

DOI: 10.17323/2587-814X.2026.1.67.85

Обзор и сравнительный анализ новых методов распределения задач между исполнителями

Тимофей Яковлевич Шевгунов^{a,b} 

E-mail: shevgunov@gmail.com

Анна Александровна Крошилина^b

E-mail: ankrosh@vk.com

^a Московский авиационный институт (национальный исследовательский университет), Москва, Россия

^b Высшая школа бизнеса, Национальный исследовательский университет «Высшая школа экономики», Москва, Россия

Аннотация

В статье рассмотрено современное состояние и представлены результаты анализа новых достижений в проблемной области автоматизированного распределения задач между исполнителями. Целью работы является выявление тенденций и закономерностей в развитии существующих методов распределения задач, определение их преимуществ и ограничений, а также обоснование необходимости разработки новых подходов и алгоритмов, обеспечивающих повышение эффективности делегирования задач исполнителям. С использованием унифицированной системы обозначений основных понятий предметной области исследования в статье проведен краткий описательный обзор десяти универсальных алгоритмов распределения задач, опубликованных за последние 20 лет. Представлены результаты сравнительного анализа, проведенного по ряду критериев, отражающих как технические, так и организационно-поведенческие аспекты функционирования алгоритмов распределения задач между исполнителями. В качестве ключевых критериев оценки рассматривались уровень учета компетенций исполнителей, адаптивность к изменениям внешних условий и состава команды, требования к полноте и структуре исходных данных, устойчивость к неполным или шумным данным, прозрачность и объяснимость принимаемых решений, вычислительная сложность, масштабируемость при увеличении числа задач и исполнителей, стоимость внедрения и сопровождения, а также ориентация на развитие персонала и совершенствование его компетенций. Проведенный сравнительный анализ позволил выявить преимущества и недостатки каждого из рассмотренных методов и позволил выработать рекомендации по наиболее продуктивному исполь-

зованию каждого из них. Анализ показал, что ни один из рассмотренных алгоритмов не может быть выделен как единый универсальный инструмент делегирования. Следует отметить, что комплексная информация о пригодности исполнителя для решения задач, требующих различных компетенций, частью алгоритмов игнорируется, а другими используется недостаточно эффективно. Последнее обосновывает открытость проблемы выработки новых подходов к распределению задач и потребность в разработке новых алгоритмов на их основе.

Ключевые слова: распределение задач, назначение исполнителей, матрица назначений, алгоритмы делегирования, round robin, фронтальный алгоритм, генетический алгоритм

Цитирование: Шевгунов Т. Я., Крошила А. А. Обзор и сравнительный анализ новых методов распределения задач между исполнителями // Бизнес-информатика. 2026. Т. 20. № 1. С. 67–85. <https://doi.org/10.17323/2587-814X.2026.1.67.85>

Введение

Одним из факторов, обуславливающих достижение высокой производительности внутренних рабочих процессов любой организации вне зависимости от ее организационно-правовой формы, является эффективность планирования и распределения задач при непосредственной реализации бизнес-процессов. Увеличение общей сложности проектов или функционирование в условиях изменяющейся среды неизбежно требуют от менеджмента организаций или их структурных единиц более точного планирования и распределения имеющихся в их распоряжении ресурсов, что является одним из важных условий, способствующих быстрой адаптации бизнес-процессов компании. При этом ограниченность трудовых ресурсов, доступных компании в краткосрочном периоде, делает важность решения проблемы их оптимального использования в течение как можно более продолжительных интервалов рабочего времени. Кроме того, существует тенденция увеличения риска принятия ошибочных решений [1] при ручном распределении задач в условиях потоков работ, близких по интенсивности к границе производственных возможностей компании. Такой риск дополнительно возрастает в тех случаях, когда задачи существенно по своим характеристикам, являются редкими или новыми, с решением которых исполнители сталкиваются впервые.

При организации процесса распределения задач в организациях типично выявляются следующие три ключевые проблемы. Первая проблема состоит в неравномерной загрузке сотрудников, приводящей к фактическим простоям или перегрузкам отдельных сотрудников или их групп. Вторая проблема заключается в отсутствии систематического механизма учета компетенций исполнителей и проверки их соответствия минимальным требованиям, удовлетворение которых позволит эффективное выполнение назначенных задач. Третья проблема проявляется в субъективности менеджеров при распределении задач, что приводит к снижению мотивации сотрудников и служит потенциальным источником нарушения деловой этики или норм, сложившихся в рамках корпоративной культуры организации. Одним из характерных признаков того, что распределение задач в организации или ее подразделениях проводится неэффективно, является систематическое нарушение исполнителями установленных сроков выполнения задач, проявляющееся в том, что не менее четверти всех назначенных задач не выполняется в срок по любым причинам независимо от того, каким именно исполнителям они были назначены. Другим характерным признаком является систематическое дублирование назначений, которое может проявляться как в назначении одной и той же задачи двум независимым исполнителям, так и в неполном делегировании задачи, при выполнении кото-

рой требуется непрерывное персональное участие менеджера для контроля ее сроков или качества.

Проведенный анализ отечественных и зарубежных публикаций с 2005 по 2025 гг., направленный на выявление новых и оригинальных методов распределения задач между исполнителями, позволил установить, что, несмотря на то что данное направление исследований не входит в число приоритетных, разработка новых методов, алгоритмов и методик делегирования продолжают оставаться предметом научных и практических изысканий отдельных научных групп. Актуальность продолжающихся исследований методов распределения задач, ключевым направлением которого является разработка универсальных алгоритмов и методик, обусловлена необходимостью формирования устойчивых механизмов оперативного перераспределения задач, способных обеспечить баланс между контролем и автономностью исполнителей.

Детальный анализ источников позволил выявить ряд новых, заслуживающих упоминания, но довольно узкоспециализированных решений, например, предназначенных для облачных дата-центров [2], распределенных вычислительных систем [3], систем поддержки разработчиков программного обеспечения [4] или систем управления многоцелевыми аэромобильными системами [5]. Алгоритмы, структурно сходные с алгоритмами распределения задач, также используются при решении задач распределения исследований между академическими школами [6], подбора научно-методической информации [7, 8], распределения потока кредитных заявок для коммерческих банков [9]. Отдельного внимания заслуживает анализ изменений, вносимых в сложившиеся процедуры делегирования в условиях обязательного использования дистанционных технологий управления [10]. Дополнительно следует отметить, что в зарубежной практике отдельные решения, направленные на повышение эффективности распределения задач [11], могут рассматриваться как результаты интеллектуальной деятельности и получать правовую охрану патентами.

Однако основное внимание авторов при анализе публикаций было направлено на выявление новых универсальных методов, которые могут быть непосредственно применены или быстро адаптированы для распределения задач между исполнителями в организациях различной отраслевой принадлежности и формы собственности. В настоящей статье проведен обзор и сравнительный анализ десяти выявлен-

ных универсальных методов, предназначенных для автоматизации распределения задач между исполнителями для компаний или их подразделений. В составленный перечень авторы включили методы, реализующие существенно различные идеи, лежащие в основе делегирования, и не добавляли в этот список варианты одних и тех же алгоритмов, незначительно отличающиеся от их основных версий. Для систематизации описаний различных методов предварительно, в разделе 1, была проведена базовая формализация обозначений для общих факторов. В разделе 2 приведены краткие содержательные описания, достаточные для составления у читателя общего представления о каждом из десяти обозреваемых алгоритмах. Формальные обозначения, используемые в описаниях некоторых из них, могут незначительно отличаться от обозначений, использованных в оригинальных публикациях. В разделе 3 проведен сравнительный анализ выбранных алгоритмов по ряду критериев, позволяющий сопоставить их собственные характеристики и определить потенциальные направления применения каждого из них. В заключении приводятся выводы по текущему обзору и рекомендации по дальнейшему развитию предметной области анализа.

1. Формализация проблемы распределения задач

Предметную область распределения задач между исполнителями можно формализовать с помощью следующих математических сущностей. Определим

$$T = \{t_n, n = \overline{1, N}\}$$

– непустое конечное множество задач (*Tasks*), которые необходимо решить;

$$E = \{e_i, i = \overline{1, I}\}$$

– непустое конечное множество исполнителей (*Employees*);

$$C = \{c_m, m = \overline{1, M}\}$$

– непустое конечное множество компетенций (*Competencies*), необходимых исполнителям из E для решения задач из множества T .

Каждая задача $t \in T$ характеризуется такими показателями: $C(t)$ – множество компетенций, необходимых для ее выполнения, $p(t)$ – приоритет задачи t , выраженный с использованием порядковой шкалы, например, «низкий», «средний», «вы-

сокий», «критический», $s(t)$ – сложность задачи t , выраженная количественно в трудоемкости, или необходимым для выполнения задачи временем, например, в часах.

Каждый исполнитель $e \in E$ характеризуется такими показателями: $C(e)$ – множество компетенций необходимых для решения задач из множества T , которыми обладает исполнитель, $a(e)$ – доступное время для выполнения задач, выраженная в единицах времени, например, в часах, $w(e)$ – текущая загруженность исполнителя e в единицах времени.

Таким образом, решаемую задачу можно сформулировать следующим образом: необходимо осуществить распределение задач из множества T между исполнителями из множества исполнителей E таким образом, чтобы были учтены как характеристики задач (требуемые компетенции, приоритет, сложность), так и характеристики исполнителей (имеющиеся компетенции, доступное время, текущая загруженность). Это позволяет отнести решаемую задачу к классу задач оптимизации, где нужно минимизировать или максимизировать некоторую целевую функцию при выполнении системы ограничений. Для решения задачи удобно ввести бинарную функцию $x(t, e)$, принимающую значение 1, если задача t назначена исполнителю e , и 0 – в противном случае.

В зависимости от управленческих целей организация может по-разному определять, каким образом следует распределять задачи между исполнителями. Примерами ключевых ориентиров могут служить снижение риска перегрузки персонала, обеспечение равномерного распределения нагрузки между членами команды или максимально полное использование доступных ресурсов. Кроме того, в стратегической перспективе становится важным развитие компетенций сотрудников, позволяющее формировать более гибкую и устойчивую структуру компании. Таким образом, выбор критерия оптимальности определяется не только текущими производственными потребностями, но и долгосрочными приоритетами бизнеса: эффективность операционной деятельности, снижение затрат, повышение устойчивости к внешним рискам и формирование кадрового потенциала. Ниже рассмотрены четыре целевые функции, которые в наибольшей степени отражают данные организационные задачи.

1.1. Минимизация нагрузки на сотрудников с высокой текущей загруженностью

Данный критерий учитывает показатель доступного времени для выполнения задач у каждого сотрудника $a(e)$, а также сложность задачи $s(t)$ и ее приоритет $p(t)$. Тогда целевая функция F_1 имеет вид:

$$F_1 = \sum_{e \in E} \sum_{t \in T} x(t, e) (a(e) - s(t)) \rightarrow \max, \quad (1)$$

процесс оптимизации которой состоит в поиске бинарных функций, обеспечивающих распределение задач сотрудникам, у которых остается больше всего свободных часов, если им будет назначена распределяемая задача. Таким образом, назначение происходит более свободному сотруднику, то есть минимизируется нагрузка на занятых исполнителей. Такой подход позволяет перераспределять работу в пользу сотрудников с меньшей занятостью, снижая риск выгорания и перегрузки. Подобный подход активно применяется в организациях, где важно поддерживать устойчивую работоспособность команды, например, в колл-центрах или IT-службах поддержки, где равномерное распределение обращений снижает вероятность отказов в обслуживании.

1.2. Балансировка нагрузки между исполнителями

Задача заключается в минимизации диспропорций распределения трудоемкости задач между сотрудниками. Рассчитывается среднее значение \bar{a} по исполнителям

$$\bar{a} = \frac{1}{\#(E)} \sum_{e \in E} a(e), \quad (2)$$

которое сравнивается поочередно с каждым значением $a(e)$. Разница между этими значениями должна стремиться к минимуму:

$$F_2 = \sum_{e \in E} |a(e) - \bar{a}| \rightarrow \min. \quad (3)$$

Такой критерий обеспечивает равномерное распределение задач, что особенно важно в командах, где предполагается взаимозаменяемость исполнителей. Примером может служить распределение рабочих заданий в группе разработчиков программного обеспечения или среди консультантов проектной команды, где чрезмерная концентрация

задач у отдельных сотрудников приводит к риску срыва сроков.

На *рисунке 1* представлена обобщенная блок-схема типового процесса распределения задач между исполнителями в организационной системе, функционирующей в условиях нестационарного потока задач и неоднородной структуры пула исполнителей. Процесс начинается с поступления новых задач, формирующих динамический входной поток T . Для каждой задачи выполняется этап предварительного анализа, включающий определение типа задачи, требований к компетенциям $C(t)$, уровня приоритета $p(t)$ и срочности выполнения. На данном этапе формируется набор формализованных характеристик, используемых далее в алгоритмической процедуре распределения.

Параллельно с пулом задач рассматривается пул исполнителей E , характеризующийся индивидуальными наборами компетенций $C(e)$, текущей загрузкой $w(e)$, доступным ресурсом $a(e)$, возможными ограничениями. Особенностью рассматриваемой модели является наличие неравномерной нагрузки исполнителей, что отражено в блоке «Пул исполнителей» и является одним из ключевых факторов при принятии решения о назначении.

На следующем этапе выбирается метод или алгоритм распределения задач. В зависимости от организационных целей и структуры входных данных могут применяться различные подходы: циклическое распределение, методы комбинаторной оптимизации, эволюционные алгоритмы, фронталь-

ные методы, нейросетевые корректоры и др. Таким образом, блок «Алгоритм распределения» отражает вариативность возможных сценариев обработки потока задач. Далее осуществляется выбор исполнителя для конкретной задачи. Решение принимается с учетом совокупности факторов: текущей загрузки, соответствия компетенций требованиям задачи, приоритетности задания, стратегических целей, включающих в качестве примера выравнивание нагрузки или развитие компетенций персонала.

После назначения задачи происходит обновление параметров состояния системы, состоящее в отражении изменений в загрузке каждого исполнителя, что обеспечивает актуальность данных в условиях нестационарного потока задач, когда новые задачи поступают до завершения ранее назначенных. Таким образом, представленный на *рисунке 1* процесс носит итеративный характер и реализует замкнутый контур управления распределением задач, обеспечивая адаптацию к изменению состава задач, состояния исполнителей и выбранной стратегии назначения.

Рассмотренная на *рисунке 1* типовая схема распределения задач между исполнителями имеет межотраслевой характер и находит применение в широком спектре организационных систем. Ниже приведены примеры из различных сфер деятельности, демонстрирующие особенности множества задач T , множества исполнителей E , ограничений и целевых характеристик функционирования.

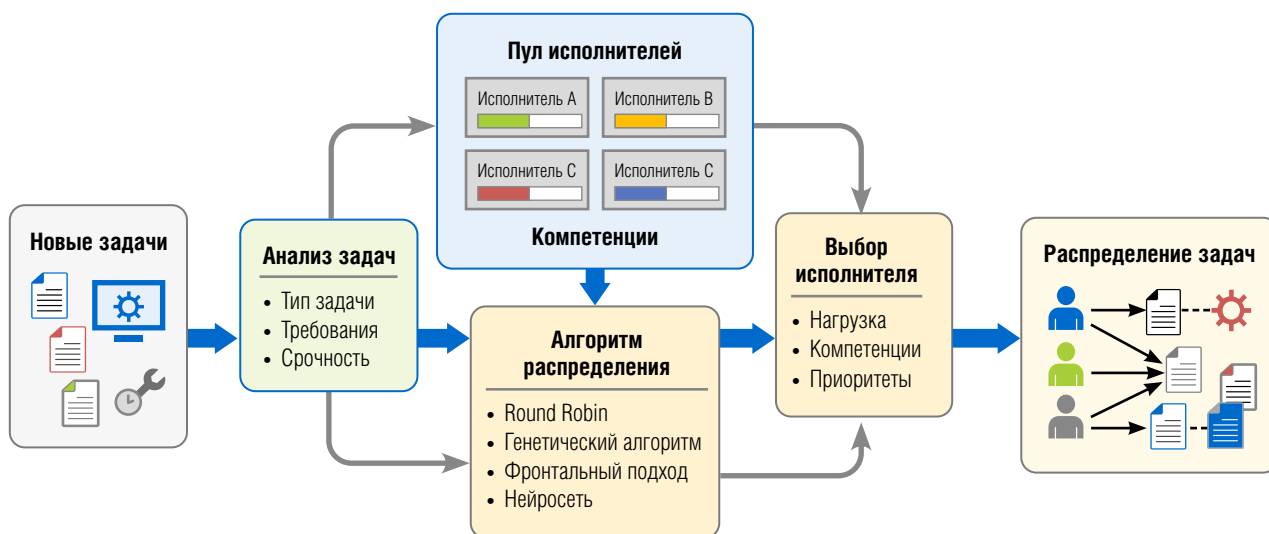


Рис. 1. Блок-схема процесса распределения задач между исполнителями.

В банковской сфере одной из типичных задач распределения является обработка кредитных заявок юридических и физических лиц. В данном случае множество T включает заявки различного уровня сложности и риска, а множество E – сотрудники подразделений андеррайтинга, риск-аналитики и комплаенса. Ограничениями выступают нормативные требования регулятора, внутренние регламенты банка, допустимые сроки рассмотрения заявок, а также лимиты загруженности специалистов. В качестве целевых бизнес-характеристик выступают среднее время рассмотрения заявки, доля одобренных заявок с допустимым уровнем риска, а также пропускная способность подразделения. Подобные задачи рассматриваются в исследованиях по оптимизации банковских бизнес-процессов и управлению операционными рисками [12].

В медицинских учреждениях распределение задач связано с назначением пациентов врачам соответствующей специализации, планированием диагностических процедур и управлением расписанием медицинского персонала [13]. В этом случае множество T включает приемы, операции и диагностические исследования, а множество E – медицинский персонал с различными компетенциями $C(e)$. Ограничениями являются нормативы времени ожидания, срочность медицинских случаев, доступность оборудования и сменный режим работы персонала. Целевые характеристики бизнеса включают суммарный поток обслуженных пациентов, среднее время ожидания и равномерность нагрузки врачей.

В строительной отрасли распределение задач возникает при планировании и координации работ на строительных объектах [14]. Множество задач T может включать подготовительные работы, монтаж конструкций, инженерные работы и контроль качества, тогда как множество E состоит из специализированных бригад и подрядных организаций. Ограничениями выступают технологическая последовательность выполнения работ, сроки поставки материалов, требования по технике безопасности и директивные сроки сдачи объекта. Целевыми показателями являются соблюдение календарного графика, минимизация простоев и снижение стоимости проекта.

В системе отправления правосудия распределение дел между судьями представляет собой задачу назначения с учетом специализации, сложности дела и текущей загрузки судейского корпуса [15].

В данном случае множество T состоит из дел различной категории, а элементами E являются судьи с различной квалификацией и специализацией. Ограничения включают процессуальные сроки рассмотрения дел, требования к равномерности нагрузки и принципы случайного распределения, закрепленные нормативными актами. Ключевыми целевыми характеристиками являются среднее время рассмотрения дела, соблюдение процессуальных сроков и балансировка нагрузки между судьями.

В сервисных организациях задачи распределения связаны с управлением очередью клиентов, распределением сервисных заявок и планированием использования оборудования [16]. Множество T формируется из поступающих заявок или обращений, а множество E – из специалистов различного профиля. Ограничениями являются стандарты обслуживания (SLA), уровень приоритета заявки, территориальная удаленность и доступность персонала. В качестве целевых характеристик рассматриваются среднее время обслуживания, процент выполненных заявок в нормативный срок и коэффициент загрузки персонала.

Приведенные примеры показывают, что формальная модель распределения задач, основанная на множествах T и E , функциях компетенций $C(t)$, $C(e)$ и бинарной функции назначения $x(t, e)$, является довольно универсальной и может быть применена к широкому спектру отраслей. При этом конкретные ограничения и целевые функции оптимизации могут существенно различаться, что может являться обоснованием дальнейшей специализации алгоритмов распределения задач, направленной на обеспечение отраслевых требований и учет характеристик входных потоков задач.

2. Алгоритмы распределения задач

На сегодняшний день существует множество подходов к распределению заданий между исполнителями, включающих как строгие методы математической оптимизации, так и эвристические алгоритмы и комбинированные схемы. Их выбор определяется характером задачи, размерностью исходных данных и требованиями к скорости получения решения. Подобные методы находят широкое применение в управлении проектами, логистике, сервисных компаниях и других областях, где

необходимо рационально использовать ресурсы и обеспечивать эффективное выполнение работ. В настоящем разделе проведен краткий обзор десяти универсальных алгоритмов распределения задач между исполнителями, опубликованных за последние 20 лет.

2.1. Алгоритм Round Robin для равномерного распределения заданий (АРРЗ) между исполнителями

Алгоритм *Round Robin* [17] является одним из типичных примеров АРРЗ, который можно описать как последовательное распределение задач между исполнителями в циклическом порядке. Предполагается, что все исполнители равны между собой, то есть отсутствуют предпочтения в пользу какого-либо исполнителя по приоритету, квалификации или строгому правилу назначения им определенного типа задач, а все задачи имеют одинаковую сложность и не имеют формально выделяемых подзадач. Целью этого алгоритма является такое распределение задач, при котором ни один исполнитель не испытывал бы перегрузки.

Формально алгоритм может быть описан так. Пусть в некоторый момент необходимо распределить множество задач T , приоритет $p(t)$ каждой задачи которого $t \in T$ известен. Для каждого исполнителя e из множества исполнителей, подходящих и готовых для выполнения таких задач, E известна текущая загруженность $w(e)$.

Задачи из множества T ранжируются по приоритету, формируется список T_p , определяющий порядок распределения задач: задачи с более высоким уровнем $p(t)$ получают преимущество в распределении. Для каждой задачи t из T_p формируется множество кандидатов $E_c(t)$. Для включения исполнителя $e \in E$ в $E_c(t)$ необходимо, чтобы у него было достаточно свободного времени для выполнения задачи t , учитывая его текущую загруженность $w(e)$ и доступное время $a(e)$:

$$w(e) + s(t) \leq a(e). \quad (4)$$

Выбор конкретного исполнителя $e \in E_c(t)$ для задачи t может осуществляться на основе дополнительных критериев, таких как минимальная загруженность $w(e)$ в момент назначения задачи или достижение близкого к равномерному распределению нагрузки между исполнителями по заверше-

нии распределения всех заданий из T . После назначения задачи t исполнителю e его загруженность обновляется:

$$w(e) \leftarrow w(e) + s(t). \quad (5)$$

Цикл повторяется до тех пор, пока не будут распределены все задачи из списка T_p .

2.2. Алгоритм оптимального распределения задач на основе матрицы назначений

Предложенная в [18] математическая модель оптимизации процесса распределения задач и трудовых ресурсов на предприятии с использованием методов комбинаторной оптимизации легла в основу алгоритма распределения задач между сотрудниками, основанного на анализе ограничений с использованием метода ветвей и границ.

В основе алгоритма лежит формирование матрицы назначений, результатом обработки которой является выбор исполнителя e из множества исполнителей-кандидатов E для каждой задачи t из множества распределяемых задач T . Для каждой задачи, наряду с приоритетом $P(t)$, описываемым по трехэлементной порядковой шкале (низкий, средний, высокий), вводится временное ограничение, описываемое датами начала $D_{start}(t)$ и окончания $D_{end}(t)$. Каждый исполнитель e характеризуется интегральным показателем $Q(e)$ и датами начала $V_{start}(e)$ и окончания $V_{end}(e)$ периода его недоступности. Интегральный показатель вычисляется как линейная комбинация

$$Q(e) = w_c Q_c(e) + w_q Q_q(e) + w_t Q_t(e), \quad (6)$$

где $Q_c(e)$ и $Q_q(e)$ – соответственно оценки объема и качества выполненной работы;

$Q_t(e)$ – оценка соблюдения сроков;

w_c, w_q, w_t – весовые коэффициенты показателей, значения которых в [18] предложено выбирать равными 0,35, 0,4 и 0,25 соответственно.

При вычислении значений всех показателей эффективности исполнителей в алгоритме изначально предложено три уровня: базовый уровень, от которого отсчитывается результат, нормальный уровень, который в обязательном порядке должен быть достигнут, и целевой уровень, к достижению которого исполнитель должен стремиться.

Для составления матрицы назначений и определения оптимального решения используется алгоритм целочисленного программирования — метод «ветвей и границ» [18]. Матрица инициализируется стоимостными значениями задач. В каждой строке матрицы назначений отыскивается минимальный элемент и вычитается из каждого элемента строки. Тем самым в матрице появится не менее одного элемента, равного нулю. Затем в каждом столбце матрицы назначений отыскивается минимальный элемент и при условии отсутствия в столбце нуля вычитается из элементов столбца. Выбирается пара (t, e) — претендент на ветвление, для которых значение элемента матрицы равно нулю. Рассчитывается коэффициент путем сложения минимального значения элемента строки-задачи и минимального значения столбца-исполнителя. Из всех коэффициентов выбирается максимальный, что и определяет оптимальное решение: задача текущей строки назначается исполнителю текущего поля. Поскольку каждому исполнителю назначается только одна работа, то из матрицы назначений удаляются столбец получившего назначение исполнителя, и строка назначенной задачи.

Алгоритм последовательно анализирует подмножества исполнителей, находит оптимальное решение для каждого подмножества задач и исключает его из дальнейшего рассмотрения. В итоге определяется наилучшее распределение задач среди всех возможных комбинаций исполнителей.

2.3. Алгоритм оптимального распределения нагрузки среди сотрудников с учетом их специализации и доступного времени на основе адаптированного генетического алгоритма

В работе [19] был предложен композиционный генетический алгоритм (ГА), использованный для решения задачи распределения учебной нагрузки. Исходные данные включали множество преподавателей — множество исполнителей E , и множество дисциплин и видов занятий — множество задач T . Дополнительно были введены следующие характеристики: $K(e)$ — квалификация исполнителя, т.е. набор дисциплин, которые преподаватель может вести, $R(t, e)$ — релевантность задачи t исполнителю e , выраженная значением из числового интервала $[0, 1]$ степень соответствия его квалификации содержанию задачи.

ГА включает в себя следующие операции: формирование начальной популяции — случайного распределения часов, селекцию — отбор сотрудников по критериям специализации и загруженности, генерацию новых поколений распределений через скрещивание и мутацию, отыскание текущего оптимального распределения нагрузки с учетом ограничений.

Алгоритм начинает работу с формирования начального поколения решений — вариантов распределения $P_0 = \{p_q^{(0)}, q = \overline{1, Q}\}$, где каждый вариант $p_q^{(0)}$ представляет отображение множества задач T на множество исполнителей E , описываемое бинарными переменными $x(t, e)$, указывающими, какая задача e закреплена за исполнителем t . Назначения производятся случайным образом, но таким образом, чтобы выполнялись ограничения. Первое состоит в том, что задача t назначается исполнителю e , только если $t \in K(e)$, а второе — в том, что для каждого исполнителя $e \in E$ общий объем назначенных задач находится в пределах допустимого интервала загрузки.

Для каждого варианта p_k вычисляется функция приспособленности $F(p_k)$. Она состоит из следующих компонент. Компонента $F_1(p_k)$ оценивает степень соответствия $R(t, e)$ для всех пар (t, e) , $F_2(p_k)$ оценивает равномерность распределения нагрузки между исполнителями, $F_3(p_k)$ оценивает интегральную по всем исполнителям близость $W(e)$ к допустимому интервалу $[A_{\min}(e), A_{\max}(e)]$. На основе значений $F(p_k)$ формируется множество наиболее приспособленных распределений P_{sel} .

После оценки решений выполняются эволюционные шаги. Из лучших вариантов формируется новое поколение путем применения операторов скрещивания и мутации: из P_{sel} случайным образом выбираются пары особей p_a и p_b , к которым применяется оператор скрещивания ГА, реализующий случайный обмен частями распределения задач, или подмножествами пар (t, e) между двумя особями. Новые особи включаются в промежуточное поколение P_{next} . К некоторым особям $p \in P_{next}$ применяется оператор мутации ГА, состоящий в случайной замене назначения некоторых задач $t \in T$ другим допустимым исполнителям $e \in E$, таким что $t \in K(e)$.

Новое поколение формируется объединением лучших особей из текущего поколения и новых особей промежуточного поколения: $P_{new} = P_{sel} \cup P_{next}$. Итерации алгоритма продолжаются до тех пор, пока

не будет достигнут критерий останова, состоящий в том, что 90% особей текущей популяции имеют одинаковую максимальную приспособленность. В результате многократных итераций формируется матрица назначений $X = \{x(t, e)\}$, которая удовлетворяет всем ограничениям и обеспечивает оптимизацию выбранных критериев. В тех случаях, когда число оптимизируемых параметров оказывается более пятидесяти, для сокращения времени получения решений могут быть использованы многоагентные генетические алгоритмы (MAGAMO) [20], позволяющие эффективно выполнять многокритериальную крупномасштабную оптимизацию.

2.4. Эвристический алгоритм оптимального распределения

Предложенный в работе [21] эвристический алгоритм оптимального распределения объектов по хранилищам имеет полиномиальную вычислительную сложность и может быть применен для решения задач из различных областей: распределения потоков параллельной обработки больших данных, складской логистики и автоматизированного составления расписания, а также адаптирован для решения задачи распределения задач между исполнителями. Это эвристический жадный алгоритм, выбор распределения на следующем шаге выполняется без учета долгосрочных последствий. Алгоритм стремится к равномерному заполнению хранилищ и минимизации разницы между наиболее и наименее загруженными хранилищами. В [21] также приведены целевые функции, формализация задачи и результаты экспериментального исследования эффективности алгоритма.

Задачи множества $t \in T$ сортируются, формируется список задач по убыванию их трудоемкости $S(t)$. Распределение задач производится в цикле до полного исчерпания списка. На каждой итерации исполнителям назначаются две задачи: наиболее и наименее трудоемкая из T . Номер кандидата-исполнителя вычисляется как результат деления номера задачи в списке по модулю на количество исполнителей. При этом, если при добавлении задачи t к исполнителю e выполняется условие:

$$w(e) + s(t) > a(e), \quad (7)$$

то эта задача не назначается данному исполнителю, а переходит к кандидату-исполнителю со следующим номером. После назначения работы исполни-

телю, выполняется обновление его загруженности (5), а назначенные задачи исключаются из списка.

В силу того, что алгоритм является полиномиальным эвристическим жадным алгоритмом, он обеспечивает высокую скорость работы и хорошее качество распределения даже при больших объемах данных в условиях жестких ограничений на время выполнения.

2.5. Алгоритм на основе случайного разбиения выборки

Для решения проблемы неравномерного распределения задач среди сотрудников, что приводит к перегрузке одних и недогрузке других, в статье [22] рассматривается программное обеспечение, позволяющая с участием менеджера-человека эффективно распределять новые задачи на основе анализа текущей загруженности сотрудников $w(e)$ и сложности задач $s(t)$. В основе ПО лежит алгоритм случайного разбиения выборки (*Random Sample Partition Algorithm, RSP*) [23], который выполняет обработку больших объемов данных за счет разбиения их на более мелкие блоки данных, доступные для непосредственного анализа человеком. Результаты анализа выводятся в удобном виде для менеджера, который затем принимает окончательное решение о распределении задач.

На основе информации о приоритете $p(t)$ и сложности $s(t)$ каждой задачи, информации о доступном времени $a(e)$ и текущей загруженности $w(e)$ исполнителя формируется множество D , содержащее данные о задачах и исполнителях. Элементами D являются пары (t, e) , дополненные характеристиками, которые позволяют оценить эффективность выполнения задач и уровень профессионального развития. Множество D разбивается случайным образом на K непересекающихся близких по мощности подмножеств $\{D_k, k = \overline{1, K}\}$. На каждом подмножестве D_k проводится анализ, направленный на определение текущей загруженности $w(e)$ каждого исполнителя, оценку сложности $s(t)$ и приоритета $p(t)$ каждой задачи, вычисление совокупной нагрузки каждого исполнителя. Результатом является информационный объект $R(D_k)$ в форме аналитического отчета, содержащего в числе прочего информацию о том, сколько задач приходится на исполнителя или группу исполнителей, о составе подмножества задач, времени их выполнения, сложности, приоритетах, а также о потенциальной

перегрузке сотрудников и их свободных ресурсах, достаточных для выполнения новых задач.

Аналитические отчеты $R(D_k)$, полученные на всех множествах разбиения, объединяются в общий результат – отчет R_{total} , представляющий собой рекомендации по распределению задач с учетом равномерности нагрузки и сложности задач. Результирующий отчет R_{total} отображается в графическом интерфейсе ПО для менеджера, позволяя последнему принять окончательное решение о назначении задач исполнителям.

Рассматриваемый алгоритм реализует сбор и структурирование данных о задачах и исполнителях, а также первичный анализ загруженности, что позволяет его рассматривать как возможный инструмент для систем поддержки принятия управленческих решений, обеспечивающий формирование рекомендаций для руководителя, занятого распределением задач и управления персоналом.

2.6. Алгоритм корректировки загрузки между исполнителями с использованием нейронных сетей

При рассмотрении задачи производственного планирования как многокритериальной задачи оптимизации авторами статьи [24] было обосновано, что внедрение традиционных систем распределения задач может быть затруднено в силу ограниченного набора фиксированных критериев, используемых в них. Поэтому для преодоления такого ограничения и возможности учета дополнительных трудно формализуемых факторов, таких как приоритеты заказов или специфика оборудования, было предложено использовать гибридный алгоритм, сочетающий традиционные методы оптимизации с возможностью корректировки плана с помощью нейронных сетей. Используемый нейросетевой корректор основан на двух типах искусственных нейронных сетей (ИНС): многослойном перцептроне (МСП) [25] и самоорганизующейся карты Кохонена (*Self-Organizing Map*, SOM) [26].

Важной особенностью алгоритма является группировка исполнителей в типы рабочих центров (РЦ) и задач – в группы задач. Все задачи из множества T группируются по их релевантности типам рабочих центров R с использованием SOM Кохонена, обучаемой с помощью алгоритма самоорганизации. Для каждой задачи t определяется тип РЦ $R(t)$, который может выполнить ее за минимальное

время. Задачи объединяются в группы $G(t)$, где каждая группа соответствует типу РЦ, наиболее подходящему для выполнения задач этой категории.

На основе исторических данных и введенных экспертных правил выполняется обучение ИНС типа МСП, который динамически корректирует приоритеты задач $p(t)$, распределение задач по группам $G(t)$, выбор исполнителя с учетом доступного времени $a(e)$ и загруженности $w(e)$. После завершения этапа коррекции задачи из множества T назначаются исполнителям из E таким образом, чтобы минимизировать общее время выполнения, выравнять загрузки $w(e)$ при сохранении приоритетов задач $p(t)$ и ограничений по доступному времени $a(e)$.

В алгоритм также заложен учет некоторых ограничений, например, не все задачи могут быть назначены любому исполнителю, а также возможность решающего назначения: если задача принадлежит высокоприоритетному заказу, она жестко закрепляется за определенным исполнителем вне зависимости от оптимальности по другим критериям.

2.7. Эвристический фронтальный алгоритм

На основе анализа решения задачи оптимального распределения между исполнителями в рамках проекта работ, связанных зависимостями, представимыми в виде ациклического ориентированного графа, в статье [27] предлагается эвристический алгоритм, позволяющий находить приемлемое распределение работ, укладывающееся в заданные временные рамки и минимизирующее стоимость выполнения проекта. В основе алгоритма лежит принцип фронтальных работ, состоящий в том, что задачи последовательно назначаются исполнителям, исходя из трудоемкости первых и доступности последних.

Для формализации дополнительно вводятся $D(t)$ – множество предшествующих задач для задачи t и $R(t) \subseteq E$ – множество исполнителей, способных выполнить задачу t , а также в трудоемкости $s(t, e)$ выполнения задачи t учитывается конкретный исполнитель e .

Фронт работ на k -м шаге F_k представляет собой подмножество задач T , все предшествующие работы которых выполнены: $D_k(t) = \emptyset$. Задачи из F_k становятся кандидатами на распределение. Для

каждой задаче $t \in F_k$ и для каждого исполнителя $e \in R(t)$ вычисляются момент времени начала выполнения задачи t исполнителем e :

$$x(e) = \max\left(y_k(e), \max_{t' \in D(t)} y(t')\right) \quad (8)$$

и завершения ее выполнения:

$$y(e) = x_k(e) + s(t, e). \quad (9)$$

Задача назначается такому исполнителю e^* , для которого $y_k(e)$ минимальна. Для этого исполнителя обновляются значения $a(e^*)$ и $w(e^*)$, а для задачи фиксируется ее время $x(t) = x_k(e^*)$ завершения $y(t) = y_k(e^*)$. Распределение задач множества F_k проводится до его исчерпания.

После назначения всех задач из текущего фронта F_k формируется новый фронт F_{k+1} , включающий задачи, у которых все предшествующие работы теперь выполнены $D_{k+1}(t) = \emptyset$, которые требуют распределения. При этом время завершения уже назначенных работ исполнителям обновляется $y_{k+1}(e) = y_k(e)$. Формирование новых фронтов задач повторяется до тех пор, пока не будет распределено все множество задач T .

После распределения всех задач выполняется проверка допустимости и оптимальности. Первая состоит в проверке соблюдения директивных сроков выполнения каждой задачи $d(t)$: т.е. $y(t) \leq d(t)$. Вторая состоит в оценке качества проекта, с использованием целевой функции, минимизация которой позволяет снизить общие затраты выполнения проекта:

$$f = \sum_{e \in E} \sum_{t \in T} p(t) x(t, e) (d(t) - y(t, e)). \quad (10)$$

Если сформированный план допустимый и оптимальный, то формируется расписание проекта и оценивается общая стоимость. Варианты распределения формируются за счет разных назначений исполнителей на работы. Так первый фронт, включающий работы без предшественников, отыскивается полным перебором всех возможных назначений. Далее, для каждой новой работы, рассматриваются все доступные исполнители, рассчитывается время выполнения с учетом занятости и зависимостей. Таким образом, разные варианты представляют собой разные сочетания назначений $x(t, e)$, а оптимальный выбирается сравнением полученных комбинаций с использованием целевой функции, обеспечивающей наименьшую стоимость проекта.

2.8. Итеративный алгоритм делегирования задач

При рассмотрении подхода к автоматизации процесса делегирования задач в проектном управлении в работе [28] была предложена модель, основанная на использовании итеративных алгоритмах, выполняемых над размещенными в БД типа NoSQL данными об опыте, доступности и предпочтениях сотрудников. Основная цель разработки такого алгоритма состояла в сокращении участия менеджера проекта в рутинном распределении задач и повышении качества назначений за счет анализа накопленных данных. Авторы представляют итеративный алгоритм делегирования задач, основанный на ключевых словах задачи t и рейтинговых показателях исполнителя e . Если стандартный алгоритм не может назначить задачу из-за недостаточного количества данных, применяется резервный механизм назначения, учитывающий набор профессиональных навыков и предпочтений исполнителей.

Алгоритм выполняется следующим образом. При добавлении новой задачи $t \in T$ извлекаются ее ключевые слова $K(t)$. Для каждого ключевого слова $k \in K(t)$ ищутся исполнители $e \in E$, имеющие наибольший рейтинг $R(e, k)$, их рейтинги суммируются, формируя общий рейтинг $R_s(e)$ для каждого исполнителя. Формируется ранжированный по убыванию $R_s(e)$ список исполнителей, по которому проводится последовательная проверка того, доступен ли очередной исполнитель, может ли он принять задачу на выполнение с учетом своей загруженности $w(e)$ и доступного времени $a(e)$. Если подходящий исполнитель найден, задача назначается ему.

Если подходящих исполнителей нет, запускается резервный алгоритм, использующий: профессиональные навыки $Q(e)$ и личные предпочтения $L(e)$ исполнителя, на основании которых вычисляется комбинированный рейтинг:

$$R_c(e) = Q(e) \cdot w_q + L(e) \cdot w_l, \quad (11)$$

где w_q и w_l — веса, определяющие важность навыков и предпочтений соответственно. Затем выбирается наиболее подходящий исполнитель аналогично тому, как это было выполнено для рейтинга $R_s(e)$. Если и в этом случае задача остается неназначенной, она помечается как требующая ручного назначения. Такие задачи назначаются менеджером проекта после алгоритмической обработки всего множества задач T .

2.9. Алгоритм делегирования задач с помощью многоагентной системы

В работе [29] представлен алгоритм планирования, декомпозиции и делегирования задач в условиях неструктурированной децентрализованной среды, где агенты обладают ограниченной информацией о своих возможностях и возможностях других агентов. Описанный подход к распределению задач с помощью многоагентной системы основан на том, что каждый агент может частично выполнить задачу и делегировать остальные ее части другим агентам. В основе подхода лежит алгоритм рекурсивной декомпозиции задач, основанный на методах прикладного искусственного интеллекта, позволяющий разбивать задачу на подзадачи, выполнять те из них, которые агент может реализовать самостоятельно и делегировать нереализуемые подзадачи другим агентам. Такой процесс назначения продолжается рекурсивно до полного распределения задачи или достижения условия останова. Алгоритм работает в условиях отсутствия полной информации о возможностях всех агентов и поддерживает частичное планирование, позволяя проводить планирование с абстрактными действиями, которые позже заменяются конкретными шагами при помощи делегирования.

Итерация алгоритма назначения исполнителей для задачи t выполняется следующим образом. Прежде всего выполняется проверка, является ли задача примитивной, или не требующей дальнейшей декомпозиции. Если это так, то среди исполнителей E определяется исполнитель e , для которого выполняется условие (4) и условие достаточности компетенций: $C(t) \subset C(e)$. Если такой исполнитель найден, то задача назначается ему; если нет, то задача помечается как нерешенная.

Если исходная задача t не является примитивной, то агент вызывает функцию декомпозиции $D(t)$, возвращающую множество ее подзадач $\{t'_k, k = \overline{1, K}\}$. Если для подзадачи t' выполняется условие $C(t') \subset C(e)$, то агент e назначает ее себе, увеличивая при этом свою загрузку (5), а если не выполняется, то задача делегируется другому агенту $e' \in E$, который обладает достаточным набором компетенций для ее выполнения: $C(t') \subset C(e')$. Делегирование подзадач продолжается до тех пор, пока все подзадачи не будут назначены или не будет достигнуто дополнительно условие останова.

В качестве дополнительного развития алгоритма может использоваться двухфазный механизм, состоящий в том, что на первой фазе агент запрашивает у других агентов готовность выполнить подзадачу, а на второй — она передается тому исполнителю, который первым подтвердил возможность ее выполнения.

2.10. Адаптивный алгоритм на основе ожиданий

Проведенная в статье [30] систематизация подходов для повышения эффективности краудсорсинга с использованием методов прикладного искусственного интеллекта позволила выделить три ключевых направления: делегирование задач, мотивацию исполнителей и контроль качества, для каждой из которых была предложена обновленная таксономия, а также проанализированы ограничения и перспективы развития. Это позволило авторам статьи предложить алгоритм делегирования сложных задач на основе WMST-модели (*Weighted Multi-Skill Tree*) [31], используемой для оценки навыков исполнителей. Целью создания алгоритма было обеспечение процедуры назначения задач исполнителям с оптимальным соответствием навыков и балансировкой загрузки.

Так для каждого исполнителя e предварительно строится WMST-модель, оценивающая его компетенции $C(e)$. При распределении множества задач T для каждой задачи t выбирается такой исполнитель e , который удовлетворяет трем условиям: соответствию компетенций решаемой задачи $C(t) \subset C(e)$, достаточности исполнительного ресурса $A(e) \geq S(t)$, и минимальной текущей нагрузки W_e . Последнее условие направлено на балансировку общей нагрузки всего множества исполнителей E . Если выбранный исполнитель временно недоступен, то алгоритм пересматривает назначение, выбирая нового исполнителя, удовлетворяющего вышеуказанным условиям.

3. Сравнительный анализ алгоритмов

Сравнительный анализ алгоритмов распределения задач между исполнителями осуществляется по ряду критериев, позволяющих всесторонне оценить их эффективность и практическую применимость. Уровень учета компетенций отража-

ет степень соответствия между характеристиками исполнителей и требованиями задач, определяя способность алгоритма обеспечивать оптимальное распределение с учетом профессиональных навыков и опыта.

Адаптивность к изменениям характеризует гибкость алгоритма и его способность корректировать решения при изменении состава исполнителей, условий выполнения или при появлении новых задач. Требования к данным показывают объем, структуру и точность исходной информации, необходимой для корректной работы алгоритма. Устойчивость к неполным или шумным данным определяет надежность алгоритма в условиях неопределенности и информационных искажений. Прозрачность и объяснимость решений связаны со степенью интерпретируемости получаемых результатов и возможностью анализа причин, повлиявших на выбор конкретного исполнителя. Вычислительная сложность характеризует затраты вычислительных ресурсов и времени, необходимых для получения решения, а масштабируемость отражает способность алгоритма сохранять эффективность при увеличении числа задач и исполнителей. Стоимость внедрения включает совокупность организационных, программных и аппаратных затрат, связанных с реализацией и сопровождением алгоритма. Наконец, ориентация на развитие персонала отражает потенциал алгоритма к использованию в целях повышения квалификации исполнителей, выявления зон дефицита компетенций и формирования оптимальной структуры распределения труда.

Анализ *таблицы 1* показывает, что различные алгоритмы обладают специфическими преимуществами и ограничениями, что определяет области их практического применения. Ниже представлены основные рекомендации по выбору подхода.

Алгоритм **Round Robin**, описанный в 2.1, представляет собой простой и прозрачный метод, применимый в условиях равномерной загрузки. Он эффективен в ситуациях, где требуется быстрое и минимально затратное распределение задач при ограниченности входных данных и высокой необходимости объяснимости решений.

Метод матрицы назначений, описанный в 2.2, рекомендуется использовать в случаях, когда доступны детализированные данные о задачах и важна высокая прозрачность принимаемых решений. Однако данный подход уступает в устойчивости

к шумным данным и обладает ограниченной масштабируемостью.

Генетический алгоритм, описанный в 2.3, целесообразно использовать для решения сложных оптимизационных задач с множеством ограничений. Его преимущество заключается в высокой адаптивности к изменяющимся условиям и способности отыскания глобально оптимальных решений. При этом он требует значительных вычислительных ресурсов и затрат времени.

Эвристический жадный алгоритм, описанный в 2.4, ориентирован на оперативное получение результата в условиях ограниченных данных и высокой динамики. Преимуществом алгоритма является высокая скорость работы и низкая вычислительная сложность, однако получаемые решения, как правило, далеки от оптимальных.

Метод RSP (Randomized Search/Planning), описанный в 2.5, демонстрирует устойчивость при работе с неполными и зашумленными данными. Он обеспечивает компромисс между случайностью и оптимальностью решений, что делает его применимым в условиях неопределенности.

ИНС (нейросетевой) корректор, описанный в 2.6, показывает высокую эффективность при реализации самообучающихся систем, предназначенных для обработки больших массивов данных. Это делает его применимым для прогнозирования и адаптивного управления, однако его внедрение требует значительных затрат и высоких вычислительных ресурсов.

Фронтальный алгоритм, описанный в 2.7, обеспечивает сбалансированные результаты в условиях средней сложности задач. Его основное преимущество заключается в универсальности, что делает его применимым при отсутствии жестких требований к оптимальности или устойчивости.

Итеративный алгоритм, описанный в 2.8, рекомендуется применять в случаях, когда решение должно постепенно уточняться. Данный подход эффективен в динамической среде, позволяя пошагово улучшать результат и адаптироваться к изменениям.

Многоагентный (МА) подход, описанный в 2.9, наиболее целесообразен для систем, в которых взаимодействуют несколько участников или подсистем. Он эффективен при высокой изменчивости среды и задачах коллективного характера.

Алгоритм на основе **WMST-модель**, описанный в 2.10, применим для задач долгосрочного стратегического планирования в масштабных системах. Он обеспечивает высокую адаптивность и устойчивость, однако требует значительных ресурсов для внедрения и сопровождения.

Заключение

Результаты анализа десяти алгоритмов распределению задач между исполнителями, опубликованных за последние 20 лет, позволяют заключить, что современные подходы, отличаясь заметным разнообразием, все еще обладают рядом существенных ограничений, сдерживающих их применение в качестве универсальных инструментов в системах автоматизации и оптимизации бизнес-процессов. Основной причиной этого является то, что большинство рассмотренных алгоритмов основаны на формальной обработке количественных характеристик таких, как загруженность сотрудников, временные затраты на вы-

полнение задач или приоритеты заданий, исключая из рассмотрения при этом индивидуальные компетенции исполнителей. Учет только количественных характеристик может приводить к тому, что назначение задач зачастую выполняется без учета соответствия реальных навыков и возможных предпочтений работников, что может отражаться негативно на общей эффективности выполнения проектов.

В то же время обладающие большей гибкостью и адаптивностью подходы, поддерживающие практические реализации управления персоналом при выполнении сложных проектов, требуют учета компетенции исполнителя посредством не скалярного параметра, бинарного или числового, но многомерной характеристики, отдельные компоненты которой различно влияют на качество, скорость и надежность выполнения задачи. Однако современные алгоритмы распределения задач недостаточно гибко учитывают этот фактор, либо вводят его в модель в упрощенном виде, без должного математического и логического обосно-

Таблица 1.

Сравнение алгоритмов на основе критериев

№	Критерий	Round Robin 2.1	Матрица назначений 2.2	Генетический алгоритм 2.3	Эвристический жадный 2.4	RSP 2.5	ИНС корректор 2.6	Фронтальный алгоритм 2.7	Итеративный алгоритм 2.8	МА подход 2.9	WMST-модель 2.10
1	Уровень учета компетенций	Нет	Lim	Lim	Нет	Lim	Lim	Lim	Lim	Lim	Lim
2	Адаптивность к изменениям	Low	Med	High	Low	Med	High	Med	High	High	High
3	Требования к данным	Min	High	Med	Min	Med	VH	Med	Med	High	High
4	Устойчивость к неполным/шумным данным	High	Low	Med	Med	High	Med	Med	Med	High	Med
5	Прозрачность и объяснимость решений	Full	High	Low	High	Med	Low	Med	Med	Low	Low
6	Вычислительная сложность	Low	High	High	Low	Med	High	Med	Med	High	High
7	Масштабируемость	High	Lim	High	High	High	Med	Med	High	High	Med
8	Стоимость внедрения	Low	High	Med	Low	Med	VH	Med	Med	High	High
9	Ориентация на развитие персонала	Нет	Part	Да	Нет	Part	Да	Part	Part	Да	Да

Обозначения: Ограниченный(ая/ые) – Lim, Минимальный(ая/ые) – Min, Низкий(ая/ие) – Low, Средний(ая/ие) – Med, Высокий(ая/ие) – High, Очень высокий(ая/ие) – VH, Частично – Part, Полный(ая/ые) – Full.

вания. Это свидетельствует о существующей потребности в разработке новых, более совершенных методов распределения задач, которые бы органично включали в себя как методы расчета на основе стандартно формализуемых числовых показателей, так и анализ компетенций как ключевой элемент при принятии решений.

Таким образом, несмотря на наличие ряда новых решений в области автоматизированного управления задачами, создание алгоритмов, способных комплексно учитывать не только временные и нагрузочные показатели, но и профессиональные навыки сотрудников остается актуальной задачей.

Такие подходы могли бы являться одним из инновационных инструментов повышения эффективности при распределении задач, обеспечивающим минимизацию риска ошибочных решений и повышения уровня мотивации персонала. ■

Благодарности

Теоретические результаты, использованные в настоящем исследовании, были получены в рамках государственного задания Министерства науки и высшего образования Российской Федерации: проект № FSFF-2026-0008.

Литература

1. Балдин К. В., Передеряев И. И., Голов Р. С. Управление рисками в инновационно-инвестиционной деятельности. М.: Дашков и К, 2017.
2. A systematic literature review on task allocation and performance management techniques in cloud data center / N. Chauhan [et al.] // *Computer Systems Science and Engineering*. 2024. Vol. 48. No. 3. P. 571–608. <https://doi.org/10.32604/csse.2024.042690>
3. Zhang P., Zhang A., Xu G. Optimized task distribution based on task requirements and time delay in edge computing environments // *Engineering Applications of Artificial Intelligence*. 2020. Vol. 94. Article 103774. <https://doi.org/10.1016/j.engappai.2020.103774>
4. Utilizing machine learning algorithms for task allocation in distributed agile software development / D. Al-Fraihat [et al.] // *Heliyon*. 2024. Vol. 10. No. 21. Article e39926. <https://doi.org/10.1016/j.heliyon.2024.e39926>
5. Skaltsis G. M., Shin H.-S., Tsourdos A. A review of task allocation methods for UAVs // *Journal of Intelligent & Robotic Systems*. 2023. Vol. 109. Article 76. <https://doi.org/10.1007/s10846-023-02011-0>
6. Подход к формированию интеллектуальной академической генеалогии с использованием больших языковых моделей / И. М. Лернер и [др.] // *Электромагнитные волны и электронные системы*. 2024. Т. 29. № 4. С. 108–120. <https://doi.org/10.18127/j5604128-202404-09>
7. К вопросу подбора учебно-методической информации для реализации адаптивной электронной образовательной среды: алгоритм априорной классификации авторов / С. Г. Григорьев и [др.] // *Информатика и образование*. 2025. Т. 40. № 2. С. 66–78. <https://doi.org/10.32517/0234-0453-2025-40-2-66-78>
8. К вопросу подбора учебно-методической информации для реализации адаптивной электронной образовательной среды: алгоритм подбора авторов литературы с учетом эмоционально-психологических особенностей пользователей на базе идей академической генеалогии / С. Г. Григорьев и [др.] // *Информатика и образование*. 2025. Т. 40. № 3. С. 69–79. <https://doi.org/10.32517/0234-0453-2025-40-3-69-79>
9. Бекларян А. Л., Акопов А. С. Имитационная модель оптимального распределения потока кредитных заявок для межрегионального центра андеррайтинга коммерческого банка // *Вестник компьютерных и информационных технологий*. 2018. № 11. С. 46–56. <https://doi.org/10.14489/vkit.2018.11.pp.046-056>
10. Тараненко Д. П., Колесников А. В. Особенности делегирования полномочий в условиях развития дистанционных технологий управления // *Нормирование и оплата труда в промышленности*. 2021. № 12. С. 56–59. <https://doi.org/10.33920/pro-3-2112-06>
11. Schaad A., Pymont B. Review mechanism for controlling the delegation of tasks in a workflow system. US Patent № 7831978. 2010.
12. Akhavein J., Frame W. S., White L. J. The diffusion of financial innovations: An examination of the adoption of small business credit scoring by large banking organizations // *Journal of Business*. 2005. Vol. 78. No. 2. P. 577–596. <https://doi.org/10.1086/427639>
13. Gupta D., Denton B. Appointment scheduling in health care: Challenges and opportunities // *IIE Transactions*. 2008. Vol. 40. No. 9. P. 800–819. <https://doi.org/10.1080/07408170802165880>
14. A BIM-based framework for construction project scheduling risk management / F. H. Abanda [et al.] // *International Journal of Computer Aided Engineering and Technology*. 2020. Vol. 12. No. 2. P. 182–218. <https://doi.org/10.1504/IJCAET.2020.105575>
15. Voigt S., El-Bialy N. Identifying the determinants of judicial performance: Taxpayers' money well spent? // *SSRN Electronic Journal*. 2013. <https://doi.org/10.2139/ssrn.2241224>

16. Aksin Z., Armony M., Mehrotra V. The modern call center: A multi-disciplinary perspective on operations management research // *Production and Operations Management*. 2007. Vol. 16. No. 6. P. 665–688. <https://doi.org/10.1111/j.1937-5956.2007.tb00288.x>
17. Yadav P. K., Saxena S. Design and implementation of Round Robin scheduling algorithm using dispatch latency // *International Online Conference on Emerging Trends in Multi-Disciplinary Research “ETMDR-2020”*. P. 356–365.
18. Еремина И. И., Лысанов Д. М. Математическая модель оптимизации процесса распределения задач и трудовых ресурсов на предприятии // *Mezhdunarodnyy nauchnyy zhurnal (American Scientific Journal)*. 2020. №. 42. С. 65–71.
19. Использование генетического алгоритма для решения задачи распределения учебной нагрузки / Е. А. Корчевская и [др.] // *Вестник вицебсага дзяржаўнага ўніверсітэта*. 2023. № 3. С. 15–19.
20. Akopov A. S., Nevencev M. A. A multi-agent genetic algorithm for multi-objective optimization // *2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK*. 2013. P. 1391–1395. <https://doi.org/10.1109/SMC.2013.240>
21. Мунерман В. И., Мунерман Д. В. Анализ алгоритма оптимального распределения // *Современные информационные технологии и ИТ-образование*. 2019. Т. 15. № 3. С. 619–625. <https://doi.org/10.25559/sitito.15.201903.619-625>
22. Automation of employee workload management using random sample partition algorithm / M. Suresh [et al.] // *International Journal of Recent Technology and Engineering*. 2020. Vol. 8. No. 6. P. 5282–5286. <https://doi.org/10.35940/ijrte.F9761.038620>
23. Salloum S., Huang J. Z., He Y. Random sample partition: A distributed data model for big data analysis // *IEEE Transactions on Industrial Informatics*. 2019. Vol. 15. No. 11. P. 5846–5854. <https://doi.org/10.1109/TII.2019.2912723>
24. Лукьянов Л. А., Спивак С. И., Христоролюбов В. Л. Нейронная сеть корректор для распределения работ в задаче внутрицехового планирования // *Вестник Башкирского университета*. 2016. Т. 21. № 4. С. 859–863.
25. Ефимов Е. Н., Шевгунов Т. Я. Разработка и исследование методики построения нейронных сетей на основе адаптивных элементов // *Труды МАИ*. 2012. № 51.
26. Кохонен Т. Самоорганизующиеся карты. М.: Лаборатория знаний, 2017.
27. Новикова Т.П., Новиков А.И. Алгоритм решения задачи оптимального распределения работ в сетевых канонических структурах // *Лесотехнический журнал*. 2014. Т. 4, № 4. С. 309–317. <https://doi.org/10.12737/8515>
28. Pop B., Boian F. Algorithms for automating task delegation in project management // *Federated Conference on Computer Science and Information Systems*. 2014. Vol. 2. P. 1191–1196. <https://doi.org/10.15439/2014F426>
29. Lopes A. L., Botelho L. M. Task decomposition and delegation algorithms for coordinating unstructured multi agent systems // *First International Conference on Complex, Intelligent and Software Intensive Systems (CISIS'07), Vienna, Austria*. 2007. P. 209–214. <https://doi.org/10.1109/CISIS.2007.52>
30. Towards AI-empowered crowdsourcing / S. Wang [et al.] // *arXiv:2212.14676*. 2022. <https://doi.org/10.48550/arXiv.2212.14676>
32. Al-Anzi F., Al-Zame K., Allahverdi A. Weighted multi-skill resources project scheduling // *Journal of Software Engineering and Applications*. 2020. Vol. 3. No. 12. P. 1125–1130. <https://doi.org/10.4236/jsea.2010.312131>

Об авторах

Тимофей Яковлевич Шевгунов

кандидат технических наук;

доцент, кафедра «Теоретическая радиотехника», Московский авиационный институт (национальный исследовательский университет), Россия, 125993, г. Москва, Волоколамское шоссе, 4;

доцент, Высшая школа бизнеса, Национальный исследовательский университет «Высшая школа экономики», Россия, 119049, г. Москва, ул. Шаболовка, д. 28/11;

E-mail: shevgunov@gmail.com

ORCID: 0000-0003-1444-983X

Анна Александровна Крошила

студентка магистратуры, Высшая школа бизнеса, Национальный исследовательский университет «Высшая школа экономики», Россия, 119049, г. Москва, ул. Шаболовка, д. 28/11;

E-mail: ankrosh@vk.com

A review and comparison of newer methods for task allocation among performers

Timofey Yakovlevich Shevgunov ^{a, b}

E-mail: shevgunov@gmail.com

Anna Alexandrovna Kroshilina ^b

E-mail: ankrosh@vk.com

^a Moscow Aviation Institute (National Research University), Moscow, Russia

^b Graduate School of Business, HSE University, Moscow, Russia

Abstract

This paper presents a description of the current state and the results of an analysis of recent advances in the problem domain of automated task distribution among employees. The purpose of the study is to identify the main trends and patterns in the development of existing task allocation methods, to determine their strengths and limitations, and to justify the need for new approaches and algorithms that can improve the efficiency of task delegation to employees. Using a unified system of notations for the key concepts of the subject area, the article provides a concise descriptive review of ten universal task distribution algorithms published over the past twenty years. The comparative analysis was carried out according to a set of criteria reflecting both the technical and the organizational-behavioral aspects of how these algorithms function. The key evaluation criteria included: the degree to which performer competencies are taken into account; adaptability to changing external conditions and team composition; requirements for completeness and structure of the input data; robustness to incomplete or noisy data; transparency and explainability of decision-making; computational complexity; scalability with an increasing number of tasks and employees; implementation and maintenance costs; and orientation toward personnel development and competence enhancement. The comparative analysis we carried out made it possible to identify the advantages and shortcomings of each method and to formulate recommendations for their most effective practical application. The results showed that none of the examined algorithms can be considered a universal tool for delegation. Furthermore, it was found that comprehensive information about a performer's suitability for solving tasks requiring diverse competencies is either ignored or insufficiently utilized by many algorithms. This observation leaves open the problem of developing new approaches to task allocation and designing new algorithms based on them.

Keywords: task distribution, task allocation, performer assignment, assignment matrix, task delegation algorithms, round-robin algorithm, front based algorithm, genetic algorithm management, product digital twin, resource digital twin

Citation: Shevgunov, T. Ya., & Kroshilina, A. A. (2026). A review and comparison of newer methods for task allocation among performers. *Business Informatics*, 20(1), 67–85. <https://doi.org/10.17323/2587-814X.2026.1.67.85>

References

1. Baldin, K. V., Perederyaev, I. I., & Golov, R. S. (2017). *Upravlenie riskami v innovatsionno-investitsionnoi deyatel'nosti [Risk management in innovation and investment activities]*. Moscow: Dashkov i K (in Russian).
2. Chauhan, N., Kaur, N., Saini, K. S., Verma, S., Alabdulatif, A., Khurma, R. A., Garcia-Arenas, M., & Castillo, P. A. (2024). A systematic literature review on task allocation and performance management techniques in cloud data center. *Computer Systems Science and Engineering*, 48(3), 571–608. <https://doi.org/10.32604/csse.2024.042690>
3. Zhang, P., Zhang, A., & Xu, G. (2020). Optimized task distribution based on task requirements and time delay in edge computing environments. *Engineering Applications of Artificial Intelligence*, 94, 103774. <https://doi.org/10.1016/j.engappai.2020.103774>
4. Al-Fraihat, D., Sharrab, Y., Al-Ghuwairi, A.-R., Alzabut, H., Beshara, M., & Algarni, A. (2024). Utilizing machine learning algorithms for task allocation in distributed agile software development. *Heliyon*, 10(21), e39926. <https://doi.org/10.1016/j.heliyon.2024.e39926>
5. Skaltsis, G. M., Shin, H.-S., & Tsourdos, A. (2023). A review of task allocation methods for UAVs. *Journal of Intelligent & Robotic Systems*, 109(4), 76. <https://doi.org/10.1007/s10846-023-02011-0>
6. Lerner, I. M., Marinosyan, A. Kh., Grigoriev, S. G., Yusupov, A. R., Anikyeva, M. A., & Garifullina, G. A. (2024). An approach to the formation of intellectual academic genealogy using large language models. *Electromagnetic Waves and Electronic Systems*, 29(4), 108–120 (in Russian). <https://doi.org/10.18127/j5604128-202404-09>
7. Grigoriev, S. G., Lerner, I. M., Marinosyan, A. Kh., Arutyunova, N. K., & Grigorieva, M. A. (2025). On the issue of educational and methodological information selection for implementing an adaptive learning management system: Algorithm of a priori authors classification. *Informatics and Education*, 40(2), 66–78 (in Russian). <https://doi.org/10.32517/0234-0453-2025-40-2-66-78>
8. Grigoriev, S. G., Lerner, I. M., Marinosyan, A. Kh., & Grigorieva, M. A. (2025). On the issue of educational and methodological information selection for implementing an adaptive learning management system: Algorithm of selecting authors of literature taking into account the emotional and psychological characteristics of users based on the ideas of academic genealogy. *Informatics and Education*, 40(3), 69–79 (in Russian). <https://doi.org/10.32517/0234-0453-2025-40-3-69-79>
9. Beklaryan, A. L., & Akopov, A. S. (2018). Simulation model of the optimal allocation credit applications for interregional underwriting center of a commercial bank. *Vestnik Komp'uternykh i Informatsionnykh Tekhnologii*, 11, 46–56 (in Russian). <https://doi.org/10.14489/vkit.2018.11.pp.046-056>
10. Taranenko, D. P., & Kolesnikov, A. V. (2021). Features of delegation of powers in the context of the development of remote control technologies. *Normirovanie i Oplata Truda v Promyshlennosti (Rationing and Remuneration of Labor in Industry)*, 12, 56–58 (in Russian). <https://doi.org/10.33920/pro-3-2112-06>
11. Schaad A., & Pyrmont B. (2010). Review mechanism for controlling the delegation of tasks in a workflow system. US Patent No. 7831978.
12. Akhavein, J., Frame, W. S., & White, L. J. (2005). The diffusion of financial innovations: An examination of the adoption of small business credit scoring by large banking organizations. *The Journal of Business*, 78(2), 577–596. <https://doi.org/10.1086/427639>
13. Gupta, D., & Denton, B. (2008). Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions*, 40(9), 800–819. <https://doi.org/10.1080/07408170802165880>
14. Abanda, F. H., Musa, A. M., Clermont, P., Tah, J. H. M., & Oti, A. H. (2020). A BIM-based framework for construction project scheduling risk management. *International Journal of Computer Aided Engineering and Technology*, 12(2), 182. <https://doi.org/10.1504/ijcaet.2020.105575>
15. Voigt, S., & El Bialy, N. (2013). Identifying the determinants of judicial performance: Taxpayers' money well spent? *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2241224>
16. Aksin, Z., Armony, M., & Mehrotra, V. (2007). The modern call center: A multi-disciplinary perspective on operations management research. *Production and Operations Management*, 16(6), 665–688. <https://doi.org/10.1111/j.1937-5956.2007.tb00288.x>
17. Yadav, P.K., Saxena, S. (2020). Design and implementation of Round Robin scheduling algorithm using dispatch latency. *International Online Conference on Emerging Trends in Multi-Disciplinary Research (ETMDR-2020)*, 356–365.
18. Eremina, I. I., & Lysanov, D. M. (2020). Matematicheskaya model' optimizatsii protsessa raspredeleniya zadach i trudovykh resursov na predpriyatii [Mathematical Model for Optimizing the Process of Task and Labor Resource Distribution in Enterprises]. *American Korchevskaya, E. A., Ermachenko, S. A., Nikonova, T. V., Markova, L. V., & Shpakova, Y. A. (2023). Ispol'zovanie geneticheskogo algoritma dlya resheniya zadachi raspredeleniya uchebnoi nagruzki [Using a genetic algorithm to solve the problem of distributing academic workload]. Vesnik Vitebskogo gosudarstvennogo universiteta*, 3(120), 15–19 (in Russian).
20. Akopov, A. S., & Hevencev, M. A. (2013). A Multi-agent Genetic Algorithm for Multi-objective Optimization. *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 1391–1395. <https://doi.org/10.1109/smcs.2013.240>
21. Munerman, V. I., & Munerman, D. V. (2019). Analysis of an optimal distribution algorithm. *Modern Information Technologies and IT-Education*, 15(3), 619–625 (in Russian). <https://doi.org/10.25559/sitito.15.201903.619-625>

22. Suresh, M., Samuel, R. B., Bhuvaneshwar, T., Jaubin, R. H., & Balaji, R. (2020). Automation of employee workload management using random sample partition algorithm. *International Journal of Recent Technology and Engineering (IJRTE)*, 8(6), 5282–5286. <https://doi.org/10.35940/ijrte.f9761.038620>
23. Salloum, S., Huang, J. Z., & He, Y. (2019). Random sample partition: A distributed data model for big data analysis. *IEEE Transactions on Industrial Informatics*, 15(11), 5846–5854. <https://doi.org/10.1109/tii.2019.2912723>
24. Luk'yanov, L. A., Spivak, S. I., & Khristolyubov, V. L. (2016). Neironnaya set' korrektor dlya raspredeleniya rabot v zadache vnutritsekhovogo planirovaniya [Neural network corrector for job distribution in intra-shop scheduling]. *Vestnik Bashkirskogo universiteta*, 21(4), 859–863 (in Russian).
25. Efimov, E. N., & Shevgunov, T. Ya. (2012). Development and analysis of the technique for the building of artificial neural networks based on adaptive elements. *Trudy MAI*, 51 (in Russian). <https://trudymai.ru/eng/published.php?ID=29159>
26. Kokhonen T. (2017). *Samoorganizuyushchiesya karty [Self-Organizing Maps]*. Moscow: Laboratoriya znaniy (in Russian).
27. Novikova, T., & Novikov, A. (2015). Algorithms for solving problems of optimum distribution work in network canonical structures. *Forestry Engineering Journal*, 4(4), 309–317 (in Russian). <https://doi.org/10.12737/8515>
28. Pop, B., & Boian, F. (2014). Algorithms for automating task delegation in project management. *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, 2, 1191–1196. <https://doi.org/10.15439/2014f426>
29. Lopes, A. L., & Botelho, L. M. (2007). Task decomposition and delegation algorithms for coordinating unstructured multi agent systems. *First International Conference on Complex, Intelligent and Software Intensive Systems (CISIS'07)*, 209–214. <https://doi.org/10.1109/cisis.2007.52>
30. Wang, S., Li, Q., Cui, L., Yan, Z., Xu, Y., Shi, Z., Min, X., Shen, Z., & Yu, H. (2022). Towards AI-empowered crowdsourcing. *arXiv:2212.14676*. <https://doi.org/10.48550/arXiv.2212.14676>
31. Al-Anzi, F. S., Al-Zame, K., & Allahverdi, A. (2010). Weighted multi-skill resources project scheduling. *Journal of Software Engineering and Applications*, 03(12), 1125–1130. <https://doi.org/10.4236/jsea.2010.312131>

About the authors

Timofey Yakovlevich Shevgunov

Candidate of Sciences (Technology);

Associate Professor, Moscow Aviation Institute (National Research University), 4 Volokolamskoe Hwy., Moscow 125993, Russia;

Associate Professor, Department of Business Informatics, Graduate School of Business, HSE University, 26–28 Shabolovka St., Moscow 119049, Russia;

E-mail: shevgunov@gmail.com

ORCID: 0000-0003-1444-983X

Anna Alexandrovna Kroshilina

Master's student, Department of Business Informatics, Graduate School of Business, HSE University, 26–28 Shabolovka St., Moscow 119049, Russia;

E-mail: ankrosh@vk.com