Linear System Solvability in the Virtual Money Exchange Rate Algorithm

Bingqing Shen, Jingzhi Guo and C. L. Philip Chen
Department of Computer Science and Information, Faculty of Science and Technology
University of Macau
Macao, China
{yb17412, jzguo, philipchen}@umac.mo

Abstract— **Virtual money exchange theory proposed in [6] is an important theory on finding a set of exchange rates between virtual currencies in rapidly-developed research area of virtual worlds. However, due to mathematical limitation of the employed algorithm, the equation solvability problem emerges in the computation of different virtual exchange rates. This paper provides a solution to identify the solvability problem in the virtual money exchange system and an algorithm to solve this problem by minimally removing unmatched supplies. Evaluation result shows that with this algorithm, the transaction histories can be maximally reserved for the sake of future exchange rate prediction.**

Keywords-VMX system; Virtual exchange rate algorithm; Exchange rate; Solvability; Directed cycle identification

I. INTRODUCTION

Virtual money exchange theory (VMX theory) proposed by Guo et al [6] is an important theory on finding a set of exchange rates between virtual currencies in rapidly-developed research area of virtual worlds. It proves the existence of a common value system and builds a virtual money exchange regime (VMX regime) to allow virtual wealth freely flow among virtual worlds in an efficient manner. Based on the common value system, a virtual exchange rate algorithm (VERA) is designed for implementing this VMX theory, which is a computational instrument calculating the exchange rates among a handful of virtual currencies. VMX theory and VERA is proved correct in [6] from economics perspective. Nevertheless, it has not looked into the mathematical level of some properties, that is, whether a solution is solvable.

VERA models the virtual currency exchange calculation into a linear system to calculate the exchange rates. Due to linear system's intrinsic properties, the equations in the virtual money exchange system (VMX system) are not always mathematically solvable. Based on the given supplies and demands of virtual currencies for exchange in the market, we may not able to find a solution for each exchange rate involved. If we cannot guarantee a deterministic equation solution in the VMX system, the practicability of VMX regime will be weakened.

In VERA, an exchange rate between any two currencies is not calculated as intuitively as only with the supply and demand between two involved currencies. Instead, the calculation requires equating the aggregated selling of one currency to buy other currencies with the aggregated buying of that currency by selling other currencies. This computation based on Pareto optimization is excellent to achieve exchange fairness, yet mathematically it is complex in determining when equations are solvable. For example, when we find a solution of the linear system is unsolvable at time (say, time slot t) of computing exchange requests, should we simply discard all the requests (time slot t) for this batch of computation and proceed to the next batch (time slot $t + 1$)? In this paper we find that there is still some space left in improving the existing virtual currency exchange rate calculation procedure.

This paper aims at examining the solvability problem appeared in VMX [6]. After investigation, it finds that the problem is causally related to the existence of unmatched supplies in supply relation. To generalize the problem, it modeled the problem as a directed cycle identification in a virtual money exchange graph (VMX graph). Through this modeling, an algorithm called unmatched supply removal algorithm (USRA) is proposed to identify and remove the unmatched supplies. This new algorithm can be safely integrated into VERA without affecting the original goal of fairness in VMX theory.

The rest of the paper is arranged as follows. Section II reviews VERA and models it in a directed cycle graph. Section III describes the newly discovered unmatched supply problem. Section IV provides some formal definitions for the new solution. Section V analyzes the exchange rate solvability. In section VI, the condition of unmatched supply problem is identified. Section VII, a resolution algorithm is designed and integrated into VMX systems. Section VII shows the correctness and evaluates the impact to market trend analysis with simulations. Finally, a conclusion is made and some contributions are enumerated.

II. RELATED WORK

A. VERA Algorithm

In VERA, the virtual money exchange rates can be computed through either sellers' exchange requests (i.e., sell-lead) or buyers' exchange requests (i.e., buy-lead) [6] without difference. In the sell-lead computing, one user makes an exchange requests by selling a certain amount of specified currency x (s_{xy}) for exchanging back some unknown amount of currency y (ud_{xy}), as formulated in (1), such that given any two currencies c_x and c_y, c_x trades back the units of c_y through the exchange rate e_{xy}. To remove ud_{yx}

and reduce the number of variables, a common virtual money c_0, called VONEY, is introduced as an intermediate virtual currency. Consequently, e_{x0} refers to a conceptual exchange rate from currency x to VONEY. This transforms (1) to (2).

$$ud_{yx} = s_{xy} \cdot e_{xy} \quad \text{(Sell-lead)} \tag{1}$$

$$e_{c_1 0} \sum_{x=c_1}^{c_n} s_{c_1 x} = \sum_{x=c_1}^{c_n} s_{xc_1} \cdot e_{xc_1} \quad \text{(Sell-lead)} \tag{2}$$

The expansion of (2) is (3), where each virtual world is assumed to have one virtual currency, such that all virtual currencies $C = \{c_1, c_2, ..., c_x, ..., c_y, ..., c_n\}$. (2) and (3) achieve the Pareto optimization and guarantee that all exchange rates e_{xy} are fair.

$$\begin{cases} e_{c_1 0} \sum_{x=c_1}^{c_n} s_{c_1 x} = \sum_{x=c_1}^{c_n} e_{x0} \cdot s_{xc_1} \\ e_{c_2 0} \sum_{x=c_1}^{c_n} s_{c_2 x} = \sum_{x=c_1}^{c_n} e_{x0} \cdot s_{xc_2} \\ \vdots \\ e_{c_4 0} \sum_{x=c_1}^{c_n} s_{c_1 x} = \sum_{x=c_1}^{c_n} e_{x0} \cdot s_{xc_4} \end{cases} \tag{3}$$

The computation of (3) can be converted to solving a linear system problem, shown in (4), by finding the solution of a matrix A, where all e_{x0} are variables that have to be solved with coefficients mapped to the elements of A. The linear equations in (3) are homogeneous, which implies that there are infinitely many solutions [7]. However, the actual rate we are looking for is a relative exchange rate between two virtual currencies, c_x and c_y, such that it can be obtained via $e_{xy} = e_{xo} / e_{yo}$. Thus, the value of e_{xo} never has to be computed.

$$A = \begin{bmatrix} \sum_{x=c_2}^{c_n} s_{c_1 x} & -s_{c_2 c_1} & \cdots & -s_{c_n c_1} \\ -s_{c_1 c_2} & \sum_{x=c_1}^{c_n} s_{c_2 x} & \cdots & -s_{c_n c_2} \\ \cdots & \cdots & \cdots & \cdots \\ -s_{c_1 c_2} & -s_{c_2 c_1} & \cdots & \sum_{x=c_1}^{c_{n-1}} s_{c_n x} \end{bmatrix} \tag{4}$$

For computational purpose, the system matrix A is transformed into the reduced row echelon form (RREF) A' as in (5). Each column i represents the intermediate exchange rate (e_{i0}) from currency i to *VONEX*. After transformation, each exchange rate can be easily calculated either by the relative value of two non-zero columns in the same row or by the transition between two exchange rates. For example, $e_{1n} = e_{10} / e_{n0} = 1 / b_1$, $e_{2n} = e_{20} / e_{n0} = 1 / b_2$, and then $e_{12} = e_{1n} / e_{2n} = b_2 / b_1$.

$$A' = \begin{bmatrix} 1 & 0 & \cdots & 0 & b_1 \\ 0 & 1 & \cdots & 0 & b_2 \\ & & \cdots & 0 & \cdots \\ 0 & 0 & \cdots & 1 & b_{n-1} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{5}$$

B. *Directed Cycle Identification*

The VMX system is modeled as a directed graph $G_E(\boldsymbol{V}, \boldsymbol{E})$ with currencies being the vertex set $\boldsymbol{V}$ and currency supplies being the edge set $\boldsymbol{E}$ (e.g., S_{xy}). With VMX graph, the solvability problem can be converted to a topological problem identifying directed cycles in G_E.

This topological problem can be addressed with at least two approaches. One approach models it as the problem of *enumerating all elementary circuits* in a directed graph. In particular, let G be a graph of n vertexes (v_1, v_2, ..., v_n). A *path* from vertex v_1 to vertex v_m contains a sequence of vertexes (v_1, v_3), (v_3, v_6), ..., (v_i, v_k), (v_k, v_m) in which the end of an edge is the start of the next edge. An *elementary path* contains no vertex at twice. If the start point and end point of a path is the same vertex, this path constitutes an *elementary circuit*. A body of work [13], [14], [15], [9] proposed different algorithms to enumerate all elementary circuits with a given graph.

Another alternative approach models the problem as *enumeration of all strongly connected components*. In a directed graph, a path containing the start and end which is the same vertex is called a *directed cycle*. A graph can be divided into arbitrary sub-graphs. If a sub-graph contains the maximal number of vertices which are all connected and form directed cycles, it is a *strongly connected component* [1]. The problem of finding all strongly connected components has been widely studied ranging from conventional computing to newly emergent parallel computing. Tarjan's algorithm [12] can accomplish the enumeration in O(|$\boldsymbol{V}$| + |$\boldsymbol{E}$|) with one stack. Kosaraju's algorithm [11] uses graph transpose to make the concept of the algorithm much simpler. This solution is useful in topological sorting, but increases the complexity in implementation. Path-based strong component algorithms [3], [2], [4] achieve linear time complexity constraint by employing two stacks, increasing space complexity. Recently, [8] and [10] seek solutions in parallel computing to achieve sub-linear time constraint.

Comparing the above two approaches, we prefer the second approach for the reason that it requires much less time complexity constraint than the elementary circuit-based approach. The algorithm in [12] is chosen for simplicity purpose.

III. THE SOLVABILITY PROBLEM

By observation, the solvability problem is related to the matching of supplies among all currency supplies. We use some examples to elaborate this problem. In the first example (Figure 1), currency C does not have any incoming supply. (6) represents the RREF of this exchange.

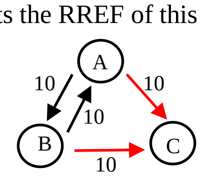


Figure 1. A Three Currencies Scenarios, scenario (a) and scenario (b)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (6)$$

The intermediate exchange rates from (6) are $e_{A0} = 0$, $e_{B0} = 0$, and e_{C0} arbitrary. Apparently, none of them is effective.

The above problem can be fixed by eliminating currency *C* in the exchange rate computation. However, the next examples collectively show a more complicated scenario by adding one more currency (currency *D*) into the exchange. Figure. 2 shows three possible VMX graphs. Likewise, the corresponding RREF matrices are in (7a), (7b), and (7c).

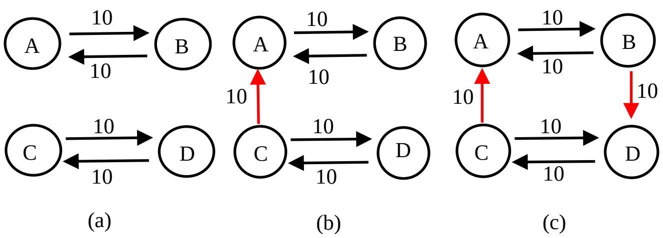


Figure. 2 Four Currencies Scenarios, Scenario (a), Scenario (b), and Scenario (c)

Observed from the RREF of the examples, the solvability of them differs related to the existence of supply S_{CA} and the supply S_{BD}. The exchange rate e_{AB} and e_{CD} can be derived from matrix (7a) while others, e_{AC}, e_{BC}, e_{AD}, e_{BD}, are not available. Once the supply S_{CA} is introduced (Figure. 2 (b)), (7b) shows that only the exchange rate e_{AB} can be solved while the intermediate exchange rates e_{C0} and e_{D0} are both equal to zero. After introducing S_{BD} (Figure. 2 (c)), surprisingly, all the exchange rates are available thought the opposite supplies S_{CA} and S_{DB} are not raised.

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7a)$$

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7b)$$

$$\begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1/2 \\ 0 & 0 & 1 & -1/2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7c)$$

We characterize the solvability problem as follows. Firstly, *orphan currency* (from a currency there exists only incoming or outgoing supply) is a sufficient condition leading to the solvability problem, but not a necessary condition. Secondly, *orphan supply* (from a currency pair there existing only one direction of supply) is a necessary condition but not sufficient to the solvability problem. Thirdly, removing orphan currencies can solve the unmatched supply problem caused by orphan currency. Fourthly, removing orphan supplies may not correctly solve the problem. The last property can be illustratively proved in Figure. 3 in which only the removal of orphan supply S_{DA} is a valid solution.

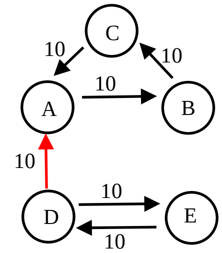


Figure. 3 A Five Currencies Scenario

IV. Formal Definitions

Before generalizing the solution, we restrict the problem scope by formally defining the system and the problem as preliminary work.

Definition 1. Exchange rate e_{xy} represents an exchange of selling *e* unit of currency *x* for currency *y*. Exchange rate e_{xy} has the following properties.

- *Duality Property*: If e_{xy} exits, e_{yx} must also exist;
- *Transitivity Property*: If e_{xa}, e_{xb}, ..., e_{iy} exist, then $e_{xy} = e_{xa} \cdot e_{ab} \cdot e_{bc} \cdot \ldots \cdot e_{iy}$. The ordered set $\{e_{xa}, e_{xb}, \ldots, e_{iy}\}$ from currency *x* to currency *y* is called a *chain*;
- *Reflexivity Property:* $e_{xx} = 1$;
- *Non-symmetry Property: $e_{xy} \neq e_{yx}$ and specifically,* $e_{xy} = 1 / e_{yx}$;
- *Non-negation Property*: $e_{xy} \geq 0$.

Definition 1' (*Basic Exchange Rate*). Exchange rate e_{x0} representing a rate from currency *x* to VONEY is called a basic exchange rate.

Definition 2 (Requested Exchange Rate). An exchange rate e_{xy} is called requested exchange rate if there exists a virtual currency supply from virtual currency *x* to virtual currency *y*.

Definition 2' (*Virtual Money Exchange Transaction or VMX Transaction*). All the requested exchange rates compose a virtual money exchange transaction or VMX transaction, denoted by ***R***.

Definition 3 (*Implicit Exchange Rate*). An exchange rate e_{xy} is called implicit exchange rate if there is no virtual currency supply from virtual currency *x* to virtual currency *y*. Implicit exchange rate set ***U*** contains all implicit exchange rates such that $\boldsymbol{R} \cap \boldsymbol{U} = \boldsymbol{\Phi}$.

Implicit exchange rate is identified and calculated with transition of requested exchange rates. For example, if requested exchange rates e_{ab}, e_{bc}, e_{cd} exist and there is no supply S_{ad}, then the implicit exchange rate e_{ad} is calculated through the chain $e_{ad} = e_{ab} \cdot e_{bc} \cdot e_{cd}$. If one requested exchange rate in the chain does not exist, neither does the implicit exchange rate.

Definition 4 (Exchange Rate Ensemble). Exchange rate ensemble is the union of requested exchange rates and implicit exchange rates: $\boldsymbol{E} = \boldsymbol{R} \cup \boldsymbol{U}$.

Definition 5 (*Zero Basic Exchange Rate*). e_{x0} is called zero basic exchange rate if $e_{x0} = 0$. $\boldsymbol{Z}$ denotes a set of zero basic exchange rates.

Definition 6 (*Free Basic Exchange Rate*). e_{x0} is a free basic exchange rate if e_{x0} has arbitrary value. $\boldsymbol{F}$ denotes a set of free exchange rates.

Definition 7 (*Solvable Exchange Rate*). An exchange rate e_{xy} is solvable if its basic exchange rates e_{x0} and e_{y0} are neither zero nor free. A solvable Exchange rate set $\boldsymbol{S}$ includes all the solvable exchange rates in a transaction.

Definition 8 (*Tangible VMX Transaction*). A virtual money exchange transaction is tangible if $\boldsymbol{R} \subset \boldsymbol{S}$.

Definition 9 (*Totally Solvable VMX Transaction*). A virtual money exchange transaction is totally solvable if $\boldsymbol{E} \subset \boldsymbol{S}$.

Definition 10 (*Solvability Problem*). There exists at least one requested exchange rate $e_{xy} \in \boldsymbol{R}$ with basic exchange rates e_{x0} and e_{y0}. If $(e_{x0} \in \boldsymbol{Z} \cup \mathbf{F}) \cup (e_{y0} \in \boldsymbol{Z} \cup \mathbf{F})$ returns true then the system matrix A is regarded as unsolvable.

Definition 10 restricts the solution scope with only tangible transactions which consist of requested exchanges In VMX graph, they are explicitly marked as directed edges. To solve the problem, firstly, we need identify the exchange solvable condition.

V. Exchange Rate Solvability Characterization

We model the virtual currency supply relation as directed graph to identify the solvablility condition. Formally, let $\boldsymbol{V}$ be the set of virtual currencies on the market for exchange, $|\boldsymbol{V}| > 1$, $\boldsymbol{E}$ be the set of supplies of any two virtual currencies. Then the VMX graph denoted by $G_E(\boldsymbol{V}, \boldsymbol{E})$ is a directed graph (or digraph) representing all supplies between different currency pairs. A directed cycle in G_E is a sequence of traversable edges such that the end of the last edge and the start of the first edge is the same vertex, namely $(v_1, v_2), (v_2, v_3), ..., (v_k, v_1)$ and $v_1, v_2, v_3, ..., v_k \in \boldsymbol{V}$.

Lemma 1. If a virtual currency exchange rate e_{xy} exists, there must exist a directed cycle between currency x and currency y.

The proof of *Lemma 1* can be easily deduced from the *duality property* in *Definition 1*, *Definition 2*, and the chained exchange rate calculation in *Definition 3*.

Definition 11 (*VMX Graph Strong Connectivity*). For any supply S_{xy} from currency x to currency y, $x \in \boldsymbol{V}$, $y \in \boldsymbol{V}$, there always exists a directed cycle which contains vertex x and vertex y, then the VMX graph G_E is *strongly connected*.

Based on *Lemma 1* and *Definition 11*, we can derive the following theorem to infer the total solvability of a VMX transaction.

Theorem 1. A virtual money exchange transaction is *totally solvable* if and only if G_E is strongly connected.

Most of the time *Theorem 1* is too strong to be practical, because the entire VMX graph is not always strongly connected. Fortunately, we only need a *tangible* solution according to *Definition 10*, which is a subset of the totally solvable solution. The tangible solution will be discussed in the next section.

VI. Unmatched Supply Identification

In a VMX graph, a *strongly connected component* [1] F_E is a sub-graph of G_E (denoted by $F_E \subset_s G_E$) It contains the maximal set of vertices from which any two vertices are strongly connected. A VMX graph G_E contains at least one strongly connected component. If a G_E is not strongly connected, it can be partitioned to two or more strongly connected components. According to the *reflexivity property* of in *Definition 1*, each vertex $v \in \boldsymbol{V}$ implicitly has an arc pointing to itself forming a directed cycle. Thereby, a strongly connected component can be as small as containing only one vertex. For any supply S_{xy}, if its vertex currencies, currency x and currency y, belong to the same directed cycle, then S_{xy} belongs to a single strongly connected component.

Theorem 2. In a VMX transaction $\boldsymbol{R}$, if and only if each supply S_{xy} belongs to a single strongly connected component, transaction $\boldsymbol{R}$ is free from solvability problems.

The approach to identify the solvability problem can be deduced from *Theorem 2* in the form of the following corollary.

Corollary 1. For a supply $S_{xy} \in \boldsymbol{V}$ in $G_E(\boldsymbol{V}, \boldsymbol{E})$. If $x \in \boldsymbol{V}_1$ in $F_{E1}(\boldsymbol{V}_1, \boldsymbol{E}_1)$, $y \in \boldsymbol{V}_2$ in $F_{E2}(\boldsymbol{V}_1, \boldsymbol{E}_1)$, $\boldsymbol{V}_1 \subset \boldsymbol{V}$, $\boldsymbol{V}_2 \subset \boldsymbol{V}$, and $\boldsymbol{V}_1 \neq \boldsymbol{V}_2$, then G_E will have a solvability problem and the supply S_{xy} for the request of $(S_{xy} \cdot e_{xy})$ is an *unmatched supply*.

Now the problem becomes how to identify and remove the unmatched supplies from a VMX graph.

VII. Solution and Algorithm Design

A. Unmatched Supply Removal Algorithm

With *Corollary 1*, we provide an algorithm called unmatched supply removal algorithm (USRA) to identify and remove all unmatched supplies with the following steps.

1) Firstly we check the connectiviy of the VMX graph. If all the edges are directly or transitively connected, then a totally solvable VMX transaction can be straightforwardly obtained based on *Theorem 1*.
2) If the VMX graph is not strongly connected, a set of strongly connected components $\boldsymbol{F}$ are identified from G_F. For a supply S_{xy}, its vertices in in G_E will be inspected whether they belongs to the same strongly connected component(*Corollary 1*). If they do not, S_{xy} will then be marked unmatched supply.
3) If a supply S_{xy} is marked unmatched supply, it will be removed from the VMX transaction $\boldsymbol{R}$.

The process will search and remove all unmatched supplies by enumerating all supplies in step (2) and step (3). The key point of algorithm design is to find an approach to identify all the strongly connected components in G_E. We employed Tanjan's algorithm [12] to achieve the objective.

B. Integration into Virtual Exchange Rate Algorithm (VERA)

Firstly, we study the impact of the unmatched supply removal on the original virtual exchange rate algorithm (VERA). Apparently if we remove the unmatched supplies at that point, the Pareto optimal has been shifted to a new point and the intrinsic value and extrinsic value are temporarily unequal [6]. In this case, the redistribution has to be applied once more by seeking a new set of exchange rates reflecting the new Pareto optimal point. It is evident that the unmatched supply removal before one Pareto optimal point search yields the same result as plugging USRA between two Pareto optimal point searches. (The proof can be found in [5].) We can then integrate USRA as a preprocess of VERA. Figure. 4 illustrates the algorithm integration in flowchart.

After calculating the requested exchange rates, the implicit exchange rates can be calculated subsequently. Calculating implicit exchange rates extends the tangible solution to the totally solvable solution as close as possible. The implicit exchange rates can be transitively calculated with requested exchange rates as formulated in *Definition 3*.

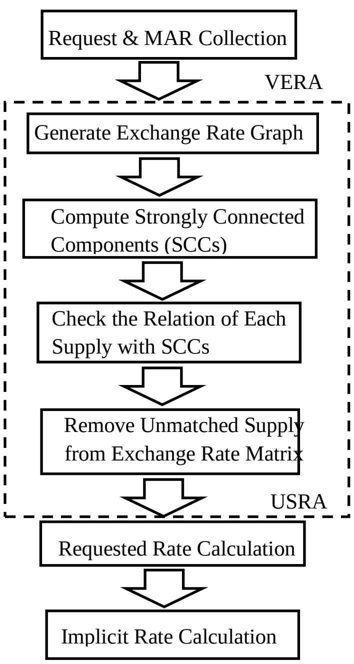


Figure. 4 Algorithm Integration Flowchart

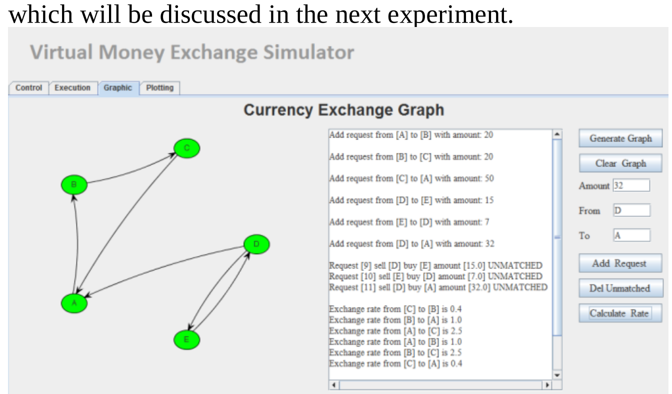
VIII. Experiments and Evaluations

To simulate USRA, we modified our virtual money exchange simulator (VMX simulator) [16] by following the flow in Figure. 4. For comparison, computations will abort as long as a zero basic exchange rate or an arbitrary basic exchange rate occurs, and all the supplies at this transaction will be discarded in the original VERA implementation. Apparently without USRA, lots of exchanges will be discarded and fewer history data can be collected.

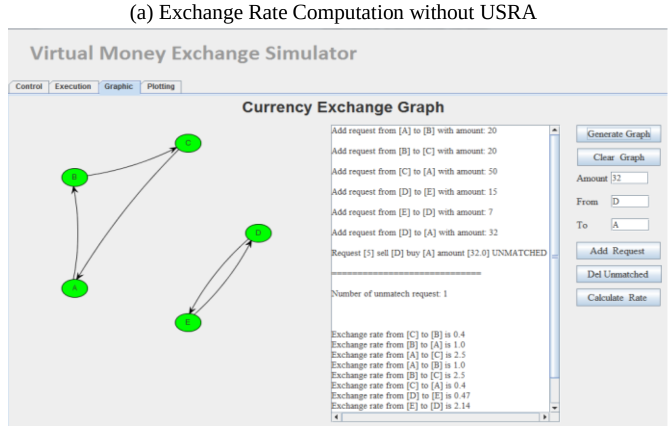
A. Single Exchange Rate Computation

This experiment validates the implementation of USRA in our VMX simulator. Figure. 5 shows a virtual currency exchange process without USRA (Figure. 5(a)) and with USRA (Figure. 5(b)). In this experiment, we deliberately generated requests with 6 supplies and different amount in each supply. In the exchange, supply S_{DA} is identified as an unmatched supply. In Figure. 5(a), only 6 exchange rates were obtained. Exchange rates e_{DE} and e_{ED} were not obtained due to solvability problem. Once the unmatched supply S_{DA} was removed, the number of obtained exchange rates is increased up to 8 (Figure. 5(b)).

This experiment not only verifies the correctness of USRA, but also implies that with USRA exchange rates can be maximally retrieved after removing unmatched supplies. The result set includes requested exchange rates as well as implicit exchange rates, enriching the market trend analysis which will be discussed in the next experiment.



(a) Exchange Rate Computation without USRA



(b) Exchange Rate Computation with USRA

Figure. 5 Unmatched Supply Removal Demonstrations in VMX Simulator

B. Influence to Market Trend

In this experiment, a series of virtual currency exchange requests are consecutively generated and computed within a given period. This experiment studies the influence of problem solving on transaction histories collected for market trend analysis. It was also be run in our VMX simulator [16].

TABLE 1 lists the experiment parameter. Totally 5 virtual currencies are involved throughout the experiments. The average number of request is around 10 in each time slot. This experiment will last 100 time slots. In each time slot, requests are randomly generated and then aggregated by the supply currency and demand currency. At the end of each time slot, the exchange rates will then be calculated with all the requests generated in the same time slot.

TABLE 1 SIMULTION PARAMETERS

Name	*Value*
Number of Currencies	5
Average Number of Request per Timeslot	10
Simulation Duration (Timeslots)	100

The experiment result is plotted in the market trend chart. We only illustrate the market trend of one currency pair (FBC – TWG), as shown in Figure. 6. Otherwise it will be messy if all plots are shown in one chart. TABLE 2 shows the statistic results of the experiment. (Currencies are arbitrarily denominated due to irrelevance to the experiment result.)

The statistic result shows that VERA can provide more historical exchange rates by integrating USRA. This result can be found in all the exchanges in TABLE 2. In this experiment, the test case with USRA has 1.24 times the number of history data in the case without USRA. We also found that this multiple monotonically increases in the number of currencies, which is not shown here. Since the future exchange rate prediction heavily relies on history market trend, we conclude that unmatched supplies removal can positively influence the prediction accuracy.

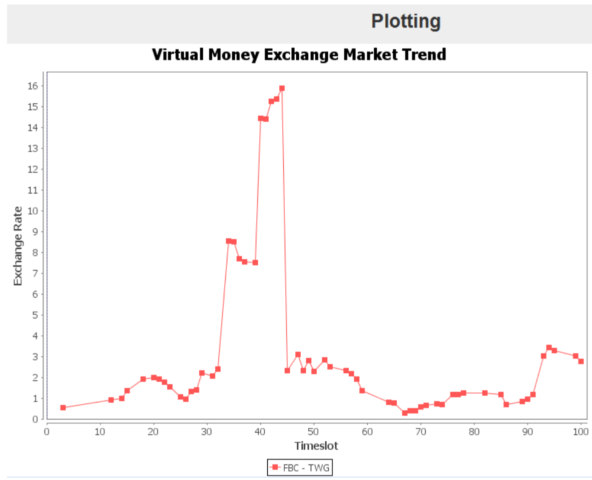
TABLE 2 SIMULTION RESULT

Currency Pair	*No. of Exchange without USRA*	*No. of Exchange with USRA*
WOG-QQB	50	51
FBC-WOG	53	56
FBC-LLD	49	58
FBC-QQB	47	55
WOG-LLD	51	56
LLD-QQB	52	56
WOG-TWG	52	53
FBC-TWG	48	60
LLD-TWG	44	57
QQB-TWG	45	55
Average	49.1	55.7
Ave. no. of matched request	7.65	9.45

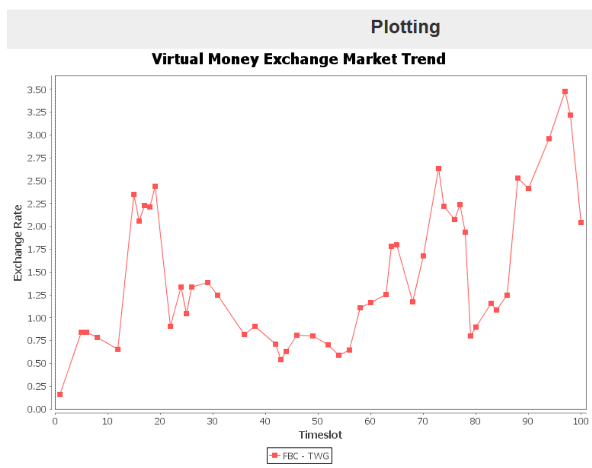
IX. CONCLUSION

This paper identifies the solvability problem in virtual currency exchange rate calculation which may prevent us from acquiring effective exchange rates. To completely solve this problem, an algorithm is designed based on unmatched supply removal. We integrate this algorithm into the original exchange rate calculation process and implement it in our simulator for evaluation.

Our work has the following contributions to the virtual currency exchange computation. First, the general condition of the solvability problem is identified with graph theory. Then, the problem is solved without changing the fairness property in exchange rate computation. With problem resolution, more exchange rates will be available in each transaction, enriching transaction histories. This has a positive influence on market trend analysis and future exchange rate prediction.



(a) Market Trend Chart without USRA, 48 Exchanges



(b) Market Trend Chart with USRA, 60 Exchanges

Figure. 6 Currency FBC to Currency TWG Market Trend Comparison

ACKNOWLEDGEMENT

This research is partially supported by the University of Macau Research Grant No. MYRG156(Y2-L2)-FST11-GJZ.

REFERENCE

[1] R. A. Brualdi, H. J. Ryser, Combinatorial matrix theory, Cambridge University Press, 1991.

[2] J. Cheriyan, K. Mehlhorn. "Algorithms for dense graphs and networks on the random access computer," Algorithmica vol. 15, no. 6, 1996, pp. 521-549.

[3] E. W. Dijkstra, E. W. Dijkstra, E. W. Dijkstra, and E. W.Dijkstra, A discipline of programming, vol. 1. Englewood Cliffs: prentice-hall, 1976.

[4] H. N. Gabow, "Path-based depth-first search for strong and biconnected components," Information Processing Letters vol 74, no. 3, 2000, pp. 107-114.

[5] M. Geilen, T. Basten, B. Theelen, and R. Otten. "An algebra of Pareto points," Fundamenta Informaticae, vol. 78, no. 1, 2007, pp. 35-74.

[6] J. Guo, A. Chow, and R.T. Wigand, "Virtual wealth protection through virtual money exchange," Electronic Commerce Research and Applications, vol. 10, no. 3, 2011, pp. 313-330.

[7] J Hefferon. Linear Algebra, released under the GNU Free Documentation License.

[8] S. J. Henstra, "An intuitive algorithm for finding strongly connected components," http://shenstra.nl/papers/2011-05-17%20Strongly%20Connected%20Components.pdf (2011), accessed 20 Feb. 2014.

[9] D. B. Johnson, "Finding all the elementary circuits of a directed graph," SIAM Journal on Computing vol. 4, no. 1, 1975, pp. 77-84.

[10] W. Schudy, "Finding strongly connected components in parallel using O (log 2 n) reachability queries," Proc. the twentieth annual symposium on Parallelism in algorithms and architectures (SPAA'07), Jun. 2008, ACM New York, NY, USA, pp. 146-151.

[11] M. Sharir, "A strong-connectivity algorithm and its applications in data flow analysis," Computers & Mathematics with Applications, vol. 7, no. 1, 1981, pp. 67-72.

[12] R. Tarjan, "Depth-first search and linear graph algorithms," SIAM journal on computing vol. 1, no. 2, 1972, pp. 146-160.

[13] J.C. Tiernan "An efficient search algorithm to find the elementary circuits of a graph," Communications of the ACM, vol. 13, no. 12, 1970, pp. 722-726.

[14] H. Weinblatt, "A new search algorithm for finding the simple cycles of a finite directed graph," Journal of the ACM (JACM), vol. 19, no. 1, 1972, pp. 43-56.

[15] R. Tarjan, "Enumeration of the elementary circuits of a directed graph," SIAM Journal on Computing, vol. 2, no. 3, 1973, pp. 211-216.

[16] J. Guo, B. Shen, C. L. P. Chen, "Augmented Simulator to Achieve a Smooth and Satisfactory Virtual Money Exchange Rate," unpublished.