ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 31 № 4 2024

Новосибирск
Издательство Института математики

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2024. Т. 31, № 4. С. 27–39

УДК 519.8+518.25 DOI: 10.33048/daio.2024.31.796

ЖАДНЫЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ С ОГРАНИЧЕННЫМИ РЕСУРСАМИ

Е. Н. Гончаров

Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия

E-mail: gon@math.nsc.ru

Аннотация. Задача календарного планирования с ограниченными ресурсами — достаточно общая задача теории расписаний, которая включает в себя ограничения предшествования работ и ресурсные ограничения. Все ресурсы возобновимы, прерывания работ запрещены. Эта задача NP-трудна в сильном смысле. Мы предлагаем новый детерминированный жадный алгоритм. Он основан на эвристике, использующей информацию, полученную в результате решения релаксированной задачи с кумулятивными ресурсами. Алгоритм протестирован на стандартных наборах данных j60, j90 и j120, предоставляемых библиотекой PSPLIB; показана его эффективность. Табл. 2, библиогр. 17.

Ключевые слова: задача календарного планирования с ограниченными ресурсами, управление проектами, возобновимые ресурсы, жадный алгоритм, PSPLIB.

Введение

Рассматривается задача календарного планирования с ограниченными ресурсами (resource-constrained project scheduling problem, RCPSP). Её можно определить как задачу комбинаторной оптимизации. Дано множество работ, частичный порядок на этом множестве задаётся ациклическим ориентированным графом. Для каждой работы известны длительность её выполнения, множество потребляемых ею ресурсов и их объёмы. Считаем, что в каждый единичный интервал горизонта планирования $\widehat{T}$ для каждого ресурса выделяется одинаковое его количество. Предполагается, что ресурсы неограниченны за пределами горизонта проекта $\widehat{T}$. Все ресурсы возобновимы. Прерывания работ не допускаются. Задача состоит в минимизации общего времени выполнения

проекта с учётом отношения предшествования работ и выполнения ресурсных ограничений. По классификации, предложенной в [1], эта задача обозначается через $PS \mid prec \mid C_{\max}$, а согласно классификационной схеме из [2] — через $m, 1 \mid cpm \mid C_{\max}$.

RCPSP принадлежит к классу NP-трудных задач [3]. Она имеет обширное применение на практике, поэтому привлекает к себе внимание многих исследователей на протяжении нескольких последних десятилетий. Об этом свидетельствуют многочисленные обзоры по методам её решения, например [4–8].

Эвристики, основанные на правилах приоритета (priority rule based heuristics), являются сравнительно хорошо исследованными методами решения RCPSP. Они относятся к так называемым методам построения. Считается, что более глубокие методы, такие как алгоритмы локального поиска, генетические алгоритмы и др., обеспечивают в большинстве случаев решения лучше, чем методы построения, поскольку они выполняют процедуру поиска, стартуя от некоторого допустимого расписания, сгенерированного одним или несколькими методами построения. Сильная сторона методов построения состоит в том, что они позволяют получать сравнительно хорошие решения с небольшой временной сложностью.

Правило приоритета — это функция, которая присваивает каждой работе j некоторое значение $v(j)$. В случае ничьей используется правило разрешения ничьей. Правила приоритета можно классифицировать по разным критериям. В зависимости от типа информации, используемой для расчёта $v(j)$, можно различать правила, основанные на сети, времени и ресурсах, а также их комбинациях. Правило, основанное на сети, использует множество всех (прямых и косвенных) последователей работы j, а основанное на времени правило использует время начала выполнения работы j в некотором оценочном расписании. В [9] была рассмотрена эвристика, которая использует наиболее поздние моменты старта работ в задаче без ресурсных ограничений (эта эвристика была рассмотрена в совокупности с дополнительными правилами). В [10] при построении стохастического жадного алгоритма в качестве $v(j)$ использовались моменты начала выполнения работы j в оценочном расписании, являющемся решением релаксированной задачи, в которой ограниченные ресурсы складируемы (кумулятивны). Для разрешения ничьей было использовано дополнительное ресурсное правило.

Ресурсное правило ориентировано на величину потребления ресурсов: приоритет отдаётся работам, наиболее полно расходующим выделенные ресурсы. Подробная классификация известных в литературе правил приоритета приведена в обзоре [4].

В этой статье предлагается новое правило приоритета, представляющее собой ресурсное правило, в котором $v(j)$ находится с учётом весовых

коэффициентов ресурсов. Эти коэффициенты вычисляются из решения вспомогательной релаксированной задачи, в которой ресурсы складируемы. В релаксированной задаче все ресурсы ранжируются по степени дефицитности, и тем самым определяются более дефицитные из них. После этого делается предположение, что ранжирование ресурсов, выполненное для релаксированной задачи, будет также справедливым и для исходной задачи. Приписываем для каждого типа ресурса вес в соответствии с его рангом и находим $v(j)$ уже в соответствии со взвешенными значениями потребляемых этой работой ресурсов.

Качество алгоритма проверено на тестовых примерах из библиотеки PSPLIB [11]. Для тестирования взяты примеры из j60, j90 и j120 размерности 60, 90 и 120 работ соответственно. Далее приводятся результаты численных экспериментов. Показано, что с использованием предложенного алгоритма получаются лучшие значения средних отклонений решений от величины критического пути в сравнении с алгоритмами, использующими правила, основанные на свойствах сети, на времени и неранжированных ресурсах. Выявлены группы примеров из PSPLIB, на которых предложенный алгоритм имеет особо значительное преимущество.

1. Постановка задачи

Задача RCPSP может быть сформулирована следующим образом. Будем рассматривать проект как ориентированный ацикличный граф $G = (N, A)$, где N — множество работ проекта, а A — множество пар работ, между которыми установлено отношение предшествования. Обозначим через $N = \{1, \ldots, n\} \cup \{0, n+1\}$ множество работ, в котором работы 0 и $n+1$ фиктивны: они задают начало и завершение всего комплекса работ. Все другие работы будем называть фактическими. Отношение предшествования на множестве N зададим множеством пар $A = \{(i, j) \in N^2 \mid i \text{ — предшественник } j\}$. Если $(i, j) \in A$, то работа j не может начаться раньше завершения работы i. В множестве A содержатся все пары $(0, j)$ и $(j, n+1)$, $j = 1, \ldots, n$.

Через K обозначим множество типов ресурсов, необходимых для выполнения работ. Горизонт планирования $\widehat{T}$ делится на временные интервалы равной длины $[t-1, t)$, $t = 1, \ldots, \widehat{T}$. Все ресурсы предполагаются возобновимыми, т. е. в каждый единичный временной интервал горизонта планирования $\widehat{T}$ для каждого типа ресурсов выделяется фиксированный объём этого ресурса $R_k \in Z^+$.

Фактическая работа j имеет детерминированную длительность своего выполнения $d_j \in Z^+$ и требует в каждую единицу времени своего выполнения $r_{jk} \geqslant 0$ единиц ресурса типа $k \in K$. Чтобы избежать неразрешимость задачи будем считать, что $r_{jk} \leqslant R_k$, $j \in N$, $k \in K$. Фиктивные

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

Введём переменные задачи. Через $s_j \geqslant 0$ обозначим момент начала выполнения j-й работы. Так как работы выполняются без прерывания, момент окончания j-й работы определяется равенством $c_j = s_j + d_j$. Определим расписание S как $(n+2)$-вектор $(s_0, \ldots, s_{n+1})$. Время завершения проекта $T(S)$ соответствует моменту завершения последней работы проекта, т. е. $T(S) = c_{n+1}$. Через $J(t) = \{j \in N \mid s_j \leqslant t < c_j\}$ обозначим множество работ, выполняемых в единичном интервале времени $[t-1, t)$ 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

$$T(S) = \max_{j \in N}(s_j + p_j) \tag{1}$$

при ограничениях

$$s_i + p_i \leqslant s_j, \quad i, j \in A, \tag{2}$$

$$\sum_{j \in J(t)} r_{jk} \leqslant R_k, \quad k \in K,\ t = 1, \ldots, \widehat{T}, \tag{3}$$

$$s_j \in \mathbf{Z}^+, \quad j \in N. \tag{4}$$

Неравенства (2) определяют ограничения предшествования работ. Соотношение (3) обеспечивает соблюдение ресурсных ограничений с возобновимыми ресурсами: суммарное количество ресурса типа k, потребляемое всеми работами, 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.

Рассмотрим также релаксированную задачу для (1)–(4), в которой условие возобновимости ресурсов ослаблено до условия их складируемости: минимизировать время завершения проекта

$$T(S) = \max_{j \in N}(s_j + p_j) \tag{5}$$

при ограничениях

$$s_i + p_i \leqslant s_j, \quad i, j \in A, \tag{6}$$

$$\sum_{t'=1}^{t} \sum_{j \in J(t')} r_{jk} \leqslant \sum_{t'=1}^{t} R_k, \quad k \in K,\ t = 1, \ldots, \widehat{T}, \tag{7}$$

$$s_j \in \mathbf{Z}^+, \quad j \in N. \tag{8}$$

Соотношения (7) обеспечивают соблюдение ресурсных ограничений со складируемыми ресурсами, а задача (5)–(8) является задачей календарного планирования со складируемыми (кумулятивными) ресурсами.

Отметим, что RCPSP с ограничениями на ресурсы складируемого типа (7) по трудоёмкости решения отличается от задачи с ограничениями с возобновимыми ресурсами (3). Если все ресурсы только складируемого типа, то для довольно широкого класса задач (с целочисленными длительностями работ) задача полиномиально разрешима [12].

Помимо полиномиального алгоритма [12] для решения RCPSP известен приближённый асимптотически точный алгоритм, который применим для более общей постановки задачи. В этой постановке ресурсные ограничения только складируемого типа, моменты начала выполнения работ могут быть неотрицательными вещественными числами, а также могут учитываться директивные сроки выполнения событий. Время работы этого алгоритма есть функция от числа работ N порядка $O(N^2)$, а относительная и абсолютная погрешности стремятся к нулю с ростом размерности задачи [13, 14]. Для приближённого решения задачи (5)–(8) в этой статье будем использовать именно этот алгоритм.

2. Алгоритм решения задачи

Для построения эвристического решения задачи RCPSP будем использовать известную последовательную схему генерации расписаний (serial schedule generation scheme, или serial SGS) [4].

Алгоритм S-SGS состоит в точности из N шагов, на каждом из которых выбирается ровно одна работа из числа нерассмотренных, и для неё определяется (назначается) минимальное время её старта, при этом соблюдаются ограничения предшествования работ и ресурсные ограничения. На каждом шаге $g = 1, \ldots, N$ известны множество уже рассмотренных работ S_g и D_g — подмножество множества не рассмотренных работ таких, что все непосредственно предшествующие им работы принадлежат множеству S_g. Руководствуясь некоторым правилом выбора $v(j)$, выбираем работу j из множества D_g и назначаем минимальное время начала её выполнения такое, что выполнены ограничения предшествования работ и ресурсные ограничения.

Порядок выбора работ играет определяющую роль при построении жадного расписания. Правило выбора наиболее приоритетной работы из множества D_g будем определять, исходя из минимального значения функции приоритетов (весовой функции) $v(j)$, $j \in D_g$. В [4] приведён обзор описанных в литературе функций приоритетов. Они могут использовать, например, структуру сети, время свершения событий, интенсивность потребления ресурсов и т. п.

Основное новшество алгоритма заключается в применяемом правиле нахождения значения $v(j)$. Это правило базируется на интенсивности потребления работой $v(j)$ ресурсов, а также на степени относительной дефицитности ресурсов. Представляем степень дефицитности ресурсов в виде весовой функции ресурсов w_k, $k \in K$. Значения w_k назначаем некоторым образом (см. разд. 3), исходя из рангов ресурсов по степени их дефицитности. Ранжирование ресурсов производим по результатам решения релаксированной задачи: чем меньше величина неиспользованного остатка ресурса в релаксированной задаче, тем он дефицитнее.

Алгоритм $\mathcal{A}_{\text{rk-res}}$

Шаг 1. Решаем задачу (5)–(8) быстрым приближённым асимптотически точным алгоритмом [13, 14] и находим оценочное расписание S^{est} и $\widehat{R}_k$, $k \in K$, — неиспользованный остаток ресурсов в момент его завершения. Ранжируем ресурсы по неубыванию $\widehat{R}_k$: ресурс с меньшим значением $\widehat{R}_k$ будем считать более дефицитным. Для удобства перенумеруем ресурсы от наиболее дефицитного ресурса к наименее дефицитному, который получает последний номер $|K|$.

Шаг 2. Назначим каждому типу ресурсов вес w_k, $k \in K$, в соответствии с произведённым ранжированием: $w_1 \geqslant w_2 \geqslant \ldots \geqslant w_{|K|}$. Подбор значений w_k, $k \in K$, может осуществляться различными способами, а численный эксперимент показал, что для разных групп примеров из PSPLIB выбор наиболее подходящих этих значений неоднозначен. Для каждой работы j определяем её весовую функцию

$$v(j) = -\sum_{k \in K} w_k r_{jk}, \quad j \in N.$$

Шаг 3. Положим $s_j := 0$, $j \in N$, $g := 1$, $S_1 := \{1\}$, $D_1 := \{j \in N \mid (1, j) \in A\}$, $\widetilde{R}_t^k := R_k$, $k \in M$, $t \in [1, \widehat{T}]$.

Шаг 4. Выбираем из множества D_g работу с минимальным значением $v(j)$. Пусть $j \in D_g$ — выбранная работа, $te_j = \max\{s_i + p_i \mid (i, j) \in A\}$. Находим для этой работы момент начала её выполнения, учитывая ограничения предшествования и ресурсные ограничения

$$s_j := \min\left\{t \geqslant te_j \mid r_{jk} \leqslant \widetilde{R}_\tau^k,\ k \in M,\ \tau \in [t, t + p_j]\right\}.$$

Полагаем $g := g + 1$. Добавляем работу j к множеству рассмотренных работ $S_g := S_{g-1} \cup \{j\}$. Пересчитываем $\widetilde{R}_\tau^k := \widetilde{R}_\tau^k - r_{jk}$, $k \in M$, $\tau \in [t, t + p_j]$.

Шаг 5. Если $g < n$, то идём на шаг 4, иначе стоп.

В результате работы алгоритма получаем приближённое решение исходной задачи — расписание s_j. Это решение можно попробовать улучшить, применяя к нему алгоритм локального улучшения решений, известный как алгоритм Forward-backward improvement (FBI) [15].

3. Численные эксперименты

Численные эксперименты проведены на компьютере с операционной системой Windows 7. Алгоритмы реализованы на языке C++ в системе Visual Studio. Для оценки производительности предложенного алгоритма использованы тестовые примеры из электронной библиотеки PSPLIB, представленной в [11]. В частности, рассмотрены наборы тестовых примеров j60, j90 и j120. Множества j60 и j90 содержат примеры с 60 и 90 работами в каждом соответственно, по 48 серий примеров в каждом множестве, по 10 экземпляров в каждой серии — всего по 480 примеров в каждом множестве. Набор данных j120 содержит 60 серий тестовых примеров, по 10 экземпляров в каждой серии — всего 600 экземпляров. Эти примеры содержатся в электронной библиотеке PSPLIB, доступной по ссылке `www.om-db.wi.tum.de/psplib/data.html`.

Представленный алгоритм $\mathcal{A}_{\text{rk-res}}$ тестируем в сравнении с тремя другими алгоритмами, построенными на трёх базовых функциях предпочтения $v(j)$. Это алгоритм $\mathcal{A}_{\text{time}}$, функция предпочтения $v(j)$ которого принимает значение времени начала работ в оценочном расписании S^{est}; алгоритм $\mathcal{A}_{\text{net}}$, в качестве $v(j)$ берём число работ (со знаком минус), непосредственно следующих за работой j в графе G, а в качестве разрешения ничьей используем время начала выполнения работы j в оценочном расписании S^{est}; и, наконец, алгоритм $\mathcal{A}_{\text{res}}$, в котором в качестве $v(j)$ используется информация об объёмах использования работой j ресурсов без учёта их ранжирования (со знаком минус).

Для запуска алгоритма $\mathcal{A}_{\text{rk-res}}$ необходимо определить применяемые в нём весовые коэффициенты ресурсов w_k, $k \in K$, в соответствии с их ранжированием, произведённым ранее. Все примеры из PSPLIB используют 4 типа ресурсов. В ходе проведения численных экспериментов опробованы несколько вариантов w_k. Оказалось, что для разных серий примеров хорошие решения получаются на разных весах ресурсов и что не существует единого доминирующего набора весов. Выделены следующие три варианта задания весов: $w = (1; 0{,}8; 0{,}6; 0{,}4)$, $w = (1; 0{,}9; 0{,}8; 0{,}7)$ и $w_k = 2 - \frac{\widehat{R}_k}{\widehat{R}_4}$, $k \in K$. Первые два варианта являются линейными монотонно убывающими функциями, а в третьем уменьшение значений w_k происходит пропорционально величинам остатков неиспользованных ресурсов в релаксированной задаче. В качестве решения будем брать лучшее из этих трёх вариантов.

Показателем качества решений для сравнения эвристических алгоритмов для задачи RCPSP принято считать среднее процентное отклонение (average percent deviation, APD) получаемых решений от их нижних границ, в качестве которых используется величина критического пути в графе G [16]. В табл. 1 приводятся полученные значения APD для алгоритмов, вошедших в тестирование, и для трёх множеств примеров из PSPLIB.

Таблица 1

Средние отклонения решений (APD, %) от длины критического пути для тестовых примеров из PSPLIB

Алгоритм	Множество тестовых примеров из PSPLIB		
	j60	j90	j120
$\mathcal{A}_{\text{time}}$	15,44	16,16	42,7
$\mathcal{A}_{\text{net}}$	16,29	14,87	40,04
$\mathcal{A}_{\text{res}}$	17,01	15,68	41,7
$\mathcal{A}_{\text{rk-res}}$	14,91	14,03	38,29

Из табл. 1 следует, что алгоритм $\mathcal{A}_{\text{rk-res}}$ показал лучшее APD по сравнению с алгоритмом $\mathcal{A}_{\text{res}}$ с ресурсной функцией предпочтения без учёта ранжирования ресурсов для всех трёх множеств примеров j60, j90 и j120. Более того, он лучший в сравнении со всеми алгоритмами, участвующими в численном эксперименте.

В табл. 1 показаны результаты, полученные на всех сериях примеров множеств j60, j90 и j120. Было бы интересно выделить те серии, на которых новый алгоритм работает особенно хорошо.

Построение каждого отдельного примера в PSPLIB определяют три параметра: сложность сети (NC), коэффициент ресурсов (RF) и ресурсный потенциал (RS). Параметр NC определяет среднее число предшественников для каждой работы, RF устанавливает средний процент количества типов ресурсов, необходимых для выполнения каждой работы, а RS устанавливает средний процент объёма выделяемых ресурсов в каждый момент времени. Нулевое значение RS соответствует минимальной потребности каждого типа ресурсов для выполнения всех работ, в то время как значение RS, равное единице, соответствует случаю неограниченных ресурсов. Значения параметров, использованные для создания примеров из наборов j60 и j90, следующие: NC $\in \{1{,}5; 1{,}8; 2{,}1\}$, RF $\in \{0{,}25; 0{,}5; 0{,}75; 1\}$ и RS $\in \{0{,}2; 0{,}3; 0{,}4; 0{,}5\}$. Известно [17], что значения параметров RF $= 1$, RS $= 0{,}2$ соответствуют достаточно сложным сериям. В множестве примеров j60 и j90 такими сериями являются j6013, j6029, j6045 и j9013, j9029, j9045 соответственно.

Значения параметров для создания примеров набора j120 следующие: NC $\in \{1{,}5; 1{,}8; 2{,}1\}$, RF $\in \{0{,}25; 0{,}5; 0{,}75; 1\}$ и RS $\in \{0{,}1; 0{,}2; 0{,}3; 0{,}4; 0{,}5\}$. Наиболее трудными сериями в j120 являются примеры j12016, j12036, j12056. Перечисленные примеры наиболее трудны, т. е. имеют наибольший средний процент отклонения APD между полученными решениями и длиной критического пути. Более подробно описание процесса создания примеров можно найти в [16].

Произведём сравнение предложенного алгоритма с известными ранее на сериях трудных примеров для каждого множества примеров j60, j90 и j120. В табл. 2 приведены значения APD для наборов трудных серий примеров для всех рассматриваемых алгоритмов.

Таблица 2

Средние отклонения решений (APD, %) от длины критического пути для трудных тестовых серий примеров из PSPLIB

Алгоритм	Множество тестовых примеров из PSPLIB		
	j60 серии 13, 29, 45	j90 серии 13, 29, 45	j120 серии 16, 36, 56
$\mathcal{A}_{\text{time}}$	92,37	88,70	180,98
$\mathcal{A}_{\text{net}}$	92,32	83,96	184,96
$\mathcal{A}_{\text{res}}$	96,17	87,89	208,29
$\mathcal{A}_{\text{rk-res}}$	89,92	82,73	172,64

Как видно из табл. 2, алгоритм $\mathcal{A}_{\text{rk-res}}$ снова превзошёл по APD как алгоритм с ресурсной функцией предпочтения без учёта ранжирования ресурсов $\mathcal{A}_{\text{res}}$, так и все другие рассматриваемые алгоритмы. Причём его преимущество над другими алгоритмами более значительно. Отметим также рост этого преимущества с увеличением размерности тестовых примеров.

Таким образом, предложенный алгоритм имеет лучшие значения APD среди тестируемых алгоритмов на всех примерах из библиотеки PSPLIB, но его преимущество особенно велико для трудных примеров — тех, в которых задействовано максимально большое количество ресурсов, а объёмы выделяемых ресурсов минимальны. Наблюдение выглядит логичным: чем больше объём выделяемых ресурсов, тем сравнительно меньший эффект имеет ранжирование ресурсов, и наоборот, чем большее количество ресурсов задействовано, тем этот эффект значительнее.

Заключение

В работе предложен детерминированный жадный алгоритм для решения задачи календарного планирования с ограниченными ресурсами

в соответствии с критерием минимизации времени выполнения всего проекта. Он основан на новом правиле приоритета работ. Это правило использует эвристику, учитывающую степень критичности (дефицитности) ресурсов, которую получаем из решения релаксированной задачи с ограничением на складируемые (кумулятивные) ресурсы. Проведены численные эксперименты на примерах из электронной библиотеки PSPLIB. Результаты вычислительных экспериментов показывают, что предлагаемый алгоритм является конкурентоспособной эвристикой. Выявлены серии примеров из PSPLIB, на которых предложенный алгоритм показал себя особенно хорошо.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF–2022–0019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Автор заявляет, что у него 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. **Blażewicz J., Lenstra J. K., Rinnooy 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., Hartmann S.** Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis // Project scheduling: Recent models, algorithms and applications. Dordrecht: Kluwer Acad. Publ., 1999. P. 147–178.
5. **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.
6. **Abdolshah M.** A review of resource-constrained project scheduling problems (RCPSP) approaches and solutions // Int. Trans. J. Eng. Manage. Appl. Sci. Technol. 2014. V. 5, No. 4. P. 253–286.
7. **Vanhoucke M.** Resource-constrained project scheduling // Project management with dynamic scheduling. Heidelberg: Springer, 2012. P. 107–137.

8. **Hartmann S., Briskorn D.** A survey of variants and extensions of the resource-constrained project scheduling problem // Eur. J. Oper. Res. 2010. V. 207, No. 1. P. 1–14.
9. **Кочетов Ю. А., Столяр А. А.** Новые жадные эвристики для задачи календарного планирования с ограниченными ресурсами // Дискрет. анализ и исслед. операций. Сер. 2. 2005. Т. 12, № 1. С. 12–36.
10. **Гончаров Е. Н.** Стохастический жадный алгоритм для задачи календарного планирования с ограниченными ресурсами // Дискрет. анализ и исслед. операций. 2014. Т. 21, № 3. С. 10–23.
11. **Kolisch R., Sprecher A.** PSPLIB — A project scheduling problem library // Eur. J. Oper. Res. 1996. V. 96, No. 1. P. 205–216.
12. **Гимади Э. Х., Залюбовский В. В., Севастьянов С. В.** Полиномиальная разрешимость задач календарного планирования со складируемыми ресурсами и директивными сроками // Дискрет. анализ и исслед. операций. 2000. Сер. 2. Т. 7, № 1. С. 9–34.
13. **Гимади Э. Х.** О некоторых математических моделях и методах планирования крупномасштабных проектов // Модели и методы оптимизации. Тр. АН СССР. Сиб. отд-ние. Ин-т математики. Вып. 10. Новосибирск: Наука, 1988. С. 89–115.
14. **Gimadi Eh. Kh., Sevastianov S. V.** On solvability of the project scheduling problem with accumulative resources of an arbitrary sign // Operations research proceedings 2002. Sel. Pap. Int. Conf. Operations Research (Klagenfurt, Austria, Sept. 2–5, 2002). Heidelberg: Springer, 2003. P. 241–246.
15. **Li K. Y., Willis R. J.** An iterative scheduling technique for resource-constrained project scheduling // Eur. J. Oper. Res. 1992. V. 56, No. 3. P. 370–379.
16. **Kolisch R., Sprecher A., Drexl A.** Characterization and generation of a general class of resource-constrained project scheduling problems // INFORMS Manage. Sci. 1995. V. 41. P. 1693–1703.
17. Project scheduling: Recent models, algorithms and applications. Dordrecht: Kluwer Acad. Publ., 1999. 546 p.

Гончаров Евгений Николаевич

Статья поступила
21 февраля 2024 г.
После доработки —
27 апреля 2024 г.
Принята к публикации
22 июня 2024 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2024. Vol. 31, No. 4. P. 27–39

UDC 519.8+518.25 DOI: 10.33048/daio.2024.31.796

A GREEDY ALGORITHM FOR THE RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM

E. N. Goncharov

Sobolev Institute of Mathematics,
4 Akad. Koptyug Avenue, 630090 Novosibirsk, Russia
E-mail: gon@math.nsc.ru

Abstract. The resource-constrained project scheduling problem (briefly RCPSP) is a general scheduling problem that includes precedence and resource constraints. Activities preemptions are not allowed. Resources are renewable and there is a unique way to perform the activities. The problem with renewable resources is NP-hard in the strong sense. We propose a new deterministic greedy algorithm. It is based on heuristics that use information obtained from a relaxing problem. The algorithm is tested with standard data sets given by Kolisch library PSPLIB for j60, j90, and j120 and found to be performing well. Tab. 2, bibliogr. 17.

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

References

1. **P. Brucker, A. Drexl, 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. Blażewicz, J. K. Lenstra,** and **A. H. G. Rinnooy Kan,** Scheduling subject to resource constraints: Classification and complexity, *Discrete Appl. Math.* **5** (1), 11–24 (1983).
4. **R. Kolisch** and **S. Hartmann,** Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis, in *Project Scheduling: Recent Models, Algorithms and Applications* (Kluwer Acad. Publ., Dordrecht, 1999), pp. 147–178.

English transl.: *Journal of Applied and Industrial Mathematics* **18** (4), 679–685 (2024), DOI: 10.1134/S1990478924040057.

5. **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).
6. **M. Abdolshah,** A review of resource-constrained project scheduling problems (RCPSP) approaches and solutions, *Int. Trans. J. Eng. Manage. Appl. Sci. Technol.* **5** (4), 253–286 (2014).
7. **M. Vanhoucke,** Resource-constrained project scheduling, in *Project Management with Dynamic Scheduling* (Springer, Heidelberg, 2012), pp. 107–137.
8. **S. Hartmann** and **D. Briskorn,** A survey of variants and extensions of the resource-constrained project scheduling problem, *Eur. J. Oper. Res.* **207** (1), 1–14 (2010).
9. **Yu. A. Kochetov** and **A. A. Stolyar,** New greedy heuristics for the scheduling problem with constrained resources, *Diskretn. Anal. Issled. Oper., Ser. 2,* **12** (1), 12–36 (2005) [Russian].
10. **E. N. Goncharov,** A stochastic greedy algorithm for the resource-constrained project scheduling problem, *Diskretn. Anal. Issled. Oper.* **21** (3), 10–23 (2014) [Russian].
11. **R. Kolisch** and **A. Sprecher,** PSPLIB — A project scheduling problem library, *Eur. J. Oper. Res.* **96** (1), 205–216 (1996).
12. **Eh. Kh. Gimadi, V. V. Zalyubovskii,** and **S. V. Sevastianov,** Polynomial solvability of scheduling problems with storable resources and directive deadlines, *Diskretn. Anal. Issled. Oper., Ser. 2,* **7** (1), 9–34 (2000) [Russian].
13. **Eh. Kh. Gimadi,** Some mathematical models and methods for scheduling large-scale projects, in *Models and Methods of Optimization* (Trudy AN SSSR, Sib. Otd., Inst. Mat., Vol. 10) (Nauka, Novosibirsk, 1988), pp. 89–115 [Russian].
14. **Eh. Kh. Gimadi** and **S. V. Sevastianov,** On solvability of the project scheduling problem with accumulative resources of an arbitrary sign, in *Operations Research Proceedings 2002* (Sel. Pap. Int. Conf. Operations Research, Klagenfurt, Austria, Sept. 2–5, 2002) (Springer, Heidelberg, 2003), pp. 241–246.
15. **K. Y. Li** and **R. J. Willis,** An iterative scheduling technique for resource-constrained project scheduling, *Eur. J. Oper. Res.* **56** (3), 370–379 (1992).
16. **R. Kolisch, A. Sprecher,** and **A. Drexl,** Characterization and generation of a general class of resource-constrained project scheduling problems, *INFORMS Manage. Sci.* **41**, 1693–1703 (1995).
17. Project Scheduling: Recent Models, Algorithms and Applications (Kluwer Acad. Publ., Dordrecht, 1999).

Evgenii N. Goncharov

Received February 21, 2024
Revised April 27, 2024
Accepted June 22, 2024