Combinatorial Optimization for Self Contained Blockchain: An Example of Useful Synergy

Presented by: Milan Todorović

Mathematical Institute of the Serbian Academy of Sciences and Arts

Seminar Decision making - Theory, technology and practice
Oct. 07, 2021

Acknowledgements

Authors: Dušan Ramljak (PSU), Tatjana Davidović, Dragan Urošević, Tatjana Jakšić Krüger, Luka Matijević, Milan Todorović, Đorđe Jovanović (MISANU)

- Grant AI4TrustBC: Advanced Artificial Intelligence Techniques for Analysis and Design of System Components Based on Trustworthy BlockChain Technology
- Miloš Simić, University of Belgrade
- Nouf Albarakati, Temple University
- West Texas A&M University, College of Engineering

Preliminary version presented at SYM-OP-IS 2021.

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Overview

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Blockchain (BC) overview

- Represents a special type of distributed data storage (ledger)
- BC types:
  - Public (permissionless)
  - Private (permissioned)
- Users (nodes, agents, participants, entities) can have different roles
- Applications: cryptocurrency, smart contracts, IoT, voting, DRM, healthcare, etc.

BC maintenance

- Issues:
  - Security of data and users (Majority attack, DDoS, Eclipse attack, etc.)
  - Users' privacy (data is public, user's identity is hidden)
  - Data consistency
  - Wasteful usage of resources
- Our focus: energy efficiency and deanonymization
- We propose a self contained BC based on:
  - Proof-of-Useful-Work concept
  - Community detection in directed graphs

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

BC background

- Autonomous (unsupervised) append-only distributed data storage
- Autonomous: the removal of external authority (third party)
- Data is submitted in a form of (append-only) transactions
- Consensus protocol controls transaction additions
- Distributed: each participant has a copy of the whole database
- Data is stored in form of blocks of transactions

Block structure

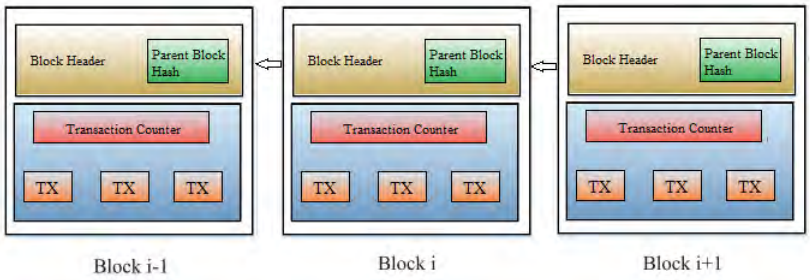
- Data is stored in form of blocks of transactions
- Blocks consist of header and body
- Header includes:
  - Block number
  - Hash value of the previous block
  - Timestamp
  - Validity code (nonce), etc.
- Body contains transactions data

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

An illustration of BC



Procedure of appending a block

- Transactions are submitted by users to transaction pool
- Special users of the BC (miners):
  - Form a block by selecting transactions from the pool
  - Execute consensus protocol
  - Publish the new block
- Block validity is checked
- Valid blocks are appended to BC
- Potential forks are resolved periodically
- Successful miners receive reward

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Consensus protocol

- Enables verification of transactions and blocks
- Eliminates the central authority
- Involves certain overhead
- Accepts or rejects each transaction from the pool
- Must be followed by all users of BC
- Ensures the agreement of all users on a common set of valid blocks

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Types of consensus protocols

- Proof-of-Work (PoW) - inversion of hash function
- Proof-of-Stake (PoS) - cryptocurrency amount guarantees trustworthiness
- Proof-of-Useful-Work (PoUW) - swaps PoW task with real-world problems
- Other consensus algorithms:
    - Proof-of-Capacity
    - Proof-of-Elapsed-Time
    - Proof-of-Burn, etc.

Proof-of-Useful-Work (PoUW)

- Proof-of-Useful-Work (PoUW) is a novel BC consensus protocol
- PoUW supports BC and contributes to solving real-world problems
- PoUW:
  - Improves efficiency of BC
  - Preserves security of BC provided by PoW
  - Provides users' equity missing in PoS
- Two sources of reward for miners:
  - Mining a block
  - Solving a problem instance
- Combination of PoW and PoUW (hybrid)

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

PoUW Optimization Insights

- PoUW involves optimization problems:
  - Travelling Salesman Problem
  - Artificial Neural Network training
  - Medical Image Processing, etc.
- Optimization problems can be addressed by:
  - Exact methods (Branch-and-... , CPLEX, Gurobi, etc)
  - Problem-specific heuristics
  - Metaheuristics (VNS, BCO, etc.)

- Our idea: Useful work of self contained BC involves its maintenance

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Pseudoanonymity vs. Deanonymization

- Pseudoanonymity: pseudonyms are known, identities should be hidden
- Transactions are made via e-wallet
- Users can generate a new e-wallet public key for every transaction
- Only public key of e-wallet available for everyone to see
- Anonymity is desirable for users but may hide malicious behaviour
- Deanonymization involves:
  - Defining similarity measures
  - Identifying patterns
  - Grouping transactions
- Clustering problem can model Deanonymization process

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Clustering problem

- Helps to discover patterns in observed data
- Cluster analysis determines the quality of clustering
  - Could be incorporated in method (aglomerative, hierarchical, etc.)
  - Done separately (centroid-based, distribution-based, density-based, etc.)
- In clustering of graph structured data similarity depends on:
  - Graph topology
  - Location of the nodes
  - Weights on nodes or edges, etc.

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

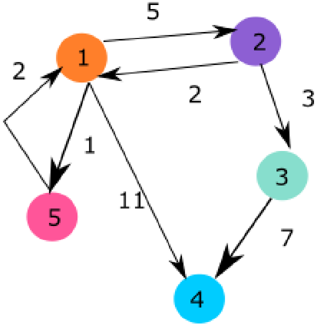
Community detection

- Community detection in case of graph-based clustering
  - Incorporates cluster analysis
  - Reveals underlying community structure
- Patterns of community detection involve
  - More interaction among nodes in the same community
  - Less interaction between the nodes that are in different communities
  - Edge directions contain useful information (e.g. sender/receiver)
- Set of BC transactions can be modeled as directed graph
- Community detection in directed graphs could be useful part of PoUW

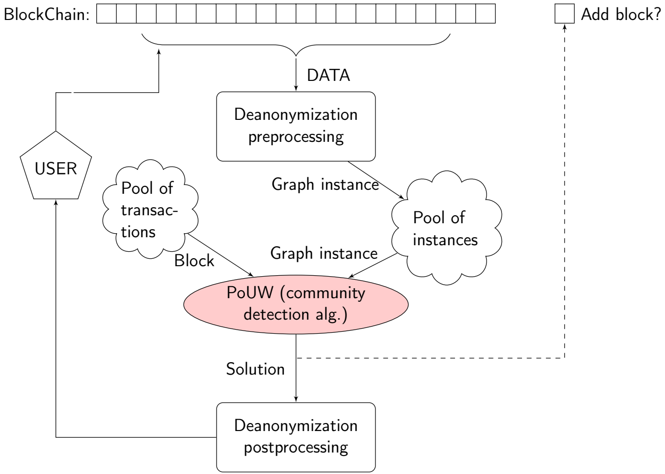
List of transactions

Tx1: Addr1 -> Addr2 (5)
Tx2: Addr2 -> Addr1 (2)
Tx3: Addr2 -> Addr3 (3)
Tx4: Addr3 -> Addr4 (7)
Tx5: Addr1 -> Addr4 (11)
Tx6: Addr1 -> Addr5 (1)
Tx7: Addr5 -> Addr1 (2)

Transaction graph



Methodology



PennState Great Valley
Mathematical Institute SANU
Science Fund of the Republic of Serbia

Formulation of Community Detection Problem: Parameters

Given is a graph $G = (V, E, W)$.
Vertices: $V = \{1, 2, \ldots, n\}$,
Arcs: $E \subseteq V^2$, $|E| = m$
$W : E \rightarrow R_+$ weights on arcs; $w_{i,j}$ is weight of arc (i, j).
C denotes the maximum number of communities
d_i^{in}, $i = 1, 2, \ldots, n$ - number of arcs ending in i
d_i^{out}, $i = 1, 2, \ldots, n$ - number of arcs beginning at i

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Formulation of Community Detection Problem: Variables

$$x_{i\ell} = \begin{cases} 1, & \text{if node } i \text{ is assigned to community } \ell, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{ij\ell} = \begin{cases} 1, & \text{if nodes } i \text{ and } j \text{ are assigned to the same community } \ell, \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{i\ell}, y_{ij\ell} \in \{0,1\}, \;\; 1 \leq i, j \leq n, \;\; 1 \leq \ell \leq C.$$

Introduce y variables to linearize problem formulation, i.e. $y_{ij\ell} = x_{i\ell} \cdot x_{j\ell}$

Formulation of Community Detection Problem: Objective function

Mixed Integer Linear Programming formulation maximizes the *modularity* Q defined as:

$$Q = \frac{1}{m} \sum_{\ell=1}^{C} \sum_{i,j=1}^{n} \left[w_{ij} - \frac{d_i^{in} d_j^{out}}{m} \right] y_{ij\ell} \qquad (1)$$

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Formulation of Community Detection Problem: Constraints

$$y_{ij\ell} \le x_{i\ell}, \quad 1 \le i, j \le n, \;\; 1 \le \ell \le C \tag{2}$$

$$y_{ij\ell} \le x_{j\ell}, \quad 1 \le i, j \le n, \;\; 1 \le \ell \le C \tag{3}$$

$$y_{ij\ell} \ge x_{i\ell} + x_{j\ell} - 1, \quad 1 \le i, j \le n, \;\; 1 \le \ell \le C \tag{4}$$

$$\sum_{\ell=1}^{C} x_{i\ell} = 1, \;\; 1 \le i \le n, \tag{5}$$

$$2 \le \sum_{i=1}^{n} x_{i\ell} \le n - 2(C-1), \;\; 1 \le \ell \le C, \tag{6}$$

$$x_{i\ell} \le \sum_{j=1}^{n} x_{j\ell-1}, \;\; 1 \le i \le n, \;\; 1 \le \ell \le C \tag{7}$$

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Basic Variable Neighborhood Search Algorithm

procedure BVNS(Problem input data, k_{max}, STOP)
    $x \leftarrow InitSolution()$
    $x_{best} \leftarrow x$
    repeat
        $k \leftarrow 1$
        repeat
            $x' \leftarrow RandomSolution(x_{best}, \mathcal{N}_k)$ ▷ Shaking
            $x'' \leftarrow LS(x')$ ▷ Local Search
            if $(f(x'') < f(x_{best}))$ **then** ▷ Move or Not (Neighborhood Change)
                $x_{best} \leftarrow x''$
                $k \leftarrow 1$
            else
                $k \leftarrow k + 1$
            end if
            $terminate \leftarrow StoppingCriterion(STOP)$
        until $(k > k_{max} \vee terminate)$
    until $(terminate)$
    return $(x_{best}, f(x_{best}))$
end procedure

VNS-based approach to Community Detection

Solution representation
$Sol = [s_1, s_2, \ldots, s_n]$
$Sol[i] = s_i$, $s_i \in \{1, 2, \ldots, c\}$, c is current number of communities

Initial solution is determined randomly
$Sol[i] = rand(1, C)$, $i = 1, 2, \ldots, n$

Neighborhood: move a node from the current community to some other (including $c + 1$).
If community ℓ is emptied c decreases.
Explored systematically once (at distance 1) in LS.
Performed randomly k times in Shaking.

Auxiliary data - reducing the complexity of calculating Q

Modularity matrix $B_{n\times n}$

$$B_{ij} = A_{ij} - \frac{d_i^{in}\, d_j^{out}}{2m}.$$

Truth matrix $T_{n\times n}$

$$T_{i,j} = \delta(Sol[i], Sol[j]),\ \ \delta(Sol[i], Sol[j]) = \begin{cases} 1, & \text{if exists } \ell \text{ s.t. } y_{ij\ell} = 1 \\ 0, & \text{otherwise} \end{cases}$$

Q is determined for initial solution, and then modified by at most $2n-2$ operations (instead of n^2).

PennState Great Valley

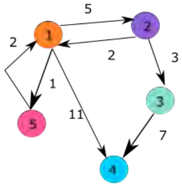
Mathematical Institute SANU

Science Fund of the Republic of Serbia

List of transactions

Tx1: Addr1 -> Addr2 (5)
Tx2: Addr2 -> Addr1 (2)
Tx3: Addr2 -> Addr3 (3)
Tx4: Addr3 -> Addr4 (7)
Tx5: Addr1 -> Addr4 (11)
Tx6: Addr1 -> Addr5 (1)
Tx7: Addr5 -> Addr1 (2)

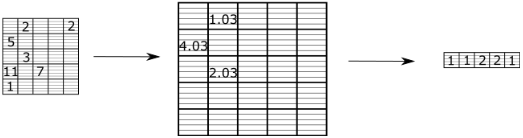
Transaction graph



Adjacency matrix

Modularity matrix

Solution



General Issues/Future work

- Correspondence between problem instance and the composed block
- The efficient exploration of miners' hardware
- Format for stating optimization problem instances
- Increasing the efficiency by parallelization (pool of miners)

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Summary and conclusion

- We considered energy efficiency and security in BC maintenance
- Energy consumption is directed to optimization problems by PoUW
- We selected MAX-SAT as a useful optimization problem
- Security problem (deanonymization) can be modeled by the community detection in directed graphs
- We combined solution approaches into a single procedure:
- Deanonymization represents the useful part of PoUW
- The resulting BC becomes self contained framework

PennState Great Valley

Mathematical Institute SANU

Science Fund of the Republic of Serbia

Thank you for your attention!

Questions?

Milan Todorović
mtodorovic@mi.sanu.ac.rs