Вестник Сыктывкарского университета.
Сер.1.Вып.4.2001

УДК 519.717

Поисковое и вершинно-поисковое число двойственных графов [1]

П.А. Головач, Ф.В. Фомин

В работе исследуются связи между поисковым и вершинно-поисковым числом двойственных графов. Наряду с установленными соотношениями формулируются и обсуждаются некоторые гипотезы, касающиеся этих связей.

Понятие поискового числа графа было введено Т.Д. Парсонсом (см. [8] и, независимо от него, Н.Н. Петровым ([11]), которые рассмотрели следующую задачу поиска. На множестве, представляющем собой связный топологический граф, находятся убегающий и группа преследователей, которые могут перемещаться по графу по непрерывным траекториям. Считается, что преследователи не видят убегающего и не располагают никакими сведениями о его положении и передвижениях. Целью преследователей является нахождение (поимка) убегающего. Убегающий считается найденым, если он оказывается в одной точке с каким-либо преследователем. Ставится задача определения минимального числа преследователей, достаточного для успешного завершения поиска при любом поведении убегающего. Эта величина и была названа поисковым числом графа.

Задача поиска привлекла к себе внимание и активно исследовалась. Выяснилось, в частности, что она имеет конкретное прикладное значение, поскольку поиск убегающего на графе можно трактовать как поиск компьютерного вируса в сети. Кроме того, оказалось, что поисковое число графа тесно связано с целой серией других численных характеристик графов, представляющих большой интерес с прикладной и теоретической точек зрения. В первую очередь следует упомянуть

[1]Работа выполнена при поддержке гранта РФФИ №01-01-00235

характеристики графов, определяемые с помощью оптимальных (по различным критериям) нумераций вершин. Это ширина ленты графа, ширина разреза, профиль, величина вершинного разделения и другие. Также выявилась связь поискового числа и таких интересных характеристик графа, как путевая и древесная ширина. Более подробную информацию о всех этих результатах можно найти в обзоре [1].

Наряду с описанной задачей поиска на графах рассматривались и другие задачи. В работах [4, 5] была введена задача, которая, по сути дела, отличалась условием успешного (с точки зрения преследователей) завершения поиска. Поиск считается успешно завершенным, если убегающий оказывается на ребре, на обоих концах которого находятся преследователи. Минимальная численность команды преследователей, достаточная для успешного завершения поиска в этом варианте задачи, называется вершинно поисковым числом графа.

В настоящей работе рассматриваются некоторые соотношения между поисковым числом планарного графа и вершинно-поисковым числом двойственного к нему графа. В диссертационной работе [10] был доказан следующий факт о графах правильных многогранников (см. рис. 1).

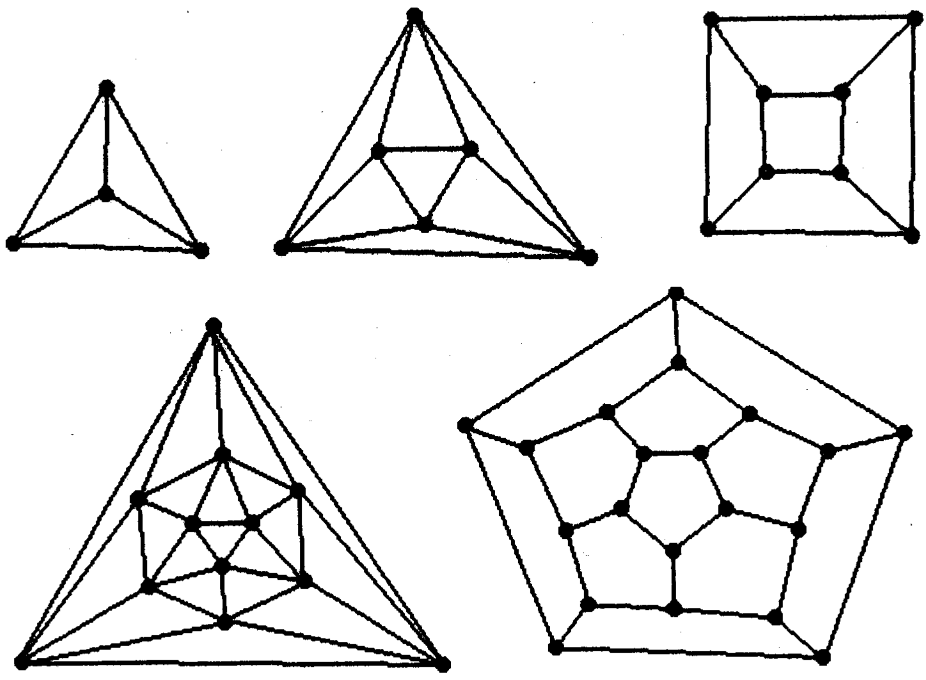

Рис. 1. Графы правильных многогранников: тетраэдра, октаэдра, куба, икосаэдра, додекаэдра.

Оказывается, поисковые числа графов тетраэдра, куба и додекаэдра совпадают с вершинно-поисковыми числами двойственных к ним графов, т.е. графов тетраэдра, октаэдра и икосаэдра. В настоящей работе мы покажем, что данное совпадение инвариантов не случайно, а является следствием более глубоких результатов о поисковом числе планарных графов.

1. Основные понятия

В данном разделе вводятся основные определения, используемые в работе. Отметим сразу, что мы используем стандартные обозначения теории графов, с которыми можно познакомиться, например, в [12].

Мы не будем приводить формальной постановки исходных «непрерывных» задач поиска, описанных во вводной части, поскольку они, как было показано в [9, 10], в конечном итоге, сводятся к более простым дискретным задачам. Можно считать, что преследователи по очереди переходят из вершины в вершину графа. Также удобно предполагать, что преследователи могут покидать граф и возвращаться на него. Для того, чтобы выяснить успешность завершения поиска, проще не рассматривать всевозможные траектории убегающего, а следить за тем, как меняется множество точек, в которых, в принципе, может находится убегающий, а также множество точек, в которых его не может быть. Таким образом, поиск можно представить как процесс «очистки» графа от «распылённого» убегающего. Соответственно, те вершины и ребра, на которых может находится убегающий, называются «загрязненными», а те, где его не может быть, — «очищенными». В итоге получается следующая задача, которую будем называть задачей поиска (1).

Пусть $G = (V, E)$ — связный граф. Определим поиск на графе G как конечную последовательность ходов, называемую ***программой поиска.*** Каждым ходом можно осуществить одно из трех действий:

1. поместить преследователя в некоторую вершину графа;
2. перевести одного из преследователей, находящегося на графе, из одной вершины в смежную ей по соединяющему их ребру;
3. удалить одного из преследователей с графа.

Считается, что в начальный момент на графе нет преследователей и все вершины и ребра являются загрязненными. Загрязненная вершина графа становится очищенной после некоторого хода, если в ней после этого хода оказывается преследователь. Загрязненное ребро (u, v)

становится очищенным после хода, если по нему перемещается преследователь, причем если преследователь перемещается из вершины u в вершину v, то либо в момент выполнения хода в вершине u должен находиться еще один преследователь, либо к моменту выполнения хода все остальные ребра, инцидентные u, должны быть очищенными. Очищенная вершина вновь становится загрязненной, если она оказывается инцидентной загрязненному ребру в момент, когда в ней нет преследователя. Соответственно, очищенное ребро становится загрязненным, если оно оказывается инцидентным загрязненной вершине.

Программа поиска называется *выигрывающей*, если после последнего хода все вершины и ребра графа являются очищенными. Если после каждого хода на графе находится не более k преследователей, то говорят, что для осуществления программы достаточно k преследователей. ***Поисковым числом*** графа G называется минимальное натуральное число k такое, что на графе G существует выигрывающая программа поиска, для осуществления которой достаточно k преследователей. Поисковое число графа G обозначается через $s(G)$.

Отметим важный факт (см. [6, 2]), который состоит в том, что если поиск можно завершить успешно, то его удается провести так, чтобы однажды очищенные вершины и ребра не становились бы загрязненными, то есть монотонно увеличивая очищенную область графа. Соответствующие программы поиска называются монотонными. Мы дополнительно будем предполагать, что при осуществлении монотонной программы не разрешается переводить преследователя из вершины в вершину по уже очищенному ребру. Выполнения этого условия легко добиться, заменяя переходы по ребрам парой ходов: удалением преследователя из одного конца ребра и помещением преследователя в другой.

Совершенно аналогично определяется задача поиска, с помощью которой вводится *вершинно поисковое* число. Эту задачу мы будем называть задачей поиска (2).

Пусть G — связный граф. Определим теперь программу поиска на графе G как конечную последовательность ходов двух видов. Каждым ходом можно осуществить следующие действия:

1. поместить преследователя в некоторую вершину графа;

2. удалить одного из преследователей с графа.

Считается, что в начальный момент на графе нет преследователей и все вершины и ребра являются загрязненными. Загрязненная вершина графа становится очищенной после некоторого хода, если в ней после этого хода оказывается преследователь. Загрязненное ребро становится очищенным после того, как в обоих его концах после очередного хода оказываются преследователи. Правила, по которым очищенные вершины и ребра могут вновь стать загрязненными, остаются прежними. *Вершинно-поисковым* числом графа G называется минимальное натуральное число k такое, что на графе G существует выигрывающая программа поиска, для осуществления которой достаточно k преследователей. Обозначается вершинно-поисковое число графа G через $ns(G)$. Так же, как в случае поискового числа, можно рассматривать монотонные программы поиска. Поисковое число графа тесно связано с численными характеристиками графов, определяемых с помощью оптимальных нумераций вершин. В данной работе нам понадобится одна из них.

Пусть $G = (V, E)$ — граф с n вершинами. Нумерацией вершин графа G называется взаимно однозначное отображение $f\colon V \to \{1, 2, \ldots, n\}$. Для графа G и нумерации f определим величину

$$cw(G, f) := \max_{1 \le i \le n-1} |\{(u, v) \in E\colon f(u) \le i,\ f(v) > i\}|.$$

Шириной разреза (cutwidth) графа G называется величина

$$cw(G) := \min\{cw(G, f)\colon f \text{ -- нумерация вершин } G\}.$$

Известно (см. [7]), что если у графа G все вершины имеют степени, не превосходящие трех, то $s(G) = cw(G)$.

Напомним в заключении этой части определение двойственного графа. Пусть G — планарный граф. Рассмотрим укладку этого графа на плоскости. Обозначим через F множество граней этой укладки. Пусть G^* — граф с множеством вершин F, две вершины которого (то есть грани G) являются смежными тогда и только тогда, когда грани являются соседними, причем число ребер, соединяющих эти вершины равно числу ребер, разделяющих грани. Граф G^* называется графом двойственным (или геометрически двойственным) графу G. Очевидно, что G^* также является планарным графом. Очевидно также, что у графа может быть несколько двойственных ему графов (в зависимости от выбора укладки). Заметим кроме того, что граф G является графом, двойственным графу G^*.

2. Поисковое число планарного графа и вершинно-поисковое число двойственного к нему графа

В этом разделе приводятся основные результаты работы. Оценим вначале поисковое число планарного графа с небольшими степенями вершин снизу через вершинно-поисковое число двойственного графа.

Теорема 1. *Пусть G — двусвязный планарный граф, все степени вершин которого не превосходят 3. Тогда $s(G) \geq ns(G^*)$.*

Доказательство. Если граф G имеет не более двух вершин, то утверждение теоремы очевидно. Предположим, что в G более двух вершин.

Рассмотрим произвольную монотонную выигрывающую программу-поиска P на графе G (в смысле задачи поиска (1)), для осуществления которой достаточно $s(G)$ преследователей. Построим, последовательно просматривая ходы этой программы, программу поиска P^* на графе G^* (уже в смысле задачи поиска (2)). Если очередным ходом программы P преследователь либо помещается на граф, либо удаляется с графа, то мы сразу переходим к следующему ходу этой программы. Предположим, что очередным ходом программы P преследователь переводится из вершины u в вершину v по ребру (u, v). Поскольку программа P является монотонной, ребро при этом становится очищенным. Ребро (u, v) разделяет грани f и g некоторой укладки графа G, являющиеся вершинами G^*. При построении программы P^* поместим преследователей в те вершины из f и g, которые еще не являются очищенными. После этого удалим преследователей из вершин G^*, у которых все инцидентные им ребра стали очищенными. Затем перейдем к следующему ходу программы P. И так далее.

Нетрудно видеть, что построенная программа P^* является монотонной выигрывающей программой поиска на графе G^* в смысле задачи поиска (2). Докажем, что для ее выполнения достаточно $s(G)$ преследователей.

Для этого убедимся, что число преследователей, находящихся после каждого хода программы P^* на графе G^*, не превосходит числа преследователей, находящихся на графе G после соответствующего хода программы P.

Рассмотрим по очереди все вершины графа G^*, в которых после некоторого хода p согласно программе P^* находятся преследователи. Пусть f — такая вершина. Будем считать, что ход p по программе P^* был построен при просмотре хода q по программе P. Вершина f графа G^* соответствует некоторой грани укладки графа G, ограниченной ребрами $(v_1, v_2), \ldots, (v_{n-1}, v_n)$, (v_n, v_1), образующими цикл в силу двусвязности графа G. Поскольку в вершине f находится преследователь, то среди ребер $(v_1, v_2), \ldots, (v_{n-1}, v_n)$, (v_n, v_1) должны быть ребра, очищенные после хода q программы P. Рассмотрим два случая.

Предположим сначала, что все эти ребра являются очищенными. Такая ситуация может возникнуть только тогда, когда ходом q очищается одно из них. Пусть, для определенности, ходом q преследователь переводится из вершины v_1 в вершину v_2. В этом случае после хода q в вершине v_2 будет не менее двух преследователей. Сопоставим двух из них вершине f графа G_2 (см. первый граф на рис. 2. Здесь и далее жирными линиями показаны очищенные ребра, а преследователи обозначены буквами P_1, P_2). Если преследователей больше двух, то выберем тех, которые были сопоставлены меньшему числу вершин графа G^*.

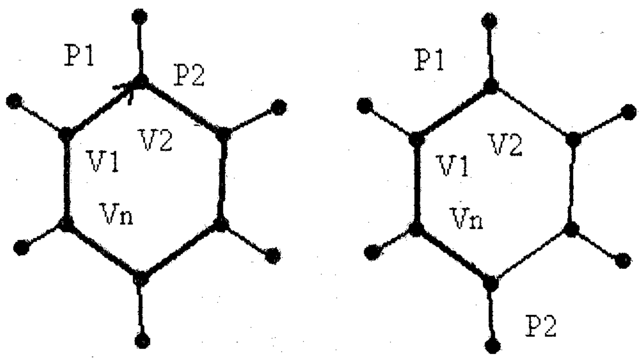

Рис. 2. К доказательству теоремы 1.

Допустим теперь, что среди ребер $(v_1, v_2), \ldots, (v_{n-1}, v_n)$, (v_n, v_1) найдутся ребра, являющиеся загрязненными после хода q. Будем считать, что ребро (v_1, v_n) является очищенным, v_i, $i \in \{1, \ldots, n-1\}$ — вершина с наименьшим номером, инцидентная загрязненному ребру (v_i, v_{i+1}), а v_j, $j \in \{2, \ldots, n\}$ — вершина с наибольшим номером, инцидентная загрязненному ребру (v_{j-1}, v_j) (см. второй граф на рис. 2). В вершинах v_i и v_j после хода q по программе P должны находиться преследователи. Сопоставим вершине f одного преследователя из v_i и одного преследователя из v_j. Если в одной из этих вершин более одного преследователя, то сопоставляем f того из них, который был сопоставлен меньшему числу вершин графа G^*.

Таким образом, каждой вершине графа G^*, в которой находятся преследователи после хода p по программе P^*, сопоставляется два преследователя, находящихся на графе G после хода q по программе P.

В силу того что вершины графа G имеют степени, не превосходящие трех, один преследователь может быть сопоставлен по описанным правилам не более, чем двум вершинам G^*. Из этого сразу следует, что число преследователей, находящихся на графе G^* после хода p, не превосходит числа преследователей, находящихся на графе G после хода q. •

Отметим, что теорема 1 остается верной и в случае, если условие двусвязности графа G в формулировке теоремы заменить на условие связности. Доказательство строится совершенно аналогично, но оно становится более громоздким, поскольку грань укладки графа G может не быть ограничена циклом графа.

Оценим теперь вершинно-поисковое число планарного графа с небольшими степенями вершин, все грани укладки которого являются треугольниками, снизу через поисковое число двойственного графа.

Теорема 2. *Пусть G — связный планарный граф, все степени вершин которого не превосходят пяти, а все грани укладки G на плоскости ограничены циклами из трех ребер. Тогда $ns(G) \geq s(G^*)$.*

Доказательство. Докажем, что $ns(G) \geq cw(G^*)$. Этого достаточно для доказательства, поскольку у графа G^* все вершины имеют степень три, и, следовательно, $cw(G^*) = s(G^*)$.

Рассмотрим произвольную монотонную выигрывающую программу поиска P на графе G (в смысле задачи поиска (2)), для осуществления которой достаточно $ns(G)$ преследователей. Построим, последовательно просматривая ходы этой программы, нумерацию вершин графа G^*. Если очередным ходом программы P преследователь удаляется с графа, то мы сразу переходим к следующему ходу этой программы. Предположим, что очередным ходом программы P преследователь перемещается в вершину v. В этом случае последовательно нумеруются вершины G^*, соответствующие граням укладки G, такие, что во всех трех вершинах, ограничивающих их, оказываются преследователи. Если таких граней найдется несколько, то они являются соседними (см. рис. 3) так как степени вершин графа G не превосходят пяти.

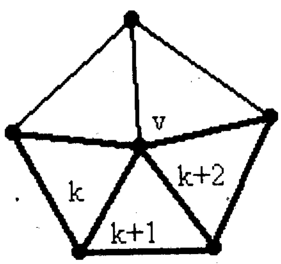

Рис. 3. К доказательству теоремы 2.

Грани нумеруются последовательно против часовой стрелки относительно вершины v, как показано на рисунке. Обозначим полученную нумерацию вершин графа G^* через f. Покажем, что $ns(G) \geq cw(G^*, f)$.

Рассмотрим произвольное $i \in \{1, \dots, n-1\}$, где n — число вершин графа G^*. Обозначим через p ход программы P, после которого получила номер i вершина $f^{-1}(i)$ графа G^*. Докажем, что число ребер (g, h) графа G^*, таких, что $f(g) \leq i$, а $f(h) > i$, не превосходит числа преследователей, находящихся на графе G после хода p. Пусть (g, h) — ребро G^*, для которого $f(g) \leq i$ и $f(h) > i$. Вершины g и h соответствуют соседним граням укладки графа G на плоскости. Пусть (u, v) — ребро G, разделяющее эти грани. Нетрудно видеть, что в вершинах u и v после хода p должны находиться преследователи. Сопоставим их ребру (g, h). Так как все вершины G имеют степени, не превосходящие пяти, каждый преследователь сопоставлен не более, чем двум ребрам G^*. Поскольку $i \in \{1, \dots, n-1\}$ было выбрано произвольно, то $ns(G) \geq cw(G^*, f)$.

Осталось заметить, что по определению ширины разреза $cw(G^*) \geq cw(G^*, f)$. •

Граф G' называется ***гомеоморфным образом*** графа G, если изоморфный графу G' граф может быть получен из графа G помещением на ребра G некоторого числа вершин степени два. Графы G и F называются ***гомеоморфными***, если у этих графов имеются изоморфные гомеоморфные образы. При доказательстве теоремы 3 мы будем использовать следующие два очевидных факта. Во-первых, поисковые числа гомеоморфных графов совпадают. Во-вторых, если граф G получен из F добавлением кратных ребер к некоторым ребрам графа F, то вершинно-поисковые числа G и F совпадают.

Объединяя теоремы 1 и 2, мы получаем следующее утверждение.

Теорема 3. *Пусть G — двусвязный планарный граф, все степени вершин которого не превосходят трех, а все грани укладки G на плоскости ограничены циклами, содержащими не более пяти ребер. Тогда $s(G) = ns(G^*)$.*

ДОКАЗАТЕЛЬСТВО. Неравенство $s(G) \geq ns(G^*)$ следует из теоремы 1. Для доказательства обратного неравенства заметим, что если граф G не имеет вершин степени три, то он, в силу двусвязности, является либо полным графом с двумя вершинами, либо циклом. Легко проверить, что в первом случае $s(G) = ns(G^*) = 1$, а во втором $s(G) = ns(G^*) = 2$. Если же граф имеет вершины степени два и три, то заменим граф G на гомеоморфнный ему граф G' без вершин степени два. Из теоремы 2 вытекает, что $s(G') \leq ns((G')^*)$. Так как G и G' гомеоморфны, то $s(G) = s(G')$, а укладка графа G на плоскости является одновременно и укладкой G'. Поскольку граф G^* получается из графа $(G')^*$ добавлением нескольких кратных ребер, то $ns(G^*) = ns((G')^*)$.

В результате, $s(G) = ns(G^*) = s(G') = ns((G')^*) = ns(G^*)$. •

3. Гипотезы и нерешенные задачи

В заключительной части работы мы приводим гипотезы о возможном обобщении теоремы 3 и о путях доказательства в более общем случае. Мы предполагаем, что выполнено следующее утверждение, которое нам не удалось доказать.

Гипотеза 1. *Пусть G — двусвязный планарный граф. Тогда* $s(G) = ns(G^*)$.

Заметим, что достаточно доказать выполнение неравенства $s(G) \geq ns(G^*)$. Для того чтобы проверить выполнение обратного неравенства, заменим каждое ребро графа G на три кратных ребра. Обозначим получившийся граф через G'. Известно, что $s(G') = ns(G)+1$ (см. [3]). Построим укладку графа G' на плоскости, используя укладку графа G (укладывая новые ребра «рядом» со старыми). Граф $(G')^*$ получается из G^* добавлением двух вершин степени два на каждое ребро графа G^*. В [3] доказано, что в этом случае $ns(G^*) = s(G^*) + 1$. Применяя неравенство $s(G') \geq ns((G')^*)$, получаем, что $ns(G) \geq s(G^*)$. Так как G^* двусвязный граф и граф G является двойственным G^*, то $s(G) \leq ns(G^*)$.

Мы считаем, что доказательство неравенства $s(G) \geq ns(G^*)$ можно свести к случаю, когда все вершины графа G имеют степени, не превосходящие трех, то есть к случаю, рассмотренному в теореме 1. Для того чтобы пояснить это, введем операцию расщепления вершины графа, обратную операции стягивания ребра.

Пусть v — вершина некоторого графа G, имеющая степень, большую трех. Заменим вершину v на две вершины u и w, соединенные ребром. Каждое ребро, инцидентное вершине v, заменяется либо ребром, ицидентным u, либо ребром, инцидентным w (вторые вершины, инцидентные этим ребрам, не меняются). Будем называть граф G' *расщеплением графа* G, если граф может быть получен из графа G последовательным применением операций расщепления вершин. Очевидно, что, расщепляя вершины графа, можно в конечном итоге получить граф, все вершины которого имеют степени, не превосходящие трех. Такое расщепление мы будем называть полным.

Нетрудно доказать, что для любого графа G можно построить такое его полное расщепление G', для которого $s(G) = s(G')$. Однако такое расщепление планарного графа G не обязательно является планарным графом.

Пусть G — связный планарный граф. Рассмотрим некоторую укладку G на плоскости. Назовем монотонную выигрывающую программу (в смысле задачи поиска (1)) на графе G правильной программой, соответствующей укладке G, если грани этой укладки можно пронумеровать таким образом, что согласно программе поиска очищаются все ребра, ограничивающие первую грань, затем все ребра, ограничивающие вторую грань, но не ограничивающие первую, и так далее.

Мы полагаем, что имеет место следующее утверждение.

Гипотеза 2. *Пусть G — связный планарный граф. Тогда для любой укладки G на плоскости существует правильная программа, для осуществления которой достаточно $s(G)$ преследователей.*

Если это действительно так, то несложно доказывается, что у любого планарного графа G существует полное расщепление G', являющееся планарным графом, для которого $s(G) = s(G')$. Причем в этом случае укладку G' на плоскости можно построить из укладки G таким образом, что граф $(G')^*$, соответствующий этой укладке, будет получаться добавлением ребер к графу G^*. Тогда имеет место неравенство $s(G) = s(G') \geq ns((G')^*) \geq ns(G^*)$.

Литература

1. **Bienstock D.** Graph searching, path-width, tree-width and related problems (a survey)//*DIMACS Ser. in Discrete Mathematics and Theoretical Computer Science. 1991. V. 5. P. 33–49.*

2. **Bienstock D., Seymour P.** Monotonicity in graph searching//*J. Algorithms. 1991. V. 12. P. 239 – 245.*

3. **Ellis J. A., Sudborough I. H., Turner J.** The vertex separation and search number of a graph//*Information and Computation. 1994. V. 113. P. 50–79.*

4. **Kirousis L. M., Papadimitriou C. H.** Interval graphs and searching//*Discr. Math. 1985. V. 55. P. 181–184.*

5. **Kirousis L. M., Papadimitriou C. H.** Searching and pebbling//*Theor. Comp. Sci. 1986. V. 47. P. 205–218.*

6. **LaPaugh A. S.** Recontamination does not help to search a graph// *J. ACM. 1993. V. 40. P. 224–245.*

7. **Makedon F. S., Sudborough I. H.** On minimizing width in linear layouts//*Discr. Appl. Math. 1989. V. 23. P. 201–298.*

8. **Parsons T. D.** Pursuit evasion in a graph//*in Theory and Application of Graphs, Y. Alavi and D. R. Lick, eds. Berlin: Springer Verlag, 1976. P. 426–441.*

9. **Головач П.А.** Эквивалентность двух формализаций задачи поиска на графе// *Вестник ЛГУ. Сер.1. 1989. Вып.1. С.10–14.*

10. **Головач П.А.** Экстремальные задачи поиска на графах. *Дисс. к.ф.-м.н.* ЛГУ, 1990.

11. **Петров Н.Н.** Задачи преследования при отсутствии информации об убегающем//*Дифференциальные уравнения. 1982. Т.18. №8. С.1345–1352.*

12. **Харари Ф.** Теория графов. М.: Мир, 1973.

Summary

Golovach P.A., Fomin F.V. Search and node search number of dual graphs

Relations between search and node search numbers of dual graphs are investigated. Some results are given and some conjectures discussed.

Сыктывкарский университет
Санкт-Петербургский университет *Поступила 20.09.2000*