

Математическое моделирование и его применение в научных исследованиях

УДК 519.7

DOI:10.25729/ESI.2026.42.2.006

Оптимизация гибкой организации большого числа работ в задаче управления командной реализацией проектов

Колесникова Светлана Ивановна, Фоменкова Анастасия Алексеевна,
Поляков Виктор Валерьевич

Санкт-Петербургский государственный университет аэрокосмического приборостроения,
Россия, Санкт-Петербург, *skolesnikova@yandex.ru*

Аннотация. Рассматривается задача управления назначением работ при большом динамически пополняемом их объеме. Несмотря на имеющиеся инструменты автоматизации разработки больших программ (AGILE-продукты, продукты Waterfall и другие), проблема оперативного и корректного вмешательства в процесс их производства при изменяющихся внешних и внутренних условиях стоит достаточно остро в силу возникающих отрицательных последствий (задержки выполнения проекта во времени, выход за рамки бюджета, незавершенность проектов из-за неудовлетворительного качества). Наиболее острая проблема в больших проектах – «оптимальное» назначение работ в соответствии с зафиксированными текущими компетенциями сотрудников, при этом отмечено в ряде исследований (опросы от Scrum Inc.), что связь между внедрением наиболее популярных продуктов Agile и ростом эффективности разработки программ носит противоречивый характер. По этой причине разработка усовершенствований имеющихся инструментов актуальна в задачах управления производством крупных программных проектов, где особенно остро стоят проблемы взаимосвязанных ресурсных ограничений: временных, стоимостных, вычислительных. Цель исследования – представление модели и реализующего ее алгоритма оптимизации процесса назначения работ с ориентацией на большие данные (с позиции субъекта-разработчика), приводящего к экспоненциальному выигрышу во времени и ресурсах при сравнении вариантов в форме альтернатив (работа, работник). Особенностью модели является оптимальный алгоритм ранжирования большого числа вариантов, выполняемый в режиме реального времени, и приводящий практически к не улучшаемому корректному решению в задаче упорядочивания динамически пополняемого множества альтернатив, что соответствует важнейшему принципу разработки «здесь и сейчас», заключающемуся в немедленной реакции на изменяющиеся условия и требования заказчика. Выигрыш достигается за счет немеханического совмещения применений трех методов: классического метода парных сравнений, его модификации, корректной в смысле аксиомы К. Эрроу о независимости выбора (предпочтений) от ранее достигнутых ранжирований, и выбранного алгоритма сортировки числовой последовательности. Приведены примеры численного моделирования, подтверждающие заявленные характеристики алгоритма выбора оптимальной альтернативы (работа, работник), в условиях больших данных.

Ключевые слова: плохо формализуемый объект, алгоритмы принятия решений, управление проектом, динамическая модель назначений работ, большие данные, корректная модификация метода парных сравнений больших данных, коллектив алгоритмов оценивания, цепь Маркова

Цитирование: Колесникова С.И. Оптимизация гибкой организации большого числа работ в задаче управления командной реализацией проектов / С.И. Колесникова, А.А. Фоменкова, В.В. Поляков // Информационные и математические технологии в науке и управлении, 2026. – № 2(42). – С. 70-86. – DOI:10.25729/ESI.2026.42.2.006.

Введение. Программный проект представляет собой продукт временной деятельности предприятия, направленной на создание уникального программного решения, удовлетворяющего потребности заказчика (человека, предприятия и др.) требуемого качества (например, [1-12]). В сфере разработки программного обеспечения на всех этапах типовых процессов (планирование, сбор требований с последующей спецификацией проекта, проектирование программной системы, реализация, тестирование, развертывание, поддержка) требуются действия (альтернативы), обоснование которых осуществляется на множестве

критериев. К наиболее распространенным технологиям управления проектами относят продукты Waterfall (каскадная методология), Agile (гибкая методология с рядом разработанных фреймворков (Scrum, Extreme Programming, Kanban и др.)), системно-инженерный подход [6], гибридный [5] и другие [7-9].

1. Краткий обзор литературы с анализом эффективности методологий управления проектами. Обзор современной литературы (например, [6-15]) с анализом текущего положения в методологиях управления проектами зачастую включает в себя доступные в открытой печати исследования с некоторой статистической информацией и выводами, склоняющими читателя к определенной точке зрения для достижения субъективных целей (с ненулевой вероятностью могущей быть опротестованной оппонентами), но факт множественности методологий свидетельствует о несовершенстве предлагаемых решений и отсутствию единых «шаблонов» в процессе управления проектами.

Следует также отметить возрастающее число сторонников движения «NoProjects», – противников использования инструментов для управления проектами в качестве инструмента для разработки программного обеспечения [11, 13] (тип проектов, который имеет самый высокий процент неудач). Так, согласно статистическим данным The Standish Group (The Chaos Report) в течение только одного года 31% исследуемых проектов завершились успешно, 50% оказались проблемными, а 19% – провальными. Аргументом для принятия такой парадигмы является следующая позиция: программа – это продукт с полным жизненным циклом создания, развития и поддержки, а не разовая акция, в отличие от проектов, фокус на которых сосредоточен только на этапе создания.

К основным причинам неуспешности проектов (в том числе согласно анализу из [8-13]) относят следующие:

- перерасход трудозатрат и/или нарушение сроков (89%);
- недостаточная квалификация топ-менеджера в каскадных проектах; что примечательно: в тех Agile-проектах, где во главе разработки стоит высококвалифицированный руководитель, провалы случаются так же часто, как и успехи;
- использование специализированных инструментов для реализации проектов при решении технических проектов (задач);
- разработка крупных проектов (табл. 1), при этом категоризация «крупный, средний, малый» рассматривается с точки зрения субъекта-разработчика;
- техническая некомпетентность (в целом: руководителя, исполнителей).

Согласно тем же отчетам [11], появляющимся дважды в год с 1994 г., у больших Agile-проектов значительно больше шансов на успех, чем у Waterfall-проектов (табл. 1), а вероятность провала Waterfall-проектов почти в 2 раза выше для средних и больших проектов.

Таблица 1. Результаты исследования Standish Group

	Agile-продукты	Каскадные продукты
Большие проекты	19%	8%
Средние проекты	31%	19%
Малые проекты	59%	45%

Таким образом, из текущего анализа [12-16] следует, что связь между внедрением продуктов Agile и ростом эффективности разработки программ имеет противоречивый характер (см., например, исследования [15]. Согласно опросам от Scrum Inc., 47 % Agile-инициатив сопровождаются задержками, выходят за рамки бюджета или приводят к недовольству клиентов.

Важно также отметить, что одной из важных проблем, влияющих на итоговые показатели реализации больших проектов [6-15], отмечена задача оптимального динамического распределения текущих работ между исполнителями, хаотичность и число которых возрастает с возникновением практически неизбежных форс-мажорных ситуаций (изменений экзо- и эндогенных факторов – требований заказчика, условий рынка, финансовых условий, состава работников и пр.).

Уместно также отметить, что, согласно анализу решений задач стратегического управления, проводимому международной консалтинговой компанией McKinsey, фирмы, преуспевающие в *персонализации* назначения работ, получают на 40% больше дохода по сравнению со средним показателем, что свидетельствует о важности корректности задачи о назначениях и перспективе конкурентного преимущества при ее реализации в условиях динамически изменяющихся наборах альтернатив большого размера.

К примерам IT-проектов с большим количеством задач и ограниченным временем на принятие решений для их распределения среди сотрудников относятся следующие:

- разработка программного обеспечения для банков: проекты по созданию или обновлению банковских систем (например, интернет-банкинга) требуют быстрого назначения разработчиков, тестировщиков и аналитиков, чтобы соответствовать строгим срокам и требованиям регуляторов;
- масштабные проекты по обслуживанию миграции данных: при миграции данных из одной системы в другую (например, при переходе на облачные решения) необходимо быстро формировать команды специалистов для анализа, очистки и переноса данных, чтобы минимизировать время простоя;
- создание мобильных приложений: в проектах по разработке мобильных приложений для крупных клиентов (например, для e-commerce) часто возникает необходимость в быстром назначении UX/UI дизайнеров, разработчиков и тестировщиков, чтобы уложиться в сроки запуска;
- внедрение ERP-систем: проекты по внедрению систем управления предприятием (например, SAP или Oracle) требуют оперативного назначения консультантов и разработчиков для настройки системы под конкретные бизнес-процессы клиента;
- проекты по кибербезопасности: в случае обнаружения уязвимостей или инцидентов в безопасности требуется быстро собрать команду специалистов для анализа ситуации и разработки мер по устранению угроз;
- разработка игр: в индустрии видеоигр часто возникают ситуации, когда необходимо быстро назначить художников, программистов и тестировщиков для выполнения задач в рамках сжатых сроков разработки и выхода на рынок;
- проекты по интеграции API: в рамках проектов по интеграции различных систем через API необходимо быстро формировать команды разработчиков и интеграторов для обеспечения совместимости и функциональности и многие другие.

Так, к примеру, в 2020 году компании Apple и Google быстро объединились для разработки технологии отслеживания контактов, чтобы помочь в борьбе с пандемией COVID-19. Это потребовало быстрой работы команд разработчиков, дизайнеров и специалистов по безопасности для создания и внедрения приложения в кратчайшие сроки.

Из анализа литературы также следует, что используемые подходы относительно процессов обработки больших данных считаются устаревшими [14-16], и ни один из инструментов работы с проектами не обладает свойством «общепринятости» ни для анализа больших данных (исключая специальные проблемно-ориентированные фреймворки), ни для оперативного распределения большого числа мелких и крупных работ в режиме реального

времени (в [14] проведен анализ 208-и последних публикаций по гибким технологиям). Таким образом, проблема big data открытая [16], несмотря на весь спектр инструментов.

2. Формализация постановки задачи о назначениях. Настоящая работа имеет целью представить корректный (безошибочный на обучающих выборках) инструмент автоматизации процесса назначений работников (в рамках задачи управления данными [17]) на основе оригинального алгоритма быстрого ранжирования работников (согласно имеющимся компетенциям) относительно возникшей работы в форс-мажорных обстоятельствах. Основой является модель последовательно выполняющихся работ и их логических взаимосвязей, заданных в виде простой однородной цепи Маркова. Особенностью процесса являются большие размерности наборов альтернатив (работа, работник).

Пусть имеется набор (смотреть, например, работы [18-20]):

$$M = \langle \mathbf{W}, \mathbf{P} \rangle,$$

где $\mathbf{W} = \{w_i | i = \overline{1, N_w}\}$ – множество работ, w_{N_w} – фиктивная работа, необходимая для обозначения завершения проекта; $\mathbf{P} = \|p_{jk} | j = \overline{1, N_w}, i = \overline{1, N_w}\|_{N_w \times N_w}$ – матрица переходов между работами, где p_{jk} – вероятность перехода от работы j к работе k ; N_w – количество работ.

Для матрицы \mathbf{P} соблюдаются естественные условия:

- $p_{jk} = 0$, если переход от работы j к работе k невозможен;
- $\sum_k p_{jk} = 1$, условие нормировки;
- $p_{jk} = 1$, при $j = k = N_w$, для обозначения завершения проекта.

Априорное распределение вероятностей работ задано в виде множества:

$$\mathbf{Q}_0 = \{q_{0_i} | i = \overline{1, N_w}\},$$

где q_{0_i} – вероятность работы w_i .

Будем полагать для определенности заданными:

- предельное (директивное) время выполнения работ в часах, заданное в виде матрицы:

$$\mathbf{TD} = \|td_i\|_{1 \times N_w},$$

где td_i – предельное (директивное) время выполнения работы w_i , $td_{N_w} = 0$ для фиктивной работы w_{N_w} ;

- предельный бюджет на выполнение каждой из работ, заданный в виде множества:

$$\mathbf{CD} = \{cd_i | i = \overline{1, N_w}\},$$

где cd_i – предельный бюджет на выполнение работы w_i , причем $cd_{N_w} = 0$ для фиктивной работы w_{N_w} ;

- штат работников разных категорий и квалификации:

$$\mathbf{E} = \{e_j | j = \overline{1, N_e}\},$$

где e_j – работник; N_e – общее количество работников;

- квалификационные группы работников и их структура:

$$\mathbf{G} = \{g_k | k = \overline{1, N_g}\},$$

где g_k – группа работников; N_g – общее количество групп; матрица принадлежности работников: $\mathbf{L} = \|l_{kj}\|_{N_g \times N_e}$, где $l_{kj} = \begin{cases} 1 \\ 0 \end{cases}$ – признак принадлежности к группе g_k работника e_j (1 – принадлежит; 0 – нет);

- сумма заработной платы в час, заданная в виде вектора:

$$\mathbf{S} = \{s_k | k = \overline{1, N_g}\},$$

где s_k – заработная плата в час работника группы g_k ;

- матрица связи между группами работников и работами (предполагается, что квалификация групп позволяет выполнить определенный вид работы):

$$\mathbf{H} = \|h_{ik}\|_{N_w \times N_g},$$

где $h_{ik} = \begin{cases} 1 \\ 0 \end{cases}$ – признак участия в работе w_i группы работников g_k (1 – участвует; 0 – нет), причем $h_{N_w N_g} = 0$ для фиктивной работы w_{N_w} ;

- оценки показателя длительности выполнения работ отдельным работником:

$$\mathbf{TP} = \|tp_{ij}\|_{N_w \times N_e},$$

где tp_{ij} – коэффициент длительности выполнения работы w_i работником e_j .

Под коэффициентом длительности выполнения работы понимается значение, на которое умножается предельное (директивное) время выполнения работы из **TD**, для последующего вычисления оценки времени.

Для матрицы **TP** соблюдаются следующие условия:

- $tp_{ij} = 0$, если работник e_j не может участвовать в работе w_i ;
- $0 < tp_{ij} \leq 1$, если работник e_j укладывается в директивные сроки работы w_i при выполнении работы без участия других работников;
- $tp_{ij} > 1$, если работник e_j не укладывается в директивные сроки работы w_i при выполнении работы без участия других работников.

На рис. 1 приведена графическая интерпретация постановки задачи, изложенной выше.

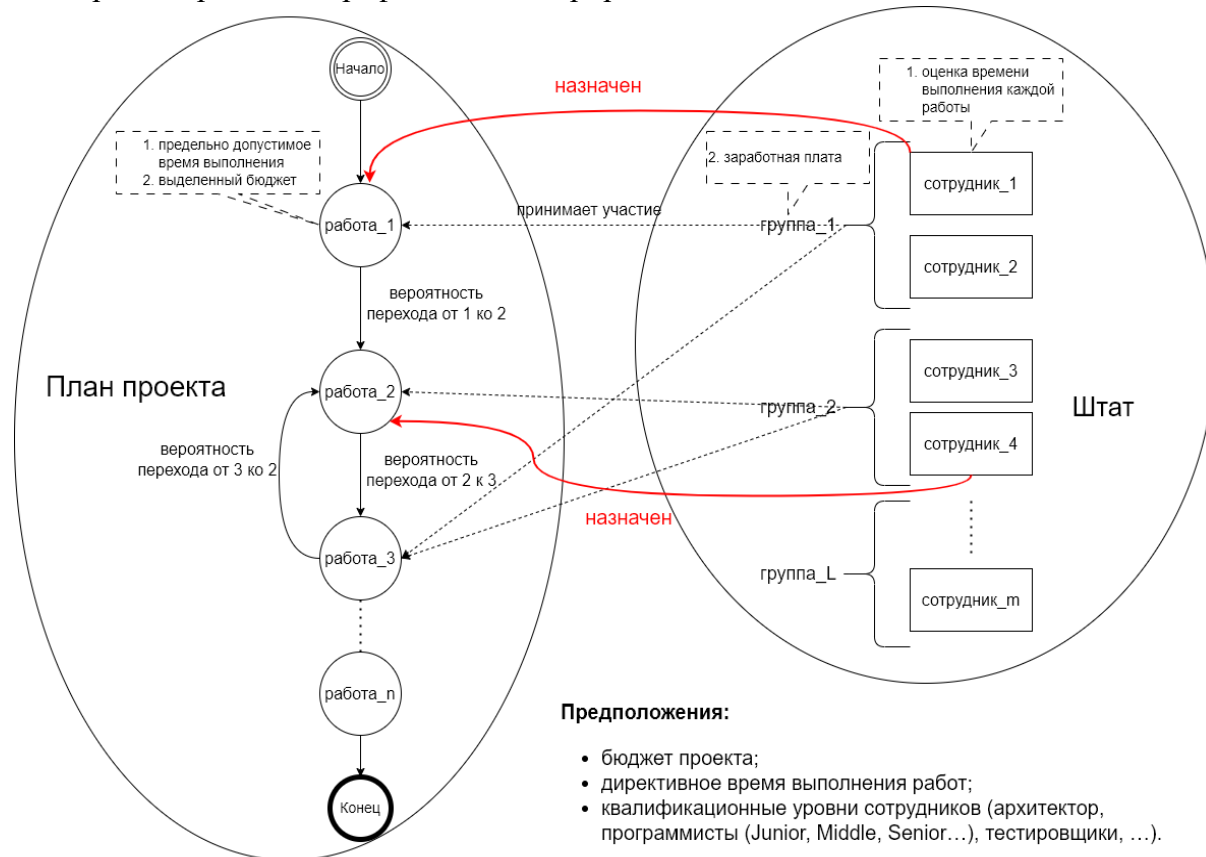


Рис. 1. Графическая интерпретация постановки задачи управления процессом командной реализации программного проекта

Пусть для каждого работника имеется «цифровая карточка» (рис. 2), содержащая сведения статистического характера, естественно обновляемые в процессе деятельности каждого работника.


Иванов	Средние числовые показатели деятельности	
Иван 	- отклонение относительно планирования спринта	46%
	- заполняемость спринта задачами	87%
	- автономность в целом:	42%
	○ степень независимости участника группы	85%
	○ уровень в системе менторства	73%
	○ степень невозвратности задач после кода-ревью/тестирования	91%
Общие статистические данные и прогноз надежности	- число завершенных задач	305
	- количество закрытых целевых установок	14
	- число спринтов с незакрытыми целями	3
	Вероятность, что задача не будет закрыта	20%

Рис. 2. Пример статистических данных в цифровой карточке работника-программиста при использовании систем управления проектами для командной работы

Ставится задача корректного ранжирования пар (работа, работник) по критериям, состав и важность которых устанавливается экспертом в зависимости от текущих особенностей проекта (длительность выполнения работы, финансовые затраты, квалификация работника и пр.) с целью совершенствования результативности решения частных задач, что будет способствовать не только повышению оперативности в принятии решений, но и корректности выбора, а, следовательно, и качества выполняемой работы.

Замечание 1. Обработка больших наборов исходных данных в режиме реального времени по нескольким критериям (показателям) – одна из основных задач в проблеме выбора. В этой связи решение задачи автоматизации процесса персонифицированных назначений в виде системы (работа, работник), или ранжирование большого числа альтернатив по предпочтительности для принятия решения о текущем выборе работника пары для фиксированного вида работы, может быть неким оптимизатором в любой известной методологии управления проектами [12-16].

В качестве альтернатив в задаче выступают всевозможные варианты назначений работников на работу:

$$A = \left\{ (w_i, e_j)_m \mid i = \overline{1, N_w}, j = \overline{1, N_e}, m = \overline{1, N_A} \right\},$$

где $(w_i, e_j)_m$ – m -я альтернатива, реализующая вариант назначения работника e_j на работу w_i ; N_A – количество альтернатив (возможных вариантов назначений).

Относительно альтернатив требуется соблюдение условий, обеспечивающих корректность их дальнейшего анализа (существование и выполнимость работ за конечное время, актуального исполнителя с определенными параметрами).

В качестве показателей оценивания выполнения задач работником выберем наиболее распространенные и используемые в реальной работе:

- автономность выполнения заданий (C1) с минимальным вовлечением других участников; например, степень $x(n)$ безвозвратности задач после проверки со значениями в зависимости от количества n отметок рецензента в тайм-трекере: $x(n), x \in (100\%, 80\%, 50\%, 0\%, 0\%, \dots 0\%), n \in [1, \infty]$, как показатель с числовой интерпретацией данной характеристики;
- полнота итерации (C2);
- относительная оценка времени выполнения работы (C3), указывающая на степень выполнения задачи работником в рамках запланированного времени:

$$T_{pc} = \sum_{i=1..k} t_{pci}; T_{fc} = \sum_{i=1..k} t_{fci}, D_c = \frac{T_{pc} - T_{fc}}{T_{pc}} \cdot 100\%,$$

где k – количество выполненных задач, T_{pc} – общее плановое время выполненных за итерацию задач, t_{pci} – плановое время на i -ю выполненную задачу, T_{fc} – общее фактическое время выполненных за итерацию задач, t_{fci} – фактическое время на i -ю задачу, D_c – отклонение от оценки по выполненным за итерацию задачам;

- денежные затраты (С5), обусловленные парой (работа, работник) и другие характеристики, обусловленные конкретным проектом.

3. Подходы к решению проблемы. К решению одной из самых востребованных задач коллективного производства продукта – организации процедуры назначения работ (или распределения пар (работа, работник)), имеющей много интерпретаций, разработаны несколько подходов (напр., обзоры в [20-31]) – аналитических и численных, на основе которых созданы многочисленные алгоритмы (что свидетельствует о сложности проблемы и об отсутствии единого подхода к ее решению, табл. 2):

- алгоритмы динамического программирования, сводящиеся к последовательности задач линейного программирования и восходящие к задаче Беллмана о кратчайшей по времени последовательности обработки деталей на нескольких станках;
- эвристические алгоритмы (экспертные решения и алгоритмы согласования частных решений, поиск оптимума на парето-множествах типа «затраты-эффект», генетические и нейросетевые алгоритмы, аукционы и др.);
- методы многокритериального принятия решений, основанные на правилах, и алгоритмы принятия решений в условиях неопределенности;
- вероятностно-статистические методы и связанные со статистической обработкой имитационные модели, близкие по реализации к моделям массового обслуживания, и другие.

Таблица 2. Краткие характеристики подходов к решению задачи о назначениях

Подходы, методы, алгоритмы	Проблемы	Отношение к неопределенности и адаптация к изменениям в процессах разработки
Доказательные аналитические методы (смотреть, например, обзоры в [20]).	Ограничение вычислительных ресурсов при больших наборах работ; требование определенных стационарных условий.	Отсутствие адаптивности относительно форс-мажорных обстоятельств, на возможные накладки в системе (работа, работник), почти неизбежно возникающие в процессе реализации проекта при параллельной разработке
Модели управления разработкой программных систем [5-17].	В условиях параллельного назначения работ на практике зачастую хаос и трудно поддающийся управлению процесс. Отсутствие автоматизации обработки больших наборов (работа, работник) в режиме реального времени. Сложность выбора модели (среди более полусотни систем управления командными проектами)	Использование принципа итеративности затруднено разными характеристиками смежных работ (приоритет, длительность, затраты) при возврате задачи на предыдущий этап.

Эвристические, вероятностно-статистические методы, имитационные модели [18, 20, 29].	Нерешенные проблемы больших данных, неприменимость статистики в хаотических режимах.	Трудоемкость процесса перенастройки параметров при изменении исходных данных.
Модели многокритериального принятия решений [24-28].	Ограничения на размер исходных данных и определенные ресурсные и временные препятствия	Негибкость к изменениям: возможны нарушения транзитивности предпочтений при увеличении множества выбора.

Таким образом, обобщая основные подходы к решению задачи корректной организации командной работы над проектом [20-31], отметим следующее:

- присущая неточность в априорных предпочтениях критериев и альтернатив по критериям может привести к неустойчивым приоритетам и ненадежному ранжированию альтернатив;
- многоцелевой характер постановок задач о назначении;
- многокритериальность, обусловленная вектором локальных критериев и интегральных показателей (например, общее время реализации проекта и совокупная стоимость проекта);
- необходимость высокопроизводительных вычислений при больших наборах исходных данных (число задач может исчисляться тысячами);
- между различными операциями и исполнителями существует взаимосвязь «многие ко многим» и др.

Уместно также отметить несколько популярных инструментов в рамках задач корректной актуализации оценок работников и формирования их приоритетов относительно конкретной работы:

- платформа Grow от компании EPAM не предоставляет возможности автоматического оценивания разработчика, но предоставляет матрицу компетенций и варианты достижения целей для каждого уровня развития;
- система оценивания работника E-Staff направлена на оценивание работников на этапе трудоустройства (с платным распространением и внедрением), формирует приоритизированный список работников, основываясь на резюме и данных из интернета; имеет ограничение на размер массива данных, обрабатываемых в реальном режиме времени и не ставит целью ранжирования предпочтений при выборе работников;
- инструменты автоматической генерации и оценки резюме на западных рынках, таких, как Zety, Resume.io, Enhancv, Rezi, использующие в том числе методы искусственного интеллекта для персонализации текстов, но не преследующие цель многокритериального ранжирования работников относительно специфики отдельной работы.

4. Предлагаемое авторами решение задачи. Модель оптимизации назначений в условиях больших данных. В основе программной реализации модели, схематично представленной на рис. 1, находится прототип со структурой, приведенной на рис. 3.

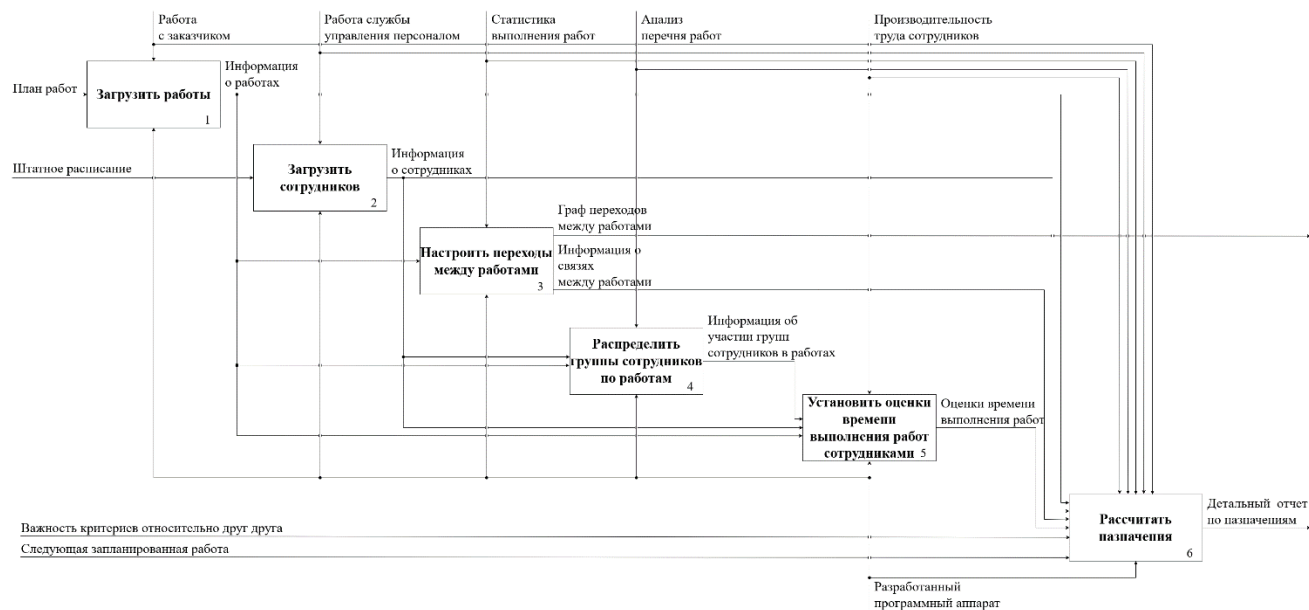


Рис. 3. Функциональная схема прототипа программы автоматизации процесса анализа в задаче оптимизации распределения работ (согласно нотации IDEF-0)

Решением основной задачи является автоматизированная процедура реализации политики назначений работников на работы, при этом важна реальность времени принятия решения при большой мощности динамически изменяемого множества альтернатив в виде пар (работа, работник). Корректный учет последнего наиболее критичен, так как в реальном рабочем процессе появление новых пар, подлежащих сравнению, удаление из набора альтернатив – явление нередкое, а, например, классический метод парных сравнений, как известно [32-35], в этом случае может выдать ошибки ранжирования в виде нарушения ранее полученных предпочтений.

Алгоритм ранжирования альтернатив относительно фиксированной работы на предмет выбора работника состоит из двух частей, первая из которых представляет собой классический метод анализа иерархий [32] (Analytical Hierarchy Process (AHP)), с выше отмеченной некорректностью при решении задачи ранжирования и весьма ограничительный при больших мощностях анализируемых множеств альтернатив и критериев.

Напомним кратко основные позиции АНР в терминах данной статьи для лучшего понимания ниже следующих алгоритмов АНР+ и АНР++ и большей прозрачности развития дальнейшей идеи применения метода оценки альтернатив для ранжирования большого объема альтернатив.

Алгоритм АНР. Как хорошо известно [32], выходом этого алгоритма является взвешенная оценка каждой альтернативы по совокупности критериев.

Входные данные: множество альтернатив $\{A_i | i = \overline{1, N_a}\}$; множество критериев $\{C_j | j = \overline{1, N_c}\}$, где N_c – количество критериев оценивания.

1. Экспертное назначение предпочтений критериев $v(C_j), j = \overline{1, N_c}, \sum_{j=1}^{N_c} v(C_j) = 1$, отражающее их относительную значимость.
2. Определение относительных предпочтений альтернатив: $d_{ij} = d(A_i/C_j), i = \overline{1, N_a}, j = \overline{1, N_c}$ («вес» альтернативы A_i в соответствии с критерием (показателем) C_j).
3. Формирование согласованных (с соблюдением свойства транзитивности) матриц парных сравнений $F_j = \|f_{ik}(j)\| = \left\| \frac{d_{ij}}{d_{kj}} \right\|_{N_a \times N_a}, i, k = \overline{1, N_a}$ относительно каждого j -го критерия,

и вычисление собственных векторов матриц парных сравнений критериев $C = \|C_{ij}\| = \|v(C_j)/v(C_l)\|$, $j, l = \overline{1, N_C}$ и альтернатив $F_j, j = \overline{1, N_C}$ с последующей их нормализацией (символ «*» означает результат операции нормализации вектора):

$$4. C^* = \|v_j^*\|_{1 \times N_C}, f_j^* = (a_{1j}^*, \dots, a_{N_A j}^*)^T, j = \overline{1, N_C}.$$

5. Получение вектора \mathbf{u} , элементы которого есть итоговые значения многокритериального оценивания альтернатив в виде их весовых коэффициентов $u_i = \sum_{j=1}^{N_C} v_j^* a_{ij}^*$, $i = \overline{1, N_A}$ с последующей их нормализацией $\mathbf{u}^* = \|u_i^*\|_{1 \times N_A}$.

Выходные данные: вектор итоговых оценок альтернатив $\mathbf{u}^* = \mathbf{u}^{\text{норм}}$.

В матричном виде итог алгоритма АНР+ имеет вид:

$$\mathbf{u} = \mathbf{F}^* \times \mathbf{C}^{*T}, \mathbf{F}^* = (f_1^* \quad \dots \quad f_{N_C}^*) = \begin{pmatrix} a_{11}^* & \dots & a_{1N_C}^* \\ \dots & \dots & \dots \\ a_{N_A 1}^* & \dots & a_{N_A N_C}^* \end{pmatrix}, \sum_{i=1}^{N_A} a_{ij}^* = 1, j = \overline{1, N_C}.$$

Замечание 2. В ряде работ [33-37] указаны нежелательные особенности данного алгоритма: нарушение соотношений для ранее полученных рангов при добавлении новых альтернатив, проблемы с непосредственным использованием числовых критериев.

Алгоритм АНР+ как корректное обобщение АНР. За счет выполнения дополнительных операций [33, 35]: а) попарной группировки компонент векторов $f_j^* = (a_{1j}^*, \dots, a_{N_A j}^*)^T, j = \overline{1, N_C}$; б) операции нормализации весовых коэффициентов по каждой группе; в) линейной скаляризации полученных оценок отдельной альтернативы из каждой группы по всем критериям, итоговые оценки весов альтернатив будут свободны от основного недостатка, а именно, – нарушения ранее достигнутых предпочтений при добавлении в набор новых альтернатив (нарушение аксиомы К. Эрроу [34] о независимости предпочтений).

Входные данные: вектор критериальных предпочтений \mathbf{C}^* и матрица \mathbf{D} , столбцы которой есть нормализованные векторы матриц парных сравнений по соответствующим критериям.

1. Формирование матрицы парных сравнений $\mathbf{D}_j, j = \overline{1, N_C}$, с элементами в виде двухкомпонентных векторов

$$\mathbf{D}_j = \|\mathbf{d}_j(i, k)\|_{N_A \times N_A}, \mathbf{d}_j(i, k) = [a_{ij}^{**}(i) \quad a_{kj}^{**}(k)], i, k = \overline{1, N_A}, j = \overline{1, N_C},$$

компоненты которых имеют значения относительных оценок по каждой паре альтернатив по отдельному критерию j (символ «**» означает результат операции нормализации в паре элементов):

$$a_{ij}^{**}(i) = \frac{a_{ij}^*(i)}{(a_{ij}^*(i) + a_{kj}^*(k))}, a_{kj}^{**}(k) = \frac{a_{kj}^*(k)}{(a_{ij}^*(i) + a_{kj}^*(k))}, i, k = \overline{1, N_A}, j = \overline{1, N_C}.$$

Замечание 3. Индексы в обозначении $\mathbf{d}_j(i, k)$ означают следующее: (i, k) есть адрес элемента матрицы \mathbf{D}_j представляющего собой относительный вес альтернатив A_i и A_k только в данной паре альтернатив по критерию C_j , и $a_{ij}^{**}(i) + a_{kj}^{**}(k) = 1, i, k = \overline{1, N_A}, j = \overline{1, N_C}$.

2. Формирование матрицы $\bar{\mathbf{D}} = \|\bar{\mathbf{d}}(i, k)\|_{N_A \times N_A}$ с векторными элементами, компоненты которых представляют собой результат скаляризации (линейную свертку) соответствующих компонент по всем критериям:

$$\bar{\mathbf{d}}(i, k) = (\bar{\mathbf{d}}(i), \bar{\mathbf{d}}(k)), \\ \bar{\mathbf{d}}(i) = \sum_{j=1}^{N_C} v_j^* a_{ij}^{**}(i), \bar{\mathbf{d}}(k) = \sum_{j=1}^{N_C} v_j^* a_{kj}^{**}(k), i, k = \overline{1, N_A}.$$

3. Нормализация элементов матрицы $\bar{\mathbf{D}}$:

$$\bar{d}^*(i) = \frac{\bar{d}(i)}{\bar{d}(i) + \bar{d}(k)}, \bar{d}^*(k) = \frac{\bar{d}(k)}{\bar{d}(i) + \bar{d}(k)}, \bar{d}^*(i) + \bar{d}^*(k) = 1, i, k = \overline{1, N_A}.$$

4. Итоговое определение весовых коэффициентов альтернатив $d(A_i), i = \overline{1, N_A}$, как сумма первых компонентов i -й строки матрицы $\bar{D}^* = \|\bar{d}^*(i, k)\|_{N_A \times N_A}, \bar{d}^*(i, k) = (\bar{d}^*(i), \bar{d}^*(k)), d(A_i) = \sum_{l=1}^{N_A} \bar{d}^*(i, l)$, с последующей нормализацией вектора весовых коэффициентов альтернатив, получаем итоговый набор $d^*(A_i), i = \overline{1, N_A}$.

Выходные данные: вектор оценок весовых коэффициентов альтернатив $d^*(A_i), i = \overline{1, N_A}$, свободных от недостатков АНР.

Замечание 4. Вычислительная трудоемкость алгоритма АНР+ оценивается как $O(N_A^2 N_C)$. Алгоритм АНР+ предъявляет достаточно высокие требования к оперативной памяти.

Финальный алгоритм АНР++, обобщающий АНР+ на ранжирование больших данных, основан на декомпозиции исходной задачи выбора на всем множестве альтернатив на последовательность задач выбора на парах альтернатив со всеми вытекающими положительными последствиями относительно проблем используемых вычислительных ресурсов.

С целью решения данной проблемы в работе [36] была предложена декомпозиция исходной задачи выбора на всем множестве альтернатив на последовательность задач выбора на паре альтернатив, выполнение каждой из которых интерпретируется как функция сравнения двух элементов согласно алгоритму сортировки числовой последовательности.

Применительно к рассматриваемым здесь задачам алгоритм АНР+, обеспечивающий непротиворечивое оценивание альтернатив при динамическом пополнении набора большой размерности, будет иметь следующее описание.

Алгоритм АНР++. Обобщение корректного алгоритма АНР+ для больших динамически пополняемых наборов альтернатив применительно к поставленной задаче имеет следующее описание.

Входные данные: вектор критериальных предпочтений C^* относительно характеристик работника (см. цифровую карточку на рис. 3) и последовательность альтернатив $A = \{A_{i_{j_0}}\}, A_{i_{j_0}} = \{w_i, e_{j_0}\}, i = \overline{1, N_w}, j_0 = \overline{1, N_e}$; индекс j_0 фиксирует вид работы, для которой осуществляется выбор работника.

Выбор алгоритма сортировки числового массива (весовых коэффициентов альтернатив):

1. Поступательное сравнение пары альтернатив A_{kj_0}, A_{lj_0} .
2. Применение алгоритма АНР+ для пары A_{kj_0}, A_{lj_0} с выходом в виде значений их весовых коэффициентов a_{kj_0}, a_{lj_0} .
3. Сравнение значений a_{kj_0}, a_{lj_0} и, при необходимости, их перестановка по убыванию весов согласно цели – получение отсортированной последовательности R .
4. Переход на шаг 1 и далее повторение п. 2–4, пока имеют место необработанные альтернативы.
5. Осуществить сортировку слиянием отсортированных частей при выборе соответствующего алгоритма сортировки при необходимости.
6. Вывод ранжированного массива оценок R .

Замечание 5. Согласно свойствам алгоритма АНР+ при добавлении в множество сравниваемых альтернатив дополнительной альтернативы ранее достигнутые предпочтения не нарушаются, но весовые коэффициенты альтернатив могут измениться. В этой связи применима любая быстрая сортировка в силу благоприятного изначального расположения элементов. В силу нарушения аксиомы К. Эрроу [34] о независимости предпочтений, алгоритм

АНР не может быть обобщен для больших данных на основе данного приема. Таким образом, решение задачи ранжирования для сколь угодно больших массивов данных (работа, работник) стало возможным только на основе АНР+ [33, 35].

Выходные данные: численно ранжированные наборы предпочтений работников относительно фиксированных работ j_0 при изменении индекса j_0 в требуемом промежутке.

Замечание 6. Алгоритмическая сложность последнего алгоритма, таким образом, определяется способом сортировки.

Сравнение решений многокритериального ранжирования набора альтернатив. Свойства алгоритма АНР++ (высокая скорость ранжирования в условиях big data, неприязнательность к используемым вычислительным ресурсам (табл. 2, рис. 4)), таким образом, обусловлены исключительно корректной модификацией АНР+ [33, 35], поскольку каждая итерация основана на применении АНР+ только для пары элементов [36].

Таблица 2. Сравнительные характеристики результатов ранжирования альтернатив-назначений в зависимости от основных параметров алгоритмов в условиях трех критериев

Параметры алгоритмов ранжирования	Длительность выполнения (с)/объем используемой памяти (мб)		
	Корректное классическое АНР+	расширение алгоритма	Алгоритм оптимизации АНР++ для больших данных
Число альтернатив (N_A)			
3000	7/648		0.116/28
10 000	-/-		0.240/51
100 000	-/-		1.965/274

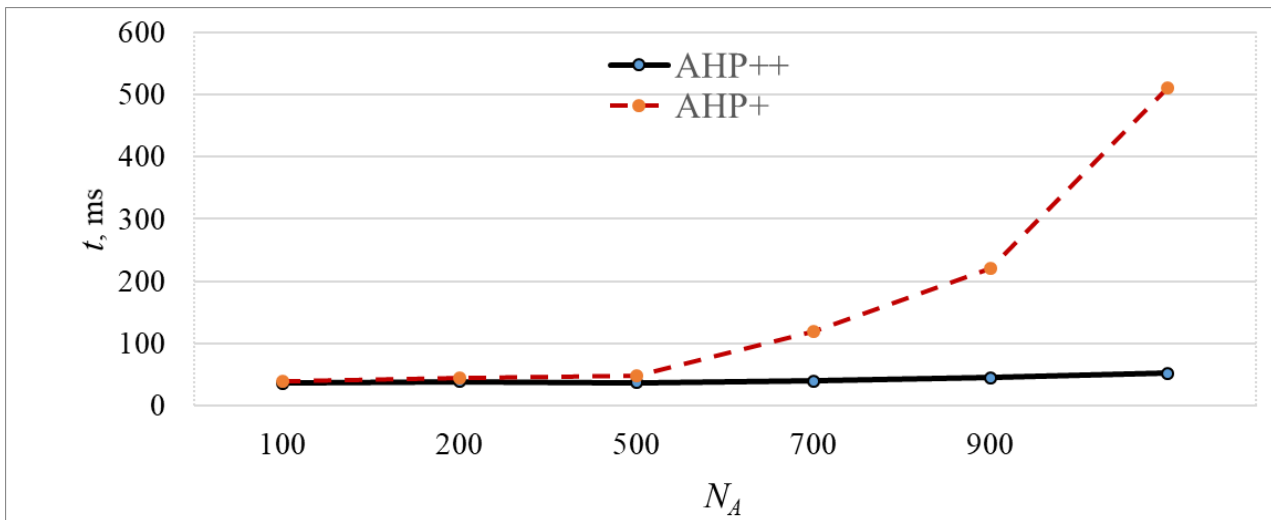


Рис. 4. Результат сравнения длительности операции ранжирования в задаче оптимизации распределения работ

На основе алгоритма АНР++ применительно к задаче выбора в контексте статьи будем иметь выходные данные (оптимальное распределение работников \mathbf{E} на отдельный вид работ \mathbf{W}), в виде матрицы:

$$\mathbf{Y} = \|\|y_{ij}\|\|_{n \times m},$$

где y_{ij} – признак назначения на работу w_i работника e_j , $i = \overline{1, N_w}$, $j = \overline{1, N_e}$; N_w – количество работ; N_e – число работников, где матрица \mathbf{Y} может содержать следующие значения:

- $y_{ij} = 1$, если работник e_j назначен на работу w_i ;
- $y_{ij} = 0$, если работник e_j не назначен на работу w_i , но может в ней участвовать;
- $y_{ij} = -1$, если работник e_j не может участвовать в работе w_i .

Заключение. Рассмотрена задача оптимизации управления назначением множества работ в процессе реализации крупных программных проектов, особенностью которых является динамическое перераспределение большого объема работ в изменяющихся условиях внешнего и внутреннего характера.

Поскольку задача оптимального распределения трудовых ресурсов имеет высокую вариативность и относится к классу задач с большим объемом данных, для ее решения привлечена корректная (непротиворечивая и алгоритмически не улучшаемая) модификация метода парных сравнений, ориентированная именно на высокоразмерные динамически изменяемые матричные массивы.

В качестве базовой функциональной модели для описания динамики выполнения работ в рамках данного исследования выбран инструмент стохастического моделирования – цепь Маркова, исключительно с целью продемонстрировать возможность формализации всего процесса реализации программного проекта большой сложности.

Заявленные свойства предложенного алгоритма, а именно, существенное преимущество по вычислительным ресурсам (табл. 2, рис. 4) перед классическим ранжированием альтернатив подтверждено численным моделированием в условиях больших данных.

Предложенный к использованию алгоритм назначений является адаптивным и позволяет корректировать связи «работа-работник» в режиме реального времени в неблагоприятных условиях, тем самым повышая устойчивость процесса планирования.

Список источников

1. Williams M. The principles of project management. Australia, Site Point Pty. Ltd., 2008, 193 p.
2. Ньютон Р. Управление проектами от А до Я. – М.: Альпина Бизнес Букс, 2007. – 180 с.
3. Макконнелл С. Сколько стоит программный проект / С. Макконнелл – М.: Русская Редакция, СПб.: Питер, 2007. – 297 с.
4. Larson E.W., Gray C.F. Project management: the management process. 8th ed. McGraw Hill, 2021, 704 p.
5. Ермаков Е.В. Применение гибридных методологий в управлении проектами / Е.В. Ермаков // Universum: экономика и юриспруденция, 2024. – 9(119).
6. Охтилев М.Ю. Производство программ. Инженерный подход: монография / М.Ю. Охтилев, В.Н. Коромысличенко, П.А. Охтилев. – Москва; Вологда: Инфра-Инженерия, 2024 – 140 с.
7. Loureiro E., Gomes B., Varajão J., Silva C. Information systems project success surveys - Insights from the last 30 years. Heliyon, 2024, vol. 10 (23), e40619, DOI: 10.1016/j.heliyon.2024.e40619.
8. Iriarte C., Bayona S. IT projects success factors: a literature review, International Journal of Information Systems and Project Management. 2020, vol. 8(2), pp. 49-78, DOI: 10.12821/ijispm080203.
9. Massimo R., Zenezini G., Ottaviani F.M. et al. Project management information systems: a systematic review. Procedia Computer Science, 2025, vol. 256, pp. 1739-1747, DOI: 10.1016/j.procs.2025.02.313.
10. PMBOK Guide. 8th ed. Newton Square, Pennsylvania, USA: Project Management Institute, 2025.
11. Standish Group et al. CHAOS report 2020. The Standish Group International, Inc. 2020, pp. 1-13.
12. Guzman E., Andres B., Poler R. Models and algorithms for production planning, scheduling and sequencing problems: a holistic framework and a systematic review. Journal of Industrial Information Integration, 2021, vol. 27, no. 2, p. 100287, DOI: 10.1016/j.jii.2021.100287.
13. Barros L., Tam C., Varajão J. Agile software development projects – unveiling the human-related critical success factors. Information and Software Technology, 2024, vol. 170, p. 107432, DOI: 10.1016/j.infsof.2024.107432.
14. Rath S.P., Jain N.K., Tomer G. et al. A systematic literature review of agile software development projects. Information and Software Technology, Elsevier, 2025.
15. Annosi M.C., Appio F.P., Martini A. Institutional context and agile team innovation: a sensemaking approach to collective knowledge creation. Technovation, 2024, vol. 129, p. 102894, DOI: 10.1016/j.technovation.2023.102894.

16. Batista T., Bronzo M., Barbosa M.W. The effects of big data analytics capabilities on agile software development practices and project performance: a dynamic capabilities view perspective. *International Journal of Managing Projects in Business*, 2025, vol. 18, no. 3, pp. 479–505.
17. Washizaki H. Guide to the software engineering body of knowledge v4.0. IEEE Computer Society, 2025, 413 p.
18. Тимофеев Д.А. Описание процессов разработки программ с помощью скрытых марковских моделей / Д.А. Тимофеев, А.В. Самочадин // *Научно-технические ведомости СПбГПУ*, 2017. – Т. 10. – № 4. – С. 70-77.
19. Savchuk T. Modeling of software development process with the Markov processes. *Eastern-European Journal of Enterprise Technologies*, 2017, vol. 3, no. 2, pp. 33-38, DOI: 10.15587/1729-4061.2017.103340.
20. Колесникова Е.В. Разработка марковской модели состояний проектно управляемой организации / Е.В. Колесникова, В.А. Вайсман, С.А. Величко // *Суч. технології в машинобуд.: зб. наук. праць. – НТУ «ХПІ», 2012. – №7. – С. 217-223.*
21. Михальцова Е.В. Разработка модели оптимизации и метода распределения работ при управлении проектами в области информационных технологий: специальность 08.00.13 «Математические и инструментальные методы экономики»: диссертация на соискание ученой степени кандидата экономических наук / Михальцова Елена Викторовна; Московский государственный университет экономики, статистики, информатики. – Москва, 2009. – 133 с.
22. Дегтерев Д.А. Математические методы решения задачи составления цеховых расписаний / Д.А. Дегтерев, А.С. Дегтерев // *Вестник Сибирского государственного аэрокосмического университета имени академика М. Ф. Решетнева*, 2011. – Т.12. – №5. – С. 29-34.
23. Карпов В.И. Оптимизация календарного плана выполнения комплекса взаимосвязанных работ в системе поддержки принятия решений / В.И. Карпов // *Прикладная информатика*, 2014. – №4(52). – С. 101-108.
24. Garcia-Zamora D., Labella Á., Ding W. et al. Large-scale group decision making: a systematic review and a critical analysis. *IEEE/CAA Journal of Automatica Sinica*, 2022, vol. 9, no. 6, pp. 949-966, DOI: 10.1109/JAS.2022.105617.
25. Magabaleh A.A., Ghraibeh L.L., Audeh A.Y. et al. Systematic review of software engineering uses of multi-criteria decision-making methods: Trends, bibliographic analysis, challenges, recommendations, and future directions. *Applied Soft Computing*, 2024, vol. 163, DOI: 10.1016/j.asoc.2024.111859.
26. Yu X., Dai H., Li L. et al. Finding the best learning to rank algorithms for effort-aware defect prediction. *Information and Software Technology*, 2023, vol. 157, p. 107162, DOI: 10.1016/j.infsof.2023.107165.
27. Xiao J., Gao Q., Yang Z. et al. Multi-round auction-based resource allocation for edge computing: maximizing social welfare. *Future Generation Computer Systems*, 2023, vol. 140, pp. 365-375, DOI: 10.1016/j.future.2022.10.028.
28. Börstler J., N. bin Ali, Petersen K. et al. Acceptance behavior theories and models in software engineering – A mapping study. *Information and Software Technology*, 2024, vol. 172, DOI: 10.1016/j.infsof.2024.107469.
29. Shao L., Liu J., Fu C. et al. Alternative ranking in trust network group decision-making: A distributionally robust optimization method. *European Journal of Operational Research*, 2025, vol. 327, iss. 3, pp. 986-1002m DOI: 10.1016/j.ejor.2025.05.052.
30. Lu C., Li X., Gao L. et al. An effective multi-objective discrete virus optimization algorithm for flexible job-shop scheduling problem with controllable processing times. *Computers & Industrial Engineering*, 2017, vol. 104, pp. 156–174, DOI:10.1016/j.cie.2016.12.020.
31. Xu W., Xu Sh., Liang J. et al. A discrete water source cycle algorithm design for solving production scheduling problem in flexible manufacturing systems. *Swarm and Evolutionary Computation*, 2025, vol. 94, DOI: 10.1016/j.swevo.2025.101897.
32. Saaty T.L. Absolute and relative measurement with the ANP. The most livable cities in the United States. *Socio-Economic Planning Sciences*, 1986, vol. 20, no. 6. pp. 327–331, DOI: 10.1016/0038-0121(86)90043-1.
33. Самохвалов Ю.Я. Групповой учет относительного превосходства альтернатив в задачах принятия решений / Ю.Я. Самохвалов // *Кибернетика и системный анализ*, 2003. – № 6. – С. 141-145.
34. Эрроу К.Дж. Коллективный выбор и индивидуальные ценности / К.Дж. Эрроу. – М.: ГУ ВШЭ, 2004. – 201 с.
35. Колесникова С.И. Модификация метода анализа иерархий для динамических наборов альтернатив / Е.В. Колесникова // *Прикладная дискретная математика*, 2009. – № 4(6). – С. 102-109.
36. Колесникова, С.И. Оптимизация алгоритма многокритериального выбора с динамически пополняемым большим набором альтернатив / С.И. Колесникова, С.А. Караванова // *Информационные технологии*, 2021. – Т. 27. – № 5. – С. 235-241.
37. Подиновский В.В. Парето-оптимальные решения многокритериальных задач / В.В. Подиновский, В.Д. Ногин. – М.: Наука, 2007. – 256 с.

Колесникова Светлана Ивановна. Д.т.н., профессор кафедры компьютерных технологий и программной инженерии Института вычислительных систем и программирования Санкт-Петербургского государственного университета аэрокосмического приборостроения. Научные интересы включают нелинейное управление с неопределенностью в описании, интеллектуальный анализ данных. AuthorID: 150312, SPIN: 1928-8626, ORCID: 0000-0001-7158-2747, skolesnikova@yandex.ru. 190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А.

Фоменкова Анастасия Алексеевна. Доцент кафедры компьютерных технологий и программной инженерии Института вычислительных систем и программирования Санкт-Петербургского государственного университета аэрокосмического приборостроения. Научные интересы: технологии программной инженерии, биотехнологии, мониторинг сложных биосистем. AuthorID: 1126590, SPIN: 3606-1805, ORCID: 0000-0002-5570-6743, a.a.fomenkova@mail.ru. 190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А.

Поляков Виктор Валерьевич. Аспирант кафедры компьютерных технологий и программной инженерии Института вычислительных систем и программирования Санкт-Петербургского государственного университета аэрокосмического приборостроения. Научные интересы: технологии разработки больших проектов, модели оптимизации обработки больших данных. AuthorID: 1267137, SPIN: 1914-4619, ORCID: 0009-0005-9314-4752, viktorpolyakov1999@mail.ru. 190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А.

UDC 519.7

DOI:10.25729/ESI.2026.42.2.006

Optimization of assignment of a large number of works in the problem of managing team project implementation

Svetlana I. Kolesnikova, Anastasiya A. Fomenkova, Viktor V. Polyakov

St.Petersburg State University of Aerospace Instrumentation,
Russia, St. Petersburg, skolesnikova@yandex.ru

Abstract. A problem of control assignment of works with a large dynamically replenished (changed) volume of them is considered. Despite existing tools for automating large-scale software development (AGILE products, Waterfall, and others), the challenge of promptly and correctly intervening in their production process under changing external and internal conditions remains acute due to the resulting negative consequences (project delays, budget overruns, and unfinished projects due to unsatisfactory quality). The most pressing issue in large projects is the optimal assignment of work in accordance with the established current competencies of employees. However, several studies (such as surveys from Scrum Inc.) have noted that the relationship between the implementation of the most popular Agile products and increased software development efficiency is inconsistent. For this reason, developing improvements to existing tools is relevant for managing the production of large software projects, where interrelated resource constraints time, cost, and computational are particularly acute. The objective of this study is to present a model and its implementing algorithm for optimizing the work assignment process with a focus on big data (from the perspective of the developer), resulting in exponential gains in time and resources when comparing options in the form of alternatives (job, worker). A distinctive feature of the model is an optimal algorithm for ranking a large number of options, executed in real time, and resulting in a virtually unimprovable correct solution to the problem of sorting a dynamically replenished set of alternatives, which corresponds to the most important principle of “here and now” development, consisting of an immediate response to changing conditions and customer requirements. The gain is achieved through a non-mechanical combination of three methods: the classical paired comparison method, its modification, correct in the sense of K. Arrow's axiom on the independence of choice (preferences) from previously achieved rankings, and the selected algorithm for sorting the numerical sequence. Examples of numerical simulation are provided, confirming the declared characteristics of the algorithm for selecting the optimal alternative (job, worker) in a big data environment.

Keywords: poorly formalizable object, decision-making algorithms, project management, dynamic work assignment model, big data, modification of the pairwise comparison method for big data, collective of evaluation algorithms, Markov chain

References

1. Williams M. The principles of project management. Australia, Site Point Pty. Ltd., 2008, 193 p.
2. Newton R. Upravleniye proyektami ot A do YA. [Project management from A to Z]. Moskva: Al'pina Biznes Buks [Moscow: Russian edition], 2007, 180 p.

3. Makkonnell S. Skol'ko stoit programmnyj proekt [How much does a software project cost]. Economics and jurisprudence: electronic scientific journal [Moscow: Russian edition], Sankt-Peterburg: Piter [St. Petersburg: Peter], 2007, 297 p.
4. Larson E.W., Gray C.F. Project management: the management process. 8th ed. McGraw Hill, 2021, 704 p.
5. Ermakov E.V. Primeneniye gibridnykh metodologiy v upravlenii proyektami [Application of hybrid methodologies in project management]. Universum: ekonomika i yurisprudentsiya: elektron. nauchn. Zhurnal, [Economics and jurisprudence: electronic scientific journal], 2024, 9(119).
6. Ohtilev M.Ju. Proizvodstvo programm. Inzhenernyj podhod: monografiya [Production of programs. The engineering approach: a monograph]. Moskva; Vologda: Infra-Inzheneriya [Moscow; Vologda: Infra-Engineering], 2024, 140 p.
7. Loureiro E., Gomes B., Varajão J., Silva C. Information systems project success surveys - Insights from the last 30 years. Heliyon, 2024, vol. 10 (23), e40619, DOI: 10.1016/j.heliyon.2024.e40619.
8. Iriarte C., Bayona S. IT projects success factors: a literature review, International Journal of Information Systems and Project Management. 2020, vol. 8(2), pp. 49-78, DOI: 10.12821/ijispm080203.
9. Massimo R., Zenezini G., Ottaviani F.M. et al. Project management information systems: a systematic review. Procedia Computer Science, 2025, vol. 256, pp. 1739-1747, DOI: 10.1016/j.procs.2025.02.313.
10. PMBOK Guide. 8th ed. Newton Square, Pennsylvania, USA: Project Management Institute, 2025.
11. Standish Group et al. CHAOS report 2020. The Standish Group International, Inc. 2020, pp. 1-13.
12. Guzman E., Andres B., Poler R. Models and algorithms for production planning, scheduling and sequencing problems: a holistic framework and a systematic review. Journal of Industrial Information Integration, 2021, vol. 27, no. 2, p. 100287, DOI: 10.1016/j.jii.2021.100287.
13. Barros L., Tam C., Varajão J. Agile software development projects – unveiling the human-related critical success factors. Information and Software Technology, 2024, vol. 170, p. 107432, DOI: 10.1016/j.infsof.2024.107432.
14. Rath S.P., Jain N.K., Tomer G. et al. A systematic literature review of agile software development projects. Information and Software Technology, Elsevier, 2025.
15. Annosi M.C., Appio F.P., Martini A. Institutional context and agile team innovation: a sensemaking approach to collective knowledge creation. Technovation, 2024, vol. 129, p. 102894, DOI: 10.1016/j.technovation.2023.102894.
16. Batista T., Bronzo M., Barbosa M.W. The effects of big data analytics capabilities on agile software development practices and project performance: dynamic capabilities view perspective. International Journal of Managing Projects in Business, 2025, vol. 18, no. 3, pp. 479–505.
17. Washizaki H. Guide to the software engineering body of knowledge v4.0. IEEE Computer Society, 2025, 413 p.
18. Timofeev D.A. Opisaniye processov razrabotki programm s pomoshh'ju skrytyh markovskikh modelej [Description of software development processes using hidden Markov models]. Nauchno-tehnicheskie vedomosti SPbGPU [Scientific and technical bulletin of SPbPU], 2017, vol. 10, no. 4, pp. 70-77.
19. Savchuk T. Modeling of software development process with the Markov processes. Eastern-European Journal of Enterprise Technologies, 2017, vol. 3, no. 2, pp. 33-38, DOI: 10.15587/1729-4061.2017.103340.
20. Kolesnikova E.V. Razrabotka markovskoj modeli sostojanij proektno upravljajemoj organizacii [Development of a Markov model of the states of a project-driven organization]. Suchasni tekhnologii v mashinobuduvanni [Modern Technologies in Mechanical Engineering], NTU "KhPI", 2012, no.7, pp. 217-223.
21. Mihal'cova E.V. Razrabotka modeli optimizacii i metoda raspredelenija rabot pri upravlenii proektami v oblasti informacionnyh tehnologij: special'nost' 08.00.13 «Matematicheskie i instrumental'nye metody jekonomiki»: dissertacija na soiskanie uchenoj stepeni kandidata jekonomicheskikh nauk [Development of an optimization model and a method for distributing work in project management in the field of information technology: specialty 08.00.13 «Mathematical and instrumental methods of Economics»: dissertation for the degree of Candidate of Economic Sciences], Moskva, 2009, 133 p.
22. Degterev D.A. Matematicheskie metody reshenija zadachi sostavlenija cehovyh raspisanij [Mathematical methods for solving the problem of drawing up shop schedules]. Vestnik Sibirskogo gosudarstvennogo aerokosmicheskogo universiteta imeni akademika M.F. Reshetneva [Bulletin of Reshetnev Siberian State Aerospace University], 2011, vol.12, no.5, pp. 29-34.
23. Karpov V.I. Optimizacija kalendarnogo plana vypolnenija kompleksa vzaimosvjazannyh rabot v sisteme podderzhki prinjatija reshenij [Optimization of the calendar plan for the execution of a complex of interrelated works in the decision support system]. Prikladnaya informatika [Applied Informatics], 2014, no.4(52), pp. 101-108.
24. Garcia-Zamora D., Labella Á., Ding W. et al. Large-scale group decision making: a systematic review and a critical analysis. IEEE/CAA Journal of Automatica Sinica, 2022, vol. 9, no. 6, pp. 949-966, DOI: 10.1109/JAS.2022.105617.

25. Magabaleh A.A., Ghraibeh L.L., Audeh A.Y. et al. Systematic review of software engineering uses of multi-criteria decision-making methods: Trends, bibliographic analysis, challenges, recommendations, and future directions. *Applied Soft Computing*, 2024, vol. 163, DOI: 10.1016/j.asoc.2024.111859.
26. Yu X., Dai H., Li L. et al. Finding the best learning to rank algorithms for effort-aware defect prediction. *Information and Software Technology*, 2023, vol. 157, p. 107162, DOI: 10.1016/j.infsof.2023.107165.
27. Xiao J., Gao Q., Yang Z. et al. Multi-round auction-based resource allocation for edge computing: maximizing social welfare. *Future Generation Computer Systems*, 2023, vol. 140, pp. 365-375, DOI: 10.1016/j.future.2022.10.028.
28. Börstler J., N. bin Ali, Petersen K. et al. Acceptance behavior theories and models in software engineering – A mapping study. *Information and Software Technology*, 2024, vol. 172, DOI: 10.1016/j.infsof.2024.107469.
29. Shao L., Liu J., Fu C. et al. Alternative ranking in trust network group decision-making: A distributionally robust optimization method. *European Journal of Operational Research*, 2025, vol. 327, iss. 3, pp. 986-1002m DOI: 10.1016/j.ejor.2025.05.052.
30. Lu C., Li X., Gao L. et al. An effective multi-objective discrete virus optimization algorithm for flexible job-shop scheduling problem with controllable processing times. *Computers & Industrial Engineering*, 2017, vol. 104, pp. 156–174, DOI:10.1016/j.cie.2016.12.020.
31. Xu W., Xu Sh., Liang J. et al. A discrete water source cycle algorithm design for solving production scheduling problem in flexible manufacturing systems. *Swarm and Evolutionary Computation*, 2025, vol. 94, DOI: 10.1016/j.swevo.2025.101897.
32. Saaty T.L. Absolute and relative measurement with the AHP. The most livable cities in the United States. *Socio-Economic Planning Sciences*, 1986, vol. 20, no. 6. pp. 327–331, DOI: 10.1016/0038-0121(86)90043-1.
33. Samohvalov, Ju.Ja. Gruppovoj uchet odnositel'nogo prevoshodstva al'ternativ v zadachah prinjatija reshenij [Group consideration of the relative superiority of alternatives in decision-making tasks]. *Kibernetika i sistemnyj analiz [Cybernetics and system analysis]*, 2003, no. 6, pp. 141-145.
34. Jerrou K. Dzh. Kollektivnyj vybor i individual'nye cennosti [Collective choice and individual values]. Moskva: GU VShJe [Moscow: HSE], 2004, 201 p.
35. Kolesnikova S.I. Modifikacija metoda analiza ierarhij dlja dinamicheskikh naborov al'ternativ [Modification of the hierarchy analysis method for dynamic sets of alternatives]. *Prikladnaja diskretnaja matematika [Applied discrete mathematics]*, 2009, no. 4(6), pp. 102-109.
36. Kolesnikova, S.I. Optimizacija algoritma mnogokriterial'nogo vybora s dinamicheski popolnjaemym bol'shim naborom al'ternativ [Optimization of a multi-criteria choice algorithm with a dynamically expanding large set of alternatives]. *Informatsionnyye tekhnologii [Information Technologies]*, 2021, vol. 27, no. 5, pp. 235-241.
37. Podinovskij V.V., Nogin V. D. Pareto-optimal'nye reshenija mnogokriterial'nyh zadach [Pareto-optimal solutions to multi-criteria problems]. Moskva: Nauka [Science], 2007, 256 p.

Kolesnikova Svetlana Ivanovna. Doctor of Technical Sciences, Professor, Department of Computer Technologies and Software Engineering, Institute of Computing Systems and Programming, St. Petersburg State University of Aerospace Instrumentation. Scientific interests include non-linear control with uncertainty in the description, data mining. AuthorID: 150312, SPIN: 1928-8626, ORCID: 0000-0001-7158-2747, skolesnikova@yandex.ru. 190000, St. Petersburg, st. Bolshaya Morskaya, 67, lit. A.

Fomenkova Anastasiya Alekseevna. Postgraduate student of the Department of Computer Technologies and Software Engineering, Institute of Computing Systems and Programming, St. Petersburg State University of Aerospace Instrumentation. Scientific interests: technologies for developing large projects, models for optimizing big data processing. AuthorID: 1126590, SPIN: 3606-1805, ORCID: 0000-0002-5570-6743, a.a.fomenkova@mail.ru. 190000, St. Petersburg, st. Bolshaya Morskaya, 67, lit. A.

Polyakov Viktor Valerievich. Associate Professor, Department of Computer Technology and Software Engineering, Institute of Computing Systems and Programming, Saint Petersburg State University of Aerospace Instrumentation. Scientific interests: software engineering technologies, biotechnology, and monitoring of complex biological systems. AuthorID: 1267137, SPIN: 1914-4619, ORCID: 0009-0005-9314-4752, viktorpolyakov1999@mail.ru. 190000, St. Petersburg, st. Bolshaya Morskaya, 67, lit. A.

Статья поступила в редакцию 18.08.2025; одобрена после рецензирования 10.02.2026; принята к публикации 14.05.2026.

The article was submitted 08/18/2025; approved after reviewing 02/10/2026; accepted for publication 05/14/2026.