번들-분해법을 이용한 대규모 비분리 콘벡스 프로그램 해법: 수치 적용결과

A Solution Method using Bundle Based Decomposition for Large-scale Nonseparable Convex Program: Computational Results

박 구 현 , 신 용 식

홍익대학교 산업공학과

<Abstract>

블록-삼각(Block-angular)구조를 갖는 선형 제약식과 분리되지 않는 콘벡스 목적함수의 대규모 비분리 콘벡스 최적화 문제의 해법으로 번들-분해법(Bundle Based Decomposition)을 이용한 알고리즘 SQA(Separable Quadratic Approximation)은 비분리 콘벡스 프로그램을 분리가능한 2차계획법(Separable Quadratic Programming) 문제로 근사화시켜 번들-분해법을 축차적으로 적용한다. 본 연구는 수렴성(local convergence & global convergence) 및 알고리즘 구현[1]에 이어 이에 대한 수치적용 결과를 중심으로 소개한다. 수치 적용은 ANSI C로 작성된 SQA프로그램을 SUN SPARC II에서 실행하였으며 이때 대규모 비분리 최적화 문제의 비분리 목적함수와 블록-삼각 구조의 선형 제약식들의 계수들은 ANSI C의 랜덤함수로 부터 임의의 값들을 이용하였다. 이와같은 다양한 비분리 콘벡스 최적화 문제에 대한 수렴성, 반복회수 및 처리시간등의 결과와 함께 GAMS/MINOS의 최적해를 소개한다.

1. 서론

다음과 같은 선형의 블록-삼각 구조를 갖는 대규모 콘벡스 최적화 문제(LCOP: Large-scale Convex Optimization Problem)를 다룬다.

$$\text{Minimize } f(x_1, x_2, \cdots, x_N) \tag{1.1}$$

Subject to

$$\begin{aligned} B_1x_1 \qquad\qquad\qquad\qquad &= b_1 \\ B_2x_2 \qquad\qquad &= b_2 \\ \cdot \qquad &= \cdot \\ B_Nx_N &= b_N \\ A_1x_1 + A_2x_2 + \cdots + A_Nx_N &\le a \\ x_1 \ge 0 \ ,\ x_2 \ge 0 \ ,\ \cdots ,\ x_N &\ge 0 \end{aligned} \tag{1.2}$$

여기서 함수 $f: R^{n_1+n_2+\cdots+n_N} \to R$ 는 $x = (x_1, x_2, \cdots, x_N) \in R^{n_1+n_2+\cdots+n_N}$ 에 대해 콘벡스(closed proper convex)이다. f는 분리가능(separable)일 필요가 없다. $a \in R^m$이고, 각각의 $i = 1, 2, \cdots, N$ 에 대해서 $b_i \in R^{m_i}$, $B_i \in R^{m_i \times n_i}$ 및 $A_i \in R^{m \times n_i}$ 이다.

(LCOP)의 목적함수 f가 선형인 경우는 Dantzig & Wolfe[2,4]의 분해법 및 Rosen[2,3,5]의 분할법등이 잘 알려져 있으며 함수 f가 덧셈에 대해 분리 가능한 경우에는 Robinson[8,9]의 번들-분해법이 매우 효과적이다. 그러나 (LCOP)와 같이 특별한 제약조건을 가지면서 목적함수가 분리되지 않는 경우의 대규모 콘벡스 최적화 문제에 대한 효율적인 해법은 잘 알려져 있지 않다.

목적함수가 미분 가능할 때 축차적으로 목적함수를 2차함수로 근사화 시켜 번들-분해법을 적용한 비분리 대규모 최적화 문제에 대한 SQA 알고리즘의 구체적 설명, 구현, 수렴성에 대해서는 [1]을 참고하도록 하며, 본 연구에서는 특히 (1.2)에서 제시한 제약식 형태를 갖고 (1.1)의 목적함수가 다양한 경우 뿐만아니라 비분리 목적함수가 대규모인 형태에 대해서 [1]에서 구현한 알고리즘 SQA를 이용하여 ANCI C로 작성한 프로그램을 SUN Sparc II에서 적용한 결과인 수렴성, 반복회수 및 계산시간등을 보인다.

2. 번들-분해법을 이용한 알고리즘

번들-분해법[8,9]은 다음 등식의 제약조건하에서 분리가능한 콘벡스 최적화 문제의 효율적인 해법이다.

$$\begin{aligned} &\text{Minimize} && \sum_{i=1}^{n} f_i(x_i) \\ &\text{Subject to} && \sum_{i=1}^{n} A_i x_i = a \end{aligned} \tag{2.1}$$

여기서 함수 $f_i: R^{n_i} \to R$ 는 x_i에 대해 콘벡스(closed proper convex)이다. A_i는

$R^{n_i} \to R^m$의 선형변환이고 $a \in R^m$이다.

문제 (2.1)의 쌍대문제[4]는

$$\text{Max} \quad g(y) = \langle a, y \rangle - \sum_{i=1}^{n} f_i^*(A_i^T y) \tag{2.2}$$

이 된다. 여기서 $f_i^*(\cdot)$는 conjugate function을 의미한다[4].

쌍대문제 (2.2)의 해를 구하기 위해서 Lemarechal의 Bundle method[8]를 적용한것이 번들- 분해법이다. 번들-분해법에서는 다음의 분할된 부문제를 풀어서 쌍대문제 (2.3)의 해를 구하게 된다.

$$\text{Minimize} \ \{ f_i(x_i) - \langle A_i^T y, x_i \rangle \} \tag{2.3}$$

번들-분해법을 이용한 SQA 알고리즘에서 (LCOP)에 Rockafellar duality[9]를 적용하면 쌍대문제는 다음과 같이 된다.

$$\text{Maximize} \ \{ g(y) \mid y \geq 0 \} \tag{2.4}$$

여기서

$$\begin{aligned} g(y) = \ & \text{Min} \ \ f(x_1, x_2, \cdots, x_N) + \\ & \qquad \langle \sum_{i=1}^{N} A_i x_i - a, y \rangle \\ & \text{s/t} \quad B_i x_i = b_i, \ x_i \geq 0, \\ & \qquad i = 1, 2, \cdots, N \\ = & - \langle a, y \rangle + \\ & \left\{ \begin{array}{l} \text{Min} \ \ f(x_1, x_2, \cdots, x_N) + \\ \qquad \langle \sum_{i=1}^{N} A_i x_i, y \rangle \\ \\ \text{s/t} \quad B_i x_i = b_i, \ x_i \geq 0, \\ \qquad i = 1, 2, \cdots, N \end{array} \right. \end{aligned} \tag{2.5}$$

함수 $g(y)$에서 $f(x_1, x_2, \cdots, x_N)$ 가 분리가능하지 않기 때문에 번들-분해법에서와 같이 분할(decomposed)된 부문제로는 해결되지 않는다.

SQA법은 함수 $f(x_1, x_2, \cdots, x_N)$ 를 분리가능한 2차함수로 근사화시켜 번들-분해법을 적용하여 보다 개선된 해를 얻어내는 방법이다. 즉 현재해 $x^k = (x_1^k, x_2^k, \cdots, x_N^k)$에서 함수 f를 2차식으로 근사화시키고, Hessian 행렬을 block-diagonal 행렬로 근사화시킨다. 즉

$$\begin{aligned} f(x) \approx \ & \frac{1}{2} \langle \nabla^2 f(x^k)(x - x^k), x - x^k \rangle + \\ & \langle \nabla f(x^k), x - x^k \rangle + f(x^k) \\ \approx \ & \frac{1}{2} \sum_{i=1}^{N} \langle M_i(x_i - x_i^k), x_i - x_i^k \rangle + \\ & \sum_{i=1}^{N} \langle q_i, x_i - x_i^k \rangle + f(x^k) \\ = \ & \frac{1}{2} \sum_{i=1}^{N} \langle M_i x_i, x_i \rangle + \\ & \sum_{i=1}^{N} \langle q_i - M_i x_i^k, x_i \rangle + \\ & \sum_{i=1}^{N} \xi_i + f(x^k) \end{aligned} \tag{2.6}$$

여기서

$$M_i = \frac{\partial^2 f(x^k)}{\partial x_i^2}, \quad q_i = \frac{\partial f(x^k)}{\partial x_i},$$

$$\xi_i = \frac{1}{2} \sum_{i=1}^{N} \langle M_i x_i^k, x_i^k \rangle - \sum_{i=1}^{N} \langle q_i, x_i^k \rangle \tag{2.7}$$

이다.

위에서와 같은 2차 함수로의 근사화 과정을 거쳐 적용하게 되는 전체 SQA알고리즘의 흐름은 다음과 같다.

[단계 0] 시작점 $x^o = (x_1^o, x_2^o, \cdots, x_N^o)$ 을 선택하고 $k = 0$ 으로 놓는다.

[단계 1] x^k에서 함수 $f(x)$를 분리가능한 2차함수로 근사화시킨다.

[단계 2] 번들-분해법을 적용하여 개선점 x^{k+1}을 계산한다. 이때 x^{k+1}이 수렴성을 만족하면 멈추고 그렇지 않으면 $k+1$을 k로 놓고 [단계 1]로 간다.

이상에서 제시한 SQA알고리즘에 대한 구체적인 설명, 구현 및 수렴성(국소 수렴성과 전체 수렴성)등은 [1]을 참고할 수 있으며 여기서는 [1]에서 제시한 SQA알고리즘에 대한 수치 적용결과를 중점적으로 제시한다.

3. 수치 적용결과

앞에서 구현한 번들-분해법을 이용한 SQA 알고리즘[1]을 ANSI C 프로그램으로 작성하여 Sun sparc II에서 다음의 여러 형태 문제를 적용한 결과와 이에 대한 GAMS/MINOS와 결과의 비교를 소개한다.

먼저 작은 규모의 비분리 2차 계획문제에 대해서 SQA 알고리즘을 적용한 문제 형태는 [문제 1]과 같이 목적함수의 매트릭스가 Positive definite인 경우로서 제약식의 형태는 [문제 1]에서 제시한 형태를 이용하되 추가되는 비분리 목적함수를 다양하게 하였다.

<표 3.1>은 2차계획문제의 분리 제약조건식은 등식(또는 부등식)이고 결합조건식이 부등식(또는 등식)일 때, 추가되는 비분리 목적함수식에 따른 SQA 알고리즘 적용결과와 GAMS/MINOS의 최적해를 비교・요약하였다. 다음은 목적함수의 형태가 다양한 경우에 대해서 적용한 것으로 [문제 2]는 목적함수가 비분리 4차 다항식을 갖는 수리 계획문제로서 4차의 목적함수를 2차식으로 근사화하여 Hessian 행렬과 Gradient 행렬을 구하여 SQA 알고리즘에 적용하였다. [문제 3]은 비분리 지수함수을 갖는 수리계획문제이고, [문제 4]는 비분리 분수식을 갖는 수리계획문제로서 이들 목적함수를 [문제 2]의 경우와 마찬가지로 2차함수로 근사화하여 SQA 알고리즘에 적용한다. <표 3.2>에는 [문제 2], [문제 3] 및 [문제 4]에 대한 SQA 알고리즘 적용결과와 GAMS/MINOS의 결과가 요약되어 있다. 특히, [문제 2]의 경우에 목적함수를 근사화하면 Hessian 행렬이 Positive Semi-definite가 되지않아 분리제약식이 부등호이고 결합 제약식이 등호인 경우에 대해서는 결과를 얻지 못한다.

대규모 비분리 2차계획문제에 대한 SQA 알고리즘의 적용은 [문제 5]의 형태로서 각 목적 함수와 블록-삼각 구조의 제약식 및 우변의 계수들은 ANSI C의 랜덤함수로 부터 얻은 결과를 이용하였으며 이때 rand()는 [0,1]사이의 값을 발생시켜주며 일양분포를 따르는 랜덤함수이다. 목적함수의 계수는 다음과 같은 절차에 따라 얻어 진다.

(1) Upper-Triangular Matrix W를 만든다.

$$\begin{cases} W_{ii} = 10 \times rand() \\ W_{ij} = 10 \times (rand() - 1.0) \quad (i < j) \end{cases} \tag{3.1}$$

(2) $M=W^TW$을 계산하면 M는 Positive Semi-definite가 되며 만일 W가 nonsingular 행렬이면 M은 Positive Definite가 된다.

블록-삼각 구조의 제약조건식의 계수는 다음 절차에 따라서 얻어진다.

(1) 분리제약식의 계수 :

$B_ix_i=b_i$에서 $B_i=(B_{i1},B_{i2},\cdots,B_{in_i})$, $x_i=(x_{i1},x_{i2},\cdots,x_{in_i})$이므로 각 B_{in_i}값은 랜덤함수를 이용하여 얻고 이에따라 $\sum_{k=1}^{n_i} B_{ik}x_{ik}$ 값을 계산하여 우변 b_i의 값을 얻게된다.

$$\begin{cases} B_i=rand()\times 10 & (i=1,\cdots,N) \\ b_i=\sum_{k=1}^{n_i} B_{ik}x_{ik} & (i=1,\cdots,N) \end{cases} \tag{3.2}$$

(2) 결합제약식의 계수 :

먼저 A_i의 계수들을 얻은 후에 $\sum_{i=1}^{N} A_ix_i$을 계산하여 부등식을 만족하는 임의의 값을 우변에 더하여 결합 제약식의 계수들을 얻는다.

$$\begin{cases} (A_i)_{jk}=rand()\times 10 & k=1,\cdots,(n_i\times N) \\ a_j=\sum_{t=1}^{N} A_tx_t + rand()\times 10 \end{cases} \tag{3.3}$$

(단, $B_t=(B_{t1},B_{t2},\cdots,B_{tn_t})$, $x_t=(x_{t1},x_{t2},\cdots,x_{tn_t})$)

(3.1), (3.2), (3.3)에의해 얻은 다양한 크기의 분리 제약식과 결합제약식을 가지는 대규모 비분리 2차계획문제는 <표 3.3>에 나타나 있다.

<표 3.3>에서 하부문제 수는 삼각-블록 구조 제약식에서 블록의 수(분리 제약식 수)에 해당하며 하부문제의 크기는 각 블록의 크기에 해당한다. 또한 밀도는 전체문제의 매트릭스 크기를 각 블록들(분리 제약식)과 결합 제약식들에 의해 만들어지는 매트릭스 요소들의 크기로 나눈 값이다.
<표 3.3>의 문제들에 대한 SQA 알고리즘 적용결과인 수렴성, 반복회수 및 계산시간결과는 <표 3.4>에 요약되어 있으며 여기서 X-norm은 수렴성의 정도($\|x^{k+1}-x^k\|$)를 나타내며 SQA-iteration은 SQA 알고리즘의 iteration수이고 BBD-iteration은 SQA 알고리즘이 끝날때까지 적용된 번들-분해법의 내부 iteration수의 합계이다.

[문제 1] 비분리 2차계획문제

$$\text{Min} \quad \frac{1}{2}\left\langle \begin{pmatrix} 5 & -1 & 1 & 0 \\ -1 & 5 & 2 & 1 \\ 1 & 2 & 5 & 0 \\ 0 & 1 & 0 & 5 \end{pmatrix}\begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \end{pmatrix}, \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \end{pmatrix} \right\rangle + \frac{1}{2}\left\langle \begin{pmatrix} 5 & 1 & -1 & 1 \\ 1 & 5 & 2 & 1 \\ -1 & 2 & 5 & -1 \\ 1 & 1 & -1 & 5 \end{pmatrix}\begin{pmatrix} x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \end{pmatrix}, \begin{pmatrix} x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \end{pmatrix} \right\rangle$$

$$+ \quad x_{11} + 2x_{12} - 2x_{13} + x_{14} - x_{21} + 2x_{22} + 3x_{13} - 5x_{14}$$

+ { 추가되는 비분리 목적함수식 }

Subject to

$$\begin{array}{lllllllll}
x_{11} - x_{12} + 3x_{13} + 4x_{14} & & & & & & & (\geq \text{ or } =) & 5 \\
5x_{11} + 2x_{12} + 2x_{14} & & & & & & & (\geq \text{ or } =) & 14 \\
 & 3x_{21} + x_{23} + 2x_{24} & & & & & & (\geq \text{ or } =) & 6 \\
 & -2x_{21} + x_{22} + x_{23} + 3x_{24} & & & & & & (\geq \text{ or } =) & 12 \\
\end{array}$$

$$\begin{array}{ll}
x_{11} + 2x_{12} + x_{13} + 5x_{14} + 2x_{21} - x_{22} + 4x_{23} + x_{24} \; [\leq \text{ or } =] & 18 \\
3x_{11} + x_{12} + 2x_{13} - x_{14} + 3x_{21} + 2x_{22} + x_{23} + 5x_{24} \; [\leq \text{ or } =] & 32
\end{array}$$

$$x_{ij} \geq 0 \,;\; i = 1, \cdots, 2 \;\&\; j = 1, \cdots, 4.$$

[문제 2] 비분리 4차 다항식을 갖는 수리계획문제

$$\begin{aligned} \text{Min} \quad & 2x_{11}^4 + x_{12}^4 + 3x_{13}^4 + 4x_{14}^4 \\ + \; & x_{21}^4 + 3x_{22}^4 + 2x_{23}^4 + 4x_{24}^4 \\ + \; & 4x_{11}^2 x_{23}^2 + 8x_{14}^2 x_{24}^2 \end{aligned}$$

Subject to [문제 1]의 제약식

[문제 3] 비분리 지수함수식을 갖는 수리계획문제

$$\begin{aligned} \text{Min} \quad & e^{x_{11}} + e^{-3x_{12}} + e^{x_{13}} + 4e^{x_{14}} \\ + \; & e^{-x_{21}} + 7e^{3x_{22}} + e^{x_{23}} + e^{-x_{24}} \\ + \; & e^{x_{13} + x_{22}} + e^{-x_{14} + x_{21}} \end{aligned}$$

Subject to [문제 1]의 제약식

[문제 4] 비분리 분수식을 갖는 수리계획문제

$$\begin{aligned} \text{Min} \quad & \frac{x_{11} + x_{21}}{10 - x_{11} - x_{21}} + \frac{x_{12} + x_{22}}{15 - x_{12} - x_{22}} \\ + \; & \frac{x_{13} + x_{23}}{15 - x_{13} - x_{23}} + \frac{x_{14} + x_{24}}{10 - x_{14} - x_{24}} \end{aligned}$$

Subject to [문제 1]의 제약식

[문제 5] 대규모 비분리 2차계획문제

$$\text{Min} \quad \frac{1}{2}\langle Mx, x \rangle + Cx$$

subject to 식 (1.2)

여기서 $x = (x_1, x_2, \cdots, x_n) \in R^{n_1 + n_2 + \cdots + n_N}$

<표 3.1> SQA 알고리즘 적용결과: 비분리 2차계획문제

번호	추가 비분리 목적함수식	분리 제약식	결합 제약식	SQA iteraton 수	SQA OBJ	GAMS OBJ
추가식이 없는경우		>=	=	2	46.313332	46.313
		=	<=	3	63.454315	63.454
1	3*x11*x24	>=	=	4	72.776184	72.776
		=	<=	5	81.817101	81.817
2	4*x14*x21	>=	=	4	47.311111	47.311
		=	<=	5	63.454315	63.454
3	5*x14*x22	>=	=	3	49.313332	49.313
		=	<=	5	68.647865	68.648
4	7*x14*x23	>=	=	8	49.271317	49.271
		=	<=	5	63.454315	63.454
5	10*x14*x24	>=	=	4	49.271320	49.271
		=	<=	5	68.647865	68.648
6	13*x13*x22	>=	=	3	46.313332	46.313
		=	<=	3	63.454315	63.454
7	8*x13*x23	>=	=	4	49.113220	49.113
		=	<=	3	63.454315	63.454
8	15*x12*x24	>=	=	4	48.354317	48.354
		=	<=	3	65.427475	65.427
9	13*x13*x22 + 4*x14*x21	>=	=	4	47.311111	47.311
		=	<=	3	63.454315	63.454
10	4*x14*x21 + 10*x14*x24	>=	=	4	49.271313	49.271
		=	<=	3	68.647865	68.648
11	3*x11*x24 + 4*x14*x21	>=	=	4	73.602753	73.603
		=	<=	3	81.81701	81.817
12	15*x12*x24 + 8*x13*x23 4*x14*x21	>=	=	3	52.573071	52.573
		=	<=	3	65.427475	65.427
13	13*x13*x22 + 8*x13*x23 7*x14*x23	>=	=	6	55.311600	55.312
		=	<=	5	63.454315	63.454
14	15*x12*x24 + 13*x13*x22 + 8*x13*x23 + 4*x14*x21	>=	=	3	52.573063	52.573
		=	<=	3	65.427460	65.427
15	15*x12*x24 + 13*x13*x22 + 8*x13*x23 + 4*x14*x21 + 5*x14*x22	>=	=	3	52.573063	52.573
		=	<=	3	74.594131	74.594

<표 3.2> SQA 알고리즘 적용결과: 비분리 수리계획문제

문제	분리 제약식	결합 제약식	SQA iteration수	SQA 최적 목적함수값	GAMS/MINOS 목적함수값	비고
[문제 2]	=	≤	5	612.069335	612.069	
	≥	=	-	-	176.800	(주1)
[문제 3]	=	≤	4	56763.399402	56763.399	
	≥	=	7	28.04437	28.020	
[문제 4]	=	≤	4	1.118862	1.119	
	≥	=	9	1.296039	1.296	

(주1) Hessian 행렬이 positive semi-definite가 안됨

<표 3.3> 대규모 비분리 2차계획문제 형태

문제 번호	전체문제 크기	결합제약식 수	하부문제 수	하부문제 크기	밀도(%)
1	123 × 200	3	40	3 × 5	4.88
2	103 × 200	3	20	5 × 10	7.77
3	54 × 200	4	10	5 × 20	16.67
4	203 × 300	3	50	4 × 6	3.45
5	214 × 300	4	30	7 × 10	5.14
6	148 × 300	8	20	7 × 15	10.14
7	256 × 400	6	50	5 × 8	3.71
8	286 × 400	6	40	7 × 10	4.55
9	148 × 400	8	20	7 × 20	10.14

<표 3.4> 대규모 비분리 2차계획문제에 대한 SQA 알고리즘 적용 결과

문제번호	X - norm	BBD-iteration	SQA-iteration	평균 BBD회수	Run-time(sec)
1	0.000021	9	4	2.25	10
2	0.000018	43	3	14.33	51
3	0.000046	33	3	11.00	60
4	0.000069	42	4	10.50	45
5	0.000043	31	4	7.75	68
6	0.000056	29	4	7.25	65
7	0.000008	36	5	7.60	77
8	0.000012	40	4	10.00	110
9	0.000081	33	5	6.60	130

주) 평균 BBD 회수 = BBD iteration ÷ SQA iteration 임

4. 결론

본 논문에서는 SQA 알고리즘[1]을 간략하게 정리하였고 다양한 형태와 크기의 문제에 대해 SQA 알고리즘을 적용한 결과를 소개하였다. 적용결과에 의하면 SQA 알고리즘은 목적함수가 Strongly Convex인 문제(QP문제일때는 2차식의 행렬이 Positive Definite인 경우)는 효율적으로 해를 구할 수 있음을 보여 주었다. 그러나 목적함수의 형태가 Strictly Convex인 문제(QP문제에서는 목적함수의 2차식 행렬이 Positive Semidefinite인 경우)는 부문제(Subproblem)인 QP를 풀기 위해서 Lemke' Algorithm[3]을 이용하기 때문에 해를 구하지 못하는 경우가 발생한다. 따라서 이에 대한 추후연구가 요구된다.

5. 참고문헌

[1] 박구현,신용식 "대규모 비분리 콘벡스 최적화 : 미분 가능한 경우", 경영과학회지 제출, 1995.

[2] Ahuja,R.K., Magnanti,T.L. & Orlin,J.B., Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, 1993.

[3] Bazaraa,M. & Shetty,C.M., Nonlinear Programming: Theory and Algorithms, John Wiley and Sons Inc., 1979.

[4] Lasdon,L.S., Optimization Theory for Large Systems, Macmillan, 1970.

[5] Lemarechal,C., "Bundle Methods in Nonsmooth Optimization", Nonsmooth Optimization, Ed. by C. Lemarechal & R. Mifflin, IIASA V3, Pergamon, 1978.

[6] Medhi,D., "Bundle-based Decomposition for Large-scale convex Optimization: Error estimate and Application to Block-angular Linear Programs", Mathematical Programming, Vol 66, 1994.

[7] Rockafellar,R.T., Conjugate Duality and Optimization, SIAM, Philadelphia, 1974.

[8] Robinson,S.M. "Bundle-based Decomposition: Description and Preliminary Results", in A. Prekopa, J. Szelezsan, and B. Strazicky, eds., System Modelling and Optimization, Springer-Verlag, 1986.

[9] Robinson,S.M. "Bundle-based Decomposition: Conditions for Convergence", Annales de l'Institute Henri Poincare: Analyse Non Lineaire 6, 1989.

[10] Rockafellar,R.T., Convex Analysis, Preceton Press, 1970.

[11] Rosen,J.B., "Primal Partition Programming for Block Diagonal Matrices", Numerische Mathematik, Vol 6, 1964.