

ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 33 № 2 2026

Новосибирск
Издательство Института математики

УСТОЙЧИВОСТЬ ВЕРШИННЫХ ПОКРЫТИЙ В ЗАДАЧЕ О ВЕЧНОМ ВЕРШИННОМ ПОКРЫТИИ

С. Ю. Утюпин

Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: stepan.utyupin@gmail.com

Аннотация. Задача о вечном вершинном покрытии является вариантом задачи о вершинном покрытии графа и может рассматриваться как динамическая игра двух сторон (атакующего и защитника) с бесконечным числом шагов. Для построения стратегии защитника задача представляется в виде динамической игры, на каждом шаге которой взаимодействие противоборствующих сторон формализуется как двухуровневая задача математического программирования. В данной работе предложена процедура, позволяющая ответить на вопрос, является ли заданное вершинное покрытие вечным. Идея процедуры заключается в рекурсивной проверке устойчивости вершинных покрытий, полученных в результате решения задач защитника и построения дополнительного графа покрытий и переходов. Табл. 2, ил. 7, библиогр. 12.

Ключевые слова: вечное вершинное покрытие, защита рёбер графа, алгоритм проверки устойчивости, динамическая игра Штакельберга.

Введение

В задаче о вершинном покрытии задан граф $G = (V, E)$, целое положительное число k и требуется ответить на вопрос, существует ли подмножество $C \subseteq V$, называемое вершинным покрытием, размера не более k , содержащее по меньшей мере одну вершину каждого ребра из E . Данная задача широко исследована и является классической. Однако существует множество различных динамических её вариантов. Такие задачи — вечной защиты рёбер или вершин графа — появились в 1990-х гг. как формализация стратегии императора Константина по защите Римской империи [1]. В них рёбра или вершины графа защищаются от бесконечной последовательности атак. Защита осуществляется с помощью

охранников, размещаемых в вершинах графа и способных перемещаться по его рёбрам после атаки. Говорят, что вершина (ребро) графа защищённое, если охранник находится в вершине или соседней (инцидентной ребру) вершине.

В настоящей работе исследуем вариант задачи, где объектами нападения являются рёбра графа. Такой вариант динамической задачи о вершинном покрытии, называемый задачей о вечном вершинном покрытии (eternal vertex cover problem), предложен в [2]. Эту задачу можно интерпретировать как бесконечную игру двух сторон — атакующего и защитника. Первоначально защитник размещает в вершинах заданного графа охранников так, что вершины, в которых размещены охранники, образуют вершинное покрытие. Далее на каждом шаге атакующий выбирает для атаки любое ребро, в одной из вершин которого нет охранника. В ответ защитник перемещает охранника из вершины атакованного ребра в другую вершину этого ребра. При этом защитник имеет право из любой вершины переместить охранника вдоль любого ребра в соседнюю вершину, если в ней не было охранника, либо охранник был перемещён из этой вершины. Также в одну вершину не могут быть перемещены несколько охранников, при этом один и тот же охранник не может быть перемещён более одного раза на одном шаге. Если реконструированное размещение охранников в вершинах образует вершинное покрытие, то начинается следующий шаг с той же последовательностью действий. Если же с помощью указанной процедуры не удаётся построить вершинное покрытие, т. е. хотя бы одно ребро окажется незащищённым, то атакующий побеждает и игра завершается. Победителем становится защитник, если он может защититься от бесконечной последовательности атак. Задача нахождения минимального числа охранников EVCN (eternal vertex cover number), образующих вершинное покрытие, для которого в рассматриваемой игре может быть построена выигрышная стратегия защитника, есть задача о вечном вершинном покрытии.

Ранее показано, что эта задача относится к классу NP-трудных [3], а также остаётся NP-трудной для двудольных графов [4]. Кроме того, задача о вечном вершинном покрытии широко исследована с точки зрения теории графов: многие работы либо предлагают алгоритмы для конкретных семейств графов, либо приводят оценки для общих или частных случаев. Так, предлагается квадратичный по времени алгоритм вычисления EVCN для хордальных графов [5], а для двусвязных хордальных графов предложен полиномиальный алгоритм [6]. Также представлены линейный алгоритм для кактусового графа [7], полиномиальные алгоритмы для цепных графов и кографов [8], а также линейный алгоритм для расщеплённых графов [8]. Для графов, построенных путём замены каждого ребра дерева произвольным элементарным двудольным графом

(или произвольной кликой), показано, что вычисление EVCN возможно за полиномиальное время [9]. Кроме того, доказано существование полиномиального алгоритма для двудольных графов [4]. Помимо этого показано, что для задачи о вечном вершинном покрытии существует 2-приближённый полиномиальный по времени алгоритм [3]. Предложена нижняя оценка для EVCN произвольного графа через мощность вершинного покрытия минимального размера, содержащего все точки сочленения данного графа [5]. Проведено исследование о связи числа EVCN с минимальным числом охранников, необходимых для бесконечной защиты вершин графа (задача о вечном доминировании) [10].

Задача проверки, является ли заданное вершинное покрытие вечным, несмотря на внешнюю близость с оригинальной постановкой, по-видимому, более сложна. В литературе на данный момент не обнаружено рассмотренной задачи в такой постановке. В данной работе рассматриваем ситуацию, когда задан граф, на структуру которого не наложено никаких ограничений, а также дано некоторое вершинное покрытие без каких-либо предположений о нём. Рассматривается вопрос, какой игрок выигрывает, если игра начнётся с заданного вершинного покрытия. Ответ на этот вопрос требует анализа всех бесконечных последовательностей атак.

На практике достаточно рассматривать последовательности некоторой достаточно большой длины из-за конечности множества возможных состояний игры, однако стоит ожидать, что эта длина будет неизвестной. Для построения стратегии защитника рассматриваемая задача представляется в виде динамической игры Штакельберга. На каждом шаге этой игры взаимодействие противоборствующих сторон формализуется в виде задачи двухуровневого математического программирования. Решение задачи верхнего уровня (задачи атакующего) определяет ребро, атака на которое приводит к максимальному числу незащищённых рёбер после реконструкции защитником размещения охранников. Решение задачи нижнего уровня (задачи защитника) при заданном решении задачи атакующего изменяет размещение охранников с целью минимизировать число незащищённых рёбер. В [11, 12] представлены формулировка данной задачи в виде двухуровневой модели целочисленно-линейного программирования и алгоритм, который отвечает на вопрос, существует ли выигрышная стратегия для защитника, позволяющая защитить заданное вершинное покрытие за заданное конечное число шагов. В данной работе предложена процедура, позволяющая ответить на вопрос, является ли заданное вершинное покрытие вечным, а также проведён анализ сложности задачи, рассматриваемой на одном шаге игры.

1. Формулировка задачи, рассматриваемой на одном шаге игры

Формулировка двухуровневой задачи математического программирования, рассматриваемой на одном шаге игры защитника и атакующего, представлена в [12]. Пусть $G = (I, J)$ — ориентированный граф без петель с множеством вершин I и дуг J . Для произвольного $j \in J$ через $i(j)$ и $k(j)$ обозначим начальную и конечную вершины дуги j соответственно, и будем называть эти вершины соседними. Отметим, что рассмотрение ориентированного графа не уменьшает общности двухуровневой задачи, рассматриваемой на каждом шаге исследуемой игры. В самом деле, в неориентированном графе каждое ребро можно заменить двумя противоположно ориентированными параллельными дугами. Будем считать, что перемещение охранника по дуге происходит от начальной вершины дуги к конечной.

Вершинным покрытием в графе G считаем всякое подмножество множества I , содержащее начальную или конечную вершину любой дуги из множества J .

Для записи задачи используются следующие обозначения:

- $x_{i0} \in \{0, 1\}$ — заданные величины, определяющие начальное для рассматриваемого шага вершинное покрытие: величина x_{i0} равна единице, если в вершине $i \in I$ размещён охранник, и нулю в противном случае;
- $x_i \in \{0, 1\}$ — переменные, задающие размещение охранников по завершении шага: величина x_i равна единице, если в вершине $i \in I$ размещён охранник, и нулю в противном случае;
- $y_j \in \{0, 1\}$ — переменные, обозначающие атакованную на шаге дугу: величина y_j равна единице, если дуга $j \in J$ атакована, и нулю в противном случае;
- $u_j \in \{0, 1\}$ — переменные, обозначающие дуги, которые по завершении шага оказались незащищёнными: величина u_j равна единице, если в начальной и конечной вершине дуги $j \in J$ не размещён охранник, и нулю в противном случае;
- $v_j \in \{0, 1\}$ — переменные, обозначающие перемещение охранников по дугам $j \in J$ на шаге: величина v_j равна единице, если охранник перемещается по дуге $j \in J$ из вершины $i(j)$ в вершину $k(j)$, и нулю в противном случае.

С использованием введённых обозначений модель, формализующая взаимодействие сторон на шаге рассматриваемой игры, записывается как следующая задача двухуровневого математического программирования:

$$\max_{(y_j)} \sum_{j \in J} u_j^*, \quad (1)$$

$$\sum_{j \in J} y_j \leq 1, \quad (2)$$

$$x_{i(j)0} \geq y_j, \quad j \in J, \quad (3)$$

$$x_{i(j)0} + x_{k(j)0} + y_j \leq 2, \quad j \in J, \quad (4)$$

$$y_j \in \{0, 1\}, \quad j \in J, \quad (5)$$

где $(x_j^*), (v_j^*), (u_j^*)$ — оптимальное решение задачи

$$\min_{(x_i), (v_j), (u_j)} \sum_{j \in J} u_j, \quad (6)$$

$$v_j \geq y_j, \quad j \in J, \quad (7)$$

$$x_{i0} \geq \sum_{j: i(j)=i} v_j, \quad i \in I, \quad (8)$$

$$1 - x_{i0} \geq \sum_{j: k(j)=i} v_j - \sum_{j: i(j)=i} v_j, \quad i \in I, \quad (9)$$

$$x_i = x_{i0} + \sum_{j: k(j)=i} v_j - \sum_{j: i(j)=i} v_j, \quad i \in I, \quad (10)$$

$$x_{i(j)} + x_{k(j)} + u_j \geq 1, \quad j \in J, \quad (11)$$

$$x_i, u_j, v_j \in \{0, 1\}, \quad i \in I, j \in J. \quad (12)$$

Данная модель, как и всякая двухуровневая задача, включает задачу верхнего уровня (1)–(5) (задачу атакующего) и задачу нижнего уровня (6)–(12) (задачу защитника).

Целевые функции (1) и (6) обеих задач показывают, сколько дуг осталось без защиты после реконструкции вершинного покрытия на рассматриваемом шаге. Неравенство (2) означает, что на рассматриваемом шаге атакованной может быть только одна дуга, а из (3) и (4) следует, что атакованной может быть только дуга, у которой охранник размещён в начальной вершине и нет охранника в конечной вершине. Ограничения (7)–(9) задачи нижнего уровня формализуют правила возможных перемещений охранников вдоль соответствующих дуг. Условие (7) означает, что если дуга атакована, то охранник перемещается в конечную вершину дуги. Из (8) следует, что если в рассматриваемом на этом шаге вершинном покрытии в вершине i размещён охранник, то его перемещение возможно не более чем по одной дуге, исходящей из этой вершины, а если в вершине i охранника нет, то перемещение охранников по дугам, исходящим из вершины i , невозможно. Аналогично из (9) с учётом условия (8) получаем, что если в рассматриваемом на этом шаге вершинном покрытии в вершине i нет охранника, то $\sum_{j: k(j)=i} v_j \leq 1$. Это означает,

что если охранник не размещён в вершине i , то не более чем по одной дуге, входящей в вершину i , может быть перемещён охранник. Если же в вершине i размещён охранник, то из неравенств (8) и (9) получаем

$$1 \geq \sum_{j: i(j)=i} v_j \geq \sum_{j: k(j)=i} v_j.$$

Это значит, что по одной из дуг, входящих в вершину i , может быть перемещён охранник, но только в случае, когда охранник, размещённый в вершине i , также перемещается. Равенство (10) определяет новое размещение охранников в результате их перемещения на шаге. Неравенство (11) заставляет переменную u_j принимать ненулевое значение, когда в начальной и конечной вершинах дуги j нет охранников.

Допустимым решением двухуровневой задачи на рассматриваемом шаге игры назовём пару $((y_j), (x_i, v_j, u_j))$, где (y_j) — допустимое решение задачи верхнего уровня, а (x_i, v_j, u_j) — оптимальное решение задачи нижнего уровня. Допустимое решение будет оптимальным, если значение целевой функции на этом решении не меньше, чем на других допустимых решениях. Вычисление оптимального решения позволяет проверить текущее вершинное покрытие на устойчивость к атаке. Если оптимальное значение целевой функции равно нулю, то ответ положительный. Если же это значение больше нуля, то существует допустимое решение задачи верхнего уровня (атакуемая дуга), для которого не существует реконструкции вершинного покрытия, приводящей к новому вершинному покрытию.

Как следует из сказанного выше, возможность восстановления (реконструкции) вершинного покрытия после атаки определяется значениями целевых функций задач верхнего и нижнего уровней в двухуровневой модели (1)–(12), рассматриваемой на каждом шаге исследуемой игры.

Вершинное покрытие $C_0 = (x_{i0})$ будем называть *устойчивым к атаке на дугу* $j_0 \in J$, если оптимальное значение целевой функции задачи защитника равно нулю при допустимом решении (y_j) , $y_{j_0} = 1$, задачи атакующего.

2. Вычислительная сложность задачи, рассматриваемой на одном шаге игры

Нетрудно видеть, что задачу проверки устойчивости к атаке на ребро можно переформулировать следующим образом. Пусть заданы граф $G = (I, J)$ с множеством вершин I и рёбер J и множество охранников $C \subset I$, размещённых в некоторых вершинах графа, образующее вершинное покрытие. Задано подвергающееся атаке ребро, на одном из концов которого нет охранника. Существует ли вершинное покрытие, которое

может быть получено из размещения охранников C так, что каждый из охранников либо перемещён в соседнюю вершину, либо остался на месте, при условии, что по атакованному ребру перемещён охранник? Назовём данную задачу *переформулированной задачей одного шага*.

Теорема 1. *Переформулированная задача одного шага NP-полна.*

ДОКАЗАТЕЛЬСТВО. Нетрудно видеть, что данная задача принадлежит классу NP.

Покажем, что она также NP-трудна. Сведём задачу о вершинном покрытии к рассматриваемой задаче. Даны граф G и натуральное число k . К графу G добавим $n - k$ вершин u_i , $i \in 1, n - k$, каждую из которых соединим со всеми вершинами G . Также добавляем вершины s и t , соединив их ребром, а также соединив s с каждой вершиной из U . Обозначим полученный граф через G' . Разместим $n + 1$ охранников во всех вершинах графа G , а также в вершине s . Нетрудно заметить, что множество охранников образует вершинное покрытие в графе G' . Выбираем для атаки ребро между вершинами s и t . Тогда G имеет вершинное покрытие размера не более k тогда и только тогда, когда для G' , указанного начального размещения охранников и выбранного ребра ответ в рассматриваемой задаче «да».

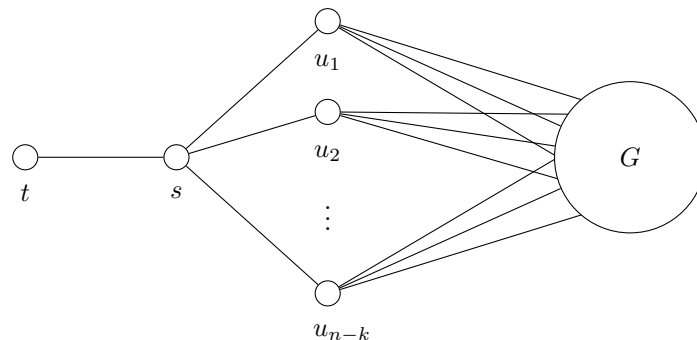


Рис. 1. Граф G'

Предположим, что C — некоторое вершинное покрытие графа G размера k . Перемещаем охранника из s в t . Тогда в G' остаётся переместить охранников из $n - k$ вершин графа G , не входящих в C , в вершины из U . Очевидно, что данное перемещение возможно осуществить таким образом, что каждый охранник перемещается в соседнюю вершину, так как u_i соединена с каждой вершиной в G . Нетрудно видеть, что новое размещение является вершинным покрытием в G' .

Пусть в G' существует новое размещение C' , удовлетворяющее требованиям рассматриваемой задачи. Так как по условиям задачи охранник перемещён вдоль атакованного ребра из s в t , а новое размещение является вершинным покрытием, во всех вершинах из U должны находиться охранники, иначе ребро (u_i, s) для некоторого $u_i \in U$ не будет защищено. Таким образом, в вершинах подграфа G графа G' находится $n + 1 - (n - k) - 1 = k$ охранников. Поскольку C' — вершинное покрытие, эти k охранников находятся в вершинах, образующих вершинное покрытие размера k в подграфе G графа G' . Теорема 1 доказана.

3. Устойчивость вершинного покрытия в игре с бесконечным числом шагов

Пусть задан неориентированный граф $G = (V, E)$.

Для данного графа можно построить эквивалентный ориентированный граф $G = (I, J)$, где $I = V$, а J построено таким образом, что для каждого ребра $(v_1, v_2) \in E$ существует пара противоположно направленных дуг $j_{v_1 v_2}, j_{v_2 v_1} \in J$, выходящих из v_1 в v_2 и наоборот соответственно.

Вершинное покрытие $C_0 = (x_{i0})$ назовём *устойчивым к атаке на ребро* $(v_1, v_2) \in E$, если оптимальное значение целевой функции задачи защитника равно нулю при допустимом решении (y_j) , $y_{j_{v_1 v_2}} + y_{j_{v_2 v_1}} = 1$, задачи атакующего.

Вершинное покрытие назовём *1-устойчивым*, если оптимальное значение целевой функции задачи атакующего равно нулю, иначе это покрытие *неустойчивое*. Таким образом, процедура проверки 1-устойчивости заданного вершинного покрытия C_0 сводится к проверке устойчивости этого покрытия к атаке всякого допустимого для атаки ребра $e \in E$.

Рассмотрим игру, состоящую из $L > 1$ шагов, с начальным вершинным покрытием C_0 . Пусть на шагах $1, 2, \dots, L - 1$ оптимальное значение целевой функции задачи атакующего равно нулю, а на шаге L больше нуля. Это означает, что в результате $L - 1$ шагов получено вершинное покрытие такое, что после атаки некоторого ребра успешная реконструкция этого вершинного покрытия невозможна. Заметим, что отсюда не следует, что при начальном покрытии C_0 у защитника нет выигрышной стратегии. Действительно, полученное к шагу L вершинное покрытие есть результат последовательных изменений вершинных покрытий, построенных на предыдущих шагах. Каждое такое изменение определяется оптимальным решением задачи защитника, которое может быть не единственным. Следовательно, при соответствующем выборе оптимальных решений на шагах, предшествующих шагу L , может быть получено вершинное покрытие, которое будет устойчивым к атаке любого ребра на шаге L .

Вершинное покрытие назовём *L-устойчивым*, если оно 1-устойчивое и для любого допустимого решения задачи атакующего существует оптимальное решение задачи защитника, дающее $(L - 1)$ -устойчивое вершинное покрытие.

Вершинное покрытие будем называть *вечно устойчивым*, или для краткости *вечным*, если оно L -устойчиво при любом L .

Для рассматриваемого графа и некоторого вершинного покрытия C_0 рассмотрим множество $\widehat{C}$ всех покрытий той же мощности, что и C_0 . Так как множество $\widehat{C}$ конечно, можем занумеровать все покрытия числами $0, \dots, |\widehat{C}|$. Обозначим множество номеров через $\widehat{N}$. Тогда C_q будет означать вершинное покрытие из $\widehat{C}$ с номером $q \in \widehat{N}$.

Для пары вершинных покрытий $C_q, C_s \in \widehat{C}$ рассмотрим множество P решений задачи защитника таких, что для каждого решения $\rho \in P$ вектор (x_{i0}^ρ) определяется покрытием C_q , а (x_i^ρ) — покрытием C_s . Для каждого такого решения определим множество

$$E_{qs}^\rho = \{(v_1, v_2) \in E \mid v_{j_{v_1 v_2}}^\rho + v_{j_{v_2 v_1}}^\rho = 1\}.$$

Пусть $E_{qs} = \bigcup_{\rho \in P} E_{qs}^\rho$ — множество всех рёбер, по которым перемещаются охранники хотя бы при некоторой реконструкции покрытия C_q в покрытие C_s . Нетрудно заметить, что $E_{qs} = E_{sq}$. Действительно, если возможно переместить охранников из покрытия C_q по некоторым рёбрам так, что новое размещение является покрытием C_s , то возможно их переместить по тем же рёбрам в обратном направлении.

Для графа G определим взвешенный *граф реконструкций* $H = (N, R)$ следующим образом: $N \subseteq \widehat{N}$, а ребро (q, s) , $q, s \in N$, имеет вес E_{qs} и принадлежит R тогда и только тогда, когда $E_{qs} \neq \emptyset$.

Граф H может содержать петли. Рассмотрим вершинное покрытие C_q и рёбра из E такие, что обе вершины каждого из этих рёбер содержатся в C_q . По условию задачи такие рёбра не могут быть подвергнуты атаке, поскольку они защищены с обеих сторон. Таким образом, петля (q, q) в графе H имеет вес E_{qq} , и множество E_{qq} состоит из рёбер, которые не могут быть атакованы.

Вершину $q \in N$ будем называть *полной*, если $\bigcup_{s \in N} E_{qs} = E$. Граф реконструкций $H = (N, R)$ будем называть *завершённым*, если любая вершина $q \in N$ полная.

Тогда для проверки того, является ли данное вершинное покрытие вечным, можно воспользоваться следующей леммой.

Лемма 1. *Если граф реконструкций $H = (N, R)$ завершённый, то для любого $q \in N$ вершинное покрытие C_q вечное.*

ДОКАЗАТЕЛЬСТВО. Предположим, что для некоторого $s \in N$ вершинное покрытие C_s не вечное. Тогда существует конечная последовательность атак $C_s \xrightarrow{e_{k_0}} C_{s_1} \xrightarrow{e_{k_1}} \dots \xrightarrow{e_{k_{L-1}}} C_{s_L} \xrightarrow{e_{k_L}}$, где $s_i \in N$ для любого $i \in \overline{1, L}$, а при атаке на ребро e_{k_L} для покрытия C_{s_L} не существует решения соответствующей задачи защитника. В то же время, так как $s_L \in N$ и граф реконструкций по условию завершённый, s_L является полной вершиной, а следовательно, найдётся вершина $p \in N$ такая, что $e_{k_L} \in E_{s_L p}$. Вершинное покрытие C_p соответствует решению задачи защитника при атаке на ребро e_{k_L} покрытия C_{s_L} ; противоречие. Лемма 1 доказана.

Лемма 2. Если вершинное покрытие C_q , $q \in \widehat{N}$, графа G вечное, то для G существует завершённый граф реконструкций $H = (N, R)$, при этом $q \in N$.

ДОКАЗАТЕЛЬСТВО. Рассмотрим $\widehat{H} = (\widehat{N}, \widehat{R})$ такой, что множество $\widehat{R}$ содержит в себе все возможные рёбра между всеми вершинами из $\widehat{N}$. Рассмотрим его компоненту связности $H = (N, R)$, $q \in N$. Покажем, что H завершённый.

Допустим, что H не завершённый. Тогда найдётся вершина $s_L \in N$ такая, что C_{s_L} — неполная вершина. Это означает, что существует ребро $e' \in E$ такое, что $e' \notin \bigcup_p E_{s_L p}$, где $p \in N$. Таким образом, при атаке на ребро e' для покрытия C_{s_L} не существует решения соответствующей задачи защитника, которое бы соответствовало некоторому покрытию, номер которого содержался бы в N . Однако, так как по условию $\widehat{R}$ содержит в себе все возможные рёбра между всеми вершинами из $\widehat{N}$, R также содержит в себе все возможные рёбра между всеми вершинами из N . Отсюда следует, что рассматриваемая задача защитника не имеет решения.

Так как H связный, существует путь между q и s_L , а значит, существует последовательность атак $C_q \xrightarrow{e_{k_0}} C_{s_1} \xrightarrow{e_{k_1}} \dots \xrightarrow{e_{k_{L-1}}} C_{s_L} \xrightarrow{e'}$, где $s_i \in N$ для любого $i \in \overline{0, L}$, приводящая к победе атакующего. Значит, вершинное покрытие C_q не будет вечным; противоречие. Лемма 2 доказана.

Таким образом, если для заданного графа удастся построить завершённый граф реконструкций такой, что проверяемое вершинное покрытие будет одной из вершин графа реконструкций, то данное вершинное покрытие будет вечным.

Опишем процедуру построения завершённого графа реконструкций $H = (N, R)$, которая состоит из определённого числа однотипных итераций. На каждой итерации проверяемое вершинное покрытие C_q , $q \in N$, исследуется на устойчивость к атакам на допустимые для атаки рёбра. Также имеется список T номеров запрещённых покрытий, которые

не могут быть включены в граф реконструкций, а также список A номеров вершинных покрытий, проверка которых начата на предыдущих итерациях, но не завершена. Элементы списка A упорядочены по порядку вхождения в список.

На первой итерации рассматривается граф реконструкций, состоящий из одной вершины, соответствующей исследуемому вершинному покрытию C_0 . Список A состоит из единственного элемента — номера вершинного покрытия C_0 , а список T пуст.

На каждой итерации, если $A \neq \emptyset$, рассматривается вершинное покрытие C_q , где номер $q \in N$ — последний в списке A . При этом номер q удаляется из A . Рассматриваемое вершинное покрытие C_q проверяется на устойчивость к атакам на рёбра из множества E_q^* , состоящее из допустимых для атаки рёбер графа G за исключением рёбер, входящих в множества E_{qs} для $s \in N$.

Такая проверка состоит из определённого числа однотипных шагов, на каждом из которых рассматривается ребро $(v_1, v_2) \in E_q^*$. Для (v_1, v_2) рассматривается задача защитника при условии $x_{i0} = 1$, если $i \in C_q$, $y_{jv_1v_2} + y_{jv_2v_1} = 1$, и дополнительных ограничениях, задаваемых вершинными покрытиями из списка T и отсекающих решения, приводящие к запрещённым покрытиям. Если $s \in T$ и $x'_i = 1$ для $i \in C_s$, то соответствующее ограничение имеет вид

$$\sum_{i \in I} x'_i x_i \leq \sum_{i \in I} x'_i - 1.$$

Пусть для ребра (v_1, v_2) оптимальное значение целевой функции задачи защитника больше нуля, а следовательно, успешная реконструкция вершинного покрытия C_q невозможна. Если C_q — исследуемое вершинное покрытие C_0 , то процедура завершается с отрицательным результатом. Иначе номер q добавляется в список T , вершина с номером q удаляется из графа H , все соседние с ней вершины добавляются в A , и начинается следующая итерация.

Пусть при атаке на рассматриваемое ребро (v_1, v_2) вершинное покрытие C_q оказывается устойчивым и получено вершинное покрытие C_s .

Если $s \notin N$, то в H добавляется вершина с номером s , соответствующая вершинному покрытию C_s , и ребро (q, s) с весом E_{qs} . Множество E_{qs} включает рёбра графа G , по которым перемещаются охранники при реконструкции вершинного покрытия C_q в вершинное покрытие C_s . Кроме того, в граф H добавляется петля (s, s) с весом E_{ss} , включающим рёбра, запрещённые для атаки вершинного покрытия C_s . Далее ребро (v_1, v_2) удаляется из множества E_q^* и, если $E_{qs} \neq \emptyset$, вершина q возвращается в список A . После этого вершина s добавляется в список A , и начинается следующая итерация.

Если полученное вершинное покрытие C_s такое, что $s \in N$, то вес E_{qs} ребра (q, s) графа H пополняется рёбрами графа G , по которым перемещаются охранники при реконструкции вершинного покрытия C_q в C_s . Далее ребро (v_1, v_2) удаляется из множества E_q^* . Если $E_q^* \neq \emptyset$, то начинается следующий шаг исследуемой итерации, иначе начинается следующая итерация.

Процедура завершается с положительным результатом тогда, когда на очередной итерации список A пуст, а все шаги по проверке текущего вершинного покрытия завершены с положительным результатом.

Если процедура завершится с положительным результатом, то для данного графа удалось построить завершённый граф реконструкций, одной из вершин которого является C_0 . Иначе все покрытия, получаемые при реконструкции в ответ на атаку на некоторое ребро вершинного покрытия C_0 , содержатся в списке T , что означает существование некоторой последовательности атак, приводящей к победе атакующего.

3.1. Показательный пример. Приведём пример работы алгоритма построения графа реконструкций $H = (N, R)$ для данного вершинного покрытия C_0 графа G . На рис. 2 представлен граф G , для которого задано вершинное покрытие $C_0 = \{a, b, c\}$.

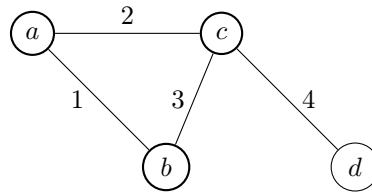


Рис. 2. Исходный граф G и покрытие C_0

На итерации 1 алгоритма граф H имеет единственную вершину, соответствующую вершинному покрытию $C_0 = \{a, b, c\}$. Эта вершина имеет петлю $(0, 0)$ с весом $E_{0,0} = \{1, 2, 3\}$, список $A = \{0\}$. Начальный вид графа H представлен на рис. 3.

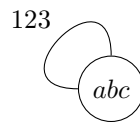


Рис. 3. Начальное состояние графа реконструкций

Сначала рассматривается вершина с номером $q = 0$ и соответствующее вершинное покрытие $C_0 = \{a, b, c\}$ проверяется на устойчивость

к атакам на рёбра из множества $E_0^* = \{4\}$. При этом номер этого вершинного покрытия удаляется из списка A .

На шаге 1 итерации 1 рассматривается задача защитника для покрытия C_0 при атаке на ребро H . Результатом решения этой задачи является вершинное покрытие $C_1 = \{a, b, d\}$, а при реконструкции C_0 в C_1 охранник перемещается только по ребру 4.

Поскольку $1 \notin N$, в множество N вершин графа H добавляется номер вершинного покрытия $C_1 = \{a, b, d\}$, а в множество R рёбер графа H — ребро $(0, 1)$ с весом $E_{0,1} = \{4\}$. Кроме того, в множество R добавляется петля $(1, 1)$ с весом $E_{1,1} = \{1\}$. После этого номер вершинного покрытия $C_1 = \{a, b, d\}$ добавляется в список A .

После шага 1 список J_0 пуст, и итерация 1 завершается. Полученный в результате итерации 1 граф H представлен на рис. 4.

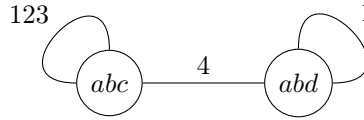


Рис. 4. Результат итерации 1

На итерации 2 имеем $A = \{1\}$, поэтому рассматриваем вершину с номером $q = 1$, соответствующую вершинному покрытию $C_1 = \{a, b, d\}$, для которого $E_1^* = \{2, 3\}$, и удаляем её номер из списка A .

На шаге 1 итерации 2 решается задача защитника для покрытия $C_1 = \{a, b, d\}$ при атаке на ребро 2. Одним из возможных решений задачи является вершинное покрытие $C_2 = \{a, c, d\}$, при построении которого охранники перемещались по рёбрам 1 и 2.

Поскольку $2 \notin N$, в множество N вершин графа H добавляется номер вершинного покрытия $C_2 = \{a, c, d\}$, а в множество R рёбер графа H — ребро $(1, 2)$ с весом $E_{1,2} = \{1, 2\}$. Кроме того, в множество R добавляется петля $(2, 2)$ с весом $E_{2,2} = \{2, 4\}$. После этого номер вершинного покрытия $C_2 = \{a, c, d\}$ добавляется в список A . Полученный в результате шага 1 итерации 2 граф H представлен на рис. 5.

На шаге 2 итерации 2 исследуется устойчивость вершинного покрытия $C_1 = \{a, b, d\}$ при атаке на ребро 3. Одним из возможных решений задачи защитника является вершинное покрытие $C_2 = \{a, c, d\}$, при построении которого охранник перемещался по ребру 3.

Поскольку $2 \in N$, множество вершин графа H не увеличивается, новые рёбра не появляются, но вес ребра $(1, 2)$ становится равным $E_{1,2} = \{1, 2, 3\}$.

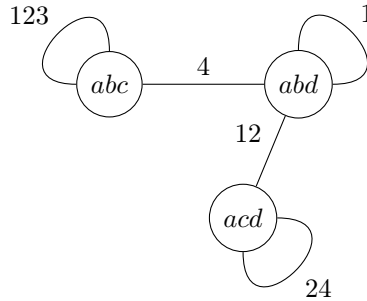


Рис. 5. Результат шага 1 итерации 2

После шага 2 список J_0 пуст, и итерация 2 завершается. Полученный в результате итерации 2 граф H представлен на рис. 6.

На итерации 3 $A = \{2\}$, поэтому исследуем вершинное покрытие с номером $q = 2$. Исключаем номер 2 из списка A , а поскольку $E_2^* = \emptyset$, завершаем итерацию. Так как $A = \emptyset$, завершаем алгоритм. Полученный граф реконструкций H , изображённый на рис. 6, полный, а одна из его вершин является номером исследуемого вершинного покрытия C_0 .

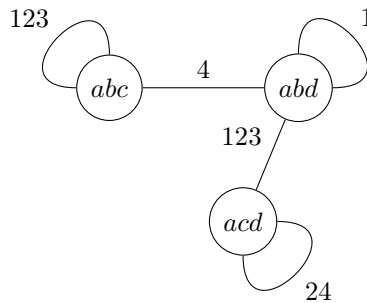


Рис. 6. Результат итерации 2

4. Вычислительные эксперименты

В экспериментах исследованы вычислительные возможности предложенного алгоритма проверки устойчивости вершинных покрытий, а также влияние на его производительность некоторых используемых процедур. Расчёты проведены на компьютере с четырёхъядерным процессором

3,3 ГГц и 32 ГБ оперативной памяти. Алгоритмы написаны на языке программирования C#, а для поиска оптимальных решений возникающих задач целочисленного линейного программирования использовался решатель Gurobi.

В экспериментах использованы случайно генерируемые связные графы. Поскольку нахождение минимального покрытия — задача труднорешаемая, в экспериментах используются покрытия, у которых ни одно из собственных подмножеств не остаётся покрытием. Алгоритмы генерации графов и вершинных покрытий приведены в [12].

4.1. Устойчивость вершинного покрытия в игре с бесконечным числом шагов. В ходе экспериментов, описанных в данном пункте, сгенерировано несколько тысяч покрытий для различных графов с 4, 8, 16 и 32 вершинами способом, описанным в предыдущих пунктах.

Структура графа реконструкций. В данном разделе представлены результаты вычислительного эксперимента, позволяющего изучить зависимость структуры получаемых графов реконструкций от размерности исходного графа. В табл. 1 показана зависимость числа вершин и рёбер в получаемом графе реконструкций от числа вершин n исходного графа.

Таблица 1

Зависимость числа вершин и рёбер в графе H от числа вершин n исходного графа

n	hv_0	hv_1	he_1
4	1,43	2,89	1,15
8	2,26	8,1	19
16	8,02	41,9	266
32	64,1	624	9954

В ходе эксперимента для каждого числа вершин были рассчитаны величины hv — среднее число вершин графа реконструкций, полученного по окончании работы алгоритма, и he — среднее число рёбер в графе реконструкций. Результаты были разделены для покрытий, которые оказались вечными (индекс 1), и для покрытий, для которых нашлась последовательность атак, приводящая к победе атакующего (индекс 0). Величина he_0 не представлена в табл. 1, так как в случае, когда для проверяемого покрытия удалось найти последовательность атак, приводящую к победе атакующего, результатом работы алгоритма является граф реконструкций, в котором отсутствуют рёбра.

Из табл. 1 видно, что уже для графов с 32 вершинами графы реконструкций имеют несколько сотен вершин. В случае, если проверяемое вершинное покрытие вечное, граф реконструкций будет иметь несколько тысяч рёбер. Таким образом, уже для графов средней размерности доказательство вечности покрытий имеет высокую сложность по памяти, так как требуется хранить всю структуру графа реконструкций.

Доля вечных покрытий. Заполненностью графа G с n вершинами и m рёбрами назовём величину

$$f = \frac{2(m - n + 1)}{(n - 1)(n - 2)}.$$

Заметим, что $f = 0$ для деревьев и $f = 1$ для полного графа.

Приведём результаты вычислительного эксперимента, демонстрирующего один из вариантов применения алгоритма по проверке вечной устойчивости вершинного покрытия. Его цель состоит в исследовании зависимости доли вечных неприводимых вершинных покрытий от заполненности графа. Случайность генерируемых графов и недетерминированность полученных неприводимых вершинных покрытий позволяет утверждать, что полученные результаты достаточно общие при достаточном размере выборки.

На рис. 7 представлена зависимость доли вечных неприводимых покрытий от заполненности графа для графов с 8, 16 и 32 вершинами.

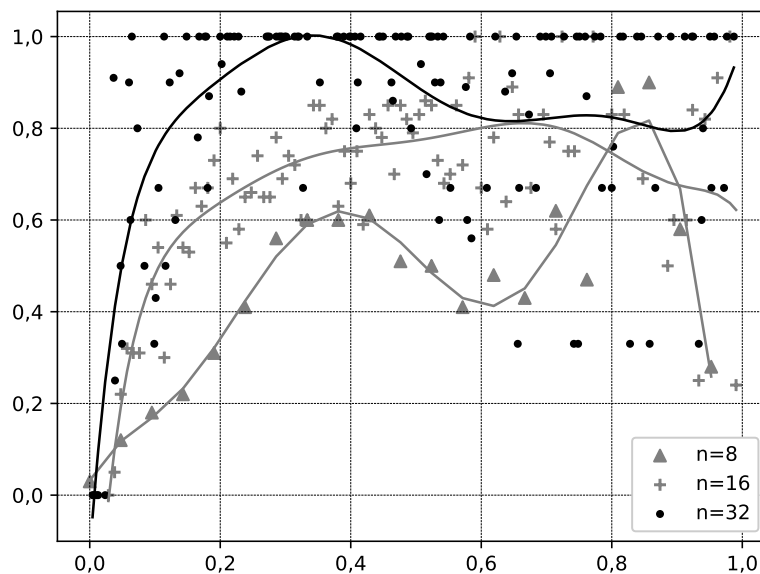


Рис. 7. Зависимость доли вечных покрытий от заполненности графа

В ходе эксперимента для каждого числа вершин построена своя зависимость. Для каждого рассматриваемого графа рассчитано значение его заполненности, затем среди покрытий, соответствующих графам с одинаковым числом заполненности, рассчитана доля вечных. Кроме того, для большей наглядности для каждой зависимости было построено приближение полиномом 8-й степени.

Из рис. 7 видно, что чем выше число вершин в графе, тем в среднем большая часть неприводимых вершинных покрытий является вечными. Можно также сделать вывод, что увеличение числа заполненности графа чаще влечёт увеличение доли вечных покрытий.

Время работы алгоритма. Проведённые вычислительные эксперименты позволяют сделать некоторые выводы о времени работы алгоритма в зависимости от числа вершин графа.

Таблица 2

Зависимость времени работы алгоритма и числа запусков оптимизации модели задачи защитника от числа вершин n

n	t_0 , с	t_1 , с	mc_0	mc_1
4	0,01	0,01	2	4
8	0,06	0,14	6	31
16	1,93	2,89	56	427
32	466	277	644	16 133

В табл. 2 представлена зависимость времени работы алгоритма при проверке вечности вершинных покрытий графов соответствующей размерности. В ходе эксперимента для каждого числа вершин рассчитаны величины t — времени работы алгоритма в секундах и mc — число запусков оптимизации модели задачи защитника. Результаты выписаны отдельно для покрытий, которые оказались вечными (индекс 1), и для покрытий, для которых нашлась последовательность атак, приводящая к победе атакующего (индекс 0).

Как видим, время работы алгоритма практически не отличается для обоих классов вершинных покрытий, т. е. доказательство вечности вершинного покрытия занимает почти столько же времени, сколько поиск последовательности атак, которая приводит к победе атакующего. Также можно видеть, что время работы алгоритма уже для графов с 32 вершинами равно примерно 5 мин.

При этом число запусков оптимизации модели задачи защитника для покрытий, оказавшихся вечными, значительно больше, чем для покрытий, для которых нашлась последовательность атак, приводящая к победе атакующего. Это можно объяснить тем, что время, затрачиваемое

на поиск оптимального решения задачи защитника с дополнительными ограничениями, возрастает с ростом числа запрещённых покрытий.

Заключение

В работе предложена процедура, позволяющая ответить на вопрос, является ли заданное вершинное покрытие вечным. В то же время, эта процедура позволяет находить сколь угодно длинные приводящие к победе атакующего последовательности атак, если они существуют.

Идея процедуры заключается в рекурсивной проверке устойчивости вершинных покрытий, полученных в результате решения задач защитника. Для построения стратегии защитника задача о вечном вершинном покрытии представляется как динамическая игра Штакельберга, на каждом шаге которой взаимодействие противоборствующих сторон формализуется как двухуровневая задача математического программирования. Предложенная процедура протестирована на различных случайно сгенерированных графах и вершинных покрытиях. Вычислительные эксперименты продемонстрировали способность данной процедуры находить решения за адекватное время для графов средней размерности.

В работе представлена реализация построения графа реконструкций вершинных покрытий на основе рекурсивного решения задач защитника. При данном подходе не гарантируется, что данный граф получится минимального размера, тем самым существует пространство для ускорения данной процедуры.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF-2026-0021). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него нет конфликта интересов.

Литература

1. Klostermeyer W. F., Mynhardt C. M. Protecting a graph with mobile guards // Appl. Anal. Discrete Math. 2016. V. 10, No. 1. P. 1–29.
2. Klostermeyer W. F., Mynhardt C. M. Edge protection in graphs // Australas. J. Comb. 2009. V. 45. P. 235–250.
3. Fomin F. V., Gaspers S., Golovach P. A., Kratsch D., Saurabh S. Parameterized algorithm for eternal vertex cover // Inf. Process. Lett. 2010. V. 110, No. 16. P. 702–706.

4. **Babu J., Misra N., Nanoti S. G.** Eternal vertex cover in bipartite graphs // Computer science — Theory and applications. Proc. 17th Int. Comp. Sci. Symp. in Russia (St. Petersburg, Russia, June 29 – July 1, 2022). Cham: Springer, 2022. P. 64–76. (Lect. Notes Comput. Sci.; V. 13296).
5. **Babu J., Prabhakaran V.** A new lower bound for the eternal vertex cover number of graphs // J. Comb. Optim. 2021. V. 44, No. 4. P. 2482–2498.
6. **Babu J., Chandran L. S., Francis M., Prabhakaran V., Rajendraprasad D., Warriar N. J.** On graphs whose eternal vertex cover number and vertex cover number coincide // Discrete Appl. Math. 2022. V. 319. P. 171–182.
7. **Babu J., Prabhakaran V., Sharma A.** A substructure based lower bound for eternal vertex cover number // Theor. Comput. Sci. 2021. V. 890. P. 87–104.
8. **Paul K., Pandey A.** Some algorithmic results for eternal vertex cover problem in graphs // WALCOM: Algorithms and computation. Proc. 17th Int. Conf. and Workshops (Hsinchu, Taiwan, Mar. 22–24, 2023). Cham: Springer, 2023. P. 242–253. (Lect. Notes Comput. Sci.; V. 13973).
9. **Araki H., Fujito T., Inoue S.** On the eternal vertex cover numbers of generalized trees // IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 2015. V. E98-A, No. 6. P. 1153–1160.
10. **Klostermeyer W. F., Mynhardt C. M.** Graphs with equal eternal vertex cover and eternal domination numbers // Discrete Math. 2011. V. 311, No. 14. P. 1371–1379.
11. **Beresnev V. L., Melnikov A. A., Utyupin S. Yu.** Representation of the eternal vertex cover problem as a dynamic Stackelberg game // Optimization and applications. Rev. Sel. Pap. 14th Int. Conf. OPTIMA 2023 (Petrovac, Montenegro, Sept. 18–22, 2023). Cham: Springer, 2023. P. 3–13. (Lect. Notes Comput. Sci.; V. 14395).
12. **Береснев В. Л., Мельников А. А., Утюпин С. Ю.** Устойчивость вершинных покрытий в игре с конечным числом шагов // Дискрет. анализ и исслед. операций. 2024. Т. 31, № 2. С. 28–45.

Утюпин Степан Юрьевич

Статья поступила

8 апреля 2026 г.

После доработки —

11 мая 2026 г.

Принята к публикации

22 мая 2026 г.

STABILITY OF VERTEX COVERS
IN THE ETERNAL VERTEX COVER PROBLEM

S. Yu. Utyupin

Sobolev Institute of Mathematics,
4 Acad. Koptug Avenue, 630090 Novosibirsk Russia
E-mail: stepan.utyupin@gmail.com

Abstract. The eternal vertex cover problem is a modification of the graph vertex cover problem and can be viewed as a dynamic two-player (attacker and defender) game with an infinite number of moves. To construct defender's strategy, the problem is represented as a dynamic game, where at each step the interaction between the opposing players is formalized as a two-level mathematical programming problem. This study proposes a procedure that allows us to determine whether a given vertex cover is eternal or not. The idea of the procedure is based on recursive verification of stability of vertex covers obtained by solving defender's problems and constructing an additional graph of covers and transitions. Tab. 2, illustr. 7, bibliogr. 12.

Keywords: eternal vertex cover, graph edge protection, stability check algorithm, dynamic Stackelberg game.

References

1. **W. F. Klostermeyer** and **C. M. Mynhardt**, Protecting a graph with mobile guards, *Appl. Anal. Discrete Math.* **10** (1), 1–29 (2016).
2. **W. F. Klostermeyer** and **C. M. Mynhardt**, Edge protection in graphs, *Australas. J. Comb.* **45**, 235–250 (2009).
3. **F. V. Fomin**, **S. Gaspers**, **P. A. Golovach**, **D. Kratsch**, and **S. Saurabh**, Parameterized algorithm for eternal vertex cover, *Inf. Process. Lett.* **110** (16), 702–706 (2010).
4. **J. Babu**, **N. Misra**, and **S. G. Nanoti**, Eternal vertex cover in bipartite graphs, in *Computer Science — Theory and Applications*, Proc. 17th Int. Comp. Sci. Symp. in Russia (St. Petersburg, Russia, June 29 – July 1, 2022) (Springer, Cham, 2022), pp. 64–76 (Lect. Notes Comput. Sci., Vol. 13296).

English transl.: *Journal of Applied and Industrial Mathematics* **20** (2) (2026).

5. **J. Babu** and **V. Prabhakaran**, A new lower bound for the eternal vertex cover number of graphs, *J. Comb. Optim.* **44** (4), 2482–2498 (2021).
6. **J. Babu**, **L. S. Chandran**, **M. Francis**, **V. Prabhakaran**, **D. Rajendraprasad**, and **N. J. Warriar**, On graphs whose eternal vertex cover number and vertex cover number coincide, *Discrete Appl. Math.* **319**, 171–182 (2022).
7. **J. Babu**, **V. Prabhakaran**, and **A. Sharma**, A substructure based lower bound for eternal vertex cover number, *Theor. Comput. Sci.* **890**, 87–104 (2021).
8. **K. Paul** and **A. Pandey**, Some algorithmic results for eternal vertex cover problem in graphs, in *WALCOM: Algorithms and Computation*, Proc. 17th Int. Conf. and Workshops (Hsinchu, Taiwan, Mar. 22–24, 2023) (Springer, Cham, 2023), pp. 242–253 (Lect. Notes Comput. Sci., Vol. 13973).
9. **H. Araki**, **T. Fujito**, and **S. Inoue**, On the eternal vertex cover numbers of generalized trees, *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E98-A** (6), 1153–1160 (2015).
10. **W. F. Klostermeyer** and **C. M. Mynhardt**, Graphs with equal eternal vertex cover and eternal domination numbers, *Discrete Math.* **311** (14), 1371–1379 (2011).
11. **V. L. Beresnev**, **A. A. Melnikov**, and **S. Yu. Utyupin**, Representation of the eternal vertex cover problem as a dynamic Stackelberg game, in *Optimization and Applications*, Rev. Sel. Pap. 14th Int. Conf. OPTIMA 2023 (Petrovac, Montenegro, Sept. 18–22, 2023) (Springer, Cham, 2023), pp. 3–13 (Lect. Notes Comput. Sci., Vol. 14395).
12. **V. L. Beresnev**, **A. A. Melnikov**, and **S. Yu. Utyupin**, Stability of vertex covers in a game with finitely many steps, *Diskretn. Anal. Issled. Oper.* **31** (2), 28–45 (2024) [Russian] [*J. Appl. Ind. Math.* **18** (2), 206–215 (2024)].

Stepan Yu. Utyupin

Received April 8, 2026

Revised May 11, 2026

Accepted May 22, 2026