УДК 007:52

#### А. Г. Ченцов, П.А. Ченцов, А.А. Петунин

## МАРШРУТНЫЕ ПРОЦЕССЫ В ЗАДАЧАХ ПОСЛЕДОВАТЕЛЬНОГО ОБХОДА МНОЖЕСТВ ПРИ НАЛИЧИИ ОГРАНИЧЕНИЙ

Институт математики и механики им Н.Н.Красовского УрО РАН

Уральский федеральный университет

*Исследуется задача экстремальной маршрутизации перемещений с ограничениями и функциями стоимости, зависящими от списка заданий. Постановка ориентирована на инженерные приложения, связанные с листовой резкой деталей на машинах с ЧПУ; возможны и другие применения (так, например, полученные результаты могут использоваться в задаче минимизации дозовой нагрузки при демонтаже системы радиационно опасных объектов при авариях на АЭС). При этом объектами посещения являются непустые конечные множества – мегаполисы. Среди ограничений особо выделяются условия предшествования, которые удается использовать для снижения вычислительной сложности. В качестве основного метода используется широко понимаемое динамическое программирование, учитывающее условия предшествования и зависимость функций стоимости от списка заданий. Упомянутая зависимость возникает, в частности, при использовании штрафов за нарушение ограничений, диктуемых технологическими требованиями. В процессе решения оптимизируются очередность выполнения заданий, конкретная траектория процесса и точка старта. Алгоритм реализован в виде программы для ПЭВМ; решены модельные примеры.*

*Динамическое программирование; допустимое решение; маршрут; траектория*

#### A.G. Chentsov, P.A. Chentsov, A.A. Petunin

## ROUTING PROCESSES IN PROBLEMS OF SEQUENTIAL TRAVERSAL OF SETS IN THE PRESENCE OF CONSTRAINTS

*The extremal problem of routing movements with constraints and cost functions depending on the list of tasks is investigated. The statement is focused on engineering applications associated with sheet cutting of parts on CNC machines; other applications are also possible (thus, obtained results can be used in the problem of minimizing the dose load during dismantling of the system of radiation hazardous elements during accidents at nuclear power plants). The objects of visit are nonempty finite sets - megalopolises. Among the restrictions, the precedence conditions stand out, which can be used to reduce the computational complexity. The main method is the widely understood dynamic programming, which takes into account the conditions of precedence and the dependence of cost functions on the list of tasks. The above-mentioned dependence arises under penalty employment under violation of constraints dictated by technological requirements. In the process of solving, the sequence of tasks and the specific trajectory of the process, as well as the starting point, are optimized. The algorithm is implemented as a PC program; model examples solved.*

*Dynamic programming; admissible solution; route; trajectory*

**Введение.** Объектом исследования в статье являются экстремальные задачи маршрутизации перемещений с ограничениями различных типов; среди последних особо выделяем условия предшествования и динамические ограничения, возникающие по мере развития процесса и проведения тех или иных работ. При соответствующей формализации возникает постановка, идейно близкая к дискретным задачам управления (имеется в виду дискретность и по времени, и по фазовому состоянию). Оптимизируется комплекс, включающий точку старта, вариант очередности исполнения заданий (далее – маршрут) и конкретную траекторию; сам возникающий при этом комплекс (триплет) именуем маршрутным процессом. Возможные применения могут быть, в частности, связаны с атомной энергетикой (см. [1-3]; имеется в виду задача минимизации дозовой нагрузки работников при демонтаже системы радиационно опасных объектов) и машиностроением (см. [4-6], где рассматривалась задача управления инструментом при фигурной листовой резке на машинах с ЧПУ); имеются и другие приложения.

В настоящей работе мы ориентируемся на применение разрабатываемых методов в машиностроении; следуем при этом конструкциям монографии [4].

Здесь первоначальная задача управления режущим инструментом с условиями предшествования и динамическими ограничениями преобразуется к математической постановке оптимизационной задачи в классе вышеупомянутых маршрутных процессов; нашей целью является нахождение глобального экстремума и соответствующего оптимального решения. Обсуждаются элементы общей теории и конструируемый на основе этой теории оптимальный алгоритм, реализованный на многоядерной ПЭВМ.

В статье используются понятия и обозначения [4,часть II], относящиеся к математической постановке, а также содержательные построения [4,часть I]. Рассматриваемая ниже задача имеет своим прототипом известную труднорешаемую задачу коммивояжера или TSP в англоязычной литературе; см. [7–12]. Однако существенные особенности качественного характера (прежде всего, наличие ограничений и, в частности, условий предшествования) мотивируют построение специализированной теории; см. [1,3–6,13,14}. Разработке упомянутых теоретических вопросов посвящены, в частности, монографии [13,14]. В настоящем изложении выделяем [4], где для актуальной инженерной задачи удается продемонстрировать ряд принципиальных положений теоретического характера и, в частности, указать возможности динамического программирования (ДП) как общего метода решения различных прикладных задач. Исследуемая задача требует строгой формализации, при построении которой используются разнообразные сведения математического характера [15–17].

**Конкретизация общей задачи.** Совсем кратко напомним сейчас некоторые построения [4, §3.3]. Полагаем здесь, что задан прямоугольник на плоскости \*\*\*\*\*1, где \*\*\*\*. Имеется раскройный план; намечены контуры попарно дизъюнктных деталей. У каждой детали имеется внешний контур и, возможно, несколько внутренних контуров (см. [4, §3.2]). Реально контуры окружены близкими к ним эквидистантами. Однако сейчас для простоты полагаем их совпадающими с контурами, то есть будем говорить о резке по контурам. По технологическим соображениям резка внутренних контуров каждой детали (если они есть) должна предшествовать резке внешнего контура. Возникает естественный вариант условий предшествования. В интересах компьютерной реализации считаем, что возле каждого контура намечены возможные точки врезки и соответствующие им точки выключения инструмента: итак, процедура врезки должным образом дискретизируется. В результате возникают непустые конечные множества – мегаполисы \*\*\*\*3, где \*\*\*4 -- натуральное число, для которого \*\*\*5. Элементами этих множеств являются точки врезки и точки выключения инструмента. Точки этих двух типов группируются в пары. Для каждого мегаполиса \*\*\*6 определено отношение в виде подмножества\*\*\*\*7; оно всякий раз состоит из упорядоченных пар. Элементами каждой такой пары являются точка врезки и соответствующая ей точка выключения инструмента. Предполагается, что инструмент, покидая точку старта, перемещается к мегаполису \*\*\*8, занумерованному первым, прибывает к некоторой (из намеченных заранее) точке врезки x, после чего выполняется врезка и работа по вырезанию контура с последующим перемещением в точку выключения инструмента \*\*\*9; затем инструмент перемещается к мегаполису, занумерованному вторым и т.д.; при этом \*\*\*10. С учетом последнего замечания можно принять, что развитие маршрутного процесса характеризуется схемой \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*11, где \*\*\*12 есть количество контуров, \*\*\*13 -- точка старта, \*\*\*14 -- перестановка индексов.

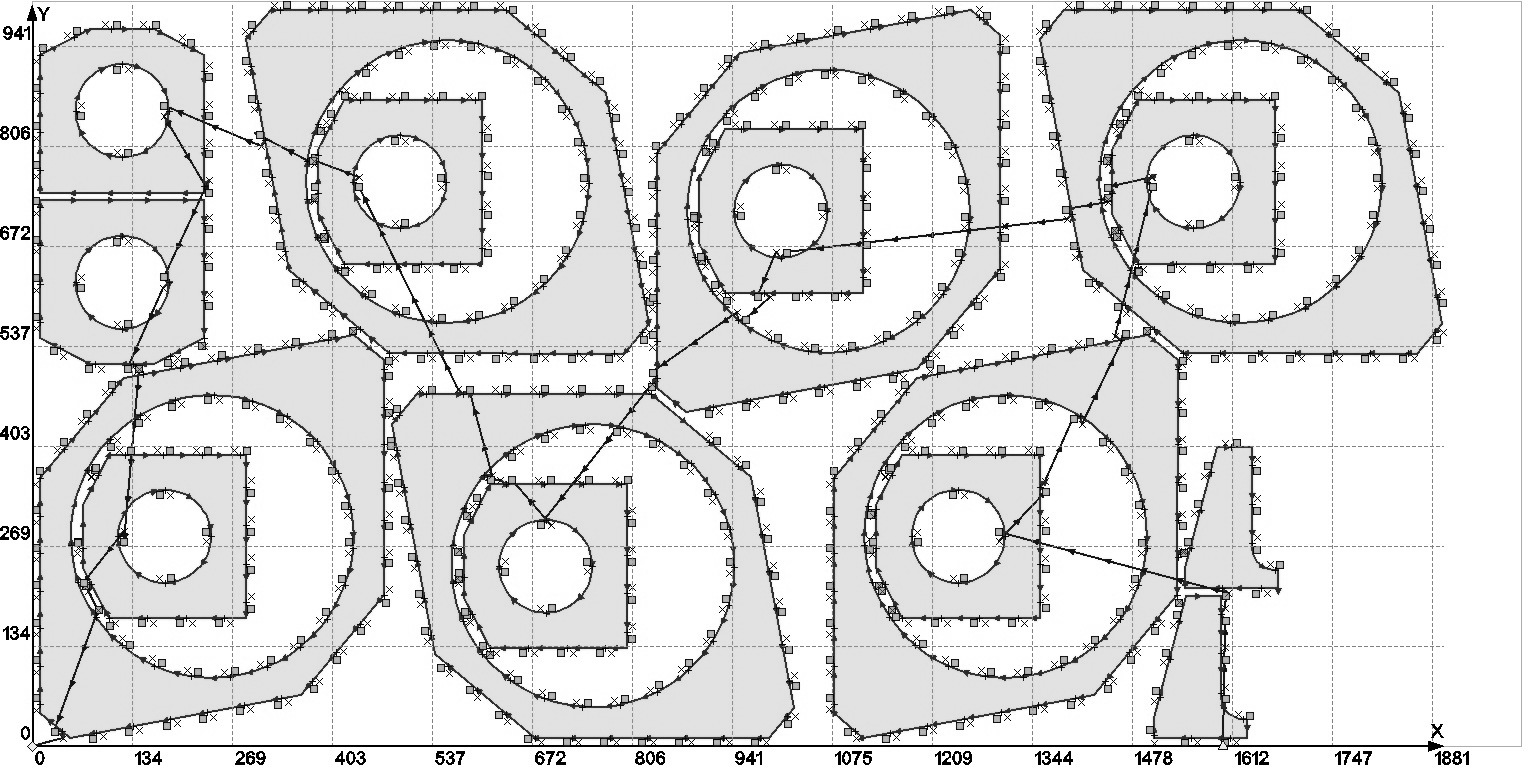
Имеются условия предшествования. В частности, как уже отмечалось, резка внутренних контуров детали должна предшествовать резке внешнего контура. Возможны и другие варианты упомянутых условий предшествования; например, может использоваться правило: сначала режутся «большие» детали. Из других ограничений сейчас отметим условия, имеющие смысл тепловых допусков (см. [18]). Имеется в виду обеспечение ситуации, при которой возле точек врезки сохранялось бы достаточно большая (в смысле площади) область невырезанного металла с тем, чтобы обеспечивался удовлетворительный отвод тепла (имеется в виду случай термической резки). Более подробное описание см. в [18]. В рассматриваемой модели резки по замкнутому контуру на этапе математической постановки исключено собственно время резки контуров, поскольку оно одинаково для всех вариантов решения и может быть легко учтено введением дополнительного слагаемого. Функция стоимости внешних перемещений определена как время, затрачиваемое в режиме холостого хода. Аналогично оценивается терминальное состояние: учитывается время перемещения до точки парковки в режиме холостого хода. Стоимость внутренних работ получается суммированием двух компонент. Одна из них определяется суммой времен, затрачиваемых на перемещение от точки врезки до точки начала реза, и от последней – до точки выключения инструмента (перемещение со скоростью рабочего хода). Вторая компонента определяется функцией штрафа и «включается» при нарушении тепловых допусков. Предполагается, что точка начала реза сопоставляется каждой паре с элементами в виде точки врезки и точки выключения инструмента. Таким образом, формируется значение суммарного времени для каждого маршрутного процесса. Более подробные сведения см. в [4, часть 1, глава 3] и в [18] (для задачи с фиксацией точки старта см. [4, часть 2]). Здесь при построении решения использовался алгоритм на основе ДП. Логическая схема данного алгоритма приведена в следующем разделе; примеры иллюстрируют ее работоспособность.

**Модельный пример: оптимальное решение задачи управления инструментом при фигурной листовой резке деталей.** В начале настоящего раздела используем стандартную конкретизацию общей задачи, а, точнее, рассматриваем вариант листовой резки деталей по замкнутому контуру. Для этого варианта будет получено оптимальное решение: построен алгоритм, реализованный в виде стандартной программы для ПЭВМ. Затем будет рассмотрен пример задачи с мультиконтурной резкой (см. [4, часть I]; точка старта в этом примере фиксирована, что соответствует постановкам [4]).

Итак, мы следуем содержательной постановке предыдущего раздела. По соображениям экономии объема ограничимся сейчас кратким описанием параметров задачи, для которой было найдено оптимальное решение. Предполагается заданным раскройный план, предусматривающий резку тридцати контуров у 16 деталей; итак, в данном примере *N* = 30 (сами детали не обсуждаем сейчас, поскольку для нашей задачи существенны только контуры). Полезно ввести специальное понятие порта, включая в это понятие триплет, содержащий точку врезки, соответствующую ей точку выключения инструмента, а также точку начала реза. Точка выключения инструмента близка к точке врезки. В примере предполагается, что общее число точек врезки (по всем контурам) равно 508. Количество адресных пар, определяющих условия предшествования в данном примере равно 20. Имеются в виду пары индексов мегаполисов (по смыслу «отправитель» и «получатель»; условия предшествования предполагают перемещение во времени только в направлении от «отправителя» к «получателю»). Имеются в виду оговоренные в предыдущем разделе условия, связывающие резку внутренних контуров (и «внутренних» деталей) и резку внешнего контура у каждой детали.

Для проведения вычислений использовался компьютер с процессором Intel i7-2630QM и 8 Гб оперативной памяти, работающий под управлением Windows 7 (64-bit). Программа разработана на языке C++, скомпилирована при помощи компилятора MinGW с использованием библиотеки Qt. Мегаполисы конструируются из точек врезки и точек выключения инструмента. Их упорядоченные пары образуют отношения, т.е. множества упорядоченных пар. Вместе с тем, с каждым мегаполисом удобно связать сейчас набор портов. Предполагается, что минимальное количество портов для мегаполиса равно 4 (при этом число «городов» мегаполиса равно 8), а максимальное количество портов для мегаполиса равно 34. Количество возможных точек старта (мощность множества упомянутого множества) равно 58. Они расположены на сторонах исходного прямоугольника с координатами углов (0,0), (0,1000), (1900,1000), (1900,0) равномерно с шагом 100 (по часовой стрелке). Точка финиша имеет координаты (0, 0). Все размеры даны в миллиметрах. Скорость холостого хода 500 мм/с, скорость реза 10 мм/с. Предполагается, что внутренние работы определяются суммой времен перемещения в режиме реза (рабочего хода) от точки врезки до точки начала реза и от этой последней точки до точки выключения инструмента, а также значения штрафной функции, определяемой подобно [18]. Для штрафной функции длина области завершения реза имеет значение 100 мм, ширина – 25 мм, пороговое значение 0,25. Если больше 25% площади области завершения реза приходится на пустоты в металле или на пространство за пределами листа, то штрафная функция имеет значение 1000000, иначе 0.

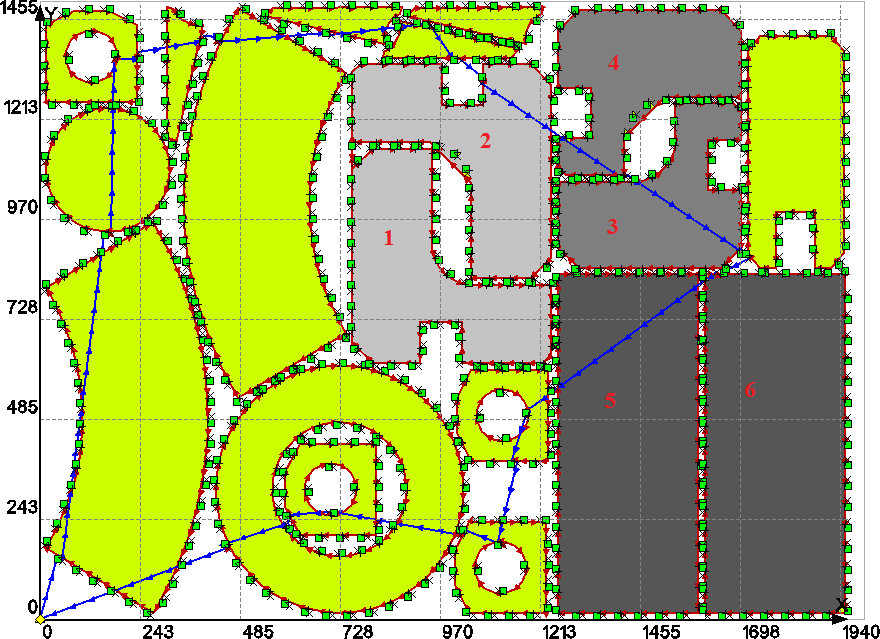
В ходе счета получено значение экстремума 67,555. Время счета 51 мин. 58 сек. Координаты точки старта (1600 мм, 0 мм). Оптимальное решение показано на рисунке 1, где, в частности, указана оптимальная точка старта.



*Рис. 1. Маршрут и трасса обхода множеств*

В связи с применением метода динамического программирования в задачах маршрутизации отметим работы [18-22].

Модель мегаполисов применима и при использовании нестандартных техник резки, в частности, т.н. мульти-контурной резки, которая предполагает использование только одной точки врезки для вырезки двух и более контуров внутри одного сегмента резки. Под сегментом резки подразумевается траектория рабочего хода инструмента между точкой врезки и соответствующей ей точкой выключения инструмента [23]. На рис. 2 показан пример расчета оптимальной траектории инструмента при резке 19 деталей, заданных 24-мя контурами. Для пар деталей, отмеченных цифрами 1–2, 3–4, 5–6 были сформированы три отдельных мегаполиса, всего 21 мегаполис. Расчет оптимальной точки старта в данном примере не производился.



*Рис. 2. Оптимальная траектория резки при использовании мульти-контурной техники резки*

Следует отметить, что описанная в статье математическая модель позволяет получать на обычном персональном компьютере точные варианты решения для задач маршрутизации инструмента машин листовой резки с ЧПУ для раскройных карт, содержащих 30 и более деталей.

**Структура алгоритма.** Используемый алгоритм соответствует конструкциям [25,26], восходящим к [14]. Важно отметить, что при должной формализации (см. общие построения [25,26]) исключительно важным здесь представляется теоретический этап, приводящий к получению уравнения Беллмана. В этой связи отметим [26, предложение 1, (2.68)], где получено упомянутое уравнение и приведено подробное доказательство. Подчеркнем, что в этом построении используется нестандартное расширение исходной задачи, которое также приведено в [26, раздел 2]. Ключевой результат – уравнение Беллмана [26, (2.36)] – используется далее для обоснования оптимальности алгоритма. Приводимая ниже схема имеет своим источником построения [14, раздел 4.9] и в полной общности также изложена в [25,26]. Мы ограничимся сейчас обсуждением ее этапов на идейном уровне.

1. **Построение существенных списков заданий.** Как уже отмечалось, для целей снижения вычислительной сложности удается использовать условия предшествования. Реализация данной процедуры связана прежде всего с тем, что насчитывание всего массива значений функции Беллмана подменяется построением некоторых ее слоев и первым шагом в этом направлении является исключение из рассмотрения некоторых списков заданий. Остающиеся после данного исключения списки называем существенными; соответствующее свойство, определяющее существенность, связано с условиями предшествования (см. [26,(2.69)]). Семейства существенных списков ранжируются по мощности, получающиеся подсемейства реализуются посредством итерационной процедуры [26,(2.71)], которая доставляет все семейство существенных списков.
2. **Построение слоев пространства позиций.** Позициями называем упорядоченные пары, элементами которых являются всякий раз состояние, определяющее положение инструмента, и список заданий, подлежащих исполнению. Позиции являются аргументами функции Беллмана (данная функция определена на пространстве позиций). В целях экономии вычислений в пространстве позиций выделяем систему взаимосвязанных слоев (см. [14, раздел 4.9]) и ограничиваемся использованием в качестве аргументов функции Беллмана только позиций из упомянутых слоев. Подчеркнем, что эффект применения слоев связан с условиями предшествования. Итак, эти условия используются «в положительном направлении» в смысле снижения вычислительной сложности.
3. **Построение системы слоев функции Беллмана.** На основе уравнения Беллмана (вид которого нам теперь известен в силу [26, предложение 1, (2.68)]) конструируем систему преобразований, действующих на функции, определяемые на слоях пространства позиций. Простейший (терминальный) слой определяется терминальной компонентой аддитивного критерия, а далее «включается» рекуррентная процедура, имеющая своей логической основой уравнение Беллмана (в этой связи см. [26,(2.76)—(2.78)]). Функции, реализуемые данной процедурой, являются сужениями нашей основной функции Беллмана. В частности, последний этап упомянутой рекуррентной процедуры определяет экстремум задачи маршрутизации с фиксированной точкой старта. Это позволяет прооптимизировать точку старта, что определяет содержание следующего этапа.
4. **Оптимизация точки старта.** После реализации последнего слоя функции Беллмана, отвечающего полному списку заданий, мы располагаем зависимостью экстремума задачи от точки старта. Минимизируя эту зависимость (а это -- совсем несложная процедура), мы определяем глобальный экстремум и оптимальную (по результату) точку старта.
5. **Построение оптимального решения.** Фиксируем оптимальную точку старта, найденную на предыдущем этапе. Далее, используя стандартную в теории ДП процедуру пошагового выбора индексов и упорядоченных пар с элементами в виде точек врезки и точек выключения инструмента, определяем допустимое решение, оптимальное для данной точки старта. Дополняя этой точкой полученное решение, реализованное в виде упорядоченной пары маршрут-траектория, получаем оптимальный маршрутный процесс. В частности, в первом примере такой процесс был построен и указана оптимальная точка старта. Во втором примере оптимизация точки старта не осуществлялась.
6. **Поддержка исследовани**й. Работа выполнена при финансовой поддержке РФФИ (грант 20-08-00873).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Коробкин В. В.*, *Сесекин А. Н.*, *Ташлыков О. Л.*, *Ченцов А. Г.* Методы маршрутизации и их приложения в задачах повышения эффективности и безопасности эксплуатации атомных станций. – 2012.
2. *Ташлыков О. Л.*, *Сесекин А. Н.*, *Щеклеин С. Е.*, *Ченцов А. Г.* Разработка оптимальных алгоритмов вывода АЭС из эксплуатации с использованием методов математического моделирования // Известия высших учебных заведений. Ядерная энергетика. – 2009. – № 2. – с. 115–120.
3. *Chentsov A. G.* Dynamic programming in routing problems (nuclear power, Engineering) // AIP Conference Proceedings. – 2020. – дек. – т. 2315, № 1. – с.040011. – ISSN 0094-243X. – DOI: [10.1063/5](https://doi.org/10.1063/5.0036656). [0036656](https://doi.org/10.1063/5.0036656).
4. *Петунин А. А.*, *Ченцов А. Г.*, *Ченцов П. А.* Оптимальная маршрутизация инструмента машин фигурной листовой резки с числовым программным управлением. Математические модели и алгоритмы. – Издательство Уральского университета, 2020. – 247 с., ISBN 978-5-7996- 3016-4.
5. *Chentsov A. G.*, *Chentsov P. A.*, *Petunin A. A.*, *Sesekin A. N.* Model of megalopolises in the tool path optimisation for CNC plate cutting machines // International Journal of Production Research. – 2018. – т. 56, № 14. – с. 4819–4830. – ISSN 0020-7543. – DOI: [10.1080/ 00207543.2017.1421784](https://doi.org/10.1080/00207543.2017.1421784).
6. *Петунин А. А.*, *Ченцов А. Г.*, *Ченцов П. А.* К вопросу о маршрутизации движения инструмента в машинах листовой резки с числовым программным управлением // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. – 2013. – 2 (169). – ISSN 2304-9766.
7. The traveling salesman problem and its variations / под ред. G. Gutin, A. P. Punnen. – Springer Science & Business Media, 2006. – ISBN 978-0-387-44459-8.
8. *Cook W. J.* In pursuit of the traveling salesman: mathematics at the limits of computation. – Princeton University Press, 2011. – ISBN 978-0-691-15270-7.
9. *Гимади Э. Х.*, *Хачай М. Ю.* Экстремальные задачи на множествах перестановок. – Екатеринбург: Изд-во УМЦ УПИ, 2016. – 220 с.
10. *Меламед И. И.*, *Сергеев С. И.*, *Сигал И. Х.* Задача коммивояжера. Вопросы теории // Автоматика и телемеханика. – 1989. – № 9. – с. 3–33.
11. *Беллман Р.* Применение динамического программирования к задаче о коммивояжере // Кибернетический сборник. – 1964. – т. 9. – с. 219–228.
12. *Хелд М.*, *Карп Р.* Применение динамического программирования к задачам упорядочения // Кибернетический сборник. – 1964. – т. 9. – с. 202–218.
13. *Петунин А. А.*, *Ченцов А. А.*, *Ченцов А. Г.*, *Ченцов П. А.* Элементы динамического программирования в конструкциях локального улучшения эвристических решений задач маршрутизации с ограничениями // Автоматика и телемеханика. – 2017. – № 4. – с. 106– 125. – ISSN 0005-2310. – DOI: [10.1134/S0005117917040087](https://doi.org/10.1134/S0005117917040087).
14. *Ченцов А. Г.* Экстремальные задачи маршрутизации и распределения заданий: вопросы теории. – М.–Ижевск: НИЦ «Регулярная и хаотическая динамика», Ижевский институт компьютерных исследований, 2008. – 240 с. – ISBN 978-5-93972-654-2.
15. *Куратовский К.*, *Мостовский А.* Теория множеств. – М.: Мир, 1970.
16. *Дьедонне Ж.* Основы современного анализа. – Мир Москва, 1964. – 430 с.
17. *Кормен Т.*, *Лейзерсон Ч.*, *Ривест Р.*, *Штайн К.* Алгоритмы. Построение и анализ:[пер. с англ.] – Издательский дом Вильямс, 2009. – ISBN 5-8459-0857-4.
18. *Ченцов А. Г.*, *Ченцов П. А.* Маршрутизация в условиях ограничений: задача о посещении мегаполисов // Автоматика и телемеханика. – 2016. – № 11. – с. 96–117.
19. *Ченцов А. Г.* К вопросу о маршрутизации комплексов работ // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. – 2013. – №1– с. 59–82. – ISSN 1994-9197.
20. *Ченцов А. Г.* Задача последовательного обхода мегаполисов с условиями предшествования // Автоматика и телемеханика. – 2014. –№ 4. – с. 170–190.
21. *Ченцов А. Г.* Одна параллельная процедура построения функции Беллмана в обобщенной задаче курьера с внутренними работами // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. – 2012. – 18 (277). – ISSN 2071-0216.
22. *Lawler E. L.* Efficient implementation of dynamic programming algorithms for sequencing problems. – Stichting Mathematisch Centrum, 1979.
23. *Petunin A.* General Model of Tool Path Problem for the CNC Sheet Cutting Machines // IFAC-PapersOnLine. – 2019. – янв. – т. 52, № 13. – с. 2662–2667. – ISSN 2405-8963. – DOI: [10.1016/j.ifacol.](https://doi.org/10.1016/j.ifacol.2019.11.609) [2019.11.609](https://doi.org/10.1016/j.ifacol.2019.11.609).
24. *Petunin A.A*.
25. *Ченцов А*.*Г*., *Ченцов А.А., Сесекин А.Н. ….. Труды ИММ УрО РАН, 2020.N 3.*
26. *Chentsov A.A., Chentsov A.G., Sesekin A.N.* An Extremal Routing Problem with Constraints and Complicated Cost Functions. 2021. River Publishers. Chapter 2. P.21-52.