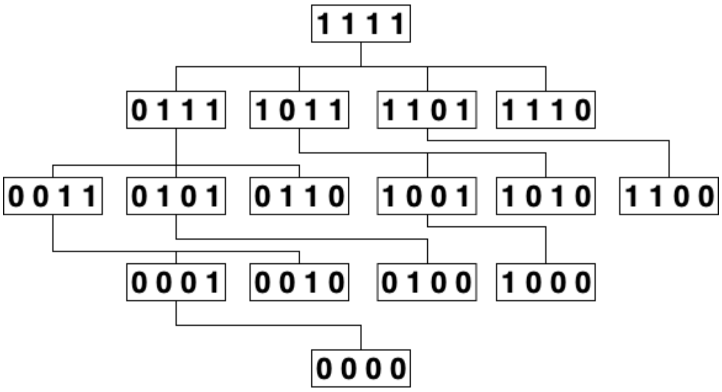


Рис. 1. Дерево решений для алгоритма, начинающегося перебор с единичного решения, при $m = 4$

Также некоторые решения более низкого уровня можно получить из нескольких решений уровня выше. Например, решение $\left[0,1,1,0\right]$ можно получить из решений $\left[0,1,1,1\right]$ и $\left[1,1,1,0\right]$. Чтобы уменьшить число рассматриваемых вариантов, можно оставить на дереве только те переходы, которые получаются заменой каждой из последних единиц в решении уровня. На рис. 1 представлены только эти переходы от решения к решению.

Например, из решения $\left[1,0,1,1\right]$ можно получить новые решения $\left[1,0,0,1\right]$ и $\left[1,0,1,0\right]$ путем замены последних двух единиц. Также может быть получено решение $\left[0,0,1,1\right]$, но это решение не является новым, так как оно уже было получено из более раннего решения $\left[0,1,1,1\right]$ (решение $\left[0,1,1,1\right]$ находится на одном уровне дерева с решением $\left[1,0,1,1\right]$, но оно будет рассмотрено раньше исходя из порядка обхода дерева).

Сформулируем следующий рекурсивный алгоритм поиска всех допустимых решений, путем неполного перебора вершин дерева, представленного на рис. 1.

Шаг 0 (предварительный). Полагаем начальное решение $X = \left[1,1,...,1\right]^{\mathrm{T}}$ (состоит из всех единиц). Далее вызывается рекурсивная функция, задающая рекурсивный алгоритм, на вход функции передается вектор X.

Шаг 1. Проверяем X по ограничениям (2). Если вектор X допустим, то проверяем, действительно ли он содержит максимальное число единиц. Для этого переходим к следующему шагу. Если вектор *X* недопустим по ограничениям (2), то переходим к шагу 3.

Шаг 2. В исходном векторе *X* последовательно в цикле заменяем каждый нуль единицей. Проверяем допустимость нового вектора после одной замены по ограничениям (2). Если хотя бы один вектор, полученный заменой одного нуля единицей, то завершение работы рекурсивного алгоритма: исходный вектор *X* недопустим, поскольку он не содержит максимальное число единиц. В противном случае вектор *X* включаем в список найденных решений и завершаем работу рекурсивного алгоритма.

Шаг 3. Находим число единиц, которые расположены подряд в конце вектора *X*. Если их количество равно нулю, то происходит завершение работы рекурсивного алгоритма. В противном случае в цикле каждую такую единицу последовательно заменяем нулем, начиная со стороны начала вектора. Новые вектора, полученные заменой одной единицы нулем, передается в качестве *X* в рекурсивный алгоритм (снова начинает работать шаг 1).

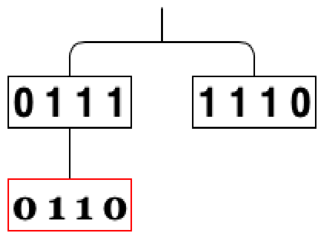


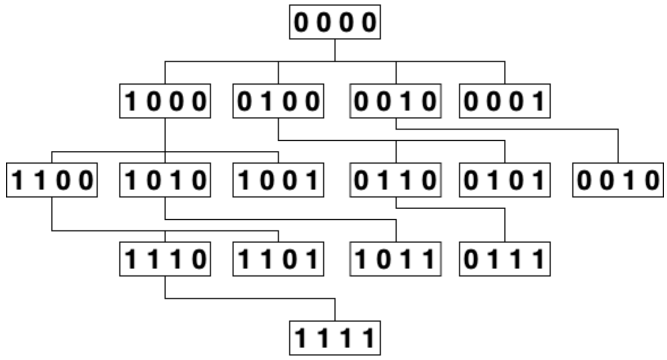
Рис. 2. Пример поиска решений с максимальным числом единиц

Примечание. На шаге 2 приходится проверять допустимость решения на предмет, содержит ли оно максимальное число единиц, путем замены каждого нуля единицей, так как при работе рекурсивной функции происходит просмотр дерева решений, представленного на рис. 1, в глубину. Возможно нахождение допустимого решения, которое уступает другим допустимым решением более высокого уровня (с большим числом единиц), которое будет найдено позже. Визуализация шага 2 представлена на рис. 2. Предположим, что решение [1,1,1,0] — допустимо. Но в связи с просмотром дерева в глубину в тот момент, когда будет найден допустимый вектор [0,1,1,0], решение [1,1,1,0] еще не будет рассмотрено алгоритмом. Поэтому следуют сделать дополнительную проверку на допустимость решения с большим числом единиц.

Алгоритм поиска допустимых решений с максимальным числом единиц, начиная с нулевого решения. По аналогии с предыдущим алгоритмом можно построить дерево решений, представленное на рис. 3 для случая $m = 4$. Отличие в том, что вершина дерева содержит нулевое решение, которое допустимо по ограничениям, решения нижних уровней получаются заменой нулей единицами. Если на каком-то уровне имеем допустимое решение, а на уровне ниже все решения, получаемые из этого решения путем замены последовательно одного нуля единицей, недопустимые, то данное решение является искомым, т. е. содержит максимальное число единиц.

По аналогии с предыдущим алгоритмом, чтобы уменьшить число рассматриваемых вариантов, можно оставить на дереве только переходы, которые получаются заменой каждого последнего нуля в решении единицей.

Рис. 3. Дерево решений для алгоритма, начинающего перебор с нулевого решения, при $m = 4$



Рассмотрим рекурсивный алгоритм поиска всех допустимых решений пошагово.

Шаг 0 (предварительный). Полагаем начальное решение $X = [0,0,\ldots,0]^{\text{т}}$ (состоит из всех нулей). Далее вызывается рекурсивная функция, задающая рекурсивный алгоритм, на вход функции передается вектор X.

Шаг 1. В цикле последние нули на конце вектора (если они есть) последовательно заменяем единицами начиная со стороны начала вектора. Новый вектор, полученный заменой одного нуля единицей, проверяется на допустимость. Если он удовлетворяет ограничениям, то он передаются в качестве вектора X в рекурсивный алгоритм (снова начинает работать шаг 1). Если новых допустимых векторов не получено, переходим к шагу 2.

Шаг 2. В исходном векторе X последовательно в цикле заменяем каждый нуль единицей. Проверяем допустимость нового вектора после одной замены по ограничениям (2). Если хотя бы один вектор, полученный заменой одного нуля единицей, допустим, то завершение работы рекурсивного алгоритма, исходный вектор X недопустим, так как он не содержит максимальное число единиц. В противном случае вектор X включаем в список найденных решений и завершаем работу рекурсивного алгоритма.

Пример решения задачи (эксперименты). Рассмотрим пример задачи. Пусть задано 8 объектов защиты, например, узлов сети. Значения ущерба и вероятности защиты объектов представлены в табл. 2.

Таблица 2

Значения ущерба и вероятностей защиты объектов

Характеристика	Номер объекта							
	1	2	3	4	5	6	7	8
w_i , у.е.	4 000	10 000	3 000	9 000	5 000	9 500	10 000	8 000
$p_{\text{пр}\,i}$	0,80	0,99	0,70	0,95	0,85	0,90	0,50	0,92

У защитника и нападающего по четыре вида ресурсов, примеры параметров ограничений для защитника и нападающего приведены в табл. 3. Параметры первого ограничения для защитника и нападающего приведены без нормировки (например, это может быть стоимость), остальные параметры представлены в нормированном виде (значение от 0 до 1).

Таблица 3

Параметры системы ограничений на ресурсы защитника и нападающего

Номер ограничения	Коэффициенты в левых частях ограничений								Ресурс
Параметры ограничений защитника a_{ki}									b_k
1	100	1000	200	900	400	500	1200	1100	3000
2	0,03	0,30	0,05	0,15	0,30	0,30	0,35	0,10	0,70
3	0,05	0,20	0,20	0,25	0,30	0,10	0,33	0,35	0,60
4	0,01	0,05	0,02	0,05	0,02	0,01	0,01	0,01	0,20
Параметры ограничений нападающего c_{ki}									d_k
1	50	600	60	500	100	120	1000	550	1500
2	0,02	0,20	0,05	0,25	0,35	0,30	0,30	0,15	0,80
3	0,02	0,30	0,30	0,35	0,40	0,15	0,35	0,30	0,80
4	0,15	0,10	0,20	0,05	0,20	0,10	0,10	0,10	0,50

Результаты решения задачи предложенными двумя алгоритмами представлены в табл. 4, 5.

Таблица 4

Результаты стороны защиты

№ п/п	Решения X	
	Начиная с нулевого решения	Начиная с единичного решения
1	11100100	00010001
2	11010000	00010010
3	11001000	00101100
4	11000010	10000101
5	11000001	10000110
6	10110100	10001100
7	10101000	10011000
8	10100010	10100001
9	10100001	10100010
10	10011000	10101000
11	10001100	10110100
12	10000110	11000001
13	10000101	11000010
14	00101100	11001000
15	00010010	11010000
16	00010001	11100100

Использование обоих алгоритмов привело к аналогичным результатам, но в разной последовательности. Размер исходной матрицы игры для примера составил 23×16.

Таблица 5

Результаты стороны нападения

№ п/п	Решения Y	
	Начиная с нулевого решения	Начиная с единичного решения
1	11100000	00010010
2	11010000	00010101
3	11001000	00100101
4	11000101	00100110
5	10110000	00101000
6	10100100	00110100
7	10100010	01010100
8	10100001	01100100
9	10011000	10000110
10	10010100	10001001
11	10010001	10001010
12	10001100	10001100
13	10001010	10010001
14	10001001	10010100
15	10000110	10011000
16	01100100	10100001
17	01010100	10100010
18	00110100	10100100
19	00101000	10110000
20	00100110	11000101
21	00100101	11001000
22	00010101	11010000
23	00010010	00010010

Заключение. В статье рассмотрены два алгоритма поиска допустимых решений защитника и нападающего с максимальным числом единиц в игровой постановке задачи, алгоритмы могут быть использованы для построения матрицы игры. Один алгоритм основан на неполном переборе решения, начиная с решения, состоящего из всех единиц, второй алгоритм основан на неполном переборе решения, начиная с решения, состоящего из всех нулей. Проверена работоспособность алгоритмов, алгоритмы позволяют получать один и тот же результат. Можно предположить, что алгоритм, основанный на переборе начиная с решения, состоящего из единиц, работает быстрее алгоритма, основанного на переборе, начиная с решения, состоящего из нулей, в случае, когда ресурсов для игроков доступно больше, т. е. в допустимых решения больше единиц, чем нулей, второй алгоритм работает быстрее, когда ресурсов меньше, т. е. в допустимых решения больше нулей, чем единиц. Однако данный вопрос требует дальнейших исследований.

Литература

[1] Chen L., Leneutre J. A game theoretical framework on intrusion detection in heterogeneous networks. *IEEE Trans. Inf. Forensics Security*, 2009, vol. 4, no. 2, pp. 165–178. DOI: 10.1109/TIFS.2009.2019154 URL: https://ieeexplore.ieee.org/document/4815406

[2] Быков А.Ю., Алтухов Н.О., Сосенко А.С. Задача выбора средств защиты информации в автоматизированных системах на основе модели антагонистической игры. *Инженерный вестник*, 2014, № 4. URL: http://engbul.bmstu.ru/doc/708106.html

[3] Быков А.Ю., Шматова Е.С. Алгоритмы распределения ресурсов для защиты информации между объектами информационной системы на основе игровой модели и принципа равной защищенности объектов. *Наука и образование: научное издание*, 2015, № 9. URL: https://technomagelpub.elpub.ru/jour/article/view/186

[4] Быков А.Ю., Панфилов Ф.А., Ховрина А.В. Алгоритм выбора классов защищенности для объектов распределенной информационной системы и размещения данных по объектам на основе приведения оптимизационной задачи к задаче теории игр с непротивоположными интересами. *Наука и образование: научное издание*, 2016, № 1. URL: http://engineering-science.ru/en/doc/830972.html

[5] Быков А.Ю., Крыгин И.А., Муллин А.Р. Алгоритм распределения ресурсов системы защиты между активами мобильного устройства на основе игры с нулевой суммой и принципа равной защищенности. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2018, № 2, с. 48–68. DOI: 10.18698/0236-3933-2018-2-48-68 URL: http://vestnikprib.ru/catalog/icec/sysan/1096.html

[6] Басараб М.А. Теория оптимизации и исследование операций. М., Изд-во МГТУ им. Н.Э. Баумана, 2012.

[7] Басараб М.А., Вельц С.В. Методы оптимизации и исследование операций в области информационной безопасности. М., Изд-во МГТУ им. Н.Э. Баумана, 2015.

Кошман Анастасия Андреевна — студентка кафедры «Информационная безопасность», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Быков Александр Юрьевич, кандидат технических наук, доцент, заместитель заведующего кафедрой «Информационная безопасность», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

ALGORITHMS OF SEARCHING FOR ADMISSIBLE SOLUTIONS TO CONSTRUCT THE MATRIX OF THE GAME IN THE PROBLEM OF CHOICE OF PROTECTION OBJECTS UNDER THE RESTRICTIONS ON RESOURCES

A.A. Koshman

anastasiay.andreevnaa@gmail.com
SPIN-code: 9147-2445

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The task of selecting the objects of protection under the restrictions on resources was considered. A task is a zero-sum game, where each player solves his own Boolean programming problem under the fixed solution of another player. Two algorithms for finding feasible solutions for the defender and the attacker with the maximum number of points in the game formulation of the problem are described. These algorithms can be used to build a game matrix. Algorithms are based on incomplete enumeration of solutions, starting with single and zero solutions. The performance of the algorithms was tested, it was proved that they allow to get the same result. An example of solving the problem is given.

Keywords

Information protection, game theory, Boolean programming, game matrix, optimal strategy, object of protection, resource restriction, incomplete enumeration, decision tree

Received 20.03.2019

References

[1] Chen L., Leneutre J. A game theoretical framework on intrusion detection in heterogeneous networks. *IEEE Trans. Inf. Forensics Security*, 2009, vol. 4, no. 2, pp. 165–178. DOI: 10.1109/TIFS.2009.2019154 URL: https://ieeexplore.ieee.org/document/4815406

[2] Bykov A.Yu., Altukhov N.O., Sosenko A.S. Problem of information security product choice in automated systems based on antagonistic play model. *Inzhenernyy vestnik* [Engineering Bulletin], 2014, no. 4. URL: http://engbul.bmstu.ru/doc/708106.html (in Russ.).

[3] Bykov A.Yu., Shmatova E.S. The algorithms of resource distribution for information security between objects of an information system based on the game model and principle of equal security of objects. *Nauka i obrazovanie: nauchnoe izdanie* [Science and Education: Scientific Publication], 2015, no. 9. URL: https://technomagelpub.elpub.ru/jour/article/view/186 (in Russ.).

[4] Bykov A.Yu., Panfilov F.A., Khovrina A.V. The algorithm to select security classes for objects in distributed information systems and place data in the objects through reducing the optimization problem to the theory of games with non-conflicting interests. *Nauka i obrazovanie: nauchnoe izdanie* [Science and Education: Scientific Publication], 2016, no. 1. URL: http://engineering-science.ru/en/doc/830972.html (in Russ.).

[5] Bykov A.Yu., Krygin I.A., Mullin A.R. Algorithms of the protection system between assets of a mobile device on the basis of zero-sum game and equal security principle. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2018, no. 2, pp. 48–68. DOI: 10.18698/0236-3933-2018-2-48-68 URL: http://vestnikprib.ru/catalog/icec/sysan/1096.html (in Russ.).

[6] Basarab M.A. Teoriya optimizatsii i issledovanie operatsiy [Optimization theory and operation research]. Moscow, Bauman MSTU Publ., 2012 (in Russ.).

[7] Basarab M.A., Vel'ts S.V. Metody optimizatsii i issledovanie operatsiy v oblasti informatsionnoy bezopasnosti [Optimization methods and operation research in field of information security]. Moscow, Bauman MSTU Publ., 2015 (in Russ.).

Koshman A.A. — Student, Department of Information Security, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Bykov A.Yu., Cand. Sc. (Eng.), Assoc. Professor, Deputy Head of Department of Information Security, Bauman Moscow State Technical University, Moscow, Russian Federation.