УДК 519.85 ISSN 1995-5499
DOI: https://doi.org/10.17308/sait/1995-5499/2022/2/72-80
Поступила в редакцию 10.03.2022
Подписана в печать 29.06.2022

АЛГОРИТМЫ СОСТАВЛЕНИЯ РАБОЧИХ ГРАФИКОВ РЕСУРСОВ ПРИ ПЛАНИРОВАНИИ ПРОЕКТА

© 2022 В. В. Коротков✉

Воронежский государственный университет
Университетская пл., 1, 394018 Воронеж, Российская Федерация

Аннотация. В работе рассматривается задача построения рабочих графиков для ресурсов, задействованных в реализации проекта. Решение данной задачи необходимо для организации рабочего процесса на уровне конкретных исполнителей. Предлагаемый подход исходит из наличия построенного общего расписания проекта и основывается на минимизации индивидуальных рисков, определяемых характеристиками распределения работ. При этом подход стремится избежать необходимость в дополнительных оценках сверх тех, что традиционно получают на более ранних этапах планирования. Это делает его легко внедряемым на практике. В статье рассматривается две возможные формулировки задачи: с однородными полностью взаимозаменяемыми ресурсами и с наличием нечетких оценок сложности выполняемых исполнителями работ. Приводится принцип переформулирования исходной задачи за счет представления ограничений на пересекающиеся работы в виде графа. Это позволяет получить все возможные конкретные варианты рабочих графиков как независимые множества вершин графа ограничений. В качестве метода решения предлагается алгоритм, основанный на методе ветвей и границ. Приводится описание его основных компонентов, включая оценку нижней границы, проверку ограничений и некоторые приемы, позволяющие повысить эффективность. Наконец, демонстрируется применение предлагаемых методов и алгоритмов на конкретных исходных данных.
Ключевые слова: календарное планирование проекта, комбинаторная оптимизация, нечеткие числа, метод ветвей и границ, теория графов.

ВВЕДЕНИЕ

Планирование — самый важный этап для успешной реализации любого проекта. Однако многообразие параметров, субъективный характер многих процессов и, как правило, недостаточность информации делают этот этап также и необычайно сложным. При этом решение некоторых задач проектного планирования успешно поддается автоматизации.

Одна из таких задач заключается в составлении расписания выполнения проекта, которое является одним из трех базовых планов наряду с бюджетом и планом по содержанию [1]. Расписание определяет конкретные сроки выполнения этапов и отдельных работ проекта. При этом обычно учитываются требуемые для выполнения работ ресурсы и их общее число, имеющееся в наличии. Однако для перехода к стадии реализации проекта затем необходимо распределить конкретных исполнителей и конкретные единицы ресурса по отдельным работам. То есть составить их рабочие графики. Подобное распределение также может быть осуществлено автоматическим образом. Для этого, как правило, отдельным единицам ресурса приписываются

✉ Коротков Владислав Владимирович
e-mail: korotkov@cs.vsu.ru

индивидуальные особенности, ограничивающие множество доступных им работ или влияющие на время их выполнения. В качестве таких особенностей может выступать опыт, навыки, индивидуальные наклонности. Тогда распределение исполнителей становится частью общей задачи построения расписания, нацеленной на сокращение сроков реализации проекта [2–4]. Такой подход требует дополнительные управленческие издержки на формальную оценку навыков отдельных сотрудников и параметров работ проекта. При этом субъективный и относительный характер вышеупомянутых характеристик затрудняет проведение подобных оценок и не гарантирует состоятельность подхода. Кроме того, внесение подобных усложнений увеличивает комбинаторную сложность задачи, которая даже в классической формулировке относится к классу NP-трудных [4].

Альтернативный подход может заключаться в составлении расписания классическими методами с последующим составлением рабочих графиков отдельных ресурсов. В этом случае распределение не влияет на общее расписание и его продолжительность, потому необходимо ввести иной критерий оптимальности. Одним из наиболее значимых показателей потенциальной успешности проекта является мера его совокупного риска, складывающегося в том числе из различных индивидуальных рисков [1]. Источником индивидуальных рисков помимо прочего может быть персонал и прочие ресурсы компании. Причем на вероятность и масштаб подобных рисков влияют характеристики построенных рабочих графиков.

Таким образом, можно сформулировать задачу распределения ресурсов по работам уже имеющегося расписания, которая ставит своей целью минимизацию рискованности процесса выполнения. Далее приведены две возможные формулировки данной задачи: с однородностью и неоднородностью ресурсов.

1. ПОСТАНОВКА ЗАДАЧИ

1.1. Постановка задачи с однородными ресурсами

Однородность ресурсов подразумевает, что всякая пара единиц ресурса одного типа взаимозаменяема. Поэтому отсутствуют какие-либо ограничения на распределение задач между ними. Однако произвольное распределение ресурсов может привести к повышению совокупного риска проекта из-за неравномерной загруженности. Выдача отдельному сотруднику большего объема работ по сравнению с другими, с одной стороны, ведет к нелинейному увеличению вероятности ассоциированного с ним персонального риска (из-за стресса и прочих субъективных факторов), с другой — к увеличению масштабов влияния этого риска. В случае недееспособности такого сотрудника или ресурса более значительная часть проекта оказывается под угрозой невыполнения или срыва сроков. Для материальных же ресурсов, например, станков, указанная проблема помимо аналогичного увеличения рисков ведет к неравномерному износу. Поэтому в целях обеспечения большей надежности процессов выполнения проекта необходимо добиться равномерной нагрузки всех единиц имеющихся ресурсов.

Пусть имеется N взаимозаменяемых единиц ресурса одного вида и M требующих его работ проекта. Обозначим τ_j продолжительности работ. Тогда можно сформулировать следующую оптимизационную задачу:

$$\sum_{i=1}^{N}\left(\sum_{j=1}^{M} x_{ij}\tau_j - \overline{\tau}\right)^2 \to \min, \tag{1}$$

$$\overline{\tau} = \frac{\sum_{j=1}^{M} r_j \tau_j}{N}, \tag{2}$$

при соблюдении ограничений:

$$\sum_{i=1}^{N} x_{ij} = r_j \qquad \forall j, \tag{3}$$

$$x_{ij}x_{ij'} = 0 \qquad \forall i, j, j' \in O_j, \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad \forall i, j. \tag{5}$$

Здесь x_{ij} — бинарная переменная, равная 1, если ресурс i выделен на выполнение работы j, r_j — число требуемых работе ресурсов, O_j — множество работ, хотя бы частично пересекающихся во времени с j.

1.2. Постановка задачи с неоднородными ресурсами

Неоднородность ресурсов подразумевает, как минимум, их неполную взаимозаменяемость. Для человеческих ресурсов источником такой неоднородности служит различие в опыте, индивидуальных способностях и прочие субъективные качества. Как правило, трудные задачи должны решаться более опытными и способными сотрудниками, а более простые можно поручить широкому кругу исполнителей. Назначение сотрудника на несоответствующую его уровню работу ведет к многократному увеличению риска превышения сроков или срыва работ. Таким образом, для уменьшения совокупного риска проекта при составлении рабочих графиков необходимо добиться максимального соответствия сложности работ уровням назначаемых на них сотрудников.

Многие методы оценки продолжительности работ проекта основаны на их предварительном ранжировании или непосредственной оценке сложности [5]. Уровень сотрудника можно также оценивать в терминах сложности решаемых им задач, что легко определяется из исторической информации или методом опроса коллег. Таким образом, необходимые для построения модели оценки уже имеются по результатам выполнения предыдущих этапов проектного планирования, их получение не сопряжено со сколь-либо значительными усилиями и временными издержками.

Традиционным способом численного представления экспертных оценок являются нечеткие числа, среди которых наиболее просты и общеупотребимы числа треугольной формы. Функция принадлежности всякого нечеткого треугольного числа $\tilde{M}$ имеет треугольную форму и определяется тройкой четких чисел (M^l, M^m, M^r), где M^l и M^r — левый и правый концы, M^m — мода.

Пусть для каждой работы получена оценка ее сложности в виде нечеткого треугольного числа $\tilde{c}_j$. Пусть также для каждой единицы человеческого ресурса путем анализа анонимного опроса сотрудников или истории предыдущих проектов получена нечеткая оценка максимальной сложности решаемых им задач $\tilde{p}_i$. Тогда различие сложности j-й задачи и уровня выделенного на ее выполнение i-го сотрудника согласно принципу обобщения Заде определяется как [6]:

$$\tilde{d}_{ij} = \tilde{p}_i - \tilde{c}_j = (p_i^l - c_j^r, p_i^m - c_j^m, p_i^r - c_j^l). \quad (6)$$

Нечеткие числа, в отличие от случайных величин, отражают не вероятность, а возможность наблюдения определенных значений. В некотором смысле возможность может рассматриваться как верхняя граница вероятности [7]. Тогда долю площади числа $\tilde{d}_{ij}$, лежащую по левую сторону от нулевой оси, можно интерпретировать как возможность несоответствия уровня работы и ее исполнителя. Необходимо добиться минимизации этой величины.

В зависимости от положения моды числа относительно нулевой оси для вычисления искомой доли площади возможны следующие случаи (рис. 1):

$$\begin{cases} 0, & l \geq 0; \\ 1, & r < 0; \\ \dfrac{S_L^-}{S} = \dfrac{\min(0,l)^2}{(m-l)(r-l)}, & m \geq 0; \\ \dfrac{S_L + S_R^-}{S} = 1 - \dfrac{\max(0,r)^2}{(r-m)(r-l)}, & m < 0. \end{cases} \quad (7)$$

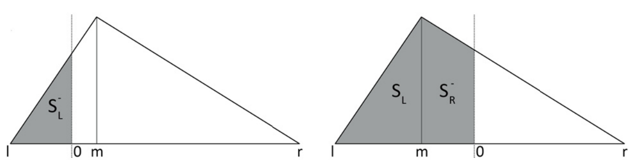

Рис. 1. Составные части площади треугольника, лежащие слева от нуля
[Fig. 1. Components of a triangle lying in the negative half-plane]

Данные значения могут быть вычислены заранее для всех сочетаний работ и исполнителей и занесены в матрицу **E**. Тогда исходная задача сводится к двухкритериальной прямоугольной задаче о назначениях с дополнительными ограничениями:

$$\begin{cases} \sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij}\mathbf{E}[i,j] \to \min \\ \sum_{i=1}^{N}\left(\sum_{j=1}^{M} x_{ij}\tau_j - \overline{\tau}\right)^2 \to \min \end{cases} \quad (8)$$

$$\sum_{i=1}^{N} x_{ij} = r_j \qquad \forall j, \quad (9)$$

$$x_{ij}x_{ij'} = 0 \qquad \forall i, j, j' \in O_j, \quad (10)$$

$$x_{ij} \in \{0,1\} \qquad \forall i, j. \quad (11)$$

Здесь (9) — ресурсные ограничения, (10) — ограничения на участие в параллельно выполняемых работах. Обозначение $\mathbf{E}[i, j]$ следует трактовать как элемент матрицы $\mathbf{E}$, лежащий на пересечении строки i и столбца j.

Для решения поставленной двухкритериальной задачи единый критерий может быть получен методом аддитивной или мультипликативной свертки [8]. В случае аддитивной свертки при этом необходимо нормировать критерии. Другой подход может заключаться в превращении критерия равномерности распределения в ограничения вида:

$$\left|\sum_{j=1}^{M} x_{ij}\tau_j - \overline{\tau}\right| \le \Delta \qquad \forall i = \overline{1, N}. \quad (12)$$

Здесь Δ — заранее задаваемое максимально допустимое отклонение загруженности отдельного ресурса от идеально равномерного. Измеряется в тех же единицах, что и продолжительности работ. Решение подобной задачи допускает некоторую перегруженность более квалифицированных кадров и неполную занятость менее квалифицированных. Но в заранее обозначенных пределах. Такие свойства решения могут быть уместны во многих практичных случаях.

1.3. Сведение к задаче на графе

Основную сложность в исходной формулировке задачи представляют ограничения на параллельное участие ресурса в нескольких работах. Представим эти ограничения в виде графа $G(V, E)$, в котором каждому узлу взаимно однозначно ставится в соответствие одна из работ проекта. Пусть наличие ребра $\{i, j\} \in E$ означает, что работы i и j пересекаются по времени, из-за чего никакой ресурс не может быть задействован в них обеих. Тогда всякий допустимый рабочий график представляет собой независимое множество вершин данного графа. Задаче поиска независимых вершин двойственна задача поиска клик в дополнении графа [9]. Обозначим дополнение графа — обратный граф G', в котором две вершины смежны, если они не смежны в G и наоборот. Клики данного графа (множества вершин, где все попарно смежны друг с другом) определяют все возможные совокупности работ, которые можно назначить одному исполнителю. Для получения списка всех клик может быть использован эффективный алгоритм, предложенный в [10].

На рис. 2 приведен пример подобного графа ограничений и соответствующего обратного графа. Все независимые множества данного графа (клики обратного графа), определяющие допустимые рабочие графики: (A), (B), (C), (D), (E), (A, B), (A, C), (A, D), (C, E), (D, E).

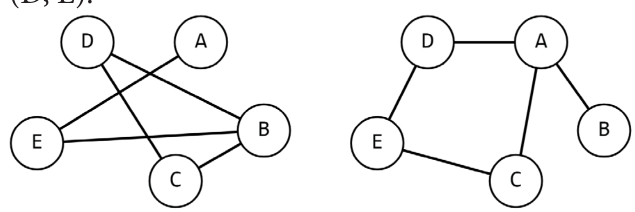

Рис. 2. Граф ограничений (слева) и его дополнение (справа)
[Fig. 2. Constraint graph (left) and its complement graph (right)]

Таким образом, исходная задача от перебора назначений на отдельные работы сводится к перебору допустимых графиков работ, а ограничения (4) и (10) опускаются из рассмотрения.

Пусть полученные допустимые графики работ закодированы в виде бинарных векторов $\mathbf{s}_k = (x_1^k, ..., x_M^k)$. Обозначим $\mathbf{r} = (r_1, ..., r_M)$ вектор ресурсных требований работ, $\mathbf{d} = (d_1, ..., d_M)$ — вектор продолжительностей. Введем переменные y_{ik}, соотносящие рабочие графики и исполнителей. Тогда все ограничения задачи можно переписать в виде:

$$\sum_{i=1}^{N}\sum_{k=1}^{P} y_{ik}\mathbf{s}_k = \mathbf{r}, \quad (13)$$

$$\mathbf{s}_k = \left(x_1^k, ..., x_M^k\right),\ x_j^k \in \{0,1\},\ k = \overline{1,P}. \quad (14)$$

Для каждого из допустимых рабочих графиков возможно заранее рассчитать суммарные нагрузки l_k и суммарные ошибки несоответствия уровней e_{ik}:

$$l_k = \mathbf{s}_k \cdot \mathbf{d} \qquad k = \overline{1,P}, \quad (15)$$

$$e_{ik} = \mathbf{s}_k \cdot \mathbf{E}[i],\ i = \overline{1,N},\ k = \overline{1,P}. \quad (16)$$

Решение сформулированной задачи может быть найдено различными методами дискретной и комбинаторной оптимизации. Далее приводится описание алгоритма, основанного на методе ветвей и границ.

2. АЛГОРИТМ РЕШЕНИЯ

2.1. Генерация начального приближения

Эффективность метода ветвей и границ главным образом обусловлена способностью отсекать целые множества решений на основе сравнения оценки нижней границы для их значений целевой функции с уже известными решениями. Поэтому для более быстрой сходимости можно сгенерировать начальное приближение — случайное решение, удовлетворяющее ограничениям. Тогда у алгоритма с самого начала будет образец для сравнения.

Начальное приближение может быть сгенерировано при помощи процедуры поиска с возвратом. Сперва для каждого из N ресурсов его рабочий график инициализируется пустым множеством работ. Затем запускается рекурсивная процедура поиска, на каждом шаге которой для очередной работы и очередной необходимой ей ресурсной позиции отбирается список кандидатов — ресурсов, чьи текущие рабочие графики не пересекаются с рассматриваемой работой по времени. Если кандидатов нет — текущая ветвь поиска завела в тупик, происходит возврат из рекурсивного вызова с неудачным результатом. Иначе, кандидаты обходятся в случайном порядке и делается попытка разместить рассматриваемую работу в рабочий график соответствующего кандидата и рекурсивно вызвать процедуру поиска для следующего шага алгоритма. Если после рассмотрения всех кандидатов так и не был найден успешный вариант размещения, происходит рекурсивный возврат с неудачным результатом.

2.2. Ветвление

Построение дерева поиска метода ветвей и границ производится следующим образом. Корень дерева соответствует всему множеству возможных решений. На каждом i-м уровне происходит ветвление на P дочерних узлов по принципу присвоения i-му ресурсу одного из P допустимых графиков. Число независимых множеств вершин графа ограничений на практичном для данной задачи размере насчитывает сотни и даже тысячи. Поэтому получается крайне ветвистое дерево, которое содержало бы в полном случае $\dfrac{P^N - 1}{P - 1}$ узлов. Однако благодаря отсечениям число развернутых узлов меньше на несколько порядков. Кроме того, имея начальное приближение, возможно предварительно значительно сократить число возможных рабочих графиков, отсеяв заведомо неудовлетворительные.

2.3. Оценка нижней границы

Обозначим h уровень, на котором расположен узел, $\mathbf{r}^+ = (\mathbf{r}_1^+, ..., \mathbf{r}_M^+)$ — вектор числа распределенных на работы ресурсов, $\mathbf{r}^- = (\mathbf{r}_1^-, ..., \mathbf{r}_M^-)$ — вектор остаточных ресурсов:

$$\mathbf{r}^+ = \sum_{i=1}^{h} \sum_{k=1}^{P} y_{ik} \mathbf{s}_k \quad (17)$$

$$\mathbf{r}^- = \mathbf{r} - \mathbf{r}^+. \quad (18)$$

Нижняя граница для критерия равномерности распределения определяется по формуле:

$$\sum_{i=1}^{h} \sum_{k=1}^{P} y_{ik} l_k + (N - h) \min_k (l_k). \quad (19)$$

Нижняя граница критерия соответствия уровней работ и исполнителей определяется путем назначения на оставшиеся незаполненными работы тех оставшихся исполнителей, для которых минимальна соответствующая величина ошибки:

$$\sum_{i=1}^{h} \sum_{k=1}^{P} y_{ik} e_{ik} + \mathbf{r}^- \cdot \mathbf{e}_{\min}^{h+1} \quad (20)$$

$$\mathbf{e}_{\min}^{h+1} = (\min(\mathbf{E}[h+1]), ..., \min(\mathbf{E}[N])).$$

2.4. Проверка ограничений

Для листового узла дерева поиска, олицетворяющего конкретное решение, проверяется условие (13).

Для внутреннего узла на уровне h проверяется непревышение числа необходимых для каждой работы ресурсов и достаточность оставшихся нераспределенными ресурсов для доукомплектации исполнителей работ:

$$\begin{cases} r_j^+ \le r_j, \\ r_j^- \le N - h, \end{cases} \forall\, j = \overline{1, M}. \qquad (21)$$

2.5. Описание алгоритма

В реализованной версии алгоритма решения неоднородной задачи в качестве целевой функции использовалась мультипликативная свертка двух рассмотренных критериев. В общем же виде алгоритм описывается следующей последовательностью шагов.

1. Из графа ограничений находится список всех возможных графиков $\mathbf{s}_k$.

2. Для всех графиков рассчитываются загруженности l_k и ошибки несоответствия e_{ik}.

3. Генерируется начальное приближение.

4. Для каждого графика оценивается нижняя граница целевой функции при его назначении каждому из N исполнителей. Если наилучшая из этих оценок оказывается хуже значения, соответствующего начальному приближению, то в дальнейшем график опускается из рассмотрения. Таким образом возможно отбраковать заведомо неудачные рабочие графики и существенно понизить сложность дальнейшего перебора.

5. Если критерий равномерности рассматривается в виде ограничений, то список всех графиков также фильтруется по условию (12).

6. Начальное приближение запоминается как текущее известное лучшее решение. Неубывающая очередь с приоритетом, хранящая узлы-кандидаты на развертывание, инициализируется корневым узлом дерева поиска и соответствующей оценкой нижней границы.

7. Если очередь пуста, то алгоритм завершен. Иначе из нее достается очередной узел.

8. Если оценка для n хуже, чем лучшее из ранее найденных решений, то переходят к шагу 7.

9. Делается проверка на выполнение ограничений (21). Если ограничения не выполняются, то переходят к шагу 7.

10. Если узел листовой, то обновляется текущее лучшее решение. Иначе генерируется P дочерних узлов, в которых первому по порядку неразмещенному ресурсу присваивается один из P рабочих графиков. Для каждого рассчитывается оценка нижней границы, и если она оказывается лучше текущего решения, узел добавляется в очередь с приоритетом.

3. ЧИСЛЕННАЯ АПРОБАЦИЯ

Описанные модели и алгоритмы были реализованы на языке Python с применением библиотек NetworkX и NumPy и апробированы на тестовых задачах. В табл. 1 приведены исходные данные одного из примеров. В расписании некоторого проекта 10 работ с указанными параметрами требуют для своего выполнения один и тот же вид ресурса. Имеется 4 единицы данного ресурса. Необходимо распределить работы по всем единицам ресурса, получив тем самым их рабочие графики.

Таблица 1. Расписание работ тестового примера
[Table 1. Test case schedule]

№	Начало	Время	Ресурсы	Сложность
1	0	3	2	(6,8,9)
2	3	5	3	(1,3,4)
3	8	1	3	(4,4,4)
4	3	3	1	(4,5,6)
5	9	2	1	(6,8,9)
6	9	4	2	(3,4,5)
7	15	5	3	(3,5,7)
8	11	6	1	(1,2,4)
9	11	4	1	(7,9,9)
10	20	4	1	(1,2,3)

В табл. 2 представлено решение, предполагающее однородность всех единиц ресурса. Как видно, нагрузка равномерно распределена, что положительным образом скажется на надежности процессов выполнения проекта. 4 полученных рабочих графика могут быть распределены произвольным образом или руководствуясь какими-либо экспертными соображениями.

Таблица 2. Решение при однородности ресурсов
[Table 2. Solution for homogeneous resources]

№	Работы	Нагрузка
1	1, 2, 6, 7	17
2	2, 3, 8, 10	16
3	1, 3, 4, 6, 7	16
4	2, 3, 5, 7, 9	17

Пусть те же 4 сотрудника имеют нечеткие оценки максимальной сложности выполняемых задач (7,8,9), (4,6,8), (4,6,7) и (2,4,5) соответственно. В табл. 3 представлено решение для подобной постановки задачи. Учет индивидуальных возможностей сотрудников позволил более предметно распределить работы. Самые сложные достались способному сотруднику №1, в то время как не настолько опытному сотруднику №4 предстоит выполнять в основном простые поручения. При этом задачи также распределены максимально равномерно. Такой подход способен значительно снизить возможные риски и повысить надежность процессов выполнения проекта.

Таблица 3. Решение при неоднородности ресурсов
[Table 3. Solution for heterogeneous resources]

№	Работы	Нагрузка
1	2, 3, 5, 7, 9	17
2	1, 3, 4, 6, 7	16
3	1, 2, 6, 7	17
4	2, 3, 8, 10	16

Время работы алгоритма зависит от качества полученного первого приближения. Для рассмотренной задачи размером в 10 работ время выполнения на компьютере с процессором i5-8250U составило не более 3 секунд для однородной задачи и 10 для неоднородной. При увеличении размеров задачи время работы алгоритма будет расти экспоненциально. Однако на практике вряд ли стоит ожидать больших размеров, т.к. построение рабочих графиков на проекте распадается на несколько задач рассмотренного типа. Каждая решается на подмножестве ресурсов определенного типа или специализации и подмножестве соответствующих работ.

ЗАКЛЮЧЕНИЕ

Рассмотренные в данной работе модели построения рабочих графиков относительно просты и соответствуют популярным методологиям проектного планирования. Отсутствие лишних компонент и параметров делает их легко внедряемыми в рабочий процесс любых организаций. При этом они позволяют добиться повышения надежности процессов выполнения проекта за счет снижения некоторых индивидуальных рисков, вносящих свой вклад в совокупный риск проекта.

Предложенный вариант формализации ограничений с применением теории графов позволяет построить эффективные алгоритмы решения. В частности, описанная реализация метода ветвей и границ непосредственно может быть реализована в программном обеспечении для проектного планирования и поддержки принятия управленческих решений. Его эффективность может быть дополнительно улучшена применением более точных, но и более затратных по вычислениям оценок нижней границы и проверок ограничений. Это предположение требует отдельного исследования.

КОНФЛИКТ ИНТЕРЕСОВ

Автор декларирует отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Павлов, А. Н.* Управление проектами на основе стандарта PMI PMBOK. Изложение методологии и опыт применения / А. Н. Павлов. – 7-е изд., электрон. – Москва : Лаборатория знаний, 2021. – 274 с.

2. *Коротков, В. В.* Планирование проекта с учётом навыков сотрудников в условиях нечёткой информации / В. В. Коротков // Информатика: проблемы, методы, технологии: сборник материалов XXI Международной научно-методической конференции (Воронеж, 11–12 февраля 2021 г.). – Воронеж, 2021. – С. 1250–1256.

3. *Зацаринный, А. А.* Моделирование процессов сетевого планирования портфеля проектов с неоднородными ресурсами / А. А. Зацаринный, В. В. Коротков, М. Г. Матвеев // Информатика и её применения. – 2019. – Т. 13, № 2. – С. 92–99.

4. *Hartmann, S.* An updated survey of variants and extensions of the resource-constrained project scheduling problem / S. Hartmann, D. Briskorn // European Journal of Operational Research. – 2022. – Vol. 297, № 1. – P. 1–14.

5. *Кон, М.* Agile: оценка и планирование проектов / М. Кон. – Москва : Альпина Паблишер, 2018. – 424 с.

6. *Шевляков, А. О.* Сравнение различных нечетких арифметик / А. О. Шевляков, М. Г. Матвеев // Искусственный интеллект и принятие решений. – 2017. – №4. – С. 60–68.

7. *Пытьев, Ю. П.* Возможность. Элементы теории и применения / Ю. П. Пытьев. – Москва : Едиториал УРСС, 2000. – 192 с.

8. *Зак, Ю. А.* Прикладные задачи многокритериальной оптимизации / Ю. А. Зак. – Москва : Экономика, 2014. – 455 с.

9. *Омельченко, А. В.* Теория графов / А. В. Омельченко. – Москва : МЦНМО, 2018. – 416 с.

10. Genome-Scale Computational Approaches to Memory-Intensive Applications in Systems Biology / Y. Zhang [et al.] // SC ‘05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing (Seattle, November 12–18, 2005). – IEEE Computer Society, 2005. – 12 p.

Коротков Владислав Владимирович — ассистент кафедры информационных технологий управления Воронежского государственного университета.
E-mail: korotkov@cs.vsu.ru
ORCID iD: https://orcid.org/0000-0002-4045-7721

DOI: https://doi.org/10.17308/sait/1995-5499/2022/2/72-80 ISSN 1995-5499
Received 10.03.2022
Accepted 29.06.2022

ALGORITHMS OF CONSTRUCTING RESOURCE WORK SCHEDULES IN PROJECT PLANNING

© 2022 V. V. Korotkov✉

Voronezh State University
1, Universitetskaya Square, 394018 Voronezh, Russian Federation

Annotation. The paper considers the problem of constructing work schedules for the resources involved in a project execution. The solution of this task is necessary to organize a workflow at the level of specific performers. The proposed approach uses existing general project schedule and is based on minimizing individual risks determined by the characteristics of the distribution of work. At the same time, the approach tries to avoid the need for additional estimates beyond those that are traditionally obtained at earlier stages of planning. This makes it easy to implement in practice. The paper considers two possible formulations of the problem: with homogeneous completely interchangeable resources and with fuzzy estimates of the complexity of the jobs performed. An approach is proposed for representing

✉ Korotkov Vladislav V.
e-mail: korotkov@cs.vsu.ru

constraints on intersecting jobs in the form of a graph to reformulate the original problem. This allows one to get all possible options for work schedules as independent sets of vertices in the constraint graph. An algorithm based on the branch and bound method is proposed as a solution method. Its main components are described, including lower bound estimation, constraint checking, and some tricks to improve efficiency. Finally, the application of the proposed methods and algorithms to specific input data is demonstrated.
Keywords: project scheduling, combinatorial optimization, fuzzy numbers, branch and bound, graph theory.

CONFLICT OF INTEREST

The author declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. *Pavlov A. N.* (2021) Upravlenie proektami na osnove standarta PMI PMBOK. Izlozhenie metodologii i opyt primeneniya [Project Management Based on the PMI PMBOK Standard. Presentation of Methodology and Application Experience]. *Moscow, Laboratoriya znanii*.

2. *Korotkov V. V.* (2021) Planirovanie proekta s uchetom navykov sotrudnikov v usloviyakh nechetkoi informatsii [Project Scheduling with Employee Skills under Fuzzy Information]. *In: IPMT 2021. Proceedings of 21st International Scientific and Methodological Conference*, 11-12 February 2021, Voronezh, Russia. Voronezh, P. 1250–1256.

3. *Zatsarinnyi A. A., Korotkov V. V. and Matveev M. G.* (2019) Modelling the Process of Network Planning of a Portfolio of Projects with Heterogeneous Resources under Fuzziness. *Informatika i ee primeneniya [Informatics and its Applications]*. 13(2). P. 92–99.

4. *Hartmann S. and Briskorn D.* (2022) An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*. 297(1). P. 1–14.

5. *Cohn M.* (2005) Agile Estimating and Planning. Translated from English by Ionov, V. (2018). *Moscow, Al'pina Pablisher*.

6. *Shevlyakov A. O. and Matveev M. G.* (2017) A comparison of different fuzzy arithmetics. *Iskusstvennyi intellekt i prinyatie reshenii [Artificial Intelligence and Decision Making]*. 4. P. 60–68.

7. *Pyt'ev Yu. P.* (2000) Vozmozhnost'. Elementy teorii i primeneniya [Possibility. Elements of Theory and Applications]. *Moscow, Editorial URSS*.

8. *Zak Yu. A.* (2014) Prikladnye zadachi mnogokriterial'noi optimizatsii [Applied Problems of Multiobjective Optimization]. *Moscow, Ekonomika*.

9. *Omel'chenko A. V.* (2018) Teoriya grafov [Graph theory]. *Moscow, MCCME*.

10. *Zhang Y., Abu-Khzam F. N., Baldwin N. E., Chesler E. J., Langston M. A. and Samatova N. F.* (2005). Genome-Scale Computational Approaches to Memory-Intensive Applications in Systems Biology. *In: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing (SC '05)*, 12–18 November 2005, Seattle.

Korotkov Vladislav V. — assistant of the department of Information Technologies in Management, Voronezh State University.
E-mail: korotkov@cs.vsu.ru
ORCID iD: https://orcid.org/0000-0002-4045-7721