Journal on
Information Technologies & Communications
http://www.ictmag.vn
Volume 2022, No.2
September

TABLE OF CONTENTS

An Experimental Study of Fast Greedy Algorithms for Fair Allocation Problems

Le Dang Nguyen[1], Trung Thanh Nguyen[2]

[1] Faculty of Computer Science, Haiphong University, Haiphong, Vietnam
[2] ORLab, Faculty of Computer Science, Phenikaa University, Hanoi, Vietnam

Correspondence: Trung Thanh Nguyen (thanh.nguyentrung@phenikaa-uni.edu.vn)
Communication: received 25 February 2022, revised 12 April 2022, accepted 20 April 2022
DOI: 10.32913/mic-ict-research.v2022.n2.1032

***Abstract*: This paper is concerned with two salient allocation problems in fair division of indivisible goods, aiming at maximizing egalitarian and Nash product social welfare. These problems are computationally NP-hard, meaning that achieving polynomial time algorithms is impossible, unless P = NP. Approximation algorithms, which return near-optimal solution with a theoretical guarantee, have been widely used for tackling the problems. However, most of them are often of high computational complexity or not easy to implement. It is therefore of great interest to explore fast greedy methods that can quickly produce a good solution. This paper presents an empirical study of the performance of several such methods. Interestingly, the obtained results show that fair allocation problems can be practically approximated by greedy algorithms.**

Keywords: *Fair allocation, exact algorithm, greedy algorithm, mixed-integer linear program,* NP*-hard.*

I. Introduction

In this paper, we study the fair allocation problem, which has shown its growing interest during last decades, with a wide range of real-world applications [1]. In short, this is a combinatorial optimization problem which asks to allocate m *discrete* items amongst a set of n agents (or players) so as to meet a certain notion of fairness. It is assumed that every item is "*indivisible*" and "*non-sharable*", that is, i) it cannot be broken in pieces before allocating to agents, and ii) it cannot be shared by two or more agents. For example, laptops and cell-phones are indivisible items which agents might not want to share with others. An allocation of items to agents is simply a partition of the whole set of items into n disjoint subsets. There are up to n^m such partitions, making the solution space large enough so that an exhaustive search for an optimal solution is impossible.

It now remains to define what a fair allocation is, a concept that is of independent interest in the field of Economic and Social Choice Theory [2, 3]. In general, there are many different ways of defining fairness, depending on particular applications. The most common way is to either use a so-called Collective Utility Function (CUF), which is a function for aggregating individual agents' utilities in a fair manner, or to follow an orthogonal method relying on determining the fair share of agents. Since we are focusing on the first method in this paper, we refer the reader to the paper [4] and the references therein for more details of the second method. Suppose that every agent evaluates the value of items through a *utility function*, which maps each subset of items to a numerical value representing the utility of the agent for the subset. Then, one can define a max-min fair allocation to be the one that maximizes the lowest utility among agents' utilities. This is also known as the Rawls aggregation function [5]. Alternatively, one can also define a max-Nash allocation so as to maximize the product of individual agents' utilities [6]. A more general utility-based fair concept, called rank-weighted utilitarianism, can be found in [7].

Our focus is on the computational problem of computing max-min allocations and max-Nash allocations. This problem belongs to the class of combinatorial NP-hard optimization problems, for which exact algorithms with a polynomial runtime is unlikely to exist [8]. Indeed, computing a max-min allocation remains NP-hard even when there are two agents having identical utility functions, as this special case can be polynomially reduced from partition, a well-known NP-complete problem which asks if one can partition m positive integers into two subsets such that the smallest sum of the integers assigned to any subset is maximized [9]. Due to the hardness result, many attempts have been put on designing approximation algorithms which run in polynomial time and produce near-optimal solution with a provable approximation guarantee. In fact, an α-approximation algorithm ($\alpha \geq 1$) for a

maximization problem can return, for any given probem instance, a solution whose value is always at least $1/\alpha$ times the optimum. Obviously, the closer factor α to 1, the better the approximation. Bezáková and Dani [10] showed that max-min allocation is NP-hard to approximate to within a factor of 2. On the positive side, Asadpour and Saberi [11] used a linear program (LP) relaxation, known as the configuration LP, to obtain a $(\sqrt{n}\log^3 n)$-approximation algorithm, and this result was then improved by Bateni et al. [12] and Chakrabarty et al. [13], independently, to a $O(n^{\epsilon}\log n)$-approximation algorithm in $n^{O(1/\epsilon)})$ time, for any constant $\epsilon > 0$. For the problem of maximizing Nash product social welfare, Nguyen and Rothe [14] was the first to study its approximability and they gave the first $O(m)$-approximation algorithm, which holds even for general *sub-additive* utility functions[1]. Since this paper was published there have been a sequence of studies improving upon this approximation factor [15–17]. The first constant approximation factor was proposed by Cole and Gkatzelis in [18], where they developed a novel technique based on the so-called spending-restricted market equilibrium to attain a factor of 2.889. The state-of-the-art approximation factor of 1.45 is obtained in [17], using a local search based approach. In an opposite direction, Lee [19] proved that there is no approximation factor better than 1.00008, unless P = NP. Narrowing the gap $(1.00008; 1.45)$ remains an open problem.

Better approximation results can be achieved for several restricted setting where the number of agents is upper bounded by a constant, or where agents utility functions are of special structures. For example, there are a fully polynomial time approximation scheme (FPTAS) for a constant number of agents [8], and a PTAS for idential agents [14, 20]. Several other works have dealt with bi-value utility functions or scoring vector-based utility functions, such as [4, 21–24], just to mention a few.

While the aformentioned approximation algorithms can provide good approximate solutions in polynomial time, most of them are mainly of theoretical interest. In fact, these algorithms are complicated, not easy to implement, and slow in practice. This motivates a comprehensive empirical study of greedy algorithms with which are typically very easy to implement with low complexity. To the best of our knowledge, this paper is the first to present a full analysis of the effectiveness of several greedy methods for solving the fair allocation problem. We give a formal description of these algorithms, analyze their theoretical performance, and evaluate their effectiveness through various experiments. The experiment shows that the greedy algorithms are quite efficient in practice in the sense that they can provide good solution in a short amount of time, when compared to exact methods.

[1] A set function u is sub-additive if $u(S \cup T) \geq u(S) + u(T)$ holds for every subsets $S, T \subseteq \mathcal{R}$.

II. Problem model

In this section, we present basic notions of fair division, define our fairness criteria, and model the existence and optimization problems that we study. We consider the basic setting of the fair allocation problem which consists of

- a set of agents $\mathcal{A} = \{a_1, \ldots, a_n\}$,
- a set of items $\mathcal{R} = \{r_1, \ldots, r_m\}$, and
- a utility function $u_i : 2^{\mathcal{R}} \to \mathbb{Z}_+$ for each agent $a_i \in \mathcal{A}$,

where $\mathbb{Z}_+$ denotes the set of nonnegative integer numbers. Let $\mathcal{U} = \{u_1, \ldots, u_n\}$. We call $\mathcal{I} = \{\mathcal{A}, \mathcal{R}, \mathcal{U}\}$ an instance of the fair allocation problem. We assume that every utility function is additive, that is, $u_i(S) = \sum_{r \in S} u_i(\{r\})$ for every $S \subseteq \mathcal{R}$. For easy of presentation, we write $u_i(r)$ instead of $u_i(\{r\})$. Also, assume that the empty set has utility 0 for every agent. A (complete) allocation is a partition $\pi = (\pi_1, \ldots, \pi_n)$ of $\mathcal{R}$, where π_i is the subset assigned to agent a_i, and $\pi_i \cap \pi_j = \emptyset$ for $i \neq j$ and $\cup_{i=1}^{n} \pi_i = \mathcal{R}$. We call π an *incomplete allocation* if $\cup_{i=1}^{n} \pi_i \subset \mathcal{R}$. Also, we say that π is a partial allocation of π' if $\pi_i \subseteq \pi'_i$ for all $i \in \{1, 2, \ldots, n\}$. We focus on max-min allocation and max-Nash allocation, which are defined through the concept of *social welfare*.

An allocation π is said to be max-min (respectively, max-Nash) if it has maximum egalitarian social welfare (respectively, maximum Nash product social welfare). We denote the problems of computing such allocations as Max-Min and Max-Nash. In the next section we will briefly review the complexity as well as algorithmic results for these problems.

Notation: For a positive integer p, we denote $[p]$ as the set $\{1, 2, \ldots, p\}$ of positive integers. Also, $\mathbf{0}$ and $\mathbf{1}$ are vectors with all 0 and 1 coordinates, respectively. Let $\mathbf{e}_i$ denotes the vector with a 1 in the i-th coordinate and 0's elsewhere. Also, we $||.||$ to denote the size of a given set.

Example 1: Consider an instance with three agents and six items, and the agent utility functions are given in the Table 1. One can check that the maximum egalitarian social welfare of the instance is 5, by taking an allocation $\pi = (\{r_5, r_2\}, \{r_4, r_6\}, \{r_3, r_1\})$. If we want to maximize the Nash product social welfare, just take the allocation $\pi' = (\{r_5, r_1\}, \{r_4, r_6\}, \{r_3, r_2\})$, with $sw_N(\pi') = (5.5.7)^{1/3} > sw_N(\pi) = (6.5.5)^{1/3}$.

III. Exact methods

The first and naive method for exactly for solving Max-Min and Max-Nash is exhaustive search which enumerates all possible allocations of m item to n agents and

TABLE I
UTILITIES OF THE AGENTS FOR EXAMPLE 1.

	r_1	r_2	r_3	r_4	r_5	r_6
a_1	1	2	2	0	4	1
a_2	1	1	0	3	1	2
a_3	1	3	4	2	0	2

picks the one of maximum egalitarian social welfare or of maximum Nash product social welfare. As mentioned earlier, this method takes $O(n^m)$ time.

A better method is to model the problems as integer (linear or quadratic) programs which can be solved by commercial solvers such as IBM ILOG CPLEX or Gurobi. A straightforward integer programming (IP) formulation for MAX-MIN can is as follows.

$$\begin{aligned}
\text{(IP)} \quad \max \quad & \min_{i=1}^{n} \left\{ \sum\nolimits_{j=1}^{m} u_i(r_j) \cdot x_{ij} \right\} && (1)\\
\text{s.t.} \quad & \sum\nolimits_{i=1}^{n} x_{ij} = 1, \qquad j \in [m], && (2)\\
& x_{ij} \in \{0,1\}, \quad i \in [n], j \in [m], && (3)
\end{aligned}$$

where the binary variable $x_{ij} = 1$ if item r_j is allocated to agent a_i, and $x_{ij} = 0$ otherwise. IP is not a linear program due to the nonlinearity of the objective function, and thus cannot be solved directly by CPLEX. The idea is to put the objective as a constraint, using an additional (fractional) variable T. As such, IP can be reformulated as the following equivalent Mixed Integer Linear Program (MILP):

$$\begin{aligned}
\text{(MILP)} \quad \max_{T \in \mathbb{R}_+} \quad & T\\
\text{s.t} \quad & \sum\nolimits_{j=1}^{m} u_i(r_j) \cdot x_{ij} \geq T, \ i \in [n],\\
& (2) - (3),
\end{aligned}$$

An integer programming formulation of MAX-NASH is similar to IP, except that the objective is replaced by the product function of agent utilities, i.e.,

$$\left(\prod_{i=1}^{n} \left\{ \sum\nolimits_{j=1}^{m} u_i(r_j) \cdot x_{ij} \right\} \right)^{1/n}. \qquad (4)$$

Again (4) is nonlinear, thus we will not be able to use CPLEX for solving it directly. To address this issue, we propose an equivalent model, called Second-Order Cone Program (SOCP), using the technique introduced by Ben-Tal and Nemirovski [25], that can be handled by CPLEX. To do so, we introduce new variables $z_i = \sum_{j=1}^{m} u_i(r_j) \cdot x_{ij}$ and write

$$\begin{aligned}
\max \quad & \eta\\
\text{s.t} \quad & 0 \leq \eta \leq \left(\prod_{i=1}^{n} z_i \right)^{1/n}, && (5)\\
& (2) - (3).
\end{aligned}$$

Let $k = \lceil \log_2 n \rceil$, the constraint (5) can be represented as

$$\begin{aligned}
& 0 \leq \eta \leq \sqrt{y_1^{k-1} y_2^{k-1}},\\
& 0 \leq y_j^l \leq \sqrt{y_{2j-1}^{l-1} y_{2j}^{l-1}}, \ j \in [2^{k-1}], \ l \in [k-1],\\
& 0 \leq y_i^0 = z_i, \ i \in [n],\\
& 0 \leq y_j^0 = \eta, \ j = n+1, \ldots, 2^k,
\end{aligned}$$

where $\eta, \{y_j^l\}$, $j = 1, \ldots, 2^k, l = 0, \ldots, k-1$, are additional variables. The above formulation is mixed integer SOCP since any constraint of the form $\{u, v, w \geq 0: \ u \leq \sqrt{vw}\}$ is equivalent to $\{u, v, w \geq 0: \ 4u^2 + (v-w)^2 \leq (v+w)^2\}$.

IV. GREEDY METHODS

In this section we describe several greedy methods with low complexity, for solving MAX-MIN and MAX-NASH. Recall that m is assumed to be at least n, otherwise the problem becomes trivial.

1. LPT algorithm

A first and naive allocation strategy is to allocate items, one by one, to agents, until no item available. In fact, the algorithm performs in m steps, at j-th step item r_j is given to an agent of smallest utility among those who have nonzero utility for r_j (tie-breaking arbitrarily). There should be at least one such agent, otherwise we can remove this item r_j from $\mathcal{R}$ without affecting the optimal value. Clearly, the complexity of the algorithm is $O(mn) = O(m^2)$, as $n \leq m$. A formal description is given in Algorithm 1. The algorithm is also known as the LPT rule, which was originally used in the context of the minimum makespan problem [26]. It was shown that such an algorithm yields a $\frac{4n-2}{3n-1}$-approximation algorithm for MAX-MIN [27] and a 1.061-approximation algorithm for MAX-NASH [23], assuming that all agents have the same utility function.

Algorithm 1: LPT

```
Inputs: 𝒜, ℛ, 𝒰.
Outputs: An allocation π.
begin
    for j := 1 to m do
        Assign item r_j to an agent who is worst-off
          among those who have non-zero utility for the
          item (tie-breaking arbitrarily);
    end
    return π;
end
```

Example 2 (continued Example 1): Consider the instance given in Example 1, the LPT algorithm will return $\pi = (\{r_1, r_5\}, \{r_2, r_4, r_6\}, \{r_3\})$. This is neither a max-min allocation nor a max-Nash allocation.

It is natural to ask if the LPT algorithm can theoretically guarantee any bound on the quality of the solution found for non-identical utility functions. We answer this question affirmatively for both MAX-MIN and MAX-NASH.

Proposition 1: The LPT algorithm does not posses an $O(m)$-approximation algorithm for both MAX-MIN and MAX-NASH.

Proof: We prove that LPT cannot give an $\frac{\epsilon}{m}$-approximation for any fixed positive constant $\epsilon < 1$. The proof is by contradiction. Consider an instance with $n = 2$ agents $\{a_1, a_2\}$, $m = 4$ items $\{r_1, r_2, r_3, r_4\}$ and the utilities of agents for items are as follows: $u_1(r_1) = 2, u_1(r_2) = 1, u_1(r_3) = 1, u_1(r_4) = M$, $u_2(r_1) = 0, u_2(r_2) = 3, u_2(r_3) = M, u_2(r_4) = 1$, where M is some positive number $M > 2$. One can easily see that the allocation returns by LPT is $\pi = (\{r_1, r_3\}, \{r_2, r_4\})$, which has the egalitarian social welfare of 4, while the max-min allocation is $\pi^* = (\{r_1, r_4\}, \{r_2, r_3\})$, which has egalitarian social welfare of $2+M$. Suppose that LPT is an $\frac{\epsilon}{4}$-approximation, then it must hold that:

$$4 \geq \frac{\epsilon}{4}(2 + M) \geq \frac{\epsilon}{2} + M\frac{\epsilon}{4}.$$

However, this is not true for $M > \frac{16}{\epsilon}$.

Similarly, $\pi = (\{r_1, r_3\}, \{r_2, r_4\})$ has the Nash product social welfare of 4, and $\pi^* = (\{r_1, r_4\}, \{r_2, r_3\})$ has the Nash product social welfare of $\sqrt{(2 + M)(3 + M)} > 4$, for $M > 2$. If LPT is an $\frac{\epsilon}{4}$-approximation, then we have that

$$4 \geq \frac{\epsilon}{4}\sqrt{(2 + M)(3 + M)} \geq \frac{\epsilon}{4}\sqrt{5 + 5M + M^2},$$

which again is not true for $M > \frac{16}{\epsilon}$.

2. Graph matching greedy algorithm

In [14] the authors proposed a graph-matching greedy algorithm (GMG) for computing an allocation of maximum Nash product social welfare. Let $G = (V_G, E_G, w)$ be a weighted bipartite graph, where $V_G = \mathcal{A} \cup \mathcal{R}$ is vertex sets including agent-vertices and of item-vertices. Each $e = (a_i, r_j) \in E$ has a weight that equals to the utility of agent a_i for item r_j, i.e., $w(e) = u_i(r_j)$. The basic idea of GMBA is to find a matching of the graph G such that the product of weights of the edges is maximized. Then, for each pair of agent-item in the matching, allocate item to its matched agent, while the remaining unmatched items are assigned arbitrarily to agents. It was prove in [14] that such an algorithm can guarantees an $O(m)$-approximation. Recently, Garg et al. [28] have modified this algorithm by repeating the matching algorithm for the remaining items several times until no items available. A formal description is given in Algorithm 2.

Basically, Algorithm 2 works in rounds, in each round n items are chosen to be allocated, one item per agent. By adding dummy items of zero utility if needed, one can assume that m is a multiple of n, that is, $m = \ell n$, for some positive integer ℓ. Hence, the number of rounds in the algorithm is ℓ. We now explain in detail how items are allocated in each round, assuming that the objective is to maximizing the Nash product social welfare. The case of the egalitarian social welfare is slightly different and is discussed later.

In the first round of the algorithm, we find a max-prod matching of G (see Definition 1), denoted as $\mathcal{M}^1$, and allocate items to agents based on this matching to obtain an incomplete allocation π^1. By removing items that have been allocated and edges in the matching, we obtain a subgraph of G. In the next round, a straightforward idea is to repeat what we have done in the first round to the subgraph and update the allocation. However, note that agents have already received a subset of items in the previous round, thus it would be fairer if the matching is done in such a way that maximizing the product of agent utilities. To this end, we replace the weight function of G by the new one which adds a utility $u_i(\pi_i^1)$ to the weight of every edge incident to agent-vertex a_i, for all $i \in [n]$. We apply this replacement for all the ℓ rounds.

For the case of max-min allocations, in each round of Algorithm 2 we find a max-min matching instead of a max-product matching. That is, we find a matching such that the minimum of weights of the edges in the matching is maximized (see Definition 1).

Definition 1: A maximum matching $\mathcal{M}$ of a weighted bipartite graph G is called

- a *max-min matching* if $\min_{e \in \mathcal{M}} w(e) \geq \min_{e \in \mathcal{M}'} w(e)$ for any other maximum matching $\mathcal{M}'$ of G;
- a *max-product matching* if $\prod_{e \in \mathcal{M}} w(e) \geq \prod_{e \in \mathcal{M}'} w(e)$ for any other maximum matching $\mathcal{M}'$ of G.

Algorithm 2: Graph-matching based Greedy (GMBA)

Inputs: $\mathcal{A}, \mathcal{R}, \mathcal{U}$.
Outputs: An allocation π.
begin
  $k \leftarrow 1$; $G^k \leftarrow G$; $w^k \leftarrow w$;
  $\pi^k \leftarrow (\emptyset, \ldots, \emptyset)$;
  while $k > \ell$ **do**
    Find a max-product matching $\mathcal{M}^k$ of G^k

$$\mathcal{M}^k \leftarrow \{(a_1, r^k_{j_1}), \ldots, (a_n, r^k_{j_n})\}$$

    Allocate items to agents based on $\mathcal{M}^k$

$$\pi^k_i \leftarrow \pi^k_i \cup \{r^k_{j_i}\}, \quad \text{for } i \in [n]$$

    $k \leftarrow k + 1$;
    Define $G^k = (V_{G^k}, E_{G^k}, w)$

$$V_{G^k} \leftarrow V_{G^{k-1}} \setminus \{r^k_{j_1}, \ldots, r^k_{j_n}\}$$
$$E_{G^k} \leftarrow E_{G^{k-1}} \setminus \mathcal{M}^k$$

    $w^k((a_i, r_j)) \leftarrow u_i(\pi^{k-1} \cup \{r_j\}),\ \forall\ (a_i, r_j) \in E_{G^k}$.
  end
  return π^ℓ
end

It was proved that both the matching can be efficiently found in polynomial time [14], by reducing to the well-known problem of finding a maximum weight matching which can be solved by the Hungarian method [29]. Alternatively, one can formulate the problem as a binary linear program in which the constraint matrix (i.e., the matrix defining the system of linear constraints) is totally unimodular[2]. It is well-known that such a program can be efficiently solved in $O(m^{2.5}L)$ time[3] [30], where L is the number of binary bits encoding the problem instance. For the sake of completeness, we provide an ILP model for the maximum weighted matching problem in Appendix.

Lemma 1 ([14]): Given a weighted complete bipartite graph G, a max-min matching and a max-product matching of G can be found in $O(m+n)^3$ time and $O((m+n)^3 L)$, respectively, where L is the number of binary bits encoding the problem instance.

Example 3 (continued Example 1): Consider the instance given in Example 1, the LPT algorithm will return $\pi = (\{r_1, r_5\}, \{r_2, r_4, r_6\}, \{r_3\})$. This is neither a max-min allocation nor a max-Nash allocation.

The Algorithm 2 runs in $\ell = m/n$ steps, and at each step $k \in [\ell]$ we run Hungarian method (or solve a BLP) to find a maximum weighted matching on a graph G^k having n agent-vertices and $m - (k-1)n$ item-vertices. Then the number of vertices of G^k at step k is equal to $(\ell + 2 - k)n$. Therefore, the overall complexity of Algorithm 2 is

$$O(n^3 \sum_{k=1}^{\ell} (\ell + 2 - k)^3) = O(n^3 \ell^4) = O(m^4/n),$$

where the second equality is due to the equality $\sum_{i=1}^{p} i^3 = (\sum_{i=1}^{p} i)^2 = \frac{1}{4}p^2(p+1)^2$, whose proof can be derived using induction on p.

Theorem 1 ([28]): Algorithm 2 is an $O(n)$-approximation algorithm for MAX-NASH.

It is not very clear if Algorithm 2 can give also an $O(n)$-approximation algorithm for MAX-MIN.

3. Picking sequences

Picking sequence has been known as simple protocol for distributing items among agents without eliciting the agents' utility functions first. This can be also seen as a decentralized approach in which agents communicate with each other to reach a final allocation. In particular, agents are asked to choose items one after the other, according to a predefined order, also called *picking policy*. Formally, a picking policy is a sequence $\sigma = (\sigma_1 \ldots \sigma_m) \in \{1, \ldots, n\}^m$, where at each step, agent a_{σ_i} picks her most preferred item among those remaining. For a formal definition of an allocation induced by a picking policy we refer to the paper by Bouveret and Lang [31]. In this paper, we consider two common policy types:

- *balanced picking policy* $(12\ldots n12\ldots n\ldots)$: agents pick items in rounds. In the first round, agents take turn to pick item: agent 1 first, then agent 2, then agent 3, and so on. Agent n is the last agent who picks item. In the next rounds, we repeat this item picking procedure until all items are allocated.
- *fair picking policy* $(1\ldots nn\ldots 1\ldots)$: according to this policy the orders in which agents take turn to pick item are different between "odd rounds" and "even rounds". In fact, in the first round, the order is the same as the one in the balanced picking policy. However, it is reversed in the second round. That is, the agent n is the first to pick item, then agent $n-1$, and so on.

Example 4 (continued Example 1): Consider the instance given in Example 1, one can see that $\pi = (\{r_5, r_2\}, \{r_4, r_6\}, \{r_3, r_1\})$ will be returned by the balanced picking policy, while $\pi' = (\{r_5, r_1\}, \{r_4, r_6\}, \{r_3, r_2\})$ will be returned by the fair picking policy.

We denote by BPG and FPG the algorithm 3 when a balanced picking policy and a fair picking policy are used, respectively. Like the LPT algorithm, these algorithms

[2] An integer matrix A is totally unimodular if if every square submatrix has determinant 0, +1 or −1. Another sufficient condition for the totally unimodularity is that no more than 2 nonzeros are in each column.

[3] To be precisely, the running time of Lee and Sidford's algorithm [30] is $\tilde{O}((nnz(A) + d^2)\sqrt{d}L)$, where $nnz(A)$ is the number of non-zero elements in A, and d is the number of A's rows.

Algorithm 3: Picking-sequence based Greedy

Inputs: $\mathcal{A}, \mathcal{R}, \mathcal{U}$ and a policy $\sigma = (\sigma_1 \dots \sigma_m)$
Outputs: An allocation π
begin
  $\tilde{\mathcal{R}} \leftarrow \mathcal{R}$;
  for $j := 1$ to m **do**
    Ask agent σ_j to pick one item from $\tilde{\mathcal{R}}$
    Let r be the item picked by σ_j and $\tilde{\mathcal{R}} \leftarrow \tilde{\mathcal{R}} \setminus \{r\}$
  end
  Let π be the obtained allocation;
  return π;
end

do not result in an $O(m)$-approximation, as shown in the proposition below.

Proposition 2: BPG and FPG algorithms do not posses an $O(m)$-approximation algorithm for both MAX-MIN and MAX-NASH.

Proof: We give a proof for FPG only, as the case of BPG can be treated similarly. The proof is by contradiction. Consider an instance with $n = 2$ agents $\{a_1, a_2\}$, $m = 2$ items $\{r_1, r_2\}$ and the utilities of agents for items are as follows: $u_1(r_1) = 1, u_1(r_2) = 1$, $u_2(r_1) = 1, u_2(r_2) = M$, where M is some positive number $M < 1$. By the FPG algorithm, agent a_1 is the first to pick item, yielding to an allocation π, which has the egalitarian social welfare of M, while the max-min allocation is $\pi^* = (\{r_2, r_1\})$, which has egalitarian social welfare of 1. Suppose that FPG is an $\frac{\epsilon}{2}$-approximation, then it must hold that:

$$M \geq \frac{\epsilon}{2} \cdot 1 = \frac{\epsilon}{2}.$$

However, this is not true for $M < \frac{\epsilon}{2}$.

Similarly, π has the Nash product social welfare of $\sqrt{M}$, and π^* has the Nash product social welfare of $1 > \sqrt{M}$, fro $M < 1$. If LPT is an $\frac{\epsilon}{2}$-approximation, then we have that

$$\sqrt{M} \geq \frac{\epsilon}{2} \cdot 1 = \frac{\epsilon}{2},$$

which again is not true for $M < \frac{\epsilon}{4}$.

V. EXPERIMENTS

In this section, we carry out comprehensive computational tests to evaluate the performance of the four greedy algorithm: LPT, GMG, FPG and BPG. In particular, we compare the solution returned by each greedy algorithm with that produced by the exact integer programming method (presented in Section III). We make use of several kinds of problem instances in the experiment, which are specified in the following.

For the evaluation purpose, we propose studying different instance sizes with a varying number of agents $n \in \{4, 8, 12, 16, 20\}$ and items $m = \{2n, 4n, 6n, 8n, 10n\}$. The largest instances are those of 20×200, which is a considerable size that can be solved by IBM-ILOG CPLEX 20.1. There are 25 combinations of n and m and for each combination we generate 10 instances and average the results. All the algorithms are implemented using Python on an INTEL Core i7, 1.5 GHz, with 16 GB of RAM. The best known solution for each one of instances is obtained by running the MILP model presented in Section III with IBM-ILOG CPLEX solver with a time limit of $T_{\max} = 300s$. The utilities of agents for items are uniformly distributed in the interval $[1, 100]$ as it is common in the literature for fair division [4, 32–34].

We propose studying additional sets of instances with scoring-vector based utilites and idenical utilities. The reason is that greedy algorithms have shown their good performance when applied to these special kinds of instances. It is worth noting that even under this restriction, both Max-Min and MAX-NASH remain NP-hard (see [21, 22]). On the other hand, fair allocation with scoring-vector based utilities has been attracted a considerable attention in the last decade. In this setting, it is given as input a (nonnegative) vector $s = (s_1, \dots, s_m)$ where $s_1 \geq \dots \geq s_m$. Each agent is asked to rank m items from the most preferred to the least preferred ones, and then assigns a utility of s_j to an item ranked at position j, for all $j \in [m]$. In our experiment, we focus on the most common scoring vector, the Borda vector $s = (m, m-1, \dots, 1)$. For example, by using the Borda vector, every agent has a utility of m for the top-ranked item and a utility of 1 for the least preferred item.

The tabulated results for each algorithm will be presented as the relative percentage deviation (RPD) from the solution obtained by the solver as follows:

$$\text{RPD}(\mathcal{I}) = \frac{\text{OPT}(\mathcal{I}) - A(\mathcal{I})}{\text{OPT}(\mathcal{I})} \times 100,$$

where $A(\mathcal{I})$ is the value returned by a given algorithm A and an instance $\mathcal{I}$, and $\text{OPT}(\mathcal{I})$ is the aforementioned 300 second CPLEX run on the same instance.

We present now the full results for all three types of instances, presented in Tables II-IV. The first observation is that GMG outperforms other three algorithms LPT, FPG and BPG for both the problems Max-Min and MAX-NASH, no matter what the instance type is used. While the LPT algorithm has a worse performance than the others in all cases, FPG and BPG give a quite similar performance (except the identical utilites), which is much better than that of LPT but slightly worse than that of GMG.

Tables II and III show the results for instances where utility functions are uniform randomly generated in the interval $[0, 100]$, and for the instances with the Borda scoring-vector-based utilities. It is observed that there are not much differences in the performance of the algorithms

TABLE II
AVERAGE RELATIVE PERCENTAGE DEVIATIONS FOR THE INTERVAL [0, 100]

		Max-Min				Max-Nash			
n	m	**LPT**	**GMG**	**FPG**	**BPG**	**LPT**	**GMG**	**FPG**	**BPG**
4	8	53.378	16.256	25.575	25.637	37.114	3.651	9.025	8.161
	16	39.221	12.132	15.658	16.171	38.13	2.312	5.137	3.991
	24	42.788	9.972	10.777	12.54	39.586	1.494	2.886	2.696
	32	41.587	8.281	9.084	7.259	38.412	0.893	2.021	2.33
	40	41.411	9.355	8.921	10.862	36.325	1.145	1.927	1.577
8	16	62.511	16.018	33.895	32.781	42.808	1.557	6.173	6.671
	32	53.426	10.571	16.455	19.641	43.38	1.304	3.243	4.822
	48	48.364	9.989	10.592	12.64	44.154	1.187	2.495	2.341
	64	48.285	7.748	9.945	9.455	44.036	0.922	1.858	1.829
	80	45.425	8.345	8.265	8.791	41.651	0.782	1.45	1.397
12	24	63.008	10.81	27.532	29.812	45.122	1.123	6.757	6.78
	48	53.662	7.652	17.43	15.201	48.772	1.162	3.564	2.884
	72	52.859	7.559	9.394	11.635	46.602	0.878	2.457	2.237
	96	50.918	6.249	9.486	8.804	46.022	0.71	1.784	1.393
	120	49.435	7.252	7.602	8.017	45.364	0.711	1.417	1.407
16	32	62.917	10.744	27.871	31.968	50.124	0.858	4.581	4.664
	64	54.707	7.164	15.504	14.564	47.53	0.671	2.406	2.229
	96	51.532	6.534	9.568	11.174	47.359	0.795	2.049	1.856
	128	50.357	5.175	10.086	8.993	47.217	0.528	1.381	1.405
	160	50.57	5.624	6.171	6.254	47.352	0.483	1.083	0.988
20	40	62.483	10.055	32.575	26.79	47.075	0.679	4.314	4.201
	80	57.116	6.582	13.627	13.685	48.345	0.622	2.345	2.508
	120	52.144	5.499	8.719	9.941	48.607	0.628	1.738	1.822
	160	52.171	4.937	7.278	8.283	48.154	0.572	1.55	1.404
	200	51.89	3.79	5.121	7.172	49.502	0.436	1.067	0.9
Average		**51.686**	**8.572**	**14.285**	**14.723**	**44.749**	**1.044**	**2.988**	**2.899**

between these two types of instances. For example, LPT gives average results of around 52% and 45% deviation from the optimal max-min solution and the optimal max-Nash solution, respectively. FPG and BPG give 12-14% for MAX-MIN and these are significantly reduced to less than 3% for MAX-NASH. The best performance is offered by GMG, which gives less than 10% for MAX-MIN and only around 1% for MAX-NASH. One can see that all the greedy algorithms achieve better results when applied to the MAX-NASH problem. This is not very surprising because while MAX-MIN aims to maximizing the utility of the worst-off agents, which may leave a large gap in the utilities between agents, MAX-MIN can well address this issue by simultaneously balancing agents utilities (see also Example 1 which shows a max-min allocation not being a max-Nash allocation).

Finally, we show the results of the identical utilities in Tables IV, which exhibits better performances compared to other two instance-types above. For the problem MAX-MIN, GMG is the best with 3.336%, which is approximately three and six times better than LPT and BPG, respectively, while FPG gives 4.178%, which almost matches the result of GMG. In the case of MAX-NASH, both GMG and FPG obtain very good results of less than 0.1%, while the results for LPT and BPG are 0.752% and 1.278%, respectively, which are also quite impressive when compared to the case of non-identical utilities. Note that the fact that LPT works well for identical utilities has been also theoretically studied in the literature [23, 27].

To get an overall picture of experimental results, Table

TABLE III
AVERAGE RELATIVE PERCENTAGE DEVIATIONS FOR THE BORDA UTILITY FUNCTIONS

		Max-Min				Max-Nash			
n	m	**LPT**	**GMG**	**FPG**	**BPG**	**LPT**	**GMG**	**FPG**	**BPG**
4	8	60.342	13.584	18.939	20.718	40.203	2.377	6.828	3.644
	16	44.802	7.343	10.215	14.565	42.097	1.379	2.968	3.459
	24	46.276	5.283	8.896	10.811	35.66	0.877	1.663	2.847
	32	37.348	4.813	6.27	6.901	35.892	0.775	1.984	1.85
	40	42.593	3.63	5.169	6.686	41.213	1.095	1.554	1.756
8	16	58.459	13.152	27.286	33.238	44.686	0.973	4.026	3.892
	32	50.28	7.544	14.782	14.56	43.658	1.024	3.262	2.486
	48	51.336	4.8	10.195	9.652	45.098	0.882	1.687	1.609
	64	48.201	5.299	7.223	7.211	44.302	0.827	1.702	1.827
	80	46.489	4.125	5.848	6.918	43.465	0.566	1.298	1.376
12	24	60.338	8.451	24.633	25.366	46.358	0.712	3.58	4.255
	48	53.928	5.455	12.522	12.514	47.124	0.932	2.902	2.369
	72	51.867	5.644	8.838	11.998	47.229	0.707	1.498	1.76
	96	52.668	4.042	6.66	7.726	45.297	0.571	1.485	1.472
	120	48.447	4.022	5.405	6.108	46.406	0.501	1.198	1.17
16	32	60.509	7.746	24.777	27.47	46.877	0.878	5.461	4.366
	64	54.402	4.86	15.319	17.302	47.553	0.679	2.306	2.477
	96	51.186	4.609	8.26	10.302	47.974	0.494	1.763	1.858
	128	51.716	4.167	6.52	7.641	46.707	0.529	1.448	1.33
	160	49.757	4.054	6.381	7.608	47.664	0.455	1.028	1.196
20	40	61.573	7.548	25.648	34.687	48.732	0.523	4.387	3.918
	80	55.005	4.209	15.34	14.411	48.847	0.542	1.942	2.024
	120	52.479	3.46	9.595	11.662	46.458	0.435	1.495	1.349
	160	51.436	3.466	6.496	6.579	47.697	0.444	1.206	1.145
	200	51.88	3.208	6.238	6.361	47.879	0.386	0.928	1.105
Average		**51.733**	**5.781**	**11.898**	**13.56**	**45.003**	**0.782**	**2.383**	**2.261**

V presents all three tested instance-types along with all four greedy algorithms. There are two global averages calculated in the table. The first one is the average results for each algorithm of all instance-types, i.e., an average relative percentage deviation from the 300s CPLEX solution for all instances. The second global average contains the three "typical" instance-types: general utilities in $[0, 100]$, borda scoring-vector-based utilities, and identical utilities in $[0, 100]$. From these averages one can remark that GMG manages to beat the other three algorithms LPT, FPG and BPG. However, to be fair, it is noticed that the latter three algorithm are more faster than GMG, in terms of worst-case running time. In particular, GMG uses, on average, less than $1/4$ second of CPU time, whereas FPG and BPG have been run for just around $1/10$ second. It is worth noting that exactly solving our problems by CPLEX takes up to 5 seconds for each instance.

Finally, our experimental results give some interesting observation about how to solve NP-hard problems Max-Min and MAX-NASH practically. While the first problem is theoretically hard to approximated to within a factor of 2, it can be well tackled in practice by, e.g., the greedy algorithm GMG, which can quickly produce a solution whose value is at least 90% that of the optimum. For the second problem, GMG can even give a much better ratio of 99%, compared to the theoretical ratio of 69% ($\approx 1/1.45$ given in [17]). On the other hand, simple greedy strategies such as FPG and BPG seems to be very good alternatives with the loss of no more than 3% in the objective values, for both the problems.

VI. CONCLUSION

We have studied empirically the four greedy algorithms, LPT, GMG, FPG and BPG, for solving two computationally

TABLE IV
AVERAGE RELATIVE PERCENTAGE DEVIATIONS FOR THE INTERVAL $[0, 100]$ AND IDENTICAL UTILITY FUNCTIONS

		Max-Min				Max-Nash			
n	m	**LPT**	**GMG**	**FPG**	**BPG**	**LPT**	**GMG**	**FPG**	**BPG**
4	8	21.605	13.649	13.649	33.454	1.701	0.659	0.659	3.084
	16	14.268	3.059	5.556	17.41	0.638	0.05	0.109	1.149
	24	9.982	1.465	2.985	12.783	0.328	0.014	0.035	0.457
	32	6.589	0.683	2.071	9.32	0.152	0.003	0.009	0.259
	40	5.323	0.409	1.245	8.032	0.114	0.001	0.01	0.174
8	16	25.157	9.365	9.365	38.935	3.008	0.572	0.572	4.846
	32	15.854	3.028	4.597	20.691	0.758	0.047	0.063	1.037
	48	9.603	1.831	2.375	14.001	0.252	0.01	0.007	0.435
	64	7.206	0.988	1.742	10.471	0.118	0.001	0.006	0.262
	80	5.498	0.48	1.294	8.793	0.136	0.001	0.005	0.168
12	24	28.776	11.456	10.42	45.524	2.916	0.233	0.233	4.778
	48	15.791	2.582	4.996	22.859	0.602	0.032	0.034	1.105
	72	10.125	1.269	2.659	15.672	0.283	0.005	0.014	0.504
	96	6.983	0.977	1.674	11.244	0.171	0.002	0.003	0.243
	120	6.287	0.526	1.144	9.137	0.1	0.001	0.003	0.166
16	32	26.993	11.006	11.006	44.936	2.481	0.369	0.369	4.899
	64	14.88	2.764	4.853	24.853	0.638	0.035	0.05	1.072
	96	10.126	1.089	1.966	15.75	0.336	0.004	0.011	0.477
	128	8.024	0.537	1.723	12.129	0.18	0.001	0.005	0.275
	160	6.318	0.473	1.166	9.524	0.086	0.001	0.001	0.167
20	40	28.681	10.023	10.023	45.073	2.611	0.155	0.155	4.454
	80	15.446	2.791	3.508	23.725	0.65	0.028	0.028	1.008
	120	11.546	1.675	2.257	16.221	0.292	0.002	0.008	0.49
	160	7.858	0.853	1.27	11.814	0.121	0.001	0.003	0.261
	200	5.919	0.409	0.899	9.432	0.131	0.001	0.002	0.183
Average		**12.994**	**3.336**	**4.178**	**19.671**	**0.752**	**0.089**	**0.095**	**1.278**

TABLE V
SUMMARY RESULTS FOR ALL INSTANCE TYPES. BOLD (ITALICS) FIGURES INDICATE BEST (WORST) RESULTS, RESPECTIVELY.

	Max-Min				Max-Nash			
Instance types	**LPT**	**GMG**	**FPG**	**BPG**	**LPT**	**GMG**	**FPG**	**BPG**
Interval $[0, 100]$	*51.686*	**8.572**	14.285	14.723	*44.749*	**1.044**	2.988	2.899
Borda utilities	*51.733*	**5.781**	11.898	13.56	*45.003*	**0.782**	2.383	2.261
Identical utilities	12.994	**3.336**	4.178	*19.671*	0.752	**0.089**	0.095	*1.278*
Average	*38.804*	**5.916**	10.12	15.985	30.168	**0.638**	1.822	2.146

NP-hard problems: egalitarian social welfare maximization and Nash product social welfare maximization. It has be proved that LPT and other two picking policy algorithms cannot give any theoretical bound on the quality of solution found. However, based on the simulation results, it is observed that all the studied greedy algorithms can quickly find near optimal solution of very good quality for various instance types, using a short amount of running time. For the future work, it would be very interesting to see if the graph-matching based greedy algorithms GMG can give an $O(n)$-approximation algorithm for MAX-MIN.

VII. Appendix

An ILP model for MWMP

For a bipartite graph $G = (V, E)$, let $\delta(v)$ be the set of edges incident to the vertex $v \in V$. An ILP formulation for MWMP is as follows.

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ \text{s.t} \quad & \sum_{e \in \delta(v)} x_e \leq 1, && v \in V, \\ & x_e \in \{0, 1\}, && e \in E, \end{aligned}$$

where $x_e = 1$ if edge e is in the matching $\mathcal{M}$, and $x_e = 0$, otherwise. Since G is complete, there are mn edges in total, and thus the number of variables x_e is mn. An equivalent matrix form of the above ILP as follows.

$$\begin{aligned} \max \quad & \mathbf{1}^T \cdot x \\ \text{s.t} \quad & Ax \leq \mathbf{1}, \\ & x \in \{0, 1\}^{mn}, \end{aligned}$$

where $x = (x_{ij})$, and $x_{ij} = 1$ if item r_j is assigned to agent a_i, and $x_{ij} = 0$, otherwise. $A \in \{0, 1\}^{(m+n)\times mn}$ is a block matrix of the form

$$\begin{bmatrix} A^{(1)}_{n\times m} & A^{(2)}_{n\times m} & \cdots & A^{(n)}_{n\times m} \\ I_{m\times m} & I_{m\times m} & \cdots & I_{m\times m} \end{bmatrix},$$

where $A^{(i)}_{n\times m}$ is the matrix of size $n \times m$, with all 1-entries in i-th row, and 0-entries in others, and $I_{m\times m}$ is identical matrix of size $m \times m$. It is not hard to see that each column of A contains exactly 2 nonzeros, making A totally unimodular. In addition, the total number of nonzeros of A is $2mn$, resulting in the complexity of $O((mn + (m+n)^2)\sqrt{m+n}L)$ when applying Lee and Sidford's algorithm on G.

Acknowledgement

We would like to thank the reviewers for very helpful comments and suggestions that helped us to improve the presentation of the manuscript. This work was supported by Vietnam National Foundation for Science and Technology Development under Project 102.01-2020.21.

References

[1] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. A. Padget, S. Phelps, J. A. R.-Aguilar, and P. Sousa, "Issues in multiagent resource allocation," *Informatica (Slovenia)*, vol. 30, no. 1, pp. 3–31, 2006.

[2] M. Fleurbaey and F. Maniquet, *A Theory of Fairness and Social Welfare*. Cambridge University Press, 2011.

[3] H. Moulin, *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, Eds. Cambridge University Press, 2016.

[4] S. Bouveret and M. Lemaître, "Characterizing conflicts in fair division of indivisible goods using a scale of criteria," *Auton. Agents Multi Agent Syst.*, vol. 30, no. 2, pp. 259–290, 2016.

[5] J. Rawls, *A Theory of Justice*. London: Oxford University Press, 1971.

[6] M. Kaneko and K. Nakamura, "The nash social welfare function," *Econometrica*, vol. 47, no. 2, pp. 423–435, 1979.

[7] H. Moulin, *Axioms of cooperative decision making*. Cambridge: Cambridge University Press, 1988.

[8] N. Nguyen, T. T. Nguyen, M. Roos, and J. Rothe, "Computational complexity and approximability of social welfare optimization in multiagent resource allocation," *Auton. Agents Multi Agent Syst.*, vol. 28, no. 2, pp. 256–289, 2014.

[9] M. Garey and D. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*. New York: W. H. Freeman and Company, 1979.

[10] I. Bezáková and V. Dani, "Allocating indivisible goods," *SIGecom Exch.*, vol. 5, no. 3, pp. 11–18, 2005.

[11] A. Asadpour and A. Saberi, "An approximation algorithm for max-min fair allocation of indivisible goods," *SIAM J. Comput.*, vol. 39, no. 7, pp. 2970–2989, 2010.

[12] M. Bateni, M. Charikar, and V. Guruswami, "Maxmin allocation via degree lower-bounded arborescences," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, M. Mitzenmacher, Ed. ACM, 2009, pp. 543–552.

[13] D. Chakrabarty, J. Chuzhoy, and S. Khanna, "On allocating goods to maximize fairness," in *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*. IEEE Computer Society, 2009, pp. 107–116.

[14] T. T. Nguyen and J. Rothe, "Minimizing envy and maximizing average nash social welfare in the allocation of indivisible goods," *Discret. Appl. Math.*, vol. 179, pp. 54–68, 2014.

[15] N. Anari, S. O. Gharan, A. Saberi, and M. Singh, "Nash social welfare, matrix permanent, and stable polynomials," in *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, ser. LIPIcs, C. H. Papadimitriou, Ed., vol. 67. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 36:1–36:12.

[16] R. Cole, N. R. Devanur, V. Gkatzelis, K. Jain, T. Mai, V. V. Vazirani, and S. Yazdanbod, "Convex program duality, fisher markets, and nash social welfare," in *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, C. Daskalakis, M. Babaioff, and H. Moulin, Eds. ACM, 2017, pp. 459–460.

[17] S. Barman, S. K. Krishnamurthy, and R. Vaish, "Finding fair and efficient allocations," in *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, É. Tardos, E. Elkind, and R. Vohra, Eds. ACM, 2018, pp. 557–574.

[18] R. Cole and V. Gkatzelis, "Approximating the nash social welfare with indivisible items," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, R. A. Servedio and R. Rubinfeld, Eds. ACM, 2015, pp. 371–380.

[19] E. Lee, "Apx-hardness of maximizing nash social welfare with indivisible items," *Inf. Process. Lett.*, vol. 122, pp. 17–20, 2017.

[20] G. J. Woeginger, "A polynomial-time approximation scheme for maximizing the minimum machine completion time," *Oper. Res. Lett.*, vol. 20, no. 4, pp. 149–154, 1997.

[21] A. Darmann and J. Schauer, "Maximizing nash product social welfare in allocating indivisible goods," *Eur. J. Oper. Res.*, vol. 247, no. 2, pp. 548–559, 2015.
[22] D. Baumeister, S. Bouveret, J. Lang, N. Nguyen, T. T. Nguyen, J. Rothe, and A. Saffidine, "Positional scoring-based allocation of indivisible goods," *Auton. Agents Multi Agent Syst.*, vol. 31, no. 3, pp. 628–655, 2017.
[23] S. Barman, S. K. Krishnamurthy, and R. Vaish, "Greedy algorithms for maximizing nash social welfare," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018, pp. 7–13.
[24] G. Amanatidis, G. Birmpas, A. Filos-Ratsikas, A. Hollender, and A. A. Voudouris, "Maximum nash welfare and other stories about EFX," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, C. Bessiere, Ed. ijcai.org, 2020, pp. 24–30.
[25] A. Ben-Tal and A. Nemirovski, "On polyhedral approximations of the second-order cone," *Math. Oper. Res.*, vol. 26, no. 2, pp. 193–205, 2001.
[26] R. L. Graham, "Bounds on multiprocessing timing anomalies," *SIAM Journal of Applied Mathematics*, vol. 17, no. 2, pp. 416–429, 1969.
[27] B. Y. Wu, "An analysis of the LPT algorithm for the max-min and the min-ratio partition problems," *Theor. Comput. Sci.*, vol. 349, no. 3, pp. 407–419, 2005.
[28] J. Garg, P. Kulkarni, and R. Kulkarni, "Approximating nash social welfare under submodular valuations through (un)matchings," in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, S. Chawla, Ed. SIAM, 2020, pp. 2673–2687.
[29] E. A. Dinic and M. A. Kronrod, "An algorithm for the solution of the assignment problem," *Math. Dokl.*, vol. 10, no. 6, pp. 1324–1326, 1969.
[30] Y. T. Lee and A. Sidford, "Efficient inverse maintenance and faster algorithms for linear programming," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, V. Guruswami, Ed. IEEE Computer Society, 2015, pp. 230–249.
[31] S. Bouveret and J. Lang, "A general elicitation-free protocol for allocating indivisible goods," in *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, T. Walsh, Ed. IJCAI/AAAI, 2011, pp. 73–78.
[32] G. Amanatidis, E. Markakis, A. Nikzad, and A. Saberi, "Approximation algorithms for computing maximin share allocations," in *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, ser. Lecture Notes in Computer Science, M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, Eds., vol. 9134. Springer, 2015, pp. 39–51.
[33] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang, "The unreasonable fairness of maximum nash welfare," in *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, V. Conitzer, D. Bergemann, and Y. Chen, Eds. ACM, 2016, pp. 305–322.
[34] H. Aziz, G. Rauchecker, G. Schryen, and T. Walsh, "Algorithms for max-min share fair allocation of indivisible chores," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, S. P. Singh and S. Markovitch, Eds. AAAI Press, 2017, pp. 335–341.

Trung Thanh Nguyen received the M.S. degree in mathematics from the Institute of Mathematics, Vietnam Academy of Science and Technology in 2007, and the Ph.D. degree in computer science from Heinrich Heine University Duesseldorf, Germany, in 2013. He is currently an Associate Professor with the Faculty of Computer Science, Phenikaa University, Vietnam.
Email: thanh.nguyentrung@phenikaa-uni.edu.vn

Le Dang Nguyen received the M.S. degree in computer science from VNU University of Engineering and Technology in 2005, and the Ph.D. degree in applied mathematics from VNU University of Science, in 2016. He is currently a senior lecturer with the Faculty of Computer Science, Haiphong University, Vietnam.
Email: nguyenld@dhhp.edu.vn

Predicting Long Non-coding RNA-disease Associations using Multiple Features and Deep Learning

Van Tinh Nguyen[1,2], Dang Hung Tran[2]

[1] Hanoi University of Industry, Hanoi, Vietnam
[2] Hanoi National University of Education, Hanoi, Vietnam

Correspondence: Dang Hung Tran. Email:hungtd@hnue.edu.vn
Communication: received 11 June 2022, revised 01 August 2022, accepted 24 August 2022
DOI: 10.32913/mic-ict-research.v2022.n2.1069

***Abstract*: Various long non-coding RNAs have been shown to play crucial roles in different biological processes including cell cycle control, transcription, translation, epigenetic regulation, splicing, differentiation, immune response and so forth in the human body. Discovering lncRNA-disease associations promotes the awareness of human complex disease at molecular level and support the diagnosis, treatment and prevention of complex diseases. It is costly, laboratory and time-consuming to discover and verify lncRNA-disease associations by biological experiments. Therefore, it is crucial to develop a computational method to predict lncRNA-disease associations to save time and resources. In this paper, we proposed a new method to predict lncRNA-disease associations using multiple features and deep learning. Our method uses a weighted K-nearest known neighbors algorithm as a pre-processing step to eliminate the impact of sparsity data problem. And it combines the linear and non-linear features extracted by singular value decomposition and deep learning techniques, respectively, to obtain better prediction performance. Our proposed method achieves a decisive performance with the best AUC and AUPR values of 0.9702 and 0.8814, respectively, under LOOCV experiments. It is superior to other state-of-the-art SDLDA and NCPLDA methods in both AUC and AUPR evaluation metrics. It could be considered as a powerful tool to predict lncRNA-disease associations.**

Keywords: *lncRNA-disease associations prediction, weighted K nearest known neighbors, singular value decomposition, feature extraction, deep learning.*

I. INTRODUCTION

Long non-coding RNAs (lncRNAs) are a class of different non-coding RNAs (ncRNAs) which have more than 200 nucleotides in length [1]. For a long time, they were considered to be noise and have no biological function. However, a diversity of researches has shown that lncRNAs have crucial roles in different biological processes including cell cycle control, transcription, translation, epigenetic regulation, splicing, differentiation, immune response and so forth in the human body [2–6] It is not surprise that the abnormal expressions, mutations and dysregulations of lncRNAs can cause numerous complex human diseases [5, 7]. Certainly, inferring lncRNA-disease associations can contribute to the awareness of human complex diseases' mechanisms at lncRNA levels as well as to discover the prognosis markers for diagnosis, treatment and prevention of complex diseases [5–7]. In fact, the number of validated lncRNA-disease associations is very limited in comparison with the number of possible associations between lncRNAs and diseases. Furthermore, discovering lncRNA-disease associations by performing the biological wet-experiments is costly, laboratory and time-consuming [5, 7]. Therefore, it is urgent and important to develop effective and meaningful computational methods to predict relationships between lncRNAs and diseases.

Over the past few years, various computational methods have been developed to predict potential lncRNA-disease associations. They can roughly be divided into the following categories: similarity network-based, multiple data sources-based, machine learning-based and multi-model integration methods [8–11]. *Firstly*, the similarity network-based methods mainly rely on known lncRNA-disease associations information to infer potential associations between lncRNAs and diseases. They are built on the basic assumption that similar diseases are likely to be associated with functionally similar lncRNAs [9]. For examples, *Xiao et al.* [12] presented a BPLLDA method to predict relationships between lncRNAs and diseases relied on simple paths with limited lengths in a heterogeneous network. *Xiao et al.* [13] proposed a weight matrix of lncRNA-disease associations prediction (WLDAP) method which based on known lncRNA-disease associations to calculate Gaussian interaction profile kernel for lncRNAs

and diseases. Then, they inferred potential lncRNA-disease associations by using a weighting model to obtain similarity scores between lncRNAs and diseases. *Chen et al.* [14] inferred latent lncRNA-disease associations by a label-propagation algorithm and random projection on a heterogeneous network constructed from known lncRNA-disease associations network, disease semantic similarity network, lncRNA functional similariry network and Gaussian interaction profile kernel for lncRNAs and diseases. Generally, this kind's models achieve high prediction performance but they require the calculation of multiple similarities.

Secondly, the multiple data sources-based methods normally integrate multiple types of biological information to reveal latent associations between lncRNAs and diseases [8, 9]. For instances, *Ding et al.* [15] presented a model called TPGLDA based on a tripartite graph constructed from lncRNA-disease associations network, disease-gene associations networks to detect lncRNA-disease associations. *Liu et al.* [16] built a lncRNA prioritization model which integrates gene-disease interactions, gene expression profiles, and lncRNA expression profiles to prioritize novel disease-associated lncRNAs. *Nguyen* and *Tran* [17] proposed an improved computational method for prediction of lncRNA-disease associations based on collaborative filtering and resource allocation which used known lncRNA-disease associations, verified lncRNA-miRNA interactions and validated miRNA-disease associations to predict lncRNA-disease associations. These methods require multiple types of biological information which are not easy to identify in the heterogeneous data sources.

Thirdly, the machine learning-based methods employed machine learning algorithms to predict lncRNA-disease associations relied on training and unlabeled samples, verified and unknown associations, respectively. Formerly, numerous meachine learning algorithms have been implemented to forecast lncRNA-disease associations such as Naïve Bayesian classifier [18, 19], Random forest [20, 21], Bagging support vector machine (SVM) [22] and matrix factorization based methods [10, 23]. Each type of machine learning algorithms has its own strengths and weaknesses. For example, the Naïve Bayesian classifier is a simple probabilistic classifier, however, it requires a naïve independence assumption that any feature of a class is independent of the other features [18]. The Random forest algorithm requires obtaining the reliable negative samples that are difficult to be collected. Therefore, randomly selecting negative samples may influence the prediction performance [20]. The SVM in [22] bases on the assumption that positive unlabeled learning problems have a particular structure that leads to instability of classifiers. Among machine learning-based methods to predict lncRNA-disease associations, the matrix factorization methods can be considered as the most popular. Normally, traditional matrix factorization based methods only extract the linear features of lncRNA-disease associations by projecting the constructed matrix into sub-matrices in lower dimension spaces. However, in fact, the lncRNA-disease associations are very complex and non-linear. Thus, traditional matrix factorization-based methods can not model such complex and non-linear associations. Recently, deep learning methods which consitute a basic part of machine learning, are applied to gather the complex and non-linear features of lncRNA-disease associations by assuming that the linear and non-linear features can reinforce each other to obtain high quality features to improve prediction performance [8]. Up to now, various deep learning methods have been developed to predict lncRNA-disease associations such as SDLDA [8], LDAPM [24], DBNLDA [25] and so on.

Finally, the multi model integration-based methods which integrated multi models into an unified one to reduce the issues of single model and increase the prediction performance [26]. For examples, the combination of matrix factorization technique and recommendation system ideas to predict lncRNA-disease associations was used in LDASR method developed *Guo et al.* [27]. *Shi et al.* [28] inferred lncRNA-disease associationsa by a VGAELDA hybrid method which integrates variational inference and graph autoencoders.

Although computational methods for lncRNA-disease association prediction have attained significant performance. However, the number of known lncRNA-disease associations between lncRNAs and diseases is very limited in comparison with the number of possible associations among them. Therefore, most of computational methods for lncRNA-disease association prediction have to deal with the problem of sparsity data.

As mentioned before, deep learning techniques can reinforce linear and non-linear features each other to improve prediction performance. Additionally, the similarity network based methods normally reach high accuracy and the problem of sparsity data needs to be solved. Hence, in this paper, we proposed a new method based on multiple features and deep learning to predict associations between lncRNAs and diseases. The proposed method based on lncRNA functional similarity, disease semantic similarity and known lncRNA-disease associations as input. Then, a weighted K-nearest known neighbors algorithm is employed as a pre-processing step to solve the problem of sparsity data. Next, singular value decomposition and deep learning techniques are used to extract linear and non-linear features of lncRNAs and diseases. The extracted linear and non-linear features are combined into a new vector before a final fully connected layer is used to forecast final prediction. The results showed that our proposed method

achieves a decisive performance with the best AUC and AUPR values of 0.9702 and 0.8814, respectively, under LOOCV experiments. It is superior to other state-of-the-art SDLDA [8] and NCPLDA [9] methods in both AUC and AUPR evaluation metrics. In case studies of Breast cancer and Hepatocellular Carcinoma, it recalled 8 and 7 known associations and it inferred 2 and 3 new associations out of top 10 predicted associations, respectively. Fortunately, all of the new predicted associations have been verified in biomedical literature. Therefore, our new proposed method could be considered as a forceful tool to predict lncRNA-disease associations.

II. MATERIALS AND METHOD

1. Materials

a) Human lncRNA-disease associations

In this paper, the data sets used in experiments were collected from the LncRNADisease database including three versions (June-2012, January-2014 and June-2015 versions). After removing the associations with abnormal lncRNA or disease names as well as merging all duplicated records, we obtained three data sets as follows. Firstly, the June-2012 version data set (marked as DS1) contained 276 lncRNA-disease associations between 112 lncRNAs and 150 diseases. Secondly, the January-2014 version data set (marked as DS2) included 319 associations between 131 lncRNAs and 169 diseases. And finally, the June-2015 version data set (marked as DS3) consisted of 621 associations between 285 lncRNAs and 226 diseases. The association density of three data sets are 1.643%, 1.441% and 0.964% for DS1, DS2 and DS3, respectively. All of these data sets were already used in the work of *Li et al.* [9].

For accessibility, we used an adjacency matrix $A^{LD} \in R^{m \times n}$ to represent lncRNA-disease associations where m is the number of rows in the matrix A^{LD} or the number of lncRNAs, n means the number of columns in the matrix A^{LD} or the number of diseases, and $A^{LD}(i, j) = 1$ if the association between lncRNA l_i and disease d_j has been validated and $A^{LD}(i, j) = 0$ for other cases.

b) Disease semantic similarity

In this paper, disease semantic similarity was estimated according to Wang et al.'s method [29]. Concretely, we collected the relationships of different diseases based on the hierarchical directed acrylic graphs (DAGs) by downloading MeSH descriptors from the National Library of Medicine (http://www.ncbi.nlm.nih.gov/) to measure the similarity among diseases. Specifically, for a disease d, its directed acrylic graph is given by $DAG(d) = (d, A_d, E_d)$, where A_d reflects the set of the disease d's ancestors and d itself, and E_d means the set of edges which point to child nodes from parent nodes in the MeSH tree. Hence, the semantic contribution of disease k to disease d is as:

$$D_d(k) = \begin{cases} k & if\, k = d \\ max\{\Delta \times D_d(k') | k' \in \text{children of k}\} & if\, k \neq d \end{cases} \quad (1)$$

Where Δ indicates a predefined semantic contribution factor with values in range (0, 1). Similar to *Wang et al.* [29], in this paper, we set Δ equal to 0.5. We computed the semantic similarity among diseases relied on the assumption that two diseases having larger parts in their DAGs tend to have higher semantic similarity as:

$$DSS(d_i, d_j) = \frac{\sum_{k \in A_{d_i} \cap A_{d_j}} (D_{d_i}(k) + D_{d_j}(k))}{\sum_{k \in A_{d_i}} D_{d_i}(k) + \sum_{k \in A_{d_j}} D_{d_j}(k)} \quad (2)$$

c) LncRNA functional similarity

Based on the assumption that functionally similar lncRNAs are often associated with similar diseases, in this work, we calculated functional similarity between two lncRNAs by measuring the semantic similarity of two disease sets which are linked to these two lncRNAs [9, 30]. Explicitly, we assumed that lncRNA l_i and lncRNA l_j were associated with n_1 and n_2 diseases, respectively, then the similarity between lncRNA l_i and lncRNA l_j can be computed as in the following equations:

$$LFS(l_i, j_j) = \frac{\sum_{d \in D_{l_j}} SS(d, D(_{l_i})) + \sum_{d \in D_{l_i}} SS(d, D(_{l_j}))}{n_1 + n_2} \quad (3)$$

$$SS(d_k, D(l_i)) = max_{d \in D(l_i)}(DSS(d_k, d)) \quad (4)$$

where LFS is the lncRNA functional similarity matrix and $D(l_i)$ means the disease set associated with lncRNA l_i.

It is important to remember that all of the known lncRNA-disease associations matrix A^{LD}, disease semantic similarity matrix DSS and lncRNA functional similarity LFS matrix are sparse. Therefore, in our proposed method, we have to solve this problem.

2. Method

a) Method overview

In overview, the proposed method contains following stages. At the first stage, we take disease semantic similarity (DSS) information, lncRNA functional similarity (LFS) information and known lncRNA-disease associations A^{LD} as inputs. And we applied a weighted K-nearest known neighbors ($WKNKN$) algorithm as a pre-processing step to solve the sparsity data problem to obtain a new lncRNA-disease association matrix A^{LD-new}. At second stage, a singular value decomposition (SVD) technique is used to decompose matrix A^{LD-new} into three matrices U, $\sum$, and

V^T to extract the linear features of lncRNAs and diseases. The rows of U indicate the linear features of lncRNAs where the columns of V^T reflect the linear features of diseases. During the third stage, we extract non-linear features of lncRNAs and diseases with deep learning technique. In deep learning part, each row of A^{LD-new} is considered as a lncRNA raw feature vector while each column of A^{LD-new} is considered as a disease raw feature vector. For each interaction in A^{LD-new} matrix, its corresponding raw feature vectors are fed into two fully connected layers with non-linear activation function to extract non-linear features. At fourth stage, the linear and non-linear features are concatenated to $Lnc_{new-fea-vec}$ and $Dis_{new-fea-vec}$ for lncRNA and disease, respectively. Next, a Hadamard product operation is employed to fuse two new feature vectors. And finally, the final prediction is obtained by applying a sigmoid activation function. The workflow of the proposed method is illustrated in Figure 1.

b) Weighted K-nearest known neighbors algorithm

As mentioned before, all of three matrices A^{LD}, DSS and LFS are sparse. To solve this problem, in this paper, we employed a weighted K-nearest known neighbors algorithm as a pre-processing step to reduce the impact of sparsity data problem and to eliminate the unknown associations in lncRNA-disease association set. It relied on the known neighbors' information of nodes by seeing the fact that many of the non-interacting lncRNA-disease pairs in A^{LD} which are unknown cases but they could potentially be true associations. Specifically, the $WKNKN$ algorithm contains following steps. *Firstly*, for each lncRNA l_i, we took its functional similarities with K known lncRNAs which are nearest to l_i and their corresponding interaction profiles to estimate the interaction likelihood profile for lncRNA l_i. *Secondly*, for each disease d_j, we chose the semantic similarities with K known diseases which are nearest to d_j and their corresponding interaction profiles to evaluate the interaction likelihood profile for disease d_j. *Thirdly*, if $A^{LD}_{ij} = 0$, we changed it by averaging the two interaction likelihood profiles. And *finally*, we used a threshold to classify the interaction likelihood profiles into two classes labeled as 0 and 1. The Algorithm 1 contains the pseudocode that describes the above steps in detail in which t is a decay term where $t \leq 1$, θ means a threshold, normally θ is set to 0.5, and $KNN()$ returns the K-nearest known neighbors in descending order based on their similarities to lncRNA l_i or disease d_j.

c) Obtaining linear features by decomposing A^{LD-new} with singular value decomposition technique

Singular value decomposition (SVD) is one of the most polular matrix factorization methods in recommendation systems [8]. It is used in our work to gather linear features

Algorithm 1: WKNKN algorithm.

Inputs: Matrices $A^{LD} \in R^{m \times n}$, $LFS \in R^{m \times m}$ and $DSS \in R^{n \times n}$, neighborhood size K, decay term t, threshold θ.
Outputs: new adjacency association matrix A^{LD-new}.
begin
  $Al = Ad = 0$ # initialize two temporary mactrices ;
  # for lncRNAs
  for $l = 1$ to m **do**
    lnn=KNN(l, LFS, K);
    for $i = 1$ to K **do**
      $w_i = t^{i-1} LFS(l, lnn_i)$;
    end
    $Z_l = \sum_{i=1}^{K} LFS(l, lnn_i)$;
    $Al(l) = \frac{1}{Z_l} \sum_{i=1}^{K} w_i A^{LD}(lnn_i)$;
  end
  # for diseases
  for $d = 1$ to n **do**
    dnn=KNN(d, DSS, K);
    for $i = 1$ to K **do**
      $w_i = t^{i-1} DSS(d, dnn_i)$;
    end
    $Z_d = \sum_{i=1}^{K} DSS(d, dnn_i)$;
    $Ad(d) = \frac{1}{Z_d} \sum_{i=1}^{K} w_i A^{LD}(dnn_i)$;
  end
  $Ald = \frac{Al+Ad}{2}$;
  $A^{LD-new} = max(A^{LD}, Ald)$;
  # classify A^{LD-new} by threshold θ.
  for $i = 1$ to m **do**
    for $j = 1$ to n **do**
      if $A^{LD-new} \geq \theta$ **then**
        $A^{LD-new}[i, j] = 1$;
      else
        $A^{LD-new}[i, j] = 0$;
      end
    end
  end
  return A^{LD-new};
end

of lncRNAs and diseases. The details of the process to obtain linear features of lncRNAs and diseases are described as follows. Let A^{LD-new} be the new association matrix of lncRNAs and diseases, the SVD of A^{LD-new} matrix is a factorization of three matrices U, $\sum$ and V^T as:

$$A^{LD-new} \approx U \sum V^T \tag{5}$$

where $U \in R^{m \times k}$ is a real matrix, $\sum \in R^{k \times k}$ is a diagonal matrix with non-negative square roots of the eigenvalues of the product $R^T R$ on the diagonal and $V^T \in R^{k \times n}$ is a real matrix. The U_i represents a linear feature vector of lncRNA l_i and the V_j^T is a linear feature vector of disease d_j. Figure 2 illustrates the process of SVD to obtain linear features of lncRNAs and diseases in our method.

d) Obtaining non-linear features with deep learning

As can be known, deep learning techniques are the basic and powerful techniques of machine learning. With multiple

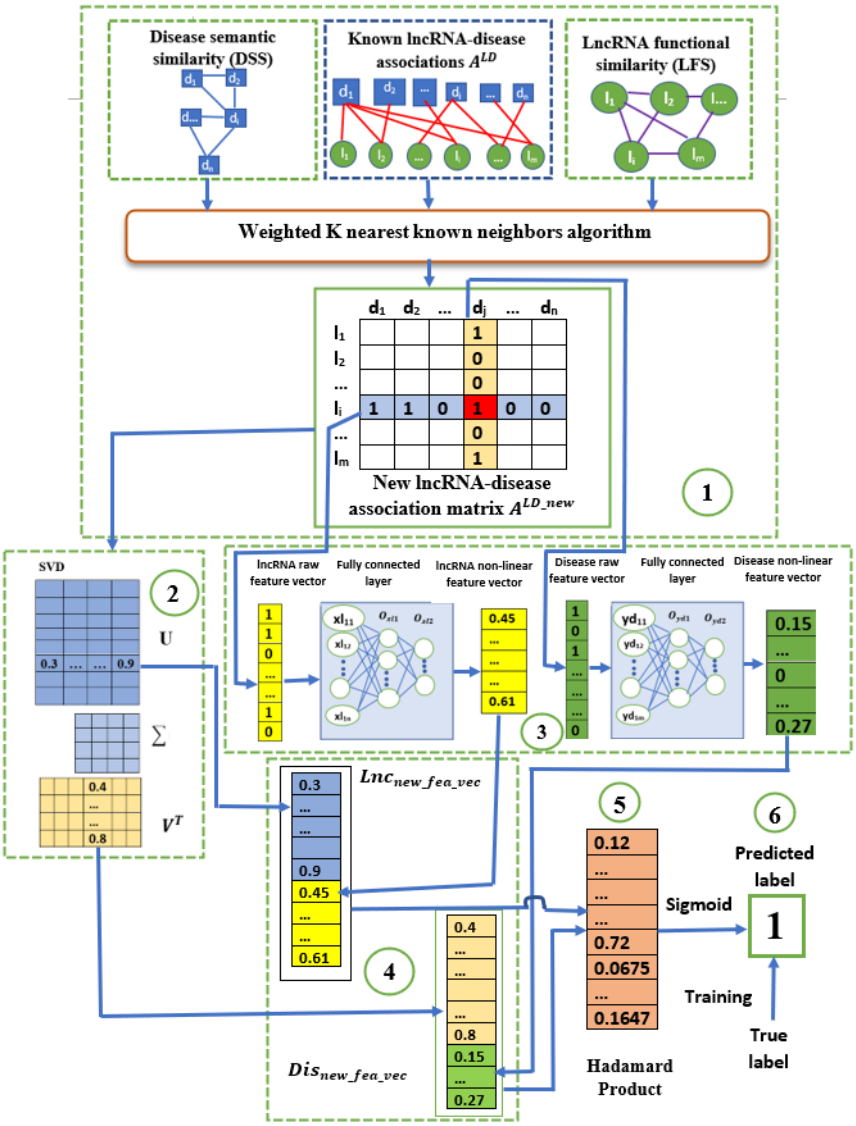


Figure 1. The workflow of the proposed method.

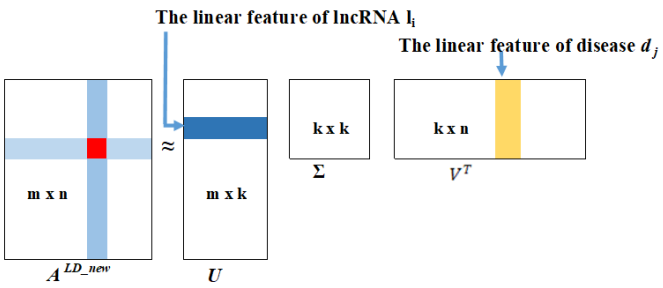


Figure 2. The process of SVD to obtain linear features of lncRNAs and diseases.

levels of representation, deep learning techniques can learn non-linear, complex and useful features. Therefore, in this work, we used a deep learning technique to obtain non-linear features of lncRNAs and diseases. Specifically, we considered the new association matrix A^{LD-new} as input of deep learning process. Similar to the work of *Zeng et al.* [8], we considered each row of the A^{LD-new} matrix as raw feature vector of lncRNA l_i while each column of A^{LD-new} matrix as raw feature vector of disease d_j. For each intersection of the A^{LD-new} matrix, its corresponding raw feature vectors are fed into two fully connected layers with a non-linear activation function to gather non-linear features of lncRNA and disease. Formally, the lncRNA and disease raw feature vectors are denoted as xl and yd, respectively. They are fed into the first fully connected layer to output O_{xl_1} and O_{yd_1} as:

$$O_{xl_1} = \delta(W_{xl_1} xl + b_{xl_1}) \quad (6)$$

$$O_{yd_1} = \delta(W_{yd_1} yd + b_{yd_1}) \quad (7)$$

where δ is the non-linear activation function, W_{xl_1} and W_{yd_1} are the first fully connected layer's weight matrices, b_{xl_1} and b_{yd_1} are bias terms of the first fully connected layer. Next, O_{xl_1} and O_{yd_1} are fed into the second fully connected layer to output the final non-linear features $O_{xl_2}$and O_{yd_2}as in the following equations:

$$O_{xl_2} = \delta(W_{xl_2}O_{xl_1} + b_{xl_2}) \tag{8}$$

$$O_{yd_2} = \delta(W_{yd_2}O_{yd_1} + b_{yd_2}) \tag{9}$$

where W_{xl_2} and W_{yd_2} are the second fully connected layer's weight matrices, b_{xl_2} and b_{yd_2} are bias terms of the second fully connected layer.

In each fully connected layer, we applied the Rectified Linear Unit ($ReLU$) activation function to obtain non-linear features as in the follow equation:

$$ReLU(x) = max(0,\ x) \tag{10}$$

e) Combining linear features and non-linear features

After obtaining linear and non-linear feature vectors by SVD and deep learning technique, respectively, we firstly concatenated them into $Lnc_{new-fea-vec}$ and $Dis_{new-fea-vec}$ for lncRNAs and diseases, respectively.

$$Lnc_{new-fea-vec} = \begin{bmatrix} U_i \\ O_{xl_2} \end{bmatrix} \tag{11}$$

$$Dis_{new-fea-vec} = \begin{bmatrix} V_j^T \\ O_{yd_2} \end{bmatrix} \tag{12}$$

where [] indicates concatenation operation.
Then, we applied a Hadamard product operation to integrate two vectors $Lnc_{new-fea-vec}$ and $Dis_{new-fea-vec}$ into a new integrated vector:

$$Vec_{new-integrated} = HadamardProduct(Lnc_{new-fea-vec}, Dis_{new-fea-vec}) \tag{13}$$

f) Obtaining final prediction

Finally, the new integrated vector $Vec_{new-integrated}$ is fed into a fully connected layer with a sigmoid activation function to obtain final prediction.

$$Sigmoid(x) = \frac{1}{1 + \exp(-x)} \tag{14}$$

$$A^{LD-final}[i, j] = Sigmoid(Vec_{new-integrated}) \tag{15}$$

We used a binary cross-entropy function in our model as the loss function as follows:

$$Loss = \sum [A^{LD-new} \log(A^{LD-final}[i, j]) + (1 - A^{LD-new}) \log(1 - A^{LD-final}[i, j])] + \gamma(|\partial^2|) \tag{16}$$

where ∂ is the weight vector, γ is a weight to balance between the experimental risk and regularized term.

III. RESULTS

1. Performance evaluation

To evaluate the proposed method's prediction performance, in this paper, we already performed Leave-one-out-cross-validation ($LOOCV$) experiments on all three data sets as described in Materials section. In $LOOCV$ experiments, each entry value equal to 1 in A^{LD} matrix is taken as test sample and the remaining associations are considered as training samples. After implementing the proposed model, we gathered $A^{LD-final}$ matrix. We repeatedly performed the process and then used the obtained results to calculate true positive rate (TPR), false positive rate (FPR), Precision and Recall as in the followings:

$$TPR = \frac{TP}{TP + FN} \tag{17}$$

$$FPR = \frac{FP}{FP + TN} \tag{18}$$

$$Precision = \frac{TP}{TP + FP} \tag{19}$$

$$Recall = \frac{TP}{TP + FN} \tag{20}$$

where TP indicates the number of true positive samples, FN means the number of false negative samples, FP is the number of false positive samples and TN reflects the number of true negative samples.

After that, we figured out Receiver Operating Characteristic (ROC) curve [31] and Precision-Recall (PR) curve [32] of proposed method on each data set. Parallely, we computed AUC (Area under ROC curve) and $AUPR$ (Area under Precision-Recall curve) values of proposed method on each data set. Figure 3 illustrates ROC curves and AUC values and Figure 4 shows PR curves and $AUPR$ values of proposed method on three data sets when the value of K in $WKNKN$ algorithm is set to 8. As can be seen, AUC values of proposed method on three data sets are similar and the best $AUPR$ value is achieved on the DS3 data set. Therefore, we chose the AUC and $AUPR$ values of proposed method on DS3 to compare with the performance of other methods in the next section.

2. Performance in comparison with other related methods

To emphasize the performance of our proposed method in comparison with other previous related methods, we compared our proposed method prediction performances, on the same data set DS3, with SDLDA and NCPLDA methods [8, 9]. Figure 5 shows the ROC curves and AUC values of the related methods and our proposed method on the same data set DS3. Figure 6 represents the PR curves and $AUPR$ values of the related methods and our

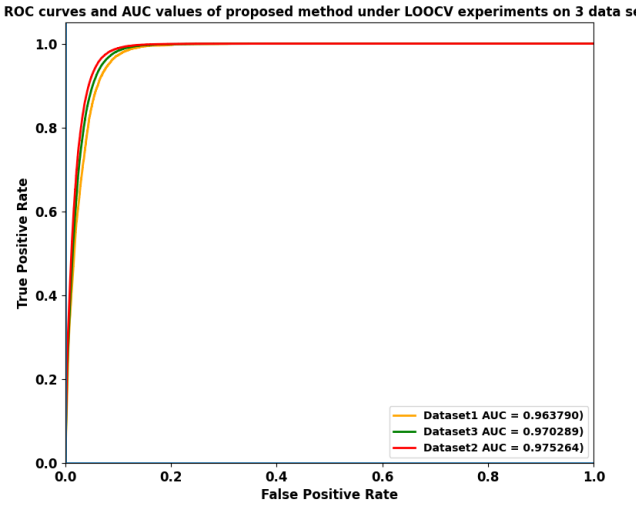


Figure 3. The ROC curves and AUC values of proposed method on three data sets when K in WKNKN is set to 8.

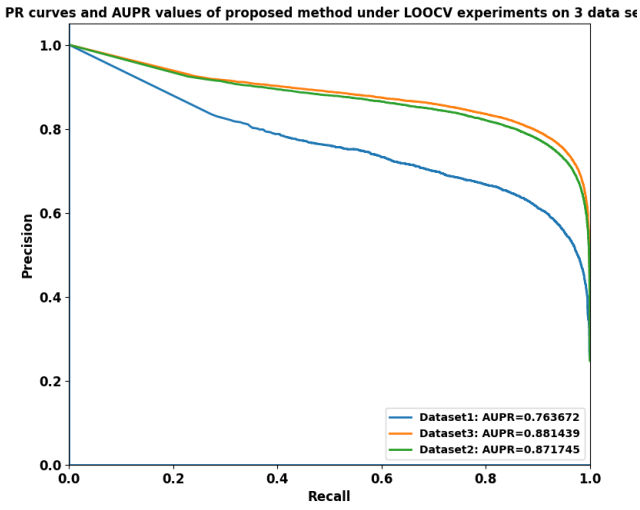


Figure 4. The PR curves and AUPR values of proposed method on three data sets when K in WKNKN is set to 8.

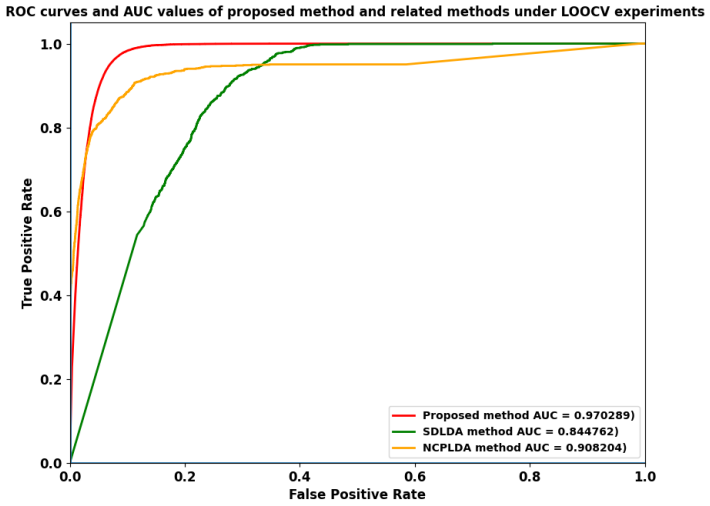


Figure 5. ROC curves and AUC values of the proposed method and related methods under LOOCV experiments.

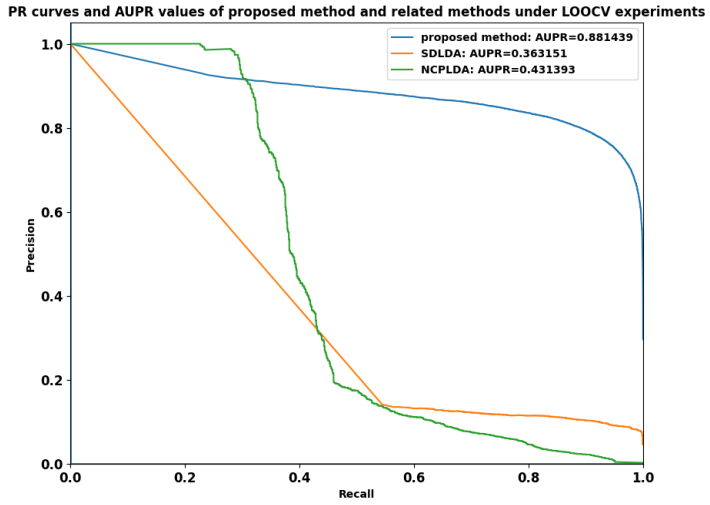


Figure 6. PR curves and AUPR values of the proposed method and related methods under LOOCV experiments.

proposed method on the same data set DS3.

As can be seen in Figure 5 and Figure 6, our new proposed method is superior to previous related methods in both AUC and AUPR terms. The highest AUPR value indicates that our proposed method can recall more known associations by reducing the impact of sparsity data problem.

3. Effects of parameters

a) Parameters of WKNKN algorithm

Considering the fact that there are some unknown lncRNA-disease associations in the matrix A^{LD}. We used the *WKNKN* algorithm as a pre-processing step to eliminate unknown values in lncRNA-disease association set relied on their known neighbors. In this algorithm, the K parameter means the number of nearest known neighbors, t reflects a decay term where $t \leq 1$, θ is a threshold. In this study, we mainly targeted on the impact of number

of nearest known neighbors to scale down the effects of sparsity data problem. It is true that when the more nearest known neighbors were selected, the more associations between lncRNAs and diseases would be added into the heterogeneous network and then the impact of sparity data problem would be reduced. However, the imbalanced data problem would again appears when the number of added associations becomes too big. Therefore, in our experiments, we repeatedly changed the value of K and t to choose the optimal result. And it showed that AUC and $AUPR$ of our proposed method reached the best values when $K = 8$ and $t = 0.7$ while θ is fixed to 0.5 as be mentioned before.

b) Implementation details and hyper-parameters

In this paper, the SVD is executed with Scipy library, the lncRNA and disease linear features are represented by two 64-dimensional vectors. The deep learning framework use Tensor flow [33]. Non-linear features of lncRNAs and diseases are extracted by ReLU activation function.

For hyper-parameters, after repeatedly performing the experiments, the optimal values of parameters are as follows. The first and second fully connected layers used 48 and 32 neurons, respectively. The dropout rate and regularization parameter γ are set to 0.05 and 0.003, respectively. The optimizer used the adaptive moment estimation (Adam). The initinal learning rate is set to 0.001. Finally, the number of neurons in the last fully connected layer is set to 48.

4. Case studies

To encourage our method's prediction reliability by evaluating AUC and $AUPR$ values under $LOOCV$ experiments, we also tested some case studies to observe the most predicted lncRNAs for any given disease. In the following sections, we showed top 10 predicted lncRNAs for case studies' results of Breast Cancer and Hepatocellular Carcinoma, respectively.

a) Breast Cancer's case study

Breast Cancer is also known as Breast Neoplasms. It is a malignant tumor that has a high mortality rate and mostly occurs in women [34]. Over the past few years, different lncRNAs have been reported to be associated with Breast Cancer. For examples, lncRNA H19 has been indicated to be associated with poor prognosis in breast cancer patients and promotes cancer stemness [35]. High expression of lncRNA MALAT1 is related to breast cancer relapse and may play a role in tumor progression [36]. The case study of Breast Cancer on data set DS3 showed that our proposed method recalled 8 known associations and it inferred 2 new associations out of top 10 predicted Breast Cancer-related lncRNAs. The Table I showed top 10 predicted Breast cancer-related lncRNAs and all predicted associations have been known or verified by biomedical literature.

TABLE I
TOP 10 PREDICTED BREAST CANCER-RELATED LNCRNAS

LncRNA	Rank	Known before	Evidences (PMID)
CDKN2B-AS1	1	1	Known association
GAS5	2	1	Known association
H19	3	1	Known association
HOTAIR	4	1	Known association
PVT1	5	1	Known association
MALAT1	6	1	Known association
MEG3	7	1	Known association
MIAT	8	0	29100300
NEAT1	9	0	32273717
XIST	10	1	Known association

TABLE II
TOP 10 PREDICTED HEPATOCELLULAR CARCINOMA-RELATED LNCRNAS

LncRNA	Rank	Known before	Evidences (PMID)
HOTAIR	1	1	Known association
MALAT1	2	1	Known association
H19	3	1	Known association
CDKN2B-AS1	4	0	32801906
PMEG3	5	1	Known association
UCA1	6	1	Known association
CCND1	7	0	32443384
PCAT29	8	1	Known association
GAS5	9	0	25120813; 26163879
AIR	10	1	Known association

b) Hepatocellular Carcinoma's case study

Hepatocellular carcinoma (HCC) is the most common primary liver malignancy and it is a leading cause of cancer-related death over worldwide [37]. Recently, many lncRNAs have been indicated to be related to hepatocellular carcinoma. For instances, lncRNA MALAT1 promotes hepatocellular carcinoma development by SRSF1 up-regulation and mTOR activation [38]. Upregulated lncRNA-UCA1 promotes to progression of hepatocellular carcinoma through inhibition of miR-216b and activation of FGFR1/ERK signaling pathway [39]. The case study of Hepatocellular carcinoma indicated that our proposed method recalled 7 known associations and derived 3 new associations out of top 10 predicted Hepatocellular carcinoma-related lncRNAs. Table II demonstrated top 10 predicted Hepatocellular carcinoma-related lncRNAs and all predicted associations have been known or verified by biomedical literature.

IV. CONCLUSION AND DISCUSSIONS

Various lncRNAs have been shown to play crucial roles in different biological processes including cell cycle control, transcription, translation, epigenetic regulation, splicing, differentiation, immune response and so forth in the

human body. Discovering associations between lncRNAs and diseases promotes the awareness of human complex diseases' mechanisms at molecular levels and support the diagnosis, treatment and prevention of complex diseases [5–7] However, the process of finding and verifying lncRNA-disease associations is costly, laboratory and time-consuming. Therefore, developing a forceful and meaningful computational method to forecast potential lncRNA-disease associations is become urgent in some recent years. In this paper, we proposed a new method to predict lncRNA-disease associations using multiple features and deep learning. It achieves a decisive performance with the best *AUC* and *AUPR* values of 0.9702 and 0.8814, respectively, under LOOCV experiments on the data set DS3 as mentioned before. It outperforms other state-of-the-art methods in both *AUC* and *AUPR* evaluation metrics.

The desirable prediction performance of our proposed method is contributed by the following factors. *Firstly*, by using multiple features, we already take into account the high accuracy advantage of the similarity network-based methods. *Secondly*, by using a *WKNKN* algorithm as a pre-processing step, we already reduced the impact of sparsity data problem. And *finally*, the combination of linear features and non-linear features extracted by singular value decomposition technique and deep learning, respectively, can reinforce each other to better predict potential lncRNA-disease associations [8]. However, there are too many parameters in deep learning model. Therefore, it is difficult to tune all hyper-parameters and try to do all possible cases of experiments.

ACKNOWLEDGMENTS

This research was supported by the Vietnam Ministry of Education and Training, Project No. B2022-SPH-04.

REFERENCES

[1] E. S. Lander, "Ribosome profiling provides evidence that large non-coding RNAs do not encode proteins," *Cell*, vol. 154, no. 1, pp. 240–251, 2014, doi: 10.1016/j.cell.2013.06.009.Ribosome.

[2] K. C. Wang and H. Y. Chang, "Molecular mechanisms of long noncoding RNAs," *Molecular*, vol. 43, no. 6, pp. 904–914, 2011, doi: 10.1016/j.molcel.2011.08.018.

[3] M. Guttman and J. L. Rinn, "Modular regulatory principles of large non-coding RNAs," *Nature*, vol. 482, no. 7385, pp. 339–346, 2012, doi: 10.1038/nature10887.

[4] V. Mohanty, Y. Gökmen-Polar, S. Badve, and S. C. Janga, "Role of lncRNAs in health and disease-size and shape matter," *Brief. Funct. Genomics*, vol. 14, no. 2, pp. 115–129, 2015, doi: 10.1093/bfgp/elu034.

[5] X. Chen, C. C. Yan, X. Zhang, and Z. H. You, "Long non-coding RNAs and complex diseases: From experimental results to computational models," *Brief. Bioinform.*, vol. 18, no. 4, pp. 558–576, 2017, doi: 10.1093/bib/bbw060.

[6] X. Li, J. Xu, Y. Xiao, and S. Ning, Non-coding RNAs in Complex Diseases. Springer Singapore, 2018.

[7] X. Lei et al., "A comprehensive survey on computational methods of non-coding RNA and disease association prediction," *Brief. Bioinform.*, vol. 00, no. August, pp. 1–31, 2020, doi: 10.1093/bib/bbaa350.

[8] M. Zeng et al., "SDLLDA: lncRNA-disease association prediction based on singular value decomposition and deep learning," *Methods*, vol. 179, no. February, pp. 73–80, 2020, doi: 10.1016/j.ymeth.2020.05.002.

[9] G. Li et al., "Prediction of LncRNA-Disease Associations Based on Network Consistency Projection," *IEEE Access*, vol. 7, pp. 58849–58856, 2019, doi: 10.1109/ACCESS.2019.2914533.

[10] Z. Xuan, J. Li, J. Yu, X. Feng, B. Zhao, and L. Wang, "A probabilistic matrix factorization method for identifying lncRNA-disease associations," *Genes (Basel).*, vol. 10, no. 2, 2019, doi: 10.3390/genes10020126.

[11] G. Xie, Z. Huang, Z. Liu, Z. Lin, and L. Ma, "NCPHLDA: A novel method for human lncRNA-disease association prediction based on network consistency projection," *Mol. Omi.*, vol. 15, no. 6, pp. 442–450, 2019, doi: 10.1039/c9mo00092e.

[12] X. Xiao et al., "BPLLDA: Predicting lncRNA-disease associations based on simple paths with limited lengths in a heterogeneous network," *Front. Genet.*, vol. 9, no. OCT, pp. 1–11, 2018, doi: 10.3389/fgene.2018.00411.

[13] G. Xie, L. Wu, Z. Lin, and J. Cui, "WLDAP: A computational model of weighted lncRNA-disease associations prediction," *Phys. A Stat. Mech. its Appl.*, vol. 558, p. 124765, 2020, doi: 10.1016/j.physa.2020.124765.

[14] M. Chen, Y. Deng, A. Li, and Y. Tan, "Inferring Latent Disease-lncRNA Associations by Label-Propagation Algorithm and Random Projection on a Heterogeneous Network," *Front. Genet.*, vol. 13, no. February, pp. 1–11, 2022, doi: 10.3389/fgene.2022.798632.

[15] L. Ding, M. Wang, D. Sun, and A. Li, "TPGLDA: Novel prediction of associations between lncRNAs and diseases via lncRNA-disease-gene tripartite graph," *Sci. Rep.*, vol. 8, no. 1, pp. 1–11, 2018, doi: 10.1038/s41598-018-19357-3.

[16] M. X. Liu, X. Chen, G. Chen, Q. H. Cui, and G. Y. Yan, "A computational framework to infer human disease-associated long noncoding RNAs," *PLoS One*, vol. 9, no. 1, 2014, doi: 10.1371/journal.pone.0084408.

[17] V. T. Nguyen and D. H. Tran, "An improved computational method for prediction of lncRNA-disease associations based on collaborative filtering and resource allocation," *2021 13th Int. Conf. Knowl. Syst. Eng.*, pp. 1–6, 2021, doi: 10.1109/kse53942.2021.9648632.

[18] J. Yu, P. Ping, L. Wang, L. Kuang, X. Li, and Z. Wu, "A novel probability model for lncRNA–disease association prediction based on the naïve bayesian classifier," *Genes (Basel).*, vol. 9, no. 7, 2018, doi: 10.3390/genes9070345.

[19] J. Yu, Z. Xuan, X. Feng, Q. Zou, and L. Wang, "A novel collaborative filtering model for LncRNA-disease association prediction based on the Naïve Bayesian classifier," *BMC Bioinformatics*, vol. 20, no. 1, pp. 1–13, 2019, doi: 10.1186/s12859-019-2985-0.

[20] D. Yao, X. Zhan, X. Zhan, C. K. Kwoh, P. Li, and J. Wang, "A random forest based computational model for predicting novel lncRNA-disease associations," *BMC Bioinformatics*, vol. 21, no. 1, pp. 1–18, 2020, doi: 10.1186/s12859-020-3458-1.

[21] R. Zhu, Y. Wang, J. X. Liu, and L. Y. Dai, "IPCARF: improving lncRNA-disease association prediction using incremental principal component analysis feature selection and a random forest classifier," *BMC Bioinformatics*, vol. 22, no. 1. 2021, doi: 10.1186/s12859-021-04104-9.

[22] W.Lan et al., "LDAP: a web server for lncRNA-disease association prediction," *Bioinformatics*, vol.33, pp458-460,

2017, doi: 10.1093/bioinformatics/btw639.
[23] G. Fu, J. Wang, C. Domeniconi, and G. Yu, "Matrix factorization-based data fusion for the prediction of lncRNA-disease associations," *Bioinformatics*, vol. 34, no. 9, pp. 1529–1537, 2018, doi: 10.1093/bioinformatics/btx794.
[24] N. Fraidouni and G. Zaruba, "A Matrix Completion Approach for Predicting lncRNA-disease association," *Int'l Conf. Bioinforma. Comput. Biol.*, pp. 1–6, 2019.
[25] M. Madhavan and G. Gopakumar, "DBNLDA: Deep Belief Network based representation learning for lncRNA-disease association prediction," *Appl. Intell.*, vol. 52, no. 5, pp. 5342–5352, 2022, doi: 10.1007/s10489-021-02675-x.
[26] C. Yan et al., "Computational Methods and Applications for Identifying Disease-Associated lncRNAs as Potential Biomarkers and Therapeutic Targets," *Mol. Ther. - Nucleic Acids*, vol. 21, pp. 156–171, 2020, doi: 10.1016/j.omtn.2020.05.018.
[27] Z. H. Guo, Z. H. You, Y. Bin Wang, H. C. Yi, and Z. H. Chen, "A Learning-Based Method for LncRNA-Disease Association Identification Combing Similarity Information and Rotation Forest," *iScience*, vol. 19, pp. 786–795, 2019, doi: 10.1016/j.isci.2019.08.030.
[28] Z. Shi, H. Zhang, C. Jin, X. Quan, and Y. Yin, "A representation learning model based on variational inference and graph autoencoder for predicting lncRNA-disease associations," *BMC Bioinformatics*, vol. 22, no. 1, pp. 1–20, 2021, doi: 10.1186/s12859-021-04073-z.
[29] D. Wang, J. Wang, M. Lu, F. Song, and Q. Cui, "Inferring the human microRNA functional similarity and functional network based on microRNA-associated diseases," *Bioinformatics*, vol. 26, no. 13, pp. 1644–1650, 2010, doi: 10.1093/bioinformatics/btq241.
[30] J. Sun et al., "Inferring novel lncRNA-disease associations based on a random walk model of a lncRNA functional similarity network," *Mol. Biosyst.*, vol. 10, no. 8, pp. 2074–2081, 2014, doi: 10.1039/c3mb70608g.
[31] H.-T. K., "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation," *Casp. J Intern Med*, vol. 4(2), pp. 627–635, 2013.
[32] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLoS One*, vol. 10, no. 3, p. e0118432., 2015, doi: 10.1371/journal.pone.0118432.
[33] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," *12th USENIX Symp. Oper. Syst. Des. Implement.*, pp. 265–283, 2016, doi: 10.1007/978-1-4842-6418-8-2.
[34] H. Jin et al., "lncRNA and breast cancer: Progress from identifying mechanisms to challenges and opportunities of clinical treatment," *Mol. Ther. - Nucleic Acids*, vol. 25, no. September, pp. 613–637, 2021, doi: 10.1016/j.omtn.2021.08.005.
[35] H. Shima et al., "Lnc RNA H19 is associated with poor prognosis in breast cancer patients and promotes cancer stemness," *Breast Cancer Res. Treat.*, vol. 170, no. 3, pp. 507–516, 2018, doi: 10.1007/s10549-018-4793-z.
[36] Z. Wang et al., "High expression of long non-coding RNA MALAT1 in breast cancer is associated with poor relapse-free survival," *Breast Cancer Res Trea*, vol. 171, no. 2, pp. 261–271, 2018, doi: 10.1007/s10549-018-4839-2.High.
[37] X. Xu et al., "The role of MicroRNAs in hepatocellular carcinoma," *J. Cancer*, vol. 9, no. 19, pp. 3557–3569, 2018, doi: 10.7150/jca.26350.
[38] P. Malakar et al., "Long Noncoding RNA MALAT1 Promotes Hepatocellular Carcinoma Development by SRSF1 Upregulation and mTOR Activation," *Cancer Res*, vol. 77, no. 5, pp. 1155–1167, 2018, doi: 10.1158/0008-5472.CAN-16-1508.Long.
[39] F. Wang et al., "Upregulated lncRNA-UCA1 contributes to progression of hepatocellular carcinoma through inhibition of miR-216b and activation of FGFR1/ERK signaling pathway," *Oncotarget*, vol. 6, no. 10, pp. 7899–7917, 2015, doi: 10.18632/oncotarget.3219.

Van Tinh Nguyen received B.S. and M.S degree in Information Technology from Hanoi University of Science and Technology, Hanoi, Vietnam in 2001 and 2006, respectively. He is a lecturer in the Falculty of Information technology, Hanoi University of Industry. Curently, he is a Ph.D. student of Hanoi National University of Education. His research includes bioinformatics, data mining, data representation and machine learning.
Email: tinhkh@gmail.com, nguyenvantinh_cntt@haui.edu.vn

Dang Hung Tran received the B.S. from Faculty of Mathematic and Informatics, Hanoi National University of Education in 2001, M.S. in Computer Science from University of Engineering and Technology, VNU in 2006, and Ph.D. degree in Knowledge Science from Japan Advance Institute of Science and Technology (JAIST) in 2009. He is currently an associate professor and dean of Faculty of Information Technology, Hanoi National University of Education (HNUE). His research includes bioinformatics, data mining and machine learning.
Email: hungtd@hnue.edu.vn

A Novel Reversible Data Hiding Based on Improved Pixel Value Ordering Method

Cao Thi Luyen[1], Pham Dinh Phong[1], Pham Hoang Hiep[2]

[1]Faculty of Information Technology, University of Transport and Communications, Hanoi, Vietnam
[2]12A3 Informatics - HUS High School For Gifted Students, VNU Hanoi - University of Science, Vietnam

Communication: received 09 February 2022, revised 03 March 2022, accepted 13 March 2022
DOI: 10.32913/mic-ict-research.v2022.n2.1064

***Abstract*: Reversible data hiding based on the preservation of sorted pixel value ordering (PVO) technique has been researched and expanded recently due to its high embedding capacity and good image quality. secret data is always embedded in the largest or smallest pixel of the sub-block. Thus, each sub-block will embed two bits. The more blocks an image has, the higher the embedding potential. In this paper, to have more sub-blocks, the paper divides the image into sub-blocks with 3 pixels and embeds 2 bits in the maximum value and 1 bit in the minimum value. Therefore, each sub-block of the proposed embedding scheme is able to hide three bits instead of the two bits as in the original scheme. The experimental results also show that the proposed scheme has a significantly higher embedding capacity.**

Keywords: *Reversible data hiding, improved pixel value ordering (IPVO).*

I. INTRODUCTION

The data transmission over the internet environment brings great benefits, besides it also has many potential risks such as: unauthorized copying, content distortion, tampering. Therefore, the issue of protecting the integrity of digital data is one of the urgent issues in the field of information security. The main methods to protect digital content include encryption, hiding, etc. This article researches on hiding information, a technique to hide secret information into digital products where it is difficult for the naked eye to tell the difference between the original product and the product containing the information. The embedded confidential information is then restored as a tool to verify the integrity of the digital product or as a basis for proving the author's copyright [1]. Hiding techniques are divided into two types: the first is the traditional steganography (it is impossible to recover the original), and the second is the reversible steganography (the embedding technique ensures the integrity of the original). Traditional data hiding techniques [2, 3] often cause permanent distortion in the original image, which limits their application in some sensitive areas such as medical image processing and military communications. Therefore, new data hiding methods called reversible (or lossless) data hiding methods are introduced to overcome this limitation. The reversible data hiding (RDH) method can not only extract the embedded data but also restore the original image. This type of reversible hiding has expanded the applicability of information hiding in life, especially in sensitive fields such as healthcare, military, and education. The hiding domain can be in the transform domain or the image space domain, but embedding information in the image space domain is considered to get high embedding capacity but good image quality. The first scheme embedding on the image space domain was proposed by Tian in 2003 with the famous name of difference extension [4]. Tian splits the image into pairs pixels to embed one bit in each pair. This method achieves the embedding capacity of 50% of the image size but distortion image high. Many authors have improved [5–7] to increase embedding and improve image quality. To increase embedding, [5, 7] expands Tian by generating multiple differences and embeddings on each difference value of the pair pixels, while [6] improves the map construction method of [4] to reduce the size of the compressed map so that there is more space to store secret information.

Tian's scheme [8–10] improvement is well known as the extension of the prediction error method to improve the quality of the information image. Instead of expanding the difference between two original pixels, [8–10] embeds information based on the prediction error of two original pixels, thereby greatly reducing image distortion when embedding information. There are also integer transformation methods [11–14], histogram shift [15–17]. Recently, appeared the PVO method [18], which hides information in a reversible manner based on the prediction error so that the pixel order after sorting is preserved. First, [18] divides the original image into none overlapping sub-blocks. Next, each

block will be sorted in ascending order. If the maximum error prediction (the smallest error prediction) is 1 (-1) then a bit is hidden in it so that the arrangement of a sequence of pixels remains unchanged to serve the information recovery process. Scheme [19] improves [18] by using blocks with zero error prediction to improve embedding. However, [19], if the block has many pixels with the largest value and many pixels with the second largest value, it is ignored. The scheme [20] is called the improved PVOK method [19] above. [20] has to transform K pixels to embed one more bit, so the image quality is significantly reduced, but the embedding capacity does not increase much. Schema GePVOK [21] continues to develop a scheme [20] to embed K bits in K pixels with the largest (smallest) value. As a result, embedding capabilities, as well as image quality, are improved [20]. The weakness of scheme [21] is that it uses two bits to store the location map while the map is also an information that needs to be embedded in the image. Map size is inversely proportional to embedding capacity. [22] not only has overcome the above disadvantage but also uses a block with a pixel value of {0,1,254,255}, so that the embedding capacity of this scheme is higher than [21]. [23] scheme proposed to divide the image into the size of three elements to create more blocks, each block is embedded two bits to improve embedding. In addition, [23] proposes a criterion for evaluating flat blocks, only the blocks considered planar should be embedded. As a result, the image quality is greatly better especially in case the embedded data is much less. [24] enlarged the block flatness assessment method and used GePVOK to improve embedding. This work will propose a new inverse algorithm that allows 3 bits to be hided in a 3-pixel subblock: 2 bits will be embedded in the maximum prediction error in a new way, and one bit will be hidden in the prediction error. The smallest prediction error is the same as IPVO. It is easy to get the proposed schema with higher embedding capacity than the related sub-block schemes with the size of 3 pixels. The article also prioritizes embedding in the image block with high flatness (flatness is evaluated as [24]) to get better the quality of images containing information. The next content will introduce IPVO scheme and flat block selection criteria in Section II. Section III proposes a reversible embedding scheme. The comparison of embedding capacity and image quality between the proposed scheme and related methods is presented in Section IV. Section V is the conclusion of the paper.

II. RELATED WORKS

1. IPVO scheme

Reversible data hiding based on pixel ordering PVO is the newest method for high embedding capacity and good quality proposed by Li et al. in 2013 [18]. PVO is improved [19] to use the whole block with zero prediction error for embedding information to improve embedding. The original image is first divided into non-intersecting blocks of n pixels $((x_1, x_2, \ldots, x_n)$. Blocks containing $\{0, 1, 254, 255\}$ will not be used for embedding and are marked in the map as 1 to avoid underflow and overflow. Here is how [19] has done to embed and extract information from a block of images.

Data embedding algorithm on a block

Input: original pixel sequence $(x_1, x_2, \ldots, x_n)$ and secret data b

Output: The sequence of pixels contains secret data $(y_1, y_2, \ldots, y_n)$.

Embedding process:

Step 1: Sort the original pixel sequences $(x_1, x_2, \ldots, x_n)$ in ascending order to get the sequence $(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)})$ $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \ldots \leq x_{\sigma(n)})$, where $\sigma : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ is the reversible mapping.

Step 2: Find the maximum and minimum prediction errors d_{max} d_{min} according to the following formula: $d_{max} = x_{\sigma(n)} - x_{\sigma(n-1)}$ and $d_{min} = x_{\sigma(1)} - x_{\sigma(2)}$

Step 3: Embedding data

Embed at the largest prediction error:

Case 1: If $d_{max} > 1$ then $x'_{\sigma(n)} = x_{\sigma(n)} + 1$

Case 2: If $d_{max} = 1$ and $\sigma(n) < \sigma(n-1)$ then: $x'_{\sigma(n)} = x_{\sigma(n)} + b$ Otherwise $x'_{\sigma(n)} = x_{\sigma(n)} + 1$

Case 3: If $d_{max} = 0$ and $\sigma(n) > \sigma(n-1)$ then: $x'_{\sigma(n)} = x_{\sigma(n)} + b$

Embed at the smallest prediction error:

Case 1: If $d_{min} < -1$ then $x'_{\sigma(1)} = x_{\sigma(1)} - 1$

Case 2: If $d_{min} = -1$ and $\sigma(1) > \sigma(2)$ then $x'_{\sigma(1)} = x_{\sigma(1)} - b$ otherwise $x'_{\sigma(1)} = x_{\sigma(1)} - 1$

Case 3: If $d_{max} = 0$ and $\sigma(1) < \sigma(2)$ then: $x'_{\sigma(1)} = x_{\sigma(1)} - b$

From the reversible mapping it is easy to infer the sequence of pixels contains secret data $(y_1, y_2, \ldots, y_n)$.

Data restoring algorithm on a block

Input: The sequence of pixels contains secret data $(y_1, y_2, \ldots, y_n)$.

Output: original pixel sequence $(x_1, x_2, \ldots, x_n)$ and secret data b

Step 1: Sort the sequences $(y_1, y_2, \ldots y_n)$ in ascending order to get the sequence $(x'_{\sigma(1)}, x'_{\sigma(2)}, \ldots, x_{\sigma(n)})'$ $x'_{\sigma(1)} \leq x'_{\sigma(2)} \leq \ldots \leq x'_{\sigma(n)}$.

The maximum and minimum prediction errors d_{max}, d_{min} are computed according to the following formula:

$d'_{max} = x'_{\sigma(n)} - x'_{\sigma(n-1)}$ and $d'_{min} = x'_{\sigma(1)} - x'_{\sigma(2)}$

Restore at the largest prediction error:

Case 1: $d'_{max} > 2$ In this case, the secret information is

0	0	0	c6
0	0	0	c5
c1	c2	c3	c4

Figure 1. Illustration of the flatness of a block

not embedded and the original pixel is restored according to the formula: $x_{\sigma(n)} = x'_{\sigma(n-1)} - 1$
Case 2: If $d'_{max} = 2$ and $\sigma(n) < \sigma(n-1)$ then: $b = 1, x_{\sigma(n)} = x'_{\sigma(n)} - 1$
Otherwise no secret data is recovered and original pixels are restored as follows: $x_{\sigma(n)} = x'_{\sigma(n)} - 1$
Case 3: $d'_{max} = 1$
If $\sigma(n) < \sigma(n-1)$ then $b = 0, x_{\sigma(n)} = x_(\sigma(n))'$
If $\sigma(n) > \sigma(n-1)$ then: $b = 1, x_{\sigma(n)}) = x'_{\sigma(n)} - 1$
Case 4:If d_{max}'=0 then $b = 0, x_{\sigma(n)} = x'_{\sigma(n)}$
Restore at the smallest prediction error:
Case 1: If d_{min}'<-2. In this case, the secret information is not embedded and the original pixel is restored according to the formula: $x_{\sigma(1)} = x'_{\sigma(1)} + 1$
Case 2: If $d'_{min} = -2$ and $\sigma(2) > \sigma(1)$ then: $b = 1, x_{\sigma(1)} = x'_{\sigma(1)} + 1$ otherwise, there is no embedded data and the original pixel is restored according to the formula $x_{\sigma(1)} = x'_{\sigma(1)} + 1$
Case 3: d'_{min}=-1 If $\sigma(2) > \sigma(1)$ then: $b = 0, x_{\sigma(1)} = x'_{\sigma(1)}$
If $\sigma(2) < \sigma(1)$ then: $b = 1, x_{\sigma(1)} = x'_{\sigma(1)} + 1$
Case 4: If $d'_{min} = 0$ then $b = 0, x_{\sigma(1)} = x'_{\sigma(1)}$.
The proposed scheme will use the embedding-extraction algorithm at the smallest value to embed one bit.

2. Determining the flatness of the block

The higher the flatness of the image blocks, the better the image quality, as well as the concealment of the scheme. [24]evaluates the flatness of the current block (blue region of Figure 1) based on neighboring pixels (yellow region of figure 1) according to formula 1:

$$T(I_i) = \frac{1}{mxn}\sqrt{\sum_{i=1}^{m\cdot n}(c_i - s)} \quad (1)$$

where $s = 1/(m+n+1)\sum_{i=1}^{m+n+1} c_i$
To determine the flat threshold α of the given image and secret information, we need to do iteratively. If $T(I_i) \le \alpha$ then the block Ii will be considered smooth and

III. THE PROPOSED SCHEME

Scheme [19] is considered to be highly efficient, so the proposed scheme applies this algorithm to embed 1 bit at the smallest prediction error. The new point of the paper is to propose a 2-bit embedding algorithm at the largest prediction error. The scheme proposed to divide the subblock by 3 pixels and use the idea of block selection of the scheme [24] to get the better imperceptibility. The embedding and extracting scheme on a block and the whole image are presented below.

1. Embedding algorithm in a block

Given subblock by three pixel $I_i = (x_1, x_2, x_3)$, threshold α and secret data $B = (b_1, b_2, b_3)$. Assume that there are no valuable pixels in the set {0,253,254,255} and $T(I_i) \le \alpha$ (T is calculated by the formula (1)), The process of embedding information on this block will be done as follows:
Step 1: Sort (x_1, x_2, x_3) in ascending order to get the sequence $(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)})$, $x_{\sigma(1)} \le x_{\sigma(2)} \le x_{\sigma(3)}$, where $\sigma : \{1, 2, n\} \to \{1, 2, n\}$ is the reversible mapping.
Step 2: Compute the largest prediction error $d_{max} = x_{\sigma(3)}) - x_{\sigma(2)}$ and the smallest prediction error $d_{min} = x_{\sigma(1)} - x_{\sigma(2)}$
Step 3: Embed the secret data
Embed bit b_1 into the smallest prediction error d_{min} by using the IPVO method (section II). Embed b_2, b_3 into the largest prediction error d_{max} as follows: If $d_{max} > 1$ then data is not embedded and $x'_{\sigma(3)} = x_{\sigma(3)} + 3$. $d'_{max} > 4$.
If $d_{max} = 1$ then data is embeded as follows:
If $b_2 = 0, b_3 = 0$ then$x'_{\sigma(3)} = x_{\sigma(3)}$. d'_{max} will equal to 1.
If $b_2 = 0, b_3 = 1$ then $x'_{\sigma(3)} = x_{\sigma(3)} + 1$. d'_{max}will equal to 2.
If $b_2 = 1, b_3 = 0$ then $x'_{\sigma(3)} = x_{\sigma(3)} + 2$ d'_{max} will equal to 3.
If $b_2 = 1, b_3 = 1$ then $x'_{\sigma(3)} = x_{\sigma(3)} + 3$. d'_{max} will equal to 4.
It is easy to see that $x_{\sigma(2)} = x'_{\sigma(2)}$ and $x'_{\sigma(1)} \le x'_{\sigma(2)} \le x'_{\sigma(3)}$
Step 4: Rearrange $(x'_{\sigma(1)}, x'_{\sigma(2)}, x'_{\sigma(3)})$ to get (y_1, y_2, y_3).
The following example describes the embedding process of the current tile (colored tile). In case of being able to embed in Fig. 2 and in case of not being able to embed in Fig. 3.

No embedded data case:

2. Extracting and restoring algorithm in a block

Given a block of 3 pixels $I_i = (y_1, y_2, y_3)$ and threshold α. Assume the location map of the block is 0 which means this block contains secret information and $T(I'_i) \le \alpha$ then the data and the host image recovery is performed as

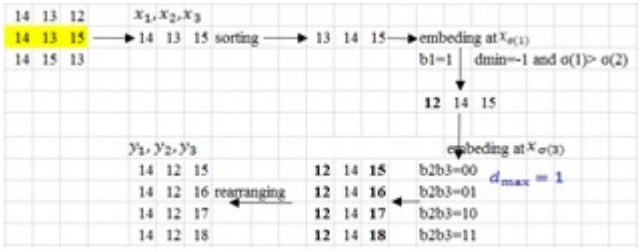


Figure 2. Example of embedded case

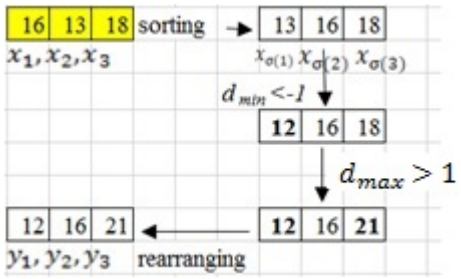


Figure 3. Example of non-embedded case

follows:

Step 1: Sort (y_1, y_2, y_3) to get an ascending sequence$(y_{\sigma(1)}, y_{\sigma(2)}, y_{\sigma(3)}$

Step 2: Compute $d'_{max} = y_{\sigma(3)} - y_{\sigma(2)}$ and $d'_{min} = y - y_{\sigma(2)}$

Step 3: Restore the data Restore bit b_1 and pixel x_1 based on the smallest predictor d'_{min} as the IPVO method presented in Section II.

Based on the largest predictor d'_{max} to recover the remaining b_2, b_3 and 2 pixels x_2, x_3 as follows:

If $d'_{max} > 4$ then no secret information is extracted, $x_{\sigma(3)} = y_{\sigma(3)} - 3$, otherwise, perform data recovery as follows:

If $d'_{max} = 1$ then $b_2 = 0, b_3 = 0$ and $x_{\sigma(3)} = y_{\sigma(3)}$.

If $d'_{max} = 2$ then $b_2 = 0, b_3 = 1$ and $x_{\sigma(3)} = y_{\sigma(3)} - 1$.

If $d'_{max} = 3$ then $b_2 = 1, b_3 = 0$ and $x_{\sigma(3)} = y_{\sigma(3)} - 2$.

If $d'_{max} = 4$ then $b_2 = 1, b_3 = 1$ and $x_{\sigma(3)} = y_{\sigma(3)} - 3$.

Step 4: Rerange $(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)})$ to get the original image (x_1, x_2, x_3).

Below is an illustration of the process of restoring the original image and embedding information. In case there is embedded data Fig. 4 and in case there is no embedded information in Fig. 5. An example of restoring the original image in case of unembedding is as follows

3. Data embedding procedure

Input: The host image I, threshold α the secret data B

Output: The stego image I'

Embedding secret information B into image I is done as follows:

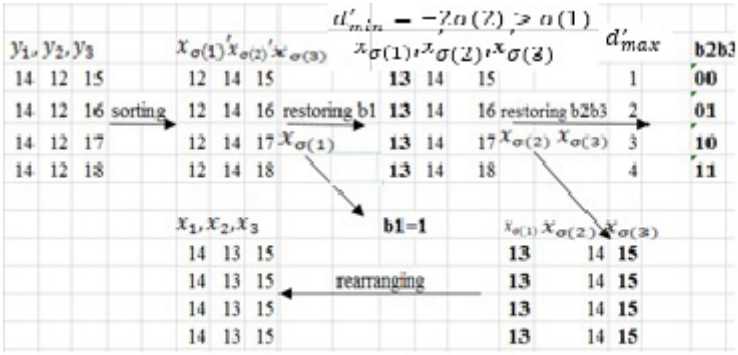


Figure 4. An example of the case of recovering block containing information

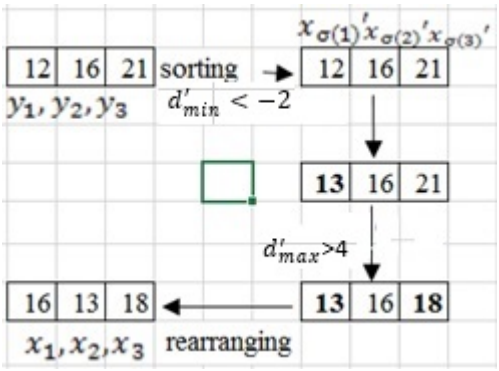


Figure 5. An example of the case of recovering block not containing information

Step 1: Divide the image into nonoverlapping subblocks of length 3 pixels (browsing the image in order from left to right and from top to bottom).

Step 2: The location map is created according to the formula below:

$$LM(I_i) = \begin{cases} 1, & \text{if } \exists x_j \in \{0, 223, 224, 225\}, j = 1, 2, 3 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Step 3: Compress the map in a lossless manner to obtain a compressed map denoted by CLM (assuming the CLM has length L).

Step 4: Specify extra information to be embedded into the image to be used for image recovery purposes as follows:

Step 5: Set the data embedded in the image as: secret data $B + LSB54$

where $LSB54$ is $(54+L)$ low bits of the embedded element data ta for extracting is explained in Step 6.

Step 6: Embed $(54 + L)$ bit B into I by traversing each sub-block in turn, assuming the i-th sub-block is denoted by I_i. If $LM(I_i) = 1$ then skip and browse to the next block, otherwise embedding in this block according to the

TABLE I
PSNR OF THE STEGO IMAGES IN MAXIMAL EMBEDDING CASE

	Information	Size of bit
1	Block size	6 bits
2	Threshold α	8 bits
3	Low bit data embedding start block position (BO)	16 bits
4	The size of compressed location map	24 bits
5	The compressed location map CLM	L

algorithm mentioned in section 3.1 get I_J. Take the low $54+L$ bits of the first pixels of I_J to get $LSB54$. Continue embedding until the end B. Locate the the first block which is embedded $LSB54(BO)$, continue to embed all $LSB54$ according to the algorithm in section3.1.

Step 7: Embeds extra information by low bit insertion into the first $54+L$ pixels. The result is stego image I'.

4. Data extracting procedure

Input: stego image I'

Output: The host image I, threshold α the secret data B

Step 1: Using the low-bit insertion method LSB extracts 6, 8, 16, and 24 bits respectively to get the tile size, threshold α, starting block position BO and compressed map size L. Continue using LSB with Next L bits to get the CLM compressed map. Extract the CLM to get the LM location map.

Step 2: Divide the image into sub-blocks of 3 pixels (1×3 or 3×1) as embedding. From the block location BO, refer to the values of $LM(j)(j > BO)$, perform $(54+L)LSB54$ bit extraction and recover the original tiles according to the algorithm mentioned in section 3.2, in blocks j has $LM(j) = 0$. Insert bits from $LSB54$ into the first $54+L$ pixels by LSB method.

Step 3: Reference $LM(i)(i >= 1)$, extract data bits from the first blocks to the $BO-1$ block corresponding to blocks with $LM(i) = 0$ in the message containing image B and recover the original image according to the algorithm described in section 3.2. By on embedded data B and the original image I fully restored.

IV. EXPERIMENTAL RESULTS

Grayscale images[1] with size 512×512 as Fig. 6 below are used to investigate the embedding ability and image quality of the proposed schema compared to schema has the same method. The test tool is Matlab2016, windows10. Below are the results of the embedding capacity survey of the methods.

Tables II and III above are the results of embedding testing when dividing 3×1 and 1×3 blocks, eliminating

[1]taken from https://sipi.usc.edu/database/

TABLE II
COMPARISON OF EMBEDDING CAPACITY WITH SUBBLOCK SIZE 1×3

Image	PVO	IPVO	PVOK	GePVO	Proposed scheme
1	28040	32681	30640	35536	49215
2	15952	22872	16776	18096	34541
3	30736	40333	35396	47222	60298
4	47155	55311	52468	61375	82427
5	43185	55395	49526	64298	82510
6	29530	34538	32730	37651	52168
7	33449	36821	35931	38350	54948
8	6328	42782	6954	17655	63918
9	29506	31899	31453	33228	47937
10	30757	32691	32462	33652	49114
TB	29464	38532	32434	38706	57708

TABLE III
COMPARISON OF EMBEDDING CAPACITY WITH SUBBLOCK SIZE 3×1

Image	PVO	IPVO	PVOK	GePVO	Proposed scheme
1	30876	36826	33678	39974	55364
2	13586	18879	14185	15198	28480
3	34332	47135	39336	54488	70712
4	47335	59182	53388	66299	88304
5	45557	57199	51898	65663	85053
6	27407	31734	30145	34392	47867
7	39620	45263	42781	47012	67694
8	6121	42219	6659	17710	63005
9	27178	29831	28799	30460	44677
10	31658	33576	33303	34540	50313
TB	30367	40184	33417	40574	60147

the compressed map storage space of the proposed schema and related schemas. The proposed method has a higher embedding capacity than PVO, IPVO, PVOK, and GePVOK methods by 19%, 15%, 18%, and 15%, respectively, If the subblock is divided as 3×1.

Next, we use $PSNR$ to test the imperceptibility of the proposed scheme and several widely PVO based schemes. $PSNR$ is calculated by using the following Equation:

$$PSNR = 10\log_{10}\frac{255^2}{MSE}, \tag{3}$$

where I, I' are the original image and the image containing data, respectively with M rows N columns, where the mean square error (MSE) between the original and embedded image is defined as:

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(I(i,j) - I'(i,j))^2, \tag{4}$$

The results are shown in the figures, from Fig. 7 to Fig. 15. The above figures show that the $PSNR$ value of the GePVO scheme is the lowest, and the second-lowest is PVOK, in most cases then the proposed scheme has the highest PSNR value. Particularly in Figure 9 then the $PSNR$ of the IPVO and the proposed schemes equal approximately. Table 3 below is the result of recording the $PSNR$ value of the methods in the case of maximum embedding.

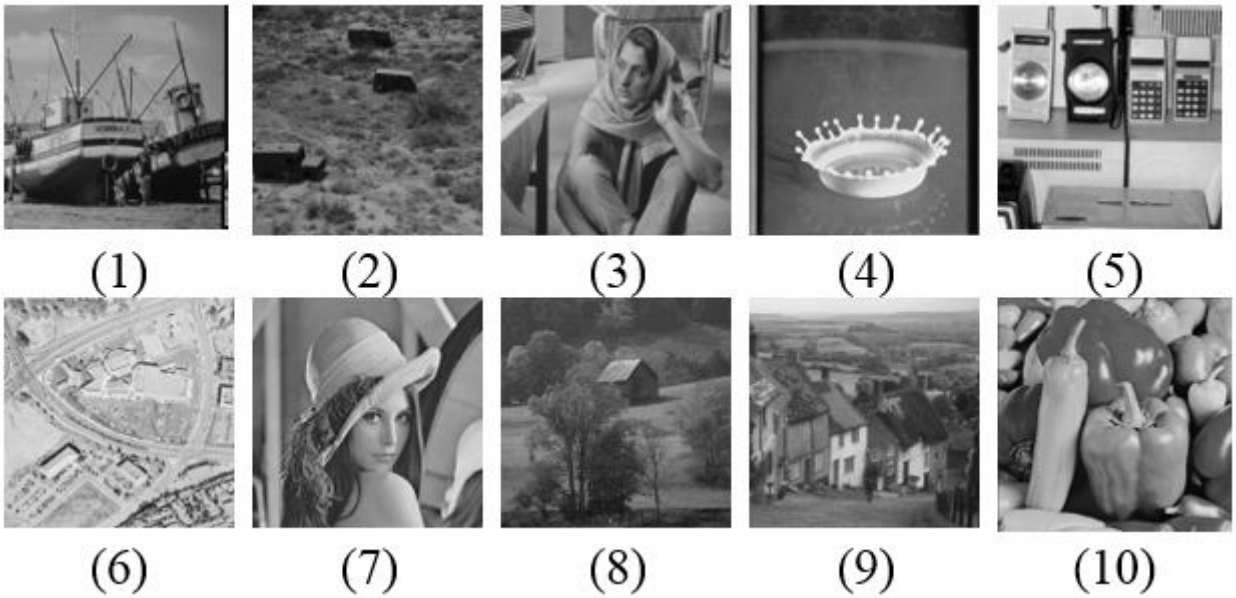


Figure 6. Experimental images

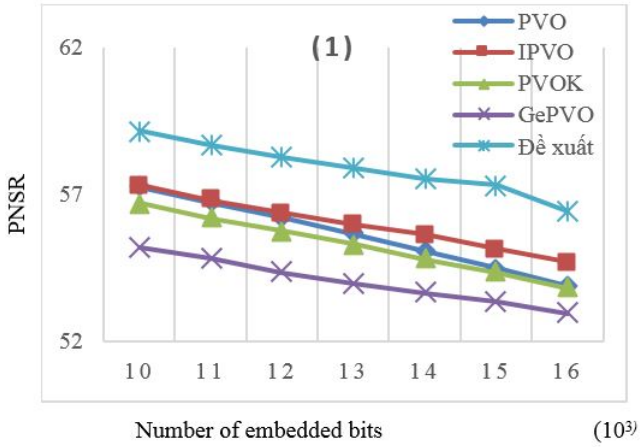


Figure 7. Comparisons in terms of quality image between methods for image(1)

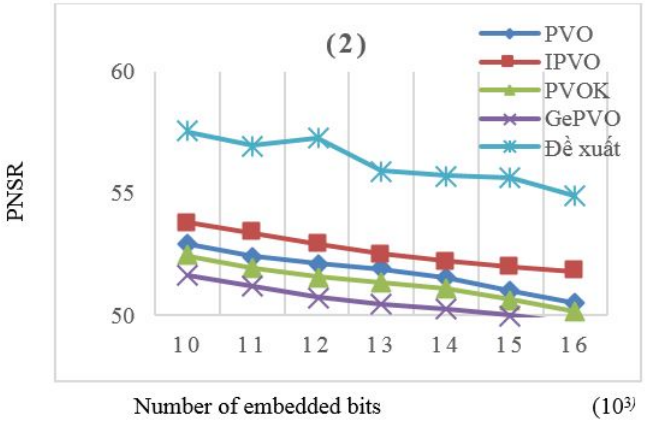


Figure 8. Comparisons in terms of quality image between methods for image(2)

With the number of embedding bits of the proposed scheme higher than that of PVO, PVOK, IPVO, and GePVO schemes at 19%, 15%, 18%, and 15%, respectively, the image quality of the proposed scheme is still high. (*PSNR* of the proposed scheme is above 40). Therefore, it can be concluded that the proposed scheme gets good image quality.

V. CONCLUSION

The article has researched and improved the reversible hiding method based on the improved pixel value ordering method, which is considered as one of the efficient reversible data hiding technique. The proposed technique

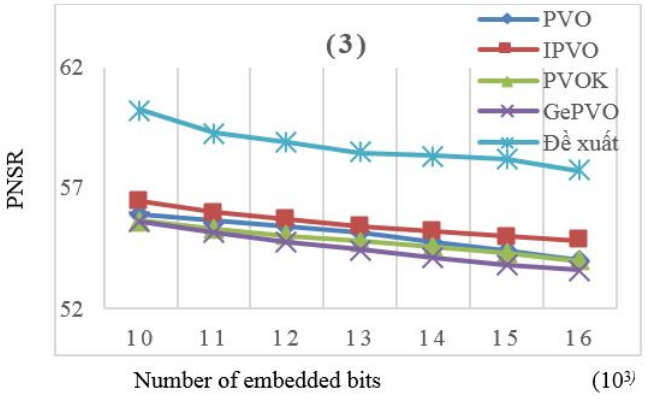


Figure 9. Comparisons in terms of quality image between methods for image(3)

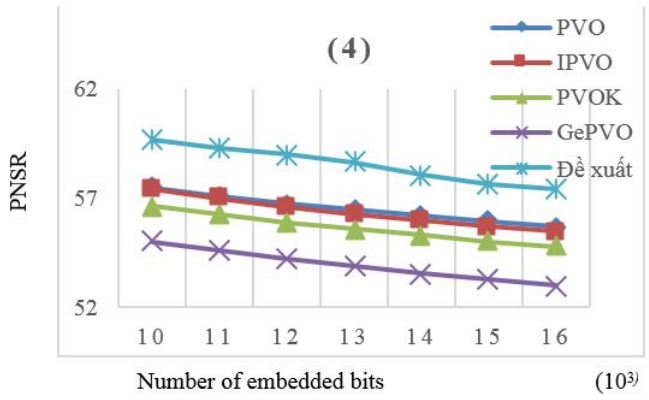


Figure 10. Comparisons in terms of quality image between methods for image(4)

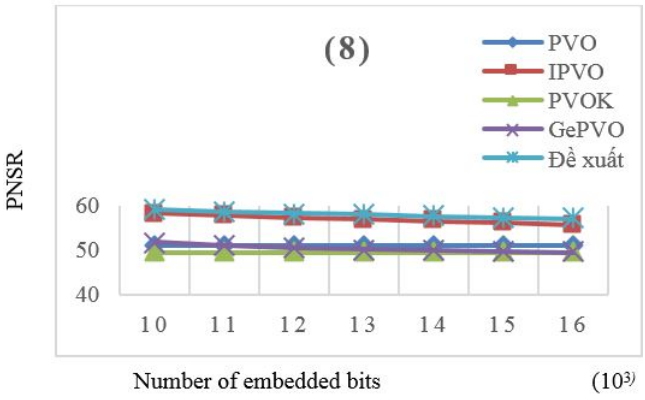


Figure 13. Comparisons in terms of quality image between methods for image(7)

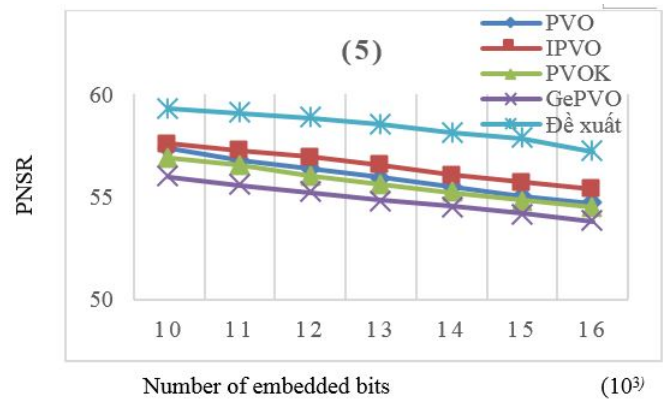


Figure 11. Comparisons in terms of quality image between methods for image(5)

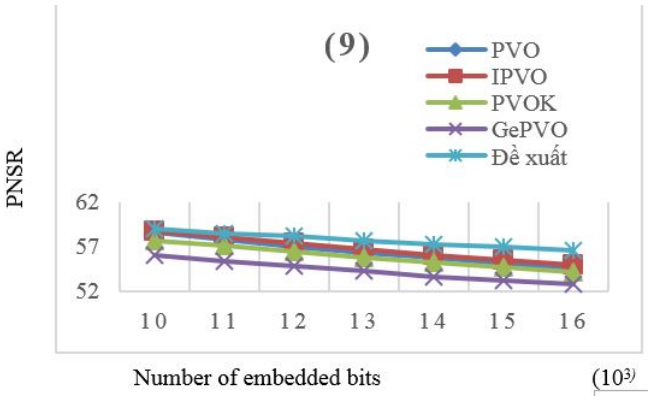


Figure 14. Comparisons in terms of quality image between methods for image(8)

image(6).

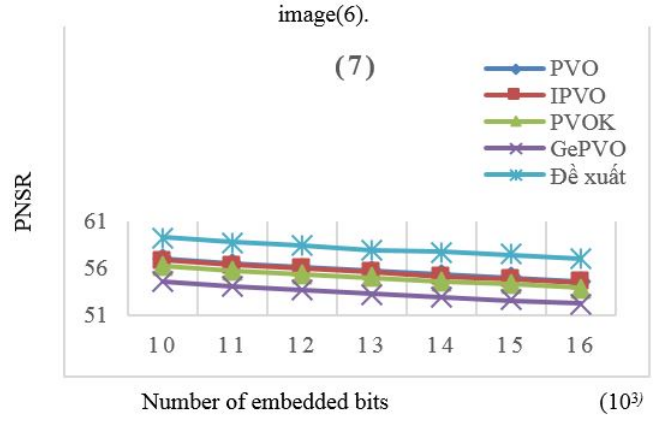


Figure 12. Comparisons in terms of quality image between methods for image(6)

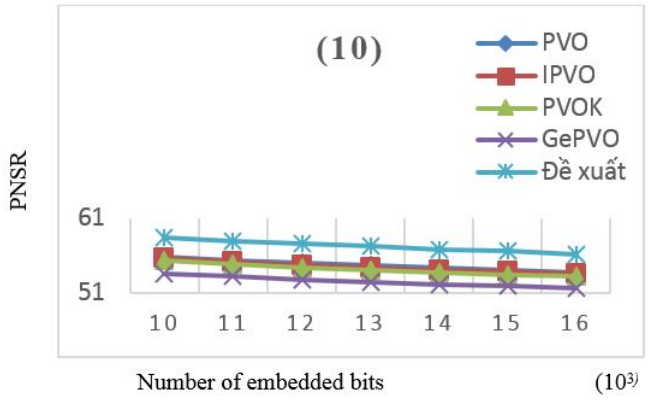


Figure 15. Comparisons in terms of quality image between methods for image(9)

TABLE IV
PSNR OF THE STEGO IMAGES IN MAXIMAL EMBEDDING CASE

Image	PVO	IPVO	PVOK	GePVO	Proposed scheme
1	57.24	57.30	56.69	55.20	51.05
2	56.71	56.79	56.16	54.79	50.55
3	56.22	56.38	55.74	54.33	50.07
4	55.64	55.98	55.29	53.97	49.66
5	55.06	55.61	54.79	53.64	49.37
6	54.47	55.16	54.33	53.33	49.04
7	51.56	52.23	51.11	50.28	45.83
8	52.93	53.82	52.46	51.66	47.28
9	52.42	53.37	51.93	51.19	46.85
10	51.91	52.53	51.36	50.49	46.12
Average	54.41	54.92	53.99	52.89	48.58

of embedding three bits per subblock with three pixels in the article has significantly enhanced embedding ability compared with schemes with the same method. The experiment results also show that the embedding capacity of the proposed method is higher than that of PVO, PVOK, IPVO, GePVO methods by 19%, 15%, 18% and 15%, respectively. The image quality of the proposed scheme is also high. The next goal is to improve the map to increase the space for embedding purposes, thereby continuing to improve the embedding capacity. Besides, we study the direction of reducing image variability to enhance image quality.

REFERENCES

[1] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan kaufmann, 2007.

[2] D. Wu and W. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern recognition letters*, vol. 24, no. 9-10, pp. 1613–1626, 2003.

[3] L. Mielikainen, "Lsb matching revisited," *signal processing lettersl*, vol. 13, no. 5, pp. 285—-287, 2006.

[4] J. Tian, "Reversible data embedding using a difference expansion," *IEEE transactions on circuits and systems for video technology*, vol. 13, no. 8, pp. 890–896, 2003.

[5] A.M.Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE transactions on image processing*, vol. 13, no. 8, pp. 1147–1156, 2004.

[6] Y. Hu, H. Lee, and J. Li, "De-based reversible data hiding with improved overflow location map," *IEEE Trans. Circuits Syst. Video Technol*, vol. 19, no. 2, pp. 250–260, 2009.

[7] C. T. Luyen and P. V. At, "An efficient reversible watermarking method and its application in public key fragile watermarking," *Applied Mathematical Sciences*, vol. 11, no. 30, pp. 1481–1494, 2017.

[8] D. Thodi and J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE transactions on image processing*, vol. 16, no. 3, pp. 721–730, 2007.

[9] W. Hong, T.-S. Chen, and C.-W. Shiu, "Reversible data hiding for high quality images using modification of prediction errors," *Journal of Systems and Software*, vol. 82, no. 11, pp. 1833–1842, 2009.

[10] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *IEEE Transactions on Information Forensics and security*, vol. 6, no. 3, pp. 873–882, 2011.

[11] G. Xuan, C. Yang, Y. Zhen, Y. Shi, and Z. Ni, "Reversible data hiding using integer wavelet transform and companding technique," in *Lecture Notes in Computer Science book series*, vol. 3304. Springer, 2004, pp. 115–124.

[12] F. Peng, X. Li, and B. Yang, "Adaptive reversible data hiding scheme based on integer transform," *Signal Process*, vol. 92, pp. 54–62, 2012.

[13] M. Arsalan, S. Malik, and A. Khan, "Intelligent reversible watermarking in integer wavelet domain for medical images," *J Syst Softw*, vol. 18, no. 4, pp. 883–894, 2012.

[14] F. Li, Q. Mao, and C. C. Chan, "A reversible data hiding scheme based on iwt and the sudoku method," *J Syst Softw*, vol. 18, no. 3, pp. 410–419, 2016.

[15] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[16] X. Wang, X. Li, B. Yang, and Z. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Process. Lett*, vol. 17, no. 6, pp. 567–570, 2010.

[17] H. Wu and J. Huang, "General framework to histogram shifting based reversible data hiding," *Signal Processl*, vol. 22, no. 6, pp. 2181–2191, 2013.

[18] X. Li, J.Li, B. Li, and B. Yang, "High fidelity reversible data hiding scheme based on pixel value ordering and prediction error expansion," *Signal Process*, vol. 93, pp. 198–205, 2013.

[19] F.Peng, X.Li, and B.Yang, "Improved pvo-based reversible data hiding," *Digital Signal Processing*, vol. 25, pp. 255–265, 2014.

[20] X.Qu and H.J.Kim, "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding," *Signal Processing*, vol. 111, pp. 249–260, 2015.

[21] J.Li, Y.H.Wu, C.F.Lee, and C.C.Chang, "Generalized pvo-k embedding technique for reversible data hiding." *Int. J. Netw. Secur.*, vol. 20, no. 1, pp. 65–77, 2018.

[22] L.Q.Hoa, C.T.Luyen, N.K.Sao, and P.V.At, "An improved reversible watermarking based on pixel value ordering and prediction error expansion," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2020, pp. 582–592.

[23] Z. Pan and E. Gao, "Reversible data hiding based on novel embedding structure pvo and adaptive block-merging strategy," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26 047–26 071, 2019.

[24] N.T.Phuc and C.T.Luyen, "A reversible data hiding based on generalized pixel value ordering," *Transport and Communications Science Journal*, vol. 72, no. 2, pp. 193–203, 2021.

Luyen Cao Thi Born on April 28, 1979 in Hung Yen. Graduated from the University in 2001, master in 2005 from the University of Science and Technology and received a doctorate in 2018 from the Military Institute of Science and Technology. Lecturer at Faculty of Information Technology - University of Transport and Commutications Areas of interest: Information hiding, digital watermarking, information security.
Email: luyenct@utc.edu.vn.

Phong Pham Dinh received a Master degree in Information Technology and a Doctor of Philosophy degree in Computer Science from University of Engineering and Technology, Vietnam National University, Hanoi in 2011 and 2018, respectively. Now, he is a lecturer of University of Transport and Communications. His research interests include hedge algebras, fuzzy systems, soft computing, data mining and machine learning.
Email: phongpd@utc.edu.vn

Hiep Pham Hoang graduated the High School for Gifted Students, Hanoi University of Science, Vietnam National University in 2022. His research interests include competitive programming, algorithms and machine learning.
Email: phamhoanghiep03092004@gmail.com

A Reliable High-speed Compact In-memory Matching Circuit for CAM-Application Based on NV-RAM

Quang-Manh Duong, Quang-Kien Trinh, Hai-Duong Nguyen, Van Phuc-Hoang, Hoang-Gia Vu, Dinh-Ha Dao, Van-Toan Tran, Duy-Manh Luong
Le Quy Don Technical University, Hanoi, Vietnam

Correspondence: Quang-Kien Trinh (kien.trinh@lqdtu.edu.vn)
Communication: received 15 July 2022, revised 31 August 2022, accepted 26 September 2022
DOI: 10.32913/mic-ict-research.v2022.n2.1060

***Abstract*: This paper presents an effective approach for implementing content address memory (CAM) based on Non-volatile random-access memory (NV-RAM) technologies. We used the 2T-2R bitcell structure implemented on a 65nm CMOS process with a special in-memory matching circuit for realizing low-delay and energy-efficient lookup operations. The simulation results on Synopsys HSPICE indicate that the proposed CAM design can achieve a search error rate of 0.03-4.61%, search energy per bit of 4.36-6.47 fJ, and an extremely small search latency varying from 0.11-0.12 ns depending on the specific design configurations.**

Keywords: *In-memory computing, NV-RAM, NV-CAM, CAM matching circuit, high-speed.*

I. INTRODUCTION

Recently, Content Addressable Memory types (CAM/ TCAM, later collectively referred to as CAM) [1] have been widely used in applications that require the capability of high-speed access as well as real-time intense processing. Instead of receiving address and returning data as in traditional memories, CAM receives input patterns and returns the corresponding address of the data being stored, providing high-frequency lookup operations with the cost of integrated density and standby power consumption. The comparison with an input pattern is performed in parallel in a single clock cycle across all the rows of storage data, which are arranged by a lookup-table structure. In general, CAM is a type of associated memory, due to the outstanding characteristic of lookup speed, commonly used in massive pattern search applications such as network routers, searching engines or virus checkers, and so on.

The basic structure of a CAM is depicted in Fig. 1(a) and usually consists of two main components: a matrix of CAM cells that stores data patterns, where the number of columns equals the width of the search pattern, and a Priority

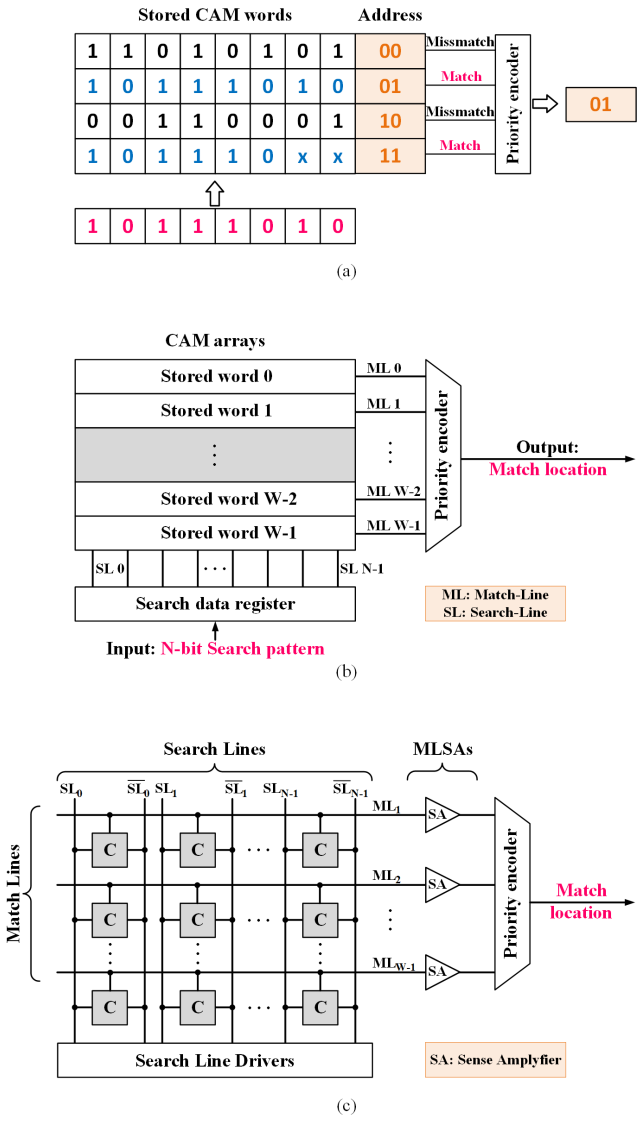


Figure 1. The basic structure and operations of a CAM.

Encoder (PE), which is used to select the address of the row with the highest priority among the similar rows that match the input pattern word. The CAM implementation diagram is shown in Fig. 1(b), where the stored words are arranged in rows, and each of those (a.k.a arrays) is connected to a single Match Line (ML), indicating that the search pattern and stored one at this location are identical (matched case) or different (mismatched case). The MLs are fed into the PE, which is normally responsible for selecting the most suitable position (with the highest priority) among them where the matched cases are found, normally the CAM row with the lower address will have a higher priority. Thus, simply, the CAM circuit receives at the input an N-bit search pattern and returns at the output the matching position of this pattern with a correspondingly selected stored word, either from a large input data space obtain a smaller data space at the output indicating the matching case in the CAM database. Finally, a practical CAM-application circuit model is depicted in Fig. 1(c) based on the diagram in Fig. 1(b) with additional peripheral circuits, including a sense amplifier at the output of the MLs (Match Line Sense Amplifiers - MLSAs) and a circuit that controls the operation of the search lines (Search Line Drivers). Each bit of the pattern word is stored in a CAM cell, which of the same column is connected to a pair of complementary Search Lines $(SL, \overline{SL})$. The lookup process in the CAM database starts from loading the search pattern into Search Line Drivers, followed by the high-level pre-charging for all the MLs, setting the matched state for all of them as well as the whole CAM arrays. In the next step, Search Line Drivers distribute search patterns through all the SL pairs, each CAM cell compares its stored bit with the value on the correspondingly connected SL pair. The comparison results of all CAM cells on the same row determine the final logical state of the MLs, which is then fed into the MLSAs. MLs whose all cells are matched retain high-prechanged, otherwise, they will be discharged to the ground if at least one cell is mismatched. The matching state of the MLs is determined by the sense amplifiers, they provide input signals to the PE to decide where the highest-priority matched location for the input search pattern.

CAM design challenges

The main design parameters that affect the performance of CAMs include the following: Energy, Area, Speed (latency), and Reliability, which are also challenges [2] when designing specific CAM applications. These parameters have different importance depending on the specific application, so it is necessary to determine which challenge as well as design parameter has the highest priority to optimize in an actual scenery. The energy consumed for the CAM operations is primarily composed of energy dissipated in ML, SL, and PE, as well as dissipated energy caused by leakage current, access transistors, decoders, search line drivers, and ML precharge circuits. Saving energy in CAM circuit design is a primarily critical research problem. Several innovations in CAM design improvement have recently been proposed such as a promising memory technology called Priority-Decision in Memory (PDM) [3] that eliminates the need for PEs to identify matched patterns, thereby significantly saving the search energy consumption; the precharge-free CAM schemes [4], which overcome the drawbacks of precharge-based designs and also improve performance during the search by reducing half of the ML evaluation time. The area (footprint) of the CAM array mainly depends on the area of the storage bit-cell. CAM based on a dynamic storage mechanism similar to the DRAM has been proposed to reduce the footprint but at the cost of refresh power and expensive process technology. Thus, several emerging memory technologies (discussed below in this section) have been proposed to address challenges for CAM both in terms of energy and area. The speed of the CAM is inversely proportional to the delay, which consists of two main components: the match delay and the delay of the PE. The match delay depends on several factors such as bitcell topology (NAND or NOR), ML capacitance, array (word) size, and the search pattern, which can be improved by a few techniques such as lowering the loading capacitance and ML voltage [5]; pipelined ML [6]. Meanwhile, multiple match detection (MMD) circuits and priority encoders (PEs) have been proposed and employed in TCAM chips [7], which show a great improvement not only in speed but also in the energy consumption of PE. Finally, the reliability of CAM is mainly governed by the characteristics of the memory employed for data storage, where RRAM or PCRAM typically suffer from poor endurance which limits the reliability of the CAM cells. In general, the reliability of CAM using CMOS is limited by the aging mechanisms of transistors, specifically, negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), and time-dependent dielectric breakdown (TDDB), impacts of NBTI and PBTI effects on ternary CAM and solutions to reduce their aging effects are proposed in [8], [9].

Emerging CAM technologies

In previous decades, materials for CAM fabrication are mainly based on SRAM technology [10] (herein called S-CAM) with a cross-coupled inverter structure which offers fast read/write speeds and relatively high reliability but is associated with some major disadvantages such as large-power dissipation for both storage and search operations, non-zero standby energy consumption, low integration density in addition to the high cost. More specifically,

besides the large cell area caused by a large number of transistors (most of which employ 16 transistors per cell), S-CAM also suffers from high standby power induced by the increasing leakage current, especially when the process of technology shrinks below 45 nm, which has become a major barrier for the compact battery-operated devices. To solve these problems, the researchers proposed using emerging non-volatile memories (NV-CAM) as an alternative to S-CAM, although there are many challenges involved in realizing NV-CAM in mass production and some limitations compared to the previous technology (S-CAM) in terms of search speed and reliability. Emerging non-volatile technologies [11] such as Resistive RAM (RRAM), Magnetoresistive RAM (MRAM), Ferroelectric RAM (FeRAM), and Phase-change RAM (PCRAM) are capable of providing excellent solutions for bitcell area and energy efficiency. On the one side, NV-CAM reduces the number of transistors in a bitcell by replacing them with alternative storage elements, on the other side, being nonvolatile energy, the power consumption of NV-CAM is reduced by effectively blocking its leakage currents during the idle mode, making zero standby power. However, NV-CAM cells suffer from low reliability during search operations because the ML voltage difference (Delta) between the matched and mismatched cases is much smaller than that of the S-CAM cell. There exists a constraint when designing NV-CAM, in which to improve the reliability of NV-CAM it is necessary to increase the value of Delta, which depends on resistance ratio and process variation. To reduce process variation, the size of transistors should be increased, leading to cell area and power overheads. A recent study [12] revealed that PCRAM is the most energy-consuming memory with a medium bitcell area, whereas RRAM, which has inherent limitations in endurance compared to the virtually infinite endurance of other emerging non-volatile memories, has a wide spectrum of cell area values, and generally consumes less energy than PCRAM. Magnetic RAM, typically Spin-Torque Transfer MRAM (STT-MRAM), with low energy consumption, relatively small bitcell area also as high integration density, excellent CMOS compatibility, and extreme long-endurance ($\sim 10^{16}$), represents a promising candidate for CAM in the future compared to the other options [13], [14].

Notable CAM designs related to this work

The design of an NV-CAM circuit to perform data search and comparison operations with dependable reliability, high speed, low energy consumption, and small area occupancy is becoming increasingly essential for In-Memory Computing (IMC) applications. A variety of NV-CAM cell structures has been proposed to overcome the above-mentioned limitations of S-CAM including 2T-2R NV-CAM cell [15], 2.5T-1R NV-CAM cell [16], 3T-2R NV-CAM cell [17], 4T-2R NV-CAM cell [18]. These NV-CAM cell designs, so-called first class, have in common the use of a small number of transistors, ranging from 2T to 4T, which gives the advantage of cell area. In addition, most of them achieve very good results in terms of average energy consumption for search operations, but no metrics regarding their reliability (i.e. Search Error Rate - SER) have been revealed. The second class of CAM-cells that are considered in this paper has rather large bitcell structures, including 9T-2R NV-CAM cell [19], 10T-4R NV-CAM cell [20], and 15T-4R NV-CAM cell [21]. Despite their large size, compared to the cells in first class, these NV-CAM cells have surprising results in the speed of look-up operations, which is demonstrated through the sensing delay. The third class consists only of a 16T-CAM cell based on traditional SRAM memory [22], which is the best S-CAM design published with outstanding reliability (absolute zero SER) and competitive figures on search energy also as sensing delay.

From all of the presented CAM-cell classes, some impressive designs have achieved excellent results only on a single design parameter, but almost none of the proposals can balance all the design criteria, which is needed for applicability. For example, design [15] is the best in the cell area but also has the worst sensing delay, while design [21] and design [22], possessing excellent results on search energy and sensing delay respectively, are the worst designs in the cell area. The restrictions on some design parameters in the prior works [15]-[22] are the fundamental motivation for us to offer a balanced solution for the NV-CAM structure that can meet most of the aforementioned design criteria at an acceptable cost. In this paper, we propose a compact 2T-2R NV-CAM cell and a reliable in-memory matching circuit that employs the proposed cell for CAM applications which ensures very small SERs, the lowest search latencies with relatively low cost of search energies.

The main contributions of our work can be summarized as the followings:

- An in-memory matching circuit based on the 2T-2R NV-CAM cell structure which performs low-latency massive parallel lookup operations.
- A variety of evaluations on the reliability of the proposed design based on the SER value and sensing margin of the ML voltage against the effects of process variation using Monte Carlo simulations.
- An in-depth analysis of all design criteria and comparisons with other proposed CAM cells that employed traditional SRAM also as various emerging memory technologies including RRAM, PCRAM, and MRAM.

The remainder of the paper is organized as follows:

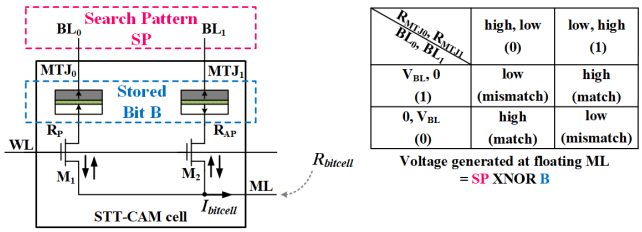


Figure 2. The structure of the proposed 2T-2R NV-CAM cell.

Section II proposes the 2T-2R bitcell configuration and the NV-CAM array architecture for the in-memory matching circuit. The impacts of process variations on the proposed design are analyzed by Monte Carlo simulation and presented in Section III. Design evaluation and comparison with several prior works are considered in Section IV. Section V concludes this work.

II. THE PROPOSED DESIGN FOR THE NV-CAM MATCHING CIRCUIT

A. 2T-2R NV-RAM Bitcell For In-Memory Matching circuit

Our proposed configuration in the bitcell level for STT-MRAM-based CAM is illustrated in Fig. 2, which can be extended to other NV-RAM devices (RRAM, PCRAM). Similar to the structure in [23], [24], a complementary bitcell structure is adopted here, which represents one stored bit in the CAM word, the complement BL to feed the search pattern bit, and the match line ML is left floating and will be used as the sensing node.

The proposed bitcell can perform a single XNOR function as described in the following. The two magnetic tunnel junctions (MTJ_0, MTJ_1) of STT-MRAM are programmed to be low (R_P) and high (R_{AP}) or revise, which is respected to '1' and '0' values and will be used as the stored bit B in the CAM word. Similarly, BL_0 and BL_1 are complementarily assigned to be (V_{BL}, 0) or (0, V_{BL}), respectively representing '1' and '0' of the input search pattern.

The gate of the access transistors M_1 and M_2 are connected to the word line WL, and the source is connected to the match line ML, two transistors are identical. It is obvious that this structure (1, 1) and (0, 0) combinations result in the same circuit, so as for (1, 0) and (0, 1). Therefore this circuit can perform the XNOR function, which is adopted as the matching operation (comparing the stored words with search pattern input) in the CAM applications.

The match line voltage V_{ML} generated by a single bitcell being activated within a matchline can be represented through the Norton equivalent circuit theorem [25].

For a single bitcell under a '1' ('0') input search pattern SP, $V_{BL,0} = V_{BL}$ and $V_{BL,1} = 0$ ($V_{BL,0} = 0$ and $V_{BL,1} = V_{BL}$) and the bitcell short-circuit current $I_{bitcell}$ is expressed as:

$$I_{bitcell}(SP=1) = \begin{cases} \frac{V_{BL}}{R_P+R_{access,0}} & if B = 1 \\ \frac{V_{BL}}{R_{AP}+R_{access,0}} & if B = 0 \end{cases} \quad (1)$$

$$I_{bitcell}(SP=0) = \begin{cases} \frac{V_{BL}}{R_{AP}+R_{access,1}} & if B = 1 \\ \frac{V_{BL}}{R_P+R_{access,1}} & if B = 0 \end{cases} \quad (2)$$

The overall resistance $R_{bitcell}$ seen from the ML terminal of the bitcell equals to $(R_{MTJ,0} + R_{access,0})||(R_{MTJ,1} + R_{access,1})$. By considering that $R_{access,0} = R_{access,1} = R_{access}$, from Eq. 1-2 the resulting voltage V_{ML} for an isolated active bitcell at position *(i, j)* within the memory array is:

$$V_{ML} = R_{bitcell} \cdot I_{bitcell,ij} = \frac{V_{BL}}{X_{ij}} \quad (3)$$

where X_{ij} is defined as:

$$X_{ij} = \begin{cases} \frac{R_P+R_{access}}{R_{bitcell}} & if \quad \overline{B_{ij} \oplus SP_J} = 1 \\ \frac{R_{AP}+R_{access}}{R_{bitcell}} & if \quad \overline{B_{ij} \oplus SP_J} = 0 \end{cases} \quad (4)$$

being B_{ij} the stored bit in CAM word, and SP_j is the input fed to column j. From Eq. 3-4 the match line voltage is proportional to the bit line voltage and directly depends on the XNOR of the search bit and the bit stored in the bitcell.

The resulting sensing margin $SM_{bitcell}$ defined as the ML voltage difference between the output XNOR is '1' and '0', which can be simplified as:

$$SM_{bitcell} = V_{BL} \cdot R_{bitcell} \left(\frac{1}{R_P + R_{access}} - \frac{1}{R_{AP} + R_{access}} \right) = V_{BL} \cdot TMR \cdot \frac{R_P}{R_P + R_{AP} + 2R_{access}} \quad (5)$$

where TMR is the MTJ technology-dependent tunneling magnetoresistance ratio $(R_{AP}-R_P)/R_P$ [26]. For extending to other NV-RAM devices (RRAM, PCRAM), we define the metric of resistance ratio $R_{ratio} = R_{AP}/R_P$. From Eq. 5, the sensing margin can be improved by adopting NV-RAM devices with a higher R_{ratio} or using the stronger access transistors, at the cost of a larger area per bitcell. The effects of the access transistors and resistance ratio (R_{ratio}) on the sensing margin, and hence on the reliability of our design will be discussed in the next section.

B. NV-CAM Array Architecture And In-Memory Matching Circuit

This section presents our proposed NV-CAM array architecture. The $M \cdot N$ array in Fig. 3 consists of M rows and N columns, with wordlines being used to enable computation in the respective rows, and the match lines as outputs. The search line drivers encode search patterns (SP) that are compared to stored bits in the selected word line by the XNOR functions, which are presented in detail in Fig. 2 and Section II. A. It could be proven that the XNOR operation in Eq. 1-3 could be generalized for the row level simply by merging all match lines of all bitcells in a row.

As all rows operate independently of each other, let us consider a single row as depicted in blue in Fig. 3. Let the number of XNOR outputs equal to '1' ('0') across the row be $N_1 (N_0 = N - N_1)$. Compared to the single bitcell resistance $R_{bitcell}$ in Section II. A, the resistance seen from the ML terminal in a row is reduced to $R_{bitcell}/N$. From Eq. 2-3, the select line voltage $V_{(ML,i)}$ in the considered row i-th is:

$$V_{ML,i} = \frac{R_{bitcell}}{N} \sum_{j=0}^{N-1} I_{bitcell,ij} = V_{BL} \frac{R_{bitcell}}{N} \sum_{j=0}^{N-1} \frac{1}{X_{ij}}$$
$$= V_{BL} \left(\frac{N_0}{N} \cdot \frac{R_{bitcell}}{R_{AP} + R_{access}} + \frac{N_1}{N} \cdot \frac{R_{bitcell}}{R_P + R_{access}} \right) \quad (6)$$

where X_{ij} is the resistance determining the bitcell current of the bitcell (i, j) in Eq. 1-3.

The matching output is determined by assigning the output OUT_i of the select line in the generic i-th row to '1' if $V_{(ML,i)} \geq V_{REF}$, and to '0' if $V_{(ML,i)} < V_{REF}$. (V_{REF} is the reference voltage between the V_{ML} in the FullMatch case and 1-bit MisMatch case). Hence, the logical output of the sense amp in the i-th row is equal to the logical conjunction of the XNOR computations across its bitcells:

$$OUT_i = \bigwedge_{j=0}^{N-1} \overline{B_{ij} \bigoplus SP_J} \quad (7)$$

In this work, a conventional current latch sense amplifier (CLSA) as in [14] is employed to generate the matching outputs, its schematic and timing diagram with Monte Carlo simulation are depicted in Fig. 4.

From Eq. 6, the row select voltage linearly depends on N_1, N and ranges from $V_{BL} \cdot \frac{R_{bitcell}}{(R_{AP}+R_{access})}$) to $V_{BL} \cdot \frac{R_{bitcell}}{(R_P+R_{access})}$, therefore, the sensing margin depends not only on the different widths of the access transistors and resistance ratio ($R_{ratio} = R_{AP}/R_P$) as aforementioned in sub-Section II.A but also the length of the stored word in CAM as illustrated in Fig. 5.

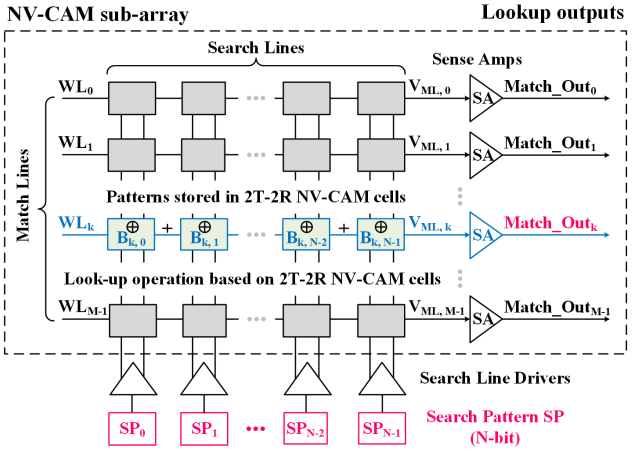


Figure 3. The proposed NV-CAM structure and operations.

As shown in Fig. 5(a), at the small CAM word length (N = 4 or 8 bits), an increase in access transistor size leads to a clear wider sensing margin. However, with a larger stored word length (i.e., N = 16, 32, or 64 bits), the sensing margin is almost independent of the access transistor.

Fig. 5(b) indicates the dependence of the sensing margin on different resistance ratios (R_{ratio}) when the width of the access transistor is 4X the minimum allowed by the technology (i.e., Width = 4W = 0.54 µm, Length = 0.06 µm). R_{ratio} is selected to evaluate in a relatively wide range of values from 2 to 100, where a small value is typical for STT-MRAM (R_{ratio} = 2), an average value commonly encountered in PCRAM (R_{ratio} = 10), and a large one correspondingly represents RRAM (R_{ratio} = 100). Similar above considerations, at a high level of the word length (N), R_{ratio} has a minor effect on the sensing margin. For example, when N = 64 bits, the SM value is 2.7 mV (17.4) when R_{ratio} = 2 (R_{ratio} = 100), meanwhile when N = 4 bits, the SM value is 37.3 mV (116.9) when R_{ratio} = 2 (R_{ratio} = 100).

Our simulation was performed by using the Hspice model with a commercial 65 nm CMOS library. Besides, the analysis of process variations was performed through the Monte Carlo simulations using statistical models for both CMOS and resistance values (the standard deviation of the resistances is 7%). For the CMOS transistors, local and global variations were accounted for by using the statistical models provided by the foundry. The impacts of process variations on our design will be discussed in the next section.

III. IMPACTS OF PROCESS VARIATIONS

Process variations degrade the sensing margin, therefore, errors in the logical value at the output of each row for the

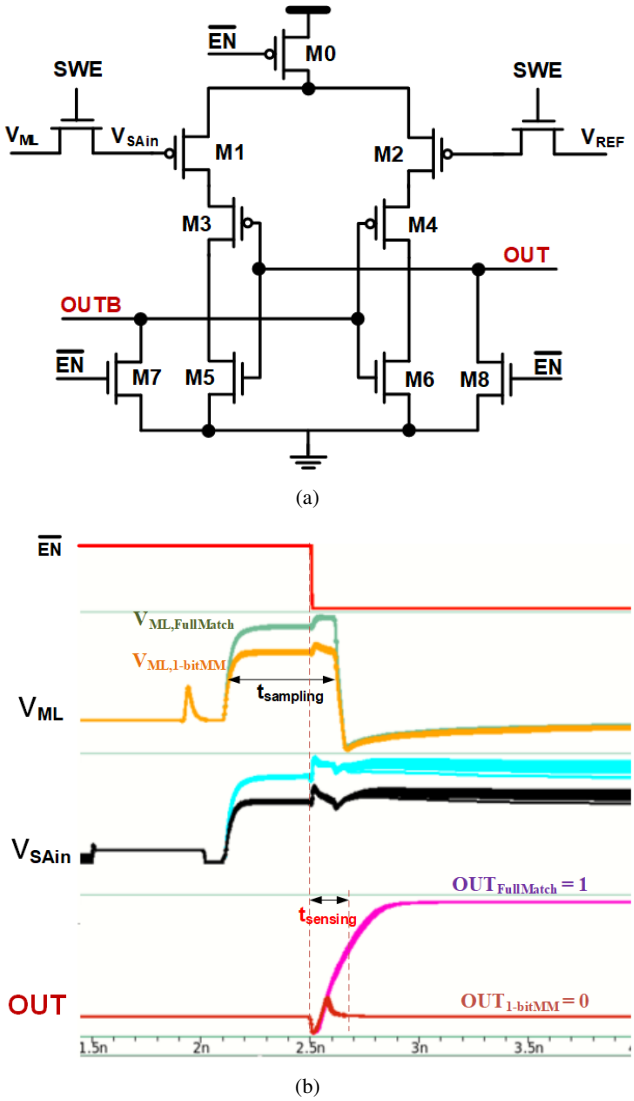


Figure 4. The schematic of CLSA (a), and corresponding timing diagram with Monte Carlo simulation (b).

matching operation are induced. To quantify the impact of variations, Monte Carlo simulations were run under global and local variations in both transistors and resistances. Typical arrays with M = 64 rows and N = 4, 8, 16, 32, and 64 columns will be investigated below. All building blocks from decoding to read-out are included to make the array fully functional and self-consistent.

To quantify the impact of process variations on robustness, we introduce, the SER is the probability of an erroneous output caused by variations in an elementary matching operation [27]:

$$SER = Pr[SM < 0] = \frac{1}{2}\left[1 + erf\left(-\frac{1}{\sqrt{2}} \cdot \frac{1}{\sigma_{SM}/\mu_{SM}}\right)\right] \quad (8)$$

In this work, we investigate SER as the worst-case

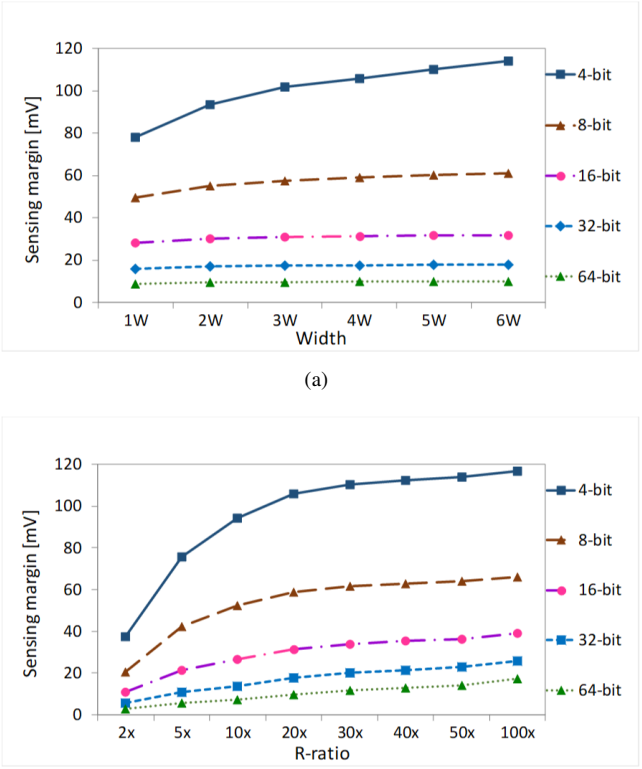


Figure 5. Sensing margin at different word lengths versus width of the access transistor with fixed R_{ratio} = 20 (a), and R_{ratio} with fixed Width = 4W (b).

search error rate (smallest sensing margin) (i.e., when distinguishing fully matched and 1-bit mismatched output). Therefore, $SM = V_{ML_{FM}} - V_{ML_{1MM}}$ is the sensing margin or the difference between ML voltages in cases of FullMatch and 1-bit MisMatch, μ_{SM} and σ_{SM} denote the mean and the standard variation of the sensing margin SM:

$$\sigma_{SM} = \sqrt{\sigma_{ML}^2 + \sigma_{REF}^2 + \sigma_{offset}^2} \quad (9)$$

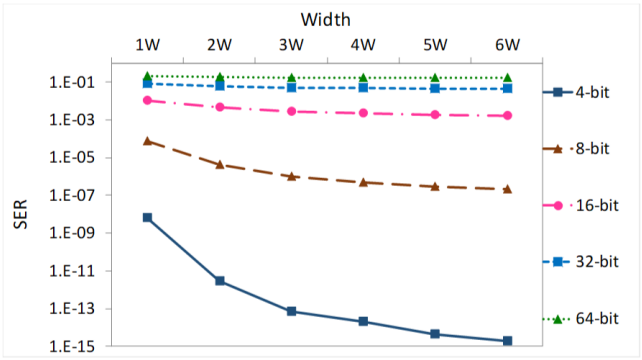
$$\mu_{SM} = |\mu_{ML} - \mu_{REF}| \quad (10)$$

where μ_{ML}, μ_{REF} (σ_{ML}, σ_{REF}) are respectively the mean (standard variation) of the match line voltage and the reference voltage. As usual, the offset of the sense amplifier has zero mean and standard deviation σ_{offset} [28].

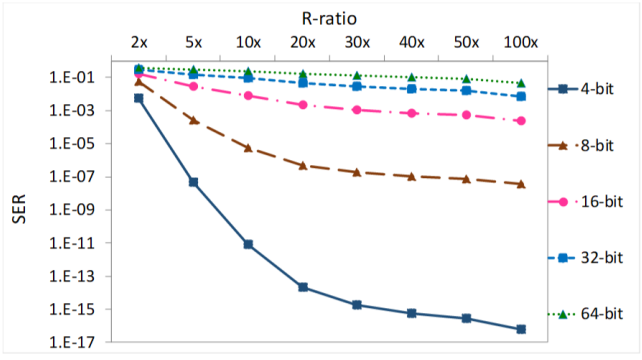
In this work, we consider σ_{offset} is 5mV for a reasonable area of circuit design (sense amplifier size of 16x transistor width and 2x length) [14].

The dependence of the SER on the width of the access transistor, the word length, and R_{ratio} are demonstrated in Fig. 6. Similar to the dependence of the sensing margin on those parameters (N, width of access transistor, R_{ratio}) shown in Fig. 5, the clear differences between the SER

value at low-level of the word-length (N = 4, or 8 bits) versus the SER value at higher N (i.e., N = 16, 32, or 64 bits) is once again depicted. At 4-bit word length, SER can reach extremely small values, specifically, it can be lowered to 6.04e-17 in the case of an R_{ratio} equal to 100, corresponding to RRAM technology. In contrast, SER values become quite large and pose some concerns when increasing the word length to 64 bits, especially, when using STT-RAM technology (R_{ratio} = 2), SER is 3.98 e-1 (i.e 39.8%). As shown in Fig. 6, SER values decrease with increasing the size of the access transistor (Fig. 6(a)) or increasing R_{ratio} (Fig. 6(b)), but the decrease rate in Fig. 6(b) is slightly faster, which presents that the width of the transistor has little effect on SER. For example, with the configuration (N = 64 bits, R_{ratio} = 20), SER only changes from 2.12e-01 (Width = 1W) to 1.66e-01 (Width = 6W). The effects of temperature on the SER are also conducted, which are revealed in Fig. 7. Ignoring the effect of word length in SER, it is clear that ambient temperature causes negligible changes of SER, which decrease from 1.7e-1 to 1.53e-1 when the temperature increases from 25 degrees to 85 degrees respectively.



(a)



(b)

Figure 6. SER at different word lengths vs. width of the access transistor with fixed R_{ratio} = 20 (a), and R_{ratio} with fixed Width = 4W (b).

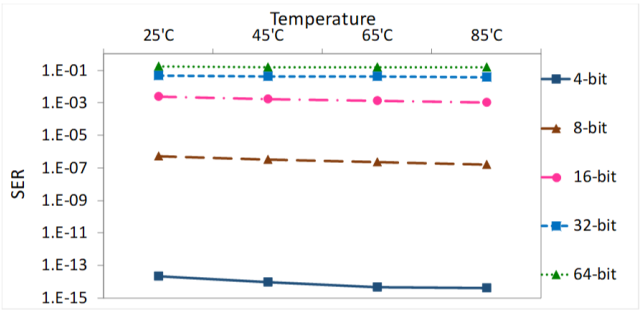


Figure 7. SER of the proposed design with different temperatures with fixed configuration (R_{ratio} = 20, Width = 4W) (b).

IV. PERFORMANCE EVALUATION AND DISCUSSIONS

We conducted the performance evaluations of the proposed NV-CAM design on energy consumption and pattern search latency, which are shown in Fig. 8 and Fig. 9 correspondingly. For the energy consumption measurement, we consider a wide range of match cases including FullMatch (FM), 1-bit MisMatch (1MM), 2-bit MisMatch (2MM), etc., up to the case of Full MisMatch (FMM). At 64-bit word length, the energy consumption of NV-CAM per bit per one access for searching based on the low-R_{ratio} device (R_{ratio} = 2) as STT-MRAM is the highest, gradually decreasing in the case of higher R_{ratio} technologies (PCRAM, RRAM). When R_{ratio} is 100, energy per bit on one search (Esearch/bit) varies from 4.36 fJ to 6.47 fJ depending on the match case.

For the measurement of search latency, we only consider typical word lengths of 16-bit, 32-bit, and 64-bit so that the comparison of search latencies (actually sensing time as illustrated in Fig. 4(b)) with other proposed designs ensures objectivity. In contrast to the decrease in power consumption, the search latency steadily increases for STT-MRAM, PCRAM, and RRAM respectively. The smallest search latencies could be achieved with NV-CAM based on STT-MRAM in the range from 0.11 ns to 0.12 ns depending on the considered word lengths.

The evaluation of the reliability and optimization of a CAM-cell design must be based on a combination of design parameters mentioned in Section I. In addition, the number of proposed designs for comparison should be large enough to ensure objectivity. For those reasons, a series of CAM designs [15]-[22] has been chosen for evaluation. The comparisons in detail over all the design aspects including Cell area, SER, Energy per bit on one search (Esearch/bit), and Sensing latency (Tsensing) are shown in Table I.

Cell area: The smallest cell area belongs to the 2T-2R cell proposed in [15], which was fabricated in the 90 nm technology process. Most of the compact CAM-

TABLE I
COMPARISON WITH PRIOR WORKS

	ISCAS'18 [22]	**TCASI'18** [21]	**TCASII'16** [20]	**JJAP'12** [19]	**JSSC'16** [18]	**E-Let'17** [17]	**ISSCC'16** [16]	**VLSI'13** [15]	**This work**
Memory type	CMOS NOR-type	MRAM	MRAM	MRAM	RRAM	RRAM/MRAM	RRAM	PCRAM	**RRAM/PCRAM/MRAM**
Technology process [nm]	65 nm	**40 nm**	45 nm	90 nm	180 nm	180 nm	65 nm	90 nm	65 nm
Cell structure	16T-SRAM	15T-4R	10T-4R	9T-2R	4T-2R	3T-2R	2.5T-1R	**2T-2R**	**2T-2R**
Cell area [μm^2]	11.43	10.76	2.78	6.84	9.7	8.6	0.59	**0.41**	0.51
Cell area [F^2]	3175	6725	1372.8	844.4	1197.5	265.4	163.8	**50.6**	142.4
Search Error Rate (SER)	**0%**	2.7%	18.5%	30%	N/A	N/A	N/A	N/A	0.03 - 4.61%[1]
Esearch/bit [fJ]	2.07	**0.17**	40.5	0.73	1.3 – 4.36[2]	0.96 – 3.67[2]	0.28 – 0.71[2]	0.75	4.36 - 6.47[2]
Tsensing [ns]	0.52	0.17	1.0	0.22	1.2	0.68	1.0	1.9	**0.11 - 0.12**[3]

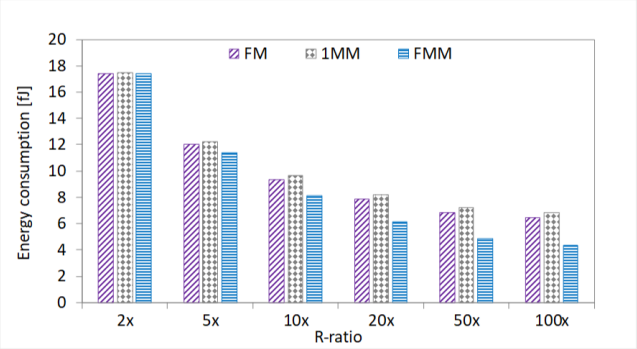


Figure 8. Energy consumption per bit in the proposed NV-CAM matching circuit on one search with fixed Width = 4W at 64-bit word length.

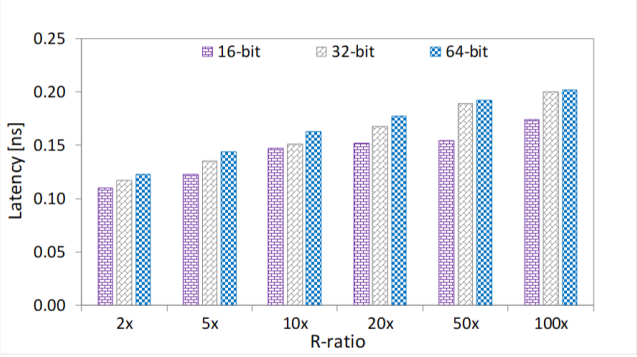


Figure 9. Sensing latency versus different R_{ratio} and word lengths.

cell structures (2.5T-1R, 2T-2R) come with the advantage of cell area. For technology independence, the cell area in minimum feature size (F^2) should be considered for the whole proposed CAM cells.

Search Error Rate: The traditional S-CAM design with 16T-SRAM based on CMOS NOR-type [22], has absolute reliability with an SER value of 0%, which outperforms all the rest.

Search energy: Usually, the search energy in CAM matching circuits is evaluated per bit on one search and this parameter is dependent on the presence of leakage current, ML discharging scheme, the applied bitline voltages, etc. The best search energy consumption is surprisingly belonging to a seemingly very bulky and power-hungry NV-CAM structure (15T-4MJT) [21] with the energy consumption per bit on one search only about 0.17 fJ.

Sensing latency: Our proposed 2T-2T cell structure has shown superiority in sensing latency compared to the remaining designs on this criterion. Moreover, this design achieved a much lower sensing latency compared to the second 2T-2R cell proposed. Besides techniques in circuit design, the distinction in the sensing latency of this work comes from the following applied measures: *First*, we used the current latch sense amplifier, which should have advantages in latency [29]. *Second*, we accept a small compromise in search energy caused by a high level of bitline voltage in return for lower sensing latency.

We summarized the highlights of all evaluated designs as follows: design [22] is the best in the classification of reliability, achieving absolute zero of SER as announced by the authors; design [21] is the best in the classification of energy-efficiency, presenting excellent results in terms of energy consumption; design [15] is the best in the classification of area occupancy according to both standard size (μm^2) and minimum feature size (F^2). Meanwhile, compared to the design [15], our proposed 2T-2R NV-CAM cell is about 2.8x larger in the normalized feature size but offers an extremely low sensing latency, approximately 15.8x-17.3x smaller. Among three CAM-cell structures based on the 65 nm technology process, design [22] despite getting good SER parameter but has a huge disadvantage related to the cell area, in contrary to design [16] and this work (take 3-nd and 2-nd place on this criterion). Compared to this work, design [16] requires two steps to perform the search operation, causing a larger sensing delay and increasing the complexity of the sensing circuit. However, our design requires higher search energy per bit than [16], which is explained by the large applied bitline voltage to achieve low SER values, while SER are absolutely not mentioned in [16].

[1]achieved with configuration (R_{ratio} = 100, the width of the access transistor = 4W, the word length is 64-bit)

[2]measured in cases of 1-bit MisMatch and Full MisMatch

[3]achieved with configuration (R_{ratio} = 2, the width of the access transistor = 4W, the word length varies from 16-bit to 64-bit)

V. CONCLUSION

Emerging non-volatile memory-based CAM cell structures (NV-CAM) have demonstrated potential ability as an alternative to traditional SRAM-based CAM cell structures integrated into frequent-search seldom-write IMC applications. In this paper, a compact 2T-2R NV-CAM cell structure is proposed with an efficient in-memory matching circuit for realizing reliable low-delay lookup operations. To verify the reliability and applicability of the proposed design, a variety of different NV-CAM designs have been chosen for comparative evaluation, which revealed that our 2T-2R CAM-cell structure outperforms the rest in search latency with an extremely small value in the range of 0.11-0.12 ns when employed STT-MRAM. In addition, our proposed design is one of the most compact bitcell structures with a cell area of only 0.51 μm^2, which can guarantee dependable lookup operations with search error rates in the range of 0.03-4.61%. Compared to the other CAM-cell structures evaluated in this paper, our proposed design is the best in search latency and at the same time is one of the most area-efficiency solutions. Moreover, this design can well meet almost the remaining design criteria with a small cost of average search energy. Due to all the presented advantages, our design could be the appropriate candidate for the NV-CAM cell under the 65 nm technology node.

ACKNOWLEDGMENT

This research is funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2018.310.

REFERENCES

[1] T. Kohonen, *Content-Addressable Memories*, ser. Springer Series in Information Sciences. Springer Berlin Heidelberg, 2012. [Online]. Available: https://books.google.com.vn/books?id=A-KpCAAAQBAJ

[2] R. Karam, R. Puri, S. Ghosh, and S. Bhunia, "Emerging trends in design and applications of memory-based computing and content-addressable memories," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1311–1330, 2015.

[3] H.-J. Tsai, K.-H. Yang, Y.-C. Peng, C.-C. Lin, Y.-H. Tsao, M.-F. Chang, and T.-F. Chen, "Energy-efficient tcam search engine design using priority-decision in memory technology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 962–973, 2017.

[4] T. Venkata Mahendra, S. Wasmir Hussain, S. Mishra, and A. Dandapat, "Energy-efficient precharge-free ternary content addressable memory (tcam) for high search rate applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 7, pp. 2345–2357, 2020.

[5] T. V. Mahendra, S. W. Hussain, S. Mishra, and A. Dandapat, "Low match-line voltage swing technique for content addressable memory," in *2019 7th International Conference on Smart Computing Communications (ICSCC)*, 2019, pp. 1–5.

[6] S. Jiang, P. Yan, and R. Sridhar, "A high speed and low power content-addressable memory(cam) using pipelined scheme," in *2015 28th IEEE International System-on-Chip Conference (SOCC)*, 2015, pp. 345–349.

[7] N. Mohan, W. Fung, and M. Sachdev, "Low-power priority encoder and multiple match detection circuit for ternary content addressable memory," in *2006 IEEE International SOC Conference*, 2006, pp. 253–256.

[8] Y.-H. Lee, I.-C. Lin, and S.-W. Wang, "Impacts of nbti and pbti effects on ternary cam," in *International Symposium on Quality Electronic Design (ISQED)*, 2013, pp. 38–45.

[9] I.-C. LIN, Y.-H. LEE, and S.-W. WANG, "Reducing aging effects on ternary cam," *IEICE Transactions on Electronics*, vol. E99.C, pp. 878–891, 07 2016.

[10] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (cam) circuits and architectures: a tutorial and survey," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, 2006.

[11] R. Bez and A. Pirovano, "Non-volatile memory technologies: emerging concepts and new materials," *Materials Science in Semiconductor Processing*, vol. 7, no. 4, pp. 349–355, 2004, papers presented at the E-MRS 2004 Spring Meeting Symposium C: New Materials in Future Silicon Technology. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369800104001003

[12] H.-S. P. Wong and S. Salahuddin, "Memory leads the way to better computing," *Nature nanotechnology*, vol. 10, no. 3, pp. 191–194, 2015.

[13] K. T. Quang, S. Ruocco, and M. Alioto, "Modeling the impact of dynamic voltage scaling on 1t-1j stt-ram write energy and performance," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 2313–2316.

[14] Q. K. Trinh, S. Ruocco, and M. Alioto, "Novel boosted-voltage sensing scheme for variation-resilient stt-mram read," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 10, pp. 1652–1660, 2016.

[15] J. Li, R. Montoye, M. Ishii, K. Stawiasz, T. Nishida, K. Maloney, G. Ditlow, S. Lewis, T. Maffitt, R. Jordan, L. Chang, and P. Song, "1mb 0.41 μm^2 2t-2r cell nonvolatile tcam with two-bit encoding and clocked self-referenced sensing," in *2013 Symposium on VLSI Technology*, 2013, pp. C104–C105.

[16] C.-C. Lin, J.-Y. Hung, W.-Z. Lin, C.-P. Lo, Y.-N. Chiang, H.-J. Tsai, G.-H. Yang, Y.-C. King, C. J. Lin, T.-F. Chen, and M.-F. Chang, "7.4 a 256b-wordlength reram-based tcam with 1ns search-time and 14× improvement in wordlength-energyefficiency-density product using 2.5t1r cell," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, 2016, pp. 136–137.

[17] C. Kim and K.-W. Kwon, "3t-2r non-volatile tcam with voltage limiter and self-controlled bias circuit," *Electronics Letters*, vol. 53, no. 13, pp. 837–839, 2017. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/el.2017.1027

[18] M.-F. Chang, L.-Y. Huang, W.-Z. Lin, Y.-N. Chiang, C.-C. Kuo, C.-H. Chuang, K.-H. Yang, H.-J. Tsai, T.-F. Chen, and S.-S. Sheu, "A reram-based 4t2r nonvolatile tcam using rc-filtered stress-decoupled scheme for frequent-off instant-on search engines used in iot and big-data processing," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 11, pp. 2786–2798, 2016.

[19] S. Matsunaga, A. Katsumata, M. Natsui, T. Endoh, H. Ohno, and T. Hanyu, "Design of a nine-transistor/two-magnetic-tunnel-junction-cell-based low-energy nonvolatile ternary content-addressable memory," *Japanese Journal of Applied*

Physics, vol. 51, no. 2, p. 02BM06, feb 2012. [Online]. Available: https://doi.org/10.1143/jjap.51.02bm06

[20] B. Song, T. Na, J. P. Kim, S. H. Kang, and S.-O. Jung, "A 10t-4mtj nonvolatile ternary cam cell for reliable search operation and a compact area," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 6, pp. 700–704, 2017.

[21] C. Wang, D. Zhang, L. Zeng, E. Deng, J. Chen, and W. Zhao, "A novel mtj-based non-volatile ternary content-addressable memory for high-speed, low-power, and high-reliable search operation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 4, pp. 1454–1464, 2019.

[22] W. Choi, K. Lee, and J. Park, "Low cost ternary content addressable memory using adaptive matchline discharging scheme," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–4.

[23] T.-N. Pham, Q.-K. Trinh, I.-J. Chang, and M. Alioto, "Stt-bnn: A novel stt-mram in-memory computing macro for binary neural networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 12, no. 2, pp. 569–579, 2022.

[24] V.-T. Nguyen, Q.-K. Trinh, R. Zhang, and Y. Nakashima, "Stt-bsnn: An in-memory deep binary spiking neural network based on stt-mram," *IEEE Access*, vol. 9, pp. 151 373–151 385, 2021.

[25] R. E. Thomas, A. J. Rosa, and G. J. Toussaint, *The analysis and design of linear circuits*. John Wiley & Sons, 2016.

[26] C. Augustine, N. N. Mojumder, X. Fong, S. H. Choday, S. P. Park, and K. Roy, "Spin-transfer torque mrams for low power memories: Perspective and prospective," *IEEE Sensors Journal*, vol. 12, no. 4, pp. 756–766, 2012.

[27] B. S. Kim, S. G. Park, Y. K. You, and S. I. Jung, "Probability & statistics for engineers & scientists," 2011.

[28] M. H. Abu-Rahma, Y. Chen, W. Sy, W. L. Ong, L. Y. Ting, S. S. Yoon, M. Han, and E. Terzioglu, "Characterization of sram sense amplifier input offset for yield prediction in 28nm cmos," in *2011 IEEE Custom Integrated Circuits Conference (CICC)*, 2011, pp. 1–4.

[29] A. K. Mishra, U. Chopra, and V. Dhandapani, "Comparative analysis in terms of power and delay of the different sense amplifier topologies," *Journal of Engg. Research EMSME Special Issue pp*, vol. 47, p. 57, 2021.

Quang-Manh Duong received BE Degree (2006) from the Moscow Aviation Institute in Moscow, Russia, and MS Degree (2012) from the Le Quy Don Technical University in electronic engineering in Hanoi, Vietnam. He was a lecturer and researcher in the Department of Microelectronics and Microprocessor Engineering, Faculty of Radio Electronics Engineering, Le Quy Don Technical University. His research interest includes low-power integrated circuit design, emerging memory technologies, and big data processing. Currently, he is working as a doctoral student at Le Quy Don Technical University.
Email: manhdq@lqdtu.edu.vn.

Quang-Kien Trinh (Member, IEEE) received a Ph.D. degree in computer engineering from the National University of Singapore, Singapore in 2018. He is working as the Deputy Head of The Department of Microprocessor Engineering, Le Quy Don Technical University, Hanoi, Vietnam. His research interests include low-power integrated circuit design, emerging memory technologies, and hardware security. He is the author and co-author of more than 40 publications (mostly IEEE) and 02 patents, and was TPC of many international conferences such as McSoC, ATC, NICS, INISCOM.
Email: kien.trinh@lqdtu.edu.vn.

Hai-Duong Nguyen received Ph.D. (2007) from The Bauman Moscow State Technical University in Russia. He specializes in teaching computer engineering since 1995 at Le Quy Don Technical University, Viet Nam. Currently, he is working as a senior researcher and the Head of the Department of Microelectronics and Microprocessor Engineering, Faculty of Radio Electronics Engineering, Le Quy Don Technical University. His research interest includes computing architecture, embedded systems, VLSI architecture for digital signal processing, and hardware security.
Email: mta.haiduongnguyen@gmail.com.

Van-Phuc Hoang received a Ph.D. degree from The University of Electro-Communications, Tokyo, Japan in 2012. He is working as an Associate Professor, and Director with the Institute of System Integration, Le Quy Don Technical University, Hanoi, Vietnam. His research interests include hardware security, digital circuits and systems, embedded systems for the Internet of Things, and VLSI architecture for digital signal processing. He was the Technical Program Chair of several IEEE international conferences such as ICDV 2017, MCSoC 2018, SigTelCom 2019, APCCAS 2020, ATC 2020, and ICICDT 2022. He is a member of IEEE.
Email: phuchv@lqdtu.edu.vn.

Hoang-Gia Vu received BS (2007) and MS (2010) Degrees in Electronic Engineering respectively from Le Quy Don Technical University, Vietnam; he received Ph.D. (2018) in Computer Engineering from Nara Institute of Science and Technology, Japan. Currently, he is a lecturer and researcher in the Department of Microelectronics and Microprocessor Engineering, Faculty of Radio Electronics Engineering, Le Quy Don Technical University. His research interest includes reconfigurable computing, computing architecture, and embedded systems.
Email: giavh@lqdtu.edu.vn.

Dinh-Ha Dao received BS (2011), MS (2015), and Ph.D. (2018) Degrees in Micro and Nanoelectronics respectively from the Faculty of Radioengineering and Electronics (Belarusian State University of Informatics and Radioelectronics, BSUIR) in Minsk, Belarus. Currently, he is a researcher in the Department of Microelectronics and Microprocessor Engineering, Faculty of Radio Electronics Engineering, Le Quy Don Technical University. He has authored or co-authored more than 30 publications. His research interest includes Sensor, Microelectronics and Semiconductor Engineering, Semiconductor Fabrication, and GaN Heterostructures.
Email: daodinhha@lqdtu.edu.vn.

Van-Toan Tran received his bachelor's and master's degrees in cybernetics and automation engineering from Le Quy Don Technical University, Hanoi, Vietnam in 2004 and 2014, respectively. He is currently pursuing a Ph.D. degree in electronic engineering with Le Quy Don Technical University. His research interest includes integrated circuit design and hardware security.
Email: trantoantbb@gmail.com.

Duy-Manh Luong received the B.S (2005) and M.S (2007) degrees from Vietnam National University, and the D.E (2016) degree in electronics engineering from the University of Electro-Communications, Japan. He worked as a postdoctoral researcher at Osaka University, Japan. He is currently deputy head of the department at Le Quy Don Technical University, Hanoi, Vietnam. His research interests include microwave semiconductor devices and circuits and THz integrated systems for wireless communication applications based on resonant tunneling diodes and photonic crystals.
Email: manhld@lqdtu.edu.vn.

Lambda Functions and Approximate Generalized Positive Boolean Dependencies

Huy Nguyen Xuan[1], Van Nguyen Thi[2]

[1] Institute of Information Technology, Vietnam Academy of Science and Technology, Ha Noi , Vietnam
[2] Hanoi Community College, Ha Noi, Vietnam

Correspondence: Van Nguyen Thi, van.cdcd@gmail.com
Communication: received 21 June, revised 9 August, accepted 30 August.
DOI: 10.32913/mic-ict-research.v2022.n2.1099

***Abstract*: The main purpose of the paper is to propose a lambda function and its apply to the concept of measure in comparing tuples of relations. Extend the generalized positive Boolean dependency to obtain a new type of dependency called approximate generalized positive Boolean dependency.**
The results can be applied in constructing more complicated databases, especially allowing extended search capabilities for the real-world data.

Keywords: *Generalized Boolean dependency, approximate generalized positive Boolean dependency, lambda function.*

I. Introduction

he theory of data dependencies play an important role in describing the real world and reflecting the data semantics in the databases. In 1970, Codd proposed the concept of relational database with the data constraints presenting as functional dependencies [5]. This concept is a useful tool for the database analysis and design.

Namely, attribute B is functionally dependent on attribute A in relation r, if each two tuples u and v in r are equal on attribute A then they must be equal on attribute B. For example, if H is the customer's height and S is his shirt size, then the functional dependency $H \to S$ requires two people to have the same height to receive the same shirt size. Similarly, the relationship between weight, age, and drug dosage should be understood as dependencies that should be relaxed.

The absolute dependencies above reveal the inadequacy when we have to deal with relations with thousands of tuples, all the while, there are only a few tuples not seem to be functional dependency accepted.

This loses the inherent flexible dependency between attributes. Therefore, from the beginning of the 80s of the last Century, studies have extended the concept of functional dependencies to the approximate dependencies. These dependencies allow us to describe the real world more accurately by replacing equality comparisons with approximate matching on tuples [3, 6, 8].

Along with the development of science and technology, these types of dependencies have been appearing more and more in the fields such as genetics, physics, molecular biology, materials technology, etc. The demand for a mathematical tool to help describe and reflect the loosened data dependence patterns in general and approximations in particular are necessary and natural [7, 9].

In this paper, continue the above research tendency, we are going to expand the general positive Boolean dependencies to obtain a new kind of dependencies, which called the approximate generalized positive Boolean dependencies.

II. Concepts and Notations

Let $U = x_1, ..., x_n$ be a finite set of Boolean variables that hold the values in the set of logical values $B = 0$ (FALSE), 1 (TRUE). Each Boolean formula can be constructed from U using the logical connectives $\wedge, \vee, \to, \neg$, and logic constants 1 and 0. A vector 0/1, $v = (v_1, ..., v_n)$ in B^n is called *an assignment*. Then each Boolean formula f, $f(v)$ is a value of f *on assignment* v. We denote e is the *unit assignment*, $e = (1, 1, ..., 1)$. A formula f is called positive if $f(e) = 1$. The formula $X \to Y, X, Y \subseteq U$ is called implication. It is clear that every implication formula is positive. Denote by $P(U)$ the set of all positive Boolean formulas over U, and by T_f the truth-table of $f, T_f = \{v \in B^n | f(v) = 1\}$. A set formulas F can be expressed as $F = \wedge\{f | f \in F\}$, so $T_F = \cap\{T_f | f \in F\}$ is the truth-table of F. It is known that $f \to g (F \to g)$ if and only if $T_f \subseteq T_g (T_F \subseteq T_g)$.

If U is a set of attributes, $A \in U$ and v is a tuple in a relation r on U, then $v.A$ is the value of variable A in tuple v. If $X \subseteq U$, then we define $v.X = \{v.A | A \in X\}$.

By convention of knowledge and database theory, some time, we use the following equivalent notations

A set of attributes (or logical variables) can be written as a sequence of symbols:

$$\{a, b, c\} \equiv abc$$
$$a \wedge b \equiv ab$$
$$a \vee b \equiv a + b$$
$$\neg a \equiv a'$$

Other traditional notations and conventions can be found in the database literature [1, 4, 5].

We assume that each domain V_x of attribute x in U contains at least two values. For each domain V_x, we consider a mapping α_x: $V_x^2 \to \{0,1\}$ which satisfies the following properties [1, 6]:

$\forall a, b \in V_x$

$A1$) *Reflexivity* $\alpha_x(a,a) = 1$

$A2$) *Symmetry* $\alpha_x(a,b) = \alpha_x(b,a)$

$A3$) *Partiality* $\exists c \in V_x : \alpha_x(a,c) = 0$.

α_x is a partial relation on domain V_x satisfying reflexivity and symmetry.

It is easy to see that equal mapping on V_x, denoted by $=_x$, satisfies the properties $(A1), (A2)$ and $(A3)$.

Each positive Boolean formula f in $P(U)$ with the given comparison α is called *generalized positive Boolean dependency* (GPBD). A pair $p = (U, F)$, where U is a finite set of attributes with the comparisons α on domains, F is a finite set of generalized positive Boolean dependencies is called *relational schema of generalized positive Boolean dependencies*.

Let $p = (U, F)$ be a schema, and r be a relation on U. For each pair of tuples $u = (u_1, u_2, ..., u_n), v = (v_1, v_2, ..., v_n)$ in r, we set a vector 0/1 $t = \alpha(u, v) = (t_1, t_2, ..., t_n) \in B^n$ where the $t_i = \alpha_i(u_i, v_i), 1 \le i \le n$.

We call the set $T_r = \{\alpha(u,v) | u, v \in r\}$ value-table of relation r on the schema p [1, 2, 3, 6].

A relation r on U satisfies a GPBD f (set of GPBDs) and is writen as $r(f)$, $(r(F)$ if $T_r \subseteq T_f (T_r \subseteq T_F)$.

Example 2.1

Consider the relation r describing individuals with the following attributes: bloodline H, father's ADN F and mother's ADN M. We see that two individuals with the same bloodline have the same father's ADN or mother's ADN. We have $H \to F + M$ is a GPBD. With relation $r(H, F, M)$ containing information about bloodline, we construct the value-table of T_r as follows:

TABLE 2.1
RELATION r WITH ATRIBUTES: BLOODLINE H, FATHER'S ADN: F AND MORTHER'S ADN: M

Tuple	H	F	M
1	h1	f0	m2
2	h2	f1	m1
3	h1	f0	m3
4	h3	f2	m4
5	h2	f3	m1

TABLE 2.2
VALUE-TABLE OF RELATION r

pair of tuples	H	F	M
(1,1)	1	1	1
(1,2)	0	0	0
(1,3)	1	1	0
(2,5)	1	0	1

TABLE 2.3
TRUTH-TABLE OF FUNCITON $H \to F + M$

H	F	M
1	1	1
1	0	1
1	1	0
0	1	0
0	0	1
0	1	1
0	0	0

Let U be a finite set of attributes. Denote by $REL(U)$ the set of all relations on U, $REL2(U)$ the set of all relations with no more than two tuples on U. Let F be a set of GPBDs and f be a GPBD. We say that

- F implies f by logic and denote as $F \models f$ if $T_F \subseteq T_f$
- F implies f by relation, and denote as $F \vdash f$ if
$$\forall r \in REL(U) : r(F) \to r(f)$$
- F implies f by no more than two tuple, denote $F \vdash_2 f$ if
$$\forall r \in REL2(U) : r(F) \Rightarrow r(f)$$

Equivalence theorem

Let F be a set of GPBDs and f be a GPBD. The following are equivalent [1, 6]:

$(i) F \models f(logicconsequence)$

$(ii) F \vdash f(relationconsequence)$

$(iii) F \vdash_2 f(two - tuplerelationconsequence)$

The equivalence theorem allows us to determine when a GPBD f is con-sequenced from a set of GPBDs F through formal logical transformations without relying on data.

III. Lambda function

Let U be an attribute set. For each attribute A in U, we define a mapping λ_A from the value domain of the attribute to the real number space as follows:

Let U be an attribute set. For each attribute A in U, we define a mapping λ_A from the value domain of the attribute to the real number space as follows:

$\lambda_A : V_A \to \mathbb{R}$

$L1)$:$\forall a, b \in V_A : a = b \Rightarrow \lambda_A(a) = \lambda_A(b)$

$L2)$: $\exists a, b \in V_A$: $\lambda_A(a) \neq \lambda_A(b)$

λ_A is called a quantitative function for the attribute A.

Example 3.1

- Let attribute A be the type string, we define $\forall s \in V_A : \lambda_A(s) = len(s)$. We have,

$\lambda_A(\text{“}Internet\text{”}) = 8$; $\lambda_A(\text{“}e - mail\text{”}) = 6$;

$\lambda_A(\text{“”}) = 0$ (the quantification of empty string is 0).

- Let attribute B be the salary coefficient we define $\forall \in V_B : \lambda_B(a) = a$. Then
$$\lambda_B(2.3) = 2.3; \lambda_B(8.0) = 8.0$$
- Let attribute C be the level of satisfaction rank (the customer for an airline

company, for example) with the values $V_C = \{verysatisfied, satisfied, normal, notsatisfied\}$ we define

$\lambda_C(verysatisfied) = 4$; $\lambda_C(satisfied) = 3$; $\lambda_C(normal) = 2$; $\lambda_C(notsatisfied) = 1$.

Theorem 3.1

Let λ_A be given, then the function α_A is defined by the following formula (*)

$$\forall a, b \in V_A: \alpha_A(a,b) \overset{def}{\Longleftrightarrow} (\lambda_A(a) = \lambda_A(b)) \quad (*)$$

Satisfy properties $A1, A2$ and $A3$ in the definition of α.

Proof

Suppose $a \in V_A$. Then $\lambda_A(a) = \lambda_A(a)$ therefore $\alpha_A(a,a) = 1$. Reflective properties $A1$ is proven.

Suppose $a, b \in V_A$. We have

$$\alpha_A(b,a) = 1 \overset{def}{\Longleftrightarrow} \lambda_A(a) = \lambda_A(b) \overset{def}{\Longleftrightarrow} \alpha_A(b,a) = 1.$$

The A2 symmetry properties is proven.

According to the definition of the Lambda function, $\exists x, y \in V_A : \lambda_A(x) \neq \lambda_A(y)$, implies $\alpha_A(x,y) = 0$.

The A3 Partiality properties is proven.

The theorem is proven

Measure

The measure d on the set V is a mapping $d : V \times V \to \mathbb{R}^+$ where $\mathbb{R}^+$ is the set of non-negative real numbers, satisfying the following properties : $\forall a, b, c \in V$.

- $Non - negativity : d(a,b) \geq 0,\ d(a,a) = 0$
- $Symmetry : d(a,b) = d(b,a)$
- $Triangle : d(a,b) + d(b,c) \geq d(a,c)$.

Example 3.2

On line (one-dimensional space), the measure between two points x and y representing the distance from x to y is calculated by $d(x,y) = ||x - y||$. In n-dimensional space, the measure between two points $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ is defined over the $Euclid$ distance as follows:

$$e(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

The A2 symmetry properties is proven.

Let $p = (U, F)$ be a Lambda functions defined for each attribute. For each tuple $u = (u_1, u_2, ..., u_n) \in r$, we define $\lambda(u)$ as a vector

$$\lambda(u) = (\lambda_1(u_1), \lambda_2(u_2), ..., \lambda_n(u_n))$$

Theorem 3.2

Given a relation r on the attribute set U and Lambda functions on each attribute A in U. Let d be an arbitrary measure in an n-dimensional vector space. Then the following d function is a measure:*

$$\forall u, v \in r : d^*(u,v) = d(\lambda(u), \lambda(v))$$

We call d^* the *inductive measure* from the d measure.

Proof

We proof that the function d^* satisfies the three properties of the measure definition. Indeed, because d is non-negative, then $d^*(u,v) = d(\lambda(u), \lambda(v)) \geq 0$. If $u = v$ then $\lambda(u) = \lambda(v)$, therefore $d^*(u,v) = d(\lambda(u), \lambda(v)) = 0$.

The symmetry property of d^* is directly implied from the symmetry property of the measure d.

Assume u, v and w are three tuples in r. Since d satisfies is a measure, d satisfies the triangular property, we have,

$$d^*(u,v) + d^*(v,w) = d(\lambda(u), \lambda(v)) + d(\lambda(v), \lambda(w)) \geq d(\lambda(u), \lambda(w)) = d^*(u,w).$$

The triangle property of d^* is proven

Theorem is proven.

IV. Some special case of generalized positive Boolean varieties

Let U be an attribute set, $X, Y \subseteq U, r$ be a relation on U.

1. Functional dependencies

A $functionaldependency$ (FD) on U is a formula $f : X \to Y$. We say the relation $rsatisfies$ FD f and write $r(X \to Y)$ if for every pair of tuples u and v in r that are equal on X then they must be equal on Y :

$$r(X \to Y) \overset{def}{\Longleftrightarrow} \forall u, v \in r: u.X = v.X \Rightarrow u.Y = v.Y$$

Thus, FD is a special case of GPDB when the positive Boolean formula is the consequence formula $X \to Y$ and the comparison is equality.

2. Weak dependencies

A formula of the form $f : \wedge X \to \vee Y$ is called weakly derived formula. A weak dependency (WD) on U is a weakly derived formula $f : \wedge X \to \vee Y$ is called weakly derived formula.

We say the relation r satisfies WD f and write $r(\wedge X \to \vee Y)$ if for every pair of tuples u and v in r that are equal on X then they must be equal on some attribute B of Y.

$$r(\wedge X \to \vee Y) \Leftrightarrow (\forall u, v \in r): (u.X = v.X) \Rightarrow \exists B \in Y: \ (u.B = v.B)$$

Thus, an WD is a special case of the positive Boolean dependency. We also have:

The $relation\ r\ satisfies\ the\ weak\ dependency\ f$ (the set of WDs F) if and only if $T_r \subseteq T_f (Tr \subseteq T_F)$.

In [2], the following results is proven:

Theorem 4.1

textitEach positive Boolean formula is equivalent to a set of weakly derived formulas (in the sense that they all have the same truth-table).

Example 4.1

The positive Boolean formula $f = (a + b) \rightarrow (c \rightarrow d)$ is equivalent to the set of weakly derived formulas $S = \{bc \rightarrow a + d, ac \rightarrow b + d, abc \rightarrow d\}$

Example 4.2

In the bloodline relation $r(H, F, M), B + M$ is weakly dependent on H.

3. Approximate Weak Dependency (AWD)

The concept of approximate functional dependencies was proposed by Jyrky Kivinen et al. in 1995 [7, 8, 9], where the term *approximation* is understood as if at least ε tuples are removed from the relation r then the rest will satisfy the given function dependency. In this paper, a general concept of approximate dependence is proposed on a set of numeric and non-numerical attributes, according to an arbitrary positive Boolean formulas with more general comparisons.

Given an attribute set U with a measure d and a weak derivation formula on $U, f : \wedge X \rightarrow \vee Y$ Let $\varepsilon_1, \varepsilon_2$ are two non-negative real numbers. The approximate weak dependency (AWD) of attribute set Y on attribute set X is an expression of the form:

$$X \ \varepsilon_1 \rightarrow \varepsilon_2 \ Y$$

We say that the relation $r(U)$ satisfies the *approximate weak dependence* $X \ \varepsilon_1 \rightarrow \varepsilon_2 \ Y$ if u and v in the relation r, the two tuples u and v have

$$\forall u,v \in r\text{: } d(u.X, v.X) \leq \varepsilon_1 \Rightarrow \exists B \in Y\text{: } d(u.B,\ v.B) \leq \varepsilon_2$$

Example 4.3

On the promotional day, a supermarket selling item H has a policy to give each customer a discount of 10 units of the assumed currency T from 1 PM onwards. Information at the cashier is given in the form of the relation r as follows:

TABLE 4.1
RELATION r

ID	Time	Quantity	Price	Total
1	10	5	10	50
2	11	8	10	80
3	12	4	10	40
4	13	5	10	40
5	14	8	10	70
6	15	4	10	30

Consider the following dependencies:

- Functional Dependency:
  $h : Quantity, price \rightarrow Total$
- Weak Dependency :
  $w : Quantity, price \rightarrow Total$
- Approximate Weak Dependency :
  $f : Quantity, price(0) \rightarrow (10)Total$

We can see that two transactions 1 and 4 buy the same quantity but at two different times, then the Total has two different values, so the relation r is not satisfied functional dependency h.

Similar, relation r is not satisfied approximate weak dependency w.

However, relation r satisfies the approximate weak dependency f: two customers buying the same quantity at different times will pay different sums: Total and Total - 10.

We call $p = (U, \varepsilon_1, \varepsilon_2, \lambda, d, F)$ a *relation schema with approximate weak dependencies* (RSAWD), where

- U is a set of n attributes, each of which is equipped with a quantitative Lambda function λ.
- d is an arbitrary measure.
- F is a set of weakly consequence formulas $\wedge X \rightarrow \vee Y, X, Y \subseteq U$
- $\varepsilon_1 \geq 0, \varepsilon_2 \geq 0$ are approximate thresholds.

Given a relation r $RSAWD$, $p = (U, \varepsilon_1, \varepsilon_2, \lambda, d, F)$. For each pair of tuples $u = (u_1, u_2, ..., u_n), v = (v_1, v_2, ..., v_n)$ in r, we define:

$$t(u, v) = (t_1, t_2, ..., t_n)$$
$$t_i = (d(\lambda(u_i), \lambda(v_i)) \leq \varepsilon)?1 : 0; 1 \leq i \leq n$$
$$\varepsilon = min(\varepsilon_1, \varepsilon_2)$$
$$T_r = \{t(u, v) | u, v \in r\}$$

and call T_r the value table of the relation r according to the scheme p.

Theorem 4.2

Let $p = (U, \varepsilon_1, \varepsilon_2, \lambda, d, F)$ and a $RSAWD$ be an AWD. Let r be a relation on U. Then,

(i) The relation r satisfies AWD f if and only if $T_r \subseteq T_f$

(ii) Relation r satisfies the set of $AWDs$ F if and only if $T_r \subseteq T_F$.

Proof

(i) Suppose the relation $r(U)$ satisfies AWD f. We need to prove $T_r \subseteq T_f$. If r is empty, then since $T_r = \emptyset \subseteq T_f$ so $T_r \subseteq T_f$. Let $\varepsilon = min(\varepsilon_1, \varepsilon_2)$ and $u \in r$. Since $t(u, u) = e \in T_r$. Since f is a positive Boolean formula, then $e \in T_f$. Suppose $t \in T_r$, by the definition of $T_r, \exists u, v \in r : t(u, v) = t$. Since relation r satisfies f, we have $f(t) = 1$, so $t \in T_f$.

In the other hand, suppose $T_r \subseteq T_f$. We need to show that relation r satisfies AWD f. For every pair of tuples u, v in r, we have $t(u, v) \in T_r$. Since $T_r \subseteq T_f$, then $t(u, v) \in T_f$. This implies $f(t(u, v)) = 1$, that r satisfies AWD f.

(ii) By the definition of T_F we have

$$T_F = \cap\{T_f | f \in F\}$$

According to the proof (i), relation r satisfies the set F if and only if for every f in F we have $T_r \subseteq T_f$. Since T_F is the intersection of $Tf, T_r \subseteq T_F$. Otherwise, if $T_r \subseteq T_F$, then $T_r \subseteq T_f$ for every f in F. By proof (i), r satisfies

all AWD f, so relation r satisfies the set of $AWDs$ F. The theorem is proved.

4. Approximate generalized positive Boolean dependency

Definition 4.1

Let U be an attribute set and f be a positive Boolean formula f over U. Let $\varepsilon_1, \varepsilon_2$ be two non-negative thresholds. Approximate generalized positive Boolean dependency (AGPBD) is a set of approximate weak dependencies S equivalent to f with the threshold $\varepsilon_1, \varepsilon_2$.

We say that the relation $r(U)$ satisfies AGPBD f if r satisfies every AWD in S.

Since f is equivalent to S, then $T_f = T_S = \cap\{T_g | g \in S\}$. We have the following result as a corollary of theorem 4.1:

Corollary 4.1

Let $p = (U, \varepsilon_1, \varepsilon_2, \lambda, d, F)$ be a RSAWD and f. Let r be a relation on U. Then,

(i) The relation r satisfies AGPBD f if and only if $T_r \subseteq T_f$.

(ii) Relation r satisfies the set AGPBD F if and only if $T_r \subseteq T_F$.

V. Conclusion

The classification and proposal of a common model for types of data dependencies is one of the issues of great interest to big data researchers. With an arbitrary measure, through Lambda functions on the value domain of the attributes, we can evaluate the approximation of tuples in the relation including numeric and non-numeric attributes of a database.

References

[1] Nguyen Xuan Huy (2006), *Logical dependencies in the database*, Statistical Publishing House (Vietnamese).

[2] Nguyen Xuan Huy, Truong Thi Thu Ha (2014), "Equivalence between positive and weak Boolean dependencies in relational databases", *Proceedings of the 17th National Conference: Some selected issues of Information and Communication Technology, Dak Lak, 30-31/10/2014, Science and Technology Publishing House*, ISBN: 978-604-67-0426-3, Hanoi, page.361-365. (Vietnamese)

[3] Nguyen Xuan Huy, Truong Thi Thu Ha, Nguyen Thi Van (2016), "Correlation between approximate functional dependency and positive Boolean dependency in general in relational databases, *Proceedings of the 19th National Conference: Some selected issues of Information and Communication Technology, Science and Technology Publishing House, ISBN: 978-604-67-0781-3, Hanoi, page.361-365. (Vietnamese)*

[4] Berman J., Blok W. J. (1988), "Positive Boolean dependencies", Inf. *Processing Letters, 27*, p.147 – 150.

[5] Codd E. F. (1970), "A Relational Model of Data for Large Shared Data Banks", *CACM 13:6*, 377-387.

[6] Huy Nguyen Xuan, Thanh Le Thi (1992), "Generalized Positive Boolan Dependencies", J. *Inf. Process. Cybern. EIK*, vol. 28, p. 363 – 370.

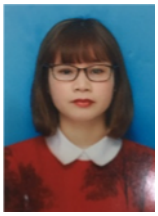
[7] Jalal Atoum (2009), "Mining Approximate Functional Dependencies from Databases", *European Journal of Scientific Research*, ISSN 1450 – 216X vol. 33 no. 2, p.338 – 346.

[8] Jyrky Kivinen et al (1995), "Approximate Inference of Funtional Dependencies from Relations", *Journal of Theoretical Computer Science*, Vol.149 Issue 1, p.129 – 149.

[9] Ronald S. King, James J. Legendre (2003), "Discovery of Functional and Approximate Functional Dependencies in Relational Databases", *J. of Appl. Math. and Decision Scienses*, 7(1), 49-59.

[10] Truong Thi Thu Ha (2017), "Correlation between logically dependent classes in the database", *PhD thesis, V-LA2/4042 – Military Technical Institute*. (Vietnamese)

Nguyen Xuan Huy, Institute of Information Technology, Vietnam Academy of Science and Technology Nguyen Xuan Huy received Math BS from the Herzen University, Saint Petersburg (1973), PhD in Computer Science (1980), and Doctor of Sciences in Computer Science (1990), Academy of Sciences, Russia. He is an associate professor and a senior researcher. He is the author of papers and books in computer science and database systems. He held positions in the departments of computer science of the State Thai Nguyen University, and Da Lat University. He is a visiting lecturer of the Ha Noi State University, Ha Noi Posts and Telecommunications Institute of Technology, State University in Ho Chi Minh city, Da Nang State University, Duy Tan University, and Qui Nhon University. Huy was the chairman of the Advisory Board of the Microsoft Project Partners In Learning (PIL) in Viet Nam (2004-2012). He is a member of Vietnam Association for Information Processing (VAIP, 1982), Vietnam Mathematical Society (VMS, 1982)
Email: nxhuy564@gmail.com.



Nguyen Thi Van received the B.S of IT at Hanoi University of Business and Technology, 2011 and M.S. degree in Computer Science at University of Information and Communication Technology, 2014. Currently working at, Hanoi Community College. Research area: Logical dependencies in databases, data models and database

Email: van.cdcd@gmail.com