

Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

На правах рукописи



Уколов Станислав Сергеевич

**Разработка алгоритмов оптимальной
маршрутизации инструмента для САПР
управляющих программ машин листовой резки
с ЧПУ**

Специальность 05.13.12 —
Системы автоматизации проектирования (промышленность)

Автореферат
диссертации на соискание учёной степени
кандидата технических наук

Екатеринбург — 2022

Работа выполнена на кафедре «Информационные технологии и автоматизация проектирования» Института новых материалов и технологий ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Научный руководитель: доктор технических наук, доцент
Петунин Александр Александрович

Официальные оппоненты: **Верхотуров Михаил Александрович**,
доктор технических наук, профессор,
ФГБОУ ВО «Уфимский государственный авиационный технический университет», г. Уфа,
заведующий кафедрой информатики;

Коновалов Анатолий Владимирович,
доктор технических наук, профессор,
ФГБУН Институт машиноведения имени Э.С. Горкунова Уральского отделения Российской академии наук, г. Екатеринбург,
заведующий лабораторией механики деформаций;

Ложников Павел Сергеевич
доктор технических наук, доцент,
ФГБОУ ВО «Омский государственный технический университет», г. Омск,
заведующий кафедрой комплексной защиты информации

Защита состоится «22» февраля 2022 г. в 14:00 на заседании диссертационного совета УрФУ 05.09.24 по адресу: 620002, г. Екатеринбург, ул. Мира, д.19, ауд. И-420 (зал Ученого совета).

С диссертацией можно ознакомиться в библиотеке и на сайте ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина», <https://dissovet2.urfu.ru/mod/data/view.php?d=12&rid=3332>

Автореферат разослан « ____ » _____ 2022 г.

Ученый секретарь
диссертационного совета



Огородникова Ольга Михайловна

Общая характеристика работы

Актуальность темы исследования. Современное производство предъявляет высокие требования к качеству заготовок и технико-экономическому уровню выпускаемой продукции, что приводит к увеличению затрат на проектирование и технологическую подготовку производства. Раскройно-заготовительные операции, являясь началом большинства производственных процессов, оказывают существенное влияние на трудоемкость и экономичность изготовления деталей. Для получения заготовок сложной геометрической формы из листового материала в условиях мелкосерийного и единичного производства широко применяются машины фигурной резки с числовым программным управлением (ЧПУ). Использование оборудования с ЧПУ, предполагает применение средств автоматизации проектирования управляющих программ (САМ-систем). При использовании современных САД/САМ систем, предназначенных для автоматизированного проектирования раскроя и подготовки управляющих программ (далее — УП) для оборудования с ЧПУ, возникает несколько различных взаимосвязанных задач, поэтому обычно проектирование УП для технологического оборудования листовой резки состоит из нескольких этапов. Первый этап предполагает предварительное геометрическое моделирование заготовок и разработку раскройной карты, здесь возникает известная задача оптимизации фигурного раскроя листового материала, которая относится к классу трудно решаемых проблем раскроя-упаковки (*Cutting & Packing*). На следующем этапе проектирования УП осуществляется процесс назначения маршрута резки — траектории перемещения режущего инструмента для полученного на первом этапе варианта раскроя, здесь возникают актуальные научно-практические задачи оптимизации маршрута режущего инструмента. Их целью обычно является минимизация стоимости и / или времени процесса резки, связанного с обработкой требуемых контуров деталей из листового материала, за счет определения оптимальной последовательности вырезки контуров и выбора необходимых точек для врезки в материал листа, а также направления движения резака с учетом технологических ограничений процесса резки. Следует отметить, что современные специализированные САПР предоставляют базовый инструментальный для решения задач рационального раскроя материалов и подготовки УП для технологического оборудования листовой резки с ЧПУ. Вместе с тем разработчики систем автоматизированного проектирования УП для оборудования листовой резки с ЧПУ не уделяют должного внимания проблеме оптимизации маршрута резки. Существующее программное обеспечение САПР не гарантирует получение оптимальных траекторий перемещения инструмента при одновременном соблюдении технологических требований резки. Зачастую пользователи САПР используют интерактивный режим проектирования УП. Кроме того, отсутствуют способы оценки точности полученных решений. В связи с этим **актуальным** направлением исследования являются вопросы разработки и применения эвристических и метаэвристических подходов, а также точных алгоритмов, которые позволяют получить решение задачи оптимальной маршрутизации режущего инструмента в режиме автоматического проектирования за приемлемое

время и обеспечивают при этом эффективные оценки результатов проектирования.

Степень разработанности темы исследования. Методы проектирования технологических процессов раскройно-заготовительного производства исследовались в работах как отечественных так и зарубежных ученых.

Разработкой алгоритмов для маршрутизации инструмента машин листовой резки с ЧПУ занимались, в частности, следующие российские исследователи: М.А. Верхотуров, Т.А. Макаровских, Р.Т. Мурзакаев, А.А. Петунин, А.Г. Ченцов, П.А. Ченцов, В.Д. Фроловский, М.Ю. Хачай, А.Ф. Таваева и др., а также зарубежные исследователи: E. Arkin, N. Ascheuer, D. Cattrysse, R. Dewil, L. Gambardella, J. Hoefl, Y. Jing, Y. Kim, M. Lee, S.U. Sherif, W. Yang и др. В подавляющем большинстве работ используется дискретизация граничных контуров деталей, что позволяет применять хорошо разработанные математические модели дискретной оптимизации. Можно отметить только отдельные публикации, где оптимизационные алгоритмы ориентированы на поиск решений в непрерывных множествах.

Задачи маршрутизации инструмента машин листовой резки с ЧПУ относятся, как известно, к NP-трудным задачам. Следует отметить, что до настоящего времени не существует единой математической модели проблемы оптимизации траектории инструмента для технологического оборудования листовой резки с ЧПУ. Имеются отдельные группы ученых, которые занимаются исследованием частных случаев этой проблемы. Кроме того, в рамках CAD/CAM систем, предназначенных для проектирования УП для машин листовой резки с ЧПУ, есть отдельные модули, которые позволяют решать в автоматическом режиме *некоторые* оптимизационные задачи, например минимизацию холостого хода инструмента, однако при этом не обеспечивают соблюдение технологических требований резки материала на машинах с ЧПУ и не позволяют получать маршруты резки, близкие к оптимальным с точки зрения критериев стоимости и времени резки с учетом рабочего хода инструмента, затрат на врезку и т.д. К тому же, реализованные в коммерческом программном обеспечении алгоритмы как правило не описываются в научной литературе.

В общей проблеме маршрутизации инструмента машин листовой резки с ЧПУ можно выделить несколько классов задач: задача непрерывной резки (ССР), задача резки с конечными точками (ЕСР), задача прерывистой резки (ICP), задача обхода многоугольников (TPP), задача коммивояжера (TSP) и обобщенная задача коммивояжера (GTSP). Любая задача оптимизации термической резки может рассматриваться как ICP, тем не менее, литература по ICP очень скудна, и большинство программных и научных статей вводят искусственные ограничения, которые упрощают ICP до задач других классов. Поиск хороших алгоритмов оптимизации с эффективными оценками точности для нескольких подклассов задачи ICP мог бы заполнить явный существующий пробел в исследованиях. В частности, отметим, что актуальна разработка алгоритмов для подкласса Segment ССР, базирующегося на понятии «сегмента резки».

Применение эффективных классических метаэвристических алгоритмов дискретной оптимизации (метод ветвей и границ, метод эмуляции отжига, метод муравьиной колонии, эволюционные алгоритмы, метод переменных окрестностей и др.) для дискретных моделей оптимизации траектории инструмента машин с ЧПУ возможно только при адаптации этих алгоритмов к требованиям технологических ограничений листовой резки. Таким образом, необходимость в создании специализированных оптимизационных алгоритмов и программного обеспечения для САПР управляющих программ машин листовой резки с ЧПУ остается доминантой развития методов решения исследуемой оптимизационной проблемы маршрутизации инструмента.

Цель работы заключается в разработке алгоритмов решения задачи оптимальной маршрутизации режущего инструмента и методик применения данных алгоритмов в системах автоматизированного проектирования УП для машин термической резки с ЧПУ. Для достижения поставленной в работе цели необходимо решить следующие **задачи**:

- Разработать точный алгоритм решения обобщённой задачи коммивояжера с ограничениями предшествования (PCGTSP), позволяющий оценивать качество решений на основе вычисления нижней оценки
- Разработать эвристики поиска оптимального положения точек врезки в контур детали и последовательности обхода контуров в процессе решения задач непрерывной резки (ССР, SCCR)
- Разработать программное обеспечение, реализующее разработанные алгоритмы
- Разработать схемы информационного обмена и методику использования алгоритмов оптимальной маршрутизации режущего инструмента в САД/САМ-системах при автоматическом проектировании управляющих программ машин листовой резки с ЧПУ.

Научная новизна результатов:

1. Разработан алгоритм ветвей и границ для обобщенной задачи коммивояжера с ограничениями предшествования PCGTSP, позволяющий строить нижние оценки для решений указанной задачи, в том числе, полученных другими алгоритмами и эвристиками. Этот алгоритм способен находить точные решения для задач значительно большей размерности, чем известные алгоритмы (до ≈ 150 кластеров в зависимости от уровня вложенности).
2. Разработаны алгоритм поиска точек врезки в контуры, не использующий механизм дискретизации, а также схема выбора последовательности резки контуров на основе метода переменных окрестностей, совместно решающие задачи ССР и SCCR.

3. Сформулированы схемы использования ограничений предшествования для уменьшения вычислительной сложности алгоритмов оптимальной маршрутизации, как в моделях дискретной, так и непрерывной оптимизации.

Теоретическая и практическая значимость работы:

1. Разработанные алгоритмы могут применяться для автоматического проектирования УП машин листовой резки с ЧПУ. Для ряда задач впервые удалось получить эффективные оценки точности решений.
2. Использование непрерывных моделей оптимизации позволяет уменьшить длину холостого хода (в некоторых случаях — до 10%) по сравнению с используемыми в настоящее время дискретными моделями.
3. Разработанные алгоритмы могут применяться для решения более общей задачи маршрутизации резки, например обобщённой сегментной резки GSCCP.
4. Разработанные схемы информационного обмена, форматы файлов и методика использования алгоритмов оптимальной маршрутизации инструмента позволяют интегрировать разработанное программное обеспечение в существующие российские САПР «Сириус» и САПР «Т-Flex»

Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ (государственный контракт № 075-03-2020-582/4).

Результаты исследований используются в учебном процессе ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина».

Методология и методы исследования. Методологическую базу исследования составили фундаментальные и прикладные работы отечественных и зарубежных ученых в области автоматизированного проектирования маршрута резки для машин листовой резки с ЧПУ, геометрического моделирования, разработки алгоритмов оптимальной маршрутизации, методы вычислительной геометрии и компьютерной графики. В качестве инструментов исследования использовались следующие методы: анализ, синтез, классификация, формализация, математические методы обработки данных. Оценка эффективности предложенных методов и алгоритмов осуществлялась с помощью вычислительных экспериментов на различных раскройных картах и тестовых примерах. Проводилось их сравнение с результатами, полученными при работе алгоритмов других авторов.

Положения, выносимые на защиту:

1. Точный алгоритм решения обобщённой задачи коммивояжера с ограничениями предшествования (PCGTSP) с обновлением нижней границы.

2. Эвристика поиска точек врезки в плоские контура, не использующая дискретизацию контура.
3. Достаточные условия, при которых полученный маршрут доставляет глобальный минимум длины холостого хода инструмента.
4. Форматы файлов и схемы для обмена геометрической и маршрутной информацией и визуализации для использования алгоритмов оптимальной маршрутизации в CAD/CAM-системах.

Достоверность результатов диссертационной работы подтверждается результатами экспериментальных исследований, приведенными в ряде публикаций и полученными при использовании методик, алгоритмов и программных средств, созданных при непосредственном участии соискателя. Основные положения диссертации были представлены на международных и всероссийских научных конференциях, опубликованы в изданиях ВАК, Scopus, WoS, известны в научном сообществе и положительно оценены специалистами.

Апробация результатов работы. Основные результаты работы докладывались и обсуждались на всероссийских и международных конференциях, в том числе:

- *Applications of Mathematics in Engineering and Economics* (AMEE'16), Созополь, Болгария, 08.06.2016 – 13.06.2016.
- *Manufacturing, Modelling, Management & Control*, (8th MiM 2016) Труа, Франция, 28.06.2016 – 30.06.2016.
- *ASRTU 2017 International Conference on Intellectual Manufacturing*, Харбин, Китайская Народная Республика, 15.06.2017 – 18.06.2017.
- *Mathematical Optimization Theory And Operations Research* (MOTOR 2019), Екатеринбург, Россия, 08.07.2019 – 12.07.2019.
- *Manufacturing Modelling, Management and Control*, (9th MiM 2019) Берлин, Германия, 28.08.2019 – 30.08.2019.
- *X Всероссийская конференция «Актуальные проблемы прикладной математики и механики»* с международным участием, посвященная памяти академика А.Ф.Сидорова и 100-летию Уральского федерального университета, пос. Абрау-Дюрсо, Россия, 01.09.2020 – 06.09.2020.
- *ICPR International Workshops and Challenges Virtual Event*, Milan, Italy, 10.01.2021 – 15.01.2021.
- *XVI Всероссийская научно-практическая конференция «Перспективные системы и задачи управления»*, Домбай, Россия, 5.04.2021 – 9.04.2021.

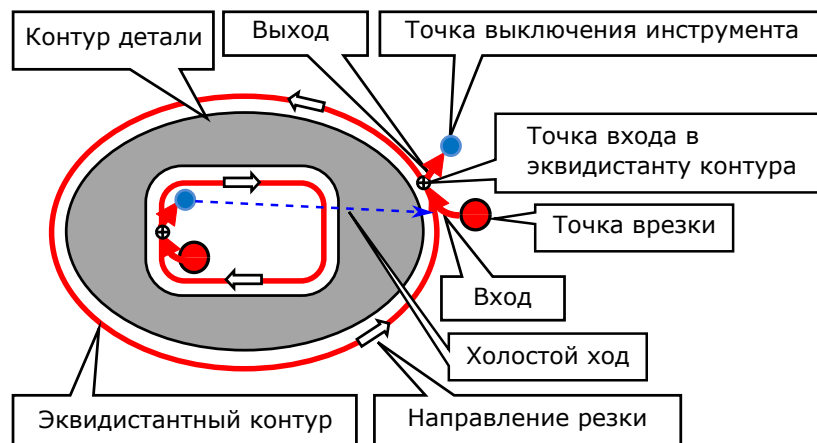


Рис. 1 – Элементы маршрута резки

- *XII International Conference Optimization and Applications (OPTIMA-2021)*, Petrovac, Черногория, 27.09.2021 – 1.10.2021.
- *XIV-я Всероссийская Мультиконференция по проблемам управления*, с. Дивноморское, Геленджик, Россия, 27.09.2021 – 02.10.2021.

Личный вклад автора состоит в проведении теоретических и экспериментальных исследований по теме диссертационной работы, проведении аналитических расчетов на основе полученных результатов. В опубликованных совместных работах постановка и разработка алгоритмов для решения задач осуществлялись совместными усилиями соавторов при непосредственном активном участии соискателя.

По теме диссертационной работы произведено 18 публикаций, в том числе 9 научных работ в рецензируемых научных журналах, определенных ВАК РФ и Аттестационным советом УрФУ, из них 8 публикаций проиндексировано в международных базах данных WoS и Scopus.

Объем и структура работы. Диссертация состоит из введения, 4 глав, заключения и 4 приложений. Полный объем диссертации составляет 135 страниц, включая 24 рисунка и 7 таблиц. Список литературы содержит 121 наименование.

Основное содержание работы

Во **введении** обосновывается актуальность диссертационной работы, определены цель и задачи исследования, представлена научная новизна, теоретическая и практическая значимость работы. Приведены основания для выполнения работы, ее апробация и структура.

В **первой главе** проводится анализ проблемы автоматизированного проектирования УП в раскройно-заготовительном производстве для оборудования термической фигурной резки с ЧПУ.

В общем случае маршрут инструмента содержит несколько компонент: начальную M_0 и возможно отличную от неё конечную точку маршрута M_{N+1} , N точек врезки M_i , $i \in \overline{1, N}$ и соответствующих им точек выключения инструмента M_i^* , рабочий ход (*сегмент резки* $S_i = M_i M_i^*$) инструмента от точки врезки M_i до точки выключения M_i^* , а также холостой ход от M_i^* до следующей точки врезки M_{i+1} , см. рис. 1. В определение маршрута также входит порядок резки, то есть последовательность посещения точек врезки $I = (i_1, i_2, \dots, i_N)$. Таким образом, маршрут резки можно определить в терминах сегментов резки как кортеж

$$\mathfrak{R} = \langle M_0, M_1, S_1, M_1^*, M_2, S_2, M_2^*, \dots, M_N, S_N, M_N^*, i_1, i_2, \dots, i_N \rangle \quad (1)$$

В качестве целевой функции при оптимизации часто используется время резки

$$T_{cut} = \frac{L_{on}}{V_{on}} + \frac{L_{off}}{V_{off}} + N_{pt} \cdot t_{pt}, \quad (2)$$

где L_{on} – суммарная длина реза; V_{on} – скорость рабочего хода режущего инструмента; L_{off} – суммарная длина холостого хода; V_{off} – его скорость; N_{pt} – количество точек врезки; t_{pt} – время, затрачиваемое на одну врезку. В подавляющем большинстве исследований, включая данную диссертационную работу, V_{on} считается константой [8].

Важнейшей экономической характеристикой качества разработанной управляющей программы является стоимость резки. По аналогии с (2) его можно определить по формуле

$$F_{cost} = L_{on} \cdot C_{on} + L_{off} \cdot C_{off} + N_{pt} \cdot C_{pt}, \quad (3)$$

где C_{on} – стоимость единицы пути с включенным режущим инструментом; C_{off} – стоимость единицы пути холостого хода; C_{pt} – стоимость одной врезки.

Задача оптимизации маршрута инструмента для машин фигурной листовой резки с ЧПУ может быть представлена в общем виде как задача минимизации некоторой числовой функции \mathfrak{F} (например, (2) или (3)) на множестве \mathfrak{E} допустимых кортежей (1):

$$\mathfrak{F}(\mathfrak{R}) \rightarrow \min_{\mathfrak{R} \in \mathfrak{E}} \quad (4)$$

В маршрут (1) помимо перестановки $I = (i_1, i_2, \dots, i_N)$ входят также точки M_i и M_i^* , которые в общем случае должны ещё быть выбраны из некоторого непрерывного множества, что делает простую формулировку задачи (4) чрезвычайно сложной в решении. Более того, набор сегментов резки S_i в общем случае не совпадает (даже в количестве) с набором контуров деталей, подлежащих резке, так как на практике используются различные техники резки: *резка по замкнутому контуру (стандартная техника)* – сегмент резки содержит ровно один замкнутый контур заготовки и вырезается целиком; *мультисегментная резка* – для вырезки одного контура используются не менее двух сегментов резки; *мультиконтурная резка* – резка предполагает вырезку нескольких контуров в одном сегменте.

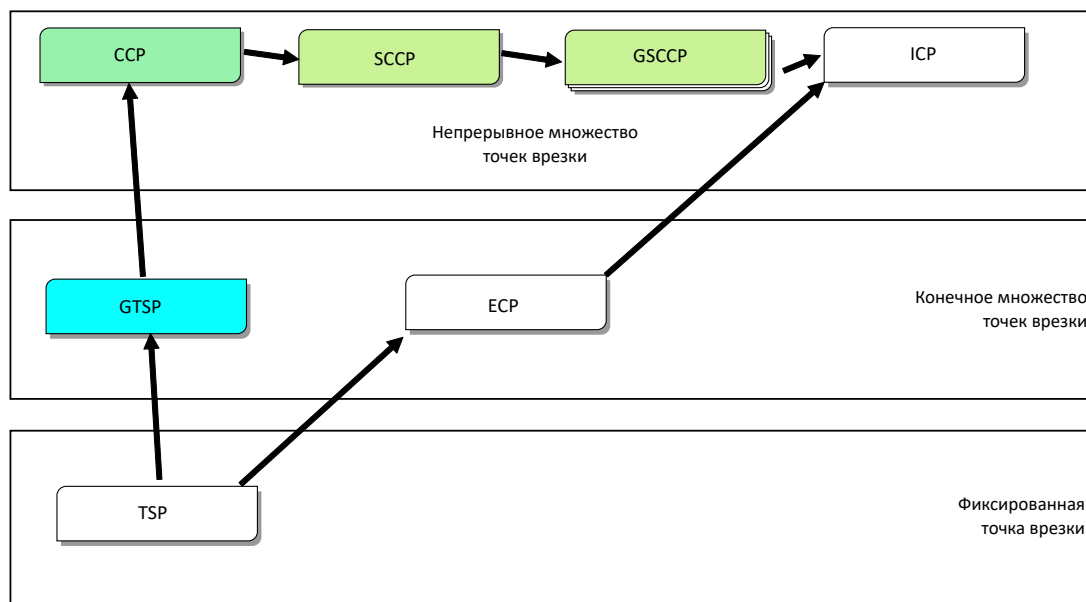


Рис. 2 – Классификация задач резки

В зависимости от используемой техники резки и способа выбора точек врезки можно выделить несколько классов задач резки¹, изображенных на рис. 2:

- **Задача непрерывной резки** (Continuous Cutting Problem, CCP): каждый контур вырезается за один раз, одним движением инструмента, но резка может начаться в любой точке контура (и заканчивается в ней же)
- **Обобщённая задача коммивояжера** (Generalized Traveling Salesman Problem, GTSP): резка может начаться в одной из заранее заданных точек на контуре (количество таких точек конечно), после этого контур вырезается целиком
- **Задача резки с конечным набором точек врезки** (Endpoint Cutting Problem, ECP): резка контура может начинаться только в заранее заданных точках на нём, но контур может вырезаться за несколько раз, частями
- **Сегментная задача непрерывной резки** (Segment Continuous Cutting Problem, SCCP): сегмент резки может быть частью контура или объединением нескольких контуров и / или их частей. Каждый сегмент вырезается целиком, от начала до конца, то есть $CCP \subset SCCP$.
- **Обобщённая сегментная задача непрерывной резки** (Generalized Segment Continuous Cutting Problem, GSCCP): подобна сегментной задаче непрерывной резки (SCCP), но разбивка на сегменты не задана заранее и сама подлежит оптимизации
- **Задача прерывистой резки** (Intermittent Cutting Problem, ICP): наиболее общая формулировка задачи резки, встречающаяся в научной литературе,

¹Dewil R. A review of cutting path algorithms for laser cutters / R. Dewil, P. Vansteenwegen, D. Cattrysse // International Journal of Advanced Manufacturing Technology. 2016. T. 87, № 5. С. 1865–1884.

контуры могут вырезаться частями, в несколько подходов, начиная с произвольной точки.

Полученный любым способом маршрут движения режущего инструмента, должен быть исполнен на конкретном промышленном оборудовании – режущей машине с ЧПУ. Это накладывает ряд существенных ограничений на решение задачи резки, то есть ограничивает множество \mathfrak{B} в (4).

Наиболее популярным и хорошо описанным в литературе является так называемое *ограничение предшествования* (Precedence Constraint, PC), возникающее из-за того, что после вырезания замкнутого контура, его внутренняя часть ничем не удерживается и может сдвигаться, поворачиваться, наклоняться или даже падать. Поэтому внутренние отверстия деталей следует вырезать до того, как будет завершена резка внешнего контура детали. Аналогично, если меньшая деталь размещается в отверстии большей детали, она должна целиком быть вырезана до того, как будет завершена резка содержащего её отверстия.

В литературе описан ещё целый ряд ограничений, накладываемых на маршрут режущего инструмента, порождаемых технологическими свойствами современных машин термической резки с ЧПУ, которые также должны учитываться при практическом применении, см. [6; 9]. В данной диссертационной работе эти технологические ограничения не рассматриваются.

Во **второй главе** рассматривается применение дискретных моделей оптимизации для проектирования УП для машин фигурной резки с ЧПУ, при этом исследуется обобщённая задача коммивояжера с ограничениями предшествования (*PCGTSP*) – хорошо известная задача комбинаторной оптимизации, имеющая множество приложений помимо оптимизации траектории режущего инструмента. К сожалению, в отличие от задачи GTSP, алгоритмические подходы к решению именно PCGTSP всё ещё малочисленны. Алгоритм [1], предложенный в данной диссертационной работе, разработан на основе синтеза нескольких идей других исследователей.

Задача определяется тройкой (G, \mathcal{C}, Π) , где $G = (V, E, c)$ – взвешенный ориентированный граф на n вершинах, задающий веса $c(u, v)$ для всех своих ребер $(u, v) \in E, u, v \in V$; $\mathcal{C} = \{V_1, \dots, V_m\}$ – разбиение вершин графа G на m кластеров; $\Pi = (\mathcal{C}, A)$ – частичный порядок, заданный на множестве кластеров. Для каждой вершины $v \in V$, за $V(v)$ обозначим (единственный) кластер $V_p \in \mathcal{C}$, такой что $v \in V_p$.

Допустимым решением задачи (G, \mathcal{C}, Π) называется тур (замкнутый путь) $T = v_1, v_2, \dots, v_m$, удовлетворяющий нескольким условиям: имеет длину $|T| = m$; начинается и заканчивается в некоторой вершине $v_1 \in V_1$; посещает каждый кластер $V_p \in \mathcal{C}$ в произвольной вершине $v_p \in V_p$; *соответствует* частичному порядку Π , то есть любой кластер V_p посещается маршрутом T только *после* всех кластеров, предшествующих ему в Π .

Каждому решению T назначается стоимость

$$\text{cost}(T) = c(v_m, v_1) + \sum_{i=1}^{m-1} c(v_i, v_{i+1}) \quad (5)$$

Требуется найти допустимый тур T минимальной стоимости $cost(T) \rightarrow \min$.

Ключевой идеей алгоритма является построение нижней оценки стоимости решения. Для этого в каждой вершине дерева поиска исходная задача разделяется на две:

1. Фиксируется *префикс* маршрута $\sigma = \{V_1, \dots, V_r\}$. Обозначим c_{\min} нижнюю границу длины всех путей, проходящих из вершин кластера V_1 в вершины кластера V_r строго в порядке σ .
2. Строится вспомогательная задача \mathcal{P} , удалением из исходной задачи всех кластеров, являющиеся внутренними в σ и соединением всех вершины из V_1 с вершинами V_r ребрами нулевого веса. Полученная таким образом задача всё ещё сложна, однако она может быть несколькими способами упрощена (релаксирована) $\mathcal{P} \rightarrow \mathcal{P}_{rel}$ и для неё найдено оптимальное решение $OPT(\mathcal{P}_{rel})$

Нижняя оценка LB находится как

$$LB = c_{\min} + OPT(\mathcal{P}_{rel}) \quad (6)$$

В качестве релаксации \mathcal{P}_{rel} задачи \mathcal{P} могут применяться:

- хорошо известная трансформация Нуна и Бина, преобразующая задачу GTSP в обычную задачу коммивояжера TSP
- построение вспомогательного *графа кластеров* H_1 с весами, индуцированными весами исходного графа G :

$$c(V_i, V_j) = \min_{v_i \in V_i, v_j \in V_j} c(v_i, v_j)$$

- построение графа кластеров H_2 , веса в котором также индуцированы весами путей длины 2 в исходном графе G :

$$c(V_i, V_j) = \min_{\substack{v_i \in V_i, v_j \in V_j \\ v_k \notin V_i \cup V_j}} \frac{c(v_i, v_k) + c(v_k, v_j)}{2}$$

при этом маршрут v_i, v_k, v_j должен удовлетворять ограничению предшествования П. Этот способ позволяет точнее учитывать взаимное расположение контуров на плоскости в случае Евклидовой задачи PCGTSP

- по аналогии с H_2 могут строиться кластерные графы с весами на основе путей большей (≥ 3) длины, однако алгоритм их расчёта существенно сложнее и не входит в рамки данной диссертационной работы

Наконец, для (быстрого) поиска нижней оценки на решение задачи $TSP(\mathcal{P}_{rel})$ можно использовать: решение задачи о минимальном остовном дереве (Minimal Spanning Arborescence, MSAP), так как $MSAP(\mathcal{P}) \leq TSP(\mathcal{P})$;

Таблица 1 – Методы оценки нижней границы

	Нун и Бин	H_1	H_2	H_{3+}
AP	E_1	L_1	L_2	-
MSAP	E_2	E_3	E_4	-
Gurobi	E_5	L_3	E_6	-

решение задачи о цикловом покрытии (Cycle cover) — оно находится как решение задачи о назначениях (Assignment Problem, AP) для двудольного графа, обе доли которого представляют \mathcal{P}_{rel} ; опять $AP(\mathcal{P}) \leq TSP(\mathcal{P})$; в некоторых случаях можно прямо решить задачу коммивояжера $TSP(\mathcal{P})$, в данной работе для этого использовался решатель Gurobi и MIP-модель ATSPxy².

Все возможные методы оценки нижней границы сведены в табл. 1, используемые в описываемой реализации обозначены L_1 – L_3 , не используемые – E_1 – E_6 . Проведённые численные эксперименты показывают, что часть методов (E_1 – E_4) систематически дают более слабые оценки, а часть (E_1 – E_2 , E_5 – E_6) требуют значительных временных затрат по сравнению с выбранными.

Получив описанными методами оценку нижней границы для данного префикса по формуле (6), алгоритм принимает решение об отсечении ветви, порождаемой текущим префиксом σ , при условии

$$LB > UB, \quad (7)$$

где UB – стоимость наилучшего известного допустимого решения исходной задачи. В данной работе для получения UB используется эвристика PCGLNS³, позволяющая буквально за несколько секунд получить близкое к оптимальному решение исходной задачи PCGTSP, что резко сокращает перебор, позволяя отбрасывать $\approx 50\%$ – 90% ветвей.

Предлагаемый алгоритм ветвей и границ обходит дерево поиска, начиная с корня $\sigma = \{V_1\}$ в ширину, для каждого узла находя оценку нижней границы (6), проверяя условие отсечения (7) и для «выживших» узлов применяя процедуру ветвления. Для этого он находит все кластера, которые можно добавить в конец текущему префиксу, не нарушая ограничение предшествования Π . По окончании подсчёта префиксов одной длины, алгоритм обновляет текущую оценку нижней границы $LB = \max(LB, LB_r)$, где

$$LB_r = \min_{|\sigma|=r} LB(\sigma),$$

в этот момент алгоритм может быть остановлен по достижении нужной точности. Если же алгоритм доходит до обработки префиксов длины $|\sigma| = m$, то для них

²Sarin S. C. New Tighter Polynomial Length Formulations for the Asymmetric Traveling Salesman Problem with and without Precedence Constraints / S. C. Sarin, H. D. Sherali, A. Bhootra // Oper. Res. Lett. NLD, 2005. Т. 33, № 1. С. 62–70.

³Khachay M. PCGLNS: A Heuristic Solver for the Precedence Constrained Generalized Traveling Salesman Problem / M. Khachay, A. Kudriavtsev, A. Petunin // Optimization and Applications. Т. 12422 / под ред. N. Olenev, Y. Evtushenko, M. Khachay, V. Malkova. Cham : Springer International Publishing, 2020. С. 196–208. (Lecture Notes in Computer Science).

вместо оценки (6) по известному порядку обхода кластеров σ находится решение исходной задачи путём поиска кратчайшего тура, посещающего кластера в этом порядке, и из всех таких решений выбирается оптимальное.

Такой алгоритм оказывается вполне работоспособен, однако в ходе его тестирования были выявлены некоторые недостатки: в некоторых задачах использованные методы оценки нижней границы (см. табл. 1) дают очень низкие значения; сведение всех путей вдоль префикса σ к минимальному c_{min} представляется слишком грубым; все префиксы σ , оканчивающиеся на один и тот же кластер V_r , но разный порядок «внутренних» кластеров, оказываются тесно связаны; это делает сложным попытки распараллелить выполнение алгоритма.

Поэтому была разработана вторая версия того же алгоритма, устроенная по классической схеме динамического программирования (DP) Хелда-Карпа⁴, модифицированной для задачи PCGTSP и дополненной стратегией отсечения (7).

Каждое состояние DP (запись в таблице поиска) соответствует частичному v - u -пути и индексируется кортежем $(V_1, \mathcal{C}', V_r, v, u)$, где V_1 и V_r – начальный и конечный кластеры маршрута, v и u – начальная и конечная его вершины ($v \in V_1$, $u \in V_r$), \mathcal{C}' – *порядковый идеал* частично упорядоченного множества \mathcal{C} и играет здесь роль, аналогичную префиксу σ в первой версии алгоритма. По определению идеала $\forall V \in \mathcal{C}', V' \in \mathcal{C} ((V', V) \in A) \Rightarrow (V' \in \mathcal{C}')$, поэтому V_1 принадлежит произвольному идеалу $\mathcal{C}' \subset \mathcal{C}$. Пусть \mathfrak{I}_k – подмножество идеалов одного размера $k \in \{1, \dots, m\}$. Очевидно, $\mathfrak{I}_1 = \{\{V_1\}\}$, остальные слои строятся индуктивно при помощи Алгоритма 1.

Для оценки производительности предложенных алгоритмов использовалась общедоступная библиотека PCGTSPLIB⁵, в качестве базы сравнения использовалось решение, полученное решателем Gurobi. Всем алгоритмам задавалось одно и то же допустимое решение, полученное эвристикой PCGLNS. В качестве критерия остановки использовалось снижение до 5% верхней оценки точности, определяемой как

$$gap = \frac{UB - LB}{LB}$$

Полученные результаты эксперимента представлены в табл. 2, которая организована следующим образом: первая группа столбцов описывает задачу, включая её обозначение ID, количество вершин n и кластеров m , а также стоимость стартового решения UB_0 , полученного эвристикой PCGLNS. Затем следуют три группы столбцов для решателя Gurobi и двух предлагаемых алгоритмов. Каждая группа содержит время счета в секундах, наилучшее значение нижней границы LB и оценку погрешности gap в процентах. Задачи, в которых один из предлагаемых алгоритмов превосходит Gurobi по производительности, выделены жирным шрифтом.

⁴Held M. A Dynamic Programming Approach to Sequencing Problems / M. Held, R. M. Karp // Journal of the Society for Industrial and Applied Mathematics. 1962. Т. 10, № 1. С. 196–210. URL: <http://www.jstor.org/stable/2098806>.

⁵Salman R. Branch-and-bound for the Precedence Constrained Generalized Traveling Salesman Problem / R. Salman, F. Ekstedt, P. Damaschke // Operations Research Letters. 2020. Т. 48, № 2. С. 163–166.

Таблица 2 – Сравнение решений задачи PCGTSP

Задача					Gurobi			Ветвей и границ			DP		
№	ID	n	m	UB ₀	Время	LB	gap, %	Время	LB	gap, %	Время	LB	gap, %
1	br17.12	92	17	43	82.00	43	0.00	11.2	43	0.00	27.3	43	0.00
2	ESC07	39	8	1730	0.24	1730	0.00	1.3	1726	0.23	8.37	1730	0.00
3	ESC12	65	13	1390	3.35	1390	0.00	4.3	1385	0.36	14.99	1390	0.00
4	ESC25	133	26	1418	10.61	1383	0.00	32	1383	0.00	60.69	1383	0.00
5	ESC47	244	48	1399	3773	1064	4.93	36000	980	42.76	36000	981	42.61
6	ESC63	349	64	62	25.35	62	0.00	1.3	62	0.00	0.52	62	0.00
7	ESC78	414	79	14872	1278.45	14630	1.66	1.3	14594	1.63	0.68	14594	1.63
8	ft53.1	281	53	6194	36000	5479	13.04	36000	4839	28.27	36000	4839	28.27
9	ft53.2	274	53	6653	36000	5511	20.7	36000	4934	34.84	36000	4940	34.68
10	ft53.3	281	53	8446	36000	6354	32.92	36000	5465	54.55	36000	5465	54.55
11	ft53.4	275	53	11822	20635	11259	5.00	35865	11274	4.86	2225	11290	4.71
12	ft70.1	346	70	32848	83.70	31521	4.21	36000	31153	5.44	36000	31177	5.36
13	ft70.2	351	70	33486	36000	31787	5.35	36000	31268	7.09	36000	31273	7.08
14	ft70.3	347	70	35309	36000	32775	7.73	36000	32180	9.72	36000	32180	9.72
15	ft70.4	353	70	44497	36000	41160	8.11	36000	38989	14.13	36000	41640	6.86
16	kro124p.1	514	100	33320	36000	29541	12.79	36000	27869	19.56	36000	27943	19.24
17	kro124p.2	524	100	35321	36000	29983	17.80	36000	28155	25.45	36000	28155	25.45
18	kro124p.3	534	100	41340	36000	30669	34.79	36000	28406	45.53	36000	28406	45.53
19	kro124p.4	526	100	62818	36000	46033	36.46	36000	38137	64.72	36000	38511	63.12
20	p43.1	203	43	22545	4691	21677	4.00	36000	738	2954.88	36000	788	2761.04
21	p43.2	198	43	22841	36000	21357	6.94	36000	749	2949.53	36000	877	2504.45
22	p43.3	211	43	23122	36000	15884	45.57	36000	898	2474.83	36000	906	2452.10
23	p43.4	204	43	66857	36000	45198	47.92	4470	66846	0.00	333.02	66846	0.00
24	prob.100	510	99	1474	36000	805	83.10	36000	632	133.23	36000	632	133.23
25	prob.42	208	41	232	13310	196	4.86	36000	149	55.70	36000	153	51.63
26	rbg048a	255	49	282	24.22	282	0.00	0.9	272	3.68	0.25	272	3.68
27	rbg050c	259	51	378	13.83	378	0.00	0.2	372	1.61	0.25	372	1.61
28	rbg109a	573	110	848	6	848	0.00	2407	812	4.43	682	809	4.82
29	rbg150a	871	151	1415	15	1382	2.38	0.4	1353	4.58	0.53	1353	4.58
30	rbg174a	962	175	1644	27	1605	2.43	0.4	1568	4.85	0.67	1568	4.85
31	rbg253a	1389	254	2376	61	2307	2.99	0.8	2269	4.72	1.42	2269	4.72
32	rbg323a	1825	324	2547	416	2490	2.29	2.0	2448	4.04	3.59	2448	4.04
33	rbg341a	1822	342	2101	18470	2033	4.97	36000	1840	14.18	36000	1840	14.18
34	rbg358a	1967	359	2080	17807	1982	4.95	36000	1933	7.60	36000	1933	7.60
35	rbg378a	1973	379	2307	32205	2199	4.91	36000	2032	13.53	36000	2031	13.59
36	ry48p.1	256	48	13135	36000	11965	9.78	36000	10739	22.31	36000	10764	22.03
37	ry48p.2	250	48	13802	36000	12065	14.39	36000	10912	26.48	36000	11000	25.47
38	ry48p.3	254	48	16540	36000	13085	26.40	36000	11732	40.98	36000	11822	39.91
39	ry48p.4	249	48	25977	36000	22084	17.62	18677	25037	3.75	14001	25043	3.73

Алгоритм 1 DP :: индуктивное построение таблицы поиска

Вход: оргграф G , частичный порядок Π , слой таблицы поиска \mathcal{L}_k

Выход: $(k + 1)$ -ый слой \mathcal{L}_{k+1}

```
1: инициализация  $\mathcal{L}_{k+1} = \emptyset$ 
2: for all  $\mathcal{C}' \in \mathfrak{I}_k$  do
3:   for all кластер  $V_l \in \mathcal{C} \setminus \mathcal{C}'$ , s.t.  $\mathcal{C}' \cup \{V_l\} \in \mathfrak{I}_{k+1}$  do
4:     for all  $v \in V_1$  и  $u \in V_l$  do
5:       if есть состояние  $S = (\mathcal{C}', U, v, w) \in \mathcal{L}_k$ , s.t.  $(w, u) \in E$  then
6:         создаем новое состояние  $S' = (\mathcal{C}' \cup \{V_l\}, V_l, v, u)$ 
7:          $S'[cost] = \min\{S[cost] + c(w, u) : S = (\mathcal{C}', U, v, w) \in \mathcal{L}_k\}$ 
8:          $S'[pred] = \arg \min\{S[cost] + c(w, u) : S = (\mathcal{C}', U, v, w) \in \mathcal{L}_k\}$ 
9:          $S'[LB] = S'[cost] + \max\{L_1, L_2, L_3\}$ 
10:        if  $S'[LB] \leq UB$  then
11:          добавляем  $S'$  к  $\mathcal{L}_{k+1}$ 
12:        end if
13:      end if
14:    end for
15:  end for
16: end for
17: return  $\mathcal{L}_{k+1}$ 
```

Для 13 из 39 задач (33%) один из предложенных алгоритмов показал лучшую производительность (в 7 случаях в точности решения, в 12 случаях во времени счета), Для 10 постановок, включая одни из самых больших *rbg323a* и *rbg358a* (1825 и 1967 вершин соответственно) было получено решение с точностью 5%. В целом, хотя Gurobi демонстрирует в среднем чуть лучшую производительность, предложенные алгоритмы за редким исключением, показывают вполне сопоставимые результаты.

В **третьей главе** исследуется применение непрерывно-дискретных моделей оптимизации в САПР УП машин листовой резки с ЧПУ на примере задачи непрерывной резки *CCP* на Евклидовой плоскости $\mathbb{R} \times \mathbb{R}$. Возьмём N попарно непересекающихся плоских контуров $\{C_1, C_2, \dots, C_N\}$. В данной работе рассматриваются только контуры C_i , состоящие из отрезков прямых линий и дуг окружностей – именно такие геометрические примитивы поддерживаются программным обеспечением современного оборудования фигурной резки с ЧПУ. Выберем также точки M_0, M_{N+1} – начало и конец маршрута резки. Задача непрерывной резки (*Continuous Cutting Problem, CCP*) состоит в поиске: N точек врезки $M_i \in C_i, i \in \overline{1, N}$ и последовательности обхода контуров C_i (перестановки N элементов $I = (i_1, i_2, \dots, i_N)$). Результатом решения задачи будет являться маршрут

$$\mathcal{R} = \langle M_0, M_{i_1}, M_{i_2}, \dots, M_{i_N}, M_{N+1} \rangle \quad (8)$$

Целевая функция сводится фактически к минимизации длины холостого хода $\mathcal{L} \rightarrow \min$, где

$$\mathcal{L} = \sum_{j=0}^N |M_{i_j} M_{i_{j+1}}|, \quad (9)$$

здесь для простоты записи мы полагаем $M_{i_0} = M_0$, $M_{i_{N+1}} = M_{N+1}$.

Кроме того, решение должно удовлетворять ограничению предшествования: если \tilde{C}_i обозначает 2-мерную фигуру, ограниченную контуром C_i (в более традиционных обозначениях $C_i = \partial\tilde{C}_i$), то $\tilde{C}_p \subset \tilde{C}_q \Rightarrow i_p < i_q$, то есть вложенный контур должен быть посещён раньше, чем содержащий его, и не все перестановки $I = (i_1, i_2, \dots, i_N)$ допустимы.

Предлагаемый алгоритм [4; 7] решения задачи: 1) удаление «внешних» контуров; 2) поиск положений точек врезки (непрерывная оптимизация); 3) поиск порядка обхода контуров (дискретная оптимизация без учёта ограничений предшествования); 4) восстановление удалённых контуров.

На первом шаге мы удаляем все контура, внутри которых содержатся другие, то есть оставляем только $\{C_i | \forall j \neq i: C_j \cap \tilde{C}_i = \emptyset\}$, тем самым как правило сокращая размерность задачи с N до некоторого $N' \leq N$.

На втором шаге мы предполагаем перестановку контуров $I = (i_1, i_2, \dots, i_N)$ фиксированной, выбираем произвольные положения точек врезки $M_i \in C_i$ на контурах и подвергаем их последовательной релаксации: для каждой точки M_i мы полагаем все остальные M_j ($i \neq j$) фиксированными и находим положение M_i , минимизирующее функционал

$$|M_{i-1} M_i| + |M_i M_{i+1}| \rightarrow \min_{M_i \in C_i}$$

На практике этот процесс очень быстро сходится, давая за время $O(N')$ позиции точек врезки на всех контурах.

На третьем шаге предполагается воспользоваться каким-либо методом дискретной оптимизации для поиска перестановки $I = (i_1, i_2, \dots, i_N)$. В данной диссертационной работе использован метод переменных окрестностей (Variable Neighborhood Search, VNS⁶). Мы также начинаем с произвольной (случайной) перестановки I , строим окрестность этой перестановки $\mathcal{N}(I)$ (например, все перестановки, полученные из неё всеми однократными попарными перестановками контуров), для каждой перестановки $I' \in \mathcal{N}(I)$ находим оптимальные позиции точек врезки для минимизации холостого хода

$$\mathcal{L}(I') = \min_{M_i \in C_i, \forall i} \mathcal{L}(M_1, M_2 \dots M_N | I')$$

(как описано выше на втором шаге алгоритма) и выбираем ту из перестановок I' , которая даёт наименьшее значение длины холостого хода (9), и к ней применяется этот же процесс. Если же понизить длину холостого хода не получается, рассматриваются всё более широкие окрестности $\mathcal{N}(I)$ (например, полученные тройными

⁶Hansen P. Variable neighbourhood search: methods and applications / P. Hansen, N. Mladenović, J. A. Moreno Pérez // Annals of Operations Research. 2010. T. 175, № 1. С. 367–407.

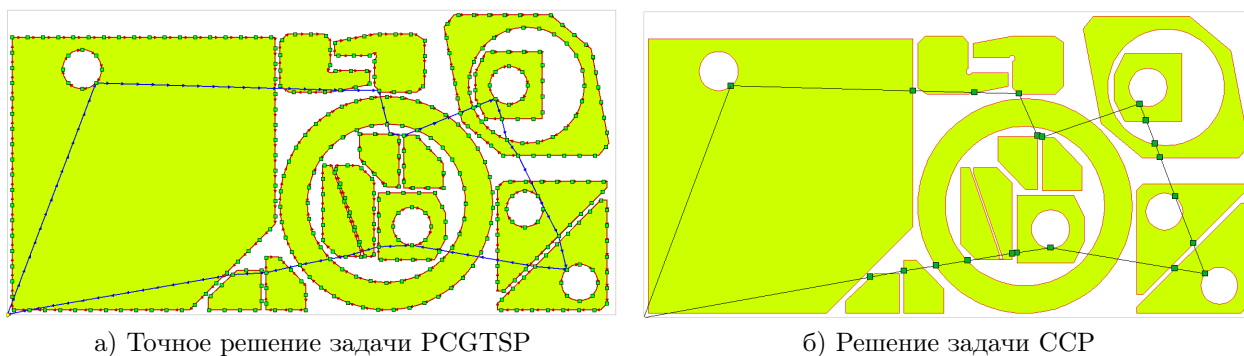


Рис. 3 – Решения задач резки для задания № 464

Таблица 3 – Сравнение качества решений задач CCP и GTSP

Задание	№ 229	№ 464	№ 3211
Кол-во деталей	11	14	17
Кол-во контуров	12	21	22
Общий периметр, м	24.609	21.717	25.051
Кол-во точек GTSP	491	429	493
\mathcal{L}_{GTSP} , м	7.729	4.743	4.557
\mathcal{L}_{CCP} , м	7.727	4.706	4.536

перестановками контуров и т.п.), пока не будет обнаружена перестановка I , которая уже не может быть улучшена. Она и считается решением задачи вместе с соответствующими ей позициями точек врезки.

На четвёртом и последнем шаге алгоритма мы восстанавливаем контуры, удалённые на первом шаге и находим точки врезки и для них как пересечение маршрута $\mathcal{R} = \langle M_0, M_1, M_2, \dots, M_{N'}, M_{N+1} \rangle$ с каждым из (удалённых) контуров $M_i = C_i \cap \mathcal{R}$, причём из нескольких таких точек выбирается самая последняя по ходу маршрута \mathcal{R} . После добавления таким образом всех «внешних» контуров и соответствующих им точек врезки, мы получаем уже полный маршрут, который посещает все исходные контуры, причём внутренние контуры посещаются строго раньше содержащих их внешних. Получаемый таким образом за линейное время $O(N)$ полный маршрут является оптимальным решением исходной задачи непрерывной резки, соблюдающим ограничение предшествования.

С практической точки зрения алгоритм работает очень хорошо, быстро находя хорошие маршруты резки. Однако, объективная оценка качества решения сложна. В данной работе в качестве базы сравнения использовался алгоритм на основе динамического программирования⁷, который находит точное решение задачи GTSP для количества контуров $N \leq 33$. Использовались несколько раскройных планов, содержащих реальные детали, см. табл. 3.

⁷Chentsov A. G. Model of megalopolises in the tool path optimisation for CNC plate cutting machines / A. G. Chentsov, P. A. Chentsov, A. A. Petunin, A. N. Sesekin // International Journal of Production Research. 2018. Т. 56, № 14. С. 4819–4830.

На рис. 3а показано точное решение задачи GTSP, а на рис. 3б — решение задачи непрерывной резки, полученное вышеописанным алгоритмом для того же раскройного плана. Видно, что оба алгоритма дают практически идентичные маршруты резки. Основное отличие вызвано необходимостью дискретизации контуров в ходе сведения задачи непрерывной резки к GTSP. Это приводит, в частности к тому, что длина холостого хода в задаче GTSP получается немного больше, чем в задаче ССР, что видно в табл. 3.

В **четвёртой главе** рассматривается методология использования алгоритмов решения разных классов задачи резки в существующих CAD/CAM-системах. Поскольку требуется обеспечить совместную работу программного обеспечения, разработанного в разное время разными командами разработчиков, чрезвычайно важными становятся вопросы организации эффективных программных интерфейсов.

Например, для хранения и обмена геометрической информацией в САПР «Сириус» используется унаследованный двоичный формат DBS [17], который при значительных достоинствах значительно затрудняет взаимодействие разных подсистем. В рамках данной диссертационной работы было принято решение использовать по возможности открытые текстовые форматы для хранения и передачи данных. В качестве основного формата был выбран формат JavaScript Object Notation (JSON⁸), ввиду того, что он является стандартом де-факто во многих современных приложениях для обмена данными, и достаточно выразителен.

```
1  [{
2    "partid": "LIST",
3    "paths": [
4      [
5        [0, 0, 0],
6        [0, 500, 0],
7        [700, 500, 0],
8        [700, 0, 0],
9        [0, 0, 0]]
10   ]},
11  {
12    "partid": "RING",
13    "paths": [
14      [
15        [205, 405, -1],
16        [205, 5, -1],
17        [205, 405, 0]],
18      [
19        [205, 305, 1],
20        [205, 105, 1],
21        [205, 305, 0]]
22   ]}]
```

Листинг 1 – JSON-файл с геометрией простейшей раскройной карты

Для хранения и обмена информацией о геометрии деталей и раскройной карты была разработана максимально простая схема JSON, пример такого файла для простейшей раскройной карты приведён в Листинге 1.

Использование JSON в качестве формата обмена данными оказалось удобным на практике, поэтому позднее были разработаны другие форматы файлов, в

⁸Introducing JSON. URL: <https://www.json.org/>.

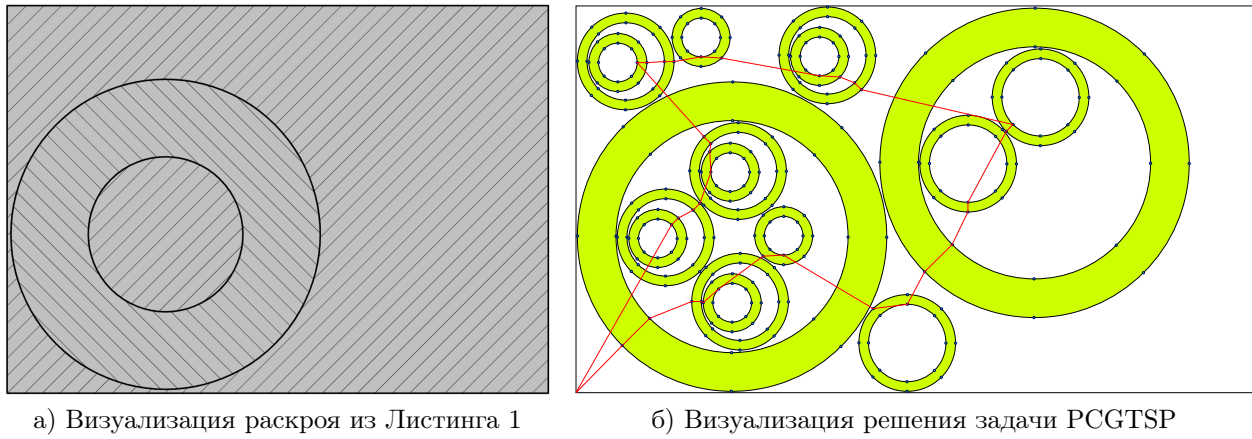


Рис. 4 – Примеры визуализации на основе SVG

частности задания на резку и результатов резки. Все они были формально описаны в виде JSON-схем⁹, см. [12].

Использование открытых форматов файлов позволило также значительно упростить вопросы, связанные с визуализацией обрабатываемой информации. Традиционно для этого приходится писать отдельный код, имеющий зачастую довольно сложную структуру и решающий множество задач, включая геометрические расчёты и организацию пользовательского интерфейса. В рамках данной диссертационной работы в качестве средства визуализации использовался экспорт в формат Scalable Vector Graphics (SVG), который является стандартом де-факто, содержит богатые возможности визуализации и широко поддержан всеми современными браузерами. Листинг 2 показывает пример простейшего SVG-файла, сгенерированного для раскройной карты, представленной на Листинге 1.

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <svg
3   xmlns="http://www.w3.org/2000/svg"
4   <g><g transform = "scale(1, -1)">
5   <path name="LIST" d="M 0 0
6   V 500
7   H 700
8   V 0
9   H 0 Z"/>
10  <path name="RING" d="M 205 405
11  A 200 200 0 0 0 405 205 A 200 200 0 0 0 205 5
12  A 200 200 0 0 0 5 205 A 200 200 0 0 0 205 405 Z
13  M 205 305
14  A 100 100 0 0 1 105 205 A 100 100 0 0 1 205 105
15  A 100 100 0 0 1 305 205 A 100 100 0 0 1 205 305 Z"/>
16 </g></g></svg>

```

Листинг 2 – SVG-файл для визуализации раскроя

Переход к широкому использованию SVG позволяет также использовать зрелые современные технологии каскадных таблиц стилей (Cascading Style Sheets, CSS) для управления внешним видом визуализации (включая цвета, заливки и штриховки и анимацию) и язык JavaScript для добавления к визуализации элементов интерактивности. Один из вариантов оформления SVG-файла из Листинга 2 приведён на рис. 4а.

⁹*Pezoa F. Foundations of JSON Schema / F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, D. Vrgoč // WWW '16: Proceedings of the 25th International Conference on World Wide Web. Republic, Canton of Geneva, CHE : International World Wide Web Conferences Steering Committee, 2016. С. 263–273.*

В ходе диссертационной работы была разработана пакет утилит [16], обеспечивающих конвертацию между различными форматами файлов (включая DBS, JSON, YAML, DXF и SVG для визуализации). Для визуализации решения задачи PCGTSP (на основе комбинации информации, полученной из нескольких источников), была разработана специализированная утилита [15]. Пример созданного ею изображения приведён на рис. 46.

В **заключении** сформулированы основные научные и практические результаты диссертационной работы.

В **приложении** приведены акт внедрения и описания применявшихся и разработанных в ходе диссертационной работы форматов файлов.

Основные результаты и выводы

В соответствии с целью и задачами исследования получены следующие научные и практические результаты:

1. Разработан алгоритм ветвей и границ для точного решения обобщённой задачи коммивояжёра с ограничениями предшествования. Он может быть реализован в классической схеме, а также в парадигме динамического программирования, при этом он допускает распараллеливание и демонстрирует лучшую производительность.
2. Предложенный алгоритм способен находить точные решения для задач большего размера, чем известные алгоритмы. В проведённых экспериментах было найдено решение для задачи со 151 кластером.
3. Данный алгоритм также решает важную задачу оценки качества решений, в том числе полученных другими алгоритмами.
4. Разработана схемы эффективного учёта ограничений предшествования для задач маршрутизации как в дискретной, так и в непрерывной схеме оптимизации.
5. Разработана основанная на геометрических соображениях эвристика оптимального размещения точек врезки на плоских контурах.
6. Доказано, что данная эвристика доставляет локальный минимум длины холостого хода и сформулированы два набора достаточных условий того, что полученное решение является глобальным минимумом.
7. Разработанные алгоритмы могут использоваться для решения задач сегментной непрерывной резки (SCCP и GSCCP) тем самым открывая подход к решению общей задачи прерывистой резки (ICP)
8. Разработаны форматы данных и алгоритмические схемы для обмена геометрической и маршрутной информацией и визуализации для использования в CAD/CAM-системах, а также алгоритмы преобразования формата файлов,

что позволило интегрировать разработанное ПО с САПР «Сириус» и T-Flex CAD.

9. Предложенные схемы информационного обмена позволяют интегрировать в одну подсистему автоматического проектирования маршрутов резки существующие алгоритмы оптимизации, включая алгоритмы, использующие схему непрерывной оптимизации.
10. Разработано программное обеспечение для реализации всех алгоритмов на языках C, Python и JavaScript.

Перспективы дальнейшей разработки темы. Можно выделить следующие направления дальнейшего развития и совершенствования алгоритмического и программного обеспечения САПР УП для оборудования листовой фигурной резки с ЧПУ:

1. Разработка методов получения нижних оценок для частичных подзадач GTSP; например, за счёт более точного учёта расстояний между узлами, а не только между кластерами.
2. Разработка метаэвристических алгоритмов дискретной оптимизации в задачах непрерывной резки (ССР, SCCR) и оценка производительности и качества получаемых алгоритмов.
3. Учет технологических требований термической резки для разработанных в данной диссертационной работе алгоритмов.
4. Интеграция разработанных алгоритмов с отечественными САПР для проектирования УП машин листовой резки

Публикации автора по теме диссертации

Статьи, опубликованные в рецензируемых научных журналах и изданиях, определенных ВАК РФ и Аттестационным советом УрФУ:

1. *Khachay M.* Problem-Specific Branch-and-Bound Algorithms for the Precedence Constrained Generalized Traveling Salesman Problem / M. Khachay, S. Ukolov, A. Petunin // Optimization and Applications. Т. 13078 / под ред. N. Olenov [и др.]. — Cham, Switzerland : Springer Nature Switzerland AG, 2021. — С. 136—148. — (Lecture Notes in Computer Science). — (0.8 п.л. / 0.27 п.л.) (Scopus).
2. *Petunin A.* Library of Sample Image Instances for the Cutting Path Problem / A. Petunin, A. Khalyavka, M. Khachay, A. Kudriavtsev, P. Chentsov, E. Polishchuk, S. Ukolov // Pattern Recognition. ICPR International Workshops and Challenges, 2021, Proceedings. — Berlin, Germany : Springer, 2021. — С. 227—233. — (0.5 п.л. / 0.07 п.л.) (Scopus).

3. *Petunin A.* A Novel Algorithm for Construction of the Shortest Path Between a Finite Set of Nonintersecting Contours on the Plane / A. Petunin, E. Polishchuk, S. Ukolov // *Advances in Optimization and Applications*. — Cham, Switzerland : Springer, 2021. — С. 70—83. — (0.9 п.л. / 0.3 п.л.) (Scopus).
4. *Петунин А. А.* Новый алгоритм построения кратчайшего пути обхода конечного множества непересекающихся контуров на плоскости / А. А. Петунин, Е. Г. Полищук, С. С. Уколов // *Известия ЮФУ. Технические науки*. — 2021. — № 1. — С. 149—164. — (1.0 п.л. / 0.3 п.л.)
5. *Petunin A. A.* Optimum routing algorithms for control programs design in the CAM systems for CNC sheet cutting machines / A. A. Petunin, P. A. Chentsov, E. G. Polishchuk, S. S. Ukolov, V. V. Martynov // *Proceedings of the X All-Russian Conference «Actual Problems of Applied Mathematics and Mechanics» with International Participation, Dedicated to the Memory of Academician A.F. Sidorov and 100th Anniversary of UrFU: AFSID-2020*. — American Institute of Physics Inc., 2020. — С. 020005. — (0.5 п.л. / 0.1 п.л.) (Scopus).
6. *Petunin A. A.* The termal deformation reducing in sheet metal at manufacturing parts by CNC cutting machines / A. A. Petunin, E. G. Polyshuk, P. A. Chentsov, S. S. Ukolov, V. I. Krotov // *IOP Publishing*. — 2020. — Т. 613. — С. 012041. — (0.5 п.л. / 0.1 п.л.) (WoS, Scopus).
7. *Petunin A. A.* On the new Algorithm for Solving Continuous Cutting Problem / A. A. Petunin, E. G. Polishchuk, S. S. Ukolov // *IFAC-PapersOnLine*. — 2019. — Т. 52, № 13. — С. 2320—2325. — (0.9 п.л. / 0.3 п.л.) (WoS, Scopus).
8. *Tavaeva A.* A Cost Minimizing at Laser Cutting of Sheet Parts on CNC Machines / A. Tavaeva, A. Petunin, S. Ukolov, V. Krotov // *Mathematical Optimization Theory and Operations Research*. — Cham, Switzerland : Springer, 2019. — С. 422—437. — (0.16 п.л. / 0.04 п.л.) (Scopus).
9. *Petunin A. A.* About some types of constraints in problems of routing / A. A. Petunin, E. G. Polishuk, A. G. Chentsov, P. A. Chentsov, S. S. Ukolov // *AIP Conference Proceedings*. — 2016. — Т. 1789, № 1. — С. 060002. — (0.9 п.л. / 0.18 п.л.) (WoS, Scopus).

Другие публикации:

10. *Петунин А. А.* Эффективная маршрутизация работа/беспилотного летательного аппарата в задачах с условиями предшествования / А. А. Петунин, М. Ю. Хачай, С. С. Уколов // *XIV Всероссийская мультikonференция по проблемам управления (МКПУ-2021)*. Т. 1. — Издательство Южного федерального университета, 2021. — С. 202—205. — (0.3 п.л. / 0.1 п.л.)
11. *Уколов С. С.* Алгоритм ветвей и границ для обобщённой задачи коммивояжера с ограничениями предшествования / С. С. Уколов, М. Ю. Хачай. — 2021. — URL: <https://github.com/ukoloff/PCGTSP-BnB>.

12. *Уколов С. С.* JSON-схемы файлов, используемых в САПР «Сириус» / С. С. Уколов, П. А. Ченцов. — 2021. — URL: <https://ukoloff.github.io/dbs.js/json-schema/>.
13. *Петунин А. А.* Алгоритмы оптимальной маршрутизации для систем автоматизированного проектирования управляющих программ машин листовой резки с ЧПУ / А. А. Петунин, П. А. Ченцов, Е. Г. Полищук, С. С. Уколов, В. В. Мартынов // Актуальные проблемы прикладной математики и механики. — Институт математики и механики УрО РАН им. Н.Н. Красовского, 2020. — С. 58—59. — (0.2 п.л. / 0.04 п.л.)
14. *Таваева А. Ф.* Разработка инвариантного модуля генерации управляющих программ для машин лазерной резки. Вопросы интеграции с САД/САМ системами / А. Ф. Таваева, Е. Н. Шипачева, П. А. Ченцов, А. А. Петунин, С. С. Уколов, А. П. Халявка // Актуальные проблемы прикладной математики и механики. — Институт математики и механики УрО РАН им. Н.Н. Красовского, 2020. — С. 70—71. — (0.2 п.л. / 0.03 п.л.)
15. *Уколов С. С.* Визуализация решения задачи PCGTSP / С. С. Уколов. — 2020. — URL: <https://ukoloff.github.io/j2pcgtsp/>.
16. *Уколов С. С.* Конвертеры открытых форматов для САПР «Сириус» / С. С. Уколов. — 2019. — URL: <https://github.com/ukoloff/dbs.js>.
17. *Уколов С. С.* Описание формата DBS / С. С. Уколов, В. И. Кротов. — 2018. — URL: <https://github.com/ukoloff/dbs.js/wiki/DBS>.
18. *Петунин А. А.* САПР «Сириус» – оптимизация раскроя и резки листовых материалов в единичном производстве / А. А. Петунин, В. И. Кротов, С. С. Уколов, В. В. Видяпин // САПР и графика. — 1999. — № 10. — С. 42—48. — (0.4 п.л. / 0.1 п.л.)

Подписано в печать __.__.2022. Формат 60×84 1/16. Усл. печ. л. 1,42.

Тираж 100 экз. Заказ № _____.