УДК 519.682.1

С. В. Листровой, д-р техн. наук
Украинская государственная академия железнодорожного транспорта
(Украина, 61050, Харьков, пл. Фейербаха, 7,
тел. (050) 9355042, E-mail: om1@yandex.ru)

О классе *NP* и *NP*-полных задачах

Показано, что SAT-задачу (satisfiability problem) нельзя считать универсальной *NP*-полной задачей, а следовательно, вопрос о существовании хотя бы одной *NP*-полной задачи остается открытым, чем объясняется безуспешность попыток установить взаимосвязь между классами *P* и *NP*.

Показано, що SAT-задачу (satisfiability problem) не можна вважати універсальною *NP*-повною задачею, а отже, питання про існування хоча б однієї *NP*-повної задачі залишається відкритим, що є поясненням безуспішності спроб встановити взаємозв'язок між класами *P* і *NP*.

К л ю ч е в ы е с л о в а: NP-полнота, полиномиальная сводимость.

Вопрос о том, действительно ли *NP*-полные задачи труднорешаемы, в настоящее время считается одним из основных открытых вопросов современной математики и теоретической кибернетики. Вопреки готовности большинства специалистов считать, что все *NP*-полные задачи труднорешаемы, прогресса ни в доказательстве ни в опровержении этого предположения не наблюдается.

Попытаемся развить теорию труднорешаемых задач, связанную с проблемой доказательства утверждений $P = NP$ или $P \neq NP$. Теория *NP*-полных задач построена для задач распознавания свойств. Задачу распознавания L можно рассматривать как состоящую из двух множеств: множества всех индивидуальных задач D_i и множества задач с ответом «да» $Y_д$, при этом $Y_д \subset D_i$.

Эти задачи состоят из двух частей. В первой части содержится описание условий задачи в терминах различных компонент: множеств, графов, чисел и др. Во второй части формулируется вопрос, на который требуется один из двух ответов: «да» или «нет». Неформально класс *NP*-полных задач определяется с помощью понятия недетерминированного алгоритма. Такой алгоритм состоит из двух этапов: угадывания и проверки. Сначала по заданной индивидуальной задаче I происходит угадывание струк-

туры S и затем, с учетом условий задачи I, осуществляется проверка детерминированным алгоритмом, которая заканчивается ответом «да» или «нет».

Как показано в [1], с помощью недетерминированного алгоритма решается задача распознавания L, если для любой индивидуальной задачи $I \in D_i$ выполнимы два следующих условия.

1. Если $I \in Y_д$, то существует такая структура S, угадывание которой приведет к тому, что стадия проверки закончится ответом «да».

2. Если $I \notin Y_д$, то не существует такой структуры S, угадывание которой для I приведет к тому, что стадия проверки закончится ответом «нет».

Понятие полиномиальной проверяемости [1] позволяет фактически выделить класс *NP*-полных задач и при этом проверяемость за полиномиальное время не обеспечивает разрешимости задачи распознавания за полиномиальное время.

Задача распознавания L называется *NP*-полной, если $L \in NP$ и любая другая задача L' из этого класса сводится к L полиномиально.

В работе [2] доказывается, что одна конкретная задача из класса *NP*, называемая SAT-задачей или задачей выполнимости булевых формул (ВЫП), обладает таким свойством, что всякая другая задача из класса *NP* может быть сведена к ней за полиномиальное время. Покажем, что данное утверждение неверно.

Формализация SAT-задачи. Рассмотрим булеву функцию $f(x_1, x_2, ..., x_n)$ в конъюнктивной форме записи:

$$f(x_1, x_2, ..., x_n) = (x_1^{\sigma_{11}} \vee x_2^{\sigma_{12}} \vee ... \vee x_n^{\sigma_{1n}}) \wedge ... \wedge (x_1^{\sigma_{m1}} \vee x_2^{\sigma_{m2}} \vee ... \vee x_n^{\sigma_{mn}}),$$

где

$$x_i^{\sigma} = \begin{cases} x_i \text{ при } \sigma = 1, \\ x_i \text{ при } \sigma = 0. \end{cases}$$

Операции $\vee$ и $\wedge$ являются булевыми и моделируют простейшие логические понятия соответственно ИЛИ и И. Для любого двоичного набора $x = (x_1, x_2, ..., x_n)$ функция принимает одно из двух возможных значений: единица или нуль. SAT-задача заключается в ответе на вопрос: существует ли набор значений переменных $x = (x_1, x_2, ..., x_n)$, обращающий функцию f в единицу.

Постановка и решение задачи. Для установления выполнимости или невыполнимости булевых функций рассмотрим сначала свойства невыполнимых функций и способы их получения.

Свойства невыполнимых булевых функций. Введем понятие минимальной конъюнктивной формы $F_л^{\min}$ произвольной булевой функции f от двух переменных с минимальным числом дизъюнктов r, при которых

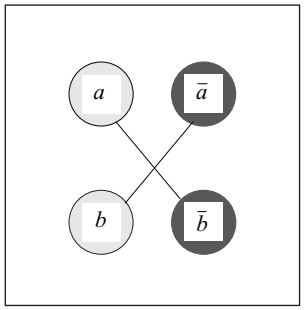

Рис. 1. Граф G_{22}

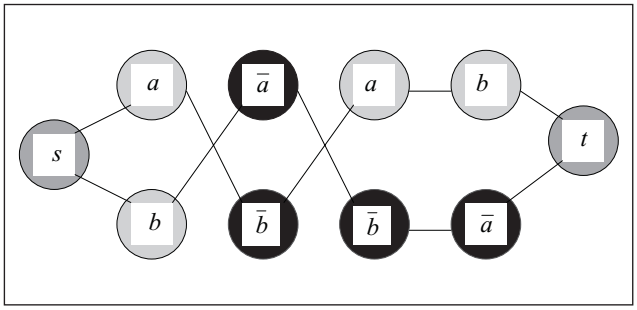

Рис. 2. Граф $G_{2л}$

$F_{л}^{\min}$ принимает значение «ложно» на всех возможных наборах функции. Для этого рассмотрим двудольный граф G_{22} (рис. 1), где одну долю образуют переменные (a, b), а вторую — переменные $(\bar{a}, \bar{b})$, ребрами соединены вершины, которые не образуют инверсные пары типа $(a, \bar{a})$.

Для возникновения противоречия в булевой функции необходимо, чтобы присутствовали переменные (a, b) и $(\bar{a}, \bar{b})$. Совершенное паросочетание в графе G_{22} обусловливает все возможные дополнения сочетаний переменных, которые можно соединить ребром, поскольку эти вершины в графе не инверсные. Дополнение дизъюнктов, содержащих переменные (a, b) и $(\bar{a}, \bar{b})$ дизъюнктами, содержащими переменные, обусловленные совершенным паросочетанием в графе G_{22}, позволит получить минимальную форму

$$F_{2л}^{\min} = (a \vee b)(\bar{a} \vee \bar{b})(a \vee \bar{b})(b \vee \bar{a}). \qquad (1)$$

Представим булеву функцию f в виде слоистого графа G, в котором каждая переменная в дизъюнктах соответствует вершине графа, а каждый слой или ярус соответствуют дизъюнкту, в котором вершины не связаны между собой.

Связи между вершинами ярусов осуществляются так. Вершина i, соответствующая переменной X_i произвольного яруса, соединяется со всеми вершинами следующего яруса, кроме вершин i^*, соответствующих переменной $\overline{X_i}$.

Вершины i и i^* будем называть инверсными вершинами графа. Порядок размещения ярусов в графе G для решаемой задачи не имеет значения. Введем также две фиктивные вершины s и t, которые связаны со всеми вершинами соответственно первого и последнего яруса графа G. Тогда граф функции (1) имеет вид, представленный на рис. 2.

Ясно, что если в графе G произвольной булевой функции f существует путь из вершины s в вершину t, в который не входят инверсные вершины, то существует набор переменных, при котором функция f принимает зна-

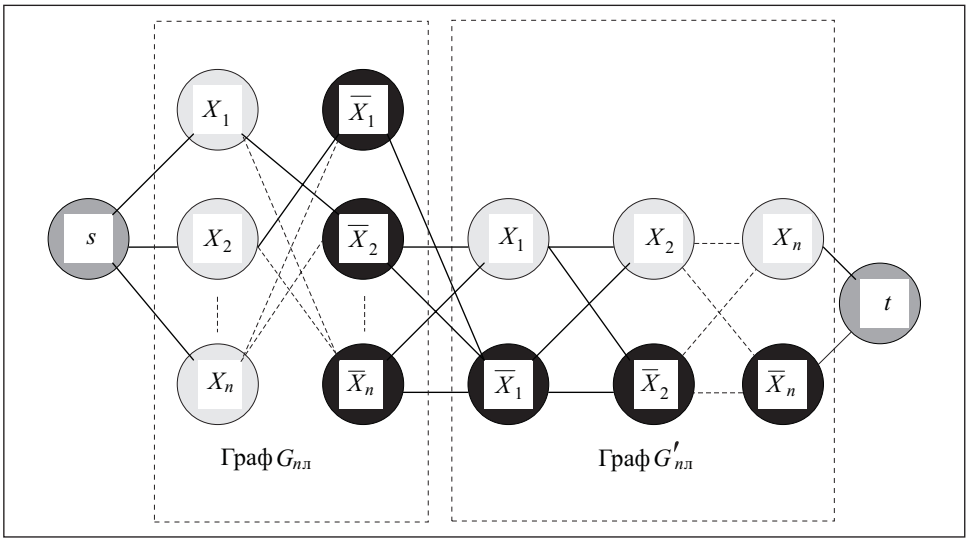

Рис. 3. Граф, порождающий функции $F_{пл}$

чение «истинно». Предположим, существует путь из s в t в графе G, не содержащий инверсных вершин $(s, i, j, ..., k, t)$. Каждой вершине соответствует переменная X_i либо $\overline{X}_i$, т.е пути $(s, i, j, ..., k, t)$ можно поставить в соответствие набор переменных $(X_i, \overline{X}_j, ..., X_k)$, среди которых нет противоречивых, что гарантирует отсутствие инверсных вершин в пути. Следовательно, можно легко указать набор переменных, при котором весь набор будет состоять из единиц. Но переменные в наборе использованы из каждого яруса графа, а это означает, что в каждом дизъюнкте присутствует единица, и функция f в этом наборе переменных принимает значение «истинно».

Таким образом, если функция f принимает на всех наборах значение «ложно», то в графе функции G не существует пути из s в t, в который не входят инверсные вершины, что видно на графе, приведенном на рис. 2. Рассматриваемые конъюнктивные формы обладают важным свойством v, а именно: изменение знака инверсии в произвольном литерале приводит к тому, что измененная функция $F_{л}$ становится выполнимой.

В этом можно легко убедиться, анализируя непосредственно граф $G_{2л}$ функции $F_{2л}^{\min}$, из которого видно, что изменения, определяемые свойством ν, приводят к появлению в графе путей, не содержащих инверсных вершин и, следовательно, функция f становится выполнимой. Если рассматривать функцию $F_{пл}$ с произвольным числом переменных и числом переменных в дизъюнкте не менее двух, то ее можно представить в виде объединения двух

графов двудольного графа $G_{пл}$ (рис. 3), в котором вершины первой доли соответствуют переменным $(\widetilde{Q}_1, \widetilde{Q}_2, ..., X_n)$, а вершины второй доли — переменным $(X_1, X_2, ..., X_n)$. Подмножества бинарных графов $G'_{пл}$, в которых связи между вершинами, соответствующими переменным $(X_1, X_2, ..., X_n)$ и $(\overline{X}_1, \overline{X}_2, ..., \overline{X}_n)$, определяются совершенными паросочетаниями в графе $G_{пл}$.

Покажем, что конъюнкция дизъюнктов, определяемых бинарными графами $\{G'_{пл}\}$, приводящими к невыполнимости функций, в общем случае равна $\widetilde{O}_1 \widetilde{O}_2 ... X_n \vee \overline{X}_1 \overline{X}_2 ... \overline{X}_n$, а произвольную функцию $F_{пл}^{\min}$ можно представить в виде

$$F_{пл}^{\min} = (\alpha_1 X_1 \vee \alpha_2 X_2 \vee ... \vee \alpha_n X_n)(\beta_1 \overline{X}_1 \vee \beta_2 \overline{X}_2 \vee ... \vee \beta_n \overline{X}_n) \times$$

$$\times (X_1 X_2 ... X_n \vee \overline{X}_1 \overline{X}_2 ... \overline{X}_n) = X_1 X_2 ... X_n (\beta_1 \overline{X}_1 \vee \beta_2 \overline{X}_2 \vee ... \vee \beta_n \overline{X}_n) \vee$$

$$\vee \overline{X}_1 \overline{X}_2 ... \overline{X}_n (\alpha_1 X_1 \vee \alpha_2 X_2 \vee ... \vee \alpha_n X_n) = \prod_{i=1}^{n} x_i f(\beta_i \overline{x}_i) \vee \prod_{i=1}^{n} \overline{x}_i f(\alpha_i x_i),$$

где

$$\alpha_i \in \{0,1\};\ \beta_i \in \{0,1\};$$

$$f(\beta_i \overline{x}_i) = (\beta_1 \overline{X}_1 \vee \beta_2 \overline{X}_2 \vee ... \vee \beta_n \overline{X}_n); \qquad (2)$$

$$f(\alpha_i x_i) = (\alpha_1 X_1 \vee \alpha_2 X_2 \vee ... \vee \alpha_n X_n).$$

Поскольку SAT-задачу можно рассматривать как задачу о покрытии, по булевой функции построим булеву матрицу B, в которой столбцам соответствуют переменные $(X_1, X_2, ..., X_n)$ и $(\overline{X}_1, \overline{X}_2, ..., \overline{X}_n)$, а строкам — дизъюнкты булевой функции. В общем случае число столбцов в матрице B равно $2n$, а число строк — числу дизъюнктов m в булевой функции. Например, для булевой функции

$$F_{пл} = (X_1 \vee X_2 \vee X_3 \vee X_4)(\overline{X}_1 \vee \overline{X}_2 \vee \overline{X}_3 \vee \overline{X}_4) \times$$

$$\times (X_1 \vee \overline{X}_3)(X_2 \vee \overline{X}_4)(X_3 \vee \overline{X}_1)(X_1 \vee \overline{X}_2)$$

матрица B имеет вид

$$B = \begin{array}{c|cccccccc} & X_1 & X_2 & X_3 & X_4 & \overline{X}_1 & \overline{X}_2 & \overline{X}_3 & \overline{X}_4 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 5 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}.$$

Столбцы, соответствующие переменным X_i и $\overline{X}_i$ в матрице B, назовем инверсными. Если в матрице B существует покрытие строк единицами, принадлежащее не инверсным столбцам, то это значит, что функция f выполнима, если такого покрытия не существует, то она невыполнима.

Итак, пусть задана булева матрица B с $2n$ столбцами и m строками, где m соответствует числу дизъюнктов в булевой функции f. Столбцы будем задавать вектором $(X_1, X_2, ..., X_n, \overline{X}_1, \overline{X}_2, ..., \overline{X}_n)$, строки — вектором $M = \{\mu_1, \mu_2, ..., \mu_m\}$. Покрытием Q строк M назовем такое множество столбцов B, которое покрывает единицами все строки M. Для определения всех покрытий матрицы используем алгебраический метод получения по импликантной таблице приведенных систем простых импликант булевых функций. Если каждый столбец из совокупности $(X_1, X_2, ..., X_n, \overline{X}_1, \overline{X}_2, ..., \overline{X}_n)$ рассматривать как «простую импликанту», покрывающую совокупность строк $M = \{\mu_1, \mu_2, ..., \mu_m\}$, а каждую строку μ_i — как набор переменных, покрываемых простыми импликантами, то матрицу B можно представить как импликантную таблицу булевой функции f. При такой интерпретации матрицы B для каждой строки μ_i дизъюнкцию столбцов b_i, покрывающих рассматриваемую строку, можно записать в следующем виде:

$$\begin{gathered} d\mu_1 = (x_1 \vee \overline{x}_k \vee ...) \\ ..... \\ d\mu_m = (x_P \vee \overline{x}_t \vee ...). \end{gathered} \tag{3}$$

Конъюнкция дизъюнкций (3) по всем строкам $\mu_1, \mu_2, ..., \mu_m$ матрицы B образует конъюнктивное представление матрицы B, содержащее все покрытия совокупности строк M:

$$k(M) = d\mu_1 \cdot d\mu_2 \cdot ... \cdot d\mu_r = (x_1 \vee \overline{x}_k \vee ...) \cdot ... \cdot (x_P \vee \overline{x}_t \vee ...).$$

Раскрывая скобки в соответствии с законами дистрибутивности, получаем дизъюнктивное представление матрицы B, образующее перечень всех возможных покрытий совокупности строк $M = \{\mu_1, \mu_2, ..., \mu_m\}$. Для выяснения существования покрытия воспользуемся принципом суперпозиции в булевой алгебре, основанным на следующих равенствах:

$$\begin{gathered} f(X_1^{\sigma 1}, X_2^{\sigma 2}, ..., X_n^{\sigma n}) = f(X_1^{\sigma 1} = 1, X_2^{\sigma 2}, ..., X_n^{\sigma n}) \vee \\ \vee f(X_1^{\sigma 1}, X_2^{\sigma 2} = 1, ..., X_n^{\sigma n}) \vee ... \vee f(X_1^{\sigma 1}, X_2^{\sigma 2}, ..., X_n^{\sigma n} = 1); \\ f'(X_1^{\sigma 1}, X_2^{\sigma 2}, ..., X_n^{\sigma n}) = f'(X_1^{\sigma 1} = 0, X_2^{\sigma 2}, ..., X_n^{\sigma n}) \vee \\ \vee f'(X_1^{\sigma 1}, X_2^{\sigma 2} = 0, ..., X_n^{\sigma n}) \vee ... \vee f'(X_1^{\sigma 1}, X_2^{\sigma 2}, ..., X_n^{\sigma n} = 0). \end{gathered} \tag{4}$$

Не нарушая принципа суперпозиции (4), исходную булеву функцию можно представить в виде

$$f(X_1^{\sigma 1},...,X_n^{\sigma n}) = f_1(X_1^{\sigma 1},...,X_n^{\sigma n}) \vee f_2(X_1^{\sigma 1},...,X_n^{\sigma n}) \vee ...$$
$$... \vee f_n(X_1^{\sigma 1},...,X_n^{\sigma n}) \vee f'_{n+1}(X_1^{\sigma 1},...,X_n^{\sigma n}) \vee f'_{n+2}(X_1^{\sigma 1},...,X_n^{\sigma n}) \vee ...$$
$$... \vee f'_{2n}(X_1^{\sigma 1},...,X_n^{\sigma n}), \qquad (5)$$

где

$$f_i(X_1^{\sigma 1},...,X_n^{\sigma n}) = X_i f(X_1^{\sigma},...,X_n^{\sigma n}); \; f'_i(X_1^{\sigma 1},...,X_n^{\sigma n}) = \overline{\widetilde{O}}_i f(X_1^{\sigma 1},...,X_n^{\sigma n}).$$

Перепишем соотношение (5) в виде

$$f(X_1^{\sigma 1},...,X_n^{\sigma n}) = (X_1 \vee X_2 \vee ... \vee X_n) f(X_1^{\sigma 1},...,X_n^{\sigma n}) \vee ...$$
$$... \vee (\overline{X}_1 \vee \overline{X}_2 \vee ... \vee \overline{X}_n) f'(X_1^{\sigma 1},...,X_n^{\sigma n}). \qquad (6)$$

Как следует из (6), для того чтобы эта булева функция была тождественно равна нулю на всех наборах переменных, необходимо чтобы функции $f_i(X_1^{\sigma 1},...,X_n^{\sigma n})$ и $f'_i(X_1^{\sigma 1},...,X_n^{\sigma n})$ были соответственно равны $(\overline{X}_1 \cdot \overline{X}_2 \cdot ... \cdot \overline{X}_n)$ и $(X_1 \cdot X_2 \cdot ... \cdot X_n)$. Следовательно, соотношение (6) вырождается в соотношение (2), что и требовалось доказать.

В дизъюнктивной нормальной форме выражение (2) имеет вид

$$F_{en}^{\min} = X_1\overline{X}_1\overline{X}_2...\overline{X}_n \vee X_2\overline{X}_1\overline{X}_2...\overline{X}_n \vee ... \vee X_n\overline{X}_1\overline{X}_2...\overline{X}_n \vee$$
$$\vee \overline{X}_1 X_1X_2 ... X_n \vee \overline{X}_2X_1X_2 ... X_n \vee ... \vee \overline{X}_nX_1X_2 ... X_n. \qquad (7)$$

Обозначим S множество индивидуальных SAT-задач. Его можно разбить на два подмножества: S^+ — подмножество выполнимых индивидуальных задач и S^- — подмножество невыполнимых индивидуальных задач. Тогда $S = S^+ \cup S^-$.

Любая невыполнимая булева функция должна содержать в себе в явном или неявном виде $F_{пл}^{\min}$. Если она содержит $F_{пл}^{\min}$ в неявном виде, то ее всегда можно преобразовать к виду (2) или (7). Для произвольной булевой функции с n переменными можно построить несколько минимальных ложных форм. Их число фактически определяется числом совершенных сочетаний в графе $G'_{пл}$. Поскольку они могут присутствовать в булевой функции в различных комбинациях, можно сделать вывод о том, что мощность множества невыполнимых индивидуальных задач S^- в экспоненциальное число раз больше чем мощности множества выполнимых индивидуальных задач S^+, т.е. справедливо неравенство

$$\left| S^- \right| >> \left| S^+ \right|. \qquad (8)$$

Как показано в работах [1—5], полиномиальная сводимость задачи распознавания I_1 к задаче распознавания I_2 означает наличие функции f, которая на основе некоторого правила Π_i отображает подмножество задач D_{i1} в подмножество задач $D_{i2}(D_{i1} \to D_{i2})$ и при этом удовлетворяет двум условиям:

1) f вычисляется с помощью полиномиального алгоритма;

2) для всех $I \in D_i$, $I \in Y_{д1}$ тогда и только тогда, когда $f(I) \in Y_{д2}$.

Рассмотрим три подмножества задач: $\{I_i\}$, $\{Z_i\}$, $\{C_i\}$. Пусть задача *I*—*NP*-полная и представляет собой универсальную задачу, задачи *Z* и *C* также являются *NP*-полными. Тогда в соответствии с введением класса *NP*-полных задач они должны полиномиально сводиться одна к другой. При этом если будет найден полиномиальный алгоритм для одной из них, то должны существовать полиномиальные алгоритмы для всех индивидуальных задач$\{I_i\}$, $\{Z_i\}$, $\{C_i\}$. Поскольку универсальной задачей может быть любая из *NP*-полных задач, должны быть справедливы все следующие сведе́ния задач:

$$\{I_i\} \to \{Z_i\} \to \{C_i\}; \tag{9}$$

$$\{I_i\} \to \{C_i\} \to \{Z_i\}; \tag{10}$$

$$\{C_i\} \to \{I_i\} \to \{Z_i\}; \tag{11}$$

$$\{C_i\} \to \{I_i\} \to \{Z_i\}; \tag{12}$$

$$\{Z_i\} \to \{C_i\} \to \{I_i\}; \tag{13}$$

$$\{Z_i\} \to \{I_i\} \to \{C_i\}; \tag{14}$$

Существуют правила Π_{iz} и Π_{zc}, которые обеспечивают сводимость задачи $I_p \to Z_p$, и при этом $\{I_p\} \in Y_{дi}$, и задачи $Z_p \to C_p$, и при этом $\{Z_p\} \in Y_{дz}$, т.е. правила преобразования Π_{iz} и Π_{zc} удовлетворяют условиям 1 и 2 полиномиальной сводимости.

Рассмотрим случай, аналогичный (8), когда структуры *S* таковы, что порождают множество индивидуальных задач$\{Z\}$, которое по мощности превышает множество индивидуальных задач$\{I\}$. Если подмножество $\{I\}$ содержит *n* индивидуальных задач, а множества $\{Z\}$ и $\{C\}$ — по $n + k$ индивидуальных задач, то для некоторого подмножества задач $\{Z_{n+1}, Z_{n+2}, ..., Z_k\}$ невозможно поставить в соответствие ни одной задачи из$\{I_i\}$. Следовательно, (9) и (10) возможны для всех задач, а (11)—(14) — не для всех, т.е. они не возможны для задач $\{C_{n+1}, C_{n+2}, ..., C_k\}$ и $\{Z_{n+1}, Z_{n+2}, ..., Z_k\}$, а значит, в этом случае утверждение о том, что все *NP*-полные задачи полиноминально сводятся одна к другой, не выполняется.

Следует заметить, что само понятие *NP*-полной задачи нуждается в уточнении. Для того чтобы *NP*-полная задача была универсальной и сво-

димой в любых направлениях внутри класса, необходимо, чтобы существовало взаимно однозначное соответствие между всеми индивидуальными задачами $\{I_i\},\{Z_i\},\{C_i\}$, т.е. для любой пары индивидуальных задач должно существовать прямое и обратное полиномиальное сведение, определяемое условиями 1 и 2.

Таким образом, если имеются подмножества задач $\{I_i\},\{Z_i\},\{C_i\}$ и мощность множества индивидуальных задач $\{I_i\}$ отличается от мощности множеств задач $\{Z_i\}$ и $\{C_i\}$, то для того чтобы доказать, что некоторая задача I является *NP*-полной, недостаточно показать, что любая индивидуальная задача $\{I_i\}$ полиномиально сводится к множеству задач $\{Z_i\}$ и $\{C_i\}$, т.е. выполняются условия 1 и 2, как это сделано при доказательстве *NP*-полноты задачи SAT в работах [2, 4, 5]. При этом следует показать, что существуют и задачи $\{I_{n+1}, I_{n+2}, ..., I_k\}$, полиномиально сводимые к задачам $\{C_{n+1}, C_{n+2}, ..., C_k\}$ и $\{Z_{n+1}, Z_{n+2}, ..., Z_k\}$, и проверяемость этих задач распознавания должна оставаться возможной за полиномиальное время.

Покажем, что SAT-задача не является универсальной. Итак, сначала Куком была доказана универсальность SAT-задачи. После того как стало известно доказательство одной *NP*-полной задачи, процесс доказательства *NP*-полноты задачи A упрощается. Для доказательства *NP*-полноты задачи $A \in NP$ достаточно показать, что какая-либо из известных *NP*-полных задач A' может быть сведена к A, так как свойство полиномиальной сводимости транзитивно, т.е. если задача A за полиномиальное время преобразуется в задачу B и если B за полиномиальное время преобразуется в C, то и A за полиномиальное время преобразуется в C.

Сначала по данной схеме была доказана *NP*-полнота шести основных задач: «трехмерное сочетание», «разбиение», «вершинное покрытие», «гамильтонов цикл», «клика». Поскольку это были первые задачи, вводимые в класс *NP*-полных после SAT-задачи, доказательство их *NP*-полноты сводилось к введению правила Π, по которому на основе некоторой произвольной SAT-задачи $y \in Y$ строилась структура S, обладающая свойством v в том и только в том случае, когда y принимает значение истинно. Например, для задачи «вершинное покрытие» в качестве структуры S использован граф G, а свойство v заключается в том, что граф G имеет вершинное покрытие с числом элементов не более K тогда и только тогда, когда выполни́м набор дизъюнкций $C = \{c_1, c_2, ..., c_m\}$, определяющих произвольную индивидуальную задачу З-ВЫП.

В общем случае SAT-задача представляет некоторое множество Y индивидуальных задач, определяемых различными способами задания логической функции. Следует заметить, что при доказательстве *NP*-пол-

ноты шести перечисленных задач сначала выбирается произвольная индивидуальная задача $y \in Y$ и по ней за полиномиальное время строится граф G, содержащий искомую структуру, обладающую нужным свойством v только в том случае, если логическая функция, соответствующая данной индивидуальной задаче, принимает значение истинно. Поскольку граф G может быть произвольным, решаем некоторую индивидуальную задачу $z \in Z$, где Z — множество задач, порождаемых использованием различных типов графов G.

При решении произвольной *NP*-полной задачи теории графов возникает обратная задача: задан произвольный граф G, и требуется установить, обладает данный граф G структурой со свойством v или нет.

Возникают вопросы: какая индивидуальная задача (y) из множества индивидуальных SAT-задач Y соответствует задаче $z \in Z$, порожденной графом G? Для всех ли задач Z существует обратное соответствие и, если существует, то как его построить по исходному графу? Будет это построение полиномиальным или нет?

В теории *NP*-полных задач [1—5] ответы на эти вопросы пока не найдены, а свойство взаимообратной полиномиальной сводимости индивидуальных задач $y \leftrightarrow z$ по умолчанию переносится на множества Y и Z.

Покажем, что если SAT-задачу рассматривать как универсальную, то схема, по которой фактически получен весь список *NP*-полных задач, не гарантирует существование соответствия задач $y \leftrightarrow z$ и их полиномиальной сводимости одна к другой.

Рассмотрим соответствие задач $y \leftrightarrow z$, в котором структуре S, обладающей свойством v, необходимо сопоставить SAT-задачу, принимающую значение истинно тогда и только тогда, когда структура S обладает свойством v.

Введем правило Π, по которому на основе некоторой произвольной SAT-задачи $y \in Y$ построена структура S, обладающая свойством v в том и только в том случае, когда y принимает значение истинно. Следует заметить, что такой подход был использован при доказательстве *NP*-полноты шести основных *NP*-полных задач.

Для того чтобы обосновать невозможность существования соответствия $y \leftrightarrow z$ при условии, что SAT-задача *NP*-полна, и при этом задача z не может быть преобразована в исходную задачу y за полиномиальное время, достаточно показать это на примере одной из шести основных *NP*-полных задач.

Рассмотрим это на примере задачи «вершинное покрытие». Приведем правило Π, использованное для доказательства *NP*-полноты задачи «вершинное покрытие»в работе [1].

Пусть $U=\{u_1, u_2, ..., u_n\}$ и $C=\{c_1, c_2, ..., c_m\}$ определяют произвольную индивидуальную задачу из З-ВЫП. Укажем граф $G(V, E)$ и положительное целое число $K \le |V|$ такое, при котором G имеет вершинное покрытие с числом элементов не более K тогда и только тогда, когда выполни́м набор дизъюнкций C. Для каждой переменной $u_i \in U$ имеется компонента $T_i = (V_i, E_i)$ набора значений истинности, где $V_i=\{u_i, \overline{u}_i\}$, $E_i=\{\{u_i, \overline{u}_i\}\}$, т.е. T_i— это пара вершин, соединенных ребром. При этом любое вершинное покрытие должно покрыть ребро из E_i, поэтому оно должно содержать, по крайней мере, одну из вершин u_i или $\overline{u}_i$.

Для каждой дизъюнкции $c_j \in C$ имеется компонента проверки выполнимости $S_j=(V'_j, E'_j)$, состоящая из трех вершин и трех соединяющих их ребер, образующих треугольник:

$$V'_j=\{a_1[j], a_2[j], a_3[j]\},$$

$$E'_j=\{\{a_1[j], a_2[j]\}, \{a_1[j], a_3[j]\}, \{a_2[j], a_3[j]\}\}.$$

Любое вершинное покрытие должно содержать хотя бы две вершины из V'_j, чтобы покрыть три ребра из E'_j. Связывающие ребра — это единственная часть конструкции, зависящая от того, какие литералы входят в дизъюнкции. Для каждой дизъюнкции $c_j \in C$ ставим в соответствие три входящих в нее литерала: x_j, l_j, p_j. Тогда связывающие ребра, исходящие из S_j, задаем в виде

$$E''_j=\{\{a_1[j], x_j\}, \{a_2[j], l_j\}, \{a_3[j], p_j\}\}.$$

Построение индивидуальной задачи «вершинное покрытие» заканчивается, если положить $K=n+2m$ и $G=(V, E)$, где

$$V=\left(\bigcup_{i=1}^{n} V_i\right)\bigcup\left(\bigcup_{j=1}^{m} V'_j\right);\ E=\left(\bigcup_{i=1}^{n} E_i\right)\bigcup\left(\bigcup_{j=1}^{m} E'_j\right)\bigcup\left(\bigcup_{j=1}^{m} E''_j\right).$$

Рассмотрим булеву функцию

$$f=(u_1\vee u_2\vee u_3)(\overline{u}_1\vee\overline{u}_2\vee u_3)(u_1\vee\overline{u}_2\vee u_3)(u_2\vee\overline{u}_1\vee u_3)\times$$

$$\times(\overline{u}_3\vee u_4\vee u_5)(\overline{u}_3\vee u_4\vee\overline{u}_5)(\overline{u}_3\vee\overline{u}_4\vee u_5)(\overline{u}_3\vee\overline{u}_4\vee\overline{u}_5). \qquad (15)$$

В соответствии с описанной процедурой *П* граф $G_1=(V_1, E_1)$ для данной функции имеет вид, приведенный на рис. 4. Число вершин, образующих минимальное покрытие в графе, не должно превысить $K=n+2m=5+16=21$. Покажем, что функция (15) невыполнима ни на одном наборе переменных. Для этого рассмотрим функцию двух переменных

$$f=(u_1\vee u_2)(\overline{u}_1\vee\overline{u}_2)(u_1\vee\overline{u}_2)(u_2\vee\overline{u}_1). \qquad (16)$$

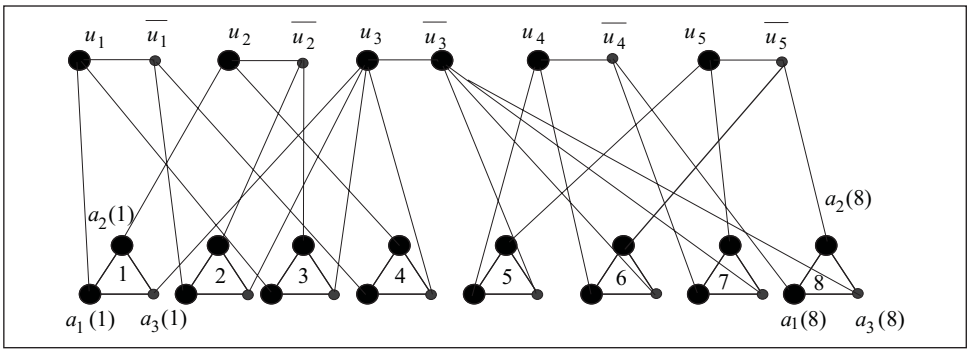

Рис. 4. Граф $G_1 = (V_1, E_1)$

Легко проверить, что функция (16) невыполнима ни на одном наборе переменных. Преобразуем (15), добавив в каждую скобку переменную u_3 и умножив функцию (16) на множитель $(\overline{u}_3 \vee u_4 \vee u_5)(\overline{u}_3 \vee u_4 \vee \overline{u}_5)(\overline{u}_3 \vee \overline{u}_4 \vee u_5) \times$ $\times (\overline{u}_3 \vee \overline{u}_4 \vee \overline{u}_5)$. Нетрудно видеть, что при этом получаем функцию (15).

Такое добавление переменных приводит к тому, что переменная u_3 будет принимать значение «ложно» в любом наборе истинности, выполняющем (15). Поэтому дизъюнкты, содержащие по две переменных в (16), эквивалентны заменяемым их дизъюнктам. Следовательно, функция (15) выполнима тогда и только тогда, когда выполнима функция (16). А поскольку (16) невыполнима ни на одном наборе переменных, то и функция (15) невыполнима ни на одном наборе переменных.

Из рис. 4 видно, что граф G_1 не содержит покрытия, состоящего из 21 вершины, а это хорошо согласуется с фактом невыполнимости функции (15) ни на одном наборе переменных. Но на графе $G_1 = (V_1, E_1)$ легко указать покрытие, состоящее из 22 вершин. На рис. 4 эти вершины обозначены более крупными кружками. При этом легко проверить, что данное покрытие в графе G_1 является минимальным.

Возникает вопрос: можно ли построить для графа G_1 такую булеву функцию, придерживаясь правила *П*, которая была бы выполнима только в случае, когда в графе G_1 минимальное покрытие не превышает 22 вершины? Очевидно, такая функция должна содержать функцию (15), и эта ее часть будет гарантией того, что покрытия, состоящего менее чем из 22-х вершин, в графе G_1 не существует, а число дизъюнктов в ее составе, необходимое для того, чтобы правило *П* выполнялось, должно быть равно (22–5)/2 = 8,5. Однако это невозможно, поскольку такая функция, во-первых, будет невыполнима на всех возможных наборах, так как включает в себя функцию (15), и, во-вторых, число дизъюнктов в функции не может быть дробным.

Если попытаться осуществить обратное преобразование для произвольного графа, учитывая правило *П*, понадобится число дизъюнктов в индивидуальной SAT-задаче, равное $m = (K - n)/2$, т.е. если $K - n$ нечетное, то число дизъюнктов будет дробным числом.

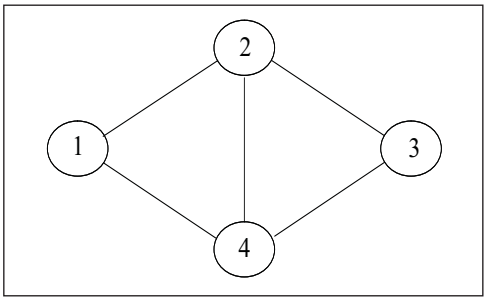

Рис. 5. Граф *G*

Следовательно, не существует такой SAT-задачи, функция которой, построенная по правилу *П* [1], принимала бы значение истинно тогда и только тогда, когда число вершин в покрытии графа G_1 не превышает 22. Таких примеров может быть столько, сколько существует невыполнимых функций, т.е. экспоненциально большое множество. Необходимо заметить, что даже если преобразование возможно, то после выполнения преобразования число истинных высказываний может оказаться экспоненциально большим, а свойству *v* исходного графа будет удовлетворять только одно истинное высказывание, обладающее некоторым свойством γ.

В работе [6] показано, что если булева функция *f*, построенная по графу $G = (V, E)$ в виде произведения дизъюнктов $(V_i \vee V_j)$, где $\{V_i\} \in \{0,1\}$, $i = \overline{1,n}$, $j = \overline{1,n}$, $i \neq j$, и при этом каждый дизъюнкт $(V_i \vee V_j)$ соответствует ребру (V_i, V_j), то все наборы переменных $\{V_i, V_j\}$, на которых она принимает значение истинно, соответствуют вершинным покрытиям в графе $G = (V, E)$.Следовательно, для перечисления всех вершинных покрытий графа $G = (V, E)$ необходимо определить те системы значений переменных $\{V_i, V_j\}$, при которых высказывание

$$f(V_1, V_2 \ldots V_n) = 1 \qquad (17)$$

истинно. Для того чтобы найти системы значений переменных $\{v_i, v_j\}$, необходимо привести левую часть (17) к минимальной дизъюнктивной нормальной форме, раскрывая скобки и пользуясь законом поглощения. Такая форма единственна ввиду отсутствия в (17) логических отрицаний.

Покажем это на примере графа *G* (рис. 5). Булева функция для графа имеет вид

$$f = (V_1 \vee V_2)(V_2 \vee V_3)(V_3 \vee V_4)(V_2 \vee V_4)(V_1 \vee V_4) =$$
$$= (V_2 \vee V_1 V_3)(V_4 \vee V_1 V_2 V_3) = V_2 V_4 \vee V_1 V_2 V_3 \vee V_1 V_3 V_4). \qquad (18)$$

Как видно из (18), в результате раскрытия скобок и приведения подобных, получен полный перечень вершинных покрытий графа *G*, приведенного

на рис. 5. Ими являются подмножества вершин: {*2*, *4*}; {*1*, *2*, *3*};{*1*, *3*, *4*}. В общем случае функция (17) содержит экспоненциальное число истинных высказываний, так как не содержит логических отрицаний. Если необходимо определить минимальное покрытие, то высказывание должно удовлетворять свойству γ, заключающемуся в том, что дизъюнкт, соответствующий минимальному вершинному покрытию, должен иметь минимальную длину, т.е. содержать наименьшее число переменных $\{V_i\}$.

Данное преобразование можно легко применить к графу G_1, но при этом для определения свойства графа *v* понадобится выполнение экспоненциального числа шагов. Если все же предположить, что в данном случае для данного преобразования существует полиномиальный алгоритм, то это будет означать, что задача «вершинное покрытие» разрешима за полиномиальное время, а это противоречит предположению о неразрешимости *NP*-полных задач за полиномиальное время.

Следовательно, если прямое и обратное преобразования существуют, то обратное преобразование может быть экспоненциальным, а не полиномиальным. Это означает, что если в примере (9)—(14) предположить, что I — это SAT-задача, то задачам $\{C_{n+1}, C_{n+2}, ..., C_k\}$ и $\{Z_{n+1}, Z_{n+2}, ..., Z_k\}$ можно поставить в соответствие задачи $\{I_{n+1}, I_{n+2}, ..., I_k\}$ такие, для которых все сводимости (9)—(14) выполнимы за полиномиальное время, но при этом «проверяемость» задачи распознавания $I_{p>n}$ может потребовать экспоненциального числа шагов.

Итак, если в качестве универсальной задачи выбрать SAT-задачу, то может возникнуть ситуация, когда полиномиальная сводимость некоторого подмножества задач внутри класса *NP*-полных задач невозможна. Это объясняется тем, что доказательство Кука справедливо для множества индивидуальных задач выполнимых булевых функций S^+, на котором, используя в качестве модели вычислителя машину Тьюринга, достаточно просто обосновывать полиномиальную сводимость. При этом полностью исключается из анализа множество индивидуальных задач S^- невыполнимых булевых функций, мощность которого существенно превышает мощность множества S^+.

Выводы. Таким образом, SAT-задачу нельзя считать универсальной *NP*-полной. Вопрос о существовании хотя бы одной *NP*-полной задачи остается открытым, поскольку универсальность ни одной задачи из класса *NP*, кроме SAT-задачи, не была доказана, и они были включены в список *NP*-полных задач лишь на основе сводимости к ним SAT-задачи.

Следовательно, известное множество задач, называемых *NP*-полными, фактически можно разделить на подмножества, внутри каждого из которых возможна полиномиальная сводимость между всеми задачами,

например вершинное покрытие, максимальное независимое множество и клика в графе. Однако, по всей видимости, сводимость между подмножествами может существовать только для некоторых индивидуальных задач [7].

Вопрос существования полиномиальной сводимости между подмножествами может оказаться алгоритмически неразрешимой задачей. Поэтому пока невозможно ответить на вопрос, существует ли хотя бы одна универсальная *NP*-полная задача, а вопрос о существовании равенства $P = NP$ просто преждевременен и пока не имеет смысла. Следует также заметить, что все выводы, сделанные в теории задач из класса *NP*, на основе предположения о «тотальной» полиномиальной сводимости в классе *NP*-полных задач могут оказаться неверными, если проблема существования универсальной задачи окажется неразрешимой.

It is shown that SAT (satisfiability) problem cannot aspire to the role of universal *NP*-full problem, but consequently a question of existence of at least one *NP*-full problem remains open, that explains the failure of all attempts to establish the relationship between the classes P and *NP*.

1. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. — М. : Мир, 1982. — 336 с.
2. *Cook S. A.* Complexity of procedures of a conclusion of theorems // Cybernetic collection of a new series. Vol. 12. — Moscow : Mir, 1975. — P. 5—15.
3. *Ахо А. V., Хопкрофт Дж. Е., Ульман Дж. D.* Построение и анализ вычислительных алгоритмов. Перевод с англ. А.О. Слисенко под ред. Ю.В. Матиясевича. — М.: Мир, 1979. — 535 с.
4. *Karp R. M.* Reducibility of Combinational Problems // Cybernetic Collection of a New Series. Vol. 12. — Moscow : Mir, 1975. — P. 16—38.
5. *Левин Л. А.* Универсальные задачи перебора//Проблемы передачи информации. — 1973. — Т. 9, вып. 3. — С. 115—116.
6. *Листровой С. В., Яблочков С. В.* Метод решения задачи определения минимальных вершинных покрытий и максимальных независимых множеств // Электрон. моделирование. — 2003. — **25**, № 2. — С. 31—40.
7. *Листровой С. В.* О полиномиальной сводимости в классе *NP*. — Ukrainian Mathematical Congress — 2009, «Algebra and Number Theory». — http://www.imath.kiev.ua/

Поступила 16.07.10;
после доработки 14.10.10

ЛИСТРОВОЙ Сергей Владимирович, д-р техн. наук, профессор, профессор кафедры специализированных компьютерных систем Украинской государственной академии железнодорожного транспорта. В 1972 г. окончил Харьковское высшее военное командно-инженерное училище. Область научных исследований — задачи дискретной оптимизации и теории графов и их приложение к анализу вычислительных систем и сетей.