

ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 33 № 2 2026

Новосибирск
Издательство Института математики

ОБ ОДНОМ МЕТОДЕ РАНЖИРОВАНИЯ РЕСУРСОВ ДЛЯ ЗАДАЧИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ С ОГРАНИЧЕННЫМИ РЕСУРСАМИ

Э. Х. Гимади^a, Е. Н. Гончаров^b

Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: ^agimadi@math.nsc.ru, ^bgon@math.nsc.ru

Аннотация. Рассматривается задача календарного планирования с ограниченными ресурсами по критерию минимума длины расписания. Все ресурсы возобновимы, прерывания операций (работ) не допускаются. Эта задача NP-трудна в сильном смысле. Ограниченность ресурсов лежит в основе сложности её решения. К числу перспективных для решения этой задачи относят эвристические алгоритмы. В данной работе предлагается ранжировать ресурсы по степени их критичности (дефицитности). Эвристические алгоритмы могут использовать это ранжирование ресурсов для построения сравнительно лучших эвристических решений. Ранжирование ресурсов получаем из релаксированной задачи, в которой ресурсы складированы. Предложенный метод ранжирования ресурсов был применён в генетическом алгоритме. Вычислительные эксперименты на примерах из электронной библиотеки PSPLIB показали его эффективность. Для нескольких примеров из PSPLIB найдены новые лучшие (ранее неизвестные) решения. Табл. 1, библиогр. 17.

Ключевые слова: задача календарного планирования с ограниченными ресурсами, возобновимые ресурсы, PSPLIB.

Введение

Рассматривается задача календарного планирования с ограниченными ресурсами (resource-constrained project scheduling problem, RCPSP). Эта задача может быть представлена следующим образом. Дан проект с множеством N операций (работ). Каждая операция j имеет длительность p_j . Прерывание операций не допускается. В связи с технологическими требованиями между некоторыми операциями существуют отношения предшествования. Они задаются наборами непосредственных предшественников P_j , указывающими, что операция j не может быть

начата до завершения каждого её предшественника из P_j . Имеется множество K типов возобновляемых ресурсов. Для каждого ресурса k в каждый единичный момент времени выделяется одинаковое количество ресурса R_k . Каждая операция j требует r_{jk} единиц ресурса типа k в каждом единичном интервале времени, в течение которого она выполняется. К числу имеющихся добавим две дополнительные операции $j = 0$ и $j = |N| + 1$, представляющие начало и завершение проекта соответственно. Обе являются «фиктивными» операциями с длительностью 0 и без запросов на ресурсы.

Предполагается, что вся информация детерминированная и известна заранее, а параметры неотрицательны и целочисленны. Расписание представляет собой назначение неотрицательного момента начала s_j каждой операции $j = 0, 1, \dots, |N| + 1$. Целью задачи RCPSP является нахождение расписания, которое приводит к кратчайшему возможному моменту завершению проекта. Время завершения проекта в расписании называется также временем выполнения расписания.

По классификации, предложенной в [1], представленная выше задача обозначается через $PS|prec|C_{\max}$, а согласно классификационной схеме из [2] — через $m, 1|cpm|C_{\max}$.

Поскольку задача RCPSP NP-трудна в сильном смысле [3], известные точные методы решения, такие как метод ветвей и границ, не подходят для решения крупномасштабных задач из-за значительного времени вычислений, поэтому точные методы неприемлемы для решения практических задач. Для решения примеров RCPSP большого размера основной интерес для исследований представляют эвристические процедуры и метаэвристики. Многие эвристические процедуры основаны на правилах приоритета и методах forward-backward improvement (FBI). К числу метаэвристических алгоритмов, предложенных для RCPSP, могут быть отнесены генетический алгоритм (ГА), поиск с запретами, имитация отжига, оптимизация муравьиной колонии, перелинковка путей (the path re-linking) и разнообразные гибридные алгоритмы. Численные эксперименты, проведённые на стандартных наборах примеров PSPLIB [4], показали высокую эффективность этих метаэвристических методов.

Задача RCPSP имеет обширное применение на практике, поэтому она привлекала к себе внимание многих исследователей на протяжении нескольких последних десятилетий. В прошлом опубликован ряд обзоров, связанных с RCPSP. Детерминированные варианты задачи и методы их решения были рассмотрены в [1]. Перечисление детерминированных вариантов и расширение базовой модели приведено в [5], но без методов решения. Эвристические методы для базового варианта рассмотрены в [6]. Обзор робастных и реактивных процедур планирования дан

в [7]. Обзор по задачам планирования проектов с ограниченными ресурсами в условиях неопределённости представлен в [8], а обстоятельный обзор гибридных метаэвристик для RCPSP — в [9]. Упомянем также обзоры [10–12].

Как утверждается в [13], решение ресурсных конфликтов лежит в основе задачи RCPSP, и многие исследователи также отмечали, что наличие ограниченных ресурсов является определяющим фактором для сложности задачи. В работе [13] авторы анализировали сложность задачи RCPSP с экспериментальной точки зрения и предложили подход к преобразованию простых задач RCPSP в очень сложные. При этом количество операций и ресурсов в таких примерах должно быть как можно меньше. Одним из шагов указанного преобразования является сокращение количества ресурсов, необходимое для поиска узких мест, определяющих сложность задачи.

Поиск узких мест среди ограниченных ресурсов являлся главной темой в работе [14]. В отличие от [13], где обнаруженные ресурсные узкие места использовались для построения трудных примеров, в [14] производилось ранжирование ресурсов по степени их дефицитности. Оба эти подхода в своей основе используют поиск узких мест задачи по использованию ресурсов, и в этом они перекликаются. Полученное ранжирование применялось далее для построения эвристического приближённого решения жадным алгоритмом. На каждом шаге этого алгоритма учитывалось применение более приоритетных ресурсов и рассматривались в первую очередь использующие их операции. В генетическом алгоритме [15] использовался оператор скрещивания родительских хромосом, а для выявления перспективных генов в этих хромосомах также применялся этот способ ранжирования ресурсов. Численные эксперименты показали эффективность генетического алгоритма с применением ранжирования ресурсов по степени их дефицитности.

В настоящей статье предлагается новый способ ранжирования ресурсов по степени их дефицитности. Для этого рассматривается релаксированная задача RCPSP со складываемыми (cumulative) ресурсами. Эта задача полиномиально разрешима [16]. В [17] был предложен модифицированный алгоритм задачи с улучшенной оценкой временной сложности. После нахождения решения релаксированной задачи мы проведём детальный анализ динамики потребления ресурсов, сравнивая её с динамикой выделения этих ресурсов. Таким образом фиксируются узкие места задачи, связанные с ресурсами, и выявляются ресурсы в этой релаксированной задаче, критически важные в каждый конкретный момент времени. В результате мы ранжируем ресурсы по степени их дефицитности, а затем делаем предположение, что такое ранжирование ресурсов

для релаксированной задачи будет также сохраняться и для исходной задачи.

Фактически предлагаемый метод ранжирования ресурсов развивает идеи из [15]. Если ранее обращалось внимание лишь на интегральные остатки неиспользованных ресурсов по окончании решения релаксированной задачи, то теперь изучается характер потребления ресурсов в динамике, что позволяет более аккуратно выявлять ресурсные коллизии.

Качество предложенного метода ранжирования ресурсов для RCPSP протестировано на серии численных экспериментов. Для их проведения использован генетический алгоритм [15], в операции кроссовера которого задействовано эвристическое правило ранжирования ресурсов. Тестовые примеры для эксперимента взяты из электронной библиотеки тестовых примеров PSPLIB [4]. Для тестирования использованы примеры из тестовых множеств j90 и j120 размерностью 90 и 120 операций соответственно.

Ниже показано, что с помощью предложенного метода ранжирования ресурсов получены лучшие решения в сравнении с алгоритмами, не использующими ранжирование ресурсов, а также использующими применённый ранее [14] интегральный способ ранжирования ресурсов. Генетический алгоритм с предложенным правилом ранжирования ресурсов получил лучшие решения для нескольких десятков примеров из j90 и j120. Для одного примера из j90 и четырёх примеров из j120 найдены лучшие (ранее неизвестные) эвристические решения. Эти решения внесены в таблицу лучших эвристических решений, расположенную на сайте PSPLIB.

1. Постановка задачи

Задачу RCPSP можно сформулировать следующим образом. Рассмотрим проект как ориентированный ациклический граф $G = (N, A)$, в котором N — множество операций проекта, а $A \subseteq N \times N$ — множество пар, отражающее отношение предшествования между операциями. Положим $N = \{1, \dots, n\} \cup \{0, n+1\}$, при этом фиктивные операции 0 и $n+1$ задают начало и завершение всего комплекса операций. Операции $1, \dots, n$ будем называть фактическими. Отношение предшествования на N определим множеством пар $A = \{(i, j) \mid i \in P_j\}$. Если $(i, j) \in A$, то операция j не может начаться раньше завершения операции i .

Через K обозначим множество типов ресурсов, необходимых для выполнения операций. Все ресурсы предполагаются возобновимыми. В каждый единичный временной интервал $[t-1, t)$, $t \in \mathbb{Z}_+$, для каждого типа ресурсов выделяется фиксированный объём этого ресурса $R_k \in \mathbb{Z}_+$.

Фактическая операция j имеет определённую длительность своего выполнения $p_j \in \mathbb{Z}_+$ и требует в каждую единицу $[t-1, t)$, $t \in \mathbb{Z}_+$, времени своего выполнения $r_{jk} \geq 0$ единиц ресурса типа $k \in K$. С целью избежать неразрешимости задачи будем предполагать, что $r_{jk} \leq R_k$, $j \in N$, $k \in K$.

Фиктивные операции 0 и $n + 1$ имеют нулевую длительность и нулевое потребление ресурсов. Фактические операции имеют ненулевую целочисленную длительность.

Введём переменные задачи. Через $s_j \geq 0$ обозначим момент начала выполнения j -й операции. Так как операции выполняются без прерывания, момент окончания j -й операции определяется равенством $c_j = s_j + p_j$. Определим расписание S как $(n + 2)$ -вектор $(s_0, \dots, s_{n+1})$. Время завершения проекта $T(S)$ соответствует моменту завершения последней операции проекта, т. е. $T(S) = c_{n+1}$. Через $J(t, S) = \{j \in N \mid s_j \leq t < c_j\}$ обозначим множество операций, выполняемых в единичном интервале времени $[t - 1, t)$, $t \in \mathbb{Z}_+$ при расписании S . Задача заключается в нахождении допустимого расписания $S = \{s_j\}$ с минимальным временем завершения проекта $T(S)$. Её можно формализовать следующим образом: минимизировать время завершения проекта

$$T(S) = c_{n+1} \quad (1)$$

при ограничениях

$$s_i + p_i \leq s_j, \quad (i, j) \in A, \quad (2)$$

$$\sum_{j \in J(t, S)} r_{jk} \leq R_k, \quad k \in K, t \in \mathbb{Z}_+ : J(t, S) \neq \emptyset, \quad (3)$$

$$s_j \in \mathbb{Z}_+, \quad j \in N. \quad (4)$$

Неравенства (2) определяют ограничения предшествования операций, а (3) обеспечивают соблюдение ресурсных ограничений с возобновимыми ресурсами: суммарное количество ресурса типа k , потребляемое всеми операциями, выполняемыми в единичном интервале времени $[t - 1, t)$, не должно превышать имеющегося в наличии количества этого ресурса в данный момент времени. Наконец, (4) определяет искомые переменные.

2. Ранжирование ресурсов

Идея нахождения относительно более перспективных ресурсов (их ранжирования по степени дефицитности) состоит в использовании релаксированной задачи для (1)–(4). Релаксация состоит в том, что теперь все ресурсы будем считать складываемыми (cumulative). По решению релаксированной задачи определяем для неё ранжирование ресурсов. Далее делаем предположение, что ранжирование ресурсов для релаксированной задачи будет также сохраняться и для исходной задачи.

Подобная идея была реализована в [14, 15]. В отличие от предлагаемого в данной работе метода ранжирования, там применялся интегральный способ определения дефицитных ресурсов, исходя из общих остатков неиспользованных ресурсов. Этот метод показал свою эффективность

при проведении численных экспериментов. Однако, он имеет и слабые стороны. Вполне возможно, что некий ресурс интегрально не проявил себя критически дефицитным, но на локальных отрезках времени именно он оказался тем узким местом, которое не позволило сократить общую длину расписания. Новый подход рассматривает процесс построения расписания в динамике, и тем самым он лишён этого недостатка.

Для задачи (1)–(4) рассмотрим её релаксацию, в которой условие возобновимости ресурсов ослаблено до условия их складировемости: минимизировать время завершения проекта

$$T(S) = c_{n+1} \quad (5)$$

при ограничениях

$$s_i + p_i \leq s_j, \quad (i, j) \in A, \quad (6)$$

$$\sum_{t'=1}^t \sum_{j \in J(t', S)} r_{jk} \leq \sum_{t'=1}^t R_k, \quad k \in K, t \in \mathbb{Z}_+: J(t, S) \neq \emptyset, \quad (7)$$

$$s_j \in \mathbb{Z}_+, \quad j \in N. \quad (8)$$

Неравенства (7) обеспечивают соблюдение ограничений для складировемых ресурсов, а соотношения (5)–(8) представляют собой задачу календарного планирования со складировемыми ресурсами.

Отметим, что задача RCPSP с ограничениями на ресурсы складировемого типа (7) по трудоёмкости решения отличается от задачи с ограничениями с возобновимыми ресурсами (3). Если все ресурсы только складировемого типа, то для довольно широкого класса задач (с целочисленными длительностями операций) задача полиномиально разрешима [16]. В [17] был предложен модифицированный алгоритм для релаксированной задачи (5)–(8) с улучшенной оценкой временной сложности, равной $O(|A| + n \log_2 n + n|K|)$, где n — число операций, K — множество типов ресурсов, $|A|$ — число пар операций, между которыми установлены отношения предшествования. Применительно к рассматриваемой постановке задачи мощность множества A имеет оценку $O(n^2)$, и тогда оценку временной сложности задачи (5)–(8) можно представить как $O(n(n + |K|))$.

Опишем алгоритм ранжирования ресурсов для релаксированной задачи (5)–(8).

ШАГ 1. Решаем задачу (5)–(8). В результате решения получаем оптимальную длину расписания T^* и моменты начала выполнения операций в оптимальном расписании $S^* = \{s_j^*\}$.

ШАГ 2. Для оптимального расписания S^* находим остатки неиспользованных ресурсов в каждый момент времени:

$$\rho_{kt} = \sum_{t'=1}^t R_k - \sum_{t'=1}^t \sum_{j \in J^*(t', S^*)} r_{jk}, \quad k \in K, t \in \overline{1, T^*}.$$

Здесь $J^*(t, S^*)$ — множество операций, выполняемых в единичном интервале времени $[t-1, t)$ при расписании S^* :

$$J^*(t, S^*) = \{j \in N \mid s_j^* \leq t < s_j^* + p_j\}.$$

ШАГ 3. Для каждого ресурса k вычисляем минимальную по всем моментам времени t величину $\rho_k^{\min}$ минимальных остатков неизрасходованных ресурсов:

$$\rho_k^{\min} = \min_{t \in \overline{1, T^*}} \rho_{kt}, \quad k \in K.$$

ШАГ 4. Сортируем по неубыванию $\rho_k^{\min}$, $k \in K$. Результат сортировки является искомым ранжированием ресурсов: ресурс с минимальным значением $\rho_k^{\min}$ будем считать наиболее дефицитным, а с максимальным значением — наименее дефицитным. В случае равенства минимальных остатков более дефицитным будем считать ресурс с меньшим интегральным значением неизрасходованных ресурсов ρ_{kT^*} .

Заметим, что значение величин ρ_{kt} получается в ходе работы алгоритма на шаге 1 в виде разности двух следующих величин: суммарного количества выделенных и суммарного количества потреблённых ресурсов. Это же замечание справедливо и для шага 3 алгоритма: минимальное количество неиспользованных ресурсов для всех моментов времени также может быть получено в ходе решения релаксированной задачи. Наконец, само ранжирование ресурсов на шаге 4 требует сортировки массива $\rho_k^{\min}$, $k \in K$, по неубыванию, оно выполняется за $O(|K| \log |K|)$ действий. Следовательно, общая временная сложность алгоритма равна $O(n(n + |K|) + |K| \log |K|)$.

3. Численные эксперименты

Для исследования эффективности предложенного метода ранжирования ресурсов использованы наборы тестовых примеров из электронной библиотеки PSPLIB [4], доступной по ссылке www.om-db.wi.tum.de/psplib. Мы использовали множества тестовых примеров j60, j90 и j120. Наборы j60 и j90 содержат примеры с 60 и 90 операциями в каждом соответственно, по 48 серий примеров в каждом наборе и по 10 экземпляров в каждой серии — всего по 480 примеров в каждом из этих наборов. Набор данных j120 содержит 60 серий тестовых примеров по 10 экземпляров в каждой серии — всего 600 экземпляров.

Пусть R_{din} — массив ресурсов в порядке их ранжирования предложенным в данной работе динамическим методом, а R_{int} — соответствующий массив ранжирования, полученный интегральным методом. Понятно, что на некоторых примерах ранжирование может совпадать, и в таких случаях у нас не будет предпосылок для обнаружения новых лучших решений за счёт более аккуратного ранжирования. В первом численном эксперименте находим на примерах из PSPLIB оба ранжирования и производим их сравнение по числу возможных совпадений.

Таблица 1

Сравнение ранжирований R_{din} и R_{int} на примерах из PSPLIB

Набор примеров из PSPLIB	Общее число примеров	Число примеров с $R_{\text{din}} = R_{\text{int}}$	Процент совпадений R_{din} и R_{int}	Число примеров с $R_{\text{din}} \neq R_{\text{int}}$	Процент несовпадений R_{din} и R_{int}
j60	480	116	24,2	364	75,8
j90	480	137	28,5	343	71,5
j120	600	152	25,3	448	74,5

Как видно из табл. 1, около трёх четвертей всех примеров из PSPLIB имеют другое, более тонко настроенное ранжирование ресурсов.

В следующем численном эксперименте исследуем эффект влияния предложенного метода ранжирования ресурсов на решения, получаемые использующим его метаэвристическим алгоритмом, в качестве которого применялся генетический алгоритм [15]. Были использованы примеры из наборов j90 и j120 электронной библиотеки PSPLIB.

Оператор кроссовера этого алгоритма действует по следующей схеме. В обоих родительских хромосомах находятся перспективные участки (гены) по принципу наилучшего использования ресурсов. Под наилучшим использованием ресурсов понимается минимальный объём неиспользованных ресурсов (потерь) среди возобновимых ресурсов. Затем эти перспективные гены передаются дочерней хромосоме. Теперь, имея ранжирование ресурсов по степени их дефицитности, стремимся передать дочерней хромосоме те гены, которые по возможности используют наилучшим способом именно более дефицитные ресурсы. Генетический алгоритм совершил две серии экспериментов по числу шагов алгоритма: 50 000 шагов и 500 000 шагов.

Численные эксперименты показали, что на примерах из набора j90 генетический алгоритм при 50 000 шагов отыскал решения с меньшей длиной расписания для 46 примеров. На примерах из j120 удалось найти 63 лучших решений. При 500 000 шагах алгоритма было найдено 15 и 29 лучших решений соответственно.

Для одного примера из j90 (пример j9013_3) удалось получить новое рекордное (неизвестное ранее) эвристическое решение. Для множества примеров j120 были обновлены решения для четырёх примеров: X11_5, X16_2, X16_3 и X51_1. Эти обновлённые решения размещены в таблице лучших эвристических решений на сайте PSPLIB.

Заключение

Для задачи календарного планирования с ограниченными ресурсами по критерию минимизации длины расписания предложен новый метод ранжирования ресурсов по степени их дефицитности. Наличие такого ранжирования ресурсов позволяет эвристическим алгоритмам и метаэвристикам находить сравнительно лучшие эвристические решения. Для определения более дефицитных ресурсов рассматривается релаксированная задача RCPSP со складываемыми ресурсами. Эта задача полиномиально разрешима. Получив решение релаксированной задачи, мы производим анализ динамики потребления ресурсов, фиксируем узкие места задачи, связанные с ресурсами, и ранжируем ресурсы по степени их дефицитности. Затем делается предположение, что такое ранжирование ресурсов для релаксированной задачи будет также сохраняться и для исходной задачи. Этот метод опробован в рамках численных экспериментов. Для проведения экспериментов использован генетический алгоритм, в операции кроссовера которого задействовано разработанное эвристическое правило ранжирования ресурсов. Тестовые примеры взяты из электронной библиотеки тестовых примеров PSPLIB [4]. Численные эксперименты показали эффективность предложенного метода.

Финансирование работы

Работа выполнена в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF-2026-0021). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Brucker P., Drexl A., Möhring R., Neumann K., Pesch E. Resource-constrained project scheduling: Notation, classification, models, and methods // Eur. J. Oper. Res. 1999. V. 112, No. 1. P. 3–41.
2. Herroelen W., Demeulemeester E., De Reyck B. A classification scheme for project scheduling // Project scheduling: Recent models, algorithms and applications. Dordrecht: Kluwer Acad. Publ., 1999. P. 1–26.

3. **Błażewicz J., Lenstra J. K., Rinnoy Kan A. H. G.** Scheduling subject to resource constraints: Classification and complexity // *Discrete Appl. Math.* 1983. V. 5, No. 1. P. 11–24.
4. **Kolisch R., Sprecher A.** PSPLIB — A project scheduling problem library // *Eur. J. Oper. Res.* 1996. V. 96. P. 205–216.
5. **Hartmann S., Briskorn D.** A survey of variants and extensions of the resource-constrained project scheduling problem // *Eur. J. Oper. Res.* 2010. V. 207. P. 1–14.
6. **Kolisch R., Hartmann S.** Experimental investigation of heuristics for resource-constrained project scheduling: An update // *Eur. J. Oper. Res.* 2006. V. 174, No. 1. P. 23–37.
7. **Herroelen W., Leus R.** Robust and reactive project scheduling: A review and classification of procedures // *Int. J. Prod. Res.* 2004. V. 42, No. 8. P. 1599–1620.
8. **Golab A., Sedgh Gooya E., Al Falou A., Cabon M.** Review of conventional metaheuristic techniques for resource-constrained project scheduling problem // *J. Proj. Manag.* 2022. V. 7, No. 2. P. 95–110. DOI: 10.5267/j.jpm.2021.10.002.
9. **Pellerin R., Perrier N., Berthaut F.** A survey of hybrid metaheuristics for the resource-constrained project scheduling problem // *Eur. J. Oper. Res.* 2020. V. 280, No. 2. P. 395–416. DOI: 10.1016/j.ejor.2019.01.063.
10. **Abdolshah M.** A review of resource-constrained project scheduling problems (RCPSP) approaches and solutions // *Int. Trans. J. Eng. Manag. Appl. Sci. Technol.* 2014. V. 5, No. 4. P. 253–286.
11. **Hartmann S., Briskorn D.** An updated survey of variants and extensions of the resource-constrained project scheduling problem // *Eur. J. Oper. Res.* 2022. V. 297, No. 1. P. 1–14. DOI: 10.1016/j.ejor.2021.05.004.
12. **Khajesaedi S., Sadjadi S., Barzinpour F., Tavakkoli-Moghaddam R.** Resource-constrained project scheduling problem: Review of recent developments // *J. Proj. Manag.* 2025. V. 10, No. 1. P. 1–26. DOI: 10.5267/j.jpm.2024.12.002.
13. **Coelho J., Vanhoucke M.** Going to the core of hard resource-constrained project scheduling instances // *Comput. Oper. Res.* 2020. V. 121. Article ID 104976. 13 p. DOI: 10.1016/j.cor.2020.104976.
14. **Гончаров Е. Н.** Жадный алгоритм для задачи календарного планирования с ограниченными ресурсами // *Дискрет. анализ и исслед. операций.* 2024. Т. 31, № 4. С. 27–39. DOI: 10.33048/daio.2024.31.796.
15. **Гончаров Е. Н., Леонов В. В.** Генетический алгоритм для задачи календарного планирования с ограниченными ресурсами // *Автоматика и телемеханика.* 2017. № 6. С. 173–189.
16. **Гимади Э. Х., Залюбовский В. В., Севастьянов С. В.** Полиномиальная разрешимость задач календарного планирования со складываемыми ресурсами и директивными сроками // *Дискрет. анализ и исслед. операций.* Сер. 2. 2000. Т. 7, № 1. С. 9–34.

17. Гимади Э. Х., Гончаров Е. Н., Штепа А. А. Быстрый алгоритм вычисления нижней оценки для решения задачи ресурсно-календарного планирования с тестированием на примерах библиотеки PSPLIB // Тр. Ин-та математики и механики. 2021. Т. 27, № 1. С. 22–36. DOI: 10.21538/0134-4889-2021-27-1-22-36.

Гимади Эдуард Хайрутдинович
Гончаров Евгений Николаевич

Статья поступила
8 апреля 2026 г.
После доработки —
15 апреля 2026 г.
Принята к публикации
22 апреля 2026 г.

ON A METHOD FOR RANKING RESOURCES
FOR THE RESOURCE-CONSTRAINED PROJECT
SCHEDULING PROBLEMÉ. Kh. Gimadi^a and E. N. Goncharov^bSobolev Institute of Mathematics,
4 Acad. Koptuyug Avenue, 630090 Novosibirsk, Russia
E-mail: ^agimadi@math.nsc.ru, ^bgon@math.nsc.ru

Abstract. We consider a resource-constrained project scheduling problem (RCPSP) with respect to the makespan minimization criterion. The problem accounts for technological constraints on activities precedence together with resource constraints. Activities preemptions are not allowed. The problem with renewable resources is NP-hard in the strong sense. Metaheuristics are one of the promising methods for solving this problem. We propose a new method for identifying the most scarce resources, which helps metaheuristics to overcome these resource conflicts and find better solutions. The proposed resource ranking method is applied in a genetic algorithm. Computational experiments with instances from the PSPLIB electronic library showed its effectiveness. New solutions are obtained for several instances, while new best (previously unknown) solutions are found for one instance from j90 and four instances from j120 datasets. Tab. 1, bibliogr. 17.

Keywords: project management, resource-constrained project scheduling problem, renewable resources, PSPLIB.

References

1. P. Brucker, A. Drexler, R. Möhring, K. Neumann, and E. Pesch, Resource-constrained project scheduling: Notation, classification, models, and methods, *Eur. J. Oper. Res.* **112** (1), 3–41 (1999).
2. W. Herroelen, E. Demeulemeester, and B. De Reyck, A classification scheme for project scheduling, in *Project Scheduling: Recent Models, Algorithms and Applications* (Kluwer Acad. Publ., Dordrecht, 1999), pp. 1–26.

3. **J. Błażewicz, J. K. Lenstra, and A. H. G. Rinnoy Kan**, Scheduling subject to resource constraints: Classification and complexity, *Discrete Appl. Math.* **5** (1), 11–24 (1983).
4. **R. Kolisch and A. Sprecher**, PSPLIB — A project scheduling problem library, *Eur. J. Oper. Res.* **96**, 205–216 (1996).
5. **S. Hartmann and D. Briskorn**, A survey of variants and extensions of the resource-constrained project scheduling problem, *Eur. J. Oper. Res.* **207**, 1–14 (2010).
6. **R. Kolisch and S. Hartmann**, Experimental investigation of heuristics for resource-constrained project scheduling: An update, *Eur. J. Oper. Res.* **174** (1), 23–37 (2006).
7. **W. Herroelen and R. Leus**, Robust and reactive project scheduling: A review and classification of procedures, *Int. J. Prod. Res.* **42** (8), 1599–1620 (2004).
8. **A. Golab, E. Sedgh Gooya, A. Al Falou, and M. Cabon**, Review of conventional metaheuristic techniques for resource-constrained project scheduling problem, *J. Proj. Manag.* **7** (2), 95–110 (2022), DOI: 10.5267/j.jpm.2021.10.002.
9. **R. Pellerin, N. Perrier, and F. Berthaut**, A survey of hybrid metaheuristics for the resource-constrained project scheduling problem, *Eur. J. Oper. Res.* **280** (2), 395–416 (2020), DOI: 10.1016/j.ejor.2019.01.063.
10. **M. Abdolshah**, A review of resource-constrained project scheduling problems (RCPSP) approaches and solutions, *Int. Trans. J. Eng. Manag. Appl. Sci. Technol.* **5** (4), 253–286 (2014).
11. **S. Hartmann and D. Briskorn**, An updated survey of variants and extensions of the resource-constrained project scheduling problem, *Eur. J. Oper. Res.* **297** (1), 1–14 (2022), DOI: 10.1016/j.ejor.2021.05.004.
12. **S. Khajesaedi, S. Sadjadi, F. Barzinpour, and R. Tavakkoli-Moghaddam**, Resource-constrained project scheduling problem: Review of recent developments, *J. Proj. Manag.* **10** (1), 1–26 (2025), DOI: 10.5267/j.jpm.2024.12.002.
13. **J. Coelho and M. Vanhoucke**, Going to the core of hard resource-constrained project scheduling instances, *Comput. Oper. Res.* **121**, ID 104976 (2020), DOI: 10.1016/j.cor.2020.104976.
14. **E. N. Goncharov**, A greedy algorithm for the resource-constrained project scheduling problem, *Diskretn. Anal. Issled. Oper.* **31** (4), 27–39 (2024) [Russian], DOI: 10.33048/daio.2024.31.796 [*J. Appl. Ind. Math.* **18** (4), 679–685 (2024)].
15. **E. N. Goncharov and V. V. Leonov**, A genetic algorithm for the resource-constrained project scheduling problem, *Avtom. Telemekh.*, No. 6, 173–189 (2017) [Russian] [*Autom. Remote Control* **78** (6), 1101–1114 (2017)].
16. **É. Kh. Gimadi, V. V. Zalyubovskiy, and S. V. Sevast'yanov**, Polynomial solvability of scheduling problems with storable resources and directive deadlines, *Diskretn. Anal. Issled. Oper., Ser. 2*, **7** (1), 9–34 (2000) [Russian].

17. **É. Kh. Gimadi, E. N. Goncharov, and A. A. Shtepa**, A fast algorithm for finding a lower bound of the solution to the resource-constrained project scheduling problem tested on PSPLIB instances, *Tr. Inst. Mat. Mekh.* **27** (1), 22–36 (2021), DOI: 10.21538/0134-4889-2021-27-1-22-36 [Russian].

Éduard Kh. Gimadi
Evgeny N. Goncharov

Received April 8, 2026
Revised April 15, 2026
Accepted April 22, 2026