Journal of the Operations Research Society of Japan
Vol. 54, No. 2-3, September 2011, pp. 86–100

PRICING SWING OPTIONS WITH TYPICAL CONSTRAINTS

Yusuke Tashiro
The University of Tokyo

(Received June 15, 2010; Revised February 22 2011)

Abstract We propose a pricing method by mathematical programming for swing options with typical constraints on a lattice model. We show that the problem of pricing typical swing options has a particular optimal solution such that there are only seven kinds of changed amounts in the solution. Using the solution, we formulate the pricing problem as a linear program. The solution can be applied to the methods of Jaillet, Ronn and Tompaidis (2004), and Barrera-Esteve et al. (2006) for improving time complexity.

Another feature of our method is the capability to price swing options in an incomplete market. In an incomplete market, the price of a swing option is defined as an upper and a lower bound of arbitrage-free prices. We formulate the problem of finding an upper bound as a linear program. For a lower bound, we give a bilinear programming formulation.

Keywords: Finance, mathematical programming, option pricing, lattice model

1. Introduction

With recent deregulation of energy markets, many derivative instruments have been designed. Some of these are swing options. Swing options are generally traded in gas and electricity markets. A holder of a swing option buys fixed amount of energy from an option seller at fixed dates, and then the holder also has rights to change the amount at some times. The amount is subject to daily and periodic (monthly or annual) constraints. The number of rights is also limited. The option holder changes the amount depending on their purpose, such as maximizing-profit and request of demand.

The valuation of swing options is known to be more difficult than that of vanilla options, because swing options have not only timing constraints but also volume constraints. Typical techniques for pricing swing options are the least-squares Monte-Carlo method and dynamic programming.

The least-squares Monte-Carlo method was applied by Longstaff and Schwartz [9] to American option pricing. Dörr [3], Meinshausen and Hambly [10], and Barrera-Esteve et al. [1] extended the least-squares Monte-Carlo method to swing options.

On the other hand, dynamic programming was studied by Jaillet et al. [7], Lari-Lavassani et al. [8] and Thompson [14]. They expressed the underlying asset price process on a lattice and computed the option price by the backward procedure.

However, these studies do not cover every setting of swing options. First, with these studies, it is difficult to consider changed volume as a continuous value, so that some discretized values are used. Thus, these studies may give biased price. Second, these studies are applicable to the pricing problem in a complete market. This setting is usual, but not adequate when a market is incomplete.

Our approach is pricing by mathematical programming. Mathematical programming is flexible enough to add constraints as conditional expressions. Swing option pricing by mathematical programming can also treat changed amount as a continuous variable. Haarbrücker and Kuhn [5], and Steinbach and Vollebrecht [13] recently studied swing option pricing. Haarbrücker and Kuhn [5] proposed the valuation of swing options with ramping constraints on a scenario tree. Steinbach and Vollebrecht [13] proposed the valuation technique by reducing a scenario tree and using a scenario fan.

Our swing option setting is typical and similar to that of Jaillet et al. [7]. Our swing option has local (daily) constraints, global (annual) constraints, and timing constraints. The first two constraints are typical constraints. Our formulation is based on a scenario lattice. However, on a lattice, a formulation that is similar to that of previous works by mathematical programming is not successful because swing options are path-dependent options. Thus, to begin with, we decompose the lattice to a tree, formulate the pricing problem on the tree, and find an optimal solution that has particular changed amounts. Then using the particular solution, we formulate the pricing problem as a linear program on the lattice. Furthermore, we apply the particular solution to the methods of Jaillet et al. [7] and Barrera-Esteve et al. [1] for improving computation time and accuracy.

An advantage of our approach is that the formulated pricing problem can be extended to that in an incomplete market. In an incomplete market, we define the pricing problem as the problems of finding an upper bound and a lower bound of "arbitrage-free prices". For American options, Föllmer and Schied [4] showed a pricing method using the Snell envelope in a discrete case. Pennanen and King [11], and Camci and Pinar [2] studied pricing of American options by stochastic programming in an incomplete market. These pricing methods are performed under the martingale probability for the underlying asset. In energy markets, it is difficult to store the underlying asset, so that the pricing is meaningless under the martingale probability for the underlying asset. However, there are futures contracts and some tradable products that relate to the underlying asset process in energy markets. We then use the martingale probability Q for these products, and we formulate the upper and lower bound problems of swing options as mathematical programming in an incomplete market.

The paper is organized as follows. Section 2 provides the definition of swing options and a formulation of the pricing problem on a tree. Section 3 shows a particular solution of the pricing problem on a tree, and using the particular solution, formulates the efficient pricing problem as mathematical programming on a lattice. Section 4 applies the solution to other pricing methods for improving upon computation time and accuracy. Section 5 focuses on the pricing problem in an incomplete market, and formulates the upper and the lower bound problem. In addition, we design a backward algorithm to compute the upper and lower bounds and show a numerical result. Section 6 concludes.

2. The Model

2.1. Swing options

There are a buyer and a seller of energy. They close a contract to buy some amount u_t of energy at a strike price of K_t at date $t = t_i$ $(i = 0, 1, \ldots, T-1)$. A swing option in this paper is defined as rights to change of delivery amount with this contract. When a swing option is added to the contract, the buyer can change the amount from u to $u + v_t$ up to $L(\leq T)$ times at $t = t_0, t_1, \ldots, t_{T-1}$ under some constraints. One of the constraints is a local constraint for DCQ (Daily Contract Quantity):

$$v_{\min} \leq v_t \leq v_{\max},$$

where $v_{\min} \leq 0$, $v_{\max} \geq 0$, and $v_{\min}$ and $v_{\max}$ are time-invariant. Total changed amount is also limited by a global constraint for ACQ (Annual Contract Quantity):

$$V_{\min} \leq \sum_{i=0}^{T-1} v_{t_i} \leq V_{\max}.$$

Furthermore, the interval of exercise is also restricted. The option holder exercising a right at t_i cannot change amount during $t_i < t < t_i + \Delta t_R$. Here Δt_R is called the refraction time.

2.2. Asset price processes and profits

We describe an asset price process on a scenario lattice. Figure 1 is an example of our scenario lattice. Such a lattice is often called a trinomial tree. Let N denote the set of nodes of a lattice, S_n the underlying asset price* at node n, and $\mathcal{N}_i$ the set of nodes at t_i. We denote by $B(n)$ and $C(n)$ the set of parents and children of node n, respectively. In this paper, we call a lattice with $|B(n)| \leq 1$ for any n as a tree. We define $p_{nk}(> 0)$ as the conditional transition probability from node n to node m ($m \in C(n)$). Concerning the probability at node n,

$$\sum_{k \in C(n)} p_{nk} = 1 \tag{1}$$

holds for each n.

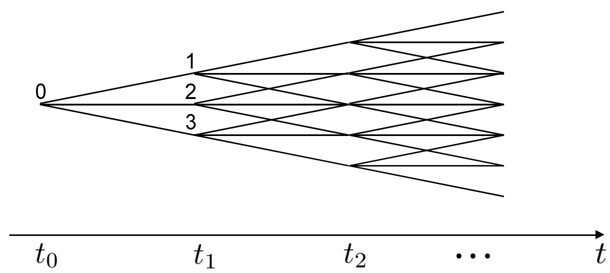

Figure 1: An example of a scenario lattice ($\mathcal{N}_1 = \{1, 2, 3\}$ in this example)

At each node, a buyer may change amount of energy. We assume a buyer faces not demand problems but financing problems. Namely, we permit a buyer to sell excess amount at a market price. Then a profit made by changed amount v_n at node n is represented as

$$v_n(S_n - K_n).$$

2.3. Pricing on a tree

We assume that a buyer is rational; thus we define the price of a swing option as the maximum expected value of the total profit. Our aim is pricing swing options by mathematical programming on a lattice. On a formulation using mathematical programming, the number of variables is proportional to the number of states on the model. However, because of the constraints for ACQ, states of a node are path-dependent. The number of states on a model is generally proportional to the number of paths.

Example 1. *Let us consider a swing option in Figure 1. We assume that a buyer exercises a right at node 1 with $v_1 = 1$ and exercises a right at node 2 with $v_2 = 2$. Then at node 6, the state from node 1 and that from node 2 are different.*

*S_n is already discounted by a risk-free asset. We also discount a strike price K_n that is equal to K_{t_i} ($n \in \mathcal{N}_i$).

For dealing with path-dependence of swing options, we first decompose the lattice into a tree and consider pricing on a tree. Here decomposing the lattice into a tree means that any node that has more than one parent is decomposed into nodes such that each node has only a parent. Then the number of paths is equal to the number of nodes in the maturity on the tree. Thus a path-dependent formulation is equivalent to that on the tree, and we consider the pricing problem on the tree.

Let $\mathcal{M}_i$ denote the set of nodes of a tree at t_i and e_n denote a variable representing whether or not a right is exercised at node n of a tree. For example, $e_n = 1$ means an exercise of a right at node n. Then the optimization problem of maximizing the expected value is as follows:

$$
\begin{array}{lrcll}
\max\limits_{v,e} & \mathrm{E}[v_n(S_n - K_n)] & & & \\
\text{s.t.} & v_{\min} e_n \le v_n & \le & v_{\max} e_n & (n \in \mathcal{M}_{T-}) \\
& V_{\min} \le \displaystyle\sum_{m \in \mathcal{A}(n)} v_m & \le & V_{\max} & (n \in \mathcal{M}_{T-1}) \\
& \displaystyle\sum_{m \in \mathcal{A}(n)} e_m & \le & L & (n \in \mathcal{M}_{T-1}) \\
& e_n & \in & \{0,1\} & (n \in \mathcal{M}_{T-}) \\
& e_n + e_m & \le & 1 & (n \in \mathcal{M}_i,\ m \in \mathcal{M}_{j,n},\ i < j,\ t_j - t_i < \Delta t_R),
\end{array}
\tag{2}
$$

where $\mathcal{M}_{T-} = \{n \mid n \in \mathcal{M}_i,\ i \le T-1\}$, $\mathcal{M}_{i,n}$ is the set of node $m \in \mathcal{M}_i$ such that m is a sink node of node n, and $\mathcal{A}(n)$ is the path history from the root to node n.

3. Pricing on a Lattice

In subsequent sections, we assume for simplicity that $t_{i+1} - t_i = \Delta t$ $(i = 0, \ldots, T-1)$ and the refraction time $\Delta t_R = \Delta t$, i.e., we exclude timing constraints. But the extension to general t_i and Δt_R is easy.

3.1. A particular solution

In Section 2, we formulated pricing on a tree made of a lattice. However, the tree may have an exponential number of nodes, so that exponential time is necessary to solve Problem (2). We aim to reduce the time to be proportional to the number of nodes on the lattice.

We focus on a value of changed amount v_n. If the value of v_n is chosen from a discrete set $\{v^1, \ldots, v^k\}$, the number of states on a lattice can be described as the number of possible combinations of the number the option was exercised with each of v^i. The following theorem shows that there is a particular optimal solution in terms of a value of v_n.

Theorem 1. *In the set of optimal solutions of Problem (2), there is a solution such that there are at most seven kinds of values of v_n in the solution.*

Before the proof of Theorem 1, we need some preparation.

First, without loss of generality we can change the constraints of Problem (2) from $\sum_{m \in \mathcal{A}(n)} e_m \le L$ to $\sum_{m \in \mathcal{A}(n)} e_m = L$, because if there is a path with $\sum_{m \in \mathcal{A}(n)} e_m < L$, the path satisfies $\sum_{m \in \mathcal{A}(n)} e_m = L$ by the exercise of residual rights with zero amount at non-exercised nodes. We name this modified problem as (P′).

We define two properties of node n:

- "bang-bang": $e_n = 1$ and $v_n \in \{v_{\min}, v_{\max}\}$,
- "non bang-bang": $e_n = 1$ and $v_n \ne v_{\min}, v_{\max}$.

We also define a property of a path l:

- "tight": $\sum_{m \in l} v_m = V_{\min}$ or $\sum_{m \in l} v_m = V_{\max}$.

Concerning Problem (P′), the next lemma holds:

Lemma 1. *Problem (P′) has an optimal solution with the following property:*

- *For any node n with "non bang-bang" such that $\sum_{m \in \mathcal{A}(n)} e_m < L$, there is a "tight" path l such that $n \in l$ and if $m \in l$ is a sink node of node n then node m is not "non bang-bang".*

Proof. First, from linearity of Problem (2) we can take an optimal solution such that for any node n with "non bang-bang" there is a "tight" path l which includes n. Let us assume that the optimal solution does not satisfy the above property. Then in the solution, for some node n with "non bang-bang" such that $\sum_{m \in \mathcal{A}(n)} e_m < L$, any "tight" path including node n includes a "non bang-bang" node m under node n. We focus on such nodes n and m. We can consider two transformations of the optimal solution with no effect on global constraints:

- to decrease the value v_n by Δ and increase the value v_m by Δ,
- to increase the value v_n by Δ and decrease the value v_m by Δ,

where Δ is a sufficiently small positive constant. At least one transformation does not reduce the objective value, because the objective function of Problem (P′) is linear and if one transformation decreases the objective value, then another transformation increases the objective value. By increasing Δ, the value v_n at not less than one node changes to $v_{\min}$ or $v_{\max}$. This change is represented in Figure 2. By repeating the transformation for node n, a path including node n satisfies the desired property. By the transformations for any node with "non bang-bang" in order from the root of the tree, the desired property is added to the optimal solution. □

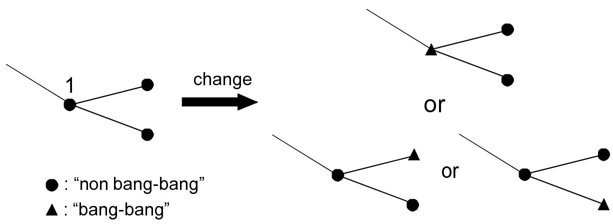

Figure 2: A transformation of the optimal solution

About node 1, the optimal solution is transformed not to reduce the objective function. Then the solution changes in two patterns. On the top of Figure, node 1 is "bang-bang". On the bottom, an upper or a lower path satisfies the property of Lemma 1.

From Lemma 1, we prove Theorem 1.

Proof. We analyze the particular solution claimed in Lemma 1. Let us look at the value of v_n such that n is "non bang-bang" step by step from the root.

Step 1: Look at v_n such that there is no source node of node n with "non bang-bang"

By Lemma 1, there is a "tight" path that includes node n and does not include a node with "non bang-bang" except n. On the path, each node m ($\neq n$) is "bang-bang" or satisfies $e_m = 0$. The number of nodes with "bang-bang" is $L-1$, and v_m at node m with "bang-bang" is equal to $v_{\max}$ or $v_{\min}$. In addition, the path has a total volume of $V_{\min}$ or $V_{\max}$ because the path is "tight". If the total volume is $V_{\min}$, v_n must have only a value. The value is $(V_{\min} - L \cdot v_{\min}) \bmod (v_{\max} - v_{\min}) + v_{\min}$. Similarly, if the total volume is $V_{\max}$, v_n must be equal to $v_{\max} - (L \cdot v_{\min} - V_{\max}) \bmod (v_{\max} - v_{\min})$. Let $(V_{\min} - L \cdot$

$v_{\min}$) mod $(v_{\max} - v_{\min}) + v_{\min}$ be equal to v^3 and $v_{\max} - (L \cdot v_{\min} - V_{\max})$ mod $(v_{\max} - v_{\min})$ be equal to v^4.

Step 2: Look at v_n such that there is a source node of node n with "non bang-bang"
Let the source node of node n with "non bang-bang" denote n'. About n, by Lemma 1, there is a "tight" path that does not include a node with "non bang-bang" except n and n'. From Step 1, a value of $v_{n'}$ is v^3 or v^4. Furthermore, the path has a total volume of $V_{\min}$ or $V_{\max}$ because of "tight". If the total volume is $V_{\min}$, v_n has only a value in regard to each v^3 or v^4. When $v_{n'} = v^3$, v_n is equal to $v_{\min}$ or $v_{\max}$ from Step 1. When $v_{n'} = v^4$, v_n is equal to $(v^3 - v^4)$ mod $(v_{\max} - v_{\min}) + v_{\min}$. On the other hand if the total volume is $V_{\max}$, when $v_{n'} = v^4$, v_n is equal to $v_{\min}$ or $v_{\max}$ from Step 1, and when $v_{n'} = v^3$, v_n is equal to $v_{\max} - (v^3 - v^4)$ mod $(v_{\max} - v_{\min})$. Let $(v^3 - v^4)$ mod $(v_{\max} - v_{\min}) + v_{\min}$ be equal to v^5 and $v_{\max} - (v^3 - v^4)$ mod $(v_{\max} - v_{\min})$ be equal to v^6.

Step 3: Look at v_n such that there are two source nodes of node n with "non bang-bang"
Let the source nodes of node n with "non bang-bang" denote n' and n''. From Step 2, $v_{n'} + v_{n''}$ is equal to $v^3 + v^6$ or $v^4 + v^5$. However, by a calculation, $v^3 + v^6$ is equal to $v_{\max} + v^4$, and $v^4 + v^5$ is equal to $v_{\min} + v^3$. Thus if $v_{n'} + v_{n''}$ is equal to $v^3 + v^6$, v_n must be v^5 from Step 2. If $v_{n'} + v_{n''}$ is equal to $v^4 + v^5$, v_n must be v^6 in the same way.

In the case that there are more than two source nodes of node n with "non bang-bang", v_n must also be v^5 or v^6 from Steps 2 and 3.

Eventually, there is a particular solution such that the value of v_n is chosen from $\{v_{\max}, v_{\min}, v^3, v^4, v^5, v^6, 0\}$ in the solution.

□

In conclusion, the possible values of v_n are as follows:

$$\begin{aligned}
v^1 &= v_{\min}, \\
v^2 &= v_{\max}, \\
v^3 &= v_{\min} + (V_{\min} - L \cdot v_{\min}) \text{ mod } (v_{\max} - v_{\min}), \\
v^4 &= v_{\max} - (L \cdot v_{\max} - V_{\max}) \text{ mod } (v_{\max} - v_{\min}), \\
v^5 &= v_{\min} + (v^3 - v^4) \text{ mod } (v_{\max} - v_{\min}), \\
v^6 &= v_{\max} - (v^3 - v^4) \text{ mod } (v_{\max} - v_{\min}), \\
v^0 &= 0.
\end{aligned} \tag{3}$$

In particular, when $v_{\min} = -1$, $v_{\max} = 1$ and $V_{\min}, V_{\max}$ have an integer value, if $e_n = 1$ then v_n must be $v_{\min}$ or $v_{\max}$. Thus the next corollary holds:

Corollary 1. *When $v_{\min} = -1$, $v_{\max} = 1$ and $V_{\min}, V_{\max}$ have an integer value, there are at most three kinds as the value of v_n for Problem (2).*

Remark 1. *For the optimal solution of swing options, a closely related result has recently been obtained in a paper [12] by Ross and Zhu, of which the author became aware after the completion of this work. Our result, Theorem 1 above, may be regarded as being essentially equivalent to their result, but our theorem covers a more general case with regard to the number of rights. Moreover, our proof is based on an approach different from [12].*

3.2. Formulating the pricing problem on a lattice

In this section, we formulate the pricing problem on a lattice with the use of the particular solution given in Section 3.1.

We can consider a state of a node as a combination of the number of exercise times with each of $v^1, \ldots, v^6$. Each the number of exercise times is not more than L, so that the number of states is not more than $\sum_{i=0}^{L} {}_{i+5}\mathrm{C}_5 = {}_{L+6}\mathrm{C}_6$. We can give more efficient representation. Let $\mathrm{Num}_n(v^i)$ denote the number of exercise times with v^i between the root and node n. The next proposition reduces the number of states:

Proposition 1. *A State of a node can be described as a combination of the number of exercise times with each of only $v^1, \cdots, v^4$ on a lattice, and the number of states is at most ${}_{L+2}\mathrm{C}_2 + 2 \cdot {}_{L+1}\mathrm{C}_2$.*

Proof. First, from the proof of Theorem 1,

$$\mathrm{Num}_n(v^3) + \mathrm{Num}_n(v^4) \le 1. \tag{4}$$

Second, we focus on $\mathrm{Num}_n(v^5)$ and $\mathrm{Num}_n(v^6)$. When $\mathrm{Num}_n(v^3) + \mathrm{Num}_n(v^4) = 0$, $\mathrm{Num}_n(v^5)$ and $\mathrm{Num}_n(v^6)$ are equal to 0 because of the proof of Theorem 1. When $\mathrm{Num}_n(v^3) + \mathrm{Num}_n(v^4) = 1$, v^5 and v^6 must be alternatively chosen from Step 3 of the proof of Theorem 1, so that

$$|\mathrm{Num}_n(v^3) - \mathrm{Num}_n(v^4) + 2\,(\mathrm{Num}_n(v^5) - \mathrm{Num}_n(v^6))\,| \le 1. \tag{5}$$

This equation means that if $\mathrm{Num}_n(v^5) > \mathrm{Num}_n(v^6)$ then $\mathrm{Num}_n(v^5) - \mathrm{Num}_n(v^6) = 1$ and $\mathrm{Num}_n(v^4) = 1$. Furthermore, because $v^4 + v^5 = v^1 + v^3$ and $v^3 + v^6 = v^2 + v^4$, $\mathrm{Num}_n(v^5)$ can be equal to $\mathrm{Num}_n(v^6)$. Moreover, $v^5 + v^6 = v^1 + v^2$ for Equation (3), and thus $\mathrm{Num}_n(v^5), \mathrm{Num}_n(v^6)$ can be equal to 0.

As a result, we can describe a state of node n as a combination of the number of exercise times with each of only $v^1, \ldots, v^4$. Then the number of the states is at most ${}_{L+2}\mathrm{C}_2 + 2 \cdot {}_{L+1}\mathrm{C}_2$ because of Equation (4) and $\mathrm{Num}_n(v^1) + \mathrm{Num}_n(v^2) + \mathrm{Num}_n(v^3) + \mathrm{Num}_n(v^4) \le L$. □

Let a state of node n denote a combination of the number of exercise times $(\mathrm{Num}_n(v^1), \mathrm{Num}_n(v^2), \mathrm{Num}_n(v^3), \mathrm{Num}_n(v^4))$, x_n^j a probability at node n with a state j, and $x_n^{i,j}$ a probability of changed amount v^i at node n with a state j. Then a profit at node n with a state j is

$$\sum_{i \in I_j} v^i (S_n - K_n) x_n^{i,j},$$

where I_j is the index set of changeable amounts in a state j. Then the pricing problem is to maximize the sum of profits at each node with each state by assigning the probability x_n^j to $x_n^{i,j}$. Figure 3 designs an example of the problem. A formulation of the problem is as follows:

$$\begin{array}{rrcll}
\max\limits_{x} & \multicolumn{4}{l}{\displaystyle\sum_{n \in \mathcal{N}_{T-}} \sum_{j \in J} \sum_{i \in I_j} v^i (S_n - K_n) x_n^{i,j}} \\
\text{s.t.} & x_0^{(0,0,0,0)} & = & 1 & \\
 & x_0^j & = & 0 & (j \in J \backslash \{(0,0,0,0)\}) \\
 & x_n^j & = & \displaystyle\sum_{i \in I_j} x_n^{i,j} & (n \in \mathcal{N}_{T-}) \\
 & x_n^{i,j} & \in & \{0, x_n^j\} & (n \in \mathcal{N}_{T-},\ i \in I_j) \quad (*) \\
 & x_{nk}^{i,j} & = & p_{nk} x_n^{i,j} & (n \in \mathcal{N}_{T-},\ k \in C(n),\ i \in I_j) \\
 & x_n^j & = & \displaystyle\sum_{m \in B(n)} \sum_{i \in I_{[j]}} x_{mn}^{i,[j-i]} & (n \in \mathcal{N} \backslash \{0\}) \\
 & x_n^j & \ge & 0 & (n \in \mathcal{N}_{T-}) \\
 & x_n^j & \ge & 0 & (n \in \mathcal{N}_T,\ |j| = L) \\
 & x_n^j & = & 0 & (n \in \mathcal{N}_T,\ |j| < L),
\end{array} \tag{6}$$

where $\mathcal{N}_{T-}$ is the node set for any $t \leq t_{T-1}$, $|j|$ is the number of exercise times in a state j, J is the feasible set of a state j, $[j-i]$ is the state† that changes to a state j by the exercise with v^i, and $I_{[j]}$ is the set of index i such that $[j-i] \geq 0$. The 3rd and 4th constraints‡ are assigning the probability x_n^j to $x_n^{i,j}$, and The 5th constraint represents a transition probability from node n to node k.

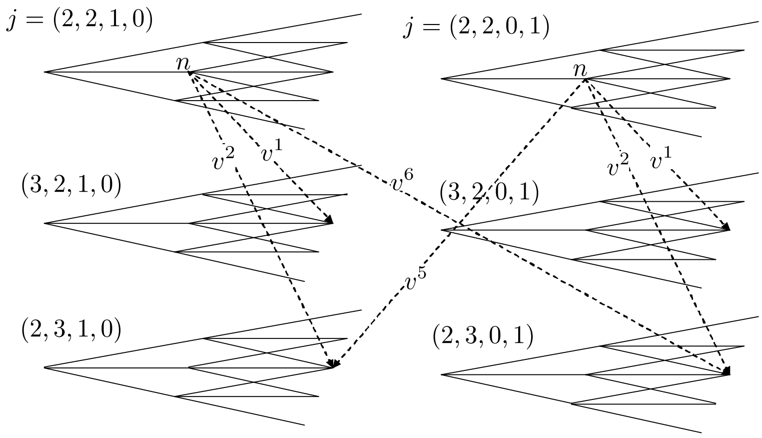

Figure 3: An example of Problem (6)

In the example, the option buyer at node n with states $j = (2,2,1,0)$ and $(2,2,0,1)$ chooses a changed volume from the set of changeable amounts. The choices are equivalent to assigning the probability at node n with the state j. Thus the choices determine the probability at nodes $C(n)$.

Problem (6) is not a linear program because of the equation $(*)$. However, Problem (6) is equivalent to a linear programming problem by the next theorem.

Theorem 2. *Let us consider the problem including the equation* $x_n^{i,j} \geq 0$ *in place of the equation* $(*)$ *in Problem (6), and call the problem as Problem A. Then Problem A has the same optimal value as Problem (6).*

Proof. In Problem A, the feasible set is convex because all constraints are linear. Then the extreme points of the feasible set are feasible points in Problem (6) because the extreme points necessarily satisfy the equation $(*)$ in Problem (6). Hence this problem has the same optimal value as Problem (6). □

4. Applying Theorem 1 to Other Methods

In Section 3, for the problem of pricing swing options on a tree, we show the presence of a particular solution such that there are at most seven kinds of changed amounts. A Markov process can be approximated as a tree, so that the particular solution actually exists and the parallel methods with Problem (6) when the price process is a Markov process. Then

†
$$[j-i] = \begin{cases} j - e_i \ (i \leq 4), \\ j - f_i \ (i \geq 5), \end{cases} \tag{7}$$

where $e_0 = (0,0,0,0)$, e_i is the ith unit vector, of which ith component is 1, $f_5 = (1,0,-1,1)$, and $f_6 = (0,1,1,-1)$.

‡The 4th constraint may be more flexible, in other words, $x_n^{i,j}$ may be more freely chosen, but the constraint is described as the equation $(*)$ for simplicity. This simplification is justified by Theorem 2.

we can apply Theorem 1 to other numerical pricing methods for improving computation time and accuracy.

4.1. Applying Theorem 1 to Jaillet et al. (2004)

In this section, we apply Theorem 1 to the method of Jaillet et al. [7].

Jaillet et al. [7] proposed a pricing method of swing options by dynamic programming approach. Their swing options are similar to ours. One difference is that their swing options allow $v_{\min}$ and $v_{\max}$ to be time-varying.

Their approach uses a multiple layer lattice. The lattice is distinguished by the number of residual rights and the sum of changed amounts. Their approach starts from the maturity date and works by backward induction.

To valuate a swing option, they discretize the changeable amount at each date. They limit the changeable amount to M kinds at even intervals, such as $1, 2, \ldots, M$. Then the total number of lattices is $\sum_{k=1}^{L} kM = \mathrm{O}(L^2M)$, and their pricing method has the time complexity $\mathrm{O}(NL^2M^2)$. When $v_{\min}$ and $v_{\max}$ are time-invariant, the discretization is not necessary from Theorem 1 and their method is refined as follows:

Theorem 3. *If $v_{\min}$ and $v_{\max}$ are time-invariant in the problem of Jaillet et al. [7], the time to solve the problem reduces $O(NL^2)$ from $O(NL^2M^2)$.*

Proof. In the case where $v_{\min}$ and $v_{\max}$ are time-invariant, the number of states is at most ${}_{L+2}\mathrm{C}_2 + 2 \cdot {}_{L+1}\mathrm{C}_2 = \mathrm{O}(L^2)$ from Proposition 1. Because a state j represents a combination of the number of residual rights and the sum of changed amounts, the number of lattices is also $\mathrm{O}(L^2)$. Furthermore, from Theorem 1 only seven kinds of the changeable amounts are necessary at each node. Thus seven times of a computation are necessary per node for backward induction. Then the time complexity is $N \cdot 7 \cdot \mathrm{O}(L^2) = \mathrm{O}(NL^2)$. □

Remark 2. *In option pricing, some dynamic programming approaches parallel mathematical programming approaches on scenario tree or lattice model. Actually, the method of Jaillet et al. [7] with Theorem 3, namely the method such that computation time is $O(NL^2)$, corresponds to our formulation in Equation (6).*

4.2. Applying Theorem 1 to the least-squares Monte-Carlo method

In this section, we apply Theorem 1 to a pricing method by the least-squares Monte-Carlo method.

Dörr [3], Meinshausen and Hambly [10], and Barrera-Esteve et al. [1] extended the least-squares Monte-Carlo method to swing options. Barrera-Esteve et al. [1] particularly focused on swing options with changeable amount. They considered the set of discrete admissible values of v_n and designed a pricing algorithm by the least-squares Monte-Carlo method. They defined the set of discrete admissible values as $\{v_{\min}, v_{\min} + \Delta v, \ldots, v_{\max} - \Delta v, v_{\max}\}$ where Δv is a positive value.

However, for our typical swing options, we can get explicit admissible values from Theorem 1. Thus we can perform the Monte-Carlo simulation faster and more accurately. Some numerical examples show the improvement.

Example 2. *We compare our method, which gives explicit admissible values, with that of Barrera-Esteve et al. [1]. Both methods are performed with 1000 paths per simulation and with six basis functions, and we set the price as the mean of 500 simulations.*

We assume that the underlying asset process $\{S_t\}$ is the following mean-reverting process:

$$\mathrm{d}X_t = -aX_t\mathrm{d}t + \sigma \mathrm{d}Z_t, \ S_t = S_0 \exp(X_t), \tag{8}$$

Table 1: Comparison of our method and Barrera-Esteve et al. ($V_{\max} = -V_{\min} = 30$)

	option price	standard error	computation time (second)
Barrera-Esteve et al.	257.49	0.17	6918
ours	257.40	0.18	2352

The simulation is performed on a computer with 2GHz CPU and 2GB memory.

Table 2: Comparison of our method and Barrera-Esteve et al. ($V_{\max} = 27.1$ and $V_{\min} = -25.25$)

	option price	standard error	computation time (second)
Barrera-Esteve et al.	233.85	-	16422
ours	232.96	0.16	2334

The simulation is performed on a computer with 2GHz CPU and 2GB memory. The standard error of Barrera-Esteve et al. [1] is blank because the option price of 233.85 is computed using linear interpolation.

where $S_0 = 100$, $X_0 = 0$, $a = 2$, *and* $\sigma = 0.1$. *For a swing option, we set parameters* $T = 20$, $\Delta t = 0.1$, $L = 15$, $K = 100$, *and* $v_{\max} = -v_{\min} = 4$. *For two kinds of global constraints we perform the simulation of the swing option pricing.*

First, we consider the global constraint of $V_{\max} = -V_{\min} = 30$. *In this case, the method of Barrera-Esteve et al. [1] with* $\Delta v = 2$ *estimates a true value, so that we compare both methods in terms of the computation time. Table 1 shows that our method is faster than that of Barrera-Esteve et al. [1].*

Second, we consider the global constraint of $V_{\max} = 27.1$ *and* $V_{\min} = -25.25$. *In this case, by setting of* $\Delta v = 0.05$ *the method of Barrera-Esteve et al. [1] estimates a true value. However,* $\Delta v = 0.05$ *is so small that the computational burden becomes high. We thus set* $\Delta v = 1$ *and evaluate the option price using linear interpolation, and then their method may estimate a biased value. Table 2 reports that our method is much faster than that of Barrera-Esteve et al. [1] because* Δv *is smaller than in the first case. In addition, because their method with* $\Delta v = 1$ *estimates a biased value, the option price of Barrera-Esteve et al. [1] is biased from ours and the method is more accurate than ours.*

5. Pricing in an Incomplete Market

5.1. Formulating the pricing problem in an incomplete market

In Sections 2, 3 and 4, we defined the price of a swing option as the expected value under the probability P. However, the probability P is not generally used in option pricing. Alternatively, the martingale probability Q that is equivalent to P is used. This pricing method is based on the arbitrage pricing theory.

Nevertheless, in some studies swing options are priced under the probability P. In energy markets the underlying asset cannot be preserved and cannot be used to hedge profits of an option, so that the definition of the martingale probability Q for the underlying asset is meaningless. However, in energy markets, futures contracts related to the underlying asset price are tradable. Thus hedging by the futures contracts allows us to define the pricing problem under the martingale probability Q for the futures contracts. If Q is unique, a market is complete; otherwise a market is incomplete.

When a market is complete, the pricing problem is obtained by replacing P by Q on Problem (6) and the price is unique. Taking the dual of Problem (6), we get the following problem:

$$\begin{array}{llll} \min\limits_{z} & z_0^{(0,0,0,0)} & & \\ \text{s.t.} & z_n^j - \sum\limits_{k\in C(n)} p_{nk} z_k^{[j+i]} \geq v^i(S_n - K_n) & (n \in \mathcal{N}_{T-},\ i \in I_j) & \\ & z_n^j \geq 0 & (n \in \mathcal{N}_T,\ |j| = L), & \end{array} \quad (9)$$

where $[j+i]$ is the state such that $[j+i]+[j-i] = 2j$. In Problem (9), z can be regarded as a contingent claim. In a complete market any contingent claim is replicatable; then, by substituting z into a portfolio of tradable assets, we can rewrite Problem (9) as the hedging problem. Let U denote the set of tradable assets and $U_n \in \mathbb{R}^{|U|}$ the set of prices of U at node n. The tradable assets include futures contracts and a risk-free asset. These prices are already discounted by the risk-free asset, and thus the risk-free asset price U_n^0 is equal to 1 for any n. Then the hedging form of Problem (9) is as follows:

$$\begin{array}{lrcll} \min\limits_{\theta} & U_0\theta_{0-}^{(0,0,0,0)} & & & \\ \text{s.t.} & U_n(\theta_{n-}^j - \theta_n^{[j+i]}) & \geq & v^i(S_n - K_n) & (n \in \mathcal{N}_{T-},\ i \in I_j) \\ & \theta_n^j & = & \theta_{k-}^j & (n \in \mathcal{N}_{T-},\ k \in C(n)) \\ & U_n\theta_{n-}^j & \geq & 0 & (n \in \mathcal{N}_T,\ |j| = L), \end{array} \quad (10)$$

where θ_n represents the holding amount of the tradable assets U at node n. Problem (10) has a similar form to the hedging problem of European and American options.

On the other hand, when a market is incomplete, the price is not unique. Let $\mathcal{Q}$ denote the set of the martingale probability Q. The price under $Q \in \mathcal{Q}$ is called the arbitrage-free price. In an incomplete market, an upper bound and a lower bound of arbitrage-free prices are important, so that we discuss these pricing problems. We obtain the pricing problems by adding the martingale condition to Problem (6). In this regard, the constraints for $x_n^{i,j}$ in Problem (6) are replaced by

$$\begin{array}{rcll} q_{nk}x_n^{i,j} & = & x_{nk}^{i,j} & (n \in \mathcal{N}_{T-}, k \in C(n),\ i \in I_j), \\ \sum\limits_{k\in C(n)} x_{nk}^{i,j}U_k & = & x_n^{i,j}U_n & (n \in \mathcal{N}_{T-},\ i \in I_j), \end{array} \quad (11)$$

where q_{nk} is an element of the probability Q. However, the constraint $q_{nk}x_n^{i,j} = x_{nk}^{i,j}$ is not necessary because the martingale condition (the second constraint) includes the constraint. We rewrite $x_{nk}^{i,j}$ to $y_{nk}^{i,j}$ for simplicity, and then the pricing problem of the upper bound is as follows:

$$
\begin{aligned}
\max_y \max_x \quad & \sum_{n\in\mathcal{N}_{T-}} \sum_{j\in J} \sum_{i\in I_j} v^i(S_n - K_n)x_n^{i,j} && \\
\text{s.t.} \quad & x_0^{(0,0,0,0)} = 1 && \\
& x_0^j = 0 && (j \in J\backslash\{(0,0,0,0)\}) \\
& x_n^j = \sum_{i\in I_j} x_n^{i,j} && (n \in \mathcal{N}_{T-}) \\
& \sum_{k\in C(n)} y_{nk}^{i,j} U_k = x_n^{i,j} U_n && (n \in \mathcal{N}_{T-},\ i \in I_j) \\
& x_n^j = \sum_{m\in B(n)} \sum_{i\in I_{[j]}} y_{mn}^{i,[j-i]} && (n \in \mathcal{N}\backslash\{0\}) \\
& y_n^{i,j} \geq 0 && (n \in \mathcal{N}_{T-},\ i \in I_j) \\
& x_n^{i,j} \geq 0 && (n \in \mathcal{N}_{T-},\ i \in I_j) \\
& x_n^j \geq 0 && (n \in \mathcal{N}_T,\ |j| = L) \\
& x_n^j = 0 && (n \in \mathcal{N}_T,\ |j| < L).
\end{aligned}
\tag{12}
$$

The upper bound problem is a linear programming problem, and easy to solve. On the other hand, the pricing problem of the lower bound is obtained by replacing $\max_y$ by $\min_y$ on Problem (12). However, the lower bound problem is a min-max programming problem, which is difficult to solve. We thus consider the dual of the lower bound problem:

$$
\begin{aligned}
\min_y \min_{z,\theta} \quad & z_0^{(0,0,0,0)} + \sum_{n\in\mathcal{N}_{T-}} \sum_{k\in C(n)} \sum_{j\in J} \sum_{i\in I_j} \left(-U_k\theta_n^{i,j} + z_k^{[j+i]}\right) y_{nk}^{i,j} && \\
\text{s.t.} \quad & z_n^j - U_n\theta_n^{i,j} \geq v^i(S_n - K_n) && (n \in \mathcal{N}_{T-},\ i \in I_j) \\
& \sum_{k\in C(n)} \sum_{i\in I_j} y_{nk}^{i,j} U_k = \sum_{m\in B(n)} \sum_{i\in I_{[j]}} y_{mn}^{i,[j-i]} U_n && (n \in \mathcal{N}_{T-}\backslash\{0\}) \\
& z_n^j \geq 0 && (n \in \mathcal{N}_T,\ |j| = L) \\
& y_n^{i,j} \geq 0 && (n \in \mathcal{N}_{T-},\ i \in I_j),
\end{aligned}
\tag{13}
$$

where $\theta \in \mathbb{R}^{|U|}$. This is a bilinear programming problem and generally easier to solve than a min-max programming problem. The second term of the objective function means the expectation value of additional borrowing (or lending) at each node. By comparing Problem (13) with Problem (10), or in other words, by comparing an incomplete market with a complete market, we verify that if y is unique then the second term of the objective function is equal to 0 in an optimal solution.

In Problem (12), all variables are local variables, namely, the constraints relate to a node, and then the variable x and y can be separately chosen at each time. Thus Problem (12) can be also solved by a backward algorithm[§]. The pricing algorithm is as follows:

1. Set $t = T - 1$.
2. At each node $n \in \mathcal{N}_t$ and in each state j, choose $x_n^{i,j} = x_n^{*,i,j}$ such that $\sum_{i\in I_j} x_n^{i,j} = 1$ and $x_n^{i,j}$ maximizes $\sum_{i\in I_j} v^i(S_n - K_n)x_n^{i,j}$. Put $x_n^{*,j} = \sum_{i\in I_j} v^i(S_n - K_n)x_n^{*,i,j}$.
3. If $t = 0$, then $\sum_{i\in I_j} x_0^{*,j}$ is the upper (lower) bound of arbitrage-free prices; otherwise set $t = t - 1$.
4. At each node $n \in \mathcal{N}_t$ and in each state j, choose $y_{nk}^{i,j} = y_{nk}^{*,i,j}$, where $k \in C(n)$, such that $\sum_{k\in C(n)} y_{nk}^{i,j} U_k = U_n$ and $y_{nk}^{i,j}$ maximizes (minimizes) $\sum_{k\in C(n)} y_{nk}^{i,j} x_k^{*,[i+j]}$. Set $\Phi_n^{*,i,j} =$

[§]The algorithm can be interpreted as the extension of Jaillet et al. [7].

$\sum_{k\in C(n)} y_{nk}^{*,i,j} x_k^{*,[i+j]}$.

5. At each node $n \in \mathcal{N}_t$ and in each state j, choose $x_n^{i,j} = x_n^{*,i,j}$ such that $\sum_{i\in I_j} x_n^{i,j} = 1$ and $x_n^{i,j}$ maximizes $\sum_{i\in I_j}(v^i(S_n - K_n) + \Phi_n^{*,i,j})x_n^{i,j}$. Put $x_n^{*,j} = \sum_{i\in I_j}(v^i(S_n - K_n) + \Phi_n^{*,i,j})x_n^{*,i,j}$. Return to Step 3.

This algorithm is sequential, so that when the problem size is large, we can save memory to solve by using this algorithm.

5.2. A numerical result

In this section, we give a numerical example of solving the upper and lower bounds of arbitrage-free prices of a swing option.

We first give the description of a swing option. We set $T = 20$, $L = 15$, $v_{\max} = -v_{\min} = 4$, $V_{\max} = 27.1$, and $V_{\min} = -25.25$. This setting is same as that in Section 4.2. The strike price K_n does not depend on nodes and the value K is given later.

Second we define a lattice. We use a trinomial tree like Jaillet et al. [7]. Figure 4 represents the trinomial tree. In each time there are three nodes, and each node can transit to any node in next time. We denote the asset prices of the upper, middle, and lower nodes by S^a, S^b, and S^c, respectively.

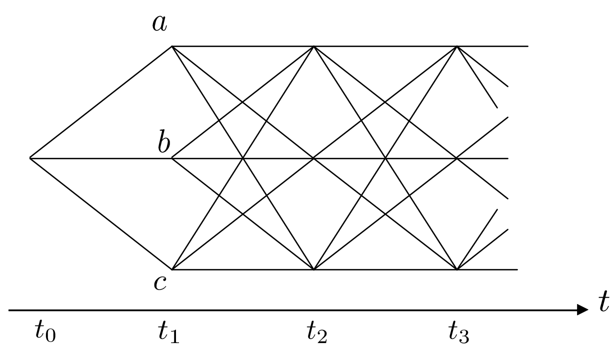

Figure 4: The trinomial tree in the numerical example

We give an the asset prices on the lattice as the approximation value of the following underlying asset process $\{S_t\}$ like Jaillet et al. [7]¶:

$$\mathrm{d}X_t = -aX_t\mathrm{d}t + \sigma\mathrm{d}Z_t, \tag{14}$$

$$S_t = S_{t_0}\exp(X_t), \tag{15}$$

where $S_{t_0} = 100$, $X_{t_0} = 0$, $a = 2$ and $\sigma = 0.1$. We also assume $t_0 = 0$ and the time step $\Delta t = 0.1$, so that $t_T = 2$. We set prices on the lattice in accordance with Hull and White [6], and then $S_t^a = 100\exp(\sigma\sqrt{3\Delta t}) = 105.63$, $S_t^b = 100$ and $S_t^c = 100\exp(-\sigma\sqrt{3\Delta t}) = 94.67$ for any t.

We assume that we can trade only a risk-free asset and a futures contract with the maturity date t_T and the risk-free rate is equal to 0. We set these as U. The price of the

¶The example does not mean the pricing under $\{S_t\}$ because $\{S_t\}$ is under a complete market. The example is completely numerical.

Table 3: The upper and lower bounds of arbitrage-free prices of the swing option

	upper bound	lower bound
$K = 102$	268.6	43.8
$K = 100$	262.3	2.2
$K = 98$	271.5	47.7
$K = 50$	1303.1	1139.7

We use CPLEX11.2 and a computer with 2GHz CPU and 2GB memory.

futures contract $F(t, t_T)$ is as follows[‖]:

$$\begin{aligned} F(t, t_T) &= \mathrm{E}_t[S_{t_T}] \\ &= S_t \exp\left(\exp(-a(t_T - t))X_t + \frac{\sigma^2}{4a}(1 - \exp(-2a(t_T - t)))\right). \end{aligned} \tag{16}$$

In the above settings, we solve the upper and lower bounds of arbitrage-free prices of the swing option. We choose some values as K and see the change of the price. Table 3 shows the result. The upper and lower bounds considerably differ in this example, because the variable y that corresponds to the martingale probability has high flexibility in the example. The difference is especially large at $K = 100$, since $S_n - K$ can take a positive and negative value and then the flexibility of y highly concerns the objective value.

6. Conclusion

In this paper, we have proposed a pricing method for swing options with typical constraints on a lattice model. Dealing with path-dependence of swing options, we find a particular solution of swing options in terms of changed amount. Using the solution, we have formulated the problem of pricing swing options as a linear program. This pricing method can naturally extend to the pricing in an incomplete market. We have formulated the problem of finding the upper and lower bounds of arbitrage-free prices as a linear program and a bilinear program, respectively. Moreover, we have shown a backward algorithm for finding the upper and lower bounds.

We have also applied the particular solution to some previous works for improving time complexity and accuracy and we have demonstrated these improvements in numerical examples. The constraints of our swing option are more limited than those of the previous works, but the constraints are typical, so that these improvements are expected to be useful of practical significance.

Acknowledgment

The author is very grateful to Kazuo Murota for giving constructive comments which lead to improvements in the presentation of the paper. This work was supported in part by Global COE Program "The research and training center for new development in mathematics", MEXT, Japan.

[‖]This pricing formula is actually incorrect in this case, because the underlying asset cannot be preserved and the pricing based on the arbitrage theory is meaningless. However, we do not need accurate prices and we only need an example of prices on the lattice, so that there is no problem.

References

[1] C. Barrera-Esteve, F. Bergeret, C. Dossal, E. Gobet, A. Meziou, R. Munos, and D. Reboul-Salze: Numerical methods for the pricing of swing options: A stochastic control approach. *Methodology and Computing in Applied Probability*, **8** (2006), 517–540.

[2] A. Camci and M.C. Pinar: Pricing american contingent claims by stochastic linear programming. *Optimization*, **58** (2009), 627–640.

[3] U. Dörr: Valuation of swing options and examination of exercise strategies by monte carlo techniques. Master Thesis, University of Oxford (2003), URL: http://www.maths.ox.ac.uk/filemanager/active?fid=918.

[4] H. follFöllmer and A. Schied: *Stochastic Finance: An Introduction in Discrete Time* (De Gruyter, Berlin, 2004).

[5] G. Haarbrücker and D. Kuhn: Valuation of electricity swing options by multistage stochastic programming. *Automatica*, **45** (2009), 889–899.

[6] J. Hull and A. White: Numerical procedures for implementing term structure models I: Single factor models. *Journal of Derivatives*, **2** (1994), 37–48.

[7] P. Jaillet, E.I. Ronn, and S. Tompaidis: Valuation of commodity-based swing options. *Management Science*, **50** (2004), 909–921.

[8] A. Lari-Lavassani, M. Simchi, and A. Ware: A discrete valuation of swing options. *Canadian Applied Mathematics Quarterly*, **9** (2001), 35–74.

[9] F.A. Longstaff and E.S. Schwartz: Valuing american options by simulation: A simple least-squares approach. *Review of Financial Studies*, **14** (2001), 113–147.

[10] N. Meinshausen and B.M. Hambly: Monte carlo methods for the valuation of multiple exercise options. *Mathematical Finance*, **14** (2004), 557–583.

[11] T. Pennanen and A. King: Arbitrage pricing of american contingent claims in incomplete markets —A convex optimization approach. Working Paper (2006), URL: http://math.tkk.fi/~teemu/american.pdf.

[12] S.M. Ross and Z. Zhu: On the structure of a swing contract's optimal value and optimal strategy. *Journal of Applied Probability*, **45** (2008), 1–15.

[13] M.C. Steinbach and H.-J. Vollebrecht: Efficient stochastic programming techniques for electricity swing options. In J. Kallrath, P. M. Pardalos, S. Rebennack, and M. Scheidt (eds.): *Optimization in the Energy Industry* (Springer, Berlin, 2009).

[14] A. Thompson: Valuation of path-dependent contingent claims with multiple exercise decisions over time: The case of take-or-pay. *Journal of Financial and Quantitative Analysis*, **30** (1995), 271–293.

Yusuke Tashiro
Department of Mathematical Informatics
Graduate School of Information Science and Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
E-mail: `yusu.tashi@gmail.com`