ISSN 1561-2430 (Print)
ISSN 2524-2415 (Online)
УДК 519.8
https://doi.org/10.29235/1561-2430-2024-60-4-303-308

Поступила в редакцию 24.07.2024
Received 24.07.2024

Я. М. Шафранский

Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Республика Беларусь

АЛЬТЕРНАТИВНОЕ ОПРЕДЕЛЕНИЕ NP-ПОЛНОТЫ В СИЛЬНОМ СМЫСЛЕ

Аннотация. Объясняются недостатки существующего определения NP-полноты в сильном смысле. Предлагается альтернативное определение, сохраняющее все существующие результаты по доказательству NP-полноты в сильном смысле задач распознавания.

Ключевые слова: задача распознавания, псевдополиномиальный алгоритм, класс *NP*, NP-полнота, NP-полнота в сильном смысле

Для цитирования. Шафранский, Я. М. Альтернативное определение NP-полноты в сильном смысле / Я. М. Шафранский // Вес. Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. – 2024. – Т. 60, № 4. – С. 303–308. https://doi.org/10.29235/1561-2430-2024-60-4-303-308

Yakov M. Shafransky

United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Republic of Belarus

ALTERNATIVE DEFINITION OF THE STRONG NP-COMPLETENESS

Abstract. The paper explains the shortcomings of the existing definition of the strong NP-completeness. An alternative definition is proposed, that preserves all existing results on proving the strong NP-completeness of decision problems. Besides, the definition of NP-completeness and the alternative definition of the strong NP-completeness have the same structure.

Keywords: decision problem, pseudo-polynomial algorithm, class *NP*, NP-completeness, strong NP-completeness

For citation. Shafransky Y. M. Alternative definition of the strong NP-completeness. *Vestsi Natsyyanal'nai akademii navuk Belarusi. Seryya fizika-matematychnykh navuk = Proceedings of the National Academy of Sciences of Belarus. Physics and Mathematics series*, 2024, vol. 60, no. 4, pp. 303–308 (in Russian). https://doi.org/10.29235/1561-2430-2024-60-4-303-308

Введение. Корректное определение основных понятий играет исключительно важную роль в изложении любой науки, особенно математики. Существующее определение NP-полноты в сильном смысле имеет, на взгляд автора, недостатки, которые приводили ранее и могут приводить в последующем к появлению ошибок при доказательстве NP-полноты (или NP-трудности) в сильном смысле тех или иных задач.

Понятие NP-полноты в сильном смысле введено в работе [1] и практически в неизменном виде включено в фундаментальную монографию [2].

Прежде чем переходить к обсуждению NP-полноты в сильном смысле, напомним несколько определений, необходимых для понимания дальнейших рассуждений.

Формально под термином «*задача распознавания* Π» понимается пара (D_Π, Y_Π), где D_Π – множество (бесконечное) всех примеров задачи, а Y_Π – множество всех ее да-примеров, т. е. примеров, которым соответствует ответ «да» (любому примеру из $D_\Pi \backslash Y_\Pi$ соответствует ответ «нет»).

Подзадача (или частный случай) Π′ задачи Π – это задача, определяемая парой $(D_{\Pi'}, Y_{\Pi'})$, где $D_{\Pi'} \subseteq D_\Pi$, а $Y_{\Pi'} = Y_\Pi \cap D_{\Pi'}$.

С любой разумной схемой кодирования e задачи Π связана пара функций Length[I] и Max[I], где I – пример задачи Π. Функция Length[I] полиномиально связана с длиной $|e(I)|$ кода примера I задачи Π: существует такой полином p, что для любого примера $I \in D_\Pi$ выполняется Length[I] $\le$ $\le p(|e(I)|)$ и $p(\text{Length}[I]) \ge |e(I)|$. Функция Max[$I$] сопоставляет примеру I величину наибольшего (по модулю) числа в формулировке примера I.

Задача П называется *нечисловой задачей*, если существует такой полином p, что для любого $I \in D_{\Pi}$ выполняется

$$\mathrm{Max}[I] \le p(\mathrm{Length}[I]). \tag{1}$$

Алгоритм решения задачи П называется *псевдополиномиальным*, если его временная сложность не превосходит некоторого полинома $p(\mathrm{Length}[I], \mathrm{Max}[I])$. Далее при необходимости используются обозначения $\mathrm{Length}_{\Pi}[I]$ и $\mathrm{Max}_{\Pi}[I]$.

Нетрудно заметить, что NP-полная нечисловая задача П не может иметь не только полиномиального, но и псевдополиномиального алгоритма решения (если $P \ne NP$), поскольку ввиду условия (1) любой псевдополиномиальный алгоритм является одновременно и полиномиальным алгоритмом решения нечисловой задачи.

Последнее свойство положено в основу следующего, введенного в работе [1], определения NP-полноты в сильном смысле.

Для задачи распознавания П и полинома p обозначим через Π_p подзадачу, включающую только те примеры I задачи П, для которых выполнено соотношение $\mathrm{Max}[I] \le p(\mathrm{Length}[I])$. Задачу П назовем NP-полной в сильном смысле, если $\Pi \in NP$ и существует такой полином p, что задача Π_p является NP-полной (см. [1, 2]).

Анализ имеющегося определения. В чем видятся недостатки приведенного определения NP-полноты в сильном смысле? Дело в том, что известны ситуации, когда из NP-полноты (или NP-трудности) подзадачи не следует NP-полнота (или NP-трудность) исходной задачи. Одна из таких ситуаций описана в [3]. Более того, известен пример, когда полиномиально разрешимая задача имеет NP-полную в сильном смысле подзадачу (см. [4]). Рассмотрим один из примеров такого рода.

Для ориентированных графов $G_1 = (V_1, A_1)$ и $G_2 = (V_2, A_2)$ обозначим через G_1SG_2 их последовательную композицию, т. е. граф $G = (V_1 \cup V_2, A)$, полученный из G_1 и G_2 добавлением всех таких новых дуг (v_1, v_2), что $v_1 \in V_1$, а $v_2 \in V_2$. В итоге в G каждая вершина множества V_1 соединена исходящей дугой с каждой вершиной множества V_2.

Дано число $n > 3$ и $n!$ ориентированных n-вершинных графов $G_1,\ldots,G_{n!}$. Пусть граф G является последовательной композицией упомянутых n-вершинных графов: $G = G_1SG_2S\ldots SG_{n!}$. Нужно определить, существует ли в графе G гамильтонов путь.

Имея в виду разумную схему кодирования, можно положить для примера I сформулированной задачи $\mathrm{Length}[I] = n^2n!$. Нетрудно заметить, что гамильтонов путь в G существует тогда и только тогда, когда такой путь существует в каждом из графов $G_1,\ldots,G_{n!}$. Даже если для анализа каждого из графов $G_1,\ldots,G_{n!}$ воспользоваться алгоритмом полного перебора, то временная сложность полученного алгоритма будет квадратичной (по отношению к $\mathrm{Length}[I]$).

Рассмотрим специальный случай задачи, в котором $G_1 = G_2 = \ldots = G_{n!}$, т. е. в качестве графа G мы имеем последовательную композицию $n!$ идентичных графов. В этом случае для задания примера задачи достаточно описать граф G_1 и указать число $n!$ копий такого графа. Соответственно, для подзадачи имеем $\mathrm{Length}'[I] = n^2$ (длина кода числа $n!$ не превосходит $n \log n$). Очевидно, в рассматриваемом случае задача определения гамильтоновости графа G эквивалентна задаче определения гамильтоновости графа G_1, что свидетельствует об NP-полноте в сильном смысле рассматриваемого специального случая.

Чем обусловлено существование таких необычных, на первый взгляд, ситуаций?

Говоря о специальном случае задачи, обычно имеют в виду не просто набор примеров задачи, являющийся подмножеством множества D_{Π}, а такой набор, который получается из исходной формулировки задачи за счет сужения области возможных значений всех или некоторых параметров задачи. Например, если в задаче о наибольшей клике ограничиться лишь планарными графами, то из NP-трудной в сильном смысле задачи мы получим полиномиально разрешимую подзадачу. Известно достаточно много примеров полиномиально разрешимых задач, являющихся подзадачами NP-полных (или NP-трудных) задач. Такое превращение трудной задачи в полиномиально разрешимую происходит за счет того, что подзадача может обладать некоторыми полезными свойствами, которыми не обладает исходная задача.

Точно так же за счет специфических свойств подзадачи иногда возможно представить примеры подзадачи кодами, длины которых существенно меньше длин кодов этих же примеров, рассматриваемых в качестве примеров исходной задачи. Именно такая ситуация возникает в представленной выше задаче о гамильтоновости последовательной композиции большого числа ориентированных графов: функции Length[I] для исходной задачи и Length′[I] для подзадачи не могут быть полиномиально связаны. Если рассматривать пример подзадачи в качестве примера исходной задачи, то в этом случае информация об идентичности графов $G_1,\ldots,G_{n!}$ отсутствует, поскольку в формулировке исходной задачи об этих графах сказано лишь, что каждый из них имеет n вершин. Поэтому при любой разумной схеме кодирования в коде примера будут представлены все $n!$ графов, так как схема кодирования может использовать лишь ту информацию, которая представлена в условии задачи, схема не может и не должна заниматься распознаванием индивидуальных особенностей отдельных примеров задачи. Что же касается подзадачи, то в ее условии имеется прямое указание на идентичность графов, что и используется схемой кодирования, приводя в итоге к построению кодов принципиально меньшей длины.

Можно возразить, что в рассматриваемой ситуации для кодирования примеров исходной задачи и примеров подзадачи используются разные схемы кодирования. В итоге алгоритм решения задачи не сможет решать примеры подзадачи: алгоритм просто «не поймет» код примера подзадачи. Эта неприятность легко преодолима. Пусть для задачи имеется некоторая разумная схема кодирования. Внесем в нее небольшое изменение: код каждого объекта из условия задачи будем сопровождать указанием количества копий этого объекта. В результате длины кодов несколько возрастут (не более чем в 2 раза), но останутся полиномиально связанными с длинами кодов при любой другой разумной схеме кодирования. При такой модификации структура кодов примера задачи и примера подзадачи будет одной и той же. В нашем примере код каждого графа в исходной задаче будет сопровождаться числом 1, а в подзадаче мы будем иметь код лишь одного графа, сопровождаемый числом $n!$.

Приведенное выше определение NP-полной в сильном смысле задачи является корректным лишь в том случае, когда в качестве кодов примеров подзадачи Π_p используются их коды из исходной задачи. Фактически вместо множества D_Π рассматривается множество кодов элементов D_Π, а D_{Π_p} – это подмножество множества этих кодов. В результате может оказаться, что коды примеров подзадачи Π_p, рассматриваемой в качестве самостоятельной задачи, не соответствуют определению разумной схемы кодирования – они являются неоправданно длинными. Такое положение вещей и приводит к недоразумениям, когда, доказав NP-полноту или NP-трудность некоторой задачи, исследователи делают ошибочное заключение об NP-полноте или NP-трудности более общей задачи, забывая о том, что коды примеров подзадачи могут быть принципиально короче кодов тех же примеров, рассматриваемых относительно более общей задачи. Такая ситуация описана, в частности, в [3].

Альтернативное определение. Предлагаемое ниже альтернативное определение NP-полной в сильном смысле задачи вообще не использует понятие подзадачи, оно основано на несколько модифицированном понятии псевдополиномиальной сводимости задач распознавания из [1, 2]. Заметим, что все известные доказательства NP-полноты (или NP-трудности) в сильном смысле получены путем построения псевдополиномиального сведения. В результате все задачи, являющиеся NP-полными (или NP-трудными) в сильном смысле относительно существующего определения, остаются таковыми и относительно нового определения. Кроме того, сама схема введения понятия NP-полноты в сильном смысле становится полностью идентичной схеме введения понятия NP-полноты (в обычном смысле).

Определим отношение псевдополиномиальной сводимости задач распознавания.

Пусть имеется пара задач распознавания Π и Π' с сопоставленными им разумными схемами кодирования.

Будем говорить, что словарная функция Φ, переводящая задачу Π в задачу Π' (т. е. сопоставляющая коду каждого примера $I \in D_\Pi$ код некоторого примера $I' \in D_{\Pi'}$), реализует *псевдополиномиальное сведение* задачи Π к задаче Π', если Φ удовлетворяет следующим условиям:

(а) для любого примера $I \in D_\Pi$ соотношение $I \in Y_\Pi$ выполняется тогда и только тогда, когда выполняется $\Phi(I) \in Y_{\Pi'}$;

(б) временная сложность вычисления функции Φ ограничена сверху некоторым полиномом q_1 от двух переменных $\mathrm{Length}_\Pi[I]$ и $\mathrm{Max}_\Pi[I]$;

(в) существует такой полином q_2 от двух переменных, что для любого $I \in D_\Pi$ выполняется соотношение

$$q_2(\mathrm{Max}_\Pi[I], \mathrm{Length}_\Pi[I]) \geq \mathrm{Max}_{\Pi'}[\Phi(I)]. \quad (2)$$

Приведенное определение псевдополиномиальной сводимости является ослабленной версией определения из [1, 2], где наряду с условиями (а), (б) и (в) присутствует еще одно условие:

(г) существует такой полином q_3, что для любого $I \in D_\Pi$ выполняется соотношение $q_3\left(\mathrm{Length}_{\Pi'}[\Phi(I)]\right) \geq \mathrm{Length}_\Pi[I]$.

Будем говорить, что *задача распознавания* Π *псевдополиномиально сводится к задаче распознавания* Π′, если для этой пары задач существует словарная функция Φ, удовлетворяющая условиям (а), (б) и (в).

Введем понятие NP-полной в сильном смысле задачи по аналогии с понятием NP-полноты (в обычном смысле).

Задачу распознавания Π будем называть NP-*полной в сильном смысле*, если $\Pi \in NP$ и любая задача из класса *NP* псевдополиномиально сводится к Π.

Напомним для сравнения: задача распознавания Π называется NP-*полной*, если $\Pi \in NP$ и любая задача из класса *NP* полиномиально сводится к Π. Как нетрудно заметить, определение NP-полноты и альтернативное определение NP-полноты в сильном смысле имеют одну и ту же структуру.

Введя новое определение NP-полноты в сильном смысле, мы потеряли все свойства, вытекающие из старого определения. Приводимые ниже утверждения описывают свойства как модифицированного отношения псевдополиномиальной сводимости, так и NP-полных в сильном смысле задач (относительно нового определения).

Лемма 1. *Отношение псевдополиномиальной сводимости является транзитивным.*

Справедливость леммы следует из транзитивности «стандартного» отношения псевдополиномиальной сводимости (соответствующее доказательство можно найти в [2]).

Отметим, что из ограниченности времени вычисления приведенной выше функции Φ полиномом от $\mathrm{Max}_\Pi[I]$ и $\mathrm{Length}_\Pi[I]$ при преобразовании примера задачи Π в пример задачи Π′ следует существование такого полинома τ, что $\tau(\mathrm{Max}_\Pi[I], \mathrm{Length}_\Pi[I]) \geq \mathrm{Length}_{\Pi'}[\Phi(I)]$.

Как обычно, символом $\propto$ обозначается отношение полиномиальной сводимости (языков и задач распознавания).

Лемма 2. *Пусть* Π и $\overline{\Pi}$ *– задачи распознавания,* $\overline{\Pi}$ *– нечисловая задача и* $\Pi \propto \overline{\Pi}$. *Тогда задача* Π *псевдополиномиально сводится к задаче* $\overline{\Pi}$.

Доказательство. Пусть Φ – словарная функция, реализующая полиномиальное сведение Π к $\overline{\Pi}$. Покажем, что Φ реализует одновременно и псевдополиномиальное сведение Π к $\overline{\Pi}$. Для этого достаточно показать существование такого полинома q_2, что для любого примера $I \in D_\Pi$ выполняется $q_2(\mathrm{Max}_\Pi[I], \mathrm{Length}_\Pi[I]) \geq \mathrm{Max}_\Pi[\Phi(I)]$. Поскольку $\overline{\Pi}$ – нечисловая задача, то существует такой полином p, что для любого примера $I' \in D_{\overline{\Pi}}$ справедливо неравенство $p(\mathrm{Length}_{\overline{\Pi}}[I']) \geq \mathrm{Max}_{\overline{\Pi}}[I']$).

Словарная функция Φ реализует полиномиальное сведение Π к $\overline{\Pi}$. Поэтому существует такой полином p', что $p'(\mathrm{Length}_\Pi[I]) \geq \mathrm{Length}_{\overline{\Pi}}[\Phi(I)]$. В совокупности с нечисловым характером задачи $\overline{\Pi}$ это влечет существование такого полинома p'', что $p''(\mathrm{Length}_\Pi[I]) \geq \mathrm{Max}_{\overline{\Pi}}[\Phi(I)]$, откуда $p''(\mathrm{Length}_\Pi[I]) + \mathrm{Max}_\Pi[I] \geq \mathrm{Max}_{\overline{\Pi}}[\Phi(I)]$. Следовательно, $q_2(\mathrm{Max}_\Pi[I], \mathrm{Length}_\Pi[I]) \geq \mathrm{Max}_{\overline{\Pi}}[\Phi(I)]$ для полинома $q_2(\mathrm{Max}_\Pi[I], \mathrm{Length}_\Pi[I]) = p''(\mathrm{Length}_\Pi[I]) + \mathrm{Max}_\Pi[I]$. Лемма 2 доказана.

Теорема 1. *Задача распознавания* Π *является* NP-*полной в сильном смысле тогда и только тогда, когда* $\Pi \in NP$ *и существует нечисловая* NP-*полная задача распознавания* $\overline{\Pi}$, *которая псевдополиномиально сводится к* Π.

Доказательство. Пусть Π – NP-полная в сильном смысле задача. Поскольку любая задача из класса *NP* псевдополиномиально сводится к Π, то в качестве $\overline{\Pi}$ можно взять любую нечисловую NP-полную задачу.

Предположим теперь, что $\Pi \in NP$ и существует нечисловая NP-полная задача $\bar{\Pi}$, которая псевдополиномиально сводится к Π. Из NP-полноты задачи $\bar{\Pi}$ и леммы 2 следует, что любая задача из класса NP псевдополиномиально сводится к $\bar{\Pi}$. Из транзитивности отношения псевдополиномиальной сводимости следует, что любая задача из класса NP псевдополиномиально сводится к Π. Теорема 1 доказана.

З а м е ч а н и е. Поскольку любая задача псевдополиномиально сводится к себе, то любая NP-полная нечисловая задача является NP-полной в сильном смысле.

Т е о р е м а 2. *Пусть задача распознавания* Π *является* NP-*полной в сильном смысле. Тогда* Π *является* NP-*полной задачей, а из псевдополиномиальной разрешимости* Π *следует полиномиальная разрешимость любой задачи из класса NP.*

Д о к а з а т е л ь с т в о. Из теоремы 1 и NP-полноты в сильном смысле задачи Π следует существование нечисловой NP-полной задачи $\bar{\Pi}$, которая псевдополиномиально сводится к Π.

Из определения функции Φ, реализующей псевдополиномиальное сведение задачи $\bar{\Pi}$ к задаче Π, следует, что временная сложность ее вычисления не превосходит некоторого полинома $q_1(\mathrm{Max}_{\Pi}[I], \mathrm{Length}_{\bar{\Pi}}[I])$. Из условия (1) получаем, что эта временная сложность не превосходит некоторого полинома $\bar{q}(\mathrm{Length}_{\bar{\Pi}}[I])$. Следовательно, Φ является одновременно и полиномиальным сведением $\bar{\Pi}$ к Π. Из $\Pi \in NP$ и $\bar{\Pi} \propto \Pi$ следует, что Π является NP-полной задачей.

Нетрудно проверить, что из условий (2) и (1) следует справедливость соотношений $\mathrm{Max}_{\Pi}[\Phi(I)] \le q_2(\mathrm{Max}_{\bar{\Pi}}[I], \mathrm{Length}_{\bar{\Pi}}[I]) \le q_2\left(p(\mathrm{Length}_{\bar{\Pi}}[I]), \mathrm{Length}_{\bar{\Pi}}[I]\right) = q'(\mathrm{Length}_{\bar{\Pi}}[I])$ для полинома $q'(x) = q_2\left(p(x), x\right)$.

Поскольку Φ является полиномиальным сведением $\bar{\Pi}$ к Π, то, очевидно, $\mathrm{Length}_{\Pi}[\Phi(I)] \le \tau'(\mathrm{Length}_{\bar{\Pi}}[I])$ для некоторого полинома $\tau'(x)$.

Предположим, что существует псевдополиномиальный алгоритм Ψ решения задачи Π, т. е. для некоторого полинома ρ время работы Ψ не превосходит $\rho(\mathrm{Max}_{\Pi}[I'], \mathrm{Length}_{\Pi}[I'])$ для любого примера $I' \in D_{\Pi}$. Для произвольного примера $I \in D_{\bar{\Pi}}$ существует такой пример $I' \in D_{\Pi}$, что $I' = \Phi(I)$. Из полученных выше оценок для величин $\mathrm{Max}_{\Pi}[I']$ и $\mathrm{Length}_{\Pi}[I']$ следует, что $\rho(\mathrm{Max}_{\Pi}[I'], \mathrm{Length}_{\Pi}[I']) \le \rho\left(q'(\mathrm{Length}_{\bar{\Pi}}[I]), \tau'(\mathrm{Length}_{\bar{\Pi}}[I])\right) = \rho'(\mathrm{Length}_{\bar{\Pi}}[I])$ для полинома $\rho'(x) = \rho\left(q'(x), \tau'(x)\right)$.

Итак, для любого примера $I \in D_{\bar{\Pi}}$ за время, ограниченное полиномом $\bar{q}(\mathrm{Length}_{\bar{\Pi}}[I]) = q_1\left(p(\mathrm{Length}_{\bar{\Pi}}[I]), \mathrm{Length}_{\bar{\Pi}}[I]\right)$, строится пример $I' = \Phi(I) \in D_{\Pi}$, а за время, ограниченное полиномом $\rho'(\mathrm{Length}_{\bar{\Pi}}[I])$ определяется, верно ли соотношение $I' \in Y_{\Pi}$. Поскольку $I \in Y_{\bar{\Pi}}$ выполняется тогда и только тогда, когда $I' \in Y_{\Pi}$, мы получаем в итоге полиномиальный алгоритм решения NP-полной задачи $\bar{\Pi}$. Теорема 2 доказана.

С л е д с т в и е. *Ни одна из* NP-*полных в сильном смысле задач не может иметь псевдополиномиального алгоритма решения, если* $P \ne NP$.

Теорема 3 ниже дает простой и удобный способ доказательства NP-полноты в сильном смысле задач распознавания.

Т е о р е м а 3. *Пусть* Π *является* NP-*полной в сильном смысле задачей распознавания. Если* Π *псевдополиномиально сводится к задаче распознавания* Π' *и* $\Pi' \in NP$, *то* Π' *является* NP-*полной в сильном смысле задачей.*

Справедливость теоремы 3 следует непосредственно из транзитивности отношения псевдополиномиальной сводимости.

Отбросив условие (г) в стандартном определении псевдополиномиальной сводимости задач, мы теоретически несколько расширили как возможности отношения псевдополиномиальной сводимости, так и класс NP-полных в сильном смысле задач, сохранив их основные свойства. Условие (г) практически всегда выполняется «автоматически» при построении псевдополиномиального сведения одной задачи к другой. Так что найти пример ситуации, в которой строится псевдополиномиальное сведение, не удовлетворяющее условию (г), не просто, хотя с точки зрения теории такая ситуация возможна.

Список использованных источников

1. Garey, M. R. “Strong” NP-completeness results: motivation, examples, and implications / M. R. Garey, D. S. Johnson // J. Ass. Comput. Machinery. – 1978. – Vol. 25, № 3. – P. 499–508. https://doi.org/10.1145/322077.322090

2. Garey, M. R. Computers and Intractability. A Guide to the Theory of NP-completeness / M. R. Garey, D. S. Johnson. – New York: W. H. Freeman and Company, 1979. – 348 p.

3. Webster, S. Note on “Parallel Machine Scheduling with Batch Setup Times” / S. Webster // Oper. Res. – 1998. – Vol. 46, № 3. – P. 423. https://doi.org/10.1287/opre.46.3.423

4. Shafransky, Y. M. On some contradictions in theory of computational complexity / Y. M. Shafransky // Third Workshop on Models and Algorithms for Planning and Scheduling. – Cambridge, 1997. – P. 42.

References

1. Garey M. R., Johnson D. S. “Strong” NP-completeness results: motivation, examples, and implications. *Journal of the Association for Computing Machinery*, 1978, vol. 25, no. 3, pp. 499–508. https://doi.org/10.1145/322077.322090

2. Garey M. R., Johnson D. S. *Computers and Intractability. A Guide to the Theory of NP-completeness.* New York, W. H. Freeman and Company, 1979. 348 p.

3. Webster S. Note on “Parallel Machine Scheduling with Batch Setup Times”. *Operations Research*, 1998, vol. 46, no. 3, p. 423. https://doi.org/10.1287/opre.46.3.423

4. Shafransky Y. M. On some contradictions in theory of computational complexity. *Third Workshop on Models and Algorithms for Planning and Scheduling.* Cambridge, 1997, p. 42.

Информация об авторе

Шафранский Яков Михайлович – кандидат физико-математических наук, доцент, ведущий научный сотрудник лаборатории математической кибернетики, Объединенный институт проблем информатики Национальной академии наук Беларуси (ул. Сурганова, 6, 220072, Минск, Республика Беларусь). E-mail: shafr-04@yandex.by

Information about the author

Yakov M. Shafransky – Ph. D. (Physics and Mathematics), Associate Professor, Leading Researcher of the Laboratory of Mathematical Cybernetics, United Institute of Informatics Problems of the National Academy of Sciences of Belarus (6, Surganov Str., Minsk, 220072, Republic of Belarus). E-mail: shafr-04@yandex.by