

УДК 519.16+519.85

ОТКАЗОУСТОЙЧИВЫЕ СЕМЕЙСТВА ПЛАНОВ ПРОИЗВОДСТВА: МАТЕМАТИЧЕСКАЯ МОДЕЛЬ, ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ И АЛГОРИТМЫ ВЕТВЕЙ И ГРАНИЦ¹⁾

© 2024 г. Ю. Ю. Огородников^{1,*}, Р. А. Рудаков^{1,**}, Д. М. Хачай^{2,***}, М. Ю. Хачай^{1,****}

¹ 620108 Екатеринбург, ул. С. Ковалевской, 16, ИММ им. Н. Н. Красовского УрО РАН, Россия

² 33405 Talence, 680 cours Libération, KEDGE Business School, France

*e-mail: yogorodnikov@imm.uran.ru

**e-mail: r.a.rudakov@gmail.com

***e-mail: daniil.khachai@kedgebs.com

****e-mail: mkhachay@imm.uran.ru

Поступила в редакцию 28.10.2023 г.

Переработанный вариант 28.11.2023 г.

Принята к публикации 05.03.2024 г.

Вопросы проектирования устойчивых к сбоям систем производства и поставок продукции составляют одно из приоритетных направлений развития современного исследования операций. Традиционный подход к моделированию таких систем основывается на привлечении вероятностных моделей, описывающих выбор возможного сценария действий в случае возникновения неполадок в производственной или транспортной сети. Наряду с рядом преимуществ данный подход обладает известным недостатком. Возникновение неполадок неизвестной природы, способных поставить под угрозу работоспособность всей моделируемой системы, существенно затрудняют его применение. В данной работе вводится в рассмотрение минимаксная задача построения отказоустойчивых планов производства (Reliable Production Process Design Problem, RPPDP), целью которой является обеспечение бесперебойного функционирования распределенной производственной системы при минимальных гарантированных издержках. Показывается, что задача RPPDP NP-трудна в сильном смысле и сохраняет труднорешаемость при достаточно специфических условиях. Для поиска точных и приближенных решений с оценками точности для данной задачи разработаны методы ветвей и границ, основанные на предложенной компактной модели смешанного целочисленного линейного программирования (Mixed Integer Linear Program, MILP) и авторской эвристике адаптивного поиска в больших окрестностях (Adaptive Large Neighborhood Search, ALNS) в рамках расширений известного MIP-солвера Gurobi. Высокая производительность и взаимодополняемость предложенных алгоритмов подтверждена результатами численных экспериментов, проведенных на разработанной авторами открытой библиотеке тестовых примеров, содержащей адаптированные постановки задач из библиотеки PCGTSP LIB. Библ. 25. Фиг. 5. Табл. 3.

Ключевые слова: задача проектирования отказоустойчивых производственных процессов, MILP-модель, метод ветвей и границ, эвристика адаптивного поиска в больших окрестностях.

DOI: 10.31857/S0044466924060058, EDN: XYUTPT

1. ВВЕДЕНИЕ

Проектирование распределенных производственных систем, сохраняющих работоспособность при условиях вероятных сбоев в цепях поставок комплектующих и готовой продукции, является одной из наиболее актуальных проблем современного исследования операций.

В процессе промышленного производства часто возникают непредвиденные обстоятельства, связанные с недостаточной надежностью цепей поставок, непредсказуемым задержкам в поступлении

¹⁾ Работа Ю. Ю. Огородникова и М. Ю. Хачая выполнена при финансовой поддержке РНФ (грант 22-21-00672).

комплектующих, недостаточными запасами критически значимых ресурсов и сбоями в транспортной сети [1]. Один из традиционных подходов к решению возникающих проблем основан на построении специализированных вероятностных моделей [2], описывающих возможные нарушения производственно-транспортной системы в контексте предопределенного набора сценариев. Несмотря на ряд преимуществ, данный подход становится малоэффективным в случае возникновения в процессе функционирования системы сбоя, неизвестного на этапе моделирования. В этой ситуации более подходящей альтернативой представляется построение минимаксных детерминированных моделей, связанных с поиском решений, оптимизирующих гарантированный результат.

Производственным процессом договоримся называть совокупность из m взаимосвязанных производственных операций, удовлетворяющих заданному отношению (частичного или линейного) порядка. Произвольная операция может быть выполнена на одном из взаимозаменяемых пунктов производства. Каждый пункт производства может выполнять единственную операцию, что приводит к образованию m производственных кластеров. В соответствии с заданным порядком продукция, произведенная в кластере-предшественнике, транспортируется в пункты производства кластера-последователя, где используется в качестве исходного ресурса.

Для описания перевозок производимой и потребляемой продукции введем в рассмотрение *транспортную сеть* — реберно-взвешенный ориентированный граф, множество вершин которого включает выделенные вершины s и t , характеризующие начало и завершение производственного процесса, кластерные вершины, соответствующие описанным выше пунктам производства, а также специализированные вершины, именуемые *транспортными хабами*. Каждый хаб обладает ценой открытия — единоразовой стоимостью включения его в производственный процесс и пропускной способностью, ограничивающей число обслуживаемых им маршрутов. Как обычно транспортные издержки моделируются весовой функцией, заданной на дугах сети.

Производственным планом мы называем подграф описанной выше транспортной сети, включающий вершины s и t , по одной вершине-представителю для каждого производственного кластера и направленные маршруты, соединяющие вершины подграфа в соответствии с заданным порядком и учетом пропускной способности транспортных хабов. Стоимость плана определяется суммарными транспортными издержками и ценами открытия используемых им хабов. По определению стоимость семейства планов совпадает с максимальной стоимостью входящих в него планов.

Цель вводимой в рассмотрение задачи проектирования отказоустойчивых производственных процессов (Reliable Production Process Design Problem, RPPDP) состоит в построении семейства минимальной стоимости, состоящего из k независимых производственных планов (пересекающихся только в начальной и конечной вершинах s и t).

2. ИЗВЕСТНЫЕ РЕЗУЛЬТАТЫ

Вероятно наиболее близкой к рассматриваемой в данной работе минимаксной задаче построения отказоустойчивых планов производства является задача о гомеоморфном вложении графа (Subgraph Homeomorphism Problem, SHP), введенная в известной работе [3]. Фактически задача SHP представляет собой семейство комбинаторных задач, параметризованное орграфами P , подлежащими вложению и именуемыми *паттернами*. В каждой задаче данного семейства требуется установить существование взаимно однозначного отображения $\chi: V(P) \rightarrow V(G)$ множества вершин паттерна P во множество вершин заданного орграфа G так, что произвольной дуге (a, b) графа P соответствует простой путь, соединяющий вершины $\chi(a)$ и $\chi(b)$ графа G , причем все построенные пути являются вершинно непересекающимися. Как показано авторами [3], труднорешаемость подклассов задачи SHP полностью определяется структурой графа P . В частности, NP-полной при каждом фиксированном $k \geq 2$ является задача о k вершинно непересекающихся путях (k -DP) (см., например, [4]), в которой для заданного набора адресных пар (s_i, t_i) , $i = \overline{1, k}$, требуется выяснить, существует ли в орграфе G семейство попарно вершинно непересекающихся s_i - t_i -путей.

Другой близкой к рассматриваемой нами задаче является известная задача о кратчайшем пути с ограничениями (Constrained Shortest Path Tour Problem, CSPTP) [5]. Условие этой задачи описы-

вается реберно-взвешенным ориентированным графом $G = (V \cup \{s, t\}, A)$, упорядоченным множеством попарно непересекающихся подмножеств вершин $\mathfrak{Z} = \{T_1, \dots, T_k\}$ и весовыми функциями $c, q: A \rightarrow \mathbb{R}^+$, определяющими транспортные издержки и пропускную способность для каждой дуги графа G . Цель задачи состоит в построении кратчайшего s - t -маршрута, посещающего некоторую систему представителей подмножеств семейства $\mathfrak{Z}$ в соответствии с заданным линейным порядком.

В работах [5, 6] обоснована труднорешаемость задачи CSPTP, предложен эвристический алгоритм жадного рандомизированного адаптивного поиска (GRASP) и несколько вариантов метода ветвей и границ. Авторами статьи [7] разработан алгоритм ветвей и оценок (branch-and-price), опирающийся на улучшенную модель смешанного целочисленного линейного программирования и показавший наилучшую производительность для данной задачи.

Задача о кратчайшем простом пути с k обязательными к посещению узлами (Shortest Simple Path Problem with k Must-Pass Nodes, SSPP- k -MPN), введенная в рассмотрение в статье [8], также представляется близкой к исследуемой нами задаче. Условие задачи SSPP- k -MPN задается реберно-взвешенным орграфом $G = (V, E, c)$, неупорядоченным подмножеством $T \subset V$ мощности k и выделенными начальной и конечной вершинами s и t . Задача состоит в построении кратчайшего простого s - t -пути, посещающего вершины множества T в произвольном порядке, причем каждую в точности один раз. В статье [9] показано, что задача SSPP- k -MPN NP-трудна в сильном смысле при произвольном фиксированном $k \geq 1$. Для данной задачи известны несколько моделей смешанно-целочисленного линейного программирования (Mixed-Integer Linear Program, MILP-моделей) [10] и специализированный метод ветвей и границ [9].

Ряд общих свойств с задачей RPPDP имеет и известная обобщенная задача коммивояжера с ограничениями предшествования (Precedence Constrained Generalized Traveling Salesman Problem, PCGTSP) [11]. Постановка PCGTSP задается реберно-взвешенным орграфом G , множество вершин которого разбито на попарно непересекающиеся подмножества-кластеры C_i , $i = 1, m$. Вспомогательный ациклический орграф Π задает частичный порядок, в соответствии с которым должны посещаться кластеры произвольным допустимым маршрутом. Цель задачи состоит в построении кратчайшего простого маршрута, начинающегося и завершающегося в начальном кластере C_1 , посещающего каждый кластер в единственной вершине в соответствии с заданным порядком. Задача PCGTSP NP-трудна в сильном смысле [12] в общей постановке и полиномиально разрешима при $m = O(\log n)$ [13]. Для нее разработана серия точных и приближенных алгоритмов, среди которых выделяются эффективные алгоритмы для ряда специальных случаев ограничений предшествования [14, 15, 16], эвристический солвер PCGLNS [17], схема динамического программирования и метод ветвей и границ [13], опирающиеся на эвристику PCGLNS [17] и подход [18]. На данный момент рекордным по производительности для данной задачи является метод ветвей и отсечений [19], эффективно использующий полиэдральные свойства допустимого множества задачи PCGTSP. С помощью данного подхода удалось существенно продвинуться в решении примеров известной библиотеки тестовых примеров PCGTSP LIB [20].

Как следует из приведенного обзора, имеющийся на данный момент математический аппарат позволяет эффективно находить кратчайшие маршруты в транспортных сетях при различных дополнительных ограничениях. Тем не менее, обсуждавшиеся выше комбинаторные задачи не вполне соответствуют моделированию устойчивых к сбоям производственных процессов, так как единственный оптимальный подграф (путь, маршрут и т.п.), поиск которого является их целью, может оказаться уязвимым к возможным неполадкам в транспортной сети.

В работе получены следующие результаты.

- Предложена постановка задачи RPPDP и обоснована ее труднорешаемость.
- Разработаны алгоритмы ветвей и отсечений, основанные на предложенной компактной MILP-модели и оригинальной эвристики адаптивного поиска и реализованные в виде расширений известного MIP-солвера Gurobi [21].
- Разработана открытая библиотека тестовых примеров для задачи RPPDP, расширяющая известную библиотеку PCGTSP LIB.

• Эффективность предложенных алгоритмов подтверждена результатами проведенных численных экспериментов.

Статья имеет следующую структуру. В разд. 3 вводится математическая постановка задачи RPPDP, труднорешаемость которой обосновывается в разд. 4. Раздел 5 содержит описание предлагаемой MILP-модели, обоснование ее корректности и компактности. Раздел 6 посвящен описанию предлагаемых алгоритмов. В разд. 7 обсуждаются результаты проведенного численного эксперимента. Наконец, в разд. 8 подводятся итоги работы.

3. ПОСТАНОВКА ЗАДАЧИ

Постановка задачи RPPDP задается упорядоченной тройкой (G, Π, k) , в которой

- $G = (V, E, c)$ — реберно-взвешенный орграф, в котором
 - задано разбиение множества вершин $V = \mathcal{M} \cup \mathcal{H}$;
 - выделены вершины s и t , представляющие собой начало и конец произвольного производственного плана;
 - $\mathcal{M} = \mathcal{M}_0 \cup \dots \cup \mathcal{M}_{m+1}$, где $\mathcal{M}_0 = \{s\}$, $\mathcal{M}_{m+1} = \{t\}$, а кластер $\mathcal{M}_j$, $j = \overline{1, m}$, объединяет однородные пункты производства, исполняющие идентичную операцию j ;
 - $\mathcal{H}$ — множество транспортных хабов; произвольный хаб $h \in \mathcal{H}$ характеризуется неотрицательными пропускной способностью q_h , ограничивающей сверху число входящих (исходящих) дуг произвольного допустимого решения, инцидентных вершине h , и ценой открытия C_h ;
 - весовая функция $c: E \rightarrow \mathbb{R}_+$ определяет транспортные издержки c_e , связанные с перевозками по дуге $e \in E$;
- $\Pi = (\{0, 1, \dots, m+1\}, A)$ — ациклический орграф, задающий порядок на множестве производственных кластеров, минимальный и максимальный элементы которого соответствуют кластерам $\mathcal{M}_0$ и $\mathcal{M}_{m+1}$;
- $k \geq 1$ — размер искомого семейства производственных планов.

Задача состоит в построении семейства попарно вершинно-непересекающихся производственных планов $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$ таких, что

- каждый план P_r представляет собой ориентированный ациклический подграф, получаемый из графа Π путем замены произвольной дуги $(i, j) \in A$ простым маршрутом $v_i, h_1, \dots, h_p, v_j$, начинающимся в кластере $\mathcal{M}_i$, посещающим некоторое число хабов, и заканчивающимся в кластере $\mathcal{M}_j$;
- стоимость каждого плана P_r состоит из суммы весов всех входящих в него дуг, а также стоимостей открытия задействованных хабов

$$\text{cost}(P_r) = \sum_{e \in P_r} c_e + \sum_{h \in P_r} C_h;$$

- семейство $\mathcal{P}$ обладает минимальной возможной стоимостью, задаваемой соотношением

$$\text{cost}(\mathcal{P}) = \max\{\text{cost}(P_r) : r = \overline{1, k}\}.$$

Фиг. 1 иллюстрирует введенную выше постановку.

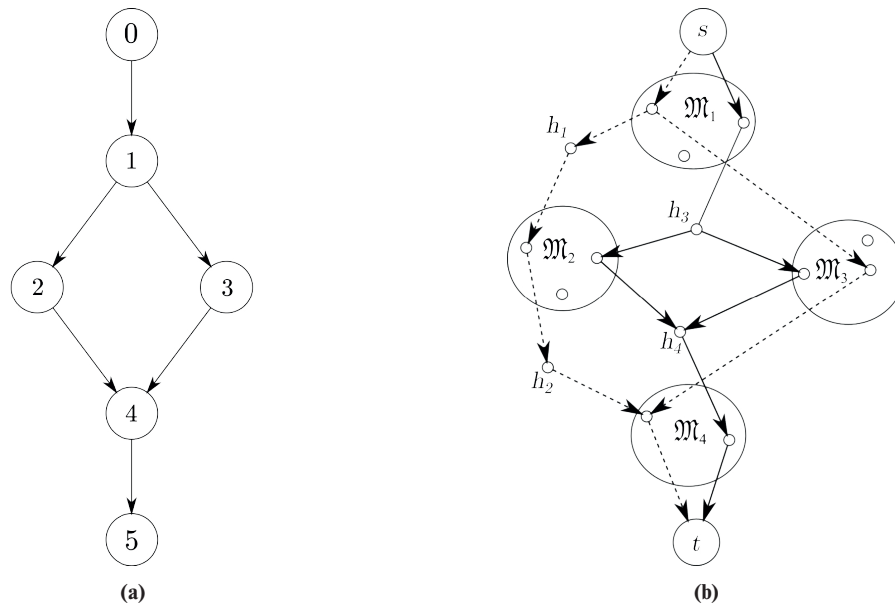
4. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ

Раздел посвящен труднорешаемости задачи RPPDP, обоснование которой опирается на полиномиальную сводимость к данной задаче NP-полной задачи 2-DP [3].

Теорема 1. *Задача RPPDP NP-трудна в сильном смысле.*

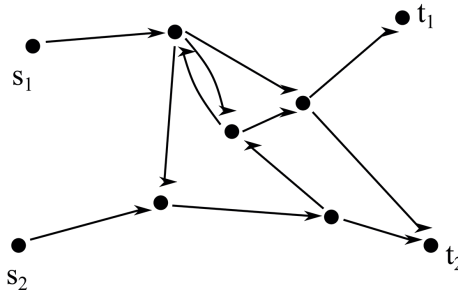
Доказательство. Рассмотрим произвольную постановку I_1 задачи 2-DP, задаваемую орграфом $G = (V, E)$ на n вершинах и адресными парами (s_1, t_1) и (s_2, t_2) , где s_1, s_2, t_1 и t_2 — попарно различные вершины графа G , и требуется проверить существование вершинно непересекающихся s_1 - t_1 и s_2 - t_2 -путей (см. фиг. 2).

Сопоставим данной постановке постановку $I_2 = (G', \Pi, 1)$ задачи RPPDP, в которой



Фиг. 1. Пример постановки RPPDP для $m = 4$ и $k = 2$. (а) граф порядка Π (б) граф G , сплошными и пунктирными линиями обозначены первый и второй планы некоторого допустимого решения.

G



Фиг. 2. Пример постановки I_1 задачи 2-DP.

1) оргграф $G' = (V', E')$ является дизъюнктивным объединением двух копий графа G , размещенных в верхнем и нижнем уровнях, дополненным стартовой и завершающей вершинами s и t и вспомогательным кластером-переключателем $\mathfrak{M}'$ (см. фиг. 3б);

2) кластеры $\mathfrak{M}_0 - \mathfrak{M}_5$ определяются соотношениями $\mathfrak{M}_0 = \{s\}$, $\mathfrak{M}_1 = \{s_1^u, s_1^l\}$, $\mathfrak{M}_2 = \{s_2^u, s_2^l\}$, $\mathfrak{M}_3 = \{t_1^u, t_1^l\}$, $\mathfrak{M}_4 = \{t_2^u, t_2^l\}$ и $\mathfrak{M}_5 = \{t\}$;

3) произвольные вершины v_p^u и v_q^l , находящиеся на различных уровнях графа G' и не лежащие в одном кластере — смежны (соединены дугами (v_p^u, v_q^l) и (v_q^l, v_p^u));

4) транспортные издержки c_e , сопоставляемые произвольной дуге $e \in E'$, задаются соотношением

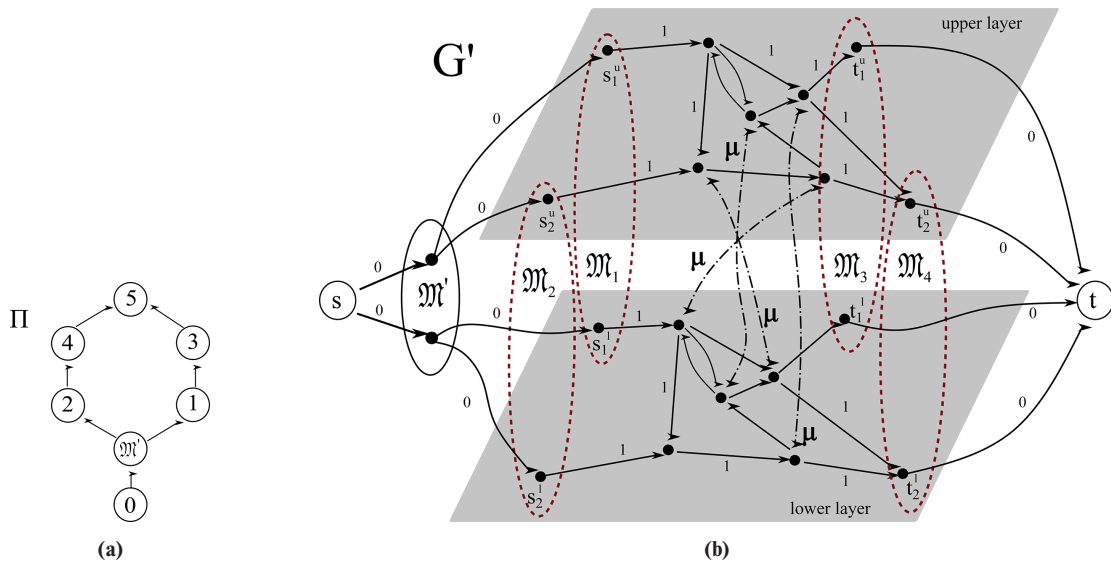
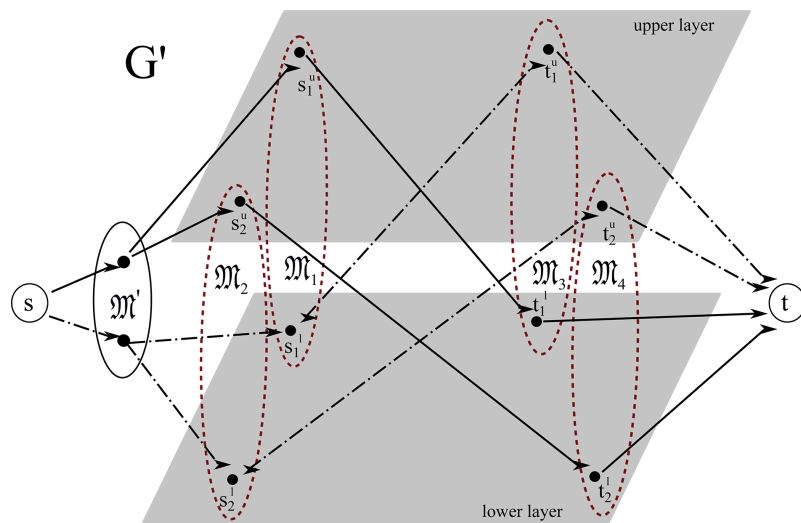
$$c_e = \begin{cases} 1, & \text{если обе вершины дуги } e \text{ лежат на одном уровне графа } G', \\ 0, & \text{если дуга } e \text{ инцидентна вершинам кластера } \mathfrak{M}' \text{ или вершине } t, \\ \mu = n^2 + 1, & \text{если } e \text{ соединяет вершины различных уровней графа } G'; \end{cases}$$

5) множество транспортных хабов задается равенством $\mathfrak{H} = (V' \setminus \mathfrak{M}') \setminus (\bigcup_{p=0}^5 \mathfrak{M}_p)$;

6) стоимость открытия C_h и пропускная способность q_h произвольного хаба $h \in \mathfrak{H}$ определяются равенствами $C_h = 0$ и $q_h = 1$, соответственно;

7) граф Π имеет вид, представленный на фиг. 3а.

По построению, для произвольной постановки I_1 сопоставленная ей постановка I_2 обладает допустимыми решениями (см. фиг. 4) и, следовательно, разрешима.

Фиг. 3. Пример постановки I_2 задачи RPPDP.Фиг. 4. Допустимые решения $\mathcal{P}_1 = \{P_1\}$ и $\mathcal{P}_2 = \{P_2\}$ произвольной постановки I_2 . Дуги плана P_1 отображены сплошными линиями, плана P_2 – пунктиром.

Для завершения доказательства теоремы убедимся, что постановка I_1 обладает положительным ответом тогда и только тогда, когда оптимальное значение соответствующей ей постановки I_2 не превосходит n^2 .

В самом деле, пусть в графе G существуют вершинно непересекающиеся s_1-t_1 и s_2-t_2 -пути. Разместив копии этих путей, например в верхнем уровне графа G' и соединив стартовую вершину s с вершинами s_1^u, s_2^u путями нулевого веса, проходящими через кластер $\mathcal{M}'$, а вершины t_1^u и t_2^u дугами нулевого веса с вершиной t , получим допустимое решение $\{P_1\}$ постановки I_2 , стоимость которого не превосходит n^2 . Следовательно, оптимальное значение задачи I_2 также удовлетворяет данной верхней оценке.

С другой стороны, пусть постановка I_2 обладает допустимым решением, стоимость которого не превосходит n^2 . С необходимостью входящий в данное решение план не содержит дуг веса $\mu = n^2 + 1$ и, следовательно, посещает вершины лишь одного из уровней графа G' . Без ограничения общности полагаем, что данный план посещает вершины верхнего уровня графа G' . По построению, он индуцирует вершинно непересекающиеся $s_1^u-t_1^u$ и $s_2^u-t_2^u$ -пути, являющиеся копиями s_1-t_1 и s_2-t_2 -путей в графе G , обуславливая положительный ответ в исходной постановке I_1 . Теорема доказана.

Отметим, что в доказательстве теоремы 1 фактически обоснована труднорешаемость задачи RPPDP при $k = 1$, нулевых стоимостях открытия и единичной пропускной способности хабов. Данные рассуждения легко могут быть распространены на случай произвольного фиксированного $k \geq 1$.

5. MILP-МОДЕЛЬ

Предлагаемая нами модель смешанного целочисленного линейного программирования оперирует двумя группами индикаторных переменных:

- $x_e^{r,a} \in \{0, 1\}$ ($r = \overline{1, k}, a \in A, e \in E$) характеризуют включение дуги e графа G в маршрут, реализующий в производственном плане r дугу a графа порядка Π ;
- $y_v^r \in \{0, 1\}$ ($r = \overline{1, k}, v \in V$) определяют использование вершины v графа G в плане r .

Кроме того, модель включает вычисляемую неотрицательную вещественную переменную cost , значение которой совпадает со значением целевой функции.

Предлагаемая MILP-модель имеет следующий вид:

$$\min \text{cost} \quad (1)$$

$$\text{s.t.} \quad \sum_{a \in A, e \in E} c_e x_e^{r,a} + \sum_{h \in \mathfrak{H}} C_h y_h^r - \text{cost} \leq 0 \quad \left(r = \overline{1, k} \right), \quad (2)$$

$$\sum_{u \in \mathfrak{M}_j \cup \mathfrak{H} : (v,u) \in E} x_{(v,u)}^{r,(i,j)} - y_v^r = 0 \quad \left(\begin{array}{l} r = \overline{1, k}, \\ (i, j) \in A, \\ v \in \mathfrak{M}_i \end{array} \right), \quad (3)$$

$$\sum_{v \in \mathfrak{M}_i \cup \mathfrak{H} : (v,u) \in E} x_{(v,u)}^{r,(i,j)} - y_u^r = 0 \quad \left(\begin{array}{l} r = \overline{1, k}, \\ (i, j) \in A, \\ u \in \mathfrak{M}_j \end{array} \right), \quad (4)$$

$$\sum_{v \in \mathfrak{M} \cup \mathfrak{H} : (v,u) \in E} x_{(v,u)}^{r,(i,j)} = 0 \quad \left(\begin{array}{l} r = \overline{1, k}, \\ (i, j) \in A, \\ u \in \mathfrak{M} \setminus \mathfrak{M}_j \end{array} \right), \quad (5)$$

$$\sum_{u \in \mathfrak{M} \cup \mathfrak{H} : (v,u) \in E} x_{(v,u)}^{r,(i,j)} = 0 \quad \left(\begin{array}{l} r = \overline{1, k}, \\ (i, j) \in A, \\ v \in \mathfrak{M} \setminus \mathfrak{M}_i \end{array} \right), \quad (6)$$

$$\sum_{v \in \mathfrak{M}_i \cup \mathfrak{H} : (v,h) \in E} x_{(v,h)}^{r,(i,j)} - \sum_{u \in \mathfrak{M}_j \cup \mathfrak{H} : (h,u) \in E} x_{(h,u)}^{r,(i,j)} = 0 \quad \left(\begin{array}{l} r = \overline{1, k}, \\ (i, j) \in A, h \in \mathfrak{H} \end{array} \right), \quad (7)$$

$$\sum_{(i,j) \in A} \sum_{v \in \mathfrak{M}_i \cup \mathfrak{H} : (v,h) \in E} x_{(v,h)}^{r,(i,j)} - q_h y_h^r \leq 0 \quad \left(r = \overline{1, k}, h \in \mathfrak{H} \right), \quad (8)$$

$$\sum_{v \in \mathfrak{M}_i} y_v^r = 1 \quad \left(r = \overline{1, k}, i = \overline{1, m} \right), \quad (9)$$

$$\sum_{r=1}^k y_v^r \leq 1 \quad (v \in V \setminus \{s, t\}), \quad (10)$$

$$x_e^{r,a} \in \{0, 1\}, y_v^r \in \{0, 1\}. \quad (11)$$

Выражения (1) и (2) задают целевую функцию, значение которой совпадает с максимальной стоимостью производственных планов, входящих в текущее решение. Уравнения (3) и (4) характеризуют начальные и конечные вершины маршрутов, реализующих дугу (i, j) графа порядка Π для каждого из k планов. Уравнения (5) и (6) гарантируют, промежуточными вершинами произвольного маршрута, реализующего дугу $(i, j) \in A$ могут быть только транспортные хабы $h \in \mathcal{H}$. Уравнение (7) является условием сохранения потока для произвольного плана r и дуги $(i, j) \in A$. Неравенство (8) задает ограничение на количество посещений хаба $h \in H$. Выражение (9) гарантирует, что каждый кластер посещается каждым планом в точности один раз. Неравенство (10) задает попарную дизъюнктивность всех производственных планов.

Проведя несложные вычисления, заметим что в предложенной модели число неизвестных составляет $O(n^2 m^2 k)$, а число ограничений — $O(nm^2 k)$. Таким образом, размер предложенной модели ограничен сверху полиномом от размера условия исходной задачи, т.е. данная модель — компактна.

Теорема 2. *Множества допустимых решений задачи RPPDP и MILP-модели (1)–(11) изоморфны.*

Доказательство. Рассмотрим произвольное допустимое решение $\mathcal{P} = \{P_1, \dots, P_k\}$ задачи RPPDP. Зададим значения индикаторных переменных $x_e^{r,a}$ и y_v^r в соответствии с входением дуг $e \in E$ и вершин $v \in V$ в каждый план r данного решения и установим значение переменной $cost$ равным стоимости $cost(\mathcal{P}) = \max\{cost(P_r) : r = 1, k\}$. Нетрудно убедиться, что построенный таким образом набор значений удовлетворяет каждому из ограничений MILP-модели (1)–(11).

С другой стороны, рассмотрим произвольное допустимое решение $(x, y, cost)$ задачи (1)–(11). Построим подграфы $P_1, \dots, P_k$, относя к подграфу P_r вершины и дуги графа G , для которых выполнено соотношение $x_e^{r,a} = 1$ (для некоторой дуги $a \in A$) и $y_v^r = 1$, соответственно. Семейство $\mathcal{P} = \{P_1, \dots, P_k\}$ является допустимым решением исходной постановки задачи RPPDP. В самом деле, справедливость ограничений (3)–(7) обеспечивает реализацию произвольной дуги $(i, j) \in A$ в каждом подграфе P_r в точности единственным маршрутом, начинающемся в некоторой вершине кластера $\mathcal{M}_i$, посещающем некоторое количество хабов и завершающемся в вершине кластера $\mathcal{M}_j$. Неравенство (8) гарантирует для произвольного хаба $h \in H$, что он используется маршрутами подграфа P_r только в случае его открытия и в пределах пропускной способности. В силу ограничения (9), произвольные планы P_{r_1} и P_{r_2} пересекаются только в вершинах s и t . Наконец, из истинности ограничения (2) следует, что стоимость плана P_r совпадает со суммарными транспортными издержками, ассоциированными с входящими в него дугами, дополненными суммарной стоимостью открытия транспортных хабов, используемых данным планом. Теорема доказана.

Приведенные ниже результаты численных экспериментов подтверждают высокое качество нижних оценок, предоставляемых вещественной релаксацией модели (1)–(11) и перспективность базирующихся на ней методов ветвления.

6. АЛГОРИТМЫ

Введенные в рассмотрение в данном разделе алгоритмы решения задачи RPPDP принадлежат семейству методов ветвей, границ и отсечений и основаны на реализации данного подхода в известном MIP-солвере Gurobi [21]. Алгоритм A_1 состоит в применении Gurobi со стандартными настройками внутренних параметров к MILP-модели (1)–(11). В отличие от него в алгоритме A_2 все используемые в солвере (встроенные) первичные эвристики заменены единственной специализированной эвристикой адаптивного поиска, описание которой приведено ниже.

6.1. Адаптивный поиск

Метаэвристический подход адаптивного поиска в больших окрестностях (Adaptive Large Neighborhood Search, ALNS), предложенный в работе [22], основан на последовательном повторении фаз разрушения / восстановления исходного допустимого решения с использованием заданного набора базовых эвристик и онлайн-обучения.

В данной работе мы развиваем модификацию ALNS, введенную в работе [23], использующую процедуры локального поиска и случайного возмущения, каждая из которых включает в себя как процесс

разрушения, так и процесс восстановления передаваемого ей решения. Договоримся называть предложенную эвристику *Reliable Production Process Design Adaptive Large Neighborhood Search (RPPD-ALNS)*. Псевдокод RPPD-ALNS представлен в алгоритме 1.

Algorithm 1. RPPD-ALNS :: общая схема

Входные данные: постановка RPPDP (G, Π, k)
Параметры: число запусков $i_{\max}$, банки процедур локального поиска $\mathfrak{Ls}$ и возмущения $\mathfrak{Sh}$
Выходные данные: результирующее допустимое решение $\overline{\mathcal{P}}$

```

1: for  $0 < i \leq i_{\max}$  do
2:   получить стартовое решение  $\mathcal{P}$  с использованием нескольких рандомизированных запусков жадного алгоритма;
3:   положить  $\mathcal{P}_{\text{best}}(i) = \mathcal{P}$ ;
4:   repeat
5:     случайным образом выбрать процедуры возмущения и локального поиска  $sh$  и  $ls$  из текущих вероятностных распределений на множествах  $\mathfrak{Sh}$  и  $\mathfrak{Ls}$ ;
6:     случайное возмущение: получить новое допустимое решение  $\mathcal{P}' = sh(\mathcal{P})$ ;
7:     локальный поиск: найти допустимое решение  $\mathcal{P}_{\text{new}} = ls(\mathcal{P}')$ 
8:     if  $\text{cost}(\mathcal{P}_{\text{new}}) < \text{cost}(\mathcal{P}')$  и не исчерпано число попыток then
9:       положить  $\mathcal{P}' = \mathcal{P}_{\text{new}}$  и вернуться на шаг 7
10:    end if
11:    if  $\text{cost}(\mathcal{P}_{\text{new}}) < \text{cost}(\mathcal{P}_{\text{best}}(i))$  then
12:       $\mathcal{P}_{\text{best}}(i) = \mathcal{P}_{\text{new}}$ 
13:    end if
14:    if критерий принятия решения выполнен для  $\mathcal{P}_{\text{new}}$  и  $\mathcal{P}$  then
15:       $\mathcal{P} = \mathcal{P}_{\text{new}}$ 
16:      запомнить улучшения, сделанные эвристиками  $sh$  и  $ls$ ;
17:    end if
18:  until не выполнится критерий остановки
19:  учитывая совершенные улучшения, обновить распределения множеств  $\mathfrak{Sh}$  и  $\mathfrak{Ls}$ 
20: end for
21: return  $\overline{\mathcal{P}} = \arg \min \{ \text{cost}(\mathcal{P}_{\text{best}}(i)) : i \in \{1, \dots, i_{\max}\} \}$ 

```

Основной цикл эвристики RPPD-ALNS состоит из $i_{\max}$ последовательных “холодных” запусков, каждый из которых начинается с нескольких попыток построения начального допустимого решения задач. Каждая такая попытка состоит в фиксации произвольного упорядочения множества дуг орграфа Π и применении жадного алгоритма для их маршрутизации в графе G .

На каждой итерации внутреннего цикла алгоритма последовательно применяются процедуры случайного возмущения sh и локального поиска ls , выбранные в соответствии с текущими вероятностными распределениями, заданными на встроенных банках эвристик $\mathfrak{Sh}$ и $\mathfrak{Ls}$. Каждая используемая нами процедура может рассматриваться как преобразование множества допустимых решений задачи, тем или иным способом трансформирующее подаваемое на вход решение. Так, цель процедуры sh состоит в возможном покидании окрестности локального оптимума, в то время как процедура локального поиска ls стремится улучшить качество найденного решения.

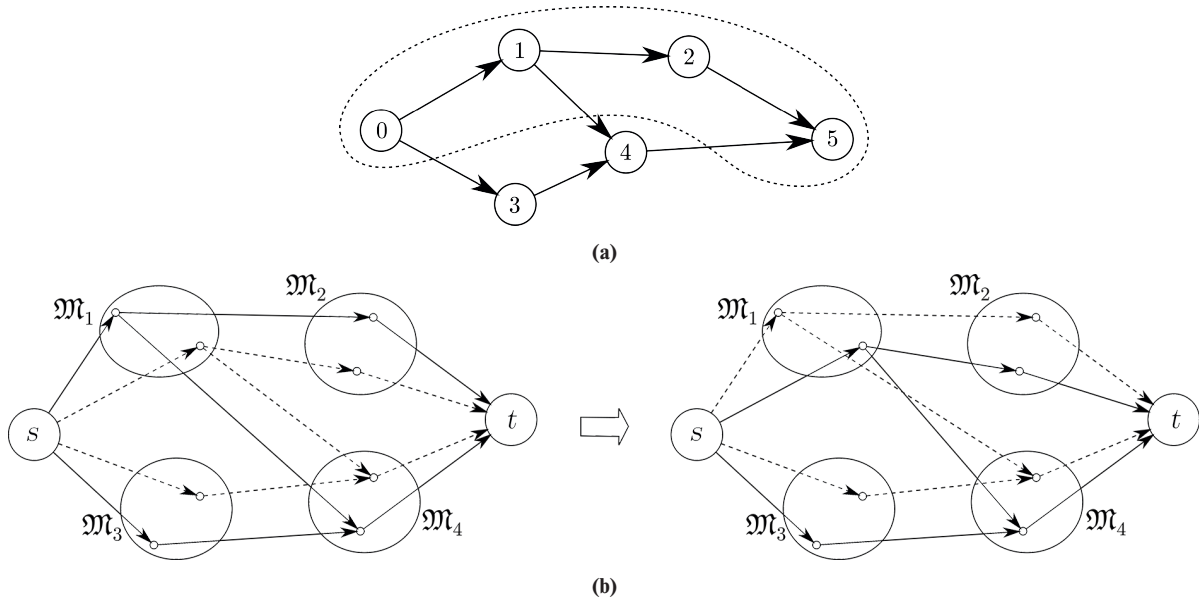
Используемое в алгоритме условие принятия найденного решения $\mathcal{P}_{\text{new}}$ традиционно опирается на метод имитации отжига (как, например, в работе [24]). При выполнении данного критерия решение $\mathcal{P}_{\text{new}}$ переводится в состояние текущего $\mathcal{P}$ и запоминаются изменения, сделанные выбранными процедурами sh и ls .

Произвольная итерация внутреннего цикла завершается проверкой критерия остановки, совпадающего в данной версии алгоритма с условием отсутствия улучшения решения на протяжении заданного числа итераций. Перед переходом на следующую итерацию обновляются вероятностные распределения на множествах $\mathfrak{Sh}$ и $\mathfrak{Ls}$ с учетом влияния конкретных эвристик на повышение качества найденных решений.

6.2. Эвристики случайного возмущения и локального поиска

Остановимся на описании множеств $\mathfrak{S}\mathfrak{h}$ и $\mathfrak{L}\mathfrak{s}$, содержащих адаптации известных алгоритмов локального поиска и случайного возмущения (см., например [25]). Каждая из представленных эвристик обрабатывает набор маршрутов, реализующих дуги графа порядка Π . В данной работе используются следующие пять процедур возмущения текущего решения, целью которых является обеспечить выход за пределы окрестности локального оптимума:

- вставка хабов случайным образом распределяет часть неиспользованных транспортных хабов между планами строящегося решения;
- обмен хабами для случайно выбранной пары планов производит обмен множествами назначенных им транспортных хабов;
- процедура переназначения кластерной вершины для случайно выбранной пары план-кластер замещает вершину кластера, включенную в заданный план, произвольной другой допустимой вершиной;
- процедура обмена кластерными вершинами для произвольно выбранной пары планов обменивает вершины, используемые данными планами в случайно выбранном кластере;
- обмен путями замещает маршрутизацию произвольного s - t -пути в графе порядка Π в случайно выбранном плане маршрутизацией этого пути в произвольном другом плане (см. фиг. 5).



Фиг. 5. Пример применения обмена s - t путями для постановки RPPDP с $m = 4$; (а) граф порядка Π , с выбранным путем $s \rightarrow M_1 \rightarrow M_2 \rightarrow t$; (б) планы в графе G до и после обмена маршрутами.

Каждая из данных процедур случайного возмущения неявным образом индуцирует специфические окрестности на множестве допустимых планов, что приводит к построению процедур локального поиска, основанных на схожих принципах. В данной версии алгоритма множество $\mathfrak{L}\mathfrak{s}$ полностью определяется множеством $\mathfrak{S}\mathfrak{h}$ и совпадает с ним по мощности.

Эвристика реализована на языке Python 3.8, с использованием библиотеки NetworkX и интегрирована в MIP-солвер Gurobi в рамках фреймворка Gurobi user callback. Исходный код размещен по адресу <https://github.com/yogorodnikov/RPPDP-instances/tree/master/RPPD-ALNS>.

7. ЧИСЛЕННЫЙ ЭКСПЕРИМЕНТ

В данном разделе описывается методика построения постановок задачи RPPDP и обсуждаются результаты оценивания производительности предложенных алгоритмов, предложенных в предыдущем разделе.

7.1. Библиотека тестовых примеров

Как и для большинства новых комбинаторных задач, тестирование алгоритмов для задачи RPPDP оказалось затруднено отсутствием общепризнанной библиотеки тестовых примеров и, как следствие, необходимостью создания такой библиотеки авторами данной статьи. Предложенная нами библиотека является адаптацией публичной библиотеки PCGTSP LIB [20], предназначенной для тестирования алгоритмов для задачи PCGTSP и восходящей к классической библиотеке TSPLIB.

В частности, каждая постановка задачи PCGTSP нами использована для построения рандомизированного семейства постановок задачи RPPDP по следующему принципу:

- граф G и функция транспортных издержек наследуются из исходной постановки PCGTSP;
- стартовой вершине постановки PCGTSP сопоставляется пара вершин s и t так, что все исходящие дуги этой вершины переназначаются вершине s , а входящие — вершине t ;
- кластеры, содержащие не менее, чем k вершин, включаются в постановку RPPDP с сохранением отношений предшествования;
- кластеры меньшего размера подлежат расформированию, входящие в них вершины объявляются транспортными хабами;
- для каждого полученного хаба h назначаются его пропускная способность q_h и стоимость его открытия C_h , равномерно распределенные в интервалах $[1, q]$ и $[1, C]$ для заданных значений параметров q и C , соответственно.

Для каждой из полученных таким образом упорядоченных пар (G, Π) формируются постановки RPPDP, (G, Π, k) с заданным числом планов k . Доступ к сформированным примерам открыт по ссылке <https://github.com/yogorodnikov/RPPDP-instances>.

7.2. Параметры эксперимента

Для проведения эксперимента были отобраны тестовые примеры, обладающие следующими характеристиками:

- число планов $k = 2, 3$;
- величина стоимости открытия хабов C ограничена сверху средним значением весов дуг графа G ;
- верхняя граница q пропускной способности хабов последовательно выбиралась из множества $\{1, 5\}$.

Для каждого сочетания графа G , числа планов k и параметра q для проведения эксперимента было сформировано по 10 случайных постановок.

Проведенный эксперимент состоял из трех последовательных стадий. Первая стадия была посвящена исследованию качества вещественной релаксации предложенной MILP-модели. Для каждой исследуемой постановки, в которой удавалось найти оптимальное значение задачи RPPDP, вычислялось относительное значение наблюдаемого разрыва целочисленности (Integrality Gap, IG) по формуле $IG = (OPT - LB)/OPT$, где OPT — оптимальное значение задачи, LB — наилучшая нижняя оценка (оптимальное значение вещественной релаксации модели). Эксперимент подтвердил высокую релаксационную способность предложенной модели для большинства постановок, что мотивировало переход к численному исследованию основанных на ней алгоритмов ветвей и границ A_1 и A_2 .

На второй стадии проведено тестирование базового алгоритма A_1 на постановках RPPDP, соответствующих значениям $q = 1$ и $q = 5$. Сравнительный анализ результатов выявил увеличение как времени выполнения первого алгоритма, так и относительной погрешности получаемых решений. Поэтому третья стадия эксперимента была посвящена численному сравнению производительности обоих алгоритмов именно на этой группе постановок.

Вычисления проводились на суперкомпьютере “Уран” Института математики и механики им. Н. Н. Красовского УрО РАН <http://parallel.uran.ru>, Intel(R) Xeon(R) 16 core CPU E5-2697 v4 @2.30 GHz 256GB RAM. Время счета для каждого экземпляра задачи ограничивалось 5 часами (18000 сек).

7.3. Результаты и обсуждение

Таблица 1 содержит данные о структуре исследуемых постановок и результаты первой фазы эксперимента: наблюдаемые значения разрыва целочисленности и времени решения релаксированных задач для постановок RPPDP. Как следует из таблицы, максимальное значение разрыва целочисленности не превысило 7.74% для постановок с $q = 1$ и 28.35% для $q = 5$ (выделено жирным шрифтом), в то время как наибольшее время счета составило 3508 и 8406 сек. Заметим, что средние значения этих величин составили 0.34% и 1.74%, соответственно. Таким образом, полученные результаты подтверждают высокую релаксационную способность предложенной MILP-модели и перспективность ее использования в методах ветвления.

Таблица 1. Исследование релаксационной способности предложенной MILP-модели. N и M — число вершин и кластеров в исходной постановке PCGTSP; m — число полученных производственных кластеров

Iname	N	M	m	$\mathcal{S}$	C	k	q = 1				q = 5			
							IG, %			Max time, sec	IG, %			Max time, sec
							Avg	Min	Max		Avg	Min	Max	
ESC0	39	8	7	4	420	2	0.56	0.56	0.56	0.01	0.56	0.56	0.56	0.0
						3	0.83	0.83	0.83	0.01	0.83	0.83	0.83	0.01
ESC12	65	13	11	5	222	2	0.54	0.54	0.54	0.02	0.54	0.54	0.54	0.02
						3	0.61	0.61	0.61	0.03	0.61	0.61	0.61	0.03
ESC25	133	26	23	5	438	2	3.24	0.1	7.74	0.02	18.14	7.27	25.31	0.06
						3	2.53	0.03	5.18	0.06	15.01	6.47	21.86	0.05
ESC47	244	48	42	11	422	2	2.03	0.12	4.33	0.11	14.46	7.16	23.04	0.1
						3	2.27	0.54	5.0	0.6	7.93	22.18	0.6	0.56
ESC63	349	64	57	10	4	2	0.12	0.0	0.21	2.2	1.66	0.46	2.82	3.72
						3	0.13	0.0	0.27	13.69	2.04	1.3	3.04	20.49
ESC78	414	79	62	28	394	2	0.01	0.0	0.05	5.15	3.12	1.51	4.43	3.02
						3	0.01	0.0	0.02	10.03	3.68	2.81	4.26	11.61
br17.10	88	17	14	6	12	2	0.59	0.0	1.52	0.04	4.46	2.06	5.84	0.03
						3	0.52	0.31	0.05	0.09	14.28	1.89	28.35	0.1
br17.12	92	17	16	4	14	2	0.63	0.21	1.05	0.03	1.62	1.14	2.48	0.03
						3	0.41	0.13	0.53	0.06	1.88	1.46	2.48	0.07
ft53.1	281	53	43	16	441	2	0.01	0.0	0.01	1.0	0.01	0.0	0.06	0.8
						3	0.01	0.0	0.01	2.73	0.02	0.0	0.06	3.0
ft53.2	274	53	41	18	420	2	0.01	0.0	0.01	1.09	0.02	0.0	0.05	1.07
						3	0.01	0.0	0.01	3.04	0.04	0.02	0.08	3.08
ft53.3	281	53	45	13	438	2	0.07	0.06	0.07	1.37	0.07	0.06	0.07	1.05
						3	0.09	0.08	0.09	3.18	0.09	0.08	0.13	3.11
ft53.4	275	53	45	14	494	2	0.03	0.03	0.03	0.8	0.04	0.03	0.09	0.63
						3	0.14	0.13	0.14	3.45	0.14	0.14	0.16	2.62
ft70.1	346	70	58	18	931	2	0.01	0.0	0.01	2.67	0.01	0.0	0.02	2.4
						3	0.01	0.0	0.01	7.99	0.01	0.0	0.02	9.4
ft70.2	351	70	58	19	878	2	0.01	0.0	0.01	3.4	0.01	0.0	0.01	4.95
						3	0.01	0.0	0.01	8.81	0.01	0.0	0.01	13.73
ft70.3	347	70	52	31	902	2	0.02	0.01	0.02	17.02	0.01	0.01	0.02	30.93
						3	0.04	0.04	0.04	21.58	0.04	0.04	0.04	14.0
ft70.4	353	70	50	32	876	2	0.17	0.17	0.17	37.33	0.17	0.17	0.17	36.31
						3	0.24	0.24	0.24	67.14	0.24	0.24	0.24	61.61
kro124p.1	514	100	78	30	1760	2	0.01	0.01	0.01	10.5	0.04	0.01	0.07	12.67
						3	0.05	0.01	0.1	12.67	0.07	0.01	0.11	11.95

Таблица 1 (окончание)

Iname	N	M	m	S	C	k	q = 1				q = 5			
							IG, %			Max time, sec	IG, %			Max time, sec
							Avg	Min	Max		Avg	Min	Max	
kro124p.2	524	100	76	40	1706	2	0.01	0.0	0.01	12.67	0.21	0.13	0.29	12.83
						3	0.01	0.0	0.01	61.53	0.2	0.13	0.27	52.71
kro124p.3	534	100	82	27	1702	2	0.04	0.03	0.05	9.58	0.11	0.05	0.14	6.82
						3	0.04	0.03	0.05	25.77	0.14	0.12	0.16	27.79
kro124p.4	526	100	80	32	1616	2	0.13	0.11	0.15	25.08	0.16	0.12	0.18	34.44
						3	0.13	0.11	0.15	157.09	—	—	—	146.28
p43.1	203	43	31	19	2188	2	0.01	0.0	0.07	1.18	0.36	0.27	0.47	1.32
						3	0.01	0.0	0.02	1.81	0.43	0.27	0.7	2.64
p43.2	198	43	34	16	2384	2	0.02	0.0	0.13	0.51	0.65	0.3	1.43	0.78
						3	0.02	0.0	0.13	1.28	0.71	0.34	1.09	1.73
p43.3	211	43	34	13	2384	2	0.03	0.0	0.08	0.34	0.16	0.08	0.28	0.42
						3	0.01	0.0	0.01	1.14	0.16	0.08	0.25	1.31
p43.4	204	43	33	15	2392	2	0.07	0.01	0.22	0.75	0.57	0.13	0.97	0.68
						3	0.17	0.03	0.53	3.9	0.68	0.17	1.01	3.57
prob.100	510	99	78	36	219	2	0.01	0.0	0.01	7.94	1.4	2.5	3.6	13.01
						3	0.01	0.0	0.01	24.67	1.92	1.19	2.86	48.52
prob.42	208	41	32	13	44	2	0.02	0.0	0.02	0.36	1.71	1.02	2.92	0.36
						3	0.03	0.03	0.03	0.63	1.63	0.86	2.55	1.16
ry48p.1	256	48	41	10	1001	2	0.01	0.0	0.01	0.14	0.01	0.0	0.01	0.09
						3	0.01	0.0	0.01	0.9	0.01	0.0	0.01	0.71
ry48p.2	250	48	39	15	982	2	0.01	0.0	0.01	0.7	0.01	0.0	0.01	0.45
						3	0.01	0.0	0.01	1.85	0.02	0.02	0.02	2.83
ry48p.3	254	48	38	14	1029	2	0.03	0.02	0.03	1.07	0.03	0.02	0.03	1.07
						3	0.02	0.02	0.02	3.26	0.02	0.02	0.02	3.25
ry48p.4	249	48	39	17	1027	2	0.16	0.15	0.16	1.64	0.16	0.15	0.16	1.07
						3	0.25	0.24	0.26	9.26	0.25	0.24	0.26	6.03
rbg048a	255	49	40	14	15	2	1.4	1.35	1.45	74.45	1.78	1.36	2.19	107.65
						3	—	—	—	405.73	—	—	—	428.34
rbg050c	259	51	43	14	17	2	—	—	—	82.48	—	—	—	80.45
						3	—	—	—	1275.09	—	—	—	248.35
rbg109a	573	110	86	36	16	2	—	—	—	2847.8	—	—	—	7621.46
						3	—	—	—	3507.6	—	—	—	8406.4

В табл. 2 приводятся результаты тестирования алгоритма A_1 : продолжительность счета и оценки относительной погрешности найденных решений $gap = (UB - LB)/LB$, где UB — вес наилучшего найденного допустимого решения, LB — значение наилучшей нижней оценки. Видно, что для 92.5% постановок при $q = 1$ найдены оптимальные решения, в то время как для оставшейся части постановок средняя величина относительной погрешности не превысила 4.6%. При $q = 5$ доля постановок задачи, для которых удалось найти оптимальное решение снизилась до 86.3%, в оставшихся случаях максимальная величина относительной погрешности составила 22.6%.

Кроме того, средний прирост времени счета составил 53.6%. В наибольшей степени это проявилось в постановках ESC63, ESC78, kro124p.X и prob.100, соответствующих одним из наиболее труднорешаемых экземпляров задачи PCGTSP [19]. Любопытно, что для некоторых немногочисленных постановок (например, rbg048a) наблюдался обратный эффект.

Таким образом, представилось разумным провести сравнение производительности алгоритма A_1 с производительностью алгоритма A_2 , опирающегося на специализированную эвристику RPPD-ALNS, причем именно на второй группе постановок с $q = 5$.

Таблица 2. Результаты тестирования алгоритма A_1

Iname	k	q = 1						q = 5					
		Time, sec			Gap, %			Time, sec			Gap, %		
		Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
ESC07	2	0.05	0.05	0.05	0.0	0.0	0.0	0.05	0.05	0.05	0.0	0.0	0.0
	3	0.07	0.06	0.1	0.0	0.0	0.0	0.1	0.08	0.13	0.0	0.0	0.0
ESC12	2	0.21	0.16	0.26	0.0	0.0	0.0	0.2	0.16	0.23	0.0	0.0	0.0
	3	0.4	0.33	0.5	0.0	0.0	0.0	0.1	0.08	0.13	0.0	0.0	0.0
ESC25	2	1.87	1.67	2.14	0.0	0.0	0.0	1.76	1.55	2.11	0.0	0.0	0.0
	3	2.72	2.55	2.94	0.0	0.0	0.0	3.56	2.53	9.0	0.0	0.0	0.0
ESC47	2	13.87	11.8	16.91	0.0	0.0	0.0	23.04	19.86	28.21	0.0	0.0	0.0
	3	22.71	18.32	35.68	0.0	0.0	0.0	38.72	33.74	47.74	0.0	0.0	0.0
ESC63	2	42.4	37.71	48.56	0.0	0.0	0.0	161.77	103.83	278.22	0.0	0.0	0.0
	3	70.74	55.53	89.36	0.0	0.0	0.0	2736.12	277.87	18000	0.05	0.0	0.47
ESC78	2	51.57	41.56	69.16	0.0	0.0	0.0	523.66	236.64	1219.75	0.0	0.0	0.0
	3	78.49	63.4	90.54	0.0	0.0	0.0	8362.78	4255.6	18000	0.2	0.0	0.6
br17.10	2	0.62	0.47	0.8	0.0	0.0	0.0	0.94	0.54	1.55	0.0	0.0	0.0
	3	1.27	0.83	2.57	0.0	0.0	0.0	2.86	0.97	6.53	0.0	0.0	0.0
br17.12	2	0.59	0.5	0.65	0.0	0.0	0.0	0.96	0.84	1.06	0.0	0.0	0.0
	3	1.06	0.79	1.31	0.0	0.0	0.0	2.7	1.62	4.54	0.0	0.0	0.0
ft53.1	2	19.51	18.53	21.47	0.0	0.0	0.0	34.4	27.5	70.9	0.0	0.0	0.0
	3	29.29	26.78	34.83	0.0	0.0	0.0	86.9	65.14	121.09	0.0	0.0	0.0
ft53.2	2	16.61	14.3	19.01	0.0	0.0	0.0	31.56	24.35	40.29	0.0	0.0	0.0
	3	23.89	20.38	28.13	0.0	0.0	0.0	105.62	69.89	153.32	0.0	0.0	0.0
ft53.3	2	17.68	16.15	22.41	0.0	0.0	0.0	30.99	27.02	37.22	0.0	0.0	0.0
	3	29.55	23.72	35.79	0.0	0.0	0.0	92.8	68.47	160.2	0.0	0.0	0.0
ft53.4	2	19.88	17.74	21.72	0.0	0.0	0.0	31.87	29.7	33.57	0.0	0.0	0.0
	3	66.67	54.5	78.75	0.0	0.0	0.0	91.17	81.15	106.04	0.0	0.0	0.0
ft70.1	2	38.17	32.3	44.43	0.0	0.0	0.0	78.38	56.58	144.66	0.0	0.0	0.0
	3	53.76	47.79	61.93	0.0	0.0	0.0	195.77	154.83	217.09	0.0	0.0	0.0
ft70.2	2	39.54	33.97	46.83	0.0	0.0	0.0	62.76	58.38	68.94	0.0	0.0	0.0
	3	53.08	48.13	60.65	0.0	0.0	0.0	167.25	139.82	202.62	0.0	0.0	0.0
ft70.3	2	76.26	66.91	90.01	0.0	0.0	0.0	244.4	133.87	440.93	0.0	0.0	0.0
	3	136.53	107.46	157.18	0.0	0.0	0.0	805.72	619.66	937.09	0.0	0.0	0.0
ft70.4	2	100.07	69.33	152.28	0.0	0.0	0.0	204.98	131.12	1473.48	0.0	0.0	0.0
	3	907.11	461.57	1563.52	0.0	0.0	0.0	4247.74	619.66	937.09	0.0	0.0	0.0
kro124p.1	2	115.6	107.5	129.8	0.0	0.0	0.0	296.72	181.35	376.27	0.0	0.0	0.0
	3	207.36	118.0	222.5	0.0	0.0	0.0	1016.04	474.89	2051.8	0.0	0.0	0.0
kro124p.2	2	147.86	116.9	232.38	0.0	0.0	0.0	597.26	425.34	1006.48	0.0	0.0	0.0
	3	271.22	244.23	363.4	0.0	0.0	0.0	4569.22	2190.44	10058.7	0.0	0.0	0.0
kro124p.3	2	171.38	118.0	205.8	0.0	0.0	0.0	421.9	333.81	509.26	0.0	0.0	0.0
	3	474.2	449.0	499.3	0.0	0.0	0.0	6502.22	3831.5	9230.8	0.0	0.0	0.0
kro124p.4	2	322.83	299.29	343.85	0.0	0.0	0.0	527.79	319.17	1164.03	0.0	0.0	0.0
	3	4006.9	1771.0	6290.2	0.0	0.0	0.0	15380.59	6526.13	18000	0.25	0.0	0.57
p43.1	2	8.47	7.6	9.94	0.0	0.0	0.0	35.12	25.47	47.81	0.0	0.0	0.0
	3	13.71	10.87	16.41	0.0	0.0	0.0	103.79	69.29	187.88	0.0	0.0	0.0
p43.2	2	7.32	6.03	8.77	0.0	0.0	0.0	17.95	12.21	22.17	0.0	0.0	0.0
	3	11.68	8.9	14.63	0.0	0.0	0.0	58.65	36.7	101.32	0.0	0.0	0.0
p43.3	2	7.06	5.9	8.34	0.0	0.0	0.0	15.9	11.13	25.75	0.0	0.0	0.0
	3	11.11	9.87	14.03	0.0	0.0	0.0	54.63	38.1	94.25	0.0	0.0	0.0
p43.4	2	6.9	5.8	8.0	0.0	0.0	0.0	22.0	14.99	39.72	0.0	0.0	0.0
	3	21.63	11.0	50.37	0.0	0.0	0.0	121.13	32.75	350.69	0.0	0.0	0.0

Таблица 2 (окончание)

Iname	k	q = 1						q = 5					
		Time, sec			Gap, %			Time, sec			Gap, %		
		Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
prob.100	2	108.8	96.9	119.86	0.0	0.0	0.0	2141.03	820.13	4525.51	0.0	0.0	0.0
	3	185.9	152.4	231.56	0.0	0.0	0.0	16060.77	6659.19	18000	0.42	0.0	0.0
prob.42	2	6.8	6.35	7.48	0.0	0.0	0.0	13.99	11.26	15.89	0.0	0.0	0.0
	3	9.67	4.6	15.8	0.0	0.0	0.0	30.93	25.93	36.6	0.0	0.0	0.0
ry48p.1	2	17.41	15.8	24.37	0.0	0.0	0.0	23.93	19.89	35.84	0.0	0.0	0.0
	3	23.8	22.52	24.37	0.0	0.0	0.0	52.04	32.75	71.43	0.0	0.0	0.0
ry48p.2	2	15.6	14.2	21.0	0.0	0.0	0.0	24.11	19.09	33.42	0.0	0.0	0.0
	3	22.02	20.8	26.7	0.0	0.0	0.0	150.4	58.32	224.49	0.0	0.0	0.0
ry48p.3	2	17.5	15.9	36.9	0.0	0.0	0.0	45.6	58.32	84.49	0.0	0.0	0.0
	3	29.42	25.78	36.88	0.0	0.0	0.0	81.4	55.36	120.94	0.0	0.0	0.0
ry48p.4	2	17.85	15.6	21.1	0.0	0.0	0.0	27.25	23.25	30.78	0.0	0.0	0.0
	3	177.7	102.8	277.8	0.0	0.0	0.0	213.09	95.17	298.57	0.0	0.0	0.0
rbg048a	2	5516.52	1524.34	10078.05	0.0	0.0	0.0	2633.12	769.66	5617.9	0.0	0.0	0.0
	3	18000	18000	18000	0.71	0.38	1.12	18000	18000	18000	0.88	0.58	1.31
rbg050c	2	18000	18000	18000	1.1	0.69	1.45	18000	18000	18000	3.95	2.63	5.44
	3	18000	18000	18000	2.35	2.02	3.17	18000	18000	18000	5.16	4.94	6.31
rbg109a	2	18000	18000	18000	3.5	2.4	4.2	18000	18000	18000	6.39	2.32	10.36
	3	18000	18000	18000	4.6	2.5	8.52	18000	18000	18000	12.55	5.85	22.6

Таблица 3 отражает результаты сравнения алгоритмов A_1 и A_2 на постановках задачи с $q = 5$. Видно, что алгоритм A_2 , использующий эвристику RPPD-ALNS, находит оптимальное решение в среднем в 90.9% случаев против 86.3% для алгоритма A_1 . В частности, оптимальные решения постановок ESC63, ESC78, и prob.100 были найдены только алгоритмом A_2 .

В 89.4% из тех случаев, где оптимальные решения были найдены обоими алгоритмами, алгоритму A_2 удавалось это сделать быстрее, в среднем на 46.8%. В частности, для постановки ft70.4 алгоритм A_2 оказался производительнее в 17 раз. В тех случаях, где оба алгоритма достигли предела времени счета, например, в задачах kro124p.4, rbg048a при $k = 3$, и rbg050c, rbg109a, алгоритму A_2 удалось найти существенно более точные приближенные решения.

Таблица 3. Результаты сравнения алгоритмов A_1 и A_2 при $q = 5$. Полужирным шрифтом выделены наилучшие полученные результаты

Iname	k	A_1						A_2					
		Time, sec			Gap, %			Time, sec			Gap, %		
		Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
ESC07	2	0.05	0.05	0.05	0.0	0.0	0.0	0.14	0.12	0.17	0.0	0.0	0.0
	3	0.1	0.08	0.13	0.0	0.0	0.0	0.23	0.21	0.25	0.0	0.0	0.0
ESC12	2	0.2	0.16	0.23	0.0	0.0	0.0	0.41	0.36	0.46	0.0	0.0	0.0
	3	0.41	0.35	0.46	0.0	0.0	0.0	0.89	0.82	0.96	0.0	0.0	0.0
ESC25	2	1.76	1.55	2.11	0.0	0.0	0.0	3.19	2.54	7.52	0.0	0.0	0.0
	3	3.56	2.53	9.0	0.0	0.0	0.0	4.47	3.81	6.51	0.0	0.0	0.0
ESC47	2	23.04	19.86	28.21	0.0	0.0	0.0	15.02	13.06	19.99	0.0	0.0	0.0
	3	38.72	33.74	47.74	0.0	0.0	0.0	30.45	21.55	54.79	0.0	0.0	0.0
ESC63	2	161.77	103.83	278.22	0.0	0.0	0.0	72.26	39.53	140.1	0.0	0.0	0.0
	3	2736.12	277.87	18000	0.05	0.0	0.47	1927.44	525.23	10472.4	0.0	0.0	0.0
ESC78	2	523.66	236.64	1219.75	0.0	0.0	0.0	338.21	166.58	745.9	0.0	0.0	0.0
	3	8362.78	4255.6	18000	0.2	0.0	0.6	6982.45	2080.9	10472.4	0.0	0.0	0.0

Таблица 3 (продолжение)

Iname	k	A ₁						A ₂					
		Time, sec			Gap, %			Time, sec			Gap, %		
		Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
br17.10	2	0.94	0.54	1.55	0.0	0.0	0.0	1.03	0.9	1.26	0.0	0.0	0.0
	3	2.86	0.97	6.53	0.0	0.0	0.0	2.59	1.79	3.84	0.0	0.0	0.0
br17.12	2	0.96	0.84	1.06	0.0	0.0	0.0	0.67	0.53	0.87	0.0	0.0	0.0
	3	2.7	1.62	4.54	0.0	0.0	0.0	1.91	1.65	2.18	0.0	0.0	0.0
ft53.1	2	34.4	27.5	40.2	0.0	0.0	0.0	18.74	17.9	19.86	0.0	0.0	0.0
	3	86.9	65.14	121.09	0.0	0.0	0.0	28.8	27.9	30.84	0.0	0.0	0.0
ft53.2	2	31.56	24.35	40.29	0.0	0.0	0.0	16.02	15.22	17.83	0.0	0.0	0.0
	3	105.62	69.89	153.32	0.0	0.0	0.0	25.96	22.97	30.35	0.0	0.0	0.0
ft53.3	2	30.99	27.02	37.22	0.0	0.0	0.0	18.4	16.97	21.51	0.0	0.0	0.0
	3	92.8	68.47	160.2	0.0	0.0	0.0	32.13	24.48	42.91	0.0	0.0	0.0
ft53.4	2	31.87	29.7	33.57	0.0	0.0	0.0	19.8	17.43	23.79	0.0	0.0	0.0
	3	91.17	81.15	106.04	0.0	0.0	0.0	67.13	55.54	103.62	0.0	0.0	0.0
ft70.1	2	78.38	56.58	144.66	0.0	0.0	0.0	36.76	32.35	40.87	0.0	0.0	0.0
	3	195.77	154.83	217.09	0.0	0.0	0.0	57.03	47.97	63.83	0.0	0.0	0.0
ft70.2	2	62.76	58.38	68.94	0.0	0.0	0.0	36.25	32.17	43.51	0.0	0.0	0.0
	3	167.25	139.82	202.62	0.0	0.0	0.0	53.69	46.78	61.91	0.0	0.0	0.0
ft70.3	2	244.4	133.87	440.93	0.0	0.0	0.0	75.15	55.1	96.61	0.0	0.0	0.0
	3	805.72	619.66	937.09	0.0	0.0	0.0	122.6	101.91	157.62	0.0	0.0	0.0
ft70.4	2	204.98	131.12	1473.48	0.0	0.0	0.0	102.9	80.68	137.17	0.0	0.0	0.0
	3	4247.74	619.66	937.09	0.0	0.0	0.0	247.82	101.91	157.62	0.0	0.0	0.0
kro124p.1	2	296.72	181.35	376.27	0.0	0.0	0.0	131.44	104.84	167.11	0.0	0.0	0.0
	3	1016.04	474.89	2051.8	0.0	0.0	0.0	209.68	179.68	279.07	0.0	0.0	0.0
kro124p.2	2	597.26	425.34	1006.48	0.0	0.0	0.0	180.78	137.2	228.41	0.0	0.0	0.0
	3	4569.22	2190.44	10058.7	0.0	0.0	0.0	335.16	252.12	219.53	0.0	0.0	0.0
kro124p.3	2	421.9	333.81	509.26	0.0	0.0	0.0	185.7	137.2	228.41	0.0	0.0	0.0
	3	6502.22	3831.5	9230.79	0.0	0.0	0.0	781.82	569.19	951.14	0.0	0.0	0.0
kro124p.4	2	527.79	319.17	1164.03	0.0	0.0	0.0	342.18	311.02	398.24	0.0	0.0	0.0
	3	15380.59	6526.13	18000	0.25	0.0	0.57	8268.88	4440.07	18000	0.05	0.0	0.28
p43.1	2	35.12	25.47	47.81	0.0	0.0	0.0	14.28	11.47	22.84	0.0	0.0	0.0
	3	103.79	69.29	187.88	0.0	0.0	0.0	53.92	35.4	73.26	0.0	0.0	0.0
p43.2	2	17.95	12.21	22.17	0.0	0.0	0.0	12.16	8.97	16.14	0.0	0.0	0.0
	3	58.65	36.7	101.32	0.0	0.0	0.0	48.87	25.03	73.67	0.0	0.0	0.0
p43.3	2	15.9	11.13	25.75	0.0	0.0	0.0	8.62	6.34	13.32	0.0	0.0	0.0
	3	54.63	38.1	94.25	0.0	0.0	0.0	21.67	13.44	31.89	0.0	0.0	0.0
p43.4	2	22.0	14.99	39.72	0.0	0.0	0.0	17.88	7.01	33.8	0.0	0.0	0.0
	3	121.13	32.75	350.69	0.0	0.0	0.0	110.87	64.17	219.0	0.0	0.0	0.0
prob.100	2	2141.03	820.13	4525.51	0.0	0.0	0.0	1801.8	564.67	2811.64	0.0	0.0	0.0
	3	16060.77	6659.19	18000	0.42	0.0	1.31	12403.5	7220.5	16305.5	0.0	0.0	0.0
prob.42	2	13.99	11.26	15.89	0.0	0.0	0.0	10.09	7.66	13.64	0.0	0.0	0.0
	3	30.93	25.93	36.6	0.0	0.0	0.0	24.9	18.28	37.88	0.0	0.0	0.0
ry48p.1	2	23.93	19.89	35.84	0.0	0.0	0.0	11.94	11.59	12.36	0.0	0.0	0.0
	3	52.04	32.75	71.43	0.0	0.0	0.0	18.02	16.09	20.3	0.0	0.0	0.0
ry48p.2	2	24.11	19.09	33.42	0.0	0.0	0.0	11.06	10.32	12.38	0.0	0.0	0.0
	3	150.4	58.32	224.49	0.0	0.0	0.0	64.82	48.33	93.95	0.0	0.0	0.0
ry48p.3	2	45.6	58.32	84.49	0.0	0.0	0.0	29.72	20.65	48.31	0.0	0.0	0.0
	3	81.4	55.36	120.94	0.0	0.0	0.0	79.2	52.18	108.58	0.0	0.0	0.0
ry48p.4	2	27.25	23.25	30.78	0.0	0.0	0.0	12.84	11.36	14.94	0.0	0.0	0.0
	3	213.09	95.17	298.57	0.0	0.0	0.0	122.59	68.73	185.67	0.0	0.0	0.0

Таблица 3 (окончание)

Iname	k	A ₁						A ₂					
		Time, sec			Gap, %			Time, sec			Gap, %		
		Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
rbg048a	2	2633.12	769.66	5617.9	0.0	0.0	0.0	2622.41	518.83	7386.01	0.0	0.0	0.0
	3	18000	18000	18000	0.88	0.58	1.31	18000	18000	18000	0.64	0.46	0.9
rbg050c	2	18000	18000	18000	3.95	2.63	5.44	18000	18000	18000	1.63	1.45	1.86
	3	18000	18000	18000	5.16	4.94	6.31	18000	18000	18000	2.6	2.04	3.48
rbg109a	2	18000	18000	18000	6.39	2.32	10.36	18000	18000	18000	2.02	1.4	3.8
	3	18000	18000	18000	12.55	5.85	22.6	18000	18000	18000	2.7	2.13	4.83

8. ЗАКЛЮЧЕНИЕ

В статье введена в рассмотрение новая комбинаторная задача проектирования отказоустойчивых производственных процессов (Reliable Production Process Design Problem, RPPDP), моделирующая эффективное построение логистических маршрутов и цепей поставок, защищенных от широкого круга возможных сбоев в транспортных сетях. Показано, что с теоретической точки зрения введенная задача близка к нескольким известным комбинаторным задачам и NP-трудна в сильном смысле.

Для эффективного поиска оптимальных и приближенных решений с оценками точности разработаны алгоритмы ветвей и границ A_1 и A_2 , основанные на предложенной компактной MILP-модели и эвристике адаптивного поиска. Для тестирования разрабатываемых алгоритмов предложена открытая библиотека постановок задачи, развивающая известную библиотеку PCGTSP LIB. Результаты численного тестирования показали, что предложенные алгоритмы успешно дополняют друг друга: с постановками малого размера успешно справляется более простой алгоритм A_1 , в то время как алгоритм A_2 , опирающийся на специализированную эвристику, демонстрирует более высокую производительность на постановках большего размера.

На данный момент открытыми остаются вопросы исследования аппроксимируемости задачи RPPDP в классе алгоритмов с априорными теоретическими оценками точности и проектирования алгоритмов ветвей и оценок (branch-and-price).

СПИСОК ЛИТЕРАТУРЫ

1. Schilling L., Seuring S. Linking the digital and sustainable transformation with supply chain practices // Int. J. Prod. Res. 2023. <https://doi.org/10.1080/00207543.2023.2173502>
2. Fan Y., Schwartz F., Vob S., Woodruff D. L. Catastrophe insurance and flexible planning for supply chain disruption management: a stochastic simulation case study // Int. J. Prod. Res. 2023. <https://doi.org/10.1080/00207543.2023.2176179>
3. Fortune S., Hopcroft J., Wyllie J. The directed subgraph homeomorphism problem // Theor. Comput. Sci. 1980. V. 10. N 2. P. 111–121.
4. Eilam-Tzoref T. The disjoint shortest paths problem // Discret. Appl. Math. 1998. V. 85. N 2. P. 113–138.
5. Ferone D., Festa P., Guerriero F., Laganà D. The constrained shortest path tour problem // Comput. Oper. Res. 2016. V. 74. P. 64–77.
6. Ferone D., Festa P., Guerriero F. An efficient exact approach for the constrained shortest path tour problem // Optim. Methods Softw. 2020. V. 35. N 1. P. 1–20.
7. Martin S., Magnouche Y., Juvigny C., Leguay J. Constrained shortest path tour problem: branch-and-price algorithm // Comp. Oper. Res. 2022. V. 144. P. 105819. <https://doi.org/10.1016/j.cor.2022.105819>
8. Saksena J. P., Kumar S. The routing problem with 'k' specified nodes // Oper. Res. 1966. V. 14. N 5. P. 909–913.
9. Kudriavtsev A., Khachay D., Ogorodnikov Y., Ren J., Shao S. C., Zhang D., Khachay M. The shortest simple path problem with a fixed number of must-pass nodes: a problem-specific branch-and-bound algorithm // LNCS. 2021. V. 12931. P. 198–210.

10. *Andrade R. C. d.* New formulations for the elementary shortest-path problem visiting a given set of nodes // *Eur. J. Oper. Res.* 2016. V. 254. N 3. P. 755–768.
11. *Gutin G., Punnen A. P.* The Traveling Salesman Problem and Its Variations. B.: Springer. 2007.
12. *Papadimitriou C.* Euclidean TSP is NP-complete // *Theor. Comput. Sci.* 1977. V. 4. P. 237–244.
13. *Khachay M., Ukolov S., Petunin A.* Problem-Specific Branch-and-Bound Algorithms for the Precedence Constrained Generalized Traveling Salesman Problem // *LNCS.* 2021. V. 13078. P. 136–148.
14. *Chentsov A. G., Khachai M. Y., Khachai D. M.* An exact algorithm with linear complexity for a problem of visiting megalopolises // *Proc. Steklov Inst. Math.* 2016. V. 295. N 1. P. 38–46.
15. *Khachai M. Y., Neznakhina E. D.* Approximation schemes for the Generalized Traveling Salesman Problem // *Proc. Steklov Inst. Math.* 2017. V. 299. Suppl. 1. P. 97–105.
16. *Khachay M., Neznakhina K.* Complexity and approximability of the Euclidean Generalized Traveling Salesman Problem in grid clusters // *Ann. Math. Artif. Intell.* 2020. V. 88. N 1. P. 53–69.
17. *Khachay M., Kudriavtsev A., Petunin A.* PCGLNS: A heuristic solver for the Precedence Constrained Generalized Traveling Salesman Problem // *LNCS.* 2020. V. 12422. P. 196–208.
18. *Morin T. L., Marsten R. E.* Branch-and-bound strategies for dynamic programming // *Oper. Res.* 1976. V. 24. N 4. P. 611–627.
19. *Khachai D., Sadykov R., Battaia O., Khachay M.* Precedence Constrained Generalized Traveling Salesman Problem: Polyhedral study, formulations, and branch-and-cut algorithm // *Eur. J. Oper. Res.* 2023. V. 309. N 2. P. 488–505.
20. *Salman R., Ekstedt F., Damaschke P.* Branch-and-bound for the Precedence Constrained Generalized Traveling Salesman Problem // *Oper. Res. Lett.* 2020. V. 48. N 2. P. 163–166.
21. Gurobi Optimization. Gurobi optimizer reference manual (2021), <https://www.gurobi.com/documentation/9.5/refman/index.html>
22. *Ropke S., Pisinger D.* An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows // *Transp. Sci.* 2006. V. 40. P. 455–472.
23. *Kalateh Ahani I., Salari M., Hosseini S. M., Iori M.* Solution of minimum spanning forest problems with reliability constraints // *Comput. Ind. Eng.* 2020. Vol. 142. P. 106365. <https://doi.org/10.1016/j.cie.2020.106365>
24. *Smith S. L., Imeson F.* GLNS: An effective large neighborhood search heuristic for the Generalized Traveling Salesman Problem // *Comp. Oper. Res.* 2017. V. 87. P. 1–19.
25. *Gendreau M., Potvin J.-Y.* Handbook of Metaheuristics. Cham: Springer. 2019.

FAULT-TOLERANT FAMILIES OF PRODUCTION PLANS: MATHEMATICAL MODEL, COMPUTATIONAL COMPLEXITY AND BRANCH AND BOUND ALGORITHMS

Yu. Yu. Ogorodnikov^{a,*}, R. A. Rudakov^{a,**}, D' M. Khachay^{b,***}, M. Yu. Khachay^{a,****}

^a*Krasovskii Institute of Mathematics and Mechanics, Ural Branch, Russian Academy of Sciences, S. Kovalevskaya st., 16,
Yekaterinburg, 620108 Russia*

^b*33405 Talence, 680 cours Libération, KEDGE Business School, France*

*e-mail: yogorodnikov@imm.uran.ru

**e-mail: r.a.rudakov@gmail.com

***e-mail: daniil.khachai@kedgebs.com

****e-mail: mkhachay@imm.uran.ru

Received 10 October, 2023

Revised 28 November, 2024

Accepted 05 March, 2024

Abstract. The design of fault-tolerant production and supply systems is one of the priority areas of development of modern operations research. The traditional approach to modeling such systems is based on the use of probabilistic models describing the choice of a possible scenario of actions in the event of failures in the production or transport network. Along with a number of advantages, this approach has a well-known drawback. The occurrence of failures of an unknown nature that can jeopardize the operability of the entire modeled system significantly complicates its application. In this paper, we introduce the minimax problem of constructing fault-tolerant production plans (Reliable Production Process Design Problem, RPPDP), the purpose of which is to ensure the smooth functioning of a distributed production system with minimal guaranteed costs. It is shown that the RPPDP problem is NP-hard in the strong sense and remains intractable under fairly specific conditions. To find exact and approximate solutions with accuracy estimates for this problem, branch and bound methods have been developed based on the proposed compact model of mixed integer linear programming (MILP) and the author's heuristics of adaptive large neighborhood search (ALNS) within the framework of extensions of the well-known Gurobi MIP-solver. High performance and complementarity of the proposed algorithms have been confirmed by the results of numerical experiments conducted on an open library of test examples developed by the authors, containing adapted problem statements from the PCGTSP LIB library.

Keywords: problem of designing fault-tolerant manufacturing processes, MILP model, branch and bound method, heuristics of adaptive search in large neighborhoods.