УДК 519.85

В.В. БОЙКО, В.Н. КУЗЬМЕНКО

УСКОРЕНИЕ СХОДИМОСТИ МЕТОДА ДЕКОМПОЗИЦИИ «PROGRESSIVE HEDGING»

Показано как метод решения многоэтапных задач стохастического программирования «Progressive Hedging» связан с методом декомпозиции по переменным первого уровня с помощью множителей Лагранжа для двухэтапных задач. Обсуждаются вопросы регулирования скорости сходимости метода и восстановления значений переменных первого уровня.

Введение. В работе [1] предложен метод (называемый «Progressive hedging» (PH) алгоритм) для решения многоэтапных задач стохастического программирования. Алгоритм основан на рассмотрении случайных событий на всех этапах (уровнях) в едином пространстве и на декомпозиции исходной задачи на каждом уровне по связывающим переменным верхних уровней.

PH алгоритм применяется не только для решения задач с непрерывными переменными, но и для решения стохастических задачах с целочисленными переменными [2].

Несмотря на то, что сходимость алгоритма в выпуклом случае доказана, регулирование процесса сходимости осуществляется одним параметром. Этот параметр одновременно регулирует изменение и прямых, и двойственных переменных, должен обеспечить согласованность этих изменений и влияет на скорость сходимости в целом.

При наличии дискретных переменных алгоритм применяется как эвристика, поскольку вопрос сходимости здесь в общем не решен.

Но и в непрерывном, и в дискретном случае вопросу выбора и изменения регулирующего параметра уделяется достаточное внимание.

Покажем как этот метод связан с другими методами декомпозиции, например, рассматриваемыми в работах Н.З. Шора [3], Н.Г. Журбенко [4], и как идеи, использованные в этих работах, можно использовать для ускорения сходимости PH алгоритма.

Рассмотрим двухэтапную задачу стохастического программирования, которая может представлена в такой форме:

$$\Psi = \min_{x \in X} F(x) = \min_{x \in X} E[Q(x, \xi)], \tag{1}$$

где $X \subseteq R^n$; $E[\cdot]$ – знак математического ожидания; $Q(x,\xi)$ целевая функция оптимального решения задачи второго этапа, условия которой зависят вектор x и случайной величины ξ, а именно:

$$Q(x,\xi) = \min_{y \in Y(x,\xi)} f(x, y, \xi). \tag{2}$$

Предполагается, что функция $f(x, y, \xi)$ выпукла по переменным $x \in R^n$ и $y \in R^m$ одновременно при любом фиксированном значении ξ.

Далее будем рассматривать случайную величину с конечным дискретным распределением $\xi \in (\xi_1, ..., \xi_r)$ и ненулевыми вероятностями $p_1, ..., p_r$. Реализации такой случайной величины будем называть сценариями. В этом случае задачи (1) и (2) можно переписать так

$$\min_{x \in X} F(x) = \min_{x \in X} \sum\nolimits_{s=1}^{r} p_s Q_s(x), \text{ где } Q_s(x) = \min_{y \in Y_s(x)} f_s(x, y). \tag{3}$$

Для декомпозиции задачи введем дубли переменных первого этапа x_s для каждого сценария, которые свяжем условием $x_s = x$. Получаем задачу

$$\min_{x \in X} F(x) = \min_{x_1, ..., x_r \in X,\ x} \left\{ \sum\nolimits_{s=1}^{r} p_s Q_s(x_s) \mid x_s = x,\ \forall s \right\}, \tag{4}$$

в которой переменные x можно не ограничивать множеством X. Поскольку эти переменные формально не используются в подзадачах, то можно избавиться от них, положив, например, $x = x_r$. В результате вместо блочной задачи (3) со связывающими переменными x получаем блочную задачу, с $r-1$ системами связывающих ограничений $x_s = x_r$, а именно:

$$\min_{x \in X} F(x) = \min_{x_1, ..., x_r \in X} \left\{ \sum\nolimits_{s=1}^{r} p_s Q_s(x_s) \mid x_s = x_r,\ s = 1, ..., r-1 \right\}. \tag{5}$$

Для дальнейшей декомпозиции применим Лагранжеву релаксацию по связывающим ограничениям. Для этого введем вектора-строки η_s множителей Лагранжа для каждого блока и запишем такую минимаксную задачу:

$$\Psi = \max_{\mu_1, ..., \mu_{r-1}} \left\{ \min_{x_1, ..., x_r \in X} \left\{ \sum\nolimits_{s=1}^{r-1} \left(p_s Q_s(x_s) + \eta_s (x_s - x_r) \right) + p_r Q_r(x_r) \right\} \right\}. \tag{6}$$

Задача минимизации в (6) разбивается на блоки

$$\min_{x_s \in X} \{ p_s Q_s(x_s) + \eta_s x_s \}, \; s = 1,\dots,r-1 \text{ и } \min_{x_r \in X} \left\{ p_r Q_r(x_r) - x_r \sum_{s=1}^{r-1} \eta_s \right\}.$$

Чтобы сделать вид блоков одинаковым введем вектор множителей для блока r, положив $\eta_r = -\sum_{s=1}^{r-1} \eta_s$. Отсюда следует условие $\sum_{s=1}^{r} \eta_s = 0$. Множители Лагранжа дальше будем называть двойственными переменными.

Введем масштабирующие множители b_s и сделаем замену двойственных переменных $\eta_s = b_s \mu_s$.

Тогда задача представима в виде

$$\Psi = \Phi(\mu) = \max_{\mu_1,\dots,\mu_r} \sum_{s=1}^{r} b_s \varphi(\mu_s), \tag{7}$$

$$\sum_{s=1}^{r} b_s \mu_s = 0, \tag{8}$$

$$\varphi(\mu_s) = \min_{x_s \in X, y_s \in Y_s(x)} \left\{ \frac{p_s}{b_s} f_s(x_s, y_s) + \mu_s x_s \right\}, \; s = 1,\dots,r, \tag{9}$$

где $\mu = (\mu_1,\dots,\mu_r) \in R^{nr}$ – вектор в объединенном пространстве двойственных переменных.

Теперь опишем PH метод, так как он основан на предложенной декомпозиции, но сначала укажем на достоинства и недостатки такой декомпозиции. Основное достоинство то, что если исходная задача (1) имеет решение, то все подзадачи минимизации (9) также имеют решения, независимо от значений двойственных переменных. Это важное свойство, так как при фиксированном векторе x в (2) множество $Y(x,\xi)$ может оказаться пустым, что требует специального алгоритма формирования результата решения подзадачи. Такой результат должен быть согласован с алгоритмом решения задачи первого уровня, а это не всегда может оказаться простым [5]. Основной недостаток этой декомпозиции – это значительное увеличение размерности задачи – количество переменных увеличивается в $2r$ раза. Однако, при этом процесс решения легко распараллеливается, что дает возможность не рассматривать пропорционального увеличения времени решения задачи.

Задача максимизации (7) – (9) может решаться различными методами. При этом важным является вопрос ограничено или нет множество X ? Конкретизация метода зависит также от свойств подзадач (9). Если эти подзадачи линейны или имеют неоднозначное оптимальное решение, то после получения оптимальных значений μ_s возникает проблема восстановления оптимальных значений переменных x_s и y_s. В работах [3, 4] предлагаются разные варианты восстановления этих переменных, так в [3] предлагается введение в подзадачи квадратичной регуляризирующей добавки.

Рассмотрим далее детали РН алгоритма, связанные с решением задачи (7) – (9). А это четыре основных момента: выбор направления движения на итерации; сохранение ограничений (8); выбор квадратичной добавки для (9); выбор шага.

Рассмотрим выбор направления и сохранения ограничений (8).

Пусть в текущей точке $\mu = (\mu_1, ..., \mu_r)$ выполнено условие (8). Пусть есть некоторое направление $d \in R^{nr}$ изменения вектора двойственных переменных, которое может нарушать условие (8). Чтобы этого избежать введем коррекцию Δd направления d, такую, чтобы ограничения (8) оставались выполненными. Вектора d, Δd, как и вектор μ, разбиваются на вектора меньшей размерности, соответствующие блокам. Тогда условие для коррекции может быть записано как $\sum_{s=1}^{r} b_s (d_s - \Delta d_s) = 0$ или $\sum_{s=1}^{r} b_s d_s = \sum_{s=1}^{r} b_s \Delta d_s$. Заметим, что такое условие не определяет коррекцию однозначно. Она должна быть такой, чтобы обеспечить возрастание целевой функции или хотя бы приближение к максимуму при движении в скорректированном направлении. Рассмотрим такой вариант коррекции $\Delta d_s = b_s c$, где c – вектор размерности n. Тогда $\sum_{s=1}^{r} b_s d_s = \sum_{s=1}^{r} b_s^2 c$ и $c = \sum_{s=1}^{r} b_s d_s \Big/ \sum_{s=1}^{r} b_s^2$. Такой вектор c делает коррекцию Δd вектором проектирования на подпространство, задаваемое (8), что непосредственно проверяется тем, что выполняется условие проектирования $(d - \Delta d)\Delta d^T = 0$ (здесь T знак транспонирования, так как вектора μ, d, Δd, c – это строки.

Таким образом, проекция произвольного направления d на подпространство (8) задается так: $d_s - b_s \sum_{\sigma=1}^{r} b_\sigma d_\sigma \Big/ \sum_{\sigma=1}^{r} b_\sigma^2$, $s = 1, ..., r$.

Если в качестве d выбран градиент $\nabla\Phi(\mu) = (b_1 x_1^*, ..., b_r x_r^*)^T$, где $x_1^*, ..., x_r^*$ вектора оптимальных значений x_s блоков, то проекция получается такой: $b_s x_s^{*T} - b_s \sum_{\sigma=1}^{r} b_\sigma^2 x_\sigma^{*T} \Big/ \sum_{\sigma=1}^{r} b_\sigma^2$. Если $b_s = 1$, $s = 1, ..., r$, имеем $x_s^{*T} - \sum_{\sigma=1}^{r} x_\sigma^{*T} \Big/ r$, т. е. от x_s^* отнимается среднее x^*. Для $b_s = p_s = 1/r$, $s = 1, ..., r$, получаем такое же направление $\left(x_s^* - E[x^*]\right)^T \Big/ r$, где $E[x^*]$ математическое ожидание случайной величины x^*, имеющей сценарии x_s^*.

Если в качестве d выбрать вектор $(x_1^*,..., x_r^*)^T$, а в качестве b_s вероятности, то проекция выглядит так: $d_s - \Delta d_s = x_s^{*T} - p_s\, E[x^*]^T \Big/ \sum_{\sigma=1}^{r} p_\sigma^2$. Если все вероятности равны, то снова получаем $d_s - \Delta d_s = x_s^{*T} - E[x^*]^T$.

PH алгоритм предлагает использовать направление изменения двойственных переменных $(x_s^* - E[x^*])^T$, а множители b_s полагают равными p_s. Это направление, не совпадает с проекцией градиента, если вероятности не равны, тем не менее сохраняет условие (8) и имеет положительное скалярное произведение с градиентом, что непосредственно проверяется

$$\sum_{s=1}^{r} p_s (x_s^* - E[x^*])^T = \sum_{s=1}^{r} p_s x_s^{*T} - E[x^*]^T \sum_{s=1}^{r} p_s = 0\,, \qquad (10)$$

$$(d - \Delta d)\Delta\Phi(\mu) = \sum_{s=1}^{r} (x_s^* - E[x^*])^T p_s x_s^* = E[x^{*2}] - E[x^*]^2 \geq 0\,. \qquad (11)$$

Равенство в (11) достигается только, если все x_s^* равны между собой.

Работа PH алгоритма начинается из точки $\mu = 0$, значения $b_s = p_s$, $s = 1,...,r$. Итерация k состоит в следующем. При заданных x_s^{*k} решить подзадачи (9) с регуляризирующей добавкой и найти x_s^{*k+1}

$$x_s^{*k+1} = \underset{x \in X,\, y \in Y_s(x)}{\arg\min} \left\{ f_s(x, y) + \mu_s^k x + \frac{\rho}{2} \left\| x - E[x^{*k}] \right\|^2 \right\},\ s = 1,...,r\,, \qquad (12)$$

изменить двойственные переменные $\mu_s^{k+1} = \mu_s^k + \rho(x_s^{*k+1} - E[x^{*k+1}])^T$, где ρ некоторый коэффициент, который вначале полагается равным 1 и может регулироваться. Алгоритм оканчивает работу, когда среднее изменение μ_s становится небольшим.

Тестовые расчеты показали, что PH алгоритм в описанном варианте имеет медленную скорость сходимости, а регулирование множителя ρ является скорее искусством, зависящим от задачи, чем правилом.

Поскольку алгоритм проектирования на подпространство ограничений (8) прост, то задачу (7) – (9) по двойственным переменным можно решать любым методом безусловной оптимизации, например, PNK-методом [6], заменив градиент на его проекцию. Если точка минимума в задаче (9) определяется однозначно при любом μ_s и множество решений ограничено, то регуляризирующая добавка может не применяться. В противном случае такая добавка может быть применена с небольшим весовым коэффициентом ε к $\|x\|^2$ так, как это описано в [3], либо к $\left\|x - E[x^{*k}]\right\|^2$ так, как это описано в [6].

Вычислительные эксперименты показали, что использование других алгоритмов решения задачи максимизации и других способов регуляризации позволяют существенно ускорить сходимость рассмотренного метода декомпозиции.

В.В. Бойко, В.М. Кузьменко

ПРИСКОРЕННЯ ЗБІЖНОСТІ МЕТОДУ ДЕКОМПОЗИЦІЇ «PROGRESSIVE HEDGING»

Показано як метод розв'язку багатоетапних задач стохастичного програмування «Progressive Hedging» пов'язаний із методом декомпозиції по змінним першого рівня за допомогою множників Лагранжа на прикладі двоетапної задачі. Обговорюються питання регулювання швидкості збіжності методу та відновлення змінних першого рівня.

V.V. Boyko, V.M. Kuzmenko

INCREASING CONVERGENCE RATE OF «PROGRESSIVE HEDGING» DECOMPOSITION ALGORITHM

It is shown relation between «Progressive Hedging» method for solving multistage stochastic optimization problems and decomposition method using Lagrange multipliers in case of Two Stage stochastic problem. Adjusting of convergence rate and restoring of first stage variables are discussed.

Список литературы

1. Rockafellar R.T., Wets R.J-B. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*. **16**. 1991. P. 119 – 147.
2. Watson J.P., Woodruff D.L. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*. **8**. 2010. P. 355 – 370.
3. Михалевич В.С., Трубин В.А., Шор Н.З. Оптимизационные задачи производственно-транспортного планирования. М.: Наука, 1986. 264 с.
4. Беляева Л.В., Журбенко Н.Г., Шор Н.З. О методе решения одного класса динамических распределительных задач. *Экономика и математические методы*. 1978. Т. 14, Вып. 1. С. 137 – 146.
5. Лаптин Ю.П. Точные штрафные функции и выпуклые продолжения функций в схемах декомпозиции по переменным. *Кибернетика и системный анализ*. 2016. Т. 52, № 1. С. 93 – 104.
6. Кузьменко В.Н., Бойко В.В. Использование PNK-метода для решения невыпуклых задач оптимизации. *Теорія оптимальних рішень.* К.: Ін-т кібернетики імені В.М. Глушкова НАН України. 2012. С. 47 – 52.

Получено 13.04.2018