

Э. ПРУУДЕН, Ю. ПРУУДЕН, Б. ТАММ

ОБ ОДНОМ ПРИБЛИЖЕННОМ МЕТОДЕ ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ЗАДАННОЙ СОВОКУПНОСТИ РАБОЧИХ ПОЗИЦИЙ

Рассматривается задача нахождения оптимального пути рабочего органа координатно-позиционного станка с программным управлением при обработке стохастически заданного множества рабочих позиций. Предлагается алгоритм для инженерного решения задачи на средних ЭВМ.

Вопрос о целесообразном прохождении некоторого заданного множества геометрических точек возникает при весьма различных ситуациях. Значительное количество задач относится к группе, получившей название «задача коммивояжера», не имеющей приемлемого универсального решения. Характерной чертой каждой входящей в эту группу задачи является наложение жестких специфических ограничений для ее решения.

Задача такого типа возникает при автоматизации подготовки программы для управления рабочим инструментом координатно-позиционного станка с программным управлением. В первую очередь здесь необходимо решить два вопроса: во-первых, как найти оптимальную последовательность перехода инструмента из одной рабочей позиции в другую и, во-вторых, как автоматизировать на ЭЦВМ расчет пути перестановки инструмента с целью получения из нее готовой управляющей программы для станка.

Наилучшее инженерное решение поставленной задачи состоит в составлении таких алгоритмов, при помощи которых при сравнительно небольшом объеме вычислительных работ получается результат, который существенно лучше результата, вероятно получаемого опытным технологом интуитивным путем.

Проведенные нами опыты показывают, что уже при наличии 15—20 стохастически расположенных точек даже опытный технолог отнюдь не всегда намечает наилучшую последовательность их обработки. На практике эта проблема еще сложнее, ибо число таких точек достигает нескольких десятков, сотен и даже тысяч и в таких случаях нельзя говорить о каких-либо разумных интуитивно найденных путях перестановки инструмента.

Оптимальным следует считать путь перестановки, минимизирующий время обработки, поэтому проблема сводится к нахождению некоторого пути, соединяющего все точки. Отметим, что специфика управления движением рабочего инструмента станка (автономные приводы для продольных и поперечных передвижений) требует истолковывать расстояние между двумя точками X_i и X_j с координатами (x_i, y_i) и (x_j, y_j) как

$$Q_{ij} = \max [|x_i - x_j|, |y_i - y_j|], \quad (1)$$

если рабочий инструмент движется одновременно в направлении обеих координатных осей, и

$$Q_{ij} = |x_i - x_j| + |y_i - y_j|, \quad (2)$$

если движения по направлениям координатных осей осуществляются одно за другим. Расстояния, выраженные формулами (1) и (2), определяют время движения инструмента из точки X_i в точку X_j и являются, таким образом, решающими при нахождении оптимального пути перестановки инструмента.

Необходимо отметить, что скорости движения рабочего инструмента в направлениях каждой из координат в общем случае не равны. Поэтому, чтобы истолковать расстояния между двумя точками в смысле, определенном выражением (1) или (2), масштаб одной из координат изменяется таким образом, чтобы скорости (в новых единицах длины на единицу времени) стали равными в обеих координатах.

Задача нахождения оптимального пути, соединяющего между собой заданную совокупность точек, рассматривалась в работах ряда авторов, где для ее решения приведены алгоритмы, основанные на применении динамического программирования или на применении теории графов (см. [1-3] и др.). Однако все эти методы требуют применения ЭВМ с очень большим объемом оперативной памяти, а также значительного количества машинного времени и поэтому неприемлемы для практического решения поставленной задачи.

Предлагаемый метод решения основан на применении теории графов. Представим рабочие позиции инструмента как точки или вершины X_i ($i = 1, \dots, n$) некоторого графа (X, U) , дугами u_i которого являются некоторые (удовлетворяющие условиям, заданным в алгоритме) возможные пути перестановки инструмента от одной рабочей позиции к следующей. К каждой дуге $u_j = (X_i, X_j)$ графа (X, U) отнесем число $q(X_i, X_j)$, которое назовем «расстоянием» дуги.

В таком представлении задача нахождения кратчайшего пути перестановки инструмента при обработке детали сводится к нахождению в графе (X, U) гамильтонова пути μ , сумма расстояний которого

$$\sum_{[X_i, X_j] \in \mu} q(X_i, X_j)$$

является наименьшей.

Из множества задач, требующих нахождения кратчайшего, охватывающего все заданные точки пути перестановки, следует выделить группу; где точки задаются в узлах некоторой определенной ортогональной сетки в декартовых или полярных координатах. Графы, соответствующие этим случаям, будем называть равномерными графами. Они встречаются в практике очень часто, решаются при помощи простых алгоритмов и поэтому подробно здесь не рассматриваются.

Значительно больший интерес представляют алгоритмы для нахождения гамильтоновых путей с наименьшей суммой расстояний в графах со стохастически расположенными вершинами.

Нижеприведенный алгоритм дает приближенные решения и является одним из возможных алгоритмов для инженерного решения поставленной задачи.

Алгоритм можно разделить на три части:

- I часть — определение графа (X, U) ;
- II часть — нахождение в графе (X, U) элементарного пути с наибольшей длиной и наименьшей суммой расстояний;
- III часть — коррекция найденного пути, если при помощи II части алгоритма не удалось определить элементарный путь длиной $n - 1$, где n — число точек в графе.

Переход к следующей части происходит только после завершения предыдущей.

I часть алгоритма. Вычисляем расстояние между каждой парой точек X_i, X_j ($i, j = 1, \dots, n$; $i \neq j$) по формуле (1) или (2) и обозначаем эти расстояния

$$q_{ij} = q(X_i, X_j).$$

Определяем множества

$$\mathfrak{M}_k^{(i)} = \{m_1^{(i)}, m_2^{(i)}, \dots, m_k^{(i)}\} \quad (k = 1, \dots, N; n-1 \geq N = \text{const}),$$

где N — ранее выбранное число, а

$$m_1^{(i)} = \min_j \{q_{ij}\}$$

$$m_2^{(i)} = \min_j \{ \{q_{ij}\} \setminus \mathfrak{M}_1^{(i)} \}$$

$$\dots \dots \dots$$

$$m_k^{(i)} = \min_j \{ \{q_{ij}\} \setminus \mathfrak{M}_{k-1}^{(i)} \}.$$

Определяем граф (X, U) , где X — множество заданных точек X_i
 $X = \{X_i\}$

и $u_{ij} = (X_i, X_j) \in U$ тогда и только тогда, если вычисленное для этой пары точек (X_i, X_j) расстояние $q_{ij} = q(X_i, X_j)$ удовлетворяет одному из условий

$$1) q_{ij} \leq m_N^{(j)} \quad \text{или} \quad 2) q_{ij} \leq m_N^{(i)}.$$

К каждой дуге u_{ij} графа (X, U) отнесем число $q_{ij} = q(X_i, X_j)$, т. е. расстояние между граничными точками X_i, X_j дуги u_{ij} . Назовем это число расстоянием дуги u_{ij} .

Определением графа (X, U) и множества $\{q_{ij}\}$ окончена I часть алгоритма.

II часть алгоритма. Работа второй части алгоритма циклическая. В каждом цикле находят в графе (X, U) некоторое количество элементарных путей μ с заданной начальной точкой X_0 (одинаковой для всех путей) и с длиной

$$l(\mu) = k,$$

где k — порядковый номер цикла.

Рассматриваем работу одного из циклов. Предполагаем, что выполнено $k-1$ циклов. В последнем цикле получено L_{k-1} элементарных путей, образующих множество

$$\{\mu_t^{(k-1)}\} = \{\mu_1^{(k-1)}, \dots, \mu_{L_{k-1}}^{(k-1)}\},$$

где
$$\mu_t^{(k-1)} = [X_0, X_{t_1}, \dots, X_{t_{k-1}}],$$

в котором $X_{t_1}, \dots, X_{t_{k-1}} \in X$.

Длина этих путей

$$l(\mu_t^{(k-1)}) = k-1.$$

Для каждого пути $\mu_t^{(k-1)}$ ($t = 1, \dots, L_{k-1}$) вычислена и сумма расстояний

$$S_t^{(k-1)} = \sum_{q=1}^{k-1} q(X_{t_{q-1}}, X_{t_q}), \quad \text{где } X_{t_0} = X_0,$$

тем самым получено множество $\{S_t^{(k-1)}\}$.

Работа алгоритма в цикле k сводится к следующему. При каждом пути $\mu_t^{(k-1)}$ берем конечную его точку $X_{t_{k-1}}$ и выделяем все дуги

$$u_{t_{k-1}, j} = (X_{t_{k-1}}, X_j) \in U, \text{ где } X_j \notin \mu_t^{(k-1)}.$$

Обозначим соответствующие дугам $u_{t_{k-1}, j}$ расстояния

$$q_{t_{k-1}, j} = q(X_{t_{k-1}}, X_j).$$

Теперь образуем все возможные элементарные пути

$$\mu_{tj}^{(k)} = \mu_t^{(k-1)} + u_{t_{k-1}, j} = [X_0, X_{t_1}, \dots, X_{t_{k-1}}, X_{t_k}] \quad (\text{здесь } X_{t_k} = X_j),$$

имеющие длину $l(\mu_{tj}^{(k)}) = k$, и вычислим все соответствующие путям $\mu_{tj}^{(k)}$ суммы расстояний

$$S_{tj}^{(k)} = S_t^{(k-1)} + q_{t_{k-1}, j}.$$

Эти суммы расстояний образуют множество $\{S_p^{(k)}\}$.

Образуем множества

$$\mathfrak{M}_s^{(k)} = \{m_1^{(k)}, m_2^{(k)}, \dots, m_s^{(k)}\} \quad (s = 1, \dots, L; L - \text{const}),$$

где L — ранее выбранное число, а

$$m_1^{(k)} = \min_p \{S_p^{(k)}\}$$

$$m_2^{(k)} = \min_p \{\{S_p^{(k)}\} \setminus \mathfrak{M}_1^{(k)}\}$$

$$\dots$$

$$m_s^{(k)} = \min_p \{\{S_p^{(k)}\} \setminus \mathfrak{M}_{s-1}^{(k)}\}.$$

Теперь образуем множество сумм расстояний $\{S_t^{(k)}\}$ ($t = 1, \dots, L_k$) так, чтобы его элементы удовлетворяли условию $S_t^{(k)} \leq m_L^{(k)}$. Выбираем из всех элементов множества $\{\mu_{tj}^{(k)}\}$ те и только те элементы, которым соответствуют элементы в множестве $\{S_t^{(k)}\}$.

Получаем множество $\{\mu_t^{(k)}\}$ ($t = 1, \dots, L_k$).

Тем самым цикл k окончен. В результате найдены L_k элементарных путей длиной k . Продолжаем такие циклы до тех пор, пока в очередном цикле $m+1$ окажется невозможным найти хотя бы один элементарный путь $\mu_t^{(m+1)}$, т. е. $\{\mu_t^{(m+1)}\} = \emptyset$.

В таком случае из путей, полученных в цикле m , выбираем путь,

$$\mu_{t_1}^m \in \{\mu_t^{(m)}\} \quad (t = 1, \dots, L_m),$$

которому соответствует минимальная сумма расстояний

$$S_{t_1}^{(m)} = \min_t \{S_t^{(m)}\} \quad (t = 1, \dots, L_m).$$

Теперь могут иметь место два случая:

1) если $m = n - 1$, то найденный элементарный путь $\mu_{i_1}^{(m)}$ имеет длину n , являясь тем самым искомым гамильтоновым путем μ_n ;

2) если $m < n - 1$, то при помощи II части алгоритма невозможно найти гамильтонового пути. В таком случае для его нахождения требуется применить третью часть алгоритма.

III часть алгоритма. Во второй части алгоритма был найден элементарный путь $\mu_{i_1}^{(m)}$ с длиной $l(\mu_{i_1}^{(m)}) = m < n - 1$ и суммой расстояний $S_{i_1}^{(m)}$.

Обозначим последовательные точки этого пути $X'_0, X'_1, X'_2, \dots, X'_m$, т. е. $\mu_{i_1}^{(m)} = [X'_0, X'_1, \dots, X'_m]$, где $X'_0 = X_0$.

Эти точки образуют множество

$$\mathfrak{X}' = \{X'_i\}, \quad \text{где } i = 0, 1, \dots, m.$$

Выделяем множество точек X^* так, чтобы

$$X^* = X \setminus \mathfrak{X}'.$$

Образуем теперь подграф (X^*, U) графа (X, U) . В подграфе (X^*, U) образуем все такие элементарные пути, начальная точка которых X'_{i_0} и точка $X'_i \in \mathfrak{X}'$, а также конечная точка X'_{i_p} и точка $X'_{i+1} \in \mathfrak{X}'$ являются попарно смежными в графе (X, U) , причем X'_i и X'_{i+1} являются граничными точками какой-то дуги пути $\mu_{i_1}^{(m)}$. Так получим в подграфе (X^*, U) множество элементарных путей, где

$$\mu_i = [\mu_{i_1}^{(m)}],$$

если $[X'_i, X'_{i_0}] \in U$, $[X'_{i_p}, X'_{i+1}] \in U$, а $[X'_i, X'_{i+1}] \in \mu_{i_1}^{(m)}$.

Теперь образуем в графе (X, U) все элементарные пути

$$\mu_{ii} = [X'_i, X'_{i_0}] + \mu_i + [X'_{i_p}, X'_{i+1}],$$

кроме того образуем в графе (X, U) все возможные пути

$$\mu_{si} = [X'_i, X_s, X'_{i+1}], \quad \text{где } X_s \in X^*, \text{ и } [X'_i, X'_{i+1}] \in \mu_{i_1}^{(m)}.$$

В итоге получим множество путей

$$\{\mu\} = \{\mu_{ii}\} \cup \{\mu_{si}\}.$$

Выбираем из множества $\{\mu\}$ некоторое число ω различных подмножеств $Q_i \subset \{\mu\}$ ($i = 1, \dots, m$), где $Q_i = \{\mu_{i_k}\}$, таким образом, чтобы:

1) ни в одном из подмножеств Q_i не было бы:

а) двух путей $\mu_{i_A}, \mu_{i_B} \in Q_i$ с одинаковыми начальными или конечными точками,

б) двух путей $\mu_{i_C}, \mu_{i_D} \in Q_i$, в последовательностях точек которых встретилась бы одна и та же точка $X_i \in X^*$;

2) суммарное количество точек $X_i \in X^*$, содержащихся в путях μ_{i_k} одного подмножества Q_i , было бы максимально возможным и равным для всех подмножеств Q_i (в пределе это количество совпадает с количеством вершин в X^*). Обозначим это количество символом m' ,

$$m' \leq n - m - 1.$$

При помощи каждого подмножества Q_i и пути $\mu_{i_1}^{(m)}$ образуем новые элементарные пути нижеследующим образом.

Разделим путь $\mu_{i_1}^{(m)}$ на $q + 1$ частичных путей, удаляя из $\mu_{i_1}^{(m)}$ q дуг так, чтобы в результате:

1) обе граничные точки любого частичного пути (кроме точек, совпадающих с граничными точками пути $\mu_{i_1}^{(m)}$) являлись граничными точками каких-то двух путей из одного подмножества $\{\mu_{R_k}\} = Q_R \subset \{Q_i\}$;

2) среди внутренних точек любого частичного пути не было ни одной граничной вершины любого пути из того же подмножества Q_R .

Обозначим образованные частичные пути

$$\mu'_1[X'_0, X'_{i_1}], \mu'_2[X'_{i_1+1}, X'_{i_2}], \dots, \mu'_k[X'_{i_{k-1}+1}, X'_{i_k}], \mu'_{k+1}[X'_{i_k+1}, X'_{i_{k+1}}], \dots, \mu'_{q+1}[X'_{i_q+1}, X'_m] \subset \mu_{i_1}^{(m)},$$

где

$$X'_{i_1}, X'_{i_1+1}, X'_{i_2}, \dots, X'_{i_{k-1}+1}, X'_{i_k}, X'_{i_k+1}, X'_{i_{k+1}}, \dots, X'_{i_q+1} \in X'$$

и

$$[X'_{i_k}, X'_{i_{k+1}}] \in \mu_{R_k} \in Q_R.$$

Теперь образуем элементарный путь μ_{Q_R} при помощи всех частичных путей μ'_k и всех путей μ_{R_k} из подмножества Q_R

$$\mu_{Q_R} = \mu'_1[X'_0, X'_{i_1}] + \mu_{R_1}[X'_{i_1}, X'_{i_1+1}] + \mu'_2[X'_{i_1+1}, X'_{i_2}] + \dots + \mu_{R_q}[X'_{i_q}, X'_{i_q+1}] + \mu'_{q+1}[X'_{i_q+1}, X'_m],$$

где

$$\mu'_1, \mu'_2, \dots, \mu'_{q+1} \subset \mu_{i_1}^{(m)} \text{ и } \mu_{R_1}, \dots, \mu_{R_q} \in Q_R, \mu_{R_1} \cup \dots \cup \mu_{R_q} = Q_R.$$

Так образуем все пути μ_{Q_i} ($i = 1, \dots, \omega$), имеющие одинаковую длину

$$l(\mu_{Q_i}) = m + m'.$$

Вычислим соответствующие этим элементарным путям суммы расстояний S_{Q_i} .

Выбираем из путей μ_{Q_i} путь μ_{Q_T} , которому соответствует минимальная сумма расстояний $S_{Q_T} = \min_i \{S_{Q_i}\}$.

Если $m + m' = n - 1$, то выбранный путь μ_{Q_T} является гамильтоновым путем в графе (X, U) и в то же время искомым приближенным решением задачи.

Если $m + m' < n - 1$, то найденный элементарный путь μ_{Q_T} не является гамильтоновым в графе (X, U) . В таком случае проведем новую коррекцию пути μ_{Q_T} следующим образом.

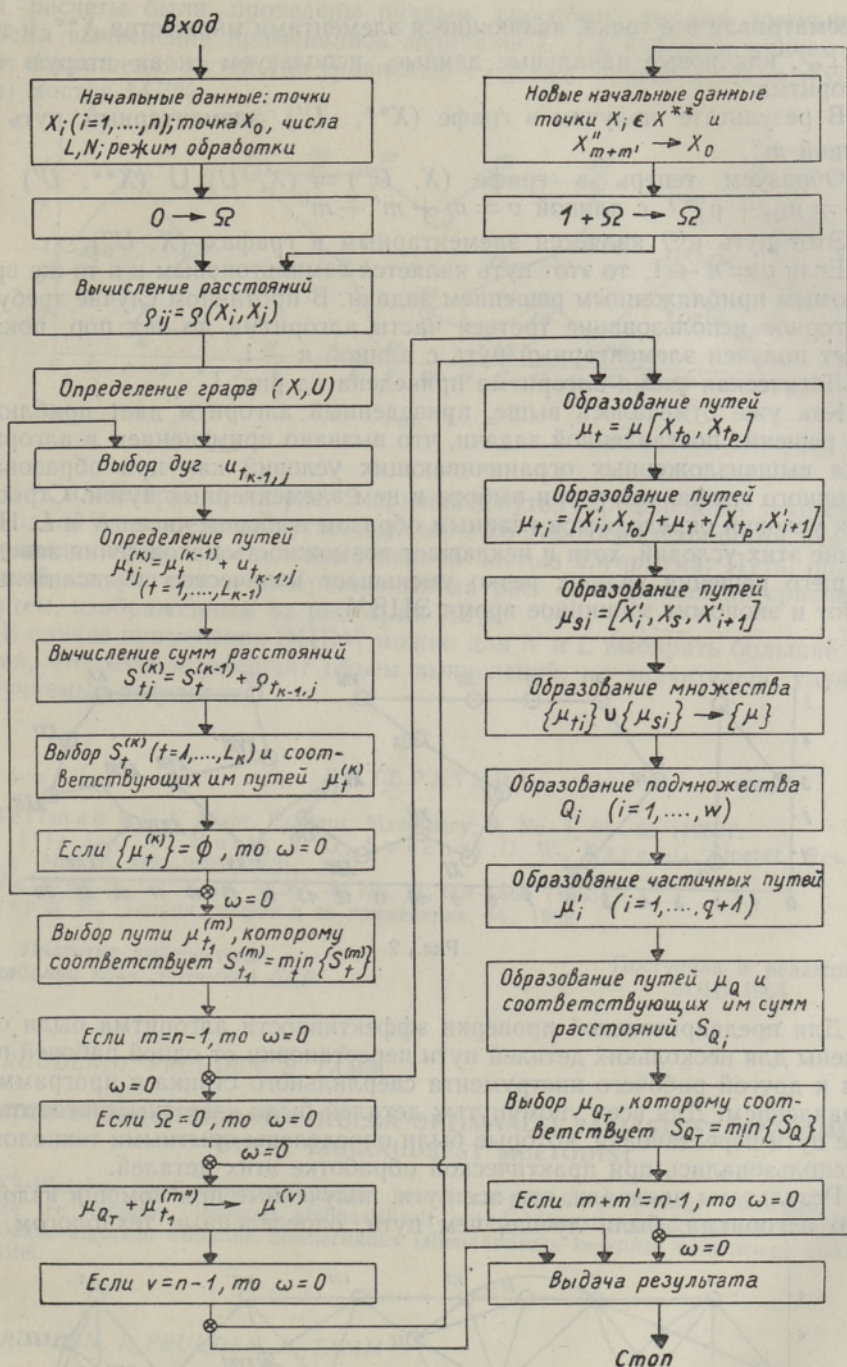


Рис. 1.

Определяем множество точек X'' , по которому проходит путь μ_{QT}

$$X'' = [X_0, X_1, \dots, X_{m+m'}], \text{ где } X_0 = X_0,$$

и множество точек $X^{**} = X \setminus X''$.

Рассматривая все точки, являющиеся элементами множества X^{**} , и точку $X_{m+m'}$, как новые начальные данные, используем снова вторую часть алгоритма.

В результате получим в графе (X^{**}, U') элементарный путь $\mu_i^{(m')}$ длиной m'' .

Образует теперь в графе $(X, U'') = (X, U) \cup (X^{**}, U')$ путь $\mu^{(v)} = \mu_{QT} + \mu_i^{(m')}$ с длиной $v = m + m' + m''$.

Этот путь $\mu^{(v)}$ является элементарным в графах (X, U'')

Если $v = n - 1$, то этот путь является гамильтоновым и в то же время искомым приближенным решением задачи. В противном случае требуется повторное использование третьей части алгоритма до тех пор, пока не будет получен элементарный путь с длиной $n - 1$.

Логическая схема алгоритма приведена на рис. 1.

Как уже отмечалось выше, приведенный алгоритм дает приближенное решение поставленной задачи, что вызвано применением в алгоритме ряда вышеизложенных ограничивающих условий как при образовании исходного графа, так и при выборе в нем элементарных путей. Строгость этих условий определяется главным образом выбором чисел N и L . Наложение этих условий, хотя и исключает возможность нахождения заведомо лучшего решения задачи, резко уменьшает количество вычислительных работ и экономит машинное время ЭЦВМ.

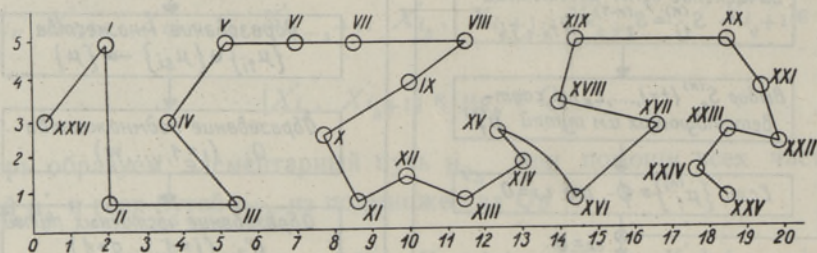


Рис. 2.

Для предварительной проверки эффективности алгоритма были определены для нескольких деталей пути перестановки от одной рабочей позиции к другой рабочего инструмента сверлильного станка с программным управлением. Для всех упомянутых деталей были известны соответствующие пути перестановки, которые были определены опытными технологами и использовались при практической обработке этих деталей.

Результаты показали, что все пути, полученные при помощи изложенного алгоритма, были лучше, чем пути, определенные технологом. Так

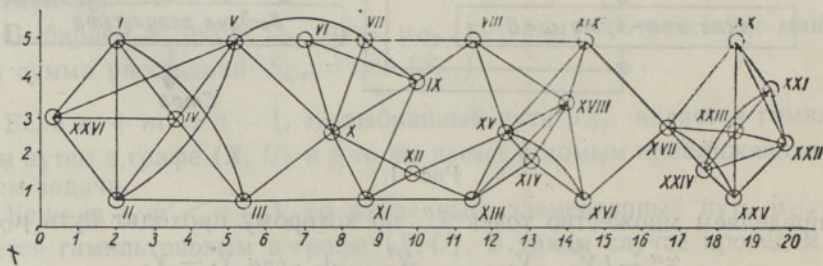


Рис. 3.

как расчеты были проведены ручным способом, то для уменьшения объема вычислений применялись величины $L = 4$ и $N = 4$, причем процент сокращения путей (по сравнению с путями, определенными технологом) достиг 14,7%.

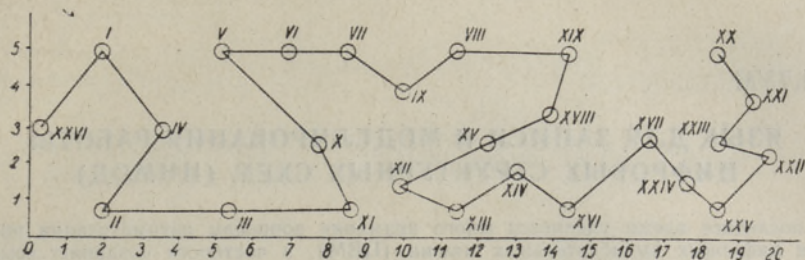


Рис. 4.

Для иллюстрации на рис. 2 приведен путь, определенный технологом для одной из вышеупомянутых деталей с 26 рабочими позициями. На рис. 3 приведен граф, определенный первой частью алгоритма. Путь, определенный алгоритмом и изображенный на рис. 4, сокращен по сравнению с путем, изображенным на рис. 2, на 7,2%.

В случае применения ЭЦВМ можно для N и L выбирать большие значения, что хотя и увеличит объем вычислений, но значительно улучшит получаемый результат.

ЛИТЕРАТУРА

1. Bellman R., J. Assoc. Comput. Machinery, 9, No. 1, 61—63 (1962).
2. Little J. D. C., Murty K. G., Sweeney D. W., Karel C., Operat. Res., 11, No. 10, 972—989 (1963).
3. Weimer D. L., "Com. ACM", 6, No. 11, 664—666 (1963).
4. Берж К., Теория графов и ее применения, М., 1962.

Институт кибернетики
Академии наук Эстонской ССР

Поступила в редакцию
31/III 1965

E. PRUUDEN, J. PRUUDEN, B. TAMM

ANTUD TÖÖPOSITSIOONIDE HULGA OPTIMAALSE JÄRJESTUSE MÄÄRAMISE ÜHEST LIGIKAUDSEST MEETODIST

Käsitletakse programmjuhtimisega koordinaat-positsioon-tööpingi tööorgani optimaalse tee leidmise ülesannet stohhastiliselt antud tööpositsioonide hulga töötlemisel. Esitatakse algoritm ülesande insenerlikuks lahendamiseks keskmise võimsusega elektron-arvutitel.

E. PRUUDEN, J. PRUUDEN, B. TAMM

ON AN APPROXIMATE METHOD FOR DETERMINING OPTIMAL SUCCESSION IN PREDESIGNED ENSEMBLE OF TOOLING POSITIONS

This paper deals with the problem of defining an optimal route for the instrument of numerically controlled positioning system in following stochastically pre-designed ensemble of tooling positions. An algorithm for constructive solution of the problem by using numerical computers is suggested.