

НАУЧНЫЕ ИССЛЕДОВАНИЯ И РАЗРАБОТКИ

**№ 3(8)/2014**

Издается с 2012 года

**РОССИЙСКИЙ ЖУРНАЛ УПРАВЛЕНИЯ ПРОЕКТАМИ**

Свидетельство о регистрации средства массовой информации ПИ № ФС77-48835 от 2 марта 2012 г.

**Издатель:**

ООО «Научно-издательский центр ИНФРА-М»  
127282, Москва, ул. Полярная,  
д. 31В, стр. 1  
Тел.: (495) 280-15-96  
Факс: (495) 280-36-29  
E-mail: books@infra-m.ru  
http://www.infra-m.ru

**Главный редактор:**

Царьков И.Н., канд. экон. наук, доцент кафедры управления проектами НИУ ВШЭ

**Ведущие рубрик:**

«Управление комплексными проектами»,  
ст. преп. НИУ ВШЭ Клименко О.А.

**Выпускающий редактор:**

Путкова А.В.

**Отдел подписки:**

Назарова М.В.  
Тел.: (495) 280-15-96, доб. 249  
e-mail: podpiska@infra-m.ru

© ИНФРА-М, 2014

Присланные рукописи не возвращаются.

Точка зрения редакции может не совпадать с мнением авторов публикуемых материалов.

Редакция оставляет за собой право самостоятельно подбирать к авторским материалам иллюстрации, менять заголовки, сокращать тексты и вносить в рукописи необходимую стилистическую правку без согласования с авторами. Поступившие в редакцию материалы будут свидетельствовать о согласии авторов принять требования редакции.

Перепечатка материалов допускается с письменного разрешения редакции.

При цитировании ссылка на журнал «НИР. Российский журнал управления проектами» обязательна.

Редакция не несет ответственности за содержание рекламных материалов.

Подписано в печать 10.09.2014.

Формат 60×90/8. Бумага офсетная.

Тираж 1000 экз. Заказ №

САЙТ: www.naukaru.ru

E-mail: mag6@naukaru.ru

**Содержание**

Обращение к читателям..... 3

**УПРАВЛЕНИЕ КОМПЛЕКСНЫМИ ПРОЕКТАМИ**

**Клименко О.А.**

Управление комплексными проектами: история вопроса, проблемы и подходы..... 4

**МЕТОДЫ И ИНСТРУМЕНТЫ УПРАВЛЕНИЯ ПРОЕКТАМИ**

**Царьков И.Н.**

Развитие методов сэмплирования в задаче сокращения продолжительности проекта с ограниченными ресурсами..... 14

**СОЦИАЛЬНО-ПСИХОЛОГИЧЕСКИЕ АСПЕКТЫ УПРАВЛЕНИЯ ПРОЕКТАМИ**

**Решетко С., Багратиони К.А.**

МВТИ как метод экспресс-диагностики поведенческих компетенций менеджеров.....25

**МЕТОДОЛОГИЯ УПРАВЛЕНИЯ ПОРТФЕЛЯМИ, ПРОГРАММАМИ И ПРОЕКТАМИ**

**Макеева О.Ю.**

Ключевые особенности реализации и планирования программы.....35

**В МИРЕ УПРАВЛЕНИЯ ПРОЕКТАМИ**

Отчет о 4-й международной конференции «Молодежь и управление проектами в России» .....46

**Stephen Heys**

Think2Impact™ – Making Systems Science Work for Dealing with Complexity in all Sectors of Society ..... 49

Информация для авторов ..... 54

## МЕТОДЫ И ИНСТРУМЕНТЫ УПРАВЛЕНИЯ ПРОЕКТАМИ

# Развитие методов сэмплирования в задаче сокращения продолжительности проекта с ограниченными ресурсами

## Sampling Methods' Evolution in Problem of Duration Reduction of Project with Limited Resources

DOI: 10.12737/5782

Получено: 08.08.2014 г. / Одобрено: 20.08.2014 г. / Опубликовано: 17.09.2014 г.

**И.Н. Царьков**

Канд. экон. наук, доцент,  
НИУ ВШЭ,  
Россия, 105187, Москва, ул. Кирпичная, 33,  
e-mail: itsarkov@hse.ru

**I.N. Tsarkov**

Ph.D. in Economics, Associate Professor,  
National Research University  
«Higher School of Economics»,  
33 Kirpichnaya st., Moscow, 105187, Russia,  
e-mail: itsarkov@hse.ru

**Аннотация**

Задача сокращения продолжительности проекта в условиях ограниченных возобновляемых ресурсов (RCPSP) уже более 50 лет является одной из самых популярных тем исследований в области математических моделей управления проектами. На сегодняшний день применение точных оптимизационных методов на практике оказывается невозможным, поэтому для решения этой задачи используют эвристические методы. Среди множества эвристических методов значительную долю занимают так называемые многопроходные методы формирования расписания, основанные на последовательном применении разных эвристических правил разрешения ресурсных конфликтов к одному и тому же проекту. Если при этом каждое новое расписание получается с использованием случайных чисел, то говорят о сэмплировании. В этой статье будет сделан обзор существующих методов сэмплирования и проведено исследование их эффективности на базе проектов PSPLIB. Кроме того, получено подтверждение, что при большом количестве итераций в методе сэмплирования параллельные схемы превосходят последовательные, и построен новый метод сэмплирования, комбинирующий различные схемы и правила приоритета.

**Ключевые слова:** расписание проекта, ограниченные ресурсы, сэмплирование, продолжительность проекта.

**Abstract**

The project duration reduction problem in the conditions of limited renewable resources is one of the most popular themes within researches in area of project management's mathematical models during more than 50 years. Nowadays applying of exact optimization methods in real practice is impossible, so heuristic methods are used in this problem resolution. Among a set of heuristic methods there is a considerable share of so-called multipass scheduling methods based on consecutive application of different heuristic rules of resource conflicts resolution to the same project. If at this each new schedule turns out with use of random numbers, they speak about sampling. In this paper a review of existing sampling methods will be made, and their efficiency research on the basis of PSPLIB projects will be conducted. Besides, has been received a confirmation that at a large number of iterations in a sampling method the parallel schemes surpass consecutive ones, and a new method of sampling combining various schemes and rules of priority has been constructed.

**Keywords:** project schedule, limited resources, sampling, project duration.

**1. Введение**

Практически в каждом проекте есть ограничения на использование возобновляемых ресурсов. Такими ресурсами обычно являются: труд отдельных специалистов и исполнителей, выполняющих работы проекта; машино-часы дорогостоящего или редкого оборудования (например, суперкомпьютеров) и другие ресурсы, которых можно потратить только ограниченное количество в каждый период выполнения проекта. При использовании подобных ресурсов нередко возникают ресурсные конфликты, заставляющие либо задержать выполнение некоторых работ проекта до того момента времени, когда освободится нужный ресурс, либо тратить дополнительные деньги на привлечение еще одной единицы занятого ресурса. Здесь и далее мы будем считать, что второй вариант недоступен. Также будем предполагать, что нет возможности прерывать выполнение работ. В этих условиях будем решать

задачу нахождения такой последовательности выполнения работ, при которой продолжительность проекта будет минимальна.

Задача сокращения продолжительности проекта в условиях ограниченных возобновляемых ресурсов (RCPSP) уже более 50 лет является одной из самых популярных тем исследований в области математических моделей управления проектами. Одна из самых первых попыток решения этой проблемы связана с методом критического пути [9]. Было предложено для разрешения ресурсных конфликтов задерживать выполнение той работы, у которой полный резерв, рассчитанный по методу критического пути (МКП) без учета ограничений на ресурсы, оказывался больше. Как показали многочисленные исследования [2; 3; 12; 6], такое правило *SLK* (с небольшими уточнениями) оказалось одним из самых лучших эвристических правил. Только в 1996 г. профессором Р. Колишем [12] было предложено

правило *WCS* (*Worst Case Slack*), которое превзошло по эффективности *SLK*.

Огромное внимание к эвристическим правилам минимизации продолжительности проекта обусловлено тем, что любой оптимальный метод является *NP*-сложным [4]. Это означает, что во многих случаях на практике найти оптимальное решение практически невозможно.

Математически строгая формулировка этой проблемы на языке целочисленного линейного программирования была предложена в 1968 г. [16].

В проекте  $n$ -работ, первая и последняя являются вехами старта и финиша соответственно. В проекте используется  $K$  различных ограниченных ресурсов. В каждый период доступно всего  $a_k$  единиц каждого ресурса. Каждая работа имеет продолжительность  $d_i$  и требует для своего выполнения  $r_{ik}$  единиц  $k$ -ресурса. Прерывать выполнение работ недопустимо.

Переменными модели являются  $x_{it}$  – булевы переменные, принимающие значение единицы, когда работа  $i$  заканчивается в момент времени  $t$ . Тогда окончание работы  $i$  будет рассчитываться по следующей формуле:

$$f_i = \sum_t t \cdot x_{it} = 1 \cdot x_{i1} + 2 \cdot x_{i2} + 3 \cdot x_{i3} + \dots,$$

где  $t$  – временные единицы проекта (например, дни).

Общая формулировка проблемы будет выглядеть так:

$\sum_{t=EFT_i}^{LFT_i} t \cdot x_{it} \rightarrow \min$  — целевая функция (окончание последней работы проекта);

$x_{it} \in \{0,1\}$  — для каждой  $i$ -работы и  $t$ -периода (булевы переменные);

$\sum_{t=EFT_i}^{LFT_i} x_{it} = 1$  — требование того, чтобы у каждой  $i$ -работы было только одно окончание;

$\sum_{t=EFT_i}^{LFT_i} t \cdot x_{it} \leq \sum_{t=EFT_j}^{LFT_j} t \cdot x_{jt} - d_j$  — для каждого отношения предшествования  $i \rightarrow j$ ;

$\sum_{i=1}^n \sum_{q=\max\{t, EFT_i\}}^{\min\{t+d_i-1, LFT_i\}} r_{iq} \cdot x_{iq} \leq a_k$  — для каждого ресурса  $k = 1, \dots, K$  и для каждого периода  $t = 1, \dots, T$ .

Другие постановки оптимизационной задачи для *RCPSP* были рассмотрены нами ранее [1].

Все эвристические методы нахождения допустимого расписания проекта в условиях ограниченных ресурсов обычно делятся на несколько классов:

- 1) конструктивные методы;
- 2) улучшающие методы;
- 3) метаэвристические методы.

Конструктивные методы позволяют с нуля построить допустимое расписание. Улучшающие используют базовое расписание для генерации нового, более качественного. Метаэвристические методы позволяют последовательно улучшать ряд базовых допустимых расписаний.

Конструктивные методы определяются тремя компонентами. Первым компонентом является *правило приоритета* (*Priority Rule*) отбора работ, с помощью которого определяется, какая работа будет задержана при возникновении ресурсных конфликтов. Вторым компонентом является схема формирования расписания (*SGS*<sup>1</sup>). Существуют две схемы: последовательная и параллельная. Схемы были тщательно исследованы в работе [13]. И последним компонентом является направление, в котором строится расписание. Можно строить расписание от первой работы к последней, можно наоборот, а можно двигаться сразу с двух сторон. В таких случаях говорят о *прямой* (*FW*), *обратной* (*BW*) и *двунаправленной* (*BI*) *схемах формирования расписания*.

Количество хороших правил приоритета – несколько штук. С учетом различных схем построения расписания получается всего несколько десятков конструктивных методов, которые способны привести к разным расписаниям. А для многих улучшающих методов (особенно для метаэвристических) необходима возможность генерации большого числа допустимых расписаний. Кроме того, стало понятно, что для повышения качества расписания для одного проекта стоит применять *различные* конструктивные методы и из результатов их работы выбирать наилучший. Такие методы называются многопроходными конструктивными, так как они предполагают, в отличие от однопроходных конструктивных, определение старта каждой работы проекта несколько раз. Именно эта операция занимает основное время при работе конструктивного метода.

Один из способов создания многопроходного конструктивного метода с большим количеством разнообразных генерируемых расписаний заключается в использовании датчика случайных чисел для генерации приоритетов работ при некоторой выбранной схеме формирования расписания. Такие методы называют *сэмплированием*. Они представляют собой разновидность метода Монте-Карло. Эффективность и особенности именно этих методов мы исследуем.

<sup>1</sup> *Scheduling Generation Schema*.

## 2. Равномерное и смещенное сэмплирование

Одним из самых удобных и эффективных практических методов решения сложных комбинаторных задач (в том числе оптимизационных) стал метод Монте-Карло, который получил второе рождение благодаря американскому проекту «Манхэттен» создания ядерной бомбы. Метод оказался чрезвычайно простым в применении и нетребовательным к вычислительным ресурсам, которые в то время были еще очень слабыми. В работе [10] утверждается, что своим названием «метод Монте-Карло» он обязан известному математику Джону фон Нейману. До этого этот метод назывался «случайное блуждание» (*random walk*).

**Равномерным сэмплированием (RAS<sup>1</sup>)** в управлении проектами называют конструктивный эвристический метод, у которого зафиксирована некоторая схема формирования расписания и направление его построения. При этом приоритеты работ выбираются случайным образом. Вероятности выбора той или иной работы равны. Получение каждого допустимого расписания этим методом будем называть *итерацией*.

В пределе с помощью равномерного сэмплирования можно перебрать все допустимые расписания. Хотя для этого и потребуется слишком много времени. Нас больше интересует способность метода быстро приближаться к оптимальному решению на самых первых итерациях. Равномерное сэмплирование с этой точки зрения будет представлять собой самую плохую скорость сокращения продолжительности, с которой мы будем сравнивать другие методы, чтобы понять, насколько они хороши.

Еще в 1953 г. [10] для исправления недостатков метода Монте-Карло был предложен **метод смещенного сэмплирования (BRS<sup>2</sup>)**, суть которого заключалась в том, чтобы повышать вероятности генерации решений, лежащих в областях, в которых наличие оптимальных решений наиболее вероятно. Эти области определяются исходя из теоретических соображений для каждой оптимизационной задачи. Подобные изменения способны существенно повысить точность метода Монте-Карло при меньшем числе итераций.

Первая серьезная попытка применения **BRS** для задач управления проектами была предпринята Дейлом Купером в 1976 г. [5]. Предлагалось для каждой итерации использовать фиксированную схему построения расписания и правило приори-

тета. Приоритеты определялись случайным образом на основе вероятностей, которые на каждом этапе применения параллельной схемы определялись следующим образом в зависимости от приоритета, полученного по зафиксированному правилу:

$$p_i = \frac{z_i}{\sum_{k \in S} z_k} \quad \text{если правило приоритета предполагает выбор работы с максимальным значением приоритета (например, MTS<sup>3</sup>);}$$

$$p_i = \frac{1/z_i}{\sum_{k \in S} 1/z_k} \quad \text{если правило приоритета предполагает выбор работы с минимальным значением приоритета (например, SLK).}$$

Здесь и далее индекс  $i$  пробегает все работы из множества  $S$  работ-претендентов, из которых осуществляется выбор на данном этапе применения последовательной или параллельной схемы.  $z_i$  представляют собой значения приоритета работы, рассчитанные по фиксированному правилу. Будем далее обозначать этот метод **BRS-C**.

По сути, значения приоритета работ используются в качестве весов при расчете вероятностей. Таким образом, в процессе сэмплирования будут получаться расписания, основная масса которых будет получаться небольшими изменениями расписания, полученного обычным конструктивным методом с использованием данного правила приоритета.

Из приведенных формул также видно, что значения приоритетов должны быть как минимум неотрицательными, а в случае необходимости минимизировать приоритет – строго положительными. Для выполнения последнего условия Купер предложил прибавлять единицу ко всем значениям приоритета. Например, если в качестве правила приоритета используется правило минимального резерва (*SLK*), то в качестве значений приоритета предлагается брать величину резерва плюс единица.

В 1989 г. в работе [3] был предложен альтернативный способ расчета вероятностей для работ в случае минимизации значений приоритета (будем в дальнейшем его обозначать **BRS-AVT**).

$$z'_i = M - z_i \quad \text{минимизация значений приоритета, где } M \text{ – достаточно большое число для исключения отрицательных значений приоритета;}$$

$$z'_i = z_i \quad \text{максимизация значений приоритета.}$$

Расчет вероятностей осуществляется по аналогичной формуле:

<sup>1</sup> *Random Sampling*.

<sup>2</sup> *Biased Random Sampling*.

<sup>3</sup> *Most Total Successors* – наибольшее число полных последователей работы.

$$p_i = \frac{z'_i}{\sum_{k \in S} z'_k}$$

Недостатком обоих подходов к *BRS* является произвольный выбор поправочных слагаемых для минимизации значений приоритета и исключение возможности выбрать работу с нулевым значением приоритета в случае их максимизации (так как вероятность будет равна нулю).

Другим серьезным недостатком является неопределенность в выборе константы  $M$ , от которой итоговые вероятности существенно зависят. Рассмотрим пример использования в качестве базового правила приоритета минимум полного резерва (*SLK*). Допустим, множество работ-претендентов  $S$  на данном этапе состоит из 3 работ, имеющих резервы 0, 1, 3 соответственно. Рассчитаем вероятности, с которыми будут выбраны эти работы при различных значениях  $M$  (рис. 1).

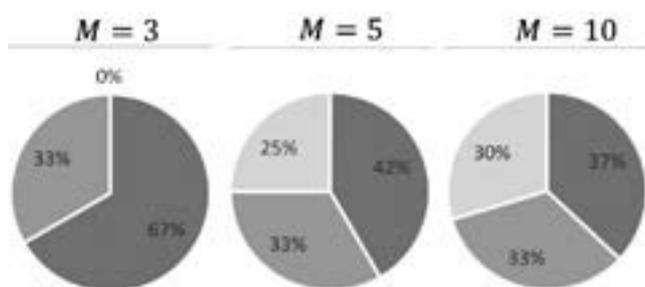


Рис. 1. Зависимость вероятностей выбора работ в методе сэмпирования от значения  $M$

В работе [19] был разработан **нормализованный метод смещенного сэмпирования (*NBRSt*)**, в котором в качестве константы  $M$  предлагается выбрать сумму максимального и минимального значений из значений приоритетов работ-претендентов.

$z'_i = \max_{k \in S} \{z_k\} - z_i + \min_{k \in S} \{z_k\}$ , если значение приоритета нужно минимизировать.

Но после такого преобразования, если в первоначальном наборе значений приоритетов были нули, то они и останутся. Поэтому, если среди значений приоритетов есть хотя бы один ноль, авторы предлагают прибавить к каждому значению  $1/10$  минимального положительного значения приоритета, если такой имеется, и единицу, если такого нет. Последнее преобразование предполагает возведение итогового значения в некоторую степень  $\alpha$ , которая является параметром этого метода.

### 3. Сэмпирование на основе скорректированных приоритетов

Другой важной вехой в развитии методов сэмпирования стала работа немецкого исследователя Андреаса Дрексла [8], который, с одной стороны, нашел способ удачным образом обобщить методы смещенного сэмпирования, а с другой стороны, ввести два важных параметра, которые будут порождать целые классы новых методов. Он предложил использовать корректировки для значений приоритетов, рассчитанных по фиксированному правилу, для того, чтобы избавиться от недостатков и ограничений, которые были присущи методам смещенного сэмпирования. Эти корректировки были достаточно универсальными, и поэтому метод Дрексла впоследствии был назван методом **смещенного сэмпирования на основе скорректированных приоритетов (*RBRSt*)**.

Схема определения вероятностей состоит из 3 этапов:

*Этап 1.* Первоначальная корректировка исходных значений приоритетов:

$z'_i = z_i - \min_{k \in S} \{z_k\}$  — в случае максимизации значений приоритета;

$z'_i = \max_{k \in S} \{z_k\} - z_i$  — в случае минимизации значений приоритета.

*Этап 2.* Последующая корректировка:

$$z''_i = (z'_i + \epsilon)^\alpha,$$

где  $\epsilon$  и  $\alpha$  — некоторые положительные параметры.

*Этап 3.* Вычисляем вероятности по формуле:

$$p_i = \frac{z''_i}{\sum_{k \in S} z''_k}$$

Параметр  $\epsilon$  позволяет дать возможность всем работам (включая те, у которых максимальное или минимальное значение приоритета), которые могут выполняться на данном этапе, положительную вероятность, т.е. возможность быть выбранными.

С помощью изменения параметра  $\alpha$  появляется возможность повлиять на степень различия вероятностей работ. Действительно, в случае  $\alpha = 0$  получаем  $z''_i = 1$  и равные вероятности выбора работ на каждом этапе, т.е. равномерное сэмпирование. В случае увеличения значения  $\alpha$  вероятность выбора наиболее приоритетной работы будет увеличиваться. В пределе будем иметь обычный конструктивный метод с детерминированным правилом приоритета. Таким образом, данный метод сэмпирования

<sup>1</sup> Normalized Biased Random Sampling.

<sup>2</sup> Regret-Based Biased Random Sampling.

рования является наиболее универсальным, включающим большинство предыдущих методов.

Рассмотрим пример, который продемонстрирует зависимость вероятностей от значений приоритетов и параметров сэмпирования. Допустим, используемое правило приоритета – минимальный полный резерв работы (*SLK*). Допустим, что на некотором этапе нам необходимо выбрать работу из некоторого множества работ-претендентов (удовлетворяющих всем ограничениям), работы которого имеют следующие значения резерва: 0, 1, 4. Так как больший приоритет получает работа с наименьшим значением резерва, то скорректированные значения приоритетов будут равны:

$$z'_0 = 4 - 0 = 4; \quad z'_1 = 4 - 1 = 3; \quad z'_2 = 4 - 4 = 0,$$

где 4 – максимум из первоначальных значений приоритетов.

Допустим, что  $\varepsilon = 1$ . Рассчитаем вероятности выбора этих работ при разных значениях  $\alpha$  (рис. 2).

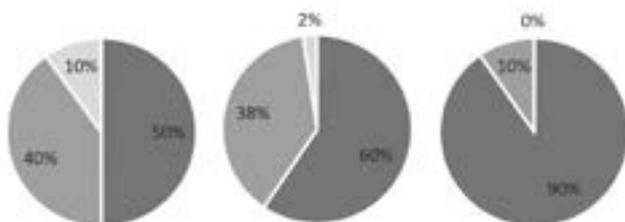


Рис. 2. Вероятности выбора работ при разных значениях  $\alpha$

В работе [19] приводится пример, показывающий недостатки использования одного и того же значения  $\varepsilon$  для всех этапов применения схемы построения расписания. Допустим, на первом этапе необходимо выбрать из 2-х работ, которые имеют значения приоритетов 1 и 9, а на втором этапе 10 и 90. Рассчитаем в обоих случаях вероятности, с которыми эти работы будут выбираться для  $\varepsilon = 1$  (табл. 1).

Таблица 1

Влияние  $\varepsilon$  на распределение вероятностей в зависимости от величины приоритетов

1-й этап	Приоритет	$z'$	$z''$	$p$
1-я работа	1	0	1	10%
2-я работа	9	8	9	90%
2-й этап	Приоритет	$z'$	$z''$	$p$
1-я работа	10	0	1	1%
2-я работа	90	80	81	99%

Для того чтобы избавиться от этого недостатка, в работе [19] предлагается рассчитывать  $\varepsilon$  динамически, т.е. на каждом этапе выполнения схемы построения расписания в зависимости от значений

приоритетов работ-претендентов следующим образом:

$$\varepsilon = \left\{ \frac{\min \text{положительное значение приоритета}}{\delta}, \right.$$

если существуют положит. знач. приор. 1, иначе).

Здесь  $\delta$  – параметр (положительное число), который обычно выбирается равным: 1, 10, 100.

Такой метод получил название *MRBRS*/ $\delta$  (*Modified RBRS*) – модифицированный метод смещенного сэмпирования на основе скорректированных приоритетов с параметром  $\delta$ .

На каждом шаге применения последовательной или параллельной схемы построения расписания  $\varepsilon$  может существенно отличаться, так как при его вычислении в числителе берется минимальное значение приоритета только из работ-претендентов.

Тщательное исследование влияния различных правил приоритета, схем формирования расписаний и различных параметров методов сэмпирования было предпринято немецкими исследователями Ширмером и Райзенбергом, авторами *MRBRS* [19]. Были получены следующие результаты:

- в методе *RBRS* с последовательной схемой лучше использовать  $\alpha = 2$ , а для параллельной  $\alpha = 1$ ;
- в методе *MRBRS*/1 с любой схемой лучше использовать  $\alpha = 2$ ;
- в методе *MRBRS*/10 с любой схемой лучше использовать  $\alpha = 1$ ;
- наиболее эффективными методами являются *RBRS* и *MRBRS*;
- с последовательной схемой лучше всего работает правило *LST* (минимальное позднее начало работы, вычисленное по методу критического пути);
- с параллельной схемой лучше всего работает правило *WCS* (см. выше);
- параллельная схема уступает последовательной при большом количестве итераций в сэмпировании.

#### 4. Сэмпирование с адаптивной схемой управления

Одними из самых совершенных методов сэмпирования являются рассмотренные выше двухпараметрические *RBRS* (зависящий от  $\alpha$  и  $\varepsilon$ ) и *MRBRS* (зависящий от  $\alpha$  и  $\delta$ ). Первая попытка их улучшения была связана с одним вопросом: можно ли создать метод, который бы на основе характеристик проекта сам бы определял наиболее эффективные значения для обоих параметров?

Такое исследование было проведено в 1996 г. для метода *RBRS* [11]. В качестве характеристик проекта были выбраны [14]:

- $Z$  – количество итераций при использовании метода сэмплирования;
- $NC$  (*network complexity*) – отношение количества связей между работами к общему количеству работ;
- $RF$  (*resource factor*) – среднее количество разных типов ресурсов, назначенных на одну работу;
- $RS$  (*resource strength*) – коэффициент сложности ресурсных конфликтов.

Были получены четкие закономерности:

- 1) чем больше число итераций  $Z$ , тем больше стоит сокращать  $\alpha$  (в оригинале от бесконечности до 1 при увеличении  $Z$  от 1 до 100). При  $Z \geq 50$  стоит выбирать  $\alpha = 1$ ;
- 2) при  $Z < 10$  стоит использовать параллельную схему формирования расписания, а при  $Z > 10$  – последовательную. При этом точка пересечения (10 в данном случае) зависит от  $RF$ .

Исходя из полученных результатов, был предложен метод адаптивного поиска *ASP*<sup>1</sup>, который в зависимости от  $Z$  подбирал  $\alpha$ , и в зависимости от  $RF$  проекта выбирал схему формирования расписания:

- если  $RF \leq 0,75$ , то используем параллельную схему при  $Z \leq 10$  и последовательную при  $Z > 10$ ;
- если  $RF > 0,75$ , то всегда используем параллельную схему.

Выбор альфы:

$Z$	1	(1;5]	(5;10]	> 10
$\alpha$	$\infty$	3	2	1

В работе Ширмера [17] было предложено использовать самообучающийся механизм определения наилучших параметров *CBR* (*Case-Based Reasoning*) для методов сэмплирования. В результате его применения были выявлены интервалы параметров *RBRS* и *MRBRS* и их соответствие классам проектов (по  $RF$ ,  $RS$ ,  $Z$  и количеству работ в проекте), что позволило усовершенствовать метод *ASP*, описанный выше. Кроме того, были обнаружены следующие интересные эффекты, отчасти реабилитировавшие использование параллельных схем даже при большом числе итераций:

- увеличение количества работ в проекте ведет к повышению эффективности параллельной схемы;
- чем сложнее задача с точки зрения ресурсных конфликтов ( $RS$  и  $RF$ ), тем эффективнее параллельные методы.

В работе [20] было предложено в использовании сэмплирования применять прямую и обратную схемы формирования расписания. При этом получается, что изменяется также и схема формирования расписания. Именно такой подход дал наилучшие результаты. Суть состоит в том, что на каждой итерации мы осуществляем несколько построений расписания:

- используем обратную параллельную схему с правилом приоритета *LFT*, попутно формируя порядок следования работ для следующего этапа;
- используем прямую последовательную схему, в которой порядок работ был определен на предыдущем этапе;
- применяем обратную последовательную схему, в которой порядок работ остался таким же, как и на предыдущих этапах.

Утверждается, что благодаря сохранению порядка работ, который определяется всего один раз, три прохода проекта на каждой итерации с точки зрения вычислительной сложности можно считать за два. Экспериментальная проверка показала, что такой метод дает наилучшие результаты из всех методов сэмплирования и дает нам основание считать, что имеет смысл продолжить исследования в этом направлении, комбинируя различные схемы построения расписаний и правила приоритета для разных итераций сэмплирования.

С начала XXI в. методы сэмплирования стали использоваться в сложных гибридных методах, сочетающих различные подходы к оптимизации продолжительности проекта в условиях ограниченных ресурсов (см., например, [18]). На сегодняшний день подобные методы являются наиболее эффективными.

## 5. Методология исследования

Целью нашего исследования является сравнительный анализ эффективности применения различных методов сэмплирования.

В данной работе будет использован массив проектов *J30* из базы *PSPLIB*<sup>2</sup> [15], сгенерированный с помощью программы *ProGen* [14; 7].

Все проекты множества *J30* состоят из 30 работ. Выполнение работ прерывать нельзя. Для выполнения каждой работы требуется от 1 до 4 типов возобновляемых ресурсов. Каждый ресурс имеет ограничение в каждом периоде выполнения проекта. Параметры каждого проекта определяются 48 комбинациями значений  $NC$ ,  $RF$ ,  $RS$ , указанных в табл. 2. Определения параметров рассмотрены выше.

<sup>1</sup> Adaptive Search Procedure.

<sup>2</sup> URL: <http://www.om-db.wi.tum.de/psplib/main.html/>

Таблица 2  
Значения параметров проектов множества *J30*

Параметр	Значения
<i>NC</i>	1.5; 1.8; 2.1
<i>RF</i>	0.25; 0.5; 0.75; 1.0
<i>RS</i>	0.2; 0.5; 0.7; 1.0

В *J30* для каждой из 48 комбинаций параметров содержится 10 проектов. Проекты, у которых  $RS = 1$ , не представляют содержательных задач, так как у них практически отсутствуют ресурсные конфликты, и их оптимальное решение получается сразу при расчете ранних сроков. Таких проектов 120, и мы их исключаем из статистики. Таким образом, исследования будут проводиться на 360 проектах.

Для каждого проекта создателями базы была вычислена оптимальная продолжительность, с которой мы будем сравнивать продолжительность, которая у нас получится после применения того или иного метода. Будем рассчитывать отклонение в процентах от оптимальной продолжительности.

Все вычисления проводились на компьютере с 4-ядерным процессором *Intel Core i5 3.4 GHz*, под управлением *Windows 8.1* в программе, написанной на языке *C#*.

## 6. Результаты исследований

Как было рассмотрено выше, одним из ключевых преимуществ методов сэмпирования, по сравнению с однопроходными конструктивными, является их способность генерировать новые расписания, которые в пределе охватывают все возможные расписания (при использовании последовательной схемы). Построим график зависимости среднего отклонения

от оптимумов по массиву *J30* в зависимости от количества итераций при использовании методов: а) равномерного сэмпирования (*RAS*); б) смещенного сэмпирования Купера (*BRS-C*); в) смещенного сэмпирования на основе скорректированных приоритетов (*RBRS*); г) модифицированного *RBRS* (*MRBRS/10*). В качестве базовой схемы формирования расписания рассмотрим прямую последовательную схему с правилом приоритета *LST*. В этом исследовании метод *RAS* играет очень важную роль, так как отражает самый неэффективный метод поиска оптимального решения. Величина отклонения от *RAS* покажет качество и целесообразность применяемых правил (рис. 3).

Видно, что наиболее качественными методами из примененных выше являются *RBRS* и *MRBRS/10*. Стоит отметить, что данный результат будет таким же при использовании других схем и правил приоритета, а также при использовании других множеств проектов (*J60*, *J90* и *J120*). Важным моментом является быстро затухающая скорость приближения к оптимальному решению: в первые несколько итераций удастся очень существенно сократить продолжительность проекта, но чем больше итераций, тем трудней становится улучшить решения.

С использованием метода *RBRS* при количестве итераций, равном 100 тыс., удалось достичь среднего отклонения 0,32%, при этом оптимальное решение было найдено для 310 проектов из 360 (86%). За следующие 100 тыс. итераций удалось найти еще 6 оптимальных решений и сократить среднее отклонение до 0,27%. За 500 тыс. итераций количество найденных оптимальных решений составило 324 при среднем отклонении от оптимальных решений

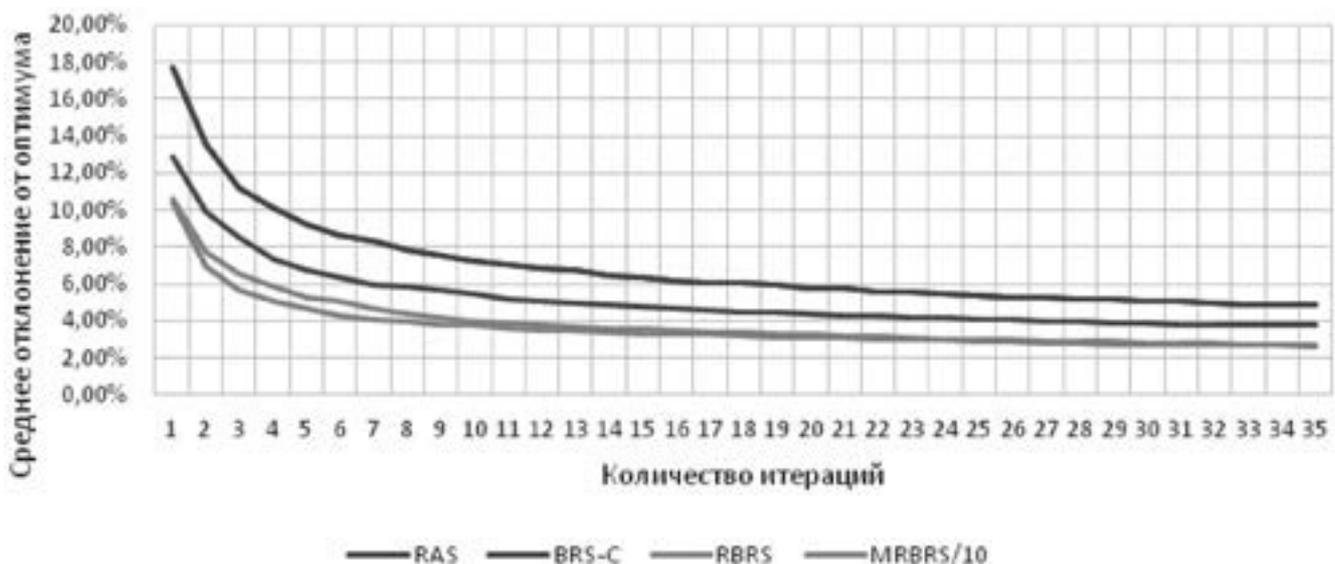


Рис. 3. Среднее отклонение от оптимальных решений разных методов сэмпирования (*Serial LST*)

0,20%. Скорость сходимости к оптимальному решению для *RBRS* можно увидеть на рис. 4. Видно, что эффективный диапазон итераций в данном случае составляет не более 60. Больше количество итераций в среднем приводит к ничтожным улучшениям расписания, которые практически не имеют значения для менеджера проекта.

Другой важный тест касается использования различных схем построения расписания. При этом с последовательной схемой использовалось одно из лучших правил для данной схемы: *LST*. Для параллельной схемы – одно из лучших именно для параллельных схем *WCS* (см. выше). Результаты представлены на рис. 5.

Важным результатом является более качественная работа параллельной схемы при небольшом количестве итераций и более качественная работа

последовательной схемы при большом количестве итераций. Этот эффект обычно объясняют тем, что пространство допустимых решений, получаемых с помощью параллельной схемы, уже пространства решений последовательной схемы. Более того, оптимальное решение может не содержаться в пространстве решений, полученных параллельной схемой. Однако такая картина существенно зависит от величины *RF* проектов. При больших значениях *RF*, параллельная схема оказывается эффективнее во всем диапазоне значений *Z*.

Оценим также эффективность использования правил приоритета *SLK* и *WCS* вместе с параллельной схемой построения расписания в методе *RBRS* (рис. 6).

Интересным фактом является высокое качество правила *WCS*, которое также является лучшим для

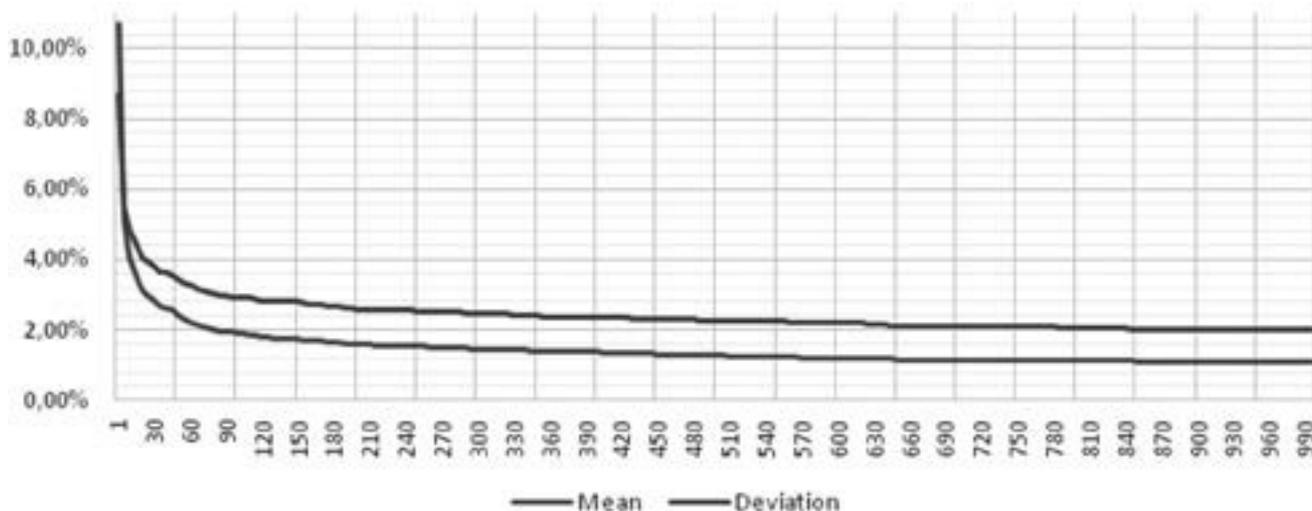


Рис. 4. Зависимость среднего отклонения и СКО от количества итераций для *RBRS*

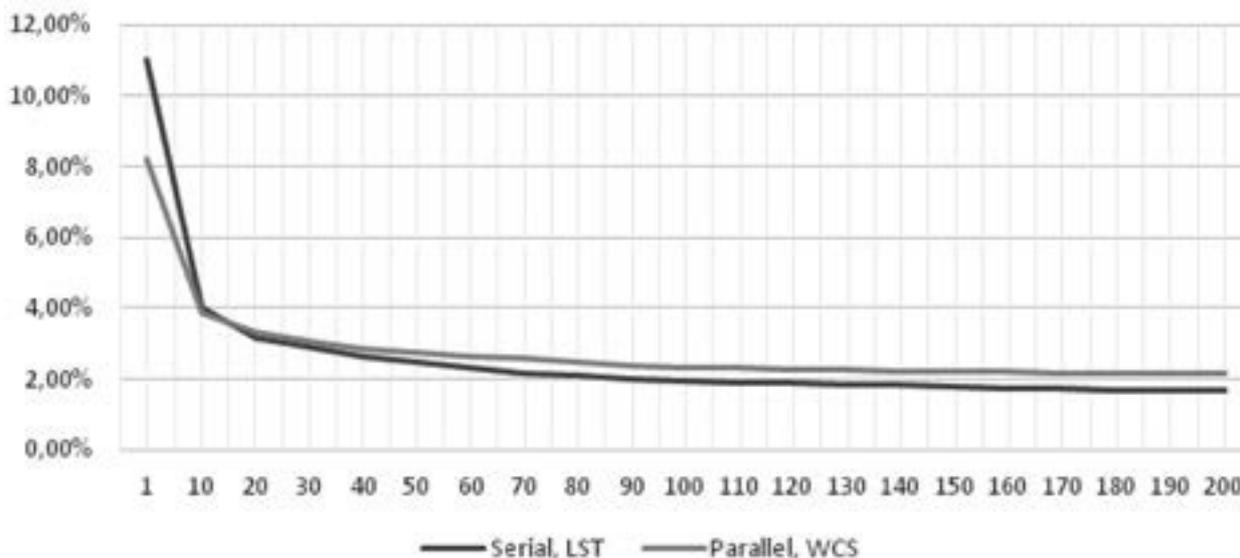


Рис. 5. Влияние схемы построения расписания (*RBRS*)

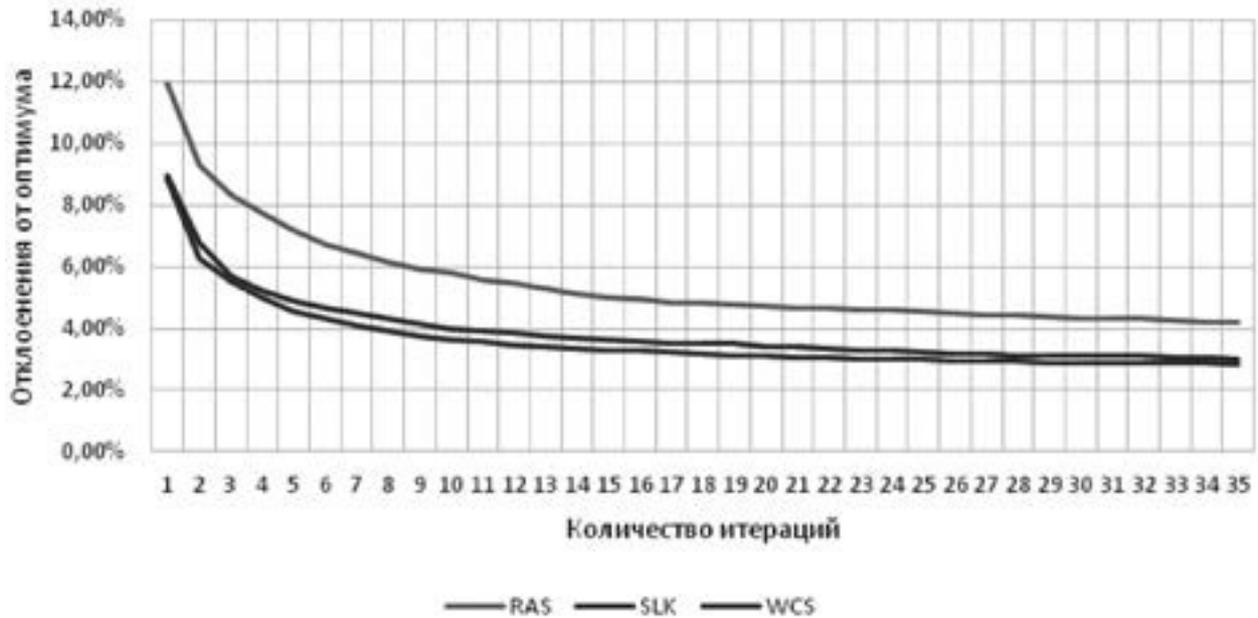


Рис. 6. Эффективность *RBRS* в зависимости от правила приоритета для параллельной схемы

однопроходных конструктивных методов при использовании параллельной схемы. При этом разница *WCS* и *SLK* не существенна, но *WCS* требует значительно больших вычислительных затрат. Также можно сказать, что правила приоритета, которые дают плохие результаты для однопроходных конструктивных, также плохо работают и в паре с методами сэмпирования.

В работе метода сэмпирования участвует схема построения расписания с некоторым правилом приоритета. Наша гипотеза состоит в том, что можно существенно повысить результативность сэмпирования, путем применения на разных его итерациях разных базовых правил приоритета и схем формирования расписаний. Для ее проверки построим систему, которая отберет для нас наилучшие правила приоритета и схемы на различных итерациях.

Будем рассматривать 46 комбинаций правил приоритета и схем, представленных в табл. 3. Подробное описание этих правил можно найти в [2].

Таблица 3

Комбинации правил приоритета и схем формирования расписаний

Priority Rule	Serial, Forward	Serial, Backward	Parallel, Forward	Parallel, Backward
WCS			X	X
IRSM			X	X
ACS			X	X
SLK	X	X	X	X
LST	X	X	X	X
LFT	X	X	X	X

Окончание табл. 3

Priority Rule	Serial, Forward	Serial, Backward	Parallel, Forward	Parallel, Backward
EST	X	X	X	X
EFT	X	X	X	X
SPT	X	X	X	X
LPT	X	X	X	X
MTS	X	X	X	X
GRPW	X	X	X	X
RED	X	X	X	X

На каждой итерации сэмпирования с использованием метода *RBRS* будем определять улучшения расписаний для всего массива проектов *J30* по каждому из 46 методов. В результате максимальную скорость снижения средней продолжительности проектов *J30*, которая только может быть получена, путем комбинаций представленных в табл. 3 схем и правил приоритета.

На рис. 7 представлены результаты работы такой системы. В левой части отображены схемы и правила приоритета, примененные на разных итерациях, в правой — зависимость отклонений от оптимальных решений от количества итераций (*Compound*). Неожиданным результатом стало отсутствие в списке примененных схем правил, которые плохо работают в однопроходных конструктивных методах (*SPT*, *LPT*, *RED* и т.д.). Также стоит отметить небольшое количество использованных обратных схем формирования расписаний. Неожиданным также стало то, что после 60 итераций все последующие

Схемы на итерациях:

1. *FW, Parallel, LFT*
2. *FW, Serial, LFT*
3. *BW, Parallel, SLK*
4. *FW, Parallel, MTS*
5. *FW, Parallel, SLK*
6. *FW, Serial, LFT*
7. *FW, Parallel, LST*
8. *BW, Parallel, SLK*
9. *FW, Parallel, MTS*
10. *FW, Parallel, LST*

...

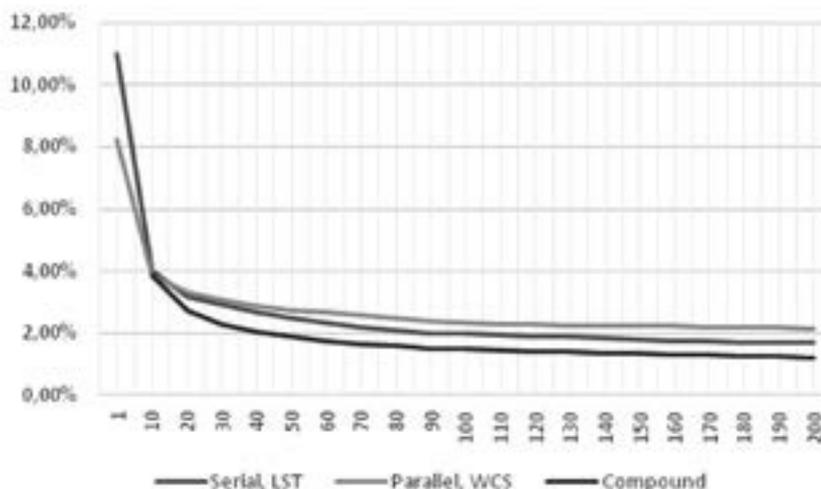


Рис. 7. Эффективность сэмпирования *RBRS* с изменяющимися правилами формирования расписаний

правила приоритета были только *WCS*. Если исключить *WCS*, *IRSM* и *ACS* из списка правил, которые можно применять вместе с сэмпированием, то при количестве итераций больше 78 мы встречаем только правило *RND* — случайный выбор приоритетов. Это означает, что преимущества использования различных правил проявляются только на первых 60 итерациях. Важным результатом также является то, что при больших значениях итераций параллельная схема оказывается эффективнее последовательной.

В методе построения списка схем и правил приоритета, примененном выше, отбор правил и схем осуществлялся на основе конкретных расписаний, полученных с помощью датчика случайных чисел, на основе перечисленных выше правил. Другими словами, остается возможность того, что если зафиксировать использование полученных схем и правил и повторить сэмпирование с этими фиксированными правилами еще раз, то результаты окажутся значительно хуже. Такой эксперимент был нами проведен. В качестве базовых методов сэмпирования циклически использовалось 10 схем, представленных на рис. 7. Результаты оказались примерно такими же, существенно превышающими результаты сэмпирования с использованием одного метода (даже самого лучшего).

### Заключение

Методы сэмпирования являются естественным развитием конструктивных однопроходных правил приоритета, позволяющих существенно улучшить характеристики проекта за первые 60 итераций. В результате исследований эффективности этих методов было выявлено:

- 1) после 60 итераций использование сложных методов сэмпирования мало отличается от случайной генерации допустимых расписаний;

- 2) параллельная схема более эффективнее последовательной даже при больших значениях количества итераций;
- 3) следует комбинировать последовательные и параллельные схемы с различными правилами приоритета на первом этапе применения сэмпирования — это позволяет существенно улучшить общие показатели метода;
- 4) существует лишь ограниченный набор лучших правил приоритета (*WCS*, *SLK*, *MTS*, *LST*, *LFT*), другие правила использовать нецелесообразно;
- 5) правило приоритета *WCS* (с параллельной схемой) является самым эффективным правилом приоритета для метода *RBRS*.

В качестве направлений для дальнейших исследований сформулируем гипотезу: изменение параметра «альфа» метода *RBRS* в зависимости от итерации, позволит еще больше улучшить показатели сэмпирования.

### Литература

1. Царьков И.Н. Исследование эффективности методов оптимизации проекта с ограниченными ресурсами. Ч. 1 // Научные исследования и разработки. Российский журнал управления проектами. 2013. Т. 2. № 3. С. 13–25. DOI: 10.12737/1240.
2. Царьков И.Н. Исследование эффективности методов оптимизации проекта с ограниченными ресурсами. Ч. 2 // Научные исследования и разработки. Российский журнал управления проектами. 2013. Т. 2. № 4. С. 3–13. DOI: 10.12737/1958.
3. Alvarez-Valdés, Tamarit. Chapter 5 - Heuristic Algorithms for Resource-Constrained Project Scheduling: A Review and an Empirical Analysis // Advances in Project Scheduling Studies in Production and Engineering Economics / Под ред. R. Słowiński, J. Węglarz. Oxford: Elsevier, 1989. P. 113–134.
4. Błazewicz J. Scheduling Subject to Resource Constraints: Classification and Complexity. Brussels: European Institute for Advanced Studies in Management, 1980.

5. Cooper D.F. Heuristics for Scheduling Resource-Constrained Projects: An Experimental Investigation // *Manag. Sci.* 1976. T. 22. № 11. P. 1186–1194.
6. Davis E.W., Patterson J.H. A comparison of heuristic and optimum solutions in resource-constrained project scheduling // *Manag. Sci.* 1975. T. 21. № 8. P. 944–955.
7. Drexl A. и др. ProGen/πx – An instance generator for resource-constrained project scheduling problems with partially renewable resources and further extensions // *Eur. J. Oper. Res.* 2000. T. 125. № 1. P. 59–72.
8. Drexl A. Scheduling of Project Networks by Job Assignment // *Manag. Sci.* 1991. T. 37. № 12. C. 1590–1602.
9. Kelley J.E. Jr, Walker M.R. Critical-path Planning and Scheduling // Papers Presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference IRE-AIEE-ACM '59 (Eastern). New York, NY, USA: ACM, 1959. C. 160–173.
10. King G.W. The Monte Carlo Method as a Natural Mode of Expression in Operations Research // *J. Oper. Res. Soc. Am.* 1953. T. 1. № 2. P. 46–51.
11. Kolisch R., Drexl A. Adaptive search for solving hard project scheduling problems // *Nav. Res. Logist. NRL.* 1996. T. 43. № 1. P. 23–40.
12. Kolisch R. Efficient priority rules for the resource-constrained project scheduling problem // *J. Oper. Manag.* 1996a. T. 14. № 3. P. 179–192.
13. Kolisch R. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation // *Eur. J. Oper. Res.* 1996b. T. 90. № 2. P. 320–333.
14. Kolisch R., Sprecher A., Drexl A. Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems // *Instituten für Betriebswirtschaftslehre der Universität at Kiel*, 1992.
15. Kolisch R., Sprecher A. PSPLIB - A project scheduling problem library: OR Software - ORSEP Operations Research Software Exchange Program // *Eur. J. Oper. Res.* 1996. T. 96. № 1. P. 205–216.
16. Pritsker A.A.B., Watters L.J. A zero-one programming approach to scheduling with limited resources. Santa Monica, Calif.: Rand Corp., 1968.
17. Schirmer A. Case-Based Reasoning and Improved Adaptive Search for Project Scheduling. University of Kiel, Germany, 1998. Вып. Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel.
18. Schirmer A. Resource-constrained project scheduling: An evaluation of adaptive control schemes for parameterized sampling heuristics // *Int. J. Prod. Res.* 2001. T. 39. № 7. C. 1343–1365.
19. Schirmer A., Riesenberger S. Parameterized Heuristics for Project Scheduling - Biased Random Sampling Methods. University of Kiel, Germany, 1997. Вып. Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel.
20. Tormos P., Lova A. An efficient multi-pass heuristic for project scheduling with constrained resources // *Int. J. Prod. Res.* 2003. T. 41. № 5. C. 1071–1086.
3. Alvarez-Valdés, Tamarit. Chapter 5 – Heuristic Algorithms for Resource-Constrained Project Scheduling: A Review and an Empirical Analysis. *Advances in Project Scheduling Studies in Production and Engineering Economics*. Oxford: Elsevier, 1989, pp. 113–134.
4. Błazewicz J. Scheduling Subject to Resource Constraints: Classification and Complexity. Brussels: European Institute for Advanced Studies in Management, 1980.
5. Cooper D.F. Heuristics for Scheduling Resource-Constrained Projects: An Experimental Investigation // *Manag. Sci.* 1976. V. 22. I. 11. pp. 1186–1194.
6. Davis E.W., Patterson J.H. A comparison of heuristic and optimum solutions in resource-constrained project scheduling // *Manag. Sci.* 1975. V. 21. I. 8. pp. 944–955.
7. Drexl A. idr. ProGen/πx – An instance generator for resource-constrained project scheduling problems with partially renewable resources and further extensions // *Eur. J. Oper. Res.* 2000. V. 125. I. 1. pp. 59–72.
8. Drexl A. Scheduling of Project Networks by Job Assignment // *Manag. Sci.* 1991. V. 37. I. 12. pp. 1590–1602.
9. Kelley J.E. Jr, Walker M.R. Critical-path Planning and Scheduling // Papers Presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference IRE-AIEE-ACM '59 (Eastern). New York, NY, USA: ACM, 1959. pp. 160–173.
10. King G.W. The Monte Carlo Method as a Natural Mode of Expression in Operations Research // *J. Oper. Res. Soc. Am.* 1953. V. 1. I 2. pp. 46–51.
11. Kolisch R., Drexl A. Adaptive search for solving hard project scheduling problems // *Nav. Res. Logist. NRL.* 1996. V. 43. I. 1. pp. 23–40.
12. Kolisch R. Efficient priority rules for the resource-constrained project scheduling problem // *J. Oper. Manag.* 1996a. V. 14. I. 3. pp. 179–192.
13. Kolisch R. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation // *Eur. J. Oper. Res.* 1996b. V. 90. I. 2. pp. 320–333.
14. Kolisch R., Sprecher A., Drexl A. Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems. *Instituten für Betriebswirtschaftslehre der Universität Kiel*, 1992.
15. Kolisch R., Sprecher A. PSPLIB - A project scheduling problem library: OR Software - ORSEP Operations Research Software Exchange Program // *Eur. J. Oper. Res.* 1996. V. 96. I. 1. pp. 205–216.
16. Pritsker A.A.B., Watters L.J. A zero-one programming approach to scheduling with limited resources. Santa Monica, Calif.: Rand Corp., 1968.
17. Schirmer A. Case-Based Reasoning and Improved Adaptive Search for Project Scheduling. University of Kiel, Germany, 1998. I. Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel.
18. Schirmer A. Resource-constrained project scheduling: An evaluation of adaptive control schemes for parameterized sampling heuristics // *Int. J. Prod. Res.* 2001. V. 39. I. 7, pp. 1343–1365.
19. Schirmer A., Riesenberger S. Parameterized Heuristics for Project Scheduling - Biased Random Sampling Methods. University of Kiel, Germany, 1997. Vyp. Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel.
20. Tormos P., Lova A. An efficient multi-pass heuristic for project scheduling with constrained resources // *Int. J. Prod. Res.* 2003. V. 41. I. 5. pp. 1071–1086.

## References

1. Tsar'kov I.N. Issledovanie effektivnosti metodov optimizatsii proekta s ogranichennymi resursami. Chast' 1 [Evaluation of Methods for Optimizing a Resource-Constrained Project. Part 1]. *Nauchnye issledovaniya i razrabotki. Rossiyskiy Zhurnal Upravleniya Proektami* [Research and Development. Russian Journal of Project Management]. 2013, V. 2. I. 3, pp. 13–25. DOI: 10.12737/1240
2. Tsar'kov I.N. Issledovanie effektivnosti metodov optimizatsii proekta s ogranichennymi resursami. Chast' 2 [Evaluation of Methods for Optimizing a Resource-Constrained