УДК 519.168

МЕТОД ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ В МИНИМАКСНОЙ ЗАДАЧЕ РАСПРЕДЕЛЕНИЯ ЗАДАНИЙ С РАВНОЦЕННЫМИ ИСПОЛНИТЕЛЯМИ

Е.Е. Иванко

В работе рассмотрен ряд специфических вариантов метода динамического программирования, используемых для решения минимаксной задачи распределения заданий при условии, что исполнители равноценны, и их порядок не важен. Для разработанных рекурсивных схем метода динамического программирования показана корректность, проводится сравнение вычислительной трудоемкости классической и предложенных схем. Демонстрируется, что использование специфики условия равноценности исполнителей позволяет сократить количество операций в рассмотренном методе динамического программирования по сравнению с классическим более чем в 4 раза, при этом при увеличении размерности исходной задачи относительный выигрыш лишь увеличивается. Одна из использованных техник сокращения вычислений – «встречное» динамическое программирование – по-видимому является общей для целого класса задач, допускающих использование при решении принципа Беллмана. Применение данной техники связано с неполным расчетом значений функции Беллмана в задаче, обладающей некоторой внутренней симметрией, после чего решение исходной задачи получается склеиванием полученных массивов значений функции Беллмана.

Ключевые слова: метод динамического программирования, распределение заданий.

Введение

Одним из классических способов исследования задачи дискретной оптимизации является метод динамического программирования (МДП) [1]. На сегодняшний день этот метод относительно редко используется собственно для нахождения оптимального решения. Для решения большинства задач либо разработаны более эффективные методы математического программирования, либо достаточными оказываются многочисленные приближенные и эмпирические алгоритмы. Способом точного решения МДП остается лишь в ряде прикладных задач, обладающих сложной структурой. В этом случае создание соответствующего МДП представляет самостоятельную сложность и является существенным шагом вперед по сравнению, например, с более простыми переборными или эмпирическими алгоритмами. Особый интерес представляют случаи, когда сложные ограничения на задачу, доставляющие трудности в формализации задачи на языке, например, линейного программированя, оказываются выгодны в процедуре МДП. Так, например, в обобщенной задаче курьера [2] удается использовать условия группировки «городов»в «мегаполисы»и условия предшествования для существенного уменьшения количества вычислений, обеспечивая высокую прикладную эффективность МДП в труднорешаемой задаче.

Несмотря на указанные ограничения, МДП остается как удобным инструментом оценки вычислительной сложности комбинаторной задачи [3, 4], так и способом ее дальнейшего качественного исследования. Так, например, в ряде работ автора [5, 6] близкие к МДП схемы

используются для построения условий устойчивости в задачах комбинаторной оптимизации при изменении множества начальных данных. Также интерес представляют различные смешанные схемы решения сложных комбинаторных задач, в которых динамическое программирование используется ограниченным образом в комбинации с быстрыми приближенными алгоритмами [7].

Настоящая работа посвящена построению специфических схем МДП, используемых для решения минимаксной задачи распределения заданий среди конечного числа равноценных исполнителей. Насколько известно автору, впервые МДП для минимаксной распределительной задачи был изложен в работе [8] и, далее, в [2, 9]. Представленные в работе алгоритмы учитывают «симметричность» распределительной задачи с равноценными исполнителями, благодаря чему удается сократить объем вычислений по сравнению с классической схемой, а также реализовать идею «встречного» динамического программирования (впервые, по-видимому, кратко описанную в работе [10] при учете ограничений в задаче линейного целочисленного программирования). Полученные результаты представляют в первую очередь теоретический интерес, углубляя представление о структуре МДП в задаче распределения заданий.

Минимаксная постановка (или bottleneck в англоязычной литературе) относительно редко рассматривается в работах, посвященных задаче распределения заданий. Между тем, существует большое число важных приложений, связанных именно с этой задачей. В качестве одного из примеров можно привести задачу мультикоммивояжер (mTSP [11]). В минимаксной постановке этой задачи требуется разбить множество «городов»между коммивояжерами таким образом, чтобы длина пути самого загруженного коммивояжера была минимальна. Задачи подобного плана возникают, например, при оптимизации перемещений бригады исполнителей в агрессивной внешней среде [7, 11]. Другой областью приложения минимаксной задачи распределения заданий является формирование наиболее вероятных эволюционных деревьев с приложениями в области биологии и лингвистики.

1. Обозначения и классическая схема

Пусть X – конечное множество распределяемых заданий, $N \in \mathbb{N}$ – количество исполнителей. Далее будем считать, что $|X| \geq N$. Для всякого подмножества $K \subseteq X$ задана функция $d(K)$ трудоемкости выполнения заданий этого подмножества одним исполнителем (в настоящей постановке мы считаем всех исполнителей равносильными, что, например, является весьма распространенным условием в прикладных постановках задачи мультикоммивояжера)

$$d\colon \mathcal{P}(X) \to \mathbb{R}^+, \tag{1}$$

где $d(\varnothing) = 0$, а под $\mathcal{P}(X)$ мы традиционно понимаем множество всех подмножеств X.

Для любого $K \subseteq X$, любого $k \in \overline{1, N}$ через $M_k(K)$ будем обозначать совокупность всех разбиений множества K, содержащих не более k непустых подмножеств:

$$M_k(K) \triangleq \{\mathcal{K} \in \mathcal{P}(\mathcal{P}(K)) \mid (|\mathcal{K}| \leq k)\&(\bigcup_{S \in \mathcal{K}} S = K)\&(\forall A \in \mathcal{K}\ \forall B \in \mathcal{K}\ (A \cap B = \varnothing) \vee (A = B))\}.$$

В данной постановке мы допускаем ситуации, в которых не все исполнители снабжаются заданиями. На практике такие ситуации возникают в случае тесной связи между заданиями, когда, например, выполнение одного из двух заданий обеспечивает средства для выполнения второго. При этом может оказаться «более выгодным»отдать исполнение обоих заданий одному исполнителю, оставив второго без заданий.

Всякий элемент $\alpha \in M_k(K)$ будем называть *распределением* множества заданий K по k исполнителям; перечисляя все непустые подмножества распределения α, будем писать $\alpha = \{K_1, \ldots, K_i\}$.

При заданных $k \in \overline{1,N}, K \subseteq X$ *стоимостью* всякого распределения $\alpha \in M_k(K)$ будем называть функцию $D\colon M_k(K) \to \mathbb{R}^+$, определяемую как максимум трудоемкости по составляющим распределение подмножествам:

$$\forall \alpha = \{K_1, \ldots, K_i\} \in M_k(K) \quad D(\alpha) \triangleq \max_{j \in \overline{1,i}} d(K_j). \tag{2}$$

Распределение $\alpha_0 \in M_k(K)$ будем называть *оптимальным* на $M_k(K)$, если справедливо

$$D(\alpha_0) = \min_{\alpha \in M_k(K)} D(\alpha). \tag{3}$$

Минимаксная задача распределения заданий заключается в нахождении стоимости оптимального распределения на $M_N(X)$ при заданных X, N, d. Отметим, что излагаемый классический метод динамического программирования и рассматриваемые ниже его модификации, позволяют помимо расчета стоимости оптимального распределения восстановить также и само оптимальное распределение, однако этой части метода динамического программирования в настоящей работе мы не касаемся. В схеме 1 изложен классический вариант МДП для нахождения стоимости оптимального распределения заданий из множества X среди N исполнителей.

Замечание. Здесь мы не станем останавливаться на подробном рассмотрении данного классического соотношения. Отметим лишь без доказательства, что всякая величина $v_i(K)$ совпадает со стоимостью оптимального распределения заданий из множества K среди i исполнителей и, в частности, величина $v_N(X)$ совпадает с искомой стоимостью оптимального распределения заданий множества X среди N исполнителей. Строгое доказательство этих фактов, а также корректности рекурсивной схемы 1 можно найти в работе [2]. В последующих же разделах сосредоточимся на сокращении количества итераций в схеме 1 при нахождении стоимости оптимального распределения на $M_N(X)$ в условиях равноценности исполнителей.

Схема 1

$$\begin{array}{rl}
1) & \forall K \subseteq X \quad v_1(K) = d(K); \\
2) & \forall K \subseteq X \quad v_2(K) = \min\limits_{K_1 \sqcup K_2 = K} \left(\max\{d(K_1), v_1(K_2)\}\right); \\
3) & \forall K \subseteq X \quad v_3(K) = \min\limits_{K_1 \sqcup K_2 = K} \left(\max\{d(K_1), v_2(K_2)\}\right); \\
\ldots & \ldots \quad \ldots \quad \ldots \\
N-1) & \forall K \subseteq X \quad v_{N-1}(K) = \min\limits_{K_1 \sqcup K_2 = K} \left(\max\{d(K_1), v_{N-2}(K_2)\}\right); \\
N) & v_N(X) = \min\limits_{K_1 \sqcup K_2 = X} \left(\max\{d(K_1), v_{N-1}(K_2)\}\right).
\end{array}$$

Выражение $K_1 \sqcup K_2 = K$ означает разбиение множества K на два подмножества K_1 и K_2: $K_1 \cap K_2 = \varnothing, K_1 \cup K_2 = K$, в котором порядок подмножеств K_1, K_2 полагается существенным.

2. Сокращенная схема МДП

Сокращение количества итераций в схеме 1 будем проводить в три последовательных этапа. Первые два сокращения, описываемые в разделах 2.1 и 2.2, очевидно следуют из отказа от учета порядка и различий на множестве исполнителей. Эти ссокращения если и не излагались ранее теоретически, то, вероятно, были учтены при написании соответствующих алгоритмов и программ. Третье сокращение, рассматриваемое в разделе 2.3, развивает идею «встречного» динамического программирования.

2.1. Отбрасывание множеств «малой» мощности

Начнем усечение схемы 1 с отказа от расчета таких значений $v_i(K)$, для которых справедливо $|K| < i$. В этом случае вместо трудоемкого перебора подмножеств $\mathcal{P}(K)$ при вычислении соответствующего минимума, достаточно приравнять $v_i(K)$ значению $v_{i-1}(K)$, расчитанному на предыдущем шаге.

Схема 2

$$1)\qquad \forall K \subseteq X \quad v_1^*(K) = d(K);$$

$$2)\qquad \forall K \subseteq X \quad v_2^*(K) = \begin{cases} \min\limits_{K_1 \sqcup K_2 = K} \left(\max\{d(K_1), v_1^*(K_2)\}\right), & \text{если } |K| \geq 2; \\ v_1^*(K), & \text{если } |K| < 2; \end{cases}$$

$$3)\qquad \forall K \subseteq X \quad v_3^*(K) = \begin{cases} \min\limits_{K_1 \sqcup K_2 = K} \left(\max\{d(K_1), v_2^*(K_2)\}\right), & \text{если } |K| \geq 3; \\ v_2^*(K), & \text{если } |K| < 3; \end{cases}$$

$$\ldots \quad \ldots \quad \ldots \quad \ldots$$

$$N-1)\qquad \forall K \subseteq X \quad v_{N-1}^*(K) = \begin{cases} \min\limits_{K_1 \sqcup K_2 = K} \left(\max\{d(K_1), v_{N-2}^*(K_2)\}\right), & |K| \geq N-1; \\ v_{N-2}^*(K), & |K| < N-1; \end{cases}$$

$$N)\qquad v_N^*(X) = \min_{K_1 \sqcup K_2 = X} \left(\max\{d(K_1), v_{N-1}^*(K_2)\}\right), \quad \text{учитывая, что } |X| \geq N.$$

Действительно, раз $i > |K|$, значит, по меньшей мере, один исполнитель остался без заданий. Пусть $\alpha \in M_i(K)$ — некоторое оптимальное распределение на $M_i(K)$, обладающее в силу сказанного в разделе 1 стоимостью $v_i(K)$. Отбрасывая из этого распределения работника без заданий, получим распределение $\beta \in M_{i-1}(K)$, содержащее тот же самый набор непустых подмножеств, что и α, а, значит, справедливо $D(\alpha) = D(\beta)$. Распределение β оптимально на $M_{i-1}(K)$ (если бы это было не так и существовало бы распределение $\beta' \in M_{i-1}(K)$: $D(\beta') < D(\beta)$, то, учитывая, что $D(\alpha) = D(\beta)$ и добавляя к β' исполнителя без заданий, можно было бы получить распределение $\alpha' \in M_i(K)$: $D(\alpha') < D(\alpha)$, что противоречит оптимальности α), следовательно, $D(\beta) = v_{i-1}(K)$, а, значит, и $v_i(K) = v_{i-1}(K)$.

2.2. Отбрасывание подмножеств «большой»мощности

Продолжим сокращение количества операций в итерационной схеме метода динамического программирования, взяв за основу теперь уже рекурсивную схему 2. Рассмотрим новую схему 3, в котором на каждом i-том шаге для всякого $K \subset X$ будем искать минимум, перебирая только такие разбиения $K = K_1 \sqcup K_2$, в которых $|K_1| \leq |K|/i$:

Схема 3

$$1)\qquad \forall K \subseteq X \quad v_1^{**}(K) = d(K);$$

$$2)\qquad \forall K \subseteq X \quad v_2^{**}(K) = \begin{cases} \min\limits_{\substack{K_1 \sqcup K_2 = K \\ |K_1| \leq |K|/2}} \left(\max\{d(K_1), v_1^{**}(K_2)\}\right), & \text{если } |K| \geq 2; \\ v_1^{**}(K), & \text{если } |K| < 2; \end{cases}$$

$$3)\qquad \forall K \subseteq X \quad v_3^{**}(K) = \begin{cases} \min\limits_{\substack{K_1 \sqcup K_2 = K \\ |K_1| \leq |K|/3}} \left(\max\{d(K_1), v_2^{**}(K_2)\}\right), & \text{если } |K| \geq 3; \\ v_2^{**}(K), & \text{если } |K| < 3; \end{cases}$$

$$\ldots \quad \ldots \quad \ldots \quad \ldots$$

$$N-1) \qquad \forall K \subseteq X \;\; v^{**}_{N-1}(K) = \begin{cases} \min\limits_{\substack{K_1 \sqcup K_2 = K \\ |K_1| \le |K|/(N-1)}} \left(\max\{d(K_1), v^{**}_{N-2}(K_2)\}\right), & |K| \ge N-1; \\ v^{**}_{N-2}(K), & |K| < N-1; \end{cases}$$

$$N) \qquad v^{**}_N(X) = \min_{\substack{K_1 \sqcup K_2 = X \\ |K_1| \le |X|/N}} \left(\max\{d(K_1), v^{**}_{N-1}(K_2)\}\right).$$

Действительно, если на i-м шаге мощность множества K_1 превосходит величину $|K|/i$, значит, в распределении заданий множества K_2 между $i-1$ исполнителем (стоимости которого соответствует функция $v^{**}_{i-1}(K_2)$) присутствует хотя бы одно подмножество мощности не более $|K|/i$. Меняя местами это подмножество и K_1, получим идентичное исходному распределение, которое встретится при переборе разбиений $K_1 \sqcup K_2 = K$, при этом будет выполнено $|K_1| \le |K|/i$. Запишем сказанное строго.

Докажем для начала простую техническую лемму, в которой будем писать $\sqcup_{i=1}^k K_i = K$, обозначая разбиение множества K на k подмножеств: $\bigcup_{i=1}^k K_i = K$ и $\forall i, j \in \overline{1,k}$ $(i \ne j) \Rightarrow (K_i \cap K_j = \varnothing)$. Порядок следования подмножеств в записи суммы полагаем существенным. В отличие от записи $\{K_1, \ldots, K_k\}$, где перечислялись только непустые подмножества, в записи $\sqcup_{i=1}^k K_i = K$ всякое подмножество K_i может быть пустым.

Лемма 1. *Пусть задано конечное множество $K \subseteq X$: $|K| > 1$, функция d из* (1) *и величина $k \in \mathbb{N}$: $|K| \ge k$, тогда*

$$\underbrace{\min_{\substack{\sqcup_{i=1}^k K_i = K \\ |K_1| \le |K|/k}} \left(\max_{i \in \overline{1,k}} \{d(K_i)\}\right)}_{(*)} = \underbrace{\min_{\substack{K_1 \subseteq K \\ |K_1| \le |K|/k}} \left(\max\left\{d(K_1), \min_{\sqcup_{i=2}^k K_i = K \setminus K_1} \left(\max_{i \in \overline{2,k}} \{d(K_i)\}\right)\right\}\right)}_{(**)}.$$

Доказательство. Пусть $\sqcup_{i=1}^k K_i^0 = K$ — разбиение K, доставляющее минимум в $(*)$, причем $|K_1^0| \le |K|/k$, тогда $(*) = \max_{i \in \overline{1,k}}\{d(K_i^0)\} = \max\{d(K_1^0), \max_{i \in \overline{2,k}}\{d(K_i^0)\}\} \ge (**)$. С другой стороны, если $\sqcup_{i=1}^k L_i^0 = K$ — разбиение K, доставляющее минимум в $(**)$, где вновь $|L_1^0| \le |K|/k$, тогда $(**) = \max\{d(L_1^0), \max_{i \in \overline{2,k}}\{d(L_i^0)\}\} = \max_{i \in \overline{1,k}}\{d(L_i^0)\} \ge (*)$. Из приведенных рассуждений следует, что $(*) = (**)$. □

Предложение 1. $(\forall k \in \overline{1, N-1},\ \forall K \subseteq X\ v^*_k(K) = v^{**}_k(K))\ \ \&\ \ (v^*_N(X) = v^{**}_N(X))$.

Доказательство. Пользуясь индукцией на номер шага k, покажем первую часть утверждения.

Б.И. $k = 1$. $\forall K \subseteq X \;\; v^{**}_1(K) = d(K) = v^*_1(K)$.

Ш.И. Пусть для некоторого $k \in \overline{1, N-1}$ справедливо: $\forall i \in \overline{1, k-1}, \forall K \subseteq X\ v^*_i(K) = v^{**}_i(K)$. Покажем, что $\forall K \subseteq X\ v^*_k(K) = v^{**}_k(K)$. Рассмотрим произвольное подмножество $K \subseteq X$. Если $|K| < k$, то по индукции и в силу схем 2 и 3 имеем $v^*_k(K) = v^*_{k-1}(K) \overset{\text{п.и.}}{=} v^{**}_{k-1}(K) = v^{**}_k(K)$. Пусть $|K| \ge k$. Величина $v^*_k(K)$ (будучи равна согласно рассуждениям раздела 2.1 величине $v_k(K)$) соответствует стоимости оптимального распределения заданий множества K среди k работников, а значит из определений (2), (3)

$$v^*_k(K) = \min_{\sqcup_{i=1}^k K_i = K} \left(\max_{i \in \overline{1,k}} \{d(K_i)\}\right). \tag{4}$$

Хотя бы одно подмножество из всякого разбиения $\sqcup_{i=1}^k K_i = K$ должно иметь мощность, не превосходящую $|K|/k$, поскольку иначе $\sqcup_{j=1}^k |K_j| > |K|$. Правая часть (4) не зависит от

порядка следования подмножеств в разбиении, а значит без ограничения общности подмножество, содержащее $|K|/k$ элементов будем обозначать как K_1, иными словами

$$\min_{\sqcup_{i=1}^k K_i = K} \left(\max_{i\in\overline{1,k}} \{d(K_i)\} \right) = \min_{\substack{\sqcup_{i=1}^k K_i = K \\ |K_1| \le |K|/k}} \left(\max_{i\in\overline{1,k}} \{d(K_i)\} \right).$$

Согласно лемме

$$\min_{\substack{\sqcup_{i=1}^k K_i = K \\ |K_1| \le |K|/k}} \left(\max_{i\in\overline{1,k}} \{d(K_i)\} \right) = \min_{\substack{K_1 \subseteq K \\ |K_1| \le |K|/k}} \left(\max \left\{ d(K_1), \min_{\sqcup_{i=2}^k K_i = K\setminus K_1} \left(\max_{i\in\overline{2,k}} \{d(K_i)\} \right) \right\} \right).$$

Величина $v_{k-1}^*(K \setminus K_1)$ совпадает со стоимостью оптимального распределения заданий множества $K \setminus K_1$ между $k-1$ исполнителем, а значит

$$\min_{\substack{K_1 \subseteq K \\ |K_1| \le |K|/k}} \left(\max \left\{ d(K_1), \min_{\sqcup_{i=2}^k K_i = K\setminus K_1} \left(\max_{i\in\overline{2,k}} \{d(K_i)\} \right) \right\} \right) = \min_{\substack{K_1 \subseteq K \\ |K_1| \le |K|/k}} \left(\max \left\{ d(K_1), v_{k-1}^*(K \setminus K_1) \right\} \right).$$

По индукции имеем

$$\min_{\substack{K_1 \subseteq K \\ |K_1| \le |K|/k}} \left(\max \left\{ d(K_1), v_{k-1}^*(K \setminus K_1) \right\} \right) = \min_{\substack{K_1 \subseteq K \\ |K_1| \le |K|/k}} \left(\max \left\{ d(K_1), v_{k-1}^{**}(K \setminus K_1) \right\} \right)$$

и, наконец, из схемы 3

$$\min_{\substack{K_1 \subseteq K \\ |K_1| \le |K|/k}} \left(\max \left\{ d(K_1), v_{k-1}^{**}(K \setminus K_1) \right\} \right) = v_k^{**}(K).$$

Вторая часть утверждения $v_N^*(X) = v_N^{**}(X)$ доказывается аналогично при $k = N, K = X$. □

2.3. Сокращение числа «слоев»

Покажем, наконец, что в схеме 3 вычисления достаточно проводить лишь «до середины»: для $i \le V \triangleq [(N-1)/2] + 1$. Запишем суженную таким образом схему 4.

Схема 4

1) $\forall K \subseteq X \quad v_1^{**}(K) = d(K)$

2) $$\forall K \subseteq X \quad v_2^{**}(K) = \begin{cases} \min\limits_{\substack{K_1 \sqcup K_2 = K \\ |K_1| \le |K|/2}} \left(\max\{d(K_1), v_1^{**}(K_2)\} \right), & \text{если } |K| \ge 2; \\ v_1^{**}(K), & \text{если } |K| < 2; \end{cases}$$

3) $$\forall K \subseteq X \quad v_3^{**}(K) = \begin{cases} \min\limits_{\substack{K_1 \sqcup K_2 = K \\ |K_1| \le |K|/3}} \left(\max\{d(K_1), v_2^{**}(K_2)\} \right), & \text{если } |K| \ge 3; \\ v_2^{**}(K), & \text{если } |K| < 3; \end{cases}$$

… … … …

V) $$\forall K \subseteq X \quad v_V^{**}(K) = \begin{cases} \min\limits_{\substack{K_1 \sqcup K_2 = K \\ |K_1| \le |K|/V}} \left(\max\{d(K_1), v_{V-1}^{**}(K_2)\} \right), & \text{если } |K| \ge V; \\ v_{V-1}^{**}(K), & \text{если } |K| < V. \end{cases}$$

Дополним схему 4 последним шагом

$$v_N^{***}(X) = \min_{K_1 \sqcup K_2 = X} \max\{v_V^{**}(K_1), v_{N-V}^{**}(K_2)\}. \tag{5}$$

В силу того, что $N - V \leq V$, оба выражения $v_V^{**}(K_1)$ и $v_{N-V}^{**}(K_2)$ конструктивно определены из итерационной схемы 4 для всех $K_1, K_2 \subseteq X$.

Предложение 2. *Величина $v_N^{***}(X)$ выражает стоимость оптимального распределения на $M_N(X)$.*

Доказательство. Рассмотрим произвольное распределение $\alpha \in M_N(X)$: $\sqcup_{i=1}^{N} L_i = X$. Разобъем распределение α на две части α_1: $\sqcup_{i=1}^{V} L_i = X_1$ и α_2: $\sqcup_{i=V+1}^{N} L_i = X_2$, где $X_1 \triangleq \bigcup_{i=1}^{V} L_i$ и $X_2 \triangleq \bigcup_{i=V+1}^{N} L_i$. В таких обозначениях очевидно $X_1 \sqcup X_2 = X, \alpha_1 \in M_V(X_1), \alpha_2 \in M_{N-V}(X_2)$, и, как следствие, учитывая определение (2), имеем $D(\alpha) = \max\{D(\alpha_1), D(\alpha_2)\}$.

Учитывая, что $\alpha_1 \in M_V(X_1)$, $\alpha_2 \in M_{N-V}(X_2)$, а согласно предложению 1 величины $v_V^{**}(X_1)$ и $v_{N-V}^{**}(X_2)$ совпадают со стоимостями оптимальных распределений на $M_V(X_1)$ и $M_{N-V}(X_2)$ соответственно, получаем неравенства $D(\alpha_1) \geq v_V^{**}(X_1)$ и $D(\alpha_2) \geq v_{N-V}^{**}(X_2)$, а значит имеем

$$D(\alpha) = \max\{D(\alpha_1), D(\alpha_2)\} \geq \max\{v_V^{**}(X_1), v_{N-V}^{**}(X_2)\} \geq \\ \min_{K_1 \sqcup K_2 = X} \max\{v_V^{**}(K_1), v_{N-V}^{**}(K_2)\} = v_N^{***}(X),$$

справедливое для всякого $\alpha \in M_N(X)$.

Итак, стоимость всякого $\alpha \in M_N(X)$ не менее $v_N^{***}(X)$. Покажем, что найдется $\alpha_0 \in M_N(X)$: $D(\alpha_0) = v_N^{***}(X)$. Пусть на множествах K_1^0 и $K_2^0 = X \setminus K_1^0$ достигается минимум правой части выражения (5). Пусть далее α_1^0 — оптимальное распределение на $M_V(K_1^0)$: $\sqcup_{i=1}^{V} L_i^1 = K_1^0$, а α_2^0 — оптимальное распределение на $M_{N-V}(K_2^0)$: $\sqcup_{i=1}^{N-V} L_i^2 = K_2^0$. Учитывая, что $K_1^0 \sqcup K_2^0 = X$, составим распределение $\alpha_0 \in M_N(X)$: $(\sqcup_{i=1}^{V} L_i^1) \bigsqcup (\sqcup_{i=1}^{N-V} L_i^2) = K_1^0 \sqcup K_2^0 = X$. По построению

$$D(\alpha_0) = \max\{D(\alpha_1^0), D(\alpha_2^0)\} = \max\{v_V^{**}(K_1^0), v_{N-V}^{**}(K_2^0)\} = \\ \min_{K_1 \sqcup K_2 = X} \max\{v_V^{**}(K_1), v_{N-V}^{**}(K_2)\} = v_N^{***}(X).$$

□

3. Сравнение числа операций

Проведем численное сравнение количества «единичных операций», требующихся для расчета стоимости оптимального распределения заданий из множества X среди N исполнителей с использованием схем 1 и 4. Единичной операцией будем считать взятие максимума из двух чисел. Именно такие операции составляют основу вычислений в схемах 1 и 4 на всех шагах, следующих за первым. Число иных операций в обеих схемах будем считать пропорциональным числу единичных операций.

Пусть далее $|X| = n$. На каждом из шагов $2, \ldots, N-1$ схемы 1 перебираются всевозможные подмножества $K \subseteq X$. Для всякого $i \in \overline{0, n}$ существует C_n^i подмножеств K мощности i. Для всякого подмножества K: $|K| = i$ существует 2^i разбиений вида $K_1 \sqcup K_2 = K$, каждому из которых соответствует одна единичная операция. Таким образом, суммарное число единичных операций на шагах $2, \ldots, N-1$ схемы 1 можно записать в виде

$$(N-2) \sum_{i=0}^{n} 2^i C_n^i.$$

Таблица

Сравнение числа операций в различных вариантах МДП

n	Классический	Встречный	Отношение
10	178171	47316	3,765
12	1598419	412110	3,878
14	14365291	3588409	4,003
16	129205699	31307602	4,126
18	1162523611	273844200	4,245
20	10461401779	2401589556	4,356
22	94147373131	21113768572	4,459
24	847305386659	186039820510	4,554
26	7625664593851	1642544688531	4,642
28	68630645800339	14527969382181	4,724
30	617674470025771	128701822410531	4,799
100	1,546e+048	2,721e+047	5,681
250	5,720e+119	9,828e+118	5,820

На последнем шаге производится перебор 2^n подмножеств множества X, каждому из которых вновь соответствует одна единичная операция. Итоговое количество единичных операций в схеме 1 можно выразить формулой

$$2^n + (N-2)\sum_{i=0}^{n} 2^i C_n^i. \tag{6}$$

В схеме 2 на всяком шаге $j \in \overline{2,\ldots,N-1}$ исключается расчет значений функции $v_j(K)$ в случае, когда количество распределяемых заданий $|K|$ превосходит число исполнителей j

$$2^n + \sum_{j=2}^{N-1}\left\{\sum_{i=j}^{n} 2^i C_n^i\right\}.$$

Далее, в схеме 3, исключая на j-том шаге из рассмотрения все разбиения $K_1 \sqcup K_2 = K$, где $|K_1| > |K|/j$, имеем

$$\sum_{k=0}^{[n/N]} C_n^k + \sum_{j=2}^{N-1}\left\{\sum_{i=j}^{n}\left(C_n^i \cdot \sum_{k=0}^{[i/j]} C_i^k\right)\right\}.$$

Наконец, сужая количество насчитываемых слоев, в схеме 4 имеем:

$$2^n + \sum_{j=2}^{[(N-1)/2]+1}\left\{\sum_{i=j}^{n}\left(C_n^i \cdot \sum_{k=0}^{[i/j]} C_i^k\right)\right\}. \tag{7}$$

В таблице n как и прежде обозначает количество распределяемых заданий; количество исполнителей N принимается равным 5; в колонке «классический» представлено значение функции (6); в колонке «встречный» – значение функции (7); в колонке «отношение» приводится отношение величины (6) к величине (7) для соответствующего значения n.

Работа проводилась при финансовой поддержке РФФИ (гранты 10-08-00484-а, 10-01-96020-р-урал-а) и программы Президиума УрО РАН (проект 09-П-1-1014).

Литература

1. Беллман, Р. Динамическое программирование / Р. Беллман. – М.: Изд-во иностр. лит., 1960. – 400 с.
2. Ченцов, А.Г. Экстремальные задачи маршрутизации и распределения заданий: вопросы теории / А.Г. Ченцов. – М.: РХД, 2007. – 240 с.
3. Held, M. A Dynamic Programming Approach to Sequencing Problems / M. Held, R.M. Karp // J. of the Society for Industrial and Applied Mathematics. – 1962. – V. 10, № 1. – P. 196–210.
4. Karp, R.M. Dynamic programming meets the principle of inclusion and exclusion / R.M. Karp // Oper. Res. Lett. – 1982. – № 1(2). – P. 49–51.
5. Иванко, Е.Е. Критерий устойчивости оптимального маршрута в задаче коммивояжера при добавлении вершины / Е.Е. Иванко // Вестн. Удмурт. ун-та. Математика. Механика. Компьютерные науки. – 2011. – № 1. – С. 58–66.
6. Иванко, Е.Е. Достаточные условия устойчивости в задаче коммивояжера / Е.Е. Иванко // Тр. Ин-та математики и механики УрО РАН. – 2011. – № 3. – С. 155–168.
7. Иванко, Е.Е. Об одном подходе к решению задачи маршрутизации перемещений с несколькими участниками / Е.Е. Иванко, А.Г. Ченцов, П.А. Ченцов // Изв. РАН. Теория и системы управления. – 2010. – № 4. – С. 63–71.
8. Коротаева, Л.Н. Об одной задаче о назначениях / Л.Н. Коротаева, Э.М. Назаров, А.Г. Ченцов // Журн. вычисл. математики и мат. физики. – 1993. – Т. 33, № 4. – С. 483–494.
9. Ченцов, А.Г. К вопросу о построении процедуры разбиения конечного множества на основе метода динамического программирования / А.Г. Ченцов, П.А. Ченцов // Автоматика и телемеханика. – 2000. – № 4. – С. 129–142.
10. Алексеев, О.Г. Комплексное применение методов дискретной оптимизации / О.Г. Алексеев. – М.: Наука, 1986. – 247 с.
11. Gutin, G. The Traveling Salesman Problem and Its Variations / G. Gutin, A. Punnen. – Berlin: Springer, 2002. – 850 p.

Евгений Евгеньевич Иванко, кандидат физико-математических наук, старший научный сотрудник отдела управляемых систем Института математики и механики Уральского отделения Российской академии наук (г. Екатеринбург, Российская Федерация), ivanko@ural.ru.

Bulletin of the South Ural State University.
Series «Mathematical Modelling, Programming & Computer Software»,
2013, vol. 6, no. 1, pp. 124–133.

MSC 90C39

Dynamic Programming Method in Bottleneck Tasks Distribution Problem with Equal Agents

E.E. Ivanko, Institute of Mathematics and Mechanics, Ural Branch, Russian Academy of Sciencies, Ekaterinburg, Russian Federation, ivanko@ural.ru

The paper considers a number of specific variants of dynamic programming method used to solve the bottleneck problem of tasks distribution in case when the performers are the same and their order is not important. Developed schemes for recursive dynamic programming method is shown to be correct, the comparison of computational complexity of the classical and the proposed schemes is done. We demonstrate that the usage of the specific conditions of performers equivalence can reduce the number of operations in the above dynamic programming method compared to the classical method more than 4 times. Herewith increase of the dimension of the original problem leads only to the increase in the relative gain. One of the techniques used to reduce computing – «counter» dynamic programming – apparently is common to a whole class of problems that allow use of the Bellman principle. The application of this technique bases on incomplete calculation of the Bellman function in problem that has some internal symmetry, then the original problem is obtained by gluing the resulting array of values of Bellman.

Keywords: dynamic programming method, tasks distribution.

References

1. Bellman R. *Dinamicheskoe programmirovanie* [Dynamic Programming]. Moscow, 1960. 400 p.
2. Chentsov A.G. *Ekstremalnie zadachi marshrutizacii i raspredeleniya zadanii: voprosi teorii* [Extreme Problems of Routing and Tasks Distribution: Theory]. Moscow, 2007. 240 p.
3. Held M., Karp R.M. A Dynamic Programming Approach to Sequencing Problems. *J. of the Society for Industrial and Applied Mathematics,* 1962, vol. 10, no. 1, pp. 196–210.
4. Karp R.M. Dynamic Programming Meets the Principle of Inclusion and Exclusion. *Oper. Res. Lett.,* 1982, no. 1(2), pp. 49–51.
5. Ivanko E.E. Stability Criterion for the Optimal Route in Traveling Salesman Problem in Case of the One City Addition [Kriterii ustoichivosti optimalnogo marshruta v zadache kommivoyazhera pri dobavlenii vershini]. *Vestnik Udmurtskogo universiteta. Matematika. Mekhanica. Kompyuternie nauki*, 2011, no. 1, pp. 58–66.
6. Ivanko E.E. Sufficient conditions of stability in Traveling Salesman Problem [Dostatochnie usloviya ustoichivosti v zadache kommivoyazhera]. *Trudi Instituta matematiki i mekhaniki UrO RAN*, 2011, no. 3, pp. 155–168.
7. Chentsov P.A., Chentsov A.G., Ivanko E.E. On One Approach to Solving Traveling Salesman Problem with Several Performers [Ob odnim podhode k resheniyuzadachi marshrutizacii peremeschenii s neskolkimi uchastnikami]. *J. of Computer and Systems Sciences International*, 2010, vol. 49, no. 4, pp. 570–578.
8. Korotaeva L.N., Nazarov E.M., Chentsov A.G. On one variant of Assignment Problem [Ob odnoi zadache o naznacheniyah]. *J. vichislitelnoi matematiki i matematicheskoi fiziki* [Computational Mathematics and Mathematical Physics], 1993, vol. 33, no. 4, pp. 483–494.
9. Chentsov P.A., Chentsov A.G. To the Problem of Finite Set Partition Construction Based on Dynamic Programming [K voprosu o postroenii proceduri razbieniya konechnogo mnozhestva na osnove metoda dinamicheskogo programmirovaniya]. *Avtomatika i telemehanika* [Automation and Remote Control], 2000, no. 4. pp. 129–142.
10. Alekseev O.G. *Kompleksnoe primenenie metodov discretnoi optimizacii* [Complex Application of Discrete Optimization Methods]. Moscow, 1986, 247 p.
11. Gutin G. *The Traveling Salesman Problem and Its Variations.* Berlin: Springer, 2002, 850 p.

Поступила в редакцию июня 5 июля 2012 г.