

О НЕКОТОРЫХ АЛГОРИТМАХ И СВОЙСТВАХ ГРАФОВ СРАВНЕНИЯ

С. Е. Маркосян

Известия Национальной Академии Наук Армении. Математика,
том 35, № 3, 2000

Если граф и его дополнение являются графами сравнения, то при любой транзитивной ориентации этих графов существуют вершины *minmin*, *minmax*, *maxmin* и *maxmax* (*minmin* – вершина, которая *min* в графе и в его дополнении, *minmax* – вершина, которая *min* в графе и *max* в его дополнении, и т.д.). Из этого свойства следует теорема Пнуели, Лемпела и Эвена о графах подстановок. Разработаны алгоритмы решения для таких задач как нахождение максимального числа попарно непересекающихся максимальных независимых множеств и наибольшего q -хроматического подграфа.

§1. ВВЕДЕНИЕ

Пусть $G = (V, E)$ – обыкновенный неориентированный граф и $G' = (V, E')$ – дополнение графа G . G будем называть графом сравнения, если его рёбра можно ориентировать так, чтобы полученный ориентированный граф $\bar{G} = (V, \bar{E})$ был бы транзитивен. Будем называть G графом ко-сравнения, если его дополнение G' является графом сравнения. Для удобства иногда будем использовать следующие обозначения: $\bar{G} = (V, \Gamma)$ и $\bar{G}' = (V, \Gamma')$, где Γ и Γ' – многозначные отображения в смысле Бержа [1], а Γ^{-1} и $(\Gamma')^{-1}$ суть обратные отображения.

Вершина $v \in V$ называется *minimal* (*min*), если $\Gamma^{-1}x = \emptyset$. Она называется *maximal* (*max*), если $\Gamma x = \emptyset$. Аналогично, эти понятия определяются для $\bar{G}'$. Вершина $v \in V$ называется *minmax*, если она *min* в $\bar{G}$ и *max* в $\bar{G}'$. Вершины, называемые *maxmin*, *minmin* и *maxmax* определяются аналогично.

В параграфе 2 доказывается следующая теорема:

Теорема 1. Пусть G – граф сравнения и ко-сравнения. Тогда при любой транзитивной ориентации графа $G = (V, E)$ и его дополнения $G' = (V, E')$ существуют вершины *maxmin*, *minmin*, *maxmax*, *minmax*.

Следствие 1.(см. [10]) Граф G является графом подстановок, если как G , так и его дополнение G' , являются графами сравнения.

Если через T , T' и P обозначим классы графов сравнения, ко-сравнения и подстановок, соответственно, то из Теоремы 1 следует, что $P = T \cap T'$.

В параграфе 3 рассмотрены следующие задачи : 1) нахождение наибольшего независимого множества ; 2) нахождение наибольшей клики ; 3) нахождение минимальной раскраски ; 4) нахождение минимального покрытия вершин кликами ; 5) нахождение максимального q -хроматического подграфа (максимального q независимого множества) ; 6) нахождение максимальных q -клик, т.е. таких непесекающихся q клик, имеющих наибольшее число вершин ; 7) нахождение максимального числа попарно непесекающихся α -независимых множеств, где α – число независимости графа (α -независимое множество содержит α вершин) ; 8) нахождение максимального числа попарно непесекающихся ω -клик, где ω – плотность графа (мощность наибольшей клики).

Задачи 1 – 8 в общем случае NP-трудны. Однако для решения большинства из этих задач разработаны эффективные алгоритмы для отдельных подклассов графов. Для графов интервалов в работах [9], [11] были разработаны эффективные алгоритмы для решения Задач 1 – 5, а для решения Задачи 8 (в §3 настоящей статьи). Задача 5 впервые была сформулирована и решена для графов интервалов в [9], [6-8]. В дальнейшем аналогичные алгоритмы были предложены в работах [13], [15]. Как вытекает из следствия, для графов перестановок все Задачи 1 – 8 имеют эффективные алгоритмы решения.

Для графов сравнения общеизвестны эффективные алгоритмы решения Задач 1 – 4. Для Задачи 6 в работе [4] разработаны некоторые алгоритмы, а для Задач 7, 8 приведены алгоритмы в §3 настоящей статье. Задачи 1, 4 решаются на основе теоремы Дилворта, применением алгоритма нахождения максимального потока (см. [3]). Все эти алгоритмы эффективны, так как их порядок не превосходит $|V|^2$ или $|V|^3$, когда применяются потоковые алгоритмы. Пока не известен эффективный алгоритм для Задачи 5. Однако, Задача 5 для графа подстановок сводится к Задаче 6 для дополнения этого графа, и поэтому эффективно решается, так как дополнение также есть граф сравнения.

В параграфе 3 мы опишем эффективные алгоритмы для оптимального решения Задач 7 и 8, и алгоритм решения Задачи 5 для графов сравнения. Для последнего алгоритма оптимальность не доказана, и поэтому важно выяснить Задача 5 NP-

трудна или нет. Также неизвестно, эта задача NP-трудна или нет даже для совершенных графов.

§2 СВОЙСТВА MAXMIN, MINMIN, MAXMAX И MINMAX

Доказательство Теоремы 1 : 1. Вначале докажем существование minmin.

Пусть $\bar{G} = (V, \bar{E}) = (V, \bar{\Gamma})$ и $\bar{G}' = (V, \bar{E}') = (V, \bar{\Gamma}')$ – транзитивно ориентированные графы, полученные после ориентации графов G и G' , соответственно. Разобьём V на подмножества $X_1, X_2, \dots, X_{\omega(G)}$ следующим образом :

$$X_1 = \{x/x \in X, \quad \Gamma^{-1}x = \emptyset\},$$

$$X_2 = \{x/x \in X \setminus X_1, \quad \Gamma^{-1}x \subseteq X_1\},$$

.....

$$X_{\omega(G)} = \{x/x \in X \setminus \bigcup_{1 \leq j \leq \omega(G)-1} X_j, \quad \Gamma^{-1}x \subseteq \bigcup_{1 \leq j \leq \omega(G)-1} X_j\},$$

где $\omega(G)$ – плотность графа. Каждое X_j – независимое множество. Разбиение $X_1, X_2, \dots, X_{\omega(G)}$ является минимальной раскраской графа, и для любой вершины $x_{\omega(G)} \in X_{\omega(G)}$ можно построить наибольшую клику $Q = \{x_{\omega(G)}, x_{\omega(G)-1}, \dots, x_1/x_j \in X_j\}$. Любой подграф $\bar{G}'_j = \bar{G}'(X_j)$ ($j = 1, 2, \dots, \omega(G)$), порожденный подмножеством X_j , является полным. Так как $\bar{G}'$ – транзитивно ориентированный граф, то вершины подмножества X_j образуют ориентированную цепь $\mu = (x_1^j, x_2^j, \dots, x_{k_j}^j)$. Предположим, что дуги графа $\bar{G}$ окрашены красным, а дуги $\bar{G}'$ синим цветом. Докажем, что вершина $x_1^1 \in X_1$ является minmin. Очевидно, все вершины подмножества X_1 являются min в графе $\bar{G}$. Остаётся показать, что x_1^1 является min в графе $\bar{G}'$, т.е. $(\Gamma')^{-1}x_1^1 = \emptyset$, или в x_1^1 не входят синие дуги. Покажем более общее утверждение : если $p < q \leq \omega(G)$, $x_q \in X_q$, то не существует дуги (x_q, x_p^1) . Если мы предположим, что такая дуга существует, то из построения множества X_j ($j = 1, 2, \dots, \omega(G)$), для x_q существует $x_{q-1} \in X_{q-1}$ такая, что $x_{q-1} \in \Gamma^{-1}x_q$ (в противном случае $x_q \in X_k$ для $k < q$). Таким образом, $\Gamma^{-1}x_q \cap X_p \neq \emptyset$, т.е. существует красная цепь между множеством X_p и вершиной x_q . Но по предположению существует синяя дуга (x_q, x_p^1) , а x_p^1 соединена синей дугой с любой вершиной $x_p^j \in X_p$. Так как граф $\bar{G}$ транзитивно ориентирован, то любая вершина $x_p^j \in X_p$ соединена синей дугой с x_q . Это противоречит с существованием красной дуги между множеством X_p и вершиной x_q . Таким образом, вершина x_1^1 является minmin.

2. Существование minmax.

Покажем теперь, что последняя вершина $x_{k,1}^1$ цепи, индуцированная подмножеством X_1 , является minmax. Как и в предыдущем пункте можно доказать, что из $x_{k,1}^1$ не выходит синих дуг. Это означает, что $x_{k,1}^1$ в $\bar{G}'$ является max. Из $x_{k,1}^1 \in X_1$ следует, что $x_{k,1}^1$ является min в $\bar{G}$, т.е. вершина $x_{k,1}^1$ является minmax.

3. Существование maxmin.

Пусть x_1^m – первая вершина из x_1^j ($j = 1, 2, \dots, \omega(G)$), из которой не выходит красная дуга. Покажем, что вершина x_1^m является maxmin. Для этого нужно показать, что x_1^m есть min в $\bar{G}'$, т.е. в x_1^m не входит синяя дуга. Из доказательства пункта 1 мы знаем, что в x_1^m не входит синяя дуга (x_j, x_1^m) , где $x_j \in X_j$, $j \geq m$. Остаётся доказать, что в x_1^m не входит синяя дуга (x_j, x_1^m) , где $x_j \in X_j$, $j < m$. Предположим, что в x_1^m входит синяя дуга (x_i^i, x_1^m) , где $x_i^i \in X_i$, $i < m$. Тогда ясно, что (x_1^i, x_1^m) также является синей дугой. Предположим, что i является последним индексом, для которого (x_1^i, x_1^m) является синей дугой. По выбору вершины x_1^m из вершины x_1^i выходят красные дуги. Если (x_1^i, y) – красная дуга, то $y \in X_j$, где $i < j < m$, так как (x_1^i, x^j) – синяя дуга для любого $x^j \in X_j$ ($j = m, m+1, \dots, \omega(G)$) (следует из того факта, что дуги (x_1^i, x_1^m) и (x_1^m, x_j) , $x_j \in X_j$, $j > m$ – синие). Пусть (x_1^i, x_t) , $x_t \in X_t$, $i < t < m$ – последняя красная дуга, выходящая из x_1^i , т.е. t – наибольший индекс, для которого дуга (x_1^i, x^t) – красная. Следовательно, x^t и x_1^m соединены синей дугой (x^t, x_1^m) , а не (x_1^m, x^t) . Так как (x^t, x_1^m) является синей дугой, то (x_1^i, x_1^m) также является синей дугой. Но i является последним индексом, для которого (x_1^i, x_1^m) является синей дугой. Это противоречие доказывает, что в x_1^m не входит синяя дуга, т.е. она является maxmin вершиной.

4. Существование maxmax.

Пусть $x_{k,m}^m$ – первая вершина из $x_{k,j}^j$, из которой не выходит красной дуги ($x_{k,j}^j$ – последняя вершина цепи $\mu_j = (x_1^j, x_2^j, \dots, x_{k,j}^j)$, индуцированная множеством X_j). Как и в пункте 3 можно доказать, что из $x_{k,m}^m$ не выходит синей дуги, т.е. $x_{k,m}^m$ является maxmax вершиной. Теорема 1 доказана.

Вышеприведенное доказательство Теоремы 1 является конструктивным, т.е. в доказательстве явно отмечено, какие вершины maxmin, minmin, minmax, maxmax. Можно привести более короткое доказательство Теоремы 1, но оно лишено этого преимущества.

Второе доказательство Теоремы 1 : Дадим новое доказательство только для одного из пунктов. Пусть $\bar{G}_\alpha$ и $\bar{G}'_\beta$ суть ориентированные графы, полученные из транзитивной ориентации α и β графов G и G' , соответственно. Предположим, что известно существование $\min\min$ вершины из вышеприведенного доказательства, и пусть мы хотим найти вершину $\min\max$ для ориентаций α и β . Возьмем ориентации α и β^{-1} , соответствующие G и G' (β^{-1} – обратная ориентация к β). Существование $\min\min$ вершины для ориентаций α, β^{-1} следует из первой части этого доказательства. Вершина $\min\max$ для ориентаций α и β . Существование $\max\min$ и $\max\max$ доказываются аналогично. Доказательство завершено.

Доказательство Следствия 1 : Пусть транзитивно ориентированные G и G' суть $\bar{G}$ и $\bar{G}'$. Построим подстановку, граф которой изоморфен G . Первую строку $(i_1, i_2, \dots, i_n)$ этой подстановки строим следующим образом : первый элемент i_1 есть вершина $\min\min$ для графов $\bar{G}$ и $\bar{G}'$, элемент i_2 – вершина $\min\min$ для подграфов $\bar{G} - \{i_1\}$ и $\bar{G}' - \{i_1\}$, элемент i_3 – вершина $\min\min$ для подграфов $\bar{G} - \{i_1, i_2\}$ и $\bar{G}' - \{i_1, i_2\}$, и т.д. Вторую строку $(j_1, j_2, \dots, j_n)$ этой подстановки строим аналогичным путём только каждый раз вместо $\min\min$ выбираем $\max\min$. Нетрудно проверить, что граф подстановки

$$\begin{pmatrix} i_1 & i_2 & \dots & i_n \\ j_1 & j_2 & \dots & j_n \end{pmatrix}$$

изоморфен графу G . Следствие доказано.

§3. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ 5, 7, 8 ДЛЯ ГРАФОВ СРАВНЕНИЯ

3.1. Алгоритм для Задачи 7.

Задача 7 для графа сравнения G находит наибольшее число попарно непересекающихся независимых множеств мощности $\alpha = \alpha(G)$ = число независимости (стабильности) графа G . Аналогично формулируется Задача 8 для максимального несмежного ω -клика и алгоритм решения которой приведен в пункте 2 этого параграфа.

Алгоритм решения Задачи 7, описанный ниже, включает алгоритм решения Задачи 4, т.е. задача минимального покрытия. Алгоритмы транзитивной ориентации обычно имеют сложность $O(n^3)$, где n – число вершин. Однако разработаны более экономные алгоритмы, имеющие сложность $O(m \cdot \Delta)$, где m – число рёбер,

а Δ – максимальная степень графа. Сложность нашего алгоритма имеет тот же порядок. Прежде чем перейти к описанию алгоритма, приведем некоторые факты для транзитивно ориентированных графов, которые будут служить теоретической основой алгоритма.

Пусть $G = (V, E)$ – граф сравнения и $\bar{G} = (V, \bar{E})$ – транзитивно ориентированный граф, полученный из G с помощью некоторой транзитивной ориентации. Если $\Delta = \{Q_1, Q_2, \dots, Q_\alpha\}$ (где α – число стабильности) – минимальное покрытие графа G , то каждой клике Q_j , $j = 1, 2, \dots, \alpha$ соответствует ориентированная цепь $\mu = (x_1^j, x_2^j, \dots, x_{k_j}^j)$, т.е. вершины клики Q_j упорядочены. Вообще, вершины графа $\bar{G}$ частично упорядочены, скажем $x > y$, если (x, y) является дугой в $\bar{G}$.

Опишем простой алгоритм нахождения наибольшего независимого множества (α -независимого множества) графа $\bar{G}$ из покрытия $\bar{\Delta} = \{\bar{Q}_1, \bar{Q}_2, \dots, \bar{Q}_\alpha\}$. Назовём x α -вершиной, если x содержится в некотором α -независимом множестве. Рассмотрим $S = \{x_1^1, x_1^2, \dots, x_1^\alpha\}$. Если (x_1^p, x_1^q) является дугой, то вершина x_1^p соединена со всеми вершинами клики Q_q , и она не является α -вершиной. Следовательно, α -независимое множество можно получить с помощью следующего алгоритма.

Алгоритм А. Положим $S = \{x_1^1, x_1^2, \dots, x_1^\alpha\}$.

1. Проверяем, содержит ли S дугу или нет. Если нет, то S является α -независимым множеством.

2. Если S содержит дугу $(x_{i_p}^p, x_{i_q}^q)$, то вместо $x_{i_p}^p$ берём $x_{i_p+1}^p$ и переходим к 1.

Для получения максимального числа попарно непересекающихся α -независимых множеств Алгоритм А изменим следующим образом.

Алгоритм В. Начинаем с $S = \{x_1^1, x_1^2, \dots, x_1^\alpha\}$. Пусть текущее состояние $S = \{x_{i_1}^1, x_{i_2}^2, \dots, x_{i_\alpha}^\alpha\}$.

1. Проверяем, содержит ли S дугу или нет. Если да – переход к 2, а если нет – переход к 3.

2. Пусть S содержит дугу $(x_{i_p}^p, x_{i_q}^q)$, тогда берём $S = (S \setminus \{x_{i_p}^p\}) \cup \{x_{i_p+1}^p\}$, если $i_p + 1 \leq k_p$, то переход к 1, в противном случае конец.

3. Берем $S = \{x_{i_1}^1, x_{i_2}^2, \dots, x_{i_\alpha}^\alpha\}$ как очередное α -независимое множество. Если $i_t + 1 \leq k_t$, $t = 1, 2, \dots, \alpha$, то переход к 1, приняв $S = \{x_{i_1+1}^1, x_{i_2+1}^2, \dots, x_{i_\alpha+1}^\alpha\}$, в противном случае конец.

Теперь мы можем полностью описать алгоритм решения Задачи 7.

Алгоритм 7. Пусть дан граф $G = (V, E)$.

1. Транзитивно ориентируем G и получаем $\bar{G} = (V, \bar{E})$.
2. Находим минимальное покрытие графа $\bar{G}$, $\bar{\Delta} = \{\bar{Q}_1, \bar{Q}_2, \dots, \bar{Q}_\alpha\}$.
3. Применяя Алгоритм В, находим максимальное число попарно непересекающихся α -независимых множеств.

Теорема 2. Алгоритм 7 даёт максимальное число попарно непересекающихся α -независимых множеств.

Доказательство : Мы должны показать, что применяя Алгоритм В к некоторому минимальному покрытию $\bar{\Delta} = \{\bar{Q}_1, \bar{Q}_2, \dots, \bar{Q}_\alpha\}$ графа $\bar{G}$, получаем максимальное число попарно непересекающихся α -независимых множеств. Легко убедиться, что для этого достаточно показать, что если $S_1 = \{x_{i1}^1, x_{i2}^2, \dots, x_{i\alpha}^\alpha\}$ и $S_2 = \{x_{j1}^1, x_{j2}^2, \dots, x_{j\alpha}^\alpha\}$ — два произвольных α -независимых множеств графа $\bar{G}$, то подмножества $S_M = \{x_{r1}^1, x_{r2}^2, \dots, x_{r\alpha}^\alpha\}$ и $S_m = \{x_{i1}^1, x_{i2}^2, \dots, x_{i\alpha}^\alpha\}$, где $x_{rk}^k = \max(x_{ik}^k, x_{jk}^k)$ и $x_{ik}^k = \min(x_{ik}^k, x_{jk}^k)$ также являются α -независимыми множествами.

Покажем, что две произвольные вершины в S_M несмежны (для S_m доказательство аналогично). Предположим, что вершины x_{r1}^1 и x_{r2}^2 смежны. Если эти две вершины принадлежат или S_1 или S_2 , то, очевидно, они не смежны. Пусть $x_{r1}^1 = x_{i1}^1$ и $x_{r2}^2 = x_{j2}^2$. Это означает, что $x_{j1}^1 < x_{i1}^1$ и $x_{i2}^2 < x_{j2}^2$. Тогда (x_{r1}^1, x_{r2}^2) не является дугой, так как из существования дуги (x_{i1}^1, x_{j2}^2) следовало бы существование дуги (x_{i1}^1, x_{i2}^2) , что противоречит условию $(x_{i1}^1, x_{i2}^2) \subseteq S_1$. Аналогично, (x_{r2}^2, x_{r1}^1) не есть дуга, так как из этого следовало бы, что $\{x_{j1}^1, x_{j2}^2\} \subseteq S_2$. Поэтому, S_M и S_m являются независимыми множествами. Теорема 2 доказана.

Из доказательства Теоремы 2 видно как можно построить максимальное число попарно непересекающихся α -независимых множеств. В каждой клике $\bar{Q}_i$ пронумеруем α -вершины натуральными числами : максимальные α -вершины в каждой клике получают номер 1, следующие α -вершины — номер 2, и т.д. Ясно, что число α -вершин в каждой клике одинаково, оно равно числу попарно непересекающихся α -независимых множеств. Все α -вершины с одинаковыми номерами образуют α -независимое множество.

3.2. Алгоритм решения Задачи 8. Доказательство оптимальности для Задачи 8 легче, чем для Задачи 7. Здесь опять используются алгоритмы транзитивной ориентации, ранжировки и максимального потока. Порядок алгоритма ранжировки меньше порядка алгоритма транзитивной ориентации и максимального

потока. Поэтому сложность этого алгоритма равна сложности последнего алгоритма. Известные алгоритмы максимального потока имеют сложность $O(n^3)$, где n – число вершин.

Пусть $G = (V, E)$ – граф сравнения, а $\bar{G} = (V, \bar{E}) = (V, \Gamma)$ – транзитивно ориентированный граф, полученный из G некоторой транзитивной ориентацией. Пусть $X_1, X_2, \dots, X_{\omega(G)}$ – подмножества вершин, полученные после ранжировки (см. §2). Построим граф $\bar{G}_1 = (V, \bar{E}_1)$ из графа $\bar{G} = (V, \bar{E})$, где $\bar{E}_1 = \{(x, y) / x \in X_i, y \in X_{i+1}, i = 1, 2, \dots, \omega - 1\}$, т.е. $\bar{G}_1$ содержит те рёбра графа $\bar{G}$, соединяющие соседние множества X_i и X_{i+1} . Ясно, что все ω -клики содержат только дуги из $\bar{G}_1$. Построим транспортную сеть $N = \bar{E}_1 \cup \{s, t\} \cup A$ на $\bar{G}_1$, добавив две вершины : вход s и выход t сети N , и пусть $A = \bar{E}_1 \cup (s, X_1) \cup (X_{\omega}, t)$, т.е. s соединена со всеми вершинами X_1 , а все вершины X_{ω} соединены с t . Пропускные способности $c(e)$ и $c(v)$ дуги $e \in A$ и вершины $v \in V$ мы полагаем равной единице и $c(s) = c(t) = \infty$. Ясно, что если найден максимальный поток в N , то каждая цепь, по которой идет поток, является ω -кликой в $\bar{G}$ (без s и t) и они не пересекаются. Число этих ω -клик максимально. Задача 8 решена.

Алгоритм 8. Пусть G – граф сравнения.

1. Транзитивно ориентируем граф G , получаем граф $\bar{G} = (V, \bar{E})$.
2. Делаем ранжировку и строим подмножества $X_1, X_2, \dots, X_{\omega(G)}$.
3. Строим граф $\bar{G}_1$ и сеть N .
4. Используя максимальный поток, строим ω -клики. Это те цепи из s в t (без s и t) по которым идет поток.

3.3. Алгоритм решения Задачи 5.

Предложенный нами алгоритм на каждом шаге работает как алгоритм решения Задачи 8, описанный в пункте 2 этого параграфа. Но на каждом шаге вместо максимального потока нам нужен минимальный вершинный разрез Y_k . Всего имеем $\omega - q$ шагов, где $\omega = \omega(G)$ – плотность графа. Пусть на текущем шаге $k = \omega, \omega - 1, \dots, q + 1$ получены графы $\bar{G}_k, \bar{G}'_k$ и минимальный вершинный разрез Y_k , а $\bar{G}_{\omega} = \bar{G}$ получен транзитивным ориентированием графа G . Граф $\bar{G}'_k$ получается из $\bar{G}_k$ и содержит только те дуги из $\bar{G}_k$, соединяющие вершины соседних подмножеств после ранжировки $\bar{G}_k$ (см. Алгоритм 8). Добавляя две вершины s и t , получаем сеть N'_k из $\bar{G}'_k$ как это сделано в Алгоритме 8. Вход s соединен со всеми вершинами первого подмножества вершин ранжировки $\bar{G}_k$,

а все вершины последнего подмножества вершин ранжировки соединены с t . Пропускные способности $c(s) = c(t) = \infty$, $c(v) = 1$ для $v \in \bar{G}'_k$, а для всех дуг пропускная способность равна 2. Для нахождения минимального вершинного разреза в сети N'_k , как обычно это делают, нужно построить новую сеть N''_k , где вершины заменены дугами. Тогда найдём минимальный разрез на N''_k , состоящий из рёбер, которым будут соответствовать вершины.

На k -м шаге мощность клики графа $\bar{G}_k = \bar{G} - Y_{k+1}$ не больше чем k . В конце работы алгоритма получим граф $\bar{G}_q$, который можно раскрасить q цветами, так как $\omega(\bar{G}_q) \leq q$. Множество вершин этого графа есть $V_q = V \setminus \bigcup_{q+1 \leq k \leq \omega} Y_k$. Теперь опишем алгоритм решения Задачи 5.

Алгоритм решения Задачи 5.

1. Построим транзитивную ориентацию графа G и получим граф $\bar{G}_\omega = \bar{G}$.
2. Определим $k = \omega$.
3. Сделаем ранжировку графа $\bar{G}_k$.
4. Построим $\bar{G}'_k$ из $\bar{G}_k$.
5. Построим N'_k из $\bar{G}'_k$.
6. В N'_k найдём минимальный вершинный разрез Y_k .
7. Построим $\bar{G}_{k-1} = \bar{G}_k - Y_k$.
8. Если $k = q + 1$, то заканчиваем, в противном случае $k = k - 1$ и переходим к 3.

Пункты 1 и 6 определяют сложность Алгоритма 5. Сложность алгоритма транзитивной ориентации есть $O(n^3)$ [10], [14], где n — число вершин графа. Сложности нахождения максимального потока и минимального разреза есть $O(n^3)$ (хотя для графов сравнения порядок — $O(n^2)$). Следовательно, сложность Алгоритма 5 зависит от сложности потокового алгоритма, т.е. если последний имеет сложность $f(n)$, то сложность Алгоритма 5 есть $O((\omega - q - 1) * f(n))$.

3.4. Примеры. Для иллюстрации работы Алгоритма 5 приведем несколько примеров. Вначале сделаем несколько замечаний.

Возьмем например $k = \omega$. Алгоритм 5 выбирает минимальное $s - t$ разделяющее множество вершин Y_ω в графе $\bar{G}_\omega$, т.е. из каждой $s - t$ цепи выбираем хотя бы одну вершину, отличную от s и t . Из теоремы Менгера следует, что число вершин в Y_ω равно наибольшему числу попарно непересекающихся (по вершинам) $s - t$ цепей в $\bar{G}_\omega$. Важно, что Алгоритм 5 выбирает не произвольное $s - t$ разделяющее множество, а такое, которое получается с помощью потокового алгоритма.

Минимальный разрез состоит из "первых старших вершин" ("первая старшая" означает, что для любой вершины $z \in Z_\omega$, где Z_ω — произвольный минимальный $s - t$ вершинный разрез, существует $y \in Y_\omega$, $y \geq z$, т.е. существует $(y - z)$ цепь). Это следует из факта, что если (V_1, V_2) — минимальный разрез, полученный из максимального потока, то для любого минимального разреза (V'_1, V'_2) , $V_1 \subseteq V'_1$ [3]. Этот факт и является главной причиной того, что Алгоритм 5 все таки может быть оптимальным.

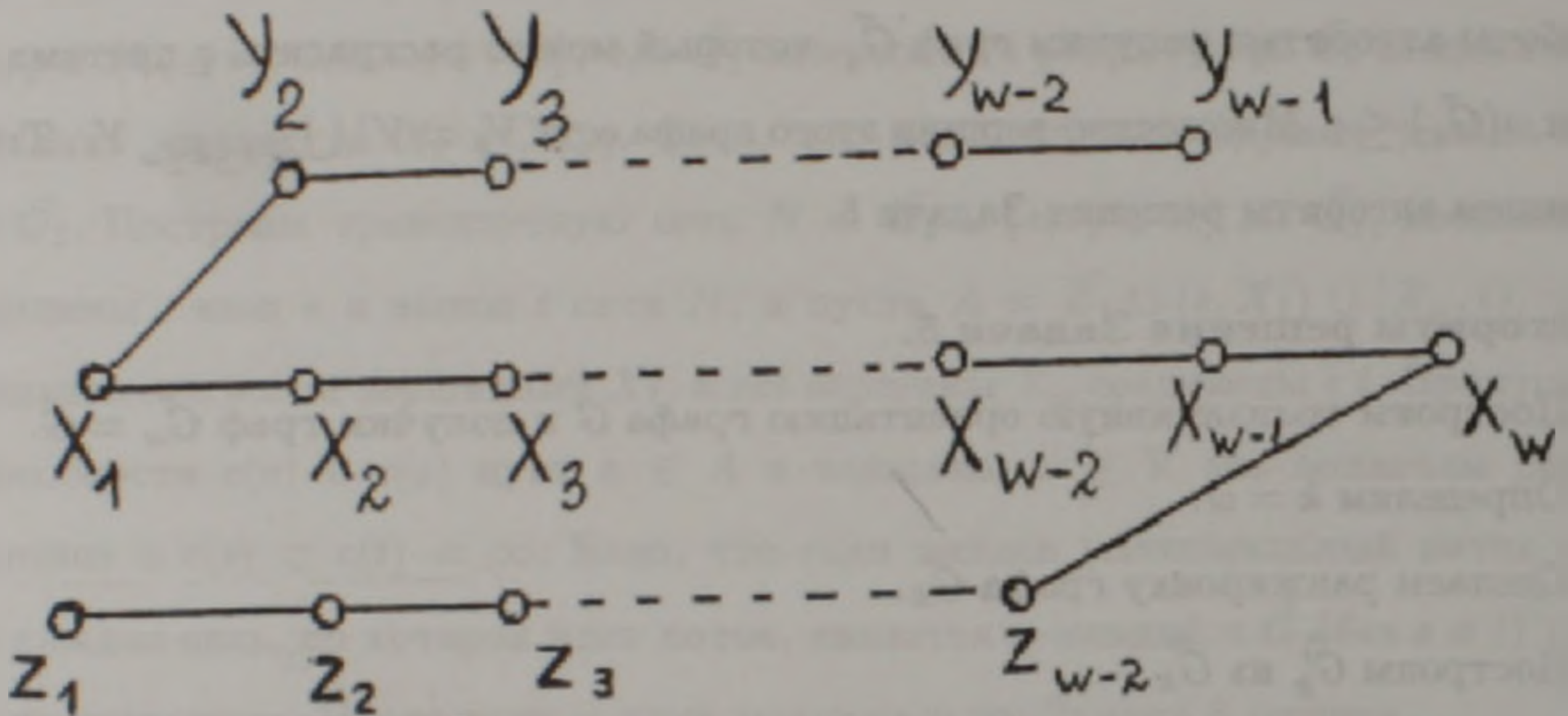


Рис. 1. Дуги графа $\bar{G}$ ориентированы слева направо. Для простоты на рисунке дуги (x_i, x_j) , (x_1, y_j) , (y_i, y_j) , (z_i, z_j) , (z_i, z_w) опущены.

Пусть $\bar{G}$ — транзитивно ориентированный граф, заданный на Рис. 1. Пусть $q = \omega - 2$, т.е. нужно найти максимальный $(\omega - 2)$ -хроматический подграф. Если возьмем минимальный разрез $Y_\omega = \{x_2\}$ в $\bar{G}_\omega = \bar{G}$, то в графе $\bar{G}_\omega - \{x_2\}$ минимальным разрезом $Y_{\omega-1}$ будет подмножество $\{x_1, x_w\}$. Полученный $(\omega - 2)$ -хроматический подграф будет $\bar{G} - \{x_2, x_1, x_w\}$, т.е. не максимальный $(\omega - 2)$ -хроматический подграф графа $\bar{G}$. На ω шаге Алгоритм 5 выберет $Y_\omega = \{x_1\}$ в качестве "первой старшей вершины". После удаления которой не остается ω -клик. На втором шаге $Y_{\omega-1} = \{x_w\}$ и полученный $(\omega - 2)$ -хроматический подграф $\bar{G} - \{x_1, x_w\}$ является максимальным. В [2] введено понятие связанного семейства кликов, из существования которого следует оптимальность q -раскраски.

Пусть $G = (V, E)$ — обыкновенный граф, $S = \{S_1, S_2, \dots, S_q\}$ — множество q независимых подмножеств, т.е. некоторая частная раскраска q цветами, а $C = \{C_j / j \in J\}$ — некоторое семейство клик. Будем говорить, что S и семейство клик C образуют связанное семейство кликов, если

$$(A1) S_i \cap S_j = \emptyset, i \neq j.$$

$$(A2) (\bigcup_{i \leq q} S_i) \cup (\bigcup_{j \in J} C_j) = V.$$

(A3) $S_i \cap C_j \neq \emptyset$ для любых i и j .

Существование связанного семейства кликов для q -раскраски, полученное Алгоритмом 5 (ранжировкой графа $\bar{G}_q$) доказывает оптимальность Алгоритма 5. Для этого достаточно найти такое семейство кликов в графе $\bar{G}$, что

1. $C_i \cap C_j = \emptyset$, $i \neq j$.
2. $(\bigcup_{j \in J} C_j) \supset Y$, $Y = (\bigcup_{q+1 \leq k \leq \omega} Y_k)$.
3. $|C_j \setminus Y| = q$, $j \in J$.

Для графа $\bar{G}$ на Рис. 1 связанное семейство кликов для получения $G_{\omega-2}$ Алгоритмом 5 может быть $C = \{C_1\}$, где $C_1 = \{x_1, x_2, \dots, x_\omega\}$ или $C = \{C_1, C_2\}$, причём $C_1 = \{x_1, y_1, \dots, y_{\omega-1}\}$, $C_2 = \{x_1, x_2, \dots, x_{\omega-2}, x_\omega\}$.

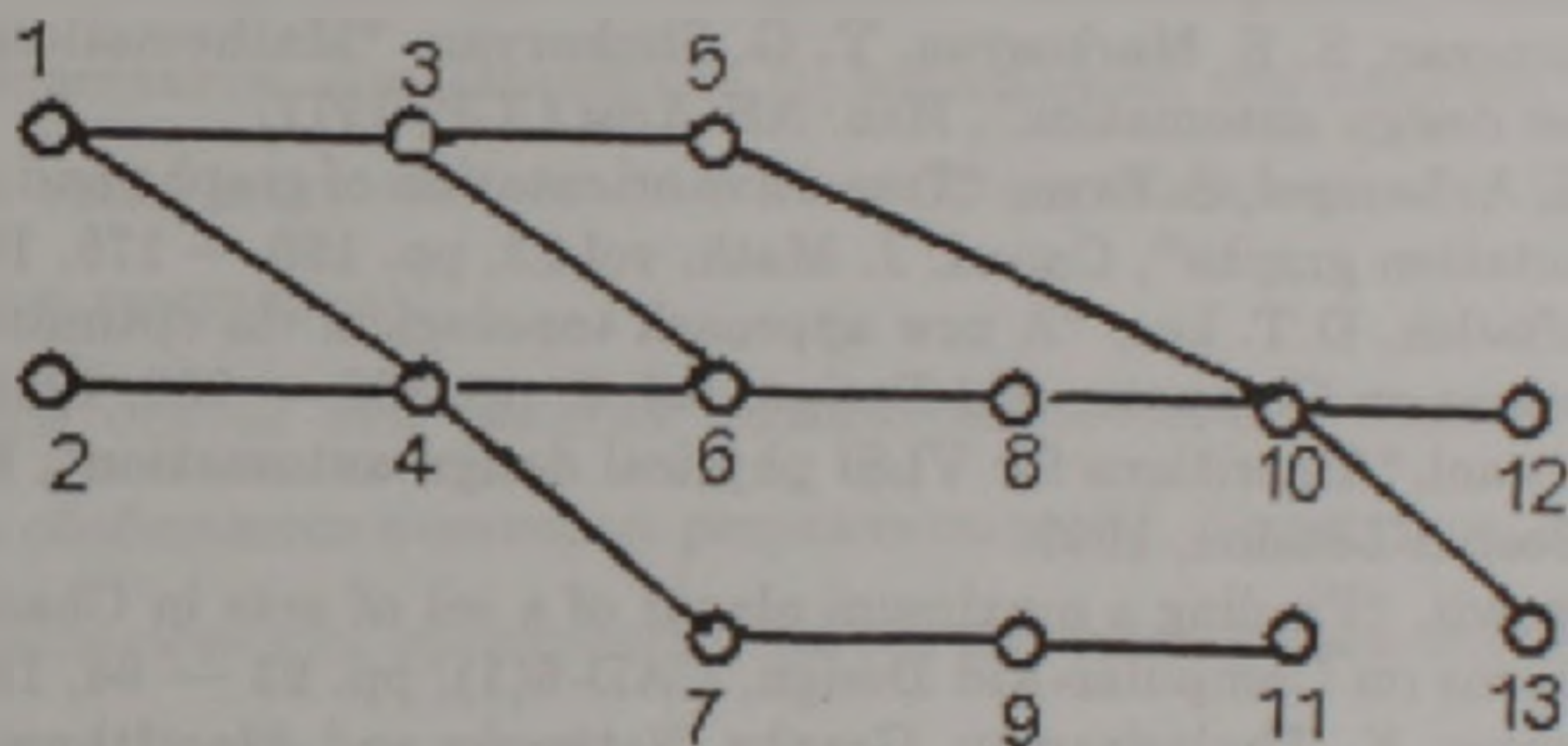


Рис. 2. Ориентация и некоторые дуги графа $\bar{G}$ на рисунке опущены.

Рассмотрим пример на Рис. 2. Пусть $\omega = 6$, $q = \omega - 2 = 4$. Алгоритм 5 на шаге ω даёт $Y_\omega = \{6\}$, хотя $\{8\}$ и $\{10\}$ также являются минимальными разрезами в $\bar{G}_\omega$. На $(\omega - 1)$ -м шаге Алгоритм 5 даёт $Y_{\omega-1} = \{1, 2\}$, хотя $\{4, 10\}$ и $\{7, 10\}$, ... также являются минимальными разрезами в $\bar{G}_{\omega-1}$.

Для полученного $Y = \{1, 2, 6\}$ связанное семейство кликов будет $C = \{C_1, C_2\}$, где $C_1 = \{1, 3, 6, 8, 10, 12\}$ и $C_2 = \{2, 4, 7, 9, 11\}$.

ABSTRACT. If a graph and its complement are comparability graphs, then for any transitive orientation of these graphs there exist a *minmin*, *minmax*, *maxmin* and *maxmax* vertices (*minmin* is the vertex that is min in both the graph and its complement, *minmax* is the vertex that is min in the graph and *max* in its complement, etc.). From this property the theorem of Pnueli, Lempel and Even about permutation graphs is obtained. Some algorithms are developed for the problems of maximal number of pairwise non-intersecting maximal independent sets and of maximal q -chromatic subgraph.

Л И Т Е Р А Т У Р А

1. C. Berge, *Theorie des Graphs et ses Applications*, Paris, 1958.
2. C. Berge, "Minmax relations for the partial q -coloring of graphs", *Discrete Math.* 74, pp. 3 — 4, 1989.
3. L. R. Ford, D. R. Fulkerson, *Flows in Networks*, Princeton Univ. Press, Princeton NJ, 1962.
4. F. Gavril, "Algorithms for maximum k coloring and k covering of transitive graphs", *Networks*, 17, pp. 465 — 470, 1987.
5. R. D. Lou, M. Sarrafzadeh, D. T. Lee, "An optimal algorithm for the maximum two-chain problem", *Proceeding of first SIAM-ACM Conference of Discrete Algorithms*, 1990.
6. С. Е. Маркосян, "О раскраске вершин графов интервалов", *Вопросы радиоэлектроники, серия ЭВТ*, вып. 4, стр. 3 — 6, 1972.
7. С. Е. Маркосян, Г. А. Газарян, "О некоторых задачах линейной трассировки", *Вопросы радиоэлектроники, серия ЭВТ*, вып. 9, стр. 16 — 23, 1974.
8. С. Е. Маркосян, А. Г. Маркосян, "Некоторые алгоритмы сравнения графов интервалов", *Кибернетика*, № 2, стр. 72 — 81, 1976.
9. A. W. Petrosyan, S. E. Markosyan, Y. G. Shukuryan, "Mathematical questions of computer design automation", *Изд. АН Арм.ССР*, 1977.
10. A. Pnueli, A. Lempel, S. Even, "Transitive orientation of graphs and identification of permutation graphs", *Canad. J. Math.* vol.23, pp. 160 — 175, 1971.
11. M. Sarrafzadeh, D.T. Lee, "A new approach topological via optimization", *IEEE Transactions on Computer-Aid Design*, vol. 8, pp. 890 — 900, 1989.
12. N. A. Serwani, "Algorithms for VLSI physical design automation", *Kluwer Acad. Publ.*, Boston-London, 1999.
13. K. J. Supowit, "Finding a maximum planar of a set of nets in Channel", *IEEE Transactions on Computer-Aid Design*, CAD-6(1), pp. 93 — 94, 1987.
14. M. N. Swamy, K. Thulasiraman, *Graphs, Networks and Algorithms*, New York, Toronto, 1981.
15. M. Yannakakis, F. Gavril, "The maximum k -colorable problem for chordal graphs", *Information processing letters*, pp. 133 — 137, 1987.

27 апреля 2000

Ереванский государственный университет
E-mail : smarkosyan@ysu.am