

УДК 517.6

© А. Г. Ченцов, П. А. Ченцов

**ОБ ОДНОЙ ЗАДАЧЕ ПОСЛЕДОВАТЕЛЬНОГО ОБХОДА МНОЖЕСТВ**

Исследуется задача последовательного обхода мегаполисов с условиями предшествования, ориентированная на применение в машиностроении при листовой резке деталей на машинах с ЧПУ. Имеется следующая особенность постановки: терминальная компонента аддитивного критерия содержит зависимость от стартовой точки. Данная особенность приводит к тому, что естественная процедура решения на основе динамического программирования должна применяться индивидуально для каждой точки старта. Целью исследования является построение оптимизирующего алгоритма для определения комплекса, включающего маршрут (способ нумерации мегаполисов), траекторию и точку старта. Предложенный алгоритм реализует идею направленного перебора точек старта. Алгоритм реализован в виде стандартной программы для ПЭВМ; решены модельные примеры.

*Ключевые слова:* маршрутная оптимизация, динамическое программирование, оптимизация точки старта.

DOI: [10.35634/vm210310](https://doi.org/10.35634/vm210310)**Введение**

В статье рассматривается задача, подобная [1, 2]: по постановке требуется осуществить посещение конечной системы мегаполисов (непустых конечных множеств) с выполнением при упомянутых посещениях некоторых работ при соблюдении условий предшествования; в формировании критерия присутствует терминальная компонента, зависящая (как от параметра) от точки старта. Требуется оптимизировать точку старта, очередность посещения мегаполисов и конкретную траекторию процесса. Для решения используется динамическое программирование (ДП), которое в данной задаче должно быть [2] индивидуальным, т. е. своим для каждой возможной точки старта. В результате возникает необходимость перебора точек старта с построением для каждой из них весьма трудоемкой процедуры на основе ДП. Отметим, что в [1, 2] указан вариант применения универсальных (относительно точек старта) вариантов ДП с целью сокращения упомянутого перебора; был построен алгоритм, реализованный на ПЭВМ. Тем не менее в целом ряде случаев существенно сократить перебор не удастся и требуется разработка новых алгоритмов, включая эвристические, для построения работоспособных процедур комплексной оптимизации. Одно из таких направлений развивается в настоящей работе.

Естественным прототипом рассматриваемой ниже задачи маршрутизации является известная труднорешаемая задача коммивояжера или TSP в англоязычных публикациях; см. [3–8] и др. В связи с применением ДП для решения задачи коммивояжера отметим [9, 10]. Аналогом исследуемой ниже задачи может служить замкнутая метрическая задача коммивояжера при условии, что положение одного из городов, подлежащих посещению можно варьировать в интересах оптимизации. В то же время исследуемая далее постановка содержит целый ряд существенных особенностей не только количественного, но и качественного характера. Среди этих особенностей особо отметим эффект ограниченный, среди которых выделяются условия предшествования. В своем исследовании мы ориентируемся на идеи ДП в реализации, являющейся существенным развитием подхода [9]; при этом условия предшествования удастся задействовать для снижения вычислительной

сложности. Применяемые ниже процедуры ориентированы на задачу управления инструментом при фигурной листовой резке на машинах с ЧПУ.

## § 1. Общие понятия и обозначения

В дальнейшем используется стандартная теоретико-множественная символика (кванторы, пропозициональные связки и др.); через  $\triangleq$  обозначаем равенство по определению,  $\emptyset$  — пустое множество. Семейством называем множество, все элементы которого сами являются множествами. Если  $x$  и  $y$  — объекты, то  $\{x; y\}$  — множество, содержащее  $x$ ,  $y$  и не содержащее никаких других элементов ( $\{x; y\}$  — неупорядоченная пара объектов  $x$ ,  $y$ ). Для любого объекта  $z$  в виде  $\{z\} \triangleq \{z; z\}$  имеем синглетон, содержащий  $z$ , т. е.  $z \in \{z\}$ . Для любых двух объектов  $a$  и  $b$  определена [11, с. 67] их упорядоченная пара (УП)  $(a, b) \triangleq \{\{a\}; \{a; b\}\}$  с первым элементом  $a$  и вторым элементом  $b$ . Если же  $h$  — какая-либо УП, то через  $\text{pr}_1(h)$  и  $\text{pr}_2(h)$  обозначаем соответственно первый и второй элементы  $h$ ,  $h = (\text{pr}_1(h), \text{pr}_2(h))$ . Для любых трех объектов  $a$ ,  $b$  и  $c$  в виде  $(a, b, c) \triangleq ((a, b), c)$  имеем их упорядоченный триплет. Соответственно для трех множеств  $A$ ,  $B$  и  $C$  полагается  $A \times B \times C \triangleq (A \times B) \times C$ . Данные соглашения традиционны; см., например, [12, гл. I].

Множеству  $H$  сопоставляем семейство  $\mathcal{P}(H)$  всех подмножеств (п/м)  $H$  и полагаем, что  $\mathcal{P}'(H) \triangleq \mathcal{P}(H) \setminus \{\emptyset\}$  (семейство всех непустых п/м  $H$ ); через  $\text{Fin}(H)$  обозначаем семейство всех конечных множеств из  $\mathcal{P}'(H)$ , т. е. семейство всех непустых конечных п/м  $H$ . Если само  $H$  — непустое конечное множество, то  $\text{Fin}(H) = \mathcal{P}'(H)$ .

Для любых двух непустых множеств  $A$  и  $B$  через  $B^A$  обозначаем множество всех отображений (функций) из  $A$  в  $B$ ; при  $f \in B^A$  и  $a \in A$  в виде  $f(a) \in B$  имеем, как обычно, значение  $f$  в точке  $a$ . Выражения  $f \in B^A$  и  $f : A \rightarrow B$  тождественны. Если  $A$  и  $B$  — непустые множества,  $f \in B^A$  и  $C \in \mathcal{P}(A)$ , то  $f^1(C) \triangleq \{f(x) : x \in C\} \in \mathcal{P}(B)$  есть образ  $C$  при действии  $f$ .

Полагаем, что  $\mathbb{R}$  — вещественная прямая,  $\mathbb{R}_+ \triangleq \{\xi \in \mathbb{R} \mid 0 \leq \xi\} = [0, \infty[$ ,  $\mathbb{N} \triangleq \{1; 2; \dots\}$ ,  $\mathbb{N}_0 \triangleq \{0\} \cup \mathbb{N} = \{0; 1; 2; \dots\}$ ;

$$\overline{p, q} \triangleq \{k \in \mathbb{N}_0 \mid (p \leq k) \& (k \leq q)\} \quad \forall p \in \mathbb{N}_0 \quad \forall q \in \mathbb{N}_0.$$

Если  $S$  — непустое множество, то через  $\mathcal{R}_+[S]$  обозначаем множество всех неотрицательных вещественнозначных (в/з) функций на  $S$ :  $\mathcal{R}_+[S] \triangleq (\mathbb{R}_+)^S$ .

Если  $K$  — непустое конечное множество, то  $|K| \in \mathbb{N}$  есть мощность (количество элементов)  $K$ ;  $|\emptyset| \triangleq 0$ . Кроме того, для непустого конечного множества  $K$  через  $(\text{bi})[K]$  обозначаем (непустое) множество всех биекций промежутка  $\overline{1, |K|}$  на  $K$  ( $(\text{bi})[K]$  — множество всех вариантов нумерации без повторений элементов  $K$  числами  $1, \dots, |K|$ ). Если  $m \in \mathbb{N}$ , то  $(\text{bi})[\overline{1, m}]$  есть множество всех перестановок элементов множества  $\overline{1, m}$ ; если  $\alpha \in (\text{bi})[\overline{1, m}]$ , то определена перестановка  $\alpha^{-1} \in (\text{bi})[\overline{1, m}]$ , для которой

$$\alpha(\alpha^{-1}(k)) = \alpha^{-1}(\alpha(k)) = k \quad \forall k \in \overline{1, m}.$$

## § 2. Постановка задачи

Фиксируем непустое множество  $X$  и  $X^0 \in \text{Fin}(X)$ . Кроме того, пусть  $N \in \mathbb{N}$  таково, что  $2 \leq N$ . Далее, фиксируем множества

$$M_1 \in \text{Fin}(X), \quad \dots, \quad M_N \in \text{Fin}(X), \quad (2.1)$$

именуемые ниже мегаполисами. Полагаем, что

$$(X^0 \cap M_j = \emptyset \quad \forall j \in \overline{1, N}) \& (M_p \cap M_q = \emptyset \quad \forall p \in \overline{1, N} \quad \forall q \in \overline{1, N} \setminus \{p\}).$$

Наконец, фиксируем  $N$  (непустых) отношений

$$\mathbb{M}_1 \in \mathcal{P}'(M_1 \times M_1), \quad \dots, \quad \mathbb{M}_N \in \mathcal{P}'(M_N \times M_N).$$

Итак,  $\emptyset \neq \mathbb{M}_1 \subset M_1 \times M_1, \quad \dots, \quad \emptyset \neq \mathbb{M}_N \subset M_N \times M_N$ . Посещение мегаполисов (2.1) связывается с выполнением определенных работ, именуемых далее внутренними: при  $j \in \overline{1, N}$  и  $z \in \mathbb{M}_j$  УП  $z$  определяет соответствующий вариант выполнения внутренних работ. Введем теперь при  $j \in \overline{1, N}$

$$(\mathfrak{M}_j \triangleq \{ \text{pr}_1(z) : z \in \mathbb{M}_j \} \in \mathcal{P}'(M_j)) \& (\mathbf{M}_j \triangleq \{ \text{pr}_2(z) : z \in \mathbb{M}_j \} \in \mathcal{P}'(M_j)),$$

выделяя для отдельного рассмотрения возможные точки прибытия в мегаполис  $M_j$  и отправления из  $M_j$  соответственно. Кроме того, имеем

$$(\mathbb{X} \triangleq X^0 \cup (\bigcup_{i=1}^N \mathfrak{M}_i) \in \text{Fin}(X)) \& (\mathbf{X} \triangleq X^0 \cup (\bigcup_{i=1}^N \mathbf{M}_i) \in \text{Fin}(X)). \quad (2.2)$$

Множества (2.2) фактически заменяют  $X$  в последующих построениях. Рассматриваемые далее процессы имеют вид

$$\begin{aligned} (x^0 \in X^0) &\rightarrow (x_{1,1} \in \mathfrak{M}_{\alpha(1)} \rightsquigarrow x_{1,2} \in \mathbf{M}_{\alpha(1)}) \rightarrow \\ &\rightarrow \dots \rightarrow (x_{N,1} \in \mathfrak{M}_{\alpha(N)} \rightsquigarrow x_{N,2} \in \mathbf{M}_{\alpha(N)}), \end{aligned} \quad (2.3)$$

где  $\alpha$  — перестановка индексов из  $\overline{1, N}$ , прямые стрелки обозначают внешние перемещения, а волнистые — перемещения при выполнении внутренних работ;  $x^0, \alpha, x_{1,1}, x_{1,2}, \dots, x_{N,1}, x_{N,2}$  выбираются исследователем в целях оптимизации соответствующего (здесь — аддитивного) критерия. Предполагается, что выбор  $\alpha$  может удовлетворять некоторым условиям предшествования.

Переходя к формализации, условимся, что  $\mathbb{P} \triangleq (\text{bi})[\overline{1, N}]$ . Для введения ограничений на выбор  $\alpha$  фиксируем множество  $\mathbf{K} \in \mathcal{P}(\overline{1, N} \times \overline{1, N})$  (итак,  $\mathbf{K}$  — множество, для которого  $\mathbf{K} \subset \overline{1, N} \times \overline{1, N}$ ) со следующим свойством:

$$\forall \mathbf{K}_0 \in \mathcal{P}'(\mathbf{K}) \exists z_0 \in \mathbf{K}_0 : \text{pr}_1(z_0) \neq \text{pr}_2(z_0) \quad \forall z \in \mathbf{K}_0. \quad (2.4)$$

Условие (2.4) подробно обсуждалось в [13, часть 2]; оно не является ограничительным в типичных случаях рассматриваемой задачи. Тогда

$$\mathbf{A} \triangleq \{ \alpha \in \mathbb{P} \mid \alpha^{-1}(\text{pr}_1(z)) < \alpha^{-1}(\text{pr}_2(z)) \quad \forall z \in \mathbf{K} \}$$

есть множество всех  $\mathbf{K}$ -допустимых (допустимых по предшествованию) перестановок.

**Замечание.** Случай  $\mathbf{K} = \emptyset$  из рассмотрения не исключается и соответствует ситуации, когда условия предшествования отсутствуют.

Введем в рассмотрение траектории, соответствующие (2.3) на идейном уровне. Для этого сначала введем множество  $\mathbb{Z}$  всех отображений из  $\overline{0, N}$  в  $\mathbb{X} \times \mathbf{X}$  (итак,  $\mathbb{Z}$  есть множество всех кортежей  $(z_i)_{i \in \overline{0, N}}$ , где  $z_0 \in \mathbb{X} \times \mathbf{X}, z_1 \in \mathbb{X} \times \mathbf{X}, \dots, z_N \in \mathbb{X} \times \mathbf{X}$ ). При  $x \in X^0$  и  $\alpha \in \mathbb{P}$  полагаем, что

$$\mathbf{Z}_\alpha[x] \triangleq \{ (z_i)_{i \in \overline{0, N}} \in \mathbb{Z} \mid (z_0 = (x, x)) \& (z_t \in \mathbb{M}_{\alpha(t)} \quad \forall t \in \overline{1, N}) \}, \quad (2.5)$$

отражая специфику (2.3); ясно, что  $\mathbf{Z}_\alpha[x] \in \text{Fin}(\mathbb{Z})$ . В виде (2.5) имеем пучок траекторий (трасс), согласованных с  $\alpha$  и стартующих из  $x$ . В дальнейшем перестановки из множества  $\mathbb{P}$  условимся именовать маршрутами, следуя традиции TSP. В виде

$$\tilde{D}[x] \triangleq \{(\alpha, (z_i)_{i \in \overline{0, N}}) \in \mathbf{A} \times \mathbb{Z} \mid (z_i)_{i \in \overline{0, N}} \in \mathbf{Z}_\alpha[x]\} \in \text{Fin}(\mathbf{A} \times \mathbb{Z}), \quad (2.6)$$

где  $x \in X^0$ , имеем множество всех допустимых решений (ДР) с точкой старта  $x$ . Наконец, в виде

$$\mathbf{D} \triangleq \{(\alpha, (z_i)_{i \in \overline{0, N}}, x) \in \mathbf{A} \times \mathbb{Z} \times X^0 \mid (\alpha, (z_i)_{i \in \overline{0, N}}) \in \tilde{D}[x]\}$$

имеем множество всех ДР формулируемой ниже полной задачи;  $\mathbf{D} \in \text{Fin}(\mathbf{A} \times \mathbb{Z} \times X^0)$ . Триплеты — элементы  $\mathbf{D}$  — будем называть маршрутными процессами.

**Функции стоимости.** Учитываем (2.2). Пусть  $\mathfrak{N} \triangleq \mathcal{P}'(\overline{1, N})$ ; итак,  $\mathfrak{N}$  есть семейство всех непустых п/м  $\overline{1, N}$ , именуемых ниже списками (заданий). Фиксируем следующие  $N+2$  функции:

$$c \in \mathcal{R}_+[\mathbf{X} \times \mathbb{X} \times \mathfrak{N}], \quad c_1 \in \mathcal{R}_+[\mathbf{M}_1 \times \mathfrak{N}], \quad \dots, \quad c_N \in \mathcal{R}_+[\mathbf{M}_N \times \mathfrak{N}], \quad f \in \mathcal{R}_+[\mathbf{M} \times X^0], \quad (2.7)$$

где (здесь и ниже)  $\mathbf{M}$  есть объединение всех множеств  $\mathbf{M}_i$ ,  $i \in \overline{1, N}$ . Функция  $c$  используется для оценивания внешних перемещений (между мегаполисами, а также из  $X^0$  к мегаполисам). Значения функций  $c_1, \dots, c_N$  оценивают внутренние работы, связанные с посещением мегаполисов. Наконец, функция  $f$  используется при оценивании терминального состояния (см.  $x_{N,2}$  в (2.3)); она включает зависимость от точки старта (элемент  $X^0$ ). Если  $x \in X^0$ ,  $\alpha \in \mathbb{P}$  и  $(z_i)_{i \in \overline{0, N}} \in \mathbf{Z}_\alpha[x]$ , то

$$\mathfrak{C}_\alpha[(z_i)_{i \in \overline{0, N}}|x] \triangleq \sum_{i=1}^N [c(\text{pr}_2(z_{i-1}), \text{pr}_1(z_i), \alpha^1(\overline{i, N})) + c_{\alpha(i)}(z_i, \alpha^1(\overline{i, N}))] + f(\text{pr}_2(z_N), x) \quad (2.8)$$

(в (2.8) используется обозначение раздела 1 для образа множества; маршруты являются перестановками и, в частности, функциями). Если  $x \in X^0$ , то (см. (2.6)) получаем  $x$ -задачу

$$\mathfrak{C}_\alpha[(z_i)_{i \in \overline{0, N}}|x] \longrightarrow \min, \quad (\alpha, (z_i)_{i \in \overline{0, N}}) \in \tilde{D}[x], \quad (2.9)$$

для которой определен экстремум  $V[x]$  в виде наименьшего из чисел

$$\mathfrak{C}_\alpha[(z_i)_{i \in \overline{0, N}}|x], \quad (\alpha, (z_i)_{i \in \overline{0, N}}) \in \tilde{D}[x],$$

а также экстремальное множество

$$(\text{SOL})[x] \triangleq \{(\alpha^0, (z_i^0)_{i \in \overline{0, N}}) \in \tilde{D}[x] \mid \mathfrak{C}_{\alpha^0}[(z_i^0)_{i \in \overline{0, N}}|x] = V[x]\} \in \mathcal{P}'(\tilde{D}[x]); \quad (2.10)$$

из (2.6), (2.10) следует, конечно, что  $(\text{SOL})[x]$  есть непустое конечное множество. В виде (2.9) имеем частную задачу экстремальной маршрутизации. Полная (основная) задача имеет вид

$$\mathfrak{C}_\alpha[(z_i)_{i \in \overline{0, N}}|x] \longrightarrow \min, \quad (\alpha, (z_i)_{i \in \overline{0, N}}, x) \in \mathbf{D}; \quad (2.11)$$

она характеризуется (глобальным) экстремумом

$$\mathbb{V} \triangleq \min_{(\alpha, (z_i)_{i \in \overline{0, N}}, x) \in \mathbf{D}} \mathfrak{C}_\alpha[(z_i)_{i \in \overline{0, N}}|x] \in \mathbb{R}_+ \quad (2.12)$$

и (непустым конечным) экстремальным множеством маршрутных процессов:

$$\text{SOL} \triangleq \{(\alpha, (z_i)_{i \in \overline{0, N}}, x) \in \mathbf{D} \mid \mathfrak{C}_\alpha[(z_i)_{i \in \overline{0, N}}|x] = \mathbb{V}\} \in \text{Fin}(\mathbf{D}). \quad (2.13)$$

Итак, ДР задачи (2.11) являются маршрутными процессами. Отметим, наконец, следующую задачу:

$$V[x] \longrightarrow \min, \quad x \in X^0,$$

для которой  $\mathbb{V}$  (2.12) является экстремумом, т. е. имеет место равенство

$$\mathbb{V} = \min_{x \in X^0} V[x], \quad (2.14)$$

а множество всех оптимальных ДР (экстремальное множество) есть

$$X_{\text{opt}}^0 \triangleq \{x^0 \in X^0 | V[x^0] = \mathbb{V}\} \in \mathcal{P}'(X^0); \quad (2.15)$$

ясно, что  $X_{\text{opt}}^0 \in \text{Fin}(X^0)$ . Конечно, нашей главной целью является нахождение  $\mathbb{V}$  и какого-либо оптимального решения из множества (2.13). Однако, ее достижение сопряжено с большими трудностями вычислительного характера. Одно из новых, в сравнении с [13, 14], осложняющих обстоятельств связано с зависимостью последней в (2.7) функции от точки старта; мы имеем здесь прежде всего осложнение с применением ДП, где возникает необходимость решения индивидуальных относительно точки старта задач (см. [2, раздел 4]). Мы напомним построения [2, раздел 4] с тем, чтобы прояснить упомянутое обстоятельство. Затем будет предложен оптимизирующий алгоритм для решения задачи (2.11), реализованный в виде стандартной программы для ПЭВМ, и рассмотрено решение модельных примеров.

### § 3. Схема индивидуального динамического программирования

В настоящем разделе в краткой форме рассматривается схема построения оптимального решения на основе ДП, включающая перебор точек из множества  $X^0$ . Мы ограничиваемся здесь алгоритмом на функциональном уровне, отсылая к [2] за подробностями.

Итак, введем оператор вычеркивания (см. [13, часть 2])  $\mathbf{I}$ , действующий в  $\mathfrak{N}$  по правилу: если  $K \in \mathfrak{N}$ , то при  $\Xi[K] \triangleq \{z \in \mathbf{K} | (\text{pr}_1(z) \in K) \& (\text{pr}_2(z) \in K)\}$  полагаем

$$\mathbf{I}(K) \triangleq K \setminus \{\text{pr}_2(z) : z \in \Xi[K]\}.$$

Данный оператор использовался в [1, 2, 13–15] и во многих других работах. Рассмотрим основные этапы построения оптимального алгоритма.

1) **Построение существенных списков заданий.** Напомним, что списками мы называем элементы  $\mathfrak{N}$ , т. е. непустые п/м  $\overline{1, N}$ . Существенными называем списки из семейства

$$\mathcal{G} \triangleq \{K \in \mathfrak{N} | \forall z \in \mathbf{K} \quad (\text{pr}_1(z) \in K) \implies (\text{pr}_2(z) \in K)\};$$

простой пример существенного списка доставляет множество  $\overline{1, N}$ . При  $s \in \overline{1, N}$  полагаем, что  $\mathcal{G}_s \triangleq \{K \in \mathcal{G} | s = |K|\}$  (семейство всех  $s$ -элементных существенных списков). Ясно, что  $\mathcal{G}_N = \{\overline{1, N}\}$  и  $\mathcal{G}_1 = \{\{t\} : t \in \overline{1, N} \setminus \mathbf{K}_1\}$ , где  $\mathbf{K}_1 \triangleq \{\text{pr}_1(z) : z \in \mathbf{K}\}$ . Кроме того, как легко видеть (см. [16])

$$\mathcal{G}_{s-1} = \{K \setminus \{t\} : K \in \mathcal{G}_s, t \in \mathbf{I}(K)\} \quad \forall s \in \overline{2, N}.$$

Тогда в виде  $\mathcal{G}_N \rightarrow \mathcal{G}_{N-1} \rightarrow \dots \rightarrow \mathcal{G}_1$  имеем рекуррентную процедуру построения существенных списков.

2) **Построение слоев пространства позиций.** Фиксируем  $x^0 \in X^0$  в качестве точки старта. Вводим в рассмотрение множества  $D_0, D_1, \dots, D_N$ . Проще всего определяются  $D_0$  и  $D_N$ . Итак, полагаем, что

$$(D_0 \triangleq \{(x, \emptyset) : x \in \bigcup_{i \in \overline{1, N} \setminus \mathbf{K}_1} \mathbf{M}_i\}) \& (D_N \triangleq \{(x^0, \overline{1, N})\}). \quad (3.1)$$

Если  $s \in \overline{1, N-1}$  и  $K \in \mathcal{G}_s$ , то последовательно определяем множества

$$\begin{aligned} \mathcal{J}_s(K) &\triangleq \{t \in \overline{1, N} \setminus K \mid \{t\} \cup K \in \mathcal{G}_{s+1}\}, \\ \mathcal{M}_s[K] &\triangleq \bigcup_{j \in \mathcal{J}_s(K)} \mathbf{M}_j, \quad \mathbb{D}_s[K] \triangleq \{(x, K) : x \in \mathcal{M}_s[K]\}. \end{aligned} \quad (3.2)$$

С учетом (3.2) полагаем при  $s \in \overline{1, N-1}$ , что множество  $D_s$  имеет вид

$$D_s \triangleq \bigcup_{K \in \mathcal{G}_s} \mathbb{D}_s[K],$$

получая непустое множество (см. [16]). В итоге получаем непустые множества  $D_0, D_1, \dots, D_N$ . При этом  $D_0 \subset \mathbf{X} \times \{\emptyset\}$ , а  $D_j \in \mathcal{P}'(\mathbf{X} \times \mathfrak{N}) \forall j \in \overline{1, N}$  (подробнее см. в [16, (3.30)]). Отметим также, что (см. [15, (3.5.17)])

$$(\text{pr}_2(z), K \setminus \{j\}) \in D_{s-1} \quad \forall s \in \overline{1, N} \quad \forall (x, K) \in D_s \quad \forall j \in \mathbf{I}(K) \quad \forall z \in \mathbb{M}_j. \quad (3.3)$$

3) **Построение слоев функции Беллмана.** Продолжаем построение решения задачи маршрутизации с фиксированной точкой старта. Последовательно строим функции  $v_0 \in \mathcal{R}_+[D_0]$ ,  $v_1 \in \mathcal{R}_+[D_1]$ ,  $\dots$ ,  $v_N \in \mathcal{R}_+[D_N]$ . Функция  $v_0 \in \mathcal{R}_+[D_0]$  определяется правилом

$$v_0(x, \emptyset) \triangleq f(x, x^0) \quad \forall x \in \bigcup_{i \in \overline{1, N} \setminus \mathbf{K}_1} \mathbf{M}_i. \quad (3.4)$$

Далее, пусть  $s \in \overline{1, N}$  и функция  $v_{s-1} \in \mathcal{R}_+[D_{s-1}]$  уже построена. В силу (3.3) это означает, в частности, что при  $(x, K) \in D_s$ ,  $j \in \mathbf{I}(K)$  и  $z \in \mathbb{M}_j$  определено значение  $v_{s-1}(\text{pr}_2(z), K \setminus \{j\}) \in \mathbb{R}_+$ . С учетом этого полагаем, что  $v_s \in \mathcal{R}_+[D_s]$  определяется правилом: при  $(x, K) \in D_s$

$$v_s(x, K) \triangleq \min_{j \in \mathbf{I}(K)} \min_{z \in \mathbb{M}_j} [c(x, \text{pr}_1(z), K) + c_j(z, K) + v_{s-1}(\text{pr}_2(z), K \setminus \{j\})]. \quad (3.5)$$

Данное правило определяет преобразование  $v_{s-1} \rightarrow v_s$  и, для своего исполнения, требует (наряду с информацией о функциях  $c$  и  $c_j$ ,  $j \in \mathbf{I}(K)$ ; имеется в виду случай, когда позиция  $(x, K)$  фиксирована) только знания  $v_{s-1}$ . С учетом (3.4), (3.5) получаем рекуррентную процедуру

$$v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_N, \quad (3.6)$$

для компьютерной реализации  $v_N$  достаточно сохранять (на каждом этапе процедуры) в памяти вычислителя только один слой функции Беллмана. С учетом (3.5) имеем, что получаемая в (3.6) функция  $v_N$  характеризуется единственным значением  $v_N(x^0, \overline{1, N})$ , которое совпадает с экстремумом нашей  $x^0$ -задачи, т. е.  $x$ -задачи (2.9) при  $x = x^0$ :

$$V[x^0] = v_N(x^0, \overline{1, N}). \quad (3.7)$$



В связи с (3.7) см., в частности, [15, (4.3.5), (4.3.8), (4.3.11)]. Заметим, что в силу (3.4) наша процедура (3.6) «привязана» к  $x^0$ , т. е. является индивидуальной. Варьируя  $x^0$ , мы получаем, конечно, возможность решения всех  $x$ -задач (2.9), что, однако, требует выполнения весьма затруднительной процедуры перебора на множестве  $X^0$  с реализацией на каждом ее этапе своего варианта (3.6).

**Замечание 1.** Подчеркнем, что в случае, рассмотренном при построении (3.6), (3.7), возможна компьютерная реализация с некоторой экономией ресурсов памяти, что уже фактически отмечалось после (3.6). В результате применения такой процедуры мы получим в итоге значение (3.7). Если это будет проделано для всех точек из  $X^0$ , мы получим в итоге (см. (2.14)) возможность найти  $\bar{V}$  (2.14) и точки множества (2.15), т. е. оптимальные точки старта, которые можно затем задействовать в сочетании с эвристиками. Подчеркнем, однако, что в случае, когда требуется построить сами оптимальные решения (см. (2.13)), процедуру (3.6) надо завершить не определением только одного значения (3.7), а построением всех функций  $v_0, v_1, \dots, v_N$ . Иными словами, при  $s \in \overline{1, N}$  по мере применения (3.5) необходимо сохранять в памяти функцию  $v_{s-1}$  наряду с вновь построенной функцией  $v_s$ .

**4) Построение оптимального решения частной задачи.** Продолжаем построение решения при фиксации  $x^0 \in X^0$ . А именно: рассмотрим построение элемента множества (2.10) при  $x = x^0$ . Полагаем, что  $z_0^{(0)} \triangleq (x^0, x^0)$ , получая УП из  $X^0 \times X^0$ . Коль скоро  $(x^0, \overline{1, N}) \in D_N$  (см. (3.1)), имеем из (3.5) и (3.7) равенство

$$V[x^0] = \min_{j \in \mathbf{I}(\overline{1, N})} \min_{z \in \mathbb{M}_j} [c(x^0, \text{pr}_1(z), \overline{1, N}) + c_j(z, \overline{1, N}) + v_{N-1}(\text{pr}_2(z), \overline{1, N} \setminus \{j\})].$$

С учетом этого находим  $\eta_1 \in \mathbf{I}(\overline{1, N})$  и  $z_1^{(0)} \in \mathbb{M}_{\eta_1}$ , для которых

$$V[x^0] = c(x^0, \text{pr}_1(z_1^{(0)}), \overline{1, N}) + c_{\eta_1}(z_1^{(0)}, \overline{1, N}) + v_{N-1}(\text{pr}_2(z_1^{(0)}), \overline{1, N} \setminus \{\eta_1\}). \quad (3.8)$$

Здесь согласно (3.3)  $(\text{pr}_2(z_1^{(0)}), \overline{1, N} \setminus \{\eta_1\}) \in D_{N-1}$ , а потому из (3.5) следует, что

$$v_{N-1}(\text{pr}_2(z_1^{(0)}), \overline{1, N} \setminus \{\eta_1\}) = \min_{j \in \mathbf{I}(\overline{1, N} \setminus \{\eta_1\})} \min_{z \in \mathbb{M}_j} [c(\text{pr}_2(z_1^{(0)}), \text{pr}_1(z), \overline{1, N} \setminus \{\eta_1\}) + c_j(z, \overline{1, N} \setminus \{\eta_1\}) + v_{N-2}(\text{pr}_2(z), \overline{1, N} \setminus \{\eta_1; j\})].$$

С учетом этого выбираем  $\eta_2 \in \mathbf{I}(\overline{1, N} \setminus \{\eta_1\})$  и  $z_2^{(0)} \in \mathbb{M}_{\eta_2}$ , для которых

$$v_{N-1}(\text{pr}_2(z_1^{(0)}), \overline{1, N} \setminus \{\eta_1\}) = c(\text{pr}_2(z_1^{(0)}), \text{pr}_1(z_2^{(0)}), \overline{1, N} \setminus \{\eta_1\}) + c_{\eta_2}(z_2^{(0)}, \overline{1, N} \setminus \{\eta_1\}) + v_{N-2}(\text{pr}_2(z_2^{(0)}), \overline{1, N} \setminus \{\eta_1; \eta_2\}), \quad (3.9)$$

где согласно (3.3)  $(\text{pr}_2(z_2^{(0)}), \overline{1, N} \setminus \{\eta_1; \eta_2\}) = (\text{pr}_2(z_2^{(0)}), (\overline{1, N} \setminus \{\eta_1\}) \setminus \{\eta_2\}) \in D_{N-2}$ . Далее процедуры, подобные (3.8), (3.9), следует продолжать вплоть до исчерпывания  $\overline{1, N}$ . В итоге будут построены маршрут  $\eta \triangleq (\eta_j)_{j \in \overline{1, N}} \in \mathbf{A}$  и траектория  $(z_i^{(0)})_{i \in \overline{0, N}} \in \mathbf{Z}_\eta[x^0]$ , для которых

$$\mathfrak{C}_\eta[(z_i^{(0)})_{i \in \overline{0, N}}] = V[x^0]; \quad (3.10)$$

иными словами,  $(\eta, (z_i^{(0)})_{i \in \overline{0, N}}) \in (\text{SOL})[x^0]$ , чем и завершается построение оптимального решения задачи (2.9), где  $x = x^0$ .

#### § 4. Оптимизирующий алгоритм выбора точки старта

Уже при умеренном значении  $|X^0|$  решение задачи (2.11) с помощью процедур, подобных рассмотренным в разделе 3 и различающихся выбором точки старта, является достаточно трудной переборной задачей. Логично поставить вопрос о «точечном» применении аппарата ДП (см. раздел 3). Ниже рассматривается один из вариантов такого применения.

В дальнейшем кортежи из  $\mathbb{Z}$  будем, как правило, обозначать одной буквой: при  $\mathbf{z} \in \mathbb{Z}$  имеется в виду, что  $\mathbf{z} = (z_i)_{i \in \overline{0, N}}$ , где  $z_j \in \mathbb{X} \times \mathbf{X}$  при  $j \in \overline{0, N}$ . Если  $\mathbf{z} \in \mathbb{Z}$  и  $x \in X^0$ , то полагаем, что отображение  $\mathbf{z}|_x \in \mathbb{Z}$  таково, что

$$(\mathbf{z}|_x(0) \triangleq (x, x)) \& (\mathbf{z}|_x(t) \triangleq \mathbf{z}(t) \quad \forall t \in \overline{1, N}). \quad (4.1)$$

Мы обращаемся здесь с элементами  $\mathbb{Z}$  как с обычными функциями, учитывая (2.2). В качестве элементов  $\mathbb{Z}$  можно, конечно, использовать точки множеств (2.5), т. е. траектории. Вполне очевидно следующее

**Предложение 1.** Если  $x \in X^0$ ,  $\alpha \in \mathbb{P}$ ,  $\mathbf{z} \in \mathbb{Z}_\alpha[x]$  и  $y \in X^0$ , то

$$\mathbf{z}|_y \in \mathbb{Z}_\alpha[y]. \quad (4.2)$$

**Доказательство.** Фиксируем  $x$ ,  $\alpha$ ,  $\mathbf{z}$  и  $y$  в соответствии с условиями, полагая для краткости, что  $y \triangleq \mathbf{z}|_y$ . Согласно (2.5) имеем  $\mathbf{z} : \overline{0, N} \rightarrow \mathbb{X} \times \mathbf{X}$  и при этом

$$(\mathbf{z}(0) = (x, x)) \& (\mathbf{z}(t) \in \mathbb{M}_{\alpha(t)} \quad \forall t \in \overline{1, N}). \quad (4.3)$$

Напомним, что согласно (2.5) справедливо следующее равенство

$$\mathbb{Z}_\alpha[y] = \{\mathbf{h} \in \mathbb{Z} \mid (\mathbf{h}(0) = (y, y)) \& (\mathbf{h}(t) \in \mathbb{M}_{\alpha(t)} \quad \forall t \in \overline{1, N})\}. \quad (4.4)$$

В силу (4.1) имеем следующие свойства кортежа  $y$  :

$$(y(0) = (y, y)) \& (y(t) = \mathbf{z}(t) \quad \forall t \in \overline{1, N}). \quad (4.5)$$

Из (4.3) и (4.5) получаем теперь, что

$$y(t) \in \mathbb{M}_{\alpha(t)} \quad \forall t \in \overline{1, N}. \quad (4.6)$$

Поскольку  $y \in \mathbb{Z}$ , получаем (см. (4.4)–(4.6)) требуемое включение (4.2).  $\square$

Рассмотрим следующий оптимизирующий алгоритм для решения задачи (2.11). Итак, выберем произвольную точку  $x_0 \in X^0$ . Используя процедуру раздела 3 при условии  $x^0 = x_0$ , получим (на основе метода ДП) значение  $V[x_0]$  и решение  $(\eta_0, \mathbf{z}_0) \in (\text{SOL})[x_0]$ , реализующее  $V[x_0]$  (см. (3.10)). Тогда согласно предложению 1 при  $y \in X^0$  имеем, что  $\mathbf{z}_0|_y \in \mathbb{Z}_{\eta_0}[y]$ . Рассмотрим один вариант задачи (2.9) при  $x = y$ , а именно: определим

$$\begin{aligned} \mathfrak{C}_{\eta_0}[\mathbf{z}_0|_y] = & \sum_{i=1}^N [c(\text{pr}_2(\mathbf{z}_0|_y(i-1)), \text{pr}_1(\mathbf{z}_0|_y(i)), (\eta_0)^1(\overline{i, N})) + \\ & + c_{\eta_0(i)}(\mathbf{z}_0|_y(i), (\eta_0)^1(\overline{i, N}))] + f(\text{pr}_2(\mathbf{z}_0|_y(N)), y). \end{aligned} \quad (4.7)$$

Отметим здесь же, что согласно (2.8) справедливо следующее равенство

$$\begin{aligned} \mathfrak{C}_{\eta_0}[\mathbf{z}_0|x_0] = & \sum_{i=1}^N [c(\text{pr}_2(\mathbf{z}_0(i-1)), \text{pr}_1(\mathbf{z}_0(i)), (\eta_0)^1(\overline{i, N})) + \\ & + c_{\eta_0(i)}(\mathbf{z}_0(i), (\eta_0)^1(\overline{i, N}))] + f(\text{pr}_2(\mathbf{z}_0(N)), x_0) = V[x_0]. \end{aligned} \quad (4.8)$$



Легко видеть, что (см. (4.1)) из (4.7), (4.8) вытекают равенства

$$\begin{aligned} \mathfrak{C}_{\eta_0}[\mathbf{z}_0|y|y] &= \mathfrak{c}(y, \text{pr}_1(\mathbf{z}_0(1)), (\eta_0)^1(\overline{1, N})) + c_{\eta_0(1)}(\mathbf{z}_0(1), (\eta_0)^1(\overline{1, N})) + \\ &+ \sum_{i=2}^N [\mathfrak{c}(\text{pr}_2(\mathbf{z}_0(i-1)), \text{pr}_1(\mathbf{z}_0(i)), (\eta_0)^1(\overline{i, N})) + \\ &+ c_{\eta_0(i)}(\mathbf{z}_0(i), (\eta_0)^1(\overline{i, N}))] + f(\text{pr}_2(\mathbf{z}_0(N)), y), \end{aligned} \quad (4.9)$$

$$\begin{aligned} V[x_0] &= \mathfrak{C}_{\eta_0}[\mathbf{z}_0|x_0] = \mathfrak{c}(x_0, \text{pr}_1(\mathbf{z}_0(1)), (\eta_0)^1(\overline{1, N})) + c_{\eta_0(1)}(\mathbf{z}_0(1), (\eta_0)^1(\overline{1, N})) + \\ &+ \sum_{i=2}^N [\mathfrak{c}(\text{pr}_2(\mathbf{z}_0(i-1)), \text{pr}_1(\mathbf{z}_0(i)), (\eta_0)^1(\overline{i, N})) + c_{\eta_0(i)}(\mathbf{z}_0(i), (\eta_0)^1(\overline{i, N}))] + \\ &+ f(\text{pr}_2(\mathbf{z}_0(N)), x_0). \end{aligned} \quad (4.10)$$

Полезно иметь в виду, что в силу биективности  $\eta_0$  справедливо равенство  $(\eta_0)^1(\overline{1, N}) = \overline{1, N}$ . Тогда из (4.9), (4.10) следует, что

$$\begin{aligned} \mathfrak{C}_{\eta_0}[\mathbf{z}_0|y|y] &= V[x_0] + (\mathfrak{c}(y, \text{pr}_1(\mathbf{z}_0(1)), \overline{1, N}) - \mathfrak{c}(x_0, \text{pr}_1(\mathbf{z}_0(1)), \overline{1, N})) + \\ &+ (f(\text{pr}_2(\mathbf{z}_0(N)), y) - f(\text{pr}_2(\mathbf{z}_0(N)), x_0)). \end{aligned} \quad (4.11)$$

Из (4.11) видно, что по известному значению  $V[x_0]$  величина  $\mathfrak{C}_{\eta_0}[\mathbf{z}_0|y|y]$  рассчитывается очень просто. Соотношение значений (4.9), (4.10) может быть произвольным (при произвольном выборе  $y$ ). В этой связи предлагается определить все значения  $\mathfrak{C}_{\eta_0}[\mathbf{z}_0|x|x]$ ,  $x \in X^0$ , и выбрать среди них наименьшее. Точнее (см. (4.11)), следует выбрать  $x_1 \in X^0$ , для которого

$$\begin{aligned} \mathfrak{c}(x_1, \text{pr}_1(\mathbf{z}_0(1)), \overline{1, N}) + f(\text{pr}_1(\mathbf{z}_0(1)), x_1) &= \\ = \min_{x \in X^0} [\mathfrak{c}(x, \text{pr}_1(\mathbf{z}_0(1)), \overline{1, N}) + f(\text{pr}_1(\mathbf{z}_0(1)), x)]. \end{aligned}$$

Далее, решаем задачу (2.9) при  $x = x_1$ , используя процедуру раздела 3 (в разделе 3 следует заменить  $x^0$  на  $x_1$ ), получая экстремум  $V[x_1]$ , для которого (см. предложение 1)

$$V[x_1] \leq \mathfrak{C}_{\eta_0}[\mathbf{z}_0|x_1|x_1].$$

Если окажется, что  $V[x_1] < V[x_0]$ , то заменяем  $x_0$  на  $x_1$  и повторяем процедуру настоящего раздела с использованием предложения 1 (в противном случае — останов алгоритма, а  $V[x_0]$  и  $(\eta_0, \mathbf{z}_0, x_0)$  выдаются в качестве приближения к решению задачи (2.11)). В целях полноты изложения совсем кратко обсудим очередной этап процедуры, предполагая, что  $V[x_1] < V[x_0]$  (данное условие можно, в принципе, проверить в режиме экономии памяти; см. замечание 1). Тогда (см. этап 4) определяем  $(\eta_1, \mathbf{z}_1) \in (\text{SOL})[x_1]$ , получая, в частности,  $(\eta_1, \mathbf{z}_1, x_1) \in \mathbf{D}$ . Рассматриваем теперь траектории  $\mathbf{z}_1|x$ ,  $x \in X^0$ . После этого выбираем  $x_2 \in X^0$  из условия

$$\begin{aligned} \mathfrak{c}(x_2, \text{pr}_1(\mathbf{z}_1(1)), \overline{1, N}) + f(\text{pr}_2(\mathbf{z}_1(N)), x_2) &= \\ = \min_{x \in X^0} [\mathfrak{c}(x, \text{pr}_1(\mathbf{z}_1(1)), \overline{1, N}) + f(\text{pr}_2(\mathbf{z}_1(N)), x)]; \end{aligned}$$

для случая  $x = x_2$  следует решить задачу (2.9) при  $x = x_2$ , используя процедуру раздела 3 и получая экстремум  $V[x_2]$ , для которого

$$V[x_2] \leq \mathfrak{C}_{\eta_1}[\mathbf{z}_1|x_2|x_2].$$

Если  $V[x_2] < V[x_1]$ , то заменяем  $x_1$  на  $x_2$  и применяем процедуру ДП, включая нахождение элемента  $(\text{SOL})[x_2]$  (в противном случае осуществляем останов алгоритма). Итак, мы последовательно применяем ДП в версии раздела 3 точно: при  $x = x_0, x = x_1, x = x_2$  и т. д.

## § 5. Частный случай: оптимизация точки старта-финиша

В настоящем разделе рассматривается вариант задачи (2.11), восходящий в идейном отношении к оптимизации параметра в замкнутой задаче коммивояжера. Рассмотрим сначала содержательную постановку, полагая в дополнение к определениям раздела 2, что траектория процесса должна вернуться в точку старта. Это перемещение также оценивается посредством соответствующей аддитивной компоненты в составе критерия. Поэтому в данном случае речь идет фактически об оптимизации точки старта-финиша.

Можно, однако, все это формализовать и в виде варианта определений раздела 2, где не требовалось возвращение траектории в точку старта. В самом деле, учет последнего перемещения можно осуществить, конкретизируя должным образом терминальное слагаемое в (2.8). Для определенности сразу будем рассматривать случай задачи на быстродействие, считая, что значения всех функций в (2.7) имеют смысл соответствующих времен, требующихся для перемещений и выполнения внутренних работ; зависимость от списка заданий в этих условиях реализуется посредством дополнительных штрафов, что позволяет учитывать некоторые ограничения динамического характера, не вводя их посредством предваряющих запретов (нарушение упомянутых ограничений делается крайне невыгодным в смысле значений критерия; подробнее см. в [17]). Итак, в дальнейшем ориентируемся на прикладную задачу управления инструментом при листовой резке деталей на машинах с ЧПУ (см. [15, 17–19] и др).

Всюду в дальнейшем  $X$  — прямоугольник на плоскости:  $X = [0, a] \times [0, b]$ , где  $a \in \mathbb{R}_+ \setminus \{0\}$  и  $b \in \mathbb{R}_+ \setminus \{0\}$ , т. е.  $a > 0$ ,  $b > 0$ . Таким образом, в виде  $X$  задан раскройный лист, на котором размещены детали, подлежащие резке (см. обсуждение в [15, § 3.3]). Для наших целей существенны только контура этих деталей (для простоты будем считать, что резка производится прямо по контурам, хотя реально для этого выбираются некоторые эквидистанты). В интересах компьютерной реализации осуществляется должная дискретизация процедур резки: возле каждого контура намечаются (достаточно часто) возможные точки врезки и отвечающие им точки выключения инструмента. В результате возникают мегаполисы  $M_j$ , отвечающие каждому своему контуру (здесь  $N$  совпадает с числом контуров, а  $j \in \overline{1, N}$ ). Для каждого контура определяется также отношение  $\mathbb{M}_j$ , составленное из УП с элементами в виде точек врезки и «привязанных» к ним точек выключения инструмента.

Условия предшествования (см. [15, § 3.3]) возникают, в частности, в результате формализации требования: у каждой детали внутренние контура должны быть вырезаны раньше внешнего. Аналогичное требование относится и к резке «внутренних» деталей (имеются в виду детали, охватываемые каким-либо внешним контуром). Итак, условия предшествования порождены естественными требованиями технологического характера. Имеются ограничения динамического характера, среди которых сейчас отметим только тепловые допуски: возле каждой точки врезки должно быть «достаточно много» невырезанного металла для отвода тепла (имеется в виду случай термической резки). Данное требование учитывается посредством введения специальных штрафных функций: нарушение тепловых допусков формально не запрещается, но делается крайне невыгодным (подробнее см. в [17]).

Мы накладываем ниже следующее условие: после завершения резки всех контуров инструмент должен возвратиться в точку старта. Данное требование может быть связано с задачами, в которых производится резка больших серий деталей с одинаковым раскройным планом. В этом случае фактический процесс резки многократно повторяется и имеет смысл оптимизировать точки старта-финиша. В качестве естественного критерия принимаем совокупное время: рассматриваем задачу на быстродействие, т. е. задачу о скорейшем завершении резки контуров всех деталей, отвечающих раскройному плану. При этом скорости холостого и рабочего хода существенно различаются.

Отметим, что (подробнее см. в [15, § 3.3]) собственно время резки контуров является (при рассматриваемом сейчас режиме резки по замкнутому контуру) одним и тем же для всех вариантов выполнения заданий. Поэтому данное время мы не включаем в критерий; оно может быть легко учтено посредством добавления дополнительного слагаемого. Тогда, в нашем толковании, процесс выполнения внутренних работ состоит в следующем: после прибытия в точку врезки (данное перемещение осуществляется в режиме холостого хода) осуществляется сама врезка и затем со скоростью рабочего хода перемещение к точке начала реза, после чего (в нашей модели) со скоростью рабочего хода реализуется перемещение к точке выключения инструмента. Затем со скоростью холостого хода осуществляется перемещение к следующему контуру. Все упомянутые перемещения оцениваются соответствующими временами. В случае, когда в районе точки врезки имеется «мало» невырезанного металла, ко времени рабочего хода добавляется большая штрафная константа (см. [17]). Значение терминальной функции  $f$  (последняя функция в (2.7)) определяется временем перемещения инструмента из точки множества  $M$  в точку  $x \in X^0$  со скоростью холостого хода; таким образом учитывается эффект, связанный с возвращением инструмента в точку старта.

## § 6. Вычислительный эксперимент

Вычисления производились на ПЭВМ с процессором Intel i7-2630QM с 8Гб оперативной памяти, работающей под управлением Windows 7 (64-bit). Для разработки программы был использован язык C++, компилятор MinGW и интерфейсная библиотека Qt.

В ходе вычислительного эксперимента были рассмотрены два примера. Первый пример демонстрирует работу алгоритма с выборкой значительной размерности. Второй показывает сравнение с точным алгоритмом, выполненным для каждой допустимой точки старта-финиша. В обоих примерах использовались функции стоимости внутренних работ, учитывающие тепловые ограничения. Именно, вокруг участка завершения реза должно быть достаточно металла для эффективного рассеивания тепла. Длина области вдоль участка завершения реза составляет 100 мм, ширина — 25 мм. Если менее четверти площади этого участка перекрывается с невырезанным металлом, то к стоимости внутренних работ добавляется штраф 1000000. Иначе функция стоимости вычисляется обычным способом (см. [17]).

**Пример 1.** Количество мегаполисов 29. Начальные точки выбираются на прямоугольнике с координатами (0 мм, 0 мм), (0 мм, 2000 мм), (965 мм, 2000 мм), (965 мм, 0 мм). Первая точка имеет координаты (0 мм, 0 мм). Остальные точки откладываются на прямоугольнике с шагом 100 мм в направлении по часовой стрелке. Координаты контуров и точек врезки, начала и завершения реза не приводятся из соображений экономии объема.

Всего были выполнены три итерации. Далее приводится значение критерия и координата точки старта-финиша для каждой итерации. Итерация 1: 66,136, (0 мм, 0 мм). Итерация 2: 65,929, (0 мм, 100 мм). Итерация 3: 65,517, (330 мм, 0 мм). На рис. 1 показан результат для третьей итерации. Время счета 1 ч. 1 мин. 30 сек. Итоговый результат 65,517.

**Пример 2.** Количество мегаполисов 15. Начальные точки выбираются на прямоугольнике с координатами (0 мм, 0 мм), (0 мм, 2000 мм), (1375 мм, 2000 мм), (1375 мм, 0 мм). Первая точка имеет координаты (0 мм, 0 мм). Остальные точки откладываются на прямоугольнике с шагом 100 мм в направлении по часовой стрелке. Координаты контуров и точек врезки, начала и завершения реза так же, как и в первом примере, не приводятся из соображений экономии объема.

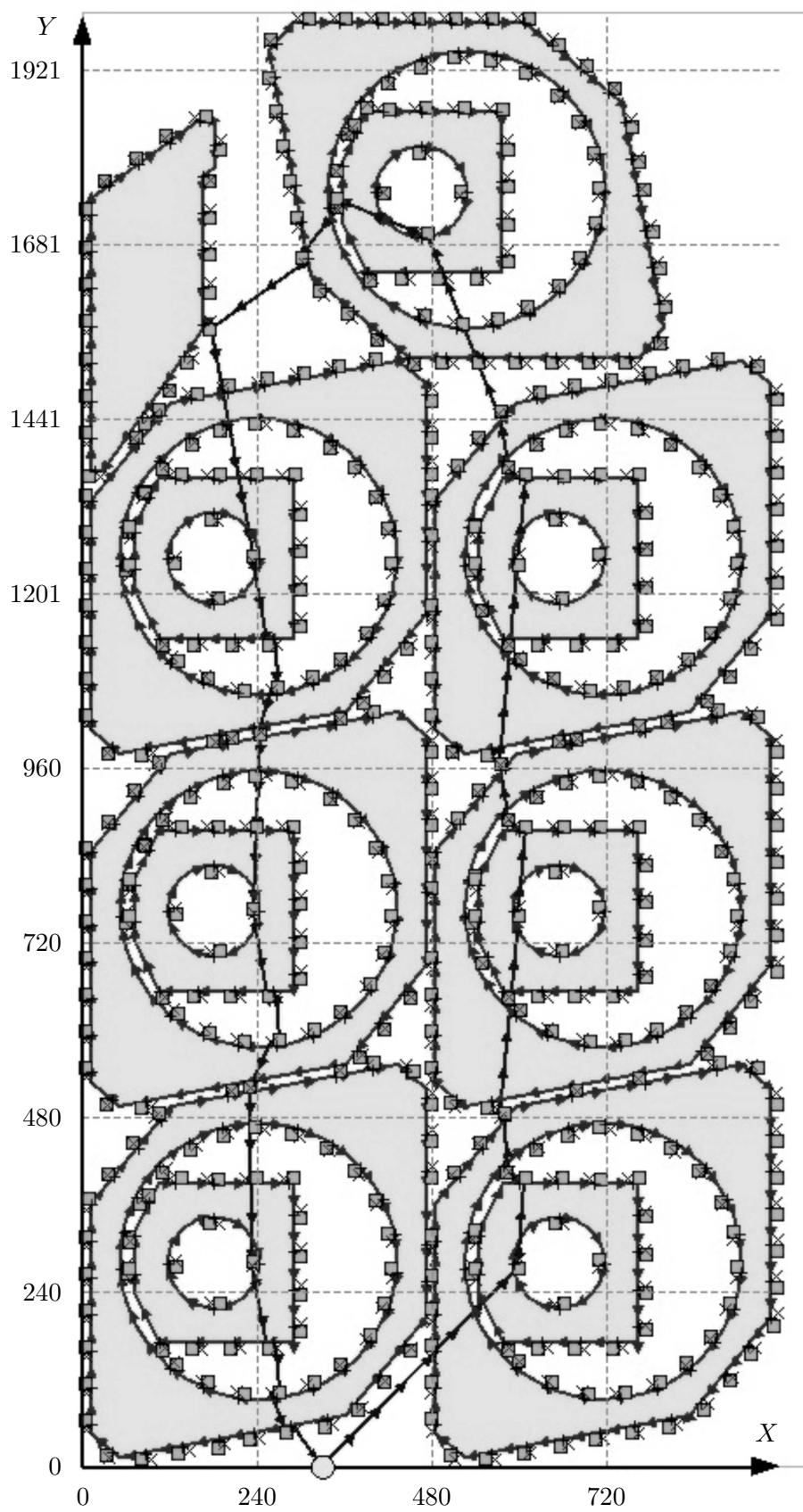


Рис. 1. Пример 1. Результат работы эвристического алгоритма

Как и в первом примере были выполнены три итерации. Далее приводятся результаты счета и соответствующие им точки старта–финиша. Итерация 1: 39,514, (0 мм, 0 мм). Итерация 2: 39,047, (350 мм, 0 мм). Итерация 3: 39,029, (450 мм, 0 мм). На рис. 2 показан результат для третьей итерации. Время счета 32 сек. Итоговый результат 39,029.

Далее был произведен расчет для всех возможных точек старта–финиша (67 точек). Получен результат 38,787. Данному результату соответствует точка старта–финиша с координатами (0 мм, 1500 мм). Время счета 11 мин. 56 сек. На рис. 3 показан результат работы точного алгоритма.

В ходе вычислительного эксперимента была подтверждена вычислительная эффективность рассматриваемого в статье эвристического алгоритма. В каждом из примеров были выполнены по три итерации, каждая из которых улучшала результат. При этом отклонение от оптимального решения оказалось незначительным, а экономия времени счета — существенной.

**Финансирование.** Исследования выполнены при финансовой поддержке РФФИ, грант 20–08–00873.

#### СПИСОК ЛИТЕРАТУРЫ

1. Chentsov A. G., Chentsov P. A. On routing problem with starting point optimization // *Ural Mathematical Journal*. 2020. Vol. 6. No. 2. P. 44–62. <https://doi.org/10.15826/umj.2020.2.005>
2. Ченцов А. Г., Ченцов П. А. К вопросу об оптимизации точки старта в задаче маршрутизации с ограничениями // *Известия Института математики и информатики Удмуртского государственного университета*. 2020. Т. 55. С. 135–154. <https://doi.org/10.35634/2226-3594-2020-55-09>
3. Gutin G., Punnen A. P. *The traveling salesman problem and its variations*. Boston: Springer, 2007. <https://doi.org/10.1007/b101971>
4. Cook W. J. *In pursuit of the traveling salesman. Mathematics at the limits of computation*. Princeton: Princeton University Press, 2012. <https://zbmath.org/?q=an:1236.00007>
5. Гимади Э. Х., Хачай М. Ю. *Экстремальные задачи на множествах перестановок*. Екатеринбург: УМЦ УПИ, 2016.
6. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Вопросы теории // *Автоматика и телемеханика*. 1989. Вып. 9. С. 3–33. <http://mi.mathnet.ru/at6414>
7. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Точные методы // *Автоматика и телемеханика*. 1989. Вып. 10. С. 3–29. <http://mi.mathnet.ru/at6433>
8. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Приближенные алгоритмы // *Автоматика и телемеханика*. 1989. Вып. 11. С. 3–26. <http://mi.mathnet.ru/at6463>
9. Беллман Р. Применение динамического программирования к задаче о коммивояжере // *Кибернетический сборник*. М.: Мир, 1964. Т. 9. С. 219–228.
10. Хелд М., Карп Р. М. Применение динамического программирования к задачам упорядочения // *Кибернетический сборник*. М.: Мир, 1964. Т. 9. С. 202–218.
11. Куратовский К., Мостовский А. *Теория множеств*. М.: Мир, 1970.
12. Дьедонне Ж. *Основы современного анализа*. М.: Мир, 1964.
13. Ченцов А. Г. *Экстремальные задачи маршрутизации и распределения заданий: вопросы теории*. М.–Ижевск: НИЦ «Регулярная и хаотическая динамика», 2008.
14. Ченцов А. Г., Ченцов А. А., Сесекин А. Н. *Задачи маршрутизации перемещений с неаддитивным агрегированием затрат*. М.: Ленанд, 2021.
15. Петунин А. А., Ченцов А. Г., Ченцов П. А. *Оптимальная маршрутизация инструмента машин фигурной листовой резки с числовым программным управлением. Математические модели и алгоритмы*. Екатеринбург: УрФУ, 2020.



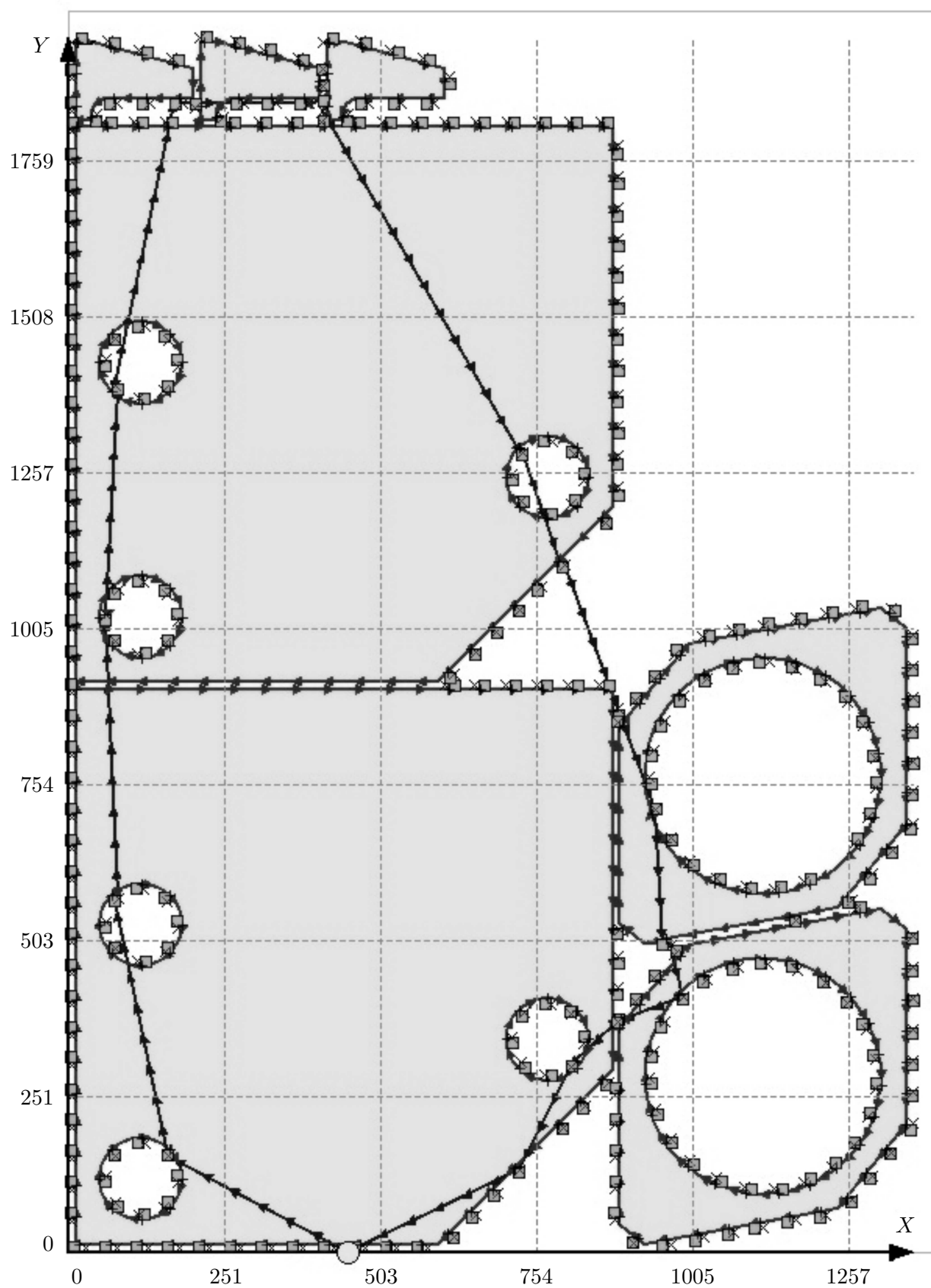


Рис. 2. Пример 2. Результат работы эвристического алгоритма





16. Ченцов А. Г. Одна параллельная процедура построения функции Беллмана в обобщенной задаче курьера с внутренними работами // Автоматика и телемеханика. 2012. Вып. 3. С. 134–149. <http://mi.mathnet.ru/at3783>
17. Ченцов А. Г., Ченцов П. А. Маршрутизация в условиях ограничений: задача о посещении мегаполисов // Автоматика и телемеханика. 2016. Вып. 11. С. 96–117. <http://mi.mathnet.ru/at14599>
18. Петунин А. А. О некоторых стратегиях формирования маршрута инструмента при разработке управляющих программ для машин термической резки материала // Вестник Уфимского государственного авиационного технического университета. 2009. Т. 13. № 2. С. 280–286. <https://elibrary.ru/item.asp?id=15134316>
19. Chentsov A. G., Chentsov P. A., Petunin A. A., Seseikin A. N. Model of megalopolises in the tool path optimization for CNC plate cutting machines // International Journal of Production Research. 2018. Vol. 56. Issue 14. P. 4819–4830. <https://doi.org/10.1080/00207543.2017.1421784>

Поступила в редакцию 11.05.2021

Ченцов Александр Георгиевич, д. ф.-м. н., член-корр. РАН, главный научный сотрудник, Институт математики и механики им. Н. Н. Красовского УрО РАН, 620108, Россия, г. Екатеринбург, ул. С. Ковалевской, 16;

профессор, Уральский федеральный университет, 620002, Россия, г. Екатеринбург, ул. Мира, 19.

ORCID: <https://orcid.org/0000-0001-6568-0703>

E-mail: [chentsov@imm.uran.ru](mailto:chentsov@imm.uran.ru)

Ченцов Павел Александрович, к. ф.-м. н., старший научный сотрудник, Институт математики и механики им. Н. Н. Красовского УрО РАН, 620108, Россия, г. Екатеринбург, ул. С. Ковалевской, 16; старший научный сотрудник, Уральский федеральный университет, 620002, Россия, г. Екатеринбург, ул. Мира, 19.

ORCID: <https://orcid.org/0000-0003-3595-0607>

E-mail: [chentsov.p@mail.ru](mailto:chentsov.p@mail.ru)

**Цитирование:** А. Г. Ченцов, П. А. Ченцов. Об одной задаче последовательного обхода множеств // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2021. Т. 31. Вып. 3. С. 487–504.

*A. G. Chentsov, P. A. Chentsov*

### On sequential traversal of sets

*Keywords:* route optimization, dynamic programming, start point optimization.

MSC2020: 49L20, 90C39

DOI: [10.35634/vm210310](https://doi.org/10.35634/vm210310)

The problem of sequential traversal of megapolises with precedence conditions is investigated; this problem is oriented to mechanical engineering — CNC metal cutting machines. There is the following setting singularity: the terminal component of additive criterion contains the dependence on the starting point. This singularity leads to the fact that the natural solution procedure based on dynamic programming must be applied individually for every starting point. The investigation goal consists in the construction of an optimizing algorithm for determining a complex including a route (a variant of megapolis numbering), a trajectory, and a starting point. The proposed algorithm realizes an idea of directed enumeration of starting points. This algorithm is realized as a program for PC; computations for model examples are made.

**Funding.** The study was funded by the Russian Foundation for Basic Research (project No. 20–08–00873).

### REFERENCES

1. Chentsov A. G., Chentsov P. A. On routing problem with starting point optimization, *Ural Mathematical Journal*, 2020, vol. 6, no. 2, pp. 44–62. <https://doi.org/10.15826/umj.2020.2.005>
2. Chentsov A. G., Chentsov P. A. To the question of optimization of the starting point in the routing problem with restrictions, *Izvestiya Instituta Matematiki i Informatiki Udmurtskogo Gosudarstvennogo Universiteta*, 2020, vol. 55, pp. 135–154 (in Russian). <https://doi.org/10.35634/2226-3594-2020-55-09>
3. Gutin G., Punnen A. P. *The traveling salesman problem and its variations*, Boston: Springer, 2007. <https://doi.org/10.1007/b101971>
4. Cook W. J. *In pursuit of the traveling salesman. Mathematics at the limits of computation*, Princeton: Princeton University Press, 2012. <https://zbmath.org/?q=an:1236.00007>
5. Gimadi E. Kh., Khachay M. Yu. *Ekstremal'nye zadachi na mnozhestvakh perestanovok* (Extreme problems on sets of permutations), Yekaterinburg: UMC UPI, 2016.
6. Melamed I. I., Sergeev S. I., Sigal I. Kh. The traveling salesman problem. I. Theoretical issues, *Automation and Remote Control*, 1989, vol. 50, issue 9, pp. 1147–1173. <https://zbmath.org/?q=an:0705.90070>
7. Melamed I. I., Sergeev S. I., Sigal I. Kh. The traveling salesman problem. II. Exact methods, *Automation and Remote Control*, 1989, vol. 50, issue 10, pp. 1303–1324. <https://zbmath.org/?q=an:0705.90071>
8. Melamed I. I., Sergeev S. I., Sigal I. Kh. The traveling salesman problem. Approximate algorithms, *Automation and Remote Control*, 1989, vol. 50, issue 11, pp. 1459–1479. <https://zbmath.org/?q=an:0704.90095>
9. Bellman R. Dynamic programming treatment of the travelling salesman problem, *Journal of the ACM*, 1962, vol. 9, issue 1, pp. 61–63. <https://doi.org/10.1145/321105.321111>
10. Held M., Karp R. M. A dynamic programming approach to sequencing problems, *Journal of the Society for Industrial and Applied Mathematics*, 1962, vol. 10, issue 1, pp. 196–210. <https://doi.org/10.1137/0110015>
11. Kuratowski K., Mostowski A. *Set theory*, Amsterdam: North-Holland, 1967. Translated under the title *Teoriya mnozhestv*, Moscow: Mir, 1970.
12. Dieudonné J. *Foundations of modern analysis*, New York: Academic Press, 1960. Translated under the title *Osnovy sovremennogo analiza*, Moscow: Mir, 1964.

13. Chentsov A.G. *Ekstremal'nye zadachi marshrutizatsii i raspredeleniya zadaniy: voprosy teorii* (Extreme problems of routing and assignment of tasks: questions of theory), Moscow–Izhevsk: Regular and Chaotic Dynamics, Institute of Computer Science, 2008.
14. Chentsov A.G., Chentsov A.A., Seseikin A.N. *Zadachi marshrutizatsii peremeshchenii s neadditivnym agregirovaniem zatrat* (Routing problems with non-additive cost aggregation), Moscow: LENAND, 2021.
15. Petunin A.A., Chentsov A.G., Chentsov P.A. *Optimal'naya marshrutizatsiya instrumenta mashin figurnoi listovoi rezki s chislovyim programmnyim upravleniem. Matematicheskie modeli i algoritmy*, (Optimal tool routing on CNC sheet cutting machines. Mathematical models and algorithms), Yekaterinburg: Ural Federal University, 2020.
16. Chentsov A.G. On a parallel procedure for constructing the Bellman function in the generalized problem of courier with internal jobs, *Automation and Remote Control*, 2012, vol. 73, issue 3, pp. 532–546. <https://doi.org/10.1134/S0005117912030113>
17. Chentsov A.G., Chentsov P.A. Routing under constraints: problem of visit to megalopolises, *Automation and Remote Control*, 2016, vol. 77, issue 11, pp. 1957–1974. <https://doi.org/10.1134/S0005117916110060>
18. Petunin A.A. About some strategies of the programming of tool route by developing of control programs for thermal cutting machines, *Vestnik Ufimskogo Gosudarstvennogo Aviatsionnogo Tekhnicheskogo Universiteta*, 2009, vol. 13, no. 2, pp. 280–286 (in Russian). <https://elibrary.ru/item.asp?id=15134316>
19. Chentsov A.G., Chentsov P.A., Petunin A.A., Seseikin A.N. Model of megalopolises in the tool path optimization for CNC plate cutting machines, *International Journal of Production Research*, 2018, vol. 56, issue 14, pp. 4819–4830. <https://doi.org/10.1080/00207543.2017.1421784>

Received 11.05.2021

Aleksandr Georgievich Chentsov, Doctor of Physics and Mathematics, Corresponding Member, Russian Academy of Science, Chief Researcher, N.N. Krasovskii Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences, ul. S. Kovalevskoi, 16, Yekaterinburg, 620108, Russia; Professor, Ural Federal University, ul. Mira, 19, Yekaterinburg, 620002, Russia.

ORCID: <https://orcid.org/0000-0001-6568-0703>

E-mail: [chentsov@imm.uran.ru](mailto:chentsov@imm.uran.ru)

Pavel Aleksandrovich Chentsov, Candidate of Physics and Mathematics, Senior Researcher, N.N. Krasovskii Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences, ul. S. Kovalevskoi, 16, Yekaterinburg, 620108, Russia.

Senior Researcher, Ural Federal University, ul. Mira, 19, Yekaterinburg, 620002, Russia;

ORCID: <https://orcid.org/0000-0003-3595-0607>

E-mail: [chentsov.p@mail.ru](mailto:chentsov.p@mail.ru)

**Citation:** A.G. Chentsov, P.A. Chentsov. On sequential traversal of sets, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2021, vol. 31, issue 3, pp. 487–504.