SG-3.3.3.1") [Scientific and technical report (interim) on the topic "Development of technology and creation of experimental software systems for controlling the configuration of onboard systems of small-mass spacecraft in the interests of increasing their survivability" (code" Technologia-SG-3.3.3.1")]. Saint Petersburg: SPIIRAN, 2018, 450 p.
22. *Kokorin S.V., Potryasaev S.A., Sokolov B.V.* Kombinirovannyy metod planirovaniya operatsiy i raspredeleniya resursov sistemy upravleniya aktivnymi podvizhnymi ob"ektami [Combined method of planning operations and resource allocation of the active mobile objects management system], *Izvestiya vuzov. Priborostroenie* [Izvestiya vuzov. Instrumentation], 2012, Vol. 55, No. 11, pp. 17-22.
23. *Potryasaev S.A.* Kompleksnoe modelirovanie slozhnykh protsessov na osnove notatsii BPMN [Complex modeling of complex processes based on BPMN notation], *Priborostroenie* [Instrumentation], 2016, No. 11, pp. 913-920.

Статью рекомендовал к опубликованию д.т.н. В.И. Пименов.

Кофнов Олег Владимирович – Санкт-Петербургский федеральный исследовательский центр Российской академии наук; e-mail: kofnov@mail.ru; г. Санкт-Петербург, Россия; тел.: +79219413288; лаборатория информационных технологий в системном анализе и моделировании; к.т.н.; с.н.с.

Потрясаев Семен Алексеевич – e-mail: spotryasaev@gmail.com; лаборатория информационных технологий в системном анализе и моделировании; д.т.н.; с.н.с.

Соколов Борис Владимирович – e-mail: sokolov_boris@inbox.ru; тел.: +79217918136; лаборатория информационных технологий в системном анализе и моделировании; д.т.н.; профессор; Заслуженный деятель науки РФ; руководитель лаборатории.

Трефилов Петр Михайлович – Федеральное государственное бюджетное учреждение науки Институт проблем управления им. В.А. Трапезникова Российской академии наук; e-mail: petertrfi@gmail.com; Москва, Россия; лаборатория №80 «Киберфизических систем»; аспирант.

Kofnov Oleg Vladimirovich – St. Petersburg Federal Research Center of the Russian Academy of Sciences, e-mail: kofnov@mail.ru; Saint Petersburg, Russia; phone: +79219413288; laboratory of information technology in system analysis and modeling, senior researcher, cand. of eng. sc.

Potriasaev Semyon Alekseevich – e-mail: spotryasaev@gmail.com; laboratory of information technology in system analysis and modeling, senior researcher; dr. of eng. sc.

Sokolov Boris Vladimirovich – e-mail: sokolov_boris@inbox.ru; phone: +78127918136; laboratory of information technology in system analysis and modeling; dr. of eng. sc.; professor; head of laboratory.

Trefilov Peter Michailovich – V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences; e-mail: petertrfi@gmail.com; Moscow, Russia; laboratory 80; postgraduate student.

УДК 519.8.А DOI 10.18522/2311-3103-2021-1-149-165

А.А. Петунин, Е.Г. Полищук, С.С. Уколов

НОВЫЙ АЛГОРИТМ ПОСТРОЕНИЯ КРАТЧАЙШЕГО ПУТИ ОБХОДА КОНЕЧНОГО МНОЖЕСТВА НЕПЕРЕСЕКАЮЩИХСЯ КОНТУРОВ НА ПЛОСКОСТИ*

Рассматривается проблема маршрутизации режущего инструмента машин листовой резки с ЧПУ для случая, когда точки врезки расположены на границах деталей, ограниченных отрезками прямых и дугами окружностей, при этом используется техника непре-

* Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ, Государственный контракт № 075-03-2020-582/4.

рывной резки (CCP), т.е. каждый контур вырезается целиком, но не используется предварительная дискретизация, то есть резка может начинаться с любой точки контура. Общая задача поиска оптимального маршрута в этом случае сводится к минимизации длины холостого хода. Показано, что она эквивалентна поиску кратчайшей ломаной с вершинами, расположенными на контурах. Предложен новый эвристический алгоритм построения такой ломаной для заранее заданного порядка обхода контуров. Показано, что получающееся решение представляет собой локальный минимум. Описаны некоторые достаточные условия, того, что решение является также глобальным минимумом, которые легко проверяются численно, а некоторые даже визуально. Описана методика автоматического учёта ограничений предшествования для практически важного случая наличия вложенных контуров, возникающих как за счёт отверстий в деталях, так и за счёт расположения мелких деталей в отверстиях крупных. При этом происходит также уменьшение размерности задачи, что положительно сказывается на времени оптимизации, особенно дискретной. Предложен эвристический алгоритм выбора порядка обхода контуров на основе метода переменных окрестностей (VNS). Описаны альтернативные подходы применения других методов дискретной оптимизации совместно с предложенным алгоритмом построения кратчайшей ломаной для решения полной задачи непрерывной резки и возникающие при этом сложности как теоретического, так и практического характера. Описано обобщение задачи непрерывной резки до более широкого класс задач сегментной резки и обобщённой сегментной резки, что позволяет продвинуться в решении общей задачи прерывистой резки. Описана схема применения предложенного алгоритма для решения задач сегментной и обобщённой сегментной резки. Приведены некоторые результаты численных экспериментов в сравнении с точным решением задачи для дискретной модели GTSP.

Задача резки; непрерывная резка; оптимизация; достаточные условия; эвристика; GTSP; метод переменных окрестностей/

A.A. Petunin, E.G. Polishchuk, S.S. Ukolov

A NEW ALGORITHM FOR CONSTRUCTING THE SHORTEST TOUR OF A FINITE SET OF DISJOINT CONTOURS ON A PLANE

The problem of tool path routing for the CNC thermal cutting machines is considered. Pierce points are located at the parts bounding contours, consisting of straight-line segments and circular arcs. Continuous cutting technique is used, each contour is cut out entirely, and no presampling occurs, so cutting can start from any point on the contour. General problem of minimizing the route length is reduced to minimizing the air move length. It is shown to be equivalent to finding the shortest polyline with vertices on the contours. New algorithm for constructing such a broken line for fixed order of contour traversing is proposed. The resulting solution is shown to be a local minimum. Some sufficient conditions are described for the it to be also a global minimum, which can be easily verified numerically, and some even visually. A technique is described for automatically taking into account precedence constraints for the practically important case of nested contours. This also decreases the size of the problem, which has a positive effect on the optimization time. A heuristic routing algorithm based on the variable neighborhood search (VNS) is proposed. Alternative approaches to the use of other discrete optimization methods along with the proposed algorithm for constructing the shortest polyline for solving the complete problem of continuous cutting, and the resulting difficulties of both theoretical and practical nature are described. The generalization of the problem of continuous cutting to a wider class of problems of (generalized) segment cutting is described, which makes it possible to advance in solving the problem of intermittent cutting. The scheme of application of the proposed algorithm for solving problems of generalized segment cutting is described. The results of numerical experiments are considered in comparison with the exact solution of the GTSP problem.

Cutting problem; continuous cutting; optimization; sufficient conditions; heuristics; GTSP; variable neighborhood search.

Введение. В процессе разработки управляющих программ для машин листовой резки листового материала с ЧПУ возникает несколько оптимизационных задач. Одна из них – минимизация длины холостого хода инструмента. Она (в некоторых случаях) может быть сведена к задаче поиска кратчайшей ломаной, вершины кото-

рой расположены на заданных плоских контурах, являющихся границами деталей. Расположение этих контуров на плоскости в свою очередь получено решением другой оптимизационной задачи – «раскроя». Обе упомянутые задачи являются NP-полными.

Задача минимизации длины холостого хода инструмента сама по себе является частным случаем общей оптимизационной проблемы маршрутизации инструмента. Её полное решение в общем случае не может быть получено в разумное время для задач типичных для современного промышленного производства (сотни граничных контуров), поэтому на практике применяются разнообразные эвристики, дающие решения приемлемого качества.

Для полного решения задачи маршрутизации режущего инструмента для машин листовой резки с ЧПУ для минимизации времени или стоимости резки требуется решить целый ряд задач. Их описание и классификация приведены подробно в [1–3], и схематически изображены на рис. 1.

♦ **Задача непрерывной резки** (Continuous Cutting Problem, CCP): каждый контур (ограничивающий одну из деталей) вырезается за один раз, одним движением инструмента, но резка может начаться в любой точке контура (и заканчивается в ней же).

♦ **Обобщённая задача коммивояжера** (Generalized Traveling Salesman Problem, GTSP): резка может начаться в одной из заранее заданных точек на контуре (количество таких точек конечно), после этого контур вырезается целиком.

♦ **Задача резки с остановками** (Endpoint Cutting Problem, ECP): резка контура может начинаться только в заранее заданных точках на нём, но контур может вырезаться за несколько раз, частями.

♦ **Сегментная задача непрерывной резки** (Segment Continuous Cutting Problem, SCCP): вводится понятие сегмента как обобщение понятия контура; сегмент может быть частью контура или объединением нескольких контуров и / или их частей. Каждый сегмент вырезается целиком, от начала до конца, таким образом $CCP \subset SCCP$.

♦ **Обобщённая сегментная задача непрерывной резки** (Generalized Segment Continuous Cutting Problem, GSCCP): подобна сегментной задаче непрерывной резки (SCCP), но разбивка на сегменты не задана заранее, и сама подлежит оптимизации.

♦ **Задача прерывистой резки** (Intermittent Cutting Problem, ICP): наиболее общая формулировка задачи резки, встречающаяся в научной литературе, контуры могут вырезаться частями, в несколько подходов, начиная с произвольной точки.

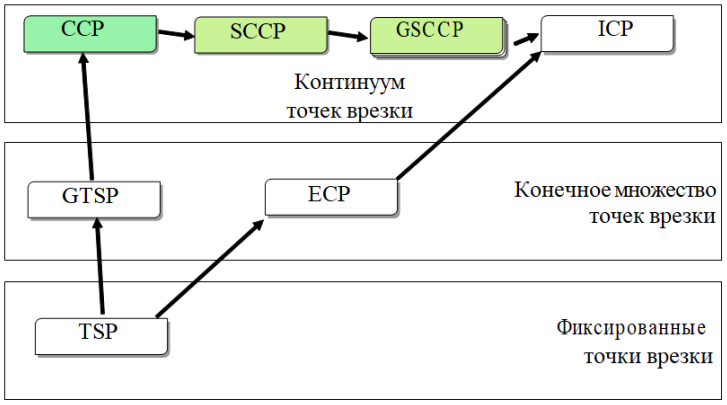


Рис. 1. Классификация задач резки

На практике задача оптимизации маршрута режущего инструмента зачастую сводится к дискретной оптимизации за счёт выбора конечного множества возможных точек врезки на контурах деталей с некоторым заранее заданным шагом ε, то есть сводится к задаче ECP [4–6] и её частному случаю – GTSP [7–10]. Задача непрерывной резки CCP тоже может сводиться к GTSP, в этом случае суммарная ошибка длины холостого хода оценивается как $N \cdot \varepsilon$, где N – количество контуров. Для достижения точности результата δ, таким образом необходимо выбирать малое $\varepsilon \approx \delta/N$, так что полное количество допустимых точек врезки растёт (как $O(N)$) и полный перебор требует экспоненциального времени. Тем не менее, этот класс задач может успешно решаться, например, средствами динамического программирования (DP) а для небольших $N \approx 30$ – даже точно (см, в частности [11]).

В данной работе рассматриваются вопросы поиска оптимального маршрута без применения дискретизации (то есть задача CCP), которые слабо освещены в открытых источниках, см. например [12; 13], где описаны некоторые эвристики.

Технологические ограничения. Тот факт, что полученный маршрут должен быть физически выполнен на конкретной машине листовой резки с ЧПУ, накладывает на последний определённые технологические ограничения.

Так называемое «ограничение предшествования» (наиболее подробно описанное в литературе), вызвано тем, что после вырезания замкнутого контура, его содержимое более ничем не удерживается и может свободно вращаться, сдвигаться и даже падать. Поэтому внутренние контура деталей должны вырезаться до того, как будет завершена резка содержащих их внешних контуров. Аналогично и детали, размещённые в отверстиях других деталей, также должны вырезаться до завершения резки содержащих их контуров.

Наконец, технология резки в большинстве случаев диктует, что резак не может двигаться прямо по контуру детали, но с некоторым сдвигом (несколько миллиметров). Этот сдвиг может вычисляться как в ходе решения задачи маршрутизации, так и после – при генерации управляющей программы на основе полученного маршрута или даже непосредственно на станке в процессе резки. Более того, точки врезки (в которых начинается резка) как правило должны находиться на ещё большем расстоянии от контуров деталей во избежание повреждения последних. В данной работе эти вопросы, тем не менее, не рассматриваются, то есть строится маршрут, проходящий в точности по контурам деталей, и точки врезки (а равно и точки окончания резки и выключения инструмента) также ищутся прямо на контурах деталей.

1. **Задача непрерывной резки.** Рассмотрим Эвклидову плоскость $\mathbb{R}^2$ и на ней фигуру B (в большинстве практических случаев – прямоугольник), ограниченную замкнутым контуром. Это – модель листового материала, подлежащего резке. Пусть N попарно непересекающихся плоских контуров $\{C_1, C_2, \dots C_N\}$ расположены внутри B, ограничивая n деталей $\{A_1, A_2 \dots A_n\}$. 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 листовой 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. [14].

Далее, внутри 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 $\overline{M_i} \in C_i, i \in \overline{1, N}$
2. Последовательности обхода контуров C_i, то есть перестановки N элементов $I = (i_1, i_2, \ldots i_N)$

Результатом решения задачи будет являться маршрут

$$\{M_0, M_{i_1}, M_{i_2}, \ldots M_{i_N}, M_{N+1}\}. \quad (1)$$

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

$$\mathcal{L} = \sum_{j=0}^{N} |M_{i_j} M_{i_{j+1}}| \quad (2)$$
$$\mathcal{L} \to min,$$

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: $\tilde{C_a} \subset \tilde{C_b}$, где $\tilde{C_a}$ обозначает 2-мерную фигуру, ограниченную контуром C_a (в более традиционных обозначениях $C_a = \partial \tilde{C_a}$).

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: $\tilde{C_a} \subset \tilde{C_b} \Rightarrow i_a < i_b$, в перестановке $I = (i_1, i_2, \ldots i_N)$. Таким образом, множество допустимых перестановок ограничено.

2. Алгоритм CCP-Relax решения задачи непрерывной резки. Предлагаемый алгоритм решения задачи непрерывной резки (см. [15]) состоит из нескольких шагов, что хорошо соответствует самой природе решаемой задачи.

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:

$$\{C_i | \forall j \neq i : C_j \cap \tilde{C_i} = \emptyset\}.$$

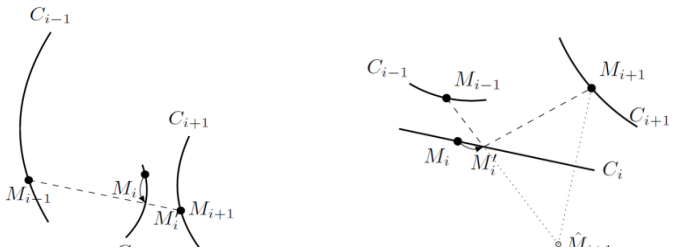
В общем случае это приводит к уменьшению (в некоторых случаях – существенному) сложности задачи (с N до некоторого N'), что в свою очередь сокращает время счёта на втором и в особенности третьем шагах алгоритма.

2.2. ***Непрерывная оптимизация.*** На этом этапе мы полагаем, что последовательность обхода контуров $I = (i_1, i_2, \ldots i_N)$ задана (фиксирована) и ищем координаты точек врезки $M_i \in C_i$ во все контуры, минимизируя полную длину холостого хода (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) при этом постоянны, так что сама функция упрощается до

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

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), рис. 2,a.

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), рис. 2,b.



a b

Рис. 2. Оптимальное положение точки врезки: a – на пересечении звена ломаной, b – с использованием принципа Ферма

Общая схема этого шага оптимизации может быть записана таким образом:

1. Выбираем произвольные начальные позиции точек врезки $M_i \in C_i, \forall i$.
2. $\forall i \in \overline{1,N}$ находим оптимальное положение M_i как описано выше за константное время.
3. Повторяем предыдущий шаг, до тех пор, пока длина холостого пути (2) не сойдётся (с некоторой наперёд заданной точностью δ).

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

2.3. ***Дискретная оптимизация.*** Наиболее вычислительно сложная задача заключается в поиске перестановки $I = (i_1, i_2, \ldots i_N)$, 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, см. [16]) по такой схеме:

1. Начальная перестановка $I = (i_1, i_2, \ldots i_N)$ выбирается произвольным образом (например, случайно)
   2. $k = 1$
3. while $k < k_{max}$:
   3.1. Выбираем перестановку $I' \in \mathcal{N}^k(I)$ из окрестности $\mathcal{N}^k(I)$, доставляющую минимум $\mathcal{L}(I')$
   3.2. if $\mathcal{L}(I') < L(I)$:
      3.2.1. $I \leftarrow I'$
      3.2.2. $k \leftarrow 1$
   3.3. else
      3.3.1. $k \leftarrow k + 1$
4. Завершение работы.

На шаге 3.1 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 $I = (i_1, i_2, \ldots i_N)$ 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 $I = (i_1, i_2, \ldots i_N)$.
- 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 3.1) на «первый подходящий» или метод Монте-Карло, но их влияние на скорость и качество получаемого решения задачи непрерывной резки требует дальнейшего исследования.

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

$$Route = \{M_0, M_1, M_2, \ldots M_{N'}, M_{N'+1}\} \qquad (3)$$

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 (3), и если таких точек несколько (что, как правило, и бывает), выбираем из них са-

мую последнюю, то есть посещаемую маршрутом позже всех других, см. рис. 3. Сам контур C_i вставляется в перестановку в месте, соответствующем выбранной точке врезки.

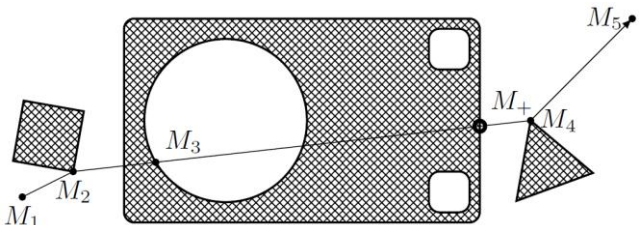


Рис. 3. Добавление точек врезки во «внешние» контура M_+

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.

3. Условия оптимальности решения непрерывной задачи. 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.

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.

На рис. 4 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.

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).

Мы начинаем с (произвольной) ломаной линии L_1. Далее для каждого контура C_i мы повторяем следующую операцию: сдвигаем вершину ломаной M_i C_i в такое положение, которое минимизирует полную длину ломаной, при этом ос-

тальные 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.

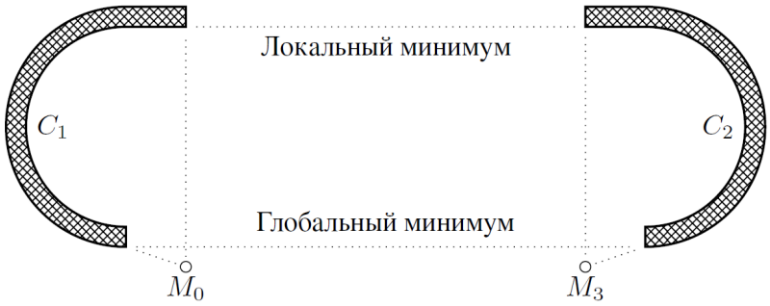


Рис. 4. Два маршрута резки, доставляющие локальный и глобальный минимум

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.

3.1. Локальный минимум. П р е д л о ж е н и е 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

Д о к а з а т е л ь с т в о. 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 $M_i(s) = M_i + s \cdot \vec{M_iM_i'}, M_{i+1}(t) = M_{i+1} + t \cdot \vec{M_{i+1}M_{i+1}'}$ $(s, t \in [0,1])$, $f(s,t) = |M_{i-1}M_i(s)M_{i+1}(t)M_{i+2}|$.

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

$$\frac{\partial f(s,t)}{\partial s}|_{(0,0)} \geqslant 0, \frac{\partial f(s,t)}{\partial t}|_{(0,0)} \geqslant 0. \quad (4)$$

Если, например, $\partial f(s,t)/\partial s|_{(0,0)} = 0$, 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

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

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 (4) не равна нулю. Обозначим $\varphi(t) = f(t,t)$.

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

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

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.

3.1. ***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 1. Пусть выполняется **одно** из следующих условий:

1. Сегмент $M_{i-1}M_{i+1}$ пересекает контур C_i, то есть $M_i \in M_{i-1}M_{i+1}$
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.

Предложение 2. Если ***Условие 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 $M_i(s) = M_i + s\cdot\vec{M_iM_i'}, M_{i+1}(t) = M_{i+1} + t\cdot\vec{M_{i+1}M_{i+1}'}(s,t \in [0,1]), f(s,t) = |M_{i-1}M_i(s)M_{i+1}(t)M_{i+2}|$.

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

$$\frac{\partial f(s,t)}{\partial s}|_{(0,0)} \geqslant 0, \frac{\partial f(s,t)}{\partial t}|_{(0,0)} \geqslant 0. \quad (5)$$

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 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. Выполняется **любое** из условий:

1. Сегмент $M_{i-1}M_{i+1}$ пересекает контур C_i: $M_i \in M_{i-1}M_{i+1}$
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 1**; иначе существовала бы лучшая вершина $M_i' \in C_i$).
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
4. Если контур C_i ограничивает собой выпуклый многоугольник $\tilde{C}_i$

4. Новый подход к задаче прерывистой резки (ICP). Задача прерывистой резки – самая общая и сложная из всех классов задач резки. Её изучение и решение представляется полезным как с теоретической точки зрения, так и для решения практических задач современного производства.

Кроме стандартной техники резки (которая фактически приводит к задаче непрерывной резки), используются и другие техники резки, например, «мульти-сегментная» и «мульти-контурная» резка. В первом случае один контур может вырезаться в несколько подходов, по частям. Во втором наоборот, несколько контуров вырезаются одним движением резака, рис. 5.

Для того, чтобы можно было описать эти техники резки в используемой математической модели, мы вводим новые понятия:

Сегмент резки $S = M\vec{M}^*$ – это траектория инструмента от точки врезки M до соответствующей точки выключения инструмента M^*.

В стандартной технике сегмент резки содержит ровно один контур детали, но в общем случае это уже не так. При мульт-сегментной резки сегмент содержит только часть контура, а при мульти-контурной – несколько контуров (или их частей). Рис. 5, таким образом, может представлять не только пример мульти-контурной резки, но также и один единственный сегмент в составе некоторой большей задачи резки.

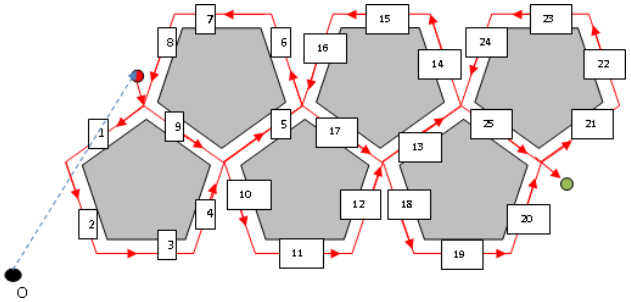


Рис. 5. Пример составного сегмента резка для шести деталей / контуров

Поскольку сегмент резки по определению содержит информацию о направлении резки, нам потребуется ещё одно понятие:

Базовый сегмент B^S – это часть сегмента резки $S = \vec{MM}^*$, без начальной и конечной части (вспомогательных движений резака, когда он движется от точки врезки к собственно контуру и отходит от вырезанного контура в конце). Базовый сегмент не содержит информации о направлении резки и является чисто геометрическим объектом.

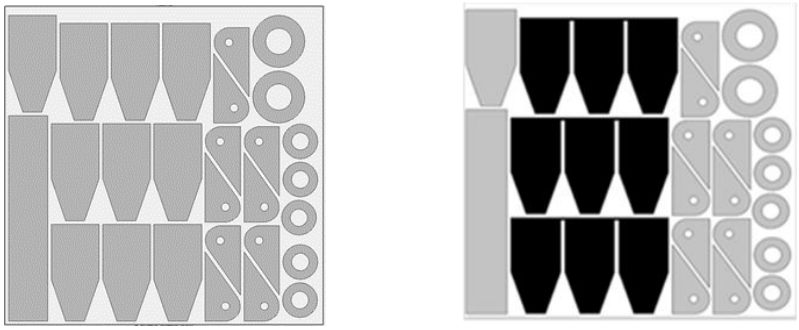
При помощи понятия базового сегмента оказывается возможным сформулировать обобщение задачи непрерывной резки:

Сегментная задача непрерывной резки (Segment Continuous Cutting Problem, SCCP) – это задача резки фиксированного набора базовых сегментов резки $SCCP = \{B^{S_i}\}$.

Описанный в разделе 2 алгоритм CCP-Relax естественным образом обобщается на решение задачи SCCP.

Заметим теперь, что если задана раскройная карта, то есть расположение всех деталей и их граничных контуров, из неё может быть сгенерирован целый *ансамбль* наборов базовых сегментов путём разбиения контуров на сегменты и объединения нескольких сегментов в один. См., например, рис. 6, где мульти-контурные сегменты выделены чёрным цветом. Это позволяет ещё больше обобщить задачу и сформулировать новый класс:

Обобщённая задача сегментной резки (Generalized Segment Continuous Cutting Problem, GSCCP) – *ансамбль* нескольких задач сегментной резки (SCCP) для одного раскройного плана: $GSCCP = \{SCCP_i\}$.



a b

Рис. 6. Ансамбль задач сегментной резки: a – стандартная резка, 45 сегментов, b – мульти-контурная резка, 39 сегментов

Введение нового класса задач резки GSCCP существенно расширяет существующую классификацию задач маршрутизации инструмента машин листовой резки с ЧПУ. Фактически, задачи SCCP и GSCCP являются частными случаями задачи ICP, содержащими конечное множество базовых сегментов резки: $CCP \subset SCCP \subset GSCCP \subset ICP$.

Общая схема решения задачи GSCCP. Предполагая ансамбль $\underline{SC}CP_i$ наборов базовых сегментов $SCCP_i = \{B^{S_j}\}$, $i \in \overline{1,T}, j \in \overline{1,K_i}$ фиксированным, используем следующую общую схему решения задачи GSCCP:

♦ Каждая из задач $SCCP_i$ решается независимо любым существующим алгоритмом, например:

1. *CCP-Relax*, описанная в разделе 2 эвристика.
2. *DP-GTSP*, точный алгоритм динамического программирования для задач небольшого размера, см. [11].

3. *Greedy-GTSP*, итеративный жадный эвристический алгоритм, см. [17].

Для алгоритмов дискретной оптимизации контуры деталей предварительно преобразуются в конечные множества допустимых точек врезки, как показано на рис. 7.

♦ Лучшее решение выбирается на основе значения целевой функции (2).

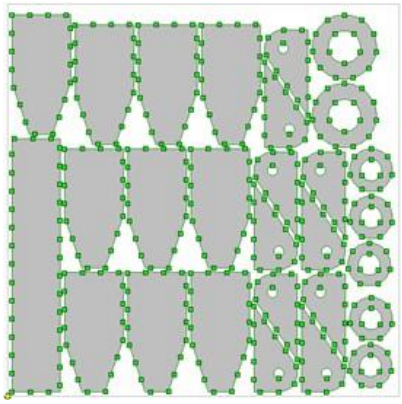
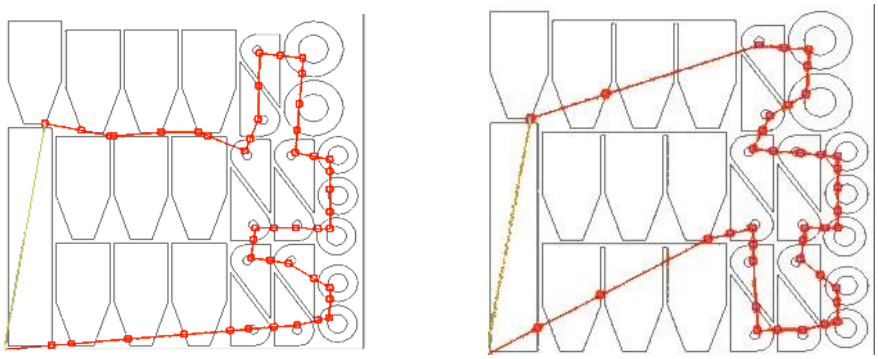


Рис. 7. Задача GTSP, полученная по задаче (S)CCP на рис. 6, 425 точек врезки

Рис. 8 показывает решения задач SCCP с рис. 6, полученные при помощи алгоритма *CCP-Relax*. Легко видеть, что маршруты действительно различаются. Более того, с практической точки зрения, различие может быть ещё более значительным, так как маршруты отличаются не только геометрически, но и количеством точек врезки, а эта операция сравнительно дорогая для современных машин резки с ЧПУ, как в смысле стоимости, так и времени работы.



a b

Рис. 8. Решение задачи GSCCP с рис. 6: a – Стандартная резка, b – Мульти-контурная резка

5. Численные эксперименты. Оценка качества решений, получаемых алгоритмом CCP-Relax проводилась на нескольких раскройных планах, содержащих реальные детали. В качестве базы сравнения был взят точный алгоритм решения задачи GTSP на основе динамического программирования (см. [11]), гарантирующий нахождение наилучшего решения для задач с небольшим количеством контуров, а также специализированная версия эвристики GNLS [18].

На рис. 9 представлено точное решение, при этом видны допустимые точки врезки, полученные дискретизацией исходных контуров деталей. На рис. 10 представлено решение задачи CCP, полученное алгоритмом CCP-Relax.

Можно отметить, что полученные маршруты практически идентичны. Небольшие отклонения вызваны процессом предварительной дискретизации (при переводе задачи CCP в GTSP). Вследствие этого алгоритм CCP-Relax формирует прямые сегменты холостого хода инструмента, а в случае задачи GTSP получаются слегка ломаные линии, что слегка увеличивает результирующую длину маршрута. Более точно это отражено в табл. 1 для нескольких раскройных планов.

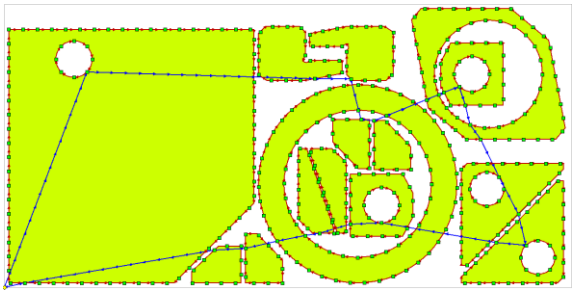


Рис. 9. Точное решение задачи GTSP, задание № 464

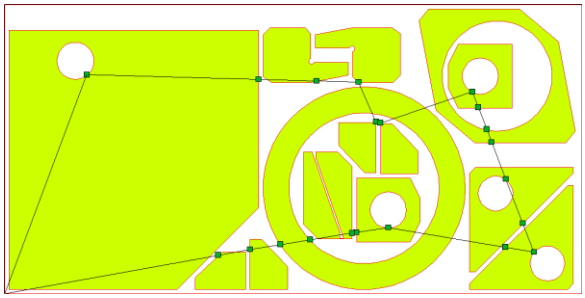


Рис. 10. Решение задачи CCP, задание № 464

Таблица 1

Сравнение решений задач CCP и GTSP

Задание	№ 229	№ 464	№ 3211	№ 20205
Деталей	11	14	17	115
Контуров	12	21	22	198
Точек GTSP	491	429	493	3917
L_{GTSP} , м	7.729	4.743	4.557	26.098
L_{CCP} , м	7.727	4.706	4.536	25.987

На рис. 11 показано решение задачи CCP (алгоритмом CCP-Relax) для раскройного плана на 198 контуров, когда точное решение не известно. Вообще, для задач такого и большего размера (а именно они представляют практический интерес), оценка оптимальности решения представляет собой довольно большие сложности. Тем не менее, сравнение с решениями хорошо исследованной задачи GTSP вполне применимо для оценки качества решения. Как известно, GTSP является NP-сложной даже на Эвклидовой плоскости [19].

Хотя из общих соображений понятно, что чем больше ограничений (имеются в виду ограничения предшествования) наложено на задачу GTSP, тем проще её решить, вопрос влияния уровня вложенности деталей на границы сложности решений ещё не достаточно исследовался и заслуживает внимания. По этому поводу хочется отметить работы [20, 21]. Для двух видов ограничений предшествования теоретически доказана полиномиальная временная сложность задачи GTSP. Один тип описан Е. Баласом в [22] для классической задачи TSP. Эффективный точный алгоритм для задачи GTSP с такими ограничениями предшествования предложен в работах [23, 24]. Маршруты, содержащие ограничения второго вида, называются квази- и псевдо-пирамидальными. Эффективные параметрические алгоритмы решения задачи GTSP с ограничениями такого вида предложены в [25, 26].

Ввиду вышеизложенного, можно заключить, что алгоритмический анализ даже задачи GTSP в приложении к маршрутизации резки всё ещё является малоисследованной областью. В частности, отсутствие эффективных моделей MILP (Mixed Integer Linear Program) для задачи GTSP делает невозможным использование современных решателей, таких как Gurobi [27] для построения верхних и нижних границ и оценки эвристических решений. Это также представляется перспективным направлением исследований.

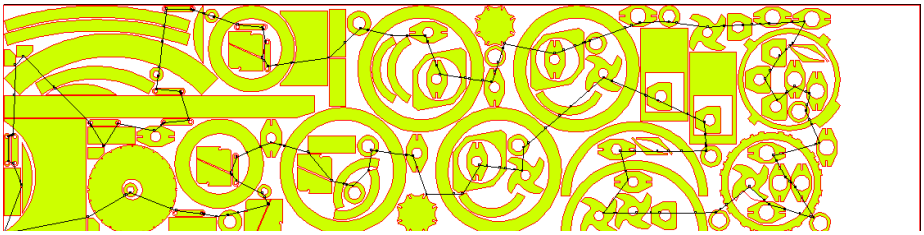


Рис. 11. Пример решения задачи CCP большого размера, задание № 20205

Заключение.

1. Задача минимизации длины холостого хода инструмента машин листовой резки с ЧПУ для задачи непрерывной резки сведена к аналогичной задаче, не содержащей ограничений предшествования, сокращая количество контуров и время работы алгоритма
2. Предложен эвристический алгоритм CCP-Relax поиска положений точек врезки в задаче непрерывной резки
3. Доказано, что алгоритм CCP-Relax при заданном порядке обхода контуров доставляет локальный минимум длины холостого хода
4. Сформулировано несколько легко проверяемых условий, при которых решение, полученное алгоритмом CCP-Relax, является глобальным минимумом.
5. Алгоритм CCP-Relax может применяться также для решения более широкого класса задач резки – SCCP (Сегментная резка) и GSCCP (Обобщённая сегментная резка), что в свою очередь является перспективным подходом к решению общей задачи ICP.
6. Ближайшим направлением дальнейших исследований является обобщение алгоритма CCP-Relax на более практический случай, когда точки врезки лежат вне контуров деталей в соответствии с технологическими требованиями современного оборудования с ЧПУ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Hoeft J., Palekar U. S.* Heuristics for the plate-cutting traveling salesman problem, *IIE Transactions*, 1997, Vol. 29, No. 9, pp. 719-731. ISSN 1573-9724. Doi: 10.1023/A:1018582320737.
2. *Dewil R., Vansteenwegen P., Cattrysse D.* A review of cutting path algorithms for laser cutters, *International Journal of Advanced Manufacturing Technology*, 2016, Vol. 87, No. 5, pp. 1865-1884. ISSN 1433-3015. Doi: 10.1007/s00170-016-8609-1.
3. *Petunin A.A., Stylios C.* Optimization Models of Tool Path Problem for CNC Sheet Metal Cutting Machines, *IFAC-PapersOnLine*, 2016, Vol. 49, No. 12, pp. 23-28.
4. *Dewil R., Vansteenwegen P., Cattrysse D.* Construction heuristics for generating tool paths for laser cutters, *International Journal of Production Research*, 2014, Vol. 52, No. 20, pp. 5965-5984.
5. *Sherif S.U., Jawahar N., Balamurali M.* Sequential optimization approach for nesting and cutting sequence in laser cutting, *Journal of Manufacturing Systems*, 2014, Vol. 33, No. 4, pp. 624-638.
6. *Imahori S. [et al.]*. Generation of cutter paths for hard material in wire EDM, *Journal of Materials Processing Technology*, 2008, Vol. 206, No. 1, pp. 453-461. ISSN 0924-0136. Doi: 10.1016/j.jmatprotec. 2007.12.039.
7. *Chentsov A. G., Chentsov A. A.* A Discrete–Continuous Routing Problem with Precedence Constraints, *Proceedings of the Steklov Institute of Mathematics*, 2018, Vol. 300, No. 1, pp. 56-71. ISSN 1531- 8605. Doi: 10.1134/S0081543818020074.
8. *Petunin A.A. [et al.]*. Elements of dynamic programming in local improvement constructions for heuristic solutions of routing problems with constraints, *Automation and Remote Control*, 2017, Vol. 78, No. 4, pp. 666-681. ISSN 1608-3032. Doi: 10.1134 / S0005117917040087.
9. Ye J., Chen Z.G. An Optimized Algorithm of Numerical Cutting-Path Control in Garment Manufacturing, *Advanced Materials Research*, 2013, Vol. 796, pp. 454-457. ISSN 1662-8985. Doi: 10.4028/ www.scientific.net/AMR.796.454.
10. Yu W., Lu L. A route planning strategy for the automatic garment cutter based on genetic algorithm, *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 379-386. ISSN 1941-0026. Doi: 10.1109/CEC.2014.6900425.
11. *Chentsov A.G. [et al.]*. Model of megalopolises in the tool path optimisation for CNC plate cutting machines, *International Journal of Production Research*, 2018, Vol. 56, No. 14, pp. 4819-4830. ISSN 0020-7543. Doi: 10.1080/00207543.2017.1421784.
12. *Arkin E.M., Hassin R.* Approximation algorithms for the geometric covering salesman problem, *Discrete Applied Mathematics*, 1994, Vol. 55, No. 3, pp. 197-218. ISSN 0166-218X. Doi: 10.1016/0166- 218X(94)90008-6.
13. *Vicencio K., Davis B., Gentilini I.* Multi-goal path planning based on the generalized Traveling Salesman Problem with neighborhoods, *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2985-2990. ISSN 2153-0866. Doi: 10.1109/IROS.2014.6942974.
14. *Dror M. [et al.]*. Touring a sequence of polygons, *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, ACM*, 2003, pp. 473-482.
15. *Petunin A.A., Polishchuk E.G., Ukolov S.S.* On the new Algorithm for Solving Continuous Cutting Problem, *IFAC-PapersOnLine*, 2019, Vol. 52, No. 13, pp. 2320-2325. ISSN 2405-8963. Doi: 10.1016/j.ifacol.2019.11.552.
16. *Hansen P., Mladenović N., Moreno Pérez J. A.* Variable neighbourhood search: methods and applications, *Annals of Operations Research*, 2010, Vol. 175, No. 1, pp. 367-407. ISSN 1572-9338. Doi: 10.1007/s10479-009-0657-6.
17. *Petunin A. A., Chentsov A. G., Chentsov P. A.* About routing in the sheet cutting, *Bulletin of the South Ural State University, Series: Mathematical Modelling, Programming and Computer Software*, 2017, Vol. 10, No. 3, pp. 25-39. ISSN 2071-0216. DOI: 10.14529/mmp170303.
18. *Smith S. L., Imeson F.* GLNS: An effective large neighborhood search heuristic for the Generalized Traveling Salesman Problem, *Computers & Operations Research*, 2017, Vol. 87, pp. 1-19. ISSN 0305-0548. Doi: 10.1016/j.cor.2017.05.010.
19. *Papadimitriou C H.* Euclidean TSP is NP-complete, *Theoretical Computer Science*, 1977, Vol. 4, pp. 237-244.
20. *Chentsov A.G., Grigoryev A.M.* A Scheme of Independent Calculations in a Precedence Constrained Routing Problem,, *SpringerLink*, 2016, pp. 121-135. DOI: 10.1007/978-3-319-44914-2_10.
21. *Saliy Y. V.* Influence of predestination conditions on the computational complexity of solution of route problems by the dynamic programming method, *Bulletin of the Udmurt University. Maths. Mechanics. Computer Science*, 2014, No. 1, pp. 76-86. ISSN 1994-9197.

22. *Balas E.* New classes of efficiently solvable generalized Traveling Salesman Problems, *Annals of Operations Research*, 1999, Vol. 86, pp. 529-558. ISSN 1572-9338. Doi: 10.1023/A:1018939709890.
23. *Chentsov A.G., Khachai M.Y., Khachai D.M.* An exact algorithm with linear complexity for a problem of visiting megalopolises, *Proceedings of the Steklov Institute of Mathematics*, 2016, Vol. 295, No. 1, pp. 38-46. ISSN 1531-8605. Doi: 10.1134/S0081543816090054.
24. *Chentsov A., Khachay M., Khachay D.* Linear time algorithm for Precedence Constrained Asymmetric Generalized Traveling Salesman Problem, *IFAC-PapersOnLine*, 2016, Vol. 49, No. 12, pp. 651- 655. ISSN 2405-8963. Doi: 10.1016/j.ifacol.2016.07.767.
25. *Khachai M.Y., Neznakhina E.D.* Approximation Schemes for the Generalized Traveling Salesman Problem, *Proceedings of the Steklov Institute of Mathematics*, 2017, Vol. 299, No. 1, pp. 97-105. ISSN 1531-8605. Doi: 10.1134/S0081543817090127.
26. *Khachay M., Neznakhina K.* Complexity and approximability of the Euclidean generalized traveling salesman problem in grid clusters, *Annals of Mathematics and Artificial Intelligence*, 2020, Vol. 88, No. 1, pp. 53-69. ISSN 1573-7470. Doi: 10.1007/s10472-019- 09626-w.
27. *Gurobi Optimization*. Gurobi optimizer reference manual, 2020. Available at: http://www.gurobi.com.

Статью рекомендовал к опубликованию д.ф.-м.н., профессор А.Н. Сесекин.

Петунин Александр Александрович – Уральский Федеральный университет; e-mail: a.a.petunin@urfu.ru; г. Екатеринбург, Россия; д.т.н.; доцент; профессор.

Полищук Ефим Григорьевич – e-mail: e.g.polishchuk@urfu.ru; к.ф.-м.н.; доцент; с.н.с.

Уколов Станислав Сергеевич – e-mail: s.s.ukolov@urfu.ru; м.н.с.

Petunin Alexander Alexandrovich – Ural Federal University; e-mail: a.a.petunin@urfu.ru; Yekaterinburg, Russia; dr. of eng. sc.; associate professor; professor.

Polishchuk Efim Grigorievich – e-mail: e.g.polishchuk@urfu.ru; cand. of phys. and math. sc.; associate professor; senior researcher.

Ukolov Stanislav Sergeevich – e-mail: s.s.ukolov@urfu.ru; junior researcher.

УДК 62-93 DOI 10.18522/2311-3103-2021-1-165-174

А.В. Логунов, А.Л. Береснев

ВОЗМОЖНОСТИ ВИБРОАКУСТИЧЕСКОГО ИССЛЕДОВАНИЯ И ДИАГНОСТИКИ ПОДВЕСКИ ТРАНСПОРТНЫХ СРЕДСТВ

Работа посвящена проблеме диагностирования подвески транспортных средств. Проблема контроля состояния подвески сейчас наиболее актуально из-за постоянного роста автопарка и ужесточения требований к безопасной эксплуатации. Своевременный и точный контроль состояния подвески способен предотвратить выход из строя целых узлов транспортного средства, а также избежать таких серьезных последствий как дорожно-транспортное происшествие. В работе подробно рассмотрены современные средства диагностики, выделены принципы работы, достоинства и недостатки, представлено обоснование выбора из существующих методов такого, который способен помочь наиболее точно и быстро обнаружить неисправность. С появлением современных технологий давно известный метод оценки состояния подвески по звуку может стать самым передовым, поскольку исключается человеческий фактор, для обработки сигнала применяется вычислительная техника анализ звукового спектра в которой осуществляется с помощью компьютерных технологий. В статье рассмотрены механизмы, которые способны генерировать звуковые сигналы. Предложенный способ диагностики позволяет выделить «полезные» звуки из общего числа шумов подвески, после сравнительного анализа указать на узел звук которого отличается от эталонного, исправного. Данное решение в диагностике позволяет существенно снизить общую трудоемкость за счет исключения частичной или полной разборки подвески, как результат несмотря на упрощение, точность обнаружения