

С.Е. Степанов, к.ф.-м.н.,
В.В. Калмыков, старший преподаватель,
К.В. Яшкин, студент
(Калужский филиал МГТУ имени Н.Э. Баумана, 248000, г. Калуга, ул. Баженова, д. 2)
E-mail: stepanov@bmstu-kaluga.ru

Алгоритмизация определения наиболее производительной последовательности обработки массива отверстий на станках с ЧПУ

Представлен оригинальный алгоритм поиска кратчайшего пути, применение которого способствует сокращению вспомогательного времени за счёт уменьшения холостого хода инструмента. Рассмотрены другие алгоритмы поиска кратчайшего пути и проведен их сравнительный анализ. Представлены объективные данные о преимуществах приведённого алгоритма в нахождении кратчайшего пути на плоскости.

Ключевые слова: оптимизация; алгоритмизация; задача коммивояжёра; жадный алгоритм; метод ветвей и границ; вычислительный эксперимент; гамильтонов путь.

S.E. Stepanov, Can. Sc. Phys-Math.,
V.V. Kalmykov, Senior lecturer,
K.V. Yashkin, Student
(Kaluga Branch of Bauman STU of Moscow, 2, Bazhenov Str., Kaluga, 248000)

Algorithmization for definition of most efficient sequence of hole block processing on NC machines

The original algorithm for the search of the shortest way the application of which contributes to the reduction of supplementary time at the expense of the decrease of tool idling is presented. There are considered other algorithms for the search of the shortest way and their comparative analysis is carried out. The objective data on advantages of the algorithm presented in finding the shortest way on a plane are presented.

Keywords: optimization; algorithmization; travelling agent's problem; eager algorithm; branch and border method; computer experiment; Hamilton's way.

Введение

В машиностроении при разработке технологических процессов для повышения производительности необходимо выбрать минимальный маршрут холостых перемещений рабочих органов современного автоматизированного оборудования [1, 2]. История решения подобных задач, в частности, задачи коммивояжёра, насчитывает без малого двести лет. Математическая формулировка данной задачи

как задачи оптимизации была дана Карлом Менгером в 1930 г.

Сегодня существует огромное разнообразие методов решения задачи коммивояжёра, поэтому зачастую трудно определить, какому методу стоит отдать предпочтение и какой из них будет наиболее качественно и быстро решать задачу оптимизации пути [3]. Самым точным методом решения задачи коммивояжёра является полный перебор всех вершин, в этом случае количество маршрутов составит значение, равное $n!$, но, как известно, эта

функция является быстро растущей и для количества вершин больше 30 является трудно осуществимой так как $(30-1)!$ будет составлять величину равную $8,84 \cdot 10^{30}$. Кроме того, задача коммивояжера относится к классу NP – полных задач, что было доказано Ричардом Карпом в 1972 г. Подробное доказательство описано в работе [4].

Математическая постановка задачи

Имеется полный взвешенный неориентированный граф $G = (X, E)$, где X – множество всех вершин, а E – множество всех рёбер. Необходимо пройти все вершины (построить гамильтонов путь) так, чтобы суммарный вес рёбер был минимальным.

Во многих производственных задачах, главной целью ставится нахождение гамильтонова пути, что снимает необходимость возвращения в исходную вершину. В качестве примера можно привести нахождение кратчайшего пути инструмента при изготовлении детали на станках с ЧПУ.

Задачи такого типа решаются методами, которые условно можно разделить на два класса – приближённые и точные. В данной работе предлагается оригинальный приближённый алгоритм реализации решения задачи коммивояжера.

Описание существующих алгоритмов

Алгоритм Литтла (Алгоритм 1) – частный случай метода ветвей и границ. Он является одним из лучших по оптимальности найденного пути, но в то же время из-за сложности его реализации производить точные расчёты на большом количестве не представляется возможным. В худшем случае алгоритм приводит к полному перебору вариантов.

Временная сложность этого алгоритма составляет $O(n^2 \cdot c^n)$, где константа c оценена в ходе вычислительного эксперимента, приведённого в работе [5].

Стоит отметить, что хоть алгоритм и является точным, но в ходе эксперимента на корпусной детали (рис.1) этим алгоритмом был найден путь, вес которого был больше, чем вес путей, найденных другими рассматриваемыми алгоритмами.

Если несколько видоизменить постановку задачи о нахождении гамильтонова контура, решаемой методом ветвей и границ, и не требовать возвращения к исходной точке, то не составит труда определить искомый путь [6].

Shortest Hamiltonian Path and Circuit (SHPAC) является точным алгоритмом, используемым для отыскания гамильтонова пути или контура минимального веса. Этот алгоритм основан на использовании упорядоченного взвешенного списка смежных вершин графа и простых свойств, которым должен удовлетворять любой путь или цикл. Временная сложность этого алгоритма $O(n^2 \cdot 2^n)$ [7].

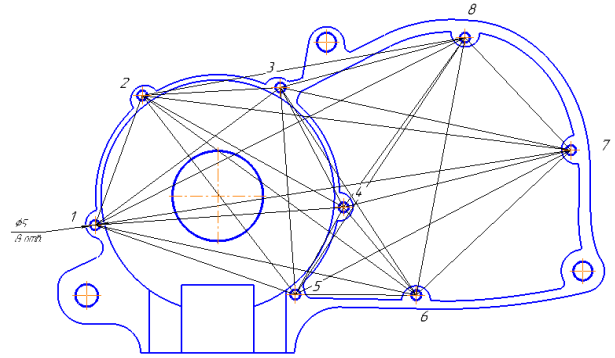


Рис.1. Главный вид корпусной детали 1

Алгоритм Донецкова А.М. (Алгоритм 2), разработанный в МГТУ им.Н.Э. Баумана [8 – 9], является приближенным и решает задачу нахождения кратчайшего гамильтонова контура с точностью большей или равной $3/4$ от оптимального, при условии симметричности графа и $2/3$ от оптимального в обратном случае. Временная сложность предложенного алгоритма равна $O(n^4)$.

Жадный алгоритм (ЖА) представляет собой приближённый класс алгоритмов. Он не всегда приводит к оптимальному решению, но во многих задачах даёт приемлемый, а в некоторых случаях и лучший результат. Жадный алгоритм подходит для довольно широкого класса задач, за счёт достаточной мощности исполнения. Если решать задачу поиска гамильтонова пути, применяя жадный алгоритм, то на каждом из этапов поиска делается выбор, который кажется лучшим, не оценивая граф целиком и, как следствие, не застрахован от ошибок. Временная сложность простейшего алгоритма, основанного на жадном, составляет $O(n^2)$.

Лучшая реализация алгоритма осуществляется на т.н. матроидах, для которых жадный метод всегда находит кратчайший путь при условии их взвешенности. Эта теория представляет практический интерес, быстро развивается и находит всё больше сфер приложения [10]. В станках с ЧПУ используется

именно этот алгоритм.

Предлагаемый алгоритм

Алгоритм 3. Так как целью разработки алгоритма является его оптимизация не только по времени, но и по качеству решения, предлагается алгоритм, который, исходя из анализа выше рассмотренных алгоритмов, основан на жадном алгоритме, как на самом оптимальном по времени, исключает возможные ошибки жадного метода, делая проверку «лучшего выбора» на каждом шаге.

Алгоритмическая сложность данного алгоритма – $O(n^3)$, что даёт преимущество над другими алгоритмами в скорости поиска кратчайшего пути за исключением жадного, временная сложность которого $O(n^2)$ и не уступающий в точности другим более сложным алгоритмам.

Данный алгоритм является наиболее оптимальным и представляет практическую ценность для любых видов задач, сводимых к взвешенному, сильно связанному графу, для которого, в свою очередь, выполняется неравенство треугольника, определяющее решение задачи коммивояжера как гамильтонова пути [11].

Теорема: если для каждой пары вершин (x, y) графа G выполняется условие:

$a(x,y) \leq a(x,z) + a(z,y)$ для всех $z \neq x, z \neq y, (1)$ то гамильтонов контур является решением общей задачи коммивояжера на графе G . Доказательство этой теоремы приводится в работе [4]

Алгоритм:

1. Нумеруются вершины графа $(1, 2, 3, \dots, n)$.
2. Выбирается точка l и находится ребро с наименьшим весом, выходящее из выбранной точки.
3. Исключаются те рёбра выбор которых приведёт к не самому благоприятному результату в имеющимся графе следующим образом:

$$(l, k) + (k, p) \leq (l, p) + (p, k), \quad (2)$$

где k – ближайшая вершина к вершине l , и p – ближайшая вершина к вершине $k, k \neq 1$, которые находятся аналогично второму шагу.

Исходя из неравенства (2) исключаются из перебора все рёбра $(l, n-1)$, веса которых заведомо больше веса (l, p) , кроме рёбер $(p, k), (l, k), (k, p)$; если их веса меньше.

1. Проверяется, не снимет ли выбранное

жадным методом, ребро (l, k) , весь эффект экономии времени при выборе следующего ребра (k, p) :

$$\text{если } (k, p) + (p, s) < (p, k) + (k, s), \quad (3)$$

где s – ближайшая вершина к вершине $p, s \neq k$, определяется аналогично второму шагу; то включается ребро (l, k) в маршрут и, считая точку k за начальную, повторяется алгоритм, начиная со второго шага.

$$\text{Если } (k, p) + (p, s) > (p, k) + (k, s), \quad (4)$$

то включается ребро (l, p) в маршрут, тем самым исключая ошибку жадного алгоритма при выборе кратчайшего гамильтонова пути. Повторяется алгоритм, начиная со второго шага считая точку p за начальную.

2. Повторяется алгоритм, начиная со второго шага, меняя только начальную точку, до тех пор, пока все вершины не будут перебраны.

Псевдокод предлагаемого алгоритма выглядит следующим образом:

Алгоритм (G, l, n)

Вход: граф $G(X, E)$ (не ориентированный) с не отрицательными длинами рёбер $\{l_e : e \in E\}$; вершина $n \in X$.

Выход: для всех вершин u , достижимых из $n, dist[u]$ будет равно расстоянию от n до u .

```

procedure Algorithm  $(G, l, n)$ 
while  $H \neq \emptyset$  do
 $u \leftarrow \text{DELETE} - \text{MIN}(H)$ 
for вершины  $j \in X$  do
 $\text{find } k \triangleright$  где  $k$  ближайшая вершина к вершине  $l$ 
 $dist[j] \leftarrow nil$ 
end for
for вершины  $k \in X$  do
 $\text{find } p, p \neq k \triangleright$  где  $p$  ближайшая вершина к вершине  $k$ 
 $dist[k] \leftarrow nil$ 
end for
for вершины  $p \in X$  do
 $\text{find } s, s \neq p \triangleright$  где  $s$  ближайшая вершина к вершине  $p$ 
 $dist[p] \leftarrow nil$ 
end for
for  $l_1 \leftarrow dist[j,k], l_2 \leftarrow dist[k,p], l_3 \leftarrow dist[p,s], l_4 \leftarrow dist[k,s]$  где  $l_1, l_2, l_3, l_4 \in l_e$  do
if  $l_2 + l_3 < l_2 + l_4$ , then
 $dist[u] \leftarrow dist[j,k]$ 
return к шагу 1, принимается  $k$  начальной вершиной
else  $dist[u] \leftarrow dist[j,p]$ 
return к шагу 1, принимается  $p$  начальной вершиной
    
```

```

end if
end for
end while
write dist [u]
end
    
```

Сравнение алгоритмов

В рамках исследования эксперимент проводился на двух типовых корпусных деталях, представленных на рис. 1 и 2, и соответствующая рис. 1 матрица смежности в табл. 1.

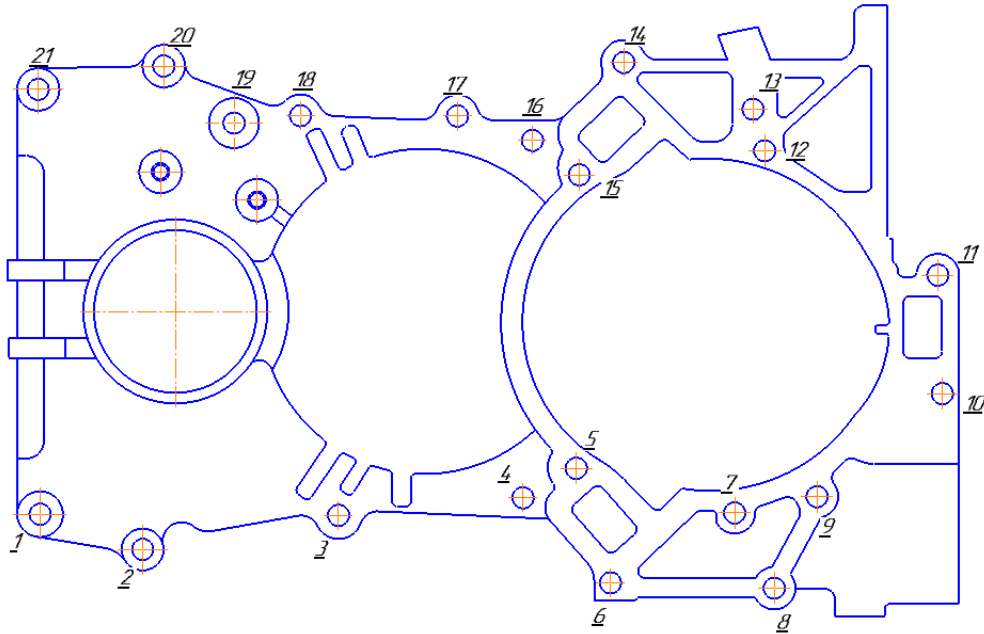


Рис. 2. Главный вид корпусной детали 2

1. Матрица смежности массива отверстий корпусной детали 1

	1	2	3	4	5	6	7	8
1	∞	77,22*	140,45	128,50	77,17	182,73	240,83	189,29
2	77,22	∞	118,16**	138,82	128,92	231,32	268,45	183,22
3	140,45	118,16	∞	55,81*	116,38	172,25	173,59	67,38
4	128,50	138,82	55,81	∞	75,82*	116,69	130,67	63,74**
5	77,17*	128,92	116,38	75,82	∞	106,52	165,61	138,82
6	182,73	231,32	172,25	116,69	106,52**	∞	86,39*	146,84
7	240,83	268,45	173,59	130,67	165,61	86,39**	∞	118,45*
8	189,29	183,22	67,38*	63,74	138,82	146,84	118,45	∞

Примечание. * Длины рёбер, которые обозначают маршрут, выбранный алгоритмом 1; ** – другими алгоритмами.

2. Длины хода инструмента на корпусной детали 1

Методы	Алгоритм 1	SHRAC, Алгоритм 2, ЖА, Алгоритм 3
L_{xx}	585,01	558,24
Примечание. L_{xx} – длина маршрута, мм		

На рис. 1 представлен эскиз корпусной детали, в котором необходимо обработать 8 отверстий диаметром 5 мм. Из условия, что дан граф $G = (X, E)$, где отверстия представляют

вершины графа X , а рёбра E представлены в виде отрезков соединяющих каждое отверстие с каждым. Необходимо задать такой путь обхода инструментом детали, чтобы длина холостого хода была наименьшей, или же найти кратчайший гамильтонов путь, при условии, что обработка ведётся на станке с ЧПУ и возвращение в исходную точку может производиться одновременно со сменой детали.

$$L_{xx} = \sum_{i,j \rightarrow \min} L_{i,j} \quad (5)$$

где L_{xx} – это минимальная сумма всех перемещений $L_{i,j}$.

Решая эту задачу, были получены следующие результаты, приведённые в табл. 2.

При условии того, что алгоритм 1 представляет собой вариацию полного перебора, оптимально справится с поставленной задачей алгоритм не смог и определил путь, который на 4,8 % длиннее путей, найденных другими алгоритмами. Что касается ЖА, на котором так же основана функция «point to point» ряда инженерных САПР (систем автоматизированного проектирования) подготовки управляющих программ для станков с ЧПУ, то основной проблемой является проблема выбора точки начала отсчёта, от выбора которой и зависит определение оптимального пути (табл. 3).

3. Длины хода инструмента на корпусной детали 2

Методы	ЖА	Алгоритм 1, SHPAC, Алгоритм 2, Алгоритм 3
L_{xx}	1429,28	1344,37
<i>Примечание.</i> L_{xx} – длина маршрута, мм		

Для второго примера детали, имеющей

двадцать одно отверстие диаметром 8 мм, были поставлены идентичные условия, представляющие его в виде графа. Чтобы не загромождать чертёж, рёбра графа были скрыты (см. рис. 2).

Исходя из полученных результатов, жадный алгоритм определил путь, который на 6 % оказался длиннее, чем путь, найденный другими рассмотренными алгоритмами.

Наилучшие значения одновременно и по времени поиска и по оптимальной длине найденного пути показал предложенный в данной статье алгоритм 3, имеющий временную сложность $O(n^3)$.

Стоит отметить, что алгоритм, предложенный Донецковым А.М., не уступал по оптимальности найденного пути, но однократно уступал по времени, которое он затрачивал. В табл. 4, приведены значения времени реализации алгоритмов, полученные на деталях, приведённых на рис. 1 и 2. Все вычисления были произведены теоретическим путём на основании известных временных сложностей всех алгоритмов. На процессоре Intel Core i3-5010U с тактовой частотой 2,1 ГГц.

4. Время реализации алгоритмов

Методы Кол-во точек	SHPAC	Алгоритм 1	Алгоритм 2	Алгоритм 3	ЖА
8	0,00008	0,0027	0,000002	0	0
21	0,44040	0,0174	0,000093	0,000004	0
60	$6,2672 \cdot 10^4$ лет	8,4560	0,006171	0,000103	0,000002
100	-	185,762	0,0476	0,000476	0,000005
200	-	919 лет	0,761905	0,003809	0,000019
300	-	-	3,857	0,012857	0,000043
1000	-	-	476,1905	0,476190	0,000476
<i>Примечания:</i> 1. Численные значения без размерности представлены в секундах. 2. Прочерки означают, что время, затрачиваемое на поиск оптимального решения велико.					

Заключение

По итогам экспериментального исследования, предлагаемый алгоритм во всех случаях выдавал аналогичный оптимальный результат, получаемый другими более сложными алгоритмами, однако, за существенно меньшее время. Жадный алгоритм превосходит по скорости предложенный, но в большинстве случаев он не позволяет определять оптимальные результаты длин маршрутов.

Таким образом, применение данного алгоритма позволит эффективно повысить производительность решения задачи коммивояжёра. Он может быть использован в современных САПР подготовки управляющих программ для станков с ЧПУ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Пронин, А.И., Жеребцов, Д.В. Оптимизация холостых перемещений инструмента при обработке деталей на

многоцелевых станках // Результаты современных научных исследований и разработок. – 2019. №1. – С. 63-66.

2. Калмыков, В.В., Барина, Д.А. Оптимизация времени смены инструментов при изготовлении деталей на фрезерных станках с ЧПУ // Электронный журнал: наука, техника и образование. – 2018. – № СВ2 (20). – С. 24-28.

3. Кравченко, И.И., Бухаров, С.В. Анализ средств автоматизации программирования оборудования, оптимизация последовательности обработки поверхностей сложных корпусных деталей // Машиностроение и компьютерные технологии. – 2018. – №. 7. – С. 31-47

4. Кормен, Томас Х. и др. Алгоритмы: построение и анализ, 3-е изд.: пер. с англ. – М.: ООО «И. Д. Вильямс», 2013. – 1328 с.: ил.

5. Костюк, Ю.Л. Эффективная реализация алгоритма решения задачи коммивояжера методом ветвей и границ: Электронный ресурс: <http://www.inf.tsu.ru/Decanat/Staff.nsf/people/KostjukJuL>

6. Майника, Э. Алгоритмы оптимизации на сетях и графах: пер. с англ. – М.: Мир, 1981. – 321с.

7. Dhananjay Mehendale Polynomial Algorithms for Shortest Hamiltonian Path and Circuit [сайт]. URL: <https://www.researchgate.net/publication/267377360>.ru (дата обращения: 14.05.2019).

8. Донецков, А.М. Подход к решению задачи коммивояжера. // Электромагнитные волны и электронные системы. – 2015. – № 7. – Т. 20. – С. 57-60.

9. Донецков, А.М. Программа обеспечение решения задачи коммивояжера. – Электромагнитные волны и электронные системы, 2018, № 3, т. 23, с. 26-30

10. Савами, М., Тхуласираман, К. Графы, сети и алгоритмы: Пер. с англ. – М.: Мир, 1984. – 455с.

11. Яшкин, К.В. Алгоритмизация гамильтонова цикла // В книге «Научно-технические технологии в приборостроении и развитие инновационной деятельности в ВУЗЕ»; Матер. регион. науч.-техн. конф. – 2019. – С. 4 - 6.

REFERENCES

1. Pronin, A.I., Zhrebtsov, D.V. Optimization of tool idle movement at parts machining on multipurpose machines //

Results of Modern Scientific Researches and Developments. – 2019. No.1. – pp. 63-66.

2. Kalmykov, V.V., Barinova, D.A. Optimization of tool change time at parts machining on NC milling machines // *Electronic Journal: Science, Engineering and Education.* – 2018. – No.SV2 (20). – pp. 24-28.

3. Kravchenko, I.I., Bukharov, S.V. Analysis of means for programming automation of equipment, optimization of sequence in processing surfaces of complex body parts // *Mechanical Engineering and Computer Technologies.* – 2018. – No.7. – pp. 31-47.

4. Kormen, Thomas H et al. Algorithms: Formation and Analysis, 3-d Edition transl. from Engl. – М.: PC “I.D. Williams”, 2013. – pp. 1328.: ill.

5. Kostyuk, Yu.L. *Efficient Realization of Algorithm for Travelling Agent's Problem Solution through Method of Branches and Borders:* electronic resource: <http://www.inf.tsu.ru/Decanat/Staff.nsf/people/KostjukJuL>

6. Mainika, E. Algorithms for Optimization of Networks and Graphs: transl. from Engl. – М.: World, 1981. – pp. 321.

7. Dhananjay Mehendale Polynomial Algorithms for Shortest Hamiltonian Path and Circuit [site]. URL: <https://www.researchgate.net/publication/267377360>.ru (address date): 14.05.2019.

8. Donetskoy, A.M. Approach to travelling agent's problem solution. // *Electromagnetic Waves and Electronic Systems.* – 2015. – No.7. – Vol.20. – pp. 57-60.

9. Donetskoy, A.M. Program-support for solution of travelling agent's problem. – *Electromagnetic Waves and Electronic Systems*, 2018, No.3. Vol. 23, pp. 26-30.

10. SavamyM., Thulasiraman, K. *Graphs, Networks and Algorithms:* transl. from Engl. – М.: World, 1984. – pp. 455.

11. Yashkin, K.V. Algorithmization of Hamilton's cycle // In book “*Science Intensive Technologies in Instrument Making, Mechanical Engineering and Innovation Activity Development in College*”; Mater. Region Scientif.-Tech. Conf. – 2019. – pp. 4-6.

Рецензент д.т.н. Н.В. Носов

