

Программное обеспечение для задачи построения траектории движения режущего инструмента

Т.А. Макаровских,
доц. каф. МиКМ, к.ф.-м.н., доц., kwark@mail.ru,
А.В. Панюков,
проф. каф. МиКМ, д.ф.-м.н., проф., a_ranyukov@mail.ru,
ЮУрГУ, г. Челябинск,
Е.А. Савицкий,
вед. инж.-прогр., kwark@mail.ru,
иностр. предпр. «АйБиЭй АйТи Парк», г. Минск, БЕЛОРУССИЯ

Большинство исследований по траекториям инструмента для режущих машин в основном посвящены определению траектории при контурной резке. Ресурсосберегающие технологии резки листовых материалов позволяют совмещать контуры вырезаемых деталей, что позволяет уменьшить количество отходов материала и сократить длину резки. Сокращение суммарной длины реза является источником еще одного набора ограничений предшествования, который можно формализовать как ограничение, называемое упорядоченным ограждением для плоских графов, являющихся гомеоморфными образами раскройного плана. В докладе рассмотрены общий случай задачи вырезания, когда допускается совмещение контуров и программное обеспечение, которое в соответствии с раскройным планом позволяет получить гомеоморфный образ графа для решения задачи маршрутизации, решает задачу для гомеоморфного образа и интерпретирует результаты решения.

Most of the research on tool paths for cutting machines is mainly devoted to determining the path during contour cutting. Resource-saving technologies of cutting sheet materials allow to combine the contours of the cut-out parts, which allows reducing the amount of waste material and reducing the length of the cutting. Reducing the total length of the cut is the source of another set of prediction constraints, which can be formalized as a constraint, called an ordered fence for flat graphs, which are homeomorphic images of the cutting plan. The report discusses the general case of the cutting problem, when contour alignment and software are allowed, which, according to the cutting plan, allows to obtain a homeomorphic image of the graph for solving the routing problem, solves the problem for the homeomorphic image and interprets the results of the solution.

Введение

Лазерная резка является одной из основных современных технологий, используемых при обработке листового материала, что делает актуальной задачу определения траектории движения режущего инструмента. Задача определения траектории заключается в определении точной последовательности резов. Развитие автоматизации производства привело к появлению технологического оборудования с числовым программным управлением, используемого для резки листовых материалов. Новые технологии позволяют осуществлять вырезание по произвольной траектории с достаточной для практики точностью.

Преимуществом при использовании лазерной резки является минимальность таких показателей как ширина реза и термические деформации.

Основными ограничениями при лазерной резке является:

- (1) все элементы внутренних контуров должны быть вырезаны прежде, чем будут полностью пройден охватывающий их контур (**OE**-охватывание [1]);
- (2) следует избегать пересечения траектории резки, касания допустимы (**NOE**-ограничение [2, 3]);
- (3) в процессе лазерной резки происходит нагревание металлического листа, поэтому, необходимо учитывать термальные эффекты [4];
- (3) ограничения на расположение точки врезки (построение **PPOE**-покрытия [5]);
- (4) общее время, требуемое на выполнение резки, представляющее суммарное время для осуществления всех вырезаний, времени на выполнение холостых переходов, времени на врезку и пр. желательно сокращать.

В [2] и [6] приводится классификация задач маршрутизации режущего инструмента и отмечается, что технологии ECP (Endpoint Cutting Problem) и ICP (Intermittent Cutting Problem) за счет возможности совмещения границ вырезаемых деталей позволяют сократить расход материала, длину резки и длину холостых проходов. Проблемы уменьшения отходов материала и максимального совмещения фрагментов контуров вырезаемых деталей решаются на этапе составления раскройного плана.

Несмотря на очевидные преимущества технологий ECP и ICP, в настоящее время большинство отечественных и зарубежных [1, 2, 7-13] работ посвящено развитию технологии GTSP (General Travelling Salesman Problem), которая не предполагает совмещение контуров вырезаемых деталей. Таким образом, при использовании данной технологии длина траектории будет равна сумме периметров всех контуров, а количество точек врезки – количеству контуров. Однако при этом проблема выполнения отмеченных выше технологических оказывается тривиальной.

Построению эффективных алгоритмов для раскройных планов, в которых допускаются совмещения границ контуров, посвящен ряд работ [14-16]. Последние достижения отражены в работах [3, 5, 17]. Целью данной работы является изложение техники программной реализации некоторых элементов CAD/CAM системы технологической подготовки процессов раскроя, связанных с нахождением маршрута режущего инструмента.

1. Представление исходных данных

Как правило, раскройный план содержит большие группы однотипных деталей, а для описания размещения детали на раскройном плане достаточно указания величин (x, y, φ) , где (x, y) – координаты базовой точки этой детали (обычно это начало координат, к которым привязаны координаты остальных точек), φ – угол поворота детали вокруг ее базовой точки. Поэтому разумно иметь базу данных типовых деталей.

Основными примитивными элементами траекторий маршрута резки являются отрезки прямых и дуги окружностей. Целая окружность представляется как объединение двух дуг с центральными углами φ и $2\pi - \varphi$, $\varphi > 0$. Для таких идентификации примитивов достаточно указать координаты (x_1, y_1) и (x_2, y_2) крайних точек v_1 и v_2 соответственно, а также $\tan(\varphi/4)$, где φ – центральный угол дуги (v_1, v_2) окружности, проходящей от точки v_1 до точки v_2 (очевидно, что для отрезка можно считать $\tan(\varphi/4) = 0$). С формальной точки плоская деталь представляет часть плоскости, ограниченной внешней границей и внутренними границами по числу дыр. Каждая граница представляет замкнутый контур, состоящий из такой последовательности примитивов, что начало следующего совпадает с концом предыдущего.

Для представления данных о детали используется формат JSON.

Пример раскройного плана с совмещенными границами контуров деталей представлен на рис. 1(а).

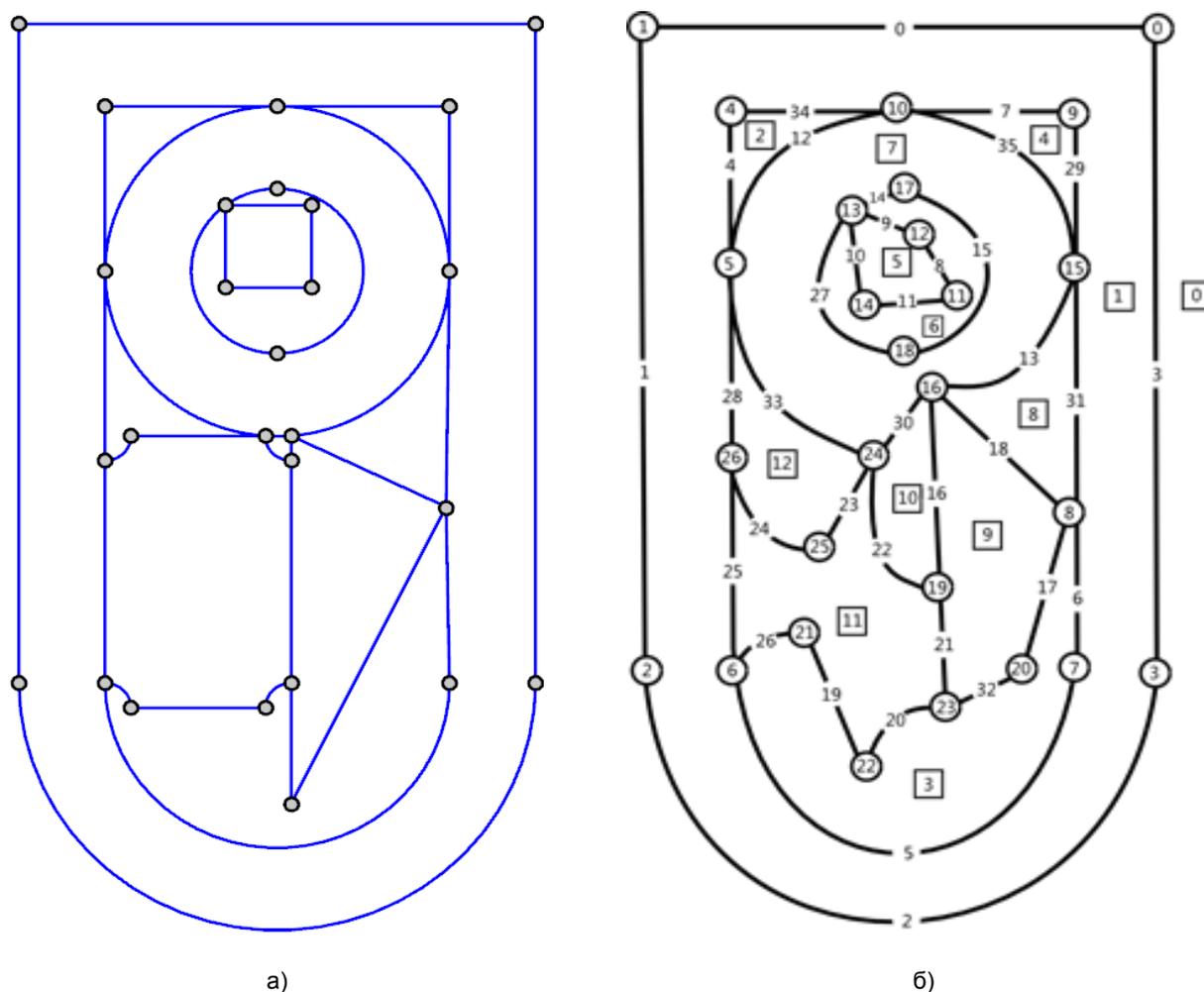


рис.1. Примеры:
 (а) раскройный план с совмещенными границами;
 (б) гомеоморфный образ раскройного плана с указанием вершин, ребер и граней

Для определения последовательности резки фрагментов раскройного плана не используется информация о форме детали, поэтому все кривые без самопересечений и соприкосновений в описании формы деталей, можно интерпретировать как ребра графа, а все точки пересечений и соприкосновений – как вершины графа. Для получения представления плоского графа, позволяющего восстановить с точностью до гомеоморфизма исходный раскройный план, необходимо и достаточно в каждой вершине зафиксировать циклический порядок на множестве инцидентных ей ребер. На рис. 1(б) представлен гомеоморфный образ раскройного плана, изображенного на рис 1(а).

Структуры данных, используемые для представления гомеоморфного образа, должны включать всю необходимую информацию для эффективной работы алгоритмов маршрутизации и интерпретации построенных маршрутов. На рис.2 представлены структуры данных для представления гомеоморфного образа раскройного плана в виде плоского графа.

```

struct Vert {
    double x,y; //Vertex coordinates
    int sv; //Connectivity component
    int rank;
    double dpth; //Vertex depth
    int deg; //Vertex degree
    int selected; //for UI (-1: no select; 0: mouse over; 1: clicked)
    int mark; //the mark used for wide-width search (isWay)
};
struct Edge { //Graph edge
    int v1, v2; //vertices id in vector V
    int sv; //Connectivity component
    int cycle1, cycle2;
    int f1, f2; // Face id
    int l1, l2, r1, r2; // Neighbors id-s in a cyclic order
    int rank;
    int dpth; //Depth of nesting (the corrected rank)
    double v1_ang, v2_ang; // Angle in radians from the 1st and 2nd vertices
    double v1_ta4, v2_ta4; //Tangent of the arc angle divided by 4
    int selected; //for UI (-1: no select; 0: mouse over; 1: clicked)
    int mark; // the mark used for wide-width search (isWay)
    double x_0, y_0, r; // The coordinates of an arc center and its radius
};
struct Face { //Graph face
    int number; //The number of face
    int dpth; // The depth of face
};

```

рис. 2. Структуры данных для представления гомеоморфного образа раскройного плана в виде плоского графа

Структура Vert содержит поля с указанием декартовых координат соответствующей точки на раскройном плане и ряд вспомогательных полей. Эти данные необходимы для интерпретатора маршрута и при заполнении полей структуры Edge.

Поля структуры Edge содержит номера $v1$, $v2$ инцидентных вершин, номера $f1$, $f2$ инцидентных граней, номера $l1$, $l2$, $r1$, $r2$ соседних в циклическом порядке ребер, значения $v1_ta4$, $v2_ta4$ величины $\tan(\varphi / 4)$, где φ – центральный угол дуги (v_1, v_2) и (v_2, v_1) соответственно, а также вспомогательные величины.

Контейнеры для приведенных структур, другие вспомогательные данные и все используемые методы их заполнения инкапсулированы в класс DataHolder.

2. Маршрутизация в связных графах

Использование плоского графа в качестве гомеоморфного образа модели режущего плана позволяет формализовать технологические ограничения на порядок вырезания фрагментов плана резки. Граф G содержит все возможные элементы траектории инструмента, тогда возникает следующая задача построения маршрутов, удовлетворяющих технологическим ограничениям: условие упорядоченного охватывания (OE -маршруты) [1] и определение траектории вырезания без самопересечений (NOE -маршруты) [3].

Определение OE -покрытия полностью конструктивно, доказательством этого факта является эффективность алгоритмов, рассмотренных в [1]. Если связный граф G не является эйлеровым, то он содержит вершины нечетной степени. В этом случае OE -маршрут состоит из k непересекающихся цепей. Задача построения такого маршрута решается алгоритмом OE -Router [17]. Если плоский граф G , представляющий образ раскройного плана, не содержит мостов (т. е. ребер инцидентных одной грани), то возможно построить OE -маршрут, в котором ребра произвольного паросочетания M и только они соответствуют холостым движениям. Отметим, что связанные плоские графы, являющиеся образами раскройных планов, не содержат мостов. Поэтому, если M является кратчайшим паросочетанием, то алгоритм M - OE -Router строит маршрут с минимальной длиной холостых переходов. Задача построения такого маршрута может быть решена алгоритмом M - OE -Router.

Итак, если граф G , представляющий образ раскройного плана, связан и не содержит мостов, то алгоритм M - OE -Router точно решает задачу, но требует определения кратчайшего паросочетания. Алгоритм OE -Router решает задачу для любого графа G , но, как известно, жадная стратегия выбора холостого хода является только 2-оптимальной.

3. Маршрутизация для несвязных графов

Часто раскройный план содержит детали с отверстиями, а также другие детали, расположенные в этих отверстиях и т. д. Плоский граф, представляющий гомеоморфный образ такого плана резки, оказывается несвязным. Для построения OE -покрытия в несвязном графе можно использовать следующие подходы: 1) определение допустимого обхода компонентов; 2) получение связного графа с использованием дополнительных ребер.

Первый подход позволяет получить очевидный алгоритм получения OE -маршрута несвязного графа. Основная идея этого алгоритма заключается в построении OE -маршрутов для каждого компонента отдельно по его рангу (в порядке уменьшения рангов компонентов). В этом случае решение проблемы маршрутизации может быть получено для каждого компонента с использованием методов из предыдущего раздела, что позволяет получить OE -маршрут для графа.

Оптимальное *NOE*-покрытие упорядоченной последовательностью *OE*-цепей

<i>OE</i> покрытие	Маршрут режущего инструмента
<i>chain</i> ₁ : v17 e14 v13 e9 v12 e8 v11 e11 v14 e10 v13 e27 v18 e15 v17 <i>chain</i> ₂ : v23 e21 v19 e16 v16 e30 v24 e33 v5 e12 v10 e35 v15 e13 v16 e18 v8 e17 v20 e32 v23 e20 v22 e19 v21 e26 v6 e25 v26 <i>chain</i> ₃ : v19 e22 v24 e23 v25 e24 v26 e28 v5 e4 v4 e34 v10 e7 v9 e29 v15 e31 v8 e6 v7 e5 v6 <i>chain</i> ₄ : v0 e0 v1 e1 v2 e2 v3 e3 v0	<pre>{ "partid": "chain_1", "paths": [[[150.0,100.0,-0.162278], [120.0,110.0,-0.0], [170.0,110.0,-0.0], [170.0,160.0,-0.0], [120.0,160.0,-0.0], [120.0,110.0,-0.720759], [150.0,200.0,-1.0], [150.0,100.0,0]]]], { "partid": "chain_2", "paths": [[[150.0,400.0,0.0], [150.0,265.0,-0.0], [150.0,250.0,0.0375], [135.0,250.0,0.374006], [50.0,150.0,0.4142], [150.0,50.0,0.4142], [250.0,150.0,0.4142], [150.0,250.0,-0.0], [250.0,300.0,-0.0], [150.0,475.0,-0.0], [150.0,400.0,-0.414213],[135.0,415.0,-0.0], [65.0,415.0,-0.414213], [50.0,400.0,-0.0], [50.0,265.0,0]]]], { "partid": "chain_3", "paths": [[[150.0,265.0,0.414213], [135.0,250.0,0.0], [65.0,250.0,0.414213], [50.0,265.0,-0.0], [50.0,150.0,-0.0], [50.0,50.0,-0.0], [150.0,50.0,-0.0], [250.0,50.0,-0.0], [250.0,150.0,-0.0], [250.0,300.0,-0.0], [250.0,400.0,1.0], [50.0,400.0,0]]]], { "partid": "chain_4", "paths": [[[300.0,0.0,0.0], [0.0,0.0,0.0], [0.0,400.0,-1.0], [300.0,400.0,0.0], [300.0,0.0,0.0]]]]]</pre>

Оптимальное *NOE*-покрытие упорядоченной последовательностью *OE*-цепей представлено в таблице 1. Первая цепь последовательно проходит квадрат и внутреннюю границу кольца. Вторая цепь последовательно проходит по правой стороне прямоугольника, внешней границе кольца, завершает вырезание треугольника и прямоугольника. Третья цепь завершает вырезание внутреннего контура окна. Четвертая цепь вырезает внешний контур окна.

Код траектории движения режущего инструмента в терминах JSON представлен во втором столбце таблицы 1. Каждой части траектории непрерывной резки соответствует отдельная цепь, которая представляет последовательность трехэлементных массивов, содержащих координаты начальной точки текущего примитива и значение $\tan(\varphi/4)$ для него. При осуществлении вырезания в соответствии с указанной последовательностью цепей будут выполнены все технологические требования.

Заключение

Технология, допускающая совмещение границ вырезаемых деталей – современная ресурсосберегающая технология резки. Существует несколько алгоритмов ее реализации. В настоящее время актуален вопрос эффективной программной реализации указанных алгоритмов. В докладе представлены результаты авторов, использованные при разработке функциональных элементов комплекса программ автоматизированной системы технологической подготовки процессов раскроя листового материала.

Литература

1. T.A. Makarovskikh, A.V. Panyukov, E.A. Savitsky, Mathematical Models and Routing Algorithms for CAD Technological Preparation of Cutting Processes, Automation and Remote Control, Vol. 78, No. 4. (2017) 868–882.
2. R. Dewil, P. Vansteenwegen, D. Cattrysse, A Review of Cutting Path Algorithms for Laser Cutters, International Journal Adv Manuf. Technol., 87 (2016) 1865–1884. DOI: 10.1007/s00170-016-8609-1
3. T. Makarovskikh, A. Panyukov, The Cutter Trajectory Avoiding Intersections of Cuts, IFAC-PapersOnLine, 50, Issue 1 (2017) 2284–2289.
4. X. Li, Zh. Liu, F. Wang, B. Yi, Y. Song, Combining physical shell mapping and reverse-compensation optimisation for spiral machining of free-form surfaces, International Journal of Production Research (2018) DOI: 10.1080/00207543.2018.1512763
5. T. Makarovskikh, A. Panyukov, Development of routing methods for cutting out details, CEUR Workshop Proceedings, 2098 (2018)249–263.
6. R. Dewil, P. Vansteenwegen, D. Cattrysse, M. Laguna, T. Vossen, An Improvement Heuristic Framework for the Laser Cutting Tool Path Problem, International Journal of Production Research, 53, Issue 6 (2015) 1761–1776. DOI: 10.1080/00207543.2014.959268.
7. A. Petunin, C. Stylios, Optimization Models of Tool Path Problem for CNC Sheet Metal Cutting Machines, IFAC-PapersOnLine, 49 (2016) 23–28.
8. A. Petunin, A.G. Chentsov, P.A. Chentsov, About routing in the sheet cutting, Bulletin of the South Ural State University, Series: Mathematical Modelling, Programming and Computer Software, 10(3) (2017) 25–39. DOI: 10.14529/mmp170303.
9. A.G. Chentsov, A.M. Grigoryev, A.A. Chentsov, Solving a Routing Problem with the Aid of an Independent Computations Scheme, Bulletin of the South Ural State University, Series: Mathematical Modelling, Programming and Computer Software, 11(1) (2018) 60–74.
10. M. Khachay, K. Neznakhina, Towards Tractability of the Euclidean Generalized Travelling Salesman Problem in Grid Clusters Defined by a Grid of Bounded Height, Communications in Computer and Information Science, 871 (2018) 68–77.
11. A. Chentsov, M. Khachay, D. Khachay, Linear Time Algorithm for Precedence Constrained Asymmetric Generalized Traveling Salesman Problem, IFAC-PapersOnLine, 49 (2016) 651–655.

12. J. Hoefl, U. Palekar, Heuristics for the Plate-cutting Travelling Salesman Problem, *IIE Transactions*, 29(9) (1997) 719–731.
13. R. Dewil, P. Vansteenwegen, D. Cattrysse, Construction heuristics for generating tool paths for laser cutters, *International Journal of Production Research*, 52(20) (2014) 5965–5984.
14. U. Manber, S. Israni, Pierce Point Minimization and Optimal Torch Path Determination in Flame Cutting, *J. Manuf. Syst.* 3(1) (1984) 81–89.
15. R.S. Garfinkel, I.R. Webb, On crossings, the Crossing Postman Problem, and the Rural Postman Problem, *Networks*, 34(3) (1999) 173–180.
16. Panyukova T.A. Constructing of OE-postman Path for a Planar Graph, *Bulletin of the South Ural State University, Series: Mathematical Modelling, Programming and Computer Software*, 7(4) (2014) 90–101. DOI: 10.14529/mmp140407.
17. T.A. Makarovskikh, A.V. Panyukov, E.A. Savitskiy, Mathematical models and routing algorithms for economical cutting tool paths, *International Journal of Production Research*, 56(3) (2018) 1171–1188. DOI: 10.1080/00207543.2017.1401746.
18. D. Crockford The application/json Media Type for JavaScript Object Notation (JSON), *Internet Engineering Task Force* (2006)
19. U. Manber, S.W. Bent, On Non-intersecting Eulerian Circuits, *Discrete Applied Mathematics*, 18 (1987) 87–94.
20. T.A. Makarovskikh, A.V. Panyukov, AOE-Trails Constructing for a Plane Connected 4-Regular Graph, *CEUR Workshop Proceedings*, 1623 (2017) 62–71. URL: <http://ceur-ws.org/Vol-1623>
21. M.R. Garey, D.S. Johnson, *Computer and intractability: a guide to the theory of NP-completeness* (1979).