TALLINNA POLÜTEHNILISE
INSTITUUDI TOIMETISED

ТРУДЫ ТАЛЛИНСКОГО
ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА

№ 411

ТРУДЫ ЭКОНОМИЧЕСКОГО ФАКУЛЬТЕТА

XXIII

ТАЛЛИН 1976

TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED
ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА

№ 411 1976

УДК 512.25
519.1
681.3
51:801

ТРУДЫ ЭКОНОМИЧЕСКОГО ФАКУЛЬТЕТА

XXШ

Вопросы исследования графов, грамматик, функции

Таллин 1976

С о д е р ж а н и е

Таллинский политехнический институт
Труды ТПИ № 411
ТРУДЫ ЭКОНОМИЧЕСКОГО ФАКУЛЬТЕТА ХХШ
Вопросы исследования графов, грамматик, функций
Редактор И. М у л л а т
Техн. редактор Л. Л о о п е р
Сборник утвержден коллегией Трудов ТПИ 30 июня 1976 г.
Подписано к печати 14 декабря 1976 г.
Бумага 60х90/16
Печ. л. 4,75 + 0,125 приложение
Уч.-изд. л. 4,2
Тираж 300
МВ - 05672
Ротапринт ТПИ, Таллин, ул. Коскла, 2/9
Зак. № 1307
Цена 42 коп.

TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED
ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА
№ 411 1976

УДК 519.1

Г.А.Вейнер

ДВЕ ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ ОПТИМАЛЬНОГО ПРИБЛИЖЕНИЯ ГРАФА ПОЛНЫМИ ПОДГРАФАМИ (обобщения)

В данной статье рассматриваются некоторые приведенные в работах [1,2] обобщения функций оценок и одновременно обобщаются доказанные там основные теоремы. В результате были получены достаточные условия, подходящие для нахождения одного частного вида класса эквивалентности отношения эквивалентности E_0. В данной работе формулируется обобщение теоремы, которая в некоторых частных случаях позволяет на графах найти все классы отношения эквивалентности E_0.

Рассмотрим кратко поставленную перед нами задачу, которая, как и в [1,2], состоит в поиске аппроксимирующего отношения эквивалентности $E \subset S \times S$ для заданного симметричного бинарного отношения $R \subset S \times S$. Целью аппроксимации является поиск такого отношения эквивалентности E_0, которое в наименьшей степени отличается от заданного отношения R, причем под наименьшей степенью отличия понимается следующее: число несовпадающих элементов заданного бинарного симметричного отношения R и отношения эквивалентности E, равное

$$\rho(E) = |R - E| + |E - R|, \tag{1}$$

должно быть минимальным. Таким образом, отношение E_0 определяется из условия

$$\rho(E_0) = \min_{E \in \varepsilon} \rho(E),$$

где ε – множество всех допустимых отношений эквивалентности на множестве S.

Далее мы рассмотрим симметричное бинарное отношение R в виде графа G. Не уменьшая общности, предположим, что граф G связен. С точки зрения постановки нашей задачи аппроксимации несущественно наличие свойства рефлексивности у отношения R , и поэтому предположим, что граф G не содержит петель. Ясно, что основным моментом поставленной задачи является задача превращения графа G в транзитивный граф так, чтобы выполнилось условие (I). Необходимые в дальнейшем определения, которые не разъяснены, используются нами исходя из соответствующих определений работ [I,2].

I. Основные понятия и определения

В приводимых ниже рассуждениях для нас существенно рассматривать связные подграфы графа G вида $G_a=(V_a,U_a)$, множество вершин которых V_a и множество ребер U_a. Предположим, что подграф G_a представляет собой максимальный полный подграф $G_0=(V_0,U_0)$ и все (максимальные полные) подграфы, инцидентные с G_0, представляют множество $T=\{G_i \mid i=1,2,\dots,k\}$. Обозначим множество индексов $I=\{1,2,\dots,k\}$. Инцидентность графов в статье [I] определяется следующим путем: максимальные полные подграфы $G_p=(V_p,U_p)$ и $G_q=(V_q,U_q)$ графа G называются инцидентными, если $V_p \cap V_q \neq \varnothing$.

Ниже мы используем следующие важные понятия грозди и максимальной грозди.

Определение I. Гроздью графа G_a нозовем множество максимальных полных подграфов $K=\{G_j \mid j\in I_k, I_k \subseteq I\}\subseteq T$, для которого подграф графа G_a, представленный в виде

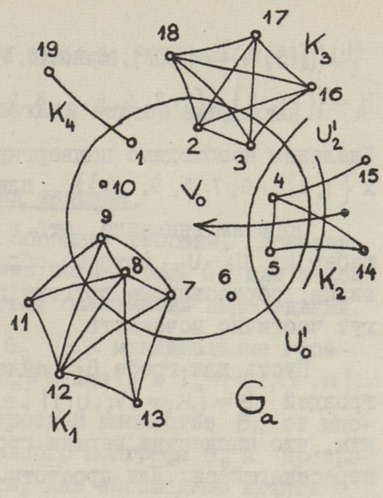
$$G_r=\Big(\{V_0\cap \bigcup_{j\in I_k} V_j\},\ \{U_0\cap \bigcup_{j\in I_k} U_j\}\Big),$$

является связным.

Множества $\{V_0\cap \bigcup_{j\in I_k} V_j\}$ и $\{U_0\cap \bigcup_{j\in I_k} U_j\}$ образуют соответственно множества вершин и ребер графа G_r.

Определение 2. Гроздь K_i назовем максимальной, если не найдется такой грозди K_m в графе G_a ,что $K_i \subset K_m$.

Пример I. На фигуре изображен граф G_a со множеством вершин $V_a=\{1,2,\dots,19\}$. У графа G_a имеется 4 максимальные грозди, составленные из множеств вершин $V_1=\{7, 8, 9, II, I2, I3\}$, $V_2=\{4, 5, I4, I5\}$, $V_3=\{2, 3, I6, I7, I8\}$ и $V_4=\{I, I9\}$. Ребра подграфа G_0 на фигуре изображены лишь частично. Причем гроздь K_1 образуется из максимальных полных подграфов на вершинах $\{7, 8, 12, 13\}$ и $\{8, 9, 11, 12\}$. Гроздь K_2 также образована из двух максимальных подграфов, в то же время грозди K_3 и K_4 состоят только из одного полного подграфа графа G_a, инцидентных подграфу G_0.



Фиг. 1.

Очевидно, что каждая максимальная гроздь K_p графа G_a соответствует определению подграфа данного графа. В этом случае, так же как и выше, множество вершин обозначается V_p и множество ребер U_p. Таким образом, $K_p = (V_p, U_p)$.

Далее мы рассматриваем случай, когда у подграфа G_a каждая пара максимальных гроздей не имеет общих вершин, вместо подграфа G_a будем говорить просто о графе G_a.

Ниже мы обращаем внимание на следующее множество вилок D. Пусть у графа G_a имеется гроздь $K_p = (V_p, U_p)$, тогда D - это множество таких вилок, одно ребро которых принадлежит множеству $U_0 - U_p$ и другое - множеству $U_p - U_0$. При удалении из графа G_a множества вилок D имеем в виду следующие три возможности:

1) можно добавить в граф G_a такое множество новых ребер H, при котором каждая вилка из D превратится в треугольник;

2) можно удалить все те ребра вилок из D, которые находятся в $U_0 - U_p$;

3) можно удалить все те ребра вилок из D, которые находятся в $U_p - U_0$.

<u>Пример 2</u>. В примере 1 графа G_a максимальная гроздь K_2 определяет множество вилок

$$D = \{\{\{(4,14),(4,15)\} \times \{(4,1),(4,2),(4,3),(4,6),(4,7),(4,8),(4,9),(4,10)\}\} \cup$$

$\cup\{\{\{(5,14)\}\times\{(5,1),(5,2),(5,3),(5,6),(5,7),(5,8),(5,9),(5,10)\}\}\}.$

$H=\{\{14,15\}\times\{1,2,3,6,7,8,9,10\}\}.$

Удалению необходимо подвергнуть множество ребер $U_0'=\{\{4,5\}\times \{1,2,3,6,7,8,9,10\}\}$ или $U_2'=\{(4,14),(4,15),(5,14)\}$.

Обратим внимание, что в случае удаления множества тех ребер из U_0-U_p, которые образуют вилки из D, некоторые вилки, образованные другими гроздьями, отличными от K_p, могут частично исчезнуть.

Пусть для графа G_0 найдено множество максимальных гроздей $B=\{K_i=(V_i,U_i)\mid i\in I_K\}$, $I_K=\{1,2,\dots,k\}$. Предположим, что множества вершин гроздей множества B – попарно непересекающиеся. Для простоты введем еще обозначения $|V_0|=\nu_0$, $|V_0\cap V_i|=\nu_i$ и $|V_i-V_0|=m_i$, если $i\in I_K$. Обозначим также через L_i, если $i\in I_K$, число тех ребер, которые соединяют вершины множеств $V_0\cap V_i$ и V_i-V_0 максимальной грозди K_i. Например, в нашем графе на фигуре для максимальной грозди K_2 число $L_2=3$.

<u>Определение 3</u>. Функции

$$f(K_i)=\frac{(\nu_0-\nu_i)m_i}{L_i} \tag{2}$$

и

$$f(G_0^i)=\frac{(\nu_0-\nu_i)m_i+\nu_i(M-m_i)}{(\nu_0-\nu_i)\nu_i}, \tag{3}$$

где $M=\bigcup\limits_{i\in I_K} m_i$, назовем функциями оценок грозди K_i и подграфа G_0 соответственно.

Обратим внимание, что $(\nu_0-\nu_i)m_i=|H|$. В числителе выражений (2) и (3) указано необходимое число добавляемых новых ребер, для того чтобы исчезло некоторое определенное число вилок, а в знаменателе указано число ребер, которое удаляется для исчезновения тех же вилок.

<u>Определение 4</u>. Функцию

$$f(K_i,G_0)=\frac{f(K_i)}{f(G_0^i)} \tag{4}$$

назовем функцией оценки грозди K_i по отношению к графу G_0.

Последнюю функцию сопоставим с вектором. При $f(K_i,G_0)>1$ вектор направлен от грозди K_i в подграф G_0, а при $f(K_i,G_0)<1$ – в обратном направлении.

Пример 3. В графе первого примера $f(G_0)=\frac{15}{8}$, $f(K_2)=\frac{16}{3}$ и $f(K_2,G_0)>1$, а соответствующий вектор направлен в граф G_0.

2. Основная теорема.

Приведенная ниже теорема обобщает теоремы, доказанные в статьях [I,2], для случаев, когда граф G или какой-нибудь его подграф G_a являются описанными выше видами.

Итак, рассмотрим в графе G_a $n \leq k$ максимальные гроздья вида $B_n=\{K_i \mid i \in I_n\}$, где $B_n \subseteq B, I_n \subseteq I_k$ и $I_n=\{1,2,\ldots,n\}$. Производя операцию отделения гроздей множества B_n от множества вершин максимального полного подграфа G_0 и превращая их в классы эквивалентности, нам необходимо изъять

$$\sum_{i=1}^{n}(v_0-\sum_{j=1}^{i}v_j)v_i \qquad (5)$$

ребер из подмножества ребер U_0 подграфа G_0. В случае, когда требуется изъять ребра гроздей, соединяющих множества вершин V_i-V_0 и $V_0 \cap V_i$, требуется разрубить

$$\sum_{i=1}^{n}L_i \qquad (6)$$

ребер. Очевидно, что приведенные выше выражения (5) и (6) остаются в силе для любого подмножества B_s максимальных гроздей B, если множество индексов I_s выполняет условие $I_s \subseteq I_k$.

Теорема. Если для графа G_a выполняется:

I) векторы функции $f(K_i,G_0)$ направлены в подграф G_0;

2) $L_i < \frac{1}{2}v_0 m_i$ и

3) $\sum_{i=1}^{k}(v_0-\sum_{j=1}^{i}v_j)v_i \geq \sum_{i=1}^{k}L_i$,

для $i=1,2,\ldots,k$, то существует такое аппроксимирующее отношение E_0, которое имеет класс эквивалентности V_0.

Доказательство. Покажем сперва, что не найдется отношения эквивалентности E_0 с классом эквивалентности V' и таким, что $V_i \cup V_0 \subseteq V'$, причем $K_i \in B$. Действительно, из условия (2) теоремы тождественным преобразованием получается, что $L_i < v_0 m_i - L_i$. Это означает, что только при соединении всех вершин множества $V_i - V_0$ с подмножеством вершин V_0 приходится добавлять $v_0 m_i - L_i$ ребер, что больше числа L_i ребер, подлежащих удалению при отделении множества вершин $V_i - V_0$ в самостоятельный класс эквивалентности. Тем самым V' не может быть классом эквивалентности отношения эквивалентности E_0.

Далее рассматриваем функцию

$$\varphi(q) = \frac{\sum_{i=1}^{q} (v_0 - \sum_{j=1}^{i} v_j)\, v_i}{\sum_{i=1}^{q} L_i}, \quad q \leqslant k, \tag{7}$$

где в числителе стоит число ребер, подлежащих изъятию из U_0 при отделении максимальных гроздей K_i, $i = 1, 2, \ldots, q$, в самостоятельные классы эквивалентности, а в знаменателе функции - число ребер, разделяющее подмножества $V_i - V_0$ от V_0. Покажем, что функция (7) монотонна. Для этого предположим сперва, что для индекса выполняется условие

$$\frac{L_{i+1}}{v_{i+1}} \geqslant \frac{L_i}{v_i}, \quad \text{где} \quad i = 1, 2, \ldots, k-1. \tag{8}$$

Выполняется также условие

$$\frac{v_0 - v_1 - \ldots - v_q}{\frac{L_q}{v_q}} \geqslant \frac{v_0 - v_1 - \ldots - v_{i+1}}{\frac{L_{i+1}}{v_{i+1}}}, \quad \text{если } q < i + 1. \tag{9}$$

Неравенство (9) выполняется, так как числитель левой части больше числителя правой части, в то время как знаменатель из-за условия (8) не больше.

Из неравенства (9) получаем

$$L_{i+1}(v_0 - v_i - \ldots - v_q)\, v_q \geqslant L_q(v_0 - v_1 - \ldots - v_{i+1})\, v_{i+1}. \tag{10}$$

$$\varphi(i) - \varphi(i+1) = \frac{(v_0 - v_1)v_1 + (v_0 - v_1 - v_2)v_2 + \ldots + (v_0 - v_1 - \ldots - v_i)v_i}{L_1 + L_2 + \ldots + L_i} -$$

$$- \frac{(v_0 - v_1)v_1 + \ldots + (v_0 - v_1 - \ldots - v_i)v_i + (v_0 - v_1 - \ldots - v_i - v_{i+1})\, v_{i+1}}{L_1 + L_2 + \ldots + L_i + L_{i+1}} =$$

$$= \frac{1}{N}\left\{ L_{i+1}\left[(v_0 - v_1)v_1 + \ldots + (v_0 - v_1 - \ldots - v_i)v_i\right] - \right.$$

$$-(\nu_0-\nu_1-\ldots-\nu_i-\nu_{i+1})\nu_{i+1}(L_1+L_2+\ldots+L_i)\}, \qquad (\text{II})$$

$$N=\sum_{j=1}^{i} L_j \cdot \sum_{j=1}^{i+1} L_j .$$

Здесь в разности (II) в свою очередь разности выражений $L_{i+1}(\nu_0-\ldots-\nu_q)\nu_q$ и $L_q(\nu_0-\ldots-\nu_{i+1})\nu_{i+1}$, где $q=1,2,\ldots,i$, не отрицательны вследствие условия (IO).

Последнее утверждение приводит к неравенству $\varphi(i)-\varphi(i+1)\geq 0$, что доказывает монотонность функции (7).

Рассмотрим теперь значения функции (7) в концах дискретного интервала $[1,\kappa]$. По предположению (3) $\varphi(\kappa)\geq 1$. Тем самым класс эквивалентности V_0 не хуже классов эквивалентности, полученных путем превращения гроздей K_i. Условие теоремы (I) тождественно условию

$$\frac{(\nu_0-\nu_i)m_i}{L_i} > \frac{(\nu_0-\nu_i)m_i+(M-m_i)\nu_i}{(\nu_0-\nu_i)\nu_i}, \qquad (\text{I2})$$

$i=1,2,\ldots,\kappa$, из которого получим

$$\frac{(\nu_0-\nu_i)\nu_i}{L_i} > 1+\frac{(M-m_i)\nu_i}{(\nu_0-\nu_i)m_i}, \qquad (\text{I3})$$

причем очевидно, что $\dfrac{(M-m_i)\nu_i}{(\nu_0-\nu_i)m_i}>0$. Тем самым при удалении точно одной грозди K_i от множества V_0 не образуется оптимального разбиения E_0. Из-за условия монотонности (7) можно заключить, что не найдется отношения эквивалентности E_0, класс эквивалентности из множества B у которого меньше κ и больше I. Это доказывает теорему.

3. Обобщения

Расширим теперь теорему на случай, когда в графе G_a найдутся гроздья, содержащие общие вершины.

Определение 5. Назовем подмножество K_t, $t\in I_\kappa$, множества максимальных гроздей B составной гроздью, если подграф $(\bigcup_{t\in I_\kappa} V_t, \bigcup_{t\in I_\kappa} U_t)$ графа G_a связен.

По аналогии с определением 2 определим понятие максимальной составной грозди.

Очевидно, что в доказательстве теоремы для случая, когда во множество максимальных гроздей графа G_a входят составные максимальные гроздья, не следует вносить существенных изменений. Теперь уже несложно показать, что такое расширение понятия максимальной грозди позволяет описать любой подграф G_a неориентированного графа G.

Нетрудно заметить, что в случае выполнения условия (I2), если числа L_i заменить на $v_i m_i$, то, опираясь на результаты работы [2], можно утверждать, что найдется такое оптимальное разбиение E_0, у которого одним из классов эквивалентности будет V_0. То же самое можно утверждать, если выполняется условие (I2), при замене числа m_i числом m_i', где для m_i' выполнено условие $v_i m_i' = L_i$.

Литература

I. Вейнер Г.А. Две вспомогательные функции оптимального приближения графа полными подграфами. - "Тр. Таллинск. политехн. ин-та", 1974, № 366.

2. Вейнер Г.А. Две вспомогательные функции оптимального приближения графа полными подграфами (дополнение). - "Тр. Таллинск. политехн. ин-та", 1975, № 386.

G. Veiner

Two Auxiliary Functions for the Optimal Approximating of a Graph by Complete Subgraphs (Generalizations)

Summary

This paper, as the previous two [1,2] , deals with the problem of finding an optimal partition E_0, which best approximates a given symmetric relation R in the sense of minimizing the number of the elements of

$$\{E_0 - R\} \cup \{R - E_0\}.$$

The main results of the papers [1,2] are generalized.

TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED
ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА
№ 4II 1976

УДК 5I9.I

Р.А.Палуоя, Р.Э.Куусик

ПОИСК ВНУТРЕННЕ УСТОЙЧИВОГО МНОЖЕСТВА НА ДЕРЕВЕ

Введение. Одним из важнейших понятий теории графов является внутренне устойчивое множество. Для решения задачи поиска наибольших внутренне устойчивых множеств в общем виде, когда граф содержит довольно много вершин, в настоящее время отсутствуют эффективные методы. Ряд алгоритмов принадлежит таким авторам, как Эдмондс, Хакими, Франк и др. По терминологии теории алгоритмов эти процедуры поиска являются итеративными с экспоненциальным ростом количества вычислений. В частном случае, например, на деревьях, можно использовать некоторые особенности. Здесь мы рассмотрим класс деревьев: конечные связные графы без циклов, имеющие по крайней мере две вершины.

Нами выбраны и доказаны следующие алгоритмы и леммы: (Используемые понятия в основном заимствованы из [2]. Алгоритмы реализованы на ЭВМ и проверены на эксперименте, что дает возможность утверждать, что они достаточно эффективны).

I. Основные понятия и теоремы

Определение I. Граф $G_1=(V_1,U_1)$ мы называем частью графа $G=(V,U)$, если $V_1\subseteq V$ и $U_1\subseteq U$.

Определение 2. Отображение вершины x Γx задается здесь следующим образом: $\Gamma x=\{y\,|\,(x,y)\in U\}$.

Определение 3. Вершина x называется соседом вершины y, если $x\in\Gamma y$.

Определение 4. Вершина x называется изолированной вершиной, если $|\Gamma x|=0$.

Определение 5. Вершина x – висячая вершина, если $|\Gamma x|=1$.

Определение 6. Цепь называется внешней цепью, когда по крайней мере одна вершина цепи является висячей вершиной.

Определение 7. Вершина называется средней вершиной, если $|\Gamma x| > 2$.

Определение 8. Множество S мы называем внутренне устойчивым множеством, если оно не содержит соседних вершин.

Определение 9. Множество S мы называем полным внутренне устойчивым множеством, если при добавлении любой вершины $V_i \notin S$ в множество S множество $\{V_i\} \cup S$ не является внутренне устойчивым множеством.

Определение I0. Наибольшее по включению элементов множество S называется здесь максимальным.

Определение II. Цепь (или цикл) называется простой, если каждое ее ребро входит в последовательность ровно один раз.

Определение I2. Граф G называется нуль-графом, если он не содержит ни одного ребра и вершины; обозначение: $G = (\Phi, \Phi)$.

Определение I3. Вершина x называется запрещенной вершиной относительно множества S, если $x \in \Gamma S$.

Определение I4. Вершину x называем свободной, если $x \notin S \cup \Gamma S$.

Теорема I. Цикломатическое число $\nu(G)$ мультиграфа G равно наибольшему количеству независимых циклов [2].

Следствие. Граф G не имеет циклов тогда и только тогда, когда $\nu(G) = 0$ [2].

Теорема 2. Всякое дерево имеет по крайней мере две висячие вершины [2].

Теорема 3. Для того, чтобы конечный связной граф был деревом, необходимо и достаточно, чтобы число его ребер было на единицу меньше числа вершин [3].

Теорема 4. Граф связен в том и только в том случае, когда он состоит из единственной компоненты [2].

Теорема 5. Пусть G' – мультиграф, полученный из мультиграфа G добавлением нового ребра между вершинами a и b;

если a и b совпадают или могут быть соединены цепью в G, то $\nu(G')=\nu(G)+1$; в противном случае $\nu(G')=\nu(G)$ [2].

Следствие. $\nu(G)\geqslant 0$ [2].

Теорема 6. Всякие две вершины дерева связаны ровно одной элементарной цепью [3].

Алгоритм DISTRIBUTION COUNT[I] мы используем для упорядочения множества M в неуменьшающемся порядке по числу соседних вершин. Множество M здесь содержит степени инцидентности вершин. В результате упорядочения мы получим множество UM. Посредством SMR обозначим номера вершин упорядоченного множества M.

2. Леммы.

Лемма I. Пусть x – висячая вершина, т.е. $|\Gamma x|=1$, и y – ее сосед и пусть соседом вершины y будут еще y_i. Тогда вершина x должна принадлежать множеству S, так как добавление y в S уменьшает мощность S.

Доказательство. Могут представиться две возможности:

I. Если $x\in S$, то множеству S не могут принадлежать вершины y, входящие в Γx.

2. Если $y\in S$, то множеству S не могут принадлежать вершины Γy. Множество Γy составляют вершины x и y_i. Таким образом, $|\Gamma y|>1$. Так как $|\Gamma y|>|\Gamma x|$, то можно заключить, что при добавлении y в множество S число запрещенных вершин (определение I3) станет больше, чем при вершине x. Отсюда и следует преимущество висячей вершины при построении множества S.

Лемма 2. Если вершина x изолированная, то $x\in S$.

Лемма 3. Если из дерева G удалить одно ребро, то число компонент связности $p=2$.

Доказательство. Пусть даны два дерева A и B. Очевидно, что $a_i\in A$ не связана цепью с вершиной $b_i\in B$. При добавлении ребра (a_i, b_i) между вершинами a_i и b_i, мы получим граф G, для которого $m_G=m_A+m_B$ [1), $n_G=n_A+n_B$. Так как

1) m_A – число ребер в дереве A; n_A – число вершин в дереве A, (m_B, n_B, m_G, n_G – соответственно).

$m_A = n_A - 1$ и $m_B = n_B - 1$, то $m_G = n_G - 2 + |\{(a_i, b_i)\}| = n_G - 1$.

Докажем, что если G связной граф, то он есть дерево. Вершина $a_i \in A$ связана цепью с каждой вершиной A, потому что A - дерево и $b_i \in B$ связана с каждой вершиной B. Любая вершина $a_j \in A$ связана цепью с любой $b_j \in B$, потому что существуют цепь (a_j, a_i), ребро (a_i, b_i) и цепь (b_i, b_j). Следовательно, существует цепь (a_j, b_j). Получаем, что G будет связным графом и, следовательно, по теореме 3 является деревом.

Обратно, удаляя из графа G ребро (a_i, b_i), мы восстанавливаем прежнюю ситуацию, т.е. имеем ровно две компоненты связности - A и B.

Лемма 4. Если из дерева удалить висячую вершину вместе с инцидентным ребром, то оставшаяся часть является деревом.

Доказательство. После удаления ребра, инцидентного с висячей вершиной, число компонент связности $p_1 = 2$ на основе леммы 3. Одной из компонент является прежняя висячая вершина. $m_1 = m - 1 = n - 2$, $n_1 = n$,

$$\nu_1 = m_1 - n_1 + p_1 = n - 2 - n + 2 = 0.$$

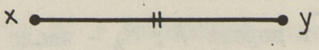
Далее, удаляя образовавшуюся изолированную вершину, получим $p_1 = 1$, $n_1 = n - 1$, $m_1 = n - 2$ и $\nu_1 = n - 2 - n + 1 + 1 = 0$. Так как $p_1 = 1$ и $\nu_1 = 0$, то, следовательно, G_1 - дерево.

Лемма 5. Если из дерева удалить висячую вершину вместе с ее соседом и с инцидентными соседу ребрами, то реализуется одна из нижеследующих возможностей, а именно:

1) нуль-граф;
2) изолированная вершина;
3) дерево;
4) лес (совокупность деревьев или изолированных вершин).

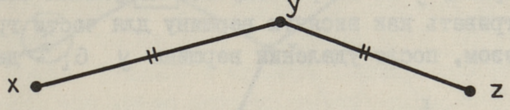
Доказательство. Пусть x - висячая вершина и y ее сосед. Соответственно существуют четыре возможности:

1. Граф G двухвершинный. Удаление x и y приводит в нуль-графу.



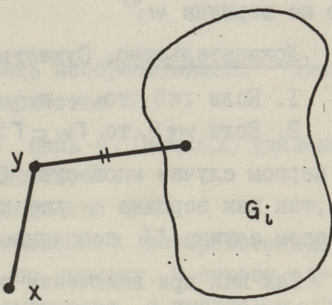
Фиг. 1.

2. Граф G трехвершинный. После удаления x и y остается изолированная вершина z.



Фиг. 2.

3. $|\Gamma y| = 2$ и $n > 3$, где n – число вершин графа G. После удаления вершины x и инцидентного ей ребра, получаем $|\Gamma y| = 1$, т.е. y превращается в висячую вершину. Возникшая часть графа будет деревом (лемма 4). Та же ситуация возникает при удалении вершины y.



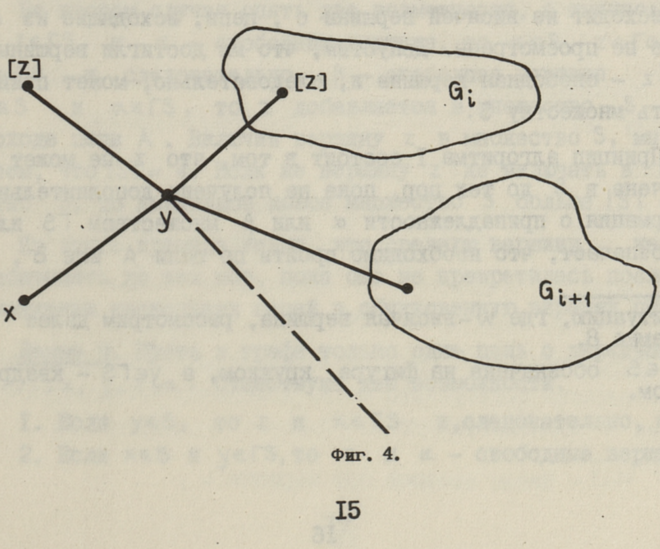
Фиг. 3.

4. $|\Gamma y| > 2$ и $n > 3$.

Соответственно существуют две возможности:

I. Сосед вершины y – висячая вершина z, т.е. $|\Gamma z| = 1$.

2. $|\Gamma z| > 1$.



Фиг. 4.

Рассмотрим их по порядку. Если из графа удалить x и y, то $|\Gamma z|=0$. Вершину y во втором случае, не умаляя общности, можно рассматривать как висячую вершину для части графа G_i. Таким образом, после удаления вершины y G_i – дерево (лемма 4).

Лемма 6. Пусть вершины x, y, z, w последовательные вершины цепи. Если $x \in S$ и $y \notin S$, т.е. $y \in \Gamma S$, и z и w – свободные вершины (определение I4), то множеству S принадлежит вершина z, но не вершина w.[1)]

Доказательство. Существуют две возможности.

I. Если $z \in S$, то w и $y \in \Gamma S$.

2. Если $w \in S$, то $\Gamma w \subset \Gamma S$.

В первом случае множество ΓS пополнилось только вершиной w, так как вершина y уже принадлежала множеству ΓS. Во втором случае ΓS пополнилось на Γw, $|\Gamma w| > 1$.

Так как при включении вершины w в множество S множество ΓS пополнилось больше чем на одну вершину, то отсюда следует преимущество вершины z при построении наибольшего внутренне устойчивого множества. Вершина z – первая свободная вершина в цепи вершин.

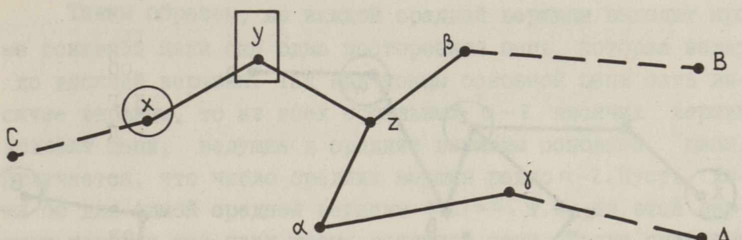
Лемма 7. Пусть дана часть графа (дерева). Пусть $x \in S$ и $y \in \Gamma S$,[2)] z, α, β, γ – свободные вершины и A, B, C – висячие вершины. Пусть при просмотре какой-либо цепи, которая исходит из висячей вершины C, цепи, исходящие из A и B, еще не просмотрены. Допустим, что мы достигли вершины z, где z – свободная вершина и, следовательно, может принадлежать множеству S.

Принцип алгоритма I состоит в том, что z не может быть включена в S до тех пор, пока не получена дополнительная информация о принадлежности α или β множествам ΓS или S. Это означает, что необходимо пройти по цепи A или B, а

---

1) Ситуацию, где w-висячая вершина, рассмотрим далее в лемме 8.

2) $x \in S$ обозначена на фигуре кружком, а $y \in \Gamma S$ – квадратом.



Фиг. 5.

при просмотре C вершину z оставить необработанной. Этот принцип не уменьшает мощности множества S.

Доказательство. Рассмотрим цепь A. Цепь(C,z) удалена из графа. Существуют две возможности.

1. Вершину z включаем в множество S.

2. При просмотре цепи C оставляем z необработанной.

Рассмотрим эти возможности по порядку. В первом случае представляются вновь две возможности, а именно: $\gamma \in \Gamma S$ и α – свободная вершина. Так как $z \in S$, то имеется противоречие с леммой 6, потому что из вершин, принадлежащих $\Gamma\alpha$, свободной является только z, а из множества Γz свободными являются α и β. Если $\gamma \in S$ и $\alpha \in \Gamma S$, то добавление вершины z в множество S произошло бы при проходе цепи A.

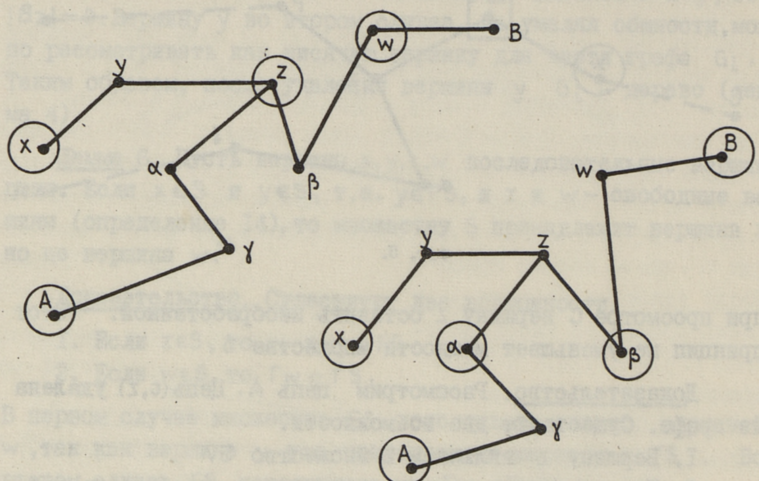
Во втором случае опять две возможности, а именно: если $\gamma \in \Gamma S$ и α – свободная вершина, то $\alpha \in S$, $z \in \Gamma\alpha$ и $z \in \Gamma S$ и, следовательно, β – свободная вершина. Если $\gamma \in S$ и $\alpha \in \Gamma S$, то z добавляется в множество S при проходе цепи A. Включив вершину z в множество S, мы получаем, что $|S| = 4$. Если же вершину z не включать в множество S, то найденное новое множество S больше $|S| = 5$.

Из этого примера видно, что среднюю вершину нельзя обрабатывать до тех пор, пока она не превратилась после прохождения нескольких цепей в обыкновенную вершину цепи.

Лемма 8. Пусть в графе только одна цепь с вершинами $A, \ldots, x, y, z, \alpha$. Существуют две возможности.

1. Если $y \in S$, то z и $x \in \Gamma S$ и, следовательно, $\alpha \in S$.

2. Если $x \in S$ и $y \in \Gamma S$, то z и α – свободные вершины.



Фиг. 6.

На основе леммы 2 висячая вершина α имеет преимущество при построении множества S перед другими вершинами цепи (лемма I). В случае, когда α - висячая вершина, возникает такая ситуация, что первой свободной вершиной при прохождении цепи является вершина z. На основе леммы 6 $z \in S$. Включение вершины z не уменьшает мощности S и, следовательно, не возникает противоречия между леммами I и 6.

<u>Доказательство</u>. Рассмотрим следующие две возможности:

I. Если $z \in S$, то $|S| = |S| + 1$ и $|\Gamma S| = |\Gamma S| + 1$.

2. Если $\alpha \in S$, то $|S| = |S| + 1$ и $|\Gamma S| = |\Gamma S| + 1$.

Таким образом, при проходе цепи нас не интересует, является ли сосед вершины z висячей вершиной или нет.

<u>Лемма 9.</u> В дереве с n висячими вершинами имеется максимально $n-2$ средних вершин.

<u>Доказательство</u>. Пусть в дереве дана цепь, из которой разветвляются посторонние цепи. Допустим, что для средних вершин i выполняется $M(i) = 3$.[1)]

[1)] $M(i)$ - число соседов при вершине i.

Таким образом, из каждой средней вершины выходит кроме основной цепи еще одна посторонняя цепь, которая ведет до висячей вершины. Так как концы основной цепи есть висячие вершины, то из всех остальных $n-2$ висячих вершин исходят цепи, ведущие в средние вершины основной цепи. Получается, что число средних вершин равно $n-2$. Пусть хотя бы для одной средней вершины $M(i)=4$, т.е. из этой вершины исходят две цепи кроме основной цепи. Тогда существуют две висячие вершины, из которых исходят цепи, пройдя по которым мы попадаем в одну и ту же среднюю вершину. Таким образом, число средних вершин уменьшилось на одну вершину и т.д.

Лемма IO. Для средних вершин дерева выполняется

$$\sum_{M(i)\geq 3}(M(i)-2)=n-2.$$

Доказательство следует непосредственно из леммы 9.

Лемма II. В результате применения алгоритма I дерево превращается в одну цепь после прохождения ровно $(n-2)$-х цепей.

Доказательство. Пусть из средней вершины i ведет цепь в висячую вершину и пусть $M(i)=3$. Если эту цепь удалить из дерева, то $M(i)=M(i)-1$. В результате такой операции ликвидировались одна висячая и одна средняя вершины.

То же можно показать и для средней вершины с $M(i)>3$. В этом случае для ликвидации одной средней вершины следует пройти $M(i)-2$ цепи и удалить из дерева $M(i)-2$ висячих вершин. В графе остается одна цепь и две висячие вершины.

Следствие. В алгоритме I требуется $n-1$ прохождений цепей.

3. Алгоритм I.

AI. На основе матрицы A находится ряд чисел $M(j)$, где j - номер вершины дерева; множества B и R сводятся к нулю.

A2. Образуется множество SMR номеров висячих вершин.

A3. Выполняем A3 — AI2 для $i=1,\ldots,N$.

A4. $k=SMR(i)$.

A5. Отыскиваем соседа j вершины k.

А6. Если $B(к)=0$[1], то выполним А7, в противном случае – А8.

А7. Приравняем $R(к)=1$[2] и $B(j)=1$.

А8. Если $M(j)\geqslant 2$, то выполняем А9, в противном случае – АIО.

А9. Приравняем $M(к)=0$ и $M(j)=M(j)-1$, выполняем АI2.

АIО. Если $M(j)=1$, то выполняем АI3.

АII. Приравняем $M(к)=0$ и $к=j$, выполняем А5.

АI2. Конец цикла.

АI3. Если $B(j)=0$, то $R(j)=1$.

Конец алгоритма.

Теорема. Найденное на основе алгоритма I множество S есть наибольшее внутренне устойчивое множество.

Доказательство. Прохождением цепей дерева и включением свободных вершин в множество S, дерево превращается в одну цепь (лемма II). После удаления каждой цепи мы получаем дерево (лемма 4). Тем самым правило алгоритма вновь применимо на каждой возникшей части дерева (теорема 2). В множество S добавляют висячие вершины (лемма I) и первые свободные вершины (лемма 6). Тем самым происходит построение множества S добавлением "лучших" вершин, которые гарантируют отыскание наибольшего внутренне устойчивого множества.

4. Алгоритм 2.

АI. Найдем число единиц строчки j матрицы A и увеличим это число на единицу, получим $M(j)$, j – номер вершины дерева.

А2. Упорядочим множество M в неуменьшающемся порядке по числу соседних вершин (алгоритм DISTRIBUTION COUNT [1]).

А3. Выполним А4–АI4. $i=1,\ldots,N$.

А4. $к=SMR(i)$.

А5. Если $UM(i)=0$, то выполняем АI6.

А6. Если $UM(i)=1$, то добавляем вершину $к$ в множество S, $M(к)=0$ и выполняем АI6.

---

[1] $B(к)=0$ – признак свободной вершины.

[2] Если $R(j)=1$, то вершина $j\in S$.

A7. Если $\cup M(i)=2$ и $M(k)=2$, то выполняем A8–A12.

A8. Найдем соседа j вершины k. Вершину k добавляем в S.

A9. Если $M(j)>3$, то найдем соседа l вершины j, при которой $l \neq k$. Для каждой вершины l $M(l)=M(l)-1$, если $M(l)\neq 0$. $M(j)=0$, $M(k)=0$, выполняем A16.

A10. Если $M(j)=2$, то $M(k)=0$, $M(j)=0$, выполняем A16. В противном случае найдем соседа l вершины j, при которой $l \neq k$ и $M(l)\neq 0$.

A11. Если $M(l)>3$, то $M(k)=0$, $M(j)=0$, $M(l)=M(l)-1$, выполняем A16.

A12. Если $M(l)=2$, то $M(k)=0$, $M(j)=0$, $M(l)=0$, добавляем вершину l в S и выполняем A16. В противном случае $M(k)=0$, $M(j)=0$, $k=l$, выполняем A8.

A13. Если $\cup M(i)=2$ и $M(k)=1$, то добавляем вершину k в S, $M(k)=0$, выполняем A16.

A14. Если $(\cup M(i)=2 \vee \cup M(i)\geq 3)$ и $M(k)=0$, то выполняем A16.

A15. Сортируем множество S (алгоритм DISTRIBUTION COUNT [1]).

A16. Конец цикла.

Конец алгоритма.

Движение начинается по цепи, исходящей из висячей вершины. Висячая вершина добавляется в S. Висячая вершина и ее сосед удаляются из дерева. Сосед соседа может превратиться в висячую вершину. Таким образом двигаемся по цепи. Если в цепи до средней вершины имеется четное число вершин, то средняя вершина не обрабатывается (лемма 7), а если пройденное число нечетное, то эта средняя вершина удаляется. В этом случае дерево распадается на изолированные вершины и деревья (лемма 5). Далее осуществляется проход цепей согласно методу, изложенному в алгоритме I. Так как построение множества S находится в соответствии с доказанными леммами, то гарантируется наибольшее внутренне устойчивое множество.

З а к л ю ч е н и е.

Рассмотренные выше алгоритмы по своим рабочим принципам достаточно различны. Поэтому необходимы оценки эффективности каждого алгоритма в специальных случаях. Эти вопросы нами не рассматриваются. Целью работы было решение задачи поиска наибольшего внутренне устойчивого множества на деревьях.

Литература

1. Knuth D.E., The Art of Computer Programming. Vol. 3. Sorting and Searching. Addison-Wesly Publishing Company, 1973.

2. Берж К. Теория графов и ее применение."ИЛ",1962.

3. Дискретная математика и математические вопросы кибернетики. Под редакцией Яблонского С.В. и Луганова О.Б. М., "Наука", 1974.

R. Paluoja, R. Kuusik

Finding the Internally Stable Set of a Tree

Summary

This paper deals with the problem of finding the maximum internally stable set of a graph if this graph is a tree.

TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED
ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА
№ 411 1976

УДК 681.3:519.1

К.К.-Э. Аллик

КОНСТРУКТИВНЫЙ ПОДХОД К ОПИСАНИЮ ЛОКАЛЬНЫХ ПРЕОБРАЗОВАНИЙ ГРАФОВ

1. Введение. Преобразования (трансформация) графов необходимы весьма часто в самых различных областях науки, в которых применяется аппаратура теории графов. К примеру, понятием преобразования графов пользуются в программировании (операторные схемы, структуры данных), математической лингвистике (семантика естественных языков), химии (синтез химических соединений). Характерно, что несмотря на полное различие в целях и интерпретации преобразований, практически во всех случаях в основу понятия преобразования графов заложен принцип локальности: преобразование влияет только на одну часть преобразуемого графа, оставляя остальную часть графа без изменений. Однако до сих пор нет достаточно точного и строгого определения этого принципа в терминах теории графов.

В настоящей работе предлагается универсальная методика описания локальных преобразований графов, которая является обобщением ряда известных конструктивных подходов к этой проблеме [6,7].

Работа носит методический характер и поэтому вместо формального изучения вопросов мы, где это только возможно, прибегаем к словесным объяснениям и графическим иллюстрациям.

2. Основные определения и примеры. Пусть $G=(V,U)$ – ориентированный граф (далее – просто граф) без параллельных дуг, где $V=V(G)$ – множество вершин, $U=U(G)$ – множество дуг, причем $U \subset V \times V$.

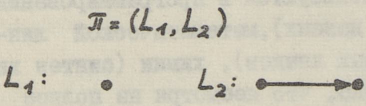
Графы $G'=(V',U')$ и $G''=(V'',U'')$ называются изоморфными ($G' \equiv G''$), если существует взаимно-однозначное соответствие

$f: V' \to V''$ такое, что $(x,y) \in U' \Leftrightarrow (f(x), f(y)) \in U''$.

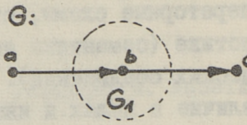
Граф $G_1 = (V_1, U_1)$ называется подграфом графа $G=(V,U)$, если $V_1 \subset V$ и $U_1 = U \cap V_1 \times V_1$.

Теперь мы в состоянии приступить к рассмотрению преобразований графов. Правило преобразования π – это упорядоченная пара $\pi = (L_1, L_2)$, где L_i – некоторые графы. Применение преобразования π к графу G означает, что некоторый подграф G_1 графа G, который изоморфен графу L_1, заменяется графом L_2, а остальная часть графа G остается без изменений.

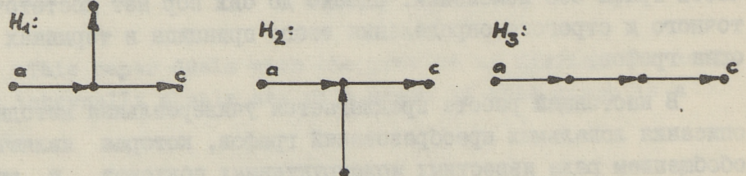
Однако, как будет видно из следующих примеров, такая схема слишком примитивна и требует серьезных уточнений.



Фиг. 1.



Фиг. 2.



Фиг. 3.

Пример I. Рассмотрим правило преобразования π, изображенное на фиг. I. Это весьма элементарное преобразование, однако применение π к подграфу G_1 графа G (фиг.2) может привести к трем различным результатам (фиг. 3). Такую неопределенность можно устранить, если установить некоторое соответствие между вершинами левой и правой части правила преобразования. Граф H_3 (фиг. 3) свидетельствует о том, что упомянутое соответствие должно как-то быть связано с правилами "привязки" графа L_2 к графу G.

Пример 2. Предположим теперь, что нам требуется преобразование типа π (фиг. I), которое применимо не к произвольным, а только к висячим вершинам графа G (к вершине с на фиг. 2). Свойство быть висячей вершиной - вполне локальное свойство и должно как-то описываться в терминах локальных преобразований графов. Это легко сделать, если применимость правила преобразования к некоторому подграфу поставить в зависимость не только от этого подграфа, но и его <u>контекста</u>, т.е. непосредственного окружения этого подграфа.

Основное отличие предлагаемой здесь методики от ранее известных подходов заключается в том, что она позволяет в правилах преобразований учитывать такого рода контекстные условия.

Нецелесообразно включать в правило преобразования все окружение заменяемого подграфа. Во-первых, это противоречит принципу локальности, и, во-вторых, приводит к немалым техническим трудностям. Поэтому мы предлагаем вместо всего окружения использовать некоторое его приближение, получающееся из исходного окружения операцией стягивания. Чтобы уточнить вышесказанное, перейдем к более точному изложению.

Говорят, что графы $G'=(V',U')$ и $G''=(V'',U'')$ являются разбиением графа $G=(V,U)$, если G' и G'' - подграфы графа G, $V' \cap V'' = \phi$ и $V' \cup V'' = V$.

Если G' и G'' - разбиение графа G, то множество дуг, не "попавших" в G' или G'', образует <u>сечение</u> $S=S(G',G'')$ графа G. Тройку $F(G)=(G',G'',S(G',G''))$ будем называть <u>разложением</u> графа G.

Пусть $F(G)$ - разложение графа G. Определим <u>стягивание</u> B сечения $S'(G',G'')$ к графу $G'=(V',U')$ следующим образом:

$$B(S,G')=(V_0,U_0),$$

где $V_0 = V' \cup \{\mu,\nu\}$; $\mu,\nu \notin V'$; $U_0 = U' \cup U_1 \cup U_2$,

причем $U_1 = \{(\mu,y) : (\exists x)[(x,y) \in S \,\&\, y \in V']\}$,

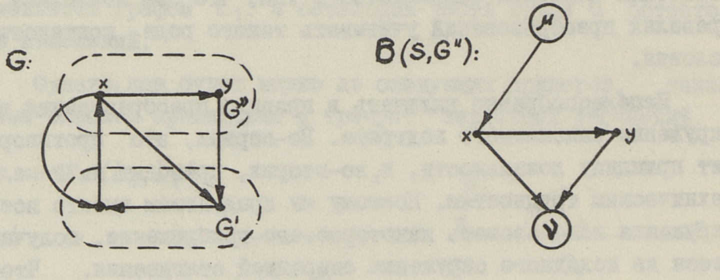
$U_2 = \{(y,\nu) : (\exists z)[(y,z) \in S \,\&\, y \in V']\}$.

Очевидно, что $B(S,G')$ - граф, который получен из графа G' добавлением двух новых вершин μ,ν и некоторых дуг. Начальные вершины всех дуг из сечения, заходящих в граф G', стянуты в вершину μ, а конечные вершины дуг, исходящих из

G' – в вершину ν, причем дуги сечения S подчинены этому процессу. Вершина μ играет роль обобщенного входа, а ν – обобщенного выхода подграфа G'.

Стягивание $B(S, G'')$ определяется аналогичным образом.

Пример 3. На фиг. 4 приведен граф G, его разбиение G' и G'', а также граф $B(S', G'')$.



Фиг. 4.

Необходимо подчеркнуть, что вершины μ и ν играют совершенно особую роль в стягивании B сечения S к графу G' (или G''). Поэтому мы будем писать $B = B(S', G')\langle\mu,\nu\rangle$. Вообще, если нам будет необходимо в некотором графе G выделить две вершины x и y, то мы будем писать $G = G\langle x, y\rangle$.

Графы $G' = G'\langle x', y'\rangle$ и $G'' = G''\langle x'', y''\rangle$ называются 2-изоморфными, если существует такой изоморфизм f графов G' и G'', что $f(x') = x''$, $f(y') = y''$ (обозначение 2-изоморфизма: $G' \doteqdot G''$).

Введем еще несколько стандартных обозначений. Пусть $G = (V, U)$ – граф, $W \subset V$, $\overline{W} = V \setminus W$, тогда

$$U^-(W) = \{(x, y) \in U : (x, y) \in W \times \overline{W}\};$$
$$U^+(W) = \{(y, z) \in U : (y, z) \in \overline{W} \times W\};$$
$$V^-(W) = \{\, x : (x, y) \in U^-(W)\};$$
$$V^+(W) = \{\, z : (y\ z) \in U^+(W)\}.$$

В частности, если $W = \{y\}$, то будем писать $U^-(y), U^+(y)$ и т.д.

3. <u>Правила преобразований графов</u>. <u>Определение I</u>. <u>Правилом подстановки</u> π будем называть четверку

$$\pi = (L_1 < \mu_1, \nu_1 >,\ L_2 < \mu_2, \nu_2 >,\ \varphi,\ \psi),$$

где I) $L_i < \mu_i, \nu_i > = (V_i, U_i)$ – некоторые графы, причем $(\mu_i, \nu_i) \notin U_i$, $U_i^-(\mu_i) = U_i^+(\nu_i) = \phi$, $i = 1, 2$;

2) φ – однозначное отображение $V_1^+(\mu_1)$ на $V_2^+(\mu_2)$;

3) ψ – однозначное отображение $V_1^-(\nu_1)$ на $V_2^-(\nu_2)$.

Графы L_i следует рассматривать как графы типа стягивания В. Вершины μ_i, ν_i – входы-выходы этих графов. Отображения φ и ψ устанавливают некоторое соответствие между вершинами в левой и правой части правила подстановки. Важно, что эти соответствия охватывают только "граничные" вершины графов L_i.

<u>Определение 2</u>. Будем говорить, что граф H <u>непосредственно выводим</u> из графа G с помощью правила подстановки π ($G \underset{\pi}{\Rightarrow} H$), если существуют разложения

$$F(G) = (G', G'', S(G', G''));$$
$$F(H) = (H', H'', S(H', H'')),$$

такие что

$$G' \equiv H'; \qquad (I)$$

$$B(S(G', G''), G'') < \mu', \nu' > \doteqdot L_1 < \mu_1, \nu_1 >; \qquad (2)$$

$$B(S(H', H''), H'') < \mu'', \nu'' > \doteqdot L_2 < \mu_2, \nu_2 >. \qquad (3)$$

В работе [7] отношение непосредственной выводимости определяется аналогичным образом, только вместо 2-изоморфизмов (2), (3) требуется изоморфизм соответствующих "чистых" (без специальных входов-выходов) графов.

<u>Определение 3</u>. Будем говорить, что правило подстановки π <u>применимо</u> к разложению $F(G)$ графа G, если выполняется условие (2) определения 2.

С практической точки зрения необходимо, чтобы левая часть правила подстановки L_1 была настолько "примитивной", чтобы установление изоморфизма (2) не было бы слишком сложной задачей.

Определение 3 является актуальным определением преобразования графов, точнее, отношения непосредственной выводимости. Однако в данном случае нас больше интересует вопрос, как, исходя из разложения $F(G)$ графа G, к которому применимо правило подстановки π, построить граф H, непосредственно выводимый из графа G.

Пусть $G''=(V'',U'')$, $L_2<\mu_2,\nu_2>=(V_2,U_2)$.

Опишем процедуру Π, которая осуществляет переход

$$\Pi:(F(G),\pi)\longrightarrow(H',H'',S_0), \tag{4}$$

где $$H''=(V_0,U_0),$$

по следующим правилам:

$$H':=G'; \tag{5}$$

$$V_0:=V_2\setminus\{\mu_2,\nu_2\}; \tag{6}$$

$$U_0:=U_2\cap V_0\times V_0; \tag{7}$$

$$S_1:=\{(x,\varphi(y)):(x,y)\in U^-(V'')\}; \tag{8}$$

$$S_2:=\{(\psi(y),z):(y,z)\in U^+(V'')\}; \tag{9}$$

$$S_0:=S_1\cup S_2. \tag{10}$$

Можно доказать, что:

- тройка (H',H'',S_0) является разложением некоторого вполне определенного графа H;
- справедливо $G\underset{\pi}{\Rightarrow}H$.

Процедура Π является универсальной в том смысле, что для разложения $F(G)$ любого графа G, к которому применимо некоторое правило подстановки π, она конструирует граф H, непосредственно выводимый из графа G.

Последовательность действий, осуществляемых процедурой Π при преобразовании (4), следующая:

I) "рядом"с исходным графом G по правилам (6), (7) строится граф H'';

2) просматриваются все дуги сечения $S(G',G'')$: один конец (в графе G') остается на месте, а другой "привязывается" с помощью отображения φ или ψ к некоторой вершине графа H'' (ориентация дуги не меняется),

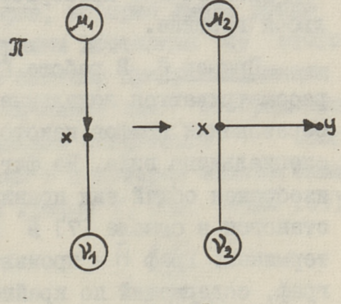
3) граф G'' удаляется из рассмотрения.

Построение (8)-(I0) сечения происходит так: для каждой дуги из сечения $S(G'G'')$ находят сначала соответствующую ей дугу в стягивании $B(S,G'')$, затем эту дугу "проектируют" с помощью изоморфизма (2) на левую часть правила подстановки L_1, после чего становится возможным применение оператора φ или ψ.

В дальнейшем, при схематическом изображении правил подстановок, мы будем вершины μ_i и ν_i отмечать особо. Всюду, где это не будет специально оговорено, будем считать, что $\varphi(x)=x$ и $\psi(x)=x$.

Пример 4. На фиг. 5 изображено правило подстановки π, применение которого к подграфу G_1 графа G (фиг. 2) теперь полностью определено; результатом преобразования является граф H_1 (фиг. 3).

Применимость правила подстановки к некоторому разложению $F(G)$ сильно зависит от структуры сечения $S(G',G'')$, т.е. контекста подвергаемого преобразованию подграфа G''. Так, например, правило подстановки π (фиг. 5) не применимо ни к одному другому подграфу графа G (фиг. 2), кроме уже указанного подграфа G_1.



Фиг. 5.

Суть преобразования, рассмотренного в примере 4, заключается в создании одной новой вершины y и дуги (x,y), где x – заданная вершина преобразуемого графа G. Остальная часть графа G, в том числе и окружение заданной вершины, особой роли не играет.

Предположим, что левая и правая часть правила подстановки состоит не только из графов $L_i=(V_i,U_i)$, но еще и из некоторого множества дуг U_i^{var}, причем $U_i^{var}\subset V_i\times V_i$, $U_i^{var}\cap U_i=\phi$, и что замена в правиле подстановки π графов L_i на графы $L_i^{max}=(V_i,U_i^{max})$, $U_i^{max}=U_i\cup U_i^{var}$ оставляет в силе определение I.

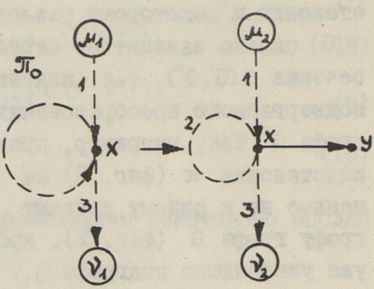
Будем считать, что правило подстановки π применимо в $F(G)= = (G',G'',S)$, если существует граф $L=(V_1,U)$, находящийся "между" графами L_1 и L_1^{max} (т.е. $U_1\subset U\subset U_1^{max}$) такой, что L 2-изоморфен графу $B(S,G'')$.

Такой подход позволяет нам описать целый класс семантически эквивалентных преобразований с помощью одного правила подстановки. Согласно вышеуказанному, дуги из U_1^{var} могут, но не должны присутствовать в G''. Следует отметить, что U_1^{var} может состоять из любых, а не только из "граничных" дуг.

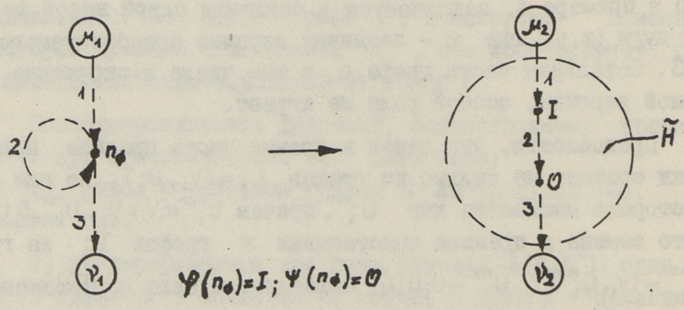
В правило подстановки необходимо еще добавить однозначное отображение $\tau : U_1^{var} \to U_2^{var}$, которое устанавливает соответствие между переменными частями в левой и правой части правила подстановки. На фигурах дуги из U_i мы будем обозначать сплошной, а из U_i^{var} – пунктирной линией. Соответствие τ укажем нумерацией дуг.

<u>Пример 5</u>. На фиг. 6 изображено правило подстановки π_0, аналогичное преобразованию π (фиг. 5), применимое к любой вершине.

<u>Пример 6</u>. В работе [7] рассматриваются локальные преобразования графов некоторого специального вида. На фиг. 7 изображен общий вид правил подстановок в смысле [7] в наших терминах. Граф $\tilde{H}$ – произвольный граф, содержащий по крайней мере две вершины: I и O.



Фиг. 6.



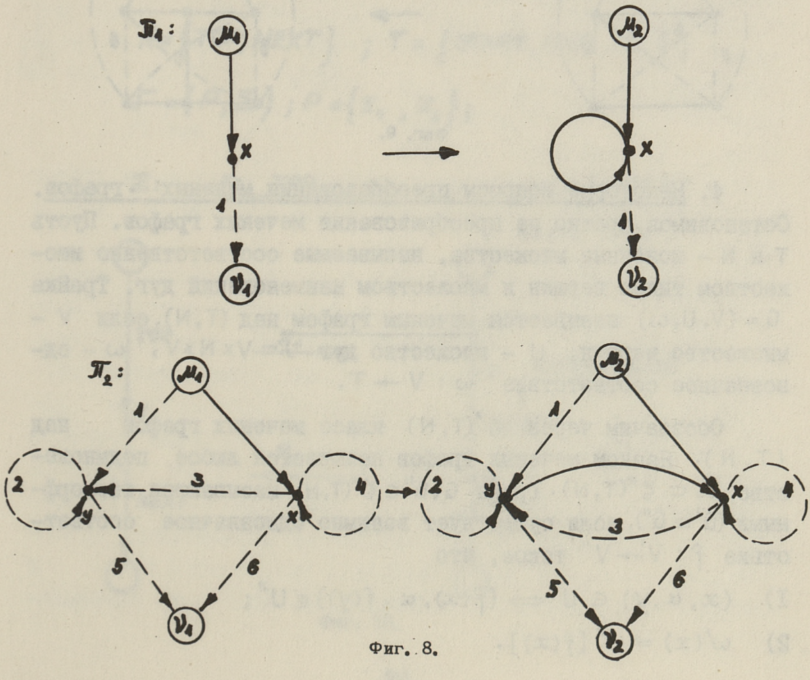
Фиг. 7.

<u>Грамматикой графов</u> Γ называется пара $\Gamma = (P, \Sigma)$, где P – конечное множество правил подстановки; Σ – некоторый "начальный"граф. Будем писать $G \underset{\Gamma}{\Rightarrow} H$, если существует $\pi \in P$, что $G \underset{\pi}{\Rightarrow} H$. Рефлексивно-транзитивное замыкание отношения $\underset{\Gamma}{\Rightarrow}$ будем обозначать $\overset{*}{\underset{\Gamma}{\Rightarrow}}$.

Г-языком графов называется класс графов
$\mathfrak{L}(\Gamma) = \{ G : \Sigma \overset{*}{\underset{\Gamma}{\Rightarrow}} G \}$.

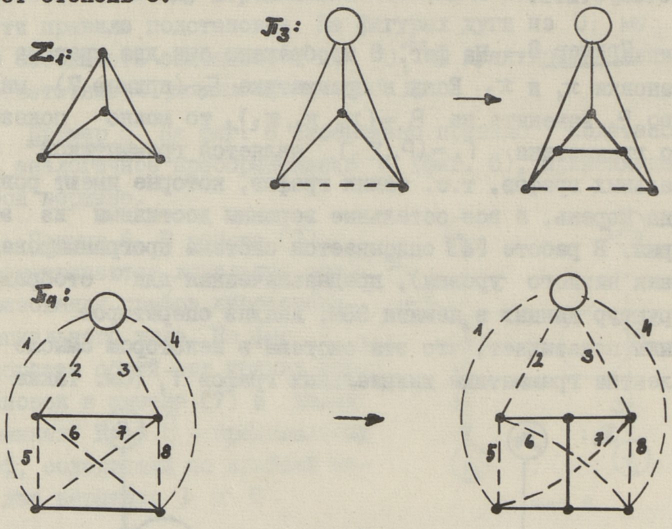
Пример 7. Пусть $P_0 = \{\pi_0\}$, где π_0 – правило подстановки, изображенное на фиг. 6,а $\Sigma_0 = \{\{x_0\}, \phi\}$. Можно показать, что грамматика $\Gamma_0 = (P_0, \Sigma_0)$ является грамматикой деревьев, т.е. любое дерево $D \in \mathfrak{L}(\Gamma_0)$. Петлю (x, x) в левой части правила можно опустить.

Пример 8. На фиг. 8 изображены еще два правила подстановки π_1 и π_2. Если в грамматике Γ_0 (пример 7) множество P_0 заменить на $P_1 = \{\pi_0, \pi_1, \pi_2\}$, то можно показать, что грамматика $\Gamma_1 = (P_1, \Sigma_0)$ является грамматикой инициальных графов, т.е. таких графов, которые имеют ровно один корень, а все остальные вершины достижимы из этого корня. В работе [4] описывается система программирования (язык низкого уровня), предназначенная для отображения структур данных в памяти ЭВМ. Анализ операторов этого языка показывает, что эта система в некотором смысле эквивалентна грамматике инициальных графов Γ_1 (см. также [5]).



Фиг. 8.

Пример 9. Разработанная здесь методика применима и для преобразования неориентированных графов. Основное различие заключается в том, что в правилах подстановки вместо двух вспомогательных вершин достаточно иметь лишь одну "внешнюю" вершину. На фиг. 9 изображена грамматика $\Gamma_2 = (\{\pi_3, \pi_4\}, \Sigma_1)$, которая является грамматикой некоторых кубических графов, т.е. графов, в которых каждая вершина имеет степень 3.



Фиг. 9.

4. Некоторые вопросы преобразования меченых графов. Остановимся кратко на преобразовании меченых графов. Пусть T и N – конечные множества, называемые соответственно множеством типов вершин и множеством наименований дуг. Тройка $G = (V, U, \omega)$ называется меченым графом над (T, N), если V – множество вершин, U – множество дуг $U \subset V \times N \times V$, ω – однозначное соответствие $\omega : V \to T$.

Обозначим через $\zeta^*(T, N)$ класс меченых графов над (T, N). Языком меченых графов называется любое подмножество $\mathcal{L} \subset \zeta^*(T, N)$. Графы $G', G'' \in \zeta^*(T, N)$ называются изоморфными $(G' \equiv G'')$, если существует взаимно однозначное соответствие $f : V' \to V''$ такое, что

1) $(x, \alpha, y) \in U' \Leftrightarrow (f(x), \alpha, f(y)) \in U''$;

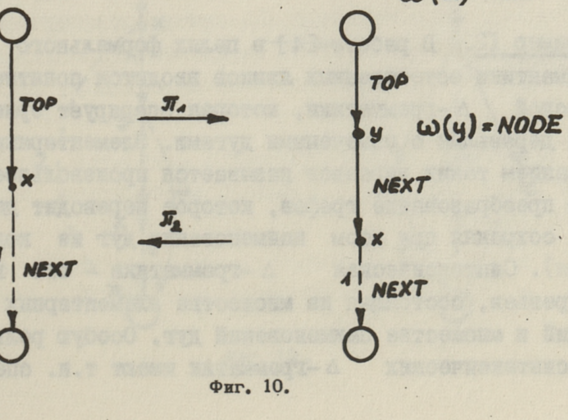
2) $\omega'(x) = \omega''[f(x)]$.

С помощью введенных понятий не представляет особого труда использовать разработанную нами методику для преобразования меченых графов. В правилах подстановки наименования дуг и типы вершин можно рассматривать как свободные параметры (не влияющие на применимость правила), или как фиксированные параметры (правило применимо, если заменяемый подграф и левая часть правила подстановки — изоморфные меченые графы). В последнем направлении можно сделать еще один шаг: в правилах подстановки дуги (вершины) имеют не одно наименование (один тип), а целое множество допустимых наименований (типов).

Пример IO. Работа [6] посвящена методике изучения структур данных, причем основное внимание уделяется рассмотрению структур данных на т.н. логическом уровне. На логическом уровне описания конкретной структуре данных соответствует меченый граф, называемый в [6] V-графом. Дуги V-графа интерпретируются как пути доступа к данным, находящимся в вершинах этого графа. Класс структур данных — это грамматика графов (в смысле настоящей статьи). В данной

$$N = \{TOP, NEXT\}\ ;\ T = \{START, NODE, NIL\}\ ;$$

$$\Gamma = (P, \Sigma)\ ;\ P = \{\pi_1, \pi_2\}\ ;$$

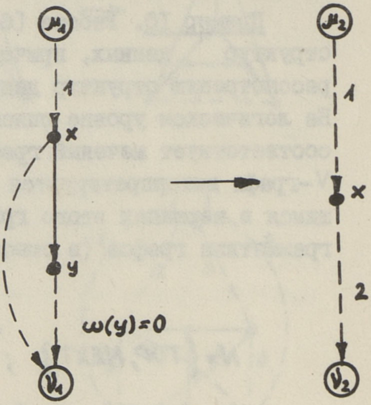


Фиг. 10.

работе по сравнению с [6], в которой не ставилась целью строгая формализация понятия преобразования, правило подстановки и процедура применения этих правил определены конструктивным образом.

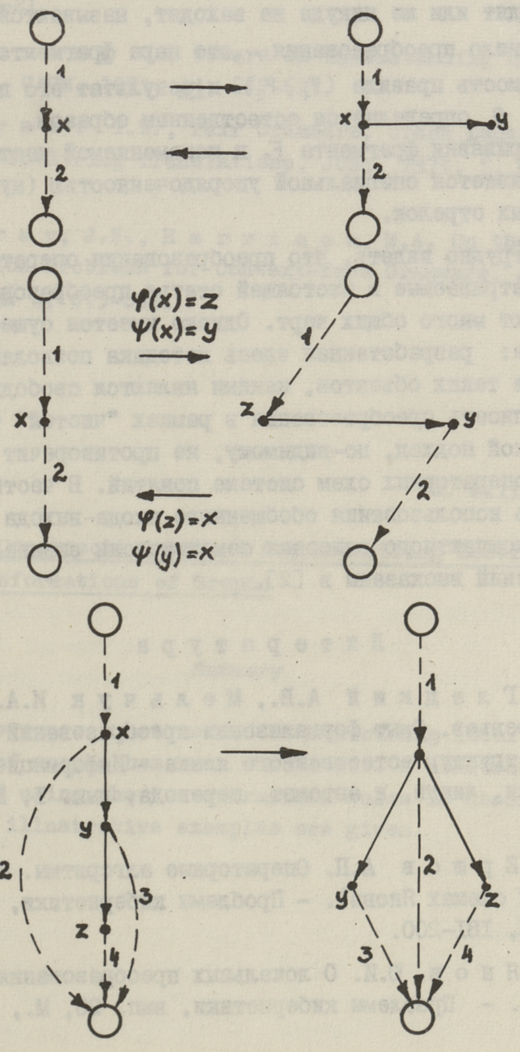
В работе [6] в качестве примера приведен ряд грамматик, порождающих типичные классы структур данных. Мы воспроизводим в терминах настоящей статьи одну из них. На фиг. I0 изображена грамматика стеков.

Пример II. Рассмотрим одну задачу, которая возникает при построении трансляторов с языков программирования. Как известно, результатом синтаксического анализа входного текста программы может быть дерево вывода D, которое, грубо говоря, является графом над $(\{0,1\}, \phi)$. Говорят, что вершины типа I имеют семантику, а вершины типа 0 таковой не имеют. В простейшем случае задача заключается в удалении из дерева D всех вершин типа 0, т.е. в преобразовании дерева вывода D в дерево т.н. H-вывода D_H [8]. На фиг. II изображено правило подстановки π, всевозможные применения которого к любому дереву вывода D преобразуют это дерево в дерево H-вывода D_H.



Фиг. 11.

Пример I2. В работе [4] в целях формального изучения семантики естественных языков вводится понятие синтаксической Δ-грамматики, которая оперирует ориентированными деревьями с помеченными дугами. Элементарным преобразованием таких деревьев называется произвольное локальное преобразование графов, которое переводит дерево в дерево, сохраняя при этом наименования дуг из контекста (сечения). Синтаксическая Δ-грамматика - это грамматика деревьев, состоящая из множества элементарных преобразований и множества наименований дуг. Особую роль в теории синтаксических Δ-грамматик имеют т.н. специальные преобразования (правила подстановки, реализующие эти преобразования, приведены на фиг. I2). Оказывается [I], что любое элементарное преобразование сводимо к специальным преобразованиям: если π – элементарное преобразование, М – множество специальных преобразований и $D_1 \underset{\pi}{\Rightarrow} D_2$, то $D_1 \overset{*}{\underset{M}{\Rightarrow}} D_2$.



Фиг. 12.

В заключение нам хотелось бы коротко остановиться на подходе к преобразованию графов, принятом в теории операторных схем [2, 3]. Основное понятие при преобразовании операторных схем – фрагмент. Фрагмент F – это часть схемы (графа), в которой каждая вершина содержит все стрелки(дуги), инцидентные этой вершине. Стрелки, которые ниоткуда не исходят или же никуда не заходят, называются свободными. Правило преобразования – это пара фрагментов (F_1, F_2). Применимость правила (F_1, F_2) и результат его применения к схеме G определяется естественным образом. Корректность привязки фрагмента F_2 к неизменяемой части схемы G обеспечивается специальной упорядоченностью (нумерацией) свободных стрелок.

Нетрудно видеть, что преобразования операторных схем и рассматриваемые в настоящей статье преобразования графов имеют много общих черт. Однако имеется существенное различие: разработанная здесь методика позволяет обойтись без таких объектов, какими являются свободные стрелки, и описать преобразования в рамках "чистой" теории графов. Такой подход, по-видимому, не противоречит принятой в теории операторных схем системе понятий. В частности, возможность использования обобщенного входа-выхода фрагмента и идея компактного описания семантически эквивалентных преобразований высказаны в [2].

Литература

I. Гладкий А.В., Мельчук И.А. Грамматика деревьев. Опыт формализации преобразований синтаксических структур естественного языка – Информационные вопр. семиотики, лингв. и автомат. перевода, вып. I, М., 1971, 16-41.

2. Ершов А.П. Операторные алгоритмы. Ш. (Об операторных схемах Янова). – Проблемы кибернетики, вып. 20, М., 1968, 181-200.

3. Янов Ю.И. О локальных преобразованиях схем алгоритмов. – Проблемы кибернетики, вып. 20, М., 1968, 201-216.

4. А л л и к К.К.-Э. М и к л и Т.И. Отображение структур данных в памяти ЭВМ. - "Тр. Таллинск. политехн. ин-та", 1974, № 366, 23-30.

5. Т о м б а к М.О., М и к л и Т.И., А л л и к К.К.-Э. Об отношении субординации. - Труды ВЦ ТГУ, 1974, вып. 30, 8-21.

6. E a r l e y J., Toward an Understanding of Data Structures, CACM, 1971, 14, 617-627.

7. P r a t t T.W., Pair Grammars, Graph Languages and String-to-String Translations, JCSS, 1971, 5, N 6, 560-595.

8. G r a y, J.N., H a r r i s o n, M.A. On the Covering and Reduction Problems for Context-free Grammars, JACM, 1972, 19, nN 4, 675-698.

K. Allik

The Constructive Method for Describing Local Transformations of Graphs

Summary

In this paper a new method of describing local transformations of graphs has been worked out. A constructive procedure of using the transformation rules is described. A number of illustrative examples are given.

4. Б а й в е р К.К.-Э., Ж и к л я Т.Н., Отображение структур данных в памяти ЭВМ. – Тр. Таллинск. политехн. ин-та, 1974, № 369, 28-30.

5. Т о м б а к М.О., ... К.К.-Э. Об операциях субординации. – Труды ВЦ ТГУ, 1974, вып. 30, 8-21.

6. E a r l e y J., Toward an Understanding of Data Structures. CACM, 1971, 14, 617-627.

7. P r a t t T.W., Pair Grammars, Graph Languages and String-to-Graph Translations. JCSS, 1971, 5, N 6, 560-595.

8. G r a y J.N., H a r r i s o n M.A., On the Covering and Reduction Problems for Context-Free Grammars. JACM, 1972, 19, N 4, 675-698.

E. Kiik

The Constructive Method for Describing of Transformations of Graphs

Summary

In this paper a new method of describing graph transformations of graphs has been worked out. A constructive procedure of using the transformation rules is described. A number of illustrative examples are given.

TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED
ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА

№ 411 1976

УДК 51:801

А.О. Вооглайд

СЕМАНТИЧЕСКОЕ РАВЕНСТВО РАСПОЗНАВАТЕЛЕЙ, РАБОТАЮЩИХ НА ГРАММАТИКЕ LR(k) И ГРАММАТИКЕ ПРЕДШЕСТВОВАНИЯ С (I/I) ОГРАНИЧЕННЫМ КАНОНИЧЕСКИМ КОНТЕКСТОМ

Введение

Данная статья является логическим продолжением статьи [10], где рассматриваются некоторые возможности подхода, данного в [3], где J.N. Grey и M.A. Harrison разбивают шаг анализа на две фазы - детектирование и редуцирование. Точнее, исследуются все возможности для редуцирования в грамматиках предшествования при помощи односимвольного контекста слева и справа от основы. Используя один результат из [I], показывается, что односимвольный контекст слева от основы для редуцирования позволяет анализировать все детерминированные языки.

В настоящей статье сравниваются распознаватель, работающий на грамматике LR(k), и распознаватель, работающий методом предшествования, в которых для редуцирования правил с одинаковой правой частью используется (I/I) ограниченный канонический контекст (распознаватель (I/I) ОККРГП). Покажем, что для любого распознавателя, работающего на λ-свободной LR(k) грамматике, можно найти распознаватель (I/I) ОККРГП такой, что для любого данного слова ими генерируются идентичные синтаксические деревья [2]. Назовем эти распознаватели семантически равными. Под деревом синтаксиса понимается дерево анализа, которое содержит только вершины, имеющие семантическое значение. Каждая вершина дерева помечена вектором значений семантических переменных.

Кроме того, получим используемые на практике преобразование грамматики SLR (I) в (I/I) ОККРГП, тест грамматики SLR (I) и алгоритм нахождения контекста LR (I).

I. Основные понятия

Обозначим через $\mathcal{A}^*$ множество всех слов в алфавите $\mathcal{A}$ и $\mathcal{A}^* = \mathcal{A}^* \setminus \{\lambda\}$, где λ - пустое слово. Если $x \in \mathcal{A}$, то $|x|$ - длина слова x. Контекстно-свободная грамматика (КГС) - это упорядоченная четверка $G = (\mathcal{V}_N, \mathcal{V}_T, P, S)$, где $\mathcal{V}_N$ - конечный алфавит нетерминальных символов, $\mathcal{V}_T$ - конечный алфавит терминальных символов такой, что $\mathcal{V}_T \cap \mathcal{V}_N = \phi$, P - множество правил подстановки вида $A \to x$, где $A \in \mathcal{V}_N$, $x \in (\mathcal{V}_T \cup \mathcal{V}_N)^*$ и S - начальный символ. Обозначим $\mathcal{V} = \mathcal{V}_N \cup \mathcal{V}_T$. Пусть $x, y \in \mathcal{V}^*$. Слово y непосредственно канонически выводимо из слова $x (x \Rightarrow y)$, если $x = uAv$, $y = uzv$ и $A \to z \in P (u \in \mathcal{V}^*, v \in \mathcal{V}_T^*)$. Слово y канонически выводимо из слова $x (x \overset{*}{\Rightarrow} y)$, если существует последовательность слов $z_0, z_1, \ldots, z_k (z_i \in \mathcal{V}^*, 0 \leqslant i \leqslant k)$ такая, что $x = x_0, y = z_k$ и $z_{i-1} \Rightarrow z_i$ $(0 < i \leqslant k)$. Последовательность $z_0, z_1, \ldots, z_k$ называется каноническим выводом y из x. Слово $x \in \mathcal{V}^*$ называется канонической сентенциальной формой (КС-формой), если $S \overset{*}{\Rightarrow} x$.

Канонический вывод $S = z_0, z_1, \ldots, z_k = x$ слова x из S называется каноническим выводом x и последовательность $z_k, \ldots, z_0$ - каноническим анализом КС-формы x. Пусть $L(G) = \{x : x \in \mathcal{V}_T^*, S \overset{*}{\Rightarrow} x\}$ есть язык, определяемый грамматикой G.

Если $S \overset{*}{\Rightarrow}$ и $Av \Rightarrow uzv$, то выделенные слово z, принадлежащее КС-форме uzv, называется основой КС-формы uzv.

Для описания некоторого метода синтаксического анализа для класса КС-грамматики K достаточно дать алгоритм, который по любой грамматике $G \in K$ и любой КС-форме x находит в G элемент, непосредственно следующий за x в каноническом анализе КС-формы, т.е. алгоритм, реализующий шаг анализа.

Разобъем шаг анализа на два действия [3]:

I. Детектирование. Найти основу z КС-формы uzv.

2. Редуцирование. Найти правило $A \to z$ такое, что $S \overset{*}{\Rightarrow} uAv \Rightarrow uzv$ и заменит основу z на A.

Универсальным методом детектирования для всех КС-языков является метод предшествования [8], суть которого в следующем. Пусть $G=(V_N, V_T, P, S)$ КС-грамматика. Определим следующие бинарные отношения на множестве $V \times V$:

$$\lambda=\{(A,B): A \to By \in P,\ y \in V^*\},\quad \lambda_T=\lambda \cap (V \times V_T),$$

$$\rho=\{(A,B): B \to xA \in P,\ x \in V^*\},$$

$$\alpha=\{(A,B): C \to xABy \in P,\ x,y \in V^*\},$$

$$\lessdot=\alpha\lambda^+,$$

$$\doteq=\alpha,$$

$$\gtrdot=(\rho^+\alpha\lambda_T^*).$$

Отношения $\lessdot, \doteq, \gtrdot$ называются отношениями предшествования.

КС-грамматика G называется грамматикой предшествования, если

I. P не содержит правил с пустой правой частью (G λ-свободная).

2. Отношения $\lessdot, \doteq, \gtrdot$ попарно непересекающиеся.

Имеет место следующая теорема [I].

<u>Теорема </u>I. Если G - грамматика предшествования и

$S \overset{*}{\Rightarrow} X_1X_2\ldots X_kAT_1T_2\ldots T_l \Rightarrow X_1X_2\ldots X_kY_1Y_2\ldots Y_mT_1T_2\ldots T_l$

в G, то

I) $X_i \lessdot X_{i+1} \vee X_i \doteq X_{i+1}$ $(1 \leqslant i < k)$;

2) $X_k \lessdot Y_1$;

3) $Y_j \doteq Y_{j+1}$ $(1 \leqslant j < m)$;

4) $Y_m \gtrdot T_1$;

5) никакие другие отношения предшествования, кроме указанных в пунктах I-4, между этими символами не имеют места.

Кроме того, известно, что по любой λ-свободной КС-грамматике G можно построить грамматику предшествования G' такую, что $\mathcal{L}(G)=\mathcal{L}(G')$ [3,6].

Для редуцирования полезно использовать ограниченный канонический контекст (ОКК). Определим (m,k)-ОКК нетерминала A для натуральных m,k (обозначаем его $C_A^{m,k}$)

$$C_A^{m,k}=\{(x,y): \#^m S \#^k \overset{*}{\Rightarrow} uxAyv;\ u,x \in V^*;\ |x|=m, |y|=k\}.$$

КСГ G называется (m,k)-ОКК редуцируемой грамматикой предшествования (ОККРГП), если G – грамматика предшествования и для любых A и B таких, что $A \to z, B \to z \in P$ имеет место

$$C_A^{m,k} \cap C_B^{m,k} = \phi .$$

Если G – (m,k) -ОККРГП и $uxzyv(|x|=m, |y|=k)$ КС-форма с основой $z(A \to z, B \to z \in P)$, то надо редуцировать при помощи $A \to z$, если $(x,y) \in C_A^{m,k}$ и при помощи $B \to z$, если $(x,y) \in C_B^{m,k}$.

Методы для нахождения $C_A^{m,k}$ известны [5], но довольно трудоемкие и соответствующие анализаторы весьма неэффективны. Если же $m, k \leq 1$, то эффективность соответствующего анализатора сравнима с эффективностью анализатора простого предшествования. Обозначим $C_A^{m/k} = C_A^{m,0} \times C_A^{0,k}$ и используя один из результатов [10], можем сформулировать следующую теорему.

<u>Теорема 2</u>. Распознаватели для классов грамматик (I/I) -ОККРГП позволяют анализировать все детерминированные языки.

3. <u>Дерево синтаксиса и семантическое равенство распознавателей</u>

Как правило, грамматика, которая описывает какой-либо детерминированный язык, не является (I/I)-ОККРГП. Следовательно, необходимо преобразовать данную грамматику в нужный вид. Но потребитель, который работает с распознавателем в системе построения трансляторов [II], должен быть уверен, что после преобразования грамматики синтаксическое дерево остается прежним.

Независимость синтаксического дерева от преобразований грамматики характеризует понятие семантического равенства распознавателей. Исходим из понятия полного покрытия [3, с.681].

Канонический вывод $z_0, z_1, \ldots, z_k$ можно представить в виде $p_1, p_2, \ldots, p_k$, где $p_i \in P$ – правило, при помощи которого z_i непосредственно выводимо из z_{i-1}. Пусть правило P_i имеет вид $A_i \to x_i$. Вводим сокращенное обозначение $(A_i \to x)_{i=1}^{k}$ для вывода $p_1, \ldots, p_k$. Пусть $H \subseteq P$ – множество таких правил, которые имеют семантическое значение. Пусть $D = (A_i \to x_i)_{i=1}^{k}$ –

канонический вывод грамматики $G=(V_N, V_T, P, S)$. H -разреженным выводом называется кортеж $D_H=(A_i \twoheadrightarrow x_i : A_i \to x_i \in H)_{i=1}^{K}$ [3,с.681]. Пусть T_D - дерево вывода, которое соответствует выводу D [3,с.682]. Пусть $Q \subseteq V_T$ - такое подмножество V_T, элементы которого имеют семантическое значение, т.е. V_T/Q содержит такие терминальные символы, содержание которых выражено в структуре дерева вывода (скобки, разделители) или содержание которых совпадает с семантикой соответствующих правил (символы операции, ключевые слова).

H, Q **-разреженным** деревом вывода D (обозначаем $T_D^{H,Q}$) называем дерево, которое получается из дерева вывода T_D следующим образом:

(I) вершинами дерева $T_D^{H,Q}$ являются все вершины дерева T_D, метки которых принадлежат множеству $H \cup Q$.

(2) в дереве $T_D^{H,Q}$ соединены x и y дугой тогда и только тогда, когда в дереве T_D существует путь $z_0, \ldots, z_l$ от вершины x до вершины y такой, что $z_i \in P/H \cdot (1 \leq i \leq l-1, z_0 = x, z_l = y)$.

Во время работы распознавателя составляется сразу дерево $T_D^{H,Q}$, которое аналогично [2, с.886] называем отмеченным синтаксическим деревом для трансляции или просто синтаксическим деревом [2, с. 722].

Пусть имеются классы КС-грамматик K_1 и K_2 такие, что для любого $G_1 \in K_1$ существует $G_2 \in K_2$, так что $\mathcal{L}(G_1) = \mathcal{L}(G_2)$. Пусть R_1 и R_2 распознаватели соответствующих классов. Если дана грамматика $G_1 \in K_1$ в виде $G_1=(V_N, V_T, P, S)$ и $H \subseteq P$, $Q \subseteq V_T$ и какое-либо слово $x \in \mathcal{L}(G_1)$ данного языка, то обозначим $R_1^{G_1,H,Q}(x)$ дерево синтаксиса слова x, построенное распознавателем R_1 соответственно данной грамматике G_1 и множествам H и Q.

Мы говорим, что распознаватели R_1 и R_2 семантически равны, если для любой грамматики $G_1 \in K_1$ и H и Q найдется такая КС-грамматика $G_2 \in K_2$ и H', что при любом $x \in \mathcal{L}(G_1)$ имеет место

$$R_1^{G_1,H,Q}(x) = R_2^{G_2,H',Q}(x).$$

Это условие гораздо строже, чем условие полного покрытия, и обеспечит независимость семантических конструкций от выбора класса грамматик анализа в пределах фиксированного множества классов грамматик.

Пусть классом грамматик K_1 будет класс $LR(k)$ грамматик и K_2 – (I/I) ОККРГП.

В статье [3] приведена следующая теорема:

<u>Теорема 3</u>. Любую λ–свободную $LR(k)$ грамматику G можно преобразовать в грамматику предшествования G' так, что

а) $\mathcal{L}(G) = \mathcal{L}(G')$;

б) G' есть грамматика $LR(k)$.

Используя результаты статьи [7], можно сформулировать следующие теоремы:

<u>Теорема 4</u>. Любую $LR(k)$ грамматику G можно преобразовать в $LR(1)$ грамматику G' так, что $\mathcal{L}(G) = \mathcal{L}(G')$.

<u>Теорема 5</u>. Любую $LR(k)$ грамматику G можно преобразовать в $SLR(1)$ грамматику G' так, что $\mathcal{L}(G) = \mathcal{L}(G')$.

Для доказательства равенства распознавателей для классов грамматик $LR(k)$ и (I/I) ОККРГП необходимо:

I) средство для описания левого контекста нетерминала;

2) преобразование $SLR(1)$ грамматик G в G' так, что
   а) $\mathcal{L}(G) = \mathcal{L}(G')$,
   б) G' является (I/I) ОККРГП;

3) чтобы все преобразования в теоремах 3–5 и в пунктах 2 сохранили начальное синтаксическое дерево.

Для выполнения этих пунктов используем понятия сложности конфликтов анализа, графа контекста $LR(1)$, даем преобразование $LR(1) \Rightarrow$ (I/I) ОКК и докажем независимость первоначального синтаксического дерева от преобразований грамматики.

4. <u>Сложность конфликтов редуцирования и анализа</u>

Как правило, грамматика, которая описывает какой-либо детерминированный язык, не является (I/I) ОККРГП. Следовательно, необходимо преобразовать данную грамматику в нужный вид. Во время преобразования грамматики меняется "трудность" грамматического разбора. Чтобы охарактеризовать эту "трудность", используем понятие конфликта анализа (КА). Под

КА понимается или конфликт редуцирования (КР), или конфликт детектирования (КД).

Под КР понимается ситуация, где для редуцирования нужен контекст, т.е. множество правил грамматики содержит по меньшей мере два правила с одинаковой правой частью $A \to x$ и $B \to x (x \in V^+)$. Обозначим данный КР через (A, B, x).

Под КД понимается ситуация, где для детектирования необходим контекст, т.е. множество правил грамматики содержит по меньшей мере два правила: $A \to x$, $B \to y (x, y \in V^+)$ и $y = \alpha x \beta, (\alpha \beta \in V^+)$. Обозначим КД через (A, B, x, y).

Сложность КА оценивается по длине контекста. Пусть D – множество значений всевозможных сложностей в классе грамматик $LR(k)$.

$D \subset (N \cup \{\#\}) \times N$, где N – множество натуральных чисел.

Определим во множестве D линеарную последовательность. Считаем априори, что при каждом $n \in N \# > n$ и $\# + n = \#$

$$d_1, d_2 \in D, \quad d_1 = (d_1^1, d_1^2), \; d_2 = (d_2^1, d_2^2);$$

$$d_1 = d_2 \leftrightarrow (d_1^1 = d_2^1) \& (d_1^2 = d_2^2);$$

$$d_1 < d_2 \leftrightarrow (d_1^2 < d_2^2) \vee [(d_1^2 = d_2^2) \& (d_1^1 < d_2^2)];$$

$$d_1 > d_2 \leftrightarrow \neg[(d_1 = d_2) \& (d_1 < d_2)].$$

Для ясности множество значений сложностей можно показать следующим образом

$$\begin{array}{llll} (0,0) & (1,0) & \ldots & (\#, 0), \\ (0,1) & (1,1) & \ldots & (\#, 0), \\ \ldots & \ldots & \ldots & \ldots, \\ (0,k) & (1,k) & \ldots & (\#, k). \end{array}$$

Сложность анализа по данной грамматике определяется максимальной сложностью КА.

В целях компактного изложения используем еще один вид сокращенной записи для обозначения значения сложности. Обозначим $(d_1/d_2), ((d_1, d_2) \in D)$ сложность конфликта, значение которого $(d_1, 0)$ или $(0, d_1)$. Пусть D_1 – множество таких сложностей. Кроме того, определим, что $(0,d) < (d/d) < (d,d)$. Продолжая деление грамматик предшествования во множестве классов грамматик предшествования K [10]

$$K = \{(d)ККРГП : d \in D \cup D_1\}$$

и используя ту же технику, можем сформулировать следующую теорему (ККРГП - каноническим контекстом редуцируемая грамматика предшествования).

Теорема 6. $n \in N$, где n - любое фиксированное число.

При каждом $i < n$ и каждом $j \in N$ существуют следующие отношения между классами грамматик:

I $(0,i)$ ОККРГП $\cap$ $(i,0)$ОККРГП $=$ $(0,0)$ ОККРГП ;

II $(0,i)$ ОККРГП $\subset$ (j/i) ОККРГП ;

III $(j,0)$ ОККРГП $\subset$ (j/i_1)ОККРГП ; $\quad 1 \leq i_1 \leq n$;

IV (j/i) ОККРГП $\subset$ (j,i) ОККРГП ;

V (j,i) ОККРГП $\subset$ $(j+1,i)$ОККРГП ;

VI $(\#,i)$ ККРГП $\subset$ $(\#,i+1)$ККРГП .

5. Граф контекста LR(1).

Для нахождения контекста какого-либо нетерминала используется понятие графа контекста LR (I).

Предполагается, что читатель знаком с основными понятиями теории графов [9].

Конечным ориентированным нагруженным графом называется упорядоченная четверка (V,D,E,f), где

I) (V,D) - конечный ориентированный граф; D - последовательность $(p_1,q_1), \ldots, (p_r, q_r)$;

2) E - непустое конечное множество (множество пометок);

3) f - отображение множества I, ..., r в $E(f(i))$ - пометка при дуге (p_i, q_i).

Пусть дана приведенная λ-свободная с граничными маркерами КСГ $G = (V_N, V_T, P, S_0)$, т.е.

I) $S_0 \to \# S \# \in P$,

2) $(\forall A \neq S_0)[(\exists x,y \in V^*)(S_0 \overset{*}{\Rightarrow} xAy)]$,

3) $(\forall A)[(\exists x \in V_T^*)(S_0 \overset{*}{\Rightarrow} x)]$.

Даем индуктированное понятие конечного ориентированного нагруженного графа контекста LR(1).

Пусть $A \in V_N$ нетерминал (нт.) такой, что $A \neq S_0$, μ_A - множество всех нетерминалов, зависящих от A, являющееся множеством вершин конструируемого графа.

Пусть множество меток E - конечное подмножество $V^* \times V$.

Рассмотрим метку при дуге как состоящую из двух частей. Различим левую и правую метку дуги, смотря со стороны стрелки.

Мы говорим:

I. Граф из одной вершины, отмеченной A, есть контекстный граф нт. A.

2. Если $X \neq S_0$ есть висячая вершина контекстного графа нт. A, то можно образовать новый контекстный граф нт. A следующим образом:

а) определим подчиненными вершине X все вершины с меткой Y такие, что существует правило $Y \to \alpha$

$$\alpha = x_1 X x_2 X \ldots X x_{n-1} X x_n \, (x_i \in V^*),$$

где $x_i (i = 1, 2, \ldots, n)$ не содержит нт. X.

б) из вершины X в вершину Y направлено $n-1$ дуг;

в) левыми метками дуг являются префиксы α:

$$x_1, x_1 X x_2, \ldots, x_1 X x_2 X \ldots X x_{n-1};$$

г) если существует такой путь из вершины A к вершине X, что множество правых меток всех образуемых дуг есть пустое множество, то правыми метками новых дуг являются соответственно первые символы слов $x_1, x_2, \ldots, x_n$. Пусть дан контекстный граф ξ_A, содержащий вершину S_0. Рассмотрим произвольный путь в графе из вершины A к вершине S_0 и образуем конкатенацию левых меток, начиная с вершины S_0. Получим один из элементов левого контекста нт. A. Если учесть правую метку этого пути и найти все первые символы слов, выводимых из этой метки, получим элемент $LR(1)$ контекста [5]. Рассматривая всевозможные пути из вершины A к вершине S_0, получим $LR(1)$ контекст нт. A.

Левый контекст нт. A есть регулярный язык и его удобно изображать при помощи регулярной формулы.

Понятие регулярной формулы в алфавите V определяется индуктивно: каждая буква из V является регулярной формулой; если W_1 и W_2 - регулярные формулы, то таковыми же являются

$$(W_1) \vee (W_2), (W_1)(W_2), (W_1)^*.$$

Иллюстрируем на примере идеи преобразования левого контекста. Пусть КСГ $G=(\{A,S\},\{0,1,a,b\},P,S_0)$.

$$
\begin{aligned}
P: S_0 &\to \# S \# \\
S &\to 01S \\
S &\to bS \\
S &\to aA \\
A &\to 01A \\
A &\to bA \\
A &\to b \\
S &\to b \\
A &\to 01 \\
S &\to 01
\end{aligned}
$$

Пусть даны языки L_1 и L_2

$$L_1=a(01)^* b^*,\ L_2=(01)^* b.$$

Данная грамматика генерирует язык

$$\mathcal{L}(G)=L_1 \vee L_2 \vee L_2 L_1.$$

КСГ G содержит конфликты редуцирования (A,S,b) и $(A,S,01)$. Если слово содердит терминал a, то надо до тех пор редуцировать слово и из него получаемые фразы при помощи нт. A, пока не появится фраза, которая не содержит терминала a. Далее надо редуцировать при помощи нт. S. Левый контекст, который определяет редуцирование (терминал a), может находиться на любом расстоянии от места редуцирования. Если требуется преобразовать КСГ G в (I/I) редуцируемую грамматику, то необходимая информация о терминале a (или ее отсутствие) переносится к месту редуцирования. Это делается следующим образом. Находится общая часть левых контекстов нт. A и нт. B, которая переименовывается по-разному. Получаем КСГ $G_1=(\{A,A_1,S,S_1,A_2,S_2,S_0\},\{a,b,0,1\},P_1,S_0)$.

$$
\begin{aligned}
P: S_0 &\to \# S \# & A_1 &\to b \\
S &\to S_1 S & S_1 &\to 01 \\
S &\to S_2 S & S_2 &\to b \\
A &\to A_1 A & A_2 &\to 01 \\
A &\to A_2 A & S &\to aA \\
A &\to b & &
\end{aligned}
$$

КСГ G_1 есть (I/I) редуцируемая грамматика.

При работе с левым контекстом полезно использовать понятие "суффиксальное пересечение". Пусть даны два слова - $x,y\in \mathcal{V}^*$. Определим максимальный суффикс этих двух слов $Sf(x,y)=z,\ x=uz,\ y=vz,\ Sf(uv)=\lambda,\ vu\neq\lambda,\ u,v\in\mathcal{V}^*$.

Пусть имеем две регулярные формулы U и V. Обозначим язык, определяемый формулой U, соответственно $\overline{U}$. Определим понятие суффиксального пересечения множеств $\overline{U},\overline{V}$ (обозначим $\overline{U}\text{-}\!A\,\overline{V}$):

$$\overline{U}\text{-}\!A\,\overline{V}=\{Sf(x,y): x\in\overline{U}\ \&\ y\in\overline{V}\}.$$

Заметим, что если $x \in \bar{U} \cap \bar{V}$, то из этого следует, что $x \notin \bar{U} \text{-}A\, \bar{V}$. Пусть имеется $LR(1)$ контекстный граф ζ_{A_i}:нт. A. Пусть левый контекст нт. A дан регулярной формулой W. Разделим формулу W на элементы $W_1, W_2, \dots, W_n$ так, что ни один элемент не содержит знака "ИЛИ" $W = \bigvee_{i=1}^{n} W_i$. Между любыми формулами W_i $(i=1,2,\dots,n)$ и множеством путей (в $LR(1)$ контекстом в графе), образующих множество левых контекстов W_i, имеется взаимнооднозначное соответствие [4]. Следовательно, взаимнооднозначное соответствие имеется и между подмножеством правил (при помощи которых генерируется соответствующая часть левого контекста) и регулярной формулой W_i.

6. Преобразование SLR (I) РГП в (I/I) ОККРГП

Далее предлагаем преобразование КСГ G в G' так, чтобы сложность фиксированных конфликтов редуцирования имела значение (I/I), если в первоначальной грамматике она имела значение $SLR(1)$. Пусть имеется грамматика предшествования $G=(V_N, V_T, P, S)$ и в ней конфликты редуцирования $A_1 \to \alpha$, $A_2 \to \alpha, \dots, A_n \to \alpha$, $\alpha \in V^+$ такие, что для любого i имеется j, $i \neq j$, $C^{1/1}_{A_i} \cap C^{1/1}_{A_j} \neq \phi$.

Для каждого A_i $i=1,2,\dots,n$ образуем $LR(1)$ контекстный граф ζ_{A_i}. По каждому ζ_{A_i} находим левый контекст нт. A_i. Найденную формулу левых контекстов W_i разделим на элементы так, чтобы они не содержали знака "ИЛИ":

$$W_i^1, \dots, W_i^{n_i}.$$

Далее предлагаем преобразование КСГ G в G' так, чтобы сложность рассматриваемых конфликтов редуцирования имела значение (I/I), если в первоначальной грамматике она имела значение (#, 0) или (0,I):

```
i: = 1(1)n-1
j: = i+1(1)n
IF C^{0,1}_{Ai} ∩ C^{0,1}_{Aj} ≠ φ THEN
BEGIN l: =1(1)n_i
      m: =1(1)n_j
IF W̄^l_i ∩ W̄^m_j ≠ φ THEN STOP A
IF W̄^l_i -A W̄^m_j ≠ φ THEN
```

```
BEGIN C_ij^lm = W̄_i^l ⊣ W̄_j^m
      COLLECT (C_ij^lm, i)
      COLLECT (C_ij^lm, j)
  END
  NEXT m
  NEXT l
END
NEXT j
NEXT i
i = 1(1)n
STRAT(i)
NEXT i
STOP B
```

COLLECT (C_{ij}^{lm}, i). Регулярная формула C_{ij}^{lm} и граф контекста LR(1) определяют подмножество правил в виде

$$\{B_к \to x_к B_{к+1} y_к : y_к, x_к \in V^*, B_к \in M_{A_i}\}$$

и для каждого правила место $x_к / B_{к+1}$ в правой стороне правила. Соберем выше определенные правила в множество P_i, если не имеется таких q' и p', что $\overline{W}_i^l \cap \overline{W}_{p'}^{q'} \neq \phi$ и $\overline{W}_{p'}^{q'} \dashv \overline{C}_{ij}^{lm} \neq \phi$, то для всех $\overline{W}_{p'}^{q'}$ и $\overline{W}_i^l$ образуем общее множество P_i.

Если имеются такие q' и p', что $\overline{W}_i^l \cap \overline{W}_{p'}^{q'} \neq \phi$ и $\overline{W}_{p'}^{q'} \dashv \overline{C}_{ij}^{lm} = \phi$ при каждом $W_{p'}^{q'}$, то включим в это множество все правила с фиксированными местами $x_к / B_{к+1}$ в их правой части, определенные регулярной формулой C_{ij}^{lm} и графом ζ_{A_p}. Если во время предыдущего прохождения алгоритма такое множество правил уже образовано, то собираем правила туда.

STRAT(i). При каждом элементе множества правил P_i преобразуем грамматику G следующим образом.

Пусть этим элементом будет $(*)$ $B_к \to x_к B_{к+1} y_к$. Если $x_к \neq \lambda$, то заменим правило $(*)$ в грамматике G двумя новыми правилами: $B_к \to D B_{к+1} y_к$, $D \to x_к$, где D – новый нт. $D \notin V_N$.

Если в границах того же множества P_i уже взято к использованию правило $D \to x_k$, то заменим правило (*) одним новым правилом $B_k \to DB_{k+1}y_k$. Если $x_k = \lambda$, то правило (*) не заменяется. Если множество правил P_i не образовано, то STRAT(i) является пустым оператором.

STOP A означает, что данное место в грамматике приводит к тому, что при помощи (#/1) контекста данная грамматика G не редуцируема. STOP B означает, что преобразуя новую грамматику в грамматику предшествования, получим грамматику, где рассматриваемый конфликт редуцирования имеет сложность (I/I). Покажем это.

Выбор множеств P_i и выполняемые левосторонние стратификации (ЛС) обеспечивают в новой грамматике G', полученной в результате преобразований, значение (I/I) сложности рассматриваемого КР. Но полученная грамматика G', как правило, не является грамматикой предшествования. Пусть $(\Delta) A \to u\mathfrak{X}\mathfrak{Y}v \in P$ есть произвольное правило, над которым совершена ЛС на месте $\mathfrak{X}/\mathfrak{Y}$. Сравним отношения предшествования, источниками которых являются правило (Δ) в грамматике G и правило $A \to B\mathfrak{Y}v$ в грамматике G':

$$R_1^G = \{(\mathfrak{X},\mathfrak{Y})\} \subset \doteq, \quad R_2^G = \{(\mathfrak{X},\mathfrak{Z}) : (\mathfrak{Y},\mathfrak{Z}) \in \lambda^+\} \subset \lessdot,$$

$$R_3^G = \{(\mathfrak{Z}_1,\mathfrak{Z}_2) : (\mathfrak{Z}_1,\mathfrak{X}) \in \rho^+ \& (\mathfrak{Y},\mathfrak{Z}_2) \in \lambda_T^*\} \subset \gtrdot,$$

$$R_1^{G'} = \{(B,\mathfrak{Y})\} \subset \doteq, \quad R_2^{G'} = \{(B,\mathfrak{Z}) : (\mathfrak{Y},\mathfrak{Z}) \in \lambda^+\} \subset \lessdot,$$

$$R_3^{G'} \cap R_3^G = \{(\mathfrak{X},\mathfrak{Z}) : (\mathfrak{Y},\mathfrak{Z}) \in \lambda_T^+\} \subset \gtrdot.$$

Из множеств видно, что КД в грамматике G' может появиться только в виде $\gtrdot \cap \doteq \neq \phi$ или $\gtrdot \cap \lessdot \neq \phi$. Такие КП называют конфликтами типа P_1. Используя данную в [6] первую часть преобразования предшествования, получаем из грамматики G' грамматику предшествования G''. Возможность использовать только алгоритм преобразования конфликтов типа P_1 появилась только потому, что не было конфликтов типа P_2 $(\doteq \cap \lessdot \neq \phi)$. Действительно, все преобразования произведены при помощи ЛС. Соответственно множествам R КД типа P_2 могут появиться только между некоторым новым нт. $B (B \to u\mathfrak{X})$ и каким-то символом $\mathfrak{Z}$. Предположим противное, что существует такой символ $\mathfrak{Z}$, что $(B,\mathfrak{Z}) \in \doteq$ и $(B,\mathfrak{Z}) \in \lessdot$. Но в таком случае имеет место $(\mathfrak{X},\mathfrak{Z}) \in \doteq$ или $(\mathfrak{X},\mathfrak{Z}) \in \lessdot$.

Следовательно, грамматика G не является грамматикой предшествования, т.е. получили противоречие.

Лемма I. Замена отношения $\mathfrak{X} \doteq \mathcal{Y}(\mathfrak{X}, \mathcal{Y} \in \mathcal{V})$ в грамматике предшествования на отношение $\mathfrak{X} \cdot> \mathcal{Y}$ возможна только при помощи алгоритма ликвидации конфликтов типа P_1.

Преобразуем полученную грамматику G' в грамматику предшествования G'' и посмотрим не появится ли новые КР, значение сложности которых больше (I/I) в течение преобразования предшествования.

I. Могли ли КР грамматики G увеличиться в грамматике G''? Сложность КР определяется контекстом. ЛС не меняет правого контекста, так как работаем с каноническим выводом. Возьмем произвольный нт. C так, что $A \in \mu_C$. Сравним контекст нт. C при помощи графа контекста до и после ЛС, получим лемму 2.

Лемма 2. В грамматике G', полученной при помощи ЛС из грамматики G, не увеличилась сложность ни одного общего для этих грамматик КР.

II. Могли ли появиться в грамматике G' новые КР, сложность которых превышает (0,I)?

Предположим, что в G' появился конфликт $(B, C, X_1 X_2 \ldots X_n)$ такой, что $C_C^{0,1} \cap C_B^{0,1} \neq \phi$, $b \in C_C^{0,1} \cap C_B^{0,1}$.

Пусть произведена ЛС следующая:

$$\begin{array}{lcl} A \to \mathfrak{X}_1 \mathfrak{X}_2 \ldots \mathfrak{X}_n \mathcal{Y}_1 \ldots \mathcal{Y}_q \in P & & A \to B\mathcal{Y}_1 \ldots \mathcal{Y}_q, \\ & \Longrightarrow & \\ (1 \le n,\ 1 \le q) & & B \to \mathfrak{X}_1 \ldots \mathfrak{X}_n,\ B \notin \mathcal{V}_n. \end{array}$$

Пусть $(H'(\mathcal{Y}) = \{\mathfrak{X}/\mathcal{Y} \overset{*}{\Rightarrow} \mathfrak{X}x,\ \mathfrak{X} \in \mathcal{V},\ x \in \mathcal{V}^*\})$.

Из ЛС следует, что $C_B^{0,1} = H^1(\mathcal{Y})$ и так как имел место КР, то во множестве P должно быть правило $C \to \mathfrak{X}_1 \mathfrak{X}_2 \ldots \mathfrak{X}_n \in P$.

Из определения контекста следует, что $S \overset{*}{\Rightarrow} xCby$, $x, y \in \mathcal{V}^*$.

Из теоремы I следует, что $(\mathfrak{X}_n, b) \in \cdot>$. Поскольку $A \to \mathfrak{X}_1 \ldots \mathfrak{X}_n \mathcal{Y}_1 \ldots \mathcal{Y}_q \in P$ и $b \in H^1(\mathcal{Y}_1)$, то есть две возможности

а) если $\mathcal{Y}_1 \in \mathcal{V}_T$, то $\mathcal{Y}_1 = b(\mathfrak{X}_n, b) \in \doteq$ и $(\mathfrak{X}_n, b) \in \cdot>$;

б) если $\mathcal{Y}_1 \in \mathcal{V}_N$, то $(\mathfrak{X}_n, b) \in <\cdot$ и $(\mathfrak{X}_n, b) \in \cdot>$.

получили противоречие, так как G-грамматика предшествования.

Лемма 3. В грамматике G', полученной из грамматики предшествования G при помощи ЛС, не появилось ни одного нового КР со сложностью больше или равной (0,I).

III. Все новые правила с одинаковыми правыми сторонами, полученные при помощи ликвидации конфликтов типа P_1, получают также одинаковые левые стороны [6]. Следовательно, между новыми нетерминалами не может появиться КР.

Все это вместе дает теорему 7.

Теорема 7. Замена отношения $X \doteq Y (X, Y \in V)$ в грамматике предшествования на отношение $X \cdot\!> Y$, при помощи алгоритма ликвидации конфликтов типа P_1, не увеличивает сложности имеющихся конфликтов анализа и не вызывает новых, сложность которых превышала бы (I/I).

Учитывая, что в произвольной SLR(1) грамматике предшествования имеется конечное число КР со сложностью больше (I/I), получаем:

Теорема 8. Произвольную SLR(1) грамматику предшествования можно преобразовать в (I/I) ОККРГП только при помощи ЛС.

7. Семантическое равенство

В интересах краткости изложения теорем 2,3,4,8 используем обозначения для классов грамматик из [3, с.692], где в клетке нижняя половина обозначает технику детектирования, а верхняя – редуцирования.

LR(K)	$\Rightarrow$	LR(K)	$\Rightarrow$	LR(1)	$\Rightarrow$	SLR(1)	$\Rightarrow$	(1/1/0KK)
LR(K)		ГП		ГП		ГП		ГП

Опираясь на статьи [3,6,7] и теорему 8, можно утверждать, что все упомянутые преобразования осуществимы при помощи трех элементарных операций: стратификация влево, стратификация вправо [6, с. 387], сдвиг основы вправо [7, с. II4].

Чтобы сохранить дерево синтаксиса в первоначальном виде, необходимо различить три вида значений семантики:

а) семантика трансляции, т.е. значение семантики, связанное с элементами множества H;

б) семантика отсутствия, т.е. значение семантики, связанное с новыми правилами, появившимися в результате право- или левосторонней стратификации или первоначально принадлежащими множеству P/H. Если стратифицируемое правило принадлежит к множеству H, то производятся следующие действия:

Пусть $(*) A \rightarrow uXYv$ правило, замененное двумя новыми правилами $(**) A \rightarrow BYv$ $(***) B \rightarrow uX$, где B - новый нетерминал и $B \notin V_N$ $u, x \in V^*$ $XY \in V$. Первоначальную семантику трансляции правила $(*)$ свяжем с правилом $(**)$, а с правилом $(***)$ свяжем семантику отсутствия;

в) семантика сдвига, т.е. значение семантики, присваиваемое правилам, которые появились во время операции сдвига основы вправо.

Если во время построения дерева синтаксиса для редуцирования использовалось правило со значениями семантики трасляции или семантики отсутствия, то построение ведется по вышеописанному алгоритму. В случае же значений семантики сдвига старшей вершиной данной основы становится вершина, полученная от редуцирования первой основы после следующего сканирования. Этим описан один шаг построения дерева синтаксиса.

Теоремы 2,3,4,8 и техника построения дерева синтаксиса позволяют доказать теорему 9.

Теорема 9. Распознаватели $LR(k)$ и (I/I) ОККРГП семантически равны.

Заменив условие $\mathcal{L}(G_1)=\mathcal{L}(G_2)$ в определении семантического равенства распознавателя на условие $\mathcal{L}(G_1) \cap (V_T \setminus Q)^* = \mathcal{L}(G_2) \cap (V_T \setminus Q)^*$, получим семантическое равенство распознавателей детерминированного языка и (I/I) ОККРГП.

Литература

1. Aho A.V., Denning P.I., Ullman J.D. Weak and Mixed Strategy Precedence Parsing. J. ACM, 1972,19, 2, 225-243.

2. Aho A.V., Ullman J.D. The Theory of Parsing, Translation and Compiling. 11: Compiling. USA, New Jersey. Prentice-Hall, 1973.

3. G r a y J.N., H a r r i s onn M.A. On the Covering and Reduction Problems for Context-Free Grammars. J. ACM, 1972, 19, 4, 675-698.

4. H u a n g J.C. A Method for Program Analysis and its Applications to Program-Correctness Problems. Department of Computer Science University of Houston. June, 1974.

5. K n u t h D.E. On the Translation of Languages from Left to Right. Inform. Contr. 1965, 8, 607-639.

6. M c A f e e J., P r e s s e r L. An Algorithm for the Design of Simple Precedence Grammars. J. ACM, 1972, 19, 3, 385-395.

7. M i c u n a s M.D., S c h n e i d e r V.B. On the Ability to Cover LR(K) Grammars with LR(I) SLR(I), and (1,1) Bounded-Context Grammars. IEEE 14-th Annual Symposion on Switching and Automata Theory, 1973, 109-121.

8. W i r t h N., W e b e r H. EULER - a Generalization of ALGOL, and its Formal Definition. Pt. I. Comm. ACM, 1966, 9, 1, 13-25.

9. Б е р ж К. Теория графов и ее применение. М., 1962.

10. В о о г л а й д А.О., Т о м б а к М.О. О проблемах редуцирования в грамматиках предшествования. - "Тр.Таллинск. политехн. ин-та", 1975, № 386, с. 23-37.

11. В о о г л а й д А.О., Т о м б а к М.О. Система построения трансляторов с LR(k) семантикой. Программирование 1976, № 5, М.

A. Vooglaid

Syntactical Equality of a Precedence Detectable (0,1) or (1,0) Context Reducible Parser and a LR(k) Parser

Summary

This paper is a continuation to [10] which dealt with the reducing problems in preceding grammars.

Refinements are given to the approach from papers [3] and [7].

The paper shows syntactical equality of a precedence detectable (0,1) or (1,0) context reducible parser and a LK(k) parser (i.e. syntax trees 2 pp. 722 of both parsers for each sentence form are equal).

TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED
ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА

№ 4II 1976

УДК 512,25/26+519.3:330.115

Э. А.-Ю. Юби

МИНИМИЗАЦИЯ ФУНКЦИИ ВЕРОЯТНОСТИ МЕТОДОМ СТАТИСТИЧЕСКИХ ИСПЫТАНИЙ

В данной работе рассматривается минимизация функции вероятности методом статистических испытаний, т.е. когда целевая функция - вероятность превышения некоторого уровня - вычисляется по статистическим испытаниям как отношение числа благоприятствующих испытаний к общему числу испытаний. Доказываются условия сходимости метода с вероятностью I. В зависимости от свойств целевой функции сравниваются различные методы минимизации, отличающиеся по числу испытаний в одной точке.

I. Постановка задачи и условия сходимости

В работе [I] рассматривается задача минимизации $v(x)=P[f(x,y)\geq 0]$, где $f: R^n \times R^m \to R^1$, $x=(x_1,x_2,\ldots,x_n)\in R^n$, $y=(y_1, y_2,\ldots,y_m)\in R^m$, m-мерный случайный вектор, обладающий плотностью $p(y)$. Закон распределения y не зависит от x. При условиях, совпадающих с условиями I) - 4) приводимой ниже теоремы I, в [I] доказывается существование производной $v'(x)$ в виде $m-1$-кратного поверхностного интеграла.

$$v'(x)=\int\limits_{S(x,0)}\frac{f'_x(x,y)}{\|f'_y(x,y)\|}p(y)\,dS(x), \quad \text{где} \quad S(x,0)=\{y: f(x,y)=0\}$$

и $v'(x)=0$, если $f(x,y)\neq 0$ для всех y. Вычисление этого интеграла является в общем случае трудной задачей, вдобавок требуется еще знание плотности $p(y)$. Рассмотрим метод минимизации $v(x)$, свободный от этих недостатков. Вероятность $v(x)$ вычисляется в каждой точке как отношение числа испытаний, в которых $f(x,y)\geq 0$, к общему числу испытаний, причем испытания независимы в разных сериях. Производная

$v'(x)$, составляющие которой вычисляются по каждой координате отдельно, заменяется ее конечно-разностной аппроксимацией. Такой метод минимизации $v(x)$ сходен с методом случайного поиска. Итак,

$$x_k = x_{k-1} - \gamma_k s_k, \tag{I}$$

где $x_k \in R^n$, $0 \leq \gamma_k$ – детерминированный скалярный множитель, s_k – случайное направление. Как и в работе [2], алгоритм минимизации $v(x)$ назовем псевдоградиентным, если

$$(v'(x_{k-1}), M s_k) \geq 0. \tag{2}$$

Обозначим через $n[f(x,y) \geq 0]$ число испытаний, в которых $f(x,y) \geq 0$ при каком-то фиксированном числе испытаний N_1 в точке x. Тогда $s_k = (s_{1k}, s_{2k}, \ldots, s_{nk})$ определим так:

$$s_{ik} = \frac{\eta[f(x_{k-1}+\Delta_{ik}e_i, y) \geq 0] - \eta[f(x_{k-1}-\Delta_{ik}e_i, y) \geq 0]}{2\Delta_{ik}N_1}, \tag{3}$$

$i = 1, 2, \ldots, n$, e_i – единичные орты, Δ_{ik} – длина "пробного" шага по i-ой координате. Для простоты будем рассматривать случай

$$\Delta_{ik} = \frac{B_i}{k^\beta}, \quad \gamma_k = \frac{A}{k^\alpha}, \quad B_i > 0, \quad A > 0. \tag{4}$$

Чтобы применить общую теорему Поляка-Цыпкина о сходимости таких алгоритмов, найдем условия, когда градиент $\nabla v(x)$ удовлетворяет условию Липшица.

<u>Теорема I</u>. Пусть для любого x и достаточно малого $\bar{x}$, $\|\bar{x}\| \leq const$, выполнены следующие условия:

I) $f'_x(x,y)$, $f'_y(x,y)$, $p(y)$ существуют и являются непрерывными;

2) $S(x,0)$ равномерно ограничена в некоторой окрестности x;

3) $f'_y(x,y)$ удовлетворяет условию Липшица по y с постоянной L_1;

4) существует $\int_{S(x,0)} dS(x) = s(x) \leq s_0$;

5) $0 < C_0 \leq l(x) \leq \|f'_y(x,y)\|$ для почти всех (п.в.) $y_0 \in S(x,0)$;

6) $\|f'_x(x+\bar{x}, y_1) - f'_x(x,y_2)\| \leq L_0\|\bar{x}\|$ для п.в. $y_2 \in S(x,0)$, $y_1 \in S(x+\bar{x}, 0)$, $\|f'_x(x,y_2)\| \leq C_1$;

7) $\|f'_y(x+\bar{x}, y_2) - f'_y(x, y_2)\| \le L_2\|\bar{x}\|$ для п.в. $y_2 \in S(x,0)$,

тогда $\|\nabla v(x+\bar{x}) - \nabla v(x)\| \le L\|\bar{x}\|$,

где $L = \max\left|\frac{c_1 s_1}{c_0}, \frac{s_0}{c_0^2}(c_1 c_3 L_1 + c_1 L_2 + c_0 L_0)\right.$, а c_3 – постоянная, такая, что

$$\max_{z_1 \in S(x+\bar{x},0)} \min_{z_2 \in S(x,0)} \|z_1 - z_2\| \le c_3\|\bar{x}\|, \quad s_1 = \max_{\|\bar{x}\| \le const} \left|\frac{s(x+\bar{x}) - s(x)}{\|x\|}\right|.$$

Доказательство:

Оценим разность $\left\|\frac{f'_x(x+\bar{x}, y_1)}{\|f'_y(x+\bar{x}, y_1)\|} - \frac{f'_x(x+\bar{x}, y_1)}{\|f'_y(x+\bar{x}, y_2)\|}\right\| \le$

$$\le c_1\left|\frac{\|f'_y(x+\bar{x}, y_2)\| - \|f'_y(x+\bar{x}, y_1)\|}{\|f'_y(x+\bar{x}, y_1)\| \; \|f'_y(x+\bar{x}, y_2)\|}\right| \le \frac{c_1}{c_0^2} L_1\|y_2 - y_1\| \le \frac{c_1 c_3}{c_0^2} L_1\|x\|,$$

где существование постоянной c_3 такой, что $\|y_2 - y_1\| \le c_3\|\bar{x}\|$, следует из доказательства теоремы I в [I].

$$\left\|\int\limits_{S(x+\bar{x},0)} \frac{f'_x(x+\bar{x}, y)}{\|f'_y(x+\bar{x}, y)\|} p(y)\, dS(x+\bar{x}) - \int\limits_{S(x,0)} \frac{f'_x(x, y)}{\|f'_y(x, y)\|} p(y)\, dS(x)\right\| \le$$

$$\le \left\|\int\limits_{S(x+\bar{x},0)} \frac{f'_x(x+\bar{x}, y)}{\|f'_y(x+\bar{x}, y)\|} p(y)\, dS(x+\bar{x}) - \int\limits_{S(x+\bar{x},0)} \frac{f'_x(x, y)}{\|f'_y(x, y)\|} p(y)\, dS(x+\bar{x})\right\| +$$

$$+ \left\|\int\limits_{S(x+\bar{x},0)} \frac{f'_x(x, y)}{\|f'_y(x, y)\|} p(y)\, dS(x+\bar{x}) - \int\limits_{S(x,0)} \frac{f'_x(x, y)}{\|f'_y(x, y)\|} p(y)\, dS(x)\right\| = I_1 + I_2.$$

$$\left\|\frac{f'_x(x+\bar{x}, y_1)}{\|f'_y(x+\bar{x}, y_1)\|} - \frac{f'_x(x, y_2)}{\|f'_y(x, y_2)\|}\right\| \le \left\|\frac{f'_x(x+\bar{x}, y_1)}{\|f'_y(x+\bar{x}, y_1)\|} - \frac{f'_x(x+\bar{x}, y_1)}{\|f'_y(x+\bar{x}, y_2)\|}\right\| +$$

$$+ \left\|\frac{f'_x(x+\bar{x}, y_1)}{\|f'_y(x+\bar{x}, y_2)\|} - \frac{f'_x(x+\bar{x}, y_1)}{\|f'_y(x, y_2)\|}\right\| + \left\|\frac{f'_x(x+\bar{x}, y_1)}{\|f'_y(x, y_2)\|} - \frac{f'_x(x, y_2)}{\|f'_y(x, y_2)\|}\right\| \le$$

$$\le \frac{c_1 c_3 L_1}{c_0^2}\|\bar{x}\| + \frac{c_1 L_2}{c_0^2}\|\bar{x}\| + \frac{c_0 L_0}{c_0^2}\|\bar{x}\|.$$

Итак, $I_1 \leqslant \frac{S_0}{C_0^2}(c_1 c_3 L_1 + c_1 L_2 + c_0 L_0)\|\bar{x}\|$.

Докажем, что $S(x) = \int\limits_{S(x,0)} dS(x)$ является непрерывной функцией от x [3], $S(x_n) \longrightarrow S(x_0)$, если $x_n \to x_0 \cdot S(x) = \int\limits_{S(x,0)} dS(x) =$

$$= \lim_{\varepsilon \to 0_+} \frac{1}{2\varepsilon} \int\limits_{|f(x,y)| \leq \varepsilon} \|f'_y(x,y)\| dy. \quad S(x_0) - S(x_n) = \lim_{\varepsilon \to 0_+} \frac{1}{2\varepsilon}\Big[\int\limits_{|f(x_0,y)| \leq \varepsilon} \|f'_y(x_0,y)\| dy -$$

$$- \int\limits_{|f(x_n,y)| \leq \varepsilon} \|f'_y(x_n,y)\| dy\Big] \leqslant \lim_{\varepsilon \to 0_+} \frac{1}{2\varepsilon}\Big[\Big|\int\limits_{|f(x_0,y)| \leq \varepsilon} \|f'_y(x_0,y)\| dy - \int\limits_{|f(x_0,y)| \leq \varepsilon} \|f'_y(x_n,y)\| dy\Big| + \Big|\int\limits_{|f(x_0,y)| \leq \varepsilon} \|f'_y(x_n,y)\| dy -$$

$$- \int\limits_{|f(x_n,y)| \leq \varepsilon} \|f'_y(x_n,y)\| dy\Big|\Big] \leqslant \lim_{\varepsilon \to 0_+} \frac{1}{2\varepsilon}\Big[L_1\|x_0 - x_n\| \int\limits_{|f(x_0,y)| \leq \varepsilon} dy + \int\limits_{\substack{|f(x_0,y)| \leq \varepsilon \\ |f(x_n,y)| > \varepsilon}} \|f'_y(x_n,y)\| dy + \int\limits_{\substack{|f(x_0,y)| > \varepsilon \\ |f(x_n,y)| \leq \varepsilon}} \|f'_y(x_n,y)\| dy\Big].$$

При стремлении $n \to \infty$ выражение в квадратных скобках стремится к нулю. Для любого x и для любого достаточно малого $\|\bar{x}\| \leqslant \text{const}$ обозначим $S_1 = \max\limits_x |S(x+\bar{x}) - S(x)| \|x\|^{-1}$. $S_1 < \infty$, так как $S(x) \leqslant S_0$.

$$I_2 \leqslant \frac{c_1}{c_0}\Big|\int\limits_{S(x+\bar{x},0)} dS(x+\bar{x}) - \int\limits_{S(x,0)} dS(x)\Big| = \frac{c_1}{c_0}|S(x+\bar{x}) - S(x)| \leqslant \frac{c_1 S_1}{c_0}\|\bar{x}\|.$$

Окончательно получим $\|\nabla v(x+\bar{x}) - \nabla v(x)\| \leqslant L\|\bar{x}\|$,

где $L = \max\Big[\frac{c_1 S_1}{c_0}, \frac{S_0}{c_0^2}(c_1 c_3 L_1 + c_1 L_2 + c_0 L_0)\Big]$.

Б.Т.Поляк и Я.З. Цыпкин [2] доказали общую теорему о сходимости псевдоградиентных алгоритмов. Приводим их теорему и два следствия из нее для случая минимизации при наличии помех.

Теорема 2. (Поляк-Цыпкин) Пусть выполнены условия:

I) $v(x) \geqslant v^* > -\infty$;

2) $\|\nabla v(x+\bar{x}) - \nabla v(x)\| \leqslant L\|\bar{x}\|$ для всех x и достаточно малого $\bar{x}, \|\bar{x}\| \leqslant \text{const}$;

3) $(\nabla v(x_{k-1}), M s_k) \geqslant 0$;

4) $M\|s_k\|^2 \leqslant \lambda_k + c_1 v(x_{k-1}) + c_2(\nabla v(x_{k-1}), M s_k)$;

5) $\gamma_k \geq 0, \quad \sum_{k=1}^{\infty} \gamma_k = \infty$;

6) $\sum_{k=1}^{\infty} \gamma_k^2 \lambda_k < \infty$;

7) $\sum_{k=1}^{\infty} \gamma_k^2 < \infty$,

тогда при любом x_0 последовательность x_k, определенная(I), п.в. такая, что существует предел $v(x_k)$, и

$$\lim_{k\to\infty} (\nabla v(x_{k-1}), M s_k) = 0 \quad \text{п.в.} \tag{5}$$

При этом предполагается при фиксированных k и $x_0, \dots, x_{k-1}$, $s_1, \dots, s_{k-1}$ существование условных математических ожиданий s и $\|s\|^2$, которые обозначены как Ms_k и $M\|s_k\|^2$. В[2] теорема доказывается с помощью теорем сходимости полумартингалов.

Следствие I. Пусть в дополнение к условиям теоремы 2 $(\nabla v(x_{k-1}), Ms_k) \geq \delta(\varepsilon) > 0$ при $v(x_{k-1}) \geq v^* + \varepsilon$ для всех $\varepsilon > 0$. Тогда

$$v(x_k) \to v^* \quad \text{п.в.} \tag{6}$$

Следствие 2. Пусть в дополнение к условиям теоремы 2, множества $\{x : v(x) \leq \text{const}\}$ ограничены и $(\nabla v(x_{k-1}), Ms_k) \geq \delta(\varepsilon) > 0$ при $\|\nabla v(x_{k-1})\| \geq \varepsilon$ для всех $\varepsilon > 0$. Тогда п.в. существует подпоследовательность k_i и точка x^* такие, что $\nabla \Delta(x^*) = 0$, $x_{k_i} \to x^*$ п.в.,

$$v(x_k) \to v(x^*) \quad \text{п.в.} \tag{7}$$

Лемма I. Пусть выполнены условия (3) и (4) и

I) $0 < \alpha \leq 1$, $0 < \beta$, $0{,}5 < \alpha - \beta$, где α и β из формулы (4);

2) $\nabla v(x)$ удовлетворяет условию Липшица;

3) для любого $i = 1, 2, \dots, n$ и для любого k $v(x)$ монотонна в промежутке $[x_{k-1} - \Delta_{ik} e_i, x_{k-1} + \Delta_{ik} e_i]$, тогда выполнены все условия теоремы 2.

Доказательство:

$$Ms_{ik}=\frac{M\eta[f(x_{k-1}+\Delta_{ik}e_i,y)\geq 0]-M\eta[f(x_{k-1}-\Delta_{ik}e_i,y)\geq 0]}{2\Delta_{ik}N_1}=$$

$$=\frac{N_1 v(x_{k-1}+\Delta_{ik}e_i)-N_1 v(x_{k-1}-\Delta_{ik}e_i)}{2\Delta_{ik}N_1}=\frac{v(x_{k-1}+\Delta_{ik}e_i)-v(x_{k-1}-\Delta_{ik}e_i)}{2\Delta_{ik}},$$

$$(\nabla v(x_{k-1}),Ms_k)=\sum_{i=1}^{n} v'_{x_i}(x_k)\frac{v(x_{k-1}+\Delta_{ik}e_i)-v(x_{k-1}-\Delta_{ik}e_i)}{2\Delta_{ik}},$$

т.е. алгоритм (I) является псевдоградиентным. (Ms_{ik} – условное математическое ожидание i-ой компоненты случайного направления движения). Условие 4) теоремы 2 тоже выполняется. $M\|s_k\|^2\leq\frac{nK^{2\beta}}{4B^2}$, где $B=\max\limits_i B_i$. Условие 6) выполнено, если $0<\alpha\leq 1$, $0<\beta$, $0{,}5<\alpha-\beta$.

Очевидно, что $v(x)$ не может быть выпуклой во всем пространстве. Отметим, что из квазивыпуклости $v(x)$ не следует, что выполнены условия следствия I или 2.

Замечание. Очевидно, что вместо (3) можно использовать

$$s_{ik}=\frac{\eta[f(x_{k-1}+\Delta_{ik}e_i,y)\geq 0;f(x_{k-1}-\Delta_{ik}e_i,y)<0]-\eta[f(x_{k-1}-\Delta_{ik}e_i,y)\geq 0;f(x_{k-1}+\Delta_{ik}e_i,y)<0]}{2\Delta_{ik}N_1}$$

Ms_{ik} от этого не изменится.

В случае, когда $v'_{x_i}(x_{k-1})=0$, а $Ms_{ik}\neq 0$ алгоритм (I) остается псевдоградиентным.

Лемма 2. (Неравенство Йенсена). Пусть x – случайная величина. $v(x):R^n\to R^1$. $v(x)$ выпукла тогда и только тогда, когда $v[M(x)]\leq M[v(x)]$.

Рассмотрим для простоты записи одномерный случай, $n=1$.

$$v(x)=P[f(x,y)\geq 0],\quad y=(y_1,\dots,y_m),\quad s(x)=\frac{\eta[f(x+\Delta,y)\geq 0]-\eta[f(x-\Delta,y)\geq 0]}{2N_1\Delta}.$$

При вычислении числа реализации $\eta[f(x+\Delta,y)\geq 0]$ и $\eta[f(x-\Delta,y)\geq 0]$ можно в обоих случаях воспользоваться одними и теми же реализациями y, или же различными реализациями. Первый метод называется в литературе методом зависимых испытаний, второй – методом независимых испытаний. Если $f(x,y)$ не задана аналитически и при фиксированном x мы можем наблюдать лишь значения $f(x,y)$, то, естественно, мы можем воспользоваться только независимыми испытаниями.

Введем в рассмотрение следующие вероятности:

$\pi_1(x,\Delta)=P[f(x+\Delta,y)<0;\ f(x-\Delta,y)<0]$, $\pi_2(x,\Delta)=P[f(x+\Delta,y)\geq 0;\ f(x-\Delta,y)<0]$,

$\pi_3(x,\Delta)=P[f(x+\Delta,y)<0;\ f(x-\Delta,y)\geq 0]$, $\pi_4(x,\Delta)=P[f(x+\Delta,y)\geq 0;\ f(x-\Delta,y)\geq 0]$,

$\pi_5(x,\Delta)=\pi_1(x,\Delta)+\pi_4(x,\Delta)$,

$\kappa(x,\Delta)=P[f(x+\Delta,y)\geq 0;\ f(x-\Delta,y)\geq 0]-P[f(x+\Delta,y)\geq 0]\cdot P[f(x-\Delta,y)\geq 0]$.

В дальнейшем вместо $\pi_i(x,\Delta)$ для простоты записи будем писать π_i. Имеют место следующие очевидные соотношения:

$$\kappa=\pi_4-v(x+\Delta)v(x-\Delta),\ \pi_1+\pi_2+\pi_3+\pi_4=1,\ \pi_2+\pi_3+\pi_5=1,$$
$$\pi_2+\pi_4=v(x+\Delta),\ \pi_3+\pi_4=v(x-\Delta).$$

Из последних двух равенств получим, что

$$\pi_2-\pi_3=v(x+\Delta)-v(x-\Delta). \tag{8}$$

В случае зависимых испытаний получим

$$\begin{aligned}\pi_2&=v(x+\Delta)-v(x+\Delta)v(x-\Delta)-\kappa,\\ \pi_3&=v(x-\Delta)-v(x+\Delta)v(x-\Delta)-\kappa,\\ \pi_5&=1-v(x-\Delta)-v(x+\Delta)+2v(x+\Delta)v(x-\Delta)+2\kappa.\end{aligned} \tag{9}$$

При независимых испытаниях имеют место такие же соотношения, но $\kappa=0$. Легко проверить, что при одном испытании $s(x)$ принимает значения $-\frac{1}{2\Delta}$, 0, $\frac{1}{2\Delta}$ с вероятностями π_3, π_5, π_2, а новая точка x_1 значения $x+\frac{\gamma}{2\Delta}$, x, $x-\frac{\gamma}{2\Delta}$ с теми же вероятностями.

Вопрос об эффективности зависимых испытаний для вычисления производной функции рассматривается, например, в [7] [9]. В [5] зависимые испытания рассматриваются для многих задач, для оценки среднего значения, плотности вероятности и т.д. При вычислении $s(x)$ по формуле (3) иногда более эффективными являются зависимые испытания (когда они нам доступны). Пусть $\pi_2\geq\pi_3$, тогда по (8) $v(x+\Delta)\geq v(x-\Delta)$. Рассмотрим, например, при одном испытании $P\{\eta[f(x+\Delta,y)\geq 0]\geq \eta[f(x-\Delta,y)\geq 0]\}$.

В случае зависимых испытаний $P\{\eta[f(x+\Delta,y)\geqslant 0]\geqslant\eta[f(x-\Delta,y)\geqslant 0]\}=1-\pi_3=1-v(x-\Delta)+v(x+\Delta)v(x-\Delta)+\kappa$, т.е. при зависимых испытаниях вероятность определения правильного направления на κ больше, чем при независимых испытаниях (тогда $\kappa\equiv 0$), если коэффициент корреляции между результатами испытаний $\kappa>0$. Точно также получается, если мы сравниваем дисперсию разности оценок средних значений $\eta[f(x+\Delta,y)\geqslant 0]$ и $\eta[f(x-\Delta,y)\geqslant 0]$. Покажем, что при достаточно малом Δ, $\kappa=\kappa(x_0\ \Delta)>0$. Исключим из рассмотрения такие x_0, что $v(x_0)=0$ или $v(x_0)=1$, не представляющих интереса. Рассмотрим последовательность Δ_κ, $\Delta_\kappa\geqslant 0$, $\Delta_\kappa\to 0$. Как следует из теоремы I в [I], в предположениях I)–4) теоремы I поверхности $S(x_0+\Delta_\kappa,0)$ и $S(x_0-\Delta_\kappa,0)$ приближаются равномерно к поверхности $S(x_0,0)$. Так как $\kappa(x_0,\Delta_\kappa)=\pi_4(x_0,\Delta_\kappa)-v(x_0+\Delta_\kappa)v(x_0-\Delta_\kappa)$, а при $\Delta_\kappa\to 0$, $\pi_4(x_0,\Delta_\kappa)\to v(x_0)$ и $\kappa(x_0,\Delta_\kappa)\to v(x_0)-v^2(x_0)>0$.

Найдем значения $Ms(x_0)$ и $Ds(x_0)$ и при любом числе испытаний N_1 в одной точке и для любого $\Delta>0$.

$$Ms(x_0)=\frac{M\eta[f(x_0+\Delta,y)\geqslant 0]-M\eta[f(x_0-\Delta,y\geqslant 0]}{2N_1\Delta}=\frac{v(x_0+\Delta)-v(x_0-\Delta)}{2\Delta}=$$

$$=\frac{\pi_2(x_0)-\pi_3(x_0)}{2\Delta}, \tag{I0}$$

$$Ds(x_0)=\frac{1}{4N_1\Delta^2}[\pi_2(x_0)+\pi_3(x_0)-(\pi_2(x_0)-\pi_3(x_0))^2], \tag{II}$$

$$v[Mx(N_1)]\leqslant Mv[x(N_1)]. \tag{I2}$$

Дисперсию $Ds(x_0)$ при одном испытании $N_1=1$ можно найти непосредственно, а при N_1, $N_1>1$ испытаниях – как дисперсию среднего арифметического независимых случайных величин.

Если мы исходим из точки x_0, то при пробном шаге Δ и при рабочем шаге γ мы приходим в случайную точку $x_0-\gamma s(x_0)$, возможные значения которой обозначим через x_{-N_1}, x_{-N_1+1}, ..., x_{-1}, x_0, x_1, ..., x_{N_1-1}, x_{N_1}. Обозначим эту случайную величину как $x(N_1)$. Итак, $Mx(1) = Mx(2) = \ldots = Mx(N_1) =$

$$= x_0 - \gamma\,\frac{\pi_2(x_0) - \pi_3(x_0)}{2\Delta} = x_0 - \gamma\,\frac{v(x_0+\Delta) - v(x_0-\Delta)}{2\Delta}.$$

По неравенству Йенсена для любой выпуклой функции в Δ окрестности $v[Mx(N_1)] \leq Mv[x(N_1)]$, т.е. при независимых или зависимых испытаниях, при любом числе испытаний в точке по критерию среднего значения $v(x)$ последнее будет ограничено снизу величиной $v[Mx(N_1)] = v[Mx(1)] = v\left[x_0 - \gamma\,\frac{v(x_0+\Delta) - v(x_0-\Delta)}{2\Delta}\right]$.

Изучим теперь поведение алгоритма на участках, где $v(x)$ сохраняет постоянное значение. Докажем одну простую лемму.

<u>Лемма 3</u>. Пусть $v(x)$ монотонная функция. Чтобы $v(x)$ была тождественно постоянна, $v(x) \equiv v_0$, необходимо и достаточно, что $Ms(x) = 0 \;\; \forall x$ и $\forall \Delta > 0$.

<u>Необходимость</u>. Берем любое $x, \Delta > 0$. Возможные значения шага $s_{-N_1}, s_{-N_1+1}, \ldots, s_{-1}, s_0, s_1, \ldots, s_{N_1-1}, s_{N_1}$ соответствуют разностям числа реализаций в точках $x+\Delta$ и $x-\Delta$, равным $-N_1, -N_1+1, \ldots, -1, 0, 1, \ldots, N_1-1, N_1$. Вероятности этих значений обозначим соответственно через $p_{-N_1}, p_{-N_1+1}, \ldots, p_{-1}, p_0, p_1, \ldots, p_{N_1-1}, p_{N_1}$.

$$P\{\eta[f(x+\Delta, y) \geq 0] - \eta[f(x-\Delta, y) \geq 0] = i\} = p_i =$$

$$= \sum_{j=i}^{N_1} C_{N_1}^{j} v_0^{j} (1-v_0)^{N_1-j} C_{N_1}^{j-i} v_0^{j-i} (1-v_0)^{N_1-j+i}.$$

$$p_{-i} = \sum_{j=i}^{N_1} C_{N_1}^{j-i} v_0^{j-i} (1-v_0)^{N_1-j+i} C_{N_1}^{j} v_0^{j} (1-v_0)^{N_1-j}, \quad i = 0, 1, \ldots, N_1.$$

Следовательно, $p_i = p_{-i}$, $i = 0, 1, \ldots, N_1$. Так как $s_i = \frac{i}{2N_1\Delta}$, $i = 0, 1, \ldots, N_1$, то $Ms(x) = \sum_{i=-N_1}^{N_1} s_i p_i = \frac{1}{2N_1\Delta} \sum_{i=-N_1}^{N_1} i p_i = 0$.

<u>Достаточность</u>: Пусть x_1 и x_2 – две любые точки, причем для определенности $x_1 < x_2$. Докажем, что $v(x_1) = v(x_2)$.

Положим $x=\frac{x_1+x_2}{2}$ и $\Delta=\frac{x_2-x_1}{2}$. Представим p_i для $i=1,\dots,N_1$ в виде $p_i=\pi_3^i Q(N_1,N_1-i)$, тогда для $i=-1,\dots,-N_1$ $p_i=\pi_2^i Q(N_1,N_1-i)$. По условию $Ms(x)=0$, но

$$Ms(x)=\frac{1}{2N_1\Delta}\sum_{i=-N_1}^{N_1} i p_i=\frac{1}{2N_1\Delta}\sum_{i=1}^{N_1} i(\pi_3^i-\pi_2^i)Q(N_1,N_1-i)=$$

$$=\frac{1}{2N_1\Delta}(\pi_3-\pi_2)\sum_{i=1}^{N_1} i\sum_{j=1}^{N_1}\pi_3^{N_1-j}\pi_2^{j-1}Q(N_1,N_1-i)=0.$$

Следовательно, $\pi_3=\pi_2$, но $\pi_2-\pi_3=v(x+\Delta)-v(x-\Delta)=0$ и $v(x_1)=v(x_2)$.

Если $v(x)$ не монотонно, то из $Ms(x)=0$ при каком-то Δ следует, что $v(x+\Delta)=v(x-\Delta)$.

Рассмотрим пример. Пусть ξ_1 и ξ_2 независимые, равномерно распределенные в промежутке $[-1,1]$ случайные величины. $v(x)=P[\xi_1 x-\xi_2\geq 0]\equiv 0{,}5$. $Ms(x)=0$ при $\forall\Delta>0$ и $\forall x$.

Если $v_x'(x_0)=0$, то необязательно $Ms(x_0)=0$. Для следующей теоремы потребуется очевидная лемма.

Лемма 4. Пусть $v(x)$ выпуклая функция $x_{-N_1},\dots,x_0,\dots,x_{N_1}$ ряд равноотстоящих точек. $l_1(x)$ прямая через точки $(x_{-N_1},v(x_{-N_1}))$ и $(x_{N_1},v(x_{N_1}))$, а $l_2(x)$ полупрямая от $(x_0,v(x_0))$ через $(x_{-N_1},v(x_{-N_1}))$ и $l_3(x)$ от $(x_0,v(x_0))$ через $(x_{N_1},v(x_{N_1}))$. Пусть $d=\frac{v(x_{N_1})+v(x_{-N_1})}{2}-v(x_0)$. Для любой точки x_{-i}, $i=1,2,\dots,N_1$ выполнено $v(x_{-i})\leq l_2(x_{-i})=l_1(x_{-i})-\frac{N_1-i}{N_1}d$.

Для любой точки $x_i, i=1,2,\dots,N_1$ выполнено $v(x_i)\leq l_3(x_i)=l_1(x_i)-\frac{N_1-i}{N_1}d$.

Доказательство леммы простое и мы его не приводим.

2. Сравнение различных алгоритмов

Теорема 3. Если $v(x)$ выпуклая в Δ окрестности точки x функция, то математическое ожидание значений $v(x)$, вычисленное при $N_1>1$ испытаниях, меньше, чем при одном испытании, $Mv[x(N_1)]\leq Mv[x(1)]$.

Доказательство: Будем считать $v(x)$ выпуклой в $(x_0-\Delta, x_0+\Delta)$. Сначала найдем общее выражение для коэффициентов p_i. При одном испытании $p_{-1}=\pi_3$, $p_0=\pi_1+\pi_4=\pi_5$, $p_1=\pi_3$. При N_1 испытаниях $p_{-k}=\pi_2^k Q(N_1,N_1-k)$, $p_k=\pi_3^k Q(N_1,N_1-k)$, $k=0,1,\dots,N_1$. Очевидно, что $Q(N_1,N_1-k)$ является многочленом степени N_1-k от π_5, π_2 и π_3, причем π_2 и π_3 всегда в одинаковой степени. $Q(N_1,N_1-k)=A_k^0\pi_5^{N_1-k}+A_k^1\pi_5^{N_1-k-2}\pi_2\pi_3+$ $+A_k^2\pi_5^{N_1-k-4}\pi_2^2\pi_3^2+\dots+A_k^l\pi_5^{N_1-k-2l}\pi_2^l\pi_3^l$,

где l - целая часть числа $\frac{N_1-k}{2}$. Нетрудно проверить, что $A_k^i=\frac{N_1!}{(k+i)!\,i!\,(N_1-k-2i)!}$.

Рассмотрим прямую $l_1(x)$ через точки $(x_{-N_1}, v(x_{-N_1}))$ и $(x_{N_1}, v(x_{N_1}))$. Так как $Mx(1)=Mx(N_1)$, $Ml_1[x(1)]=Ml_1[x(N_1)]$, т.е. $l_1(x_{-N_1})\pi_2+l_1(x_0)\pi_5+l_1(x_{N_1})\pi_3=l_1(x_{-N_1})p_{-N_1}+\dots+l_1(x_0)p_0+\dots+l_1(x_{N_1})p_{N_1}$,

где p_i вычисляются по вышеприведенным формулам для функции $v(x)$. Преобразуем последнее равенство к виду

$$l_1(x_{-N_1})(\pi_2-p_{-N_1})-l_1(x_{-N_1+1})p_{-N_1+1}-\dots-l_1(x_{-1})p_{-1}+$$
$$+(\pi_5-p_0)l_1(x_0)-l_1(x_1)p_1-\dots-l_1(x_{N_1-1})p_{N_1-1}+l_1(x_{N_1})(\pi_3-p_{N_1})=0. \qquad (13)$$

Если $\pi_5-p_0\leqslant 0$, то все слагаемые, кроме первого и последнего, отрицательные, и выражение слева лишь увеличивается, если вместо $l(x_i)$ писать $v(x_i)$, $i=-N_1+1,\dots,N_1-1$,

$v(x_{-N_1})=l_1(x_{-N_1})$, $v(x_{N_1})=l_1(x_{N_1})$. $v(x_{-N_1})(\pi_2-p_{-N_1})-$ $-v(x_{-N_1+1})p_{-N_1+1}-\dots-v(x_{-1})p_{-1}+(\pi_5-p_0)v(x_0)-v(x_1)p_1-$ $\dots-v(x_{N_1-1})p_{N_1-1}+v(x_{N_1})(\pi_3-p_{N_1})\geqslant 0$,

т.е. $Mv[x(1)]\geqslant Mv[x(N_1)]$.

Рассмотрим случай $\pi_5-p_0>0$. Как в лемме 4, рассмотрим прямую $l_1(x)$ и лучи $l_2(x)$ и $l_3(x)$. Если в (13) для $i=\pm1,\pm2,\dots,\pm N_1$, $l_1(x_i)$ заменить на $v(x_i)$, то по лемме 4 значение выражения увеличится от этого не менее, чем на $\sum_{i=1}^{N_1}(p_{-i}+p_i)\frac{N_1-i}{N_1}d$. При замене $l_1(x_0)$ на $v(x_0)$

левая сторона (I3) уменьшается на $d(\pi_5 - p_0)$. Надо доказать еще, что

$$d(\pi_5 - p_0) \leq \sum_{i=1}^{N_1} (p_{-i} + p_i) \frac{N_1 - i}{N_1} d,$$

$$N_1 \pi_5 \leq N_1 p_0 + \sum_{i=1}^{N_1} (p_{-i} + p_i)(N_1 - i). \qquad (I4)$$

Докажем по индукции по N_1, что

$$N_1 \pi_3 \geq p_1 + 2 p_2 + \ldots + N_1 p_{N_1}. \qquad (I5)$$

При $N_1 = 2$, $p_1 = 2\pi_3\pi_5$, $p_2 = \pi_3^2$ и $2\pi_3 \geq 2\pi_3\pi_5 + 2\pi_3^2$ выполняется. Вместо p_K будем писать $p_K(N_1)$. Имеет место очевидное рекуррентное соотношение $p_K(N_1) = p_{K-1}(N_1-1)\pi_3 + p_K(N_1-1)\pi_5 + p_{K+1}(N_1-1)\pi_2$. Пусть выполнено индукционное предположение $a = (N_1-1)\pi_3 \geq p_1(N_1-1) + 2p_2(N_1-1) + \ldots + (N_1-1)p_{N_1-1}(N_1-1) = b$. Докажем, что

$$A = N_1 \pi_3 \geq p_1(N_1) + 2p_2(N_1) + \ldots + N_1 p_{N_1}(N_1) = B$$

$$A = N_1 p_3 \geq p_0(N_1-1)\pi_3 + p_1(N_1-1)\pi_5 +$$

$$+ p_2(N_1-1)\pi_2 + 2[p_1(N_1-1)\pi_3 + p_2(N_1-1)\pi_5 + p_3(N_1-1)\pi_2] + \ldots$$

$$+ K[p_{K-1}(N_1-1)\pi_3 + p_K(N_1-1)\pi_5 + p_{K+1}(N_1-1)\pi_2] + \ldots$$

$$+ (N_1-2)[p_{N_1-3}(N_1-1)\pi_3 + p_{N_1-2}(N_1-1)\pi_5 + p_{N_1-1}(N_1-1)\pi_2] +$$

$$+ (N_1-1)[p_{N_1-2}(N_1-1)\pi_3 + p_{N_1-1}(N_1-1)\pi_5] + N_1 p_{N_1-1}(N_1-1)\pi_3 = B.$$

$$B = b\pi_5 + b\pi_3 + b\pi_2 + \pi_3 \sum_{i=1}^{N_1-1} p_i(N_1-1) - \pi_2 \sum_{i=1}^{N_1-2} p_i(N_1-1) -$$

$$- \pi_2(N_1-1) p_{N_1-1}(N_1-1) + \pi_3 p_0(N_1-1).$$

$$B = b + \pi_3 \sum_{i=0}^{N_1-1} p_i(N_1-1) - \pi_2 \sum_{i=1}^{N_1-2} p_i(N_1-1) - \pi_2(N_1-1) p_{N_1-1}(N_1-1).$$

Необходимо, что $b + \pi_3 \geq B$, т.е. $\pi_3 \geq \pi_3 \sum_{i=0}^{N_1-1} p_i(N_1-1) - \pi_2 \sum_{i=1}^{N_1-2} p_i(N_1-1) - \pi_2(N_1-1) p_{N_1-1}(N_1-1)$.

Последнее неравенство очевидно выполняется. Итак,

$$N_1\pi_3 \geq p_1(N_1) + 2p_2(N_1) + \ldots + N_1 p_{N_1}(N_1).$$

Ввиду неравенства $Mx(1) = Mx(N_1)$, получим

$$-N_1\pi_2 + N_1\pi_3 = -N_1 p_{-N_1}(N_1) + (-N_1+1)p_{-N_1+1}(N_1) + \ldots + (-1)p_{-1}(N_1) + p_1(N_1) + \ldots + (N_1-1)p_{N_1-1}(N_1) + N_1 p_{N_1}. \quad (16)$$

Очевидно,

$$N_1(\pi_2 + \pi_3 + \pi_5) = N_1[p_{-N_1}(N_1) + \ldots + p_0(N_1) + \ldots + p_{N_1}(N_1)]. \quad (17)$$

Складывая (16) и (17) и вычитая (15), умноженное на два, получим $N_1\pi_5 \leq p_{-N_1}(N_1) + \ldots + (N_1-1)p_{-1}(N_1) + N_1 p_0(N_1) + (N_1-1)p_1(N_1-1) + \ldots + p_{N_1-1}(N_1-1)$, что и доказывает (14). В случае $\pi_3 = 0$ надо провести прямую через $(x_{-N_1}, v(x_{-N_1}))$ и $(x_0, v(x_0))$. Тогда

$$l(x_{-N_1})\pi_2 + l(x_0)\pi_5 = l(x_{-N_1})p_{-N_1} + l(x_{-N_1+1})p_{-N_1+1} + \ldots + l(x_0)p_0,$$

$$v(x_{-N_1})\pi_2 + v(x_0)\pi_5 \geq v(x_{-N_1})p_{-N_1} + v(x_{-N_1+1})p_{-N_1+1} + \ldots + v(x_0)p_0,$$

что и требовалось доказать.

Отметим, что теорема справедлива как при независимых, так и при зависимых испытаниях, а также при отрицательном коэффициенте корреляции между результатами испытаний.

Аналогично доказывается, что при вогнутом в Δ окрестности x функции $v(x)$, $Mv[x(N_1)] \geq Mv[x(1)]$.

<u>Следствие I</u>. При $N_1 \to \infty$ $Mv[x(N_1)] \to v(a)$.

<u>Доказательство</u>: Рассмотрим независимые случайные величины $x(1)$, принимающие значения $x_{-1}(1)$, $x_0(1)$, $x_1(1)$ с вероятностями π_2, π_5 и π_3 соответственно. $x(2)$ принимает значения $x_{-2}(2)$, $x_{-1}(2)$, $x_0(2)$, $x_1(2)$, $x_2(2)$ с вероятностями π_2^2, $2\pi_2\pi_5$, $\pi_5^2 + 2\pi_2\pi_3$, $2\pi_3\pi_5$, π_3^2. $x_{-1}(1) = x_{-2}(2)$, $x_1(1) = x_2(2)$. Очевидно, что $x(2) = \frac{x(1)+x(1)}{2}$. Точно так же $x(N_1) = \frac{x(1)+\ldots+x(1)}{N}$.

$$Mx(1) = Mx(2) = \ldots = Mx(N_1) = x_0 - \frac{\gamma}{2\Delta}[\pi_2(x_0) - \pi_3(x_0)] = a.$$

Тогда по теореме Колмогорова [4] $x(N_1) \to a$ при $N_1 \to \infty$ с вероятностью I. Наряду со случайной величиной $x(N_1)$ рассмотрим случайную величину $v(N_1)$, $v(N_1) = v[x(N_1)]$. Если $v(x)$ линейна от x_{-N_1} до x_{N_1}, то $Mv[x(N_1)] = Mv(N_1) \equiv v(a)$. При выпуклом в Δ окрестности $v(x)$ по неравенству Йенсена $Mv(N_1) \geq v(a)$. Как легко видеть $v(N_1) \to v(a)$ при $N_1 \to \infty$ с вероятностью единицы. Действительно, при любом $\varepsilon > 0$ сходится ряд $\sum P[|v(N_1) - v(a)| \geq \varepsilon]$, так как сходится ряд $\sum P[|x(N_1) - a| \geq \varepsilon_1]$, где ε_1 определяется из условия, что из $|v_i(N_i) - v(a)| \geq \varepsilon$ следует $|x_i(N_1) - a| \geq \varepsilon_1$ (теорема Колмогорова о трех рядах). При любом $\varepsilon > 0$ и $\gamma > 0$ существует N_0, что для всех $N_1 \geq N_0$ $P[|v(N_1) - v(a)| < \varepsilon] > 1 - \gamma$. Обозначим для $N_1 \geq N_0$ $P[|v(N_1) - v(a)| < \varepsilon] = 1 - \gamma(N_1)$. Тогда $\gamma(N_1) \leq \gamma$. Очевидно, что условное среднее $Mv_\varepsilon(N_1)$, вычисленное по $v_i(N_1)$, удовлетворяющим условию $|v_i(N_1) - v(a)| < \varepsilon$, удовлетворяет условию $\left|\frac{Mv_\varepsilon(N_1)}{1-\gamma(N_1)} - v(a)\right| < \varepsilon$. Оценим теперь разность $|Mv(N_1) - v(a)|$.

$$|Mv(N_1) - v(a)| \leq |Mv(N_i) - Mv_\varepsilon(N_1)| + \left|Mv_\varepsilon(N_1) - \frac{Mv_\varepsilon(N_1)}{1-\gamma(N_1)}\right| +$$

$$+\left|\frac{Mv_\varepsilon(N_1)}{1-\gamma(N_1)} - v(a)\right| \leq \gamma(N_1) + \frac{\gamma(N_1)\,Mv_\varepsilon(N_1)}{1-\gamma(N_1)} + \varepsilon \leq \gamma(N_1) + \frac{\gamma(N_1)}{1-\gamma(N_1)} + \varepsilon .$$

Итак, разность $Mv(N_1) - v(a)$ при увеличении N_1 может быть сделана сколь угодно малой. $Mv(N_1)$ стремится сверху к $v(a)$ при $N_1 \to \infty$.

<u>Следствие 2</u>. Если $v(x)$ линейна в Δ окрестности точки x, то из $Mx(1) = Mx(N_1)$ следует, что $Mv[x(1)] = Mv[x(N_1)]$, т.е. при сравнении по среднему значению $v(x)$ безразлично, какое число испытаний проводить в одной точке, если не принимать во внимание затраты на испытания. Если $v(x)$ выпукла в Δ окрестности точки x, то лучше проводить побольше испытаний, а в случае вогнутой в Δ окрестности точки x – по одному испытанию.

Если x^* — точка минимума и она единственна, а в качестве критерия сравнения различных способов действия выбрать $\min_x M\|x-x^*\|$, то все три случая дают одинаковое значение минимума.

Теперь возникает вопрос о сравнении методов минимизации $v(x)$ при условии $N_1N_2=\mathrm{const}$, где N_1 - число испытаний в одной точке; N_2 - число шагов. После задания критерия оптимальности эта задача сама по себе является задачей дискретного стохастического программирования. Рассмотрим простейший случай, $\mathrm{const}=2$. Согласно [8] сравнение различных алгоритмов оптимизации разумно проводить при условии равенства длин шагов и тождественности зон сбора информации. Последнее условие у нас не может быть выполнено, поэтому ограничимся лишь равенством длин шагов. Итак, задача такая: что лучше, проводить ли в точке x_0 два испытания при рабочем шаге $s(x_0)$, или в точке x_0 одно испытание при рабочем шаге $\frac{s(x_0)}{2}$, а в следующей точке (т.е. в одной из точек x_{-1}, x_0, x_1) второе испытание при рабочем шаге $\frac{s(x_0)}{2}$. Длины пробных шагов будем считать одинаковыми. Обозначим соответствующие случайные величины через $x(2)$ и $x(1,1)$.

$$Mv[x(2)]=\pi_2^2(x_0)v(x_{-2})+2\pi_2(x_0)\pi_5(x_0)v(x_{-1})+[\pi_5^2(x_0)+$$
$$+2\pi_2(x_0)\pi_3(x_0)]v(x_0)+2\pi_3(x_0)\pi_5(x_0)v(x_1)+\pi_3^2(x_0)v(x_2).$$

$$Mv[x(1,1)]=\pi_2(x_0)\pi_2(x_{-1})v(x_{-2})+[\pi_2(x_0)\pi_5(x_{-1})+\pi_5(x_0)\pi_2(x_0)]v(x_{-1})+$$
$$+[\pi_5^2(x_0)+\pi_3(x_0)\pi_2(x_1)+\pi_2(x_0)\pi_3(x_{-1})]v(x_0)+[\pi_3(x_0)\pi_5(x_1+$$
$$+\pi_5(x_0)\pi_3(x_0)]v(x_1)+\pi_3(x_0)\pi_3(x_1)v(x_2).$$

Спрашивается, когда выполнено условие

$$A=[(v_{-2}-v_{-1})(\pi_2(x_{-1})-\pi_2(x_0))+(v_0-v_{-1})(\pi_3(x_{-1})-\pi_3(x_0)]\pi_2(x_0)-$$
$$-[(v_1-v_0)(\pi_2(x_1)-\pi_2(x_0))-(v_2-v_1)(\pi_3(x_1)-\pi_3(x_0))]\pi_3(x_0)\geq 0\,? \tag{18}$$

Если $v(x)$ линейна от $x_{-1}-\Delta$ до $x_1+\Delta$, то очевидно $A=0$ (т.к. $\pi_2(x)-\pi_3(x)$ постоянна), т.е. в случае линейной в $\Delta+x_1-x_0$ окрестности $v(x)$ опять оба алгоритма дают одинаковый результат. Хотя для "большинства" выпуклых в $\Delta + x_1 - x_0$ окрестности $v(x)$ $A \geq 0$, нетрудно построить пример, когда это не так. Если $v(x)$ линейна от $x_{-1} - \Delta$ до $x_0 + \Delta$, выпукла от $x_0 + \Delta$ до $x_1 + \Delta$, то $A < 0$. Точно такое же имеет место при $N_1 N_2 > 2$. Для выпуклой в Δ окрестности x $v(x)$ нельзя получить простых аналитических условий того, что один алгоритм лучше другого, так как приходится принимать во внимание значения $v(x)$ во многих точках. Если бы такие условия и удалось получить, их надо проверить еще на примерах. Вообще говоря, при сравнении при условии $N_1 N_2 = \mathrm{const}$ следовало бы принимать во внимание и затрату машинного времени, которая неодинаковая в обоих случаях.

При сравнении по критерию $\min\limits_{x} M \|x - x^*\|$ при условии $N_1 N_2 = 2$ в формуле (I8) заменим $v(x)$ на x, а через τ_2 обозначим шаг точек $x_i, \tau_2 = x_i - x_{i-1}$, $i = -1, 0, 1, 2$. Тогда получим

$$B = \tau_2 \{ [\pi_2(x_0) - \pi_3(x_0) - (\pi_2(x_{-1}) - \pi_3(x_{-1}))] \pi_2(x_0) - \\ - [\pi_2(x_1) - \pi_3(x_1) - (\pi_2(x_0) - \pi_3(x_0))] \pi_3(x_0) \}. \tag{I9}$$

Очевидно, что $B = Mx(1,1) - Mx(2) \geq 0$, если $\pi_2(x) - \pi_3(x) = v(x+\Delta) - v(x-\Delta)$ является вогнутой функцией по x. Вводим обозначение $R_2(x_{-1}, x_0, x_1) = [\pi_2(x_0) - \pi_3(x_0) - (\pi_2(x_{-1}) - \pi_3(x_{-1}))] \pi_2(x_0) - [\pi_2(x_1) - \pi_3(x_1) - (\pi_2(x_0) - \pi_3(x_0))] \pi_3(x_0)$.

Тогда необходимое и достаточное условие того, что в точке x_0 лучше проводить по два испытания, такое:

$$\tau_2 R_2(x_{-1}, x_0, x_1) \geq 0, \tag{20}$$

где τ_2 из (I9)

При сравнении по критерию $\min\limits_{x} M \|x - x^*\|$ при условии $N_1 N_2 = 3$, необходимым и достаточным условием будет

$$\tau_3 \pi_2(x_0) R_3(x_{-2}, x_{-1} x_0) + \tau_3 (2 + \pi_5(x_0)) R_3(x_{-1}, x_0, x_1) + \\ + \tau_3 \pi_3(x_0) R_3(x_0, x_1, x_2) \geq 0. \tag{2I}$$

Аналогично могут быть выведены необходимые и достаточные условия и для остальных значений $N_1 N_2$.

Итак, достаточным условием того, что лучше проводить в точке x_0 по N_1 испытаний, чем в каждой точке по одному испытанию, в случае выпуклой в Δ окрестности точки x_i $V(x)$ по критерию $\min\limits_{x} M \| x - x^* \|$ является условие (22).

$$R_{N_1}(x_{-i}, x_{-i+1}, x_{-i+2}) \geq 0, \quad i = N_1 - 1, N_1 - 2, \ldots, -(N_1 - 1). \qquad (23)$$

Когда испытания зависимы и в добавок $\pi_3(x) \equiv 0$, то в случае выпуклой в Δ окрестности точки x $V(x)$ всегда лучше проводить по одному испытанию в точке, как следует из (I8) и (2I). При вогнутой в Δ окрестности x $V(x)$ все наоборот.

В заключение отметим, что хотя большинство утверждений было приведено для одномерного x, они применимы и для многомерного x, так как мы используем покоординатное вычисление $s(x)$.

Как следует из приводимых ниже примеров, алгоритмы минимизации $V(x)$ по методу статистических испытаний иногда сходятся очень медленно, и поэтому было уделено столько внимания на сравнение различных вариантов.

3. Решение практических задач

Рассмотрим теперь практическое решение задачи. Поскольку при сравнении в предположении $N_1 N_2 = \mathrm{const}$ условия, при которых один алгоритм лучше другого по критерию $M V(x)$, довольно сложные, и при отсутствии априорной информации о свойствах $V(x)$ мы можем судить о ней лишь по испытаниям Y, то приходится ограничиться следующим правилом: при выпуклой в Δ окрестности точки x $V(x)$ проводить по N_1 испытаний в каждой точке, а при вогнутой в Δ окрестности точки x $V(x)$ по одному испытанию в каждой точке.

Рассмотрим такой приближенный способ определения выпуклости $V(x)$ в Δ окрестности точки x. В ходе решения задачи запоминается какое-то количество точек x_k и значений $V(x_k)$. По методу наименьших квадратов проводится парабола $z = ax^2 + bx + c$. Если $a > 0$, то будем считать $V(x)$ выпуклой в Δ окрестности, при $a < 0$ вогнутой в Δ окрестности и при $a = 0$ линейной в Δ окрестности точки x. Решение практических задач показывает, что приведенное правило является достаточно хорошим, если размах точек x_k достаточно большой.

Когда мы можем воспользоваться зависимыми испытаниями, можно проводить параллельно вычисления по двум методам.

Рассмотрим минимизацию $v(x_1,x_2)=P[y_1(1-1/(1+x_1^2+x_2^2))-y_2\geq 0]$, где y_1 и y_2 - равномерно распределенные независимые случайные величины в промежутке [0,I]. $v(x_1,x_2)$ удовлетворяет условиям теорем I и 2. Очевидно $\min_x v(x)=0$, $v(0,0)=0$. Начальные данные: $x_{10}=10$, $x_{20}=8$, $\Delta_{1k}=\frac{2}{k^{\frac{1}{8}}}$, $\Delta_{2k}=\frac{2}{k^{\frac{1}{8}}}$, $\gamma_k=\frac{2}{k^{\frac{3}{4}}}$. I700 испытаний были проведены за I час I0 минут. Задача решалась на машине "Минск-22", программа была составлена на языке МАЛГОЛ-73. Результаты вычисления приведены в таблице. I_1 и I_2 означают соответственно число испытаний, таких что $s_1(x)$ и $s_2(x)$ отличны от нуля, I_3 - общее число испытаний. Были использованы зависимые испытания.

x_1	I0	I0	9	7,72	7,I2	6,34	6,00	5,64
I_1	0	0	I	7	I2	2I	26	32
x_2	8	8	7	6,03	5,72	4,82	4,45	4,20
I_2	0	0	I	5	7	I5	I9	23
I_3	I	I00	I000	3000	5000	7000	9000	I0500

x_1	5,3I	4,98	3,87	2,80	0,042
I_1	38	45	73	I09	246
x_2	3,8I	3,64	2,80	2,50	-0,042
I_2	29	32	49	76	I42
I_3	II500	I2500	I4000	I6000	I7000

Когда мы можем воспользоваться зависимыми испытаниями, алгоритм сходится быстрее (особенно когда еще $\pi_3(x)\equiv 0$.). В противном случае алгоритм сходится иногда очень медленно и экономия машинного времени становится серьезной проблемой. Например, для минимизации $v(x)=P(x-y\geq 0)$, где y равномерно распределена в промежутке [0,I], $x_0=0,7$ потребовалось в случае независимых испытаний 20 минут, чтобы достичь точки $x=0{,}025$. При этом было проведено 2000 испытаний таких, что $s(x) \neq 0$. Начальный пробный шаг был 0,1, рабочий шаг 0,01.

А для минимизации $v(x) = P[u(x) - y \geq 0]$,

где $$u(x) = \begin{cases} 0{,}2k + 0{,}1(x - 2k)^2, & 2k \leq x \leq 2k+1 \\ 0{,}1(2k+2) - 0{,}1(x - 2k - 2)^2, & 2k+1 \leq x \leq 2k+2, \end{cases}$$

а $k = 0, 1, 2, 3, 4$, y равномерно распределена в промежутке [0, 10] $x_0 = 4,5$, потребовалось в случае независимых испытаний 5 часов, чтобы достичь точки $x = 3{,}52$. При этом было проведено 5170 испытаний, таких что $S(x) \neq 0$, а общее количество испытаний было 12 069. Начальный пробный шаг был 3, рабочий шаг тоже 3. Такая затрата машинного времени объясняется тем, что в рассматриваемом участке производная $v(x)$ мало отличается от нуля и приходится отвергать много реализаций. Кроме того, процедуры вычисления значений $v(x)$ и ее производных работают много времени, в то время как для вычисления значений $v(x) = P[x - y \geq 0]$ и ее производных не было составлено процедур. Количество испытаний в одной точке в обоих случаях было 10.

Существенно большая скорость может быть достигнута использованием алгоритмов минимизации с самообучением (например, когда длина шага зависит от величины статистической производной $v(x)$), но это выходит за рамки данной статьи.

Если мы хотим вычислить значение $v(x)$ в какой-нибудь точке более точно (например, в точке, где $v(x)$ принимает минимальное значение), то формулы вычисления $v(x)$ как эмпирической частоты с заданным уровнем значимости приведены в [5] с. 244.

В заключение хочу выразить благодарность Э.В. Райку за помощь.

Литература

1. Райк Э. Дифференцируемость по параметру функции вероятности и стохастический псевдоградиентный метод для ее оптимизации.-"Известия АН ЭССР. Физика. Математика." 1975, № 1.

2. Поляк Б.Т., Цыпкин Я.З. Псевдоградиентные алгоритмы адаптации и обучения. -"Автоматика и телемеханика", 1973, № 3.

3. Ш в а р ц Л. Анализ, т. I, М., "Мир", 1972.

4. Г н е д е н к о Б.В. Курс теории вероятностей. М., Физматгиз, 1961.

5. П о л л я к Ю.Г. Вероятностное моделирование на ЭВМ. М., "Советское Радио", 1971.

6. С о б о л ь И.М. Численные методы Монте-Карло. М., "Наука", 1973.

7. Е р м а к о в С.М., Метод Монте-Карло и смежные вопросы. М., "Наука", 1971.

8. Р а с т р и г и н Л.А. Случайный поиск в процессах адаптации. М., "Наука", 1973.

E.Übi

Monte Carlo Method for Minimization of a Probability Function

Summary

To minimize the probability function $v(x)= P[f(x,y)\geqslant 0]$ the Monte Carlo method is used. The Monte Carlo methods differing in the number of trials at one point are compared. The conditions under which the algorithm converges with the probability one are found.

Цена 42 коп.