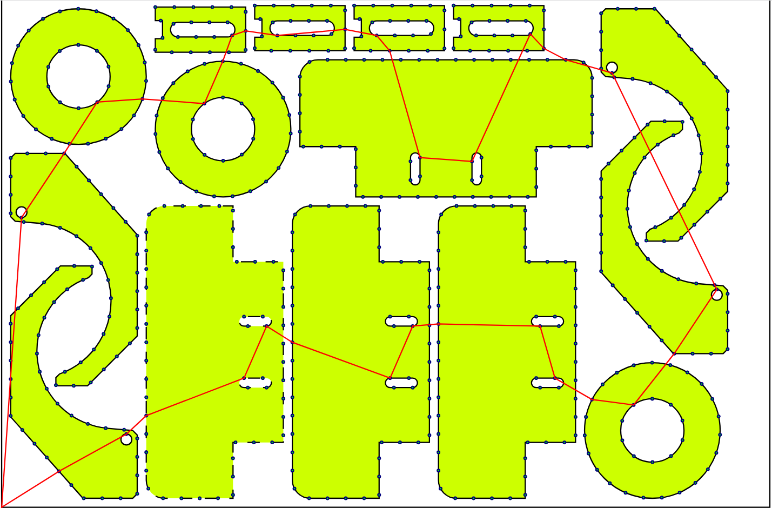
Ответы на вопросы / замечания официального оппонента д.т.н., профессора **Верхотурова Михаила Александровича**

1. *В диссертационной работе использовано несколько языков программирования – С, Julia, Python, JavaScript. Чем вызвано такое разнообразие? Каковы причины выбора этих языков программирования?*

Прежде всего тем, что алгоритмы разрабатывались в разных научных группах, на выбор языка влияли ранее использованные языки. На выбор языка Python повлияло также наличие специализированных библиотек для работы с графами и матрицами, которые позволили значительно повысить суммарное быстродействие алгоритма решения задачи PCGTSP, хотя возможно следует исследовать вопрос миграции на высокопроизводительный компилируемый язык, например С++. Язык C оказался востребован также благодаря производительности и потому, что для алгоритма решения задачи CCP оказалось возможным обойтись без сложных структур данных. Язык Julia рассматривался как возможная альтернатива – «Python с производительностью C++», но на данный момент используется только для исполнения эвристики PCGLNS. Использование языка JavaScript мотивировано прежде всего его доминированием (хотя уже и не абсолютным) в современных браузерах.

1. *Численные эксперименты второй главы (решение задачи PCGTSP) проводились на открытой библиотеке PCGTSPLIB, содержащей абстрактные экземпляры обобщённой задачи коммивояжера с ограничениями предшествования, зачастую асимметричные. В то же время автор принимал участие в разработке библиотеки экземпляров задач именно маршрутизации инструмента, где матрица расстояний и частичный порядок определяются геометрией плоских деталей. Проведение численных экспериментов над этой библиотекой дало бы более наглядные результаты и лучше бы продемонстрировало возможность применения алгоритма для проектирования управляющих программ машин листовой резки с ЧПУ.*

Действительно, в данный момент как раз идут исследования всех описанных алгоритмов (а также динамического программирования по схеме Беллмана, разработанного группой А. Г. Ченцова и активно использовавшегося в диссертационной работе) на библиотеке задач резки. Для этого в неё были добавлен целый ряд раскройных карт, содержащих несколько десятков (30–40) контуров, представляющих особый интерес. На рисунке раскройная карта p1xl\_24, 559 точек врезки, 34 контура, длина маршрута 2595 мм. На момент разработки алгоритма ветвей и границ для PCGTPS он, конечно, тестировался на библиотеке задач резки, но только на задачах небольшого размера, поэтому было принято решение в качестве основной использовать библиотеку PCGTSPLIB.

1. *Вопросы оптимизации раскроя намеренно не включены в круг исследуемых в диссертационной работе. Тем не менее, интересны перспективы применения описанных в алгоритмах для совместной оптимизации процессов раскроя и резки плоских деталей.*

Да, задача совместной оптимизации раскроя и резки (Integrated Nesting & Routing Problem) с одной стороны весьма сложна, с другой же – перспективна с точки зрения повышения эффективности производства. Активных исследований по ней в данный момент не ведётся, но уже показано, что решение задачи INRP способно изменять маршрут инструмента, а также время и стоимость резки. В предварительных исследованиях использовался алгоритм А. Г. Ченцова, но новые алгоритмы могут использоваться вместо него, например для раскройных планов большего размера и тем самым применяться в исследованиях задачи INRP.

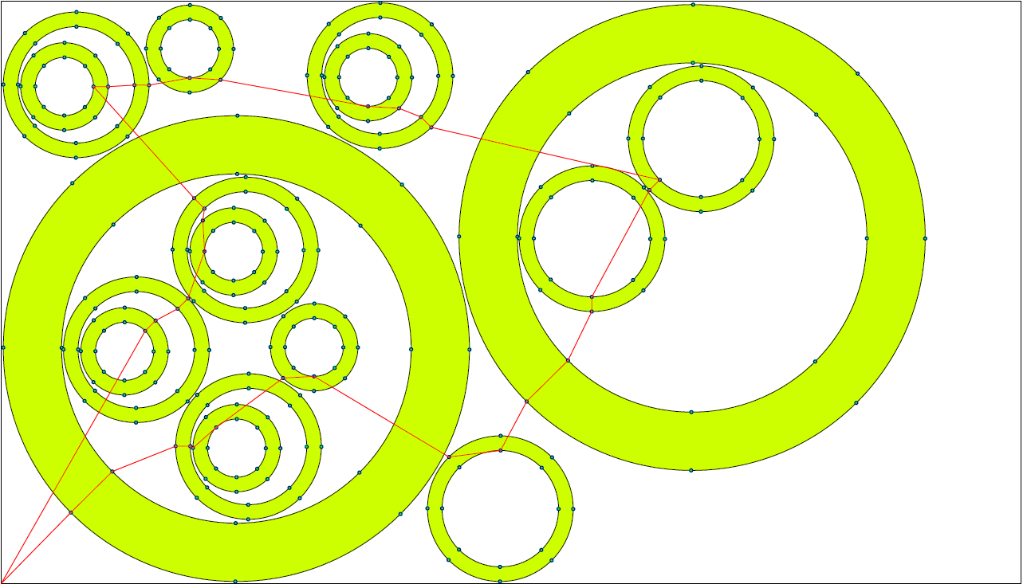
1. *Из текста диссертации непонятна методика получения оценок различных методов построения нижней оценки в табл. 2.2 на стр. 44*

Данные оценки являются не строгими, они были получены в диссертационной работе как эмпирические и использовались для выбора методов построения нижней оценки решения задачи PCGTSP. Фактически использовались как раз раскройные планы из библиотеки задач резки сравнительно небольшого размера (например, e5x\_1 с 23 кластерами), для них выполнялся полный алгоритм ветвей и границ со всеми вариантами расчёта и собиралась статистика по полученным оценкам для каждого варианта. Распределение оценок оказывается, как и ожидалось, близким к нормальному, по этим распределениям и получены цифры в табл. 2.2 на стр. 44. Разумеется, при использовании других тестовых задач, они вероятно немного поменяются, но вряд ли это изменит выбор методов оценки для алгоритма.

1. *Чем можно объяснить использование диссертантом метода обхода в ширину на стр. 48, а не, например, обхода в глубину?*

В оригинальной работе Салмана, послужившей источником вдохновения, как раз использовался поиск в глубину, это было вызвано желанием как можно быстрее получить какое-нибудь решение задачи PCGTSP для того, чтобы заработал механизм отсечения ветвей. В данной же диссертационной работе используется эвристика PCGLNS, которая очень быстро (за несколько секунд для задач на 20–30 кластеров) находит решение близкое к оптимальному. Поэтому и применён обход в ширину, он даёт возможность очень быстро начать получать нижние оценки, так как для построения оценки нужно обработать все префиксы одной длины. Но при этом точное решение будет получено только после полного обхода всего дерева, что требует экспоненциального времени. Таким образом, выбор метода обхода вызван переносом приоритетов с получения близких к оптимальным решений на получение нижних оценок.

1. *Можно ли сравнить эффективность получения точного решения задачи GTSP алгоритмами, разработанными А. Г. Ченцовым, и алгоритмами, предложенными автором диссертации?*

Несмотря на то, что алгоритмы используют динамическое программирование, схемы их работы существенно отличаются. Алгоритм А. Г. Ченцова должен отработать полностью и гарантированно даёт точное решение. Время его работы колеблется от десятков секунд до нескольких часов, в зависимости от сложности задачи, кроме того, существует граница сложности, максимальное количество контуров (порядка 33). На рис. 2.1 на стр.  38 как раз приведён эмпирически подобранный пример (34 контура), который (на момент исследования) не решался алгоритмом А. Г. Ченцова, но легко решался алгоритмом ветвей и границ. Последний же работает так: он быстро (за несколько секунд) выдаёт решение близкое к оптимальному, ещё через несколько секунд получает первую нижнюю оценку, которую начинает постепенно улучшать. Он непременно получит точное решение, но также только в конце работы алгоритма, к сожалению, оценка времени всё равно остаётся экспоненциальной. Можно говорить, что новый алгоритм работает в целом медленнее, обычно его время работы начинается с 10–15 минут и может занимать даже сутки. Можно предположить, что учёт ограничений предшествования в нём реализован лучше, большая вложенность контуров заметно сокращает время счёта, однако это конечно интересно исследовать явно. Таким образом, он заметно быстрее даёт решение, близкое к оптимальному, но дольше – оптимальное решение.

С другой стороны, алгоритм А. Г. Ченцова явно более зрелый, например, он способен учитывать ограничения термической резки (правила жесткости листа и детали), которые пока не реализованы в алгоритмах диссертационной работы.

Ответы на вопросы / замечания официального оппонента д.т.н., профессора **Коновалова Анатолия Владимировича**

1. *В описании численных экспериментов для алгоритма решения задачи PCGTSP (вторая глава) указано, что они проводились на вычислительном кластере «Уран» института математики и механики им. Н. Н. Красовского Уральского отделения Российской академии наук и приведены его характеристики – 16-ядерные lntel Xeon, 128G RAM. На каком оборудовании проводились численные эксперименты третьей главы с задачей непрерывной резки?*

Данный алгоритм имеет гораздо более скромные требования к аппаратному обеспечению, поэтому расчёты велись на обычных персональных компьютерах и ноутбуках, например 4-ядерный Intel Core i7 CPU, 8 Gb RAM под управлением ОС Windows 7.

1. *При разработке управляющих программ для оборудования термической резки с ЧПУ важным является учёт тепловых деформаций, возникающих в процессе резки. Каким образом возможно реализовать такой учёт в описанных в диссертационной работе алгоритмах?*

На данный момент существуют два основных подхода к учёту тепловых деформаций – 1) использование эмпирических правил типа «жёсткости детали» и «жёсткости листа», которые фактически приводят к некоторым геометрическим вычислениям в ходе выполнения алгоритмов и 2) прямой расчёт тепловых полей, возникающих в процессе резки. Второй подход значительно более трудоёмкий, поэтому вероятно есть смысл начать с первого. На этом пути не видится непреодолимых трудностей, тем более что эта работа уже была проделана в научной группе А. Г. Ченцова. Построение зон жесткости для префикса σ представляется вполне возможным в случае алгоритма ветвей и границ. С эвристикой же поиска точек врезки для задачи непрерывной резки ситуация сложнее. Во-первых, её придётся дооснастить механизмом временного восстановления внешних контуров, удалённых на первом шаге алгоритма для учёта ограничений предшествования. Во-вторых, тут мне кажется важным, что алгоритм даёт несколько возможных точек врезки в контур. В существующей версии мы выбираем любую случайным образом или самую последнюю во «внешнем» контуре, и в том, и в другом случае для простоты. Вероятно, потребуется отдельная процедура выбора точки врезки из нескольких возможных, чтобы не создавать «столкновений» точек врезки.

1. *В доказательстве утверждения 3.1 на стр. 68 функция φ(t) названа выпуклой, во избежание двусмысленности следовало бы уточнить, что она является выпуклой вниз, а не вверх, что подтверждается тем, что её вторая производная неотрицательна.*

Совершенно верно, фактически в тексте диссертации термин «выпуклый» используется в двух смыслах – для выпуклых множеств (например, на стр. 70) и для выпуклых функций, и может создаваться путаница. В данном случае имелась в виду функция, выпуклая вниз.

1. *В тексте диссертации списки оформлены вразнобой, иногда пункты списков начинаются со строчной буквы, иногда с прописной, в конце ставится иногда точка, иногда точка с запятой, а иногда ничего.*

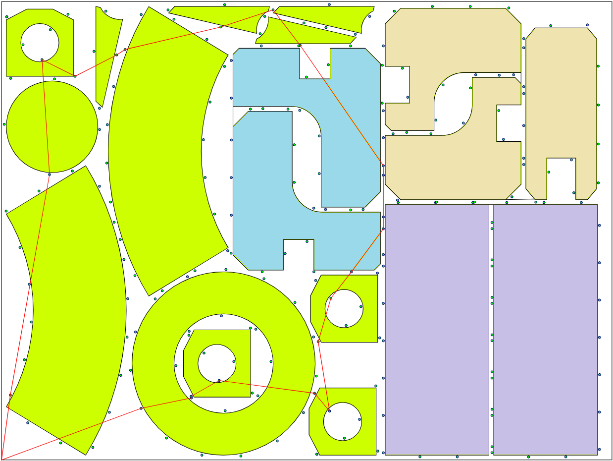
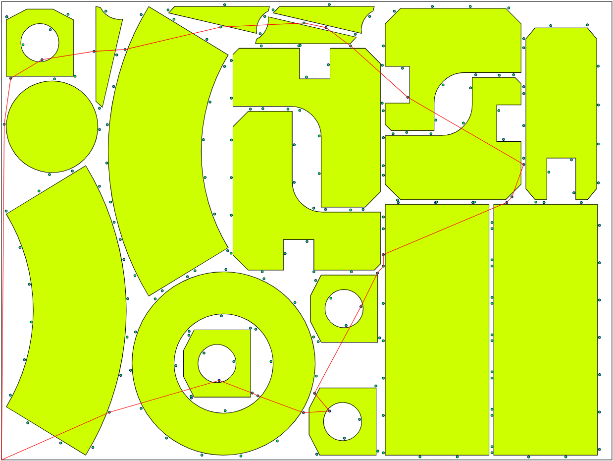
Спасибо за замечание. Оформлению списков действительно не было уделено достаточно внимания при работе над текстом диссертации.

1. *В тексте диссертации также имеются опечатки, так в формуле на стр. 18 вместо CJ должно быть Cj по смыслу.*

Действительно, хотя при подготовке текста диссертации использовались средства проверки правописания, отловить все ошибки они не в состоянии, в особенности опечатки в математических формулах.

Ответы на вопросы / замечания официального оппонента д.т.н., доцента **Ложникова Павла Сергеевича**

1. *Во введении (раздел 1.1) описаны различные техники резки, применяемые в современном производстве, а именно: стандартная, мультиконтурная и мультисегментная. В то же время, описанные алгоритмы ориентированы на использование только резки по замкнутому контуру, то есть стандартной. Каким образом возможно использовать нестандартные техники резки в рамках разработанного в диссертации алгоритмического обеспечения для решения задач оптимальной маршрутизации инструмента?*

Здесь, как мне кажется, нужно разделить случаи мультиконтурной и мультисегментной резки. На случай мультиконтурной резки оба алгоритма обобщаются почти автоматически, просто сегмент резки становится сложнее. В диссертации на стр. 78 фактически приведён пример мультиконтурной резки, осуществлённой алгоритмом непрерывной резки из третьей главы. Уже в начале этого годы были проведены аналогичные исследования для алгоритма ветвей и границ, пример на рисунке. Таким образом, оба алгоритма могут использоваться для мультиконтурной резки. А вот мультисегментная резка, когда сегмент резки теряет свойство замкнутости, – более сложный случай. Фактически задача упрощается, на сегменте остаются только две точки врезки, но теряется свойство, что маршрут покидает сегмент в той же точке. Это требует доработки обоих алгоритмов диссертационной работы.

1. *Алгоритм решения задач PCGTSP, описанный в Главе 2, решает их в самой общей постановке, для произвольных входных данных, тогда как задачи, возникающие при оптимизации маршрута резки, существенно эвклидовы. Каким образом учитывается это их свойство в алгоритме? Возможно ли использовать геометрические соображения для улучшения работы алгоритма?*

Да, действительно, алгоритм ветвей и границ разрабатывался для решения задач PCGTSP в произвольном виде, в том числе асимметричных и основные результаты получены на библиотеке PCGTSPLIB, содержащей некие абстрактные постановки. В то же время алгоритм отлаживался и на задачах, полученных из реальных задач маршрутизации инструмента, состоящих из контуров на плоскости, и действительно возникла потребность внести в алгоритм некоторые блоки, учитывающие специфику таких задач. В частности, именно из этих соображений в алгоритм было добавлено построение графа кластеров H2 на основе цепей длины 2, как описано на стр. 42, что позволяет точнее учитывать взаимное расположение контуров на плоскости. Эта идея оправдала себя, примерно в половине случае такие оценки оказываются точнее. Возможно также использование более длинных цепей, но это алгоритмически заметно сложнее. В то же время, использование именно геометрических соображений в текущем дизайне алгоритма вряд ли возможно, потому что исходный файл задания уже очищен от геометрической информации, хотя конечно её можно туда добавить, если появится идея, как её использовать.

1. *Если результаты работы алгоритма сопоставимы с результатами, полученными универсальным решателем Gurobi, то в чём смысл разработки специализированного алгоритма?*

Во-первых, решатель Gurobi является коммерческим программным продуктом, требующим лицензирования для своего использования, поэтому он может оказаться недоступен в некоторых случаях. Во-вторых, текущая версия алгоритма определённо не является идеальной и может быть улучшена, как с точки зрения оптимизации, так и путём, например разработки новых методов получения нижних оценок. Уточнение последних сразу же сказывается на производительности алгоритма. Ну и в-третьих, в процессе разработка алгоритма появилось несколько плодотворных идей, которые могут оказаться полезными и в других исследованиях.

1. *Английские аббревиатуры систематически вводятся без расшифровки, например САМ на стр. 5 или названия классов задач резки (ССР, ЕСР, ICP и т. д.) на стр. 7. Часть из них приведена в списке сокращений на стр. 97, но не все.*

Действительно, при работе над текстом диссертации следовало уделить внимание этому аспекту, так как многие из этих аббревиатур воспринимаются как общепринятые, но это, разумеется, не так. Аббреиватуры классов задач расшифрованы на стр. 29, следовало сделать это раньше по тексту.

1. *В тексте работы имеются грамматические ошибки. Так, на стр. 5, 7 и 31 пропущен дефис в термине «САD/САМ-система», хотя в других местах он употребляется.*

Спасибо за замечание.

Ответы вопросы / замечания по автореферату к.т.н., доцента **Захаровой Галины Борисовны**

1. *На рис. 2 на стр. 10 приведена классическая задача коммивояжера TSP как один из частных случаев задачи резки, хотя как правило последняя сводится к обобщённой задаче коммивояжера GTSP. Не ясно, каким образом простая задача коммивояжера может применяться для поиска оптимального решения задачи маршрутизации режущего инструмента.*
2. *Понятно, что ограничение предшествования сокращает время счёта для алгоритма главы три, не использующего дискретизацию контуров. А как аналогичный эффект достигается для алгоритма ветвей и границ второй главы?*

Алгоритм ветвей и границ использует ограничения предшествования в нескольких местах. Во-первых, при релаксации исходной задачи PCGTSP в задачу ATSP можно уничтожить часть рёбер графа, движение по которым фактически невозможно из-за ограничений предшествования. В ходе диссертационной работы удалось сформулировать более строгие ограничения на допустимые рёбра, чем те, которые кажутся интуитивно понятными и описаны в литературе. Тем самым задача ATSP упрощается и может решаться за меньшее время. Во-вторых, в момент ветвления не все кластеры являются достижимыми из последнего кластера префикса σ и здесь тоже более строгий анализ позволил сократить размер дерева поиска, а значит и время счёта.

1. *Не для всех реализованных в работе алгоритмов приведено время счета. Хотелось бы получить представление о сравнительном быстродействии этих алгоритмов.*

Ответы на вопросы / замечания по автореферату д.т.н., профессора **Мельникова Андрея Витальевича**

1. *В автореферате при описании разработанных в работе алгоритмов решения задач маршрутизации говорится только об ограничениях типа условий предшествования, при этом не обговаривается учёт других технологических ограничений термической резки.*

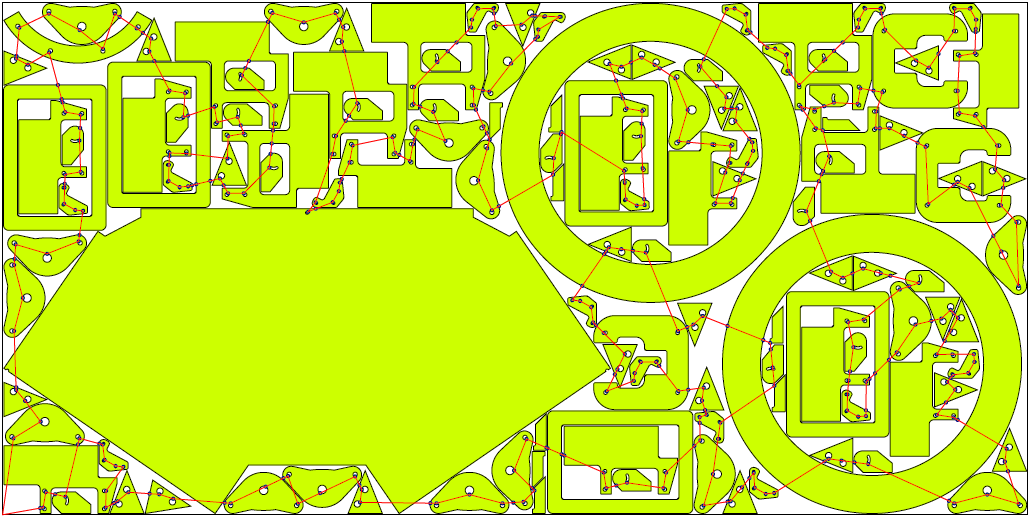
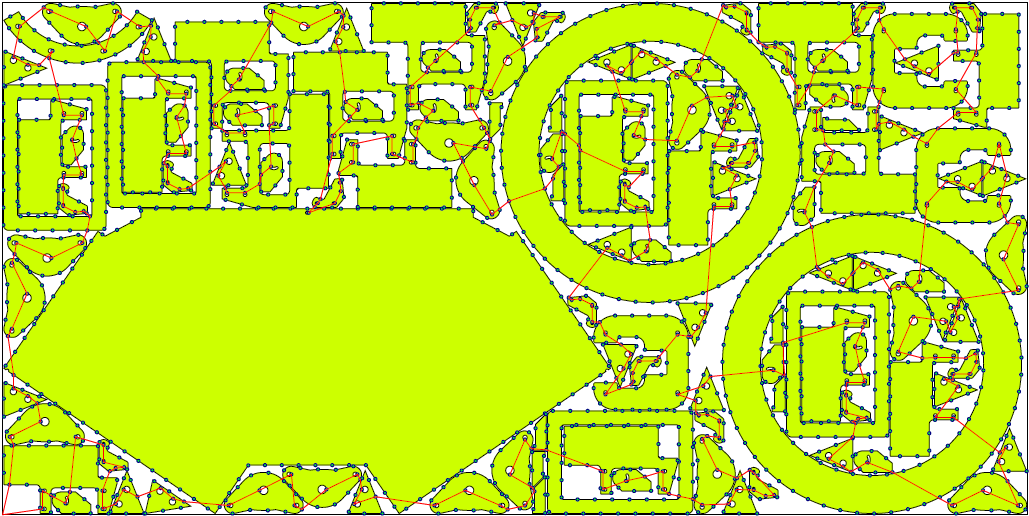
Это действительно так, в первоначальном виде алгоритмы учитывают только условия предшествования. Учёт других ограничений (например, на позиции точек врезки, правила жесткости листа и детали) требует дальнейших исследований. Отчасти этот путь уже был проделан в научной группе А. Г. Ченцова, что позволяет надеяться на возможность такого развития. Можно ожидать также, что по своему внутреннему устройству алгоритм ветвей и границ окажется легче доработать для учёта технологических ограничений термической резки, чем непрерывно-дискретный алгоритм, хотя и тут уже есть некоторые наработки.

1. *На стр. 18 упомянуто, что алгоритм, использующий модель непрерывно-дискретной оптимизации, сравнивается с точным алгоритмом А. Г. Ченцова, разработанным на основе метода динамического программирования. В дополнение к этому следовало бы привести сравнение и с алгоритмом ветвей границ, разработанным в диссертационной работе. Имеется в виду и факт получения глобального экстремума и вычислительные затраты на его получение.*

Ответы на вопросы / замечания по автореферату д.т.н. **Вохминцева Александра Владиславовича**

1. *Хотелось бы сравнить результаты работы алгоритмов второй и третьей глав на одних и тех же раскройных планах, что позволило бы лучше судить о применимости этих алгоритмов для разработки управляющих программ для машин листовой резки с ЧПУ.*

Отчасти этот вопрос освещён в тексте диссертации на стр. 54 и 76, где приведены результаты работы обоих алгоритмов для задач большой размерности (423 и 620 контуров).



Видно, что в значительной части маршруты резки, полученные разными алгоритмами, совпадают. В задачах меньшей размерности совпадение как правило полное. При этом длина полученного маршрута как правило немного короче для случая непрерывно-дискретной оптимизации, что легко объяснимо. Разница может составлять от 1% до 10% для сложных раскройных планов.

Действительно, подобное сравнение напрашивается, хотя и требуется продумать его методику. В настоящее время готовится большой эксперимент по сравнению производительности всех алгоритмов (обоих, представленных в диссертации), алгоритма А. Г. Ченцова на основе динамического программирования и при помощи решателя Gurobi. Эксперимент будет проводиться на открытой библиотеке заданий для резки.

1. *В автореферате отсутствуют экспериментальные данные о быстродействии реализации алгоритма, использующего непрерывно-дискретную оптимизацию.*
2. *Интересно было бы исследовать влияние уровня вложенности деталей в раскройном плане на быстродействие описанных алгоритмов.*

Вопрос действительно интересный, и также требуется проведение численных экспериментов для его решения. Даже из анализа содержимого таблицы 2.3 на стр. 55 становится понятным, что уровень вложенности контуров крайне сильно влияет на производительность алгоритмов. Именно высокая вложенность, вероятно, позволила получить решение для задачи размерностью 151 кластер из библиотеки PCGTSPLIB, что значительно превышает предыдущий «рекорд» порядка 33 контуров. В тексте диссертации на стр. 52 приводится оценка трудоёмкости алгоритма путём вычисления мощности множества идеалов, порождённых частичным порядком, то есть вложенностью. Однако интересно соотнести эти соображения с характеристиками раскройного плана.

Ответы на вопросы / замечания по автореферату д.т.н., профессора **Мартынова Виталия Владимировича**

1. *К недостаткам автореферата можно отнести почти полное отсутствие описания деталей программной реализации алгоритмов, приведён псевдокод только одного программного блока*
2. *С точки зрения оформления неудачно выбрано расположение в тексте рисунка 1 на стр. 8.*

Действительно, для вёрстки текста диссертации и автореферата использовалась система компьютерной вёрстки LaTeX, которая сама определяет расположение рисунков и таблиц в тексте, руководствуясь некоторыми сложными алгоритмами. При финальной вёрстке этот алгоритм нашёл не очень удачное место для рисунка, разбив список научных конференций. Это можно было легко поправить вручную, но было замечено слишком поздно. Таким образом, получилась демонстрация достоинств и недостатков полуавтоматических механизмов раскроя, применении пусть и не к машиностроению, но к полиграфии.

1. *Сформулированные основные результаты и выводы по работе не коррелируют однозначно с поставленными задачами.*