DAOCROSS AND DAOSWAP

G.P. SAGGESE AND PAUL SMITH

Contents

1. Introduction

DaoCross and DaoSwap are smart contracts[1] that facilitate on-chain ERC-20[2] token exchange. DaoSwap provides a mechanism for efficient relative price discovery, and DaoCross enables parties to engage in zero market impact transactions with price improvement.

DaoSwap performs price discovery by matching supply and demand at regular time intervals. Supply and demand is expressed through a collection of buy and sell limit orders, and the clearing price is determined by a Walrasian-style auction ([12]).

DaoCross is a decentralized liquidity pool that crosses buy and sell orders with respect to an external reference price, i.e., a price oracle. This allows traders to interact directly with each other and share price improvement with respect to exchanges, e.g., by trading at bid-ask midpoint.

Date: April 4, 2023.
With contributions from Tamara Jordania, Samarth KaPatel, Grigorii Pomazkin, and Daniel Yachmenev.

[1] https://ethereum.org/en/developers/docs/smart-contracts/

[2] https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

Additionally, order intentions are not publicly disclosed prior to a cross, which enables block or other high-volume traders to execute quickly with minimal market impact.

DaoSwap and DaoCross differ in how the token exchange rate is determined, but both share several common advantages:

(1) Low Cost: they are a DeFi primitive that allows trading tokens peer-to-peer without incurring spread costs
(2) Capital Efficiency: participants provide liquidity only when they wish to trade, and only in the amount they wish to trade
(3) No Impermanent Loss: liquidity providers are not exposed to the risk of impermanent loss (a form of unrealized loss that becomes realized upon withdrawing liquidity), unlike the case with other decentralized exchange protocols based on automated market makers
(4) No Intermediary Custody: exchanged tokens always remain under the control of their respective owners
(5) No Rent Extraction from High-Frequency Traders: speed alone does not confer an advantage to traders, as the discrete timing prevents predatory tactics such as front-running, limit order scalping, and spoofing, which are known issues seen with continuous limit order books
(6) Ease-of-use: the interaction between buyers and sellers occurs through smart contracts, with a website front-end available

2. Matching liquidity

Suppose there are two parties, Alice an `ETH` holder and Bob a `wBTC` holder (wrapped bitcoin, an ERC-20 compliant coin that tracks bitcoin), who wish to = swap tokens at a competitive rate. Suppose also that there are many additional `ETH` and `wBTC` holders who may be interested in participating in a swap. How should the liquidity be pooled? At first glance, it appears that there should be a single pool of liquidity. The wrinkle manifests in determining how to express a desire to trade. A simple expression of a desire to trade is a limit order representing a commitment to trade up to q in quantity of a token at a token exchange rate up to p. But what if some parties express quantity in terms `ETH` and and some in terms of `wBTC`, and some parties express limit prices in terms of `ETH` per `wBTC`, while others express limit prices in terms of `wBTC` per `ETH`? Under these conditions, setting a single clearing price and determining fill prioritization rules raises several questions. We will consider and address these in the sequel, but to begin, we will discuss a more constrained setting.

2.1. **Breaking the symmetry.**

To simplify, we follow the practice of FX (foreign exchange) futures markets, which break the symmetry by specifying two non-interchangeable roles:

(1) a *base* currency (for quantity)
(2) a *quote* currency (for price)

The roles are expressed by writing *base currency/quote currency*, e.g., "EUR/USD".

A limit order then consists of a quantity expressed in the units of the base currency and a price expressed in terms of the quote currency (per unit or suitable multiple of base currency). See, for example, [6] (e.g., footnotes on page 25) for usage of this terminology. See [7] for how this works in practice for Euro FX contracts, where contract units are denominated in Euros and price is quoted in U.S. dollars and cents per Euro increment.

By breaking the symmetry, one may run a standard limit order book, hold a classical Walrasian-style auction, and handle size quantization effects by using one of a variety of priority rules based on quantity, price, and time. See [4] for discussion around priority rule variations and their effects on market quality outcomes.

An effect of breaking the symmetry in this way is that either liquidity for a pair of currencies may be expressed in only one form, or liquidity for a pair of currencies is split across two separate

contracts (with roles of base and quote token reversed), each with its own limit order book. Using our example, this would mean treating `wBTC/ETH` and `ETH/wBTC` as separate token pairs, even though the union of the underlying parties and sources of liquidity are the same.

2.2. **Basket case.** It seems plausible that by retaining the symmetry in the two-token case and instead treating a pair of tokens as a single source of liquidity, one may contrive a more efficient exchange mechanism.

This minor expansion, however, suggests widening even further the universe of eligible swaps. For example, if there are three tokens available for exchange, one could contemplate many-for-one, one-for-many, or many-for-many token swaps, all to be carried out in an atomic fashion, and perhaps with complex constraints. A toy case along these lines one may consider is as follows:

Problem 1 (Triangular liquidity)**.** Suppose there are the following three parties:

- Party A, who holds `wBTC` and wants `ETH`
- Party B, who holds `ETH` and wants `DAI`
- Party C, who holds `DAI` and wants `wBTC`

How should an auction be structured to "best" facilitate exchange?

While there may be other use cases and even demand for such auctions, unfortunately, the problem in general is NP-complete (e.g., [13]), which is a significant limiting factor in terms of feasibility and auction expense.

3. Protocol introduction

In the previous section, we discussed the notions of base currency and quote currency from foreign exchange. Here we extend them to tokens.

3.1. **Base/quote tokens.** A token (ordered) pair consists of a *base token* and a *quote token*. The shorthand notation for expressing the pair is *base token/quote token* (e.g., "`wBTC/ETH`"), and as such indicates the respective roles of the tokens.

By convention, *quantity* to exchange is expressed in terms of the base token, and *price* is a token exchange rate expressed in terms of the quote token per unit of base token (or multiple thereof).

Example 1 (base/quote tokens). By way of example, in the pair `wBTC/ETH`, `wBTC` is the base token and `ETH` is the quote token. Interest to trade is expressed in terms of orders with `wBTC` as base token and `ETH` as quote token, as follows:

- A "buy" order represents a commitment to purchase the base token `wBTC` and pay with the quote token `ETH`
- A "sell" order represents a commitment to sell the base token `wBTC`) and receive the quote token `ETH`.
- Quantity q is in terms of the base token ("buy/sell a certain amount of `wBTC`")
- Limit price is in terms of the quote token ("buy/sell `wBTC` with a limit price of p `ETH` per `wBTC`")

A party who places a "buy" order must have the quote token (`ETH`) in custody, i.e., in its wallet. A party who places a "sell" order must have the base token (`wBTC`) in custody.

3.2. **Order attributes and notation.** We introduce the following tuple notation for a general limit order (valid for both DaoCross and DaoSwap).

Definition 1 (Limit order)**.** A DaoCross or DaoSwap limit order is represented by a tuple of the form

(`timestamp`,
`action`,
`quantity`,
`base_token`,
`limit_price`,
`quote_token`,
`deposit_address`)

The quantities are arranged to make the order simple to read in natural language: "At timestamp `timestamp` create an order to `action` a `quantity` of the token `base_token` for a limit price of `limit_price` with respect to the token `quote_token` and deposit the resulting tokens at `deposit_address`."

We have previously discussed the roles of `action`, `quantity`, `base_token`, `limit_price`, and `quote_token`. The roles of `timestamp` include determining eligibility in a swap or cross and can extend to influencing order fill priority. The `deposit_address` attribute mirrors standard functionality available in traditional finance, e.g., in purchasing T-Bills on TreasuryDirect[3]. This feature facilitates account management through the use of multiple wallets. For example, a miner may have a supply of `BTC` collected and held in one wallet which it would like to systematically diversify into `ETH` and perhaps other tokens. In specifying a deposit address, it may use a separate wallet to collect incoming `ETH`.

For brevity, in the sequel we may:

- omit `timestamp` and `deposit_address` when not relevant to the discussion
- omit the (infinite) `limit_price` for market orders

Example 2 (Limit order notation)*.* The order

(`1678660406`, `buy`, `3.2`, `ETH`, `4.0`, `wBTC`, `0xdeadc0de`)

corresponds to the natural language description: "At timestamp `Mon Mar 13 2023 02:33:25 GMT+0000`, the user commits to `buy` up to `3.2` units of `ETH` in exchange for `wBTC` up to a `limit_price` of `4.0` `wBTC` per `ETH` with proceeds deposited at `0xdeadc0de`".

Next we introduce some examples of market order and limit order behavior when a clearing rate has been set. In particular, consider a swap for the tokens `ETH`, `wBTC` and assume that the exchange rate between `ETH` and `wBTC` is fixed at 0.2 (i.e., 0.2 `wBTC` can be exchanged for 1 `ETH` and vice versa).

Example 3 (Market order notation and clearing)*.* The following market orders omit `timestamp`, `limit_price`, and `deposit_address` in favor of emphasizing the amounts of tokens exchanged:

- An order `(buy, 1.0, ETH, wBTC)` means buying 1 `ETH` in exchange for 0.2 `wBTC`
- An order `(sell, 1.0, ETH, wBTC)` means selling 1 `ETH`, receiving the corresponding amount of 0.2 `wBTC`
- An order `(buy, 1.0, wBTC, ETH)` means buying 1 `wBTC`, paying with 5 `ETH`
- An order `(sell, 1.0, wBTC, ETH)` means exchanging 1 `wBTC` in return for 5 `ETH`

Next we consider the behavior of limit orders under the same prevailing exchange rate (and we assume that there is sufficient supply of tokens to fully fill the orders).

[3] https://www.treasurydirect.gov

Example 4 (Executable buy limit order). A limit order

`(buy, 1.0, ETH, 0.5, wBTC)`

means "`buy` up to 1 `ETH` in exchange for `wBTC` at a rate up to 0.5 `wBTC` per `ETH`." In this case, since the price of one `ETH` is equal to 0.2 `wBTC`, the order can be executed at the prevailing market rate.

Example 5 (Non-executable buy limit order). On the other hand, a limit order

`(buy, 1.0, ETH, 0.1, wBTC)`

requires that the rate of `wBTC` per `ETH` be lower than the current market rate, and so the limit price prevents a token swap from being carried out.

Example 6 (Non-executable sell limit order). A limit order

`(sell, 1.0, ETH, 0.5, wBTC)`

means "`sell`up to 1 unit of `ETH` in exchange for `wBTC` at a rate down to 0.5 `wBTC` per `ETH`." Since the current rate of `wBTC` per `ETH` is 0.2, which is below the limit price of 0.5, the order cannot be executed.

3.3. **Limit order as a set of linear inequalities.** Any limit order as defined above can be translated into inequalities involving quantity of exchanged tokens and token exchange rates. Multiple limit orders can be converted into a system of inequalities, which collectively constrain potential exchange outcomes.

In both DaoCross and DaoSwap, a swap between a given base/quote token pair occurs at a single exchange rate, i.e., the clearing price is the same for all executed orders. On the other hand, quantity exchanged is specific to each order and each order's constraint on quantity exchanged cannot be violated.

In the sequel we use an asterisk to denote clearing quantity and exchange rate:

- $q^*_{base}(o_i)$ denotes quantity of the base token exchanged in a swap from order o_i
- $p^*_{quote\_per\_base}(o_i)$ denotes exchange rate at

Note that

$$p^*_{quote\_per\_base}(o_i) = 1/p^*_{base\_per\_quote}(o_i)$$

An order of the form $o_1 =$ `(buy, 2, A, 3, B)` means "buy 2 units of token A in exchange for token B with a limit price up to 3 B per unit of A" and it corresponds to the following constraint on realized quantity exchanged and clearing exchange rate:

$$(p^*_{B\_per\_A}(o_1) \leq 3) \wedge (0 \leq q^*_A(o_1) \leq 2) \vee (q^*_A(o_1) = 0)$$

The constraint on the quantity exchanged is conditioned on the corresponding price satisfying the desired limit price constraint, otherwise the swap cannot be carried out and the exchanged quantity is 0.

In the same way, an order like $o_2 =$ `(sell, 3, A, 2, B)` means "sell 3 units of token A paying in token B with a limit price for one unit of A of at least 2 B" which corresponds to

$$(p^*_{B\_per\_A}(o_2) \geq 2) \wedge (0 \leq q^*_A(o_2) \leq 2) \vee (q^*_A(o_2) = 0)$$

A market order (i.e., without a limit price) has no constraint on exhange rate and thus is reduced to

$$0 \leq q^*_A(o_1) \leq 2$$

3.3.1. *Order equivalence.* Let

$$o_1 = (\texttt{buy}, q, A, p_{B\_per\_A}, B).$$

Suppose that $p^*_{B\_per\_A}$ is fixed and known. Then the order

$$o_2 = (\texttt{sell}, q', B, 1/p_{B\_per\_A}, A)$$

may be handled in order crossing in the same way that o_1 may be provided that

$$q' = q \cdot p^*_{B\_per\_A}$$

4. Reference clearing prices and prioritization

DaoCross relies on a *price oracle* for determining the effective token exchange rate in a cross. The price oracle may come from a lit exchange, centralized or decentralized, or even on-chain automated market makers such as Uniswap ([2, §2.2]).

The case of using an automated market maker as a price oracle is relatively straightforward, provided there is an automated market maker trading the target currency pair with sufficient liquidity.

To use an exchange as a price reference, we must consider some additional steps. First, exchanges typically use a dollar stablecoin (e.g., USDC, USDT) as the quote currency and all other currencies as base currencies. Our example of BTC/ETH would be considered a cross pair in the world of traditional finance. Because most cross pairs are not traded directly on exchanges, we must derive a clearing price from pairs involving stablecoins.

4.1. **Lit exchange bid-ask midpoint reference price.** Let $\text{bid}_{\text{BTC}}, \text{ask}_{\text{BTC}}$ be stablecoin-denominated top-of-book bid-ask prices for `wBTC` (e.g., expressed in `USDC`), and $\text{bid}_{\text{ETH}}, \text{ask}_{\text{ETH}}$ be the analogous prices for `ETH`. Then, a commitment to buy one `wBTC` and pay `ETH` would clear at a midpoint price of

$$\texttt{wBTC/ETH}_{\text{midpoint}} = \frac{\text{bid}_{\text{BTC}} + \text{ask}_{\text{BTC}}}{\text{bid}_{\text{ETH}} + \text{ask}_{\text{ETH}}}$$

expressed in BTC per ETH. Note that if the dollar price of BTH is significantly more than the dollar price of ETH, then this price implies paying many multiples of ETH for BTC (as is the case at the time of this writing).

In general, DaoCross may use the following reference price for a base/quote token pair, where the bids and asks come from a lit exchange and are expressed in terms of stablecoin prices:

$$\frac{\text{bid}_{\text{quote token}} + \text{ask}_{\text{quote token}}}{\text{bid}_{\text{base token}} + \text{ask}_{\text{base token}}} \tag{1}$$

When using either an on-chain price oracle or an exchange as a price oracle, it is possible to perform a time or volume weighted average of prices so as to avoid extreme price variations and to mitigate the risk and effects of any market manipulation attemps.

4.2. **Order crossing.** At regular time intervals, order submissions are cut off and reference prices are determined. An element of randomness is used to determine order cutoff times and reference price cutoff times in order to mitigate manipulative behaviors.

An order is eligible for matching if its timestamp is within the cutoff window and if the external reference price does not exceed its limit price.

Except on occasions where eligible buy/sell volume is perfectly matched, not all orders can be fully crossed. There are also discretization effects to consider arising from discrete order sizes and a discrete price grid [5][§3.2.1]. See [4] for a discussion of the trade-offs surrounding different prioritization rule choices. See [11] for the prioritization rules used by one of Morgan Stanley's equity liquidity pools.

4.2.1. *Priority rules.* DaoCross prioritizes fills according to:

- Volume (higher volume comes first in priority)
- Price (higher limit price breaks volume ties)
- Timestamp (earlier timestamp breaks ties in volume and price)

If, in the unlikely scenario that all three of these parameters perfectly agree, a certain priority is not guaranteed.

4.2.2. *Matching algorithm.* The mechanism for matching eligible buy and sell orders consists of two priority queues, one for eligible buy orders and one for eligible sell orders. Priority is determined according to the volume/price/timestamp priority rules introduced above. Top-of-queue orders are compared, and the lesser of the two volumes is fully filled. Once an order is fully filled at the established reference price. One an order is fully filled, it is removed from the priority queue. The procedure continues until one of the two priority queues is empty.

4.2.3. *Computational complexity.* Let n denote the sum (or max) of the number of eligible buy and sell orders. Priority queue construction occurs in $O(n)$ time, order removal costs $O(n \log n)$, and order remove occurs $O(n)$ times. So, the computational time complexity of the task is $O(n \log n)$. Memory requirements are $O(n)$.

4.2.4. *DaoSwap variant.* The mechanics for DaoSwap are similar to those above, with the primary difference being that an auction is used to determine a single clearing price. Variations in prioritization rules also possible.

5. Formulation of the general DaoSwap problem

The general problem of determining the token equilibrium price and the allocation among the swap participants can be formulated in terms of the inequalities on quantities and prices corresponding to the orders and on the need that the total quantity exchanged of each token be preserved across the swaps, i.e., the quantity bought for each token is equal to the quantity sold.

5.1. **Limit order inequalities.** Let $\{o_i\}_{i=1}^n$ be a set of limit orders, each of the form

$$o_i = (a_i, q_i, \texttt{base\_token}_i, p_i, \texttt{quote\_token}_i)$$

where the quote token is implicit in the quantity q_i and the quote and base tokens are implicit in the limit price p_i. To encode directionality in the limit price inequality, we introduce

$$\mathcal{I}(a) := \begin{cases} +1 \text{ for } a = \texttt{buy} \\ -1 \text{ for } a = \texttt{sell} \end{cases}$$

For each i, define

$$c_i := \left(p^*_{\texttt{quote\_token}_i \textit{-per-} \texttt{base\_token}_i} \leq \mathcal{I}(a_i) \cdot p_i \right)$$

where $p^*_{\texttt{quote\_token}_i \textit{-per-} \texttt{base\_token}_i}$ is the unique clearing price for the indicated base token/quote token ordered pair. TODO(Paul): revive the "to token" notation. Then, each limit order o_i may be expressed as

$$c_i \wedge (q_i^* \leq q_i) \vee (q_i^* = 0)$$

5.2. **No arbitrage.** Let T denote the intersection of the union of all base tokens and the union of all quote tokens:

$$T := (\cup_i \texttt{base\_token}_i) \cap (\cup_i \texttt{quote\_token}_i)$$

The set of tokens T is the set of tokens eligible for swapping. We require the following *no arbitrage* constraints be satisfied for all tokens $u, v, w \in T$ involved in a swap:

$$p^*_{u\_per\_v} \cdot p^*_{v\_per\_w} = p^*_{u\_per\_w}$$

Note that, in the case $w = u$, this enforces a unique exchange rate between a pair of tokens (changing the roles of the base and quote tokens inverts the exchange rate).

5.3. **Conservation of exchanged value.** For each order i, let $\tilde{q}^*_i$ denote the quantity of $\texttt{quote\_token}_i$ exchanged. Then we must have

$$\tilde{q}^*_i = q^*_i \cdot p^*_{\texttt{quote\_token}_i\_per\_\texttt{base\_token}_i}$$

for each i. In effect, this enforces the market clearing price for each order.

5.4. **Conservation of exchanged tokens.** Assume by convention that a buy order removes base tokens from the system and provides quote tokens to the system, whereas a sell order does the opposite.

For each token $u \in T$, define the indicator function $\mathcal{T} : T \to \{0, 1\}$ via

$$\mathcal{T}_u(v) = \begin{cases} 1 \text{ if } v = u \\ 0 \text{ if } v \neq u \end{cases}$$

Then, for each $u \in T$, the following conservation law must hold:

$$\sum_i \mathcal{T}_u(\texttt{base\_token}_i) \cdot q^*_i = -\sum_i \mathcal{T}_u(\texttt{quote\_token}_i) \cdot \tilde{q}^*_i$$

Note that this translates into one equality per token participating in the swap. Note also that this is a global constraint on the swap rather than local to each order; in particular, it expresses the notion that each filled order must have, across the collection of orders, suitable counterparties.

5.5. **Combining the constraints.** We now collect the constraints that must be satisfied:

$$\begin{cases} c_i \wedge (q^*_i \leq q_i) \vee (q^*_i = 0) & \forall i \in \{1, \ldots, n\} \\ \tilde{q}^*_i = q^*_i \cdot p^*_{\texttt{quote\_token}_i\_per\_\texttt{base\_token}_i} & \forall i \in \{1, \ldots, n\} \\ p^*_{u\_per\_v} \cdot p^*_{v\_per\_w} = p^*_{u\_per\_w} & \forall u, v, w \in T \\ \sum_{j=1}^n \mathcal{T}_u(\texttt{base\_token}_j) \cdot q^*_j = -\sum_{j=1}^n \mathcal{T}_u(\texttt{quote\_token}_j) \cdot \tilde{q}^*_j & \forall u \in T \end{cases} \tag{2}$$

5.6. **Solution of the general DaoSwap problem.** The general solution of the DaoSwap problem must satisfy a set of non-linear, combinatorial constraints, whose solution is NP-complete.

It can still be solved in an effective way with solvers like ...

TODO(gp): explain better, cite

5.7. **Reformulating the DaoCross problem.** In the DaoCross set-up, the price of the tokens involved in the swap is determined by an external oracle. This allows simplifying the general problem by applying the equivalence principle between orders and removing the non-linearity from the set of inequalities above.

In fact, the actual prices p^* compared to the limit price verifies or not the boolean condition c_i, which then allows specifying the quantity constraints:

$$\begin{cases} cond_i := to\_ineq(a(o_i))(p_{base}(o_i) \leq lp_{base}(o_i)p_{quote}(o_i)) \\ 0 \leq q_{base}(o_i) \leq q_{base} \text{ if } cond_i \\ q_{base}(o_i) = 0 \text{ if } \neg cond_i \end{cases} \tag{3}$$

The problem then becomes a set of linear inequalities that can be solved with various efficient methods (e.g., simplex-method).

TODO(gp): check

6. Comparison with automated market makers

Uniswap doesn't support limit orders - v3 introduced some steps towards incorporating - DaoSwap/Cross supports limit orders

Because liquidity providers and liquidity takers are in general different users operating at different time scales, Uniswap is affected by impermanent loss - DaoSwap/Cross is not affected by that

Uniswap requires multiple swaps and fees for arbitrary tokens - DaoSwap/Cross pools all the liquidity in a single optimization problem

References

[1] Hayden Adams, *Uniswap Whitepaper* (2018), available at `https://hackmd.io/s/HJ9jLsfTz`.

[2] Hayden Adams, Noah Zinsmeister, and Dan Robinson, *Uniswap v2 Core* (March 2020), available at `https://uniswap.org/whitepaper.pdf`.

[3] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson, *Uniswap v3 Core* (March 2021), available at `https://uniswap.org/whitepaper-v3.pdf`.

[4] Alejandro Bernales, Daniel Ladley, Evangelos Litos, and Marcela Valenzuela, *Alternative Execution Priority Rules in Dark Pools* (July 22, 2022), available at `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4169352`.

[5] Jean-Philippe Bouchaud, Julius Bonart, Jonathan Donier, and Martin Gould, *Trades, Quotes and Prices: Financial Markets Under the Microscope*, Cambridge University Press, 2018.

[6] CME Group, *2023 FX Product Guide*, available at `https://www.cmegroup.com/trading/fx/files/fx-product-guide-2023-us.pdf`.

[7] ———, *Euro FX Futures - Contract Specs*, available at `https://www.cmegroup.com/markets/fx/g10/euro-fx.contractSpecs.html`.

[8] Robin Hanson, *Combinatorial Information Market Design*, Information Systems Frontiers **5** (2003), 107-119, available at `https://doi.org/10.1023/A:1022058209073`.

[9] Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang, *Automated Market Making and Loss-Versus-Rebalancing* (2022), available at `https://doi.org/10.48550/arXiv.2208.06046`.

[10] Morgan Stanley, *Morgan Stanley Dark Pools*, available at `https://www.morganstanley.com/disclosures/morgan-stanley-dark-pools`.

[11] ———, *MS Pool ATS-N Filings*, available at `https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&filenum=013-00117`.

[12] *Walrasian auction*, available at `https://en.wikipedia.org/wiki/Walrasian_auction`.

[13] Mu Xia, Jan Stallaert, and Andrew B. Whinston, *Solving the combinatorial double auction problem*, Journal of Operation Research **164** (2005), 239-251, available at `https://www.sciencedirect.com/science/article/abs/pii/S0377221703008981`.