Modern Applied Science

www.ccsenet.org/journal.html

Vol. 2, No. 5

September 2008

# Adaptive Multicut Aggregation Method for Solving Two-stage Stochastic Convex Programming with Recourse

Jingsheng Liu(Corresponding author)
College of Information Science and Engineering
Shandong University of Science and Technology
579 Qian Wan Gang Street, Huangdao District, Qingdao 266510, China
E-mail:jingsheng2008@163.com
Changyin Zhou
College of Information Science and Engineering
Shandong University of Science and Technology
579 Qian Wan Gang Street, Huangdao District, Qingdao 266510, China
E-mail: zhoucy@sdust.edu.cn
Xiuping Zhang
College of Information Science and Engineering
Shandong University of Science and Technology
579 Qian Wan Gang Street, Huangdao District, Qingdao 266510, China
E-mail: pingxiu8384@163.com

**Abstract**

In this paper we extend directly adaptive multicut aggregation method of Svyatoslav Trukhanov, Lewis Ntaimo and Andrew Schaefer to solving two-stage problems of stochastic convex programming. The implement of the algorithm is simple, so less computation work is needed. The algorithm has certain convergence.

**Keywords:** Multicut aggregation, Adaptive cut, Stochastic programming, Stochastic convex programming

## 1. Introduction

(Birge, J. R. & Louveaux, F. V.,1997)L-shaped algorithm for stochastic programming problems with recourse generally generate a single cut at each major iteration. While(Ruszczynski, A. & Shapiro, A.,2004) the multicut version of the algorithm allows for cuts up to the number of outcomes to be placed at once. So the L-shaped algorithm tends to have more major iterations than the multicut algorithm, however the multicut algorithm have more optimality cuts, the size of master problem relatively big. To settle the disadvantage of L-shaped algorithm and multicut algorithm, Svyatoslav Trukhanov, Lewis Ntaimo and Andrew Schaefer (Trukhanov, S.,Ntaimo, L., & Schaefer, A.,2007) have proposed the adaptive multicut aggregation algorithm for solving two stage stochastic linear programming with recourse. The computational results of the algorithm shows that the algorithm is more effective than L-shaped algorithm and multicut algorithm.

Two-stage problems of stochastic convex programming are difficult for solving. For solving they are transformed into problems of linear programming by linearization, but the size of the problems is greatly increased, the problems are difficult to solve. In this paper we extend directly adaptive multicut aggregation method for solving two-stage stochastic linear programming problems to two-stage stochastic convex programming with recourse. The method is simple and has certain convergence.

## 2. Two-Stage Stochastic Convex Programming

A two-stage stochastic convex programming with fixed recourse in the extensive form can be given as follow:

$$\min_{x\in X}\left\{f(x):=f_1(x)+\sum_{s=1}^{S}p_sQ^s(x)\right\} \tag{1}$$

Where

$$Q^s(x)=\min_{y\in Y^s}f_2^s(x) \tag{2}$$

$$\text{s.t. } T^sx+Wy=h^s$$

$$y\in Y^s$$

Where s is the scenario(outcome). $f_1(x):R^{n_1}\to R$ and $f_2^s(x):R^{n_2}\to R\,,s=1,2,3\cdots S$ ,are convex functions. $X$ and $Y^s$ are bounded convex sets. $T^s=T(s)$ is $m_2\times n_1$ matrix, $W$ is $m_2\times n_2$ matrix, $h^s=h(s),h^s$ is $m_2$ vector.

### 3. Adaptive Multicut Aggregation Method

The adaptive multicut aggregation method (Trukhanov, S.,Ntaimo, L., & Schaefer, A.,2007) has point out that this method has two advantages comparing L-shaped algorithm and multicut algorithm. First, it use more information from subproblems, which assumes adding more than one cut at each major iteration avoiding 'information loss'; Second ,it can keep the size of master problem relatively small, which requires to limit the number of cuts added to master problem. The adaptive multicut aggregation method is required to partition the sample space $S$ into $D$ aggregates $\{S_d\}_{d=1}^{D}$, such that $S=S_1\bigcup S_2\bigcup\cdots\bigcup S_D$, $\forall i\neq j,S_i\bigcap S_j=\varnothing$. The sample space $S$ can be partitioned into some subsets based on some aggregation rules. So each subset can generate a optimality cut, and it introduce $D$ different optimality variables $\theta_d$.

At the k-th iteration, let the scenario set partition be $S(k)=\left\{S_1,S_2,\cdots S_{L_k}\right\}$,where the set $S_i,i=1,2,\cdots L_k$, are disjoint, that is $\forall i\neq j,S_i\bigcap S_j=\varnothing$ and $\bigcup_{i=1}^{L_k}S_i=S$. Let $F(k)$ denote the set of iteration numbers up to $k$ where all subproblems are feasible and optimality cuts are generated .The optimality cut at k-th iteration has following form:

$$\overline{\alpha}_d^t+\left(\overline{g}_d^t\right)^T x\leq\theta_d,t\in F(k),d\in S(k) \tag{3}$$

Where $\overline{\alpha}_d^t,\overline{g}_d^t$ is defined in (14). Then the method generated $|S(k)|$(the number of element in $S(k)$ ) different optimality cuts. If for every $d\in S(k)$, the following formula hold:

$$\overline{\alpha}_d^t+\left(\overline{g}_d^t\right)^T x^k\geq\theta_d,t\in F(k),d\in S(k) \tag{4}$$

the algorithm stop, $x^k$ is optimal, otherwise the algorithm add optimality cut like (3) to the master problem.

At each iteration, the number of aggregation $D$ must satisfy $1<D<|S(k)|$. The master problem will have 'adaptive' optimality variables $\theta_d$, that is ,the number of optimality variables will change during the course of the algorithm. The goal is to let the algorithm use more information and then settle for a level of aggregation that tends to faster convergence to the optimal solution. Svyatoslav Trukhanov, Lewis Ntaimo and Andrew Schaefer (Trukhanov, S., Ntaimo, L., & Schaefer, A.,2007) has proposed the redundancy threshold $\delta\,(0<\delta<1)$to be a aggregation rule. In the master problem, if the optimality cuts contain 'little' information about the optimal solution, then these cuts are inactive. These inactive optimality cuts can be aggregated into one cut without information loss. Conside some iteration k after solving the master problem for some aggregate $d\in S(k)$, let $f_d$ be the number of iterations when optimality cuts corresponding to $d$ are redundant. If $\dfrac{f_d}{|F(k)|}>\delta$, then all $d$ are combined to form one aggregate, and there is a new optimality cut generated in the next iteration. As a supplement to the redundancy threshold, a bound on the minimum and maximum number of aggregation $|S(k)|$ should be imposed. This prevents the algorithm leading to the L-shaped algorithm (highest level of cut aggregation) and multicut algorithm(no cut aggregation).

### 4. description of the algorithm

#### *4.1 the master problem*

For $S_i\in S(k)=\left\{S_1,S_2,\cdots S_{L_k}\right\}$, the aggregation probability of $S_i$ is defined as $p_{S_i}=\sum_{s\in S_i}p_s$ with $\sum_{i=1}^{L_k}p_{S_i}=\sum_{s\in S}p_s=1$. Then the master problem has the following form:

$$\min \sum_{d\in S(k)} \theta_d \tag{5a}$$

$$\text{s.t.} \quad \overline{\alpha}_d^j + \left(\overline{\beta}_d^j\right)^T x \le \theta_d, j \in F(k), d \in S(k) \tag{5b}$$

$$\beta^{j,s} + \left(\gamma^{j,s}\right)^T x \le 0, j \in \{1, \cdots k\} \setminus F(k), s \in S \tag{5c}$$

$$x \in X \tag{5d}$$

Where (5b) is the optimality cut, (5c) is the feasibility cut.

*4.2 feasibility cut*

The method of getting feasibility cut is the same as one for stochastic linear programming(Ruszczynski, A., & Shapiro, A.,2004).

Solving following programming:

$$\min_{y,z} \|z\| \tag{6}$$

$$\text{s.t.} \quad Wy + z = h^s - T^s x$$

$$y \in Y^s$$

Where $z$ is the artificial variable, $\|\cdot\|$ is a norm on the space $R^{m_2}$.Let $U^s(x)$ be the optimal of problem (6).If $U^s(x^k) = 0$, $\forall s \in S$ ,then $x^k$ is the feasible point of problem (2),otherwise feasibility cut like (5c) is generated $\beta^{k,s} + \left(\gamma^{k,s}\right)^T x \le 0$,

$$\text{where} \quad \begin{cases} \gamma^{k,s} \in \partial U^s(x^k) \\ \beta^{k,s} = U^s(x^k) - (\gamma^{k,s})^T x^k \end{cases} \tag{7}$$

$$\text{and} \quad U^s(x) \ge U^s(x^k) + (\gamma^{k,s})^T (x - x^k) = \beta^{k,s} + \left(\gamma^{k,s}\right)^T x \tag{8}$$

*4.3 optimality cut*

Solving the subproblem (2), let $\pi_s^k$ be the dual multipliers associated with an optimal solution of problem (2).then calculate:

$$\begin{cases} g^{k,s} = -\left(T^s\right)^T \pi_s^k \\ \alpha^{k,s} = \left(h^s\right)^T \pi_s^k \end{cases} \tag{9}$$

For $f_1(x)$ the optimality cut at $x^k$ is

$$f_1(x) \ge \alpha^{k,0} + \left(g^{k,0}\right)^T x \tag{10}$$

$$\text{where} \quad \begin{cases} g^{k,0} \in \partial f_1(x^k) \\ \alpha^{k,0} = f_1(x^k) - \left(g^{k,0}\right)^T x^k \end{cases} \tag{11}$$

So for $f(x)$ the optimality cut at $x^k$ is:

$$f^s(x) \ge \overline{\alpha}^{k,s} + \left(\overline{g}^{k,s}\right)^T x \tag{12}$$

$$\text{where} \quad \begin{cases} \overline{g}^{k,s} = g^{k,0} + g^{k,s} \\ \overline{\alpha}^{k,s} = \alpha^{k,0} + \alpha^{k,s} \end{cases} \tag{13}$$

*4.4 Update Cut Aggregation Level*

Generate aggregation $S(k)$ using $S(k-1)$ based on some aggregation rules, each element of $S(k)$ is a union of some elements from $S(k-1)$.For example , $d_1, d_2, \cdots d_j \in S(k-1)$ are aggregated into $d$ , $d \in S(k)$ ,then $d = \bigcup_{i=1}^{j} d_j$

and $p_d = \sum_{i=1}^{j} p_{d_j}$ ,Master problem will be modified by removing variables $\theta_{d_1} \cdots \theta_{d_j}$ and introducing the new one $\theta_d$ .For each major iteration ,the optimality cut is updated as following:

$$\begin{cases} \overline{g}_d^t = p_d \sum_{i=1}^{j} \left(1/p_{d_i}\right) \overline{g}_{d_i}^t, \forall d \in S(k), t \in F(k) \\ \overline{\alpha}_d^t = p_d \sum_{i=1}^{j} \left(1/p_{d_i}\right) \overline{\alpha}_{d_i}^t, \forall d \in S(k), t \in F(k) \end{cases} \tag{14}$$

where

$$\begin{cases} \overline{g}_{d_i}^t = \sum_{s \in S_{d_i}} p_s \overline{g}^{t,s} \\ \overline{\alpha}_{d_i}^t = \sum_{s \in S_{d_i}} p_s \overline{\alpha}^{t,s} \end{cases} \tag{15}$$

With above formula, a new aggregation optimality cut is generated:

$$\overline{\alpha}_d^t + \left(\overline{g}_d^t\right)^T x \le \theta_d, t \in F(k), d \in S(k) \tag{16}$$

## 5. Algorithm

Step0: Let $k = 0, F(0) = \phi$, initialize $S(0), \theta_i = -\infty, i \in S(0)$;

Step1: For every $s \in S$, solve subproblem (6) at $x = x^k$ :

(a) if $U^s(x^k) = 0$ ,construct the optimality cut (12.);

(b) if $U^s(x^k) > 0$, construct the feasibility cut (8);

Step2: Generate $S(k)$ using $S(k-1)$ based on some aggregation rules, construct aggregation optimality cut (16) ;

Step3: if $f(x^k) = \sum_{d=1}^{|S(k)|} p_d \theta_d$ , $\sum_{d=1}^{|S(k)|} p_d \theta_d = \theta^k$ stop, otherwise $F(k) = F(k-1) \bigcup \{k\}$, continue;

Step4: Solve the master problem (5),if it is infeasible, stop, otherwise denote by $\left(x^{k+1}, \theta^{k+1}\right)$ it's solution, $k = k+1$,and go to step1;

## 6. convergence of the algorithm

Assumption: There exists a constant $C$ such that $\left\| \overline{g}_d^k \right\| \le C$, for all $k = 1, 2, \cdots$ and $d \in S(k)$.

Lemma 1(Ruszczynski, A., & Shapiro, A.,2004): For every $s \in S$, the number of iterations for which $U^s(x^k) > 0$ is finite.

Theorem 1: If problem (1) has no feasible solutions the algorithm will stop at step4 after finitely many iterations; If problem (2) has feasible solutions then the method either stops at stop3 at an optimal solution or generates a sequence of point $\{x^k\}$ ,such that $\lim_{k \to +\infty} f(x^k) = f^*$ ( $f^*$ is the optimal of problem (1)).

Proof: Since the master problem is a relaxation of (1), if the method stops at step4, the original problem is infeasible. Also, It always have $\theta^k \le f^*$, so the method can stop at step3 only if $x^k$ is optimal. It remains to analyze the case of infinitely many steps.

The construction and the use of feasibility cuts is the same as in the linear case. By lemma1 It is true that if the problem has no feasible solutions, the method will discover this after finitely many iterations. Moreover, if feasible points exist and the method does not stop at an optimal solution, It must have $f(x^k) < +\infty$ for all sufficiently large $k$ .

For $\varepsilon > 0$ define $K_\varepsilon = \left\{k : f^* + \varepsilon < f(x^k) < +\infty\right\}$ .Let $k_1, k_2 \in K_\varepsilon$ with $k_1 < k_2$ .

Since $f(x^{k_1}) > f^* + \varepsilon$ and $f^* \ge \theta^{k_1}$ there will be many new optimality cuts at $x^{k_1}$ . It will be in the master from $k_1$ on, so it has to be satisfied at $x^{k_2}$ :

$$f(x^{k_1}) + \left( \sum_{d \in S(k_1)} \overline{g}_d^{k_1} \right)^T (x^{k_2} - x^{k_1}) \le \theta^{k_2} \le f^* \tag{17}$$

On the other hand $\varepsilon < f(x^{k_2}) - f^*$ (18)

By (17) and (18), It must have the following formula :

$$\varepsilon < f(x^{k_2}) - f(x^{k_1}) - \left( \sum_{d \in S(k_1)} \overline{g}_d^{k_1} \right)^T (x^{k_2} - x^{k_1}) \quad (19)$$

The function $f(x)$ is subdifferentiable in its domain and $X$ is compact, so there is a constant $C$ such that $f(x_1) - f(x_2) \le C\|x_1 - x_2\|$, for all $x_1, x_2 \in domf(x) \cap X$ .By assumption, a $C$ big enough can be chose so that $\left\|\overline{g}_d^k\right\| \le C$ for all $k$ . It follows that

$$\varepsilon < \left(|S(k_1)| + 1\right) C \left\|x^{k_1} - x^{k_2}\right\| \quad \text{for all} \quad k_1, k_2 \in K_\varepsilon \quad (21)$$

Since the set $X$ is compact, (21) implies that the set $K_\varepsilon$ is finite for each $\varepsilon > 0$ , then $\lim_{k \to +\infty} f(x^k) = f^*$ .

## 7. Summary

This paper extend directly adaptive multicut aggregation method for solving two-stage stochastic linear programming with recourse to solving two stage problems of stochastic convex programming with recourse. The algorithm of the method is exactly described and its convergence proof is given. In the future, more work should be done to find the effective aggregation rules and apply for multi-stage stochastic programming.

## References

Birge,J.R.,&Louveaux,F.V.(1997). Introduction to Stochastic Programming. Springe-Verlag, (chapter2).

Ruszczynski, A. & Shapiro, A.(2004). Stochastic programming. Handbooks in operations research and management science 10. Elsevier Science, (chapter3).

Trukhanov, S.,Ntaimo, L., & Schaefer, A.(2007). On Adaptive Multicut Aggregation for Two-Stage Stochastic Linear Programs with Recourse.[online] Available: http://www.optimization-online.org/ DB_HTML/2007/11/1849.html (Novenmber, 2007).

Van Slyke, R., & Wets, R. B. (1969). L-shaped linear programs with application to optimal control and stochastic programming. SIAM Journal on Applied Mathematics, 17, 638–663.

Higle, J., & Zhao, L.(2007). Adaptive and nonadaptive samples in solving stochastic linear programs Computational investigation, Submitted to Computational Optimization and Applications.

Higle, J., & Sen, S.(1991). Stochastic decomposition: An algorithm for two-stage stochastic linear programs with recourse. Mathematics of Operational Research, 16, 650–669.

Birge, J.(1985). Aggregation bounds in stochastic linear programming. Mathematical Programming, 31, 25–41.