

An Introduction to U-Mart

Hajime KITA (National Institution for Academic Degrees, kita@niad.ac.jp)

May 24th., 2000

This document describes the outline of futures trade on U-Mart. For the details of U-Mart itself, please refer to "How to Participate in U-Mart Experiment System".

1 What is Futures Trade?

Futures trade is a trade to buy and sell goods at a certain point of time in the future (due date). The price of a futures good is determined as an actual market price (spot price) at the due date and is indeterminable before the date. This is why the futures markets exist. For example, you have bought a futures good at a YEN, and its spot price at the due date is b Yen. If $a < b$, you gain a profit from this futures transaction. If $a > b$, you suffer a loss from the transaction¹. In futures markets, it is also allowed to enter a reverse transaction before due date to make settlement with the balance between the futures buying price and futures selling price.

U-Mart deals in stock index futures. Stock index is commonly used to see the stock price level and is defined as a weight average (or a simple average) price of listed stocks. In stock index futures, stock index is considered as a price of a fictitious good and traded in the futures market. The actual good can not be obtained even at the due date because it is fictitious. Therefore, members of the market make settlements with the balance between buying or selling prices and the spot price at the due date. This method is called "closing out positions".

In U-Mart, a real stock index futures is traded fictitiously in a virtual market. It differs from real futures market in the following points²:

1. U-Mart allows futures trade only. Spot trade is not available.

¹ To be precise, we need to consider interest for managing the fund a YEN on markets. This explanation ignores interest for simplicity.

² When U-Mart is operated in parallel with real spot markets, it is possible in theory that the participants of U-Mart trade in real spot market or that traders in real spot market consult U-Mart, but it is not much practical.

2. The futures price formed in U-Mart is fictitious and does not affect real spot markets.

2 Trading Procedure in U-Mart

This section describes the trading procedure in U-Mart. U-Mart is designed based on actual futures market systems, but it is much simpler than the actual markets to avoid too much complexity. Figure 1 illustrates the procedure from placing orders through making settlements. The following sub-sections describes each process according to this flow.

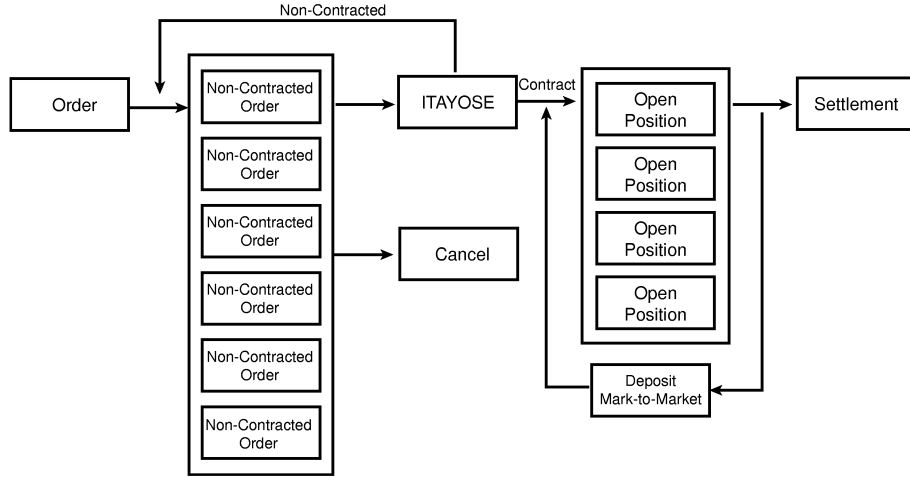


Figure 1: Trading Procedure in U-Mart

2.1 Participate in U-Mart

The members (trading programs) who participate in U-Mart are given a certain amount of fund at the beginning of experiments. Each member aims at making profit from this fund by trading in U-Mart.

2.2 Orders

Orders are placed by specifying the followings:

- brand (currently, a single brand),
- buying or selling,

- quantity, and
- price.

The price can be specified in the following two ways:

Market Order place an order at the price formed in the market, without regard to how much the price is, or

Limit Order place an order with a price limit.

- A buying order with a price limit p represents an order "to buy when price is not higher than p ; otherwise not to buy".
- A selling order with a price limit p represents an order "to sell when price is not lower than p ; otherwise not to sell".

2.3 Contracts

The exchange (U-Mart server) arranges contracts by matching buying and selling orders placed in the market.

Actual markets handle orders in different ways at or after opening the market.

Itayose : This method is used at opening a market. An exchange collects the orders accepted before opening the market, and contracts them at the prices which maximizes the trading volume.

Intraday : This method is used after opening a market. An exchange matches a new order with existing orders as soon as it is placed.

U-Mart repeats Itayose at regular intervals, which facilitates synchronization of transactions even when an experiment is conducted via network. Refer to Appendix A for the details of the algorithms to determine price, quantity, and orders to be contracted.

The contracted orders are called "open positions". A member has to settle the balance between buying and selling open positions ("position balance") at the due date. The overall status of open positions (buying or selling, and its quantity) of a member is simply called "position".

2.4 Margin and Mark-to-Market

Settlements of futures trades are made in the future on due date. Therefore, it is uncertain if a member can fulfill contracts at the date. To signify the capacity to fulfill contracts, members are required to deposit "margin" money (a certain percentage of trading amount) to the exchange.

A margin is determined based on position balance. Assuming that margin m is required per unit, and you have selling position $s(t)$ and buying position $b(t)$ at a certain time t , the margin to be deposited is calculated as:

$$|s(t) - b(t)| \times m$$

Since futures transactions take long time to be settled, carrying over losses up to the due date increases the default risk of contracts. To avoid this, the exchange evaluates members' positions everyday and pays or receives margin for the unrealized profit or loss. This process is called "Mark-to-Market".

In U-Mart, the mark-to-market is performed at the end of every virtual day, using closing price of the day to evaluate members' positions.

2.5 Settlements

Members need to settle their open positions on the due date. Open positions are evaluated with spot price (the actual price of the index at the due date) and settled at the balance between futures prices and the spot price.

2.6 Bankruptcy

A member goes into bankruptcy when he/she is unable to settle his/her position due to the short of fund, and is not allowed to make any more deal³. The exchange takes over the transactions of bankrupt members and make settlement on due date.

References

- [1] How to Participate in U-Mart Experiment System
- [2] Osaka Securities Exchange: A Guide for Stock Price Index Futures Trading (1999)

A Appendix: Itayose Algorithm of U-Mart

A.1 Introduction

U-Mart stores orders from participants for a certain period, then determines a price by Itayose algorithm and processes contracts of the stored orders. The highest computational

³ U-Mart has capability to loan to members up to a certain limit. In this explanation, it is omitted for simplicity.

efficiency is required for the algorithm to operate the server efficiently and to shorten the period of closing market. Making the algorithm definite enhances the market transparency and helps participants understand the U-Mart operation. This appendix shows the one implementation of the algorithm.

A.2 Basic Idea of Price/Quantity Determination

In U-Mart, an order O_i is represented as a set of price, quantity, buying/selling, and Itayose term (p_i, q_i, bs_i, t_i) . The price p_i in a buying order ($bs_i = \text{BUY} = 2$) or in a selling order ($bs_i = \text{SELL} = 1$) means the following, where the subscript i represents the order number:

- Buying order: buy quantity q_i when the price is not higher than p_i , or
- Selling order: sell quantity q_i when the price is not lower than p_i .

In here, a demand curve $B(p)$ and a supply curve $S(p)$ are defined as follows (see Figure 1):

$$B(p) = \sum_{i, bs_i=\text{BUY}, p_i \geq p} q_i = \sum_{i, bs_i=\text{BUY}} q_i - \sum_{i, bs_i=\text{BUY}, p_i < p} q_i \quad (1)$$

$$S(p) = \sum_{i, bs_i=\text{SELL}, p_i \leq p} q_i \quad (2)$$

The right side of the Formula (1) is useful as a method to calculate $B(p)$ successively when incrementing the price p from 0. The maximum quantity of contracts are achieved by determining contract price and quantity from the intersecting point of these curves. However, contract price generally is not determined uniquely because of the discrete variation of demand and supply curves. It also has possibility that some orders are only partially contracted (partial contract). Considering these aspects, some rules are required for defining which orders are contracted and how the price is determined. U-Mart complies with the following rules:

1. market orders are contracted prior to limit orders (however, when every buying and selling order is market order, the price can not be determined and no contract is made),
2. selling limit orders with lower prices are contracted before higher ones,
3. buying limit orders with higher prices are contracted before lower ones,
4. when the same type (buying/selling) of orders at the same limit price are placed, the one placed in earlier Itayose term gets priority,

5. orders get random priority when they are placed in the same Itayose term with the same price and type⁴ ,
6. a contract price is determined as a mean value of the following two values (see Figure 5):
 - the greater value between "the highest limit price of contracted selling orders" and "the highest limit price of non-contracted buying orders", and
 - the smaller value between "the lowest limit price of contracted buying orders" and "the lowest limit price of non-contracted selling orders",

and the fraction is rounded up^{5 6} .

Although an exceptional handling is required when every order is market order, the algorithm does not explicitly consider market orders because it can treat:

- selling market orders as selling limit orders at price 0, and
- buying market orders as buying limit orders at higher prices than the highest limit price.

A.3 Basic Idea of Algorithm

The algorithm is composed of the following processes (see Figure 3):

- align and search orders O_i in the lexicographic sequence in order type, price, selling/buying, and Itayose term (The order types are in sequence of selling market orders, limit orders, and buying market orders, price is in ascending sequence, selling/buying is in sequence of selling and buying, and Itayose term is in ascending for selling orders and descending for buying orders). And orders in the same Itayose term are align in random sequence,
- successively construct demand curves $B(p_i)$ and supply curves $S(p_i)$ at the price p_i for orders O_i ,
- while the demand curve is higher than the supply curve,

⁴ The problem is that this rule gives an advantage to someone who divide an order into two or more.

⁵ The "non-contracted order" is not detected when every buying order or every selling order is contracted. In this case, the highest (or the lowest) limit price of the other detected non-contracted orders is the contract price.

⁶ As shown in Figure 4, it is possible that non-contracted orders exist with the limit prices between "the highest limit price of contracted selling orders" and "the lowest limit price of contracted buying orders".

- O_i is not contracted if it is a buying order, or
- O_i is contracted if it is a selling order,
- when the demand curve and the supply curve intersect, partially contract order O_i ,
- while the demand curve is lower than the supply curve,
 - O_i is contracted if it is a buying order, or
 - O_i is not contracted if it is a selling order.

The search sequence is rather complicated because (see Figure 2):

- batch processing of buying and selling orders enables a simpler algorithm,
- however, the priority in prices are in reverse order on buying and selling orders, while both types of orders in earlier Itayose term get priority, and
- to maximize the contracted quantity, selling orders must be counted first to process buying and selling orders with the same limit price properly.

A.4 Implementation of Algorithm

A pseudo-code is shown below. It assumes that the orders are already aligned.

```

/* Initialization */
  Calculate the sum total of selling quantity, TB.

  B = TB;
  S = 0;
  MMAX = 0; /* Maximum contracted quantity */
  P_MAX_SELL = 0; /* Maximum limit price of contracted selling order */
  P_MIN_NOT_SELL = maxprice+1; /* Minimum limit price of non-contracted selling or
  P_MAX_NOT_BUY = 0; /* Maximum limit price of non-contracted buying order */
  P_MIN_BUY = maxprice+1; /* Minimum limit price of contracted buying order */

/* Search and Process of Each Order */
for (i=0;i<numorder;i++) {
  P = price[i];
  if (buysell[i]==BUY) { /* Process of buying orders */
                          /* Demand curve is updated after processing orders. */
    M = min(B,S);
  }
}

```

```

if (M <= B-quantity[i]) {
    contract[i] = NON-CONTRACTED;
    if (P_MAX_NOT_BUY < P) {
        P_MAX_NOT_BUY = P;
    }
} else if (M < B) {
    contract[i] = PARTIALLY-CONTRACTED;
    if (P_MIN_BUY > P) {
        P_MIN_BUY = P;
    }
} else {
    contract[i] = CONTRACTED;
    if (P_MIN_BUY > P) {
        P_MIN_BUY = P;
    }
}
B -= quantity[i]; /* Update demand curve. */
}
if (buysell[i]==SELL) { /* Process of selling orders */
    S += quantity[i]; /* Update supply curve. */
    M = min(B,S);
    if (M>MMAX) {
        MMAX = M;
    }
    if (M == S) {
        contract[i] = CONTRACTED;
        if (P_MAX_SELL < P) {
            P_MAX_SELL = P;
        }
    } else if (M > S - quantity[i]) {
        contract[i] = PARTIALLY-CONTRACTED;
        if (P_MAX_SELL < P) {
            P_MAX_SELL = P;
        }
    } else {
        contract[i] = NON-CONTRACTED;
        if (P_MIN_NOT_SELL > P) {
            P_MIN_NOT_SELL = P;
        }
    }
}

```


}
}

A.5 Computational Amount of Algorithm

The computational amount of this algorithm is estimated as: $2o(N) + o(N \log N) = o(N \log N)$, where:

number of orders: N ,

computational amount to calculate total quantity of buying orders: $o(N)$,

computational amount to align orders, with efficient alignment algorithm: $o(N \log N)$, and

computational amount to process contracts: $o(N)$.

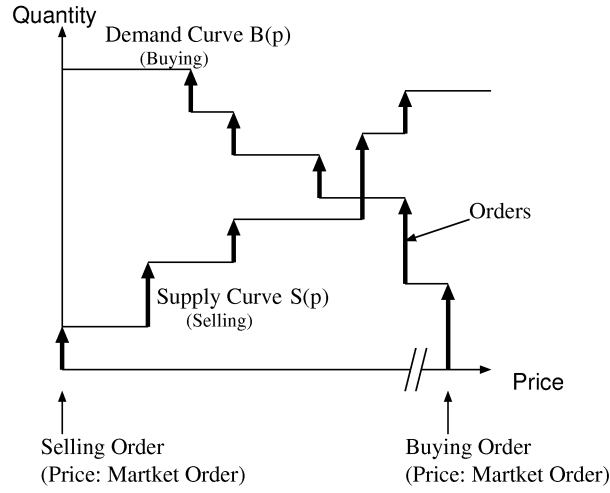


Figure 2: Demand and Supply Curves

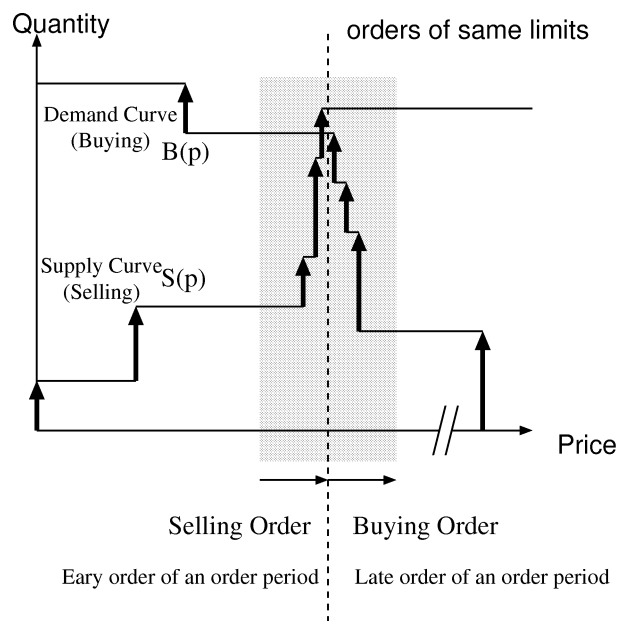
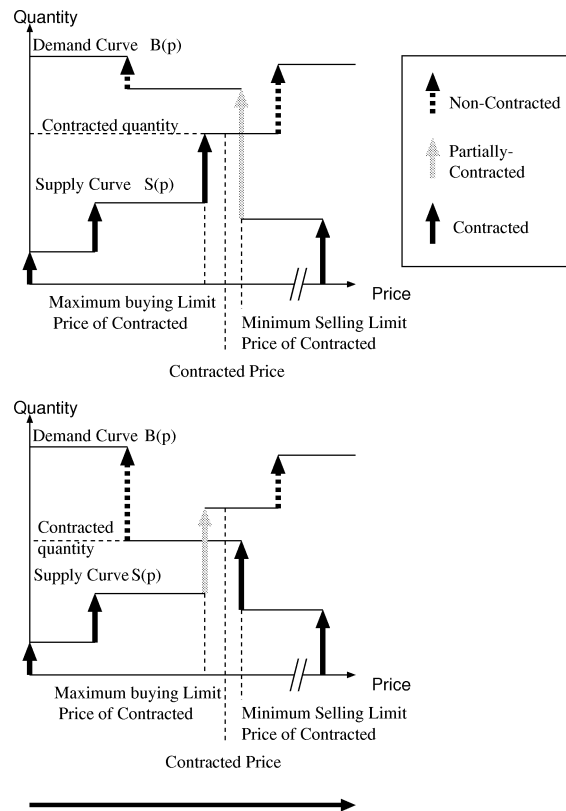


Figure 3: Orders with Same Limit Price



It can determine the contracted of selling order, and non-contracted of buying order as the cheap order of price

Figure 4: Determination of Contracts

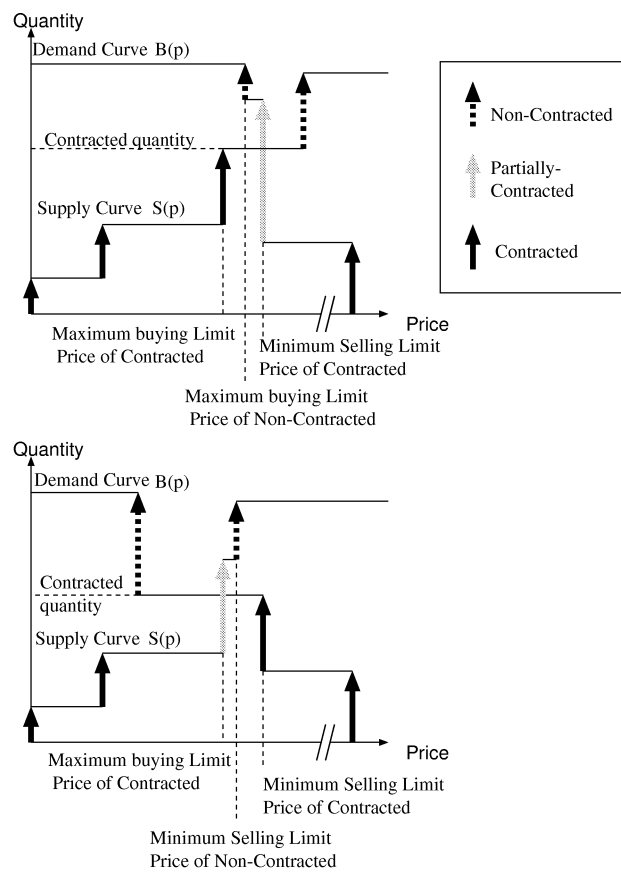


Figure 5: Relation between Contract Prices and Limit Prices

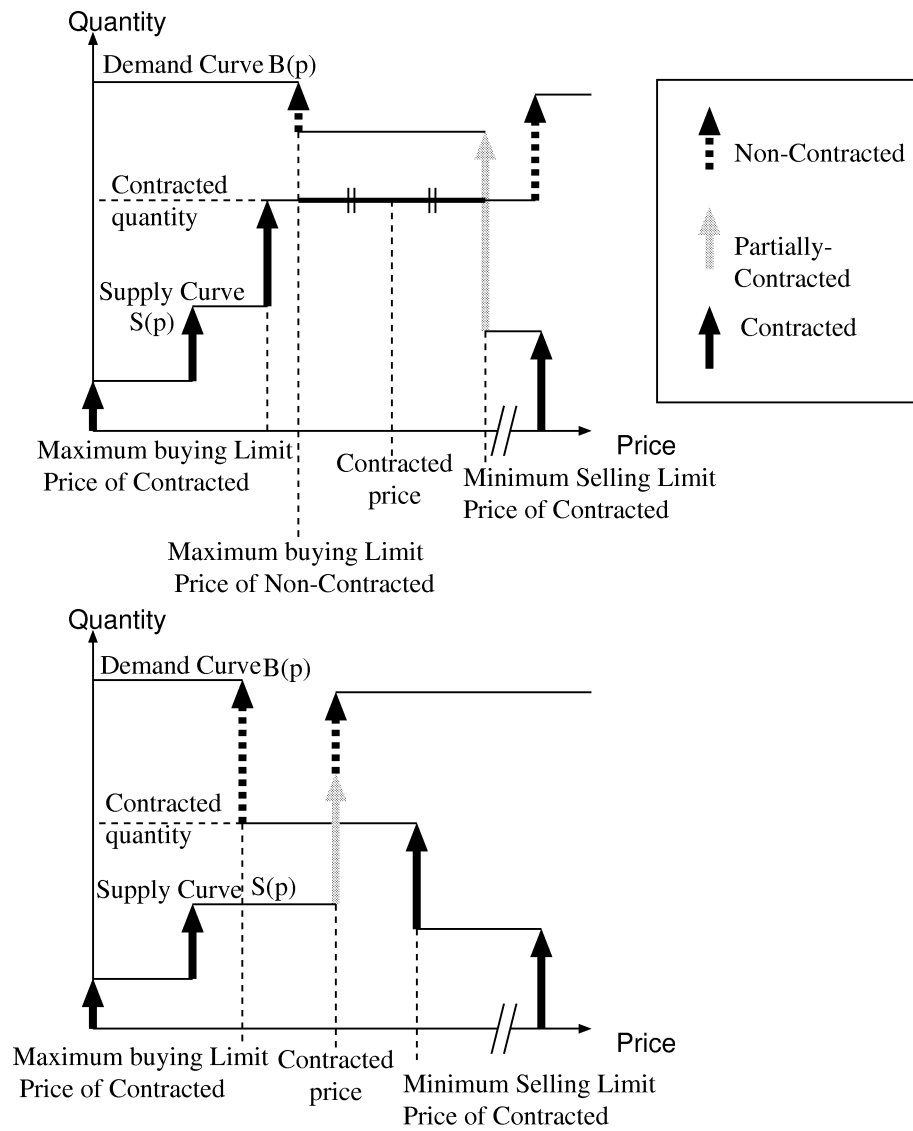


Figure 6: Contract Prices