經營科學
第30卷 第3號
2013年 11月, pp.55-70
http://dx.doi.org/10.7737/KMSR.2013.30.3.055

평균/VaR 최적화 모형에 의한 전환사채 주식전환 비중 결정*

박 구 현**

Determination Conversion Weight of Convertible Bonds Using Mean/Value-at-Risk Optimization Models

Koohyun Park**

■ Abstract ■

In this study we suggested two optimization models to determine conversion weight of convertible bonds. The problem of this study is same as that of Park and Shim [1]. But this study used Value-at-Risk (VaR) for risk measurement instead of CVaR, Conditional-Value-at-Risk. In comparison with conventional Markowitz portfolio models, which use the variance of return, our models used VaR.

In 1996, Basel Committee on Banking Supervision recommended VaR for portfolio risk measurement. But there are difficulties in solving optimization models including VaR. Benati and Rizzi [5] proved NP-hardness of general portfolio optimization problems including VaR. We adopted their approach. But we developed efficient algorithms with time complexity $O(n\log n)$ or less for our models. We applied examples of our models to the convertible bond issued by a semiconductor company Hynix.

Keywords : Value-at-Risk(VaR), Markowitz Model, Portfolio Optimization, Convertible Bond

논문접수일 : 2013년 07월 23일　　논문게재확정일 : 2013년 08월 19일
논문수정일 : 2013년 08월 14일
* 이 논문은 2011년도 홍익대학교 학술연구진흥비에 의하여 지원되었음.
** 홍익대학교 산업공학과, khpark@hongik.ac.kr

1. 서 론

본 연구에서 우리는 전환사채(Convertible Bond)의 주식 전환량을 결정하는 최적화 모형을 제시하고, 효율적 해법을 제시하려 한다. 전환사채는 전환사채 보유자가 해당 기업의 주식가격이 전환가격 이상이 되면 주식으로 전환할 수 있는 권리가 주어진 채권이다. 전환사채를 보유한 투자자는 채권 만기일까지 채권을 보유하고 있다가 만기 지불액을 받을 수도 있고, 도중에 채권은 포기하고 주식으로 전환하여 주식으로 보유할 수도 있다. 이러한 전환사채는 주식으로 전환할 수 있는 전환가격 및 채권 1좌로 전환할 수 있는 주식수인 전환비율을 미리 정하여 발행하게 되어있다.

전환사채 관련 연구는 전환사채의 가치평가에 관한 연구가 대부분으로 전환사채가 일반 채권에 주식전환 청구권인 콜-옵션이 추가된 것이라 생각하여 Black-Scholes 및 Merton의 옵션가격 결정모형에 기초하고 있으며, 대표적인 연구로는 Ingersoll[9], Brennan and Schwartz[6], Tsiveriotis and Fernandes[14], Ayache et al.[4]가 있다. 이러한 연구들은 대부분 경계조건이 있는 확률 편미분 방정식으로 모형화하여 수치해석 방법에 의해 가치를 평가한다.

이와 같이 미국형 옵션의 가치평가 모형으로 평가한 전환사채 가치보다 주식의 가격이 크게 될 때는 모두 주식으로 전환하는 것이 좋다는 것을 의미한다. 그러나 이는 위험 중립의 관점, 즉 투자자가 시장에서 위험회피를 할 경우의 전략이다. 그러나 일반 투자자나 펀드 등의 기관투자자의 경우 이러한 위험회피 활동을 하지 않기 때문에, 주식가격이 전환가격 이상 될 때, 보유 전환사채 전량을 주식으로 전환할 것인지, 아니면 보유량 중 일부만 주식으로 전환할 것인지를 결정함으로 위험회피 투자를 하게 된다. 이와 같은 위험회피를 고려한 전환사채의 주식전환 비중을 결정하는 연구로서 박구현, 심은택[1]이 있다. 그 연구에서는 주식 전환량 결정 문제를 위험척도 CVaR(Conditional Value at Risk)이 포함된 확률적 최적화 문제로 모형화하고, Rockafellar and Uryasev[11, 12]의 시나리오 근사해법을 적용하여 선형계획문제로 전환하여 해를 구하였다. 정재우, 민대기[2]는 이를 장기 전원 포트폴리오의 평가에 활용하였다.

위험척도 CVaR은 Artzner et al.[3]가 정의하는 일관성있는 위험척도(coherent risk measure)의 4가지 조건을 모두 만족하고 있다. 즉 한 대안이 다른 대안보다 어떤 상황에서도 이익이 많다면 위험은 적어야 하며(monotonicity), 분산 투자의 위험은 개별 투자 위험의 합보다 같거나 작아야 하며(sub-additivity), 투자가 상수배 증가하면 위험도 비례해서 증가하여야 하며(positive homogeneity), 고정수익률을 갖는 투자가 포함되면 전체 투자 위험에서 고정수익률 투자금은 제외되어야 하는(translational invariance) 조건이 그것이다. CVaR은 이러한 4가지 조건을 모두 만족하여 일관성있는 위험척도이나, 본 연구에서 고려하고자 하는 위험척도인 VaR(Value at Risk)은 위의 2번째 조건인 sub-additivity 조건을 만족시키지 못한다. 따라서 앞선 연구[1]를 포함한 대부분의 많은 연구[7, 8, 10, 13]에서는 VaR 대신에 CVaR를 이용하였다. 그러나 이러한 이유 때문에 지난 17년간 금융사회에서 가장 주목 받아 온 위험 척도인 VaR에 대해 연구가 상대적으로 적었다고 볼 수 있다. 본 연구는 CVaR 대신 일관성있는 위험척도는 아니지만 VaR 위험척도를 적용하려 한다. 왜냐하면 VaR은 주요 선진국의 중앙은행 및 은행감독 당국의 대표들로 구성되는 바젤위원회에서 1996년 BIS(Bank for International Settlements)비율을 대신하여 사용하기로 결정한 위험척도이기 때문이다. VaR은 물론 은행의 위험관리를 위해 제시된 것이기는 하나 다른 금융기관에서도 하나의 위험관리 목적으로 사용할 수 있으며, 펀드를 통해 전환사채에 투자된 자산을 자산운용사가 운용한다고 하면 자산운용사는 VaR을 고려할 것이라고 생각할 수 있다.

본 연구에서는 주식 전환량 결정 문제를 투자대안이 2개인 포트폴리오 모형으로 표현하여 최적해

를 구한다. 즉 채권(전환사채)과 주식(전환되는 주식) 두 대안에 대한 가중치를 결정함으로 최적해를 제시하려 한다. 이를 위해 본 연구에서는 기존의 평균/분산 포트폴리오 모형 대신 평균/VaR 포트폴리오 모형에 대한 Benati and Rizzi[5]의 연구를 적용한다. 위험척도로서 VaR을 분산대신 적용하면 최적화 문제가 복잡해진다. Benati and Rizzi는 일반적인 평균/VaR 포트폴리오 모형이 NP-hard 문제라는 것을 증명하였으며, 정수계획문제로 모형화하여 상용 해찾기(CPLEX) 도구를 적용하여 해를 구하였다. 본 연구에서는 그들이 제시한 일반적인 모형을 전환사채 주식전환 비중을 결정하는 문제에 적용한다. 또한 본 연구의 문제 특성을 고려하여 효율적인 해법을 제시한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 VaR 정의와 전환사채 주식전환 비중 결정 문제를 VaR이 포함된 2종류의 포트폴리오 문제로 모형화하고, 이를 선형 정수계획문제로 표현한다. 제 3장에서는 제 2장에서 제시한 선형 정수계획문제의 효율적인 해법을 제시한다. 해법은 문제별로 모수에 따라 3가지 경우로 나누어 각각의 알고리즘을 제시한다. 제시한 해법의 실제 적용 예는 제 4장에서 보인다. 마지막으로 제 5장은 결론이다.

2. 전환사채 주식전환 비중 결정 모형

전환사채 주식전환 비중 결정을 위한 최적화 모형을 전개하기 위해 몇 가지 가정을 한다. 첫째 전환사채는 만기까지 이자지급이 없는 무이표채라고 가정한다. 둘째, 전환사채를 주식으로 전환한 이후에는 배당이 없다고 가정한다. 셋째, 이자율을 고려하지 않는다. 여기서는 모형 및 알고리즘 표현의 간편성을 위함이다. 전환사채 주식전환 비중 결정을 위한 CVaR의 연구[제1장, 제5장]에서처럼 이들 가정에 대해 이표이자, 주식배당 및 이자율을 고려하여 모형을 일반화 할 수 있다. 표현은 복잡해지지만 일반 모형으로 확장될 수 있다.

최적화 모형을 위해 다음 표현들을 정의한다.

b : 채권 1좌의 만기 지불액
m : 채권 1좌당 전환되는 주식의 수(전환비율)
i : 과거 데이터의 시점 $i(i=1,2,\cdots,n)$
s_i : 시점 i에서 주식의 가격$(i=1,2,\cdots,n)$
y_i : 보조변수, 시점 i에서 해당 주식의 가격이 VaR에 영향을 미치는 지의 여부를 나타냄
$y_i=0$ 이면 VaR에 영향을 주고, $y_i=1$이면 VaR에 영향을 주지 않는다$(i=1,2,\cdots,n)$.
λ : 의사결정변수, 채권 보유량 중 주식전환 비중 $(0 \le \lambda \le 1)$
$\lfloor x \rfloor$: x보다 크지 않은 정수를 의미한다.

X를 만기 시점에서 투자 수익을 나타내는 확률변수라고 하자. 그리고 $F_X(\cdot)$를 X의 누적분포함수라고 할 때, 임계값 α에 대한 Value-at-Risk, $VaR_\alpha(X)$는 누적분포함수의 α-변위값으로 아래와 같이 정의된다.

정의 1 : 임계값이 $\alpha \in (0,1)$일 때, X의 α에 대한 VaR은 아래와 같이 정의된다.

$$VaR_\alpha(X) = \inf\{x \mid F_X(x) \ge \alpha\}$$

만약 누적분포함수 $F_X(\cdot)$가 연속이고 엄격한 단조증가(strictly increasing)이면 $VaR_\alpha(X) = F_X^{-1}(\alpha)$이 된다.

이제 전환사채의 주식전환 비중을 결정하기 위한 최적화 모형을 제시한다. 이를 위해 R_s을 전환사채 1좌를 주식으로 전환하였을 때 만기 시점에 주식으로 인한 수익을 나타내고, R_b는 주식전환 없이 채권으로 보유하였을 때 만기 시점의 수익을 나타낸다고 하자. 그러면 이들 두 투자대안에 대한 포트폴리오 최적화 모형을 이용하여 전환사채의 주식전환 비중을 결정할 수 있다. 모형에서 X는 채권 1좌의 수익을 나타내는 확률변수이지만, 모형으로부터 얻어지는 주식전환 비중은 소유 채권 수

에 관계없이 전체 채권 수에 대한 주식전환 비중으로 해석할 수 있다.

전통적인 평균-분산 포트폴리오 최적화 모형에서 분산 대신에 본 연구에서는 VaR을 적용한다. 이를 위해 두 모수로서 임계 확률값 α와 이에 대응되는 수익을 r^{α}라고 놓는다. 즉 $F_X(r^{\alpha}) = \Pr[X \le r^{\alpha}] > \alpha$ 된다면, 이러한 투자 X는 임계값 α를 넘어서므로 허용하지 않겠다는 것이다. 따라서 임계값 α와 대응되는 수익 r^{α}를 허용하는 투자 포트폴리오는 아래의 최적화 모형으로 표현할 수 있다.

(모형 1) Maximize $E[X]$ (1)

$$\text{Subject to } \Pr[X \le r^{\alpha}] \le \alpha^{VaR} \quad (2)$$
$$X = \lambda R_s + (1-\lambda) R_b \quad (3)$$
$$0 \le \lambda \le 1 \quad (4)$$

모형 1이 위험 조건 하에서 기대수익을 최대화하는 것이라면 모형 2는 수익 조건 하에서 위험을 최소화하는 것이다.

(모형 2) Minimize α^{VaR} (5)

$$\text{Subject to } E[X] \ge r_{req} \quad (6)$$
$$\Pr[X \le r^{\alpha}] \le \alpha^{VaR} \quad (7)$$
$$X = \lambda R_s + (1-\lambda) R_b \quad (8)$$
$$0 \le \lambda \le 1 \quad (9)$$

모형 1에서 α^{VaR}는 VaR를 고려하는 모형에서 임계 확률값 α를 의미하며, 모형 2에서 r_{req}는 최소요구 기대수익을 의미한다. 이제 모형 1을 과거 주식가격 데이터를 반영하여 전환사채 주식전환 비중(λ)을 결정하는 문제로 표현하면 아래의 문제(P1)이 된다.

(P1) Maximize $(1-\lambda)b + \frac{\lambda}{n}\sum_{i=1}^{n} ms_i$ (10)

$$\text{Subject to } (1-\lambda)b + \lambda ms_i \ge \beta_{VaR}\, b y_i,$$
$$i = 1, 2, \cdots, n \quad (11)$$
$$\frac{1}{n}\sum_{i=1}^{n}(1-y_i) \le \alpha^{VaR} \quad (12)$$
$$y_i = 0 \text{ 또는 } 1,\ i = 1, 2, \cdots, n \quad (13)$$
$$0 \le \lambda \le 1 \quad (14)$$

모형 1과 문제(P1)을 비교하면, 채권 1좌를 주식으로 전환하여 얻게 되는 기대수익은 $R_s = \frac{1}{n}\sum_{i=1}^{n} ms_i$ 이 되며, 채권을 주식으로 전환하지 않고 그대로 보유하여 얻게 되는 수익은 $R_b = b$가 된다. 따라서 모형 1의 식 (2)~식 (3)은 문제(P1)에서는 식 (11)~식 (13)으로 표현된다. 여기서 식 (2)의 r^{α}는 식 (11)에서 $\beta_{VaR} b$로 표현하였다. 식 (11)에서 수익을 나타내는 좌변이 원하는 수준인 우변 값 r^{α} ($= \beta_{VaR} b$) 이상이 되기를 바라는데 그와 같은 경우는 $y_i = 1$이 되고, 그렇지 못한 경우는 $y_i = 0$이 된다. 식 (12)는 $y_i = 0$이 될 확률이 α^{VaR} 이하이어야 한다는 조건이다.

모형 2 역시 과거 주식가격 데이터를 반영하여 전환사채 주식전환 비중(λ)을 결정하는 문제로 표현하면 문제(P2)가 된다. 모형 2의 식 (6)에서 요구하는 기대수익 r_{req}은 식 (16)의 $\beta^* b$로 표현하였다.

(P2) Minimize α^{VaR} (15)

$$\text{Subject to } (1-\lambda)b + \frac{\lambda}{n}\sum_{i=1}^{n} ms_i \ge \beta^* b \quad (16)$$
$$(1-\lambda)b + \lambda ms_i \ge \beta_{VaR}\, b y_i,$$
$$i = 1, 2, \cdots, n \quad (17)$$
$$\frac{1}{n}\sum_{i=1}^{n}(1-y_i) \le \alpha^{VaR} \quad (18)$$
$$y_i = 0 \text{ 또는 } 1,\ i = 1, 2, \cdots, n \quad (19)$$
$$0 \le \lambda \le 1 \quad (20)$$

3. 효율적 해법

본 장에서는 수익 최대화 문제인 문제(P1)에 대

해 모수 β_{VaR}에 따라 3개의 알고리즘을 제안하고, 위험 최소화 문제인 문제(P2)에 대해서도 모수 β^*에 따라 역시 3개의 알고리즘을 제안한다. β_{VaR} 및 β^*는 모두 채권의 만기지불액 b의 배수를 나타내는 모수들이다.

문제(P1)은 기대수익이 $\beta_{VaR}b$ 이하가 될 위험확률이 α^{VaR} 이하가 되어야 하는 조건하에서 기대수익을 최대화하는 문제이고, 문제(P2) 기대수익은 $\beta^* b$ 이상 되어야 한다는 조건하에서 기대수익이 $\beta_{VaR}b$ 이하가 될 위험(α^{VaR})을 최소화하는 문제이다. 문제(P1)은 $\beta_{VaR}<1$, $\beta_{VaR}=1$, $\beta_{VaR}>1$에 따라 알고리즘이 각각 제안되며, 문제(P2)에 대해서도 $\beta^*<1$, $\beta^*=1$, $\beta^*>1$에 따라 3개의 알고리즘이 제안된다. 이를 정리하면 <표 1>과 같다.

〈표 1〉 문제, 알고리즘 및 모수조건

문제	모수 조건	알고리즘
문제(P1)	$\beta_{VaR}<1$	알고리즘 I
	$\beta_{VaR}=1$	알고리즘 II
	$\beta_{VaR}>1$	알고리즘 III
문제(P2)	$\beta^*<1$	알고리즘 IV
	$\beta^*=1$	알고리즘 V
	$\beta^*>1$	알고리즘 VI

3.1 수익 최대화 문제(P1) 해법

(i) $\beta_{VaR}<1$ 경우

문제(P1)의 목적함수는 $(1-\lambda)b+\frac{\lambda}{n}\sum_{i=1}^{n} ms_i = b+\lambda\left\{\frac{1}{n}\sum_{i=1}^{n}(ms_i)-b\right\}$이므로 $\frac{1}{n}\sum_{i=1}^{n}ms_i>b$이면 λ를 최대화하고, $\frac{1}{n}\sum_{i=1}^{n}ms_i<b$이면 λ를 최소화한다. $\frac{1}{n}\sum_{i=1}^{n}ms_i=b$이면 어떤 $\lambda(0\le\lambda\le 1)$에 대해서도 목적함수 값은 b로 일정하다.

알고리즘 I($\beta_{VaR}<1$)

(1) $\frac{1}{n}\sum_{i=1}^{n}ms_i\le b$이면, $\lambda^*=0$으로 놓고 멈춘다. 그렇지 않으면 다음 단계로 간다.

(2) $\left\{\frac{(1-\beta_{VaR})b}{b-ms_i}\mid ms_i<\beta_{VaR}b,\ i=1,\cdots,n\right\}$를 내림차순으로 아래와 같이 정렬한다.

$$w_1\ge w_2\ge\cdots\ge w_{j^*}\ge w_{j^*+1}\ge\cdots\ge w_k$$

여기서 $k=\sum_{i=1}^{n}I_{\{ms_i<\beta_{VaR}b\}}$이고, $j^*=k-\lfloor n\alpha^{VaR}\rfloor$이다. $\lambda^*=\min\left[1,\ w_{j^*}\right]$로 놓는다. □

알고리즘 단계 (2)에서 $I_{\{ms_i<\beta_{VaR}b\}}$는 지시함수로서 다음과 같은 값을 갖는다.

$$I_{\{ms_i<\beta_{VaR}b\}}=\begin{cases}1,\ ms_i<\beta_{VaR}b\text{이면}\\ 0,\ \text{그렇지않으면}\end{cases}$$

정리 1 : $\beta_{VaR}<1$ 경우, 문제(P1)은 가능해를 가지며, 알고리즘 I에서 구한 λ^*가 최적해(λ)가 된다.

(증명) 부록 참조 □

(ii) $\beta_{VaR}=1$ 경우

이 경우도 목적함수는 $b+\lambda\left\{\frac{1}{n}\sum_{i=1}^{n}(ms_i)-b\right\}$이므로, $\frac{1}{n}\sum_{i=1}^{n}ms_i$가 b보다 큰 경우, 작은 경우, 같은 경우에 따라 λ를 최대화하거나, λ를 최소화하거나, λ에 무관하게 동일한 값을 갖게 된다.

알고리즘 II($\beta_{VaR}=1$)

(1) $\frac{1}{n}\sum_{i=1}^{n}ms_i\le b$ 이면, $\lambda^*=0$으로 놓고 멈춘다.

그렇지 않으면 다음 단계로 간다.

(2) $k=\sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR} b\}}$ 로 놓는다.

(3) $k \le \lfloor n\alpha^{VaR} \rfloor$ 이면, $\lambda^* = 1$ 로 놓고, 그렇지 않으면 $\lambda^* = 0$ 로 놓는다. □

정리 2 : $\beta_{VaR} = 1$ 경우, 문제(P1)은 가능해를 가지며, 알고리즘 II에서 구한 λ^*가 최적해(λ)가 된다.

(증명) 부록 참조 □

(iii) $\beta_{VaR} > 1$ 경우

이 경우도 목적함수는 $b+\lambda\left\{\frac{1}{n}\sum_{i=1}^{n}(ms_i)-b\right\}$이므로, $\frac{1}{n}\sum_{i=1}^{n} ms_i$ 가 b보다 큰 경우, 작은 경우, 같은 경우에 따라 λ를 최대화하거나, λ를 최소화하거나, λ에 무관하게 동일한 값을 갖게 된다.

알고리즘 III($\beta_{VaR} > 1$)

(1) $k=\sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR} b\}}$ 로 놓는다.

(2) $\left\{\frac{(\beta_{VaR}-1)b}{ms_i - b} | ms_i \ge \beta_{VaR} b,\ i=1, \cdots, n\right\}$ 를 1을 포함하여 내림차순으로 아래와 같이 정렬한다. 단 $1 > w_1$ 이면, $j^* = 0$이다.

$$w_1 \ge w_2 \ge \cdots \ge w_{j^*} > 1 \ge w_{j^*+1} \ge \cdots \ge w_{n-k}$$

(3) $\delta = \lfloor n\alpha^{VaR} \rfloor - k - j^*$ 라고 놓는다. 만일 $\delta < 0$ 이면, 멈춘다. 문제(P1)은 가능해가 존재하지 않는다. 그렇지 않으면 다음 단계로 간다.

(4) $\frac{1}{n}\sum_{i=1}^{n} ms_i \ge b$ 이면, $\lambda^* = 1$ 로 놓고, 그렇지 않으면 $\lambda^* = \min\left[1, w_{j^*+\delta}\right]$ 로 놓는다. □

정리 3 : $\beta_{VaR} > 1$ 경우 알고리즘 III을 적용하였을 때, 만약 $\delta < 0$ 이면 문제(P1)은 가능해가 존재하지 않는다. $\delta \ge 0$ 이면 알고리즘 III가 제공하는 λ^*가 문제(P1)의 최적해(λ)가 된다.

(증명) 부록 참조 □

따름정리 1 : 문제(P1)에 대한 알고리즘 I, III의 시간 복잡도는 모두 $O(n\log n)$ 이고, 알고리즘 II의 시간 복잡도는 $O(n)$ 이다.

(증명) 알고리즘 I 및 알고리즘 III의 단계별 계산량을 구하면, 이들 알고리즘의 시간 복잡도는 정렬 단계에 의해 결정된다. 따라서 두 알고리즘의 시간 복잡도는 정렬 알고리즘의 시간 복잡도인 $O(n\log n)$ 이 된다. 알고리즘 II의 단계 (1) 및 단계 (2)의 계산량은 최대 $O(n)$ 이며, 따라서 이 알고리즘의 시간 복잡도는 $O(n)$ 이다. □

3.2 위험 최소화 문제(P2) 해법

(i) $\beta^* < 1$ 인 경우

$\frac{1}{n}\sum_{i=1}^{n} ms_i \ge \beta^* b$라고 가정하자. 그러면 $b > \beta^* b$ 이므로 모든 $\lambda (0 \le \lambda \le 1)$에 대해 제약식 (16)은 항상 성립한다. 그러나 만일 $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$이라면,

$$\lambda \le \lambda_{\max} \equiv \frac{(1-\beta^*)b}{b - \frac{1}{n}\sum_{i=1}^{n} ms_i}$$ 이 되어야 제약식 (16)이 성립한다. 이 경우 $b - \frac{1}{n}\sum_{i=1}^{n} ms_i > b - \beta^* b$이므로 $\lambda_{\max} < 1$ 이 된다.

알고리즘 IV ($\beta^* < 1$)

(1) $\beta_{VaR} \le 1$ 이면 $\lambda^* = 0$, $\alpha^* = 0$ 으로 놓고 멈춘

다. $\beta_{VaR} > 1$ 이면, 두 경우로 나누어 $\frac{1}{n}\sum_{i=1}^{n} ms_i \geq \beta^* b$이면 단계 (2)로 가고, $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$ 이면 단계 (3)로 간다.

(2) $k = \sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR} b\}}$ 로 놓는다. $\lambda^* = 1$ 및 $\alpha^* = \frac{n-k}{n}$ 로 놓고 멈춘다.

(3) $k = \sum_{i=1}^{n} I_{\{ms_i \geq \beta_{VaR} b\}}$ 로 놓는다.

(4) $\lambda^* = \lambda_{\max}$ 로 놓고, $\left\{ \frac{(\beta_{VaR} - 1)b}{ms_i - b} | ms_i \geq \beta_{VaR}\, b,\ i = 1, \cdots, n \right\}$를 $\lambda_{\max}$ 를 포함하여 내림차순으로 아래와 같이 정렬한다. 단 $\lambda_{\max} > w_1$ 이면, $j^* = 0$ 이다.

$$w_1 \geq w_2 \geq \cdots \geq w_{j^*} > \lambda_{\max} \geq w_{j^*+1} \geq \cdots \geq w_k$$

$\alpha^* = \frac{k - j^*}{n}$ 로 놓고 멈춘다. □

정리 4 : $\beta^* < 1$ 경우를 가정한다. 문제(P2)는 가능해가 존재하며 알고리즘 IV에서 구한 λ^*와 α^*가 문제(P2)의 최적해(λ, α^{VaR})가 된다.

(증명) 부록 참조 □

(ii) $\beta^* = 1$ 인 경우

$\frac{1}{n}\sum_{i=1}^{n} ms_i \geq b$라고 가정하자. 그러면 모든 $\lambda(0 \leq \lambda \leq 1)$에 대해 제약식 (16)은 항상 성립한다. 그러나 만일 $\frac{1}{n}\sum_{i=1}^{n} ms_i < b$이면 $\lambda = 0$ 인 경우만 제약식 (16)이 성립한다.

알고리즘 V($\beta^* = 1$)

(1) $\beta_{VaR} \leq 1$이면 $\lambda^* = 0$, $\alpha^* = 0$ 으로 놓고 멈춘다. $\beta_{VaR} > 1$이면, 두 경우로 나누어 $\frac{1}{n}\sum_{i=1}^{n} ms_i \geq b$이면 단계 (2)로 가고, $\frac{1}{n}\sum_{i=1}^{n} ms_i < b$이면 단계 (3)로 간다.

(2) $k = \sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR} b\}}$ 로 놓는다. $\lambda^* = 1$ 및 $\alpha^* = \frac{n-k}{n}$ 로 놓고 멈춘다.

(3) $\lambda^* = 0$ 및 $\alpha^* = 1$로 놓는다. □

정리 5 : $\beta^* = 1$ 경우를 가정한다. 문제(P2)는 가능해가 존재하며 알고리즘 V에서 구한 λ^*와 α^*가 문제(P2)의 최적해(λ, α^{VaR})가 된다.

(증명) 부록 참조 □

(iii) $\beta^* > 1$ 인 경우

만약 $\frac{1}{n}\sum_{i=1}^{n} ms_i \geq \beta^* b$이면, 제약식 (16)이 성립되기 위해서 $\lambda \geq \lambda_{\min} \equiv \frac{(\beta^* - 1)b}{\frac{1}{n}\sum_{i=1}^{n}\{ms_i - b\}}$ 이어야 한다.

가정에서 $\frac{1}{n}\sum_{i=1}^{n}\{ms_i - b\} \geq (\beta^* - 1)b0$이 성립되므로 $\lambda_{\min} \leq 1$ 이 된다. 만약 $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$이면, 제약식 (16)은 성립되지 않는다. 따라서 문제(P2)는 가능해를 갖지 않는다.

알고리즘 VI ($\beta^* > 1$)

(1) $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$이면, 멈춘다. 문제(P2)는 가능해를 갖지 않는다. 그렇지 않다면, 다음 단계로 간다.

(2) $\beta_{VaR} \geq 1$인 경우는 단계 (3)으로 가고, $\beta_{VaR} < 1$인 경우는 단계 (4)로 간다.

(3) $k=\sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR} b\}}$로 놓는다. $\lambda^* = 1$ 및 $\alpha^* = \frac{n-k}{n}$로 놓고 멈춘다.

(4) $\left\{ \frac{(1-\beta_{VaR})b}{b-ms_i} | ms_i < \beta_{VaR} b,\ i=1, \cdots, n \right\}$를 $\lambda_{\min}$를 포함하여 내림차순으로 아래와 같이 정렬한다. 단 $\lambda_{\min} > w_1$이면, $j^* = 0$이다. 여기서 $k = \sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR} b\}}$이다.

$$w_1 \ge w_2 \ge \cdots \ge w_{j^*} \ge \lambda_{\min} > w_{j^*+1} \ge \cdots \ge w_k$$

$\lambda^* = w_{j^*}$ 및 $\alpha^* = \frac{k-j^*}{n}$로 놓고 멈춘다. □

정리 6 : $\beta^* > 1$ 경우를 가정한다. $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$이면 문제(P2)는 가능해를 갖지 않는다. 그렇지 않다면 알고리즘 VI에서 구한 λ^*와 α^*가 문제(P2)의 최적해(λ, α^{VaR})가 된다.

(증명) 부록 참조 □

따름정리 2 : 문제(P2)에 대한 알고리즘 IV, VI의 시간 복잡도는 모두 $O(n\log n)$이고, 알고리즘 V의 시간 복잡도는 $O(n)$이다.

(증명) 알고리즘 IV, VI의 단계별 계산량을 구하면, 이들 알고리즘의 시간 복잡도는 정렬 단계에 의해 결정된다. 따라서 두 알고리즘의 시간 복잡도는 정렬 알고리즘의 시간 복잡도인 $O(n\log n)$이 된다. 알고리즘 V의 단계 (1) 및 단계 (2)의 계산량은 최대 $O(n)$이며, 따라서 이 알고리즘의 시간 복잡도는 $O(n)$이다. □

4. 적용 예

본 연구에서 제시한 모형을 하이닉스 전환사채의 경우에 적용하고자 한다. 하이닉스 전환사채는 20XX년 9월 5일 발행하여 만기가 20YY년 9월 5일인 5년짜리 전환사채로 만기보장수익률은 5.8%이고 표면이자율은 3%이다. 본 모형에서는 무이표채를 가정하므로 표면이자율을 포함하여 수익률 5.5%로 연 4회 5년간 복리 계산하여 만기일에 원금의 133.3365%를 일시 상환한다고 가정한다. 전환가격은 23,328원으로 주식가격이 전환가격 이상일 경우, 채권 1좌(1만원)당 보통주 전환은 1주인 100%의 전환비율을 갖는다.

아래 <그림 1>은 전환사채의 해당 주식인 하이

〈그림 1〉 하이닉스 주가 그래프(주별주가, 전환가격, 발행시점, 모형적용시점)

닉스의 주별 주식가격의 변화와 전환가격을 표시하였다. 가격은 주별 종가 기준이다. 그림에서 전환사채가 발행한 시점은 193주째로 그때의 주식가격은 20,400원이었고, 262주째에 주식가격이 최초로 전환가격 이상 되었다. 따라서 주식으로 전환할 보유 채권의 비중을 결정하기 위해 제 2장, 제 3장에서 제시한 모형과 알고리즘을 적용한다. 모형 적용에서 주식가격의 과거 데이터를 1년 52주로 할 경우는 211주~262주 데이터를 이용하고, 과거 데이터를 2년(105주) 또는 3년(157주)으로 할 경우는 각각 158주~262주 데이터, 106주~262주 데이터를 사용한다. 본 연구에서는 1년 52주의 경우를 중심으로 적용결과를 소개한다.

수익 최대화 문제인 문제(P1)에 대해 모수 β_{VaR}에 따라 알고리즘 I~III을 적용한 결과는 <표 2>과 같다. 알고리즘에 의해 구한 최적 λ(전환비중)

〈표 2〉 문제(P1)에 알고리즘 I, II, III 적용결과

적용 알고리즘	β_{VaR}	α^{VaR}	λ(전환비중)*	목적함수 값*
알고리즘 I	0.9	0.00	0.209	13,829.9
		0.01	0.209	13,829.9
		0.05	0.244	13,914.5
		0.10	0.306	14,061.3
		0.20	1.000	15,709.6
		0.30	1.000	15,709.6
알고리즘 II	1.0	0.1	0.000	13,333.6
		0.2	0.000	13,333.6
		0.3	0.000	13,333.6
		0.4	1.000	15,709.6
		0.5	1.000	15,709.6
알고리즘 III	1.2	0.4	infeasible	
		0.5	infeasible	
		0.6	1.000	15,709.6
		0.7	1.000	15,709.6

주) * 본 연구에서 제안한 알고리즘 I~III에 의해 구한 최적 λ(전환비중) 및 목적함수 값은 문제(P1)을 분지-한계법에 기초한 상용 해찾기를 적용한 결과와 비교했을 때, 모든 경우 동일하였다.

및 목적함수 값은 문제(P1)을 분지-한계법에 기초한 상용 해 찾기를 적용한 결과와 비교했을 때, 모든 경우 동일하였다.

위험 최소화 문제인 문제(P2)에 대해 모수 β^*에 따라 알고리즘 IV~VI을 적용한 결과는 <표 3>과 같다. 이 경우도 알고리즘에 의해 구한 최적 목적함수 값(α^{VaR})은 문제(P2)를 분지-한계법에 기초한 상용 해찾기를 적용한 결과와 비교했을 때, 모든 경우 동일하였다. 최적 전환비중 λ은 $\beta_{VaR} > 1$일 때, 다른 값(예 : 0.8841)이 나온 경우가 있으나 동일한 목적함수 값을 갖는 복수 최적해를 갖는 경우에 해당된다.

〈표 3〉 문제(P2)에 알고리즘 IV, V, VI 적용결과

적용 알고리즘	β^*	β_{VaR}	α^{VaR} (목적함수 값)*	λ (전환비중)**
알고리즘 IV	0.9	0.9	0.0000	0.0000
		1.0	0.0000	0.0000
		1.2	0.5192	1.0000 또는 0.8841
알고리즘 V	1.0	0.9	0.0000	0.0000
		1.0	0.0000	0.0000
		1.2	0.5192	1.0000
알고리즘 VI	1.1	0.9	0.1923	0.9998
		1.0	0.3269	1.000
		1.2	0.5192	1.0000 또는 0.8841
	1.2	0.9	infeasible	
		1.0	infeasible	
		1.2	infeasible	

주) * 본 연구에서 제안한 알고리즘 IV~VI에 의해 구한 최적 목적함수 값인 α^{VaR} 은 문제(P2)를 분지-한계법에 기초한 상용 해찾기를 적용한 결과와 비교했을 때, 모든 경우 동일하였다.
** 본 연구에서 제안한 알고리즘 IV~VI에 의해 구한 최적 전환비중 λ은 문제(P2)를 분지-한계법에 기초한 상용 해찾기를 적용한 결과와 비교했을 때, $\beta_{VaR} > 1$일 때, 일부 다른 값(예, 0.8841)이 나온 경우가 있으나 동일한 목적함수 값을 갖는 복수 최적해를 갖는 경우에 해당된다.

5. 결 론

본 연구는 전환사채 보유자가 주식전환이 가능한 시점에서 전환사채의 보유량 중 얼마나 주식으로 전환할 것인지, 전환 비중을 결정하는 최적화 모형과 이에 대한 효율적인 해법을 제시하였다. 제시한 최적화 모형은 투자대안이 2개인 포트폴리오 모형에 해당된다. 즉 전환사채를 채권으로 유지하는 대안과 전환사채를 전환하여 주식으로 보유하는 대안에 대해 최적가중치를 결정함으로 최적해를 제시하였다. 이를 위해 본 연구에서는 기존의 Markowitz 평균/분산 포트폴리오 모형에서 분산 대신 1996년 바젤위원회에서 결정한 VaR을 사용하였다.

또한 본 연구에서는 Markowitz 포트폴리오 선택 모형처럼 2가지 종류의 모형을 제시하였는데, 첫 번째 모형은 최대허용 위험(VaR) 조건하에서 기대수익을 최대화하는 모형이고, 둘째 모형은 최소요구 기대수익 조건하에서 위험(VaR)을 최소화하는 모형이다. 제안한 최적화 모형들은 정수계획 문제로 표현되며, 이는 일반적인 평균/VaR 포트폴리오 모형에 대한 Benati and Rizzi[5]의 결과를 적용한 것이다. 그들은 일반적인 평균/VaR 포트폴리오 모형이 NP-hard 문제라는 것을 증명하였으며, 상용 해찾기 도구를 적용하여 해를 구하였다. 그러나 본 연구에서는 전환사채 주식전환 비중 결정 문제가 갖는 문제 특성을 이용하여 모형별로 3가지 경우로 분류하여 각각에 대해 효율적인 해법을 제시하였다. 제시한 해법들은 시간 복잡도가 최대 $O(n\log n)$ 또는 그 이하로 매우 효율적이다.

본 연구에서 제시한 모형과 해법에 대해 실제 하이닉스 전환사채의 예에 적용하였다. 다양한 모수 값에 대해 적용한 결과, 본 연구에서 제시한 해법의 정확성과 효율성을 확인할 수 있었다.

참 고 문 헌

[1] 박구현, 심은택, "전환사채 주식전환을 위한 조건부 VaR 최적화", 「경영과학」, 제28권, 제2호(2011), pp.1-16.

[2] 정재우, 민대기, "불확실성을 고려한 장기 전원 포트폴리오의 평가", 「한국경영과학지」, 제37권, 제3호(2012), pp.135-150.

[3] Artzner, P., F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, Vol.9(1999), pp.203-228.

[4] Ayache, E., P.A. Forsyth, and K.R. Vetzal, "The valuation of convertible bonds with credit risk," *Journal of Derivatives*, Vol.11 (2003), pp.9-29.

[5] Benati, S. and A. Rizzi, "A mixed integer linear programming formulation of the optimal mean/Value-at-Risk portfolio problem," *European Journal of Operational Research*, Vol.176(2007), pp.423-434.

[6] Brennan M.J. and E.S. Schwartz, "Convertible bonds : Valuation and optimal strategies for call and conversion," *Journal of Finance*, Vol.32(1977), pp.1699-1715.

[7] Habian, C.I., "Handling CVaR objectives and constraints in two-stage stochastic models," *European Journal of Operational Research*, Vol.191(2008), pp.888-911.

[8] Huang, D., S. Zhu, F.J. Fabozzi, and M. Fukushima, "Portfolio selection under distributional uncertainty : A relative roburst CVaR approach," *European Journal of Operational Research*, Vol.203(2010), pp.185-194.

[9] Ingersoll, J.E., "A contingent-claims valuation of convertible securities," *Journal of Financial Economics*, Vol.4(1977), pp.289-322.

[10] Kunzi-Bay, A. and J. Mayer, "Computational aspects of minimizing conditional value-at-risk," *Computational Management Science*, Vol.3(2006), pp.3-27.

[11] Rockafellar, R.T. and S. Uryasev, "Optimi-

zation of conditional value-at-risk," *The Journal of Risk*, Vol.2(2000), pp.21-41.

[12] Rockafellar, R.T. and S. Uryasev, "Conditional value-at-risk for general loss distributions," *The Journal of Banking and Finance*, Vol.26(2002), pp.1443-1471.

[13] Tsai, J.T., J.L. Wang, L.Y. Tzeng, "On the optimal product mix in life insurance companies using conditional value at risk," *Insurance : Mathematics and Economics*, Vol.46(2010), pp.235-241.

[14] Tsiveriotis, K. and C. Fernandes, "Valuing convertible bonds with credit risk," *Journal of Fixed Income*, Vol.8(1998), pp.95-102.

〈부　록〉

1. 정리 1 증명

$\beta_{VaR} < 1$이면, 제약식 (11)의 우변에서 $\beta_{VaR}b$의 값이 주식전환 없이 채권으로 만기까지 보유할 경우의 수익 b 이하이므로 항상 가능해를 갖는다. 즉 $\lambda=0$ 및 모든 i에 대해 $y_i=1$으로 놓으면 제약식 (11)~식 (14)를 모두 만족함을 알 수 있다.

알고리즘의 앞문단에서 언급한 바와 같이 문제 (P1) 목적함수를 최대화하는 것은 $\frac{1}{n}\sum_{i=1}^{n} ms_i$와 b의 크기 비교에 따라 λ값을 최대화 또는 최대화하게 된다. 먼저, $\frac{1}{n}\sum_{i=1}^{n} ms_i < b$이면 λ를 최소화해야 한다. 즉 $\lambda^*=0$로 놓는다. 그러면 모든 i에 대해 $y_i=1$으로 제약식 (11)~식 (14)이 만족되고 목적함수 값은 b가 된다. $\frac{1}{n}\sum_{i=1}^{n} ms_i = b$이면 어떤 $\lambda(0 \le \lambda \le 1)$에 대해서도 목적함수 값은 b로 일정하다. 이제 $\frac{1}{n}\sum_{i=1}^{n} ms_i > b$를 가정한다. 이때는 제약식을 만족하는 λ중에서 최대값을 선택하면 된다. 즉 제약식 (12)를 만족하는 최대값 λ를 구하면 된다. 첫째, $ms_i \ge \beta_{VaR}b$ 인 경우는 모든 $\lambda(0 \le \lambda \le 1)$에 대해서 제약식 (11)이 $y_i=1$로 만족된다. 둘째, $ms_i < \beta_{VaR}b$인 경우는 $\lambda \le \frac{(1-\beta_{VaR})b}{b-ms_i}$일 때는 제약식 (11)이 $y_i=1$로 만족되고, $\lambda > \frac{(1-\beta_{VaR})b}{b-ms_i}$일 때는 제약식 (11)이 $y_i=0$이 되어야 만족된다. 제약식 (12)를 만족하기 위해서는 제약식 (11)이 $y_i=0$로 만족되는 개수가 $\lfloor n\alpha^{VaR} \rfloor$ 이하여야 한다. 따라서 $y_i=0$로 만족되는 개수가 $\lfloor n\alpha^{VaR} \rfloor$일 때 λ가 최대값을 갖는 경우이다. 이와 같이 $y_i=0$이 되는 개수가 $\lfloor n\alpha^{VaR} \rfloor$이 되게 하기 위해 $\left\{\frac{(1-\beta_{VaR})b}{b-ms_i} | ms_i < \beta_{VaR}b,\ i=1, \cdots, n\right\}$를 내림차순으로 아래와 같이 정렬한다. 여기서 $k=\sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR}b\}}$이다.

$$w_1 \ge w_2 \ge \cdots \ge w_{j^*} \ge w_{j^*+1} \ge \cdots \ge w_k$$

그러면 $\lambda^* = w_{j^*}$로 놓는다. 여기서 $j^* = k - \lfloor n\alpha^{VaR} \rfloor$이다. 그러면, $y_i=0$로 제약식 (11)이 만족되는 개수 $\le \lfloor n\alpha^{VaR} \rfloor$이다. 따라서 $0 \le \lambda \le 1$범위를 고려하여, $\lambda^* = \min[1,\ w_{j^*}]$로 놓는다. □

2. 정리 2 증명

$\lambda=0$ 및 모든 i에 대해 $y_i=1$로 놓으면 제약식 (11)~식 (14)를 모두 만족한다. 따라서 문제 (P1)은 가능해를 갖는다. 이제 제약식 (11)을 고려한다. 첫째, $ms_i \ge \beta_{VaR}b$인 경우는 모든 $\lambda(0 \le \lambda \le 1)$에 대해서 제약식 (11)이 $y_i=1$로 만족된다. 둘째, $ms_i < \beta_{VaR}b$인 경우는 $\lambda=0$이면 제약식 (11)이 $y_i=1$로 만족되나, $\lambda \ne 0$이면 제약식 (11)이 $y_i=0$이 되어야 만족된다. 이제 제약식 (12)를 만족케 하기 위해 3경우로 나누어 고려한다.

먼저 $\frac{1}{n}\sum_{i=1}^{n} ms_i < b$인 경우를 고려한다. 이때는 λ를 최소화하면 된다. 즉 $\lambda^*=0$로 놓으면 제약식 (11)이 모두 $y_i=1$로 만족된다. 따라서 제약식 (12)가 좌변 0으로 성립된다. $\frac{1}{n}\sum_{i=1}^{n} ms_i = b$ 경우는 λ에 무관하게 목적함수 값이 동일한 값 b를 가지므로 알고리즘처럼 $\lambda^*=0$로 놓을 수 있다. 이제 $\frac{1}{n}\sum_{i=1}^{n} ms_i > b$를 가정한다. 그러면 λ를 최대화하여야 한다. 만약 $\lambda > 0$ 값을 갖는다고 하자. 그러면 제약식 (12)로부터 ($y_i=0$인 경우의 수)$/n \le \alpha^{VaR}$이 되어야 한다. 따라서 즉 알고리즘에서 구한 $k \le \lfloor n\alpha^{VaR} \rfloor$이면, $\lambda^*=1$으로 놓을 수 있다.

그렇지 않다면 제약식 (12)를 만족하기 위해 $\lambda^* = 0$로 놓을 수밖에 없다. 이때 $y_i = 0$인 경우의 수는 0개이다. □

3. 정리 3 증명

제약식 (11)을 고려한다. 첫째, $ms_i < \beta_{VaR} b$인 경우는 모든 $\lambda(0 \le \lambda \le 1)$에 대해서 제약식 (11)이 $y_i = 0$이 되어야 만족된다. 둘째, $ms_i \ge \beta_{VaR} b$인 경우는 $\lambda \ge \dfrac{(\beta_{VaR}-1)b}{ms_i - b}$ 일 때는 제약식 (11)이 $y_i = 1$로 만족되나, $\lambda < \dfrac{(\beta_{VaR}-1)b}{ms_i - b}$ 일 때는 제약식 (11)이 $y_i = 0$이 되어야 만족된다. 이제 제약식 (12)를 만족케 하기 위해 제약식 (11)이 $y_i = 0$이 되어야 만족되는 경우의 수를 계산한다. 따라서 첫째 경우에서 제약식 (11)이 $y_i = 0$이 되어야 만족되는 경우의 수는 $k = \sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR} b\}}$이다. 둘째 경우에서 제약식 (11)이 $y_i = 0$이 되어야 만족되는 경우의 수를 구하기 위해 $\left\{ \dfrac{(\beta_{VaR}-1)b}{ms_i - b} | ms_i \ge \beta_{VaR} b,\ i = 1, \cdots, n \right\}$를 1을 포함하여 내림차순으로 아래와 같이 정렬한다.

$$w_1 \ge w_2 \ge \cdots \ge w_{j^*} > 1 \ge w_{j^*+1} \ge \cdots \ge w_{n-k}$$

그러면 둘째 경우에서 제약식 (11)이 $y_i = 0$이 되어야 만족되는 경우의 수는 적어도($\lambda = 1$로 놓아도) j^*개 이다. 따라서 $k + j^* > \lfloor n\alpha^{VaR} \rfloor$ 이면, 즉 $\delta = \lfloor n\alpha^{VaR} \rfloor - k - j^* < 0$이면, 제약식 (12)는 만족되지 않는다. 따라서 문제(P2)는 가능해가 존재하지 않는다.

이제 $\delta \ge 0$라고 하자. 이제 목적함수와 관련하여 3 경우로 나누어 고려한다. 먼저 $\frac{1}{n}\sum_{i=1}^{n} ms_i > b$라면, λ를 최대화하여야 한다. $\lambda^* = 1$로 놓으면 된다. 다음으로 $\frac{1}{n}\sum_{i=1}^{n} ms_i = b$ 경우는 λ에 무관하게 목적함수 값이 동일한 값 b를 가지므로 알고리즘처럼 $\lambda^* = 1$로 놓을 수 있다. 마지막으로 $\frac{1}{n}\sum_{i=1}^{n} ms_i < b$인 경우를 가정한다. 그러면 $\lambda^* = \min[1, w_{j^*+\delta}]$로 놓음으로 λ를 최소화한다. □

4. 정리 4 증명

먼저 $\beta_{VaR} \le 1$라고 가정한다. 목적함수 α^{VaR}의 최소값은 제약식 (18)로부터 결정되므로 제약식 (17)이 성립되기 위해 $y_i = 0$이 되어야 하는 경우의 수가 최소이면 된다. 먼저 $\beta_{VaR} < 1$인 경우부터 고려한다. $b > \beta_{VaR} b$이므로 $ms_i \ge \beta_{VaR} b$이면 모든 λ에 대해서 제약식 (17)이 $y_i = 1$로 만족된다. 만약 $ms_i < \beta_{VaR} b$이면 $\lambda \le \dfrac{(1-\beta_{VaR})b}{b - ms_i}$ 일 때, 제약식 (17)이 $y_i = 1$로 만족된다. 따라서 $\lambda^* = 0$로 놓으면 모든 제약식 (17)이 $y_i = 1$로 만족된다. 즉 $\alpha^* = 0$이다. 이제 $\beta_{VaR} = 1$인 경우 고려한다. $b = \beta_{VaR} b$이므로 $ms_i \ge \beta_{VaR} b$이면 제약식 (17)은 모든 λ에 대해서 $y_i = 1$로 만족된다. 만약 $ms_i < \beta_{VaR} b$이면 $\lambda = 0$일 때는 $y_i = 1$로 만족되고, $\lambda \ne 0$일 때는 $y_i = 0$이 되어야 만족한다. 따라서 $\lambda^* = 0$로 놓으면 모든 제약식 (17)을 $y_i = 1$로 만족된다. 즉 $\alpha^* = 0$이다.

이제 $\beta_{VaR} > 1$를 가정한다. 알고리즘처럼 $\frac{1}{n}\sum_{i=1}^{n} ms_i \ge \beta^* b$인 경우와 $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$인 경우로 나누어 고려한다.

(i) $\frac{1}{n}\sum_{i=1}^{n} ms_i \ge \beta^* b$ 경우

알고리즘 IV 앞 문단에서 언급한 바와 같이 이 경

우는 모든 $\lambda(0 \le \lambda \le 1)$에 대해 제약식 (16)은 항상 성립한다. 따라서 제약식 (17)에서 $y_i = 0$이 되어야만 그 부등식이 만족하게 되는 경우의 수를 최소화시켜 제약식 (18) 우변 α^{VaR}를 최소화시킨다. 첫째, $ms_i < \beta_{VaR} b$라면, 제약식 (17)은 모든 λ에 대해 $y_i = 0$이어야 만족된다. 둘째, $ms_i = \beta_{VaR} b$라면, $\lambda = 1$일 때만 제약식 (17)은 $y_i = 1$로 만족된다. 셋째, $ms_i > \beta_{VaR} b$라면, $\lambda \ge \dfrac{(\beta_{VaR}-1)b}{ms_i - b}$일 때는 제약식 (17)이 $y_i = 1$로 만족된다. 따라서 둘째 및 셋째 경우를 모두 $y_i = 1$로 만족케 하기 위해 $\lambda^* = 1$로 놓는다. 그러면 제약식 (17)이 $y_i = 0$이 되어야 만족되는 경우의 수는 $k = \sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR} b\}}$이 되며, 따라서 최소값 α^{VaR}은 $\alpha^* = (n-k)/n$이 된다.

(ii) $\dfrac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$ 경우

알고리즘 IV 앞 문단에서 언급한 바와 같이 $\lambda \le \lambda_{\max} \equiv \dfrac{(1-\beta^*)b}{\frac{1}{n}\sum_{i=1}^{n}(b - ms_i)}$이 되어야 제약식 (16)이 성립한다. 이제 $\lambda \le \lambda_{\max}$를 가정한다. 제약식 (17)을 고려한다. 첫째, $ms_i < \beta_{VaR} b$라면 $\lambda \in [0, \lambda_{\max}]$인 모든 λ에 대해 $y_i = 0$이어야 제약식 (17)이 만족된다. 둘째, $ms_i \ge \beta_{VaR} b$라면, $\lambda \ge \dfrac{(\beta_{VaR}-1)b}{ms_i - b}$일 때 제약식 (17)이 $y_i = 1$로 만족된다. 그러나 $\lambda \le \lambda_{\max}$이어야 하므로 $\left\{\dfrac{(\beta_{VaR}-1)b}{ms_i - b} \mid ms_i \ge \beta_{VaR} b,\ i = 1, \cdots, n\right\}$를 $\lambda_{\max}$를 포함하여 내림차순으로 아래와 같이 정렬한다. 여기서 $k = \sum_{i=1}^{n} I_{\{ms_i \ge \beta_{VaR} b\}}$이다.

$$w_1 \ge w_2 \ge \cdots \ge w_{j^*} > \lambda_{\max} \ge w_{j^*+1} \ge \cdots \ge w_k$$

둘째 경우에서 제약식 (17)이 $y_i = 1$로 만족되는 경우의 수를 최대로 하기 위해서, 즉 제약식 (17)이 $y_i = 0$이어야 만족되는 경우의 수를 최소화하기 위해 $\lambda^* \in [w_{j^*+1}, \lambda_{\max}]$여야 한다. 알고리즘에서는 $\lambda^* = \lambda_{\max}$로 놓았다. 그러면 제약식 (17)이 $y_i = 0$이 되어야 만족하는 경우의 수는 첫째 경우 전부와 둘째 경우 일부를 합한 $(n-k)+j^*$이 된다. 따라서 최소값 α^{VaR}은 $\alpha^* = 1 - \dfrac{n-k+j^*}{n} = \dfrac{k-j^*}{n}$이 된다. □

5. 정리 5 증명

먼저 $\beta_{VaR} \le 1$라고 가정한다. 목적함수 α^{VaR}의 최소값은 제약식 (17)이 성립되기 위해 $y_i = 0$이 되어야 하는 경우의 수가 최소일 때이다. 먼저 $\beta_{VaR} < 1$인 경우부터 고려한다. 알고리즘 앞 문단에서 언급한 바와 같이 $\dfrac{1}{n}\sum_{i=1}^{n} ms_i \ge b$라고 가정하자. 그러면 모든 $\lambda(0 \le \lambda \le 1)\lambda$에 대해 제약식 (16)은 성립된다. 이제 제약식 (17)을 고려하자. 첫째, $ms_i \ge \beta_{VaR} b$이면 모든 λ에 대해 제약식 (17)이 $y_i = 1$로 만족된다. 둘째, $ms_i < \beta_{VaR} b$이면 $\lambda \le \dfrac{(1-\beta_{VaR})b}{b - ms_i}$일 때 제약식 (17)이 $y_i = 1$로 만족된다. 따라서 $\lambda^* = 0$로 놓으면 모든 제약식 (17)이 $y_i = 1$로 만족된다. 따라서 이때 $\alpha^* = 0$이다. 만약 $\dfrac{1}{n}\sum_{i=1}^{n} ms_i < b$이면 $\lambda^* = 0$일 때만 제약식 (16)이 만족된다. 이때 모든 제약식 (17)이 $y_i = 1$로 만족된다. 따라서 $\alpha^* = 0$이 된다.

이제 $\beta_{VaR} = 1$인 경우 고려한다. 먼저 $\dfrac{1}{n}\sum_{i=1}^{n} ms_i \ge b$라고 가정한다. 첫째, $ms_i \ge \beta_{VaR} b$이면 제약식 (17)은 모든 λ에 대해 제약식 (17)이 $y_i = 1$로 만족된다. 둘째, $ms_i < \beta_{VaR} b$이면 $\lambda = 0$일 때 제

약식 (17)이 $y_i=1$로 만족되고, $\lambda \neq 0$일 때는 $y_i=0$이어야 만족한다. 따라서 $\lambda^*=0$로 놓으면 모든 제약식 (17)이 $y_i=1$로 만족된다. 따라서 이때 $\alpha^*=0$이다. 만약 $\frac{1}{n}\sum_{i=1}^{n} ms_i < b$이면 $\lambda^*=0$일 때만 제약식 (16)이 만족된다. 이때 모든 제약식 (17)이 $y_i=1$로 만족된다. 따라서 $\alpha^*=0$이 된다.

이제 $\beta_{VaR}>1$를 가정한다. 알고리즘처럼 $\frac{1}{n}\sum_{i=1}^{n} ms_i \geq \beta^* b$인 경우와 $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$인 경우로 나누어 고려한다.

(i) $\frac{1}{n}\sum_{i=1}^{n} ms_i \geq \beta^* b$ 경우

알고리즘 V 앞 문단에서 언급한 바와 같이 이 경우는 모든 $\lambda(0 \leq \lambda \leq 1)$에 대해 제약식 (16)이 만족된다. 따라서 제약식 (17)에서 $y_i=0$이 되어야만 그 부등식이 만족하게 되는 경우의 수를 최소화시켜 제약식 (18) 우변 α^{VaR}를 최소화시킨다. 첫째, $ms_i < \beta_{VaR} b$라면, 제약식 (17)은 모든 λ에 대해 $y_i=0$이어야 만족된다. 둘째, $ms_i = \beta_{VaR} b$라면, $\lambda=1$일 때만 제약식 (17)이 $y_i=1$로 만족된다. 셋째, $ms_i > \beta_{VaR} b$라면, $\lambda \geq \frac{(\beta_{VaR}-1)b}{ms_i - b}$일 때 제약식 (17)이 $y_i=1$로 만족된다. 따라서 둘째 및 셋째 경우를 모두 $y_i=1$로 만족케 하기 위해 $\lambda^*=1$로 놓는다. 그러면 제약식 (17)이 $y_i=0$이어야 만족되는 경우의 수는 $k=\sum_{i=1}^{n} I_{\{ms_i < \beta_{VaR} b\}}$이며, 따라서 최소값 α^{VaR}은 $\alpha^*=(n-k)/n$이 된다.

(ii) $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$ 경우

알고리즘 V 앞 문단에서 언급한 바와 같이 $\lambda=0$이 되어야 제약식 (16)이 성립된다. 이때 제약식 (17)은 모두 $y_i=0$이어야만 만족되므로, $\alpha^*=1$이 된다. □

6. 정리 6 증명

먼저 문제(P2)가 가능해를 갖지 않는 경우를 증명한다. 즉 $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$임을 가정한다. 그러면 $\frac{1}{n}\sum_{i=1}^{n}\{ms_i - b\} < (\beta^*-1)b$이 된다. 만약 $\frac{1}{n}\sum_{i=1}^{n}\{ms_i - b\} > 0$이면, $0 \leq \lambda \leq 1$범위의 모든 λ에 대해서 아래의 관계가 성립된다.

$$\frac{\lambda}{n}\sum_{i=1}^{n}\{ms_i - b\} \leq \frac{1}{n}\sum_{i=1}^{n}\{ms_i - b\} < (\beta^*-1)b$$

첫째 부등식의 좌변과 둘째 부등식의 우변으로부터 $(1-\lambda)b + \frac{\lambda}{n}\sum_{i=1}^{n} ms_i < \beta^* b$이 성립된다. 따라서 제약식 (16)이 성립되지 않는다. 이제 $\frac{1}{n}\sum_{i=1}^{n}\{ms_i - b\} \leq 0$라고 한다면 $(\beta^*-1)b > 0$이므로 $0 \leq \lambda \leq 1$ 범위의 모든 λ에 대해서도 $\frac{\lambda}{n}\sum_{i=1}^{n}\{ms_i - b\} < (\beta^*-1)b$가 성립된다. 따라서 $(1-\lambda)b + \frac{\lambda}{n}\sum_{i=1}^{n} ms_i < \beta^* b$이 성립된다. 역시 제약식 (16)이 성립되지 않는다. 따라서 $\frac{1}{n}\sum_{i=1}^{n} ms_i < \beta^* b$이면 문제(P2)는 가능해를 갖지 않는다.

이제 $\frac{1}{n}\sum_{i=1}^{n} ms_i \geq \beta^* b$를 가정한다. 그러면 알고리즘 VI의 앞 문단에서 언급한 바와 같이 제약식 (16)이 만족하기 위해서 $\lambda \geq \lambda_{\min}$이어야 한다. 이제 $\lambda \geq \lambda_{\min}$에 대해 제약식 (17)이 $y_i=0$이어야만 만족되는 경우의 수를 최소화되게 하려 한다. 이를 위하여 $\beta_{VaR}>1$인 경우, $\beta_{VaR}=1$인 경우, β_{VaR}

<1인 경우로 나누어 생각한다.

(i) $\beta_{VaR}>1$인 경우

첫째, $ms_i<\beta_{VaR}b$라면, 제약식 (17)은 모든 λ에 대해 $y_i=0$이어야 만족된다. 둘째, $ms_i=\beta_{VaR}b$라면, $\lambda=1$일 때 제약식 (17)은 $y_i=1$로 만족된다. 셋째, $ms_i>\beta_{VaR}b$라면, $\lambda\geq\dfrac{(\beta_{VaR}-1)b}{ms_i-b}$일 때 제약식 (17)이 $y_i=1$로 만족된다. 따라서 둘째 및 셋째 경우를 모두 $y_i=1$로 만족케 하기 위해 $\lambda^*=1$로 놓는다. 그러면 제약식 (17)이 $y_i=0$이 되어야 만족되는 경우의 수는 $k=\sum_{i=1}^{n} I_{\{ms_i<\beta_{VaR}b\}}$이 되며, 따라서 최소값 α^{VaR}은 $\alpha^*=(n-k)/n$이 된다.

(ii) $\beta_{VaR}=1$인 경우

첫째, $ms_i\geq b$인 경우는 모든 λ에 대해 제약식 (17)이 $y_i=1$로 만족된다. 둘째, $ms_i<b$인 경우는 $\lambda=0$일 때만 제약식 (17)이 만족된다. 그러나 $\lambda\geq\lambda_{\min}$여야 하므로, $\lambda^*=1$로 놓는다. 그러면 두 번째 경우에 해당되는 수만큼 $y_i=0$이어야 하므로 최소값 α^{VaR}은 $\alpha^*=(n-k)/n$이 된다. 여기서 $k=\sum_{i=1}^{n} I_{\{ms_i<\beta_{VaR}b\}}$이다. 따라서 $\beta_{VaR}>1$인 경우와 동일한 λ^*와 α^*값을 갖는다.

(iii) $\beta_{VaR}<1$인 경우

첫째, $ms_i\geq\beta_{VaR}b$인 경우는 모든 λ에 대해 제약식 (17)이 $y_i=1$로 만족된다. 둘째, $ms_i<\beta_{VaR}b$인 경우는 $\lambda\leq\dfrac{(1-\beta_{VaR})b}{b-ms_i}$일 때 제약식 (17)이 $y_i=1$로 만족된다. 그러나 $\lambda\geq\lambda_{\min}$이어야 하므로 $\left\{\dfrac{(1-\beta_{VaR})b}{b-ms_i}\middle| ms_i<\beta_{VaR}b,\ i=1,\cdots,n\right\}$를 $\lambda_{\min}$을 포함하여 내림차순으로 아래와 같이 정렬한다. 여기서 $k=\sum_{i=1}^{n} I_{\{ms_i<\beta_{VaR}b\}}$이다.

$$w_1\geq w_2\geq\cdots\geq w_{j^*}\geq\lambda_{\min}>w_{j^*+1}\geq\cdots\geq w_k$$

둘째 경우에서 제약식 (17)이 $y_i=1$로 만족되는 경우의 수를 최대로 하기 위해 $\lambda^*=w_{j^*}$로 놓는다. 그러면 제약식 (17)이 $y_i=0$이어야 만족하는 경우의 수는 최소가 되며, 둘째 경우에서 $k-j^*$개 임을 알 수 있다. 따라서 최소값 α^{VaR}은 $\alpha^*=1-\dfrac{n-(k-j^*)}{n}=\dfrac{k-j^*}{n}$이 된다. □