Министерство высшего и среднего специального образования УССР

Полтавский инженерно-строительный институт

УДК 519.852.67

О.А.Емец

О РЕШЕНИИ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С БОЛЬШИМ КОЛИЧЕСТВОМ ОГРАНИЧЕНИЙ

Полтава-1983

При разработке и эксплуатации различных АСУ часто приходится решать задачи линейного программирования (ЛП), значительная часть которых имеет большую размерность. Для решения таких задач разработан ряд декомпозиционных методов, базирующихся, в основном, на учете специальной (блочной, окаймленной и т.п.) структуры матрицы ограничений и восходящих к методу разложения Данцига-Вулфа (см.,например, [I]). В настоящей работе рассматривается один метод решения задач ЛП с матрицей общего вида, в которых количество ограничений значительно превосходит количество неизвестных. С помощью этого метода исходная задача ЛП заменяется набором задач ЛП со значительно меньшим количеством ограничений. Будем называть далее этот метод методом последовательного подсоединения ограничений (МППО). Выбор такого названия объясняется сущностью метода изложенного ниже.

Итак, пусть необходимо найти

$$\min_{(x_1, x_2,\dots, x_n)\in D_\tau} \sum_{i=1}^{n} c_i x_i , \tag{I}$$

где $\tau=0$ и область D_0 определяется системой S , состоящей из m ограничений:

$$\sum_{i=1}^{n} a_{ij} x_i \le b_j , \quad j=1,2,\dots,m. \tag{2}$$

Предполагается, что $m \gg n$, а функция цели на области D_0 ограничена снизу.

Рассмотрим МППО решения задачи (I)-(2). Сруктурно он может быть представлен следующим образом. На первом этапе формируется система S_1 линейных ограничений, определяющая некоторую область D_1 :

$$\sum_{i=1}^{n} a'_{it} x_i \le b'_t , \quad t=1,2,\dots,\tau. \tag{3}$$

При этом система S_1 должна удовлетворять следующим трем условиям: I) $S_1 \subset S$, 2) $\tau \ll m$, 3) функция цели на области D_1 ограничена снизу. Заметим, что имеющийся произвол в выборе системы S_1 позволяет формировать ее таким образом, чтобы решение задачи (I),(3) было как можно ближе к решению задачи (I),(2).

На втором этапе МППО находится решение задачи (I) на области D_τ (при текущем значении индекса τ ($\tau \geq$ I)) и определяется точка $x^\tau \in D_\tau$, доставляющая это решение.

Третий этап состоит в проверке выполнения условия

$$x^\tau \in D_0 \tag{4}$$

Теорема I. Если $x^\tau \in D_0$, то $\sum_{i=1}^{n} c_i x_i^\tau$ - это решение задачи (I),(2).

Доказательство. По построению $D_\tau \supset D_0$. Следовательно,

$$\min_{x \in D_0} \sum_{i=1}^{n} c_i x_i = C(x') \geq \min_{x \in D_\tau} \sum_{i=1}^{n} c_i x_i = C(x''),$$

но т.к. $x'' = x^\tau \in D_0$, то $C(x') = C(x'') = C(x^\tau)$, что и требовалось показать.

Таким образом, если условие (4) справедливо, то это означает, что x^τ доставляет решение задаче (I),(2), т.е. процесс решения заканчивается. В противном случае необходим четвертый этап, который состоит в формировании системы ограничений $S_{\tau+1}$. При этом к системе присоединяется часть или все неравенства системы $S \setminus S_\tau$, которые в точке x^τ не выполняются, т.е. область D_τ заменяется на $D_{\tau+1}$, $D_\tau \supset D_{\tau+1}$. Полагаем τ равным τ+I. После этого переходим ко второму этапу. Имеет место следующая теорема.

Теорема 2. МППО заканчивает работу после конечного числа повторений своих этапов.

Доказательство. Для доказательства достаточно заметить, что S - конечное множество. Следовательно, множество непересекающихся подмножеств S так же конечно. Поэтому на третьем этапе МППО лишь конечное число раз будет возвращение на этап 2, что и доказывает теорему.

Заметим, что МППО идейно близок методу отсекающих гиперплоскостей в выпуклом программировании [2], для которого имеется положительный опыт решения ряда важных практических задач (задачи массового раскроя, производственного планирования и др.) [3]. Отметим также идейное родство МППО и методов последовательного погружения надграфика целевой функции в задачах оптимизации [4].

Рассмотрим применение МППО к решению следующей задачи.

Найти

$$\tilde{x} = \arg\min_{x \in D_0} \sum_{i=1}^{p+1} \mu_i x_i , \qquad (5)$$

где $0 < \mu_1 < \ldots < \mu_{p+1}$, а область D_0 определяется следующей системой ограничений

$$\sum_{t=0}^{K_{im_i}-1} x_{\gamma_{im_i}+t} + d_{im_i} \leq x_{p+1}, \; i=1,2,\ldots,K; \qquad (6)$$

$$\sum_{t=0}^{K_{ij}-1} x_{\gamma_{ij}+t} \leq C_{i(j+1)} - d_{ij}; \; i=1,2,\ldots,K; \; j=1,2,\ldots,m_i-1; \qquad (7)$$

$$\sum_{t=1}^{i} x_{\alpha_t} \geq \sum_{\ell=1}^{i} b_\ell , \quad \forall \alpha_t \in \{1,2,\ldots,p\}, \qquad (8)$$

$$\alpha_t \neq \alpha_q , \; \forall t \neq q, \quad t,q = 1,2,\ldots,i; \; i=1,2,\ldots,p-1;$$

$$\sum_{i=1}^{p} x_i = \sum_{i=1}^{p} b_i ; \qquad (9)$$

$$x_q \geq x_{q+1} ; \qquad (10)$$

где $q \in \{1, 2, \ldots, p\} \setminus \{\sum_{\alpha=1}^{i} \sum_{\beta=1}^{i} K_{\alpha\beta};\ j=1,2,\ldots,m_i;$ $i=1,2,\ldots,K\}$.

Здесь γ_{ij} вычисляются по формулам

$$\gamma_{ij} = \sum_{t=1}^{i-1} \sum_{q=1}^{m_t} K_{tq} + \sum_{q=1}^{j} K_{iq} + 1;\ j=1,2,\ldots,m_i;\ i=2,3,\ldots,K; \quad (\mathrm{I}1)$$

$$\gamma_{1j} = \sum_{q=1}^{j} K_{1q} + 1;\quad j=1,2,\ldots,m_1. \quad (\mathrm{I}2)$$

Параметры модели (5)-(I2) будут описаны ниже. Здесь же заметим только, что

$$0 \leqslant b_i \leqslant b_{i+1};\quad i=1,2,\ldots,p-1; \quad (\mathrm{I}3)$$

$$\sum_{i=1}^{K} \sum_{j=1}^{m_i} K_{ij} = p;\quad p \geqslant n; \quad (\mathrm{I}4)$$

$$a_{n-i+1} = b_{p-i+1},\ i=1,2,\ldots,n;\quad \sum_{i=1}^{n} a_i = \sum_{j=1}^{p} b_j; \quad (\mathrm{I}5)$$

Поскольку в рамках данной работы процесс получения модели (5)-(I2) не принципиален, приведем только пример задачи, при решении которой (на одном из этапов) используется модель (5)-(I2). Эта задача является обобщением известной задачи о распределении заданий по процессорам [5] или задачи Форда-Фалкерсона об организации параллельных работ при нулевом времени перехода от работы к работе [6].

Задача. Имеется множество K равнопроизводительных процессоров $P_1, P_2, \ldots, P_K$ и n независимых заданий $J_1, J_2, \ldots, J_n$, которые нужно выполнить. Процессоры могут работать одновременно, и любое задание можно выполнять на любом процессоре. Если задание загружено в процессор, оно остается там до конца обработки, которая должна быть непрерывной. Время обработки задания J_i известно, оно равно a_i, $i = 1,2, \ldots, n$. Предполагается, что для процессора P_i определено

m_i ($m_i \geqslant 0$) упорядоченных по началу временных промежутков I_{ij}, когда этот процессор не может выполнять задания, $j=1,2,\ldots,m_i$, $i=1,2,\ldots,K$. Заданы моменты начала и конца промежутка I_{ij}: c_{ij} и d_{ij} соответственно ($c_{i1}=0$, $c_{ij} \leqslant d_{ij}$), $j=1,2,\ldots,m_i$, $i=1,2,\ldots,K$. Задано также такое число λ, что $d_{im_i}<\lambda$, $i=1,2,\ldots,K$. Требуется установить, имеется ли такое распределение заданий по процессорам, при котором общее время выполнения заданий не превышает λ. При этом возможно предварительное задание параметров K_{ij}, под которыми понимаются максимальные числа заданий, которые могут быть выполнены на процессоре P_i после запрещенного для выполнения задания промежутка I_{ij} до следующего, если он еще имеется, $j=1,2,\ldots,m_i$; $i=1,2,\ldots,K$.

Один из возможных путей решения сформулированной задачи обсуждался в [7]. Он состоит из следующих шагов: I) построение комбинаторной модели задачи как оптимизационной задачи на множестве из вещественных чисел с повторениями, 2) отображение этого множества в арифметическое эвклидово пространство [8], 3) построение модели (5)-(I2), 4) решение задачи (5)-(I2) с помощью МППО, 5) формирование решения исходной задачи по решению задачи (5)-(I2).

Рассмотрим интерисующее нас применение МППО к решению задачи (5)-(I2).

Сформируем систему ограничений S_1 следующим образом. Включим в нее ограничения (6),(7),(9),(I0). Заметим, что поскольку система (8) содержит 2^p-2 ограничений, то существенным моментом является организация третьего этапа МППО.

Воспользуемся следующим фактом.

Утверждение I. Пусть $y=(y_1,y_2,\ldots,y_p)\in R^p$ и

$$y_j \le y_{j+1}, \quad j = 1, 2, \ldots, p-1. \qquad (I6)$$

Тогда из выполнения при $x \equiv y$ ограничения для $i = i_0$

$$\sum_{j=1}^{i} x_j \ge \sum_{j=1}^{i} b_j$$

следует выполнение и всех остальных ограничений системы (8) для тех же $i \equiv i_0$ и $x \equiv y$.

Доказательство. Справедливость утверждения непосредственно следует из предположения (I6) и равенства правых частей ограничений с одинаковым значением параметра i в системе (8).

Таким образом, из утверждения I следует, что для проведения проверки (4) достаточно проверить справедливость не более p -I ограничений.

Количество иттераций МППО существенно зависит от стратегии формирования системы $S_{\tau+1}$. Рассмотрим рациональные в этом смысле стратегии.

Будем называть множество точек $x = (x_1, x_2, \ldots, x_p) \in R^p$, удовлетворяющих системе ограничений (8),(9) (при условии (I3)), общим перестановочным многогранником (ОПМ), который определяется вещественными константами $b_1, b_2, \ldots, b_p$ .. Имеет место следующее утверждение.

Утверждение 2. Если $x = (x_1, x_2, \ldots, x_p)$ - вершина ОПМ, то верно

$$\{\alpha_1\} \subset \{\alpha_1, \alpha_2\} \subset \ldots \subset \{\alpha_1, \alpha_2, \ldots, \alpha_{p-1}\}, \qquad (I7)$$

$$\sum_{t=1}^{i} x_{\alpha_t} = \sum_{t=1}^{i} b_t, \quad \forall i = 1, 2, \ldots, p-1. \qquad (I8)$$

И наоборот, если верны условия (I7),(I8), то x - вершина ОПМ.

Доказательство. Доказательство утверждение 2 следует из теоремы 3.4 главы 5 [9] и условия (I3).

Из утверждения 2 следует, что рациональной стратегией формирования системы $S_{\tau+1}$ является такая, при которой поочередно для всех различных i из системы (8) выбираются и присоединяются к S_τ по одному невыполняющемуся в x^τ неравенству. При этом возможны две следующие ситуации:

I) $S_{\tau+1}$ содержит кроме S_τ по одному невыполняющемуся в x^τ неравенству для всех (возможных) i из (8),

2) $S_{\tau+1}$ содержит кроме S_τ одно невыполняющееся в x^τ неравенство из (8) для $i = i_{\tau+1}$, которые определяется следующим образом: для $\tau = I$ $i_{\tau+1} = I$, для $\tau > I$ $i_{\tau+1} = i_\tau + I \pmod{(p-1)}$. Если для $i_{\tau+1}$, вычисленного как описано выше, все ограничения из (8) выполняются в точке x^τ, то $i_{\tau+1}$ увеличиваем на на единицу (по $mod(p-1)$) до тех пор, пока найдется соответствующее ограничение, не выполняющееся в x^τ.

МПЮ для решения задачи (5)-(I2) реализован в виде программы на языке ФОРТРАН-IУ ДОС ЕС. На втором этапе была использована стандартная подпрограмма решения задач ЛП DSIMLI. Ниже приводятся результаты счета задачи (5)-(I2) по МПЮ с количеством ограничений более двух миллиардов (неравенств, входящих в систему (8), было $2^{31}-2$).

Пример. Задано $n=31$, $K=4$, $m_1=m_2=I$, $m_3=m_4=2$, $K_{11}=6$, $K_{21}=9$, $K_{31}=5$, $K_{32}=2$, $K_{41}=5$, $K_{42}=4$, $c_{11}=0$, $d_{11}=40$, $c_{21}=d_{21}=0$, $c_{31}=d_{31}=60$, $c_{41}=d_{41}=50$, $a=(a_1,a_2,\dots,a_n)=$ $=(I,I,I,2,2,2,2,2,3,5,5,6,7,7,8,9,9,I0,I0,I0,I0,II,II,II,I3,$ $I7,2I,22,24,44,49)$, $\mu_i = i$, $i=I,2,\dots,p$, $\mu_{p+1}=99$. При формировании $S_{\tau+1}$ использовалась вторая из возможных и описанных выше ситуаций.

Решение $\tilde{x}$ задачи (5)-(I2) получено при $\tau = II7$ и имеет вид: $\tilde{x}$ =(21,7475; II; IO; 7; 2; 2; 48,99; I7,7575; IO; IO; 3; 2; I; I; 0.99; 34; II; 8; 5; 2; 23,7475; IO; 25; II; 7; 5; I; I8,7475; IO; 9; 6; 93,7475).

В заключении отметим ряд на наш взгляд достоинств МППО: I) МППО идейно прост и поэтому легок в понимании и реализации, 2) МППО позволяет расширить возможности имеющегося математического обеспечения ЭВМ для решения задач ЛП с большим количеством ограничений путем включения имеющихся эффективных программ и алгоритмов в реализацию второго этапа МППО, 3) МППО позволяет учитывать специфику решаемой задачи, например так, как это было сделано при решении задач (5)-(I2), с тем, чтобы уменьшить количество иттераций, 4) МППО показал себя работоспособным при решении задач вида (5)-(I2) с количеством ограничений до немногим более двух миллиардов (с большими системами эксперименты не проводились).

Статья является расширенным вариантом доклада, подготовленного к семинару [IO].

Литература

1, Зуховицкий С.И., Авдеева Л.И. Линейное и выпуклое программирование.-М.:Наука,1967. - 460 с.

2. Ермольев Ю.М., Ляшко И.И., Михалевич В.С., Тюптя В.И. Математические методы исследования операций.-Киев: Вища школа,1979.-312с.

3. Романовский И.В., Станевичюс А.-И. А. Програмное обеспечение сиплекс-метода для задач больших размеров.-В кн.: Тезисы докл. Ш Всесоюз. конф. по исслед. операций.Горький, 1978, с. 12-14.

4. Булатов В.П. Методы погружения в задачах оптимизации. - Новосибирск:Наука,1977.-158с.

5. Гудман С., Хидетниеми С. Введение в разработку и анализ алгоритмов.-М.:Мир,1981.-368с.

6. Юдин Д.Б., Горяшко А.П., Немировский А.С. Математические методы оптимизации устройств и алгоритмов АСУ.-М.: Радио и связь, 1982.-288с.

7. Стоян Ю.Г., Емец О.А. Об оптимизации на перестановках с использованием больших задач линейного программирования: модели, способы и алгоритмы.-В кн.: Системы прогр. обеспечения решения задач оптим. планир. Седьмой всесоюз. симпоз.(г. Нарва-Йыэссуу, 16-24 апр. 1982г.).Крат. тез. докл.-М.,1982,с.100.

8. Стоян Ю.Г. Некоторые свойства специальных комбинаторных множеств.-Харьков,1980.-22с.(Препринт/Ин-т пробл. машиностроения АН УССР.:85).

9. Емеличев В.А., Ковалев М.М., Кравцов М.К. Многогранники, графы, оптимизация.-М.:Наука,1981.-344с.

10. Емец О.А. О расширении возможностей МО ОАСУ при ре-

шении задач ЛП с большим количеством ограничений посредством применения МППО. – В кн.: Всесоюзный науч.-практ. семинар "Прикл. аспекты упр. слож. системами" (г. Кемерово, 22-24 марта 1983 г.): Тез. докл., II ч. – М., 1983, с. 228.

Печатается в соответствии с решением совета Полтавского инженерно-строительного института от 28 мая I983 г.

14.06.83

В печать I7.06.83

Тир. Цена Зак.

252I7I, Киев-I7I, ул. Горького, I80
УкрНИИНТИ Госплана УССР