Федеральное государственное автономное образовательное учреждение
высшего образования «Уральский федеральный университет имени первого
Президента России Б.Н. Ельцина»
Институт новых материалов и технологий
Кафедра информационных технологий и автоматизации проектирования

На правах рукописи

Уколов Станислав Сергеевич

Разработка алгоритмов оптимальной маршрутизации режущего инструмента для машин термической резки с ЧПУ

Специальность 05.13.12 —
«Системы автоматизации проектирования (промышленность)»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук, доцент
Петунин Александр Александрович

Екатеринбург — 2021

Оглавление

Введение

Актуальность темы исследования. Современное производство предъявляет высокие требования к качеству заготовок, технико-экономическому уровню выпускаемой продукции, что приводит к увеличению затрат на проектирование и технологическую подготовку производства. Одним из направлений повышения эффективности использования производственных ресурсов является совершенствование безотходных технологий в металлообрабатывающих производствах и возрастание степени их автоматизации.

Раскройно-заготовительные операции, являясь началом большинства производственных процессов, оказывают существенное влияние на трудоёмкость и экономичность изготовления детали. 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). К данному типу оборудования относятся станки газовой, лазерной, плазменной, электроэрозионной и гидроабразивной резки металла. Станки листовой резки имеют множество преимуществ: возможность обработки многих видов материалов различной толщины, высокая скорость резки, возможность обработки контуров различной сложности, адаптация к постоянным изменениям номенклатуры выпускаемой продукции. Использование оборудования с ЧПУ, предполагает применение средств автоматизации проектирования управляющих программ (CAM-систем). При использовании современных CAD/CAM систем, предназначенных для автоматизированного проектирования раскроя и подготовки управляющих программ (далее – УП) для машины 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 листового материала. На этапе проектирования раскроя возникает известная задача оптимизации фигурного раскроя листового материала,

которая с точки зрения геометрической оптимизации относится к классу трудно решаемых проблем раскроя-упаковки (*Cutting & Packing*). На следующем этапе проектирования УП осуществляется процесс назначения траектории перемещения режущего инструмента (маршрута резки) для полученного на первом этапе варианта раскроя. На этом этапе возникают актуальные научно-практические задачи оптимизации маршрута режущего инструмента. Целью этих задач 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 термической врезки в листовой материал с учетом технологических ограничений процесса резки. Следует отметить, что современные специализированные САПР предоставляют базовый инструментарий 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, обусловленных необходимостью уменьшения термических деформаций материала, которые могут приводить к существенным искажениям геометрии вырезаемых деталей. Отметим также, что пользователи САПР зачастую используют интерактивный режим проектирования УП. При этом при проектировании маршрута резки зачастую применяется стандартная техника резки «по замкнутому контуру» и путь инструмента строится только с точки зрения минимизации холостых переходов режущего инструмента. В связи с этим актуальным направлением исследования является применение эвристических и метаэвристических подходов, которые позволяют получить решение задачи оптимальной маршрутизации режущего инструмента за приемлемое время.

Степень разработанности темы исследования. Методы проектирования технологических процессов раскроя, включая методы формирования маршрута резки, исследовались в работах как отечественных так и зарубежных ученых. Хотя разработка оптимизационных методов решения задачи раскроя не входит в круг рассматриваемых в диссертационной работе задач, тем не менее, следует

упомянуть о значительном вкладе советских и российских исследователей в теорию оптимизации раскроя-упаковки. Работы в этой предметной области были начаты выдающимися учёными В.А. Залгаллером и Л.В. Канторовичем и продолжены в уфимской научной школе Э.А. Мухачевой и её учениками: А.Ф. Валеевой, М.А. Верхотуровым, В.М. Картаком, В.В. Мартыновым, А.С. Филипповой и др. Методологические и теоретические основы создания САПР листового раскроя были заложены Н.И. Гилем, А.А. Петуниным, Ю.Г. Стояном, В.Д. Фроловским.

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 и др., а также зарубежные исследователи: E. Arkin, N. Ascheuer, D. Cattrysse, R. Dewil, L. Gambardella, J. Hoeft, Y. Jing, Y. Kim , M. Lee, S.U. Sherif, W. Yang и др. В подавляющей части работ используется дискретизация граничных контуров деталей, что позволяет применять различные хорошо разработанные математические модели дискретной оптимизации. Можно отметить только отдельные публикации, где оптимизационные алгоритмы ориентированы на поиск решений среди континуальных множеств.

Общеизвестно, что задачи маршрутизации инструмента машин листовой резки с ЧПУ относятся к NP-трудным задачам При этом следует отметить, что в настоящее время 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 раскроя и управляющих программ для машин листовой резки с ЧПУ, есть отдельные модули, которые позволяют решать 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 критерия стоимости резки с учетом рабочего хода инструмента, затрат на врезку и т.д. Следует подчеркнуть, что алгоритмы, реализованные в коммерческом программном обеспечении, не описываются, как правило, в научной литературе.

В общей проблеме маршрутизации инструмента машин листовой резки с ЧПУ можно выделить несколько классов задач: задача непрерывной резки (CCP), задача резки с конечными точками (ECP), 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 или эффективного упрощения ICP мог бы заполнить явный и существующий пробел в исследованиях.

В целом можно отметить, что за рамками исследований современных отечественных и зарубежных коллег остаются 3 принципиальных момента:

1. Разработка алгоритмов, обеспечивающих получение глобального оптимума оптимизационной задачи маршрутизации инструмента.

2. Отсутствие адекватного учета тепловых искажений заготовок при термической резке с целевой функцией стоимости, что приводит к не технологичным решениям и искажению геометрии получаемых заготовок.

3. Рассмотрение задач маршрутизации из класса ICP и задач с набором возможных точек врезки из континуального множества.

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 постановок задач, алгоритмов и программного обеспечения представляется доминантой развития методов решения исследуемой оптимизационной проблемы маршрутизации инструмента машин листовой резки с ЧПУ.

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 эвристику поиска оптимального положения точек врезки в контур детали в процессе решения задачи непрерывной резки
- Подобрать алгоритм поиска решения задачи дискретной оптимизации допускающий совместное использование с разработанной эвристикой поиска точек врезки
- Разработать точный алгоритм решения обобщённой задачи коммивояжера с ограничениями предшествования (PCGTSP), позволяющий оценивать качество решений на основе вычисления нижней оценки
- Разработать программное обеспечение, реализующее разработанные алгоритмы
- Разработать методику использования алгоритмов оптимальной маршрутизации режущего инструмента в CAD/CAM-системах на примере САПР «Сириус»

Научная новизна результатов.

1. Разработан эвристический алгоритм непосредственного решения задачи непрерывной резки без сведения к обобщённой задаче коммивояжера и применения дискретизации
2. Предложена схема учёта ограничений предшествования на основе геометрических соображений, эффективно уменьшающая вычислительную сложность задачи
3. Разработан точный алгоритм, позволяющий находить нижние оценки для обобщённой задачи коммивояжера с ограничениями предшествования
4. Сформулированы более строгие ограничения, позволяющие сократить дерево перебора для обобщённой задачи коммивояжера с ограничениями предшествования

Теоретическая и практическая значимость работы заключается:

1. Прямое решение задачи непрерывной резки позволяет уменьшить длину холостого хода режущего инструмента по сравнению с используемыми в настоящее время технологиями на основе решения задачи GTSP и её аналогов

2. Новая схема учёта ограничений предшествования для задачи непрерывной резки, уменьшая размерность задачи и время счёта, позволяет получать решения задач большего размера, в особенности, имеющих большую вложенность контуров

3. Кроме ускорения процесса проектирования УП, алгоритм решения задачи непрерывной резки также может применяться как составная часть других алгоритмов, что в свою очередь позволяет использовать его для решения задач класса GSCCP (обобщённая сегментная непрерывная резка), то есть искать подходы к решению самой общей задачи маршрутизации режущего инструмента – ICP (прерывистой резки)

4. Предложенные схемы учёта ограничений предшествования и поиска точек врезки могут сочетаться с другими алгоритмами дискретной оптимизации, кроме использованного в данной работе метода переменных окрестностей, что может привести к созданию новых алгоритмов решения задачи непрерывной резки

5. Представленные в данной диссертационной работе необходимые условия глобального минимума могут быть использованы при разработке новых алгоритмов, например, для ограничения перебора или наоборот, его возобновления с целью улучшения качества решения.

6. Предложенный алгоритм решения обобщённой задачи коммивояжера с ограничениями предшествования способен находить точные решения для задач, которые были ранее недоступны для точного решения

7. За счёт вычисления нижних оценок, данный алгоритм может использоваться также для оценки качества решений, полученных при помощи других алгоритмов

8. В предложенной схеме использованы несколько способов получения нижней оценки, что позволяет получать более точные результаты и сокращать объём перебора

9. Допустимо использование также других способов получения оценок, что открывает дорогу новым теоретическим и практическим исследованиям

10. Предложенные алгоритмы могут использоваться для подготовки УП для машин термической резки с ЧПУ в составе САПР «Сириус», а также других CAD/CAM-систем

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. Схема сведения задачи непрерывной резки с ограничениями предшествования к аналогичной меньшего размера и не имеющей ограничениями

2. Эвристика поиска точек врезки в плоские контура, не использующая дискретизации

3. Достаточные условия, при которых полученный маршрут доставляет глобальный минимум длины холостого хода инструмента

4. Схема использования ограничений предшествования для сокращения перебора в обобщённой задаче коммивояжера

5. Точный алгоритм решения обобщённой задачи коммивояжера с ограничениями предшествования с обновлением нижней границы

6. Форматы файлов 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*, Созополь, Болгария, 2016 год

– *MiM2016: on Manufacturing, Modelling, Management & Control*, Труа, Франция, 2016 год

– *ASRTU 2017 International Conference on Intellectual Manufacturing*, Харбин, Китайская Народная Республика, 2017 год

– *Mathematical Optimization Theory And Operations Research*, Екатеринбург, Россия, 2019 год

– *Manufacturing Modelling, Management and Control - 9th MIM 2019*, Берлин, Германия, 2019 год

– *XVI Всероссийская научно-практическая конференция «Перспективные системы и задачи управления»*, Домбай, Россия, 2021 год

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 опубликовано ? научных работ, среди которых ? статей в журналах, определенных ВАК и Аттестационным советом УрФУ, включая ? статей в изданиях, индексируемых в международных базах WoS и Scopus.

Структура и объем диссертации. Диссертация состоит из введения, четырех глав, заключения, списка литературы и ? приложений. Общий объем диссертации составляет ? с., в том числе ? рисунков, ? таблиц. Список литературы включает ? наименований.

Глава 1

Задача оптимизации маршрута режущего инструмента для машин термической резки с ЧПУ

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam.

Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad

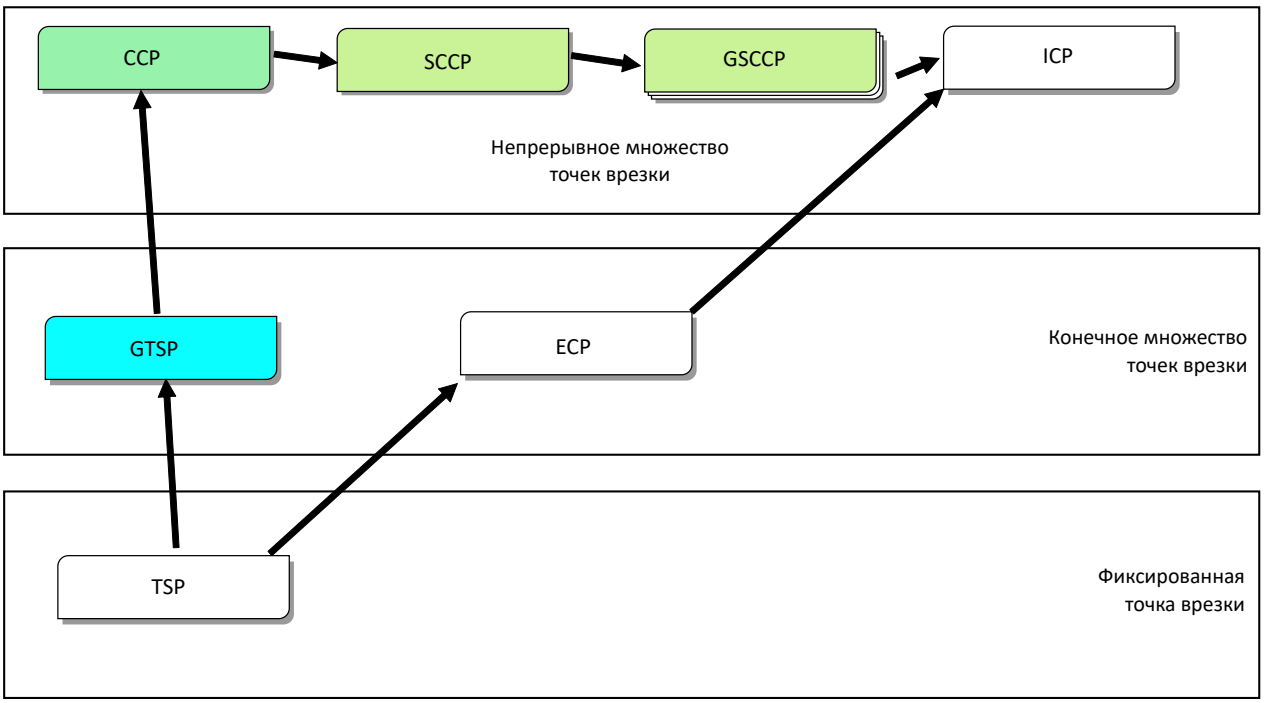


Рис. 1.1. Классификация задач резки

litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

1.1. Выводы по Главе 1

1.

Глава 2

Эвристический алгоритм решения задачи непрерывной резки CCP

2.1. Постановка задачи

Рассмотрим евклидову плоскость $\mathbb{R} \times \mathbb{R}$ и на ней фигуру $\mathcal{B}$ (в большинстве практических случаев – прямоугольник), ограниченную замкнутым контуром. Это – модель листового материала, подлежащего резке. Пусть N попарно непересекающихся плоских контуров $\{C_1, C_2, ...C_N\}$ расположены внутри $\mathcal{B}$, ограничивая n деталей $\{A_1, A_2...A_n\}$. Деталь может быть ограничена одним или несколькими контурами (одним внешним и несколькими отверстиями), так что в общем случае $n \leqslant N$.

Контуры C_i могут быть произвольной формы, но мы будем рассматривать только состоящие из (конечного числа) отрезков прямых линий и дуг окружностей, так как именно такие геометрические примитивы поддерживаются программным обеспечением современных машин термической резки с ЧПУ. Частный случай, когда контура состоят только из отрезков прямых, сводится к одному из вариантов задачи обхода прямоугольников (Touring Polygon Problem, TPP), см. [2].

Далее, внутри $\mathcal{B}$ (как правило, на границе) выберем две точки и обозначим их M_0, M_{N+1} (почти всегда $M_0 = M_{N+1}$), которые будут использоваться как начало и конец маршрута резки.

Задача непрерывной резки (Continuous Cutting Problem, CCP) состоит в поиске:

1. N штук точек врезки $M_i \in C_i, i \in \overline{1, N}$

2. Последовательности обхода контуров C_i, то есть перестановки N элементов $I = (i_1, i_2, ...i_N)$

Результатом решения задачи будет являться маршрут

$$\mathfrak{R} = \langle M_0, M_{i_1}, M_{i_2}, \dots M_{i_N}, M_{N+1} \rangle \tag{2.1}$$

Целевая функция в данном случае сильно упрощается по сравнению с общей задачей маршрутизации резки и сводится фактически к минимизации длины холостого хода:

$$\mathcal{L} = \sum_{j=0}^{N} |M_{i_j} M_{i_{j+1}}| \tag{2.2}$$
$$\mathcal{L} \to \min$$

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

Кроме того, мы потребуем, чтобы искомое решение задачи удовлетворяло описанному выше ограничению предшествования.

Хотя контуры C_i по условию не пересекаются, они могут быть вложены друг в друга: $\widetilde{C}_a \subset \widetilde{C}_b$, где $\widetilde{C}_a$ обозначает 2-мерную фигуру, ограниченную контуром C_a (в более традиционных обозначениях $C_a = \partial\widetilde{C}_a$). В общей задаче маршрутизации режущего инструмента это соответствует двум разным случаям (наличие отверстий в деталях с одной стороны и размещение меньших деталей в отверстиях больших), но в нашем случае оба этих варианта обрабатываются одинаково.

Если один контур расположен внутри другого, то внутренний должен быть вырезан (посещён) ранее, чем внешний: $\widetilde{C}_a \subset \widetilde{C}_b \Rightarrow i_a < i_b$, в перестановке $I = (i_1, i_2, ...i_N)$. Таким образом, множество допустимых перестановок ограничено.

2.2. Алгоритм решения задачи непрерывной резки

Предлагаемый алгоритм решения задачи непрерывной резки (см. [7; 10]) состоит из нескольких шагов, что хорошо соответствует самой природе решаемой задачи.

2.2.1. Удаление «внешних» контуров

Для автоматического соблюдения ограничения предшествования, мы начинаем с удаления всех контуров, внутри которых есть вложенные контура, так, чтобы остались только:

$$\{C_i | \forall j \neq i : C_j \cap \widetilde{C}_i = \varnothing\}$$

В общем случае это приводит к уменьшению (в некоторых случаях – существенному) сложности задачи (с N до некоторого N'), что в свою очередь сокращает время счёта на втором и в особенности третьем шагах алгоритма.

2.2.2. Непрерывная оптимизация

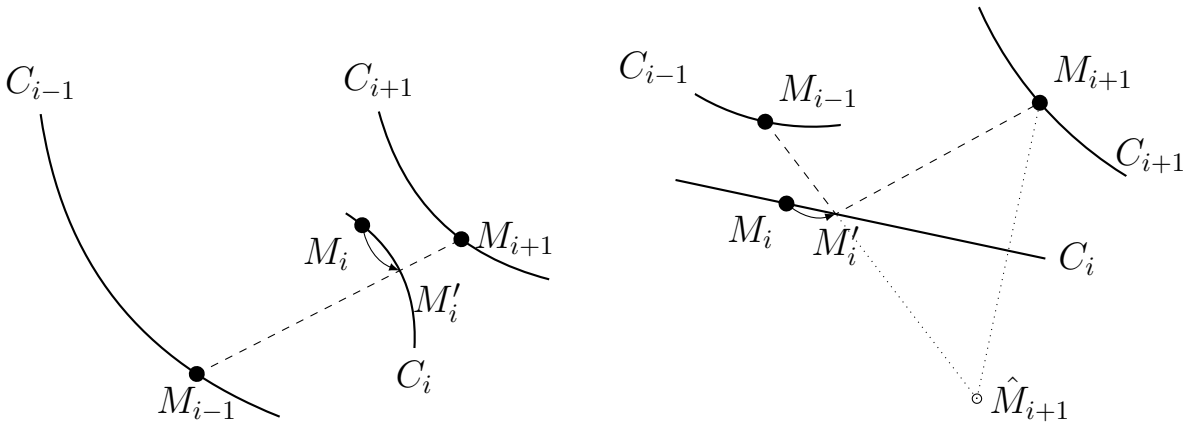
На этом этапе мы полагаем, что последовательность обхода контуров $I = (i_1, i_2, ...i_N)$ задана (фиксирована) и ищем координаты точек врезки $M_i \in C_i$ во все контура, минимизируя полную длину холостого хода (2.2). Для этого, начальные позиции точек врезки выбираются произвольным образом (например, случайно) и затем положение одной (каждой) из точек M_i изменяется, а все остальные остаются неподвижны: $\mathcal{L}(M_i) \to \min$. Большинство слагаемых в целевой функции (2.2) при этом постоянны, так что сама функция упрощается до

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

Несложный геометрический анализ показывает, что если точки M_{i-1} и M_{i+1} расположены по разные стороны сегмента контура C_i, то оптимальное положение точки врезки M_i оказывается на пересечении с этим сегментом: $M_i = M_{i-1}M_{i+1} \cap C_i$ (если, конечно, такое пересечение существует; в противном случае решением будет один из концов сегмента), см. рис. 2.1а.

Если же точки располагаются с одной стороны сегмента, решение легко находится при помощи *принципа Ферма*, или другими словами правила «угол падения равен углу отражения» (или опять на одном из концов сегмента), см. рис. 2.1б.

Поиск позиции точки врезки выполняется для всех N контуров поочерёдно, и весь этот процесс повторяется пока длина холостого пути (2.2) не стабилизируется с некоторой наперёд заданной точностью δ (на шаге 4), см. Алгоритм 1.



(а) На пересечении звена ломаной (б) С использованием *принципа Ферма*

Рис. 2.1. Оптимальное положение точки врезки

Алгоритм 1 CCP :: Релаксация позиций точек врезки

1: инициализация $M_i \in C_i, \forall i$ {Случайным образом}
2: $d \leftarrow 0$
3: $d_0 \leftarrow \infty$
4: **while** $|d - d_0| > \delta$ **do**
5: **for all** $i \in \overline{1, N}$ **do**
6: $M_i \leftarrow \arg\min_{M_i \in C_i} |M_{i-1}M_i| + |M_iM_{i+1}|$
7: $d_0 \leftarrow d$
8: $d \leftarrow \mathcal{L}(M_1, M_2, \ldots, M_N)$ {см. (2.2)}
9: **end for**
10: **end while**
11: **return** $\{M_1, M_2, \ldots, M_N\}$

На практике весь процесс хорошо сходится за время $O(N)$ и поэтому многократно применяется в качестве подпрограммы на следующем шаге.

2.2.3. Дискретная оптимизация

Наиболее вычислительно сложная задача заключается в поиске перестановки $I = (i_1, i_2, ...i_N)$, минимизирующей полную длину холостого хода $\mathcal{L} \to \min$. Фактически, это решение задачи коммивояжёра (Traveling Salesman Problem, TSP), только длина пути вычисляется не аддитивно, а при помощи процесса непрерывной оптимизации, описанного на предыдущем шаге.

В данной работе для поиска такой перестановки применяется метод переменных окрестностей (Variable Neighborhood Search, VNS [3]), см. Алгоритм 2.

Алгоритм 2 CCP :: Дискретная оптимизация

1: Инициализация $I = (i_1, i_2, ...i_N)$ {выбирается случайным образом}
2: $k \leftarrow 1$
3: **while** $k < k_{max}$ **do**
4: $I' = \arg\min_{I' \in \mathcal{N}^k(I)} \mathcal{L}(I')$
5: **if** $\mathcal{L}(I') < \mathcal{L}(I)$ **then**
6: $I \leftarrow I'$
7: $k \leftarrow 1$
8: **else**
9: $k \leftarrow k + 1$
10: **end if**
11: **end while**
12: **return** I

На шаге 4 многократно применяется непрерывная оптимизация из предыдущего этапа:

$$\mathcal{L}(I') = \min_{M_1, M_2 \ldots M_N} \mathcal{L}(M_1, M_2 \ldots M_N | I')$$

Окрестности $\mathcal{N}^k(I)$ различного размера конструируются разнообразными способами, например:

- Все возможные парные перестановки (то есть, окрестности размера 1 в смысле транспозиционной метрики)
- Циклические перестановки 3 контуров. Поскольку всего таких перестановок получается $O(N^3)$, выбираются только те из них, в которых задействованные контуры расположены в исходной перестановке $I = (i_1, i_2, ...i_N)$ не далее, чем на предопределённом расстоянии друг от друга; это предопределённое расстояние является параметром алгоритма
- Подобным же образом, выбираются циклические перестановки 4 контуров, лежащих не далее заданного расстояния друг от друга в исходной перестановке $I = (i_1, i_2, ...i_N)$

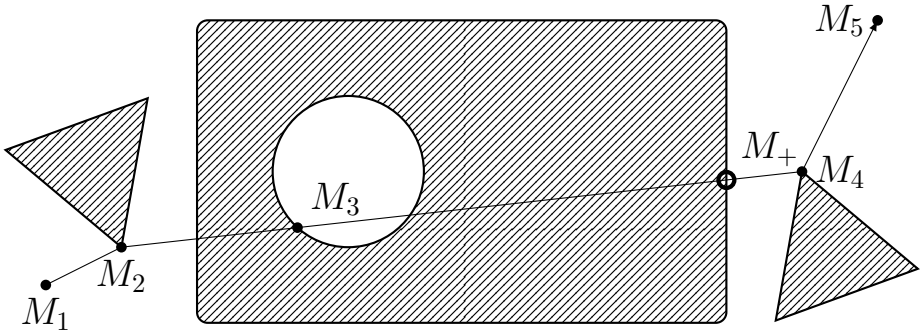


Рис. 2.2. Добавление точек врезки во «внешние» контура M_+

– Выбирается последовательный блок контуров произвольной длины и к нему применяется циклический сдвиг

– Все контуры в последовательном блоке контуров (произвольной длины) переставляются в обратном порядке

– Перестановка двух последовательных (но не смежных) блоков контуров

– Циклическая перестановка нескольких последовательно расположенных последовательных блоков контуров (произвольной но одинаковой длины)

– И ещё порядка десяти других способов генерации «близких» к исходной перестановок

Если размер некоторой окрестности $\mathcal{N}^k(I)$, получаемой одним из способов, оказывается слишком большим (что приводит к увеличению времени счёта), он легко может быть ограничен при помощи введения дополнительного параметра алгоритма, подобно тому, как это сделано для тройных и четверных циклических перестановок.

Кроме того, сам метод переменных окрестностей допускает несколько вариантов применения, например замена полного перебора (на шаге 4) на «первый подходящий» или метод Монте-Карло, но их влияние на скорость и качество получаемого решения задачи непрерывной резки требует дальнейшего исследования.

2.2.4. Восстановление удалённых контуров

На предыдущем шаге мы получили последовательность обхода контуров и точки врезки для каждого из них, но только для тех, которые не содержат

других контуров внутри себя. Теперь необходимо дополнить эту последовательность оставшимися контурами (удалёнными на первом шаге) и точками врезки для них, причём таким образом, чтобы ограничение предшествования оказалось соблюдённым.

Заметим, что маршрут, полученный к этому моменту

$$\mathfrak{R} = \langle M_0, M_1, M_2, \dots M_{N'}, M_{N'+1} \rangle \tag{2.3}$$

обязательно пересекает все исходные контуры C_i, потому что для контуров, сохранённых на первом шаге, он их явно посещает по построению шагов 2 и 3, а для остальных контуров – ввиду того, что каждый из них включает в себя один из посещённых контуров, а начальная и конечная точка M_0 и M_{N+1} всегда выбираются снаружи всех контуров C_i.

Таким образом, для каждого «внешнего» контура C_i, который пока не включён в маршрут резки, мы находим все точки пересечения с полученным маршрутом (2.3), и если таких точек несколько (что как правило и бывает), выбираем из них самую последнюю, то есть посещаемую маршрутом позже всех других, см. рис. 2.2. Сам контур C_i вставляется в перестановку в месте, соответствующем выбранной точке врезки.

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. Полная длина маршрута при этой операции, очевидно, не меняется.

Легко понять, что получаемый таким образом полный маршрут является оптимальным решением исходной задачи непрерывной резки. Действительно, если бы существовал более короткий маршрут, посещающий все контура, из него можно было бы просто удалить точки врезки, лежащие на «внешних» контурах, получив тем самым маршрут, обходящий «внутренние» контура и имеющий ту же, то есть меньшую длину. Таким образом, существует лучшее решение для задачи обхода части контуров без учёта ограничений предшествования, но это по предположению невозможно.

Таким образом, мы строго выполняем ограничение предшествования, тратя на это линейное время $O(N)$.

На этом выполнение предложенного эвристического алгоритма решения задачи непрерывной резки завершается.

2.3. Экстремальные свойства получаемого решения

С практической точки зрения, вышеописанный алгоритм оказывается вполне работоспособным и даёт хорошие результаты, как в смысле времени работы, так и качества получаемых решений. Однако, это чисто эмпирический результат, так что было бы интересно получить теоретические оценки качества работы данного алгоритма.

На рис. 2.3 показан пример маршрута резки, который не являясь кратчайшим, тем не менее неё может быть приведён к таковому никакими индивидуальными сдвигами точек врезки. Таким образом, возникает вопрос, при каких условиях предлагаемый алгоритм гарантирует получение действительно оптимального маршрута, то есть другими словами, глобального минимума оптимизационной задачи.

Наиболее важным и одновременно наиболее сложным является, конечно, третий этап – дискретная оптимизация (фактически решение задачи коммивояжёра), как с теоретической, так и с практической точки зрения. В данной же работе исследуется второй шаг алгоритма – непрерывная оптимизация. Оказывается возможным сформулировать некоторые утверждения относительно получаемых в её ходе решений.

Итак, рассмотрим следующую задачу: пусть контуры C_i на плоскости состоят только из отрезков прямых линий, они не пересекаются и не вложены друг в друга. Порядок обхода контуров заранее задан. Требуется найти кратчайшую ломаную линию, чьи вершины лежат на заданных контурах (в указанном порядке).

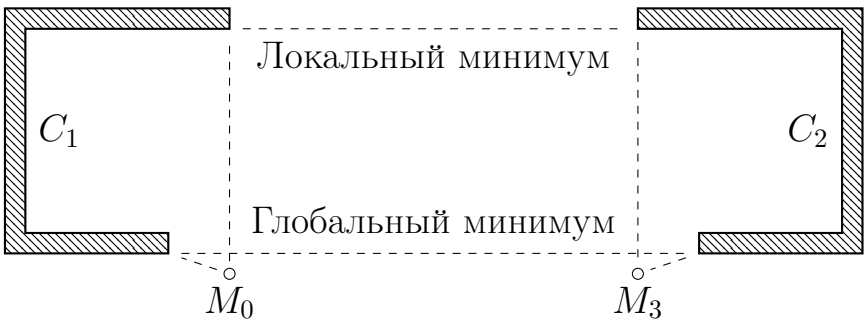


Рис. 2.3. Два маршрута резки, доставляющие локальный и глобальный минимум

Мы начинаем с (произвольной) ломаной линии L_1. Далее для каждого контура C_i мы повторяем следующую операцию: сдвигаем вершину ломаной $M_i \in C_i$ в такое положение, которое минимизирует полную длину ломаной, при этом остальные вершины M_j $(j \neq i)$ неподвижны. Это сводится к несложной геометрической задаче нахождения точки, для которой сумма расстояний до двух других фиксированных точек минимальна.

В процессе таких пошаговых сдвигов мы получаем последовательность ломаных линий $\{L_k\}$, причём последовательность длин этих ломаных линий по построению монотонно убывает. Обозначим $m = \inf |L_k|$ (точная нижняя граница длин ломаных линий) и пусть L_* – ломаная линия длины m, то есть предельная точка последовательности ломаных линий $\{L_k\}$. В качестве метрики на пространстве ломаных линий можно использовать сумму расстояний между вершинами с одинаковыми номерами.

В реальных примерах стабилизация последовательности ломаных происходит буквально в течение нескольких итераций (не более 10) и ломаная L_* действительно получается во всех численных экспериментах

По построению, длина ломаной L_* не может быть уменьшена никаким сдвигом одной из её вершин (в рамках содержащего её контура), также как и сдвигом любого количества её вершин, не являющихся соседними.

2.3.1. Локальный минимум

Утверждение 2.1. *Сдвиг нескольких соседних вершин ломаной L_* таким образом, что сдвигаемые вершины остаются на тех же самых сегментах контуров, не приводит к уменьшению полной длины ломаной.*

Доказательство. Рассмотрим сдвиг *двух* соседних вершин. Обозначим четыре последовательных вершины ломаной L_* за $M_{i-1}, M_i, M_{i+1}, M_{i+2}$.

Пусть точки $M_i \in S_i, M_{i+1} \in S_{i+1}$ лежат на прямолинейных сегментах соответствующих контуров $S_i \subset C_i, S_{i+1} \subset C_{i+1}$.

Докажем, что:

$$\forall M_i' \in S_i, M_{i+1}' \in S_{i+1} : |M_{i-1}M_i'M_{i+1}'M_{i+2}| \geqslant |M_{i-1}M_iM_{i+1}M_{i+2}|$$

Для произвольной вершины $M_i' \in S_i$: $|M_{i-1}M_i'M_{i+1}M_{i+2}|$ минимально, когда $M_i' = M_i$, и аналогично для произвольной вершины $M_{i+1}' \in S_{i+1}$: $|M_{i-1}M_iM_{i+1}'M_{i+2}|$ минимально, когда $M_{i+1}' = M_{i+1}$.

Предположим, что

$$\exists M_i' \in S_i, \exists M_{i+1}' \in S_{i+1} : |M_{i-1}M_i'M_{i+1}'M_{i+2}| < |M_{i-1}M_iM_{i+1}M_{i+2}|$$

Очевидно, что $M_i' \neq M_i, M_{i+1}' \neq M_{i+1}$.

Введём обозначение $M_i(s) = M_i + s \cdot \overrightarrow{M_iM_i'}$, $M_{i+1}(t) = M_{i+1} + t \cdot \overrightarrow{M_{i+1}M_{i+1}'}$ $(s, t \in [0, 1])$, $f(s, t) = |M_{i-1}M_i(s)M_{i+1}(t)M_{i+2}|$.

Но положения вершин M_i and M_{i+1} выбраны так, что:

$$\frac{\partial f(s,t)}{\partial s}\Big|_{(0,0)} \geqslant 0, \frac{\partial f(s,t)}{\partial t}\Big|_{(0,0)} \geqslant 0 \tag{2.4}$$

Если, например, $\partial f(s,t)/\partial s\big|_{(0,0)} = 0$, то это может быть только если точки M_{i-1}, M_i, M_{i+1} лежат на одной прямой (когда M_{i-1} и M_{i+1} по одну сторону от S_i; в противном случае заменяем M_{i-1} на её отражение относительно S_i). Таким образом, если одновременно

$$\frac{\partial f(s,t)}{\partial s}\Big|_{(0,0)} = 0 = \frac{\partial f(s,t)}{\partial t}\Big|_{(0,0)}$$

то значит все четыре точки $M_{i-1}, M_i, M_{i+1}, M_{i+2}$ лежат на одной прямой и проходящая через них ломаная фактически является отрезком прямой и заведомо кратчайшая, не может быть сделана короче.

Рассмотрим теперь случай, когда хотя бы одна из производных в (2.4) не равна нулю. Обозначим $\varphi(t) = f(t, t)$.

$$\frac{d\varphi}{dt}\Big|_0 = \frac{\partial f(s,t)}{\partial s}\Big|_{(0,0)} + \frac{\partial f(s,t)}{\partial t}\Big|_{(0,0)} > 0$$

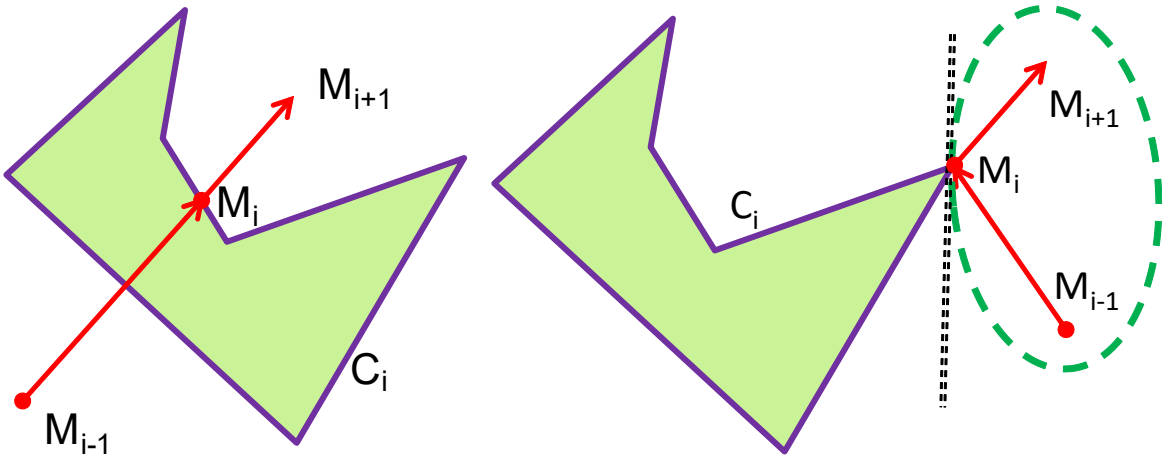
Это значит, что $\exists \tau^* \in [0, 1]$: $\varphi(\tau^*) > \varphi(0)$. Но по нашему предположению $\varphi(1) < \varphi(0)$, то есть $\varphi(0) < \varphi(\tau^*) > \varphi(1)$.

Теперь заметим, что $\varphi(t)$ представляет собой сумму четырёх слагаемых вида $\sqrt{(a + b \cdot t)^2 + (c + d \cdot t)^2}$, и каждое из этих слагаемых имеет положительную вторую производную, так что и $d^2\varphi(t)/dt^2 \geqslant 0$, а значит функция $\varphi(t)$ является выпуклой на $[0, 1]$ и не может принимать во внутренней точке интервала значения большие, чем на его концах.

Значит, наше предположение невозможно.

Мы рассмотрели сдвиг *двух* соседних вершин ломаной. Для большего числа соседних вершин доказательство аналогично, только более громоздко.

Окончательно, мы доказали, что ломаная линия L_* представляет собой **локальный** минимум. □



(а) Маршрут пересекает контур (б) Касательная отделяет контур

Рис. 2.4. Достаточные условия глобального минимума

2.3.2. Глобальный минимум

Перейдём к *достаточному* условию того, что ломаная L_* является глобальным минимумом.

Пусть $M_i \in C_i$ – вершина ломаной L_*. Соседние вершины M_{i-1} и M_{i+1} находятся вне C_i, поскольку по условию задачи мы исключили вложенные контуры По построению L_*: $\forall M_i' \in C_i : |M_{i-1}M_i|+|M_iM_{i+1}| \leqslant |M_{i-1}M_i'|+|M_i'M_{i+1}|$.

Условие 2.1. Пусть выполняется **одно** из следующих условий (см. рис. 2.4):

1. Сегмент $M_{i-1}M_{i+1}$ пересекает контур C_i, то есть $M_i \in M_{i-1}M_{i+1}$, рис. 2.4а
2. Касательная в точке M_i к эллипсу с фокусами M_{i-1} и M_{i+1} и проходящему через M_i, разделяет эллипс и контур C_i, рис. 2.4б

Утверждение 2.2. *Если условие 2.1 выполняется для (всех контуров) L_*, то при сдвиге нескольких соседних вершин ломаной L_* в рамках содержащих их контуров, полная длина ломаной не уменьшается, то есть ломаная L_* представляет собой глобальный минимум.*

Доказательство. Используя те же обозначения, рассмотрим четыре соседних вершины $M_{i-1}, M_i, M_{i+1}, M_{i+2} \in L_*$. $M_i \in C_i, M_{i+1} \in C_{i+1}$.

Предположим, что

$$\exists M_i' \in C_i, \exists M_{i+1}' \in C_{i+1} : |M_{i-1}M_i'M_{i+1}'M_{i+2}| < |M_{i-1}M_iM_{i+1}M_{i+2}|$$

Снова, $M_i' \neq M_i, M_{i+1}' \neq M_{i+1}$.

Обозначим $M_i(s) = M_i + s \cdot \overrightarrow{M_i M_i'}$, $M_{i+1}(t) = M_{i+1} + t \cdot \overrightarrow{M_{i+1} M_{i+1}'}$ ($s, t \in [0, 1]$), $f(s, t) = |M_{i-1} M_i(s) M_{i+1}(t) M_{i+2}|$.

Но теперь условие 2.1 гарантирует, что $f(s, 0) \geqslant f(0, 0)$ для $s \in [0, 1]$, то есть снова

$$\frac{\partial f(s,t)}{\partial s}\Big|_{(0,0)} \geqslant 0, \frac{\partial f(s,t)}{\partial t}\Big|_{(0,0)} \geqslant 0 \tag{2.5}$$

Поэтому остальная часть предыдущего доказательства повторяется отсюда без изменений. □

Предположим, что кроме L_* существует другая ломаная, также являющаяся глобальным минимумом. Тогда из доказательства следует, что они представляют собой одну и ту же линию (как множество точек) и отличаются только положением вершин, то есть тем, какие именно пересечения с контурами выбраны в качестве вершин ломаных линий.

Условие 2.1 легко проверяется программно, однако его можно ещё упростить с тем, чтобы в большинстве практических случаев его можно было проверить просто визуально, буквально «на глаз».

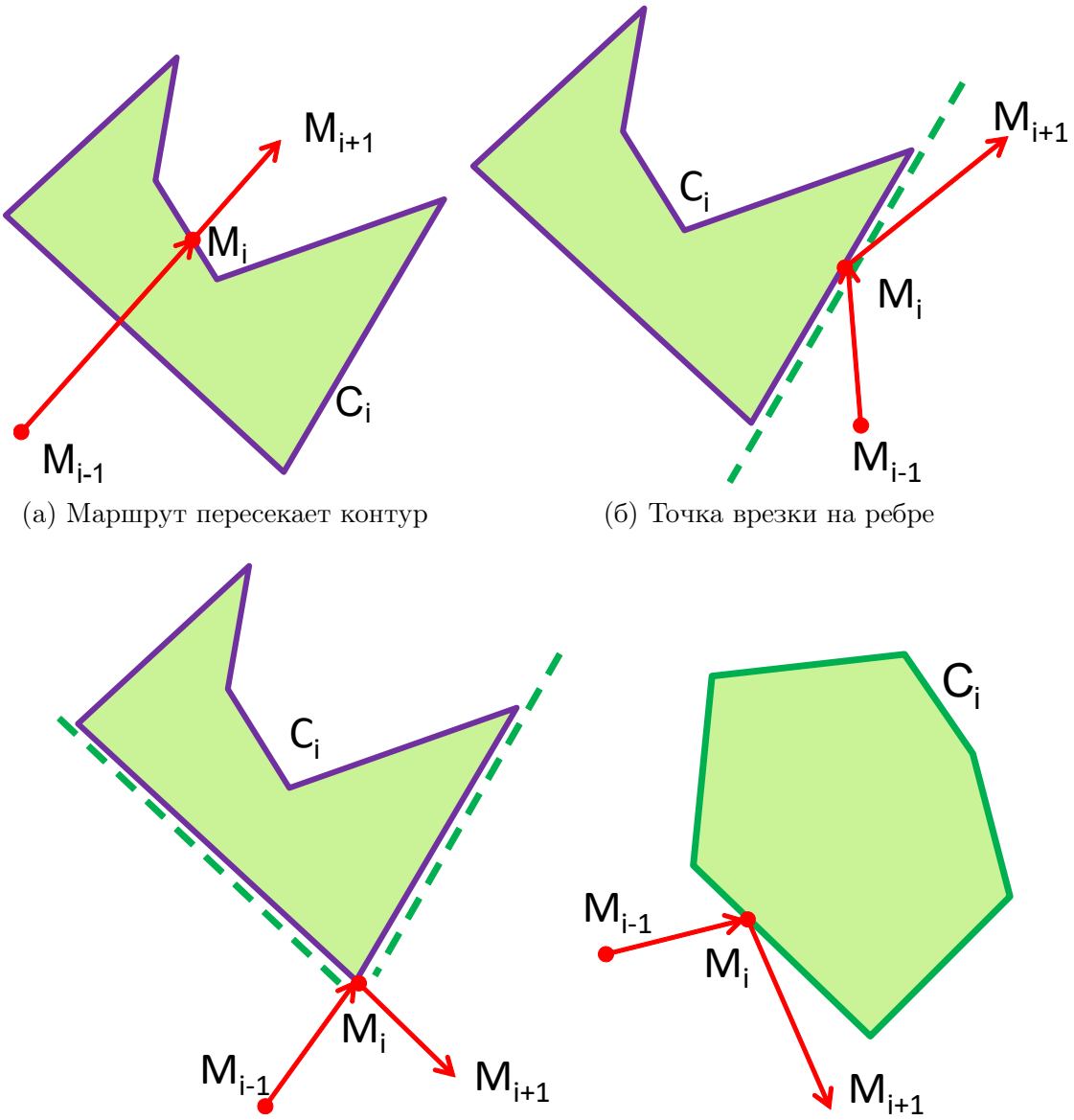
Условие 2.2. Выполняется **любое** из условий (см. рис. 2.5):

1. Сегмент $M_{i-1} M_{i+1}$ пересекает контур C_i: $M_i \in M_{i-1} M_{i+1}$, рис. 2.5а

2. Если вершина M_i является внутренней точкой одного из отрезков контура C_i и при этом весь контур расположен по одну сторону линии, проходящей через этот отрезок (это и есть касательная, которую использует условие 2.1; иначе существовала бы лучшая вершина $M_i' \in C_i$), рис. 2.5б

3. Если вершина M_i является также вершиной контура C_i (принадлежит сразу двум его отрезкам) и при этом весь контур находится внутри угла, образованного лучами, идущими из вершины M_i вдоль этих двух отрезков, рис. 2.5в

4. Если контур C_i ограничивает собой выпуклый многоугольник $\widetilde{C}_i$, рис. 2.5г



(а) Маршрут пересекает контур (б) Точка врезки на ребре

(в) Врезка в вершину контура (г) Выпуклый контур

Рис. 2.5. Ослабленное условие глобального минимума

2.4. Численные эксперименты

Для оценки качества решений, получаемых описанным эвристическим алгоритмом, использовались несколько раскройных планов, содержащих реальные детали. В качестве базы сравнения использовался алгоритм [1], решающий задачу GTSP и дающий точное решение для количества контуров $N \leqslant 33$.

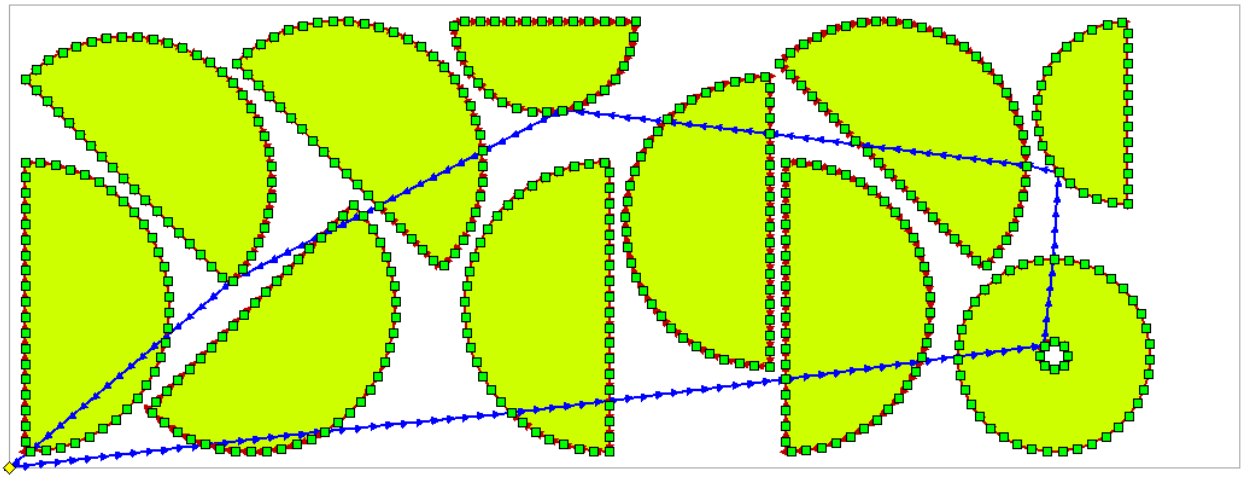
Результаты экспериментов сведены в Табл. 2.1, полученные решения приведены на рис. 2.6, рис. 2.7, рис. 2.8 и рис. 2.9.

Видно, что оба алгоритма дают практически идентичные маршруты резки. Основное отличие вызвано необходимостью дискретизации контуров в ходе

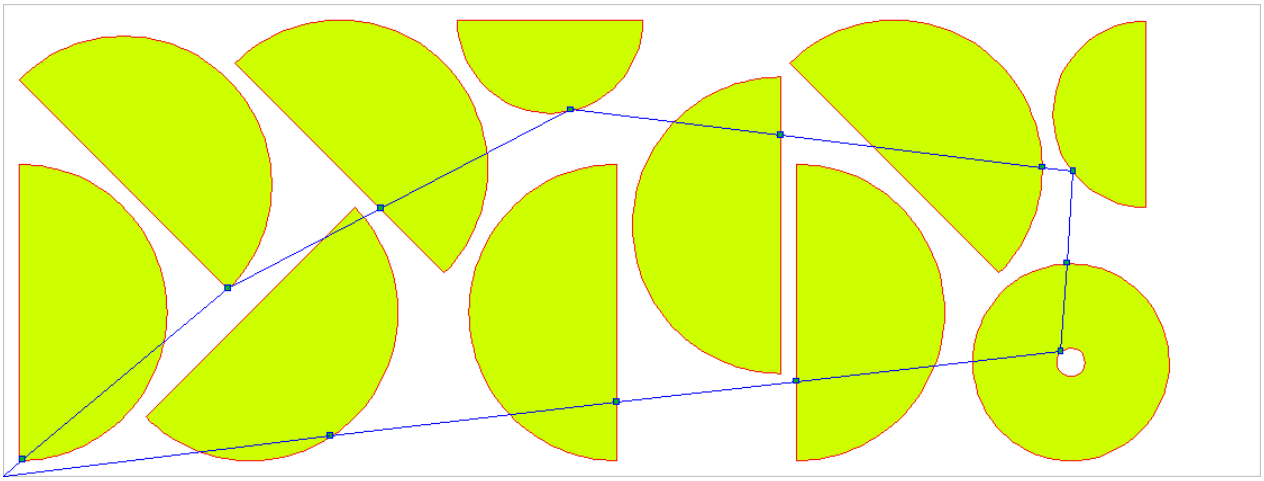
Таблица 2.1

Сравнение качества решений задач CCP и GTSP

Задание	№ 229	№ 464	№ 3211	№ 20205
Кол-во деталей	11	14	17	115
Кол-во контуров	12	21	22	198
Общий периметр, м	24.609	21.717	25.051	143.467
Кол-во точек GTSP	491	429	493	3917
$\mathcal{L}_{GTSP}$, м	7.729	4.743	4.557	26.098
$\mathcal{L}_{CCP}$, м	7.727	4.706	4.536	25.987
См.	Рис. 2.6	Рис. 2.7	Рис. 2.8	Рис. 2.9

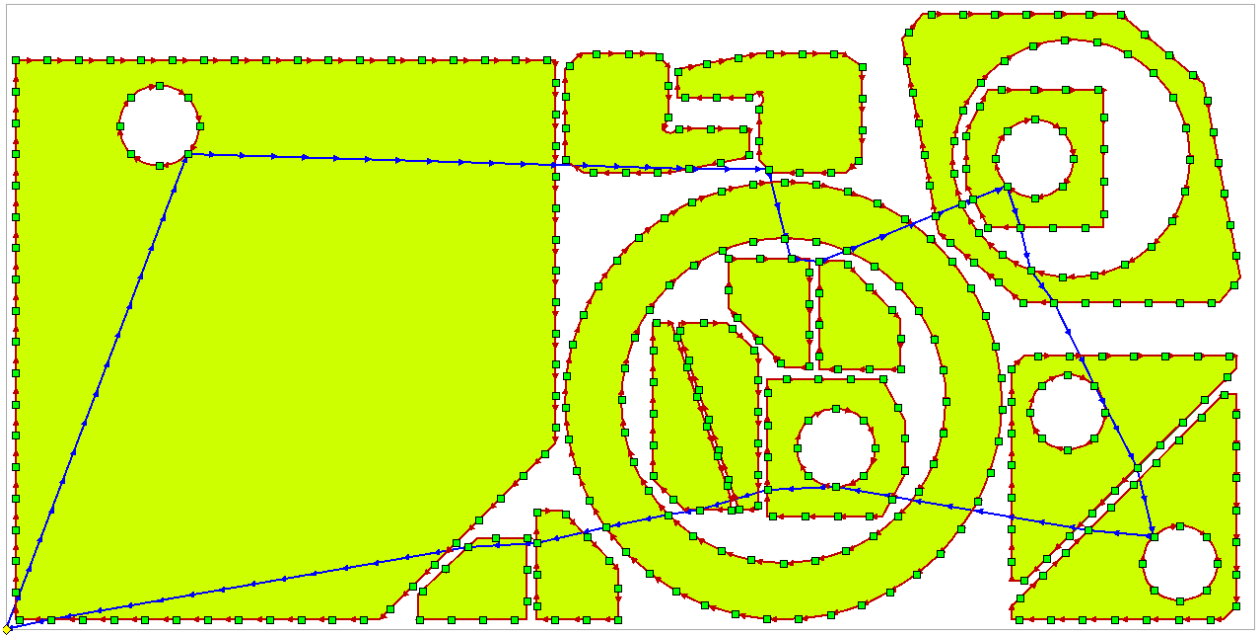


(а) Точное решение задачи GTSP

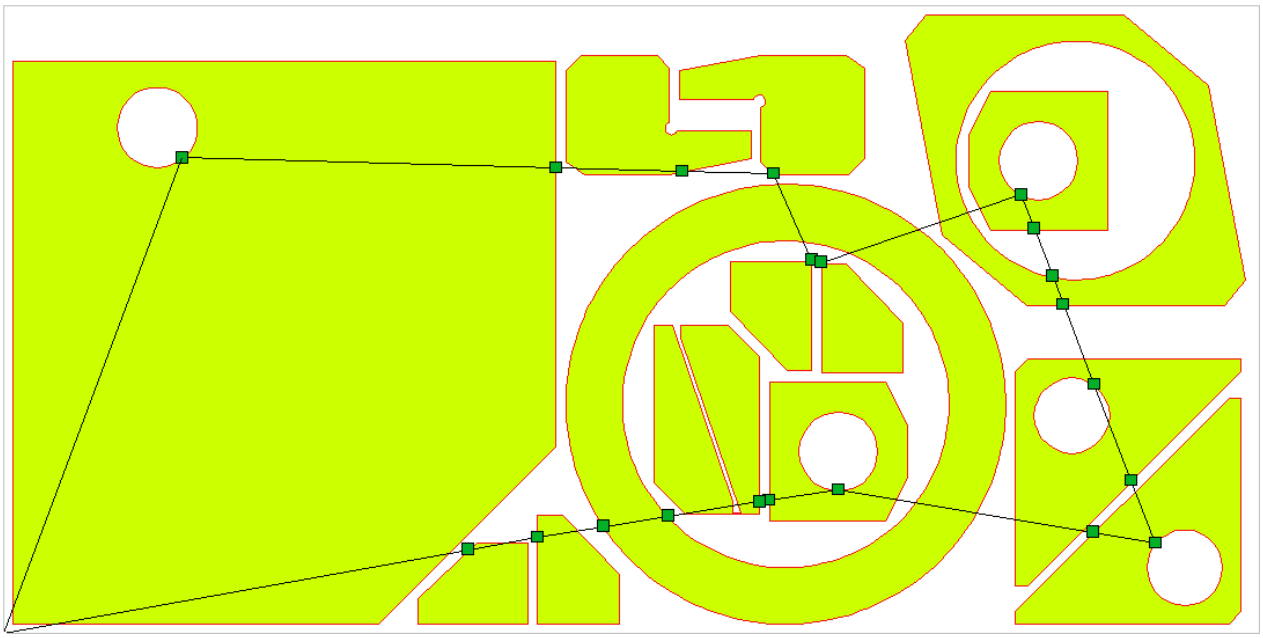


(б) Решение задачи CCP

Рис. 2.6. Решения задач резки для задания № 229

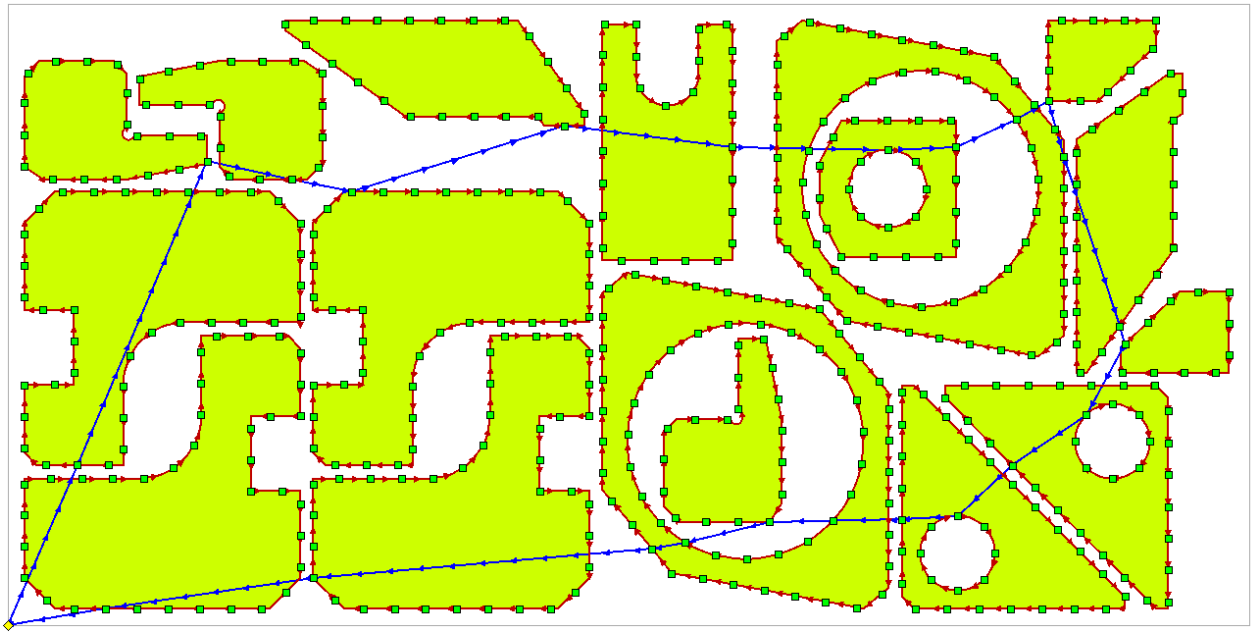


(а) Точное решение задачи GTSP

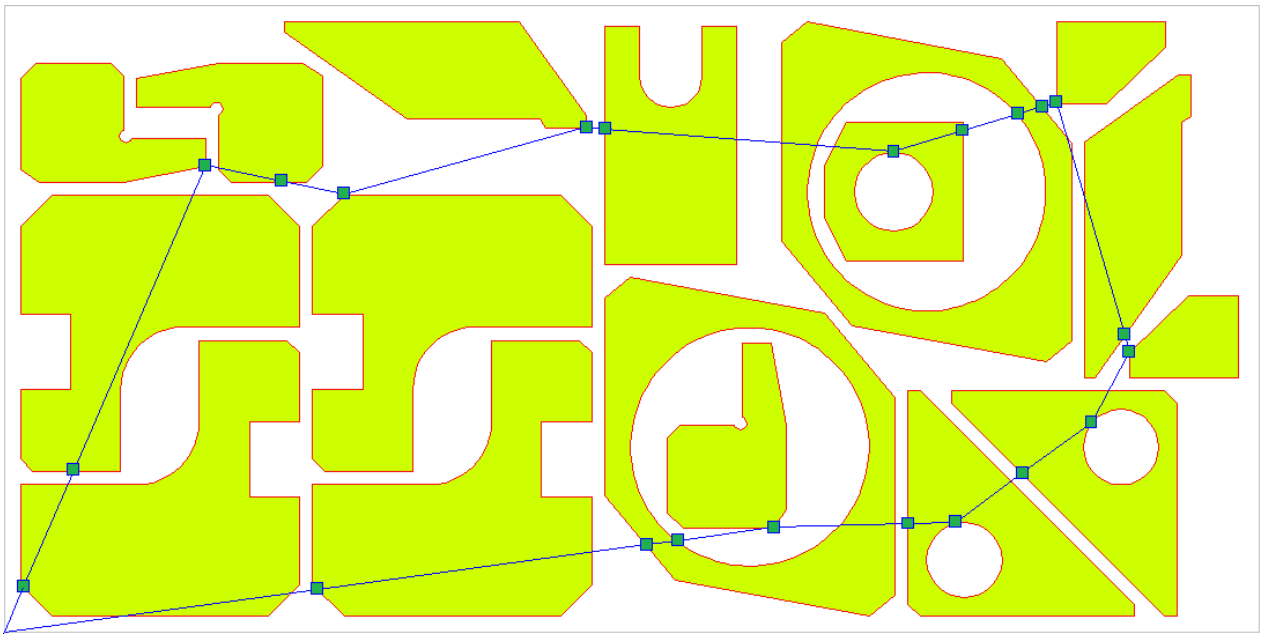


(б) Решение задачи CCP

Рис. 2.7. Решения задач резки для задания № 464



(а) Точное решение задачи GTSP



(б) Решение задачи CCP

Рис. 2.8. Решения задач резки для задания № 3211

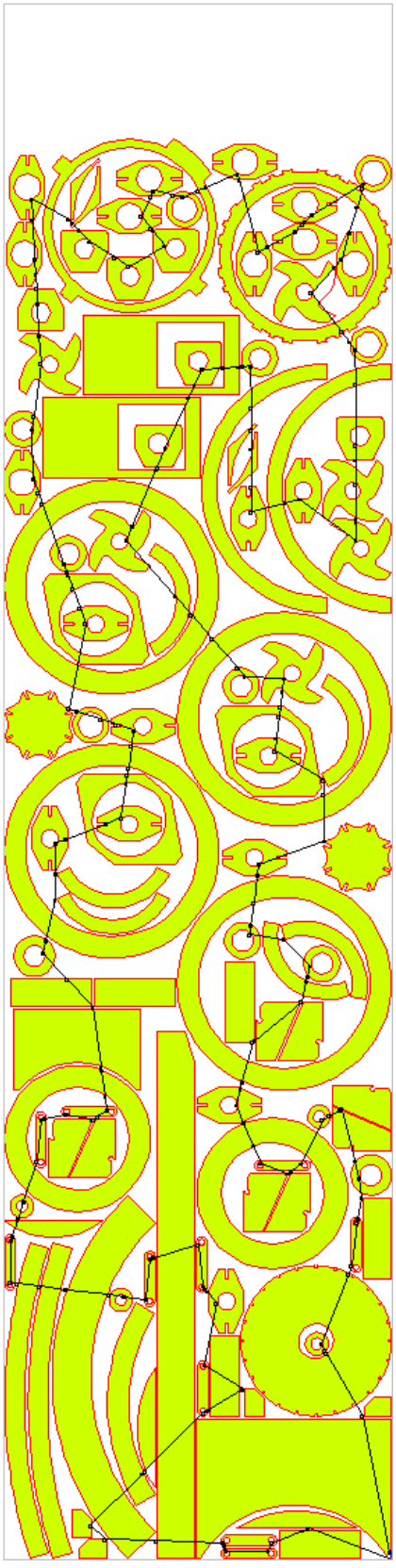


Рис. 2.9. Пример решения задачи ССР большого размера, задание № 20205

сведения задачи непрерывной резки к GTSP. Характерной особенностью решения задачи непрерывной резки является то, что оно содержит много прямолинейных сегментов, разделяемых точками врезки, но фактически лежащих на одной прямой. Аналогичные сегменты решения задачи GTSP имеют небольшие изломы, поскольку возможные координаты точек резки фиксированы заранее и не могут попасть на одну прямую. Поэтому общая длина холостого хода в общем случае получается чуть больше, чем для «честного» решения задачи непрерывной резки, что и отображено в Табл. 2.1.

Интересно, что Условие 2.1 на стр. 26 соблюдено для всех контуров на рис. 2.7б, то есть изображённое там решение действительно представляет собой глобальный минимум (длины холостого хода). В то же время, более простое Условие 2.2 на стр. 27 соблюдено для *почти* всех контуров, кроме одного (невыпуклой детали сверху по центру). Это довольно редкая с практической точки зрения ситуация, тем не менее, она показывает, что две формулировки достаточного условия глобальности минимума не эквивалентны.

2.5. Обобщение на задачи сегментной резки SCCP / GSCCP

Описанный выше алгоритм разрабатывался и применяется для решения задачи непрерывной резки CCP, однако последняя представляет собой весьма частный случай самой общей формулировки задачи оптимальной маршрутизации режущего инструмента, на данный момент это задача прерывистой резки (Intermittent Cutting Problem, ICP). Она всё ещё слабо исследована и представляет существенный научный интерес как с теоретической точки зрения, так и в смысле практического использования.

Тогда как задача непрерывной резки возникает фактически при использовании так называемой *стандартной техники резки*, существуют и более сложные, прежде всего *мульти-сегментная* и *мульти-контурная* техники резки. Первая характеризуется тем, что контур детали вырезается в несколько проходов, с использованием нескольких точек врезки. Вторая же наоборот, вырезает несколько контуров деталей за один проход, как это показано на рис. 2.10.

Для включения этих техник в исследование, полезно расширить понятие контура детали и ввести новый термин [8]:

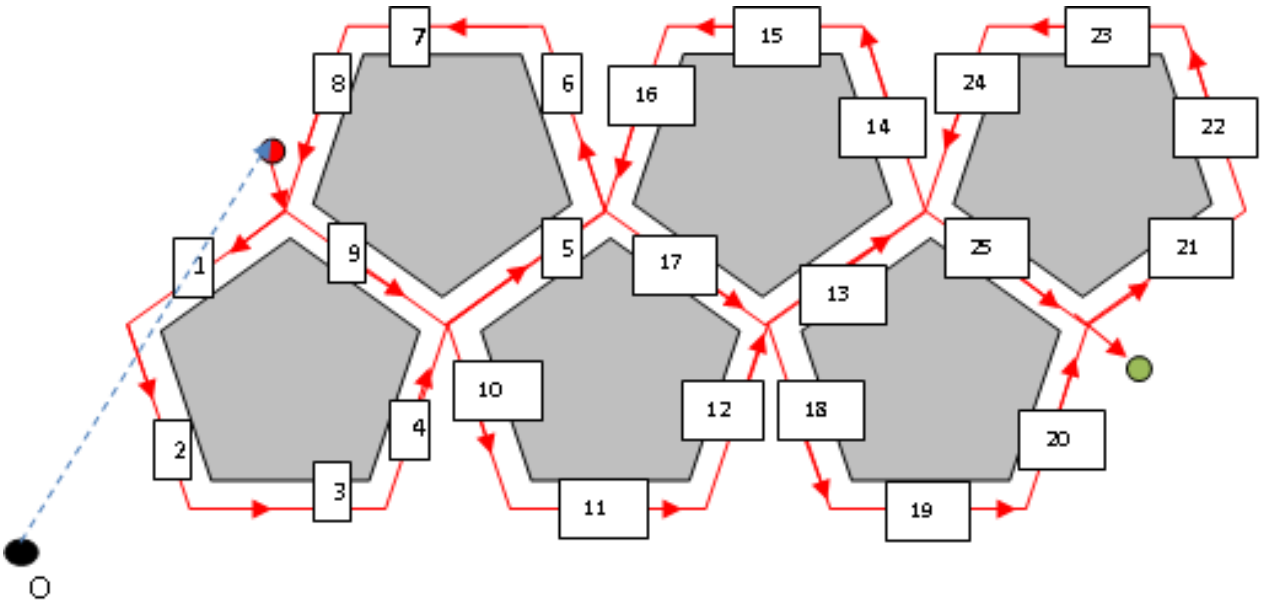


Рис. 2.10. Пример составного сегмента резки, содержащего 6 контуров (деталей)

Сегмент резки $S = \overrightarrow{MM^*}$ — это часть траектории режущего инструмента от точки врезки M до соответствующей точки выключения инструмента M^*.

Сегмент содержит в себе вход в контур (*lead-in*) и выход из него (*lead-out*), а также часть или целый контур детали или нескольких деталей. В случае стандартной техники резки имеется однозначное соответствие между контурами деталей и сегментами резки, но в общем случае оно отсутствует. Фактически, пример мульти-контурной резки на рис. 2.10 также может представлять собой один единственный сегмент резки в составе некоторой большой задачи.

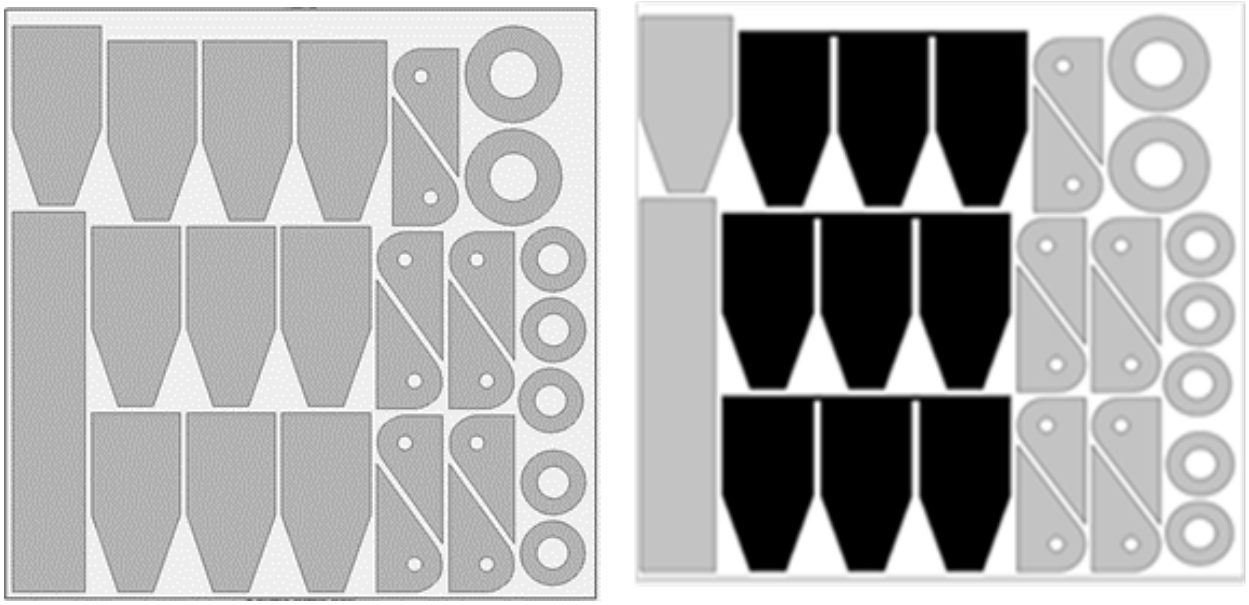
Ввиду того, что сегмент резки по определению содержит в себе направление резки (от точки врезки M до точки выключения инструмента M^*), нам потребуется ещё более общее понятие:

Базовый сегмент резки B^S — это часть сегмента резки $S = \overrightarrow{MM^*}$ без участков входа и выхода (lead-in и lead-out).

Базовый сегмент содержит только геометрию (части) контуров, подлежащих резке, и не содержит информации о направлении резки.

При помощи понятия базового сегмента формулируется обобщение задачи непрерывной резки:

Задача непрерывной сегментной резки (Segment Continuous Cutting Problem, $SCCP$) — это задача резки для фиксированного набора базовых сегментов резки: $SCCP = \{B^{S_i}\}$.



(а) Стандартная резка, 45 сегментов (б) Мульти-контурная резка, 39 сегментов

Рис. 2.11. Ансамбль задач сегментной резки

Описанный в данной главе алгоритм, разработанный для решения задачи непрерывной резки, естественным образом обобщается на решение задачи непрерывной сегментной резки.

Далее, для произвольного раскройного плана (то есть фиксированного расположения деталей A_i и контуров C_i на листе $\mathcal{B}$), в общем случае можно сгенерировать целый *ансамбль* наборов базовых сегментов B^{S_i}, отвечающих заданным контурам деталей C_i. Например, для раскроя на рис. 2.11а можно построить задачу $SCCP$ меньшего размера за счёт объединения некоторых контуров в базовые сегменты, как показано на рис. 2.11б. Это наблюдение приводит нас к ещё более общей задаче резки:

Обобщённая задача непрерывной сегментной резки (Generalized Segment Continuous Cutting Problem, $GSCCP$) — это ансамбль из нескольких задач $SCCP$, полученных из одного раскройного плана: $GSCCP = \{SCCP_i\}$.

Новые классы $SCCP$ и $GSCCP$ значительно расширяют существующую классификацию задач резки для машин термической резки с ЧПУ. Фактически они представляют собой подклассы наиболее общей задачи ICP, состоящие из конечного набора базовых сегментов резки.

$$CCP \subset SCCP \subset GSCCP \subset ICP$$

2.5.1. Общая схема решения задачи GSCCP

Считая заданным ансамбль $\{SCCP_i\}$ наборов базовых сегментов $SCCP_i = \left\{B^{S_j}\right\}$, $i \in \overline{1,T}, j \in \overline{1,K_i}$, будем решать задачу $GSCCP$ следующим образом:

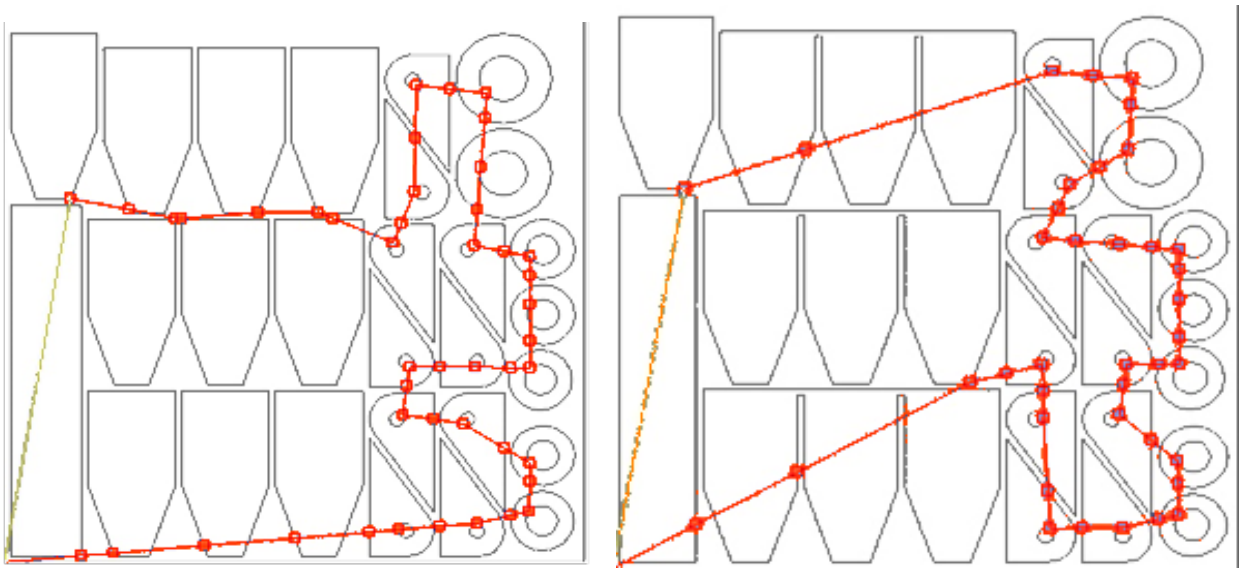
- Каждая задача $SCCP_i$ решается независимо одним из существующих алгоритмов, например:
  - Описанный выше эвристический алгоритм решения задачи непрерывной резки, см. Главу 2.
  - Алгоритм ветвей и границ для обобщённой задачи коммивояжера с ограничениями предшествования, см. Главу 3.
  - Алгоритм на основе динамического программирования, дающий точное решение для задач ограниченного размера, см. [1].
  - Итеративный жадный эвристический алгоритм, см. [6]
  - ...

  Для дискретных алгоритмов предварительно проводится дискретизация контуров и построение конечного множества допустимых точек врезки.
- Выбирается лучшее решение, минимизирующее целевую функцию (2.2).

Пример решения задачи $GSCCP$, представленной на рис. 2.11, приведён на рис. 2.12. Использован алгоритм решения задачи непрерывной резки (без применения дискретизации контуров деталей). Видно, что для разных постановок задач $SCCP$ действительно получаются разные маршруты движения режущего инструмента. Более того, на практике различие оказывается ещё более значительным, поскольку получаемые решения отличаются также количеством точек врезки, а эта операция обычно вносит существенные расходы, как в смысле стоимости резки, так и затрачиваемого времени.

2.6. Выводы по Главе 2

1. Разработана и реализована оригинальная эвристика поиска оптимальных положений точек врезки на контурах деталей.



(а) Стандартная резка (б) Мульти-контурная резка

Рис. 2.12. Решение задачи GSCCP на рис. 2.11

2. Эта эвристика может сочетаться с алгоритмами дискретной оптимизации для получения полного алгоритма решения задачи непрерывной резки CCP

3. Полученные решения оказываются вполне сравнимы с решениями, получаемыми более традиционными методами, путем сведения задачи CCP к задаче GTSP за счёт дискретизации контуров деталей. Более того, длина холостого хода для решений, даваемых разработанным алгоритмом, систематически оказывается чуть лучше, чем для ранее использовавшихся алгоритмов.

4. Описанный алгоритм может также применяться для решения задач более высокого класса — SCCP и GSCCP (непрерывной сегментной резки), что открывает дорогу к исследованиям в области задачи прерывистой резки (ICP).

5. Перспективными представляются также следующие направления исследований:

   – Использование разработанной эвристики в сочетании с другими алгоритмами дискретной оптимизации

   – Учёт других технологических ограничений термической резки, кроме ограничения предшествования

Глава 3

Точный алгоритм решения обобщенной задачи коммивояжера с ограничениями предшествования PCGTSP

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam.

Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad

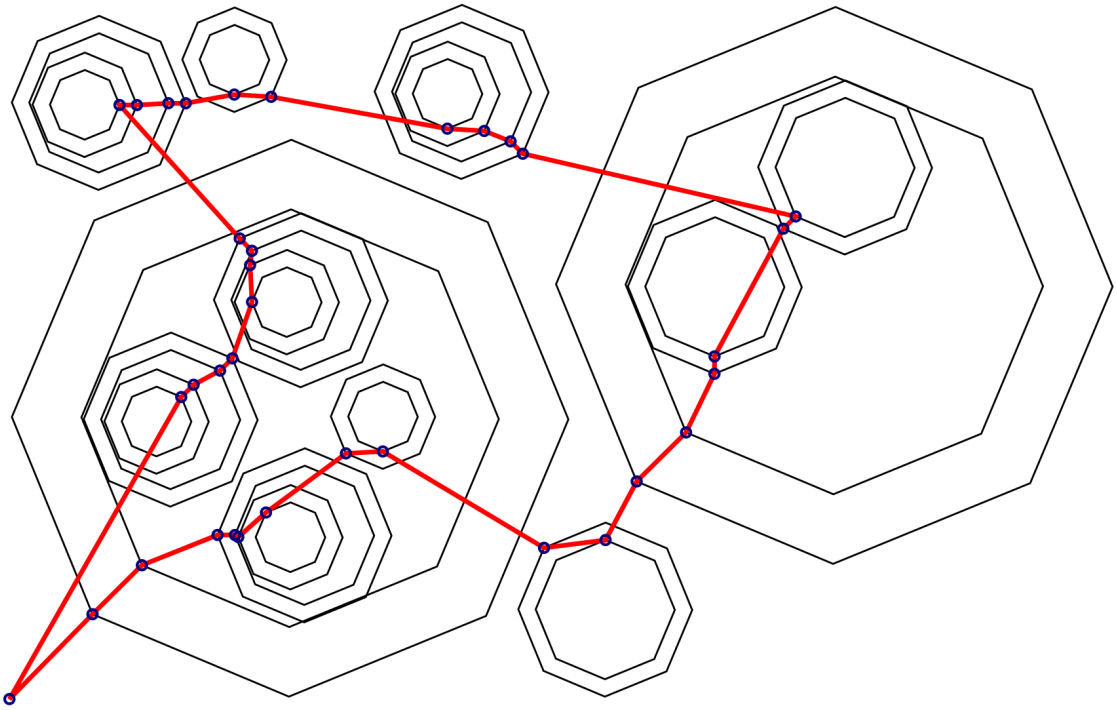


Рис. 3.1. Пример решения задачи PCGTSP, полученного эвристикой PCGLNS

litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

3.1. Численные эксперименты

...

Полученные результаты эксперимента представлены в табл. 3.1, 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.

3.2. Выводы по Главе 3

1.

Таблица 3.1

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**

Глава 4

Название в разработке

...

Пример файла геометрии для простейшей раскройной карты приведён в Листинге 4.1.

```
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 4.1. JSON-файл с геометрией простейшей раскройной карты

Листинг 4.2 показывает пример простейшего SVG-файла, сгенерированного для раскройной карты, представленной на Листинге 4.1.

```
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 4.2. SVG-файл для визуализации раскроя из Листинга 4.1

Один из вариантов оформления SVG-файла из Листинга 4.2 приведён на рис. 4.1. Пользовательский интерфейс (масштабирование и прокрутка) обеспечивается при помощи подключения библиотеки с открытым кодом svg-pan-zoom [5].

Для визуализации решения задачи PCGTSP (на основе комбинации информации, полученной из нескольких источников), была разработана специализированная утилита [11], первоначально в форме утилиты командной строки но позднее преобразованная для удобства использования в Single Page Application (SPA). Пример созданного ею изображения приведён на рис. 3.1, стр. 40.

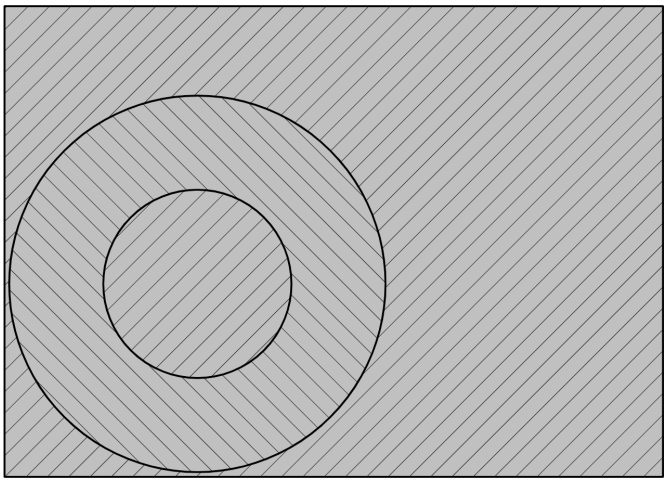


Рис. 4.1. Визуализация раскроя из Листинга 4.1

4.1. Выводы по Главе 4

1.

Заключение

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

1. Разработана схема эффективного учёта ограничений предшествования для задачи непрерывной резки.

2. Разработана основанная на геометрических соображениях эвристика размещения точек врезки на плоских контурах.

3. Доказано, что данная эвристика доставляет локальный минимум длины холостого хода и сформулированы два набора достаточных условий того, что полученное решение является глобальным минимумом.

4. Эти две схемы могут соединяться с различными алгоритмами дискретной оптимизации, тем самым получается полный алгоритм решения задачи непрерывной резки. В данной работе для дискретной оптимизации используется метод переменных окрестностей.

5. Полученный алгоритм даёт решения хорошего качества за разумное время. В случае, когда известно точное решение соответствующей обобщённой задачи коммивояжера, они визуально совпадают, но длина холостого хода для решения задачи непрерывной резки оказывается короче за счёт отсутствия дискретизации.

6. Продемонстрировано, что полученный алгоритм может использоваться для решения задач сегментной непрерывной резки (SCCP) и обобщённой сегментной непрерывной резки (GSCCP), тем самым открывая подход к решению общей задачи прерывистой резки (ICP)

7. Разработан алгоритм Branch-and-Bound для точного решения обобщённой задачи коммивояжёра с ограничениями предшествования, рассчитывающий нижнюю границу

8. Алгоритм способен находить точные решения для задач большего размера, чем другие алгоритмы. В проведённых экспериментах было найдено решение для задачи со 151 кластером.

9. Данный алгоритм также решает важную задачу оценки качества решений, полученных другими алгоритмами, даже в случае, когда нахождение точного решения непрактично.

10. Алгоритм может быть реализован в классической схеме, а также в парадигме динамического программирования. В последнем случае он естественным образом допускает распараллеливание и демонстрирует лучшую производительность.

11. Разработаны форматы данных для обмена геометрической и маршрутной информацией и визуализации для использования в CAD/CAM-системах, а также алгоритмы преобразования на примере САПР «Сириус». Аналогичные конвертеры могут легко разрабатываться для других САПР.

12. 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. Учёт дополнительных ограничений в алгоритме решения задачи непрерывной резки, в частности того, что точка врезки располагается не на самом контуре, а на некотором расстоянии от него, а также того, что существуют зоны, где не могут размещаться зоны врезки, а также других технологических ограничений, порождаемых современным оборудованием термической резки с ЧПУ

2. Использование других алгоритмов дискретной оптимизации в задаче непрерывной резки и оценка производительности и качества получаемых алгоритмов.

3. Разработка других методов получения оценок для частичных подзадач GTSP с целью повышения нижней границы; например, за счёт более точного учёта расстояний между точками, а не только между кластерами.

Список литературы

1. *Chentsov 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.
2. *Dror M.* Touring a sequence of polygons / M. Dror, A. Efrat, A. Lubiw, J. S. Mitchell // Proceedings of the thirty-fifth annual ACM symposium on Theory of computing. — ACM. 2003. — С. 473—482.
3. *Hansen P.* Variable neighbourhood search: methods and applications / P. Hansen, N. Mladenović, J. A. Moreno Pérez // Annals of Operations Research. — 2010. — Т. 175, № 1. — С. 367—407.
4. Introducing JSON. — URL: `https://www.json.org/`.
5. JavaScript library that enables panning and zooming of an SVG in an HTML document, with mouse events or custom JavaScript hooks. — URL: `https://github.com/bumbu/svg-pan-zoom`.
6. *Petunin A. A.* About routing in the sheet cutting / A. A. Petunin, A. G. Chentsov, P. A. Chentsov // Bulletin of the South Ural State University, Series: Mathematical Modelling, Programming and Computer Software. — 2017. — Т. 10, № 3. — С. 25—39.
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.
8. *Petunin A.* General Model of Tool Path Problem for the CNC Sheet Cutting Machines / A. Petunin // IFAC-PapersOnLine. — 2019. — Т. 52, № 13. — С. 2662—2667.

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.

10. *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.

11. *Уколов С. С.* Визуализация решения задачи PCGTSP / С. С. Уколов. — URL: `https://ukoloff.github.io/j2pcgtsp/`.

12. *Уколов С. С.* JSON-схемы файлов, используемых в САПР «Сириус» / С. С. Уколов, П. А. Ченцов. — URL: `https://ukoloff.github.io/dbs.js/json-schema/`.

Список иллюстраций

Список таблиц

Приложение А

JSON-схемы

В ходе диссертационной работы был разработан ряд форматов обмена информацией [12], все они представляют собой современный *стандарт де-факто* JSON [4] и их описание оформлено в виде JSON-схем [9].

А.1. Сведения о геометрии деталей и раскроя

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://ukoloff.github.io/dbs.js/json-schema/dbs.json",
  "$ref": "#/$defs/dbs",
  "$defs": {
    "point": {
      "title": "Point on the plain",
      "description": "X, Y and bulge (i.e. tan(angle / 4) for an arc)",
      "type": "array",
      "items": {
        "type": "number"
      },
      "minItems": 2,
      "maxItems": 3,
      "default": 0.0
    },
    "path": {
      "title": "2D contour",
      "description": "Polyline consisting of line segments and arcs",
      "type": "array",
      "items": {
        "$ref": "#/$defs/point"
      },
```

```
24        "minItems": 2
25      },
26      "part": {
27        "title": "2D part",
28        "description": "Collection of plain contours",
29        "properties": {
30          "partid": {
31            "type": "string"
32          },
33          "paths": {
34            "type": "array",
35            "items": {
36              "$ref": "#/$defs/path"
37            }
38          },
39          "area": {
40            "type": "number"
41          },
42          "perimeter": {
43            "type": "number"
44          }
45        },
46        "required": [
47          "partid",
48          "paths"
49        ]
50      },
51      "dbs": {
52        "title": "DBS file",
53        "description": "Collection of 2D parts",
54        "type": "array",
55        "items": {
56          "$ref": "#/$defs/part"
57        },
58        "minItems": 1
59      }
60    }
61  }
```

Листинг А.1. Файл геометрии деталей и раскроя

А.2. Задание на резку

```
1  {
```

```
2   "$schema": "https://json-schema.org/draft/2020-12/schema",
3   "$id": "https://ukoloff.github.io/dbs.js/json-schema/rm-task.json",
4   "$ref": "#/$defs/task",
5   "$comment": "Generated by https://app.quicktype.io/",
6   "$defs": {
7     "point": {
8       "title": "Pierce poitnt",
9       "description": "Feasible pierse point to cut-in",
10      "type": "object",
11      "additionalProperties": false,
12      "properties": {
13        "Index": {
14          "type": "integer"
15        },
16        "ZeroBasedIndex": {
17          "type": "integer"
18        },
19        "GlobalIndex": {
20          "type": "integer"
21        },
22        "ZeroBasedGlobalIndex": {
23          "type": "integer"
24        },
25        "X": {
26          "type": "number"
27        },
28        "Y": {
29          "type": "number"
30        },
31        "UseCircuitAndFinishCutPoint": {
32          "type": "boolean"
33        }
34      },
35      "required": [
36        "GlobalIndex",
37        "Index",
38        "UseCircuitAndFinishCutPoint",
39        "X",
40        "Y",
41        "ZeroBasedGlobalIndex",
42        "ZeroBasedIndex"
43      ]
44    },
45    "contour": {
```

```
46        "title": "2D contour",
47        "description": "Sequence of 2D points with nested contours allowed",
48        "type": "object",
49        "additionalProperties": false,
50        "properties": {
51          "Index": {
52            "type": "integer"
53          },
54          "ZeroBasedIndex": {
55            "type": "integer"
56          },
57          "Points": {
58            "type": "array",
59            "items": {
60              "$ref": "#/$defs/point"
61            }
62          },
63          "NestedContours": {
64            "$ref": "#/$defs/contours"
65          }
66        },
67        "required": [
68          "Index",
69          "Points",
70          "ZeroBasedIndex"
71        ]
72      },
73      "contours": {
74        "title": "Several contours",
75        "type": "array",
76        "items": {
77          "$ref": "#/$defs/contour"
78        },
79        "minItems": 1
80      },
81      "params": {
82        "title": "Configuration for RouteManager",
83        "type": "object",
84        "additionalProperties": false,
85        "properties": {
86          "StartX": {
87            "type": "number"
88          },
89          "StartY": {
```

```
90            "type": "number"
91          },
92          "TerminalMotion": {
93            "type": "boolean"
94          },
95          "FinishX": {
96            "type": "number"
97          },
98          "FinishY": {
99            "type": "number"
100         },
101         "SheetMinX": {
102           "type": "number"
103         },
104         "SheetMaxX": {
105           "type": "number"
106         },
107         "SheetMinY": {
108           "type": "number"
109         },
110         "SheetMaxY": {
111           "type": "number"
112         },
113         "ToolIdlingSpeed": {
114           "type": "number"
115         },
116         "ToolCutSpeed": {
117           "type": "number"
118         },
119         "PiercingDuration": {
120           "type": "number"
121         }
122       },
123       "required": [
124         "FinishX",
125         "FinishY",
126         "PiercingDuration",
127         "SheetMaxX",
128         "SheetMaxY",
129         "SheetMinX",
130         "SheetMinY",
131         "StartX",
132         "StartY",
133         "TerminalMotion",
```

```
134         "ToolCutSpeed",
135         "ToolIdlingSpeed"
136       ]
137     },
138     "task": {
139       "title": "Task for RouteManager",
140       "description": "JSON version of RDF file, containing all data for running R
141       "type": "object",
142       "additionalProperties": false,
143       "properties": {
144         "TaskData": {
145           "$ref": "#/$defs/params"
146         },
147         "Contours": {
148           "$ref": "#/$defs/contours"
149         }
150       },
151       "required": [
152         "Contours",
153         "TaskData"
154       ]
155     }
156   }
157 }
```

Листинг А.2. Задание на резку

А.3. Результат резки

```
1  {
2    "$schema": "https://json-schema.org/draft/2020-12/schema",
3    "$id": "https://ukoloff.github.io/dbs.js/json-schema/rm-result.json",
4    "$ref": "#/$defs/result",
5    "$comment": "Generated by https://app.quicktype.io/",
6    "$defs": {
7      "mode": {
8        "title": "Mode of tool motion",
9        "type": "string",
10       "enum": [
11         "air",
12         "cut"
13       ]
14     },
15     "point": {
```

```
16        "title": "2D point",
17        "description": "X, Y and angle (for arc)",
18        "type": "array",
19        "items": {
20          "type": "number"
21        },
22        "minItems": 3,
23        "maxItems": 3
24      },
25      "points": {
26        "title": "2D contour",
27        "description": "Sequence of 2D points",
28        "type": "array",
29        "items": {
30          "$ref": "#/$defs/point"
31        },
32        "minItems": 2
33      },
34      "path": {
35        "title": "Part contour",
36        "description": "Single 2D contour of initial Task",
37        "type": "object",
38        "additionalProperties": false,
39        "properties": {
40          "part.id": {
41            "type": "integer"
42          },
43          "id": {
44            "type": "integer"
45          },
46          "points": {
47            "$ref": "#/$defs/points"
48          }
49        },
50        "required": [
51          "id",
52          "part.id",
53          "points"
54        ]
55      },
56      "job": {
57        "title": "Task solved",
58        "description": "Some information on the Task solved by RouteManager",
59        "type": "object",
```

```
60        "additionalProperties": false,
61        "properties": {
62          "paths.count": {
63            "type": "integer"
64          },
65          "parts.count": {
66            "type": "integer"
67          },
68          "total.points.count": {
69            "type": "integer"
70          },
71          "total.address.pairs.count": {
72            "type": "integer"
73          },
74          "list": {
75            "$ref": "#/$defs/points"
76          },
77          "paths": {
78            "type": "array",
79            "items": {
80              "$ref": "#/$defs/path"
81            }
82          }
83        },
84        "required": [
85          "list",
86          "parts.count",
87          "paths",
88          "paths.count",
89          "total.address.pairs.count",
90          "total.points.count"
91        ]
92      },
93      "toolpath": {
94        "title": "Tool path",
95        "description": "Sequence of points visited by tool during cutting process",
96        "type": "object",
97        "additionalProperties": false,
98        "properties": {
99          "point.id": {
100           "type": "integer"
101         },
102         "mode": {
103           "$ref": "#/$defs/mode"
```

```
104         },
105         "point": {
106           "$ref": "#/$defs/point"
107         },
108         "part.id": {
109           "type": "integer"
110         },
111         "id": {
112           "type": "integer"
113         }
114       },
115       "required": [
116         "mode",
117         "point",
118         "point.id"
119       ]
120     },
121     "results": {
122       "type": "object",
123       "additionalProperties": false,
124       "properties": {
125         "result": {
126           "type": "number"
127         },
128         "route.length.no.contours": {
129           "type": "number"
130         },
131         "route.length": {
132           "type": "number"
133         },
134         "count.time": {
135           "type": "string",
136           "format": "time"
137         },
138         "toolpath": {
139           "type": "array",
140           "items": {
141             "$ref": "#/$defs/toolpath"
142           }
143         }
144       },
145       "required": [
146         "count.time",
147         "result",
```

```
148           "route.length",
149           "route.length.no.contours",
150           "toolpath"
151         ],
152         "title": "Results"
153       },
154       "result": {
155         "title": "Solution obtained by RouteManager",
156         "type": "object",
157         "additionalProperties": false,
158         "properties": {
159           "job": {
160             "$ref": "#/$defs/job"
161           },
162           "results": {
163             "$ref": "#/$defs/results"
164           }
165         },
166         "required": [
167           "job",
168           "results"
169         ]
170       }
171     }
172   }
```

Листинг А.3. Решение задачи резки