运筹学与控制论

DOI:10.3969/J. ISSN. 1672-6693. 2010. 06. 002

增广拉格朗日函数的两种可分化方法之比较*

王　磊，白富生
(重庆师范大学 数学学院，重庆 400047)

摘要:可分方法用于将一个复杂的大规模优化问题分解成各个子问题进行求解。增广拉格朗日松弛方法的主要缺点是由其引入的二次项是不能分离的。为了处理这种增广拉格朗日函数的不可分离性,可将辅助问题原理方法或分块坐标下降方法应用于增广拉格朗日松弛方法。与已有文献中对带有约束条件 $x-\bar{x}=0$ 的优化问题进行这两种可分方法的比较不同,本文对带有更一般的约束条件——线性约束 $z=Ax$ 的优化问题进行这两种可分化方法的比较;最后给出的两个算例证实了本文的理论分析结果——在处理不可分离的增广拉格朗日函数的时候,在一定条件下,分块坐标下降法往往比辅助问题原则法更快得到最优值。

关键词:可分化方法;增广拉格朗日松弛;辅助问题原理;分块坐标下降
中图分类号:O221.2　　文献标识码:A　　文章编号:1672-6693(2010)06-0007-05

1 预备知识

为了解决数学规划中的大规模问题,可分化方法的产生是有必要的。可分化方法又称为分解方法,可应用于多区域电力系统分析、网络设计、价格决策管理等模型中[1-5],早在上世纪 60 年代就已提出,如 Dantzig-Wolfe 分解和 Benders 分解[6]。后来有不少相关的研究成果问世,现在拉格朗日松弛(LR)方法是使用最广泛的方法。而在 LR 方法中,有两个主要的方法:经典拉格朗日松弛(The classical Lagrangian relaxation)方法[7-9]和增广拉格朗日松弛(The augmented Lagrangian relaxation)方法[10-12],分别简记为 CLR 方法、ALR 方法。CLR 方法在求解优化问题时,由于对偶间隙的存在常常会得到一个不可行的解,而 ALR 方法好于 CLR 方法的一个方面就是它可能会得到一个可行解;但是在解非凸问题时,ALR 方法可能会得到一个局部最优值;除此之外,ALR 方法另一个主要缺点是由增广拉格朗日函数引入的二次项是不可分离的。

为了克服这个缺点,使用最广的方法是辅助问题原则 APP 方法[13]和分块坐标下降(BCD)方法[14]。在文献[15]中,作者用 APP 方法和 BCD 方法比较求解带有约束 $x-\bar{x}=0$ 的优化问题的快慢,结果表明分块坐标下降法比辅助问题原则法更快得到最优值。本文将文献[15]中的约束推广为更一般的线性约束 $z=Ax$,然后去比较在这种条件下的 APP 方法和 BCD 方法。即将约束进行了推广,使约束条件更一般化,扩大了应用的范围。

本文分为两个部分。第一部分介绍 ALR 方法、APP 方法和 BCD 方法;第二部分从理论和算例比较了两种可分化方法,即 APP 法和 BCD 法。

1.1 增广拉格朗日松弛方法

为了解决以下原始问题($x\in\mathbf{R}^n,z\in\mathbf{R}^m$)

$$\min f(x)+g(z) \qquad (1)$$
$$\text{s.t. } z=Ax, x\in D, z\in\overline{D}$$

其中 $f:\mathbf{R}^n\to(-\infty,+\infty]$,$g:\mathbf{R}^m\to(-\infty,+\infty]$,且 A 是一个 $m\times n$ 矩阵。

(1)式的拉格朗日函数即 $L(x,z,\lambda):=f(x)+g(z)+\lambda^{\mathrm{T}}(Ax-z)$,用它可以得到其对偶问题 $\max\limits_{\lambda\in\mathbf{R}^m} q(\lambda)$,

* 收稿日期:2010-04-12
资助项目:国家自然科学基金(No. 10171118)
作者简介:王磊,男,硕士研究生,研究方向为最优化理论与算法;通讯作者:白富生,E-mail:fsbai@cqnu.edu.cn

其中对偶函数为 $q(\lambda) := \min_{x \in D, z \in \bar{D}} f(x) + g(z) + \lambda^{\mathrm{T}}(Ax - z)$。即对于固定的拉格朗日乘子，要得到以上问题的最优解，可以分解为求两个独立问题的最优解。但是在非凸情况下，由于对偶间隙的存在，可能降低了这个方法的可应用性；即使在凸的情况下由于对偶问题 $q(\lambda) := \min_{x \in D, z \in \bar{D}} f(x) + g(z) + \lambda^{\mathrm{T}}(Ax - z)$ 的非光滑性，也可能存在不足的地方。

通过使用增广拉格朗日函数，有许多方法凸化参数问题，在下节将介绍其中两种方法。为了把对偶问题分解为两个子问题，一个定义在 D 上，另一个定义在 $\bar{D}$ 上，将增广拉格朗日函数中的耦合约束 $Ax - z$ 松弛，这意味着在目标函数 $f(x) + g(z)$ 中增加了拉格朗日项 $\lambda^{\mathrm{T}}(Ax - z)$ 和惩罚项 $\frac{c}{2}\|Ax - z\|^2$。形成的这种最大—最小化问题称为对偶问题，即 $\max_{\lambda \in \mathbf{R}^m}\left\{\min_{x \in D, z \in \bar{D}} f(x) + g(z) + \lambda^{\mathrm{T}}(Ax - z) + \frac{c}{2}\|Ax - z\|^2\right\}$。

通常情况下，增广拉格朗日函数被定义为

$$L_c(x, z, \lambda) := f(x) + g(z) + \lambda^{\mathrm{T}}(Ax - z) + \frac{c}{2}\|Ax - z\|^2 \tag{2}$$

相应的对偶函数为 $q_c(\lambda) := \{\min_{x \in D, z \in \bar{D}} L_c(x, z, \lambda)\}$。于是，对偶问题可描述为 $\max_{\lambda \in \mathbf{R}^m} q_c(\lambda)$。

注意到，增广拉格朗日函数(2)式，由于有二次惩罚项 $\frac{c}{2}\|Ax - z\|^2$ 而不能将其分离开。

1.2 可分化方法

本文的焦点是基于不可分的增广拉格朗日函数的最小化 $\min_{x \in D, z \in \bar{D}} f(x) + g(z) + \lambda^{\mathrm{T}}(Ax - z) + \frac{c}{2}\|Ax - z\|^2$，将其分化为小的子问题这种方法称为辅助问题原则方法(APP)。粗糙地说，APP 方法将拉格朗日函数中的二次项 $\frac{c}{2}\|Ax - z\|^2$ 在当前迭代 (x_n, z_n) 处线性化，再增加一个二次的可分离项 $(\frac{b}{2})(\|x - x_n\|^2 + \|z - z_n\|^2)$，于是有

$$\min_{x \in D, z \in \bar{D}} f(x) + g(z) + \lambda^{\mathrm{T}}(Ax - z) + c(Ax_n - z_n)^{\mathrm{T}}(Ax - z) + (\frac{b}{2})(\|x - x_n\|^2 + \|z - z_n\|^2)$$

增加的二次项 $(\frac{b}{2})(\|x - x_n\|^2 + \|z - z_n\|^2) = (\frac{b}{2})\|(x, z)^{\mathrm{T}} - (x_n, z_n)^{\mathrm{T}}\|^2$

的主要作用是使部分项线性化的增广拉格朗日函数成为严格凸函数。

于是，由于 L_c 中添加了这个逼近项，求解 L_c 的最小化问题可以分化为求解两个子问题的最值((x_{n+1}, z_{n+1}) 是新的迭代)

$$x_{n+1} := \operatorname*{argmin}_{x \in D} f(x) + \lambda_n^{\mathrm{T}} Ax + c(Ax_n - z_n)^{\mathrm{T}} Ax + (\frac{b}{2})\|x - x_n\|^2$$

$$z_{n+1} := \operatorname*{argmin}_{z \in \bar{D}} g(z) - \lambda_n^{\mathrm{T}} z - c(Ax_n - z_n)^{\mathrm{T}} z + (\frac{b}{2})\|z - z_n\|^2$$

另一个可供选择的方法是分块坐标下降(BCD)法，也称为非线性高斯—赛德尔方法。与 APP 方法不同的是，BCD 方法直接将 L_c 最小化。当在定义域 D 内最小化时，定义域 $\bar{D}$ 内的变量在它们当前最好的值固定，这个值记作 z_n 类似地，当在定义域 $\bar{D}$ 内最小化时，定义域 D 内的变量在它们当前最好的值固定，这个值记作 x_{n+1}，依此下去。在凸的情况下，收敛性能够保证[14]。于是，L_c 的最小化问题可分解为两个子问题((x_{n+1}, z_{n+1}) 是新的迭代)

$$x_{n+1} := \operatorname*{argmin}_{x \in D} f(x) + \lambda_n^{\mathrm{T}} Ax + (\frac{c}{2})\|Ax - z_n\|^2, z_{n+1} := \operatorname*{argmin}_{z \in \bar{D}} g(z) - \lambda_n^{\mathrm{T}} z + (\frac{c}{2})\|Ax_{n+1} - z\|^2$$

1.3 可分化算法

这部分用一个熟知的事实：对偶函数在当前对偶迭代的梯度等于松弛约束在当前原迭代的值，即 $\nabla q_c(\lambda_n) = (Ax_n - z_n)$ 带有辅助问题原则(APP)的增广拉格朗日松弛(ALR)方法可以总结为以下的 ALR + APP 算法(ALR 算法也称为乘子法)。

ALR + APP 算法

第一步,核对终止准则。如果对偶函数的梯度的范数 $\|Ax_n - z_n\| < \varepsilon$,则停止,$(x_n, z_n, \lambda_n)$ 是最优解;

第二步,计算 $x_{n+1} := \underset{x \in D}{\operatorname{argmin}} f(x) + \lambda_n^{\mathrm{T}} Ax + c_n (Ax_n - z_n)^{\mathrm{T}} Ax + \left(\frac{b_n}{2}\right) \|x - x_n\|^2$;

第三步,计算 $z_{n+1} := \underset{z \in \bar{D}}{\operatorname{argmin}} g(z) - \lambda_n^{\mathrm{T}} z - c_n (Ax_n - z_n)^{\mathrm{T}} z + \left(\frac{b_n}{2}\right) \|z - z_n\|^2$;

第四步,对偶变量的更新。令 $\lambda_{n+1} = \lambda_n + c_n (Ax_{n+1} - z_{n+1})$;

第五步,罚参数更新。如果 $\|Ax_{n+1} - z_{n+1}\| > \alpha \cdot \|Ax_n - z_n\|$,则令 $c_{n+1} := \beta c_n$,$b_{n+1} := \gamma c_{n+1}$。一个合适的选择是 $\alpha := 1.10$,$\beta := 2$,$\gamma := 2$。

类似地,带有分块坐标下降(BCD)的 ALR 方法可以总结为以下的 ALR + BCD 算法。

ALR + BCD 算法

第一步,核对终止准则。如果对偶函数的梯度的范数 $\|Ax_n - z_n\| < \varepsilon$,则停止。$(x_n, z_n, \lambda_n)$ 是最优解;

第二步,计算 $x_{n+1} := \underset{x \in D}{\operatorname{argmin}} f(x) + \lambda_n^{\mathrm{T}} Ax + \left(\frac{c_n}{2}\right) \|Ax - z_n\|^2$;

第三步,计算 $z_{n+1} := \underset{z \in \bar{D}}{\operatorname{argmin}} g(z) - \lambda_n^{\mathrm{T}} z + \left(\frac{c_n}{2}\right) \|Ax_{n+1} - z\|^2$;

第四步,对偶变量的更新。令 $\lambda_{n+1} = \lambda_n + c_n (Ax_{n+1} - z_{n+1})$;

第五步,罚参数更新。如果 $\|Ax_{n+1} - z_{n+1}\| > \alpha \cdot \|Ax_n - z_n\|$,则令 $c_{n+1} := \beta c_n$。一个合适的选择是 $\alpha := 1.10$,$\beta := 2$。

2 理论比较

这一部分从理论的观点比较了 APP 法和 BCD 法。

命题 1　辅助问题原则 APP 法中的迭代等价于

$$x_{n+1} := \underset{x \in D}{\operatorname{argmin}} f(x) + \lambda_n^{\mathrm{T}} Ax + \left(\frac{c_n}{2}\right) \|Ax - z_n\|^2 + \frac{b_n}{2} \|x - x_n\|^2 - \frac{c_n}{2} \|Ax - Ax_n\|^2 \tag{3}$$

$$z_{n+1} := \underset{z \in \bar{D}}{\operatorname{argmin}} g(z) - \lambda_n^{\mathrm{T}} z + \left(\frac{c_n}{2}\right) \|Ax_n - z\|^2 + \left(\frac{b_n - c_n}{2}\right) \|z_n - z\|^2 \tag{4}$$

其中 (x_n, z_n) 是当前的迭代的输入对,而 (x_{n+1}, z_{n+1}) 是下个迭代的输出对。

证明　考虑

$$c(Ax_n - z_n)^{\mathrm{T}} Ax + \left(\frac{b}{2}\right) \|x - x_n\|^2 = c(Ax_n)^{\mathrm{T}} Ax - c z_n^{\mathrm{T}} Ax + \frac{b-c}{2} \|x - x_n\|^2 + \frac{c}{2} \|x - x_n\|^2 =$$

$$\frac{b-c}{2} \|x - x_n\|^2 + \frac{c}{2} \|Ax\|^2 - c z_n^{\mathrm{T}} Ax + \frac{c}{2} \|z_n\|^2 - \frac{c}{2} \|Ax\|^2 - \frac{c}{2} \|z_n\|^2 + \frac{c}{2} \|x\|^2 - c x^{\mathrm{T}} x_n + \frac{c}{2} \|x_n\|^2 + c \|A\|^2 x^{\mathrm{T}} x_n =$$

$$\frac{b-c}{2} \|x - x_n\|^2 + \frac{c}{2} \|Ax - z_n\|^2 + \frac{c}{2} \left(- \|Ax\|^2 - \|z_n\|^2 + \|x\|^2 - 2x^{\mathrm{T}} x_n + \|x_n\|^2 + 2x^{\mathrm{T}} A^{\mathrm{T}} A x_n \right) =$$

$$\frac{b-c}{2} \|x - x_n\|^2 + \frac{c}{2} \|Ax - z_n\|^2 + \frac{c}{2} \left[\|x - x_n\|^2 - \|Ax - Ax_n\|^2 + \|Ax_n\|^2 - \|z_n\|^2 \right] =$$

$$\frac{c}{2} \|Ax - z_n\|^2 + \frac{b}{2} \|x - x_n\|^2 - \frac{c}{2} \|Ax - Ax_n\|^2 + \frac{c}{2} \left(\|Ax_n\|^2 - \|z_n\|^2 \right)$$

则在 D 上的最小化　$\min\limits_{x \in D} f(x) + \lambda_n^{\mathrm{T}} Ax + c(Ax_n - z_n)^{\mathrm{T}} Ax + \left(\frac{b}{2}\right) \|x - x_n\|^2$

可以写为

$$\min_{x \in D} f(x) + \lambda_n^{\mathrm{T}} Ax + \frac{c}{2} \|Ax - z_n\|^2 + \frac{b}{2} \|x - x_n\|^2 - \frac{c}{2} \|Ax - Ax_n\|^2 + \frac{c}{2} \left(\|Ax_n\|^2 - \|z_n\|^2 \right) \tag{5}$$

类似地,考虑 $-c(Ax_n - z_n)^{\mathrm{T}} z + \left(\frac{b}{2}\right) \|z - z_n\|^2 = -c(Ax_n)^{\mathrm{T}} z + c(z_n)^{\mathrm{T}} z + \left(\frac{b-c}{2}\right) \|z - z_n\|^2 + \left(\frac{c}{2}\right) \|z - z_n\|^2 =$

$$\left(\frac{b-c}{2}\right)\|z-z_n\|^2+\left(\frac{c}{2}\right)\|z_n\|^2-c(z_n)^{\mathrm T}z+\left(\frac{c}{2}\right)\|z\|^2-c(Ax_n)^{\mathrm T}z+c(z_n)^{\mathrm T}z=$$

$$\left(\frac{b-c}{2}\right)\|z-z_n\|^2+\left(\frac{c}{2}\right)\|Ax_n-z\|^2-\frac{c}{2}(\|Ax_n\|^2-\|z_n\|^2)$$

则在$\overline{D}$上的最小化$\min\limits_{z\in\overline{D}} g(z)-\lambda_n^{\mathrm T}z-c(Ax_n-z_n)^{\mathrm T}z+\left(\frac{b}{2}\right)\|z-z_n\|^2$,可以写为

$$\min_{z\in\overline{D}} g(z)-\lambda_n^{\mathrm T}z+\left(\frac{b-c}{2}\right)\|z-z_n\|^2+\left(\frac{c}{2}\right)\|Ax_n-z\|^2-\frac{c}{2}(\|Ax_n\|^2-\|z_n\|^2) \tag{6}$$

于是,目标函数(5)式和目标函数(6)式中的固定项$\frac{c}{2}(\|Ax_n\|-\|z_n\|)$和$\frac{c}{2}(\|z_n\|-\|Ax_n\|)$可以不要,这也不会改变最优的迭代。因此,APP法中的迭代等价于

$$x_{n+1}:=\underset{x\in D}{\operatorname{argmin}} f(x)+\lambda_n^{\mathrm T}Ax+\left(\frac{c_n}{2}\right)\|Ax-z_n\|^2+\frac{b_n}{2}\|x-x_n\|^2-\frac{c_n}{2}\|Ax-Ax_n\|^2$$

$$z_{n+1}:=\underset{z\in\overline{D}}{\operatorname{argmin}} g(z)-\lambda_n^{\mathrm T}z+\left(\frac{c_n}{2}\right)\|Ax_n-z\|^2+\left(\frac{b_n-c_n}{2}\right)\|z_n-z\|^2$$

这就是要证明的结果。 证毕

推论1[15] 对于$b=c,A=I$,APP法就是非线性高斯—赛德尔或BCD法的雅可比式。

证明 比较命题1和以下的BCD法(非线性高斯—赛德尔法)

$$x_{n+1}:=\underset{x\in D}{\operatorname{argmin}} f(x)+\lambda_n^{\mathrm T}Ax+\left(\frac{c}{2}\right)\|Ax-z_n\|^2,\ z_{n+1}:=\underset{z\in\overline{D}}{\operatorname{argmin}} g(z)-\lambda_n^{\mathrm T}z+\left(\frac{c}{2}\right)\|Ax_{n+1}-z\|^2 \tag{7}$$

对于$b=c,A=I$,(3)~(4)式的最后一项就没有了。因此,这两种方法的不同之处仅在于:在(4)式中,APP法用的是旧的迭代x_n(雅可比方法),然而在(7)式中BCD法用的是新的迭代x_{n+1}(非线性高斯—赛德尔法)。 证毕

众所周知的是在数值分析中非线性高斯—赛德尔法很可能优于雅可比法。主要原因就是非线性高斯—赛德尔方法在当前的迭代中就直接使用新产生的x_{n+1},而雅可比方法在下个迭代开始才使用新产生的x_{n+1}(在当前的迭代中使用的是x_n)。因此,在$b=c,A=I$这种情况下,与APP方法相比,BCD方法能更快的收敛到最优值。

在命题1中,如果考虑目标函数在$D\times\overline{D}$上的最小化,当$A=I$时,(3)式中的项$\frac{b}{2}\|x-x_n\|^2-\frac{c}{2}\|Ax-Ax_n\|^2$和(4)式中的项$\left(\frac{b-c}{2}\right)\|z-z_n\|^2$可以合并写成$\left(\frac{b-c}{2}\right)\|(x,z)^{\mathrm T}-(x_n,z_n)^{\mathrm T}\|^2$,若$b=c$,则前式将不存在;但是对于任意的$b>c$,它将惩罚远离$(x_n,z_n)$的迭代对$(x_{n+1},z_{n+1})$。

当$A\neq I$,有

$$\frac{b}{2}\|x-x_n\|^2-\frac{c}{2}\|Ax-Ax_n\|^2\geqslant\frac{b}{2}\|x-x_n\|^2-\frac{c}{2}\|A\|^2\|x-x_n\|^2=\frac{b-c\|A\|^2}{2}\|x-x_n\|^2 \tag{8}$$

因此,当$\|A\|^2>1$且对于任意的$b>c\|A\|^2$,或者当$\|A\|^2<1$且对于任意的$b>c$都有(8)式中的结果。

综上所述,一般来说,比起$b=c,A=I$这种情况,对于$\begin{cases}\|A\|^2\leqslant 1\\ b>c\end{cases}$或$\begin{cases}\|A\|^2>1\\ b>c\|A\|^2\end{cases}$。APP方法很可能要更多的迭代才能达到最优点,因为前两式的惩罚从当前迭代向后一个迭代的长步长。

3 数值算列

例1

$$\min_{x=(x_1,x_2)\in\mathbf{R}^2,z\in\mathbf{R}} f(x)+g(z)=x_1^2+x_2^2+z^2$$

$$\text{s.t.}\quad (1\quad 1)\begin{pmatrix}x_1\\ x_2\end{pmatrix}=z$$

很容易知道最优解为 $x^* = (0,0)$,$z^* = 0$。在 ALR + APP 算法和 ALR + BCD 算法中,令初始罚参数 c_0 等于 1。参数 b_0 等于 3,常数 α,β,γ 分别等于 1.1,2,2。前面提到的两种方法解决这个问题的一些数据在下表 1 中。

例 2

$$\min_{x=(x_1,x_2)\in \mathbf{R}^2, z\in \mathbf{R}} f(x) + g(z) = 2x_1^2 + x_2^2 + x_3^2 + 2z^2$$

$$\text{s.t.} \begin{pmatrix} 0.5 & -0.5 & 0.25 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = z$$

与例 1 类似地,有下面的表 2。

表 1　例 1 的两种算法比较

方法	(x^0,z^0)	(x^*,z^*)	$f(x^*)+g(z^*)$	迭代次数 k
ALR + APP	$\begin{pmatrix}0.4\\0.4\\0.4\end{pmatrix}$	$\begin{pmatrix}0.048\,1\text{e}-004\\-0.140\,5\text{e}-004\\-4.440\,9\text{e}-016\end{pmatrix}$	2.205 8e−010	24
ALR + BCD	$\begin{pmatrix}0.4\\0.4\\0.4\end{pmatrix}$	$\begin{pmatrix}0.069\,1\text{e}-005\\0.848\,6\text{e}-005\\-4.440\,9\text{e}-016\end{pmatrix}$	7.249 3e−011	17

表 2　例 2 的两种算法比较

方法	(x^0,z^0)	(x^*,z^*)	$f(x^*)+g(z^*)$	迭代次数 k
ALR + APP	$\begin{pmatrix}0.4\\0.4\\0.4\\0.4\end{pmatrix}$	$\begin{pmatrix}-0.010\,2\text{e}-003\\0.043\,0\text{e}-003\\0.130\,9\text{e}-003\\-4.440\,9\text{e}-016\end{pmatrix}$	1.919 4e−008	18
ALR + BCD	$\begin{pmatrix}0.4\\0.4\\0.4\\0.4\end{pmatrix}$	$\begin{pmatrix}0.006\,6\text{e}-005\\-0.013\,3\text{e}-005\\0.006\,6\text{e}-005\\0.011\,6\text{e}-016\end{pmatrix}$	5.790 5e−004	9

从表 1,2 中容易看到,一般来说,BCD 法比 APP 法更快得到最优值。

4 结论

从理论上和算例上看,在处理不可分离的增广拉格朗日函数的时候,在一定条件下,分块坐标下降法往往比辅助问题原则法更快得到最优值。

参考文献:

[1] Virmani S,Adrian C. Implementation of a Lagrangian relaxatio based unit commitment problem[J]. IEEE Transactions on Power Systems,1989,4:1373-1380.

[2] Pellegrino F,Renaud A. Bundle and augmented Lagrangian methods for short-term unit commitment[C]//Germany:12th Power System Computation Conference,Dresden,1996:730-739.

[3] Heredia F J. Constrained nonlinear network flow problems through projected Lagrangian methods,problems in modern applied mathematics[C]//Mastorakis N E. Mathematics and computers in science and engineering. Athens,Greece:World Scientific and Engineering Society Press,2000:406-411.

[4] Engelmann B. Convexification and decomposition of separable non-convex optimization problem[J]. Optimization,1992,26:61-82.

[5] Chen Gong,Teboulle M. A proximal-based decomposition method for convex minimization problems[J]. Mathematical Programming,1994,64:81-101.

[6] Mahey P. Decomposition methods for mathematical programming[M]//Pardalos P,Resende M G C. Handbook of applied optimization. Oxford:Oxford University Press,2002.

[7] Merlin A,Sandrin P. A new method for unit commitment at Electricite′ de France[J]. IEEE Transactions on Power Apparatus and Systems,1983,102:1218-1225.

[8] Tong S K,Shahidehpour S M. An innouatiue approach to generation scheduling in Large-ScaleHydro-Thermal power systems with fuel constrained units[J]. IEEE Transactions on Power Systems,1990,5:665-673.

[9] JIME′NEZ N,Conejo A. Short-Term hydrothermal coordination by Lagrangian relaxation:solution of the dual problem[J]. IEEE Transactions on Power Systems,1999,14:89-95.

[10] Beltran C,Heredia F J. Short-Term hydrothermal coordination by augmented Lagrangian relaxation:A new multiplier updating[J]. Investigacio′n Operativa,1999,8:63-76.

[11] Batut J,Renaud A. Daily generation scheduling optimization with transmission constraints:a new class of algorithms[J]. IEEE Transactions on Power Systems,1992,7:982-989.

[12] BALDICK R. The generalized unit commitment problem [J]. IEEE Transactions on Power Systems, 1995, 10: 465-475.

[13] Cohen G. Auxiliary problem principle and decomposition of optimization problems[J]. Journal of Optimization Theory and Applications, 1980, 32: 277-305.

[14] Bertsekas D P. Nonlinear programming[M]. Belmont: Athena Scientific, 1995.

Operations Research and Cybernetics

Comparison of Two Decomposition Approaches with the Augmented Lagrangian Function

WANG Lei, *BAI Fu-sheng*

(College of Mathematics Science, Chongqing Normal University, Chongqing 400047, China)

Abstract: The decomposition methods are used to solve large-scale optimization problems by decomposition them into sub-problems. The main drawback of the augmented Lagrangian relaxation method is that the quadratic term introduced by the augmented Lagrangian is not separable. To cope with the non-separability of the augmented Lagrangian function, we can apply auxiliary problem principle (APP) method or block coordinate descent (BCD) method to the augmented Lagrangian relaxation method. Compared with the literature [24] in solving the optimization problem with constraints $x-\bar{x}=0$, we compare these two decomposition methods solving optimization problem with more general constraints——linear constrains $z=Ax$. Two numerical examples are to show comparison of theoretical validation—In dealing with non-separable augmented Lagrangian function, we can often expect faster performance of the BCD method compared to the APP method under certain conditions.

Key words: decomposition methods; augmented Lagrangian relaxation; auxiliary problem principle; block coordinate descent

(责任编辑 黄 颖)