Formal Analysis of Medical Systems using Multi-Agent Systems with Information Sharing

Bogdan Aman Gabriel Ciobanu

Abstract

Improving safety is a main objective for medical systems. To assist the modelling and formal analysis of medical systems, we define a language for multi-agent systems handling information, timed communication, and timed migration. We use a simplified airway laser surgery scenario to demonstrate our approach. An implementation in Maude is presented; we use the strategies allowed by Maude to guide the rules application in order to decrease substantially the number of possible executions and results in the highly nondeterministic and concurrent multi-agent systems. Finally, we present how the executable specifications can be verified with the model-checking tools in Maude to detect the behavioural problems or desired properties of the agents.

Keywords: multi-agent systems, rewriting engine Maude, strategies and model-checking, example of airway laser surgery.

MSC 2020: 68Q42, 68Q60, 68Q85, 93A16.

ACM CCS 2020: Theory of computation → Equational logic and rewriting, Software and its engineering → Model checking, Theory of computation → Process calculi.

1 Introduction

Information about patients history, diagnostics, drugs, and treatment methods is growing very fast and became more distributed in various locations. The challenge is to collect properly the relevant information and to use it smarter. Software agents can be used in medicine to collect information from many different locations and provide relevant

doi:10.56415/csjm.v32.01

assistance by presenting an integrated view and unexpected relationships or treatment procedures. These agents support decision-making by accessing distributed resources and coordinating the actions in complex medical processes, and potentially avoid failures in medical procedures. Agent-based systems overcome the weakness of centralized systems, improving the performance and providing flexibility, scalability, and robustness.

Software agents work in heterogeneous and distributed networks. An agent operates without the direct intervention of humans and interacts with other agents. A multi-agent system is a collection of autonomous software agents coordinated to solve larger problems. Usually, the information and knowledge required to solve a large problem are distributed in several locations of the network; information is obtained by moving the agents from a location to another. Multi-agent systems coordinate the actions and interactions of these migrating agents to provide solutions for a complex problem.

There already exist articles presenting some advantages of the agent technology and case studies of multi-agent systems in real medical domains [1], [2]. In this article, we present something new: how to use execution strategies in multi-agent systems in medical environments to reduce their evolution, and how to verify automatically their behaviour.

In multi-agent systems, an agent can be characterized by several properties (e.g., cooperation, learning, mobility [3]). In this article, we consider a language of multi-agent systems named iMAS , which is an extension of TiMo [4] with timeouts for communication and mobility, located agents acting in parallel locations and able to migrate between locations. The interaction among agents is given by message-passing communication. Such a system has public information that can be accessed by all agents, while each agent has its private information.

We consider the next example taken from [5] which illustrates the multi-agent systems with information sharing; this example involves information sharing and mobility in space and time: "For airway laser surgery, there are two potential dangers: (1) an accidental burn if both laser and ventilator are activated; and (2) a low-oxygen shock if the Saturation of Peripheral Oxygen (SpO) level of the patient decreases below a given threshold (assume 95%). To prevent the potential dangers when the surgery starts, the airway laser turns on and notifies the ventilator to turn off; and when the patient's SpO level becomes below 95%, the ventilator turns on and notifies the airway laser to turn off."

We provide an implementation in the rewriting engine Maude, and use the strategies allowed by Maude to guide the rules application in order to decrease substantially the number of possible results in the highly nondeterministic and concurrent multi-agent systems. Moreover, we verify their behaviour by using Maude model-checking tools.

The structure of the paper is as follows: Section 2 presents the syntax and semantics of our language iMAS and illustrates them with a running example. Section 3 contains an implementation of iMAS in Maude. In Section 4, we use strategies available in Maude, and verify that the agents behave as intended by means of model-checking tools of Maude. Conclusion and references end the article.

2 Multi-Agent Systems with Information Sharing: Syntax and Semantics

We consider a language named iMAS , where 'i' stands for 'information' and 'MAS ' stands for 'multi-agent system'. Table 1 contains the *syntax* of iMAS, where:

∗ *Loc*= $\{l, l', \ldots\}$ is a location set, *Chan*=$\{a, b, \ldots\}$ is channel set, *Id*= $\{id, \ldots\}$ is a name set for recursive processes, and $\mathcal{N} = \{N, N', \ldots\}$ is a network set;

∗ $id(v) \overset{def}{=} P_{id}$, for all $id \in Id$, is a unique process definition; $t \in \mathbb{N}$ is an action timeout, $k \in \mathbb{Z}$ is a threshold, u is a variable, v is an expression over variables and values, f is a field, and p is the type of the accessed information: either *private* to indicate the information belonging to an agent or *public* to indicate the information belonging to the entire location. If $Q(u)$ is a process definition, where $Q \in Id$ and $v_1 \neq v_2$, then $Q(v_1)$ and $Q(v_2)$ are different.

An agent $P \rhd I$ behaves according to process P and has *private* information I. An agent $\mathsf{go}^t\, l\ \mathsf{then}\ P \rhd I$ cannot move for t units of

Table 1. Syntax of our Multi-Agent Systems

Processes	$P, Q ::=$	$\mathsf{go}^t\ l$ then P	(move)
	$\mid$	$a^{\Delta t}!\langle v\rangle$ then P else Q	(output)
	$\mid$	$a^{\Delta t}?(u)$ then P else Q	(input)
	$\mid$	if $test$ then P else Q	(branch)
	$\mid$	$stop$	(termination)
	$\mid$	$id(v)$	(recursion)
	$\mid$	$updA(p, f, v)$ then P	(asynch update)
	$\mid$	$updS(a, p, f, v)$ then P	(synch update)
Information	$I ::=$	$\emptyset \mid \langle f; v\rangle \mid I\ I$	
Tests	$test ::=$	$true \mid \neg test \mid test \wedge test$	
		$\mid get(p, f) > k \mid \ldots$	$p \in \{private, public\}$
Agents	$A, B ::=$	$P \triangleright I$	
Set of Agents	$\tilde{A} ::=$	$\mathbf{0} \mid \tilde{A} \parallel A$	
Networks	$N ::=$	$void \mid l[[I \triangleleft \tilde{A}]] \mid M \mid M$	

time; afterwards the agent $P \triangleright I$ moves at location l. Since l can be instantiated after communication, agents can adapt their behaviours.

An agent $a^{\Delta t}!\langle v\rangle$ then P else $Q \triangleright I$ waits for up to t units of time to communicate the value v on channel a to an agent $a^{\Delta t'}?(x)$ then P' else $Q' \triangleright I'$ awaiting at the same location a value to be written on variable x for up to t' units of time. If communication happens, the agents become $P \triangleright I$ and $\{v/x\}P' \triangleright I'$ (all the free occurrences of the variable x are replaced by value v in P') and are available at the same location. If the timers expire, then the agents become $Q \triangleright I$ and $Q' \triangleright I'$.

An agent if $test$ then P else $Q \triangleright I$ checks $test$ using the available information (*public* and *private*). If the *test* returns *true*, then the agent becomes $P \triangleright I$; otherwise, the agent becomes $Q \triangleright I$. For example, a test $get(private, f) > k$ returns *true* only if the value stored in the field f of the private information I is greater than k.

The agent $updA(p, f, v)$ then $P \triangleright I$ with $p \in \{private, public\}$ has two possible outcomes: (i) if the p information does not have a field f, then a new piece of information $\langle f; v\rangle$ is added to p; (ii) if the p information does have a field f, then its value is updated to v. The agent continues with the same process P.

The agent $\mathit{updS}(a, p, f, v)$ then $P \rhd I$ has a similar behaviour as $\mathit{updA}(p, f, v)$ then $P \rhd I$, except that the update is performed only if there exists another agent $\mathit{updS}(a, p', f', v')$ then $P' \rhd I'$ at the same location and using the same channel a that is ready to perform an update. $A = 0 \rhd I$ has no action to execute and terminates.

A network is composed of parallel locations of the form $l[[I \lhd \tilde{A}]]$, where l is a location with *public* information I and a set $\tilde{A}$ of agents; an empty location is denoted by $l[[\emptyset \lhd \mathbf{0}]]$.

The structural equivalence rearranges agents so that they interact. This is needed in the *operational semantics* presented in Table 2; we present only some of the rules as the others are similar. We denote by $N \xrightarrow{\Lambda} N'$ a network N that transforms into a network N' by executing the multiset of actions Λ.

In rule (Stop), we use $\not\rightarrow$ to denote that no action can be executed if no agents are available. Rule (Com) models two agents at the same location l able to communicate on the same channel a. After a successful communication, the agents become $P_1 \rhd I_1$ and $\{v/u\}P_2 \rhd I_2$.

Rule (Put0) is used for an agent to remove its current output action if its timer is zero. Afterwards, the agent can execute Q using the unchanged information I. A similar rule (Get0) is available for the input action. Since rule (Com) can be applied even when the timers are zero, it follows that one of the rules (Com), (Put0), and (Get0) is nondeterministically applied.

Rule (Move0) is used when an agent migrates from location l to location l' and becomes $P \rhd I$. Rule (IfT) is used when the test performed by an agent returns *true*. A similar rule (IfF) is available when the *test* is *false*.

Rules (CrtPr) and (UpdPr) are used when an agent extends or updates, in an asynchronous manner, its *private* or *public* information; afterwards the agent becomes $P \rhd I\langle f; v\rangle$. Similar rules are available for the synchronous updates *updS*.

Rule (Call) transforms an agent $id(v) \rhd I$ into $\{v/u\}P_{id} \rhd I$. Other rules are available to compose smaller subnetworks, and to apply the structural equivalence over networks.

Table 2. Operational Semantics for our Multi-Agent Systems

(Stop)
$$l[[I \lhd \mathbf{0}]] \not\rightarrow$$

(Com)
$$\begin{array}{c} l[[I_l \lhd a^{\Delta t_1}!\langle v\rangle \ \mathsf{then}\ P_1 \ \mathsf{else}\ Q_1 \rhd I_1 \\ \ ||\ a^{\Delta t_2}?(u) \ \mathsf{then}\ P_2 \ \mathsf{else}\ Q_2 \rhd I_2 \ ||\ \tilde{A}]] \\ \xrightarrow{a!?@l} l[[I_l \lhd P_1 \rhd I_1 \ ||\ \{v/u\}P_2 \rhd I_2 \ ||\ \tilde{A}]] \end{array}$$

(Put0)
$$l[[I_l \lhd a^{\Delta 0}!\langle v\rangle \ \mathsf{then}\ P \ \mathsf{else}\ Q \rhd I \,||\, \tilde{A}]] \xrightarrow{a!^{\Delta 0}@l} l[[I_l \lhd Q \rhd I \,||\, \tilde{A}]]$$

(Move0)
$$\begin{array}{c} l[[I_l \lhd \mathsf{go}^0\, l' \ \mathsf{then}\ P \rhd I \ ||\ \tilde{A}]] \mid l'[[I'_l \lhd \tilde{B}]] \\ \xrightarrow{l \rhd l'} l[[I_l \lhd \tilde{A}]] \mid l'[[I'_l \lhd P \rhd I \ ||\ \tilde{B}]] \end{array}$$

(IfT)
$$\begin{array}{c} \mathit{test}@(I\ I_l) = \mathit{true} \text{ implies} \\ l[[I_l \lhd \mathsf{if}\ \mathit{test}\ \mathsf{then}\ P \ \mathsf{else}\ Q \rhd I \ ||\ \tilde{A}]] \xrightarrow{\mathit{true}@l} l[[I_l \lhd P \rhd I \,||\, \tilde{A}]] \end{array}$$

(CrtPr)
$$\begin{array}{c} \nexists\langle f; v'\rangle \in I \text{ implies} \\ l[[I_l \lhd \mathit{updA}(\mathit{private}, f, v) \ \mathsf{then}\ P \rhd I \ ||\ \tilde{A}]] \\ \xrightarrow{\mathit{create}_{lf}@l} l[[I_l \lhd P \rhd I\langle f; v\rangle \ ||\ \tilde{A}]] \end{array}$$

(UpdPr)
$$\begin{array}{c} l[[I_l \lhd \mathit{updA}(\mathit{private}, f, v) \ \mathsf{then}\ P \rhd I\langle f; v'\rangle \ ||\ \tilde{A}]] \\ \xrightarrow{\mathit{upd}_{lf}@l} l[[I_l \lhd P \rhd I\langle f; v\rangle \ ||\ \tilde{A}]] \end{array}$$

(Call)
$$\begin{array}{c} l[[I_l \lhd \mathit{id}(v) \rhd I \ ||\ \tilde{A}]] \xrightarrow{\mathit{call}@l} l[[I_l \lhd \{v/u\}P_{\mathit{id}} \rhd I \ ||\ \tilde{A}]], \\ \text{where } \mathit{id}(u) \overset{\mathit{def}}{=} P_{\mathit{id}} \end{array}$$

(DStop)
$$l[[I_l \lhd \mathbf{0}]] \overset{t}{\rightsquigarrow} l[[I_l \lhd \mathbf{0}]]$$

(DPut)
$$\begin{array}{c} t \geq t' \geq 0 \text{ implies } l[[I_l \lhd a^{\Delta t}!\langle v\rangle \ \mathsf{then}\ P \ \mathsf{else}\ Q \rhd I]] \\ \overset{t'}{\rightsquigarrow} l[[I_l \lhd a^{\Delta t - t'}!\langle v\rangle \ \mathsf{then}\ P \ \mathsf{else}\ Q \rhd I]] \end{array}$$

(DGet)
$$\begin{array}{c} t \geq t' \geq 0 \text{ implies } l[[I_l \lhd a^{\Delta t}?(u) \ \mathsf{then}\ P \ \mathsf{else}\ Q \rhd I]] \\ \overset{t'}{\rightsquigarrow} l[[I_l \lhd a^{\Delta t - t'}?(u) \ \mathsf{then}\ P \ \mathsf{else}\ Q \rhd I]] \end{array}$$

(DMove)
$$\begin{array}{c} t \geq t' \geq 0 \text{ implies } l[[I_l \lhd \mathsf{go}^t\, l' \ \mathsf{then}\ P \rhd I]] \\ \overset{t'}{\rightsquigarrow} l[[I_l \lhd \mathsf{go}^{t-t'}\, l' \ \mathsf{then}\ P \rhd I]] \end{array}$$

We denote by $N \overset{t}{\rightsquigarrow} N'$ a network N that transforms into a network N' after t units of time. In rule (DStop), the network $l[[I \lhd \mathbf{0}]]$ is not affected by the passing of time. To decrease action timers, we use the rules (DPut), (DGet), and (DMove), while other rules are used to compose smaller subnetworks and to apply the structural equivalence.

A derivation $N \overset{\Lambda,t}{\Longrightarrow} N'$, where $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ is a multiset of actions and t is a timeout, denotes a complete computational step:

$$N \xrightarrow{\lambda_1} N_1 \ldots N_{k-1} \xrightarrow{\lambda_k} N_k \overset{t}{\rightsquigarrow} N'.$$

By $N \Longrightarrow^* N'$ we denote an iMAS network N that transforms into a network N' after zero or more action steps followed by a time step.

In our setting, the passing of time is deterministic. The following theorem claims that time passing does not introduce nondeterminism in the evolution of a network.

Theorem 1. *The next statements hold for any three networks N, N', and N'':*

1. *if $N \overset{0}{\rightsquigarrow} N'$, then $N = N'$;*
2. *if $N \overset{t}{\rightsquigarrow} N'$ and $N \overset{t}{\rightsquigarrow} N''$, then $N' = N''$.*

Proof. By induction on the structure of N, as in [6]. □

The following theorem claims that when only time rules can be applied for two time steps of lengths t and t'', then the rules can be applied also for a time step of length $t + t'$. This ensures that the evolution is smooth (without gaps).

Theorem 2. *If $N \overset{t}{\rightsquigarrow} N'' \overset{t'}{\rightsquigarrow} N'$, then $N \overset{t+t'}{\rightsquigarrow} N'$.*

Proof. By induction on the structure of N, as in [6]. □

Example 1. *Let us consider the example mentioned before. For this scenario, we use the following notations for the descriptions of the involved agents: sl (start laser), lv (laser ventilator), stateL (state laser), stateV (state ventilator), and SpO (Saturation of Peripheral Oxygen).*

The entire system can be described as a network:

$$
\begin{aligned}
&\mathit{LaserSurgerySystem} = \mathit{SurgeryRoom}[[\mathit{empty} \lhd \\
&\quad \mathit{Surgeon} \rhd \mathit{empty}\ ||\mathit{Laser} \rhd \langle \mathit{stateL}; \mathit{0}\rangle \\
&\quad ||\,\mathit{Ventilator} \rhd \langle \mathit{stateV}; \mathit{1}\rangle\langle \mathit{SpO}; \mathit{98}\rangle]] \\
&\ |\mathit{LockerRoom}[[\mathit{empty} \lhd \mathit{Zero}]]
\end{aligned}
$$

where:

$$
\begin{aligned}
&\mathit{Surgeon} = sl^{\Delta 1}!\langle \mathit{yes}\rangle \ \mathsf{then}\ \mathit{Surgeon}\ \mathsf{else}\ \mathit{Surgeon} \\
&\mathit{Laser} = sl^{\Delta 1}?(x)\ \mathsf{then}\ \mathit{updS}(\mathit{lv}, \mathit{private}, \mathit{stateL}, \mathit{1}) \\
&\qquad\qquad\qquad \mathsf{then}\ \mathit{updS}(\mathit{lv}, \mathit{private}, \mathit{stateL}, \mathit{0})\ \mathsf{then}\ \mathit{Laser} \\
&\qquad\qquad \mathsf{else}\ \mathit{Surgeon} \\
&\mathit{Ventilator} = \mathit{updS}(\mathit{lv}, \mathit{private}, \mathit{stateV}, \mathit{0})\ \mathsf{then}\ \mathit{Ventilator}' \\
&\mathit{Ventilator}' = \mathsf{if}\ \mathit{get}(\mathit{private}, \mathit{SpO}) > \mathit{95} \\
&\quad \mathsf{then}\ \mathit{updA}(\mathit{private}, \mathit{SpO}, \mathit{get}(\mathit{private}, \mathit{SpO}) - \mathit{1}) \\
&\qquad \mathsf{then}\ \mathit{wait}^{\Delta 1}?(y)\ \mathsf{then}\ \mathit{stop}\ \mathsf{else}\ \mathit{Ventilator}' \\
&\quad \mathsf{else}\ \mathit{updA}(\mathit{private}, \mathit{SpO}, \mathit{98}) \\
&\qquad \mathsf{then}\ \mathit{updS}(\mathit{lv}, \mathit{private}, \mathit{stateV}, \mathit{1})\ \mathsf{then}\ \mathit{Ventilator}
\end{aligned}
$$

3 Implementing Multi-Agent Systems with Information Sharing

We provide an implementation for our language iMAS . Maude is a high-level language and a rewriting platform; it is part of a high-performance system supporting executable specifications in rewriting logic. Rewriting logic [7] is basically a framework which combines term rewriting with equational logic. We use Maude [8] extended with time aspects taken from Real-Time Maude [9].

In order to implement the multi-agent systems of iMAS , we consider sorts corresponding to sets from our language:

```
sorts Location Channel Process GlobalSystem .
```

For the iMAS operators in Table 1, the attached attribute `ctor` marks a constructor, while attribute `prec` followed by a value marks a precedence among operators. Moreover, in real-time Maude we attach the attributes `comm` and `assoc` to mark commutative and associative operators. For example:

```
op < _ ; _ > : Field Nat -> Inf .
op _ |> _ : Process Inf -> Agent [ctor] .
op _||_ : Agent Agent -> Agent [ctor prec 5 comm assoc] .
```

Since some rules of Table 2 have hypotheses, the corresponding rules in Maude are conditional. To identify the corresponding rule of Table 2, we use similar names for each of the below rewrite rules. For example:

```
crl [UpdatePrivate] : k[[I <| ((update(private,f,v)
   then (P)) |> (I' < f ; v' >)) || A]]
=> k[[I <| (P |> (I'< f ; v >)) || A]] if A =/= Zero .
```

As Maude does not support infinite computations, the recursion operator of iMAS is not directly encodable into Maude. Thus, we encode each $id(v)$ into a construction $id(v, b)$, where b is a Boolean value b limiting the unfolding until the first occurrence of *not* b (to transform *not* b into b, an evolution rule should be used again). For example:

```
op Laser : Bool -> Process [ctor] .
ceq Laser(b) =  ((sl ^ 1 ? ( x ))
    then (updateS(lv,private,stateL,1)
        then (updateS(lv,private,stateL,0)
            then Laser(not b)))
    else Laser(not b) ) if b == true .

crl [UnfoldLaser] : k[[I1 <| ((Laser(b)) |> I2 ) || B]]
=> k[[I1 <| ((Laser(not b)) |> I2 ) || B]]
    if b == false /\  B =/= Zero .
crl [UnfoldLaser] : k[[I1 <| ((Laser(b)) |> I2 )]]
=> k[[I1 <| ((Laser(not b)) |> I2 )]]  if b == false .
```

The definitions for *Surgeon* and *Ventilator* are similar.

The rule `[tick]` models the maximum passage of time.

```
crl [tick] : {M} => {delta(M, mte(M))}
    if mte(M) =/= INF and mte(M) =/= 0 .
```

The rule `[tick]` decreases time by means of function `delta`, and the value of the maximal passed time is computed by the function `mte`.

We show that the transition system associated with the rewrite theory in our Maude specification coincides with the reduction semantics for the multi-agent system. Given a system M, we use $\psi(M)$ to denote its representation in Maude. Also, $\mathcal{R}_{\mathcal{D}}$ denotes the rewrite theory mentioned previously in this section by the rewrite rules, and also by the additional operators and equations of these rewrite rules.

The next result relates the structural congruence in our multi-agent language iMAS with the equational equality of the rewrite theory.

Lemma 1. *$M \equiv N$ if and only if $\mathcal{R}_D \vdash \psi(M) = \psi(N)$.*

Proof. $\Rightarrow$: By induction on the congruence rules of our language iMAS .
$\Leftarrow$: By induction on the equations of the rewrite theory $\mathcal{R}_D$. □

The following result emphasizes the operational correspondence between the high-level systems and their translations into a rewrite theory. Generically, by $M \rightarrow N$ is denoted any rule of Table 2.

Theorem 3. *$M \rightarrow N$ if and only if $\mathcal{R}_D \vdash \psi(M) \Rightarrow \psi(N)$.*

Proof. $\Rightarrow$: By induction on the derivation $M \rightarrow N$.
$\Leftarrow$: By induction on the derivation $\mathcal{R}_D \vdash \psi(M) \Rightarrow \psi(N)$. □

4 Strategies and Model-Checking Multi-Agent Systems with Information Sharing

A strategy controls the rewriting steps such that each step obeys the strategy. The result of applying a strategy is the subset of computations produced according to the strategy.

In Maude, a strategy language able to control explicitly the application of rules was presented in [10]. The command for executing a strategy expression `alpha` applied to a term `t` is `srewrite t using alpha`; its output enumerates the solutions that are obtained after this controlled rewriting. Multiple solutions are possible because strategies do not remove the nondeterminism.

The elementary building block of the strategy language is the application of a rule, and the most basic form is the strategy `all` that does not use any restriction when applying the rules. The iteration `(all)*` runs the strategy `all` zero or more times consecutively. For example, the command

```
srew {LaserSurgerySystem} using (all)* .
```

returns a number of 78 solutions. However, this number of solutions can be reduced if we consider a sort of priority on rules. In what follows, besides the iteration strategy α^*, we consider several others: (i) the strategy `idle` that returns the initial term; (ii) the disjunction (or alternative) strategy $\alpha \mid \beta$ that executes α or β, and (iii) the conditional strategy $\alpha?\beta : \gamma$ that executes α and then β on its results; if α does not produce any result, it executes γ on the initial term.

The simplest strategy we consider is to apply the `[Input0]` and `[Output0]` rules only if any other rule is not applicable, and the time rule `[tick]` last. Formally:

```
sd step := ((Comm | IfT | IfF | UnfoldSurgeon
    | UnfoldLaser | UnfoldVentilator | UpdatePrivate
    | UpdatePublic | CreatePrivate | CreatePublic
    | UnfoldScheduleBus | Move)
    ? idle : (Input0 | Output0)) .
sd mtestep := (step ? idle : tick )* .
```

In this case, by running the system *LaserSurgerySystem* using the strategy `mtestep`, the number of solutions is decreasing to 53. This number of solutions can be further reduced by considering the strategy `mtestep1` obtained by removing the number of alternative strategies of the form $\alpha \mid \beta$; this leads to an important decrease in the state space, namely 28 solutions.

Since in iMAS systems, the number of possible applicable rules is high, it turns out to be necessary to use software verification. Various properties of the iMAS systems controlled by using strategies can be analyzed and verified automatically by using the unified Maude model-checking tool umaudemc [11]. In this way, we can check by using CTL*. CTL* [12] is a branching-time temporal logic that extends both LTL [13] and CTL [14]. We can check by formulae of the form: (i) $\mathsf{E}\ \langle\ \rangle\ \phi$ (checks the reachability: there exists a path such that eventually ϕ is satisfied); (ii) $\mathsf{A}\ [\ ]\ \phi$ and $\mathsf{E}\ [\ ]\ \phi$ (checks the safety: something bad will never happen); (iii) $\mathsf{A}\ \langle\ \rangle\ \phi$ and $\phi \leadsto \psi$ (checks the liveness: something good will eventually happen).

The command-line for the umaudemc tool is the following:

```
umaudemc check < file name > < initial term >
    < formula > [ < strategy > ] .
```

To illustrate such a verification, we check some CTL* properties of our running example. Namely, according to [5], the simplified airway laser surgery scenario has two safety properties, i.e., P1: the laser and the ventilator can not be activated at the same time; and P2: the patient's SpO level can not be smaller than 95%. In what follows, we show the outcome of verifying these properties:

```
$ umaudemc check iMASLaserSurgery.maude {LaserSurgerySystem}
    ' A [] ((InfInLocation( < stateL ; 1 > , SurgeryRoom)
    /\ InfInLocation( < stateV ; 0 > , SurgeryRoom))
    \/ (InfInLocation( < stateL ; 0 > , SurgeryRoom)
    /\ InfInLocation( < stateV ; 1 > , SurgeryRoom)))
    ' mtestep2
The property is satisfied in the initial state (56 system
states, 587 rewrites, holds in 56/56 states)

$ umaudemc check iMASLaserSurgery.maude {LaserSurgerySystem}
    ' A [] Saturation(SpO) ' mtestep2
The property is satisfied in the initial state (56 system
states, 475 rewrites, holds in 56/56 states)
```

5 Conclusion

Outperforming the advantages of the agent technology and case studies of multi-agent systems presented in real medical domains [1],[2], in this article we show how to use strategies in multi-agent systems in medical environments in order to reduce their evolution, and how to verify automatically their behaviour by using rewriting platform Maude. Both qualitative aspects (e.g., reachability, safety, liveness) and quantitative aspects (e.g., related to stored information) are presented.

The prototyping language iMAS can be viewed as a member of the TiMo family; this family is generated by a process calculus in which processes can migrate between explicit locations in order to perform local communications with other processes. The initial version of TiMo (presented in [4]) generated various extensions: with access permissions in perTiMo [15], with real-time in rTiMo [16], combining TiMo and the bigraphs [17] to obtain the BigTiMo calculus [18]. Related to the approach presented in this article, we mention [19], in which repuTiMo describes agents with reputation, and [6], in which knowTiMo deals with knowledge of agents represented as sets of trees whose nodes carry information. A related approach is presented in [20], where a Java-based software allows the agents to perform timed migration like in TiMo . In [21], it is given a translation of TiMo into a Real-Time Maude rewriting language.

References

[1] H. Lieberman and C. Mason, “Intelligent agent software for medicine,” *Studies in Health Technology and Informatics*, vol. 80, pp. 99–109, 2002.

[2] D. Isern, D. Sánchez, and A. Moreno, “Agents applied in health care: A review,” *International Journal of Medical Informatics*, vol. 79, no. 3, pp. 145–166, 2010.

[3] M. J. Wooldridge, *An Introduction to MultiAgent Systems, Second Edition.* Wiley, 2009.

[4] G. Ciobanu and M. Koutny, “Timed mobility in process algebra and Petri nets,” *Journal of Logic and Algebraic Programming*, vol. 80, no. 7, pp. 377–391, 2011.

[5] C. Guo, Z. Fu, Z. Zhang, S. Ren, and L. Sha, “Design verifiably correct model patterns to facilitate modeling medical best practice guidelines with statecharts,” *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6276–6284, 2019.

[6] B. Aman and G. Ciobanu, “Knowledge dynamics and behavioural equivalences in multi-agent systems,” *Mathematics*, vol. 9, no. 22, 2021.

[7] J. Meseguer, “Twenty years of rewriting logic,” *Journal of Logic and Algebraic Programming*, vol. 81, no. 7-8, pp. 721–781, 2012.

[8] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. L. Talcott, Eds., *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, ser. Lecture Notes in Computer Science. Springer, 2007, vol. 4350.

[9] P. C. Ölveczky and J. Meseguer, “The Real-Time Maude tool,” in *14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008*, ser. Lecture Notes in Computer Science, C. R. Ramakrishnan and J. Rehof, Eds., vol. 4963. Springer, 2008, pp. 332–336.

[10] F. Durán, S. Eker, S. Escobar, N. Martí-Oliet, J. Meseguer, R. Rubio, and C. L. Talcott, “Programming and symbolic computation in Maude,” *Journal of Logical and Algebraic Methods in Programming*, vol. 110, 2020.

[11] R. Rubio, N. Martí-Oliet, I. Pita, and A. Verdejo, “Model checking strategy-controlled systems in rewriting logic,” *Automated Software Engineering*, vol. 29, no. 1, p. 7, 2022.

[12] E. A. Emerson and J. Y. Halpern, “”Sometimes” and ”Not Never” revisited: on branching versus linear time temporal logic,” *Journal of the ACM*, vol. 33, no. 1, pp. 151–178, 1986.

[13] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science (SFCS 1977)*, 1977, pp. 46–57.

[14] E. M. Clarke and E. A. Emerson, "Design and synthesis of synchronization skeletons using branching-time temporal logic," in *Workshop on Logics of Programs*, ser. Lecture Notes in Computer Science, D. Kozen, Ed., vol. 131. Springer, 1981, pp. 52–71.

[15] G. Ciobanu and M. Koutny, "Timed migration and interaction with access permissions," in *17th International Symposium on Formal Methods, FM 2011*, ser. Lecture Notes in Computer Science, M. J. Butler and W. Schulte, Eds., vol. 6664. Springer, 2011, pp. 293–307.

[16] B. Aman and G. Ciobanu, "Real-time migration properties of rTiMo verified in Uppaal," in *11th International Conference on Software Engineering and Formal Methods, SEFM 2013*, ser. Lecture Notes in Computer Science, R. M. Hierons, M. G. Merayo, and M. Bravetti, Eds., vol. 8137. Springer, 2013, pp. 31–45.

[17] R. Milner, *The Space and Motion of Communicating Agents.* Cambridge University Press, 2009.

[18] W. Xie, H. Zhu, M. Zhang, G. Lu, and Y. Fang, "Formalization and verification of mobile systems calculus using the rewriting engine maude," in *2018 IEEE 42nd Annual Computer Software and Applications Conference, COMPSAC 2018*, S. Reisman, S. I. Ahamed, C. Demartini, T. M. Conte, L. Liu, W. R. Claycomb, M. Nakamura, E. Tovar, S. Cimato, C. Lung, H. Takakura, J. Yang, T. Akiyama, Z. Zhang, and K. Hasan, Eds. IEEE Computer Society, 2018, pp. 213–218.

[19] B. Aman and G. Ciobanu, "Dynamics of reputation in mobile agents systems and weighted timed automata," *Information and Computation*, vol. 282, p. 104653, 2022.

[20] G. Ciobanu and C. Juravle, “Flexible software architecture and language for mobile agents,” *Concurrency and Computation: Practice and Experience*, vol. 24, no. 6, pp. 559–571, 2012.

[21] B. Aman and G. Ciobanu, “Verification of multi-agent systems with timeouts for migration and communication,” in *16th International Colloquium on Theoretical Aspects of Computing, ICTAC 2019*, ser. Lecture Notes in Computer Science, R. M. Hierons and M. Mosbah, Eds., vol. 11884, 2019, pp. 134–151.

Bogdan Aman, Gabriel Ciobanu

Received February 14, 2024
Revised February 16, 2024
Accepted February 21, 2024

Bogdan Aman
ORCID: https://orcid.org/0000-0001-7649-8181
Institute of Computer Science, Romanian Academy, Iasi Branch
Str. Teodor Codrescu 2, 700481, Iaşi, Romania
E–mail: bogdan.aman@iit.academiaromana-is.ro

Gabriel Ciobanu
ORCID: https://orcid.org/0000-0002-8166-9456
Academia Europaea, www.ae-info.org/ae/Member/Ciobanu_Gabriel
E–mail: gabriel.ciobanu@iit.academiaromana-is.ro

Computer Science Journal of Moldova, vol.**32**, no.1(94), 2024

Outer independent total double Italian domination number

Seyed Mahmoud Sheikholeslami, Lutz Volkmann

Abstract

If G is a graph with vertex set $V(G)$, then let $N[u]$ be the closed neighborhood of the vertex $u \in V(G)$. A total double Italian dominating function (TDIDF) on a graph G is a function $f : V(G) \to \{0, 1, 2, 3\}$ satisfying (i) $f(N[u]) \geq 3$ for every vertex $u \in V(G)$ with $f(u) \in \{0, 1\}$ and (ii) the subgraph induced by the vertices with a non-zero label has no isolated vertices. A TDIDF is an outer-independent total double Italian dominating function (OITDIDF) on G if the set of vertices labeled 0 induces an edgeless subgraph. The weight of an OITDIDF is the sum of its function values over all vertices, and the outer independent total double Italian domination number $\gamma^{oi}_{tdI}(G)$ is the minimum weight of an OITDIDF on G. In this paper, we establish various bounds on $\gamma^{oi}_{tdI}(G)$, and we determine this parameter for some special classes of graphs.

Keywords: (Total) double Italian domination. Outer independent (total) double Italian domination.

MSC 2010: 05C69.

1 Introduction

For notation and graph theory terminology, we in general follow Haynes, Hedetniemi and Slater [11]. The starting point of Roman and Italian domination in graphs, as well as all its variants, can be attributed to the mathematical formalization of a defensive model of the Roman Empire described by Stewart in [18]. The formal definition was given by Cockayne et al. [9] as follows. Given a graph $G = (V, E)$,

doi:10.56415/csjm.v32.02

with vertex set $V = V(G)$ and edge set $E = E(G)$, a *Roman dominating function* (RDF) on G is a function $f : V \to \{0, 1, 2\}$, that assigns labels to vertices of G, such that every vertex labeled with 0 must be adjacent to a vertex with a label 2. The sum of all vertex labels, $w(f) = f(V) = \sum_{v \in V} f(v)$, is called the weight of the RDF f and the minimum weight over all possible RDF's is the *Roman domination number*, $\gamma_R(G)$, of the graph G. For the sake of simplicity, an RDF with minimum weight is known as a $\gamma_R(G)$-function or a γ_R-function of G. Clearly, there is a one-on-one relation between Roman dominating functions and the set of subsets $\{V_0^f, V_1^f, V_2^f\}$ of $V(G)$, where $V_i^f = \{v \in V \mid f(v) = i\}$. That is why an RDF f is usually represented as $f = (V_0^f, V_1^f, V_2^f)$ or simply by (V_0, V_1, V_2), if there is no possibility of confusion. An *Italian dominating function* (IDF) on a graph G is defined in [5] as a function $f : V \to \{0, 1, 2\}$ satisfying $f(N(u)) \geq 2$ for each vertex u with $f(u) = 0$. An IDF $f = (V_0, V_1, V_2)$ is an *outer independent total Italian dominating function* (OITIDF) if V_0 is an independent set and $G[V_1 \cup V_2]$ is a subgraph without isolated vertices. The *outer independent total Italian domination number* $\gamma_{tI}^{oi}(G)$ equals the minimum weight of an OITIDF on G, and an OITIDF of G with weight $\gamma_{tI}^{oi}(G)$ is called a $\gamma_{tI}^{oi}(G)$-function. For more details on Roman and Italian domination and its variants, the reader can consult the following book chapters [6], [7] and the survey [8].

In this paper, we only consider simple graphs $G = (V, E)$ with vertex set $V = V(G)$ and edge set $E = E(G)$. The *size* of a graph is its number of edges and its *order* is the number of elements in V. The *open* (resp. *closed*) *neighborhood* $N(v)$ (resp. $N[v]$) of a vertex v is the set $\{u \in V(G) \mid uv \in E(G)\}$ (resp. $N[v] = N(v) \cup \{v\}$). The number of adjacent vertices with v is its *degree*, $\deg(v) = |N(v)|$. We denote by $\delta = \delta(G)$ (resp., $\Delta = \Delta(G)$) the *minimum* (resp., *maximum*) *degree* of a graph G. A *leaf* in a graph is a vertex whose degree is equal to 1 and its neighbor is a *support vertex.* Let $\alpha(G)$ and $\beta(G)$ be the *independence number* and the *covering number* of a graph, respectively. If G is a graph of order n without isolated vertices, then $\alpha(G) + \beta(G) = n$.

We denote by P_n the *path graph* of order n, and by C_n the *cycle graph* of order n. The *corona* of a graph G denoted $G \circ K_1$, is the graph formed from a copy of G by adding for each $v \in V$, a new vertex v'

and the edge vv'.

A function $f : V \rightarrow \{0, 1, 2, 3\}$ is an *outer independent total double Roman dominating function* (OITDRDF) on a graph G if it meets the following requirements:

- Every vertex $v \in V$ with $f(v) = 0$ is adjacent to either a vertex w such that $f(w) = 3$ or to two vertices $w, w' \in V$ with $f(w) = f(w') = 2$.
- Every vertex $v \in V$ with $f(v) = 1$ is adjacent to a vertex $w \in V$ with $f(w) \geq 2$.
- The set of vertices with weight 0 induces an edgeless subgraph and the set of vertices with positive weight induces an isolated-free vertex subgraph.

The *outer independent total double Roman domination number* (*OITDRD-number for short*) $\gamma^{oi}_{tdR}(G)$ equals the minimum weight of an OITDRDF on G, and an OITDRDF of G with weight $\gamma^{oi}_{tdR}(G)$ is called a $\gamma^{oi}_{tdR}(G)$-function. The outer independent total double Roman domination was investigated by Teymourzadeh and Mojdeh [17]; Abdollahzadeh Ahangar, Chellali, Sheikholeslami, and Valenzuela-Tripodoro [2]; and Sheikholeslami and Volkmann [16].

In [12], Mojdeh and Volkmann defined a variant of double Roman domination, namely double Italian domination. A *double Italian dominating function* (DIDF) on a graph G is a function $f : V(G) \rightarrow \{0, 1, 2, 3\}$ satisfying $f(N[u]) \geq 3$ for every vertex $u \in V(G)$ with $f(u) \in \{0, 1\}$. According to Shao, Mojdeh, and Volkmann [15], a DIDF on a graph G with no isolated vertices is a *total double Italian dominating function* (TDIDF) if the subgraph induced by the vertices of the positive label has no isolated vertices. The *total double Italian domination number* $\gamma_{tdI}(G)$ is the minimum weight of a TDIDF on G. A TDIDF on G with weight $\gamma_{tdI}(G)$ is called a $\gamma_{tdI}(G)$*-function.* An *outer independent double Italian dominating function* (OIDIDF) of a graph G is a DIDF for which the vertices with weight 0 are independent. The *outer independent double Italian domination number* $\gamma_{oidI}(G)$ is the minimum weight of an OIDIDF on G (see [1], [3], [4], [19]). An OIDIDF on G with weight $\gamma_{oidI}(G)$ is called a $\gamma_{oidI}(G)$*-function.*

Our aim in this work is to continue the study of a new variation of Italian domination, namely the outer independent total double Italian domination. A TDIDF is an *outer independent total double Italian dominating function* (OITDIDF) on G if the set of vertices with weight 0 induces an edgeless subgraph. The *outer independent total double Italian domination number* (*OITDID-number for short*) $\gamma_{tdI}^{oi}(G)$ equals the minimum weight of an OITDIDF on G, and an OITDIDF of G with weight $\gamma_{tdI}^{oi}(G)$ is called a $\gamma_{tdI}^{oi}(G)$-function.

In this paper, we present basic properties and sharp bounds for the outer independent total double Italian domination number. In addition, we determine this parameter for special classes of graphs.

If G is a graph without isolated vertices, then the definitions lead to $\gamma_{tdI}(G) \leq \gamma_{tdI}^{oi}(G) \leq \gamma_{tdR}^{oi}(G)$.

We make use of the following results in this paper.

Theorem 1. *[10],[13] For a graph G with even order n and no isolated vertices, $\gamma(G) = n/2$ if and only if the components of G are the cycle C_4, or the corona $H \circ K_1$ for any connected graph H.*

Proposition 2. *[15] If C_n is a cycle of length $n \geq 3$, then $\gamma_{tdI}(C_n) = n$. If P_n is a path of order $n \geq 2$, then $\gamma_{tdI}(P_n) = n + 2$ when $n \equiv 1 \pmod 3$ and $\gamma_{tdI}(P_n) = n + 1$ otherwise.*

Proposition 3. *[16] If G is a graph of order n without isolated vertices, then $\gamma_{tdR}^{oi}(G) \leq 2n - \Delta(G)$.*

Proposition 4. *[2] For $n \geq 3$,*

$$(i)\ \gamma_{tdR}^{oi}(P_n) = \begin{cases} 6 & \text{if} \quad n = 4, \\ \lceil \frac{6n}{5} \rceil & \quad \text{otherwise.} \end{cases}$$

$$(ii)\ \gamma_{tdR}^{oi}(C_n) = \lceil \tfrac{6n}{5} \rceil.$$

The next lemma is easy to see, and therefore its proof is omitted.

Lemma 5. *Let G be a graph without isolated vertices. If v is a support vertex and u a leaf neighbor of v, then for any OITDIDF f of G, we have $f(u) + f(v) \geq 3$ and $f(v) \geq 1$.*

2 Special classes of graphs

In this section, we determine the outer independent total double Italian domination number for cycles, paths, and complete t-partite graphs.

Proposition 6. *If C_n is a cycle of length $n \geq 3$, then $\gamma^{oi}_{tdI}(C_n) = n$. If P_n is a path of order $n \geq 2$, then $\gamma^{oi}_{tdI}(P_n) = n + 2$ when $n \equiv 1 \pmod 3$ and $\gamma^{oi}_{tdI}(P_n) = n + 1$ otherwise.*

Proof. Define the function $f : V(C_n) \to \{0, 1, 2, 3\}$ by $f(x) = 1$ for each vertex $x \in V(C_n)$. Clearly, f is an OITDIDF on C_n of weight n, and thus $\gamma^{oi}_{tdI}(C_n) \leq n$. Using Proposition 2 we obtain

$$n = \gamma_{tdI}(C_n) \leq \gamma^{oi}_{tdI}(C_n) \leq n,$$

and thus $\gamma^{oi}_{tdI}(C_n) = n$.

Let now $P_n = v_1 v_2 \dots v_n$. If $n = 3t + 1$ with an integer $t \geq 1$, then define f by $f(v_1) = f(v_n) = 2$ and $f(v_i) = 1$ for $2 \leq i \leq n - 2$. Then f is an OITDIDF on P_n of weight $n + 2$, and thus $\gamma^{oi}_{tdI}(P_n) \leq n + 2$. Using Proposition 2, we obtain

$$n + 2 = \gamma_{tdI}(P_n) \leq \gamma^{oi}_{tdI}(P_n) \leq n + 2,$$

and so $\gamma_{tdI}(P_n) = n + 2$ when $n \equiv 1 \pmod 3$.

Let next $n = 3t$ with an integer $t \geq 1$. Define f by $f(v_{3i}) = 0$ for $1 \leq i \leq t - 1$, $f(v_{3t}) = 1$, $f(v_{3i-1}) = 2$ for $1 \leq i \leq t$, and $f(v_{3i-2}) = 1$ for $1 \leq i \leq t$. Then f is an OITDIDF on P_n of weight $n + 1$, and thus $\gamma^{oi}_{tdI}(P_n) \leq n + 1$. It follows from Proposition 2 that

$$n + 1 = \gamma_{tdI}(P_n) \leq \gamma^{oi}_{tdI}(P_n) \leq n + 1$$

and so $\gamma_{tdI}(P_n) = n + 1$ in this case. Finally, let $n = 3t + 2$ with an integer $t \geq 0$. Define f by $f(v_{3i}) = 0$ for $1 \leq i \leq t$, $f(v_{3i-1}) = 2$ for $1 \leq i \leq t + 1$, and $f(v_{3i-2}) = 1$ for $1 \leq i \leq t + 1$. Then f is an OITDIDF on P_n of weight $n + 1$, and thus $\gamma^{oi}_{tdI}(P_n) \leq n + 1$. Again Proposition 2 leads to the desired result. □

Proposition 4 implies $\gamma^{oi}_{tdR}(C_n) = \lceil \frac{6n}{5} \rceil$ and $\gamma^{oi}_{tdR}(P_n) = \lceil \frac{6n}{5} \rceil$ for $n \geq 5$. Therefore, Proposition 6 shows that the difference $\gamma^{oi}_{tdR}(G) - \gamma^{oi}_{tdI}(G)$ can be arbitrarily large.

Proposition 7. *If $K_{p.q}$ is the complete bipartite graph with $3 \leq p \leq q$, then $\gamma_{tdI}^{oi}(K_{p,q}) = p + 2$.*

Proof. Let X, Y be a bipartition of $K_{p,q}$ with $|X| = p$ and $|Y| = q$, and let $f = (V_0, V_1, V_2, V_3)$ be an OITDIDF on $K_{p,q}$. If $|V_0| = 0$, then $\gamma_{tdI}^{oi}(K_{p,q}) \geq p + q > p + 2$. So let now $|V_0| \geq 1$, and assume, without loss of generality, that $V_0 \subseteq Y$. This implies $f(X) \geq p$ and $f(Y) \geq 1$. If $f(X) \geq p + 1$, then $\gamma_{tdI}^{oi}(K_{p,q}) \geq f(X) + f(Y) \geq p + 2$. In the remaining case that $f(X) = p$, we deduce that $f(x) = 1$ for each $x \in X$, and therefore $f(Y) \geq 2$. Also in this case we obtain $\gamma_{tdI}^{oi}(K_{p,q}) \geq p + 2$.

Conversely, let $w \in Y$. Define the function g by $g(x) = 1$ for $x \in X$, $g(w) = 2$ and $g(y) = 0$ for $y \in Y \setminus \{w\}$. Then g is an OITDIDF on $K_{p,q}$ of weight $p+2$. Hence $\gamma_{tdI}^{oi}(K_{p,q}) \leq p+2$ and so $\gamma_{tdI}^{oi}(K_{p,q}) = p+2$. □

Note the following completion to Proposition 7.

Proposition 8. *If $q \geq 2$, then $\gamma_{tdI}^{oi}(K_{1,q}) = 4$. If $q \geq 3$, then $\gamma_{tdI}^{oi}(K_{2,q}) = 5$.*

Proposition 9. *Let $G = K_{n_1,n_2,\ldots,n_t}$ be the complete t-partite graph with $n_1 \leq n_2 \leq \cdots \leq n_t$, $n = n_1 + n_2 + \cdots + n_t$ and $t \geq 3$. If $n_1 = 1$ and $t = 3$, then $\gamma_{tdI}^{oi}(G) = n + 1 - n_t$ and $\gamma_{tdI}^{oi}(G) = n - n_t$ otherwise.*

Proof. Let $X_1, X_2, \ldots, X_t$ be the partite sets of G with $|X_i| = n_i$ for $1 \leq i \leq t$. If $f = (V_0, V_1, V_2, V_3)$ is an OITDIDF on G, then $|V_0| \leq n_t$, and therefore $\gamma_{tdI}^{oi}(G) \geq n - n_t$. If $n_1 \geq 2$ or $t \geq 4$, then the function g with $g(x) = 0$ for $x \in X_t$ and $g(x) = 1$ for $x \in V(G) \setminus X_t$, is an OITDIDF on G of weight $n - n_t$. Hence $\gamma_{tdI}^{oi}(G) \leq n - n_t$ and so $\gamma_{tdI}^{oi}(G) = n - n_t$ in this case.

Let now $t = 3$ and $n_1 = 1$ with $X_1 = \{w\}$. If $|V_0| = 0$, then $\gamma_{tdI}^{oi}(G) \geq n \geq n + 1 - n_t$. So let now $|V_0| \geq 1$, and assume, without loss of generality, that $V_0 \subseteq X_3 = X_t$. This implies $f(X_1) \geq n_1$ and $f(X_2) \geq n_2$. If $f(X_3) \geq 1$, then we deduce that $\gamma_{tdI}^{oi}(G) \geq f(X_1) + f(X_2) + f(X_3) \geq n_1 + n_2 + 1 = n + 1 - n_3 = n + 1 - n_t$. Let now $f(X_3) = 0$. If $f(X_2) \geq n_2+1$, then $\gamma_{tdI}^{oi}(G) \geq f(X_1)+f(X_2)+f(X_3) \geq n_1 + n_2 + 1 = n + 1 - n_t$. In the remaining case that $f(X_2) = n_2$, we deduce that $f(x) = 1$ for each $x \in X_2$, and therefore $f(w) \geq 2$. Also in this case we obtain $\gamma_{tdI}^{oi}(G) \geq n+1-n_t$. Conversely, define the function g by $g(x) = 1$ for $x \in X_2$, $g(w) = 2$ and $g(x) = 0$ for $x \in X_3$. Then g is an OITDIDF on G of weight $n+1-n_t$. Hence $\gamma_{tdI}^{oi}(G) \leq n+1-n_t$ and so $\gamma_{tdI}^{oi}(G) = n+1-n_t$. □

3 Bounds on $\gamma_{tdI}^{oi}(G)$

In this section, we establish upper and lower bounds for the outer independent total double Italian domination number of graphs. We start with a simple result.

Proposition 10. *Let G be a graph of order n. If $\delta(G) \geq 2$, then $\gamma_{tdI}^{oi}(G) \leq n$.*

Proof. If $\delta(G) \geq 2$, then define the function $f : V(G) \to \{0, 1, 2, 3\}$ by $f(x) = 1$ for each vertex $x \in V(G)$. Clearly, f is an OITDIDF on G of weight n, and thus $\gamma_{tdI}^{oi}(G) \leq n$. □

For connected graphs G with minimum degree at least two and $\Delta(G) = n(G) - 1$, we can improve the bound of Proposition 10 slightly.

Proposition 11. *Let G be a connected graph of order $n \geq 4$ with $\delta(G) \geq 2$ and $\Delta(G) = n-1$. Then $\gamma_{tdI}^{oi}(G) \leq n-1$. This bound is sharp for complete graphs.*

Proof. Let $v \in V(G)$ be a vertex of degree $n-1$. If G is complete, then $\gamma_{tdI}^{oi}(G) = n-1$ by Proposition 9. If G is not complete, then two neighbors, say w and z of v are independent. Since $n \geq 4$ and $\delta(G) \geq 2$, we observe that $(\{w, z\}, V(G) \setminus \{v, w, z\}, \{v\}, \emptyset)$ is an OITDIDF on G of weight $n-1$. Thus $\gamma_{tdR}^{oi}(G) \leq n-1$, and the proof is complete. □

Proposition 12. *If G is a graph of order n without isolated vertices, then $\gamma_{tdI}^{oi}(G) \leq \gamma(G) + n \leq n + \lfloor \frac{n}{2} \rfloor$.*

Proof. Given a minimum dominating set D of G, the function f defined by $f(x) = 2$ if $x \in D$ and $f(x) = 1$ otherwise, is an OITDIDF on G, implying that $\gamma_{tdI}^{oi}(G) \leq |D| + n = \gamma(G) + n$. Using Ore's result $\gamma(G) \leq \lfloor \frac{n}{2} \rfloor$ for graphs without isolated vertices, we obtain $\gamma_{tdI}^{oi}(G) \leq n + \lfloor \frac{n}{2} \rfloor$. □

The next result is an immediate consequence of Theorem 1 and Propositions 6 and 12.

Corollary 13. *If G is a graph of order n without isolated vertices, then $\gamma^{oi}_{tdI}(G) \leq \frac{3}{2}n$ with equality if and only if the components of G are the corona $H \circ K_1$ for any connected graph H.*

Next we focus on graphs with minimum degree at least three. A set of vertices $P \subseteq V(G)$ is a 2-*packing* of G if the distance in G between any pair of distinct vertices from P is larger than two. The maximum cardinality of a 2-packing of G is the *packing number* of G and is denoted by $\rho(G)$.

Theorem 14. *If G is a graph of order n with $\delta(G) \geq 3$, then $\gamma^{oi}_{tdR}(G) \leq n - \rho(G)$. Moreover, this bound is sharp.*

Proof. Suppose that $A = \{v_1, v_2, \ldots, v_\rho\}$ is a 2-packing of G. Define the function f by $f(v_i) = 0$ for $1 \leq i \leq \rho$ and $f(x) = 1$ for $x \in V(G)\backslash A$. Since $\delta(G) \geq 3$, f is an OITDIDF on G, and thus $\gamma^{oi}_{tdR}(G) \leq n - \rho(G)$.

Now let $H_1, H_2, \ldots, H_t$ be isomorphic to the complete graph K_p with $p \geq 4$, and let $a_i, b_i \in V(H_i)$ for $1 \leq i \leq t$. Define H as $H_1 \cup H_2 \cup \ldots \cup H_t$ together with the edges $b_i a_{i+1}$ for $1 \leq i \leq t-1$. If $g = (V_0, V_1, V_2, V_3)$ is an OITDIDF on H, then we observe that $|V_0 \cap V(H_i)| \leq 1$ for $1 \leq i \leq t$, and therefore $|V_0| \leq t$. It follows that

$$\omega(g) = |V_1| + 2|V_2| + 3|V_3| \geq |V_1| + |V_2| + |V_3| = n(H) - |V_0| \geq n(H) - t.$$

Since $\rho(H) = t$, the bound above implies $\gamma^{oi}_{tdR}(H) \leq n(H) - \rho(H)$. Hence we obtain $\gamma^{oi}_{tdR}(H) = n(H) - t = n(H) - \rho(H)$. □

Using Proposition 3, we get the next result.

Theorem 15. *Let G be a connected graph of order $n \geq 2$. Then $\gamma^{oi}_{tdI}(G) \leq 2n - \Delta(G)$ with equality if and only if $G = F \circ K_1$, where F is a connected graph with maximum degree $\Delta(F) = n(F) - 1$.*

Proof. Proposition 3 implies that $\gamma^{oi}_{tdI}(G) \leq 2n - \Delta(G)$.

If $G = F \circ K_1$, where F is a connected graph with maximum degree $\Delta(F) = n(F) - 1$, then $\Delta(G) = n(F)$ and $\gamma^{oi}_{tdI}(G) = 3n(F) = n(G) + n(F) = 2n(G) - n(F) = 2n(G) - \Delta(G)$.

Conversely assume that $\gamma_{tdI}^{oi}(G) = 2n - \Delta(G)$. Proposition 10 yields $\delta(G) = 1$. If $\Delta(G) = 2$, then G is a path and applying Proposition 6, we obtain $G \in \{P_2, P_4\}$ and so G satisfied the condition. Assume that $\Delta(G) \geq 3$. Let v be a vertex of maximum degree $\Delta(G)$ and let $N(v) = \{u_1, u_2, \ldots, u_t\}$. Assume that $X = V(G) - N[v]$. If $X = \emptyset$, then suppose, without loss of generality, that $\deg(u_1) = 1$ and define the function f on G with $f(v) = 3$, $f(u_1) = f(u_2) = 0$ and $f(x) = 1$ otherwise. Clearly, f is an OITDIDF of G of weight $n < 2n - \Delta(G)$, which is a contradiction. Thus $X \neq \emptyset$. Let $X = \{w_1, w_2, \ldots, w_s\}$. If there is an edge $w_iw_j \in E(G)$, then the function f defined on G by $f(w_i) = 1$, $f(u_1) = \cdots = f(u_t) = 1$ and $f(x) = 2$ otherwise, is an OITDIDF of G of weight $2n - \Delta(G) - 1 < 2n - \Delta(G)$, which is a contradiction. Thus X is an independent set. If some w_i has two neighbors in $N(v)$, then the function f defined on G by $f(w_i) = 1$, $f(u_1) = \cdots = f(u_t) = 1$ and $f(x) = 2$ otherwise, is an OITDIDF of G of weight $2n - \Delta(G) - 1 < 2n - \Delta(G)$, which is a contradiction. Therefore, each vertex in X has exactly one neighbor in $N(v)$ because G is connected. Hence $\deg(w_i) = 1$ for each $1 \leq i \leq s$. If some u_i has two neighbors in X, say w_1, w_2, then the function f defined on G by $f(u_i) = 3$, $f(w_1) = f(w_2) = 0$, $f(u_1) = \cdots = f(u_t) = 1$, and $f(x) = 2$ otherwise, is an OITDIDF of G of weight less than $2n - \Delta(G)$, which is a contradiction. Thus, each u_i is adjacent to at most one vertex in X. Assume, without loss of generality, that $w_iu_i \in E(G)$ for each $1 \leq i \leq s$. If $s = t$, then the function f defined on G by $f(v) = 1$, $f(u_1) = \cdots = f(u_t) = 3$, and $f(x) = 0$ otherwise, is an OITDIDF of G of weight less than $2n - \Delta(G)$, a contradiction. Hence $t > s$. If $\deg(u_i) \geq 2$ for some $i \in \{s+1, \ldots, t\}$, say $i = t$, then the function f defined on G by $f(v) = 2$, $f(u_1) = \cdots = f(u_s) = 3$, $f(u_i) = 1$ for $s + 1 \leq i \leq t - 1$, and $f(x) = 0$ otherwise, is an OITDIDF of G of weight less than $2n - \Delta(G)$, a contradiction. Thus $\deg(u_i) = 1$ for each $i \in \{s+1, \ldots, t\}$. If $t - s \geq 2$, then the function f defined on G by $f(v) = 3$, $f(u_1) = \cdots = f(u_s) = 3$, and $f(x) = 0$ otherwise, is an OITDIDF of G of weight less than $2n - \Delta(G)$, a contradiction, yielding $t = s + 1$. Thus, $G = F \circ K_1$, where $F = G - \{w_1, w_2, \ldots, w_s, u_t\}$ and that $\Delta(F) = n(F) - 1$. This completes the proof. □

Next we present an upper bound on $\gamma_{tdI}^{oi}(G)$ in terms of $\gamma_{oidI}(G)$.

Theorem 16. *If G is a graph without isolated vertices, then*

$$\gamma_{tdI}^{oi}(G) \leq \left\lfloor \frac{1}{2}(3\gamma_{oidI}(G)-1)\right\rfloor .$$

The bound is sharp for the stars $K_{1,p}$ for $p \geq 2$, and the complete bipartite graphs $K_{2,q}$ for $q \geq 3$.

Proof. Let $f=(V_0,V_1,V_2,V_3)$ be a $\gamma_{oidI}(G)$-function, and assume that $V_1\cup V_2\cup V_3=\{v_1,v_2,\ldots,v_t\}$. If $V_0=\emptyset$, then $\gamma_{tdI}^{oi}(G)=\gamma_{oidI}(G)\leq \left\lfloor \frac{1}{2}(3\gamma_{oidI}(G)-1)\right\rfloor$, as desired. Hence assume that $V_0\neq\emptyset$. Now let $H_1,H_2,\ldots,H_r$ be the connected components of the subgraph $G[V_1\cup V_2\cup V_3]$, and let $w\in V_0$. Assume, without loss of generality, that $N(w)\cap V(H_i)\neq\emptyset$ for each i with $1\leq i\leq s$. We observe that $s\leq r\leq t$, $\sum_{i=s+1}^{r}\sum_{x\in V(H_i)} f(x)\geq 2(r-s)$ and $\sum_{i=1}^{s}\sum_{x\in V(H_i)} f(x)\geq 3$. Therefore, it follows that

$$\gamma_{oidI}(G)=\sum_{i=1}^{t} f(v_i)=\sum_{i=1}^{s}\sum_{x\in V(H_i)} f(x)+\sum_{i=s+1}^{r}\sum_{x\in V(H_i)} f(x)\geq 3+2(r-s),$$

and thus

$$(r-s)\leq \frac{1}{2}(\gamma_{oidI}(G)-3).$$

Now let, without loss of generality, $H_{s+1},H_{s+2},\ldots,H_k$ $(k\leq r)$ be exactly the components of order one. Since G is a graph without isolated vertices, we can choose a vertex $w_i\in V_0$ for each $s+1\leq i\leq k$ such that w_i has a neighbor in H_i. Then the function g defined by $g(w)=g(w_i)=1$ for each $s+1\leq i\leq k$ and $g(x)=f(x)$ otherwise is an OITDIDF on G, and thus

$$\gamma_{tdI}^{oi}(G)\leq \omega(g)\leq \omega(f)+(k-s)+1\leq \omega(f)+(r-s)+1\leq \frac{1}{2}(3\gamma_{oidI}(G)-1).$$

This leads to the desired bound.

Proposition 8 implies $\gamma_{tdI}^{oi}(K_{1,p})=4$ and $\gamma_{tdI}^{oi}(K_{2,q})=5$. Since $\gamma_{oidI}(K_{1,q})=3$ and $\gamma_{oidI}(K_{2,q})=4$, we deduce that

$$\gamma_{tdI}^{oi}(K_{1,q})=4=\left\lfloor \frac{1}{2}(3\gamma_{oidI}(K_{1,q})-1)\right\rfloor$$

and

$$\gamma_{tdI}^{oi}(K_{2,q}) = 5 = \left\lfloor \frac{1}{2}(3\gamma_{oidI}(K_{2,q}) - 1) \right\rfloor .$$

□

In what follows we establish some lower bounds on $\gamma_{tdR}^{oi}(G)$. The proof of the next result is similar to the proof of Theorem 3.1 in [1].

Theorem 17. *Let G be a graph of order n with minimum degree $\delta \geq 1$ and maximum degree Δ. Then*

$$\gamma_{tdR}^{oi}(G) \geq \left\lfloor \frac{\delta n}{\Delta + \delta - 1} \right\rfloor + 1,$$

and this bound is sharp.

Proof. Let $f = (V_0, V_1, V_2, V_3)$ be a $\gamma_{tdI}^{oi}(G)$-function. Since V_0 is an independent set, every vertex of V_0 has at least δ neighbors in $V_1 \cup V_2 \cup V_3$. In addition, every vertex of $V_1 \cup V_2 \cup V_3$ has at most $\Delta - 1$ neighbors in V_0. Therefore, it follows that

$$\delta(n - |V_1| - |V_2| - |V_3|) = \delta|V_0| \leq (\Delta - 1)(|V_1| + |V_2| + |V_3|),$$

and thus

$$\frac{\delta n}{\Delta + \delta - 1} \leq |V_1| + |V_2| + |V_3| = \gamma_{tdI}^{oi}(G) - |V_2| - 2|V_3|.$$

If $V_2 \cup V_3 \neq \emptyset$, then the last inequality chain leads to the desired bound. If $V_2 \cup V_3 = \emptyset$, then each vertex of V_1 is adjacent to at least two vertices of V_1, and we obtain analogously

$$\gamma_{tdI}^{oi}(G) \geq \frac{\delta n}{\Delta + \delta - 2} > \frac{\delta n}{\Delta + \delta - 1}.$$

Since $\gamma_{tdI}^{oi}(G)$ is an integer, we deduce the desired bound also in this case.

For each integer $p \geq 3$, let H_{3p} be the graph obtained from a cycle C_p by adding $2p$ new vertices and joining each new vertex to all vertices of C_p. Then we observe that $n(H_{3p}) = 3p$, $\Delta(H_{3p}) = 2p+2$, $\delta(H_{3p}) = p$ and $\gamma_{tdI}^{oi}(H_{3p}) = p = \left\lfloor \frac{3p^2}{3p+1} \right\rfloor + 1$. □

Theorem 18. *If G is a connected graph of order $n \geq 2$, then $\gamma_{tdI}^{oi}(G) \geq \beta(G)$. Furthermore, this bound is sharp for the complete t-partite graph $G = K_{n_1,n_2,\ldots,n_t}$ with $n_1 \leq n_2 \leq \ldots \leq n_t$ and $t \geq 4$, or $t = 3$ and $n_1 \geq 2$.*

Proof. Let $f = (V_0, V_1, V_2, V_3)$ be an OITDIDF on G. Then $|V_0| \leq \alpha(G)$, and therefore

$$\beta(G) = n - \alpha(G) \leq n - |V_0| = |V_1| + |V_2| + |V_3|.$$

Thus

$$\gamma_{tdI}^{oi}(G) = |V_1| + 2|V_2| + 3|V_3| \geq |V_1| + |V_2| + |V_3| \geq \beta(G),$$

as desired.

Observation 9 shows that

$$\gamma_{tdI}^{oi}(K_{n_1,n_2,\ldots,n_t}) = n - n_t = \beta(K_{n_1,n_2,\ldots,n_t})$$

if $n_1 \leq n_2 \leq \ldots \leq n_t$ and $t \geq 4$, or $t = 3$ and $n_1 \geq 2$. □

Theorem 19. *If G is a connected graph of order $n \geq 2$, then $\gamma_{tdI}^{oi}(G) \leq 2\gamma_{tI}^{oi}(G) - 1$. The bound is sharp for any graph G with $\gamma_{tI}^{oi}(G) = 2$.*

Proof. If $f = (V_0, V_1, V_2)$ is a $\gamma_{tI}^{oi}(G)$-function, then $\gamma_{tI}^{oi}(G) = |V_1| + 2|V_2|$. If $V_1 = \emptyset$, then $|V_2| \geq 2$ and the function $(V_0, \emptyset, \emptyset, V_2)$ is an OITDIDF on G of weight $3|V_2|$, and so

$$\gamma_{tdI}^{oi}(G) \leq 3|V_2| \leq 4|V_2| - 2 = 2\gamma_{tI}^{oi}(G) - 2.$$

Assume that $V_1 \neq \emptyset$ and let $w \in V_1$. Now the function $(V_0, \{w\}, V_1 \setminus \{w\}, V_2)$ is an OITDIDF on G. If $V_2 \neq \emptyset$, then

$$\gamma_{tdI}^{oi}(G) \leq 1 + 2(|V_1| - 1) + 3|V_2| < 2|V_1| + 4|V_2| - 1 = 2\gamma_{tI}^{oi}(G) - 2.$$

Suppose that $V_2 = \emptyset$. Then we have

$$\gamma_{tdI}^{oi}(G) \leq 1 + 2(|V_1| - 1) \leq 2|V_1| - 1 = 2\gamma_{tI}^{oi}(G) - 1.$$

□

4 Nordhaus-Gaddum type inequalities

In this section, we present Nordhaus-Gaddum type inequalities for the outer independent total double Italian domination number. Let $\mathcal{G}$ be a family of graphs G such that G is obtained from a complete graph K_p, $(p \geq 4)$, an empty graph $\overline{K_s}$, where $s \geq \left\lceil \frac{3p}{p-3} \right\rceil$ and a new vertex u, by joining u to every vertex of K_p and joining each vertex of $\overline{K_s}$ to at least three vertices of K_p such that each vertex of K_p is non-adjacent to at least three vertices of $\overline{K_s}$. It is clear from the construction of G that $G \in \mathcal{G}$ if and only if $\overline{G} \in \mathcal{G}$. The proof of the following result can be found in [3].

Theorem 20. *If G and $\overline{G}$ are connected graphs of order $n \geq 4$, then*

$$\gamma_{oidI}(G) + \gamma_{oidI}(\overline{G}) \geq n - 1,$$

with equality if and only if $G \in \mathcal{G}$.

Since for any graph G without isolated vertices, $\gamma^{oi}_{tdI}(G) \geq \gamma_{oidI}(G)$, we get the following result.

Theorem 21. *If G and $\overline{G}$ are connected graphs of order $n \geq 4$, then*

$$\gamma^{oi}_{tdI}(G) + \gamma^{oi}_{tdI}(\overline{G}) \geq n - 1,$$

with equality if and only if $G \in \mathcal{G}$.

Theorem 22. *Let G and $\overline{G}$ be graphs without isolated vertices of order n. If $\delta(G) = \delta(\overline{G}) = 1$, then*

$$\gamma^{oi}_{tdI}(G) + \gamma^{oi}_{tdI}(\overline{G}) \leq 2n + 4.$$

The equality holds if and only if $G = P_4$.

If $\delta(G) \geq 2$ and $\delta(\overline{G}) \geq 2$, then

$$\gamma^{oi}_{tdI}(G) + \gamma^{oi}_{tdI}(\overline{G}) \leq 2n.$$

This bound is sharp for C_5.

If $\delta(G) \geq 2$ or $\delta(\overline{G}) \geq 2$, then

$$\gamma^{oi}_{tdI}(G) + \gamma^{oi}_{tdI}(\overline{G}) \leq 2n + \left\lfloor \frac{n}{2} \right\rfloor.$$

This bound is sharp for C_4.

Proof. If $\delta(G)=\delta(\overline{G})=1$, then it follows from Theorem 15 that

$$\begin{aligned}\gamma_{tdI}^{oi}(G)+\gamma_{tdI}^{oi}(\overline{G}) &\leq (2n-\Delta(G))+(2n-\Delta(\overline{G}))\\ &= (2n-(n-2))+(2n-(n-2))=2n+4.\end{aligned}$$

If $G=P_4$, then $\overline{G}=P_4$, and Proposition 6 implies that $\gamma_{tdI}^{oi}(G)+\gamma_{tdI}^{oi}(\overline{G})=6+6=2n+4$. Conversely, let $\gamma_{tdI}^{oi}(G)+\gamma_{tdI}^{oi}(\overline{G})=2n+4$. Then we deduce from the above inequality chain that $\gamma_{tdI}^{oi}(G)=2n-\Delta(G)$ and $\gamma_{tdI}^{oi}(\overline{G})=2n+\Delta(\overline{G})$. Theorem 15 implies that $G=F\circ K_1$ and $\overline{G}=F'\circ K_1$, where F and F' are connected graphs with $\Delta(F)=n(F)-1$ and $\Delta(F')=n(F')-1$. If $\Delta(F)\geq 2$, then we have $\delta(\overline{G})\geq 2$ which is a contradiction. Hence $\Delta(F)=1$, and so $F=K_2$. Thus $G=P_4$.

Proposition 10 implies $\gamma_{tdI}^{oi}(G)+\gamma_{tdI}^{oi}(\overline{G})\leq 2n$ immediately when $\delta(G)\geq 2$ and $\delta(\overline{G})\geq 2$. According to Proposition 6, this bound is sharp for C_5.

Finally assume, without loss of generality, that $\delta(\overline{G})\geq 2$. We deduce from Propositions 12 and 10 that

$$\gamma_{tdI}^{oi}(G)+\gamma_{tdI}^{oi}(\overline{G})\leq n+\left\lfloor\frac{n}{2}\right\rfloor+n=2n+\left\lfloor\frac{n}{2}\right\rfloor.$$

If $\overline{G}=C_4$, then $G=2P_2$, and we observe that $\gamma_{tdI}^{oi}(G)+\gamma_{tdI}^{oi}(\overline{G})=4+6=2n+\left\lfloor\frac{n}{2}\right\rfloor$. □

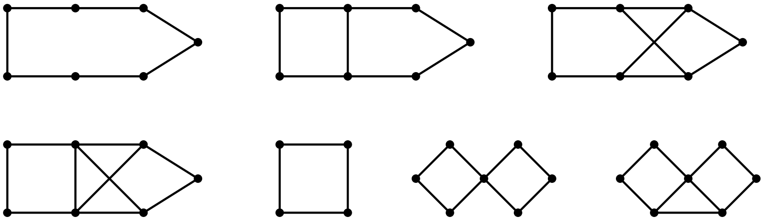


Figure 1. Graphs in family $\mathcal{A}$

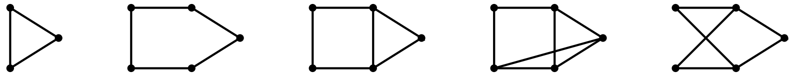


Figure 2. Graphs in family $\mathcal{B}$

Using Theorem 14, we improve Theorem 22 for $n \geq 12$. We recall the definition of some families of graphs. Let $\mathcal{A}$ be the collections of graphs in Figure 1 and $\mathcal{B}$ be the collections of graphs in Figure 2. Let

$$\mathcal{G}_1 = \{C_4\} \cup \{G \mid G = H \circ K_1, \text{ where } H \text{ is connected}\}$$

and

$$\mathcal{G}_2 = \mathcal{A} \cup \mathcal{B} - \{C_4\}.$$

For any graph H, let $\mathcal{S}(H)$ denote the set of connected graphs, each of which can be formed from $H \circ K_1$ by adding a new vertex x and edges joining x to one or more vertices of H and define

$$\mathcal{G}_3 = \cup_H \mathcal{S}(H),$$

where the union is taken over all graphs H. Let y be a vertex of a copy of C_4 and for $G \in \mathcal{G}_3$, let $\theta(G)$ be the graph obtained by joining G to C_4 with the single edge xy, where x is the new vertex added in forming G. Define

$$\mathcal{G}_4 = \{\theta(G) \mid G \in \mathcal{G}_3\}.$$

Next, let uvw be a path P_3. For any graph H, let $\mathcal{P}(H)$ be the set of connected graphs which may be formed from $H \circ K_1$ by joining each of u and w to one or more vertices of H. Then define

$$\mathcal{G}_5 = \bigcup_H \mathcal{P}(H).$$

Let H be a graph $X \in \mathcal{B}$. Let $\mathcal{R}(H, X)$ be the set of connected graphs which may be obtained from $H \circ K_1$ by joining each vertex of $U \subseteq V(X)$ to one or more vertices of H such that no set with fewer than $\gamma(X)$ vertices of X dominates $V(X) - U$. Then define

$$\mathcal{G}_6 = \bigcup_{H,X} \mathcal{R}(H, X).$$

The proof of the following result can be found in [14], [20].

Theorem 23. *A connected graph G satisfies $\gamma(G) = \lfloor \frac{n(G)}{2} \rfloor$ if and only if $G \in \cup_{i=1}^{6} \mathcal{G}_i$.*

Lemma 24. *If $G \in \cup_{i=1}^{6} \mathcal{G}_i$ and $n(G) \geq 12$, then $\gamma_{tdI}^{oi}(\overline{G}) \leq n - 2$.*

Proof. Let $G \in \cup_{i=1}^{6} \mathcal{G}_i$. Since $n(G) \geq 12$, we have $G \notin \mathcal{G}_2$. Let $G \in \mathcal{G}_1$ and let $G = H \circ K_1$. We deduce from $n(G) \geq 12$ that $n(H) \geq 6$. If $z \in V(H)$ and z' is a leaf adjacent to z in G, then the function f defined on $V(\overline{G})$ by $f(z) = f(z') = 0$ and $f(s) = 1$ otherwise, is an OITDIDF on $\overline{G}$ of weight $n - 2$, and so $\gamma_{tdI}^{oi}(\overline{G}) \leq n - 2$, the required bound.

If $G \in \mathcal{G}_3 \cup \mathcal{G}_4 \cup \mathcal{G}_5 \cup \mathcal{G}_6$, then G has at least four leaves and the function f defined above leads to $\gamma_{tdI}^{oi}(\overline{G}) \leq n - 2$. □

Theorem 25. *If G and $\overline{G}$ are graphs without isolated vertices of order $n \geq 12$, then*

$$\gamma_{tdI}^{oi}(G) + \gamma_{tdI}^{oi}(\overline{G}) \leq 2n + \left\lfloor \frac{n}{2} \right\rfloor - 2.$$

Proof. If $\delta(G) = \delta(\overline{G}) = 1$ or $\delta(G) \geq 2$ and $\delta(\overline{G}) \geq 2$, then the desired result follows from Theorem 22. Let now, without loss of generality, $\delta(G) = 1$ and $\delta(\overline{G}) \geq 2$. If $\delta(\overline{G}) = 2$, then $\Delta(G) = n - 3$, and thus we deduce from Theorem 15 and the hypothesis $n \geq 12$ that

$$\begin{aligned}
\gamma_{tdI}^{oi}(G) + \gamma_{tdI}^{oi}(\overline{G}) &\leq (2n - \Delta(G)) + (2n - \Delta(\overline{G}) - 1) \\
&= (2n - (n - 3)) + (2n - (n - 2) - 1) \\
&= 2n + 4 \leq 2n + \left\lfloor \frac{n}{2} \right\rfloor - 2.
\end{aligned}$$

Assume that $\delta(\overline{G}) \geq 3$. If $G \in \cup_{i=1}^{6} \mathcal{G}_i$, then by Lemma 24 and Proposition 12, we obtain

$$\gamma_{tdI}^{oi}(G) + \gamma_{tdI}^{oi}(\overline{G}) \leq n + \left\lfloor \frac{n}{2} \right\rfloor + n - 2 = 2n + \left\lfloor \frac{n}{2} \right\rfloor - 2.$$

If $G \notin \cup_{i=1}^{6} \mathcal{G}_i$, then using Theorems 23, 14 and Proposition 12, we obtain

$$\gamma_{tdI}^{oi}(G) + \gamma_{tdI}^{oi}(\overline{G}) \leq n + \left\lfloor \frac{n}{2} \right\rfloor - 1 + (n - 1) = 2n + \left\lfloor \frac{n}{2} \right\rfloor - 2.$$

□

References

[1] N. A. Abd Aziz, H. Kamrulhaili, F. Azviv, and N. Jafari Rad, “On the outer-independent double Italian domination number,” *Electron. J. Graph Theory Appl.*, vol. 10, pp. 365–374, 2022.

[2] H. Abdollahzadeh Ahangar, M. Chellali, S. M. Sheikholeslami, and J. C. Valenzuela-Tripodoro, “On the outer independent total double Roman domination in graphs,” *Mediterr. J. Math.*, vol. 20, Article ID 171, 2023.

[3] F. Azvin, N. Jafari Rad, and L. Volkmann, “Bounds on the outer-independent double Italian domination number,” *Commun. Comb. Optim.*, vol. 6, pp. 123–136, 2021.

[4] M. Benatallah, “Outer-independent double Italian domination in graphs,” Manuscript, 2021.

[5] M. Chellali, T. W. Haynes, S.T. Hedetniemi, and A. MacRae, “Roman {2}-domination,” *Discrete Appl. Math.*, vol. 204, pp. 22–28, 2016.

[6] M. Chellali, N. Jafari Rad, S. M. Sheikholeslami, and L. Volkmann, “Roman domination in graphs,” in *Topics in Domination in Graphs*, T. W. Haynes, S. T. Hedetniemi, and M. A. Henning, Eds. Springer, 2020, pp. 365–409.

[7] M. Chellali, N. Jafari Rad, S. M. Sheikholeslami, and L. Volkmann, “Varieties of Roman domination,” in *Structures of Domination in Graphs*, T. W. Haynes, S. T. Hedetniemi, and M. A. Henning, Eds. Springer, 2021, pp. 273–307.

[8] M. Chellali, N. Jafari Rad, S. M. Sheikholeslami, and L. Volkmann, “Varieties of Roman domination II,” *AKCE J. Graphs Combin.*, vol. 17, pp. 966–984, 2020.

[9] E. J. Cockayne, P. A. Dreyer, S. M. Hedetniemi, and S. T. Hedetniemi, “Roman domination in graphs,” *Discrete Math.*, vol. 278, pp. 11–22, 2004.

[10] J. F. Fink, M. S. Jacobson, L. F. Kinch, and J. Roberts, "On graphs having domination number half their order," *Period. Math. Hungar.*, vol. 16, pp. 287–293, 1985.

[11] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*, New York: Marcel Dekker, Inc., 1998.

[12] D. A. Mojdeh and L. Volkmann, "Roman {3}-domination (double Italian domination)," *Discrete Appl. Math.*, vol. 283, pp. 555–564, 2020.

[13] C. Payan and N. H. Xuong, "Domination-balanced graphs," *J. Graph Theory*, vol. 6, pp. 23–32, 1982.

[14] B. Randerath and L. Volkmann, "Characterization of graphs with equal domination and covering number," *Discrete Math.*, vol. 191, pp. 159–169, 1998.

[15] Z. Shao, D.A. Mojdeh, and L. Volkmann, "Total Roman {3}-domination in graphs," *Symmetry*, vol. 12, Article No. 268, 15 p., 2020.

[16] S. M. Sheikholeslami and L. Volkmann, "New bounds on the outer independent total double Roman domination number," *Discrete Math. Algorithms Appl.*, to be published.

[17] A. Teymourzadeh and D. A. Mojdeh, "Covering total double Roman domination in graphs," *Commun. Comb. Optim.*, vol. 8, pp. 115–125, 2023.

[18] I. Stewart, "Defend the Roman Empire!," *Sci. Amer.*, vol. 281, no. 6, pp. 136–139, 1999.

[19] L. Volkmann, "Restrained double Italian domination in graphs," *Commun. Comb. Optim.*, vol. 8, pp. 1–11, 2023.

[20] B. Xu, E. J. Cockayne, T. W. Haynes, Stephen T. Hedetniemi, and Z. Shangchao, "Extremal graphs for inequalities involving domination parameters," *Discrete Math.*, vol. 216, pp. 1–10, 2000.

Outer independent total double Italian domination number

Seyed Mahmoud Sheikholeslami, Lutz Volkmann

Received May 17, 2023
Accepted June 30, 2023

Seyed Mahmoud Sheikholeslami
ORCID: https://orcid.org/0000-0003-2298-4744
Department of Mathematics
Azarbaijan Shahid Madani University
Tabriz, I. R. Iran
E–mail: s.m.sheikholeslami@azaruniv.ac.ir

Lutz Volkmann
ORCID: https://orcid.org/0000-0003-3496-277X
Lehrstuhl C für Mathematik
RWTH Aachen University
52062 Aachen, Germany
E–mail: volkm@math2.rwth-aachen.de

On the trees with maximum Cardinality-Redundance number

Elham Mohammadi, Nader Jafari Rad

Abstract

A vertex v is said to be over-dominated by a set S if $|N[u] \cap S| \geq 2$. The cardinality–redundance of S, $CR(S)$, is the number of vertices of G that are over-dominated by S. The cardinality–redundance of G, $CR(G)$, is the minimum of $CR(S)$ taken over all dominating sets S. A dominating set S with $CR(S) = CR(G)$ is called a $CR(G)$-set. In this paper, we prove an upper bound for the cardinality–redundance in trees in terms of the order and the number of leaves, and characterize all trees achieving equality for the proposed bound.

Keywords: Dominating set, Cardinality-Redundance, trees.
MSC 2020: 05C69.

1 Introduction

We consider here undirected and simple graphs $G = (V, E)$ with vertex set $V(G)$ and edge set $E(G)$. The *order* of G is given by $n = n(G) = |V(G)|$. The *open neighborhood* $N(v)$ of a vertex v is the set of vertices that are adjacent to v, and the *close neighborhood* $N[v]$ is $N(v) \cup v$. For any subset $S \subseteq V(G)$, denote $N(A) = \cup_{v \in A} N(v)$ and $N[A] = \cup_{v \in A} N[v]$. The *degree* of v is the cardinality of $N(v)$, denoted by $\deg(v)$. A vertex v is said to be a *leaf* if $\deg(v) = 1$. A vertex is a *support* vertex if it is adjacent to a leaf. We denote by $L(G)$ and $S(G)$ the collections of all leaves and support vertices of G, respectively. We also denote by $L(v)$ the leaves adjacent to v. A *star* is the graph $K_{1,k}$, where $k \geq 1$. Further if $k > 1$, the vertex of degree k is called the *center* vertex of the star, while if $k = 1$, arbitrarily designate either

doi:10.56415/csjm.v32.03

vertex of P_2 as the center. A *double star* is a tree with precisely two vertices of degree at least two, namely the centers of the double star. We denote by $S(a, b)$ a double star in which the centers have degrees a and b. We call a double star *strong* if at least one of its centers has degree at least three. A *bipartite graph* is a graph G that the vertex set can be partitioned into two sets X and Y such that any edge of G has one end-point in X and the other end-point in Y. The *diameter*, $diam(G)$, of a graph G is the maximum distance among all pairs of vertices in G. A *diametrical path* in G is a shortest path whose length is equal to the diameter of G. A *rooted tree* T distinguishes one vertex r called the *root*. For each vertex $v \neq r$ of T, the parent of v is the neighbor of v on the unique (r, v)-path, while a child of v is any other neighbor of v. If T is a rooted tree, then for any vertex v, we denote by T_v the sub-rooted tree rooted at v.

A set $S \subseteq V$ of vertices in a graph is called a *dominating set*, if every vertex $v \in V$ is either an element of S or is adjacent to an element of S. The *domination number* $\gamma(G)$ of a graph G is the minimum cardinality of a dominating set among all dominating sets of G. For fundamentals of domination theory in graphs, we refer the reader to the so-called domination books by Haynes, Hedetniemi, and Slater [2], [3].

Johnson and Slater [4] introduced the concept of cardinality–redundance in graphs. A vertex v is said to be *over-dominated* by a set S if $|N[u] \cap S| \geq 2$. The *cardinality–redundance* of S, $CR(S)$, is the number of vertices of G that are over-dominated by S. The cardinality–redundance of G, $CR(G)$, is the minimum of $CR(S)$ taken over all dominating sets S. A dominating set S with $CR(S) = CR(G)$ is called a $CR(G)$-set. The concept of cardinality–redundance was further studied in, for example, [6], in which the authors presented several extremal Problems Related to the Cardinality–redundance on graphs G with $CR(G) = 0, 1, 2$.

In this paper, we prove an upper bound for the cardinality–redundance in trees in terms of the order and the number of leaves, and characterize all trees achieving equality for the proposed bound.

2 Main result

In this section, we prove an upper bound for the cardinality-redundance of a tree in terms of the order and the number of leaves, and characterize trees achieving equality of the given bound. For this purpose, we first introduce the following family of trees $\mathcal{T}$. Let $\mathcal{T}$ be the family of all trees T that can be obtained from a sequence $T_0, T_1, ..., T_{k-1}, T_k$, where T_0 is a strong double star, and if $k \geq 1$, then T_i is obtained from T_{i-1} by the following operation for each $i = 1, 2, ..., k-1$:

Operation $\mathcal{O}$: Add a strong double star and join one of its centers to a support vertex of T_{i-1}.

To prove our main result, we need a series of lemmas.

Lemma 1. *If $T' \in \mathcal{T}$ and T is obtained from T' by Operation $\mathcal{O}$, then $CR(T) = CR(T') + 1$.*

Proof. Let $T' \in \mathcal{T}$ and T be obtained from T' by adding a $S(a,b)$ with $\deg(b) \geq 3$, and joining a to a support vertex c of T'. If S' is a $CR(T')$-set, then $S = S' \cup \{b\} \cup L(a)$ is a dominating set for T, and $CR(S) = CR(S') + 1$, since a is over-dominated by S. Thus, $CR(T) \leq CR(S) = CR(S') + 1$. Now let S be a $CR(T)$-set. Since S is a dominating set for T, we find that $|S \cap V(S(a,b))| \geq 2$. Assume that $c \in S$. If $a \notin S$, then $L(a) \subseteq S$ and a is over-dominated by S. Then $S' = S - V(S(a,b))$ is a dominating set for T' with $CR(S') \leq CR(S) - 1$, and thus, $CR(T') \leq CR(T) - 1$. Thus, assume that $a \in S$. Then both a and b are over-dominated by S, since $S \cap (\{b\} \cup L(b)) \neq \emptyset$. Now $S' = S - V(S(a,b))$ is a dominating set for T' with $CR(S') \leq CR(S) - 2$, and thus, $CR(T') \leq CR(T) - 2$.

We next assume that $c \notin S$. Then $L(c) \subseteq S$. If $a \notin S$, then $L(a) \subseteq S$ and at least one vertex of $S(a,b)$ is over-dominated by S, since $S(a,b)$ is a strong double star. Then $S' = S - V(S(a,b))$ is a dominating set for T' with $CR(S') \leq CR(S) - 1$, and thus, $CR(T') \leq CR(T) - 1$. Thus, assume that $a \in S$. Then b is over-dominated by S, since $S \cap (\{b\} \cup L(b)) \neq \emptyset$. Then $S' = S - V(S(a,b))$ is a dominating set for T' with $CR(S') \leq CR(S) - 1$, and thus, $CR(T') \leq CR(T) - 1$. We conclude that $CR(T) = CR(T') + 1$. □

The following is an immediate consequence of the definition of the family of $\mathcal{T}$.

Lemma 2. *If $T \in \mathcal{T}$ has n vertices and ℓ leaves, then:*
(1) $V(T) = L(T) \cup S(T)$.
(2) $CR(T) = \frac{n-\ell}{2}$.

Proof. (1) is obvious.
(2) Let X and Y be the partite sets of T. Without lost of generally let

$$|S(T) \cap X| \leq |S(T) \cap Y|.$$

Clearly, Y is a dominating set for T and

$$CR(Y) = |S(T) \cap X|.$$

Thus,

$$CR(G) \leq CR(Y) = |S(T) \cap X| \leq \frac{|S(T)|}{2} \leq \frac{n-\ell}{2}.$$

We next prove that $CR(T) \geq \frac{n-\ell}{2}$. Note that T is obtained from a sequence $T_0, T_1, ..., T_{k-1}, T_k$, where T_0 is a strong double star, and if $k \geq 1$, then T_i is obtained from T_{i-1} by the Operation $\mathcal{O}$, for each $i = 1, 2, ..., k-1$. We use an induction on k (the number of times that the Operation $\mathcal{O}$ is performed to construct T). For the base step $k = 0$, it is clear that $CR(T) = 1 = \frac{n-\ell}{2}$. Assume the result is true for any tree $T' \in \mathcal{T}$ arisen by applying $k' < k$ operations. Now consider the tree T, and let $T' = T_{k-1}$. Assume that T is obtained from T' by adding a strong double star $S(a, b)$ and joining a to a support vertex c of T'. By the inductive hypothesis, $CR(T') \geq \frac{n'-\ell'}{2}$, where $n' = n(T')$ and $\ell' = \ell(T')$. By Lemma 1, $CR(T) = CR(T') + 1$. Then,

$$\begin{aligned} CR(T) &= CR(T') + 1 \\ &\geq \frac{n'-\ell'}{2} + 1 \\ &= \frac{n - (\deg(a) - 1) - \deg(b) - (\ell - (\deg(a) - 2) - (\deg(b) - 1))}{2} \\ &+ \ 1 = \frac{n-\ell}{2}, \end{aligned}$$

as desired. □

Lemma 3. *If $T \in \mathcal{T}$ has n vertices and ℓ leaves, and S is a $CR(T)$-set, then:*
(1) S contains precisely half of members of $S(T)$.
(2) For each vertex $x \in S(T)$, if $x \in S$, then $L(x) \cap S = \emptyset$ and if $x \notin S$, then $L(x) \subseteq S$.
(3) If X and Y are partite sets of T, then $CR(X) = CR(Y) = \frac{n-\ell}{2}$.

Proof. (1) and (2) are obvious.
(3). We prove this by an induction on the number of times that the Operation $\mathcal{O}$ is performed to construct T. The result is obvious if T is a strong double star. Assume the result holds if T is obtained by applying Operation $\mathcal{O}$, $k' < k$ times, and now T is obtained from a sequence $T_0, T_1, ..., T_{k-1}, T_k$, where T_0 is a strong double star and T_i is obtained from T_{i-1} by the Operation $\mathcal{O}$, for each $i = 1, 2, ..., k-1$. Let X_{k-1} and Y_{k-1} be partite sets of T_{k-1}, and let T is obtained by adding the center a of a strong double star $S(a, b)$ to a support vertex c of T_{k-1}, and without loss of generality, assume that $c \in X$. By Lemma 1, $CR(T) = CR(T_{k-1}) + 1$. By the inductive hypothesis, $CR(X_{k-1}) = CR(Y_{k-1}) = \frac{n(T_{k-1})-\ell(T_{k-1})}{2}$. Let $X_k = X_{k-1} \cup \{b\} \cup L(a)$, and $Y_k = Y_{k-1} \cup \{a\} \cup L(b)$. Then $X_k = X_{k-1} \cup \{b\} \cup L(a)$ is a dominating set for T with $CR(X_k) = CR(X_{k-1}) + 1 = \frac{n(T_{k-1}-\ell(T_{k-1})}{2} + 1 = \frac{n-\ell}{2}$, since a is over-dominated by X_k. Similarly, from $CR(Y_{k-1}) = \frac{n(T_{k-1})-\ell(T_{k-1})}{2}$ we obtain that $Y_k = Y_{k-1} \cup \{a\} \cup L(b)$ is a dominating set for T with

$$CR(Y_k) = CR(Y_{k-1}) + 1 = \frac{n(T_{k-1}) - \ell(T_{k-1})}{2} + 1 = \frac{n-\ell}{2},$$

since b is over-dominated by Y_k and c is a support vertex of T_{k-1}. □

The following is a direct consequence of Lemma 3, Part (3).

Corollary 1. *If $T \in \mathcal{T}$ has n vertices and ℓ leaves, then for any vertex x, there is a $CR(T)$-set S with $CR(S) = \frac{n-\ell}{2}$ and $x \notin S$.*

We are now ready to present the main result of this paper.

Theorem 1. *If T is a tree of order $n \geq 3$ with $\ell = \ell(T)$ leaves, then $CR(T) \leq \frac{n-\ell}{2}$, with equality if and only if $n = 2$ or $T \in \mathcal{T}$.*

Proof. The result is obvious for $n = 2$; thus, assume that $n \geq 3$. We prove by induction on n that $CR(T) \leq (n - \ell)/2$, and if equality holds, then $T \in \mathcal{T}$. We root T at a leaf x_0 of a diametrical path $P_0 : x_0, x_1, ..., x_d$, where d is the diameter of T. For the base step of the induction, we assume that $d = 1$. Then T is a star, and it is evident that $CR(T) = 0 < \frac{n-\ell}{2}$. If $d = 3$, then T is a double star $S(a, b)$. If $\deg(a) = \deg(b) = 2$, then the leaves of T form a dominating set implying that $CR(T) = 0 < \frac{n-\ell}{2}$. Thus, assume that $\deg(b) \geq 3$. Then $CR(T) = 1 = \frac{n-\ell}{2}$. We thus assume that $d \geq 4$.

Assume that $\deg(x_{d-2}) = 2$. Let $T' = T - T_{x_{d-2}}$. By the inductive hypothesis,

$$CR(T') \leq \frac{n' - \ell'}{2} = \frac{n - (\deg(x_{d-1}) + 1) - \ell'}{2}.$$

Observe that $\ell - (\deg(x_{d-1}) - 1) \leq \ell' \leq \ell - (\deg(x_{d-1}) - 1) + 1$. Thus,

$$\begin{aligned} CR(T') &\leq \frac{n' - \ell'}{2} \leq \frac{n - (\deg(x_{d-1}) + 1) - (\ell - (\deg(x_{d-1}) - 1))}{2} \\ &= \frac{n - \ell - 2}{2}. \end{aligned}$$

If $CR(T') < \frac{n'-\ell'}{2}$ and S' is a $CR(T')$-set, then $S = S' \cup \{x_{d-1}\}$ is a dominating set for T with $CR(S) \leq CR(S') + 1$. Then $CR(T) \leq CR(S) < \frac{n'-\ell'}{2} + 1 = \frac{n-\ell-2}{2} + 1 = \frac{n-\ell}{2}$. Thus, assume that $CR(T') = \frac{n'-\ell'}{2}$. By the inductive hypothesis, $T' \in \mathcal{T}$. By Corollary 1, there is a $CR(T')$-set S' with $CR(S') = \frac{n'-\ell'}{2}$ and $x_{d-3} \notin S'$. Then $S = S' \cup \{x_{d-1}\}$ is a dominating set for T with $CR(S) = CR(S')$. Thus,

$$CR(T) \leq CR(S) = CR(S') = \frac{n' - \ell'}{2} = \frac{n - \ell - 2}{2} < \frac{n - \ell}{2}.$$

We thus assume that $\deg(x_{d-2}) \geq 3$. Note that x_{d-1} is a child of x_{d-2} which is a support vertex. Suppose that x_{d-2} has at least two children which are support vertices. Let x_{d-1}, z_1,...,z_k be the children of x_{d-2} which are support vertices, where $k \geq 1$. Let $T' = T - T_{x_{d-2}}$.

By the inductive hypothesis,

$$\begin{aligned} CR(T') &\leq \frac{n'-\ell'}{2} \\ &= \frac{n-\deg(x_{d-1})-\sum_{i=1}^{k}\deg(z_i)-1-|L(x_{d-2})|-\ell'}{2}. \end{aligned}$$

Observe that $\ell' \geq \ell-(\deg(x_{d-1})-1)+\sum_{i=1}^{k}(\deg(z_i)-1)-|L(x_{d-2})|$. Thus, since $k \geq 1$, we obtain that

$$CR(T') \leq \frac{n'-\ell'}{2} \leq \frac{n-\ell-2-k}{2} \leq \frac{n-\ell-3}{2}.$$

Let S' be a $CR(T')$-set. Then $S=S'\cup L(x_{d-2})\cup\{x_{d-1},z_1,...,z_k\}$ is a dominating set for T with $CR(S)=CR(S')+1$. Thus,

$$CR(T) \leq CR(S)=CR(S')+1 \leq \frac{n-\ell-3}{2}+1 < \frac{n-\ell}{2}.$$

We thus assume that x_{d-1} is the only child of x_{d-2} which is a support vertex. Then $T_{x_{d-2}}$ is a double star. Let $T'=T-T_{x_{d-2}}$. By the inductive hypothesis,

$$CR(T') \leq \frac{n'-\ell'}{2} = \frac{n-(\deg(x_{d-2}-2)-(\deg(x_{d-1}+1)-\ell'}{2}.$$

Observe that $\ell' \geq \ell-\deg(x_{d-1})-\deg(x_{d-2})+3$. Thus,

$$\begin{aligned} CR(T') &\leq \frac{n'-\ell'}{2} \leq \frac{n-(\deg(x_{d-2}-2)-(\deg(x_{d-1}+1)-\ell'}{2} \\ &\leq \frac{n-\ell-2}{2}. \end{aligned}$$

Let S' be a $CR(T')$-set. Then $S=S'\cup L(x_{d-2})\cup\{x_{d-1}\}$ is a dominating set for T with $CR(S)=CR(S')+1$. Thus,

$$CR(T) \leq CR(S)=CR(S')+1 \leq \frac{n-\ell-2}{2}+1 \leq \frac{n-\ell}{2}. \qquad (1)$$

We thus proved that $CR(T) \leq \frac{n-\ell}{2}$. Assume the equality holds. Following the above proof, we find that if $d=3$, then T is a strong double star that belongs to $\mathcal{T}$, or equality in (1) holds. If $CR(T') < \frac{n'-\ell'}{2}$, then considering $S = S' \cup L(x_{d-2}) \cup \{x_{d-1}\}$, we find that $CR(T) \leq CR(S) = CR(S') + 1 < \frac{n-\ell-2}{2} + 1 \leq \frac{n-\ell}{2}$, a contradiction. Thus, $CR(T') = \frac{n'-\ell'}{2}$. By the inductive hypothesis, $T' \in \mathcal{T}$. If $\deg(x_{d-3}) = 2$, then $\ell' = \ell - (\deg(x_{d-1}) - 1) - (\deg(x_{d-2}) - 2) + 1$, and so $CR(T') \leq \frac{n-\ell-3}{2}$, and we obtain that $CR(T) < \frac{n-\ell}{2}$, a contradiction. Thus, $\deg(x_{d-3}) \geq 3$, that is, x_{d-3} is a support vertex of T'. Thus, T is obtained from T' by Operation $\mathcal{O}$. Consequently, $T \in \mathcal{T}$.

The converse follows by Lemma 2, Part (2). □

References

[1] E. DeLaViña, R. Pepper, and W. Waller, "Lower bounds for the domination number," *Discuss. Math. Graph Theory*, vol. 30, pp. 475–487, 2010.

[2] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*, New York: Marcel Dekker, In c., 1998.

[3] *Domination in Graphs: Advanced Topics*, T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, Eds. New York: Marcel Dekker, 1998.

[4] T.W. Johnson and P.J. Slater, "Maximum independent, minimally c-redundant sets in graphs," *Congr. Numer.*, vol. 74, pp. 193–211, 1990.

[5] M. Lemańska, "Lower bound on the domination number of a tree," *Discuss. Math. Graph Theory*, vol. 24, pp. 165–170, 2004.

[6] D. McGinnis and N. Shank, "Extremal Problems Related to the Cardinality-Redundance of Graphs," *Australas. J. Combin*, vol. 87, pp. 214–238, 2023.

Elham Mohammadi[1], Nader Jafari Rad[2]

Received October 01, 2023
Accepted November 16, 2023

[1,2]Department of Mathematics
Shahed University Tehran, Iran

[1]Elham Mohammadi
E–mails: elhammohammadi495@gmail.com, elhammohammadi495@shahed.ac.ir

[2]Nader Jafari Rad
ORCID: https://orcid.org/0000-0001-7406-1859
E–mail: n.jafarirad@gmail.com

Computer Science Journal of Moldova, vol.**32**, no.1(94), 2024

Vector finite fields of characteristic two as algebraic support of multivariate cryptography

Alexandr Moldovyan, Nikolay Moldovyan

Abstract

The central issue of the development of the multivariate public key algorithms is the design of reversible non-linear mappings of n-dimensional vectors over a finite field, which can be represented in a form of a set of power polynomials. For the first time, finite fields $GF\left((2^d)^m\right)$ of characteristic two, represented in the form of m-dimensional finite algebras over the fields $GF(2^d)$ are introduced for implementing the said mappings as exponentiation operation. This technique allows one to eliminate the use of masking linear mappings, usually used in the known approaches to the design of multivariate cryptography algorithms and causing the sufficiently large size of the public key. The issues of using the fields $GF\left((2^d)^m\right)$ as algebraic support of non-linear mappings are considered, including selection of appropriate values of m and d. In the proposed approach to development of the multivariate cryptography algorithms, a superposition of two non-linear mappings is used to define resultant hard-to-reverse mapping with a secret trap door. The used two non-linear mappings provide mutual masking of the corresponding reverse maps, due to which the size of the public key significantly reduces as compared with the known algorithms-analogues at a given security level.

Keywords: finite fields, finite algebras, non-linear mapping, system of power equations, post-quantum cryptography, multivariate cryptography.

MSC 2010: 68P25, 68Q12, 68R99, 94A60, 16Z05, 14G50

doi:10.56415/csjm.v32.04

1 Introduction

Multivariate public-key cryptography (MPC) is one of the attractive directions of post-quantum cryptography [1]. It exploits the computational difficulty of solving systems of many power equations with many unknowns. Quantum computers are not efficient for solving the said problem, therefore, the MPC cryptalgorithms are secure against quantum attacks [1], [2]. The MPC algorithms have sufficiently high performance and a small size of digital signature and are promising for practical applications in the coming post-quantum era [3], [4]. However, the known MPC algorithms have a significant drawback for practical application, which is the very large size of the public key.

The present paper considers a novel concept of the design of MPC algorithms, which consists in the use of two non-linear mappings specified in the form of exponentiation operations in finite fields $GF\left((2^d)^m\right)$ defined in the form of finite m-dimensional algebras [5]. The main meaning of the used vector form [5] for specifying finite fields $GF\left((2^d)^m\right)$ is that the result of exponentiation operations can be effectively obtained as a calculation of the values of m polynomials over the field $GF\left(2^d\right)$. Selecting appropriate values of d and m allows a significant reduction of the public key size at a given security level.

2 Preliminaries

In the MPC algorithms, the public key is usually specified as a hard to reverse non-linear mapping Π of an n-dimensional vectors over a finite field $\mathbb{F}_q$ into a u-dimentional vectors over $\mathbb{F}_q$ $(u \geq n)$ [1], [3], the said non-linear mapping being set in the form of a set of u power polynomials (usually of degree two) in n variables and having a secret trapdoor. Using the latter, the owner of the public key can perform decryption of ciphertexts and generate digital signatures.

The development of an MPC algorithm is connected with specifying a set of u secret power polynomials $f_j\left(x_1, x_2, \ldots, x_n\right)$ over $\mathbb{F}_q$, where $j = 1, \ldots u$, which define a reversible nonlinear mapping $\Psi : \mathbb{F}_q^n \to \mathbb{F}_q^u$. Then, using two linear maps $\Lambda_1 : \mathbb{F}_q^n \to \mathbb{F}_q^n$ and $\Lambda_2 : \mathbb{F}_q^u \to \mathbb{F}_q^u$ (for example, implemented as multiplication of the n-dimensional and u-

dimensional vectors by $n \times n$ and $u \times u$ secret matrices, correspondingly), calculate the set of u polynomials $p_j(x_1, x_2, \ldots, x_n)$ over $\mathbb{F}_q$, where $j = 1, \ldots u$, which define the mapping:

$$\mathit{\Pi} = \mathit{\Lambda}_2 \circ \mathit{\Psi} \circ \mathit{\Lambda}_1. \tag{1}$$

From a known set of polynomials $\mathit{\Pi}$, it is easy to find the image $Z = \mathit{\Pi}(X)$ of some vector X, but it is computationally difficult to calculate the vector-preimage V for a given random vector R. However, the creator (owner) of the public key $\mathit{\Pi}$ can effectively calculate the vector V:

$$V = \mathit{\Lambda}_1^{-1}\left(\mathit{\Psi}^{-1}\left(\mathit{\Lambda}_2^{-1}(R)\right)\right) = \mathit{\Pi}^{-1}(R). \tag{2}$$

The public key represents the superposition (1) of secret mappings. However, the mapping $\mathit{\Pi}$ is given as a set of power polynomials $p_j(x_1, x_2, \ldots, x_n)$, and the public encryption of a message M, represented in the form of n-dimensional vector, is performed as calculation of u coordinates of the u-dimensional vector $C = \mathit{\Pi}(M)$:

$$c_1 = p_1(x_1, x_2, \ldots, x_n); \quad c_2 = p_2(x_1, x_2, \ldots, x_n); \quad \ldots \quad .$$
$$c_u = p_u(x_1, x_2, \ldots, x_n).$$

If a ciphertext $C = (c_1, c_2, \ldots, c_u)$ is given, then a potential adversary can find the source message, solving the system of u power equations with n unknowns $x_1, x_2, \ldots, x_n$, which is defined by the latter formulas. Such attacks on the MPC algorithms are called direct. The best known direct attacks are based on using so-called algorithms F4 [6] and F5 [7].

The owner of the public key $\mathit{\Pi}$ can decipher the ciphertext C as follows: $M = \mathit{\Lambda}_1^{-1} \circ \mathit{\Psi}^{-1} \circ \mathit{\Lambda}_2^{-1}(C)$.

A digital signature can be calculated in the form of n-dimensional vector S as follows:

1. Calculate the hash value from a message M to be signed and represent it in the form of u-dimensional vector H.

2. Find preimage S of the vector H: $S = \mathit{\Lambda}_1^{-1} \circ \mathit{\Psi}^{-1} \circ \mathit{\Lambda}_2^{-1}(H)$.

The signature verification algorithm includes the next two steps:

1. Compute the image H' of the signature S: $H' = \mathit{\Pi}(S)$.

2. Calculate the hash value from the message M and represent it as an u-dimensional vector H. If $H = H'$, then the signature S is genune, else the signature is false.

The role of linear mappings Λ_1 and Λ_2 is to mask a secret trapdoor that allows you to invert the mapping Π. The present paper consideres a new technique for designing the mapping Π, characterized in using two reversible nonlinear mappings Ψ_1^{-1} and Ψ_2^{-1} ensuring the rejection of the use of masking linear mappings. The said two mappings are set using exponentiation operations in the vector finite fields [5], namely, in the fields specified in the form of finite algebras over $\mathbb{F}_q$, where $q = 2^d$; $d \geq 5$.

Suppose an m-dimensional vector space is set over a finite field $GF(q)$, where q is a prime number or a power of a prime number. If a multiplication operation that is distributive at the left and at the right relatively addition operation is defined additionally, then we have m-dimensional finite algebra. We will use the following two notations of the vector A: $A = (a_1, a_2, \ldots a_m) = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + \ldots, a_m\mathbf{e}_m$, where $\mathbf{e}_1, \mathbf{e}_2 + \ldots, \mathbf{e}_m$ are basis vectors. The multiplication of two vectors A and $B = (b_1, b_2, \ldots b_m)$ is defined as follows:

$$AB = \sum_{i,j=1}^{m} a_i b_j \left(\mathbf{e}_i \mathbf{e}_j\right),$$

where every product $\mathbf{e}_i\mathbf{e}_j$ is to be substituted by a one-component vector $\mu\mathbf{e}_k$ ($\mu \neq 1$ is called structural constant) indicated in the cell at the intersection of the ith row and jth column of so-called basis vector multilication table (BVMT). In [5], it had been shown, if $m \geq 2$ divides the value $q - 1$, then it is possible to specify a BVMT such that the algebra is the finite field $GF(q^m)$.

Table 1, where $\pi = \mu\epsilon\tau^{-1}$, presents a general form of BVMT with three different structural constants μ, ϵ, and τ, which was introduced for specifying the vector finite fields of arbitrary dimension $m \geq 2$ [5]. For a given value of m, there are various types of BVMTs by which vector fields can be specified. Every of these BVMTs can include from one to m different structural constants with their different distributions across the cells of the table. The vector fields are set by selecting suitable values of the structural constants.

Table 1. A general form of BVMT for defining the vector fields $GF(q^m)$ [5] $(m \geq 2)$.

$\cdot$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\cdots$	$\mathbf{e}_{m-1}$	$\mathbf{e}_m$
$\mathbf{e}_1$	$\tau\mathbf{e}_1$	$\tau\mathbf{e}_2$	$\tau\mathbf{e}_3$	$\tau\mathbf{e}_4$	$\tau\cdots$	$\tau\mathbf{e}_{m-1}$	$\tau\mathbf{e}_m$
$\mathbf{e}_2$	$\tau\mathbf{e}_2$	$\epsilon\mathbf{e}_3$	$\epsilon\mathbf{e}_4$	$\epsilon\cdots$	$\epsilon\mathbf{e}_{m-1}$	$\epsilon\mathbf{e}_m$	$\pi\mathbf{e}_1$
$\mathbf{e}_3$	$\tau\mathbf{e}_3$	$\epsilon\mathbf{e}_4$	$\epsilon\cdots$	$\epsilon\mathbf{e}_{m-1}$	$\epsilon\mathbf{e}_m$	$\pi\mathbf{e}_1$	$\mu\mathbf{e}_2$
$\mathbf{e}_4$	$\tau\mathbf{e}_4$	$\epsilon\cdots$	$\epsilon\mathbf{e}_{m-1}$	$\epsilon\mathbf{e}_m$	$\pi\mathbf{e}_1$	$\mu\mathbf{e}_2$	$\mu\mathbf{e}_3$
$\cdots$	$\tau\cdots$	$\epsilon\mathbf{e}_{m-1}$	$\epsilon\mathbf{e}_m$	$\pi\mathbf{e}_1$	$\mu\mathbf{e}_2$	$\mu\mathbf{e}_3$	$\mu\ldots$
$\mathbf{e}_{m-1}$	$\tau\mathbf{e}_{m-1}$	$\epsilon\mathbf{e}_m$	$\pi\mathbf{e}_1$	$\mu\mathbf{e}_2$	$\mu\mathbf{e}_3$	$\mu\cdots$	$\mu\mathbf{e}_{m-2}$
$\mathbf{e}_m$	$\tau\mathbf{e}_m$	$\pi\mathbf{e}_1$	$\mu\mathbf{e}_2$	$\mu\mathbf{e}_3$	$\mu\cdots$	$\mu\mathbf{e}_{m-2}$	$\mu\mathbf{e}_{m-1}$

3 Specifying the vector finite fields $GF\left(\left(2^d\right)^m\right)$

In the introduced method for the development of the MPC algorithms, we use the vector finite fields set over the fields $GF\left(2^d\right)$, elements of which are the binary polynomials of the degree less or equal to $d-1$. The multiplication operation in $GF\left(2^d\right)$ is specified as multiplication of binary polynomials modulo a low-weight binary irreducible polynomial (in order to reduce the computational complexity of the multiplication in $GF\left(2^d\right)$). The values of d define the values of m for which the m-dimensional algebra is a vector field $GF\left(\left(2^d\right)^m\right)$, since for the latter, it is required to fulfill the condition $m|2^d-1$. Suitable values of d and m are shown in Table 2, where the cases $d = 8,\ 16,\ 24$ are of preferable interest from the practical point of view.

Consider the case of using the vector fields set over $GF\left(2^8\right)$ relating to the development of an MPC algorithm with a public key $\mathit{\Pi} = \mathit{\Psi}_2 \circ \mathit{\Psi}_1$ defining the mapping $\mathbb{F}_{256}^{85} \to \mathbb{F}_{256}^{85}$. The input 85-dimensional vector X is represented as concatenation of 17 vectors of dimension 5 ($j = 1, 2, \ldots, 17$):

$$X = (X_1, X_2, \ldots X_{17})\,, \text{ where } X_j = (x_1^{(j)}, x_2^{(j)}, \ldots x_5^{(j)}).$$

The mapping $\mathit{\Psi}_1$ is specified as exponentiation of every vector X_j to the power 257 in a unique vector field $GF\left(\left(2^8\right)^5\right)$. Since the integer 257

Table 2. Suitable values of d and m.

d	2^d-1	m	d	2^d-1	m
5	31	31	18	$3^3\cdot 7\cdot 19\cdot 73$	$7;9;19;$ $27;73$
6	$3^2\cdot 7$	$3;7;9;21$	20	$17\cdot 61681$	17
8	$3\cdot 5\cdot 17$	$3;5;15;17$	21	$7^2\cdot 127\cdot 337$	$7;49;127$
9	$7\cdot 73$	$7;73$	22	$3\cdot 23\cdot 89\cdot 683$	$3;23;89$
14	$3\cdot 43\cdot 127$	$3;43;127$	23	$47\cdot 178481$	47
15	$7\cdot 31\cdot 151$	$7;31$	24	$3^2\cdot 5\cdot 7\cdot 13\cdot$ $\cdot 17\cdot 241$	$5;7;9;$ $13;15;17$
16	$3\cdot 5\cdot 17\cdot 257$	$3;5;15;17$	...	...	...

is mutaully prime with the integer $2^{40}-1$ (order of the multiplicative group of $GF\left(\left(2^8\right)^5\right)$), the latter operation defines bijective nonlinear mapping $Y_j=\varPsi_{1(j)}(X_j)$. The inverse mapping $X_j=\varPsi^{-1}_{1(j)}(Y_j)$ can be performed as exponentiation to the power $b=551894941568$, since $b\equiv 257^{-1} \bmod 2^{40}-1$.

To specify the 17 unique mappings $\varPsi_{1(j)}$ $(j=1,2,\ldots,17)$, we define two different types of the vector fields $GF\left(\left(2^8\right)^5\right)$ using two different BVMTs with 5 structural constants ϵ, λ, μ, σ, and τ, shown in Tables 3 and 4. The values of the said constants are generated at random and independently for each of the mappings $\varPsi_{1(j)}$.

The fact that a field is formed for a given set of values of structural constants is ensured by checking experimentally the existence of a field element whose order is equal to $2^{40}-1$. If there is no such element, then another set of random values of structural constants is generated and the specified check is repeated until the presence of a generator of a cyclic group of the order $2^{40}-1$ is established. The latter fact will mean the reversibility of every non-zero 5-dimensional vector, i.e., it will mean the formation of a vector field $GF\left(\left(2^8\right)^5\right)$.

For the values $j=1,2,3,5,6,8,9,10,12,13,15,16$, and 17, the mapping $\varPsi_{1(j)}$ is defined as the exponentiation operation (represented in the form of a set of polynomials over $GF(2^8)$) in the vector finite field $GF\left(\left(2^8\right)^5\right)$ set by the BVMT of the first kind represented by Table 3. For the values $j = 4, 7, 11$, and 14, the mapping $\Psi_{1(j)}$ is specified as a set of five polynomials over $GF(2^8)$, the values of which define the result of exponentiating to the degree 257 in the vector finite field $GF\left(\left(2^8\right)^5\right)$ set by the BVMT of the second kind represented by Table 4.

Table 3. The BVMT of the first kind for specifying the vector field $GF\left(\left(2^8\right)^5\right)$.

$\cdot$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_1$	$\lambda\mu\mathbf{e}_4$	$\lambda\mu\mathbf{e}_5$	$\tau\mathbf{e}_1$	$\lambda\sigma\mathbf{e}_2$	$\epsilon\lambda\mu\sigma\tau^{-1}\mathbf{e}_3$
$\mathbf{e}_2$	$\lambda\mu\mathbf{e}_5$	$\epsilon\mu\mathbf{e}_1$	$\tau\mathbf{e}_2$	$\epsilon\lambda\mu\sigma\tau^{-1}\mathbf{e}_3$	$\epsilon\mu\mathbf{e}_4$
$\mathbf{e}_3$	$\tau\mathbf{e}_1$	$\tau\mathbf{e}_2$	$\tau\mathbf{e}_3$	$\tau\mathbf{e}_4$	$\tau\mathbf{e}_5$
$\mathbf{e}_4$	$\lambda\sigma\mathbf{e}_2$	$\epsilon\lambda\mu\sigma\tau^{-1}\mathbf{e}_3$	$\tau\mathbf{e}_4$	$\lambda\sigma\mathbf{e}_5$	$\epsilon\sigma\mathbf{e}_1$
$\mathbf{e}_5$	$\epsilon\lambda\mu\sigma\tau^{-1}\mathbf{e}_3$	$\epsilon\mu\mathbf{e}_4$	$\tau\mathbf{e}_5$	$\epsilon\sigma\mathbf{e}_1$	$\epsilon\sigma\mathbf{e}_2$

Table 4. The BVMT of the second kind for specifying the vector field $GF\left(\left(2^8\right)^5\right)$.

$\cdot$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_1$	$\tau\mathbf{e}_1$	$\tau\mathbf{e}_2$	$\tau\mathbf{e}_3$	$\tau\mathbf{e}_4$	$\tau\mathbf{e}_5$
$\mathbf{e}_2$	$\tau\mathbf{e}_2$	$\epsilon\lambda\mathbf{e}_3$	$\epsilon\sigma\mathbf{e}_4$	$\epsilon\lambda\mathbf{e}_5$	$\epsilon\lambda\mu\sigma\tau^{-1}\mathbf{e}_1$
$\mathbf{e}_3$	$\tau\mathbf{e}_3$	$\epsilon\sigma\mathbf{e}_4$	$\epsilon\sigma\mathbf{e}_5$	$\epsilon\lambda\mu\sigma\tau^{-1}\mathbf{e}_1$	$\lambda\sigma\mathbf{e}_2$
$\mathbf{e}_4$	$\tau\mathbf{e}_4$	$\epsilon\lambda\mathbf{e}_5$	$\epsilon\lambda\mu\sigma\tau^{-1}\mathbf{e}_1$	$\lambda\mu\mathbf{e}_2$	$\lambda\mu\mathbf{e}_3$
$\mathbf{e}_5$	$\tau\mathbf{e}_5$	$\epsilon\lambda\mu\sigma\tau^{-1}\mathbf{e}_1$	$\mu\sigma\mathbf{e}_2$	$\lambda\mu\mathbf{e}_3$	$\mu\sigma\mathbf{e}_4$

Selection of the power 257 for specifying the mapping $\Psi_{1(j)}$ is determined by the purpose of defining a bijective nonlinear mapping of 5-dimensional vectors as calculation of five quadratic polynomials including three terms. Indeed, from Table 1, taking into account that in $GF(2^8)$ we have $v + v = 0\ \forall v \in GF(2^8)$, the exponentiation of the vector V in $GF\left(\left(2^8\right)^5\right)$ to the degree 2^i, where $i = 1, 2, \ldots, 8$, can be performed as calculation of monomials of the form $k_1^{(i)} v_1^{2^i}$, $k_2^{(i)} v_2^{2^i}$, $\ldots$ $k_5^{(i)} v_5^{2^i}$ (where coefficients $k^{(i)}$ represent products of some powers of structural constants):

$$\begin{aligned}
V^2 &= (v_1, v_2, \ldots, v_5)^2 = \left(\epsilon\mu v_2^2,\ \epsilon\sigma v_5^2,\ \tau v_3^2,\ \lambda\mu v_1^2,\ \lambda\sigma v_4^2\right);\\
V^4 &= \left(V^2\right)^2 = \left(\epsilon^3\mu\sigma^2 v_5^4,\ \epsilon\lambda^2\sigma^3 v_4^4,\ \tau^3 v_3^4,\ \epsilon^2\lambda\mu^3 v_2^4,\ \lambda^3\mu^2\sigma v_1^4\right);\\
V^8 &= \left(V^4\right)^2 = \left(\epsilon^3\lambda^4\mu\sigma^6 v_4^8,\ \epsilon\lambda^6\mu^4\sigma^3 v_1^8,\ \tau^7 v_3^8,\ \epsilon^6\lambda\mu^3\sigma^4 v_5^8,\ \epsilon^4\lambda^3\mu^6\sigma v_2^8\right);\\
V^{16} &= \big(\epsilon^3\lambda^{12}\mu^9\sigma^6 v_1^{16}, \epsilon^9\lambda^6\mu^{12}\sigma^3 v_2^{16}, \tau^{15} v_3^{16}, \epsilon^6\lambda^9\mu^3\sigma^{12} v_4^{16},\big)\\
&\qquad \epsilon^{12}\lambda^3\mu^6\sigma^9 v_5^{16}\big);\\
V^{32} &= \big(\epsilon^{19}\lambda^{12}\mu^{25}\sigma^6 v_2^{32}, \epsilon^{25}\lambda^6\mu^{12}\sigma^{19} v_5^{32}, \tau^{31} v_3^{32}, \epsilon^6\lambda^{25}\mu^{19}\sigma^{12} v_1^{32},\big)\\
&\qquad \epsilon^{12}\lambda^{19}\mu^6\sigma^{25} v_4^{32}\big);\\
V^{64} &= \big(\epsilon^{51}\lambda^{12}\mu^{25}\sigma^{38} v_5^{64}, \epsilon^{25}\lambda^{38}\mu^{12}\sigma^{51} v_4^{64}, \tau^{63} v_3^{64}, \epsilon^{38}\lambda^{25}\mu^{51}\sigma^{12} v_2^{64},\big)\\
&\qquad \epsilon^{12}\lambda^{51}\mu^{38}\sigma^{25} v_1^{64}\big);\\
V^{128} &= \big(\epsilon^{51}\lambda^{76}\mu^{25}\sigma^{102} v_4^{128}, \epsilon^{25}\lambda^{102}\mu^{76}\sigma^{51} v_1^{128}, \tau^{127} v_3^{128},\\
&\qquad \epsilon^{102}\lambda^{25}\mu^{51}\sigma^{76} v_5^{128}, \epsilon^{76}\lambda^{51}\mu^{102}\sigma^{25} v_2^{128}\big);\\
V^{256} &= \big(\epsilon^{51}\lambda^{204}\mu^{153}\sigma^{102} v_1, \epsilon^{153}\lambda^{102}\mu^{204}\sigma^{51} v_2, v_3, \epsilon^{102}\lambda^{153}\mu^{51}\sigma^{204} v_4,\\
&\qquad \epsilon^{204}\lambda^{51}\mu^{102}\sigma^{153} v_5\big).
\end{aligned}$$

Using Table 3, calculation of the vector $U = (u_1, u_2, \ldots, u_5) = V^{257} = V^{256}V$ gives the next result:

$$\begin{aligned}
u_1 &= \epsilon^{154}\lambda^{103}\mu^{204}\sigma^{51} v_2^2 + \left(\tau + \epsilon^{51}\lambda^{204}\mu^{153}\sigma^{102}\tau\right) v_1 v_3 +\\
&\qquad + \left(\epsilon^{205}\lambda^{51}\mu^{102}\sigma^{154} + \epsilon^{103}\lambda^{153}\mu^{51}\sigma^{205}\right) v_5 v_4;\\
u_2 &= \left(\epsilon^{102}\lambda^{154}\mu^{51}\sigma^{205} + \epsilon^{51}\lambda^{205}\mu^{153}\sigma^{103}\right) v_4 v_1 +\\
&\qquad + \left(\tau + \epsilon^{153}\lambda^{102}\mu^{204}\sigma^{51}\tau\right) v_3 v_2 + \epsilon^{205}\lambda^{51}\mu^{102}\sigma^{154} v_5^2;\\
u_3 &= \left(\epsilon^{205}\lambda^{52}\mu^{103}\sigma^{154}\tau^{-1} + \epsilon^{51}\lambda^{204}\mu^{153}\sigma^{102}\tau^{-1}\right) v_5 v_1 +\\
&\qquad + \left(\epsilon^{103}\lambda^{154}\mu^{52}\sigma^{205}\tau^{-1} + \epsilon^{154}\lambda^{103}\mu^{205}\sigma^{52}\tau^{-1}\right) v_2 v_4 + \tau v_3^2;
\end{aligned}$$

$$\begin{aligned}
u_4 &= \epsilon^{51}\lambda^{205}\mu^{154}\sigma^{102}v_1^2 + \left(\epsilon^{205}\lambda^{51}\mu^{103}\sigma^{153} + \epsilon^{154}\lambda^{102}\mu^{205}\sigma^{51}\right) v_2v_5+\\
&\quad + \left(\epsilon^{102}\lambda^{153}\mu^{51}\sigma^{204}\tau + \tau\right) v_3v_4;\\
u_5 &= \left(\epsilon^{153}\lambda^{103}\mu^{205}\sigma^{51} + \epsilon^{51}\lambda^{205}\mu^{154}\sigma^{102}\right) v_1v_2+\\
&\quad + \left(\epsilon^{204}\lambda^{51}\mu^{102}\sigma^{153}\tau + \tau\right) v_3v_5 + \epsilon^{204}\lambda^{52}\mu^{102}\sigma^{154}v_4^2.
\end{aligned} \tag{3}$$

In a similar way, the exponentiating to the degree 257 in the vector finite field $GF\left(\left(2^8\right)^5\right)$ set by Table 4 can be represented as the calculation of the values of five trinomials over the field $GF\left(2^8\right)$. The proposed nonlinear mapping $\Psi_1(X)$ is specified as performing seventeen mappings $\Psi_{1(j)}$, when the vector X is represented in the form of cancatenation of the vectors $X_1, X_2, \ldots, X_{17}$:

$$\Psi_1\left(X_1, X_2, \ldots, X_{17}\right) = \left(\Psi_{1(1)}\left(X_1\right), \Psi_{1(2)}\left(X_2\right), \ldots, \Psi_{1(17)}\left(X_{17}\right)\right),$$

where every coordinate of the output 85-dimensional vector $Y = \Psi_1\left(X\right)$ is calculated as a value of some trinomial over $GF\left(2^8\right)$. The inverse mapping $X = \Psi_1^{-1}\left(Y\right)$ is performed by the formula

$$\begin{aligned}
X = \left(X_1, X_2, \ldots, X_{17}\right) &= \Psi_1^{-1}\left(Y_1, Y_2, \ldots, Y_{17}\right) =\\
&= \left(\Psi_{1(1)}^{-1}\left(X_1\right), \Psi_{1(2)}^{-1}\left(X_2\right), \ldots, \Psi_{1(17)}^{-1}\left(X_{17}\right)\right),
\end{aligned}$$

where mappings $\Psi_{1(j)}^{-1}\left(X_j\right)$ are performed as exponentiation to the power $b = 551894941568$ in seventeen different modifications of the field $GF\left(\left(2^8\right)^5\right)$. Every of the said modifications is characterized in using a unique set of values of structural constants ϵ, λ, μ, σ, and τ.

The nonlinear mapping Ψ_2 is specified, using five unique mappings $\Psi_{2(j)}$ performed as exponentiations to the power 257 in five unique modifications of the field $GF\left(\left(2^8\right)^{17}\right)$, implementation of every of the exponentiations being performed as computation of seventeen different power polynomials (over $GF\left(2^8\right)$) every of which contains nine terms. The vector finite field $GF\left(\left(2^8\right)^{17}\right)$ is set by BVMTs with the basis vector distribution shown in Table 1 and with 17 different structural constants (for prime values of the dimension m it is sufficiently simple to find distributions of m different structural constants, for which the multiplication operation is commutative and associative).

Suppose the 85-dimensional vector $Y = (Y_1, Y_2, \dots Y_5)$ is represented as concatenation of 5 vectors $Y_i = \left(y_1^{(i)}, y_2^{(i)}, \dots y_{17}^{(i)}\right)$ of the dimension 17 ($i = 1, 2, \dots, 5$). The mapping Ψ_2 is specified as exponentiation of every vector Y_i to the power 257 in a unique vector field $GF\left(\left(2^8\right)^{17}\right)$. Since the integer 257 is mutually prime with the integer $2^{136} - 1$ (order of the multiplicative group of $GF\left(\left(2^8\right)^{17}\right)$), the latter operation defines bijective nonlinear mapping $Z_i = \Psi_{2(i)}(Y_i)$. The inverse mapping $Y_i = \Psi_{1(i)}^{-1}(Z_i)$ can be performed as exponentiation to the power

$$b' = 43725622121389384503558299750298495778688,$$

since $b' \equiv 257^{-1} \bmod 2^{136} - 1$. To specify five unique mappings $\Psi_{2(i)}$ ($i = 1, 2, \dots, 5$), one is to set five different sets of the values of structural constants in the BVMT specifying the vector field $GF\left(\left(2^8\right)^{17}\right)$. The values of the said constants are generated at random but so that the said field is set. Thus, the mapping $Z = \Psi_2(Y)$ is described as follows:

$$Z = (Z_1, Z_2, \dots, Z_5) = \Psi_2\left(Y_1, Y_2, \dots, Y_5\right) = \\ \left(\Psi_{2(1)}\left(Y_1\right), \Psi_{2(2)}\left(Y_2\right), \dots, \Psi_{2(5)}\left(Y_5\right)\right),$$

where every coordinate of the output 85-dimensional vector $Z = \Psi_2\left(Y\right)$ is calculated as a value of some power polynomial (containing 9 terms) over $GF\left(2^8\right)$. We do not provide a set of 17 square polynomials describing the mappings $\Psi_{2(i)}$, since this would require a rather cumbersome table with 17 structural constants. This can be done similarly to the case of description of the mappings $\Psi_{1(j)}$. Note that the polynomials describing the mappings $\Psi_{2(i)}$ contain 9 terms of the second degree. (It is reasonable to leave a detailed consideration of this issue, including the generation of a BVMT for the case $m = 17$ with 17 different distributions of structural constants, for the stage of software implementation of the algorithm.)

The inverse mapping $Y = \Psi_2^{-1}\left(Z\right)$ is performed by the formula

$$Y = (Y_1, Y_2, \dots, Y_5) = \Psi_2^{-1}\left(Z_1, Z_2, \dots, Z_5\right) = \\ = \left(\Psi_{2(1)}^{-1}\left(Z_1\right), \Psi_{2(2)}^{-1}\left(Z_2\right), \dots, \Psi_{2(5)}^{-1}\left(Z_5\right)\right),$$

where mappings $Y_i = \Psi_{2(i)}^{-1}(Z_i)$, for $i = 1, 2, \ldots, 5$, are performed as exponentiation to the power b' in five different modifications of the field $GF\left(\left(2^8\right)^{17}\right)$. Every of the said modifications is characterized by using a unique set of values of 17 structural constants.

Obviously, the generated public key Π represents a set of 85 polynomials (of the 4th degree), whose variables are the coordinates of the input vector X. Every polynomial contains 81 terms, the latter being ordered in the lexicographic order of the products of the variables. Assuming the term ordering convention, each polynomial can be specified as a set of 8-bit coefficients. Therefore, the size of public key is equal to $85 \cdot 81 = 6885$ bytes ($\approx$7 kB. The 170-byte secret key represents the set of 170 structural constants used to specify 17 modifications of the field $GF\left(\left(2^8\right)^5\right)$ and 5 modificatios of the field $GF\left(\left(2^8\right)^{17}\right)$.

Each public key coefficient is the product of some set of structural constants. However, the calculation of structural constants from known coefficients is associated with the solution of a system of equations of high degree (>50; see formulas (3)), which includes 170 unknowns. The latter represents a specific structural attack against the introduced MPC algorithm, which is similar to a standard direct attack representing solving a system of 85 equations (set by the public-key polynomials) of the 4th degree with 85 unknowns. Due to the supposed computational difficulty of the said structural attack, the calculation of mappings Ψ_1 and Ψ_2 (such that $\Pi = \Psi_2 \circ \Psi_1$) by the public key seems to be a computationally infeasible task.

4 Discussion

Using the data in Table 1, by analogy with the proposed algorithm, other MPC algorithms can be developed. Besides, linear mappings (for example, permutations of the coordinates of the input vector) can be used additionally, which do not lead to an increase in the number of terms in the public key polynomials. It is also of interest to specify non-linear mappings in the form of exponentiation operations in vector fields $GF\left((p)^m\right)$ with an odd characteristic p.

In general, the proposed approach provides quite ample opportuni-

Table 5. The minimum number of equations in $GF(q)$ for the case $u = n$ [1].

$L = \dots$	2^{80}	2^{100}	2^{128}	2^{192}	2^{256}
$q = 16$	30	39	51	80	110
$q = 31$	28	36	49	75	103
$q = 256$	26	33	43	68	93

ties for developing algorithms with a relatively small size of the public key, when providing a given security level. When estimating the security of the MPC algorithms, two types of attacks are distinguished: i) direct attacks and ii) structural ones. An attack of the first type consists in solving a system of power equations given by the public key polynomials for some vector $Z = (z_1, z_2, \dots, z_u)$. The solution gives preimadge $X = (x_1, x_2, \dots, x_n)$ of the vector Z, i. e., $X = \Pi^{-1}(Z)$. The computational difficulty of the best direct attack defines the upper limit of the MPC algorithms' security. The best-known methods for solving a system of many power equations with many unknowns use the algorithms F4 [6] and F5 [7]. The computational difficulty of the complexity of those methods depends exponentially on the number of equations and weakly depends on the degree of the equations and on the order of the field in which the equations are given. Table 5 shows the minimum number of equations (for the case $n = u$) that are required to get a given security level.

In the introduced algorithm, we have $n = u = 85$ and $q = 256$, therefore, the security against direct attack can be evaluated as 2^{192}. Modifications of the MPC algorithms, Rainbow [8] (signature finalist of the NIST competition for development of the post-quantum public-key standards) and GeMSS [9] (alternative algorithm participated in the third round of the NIST competition), have the size of public key equal to ≈260 kB and ≈1300 kB for the case of the 2^{192} security level, correspondingly (≈40 and ≈190 times more than the proposed algorithm).

Thus, the proposed method represents a significant interest in developing the MPC algorithms with a practical size of the public key.

In addition, the size of the secret key (170 bytes) is also significantly smaller against Rainbow (≈600 kB) and GeMSS (≈35 kB). Obviously, the essential advantage of algorithms Rainbow and GeMSS is that the resistance to structural attacks of various modifications of the algorithms was considered within a fairly long period of time. Study of the security of the introduced algorithm against potential structural attacks is connected with the following two items:

i) due to significantly different designs, the known structural attacks are hardly applicable to the proposed algorithm;

ii) new stuructural attacks are to be considered.

A natural structural attack against the proposed algorithm is the calculation (by the known coefficients of the public key polynomials) of 170 structural constants used to define 17 modifications of the vector field $GF\left(\left(2^8\right)^5\right)$ and 5 modifications of the field $GF\left(\left(2^8\right)^{17}\right)$. This structural attack is similar to the direct attack, since it is connected with solving a system of many power equations with many unknowns, given in $GF(2^8)$. In this structural attack, we have 6885 power equations and 170 unknowns. A potential attacker may attempt to select different subsets of power equations, for which the solution has a lower computational complexity. However, taking into account the significantly larger number of unknowns (170 versus 85), we can expect that the complexity of this structural attack will be higher compared to the direct attack.

In fact, the proposed specific algorithm is an illustration of the proposed new paradigm for constructing MPC algorithms, and there are significant reserves for the development of new algorithms and their modification taking into account structural attacks that may appear in the future.

The development of new structural attacks on the algorithms developed in the framework of the proposed approach and their detailed consideration represent independent research tasks.

It is also of interest to use exponentiation operations in the vector finite fields $GF\left((2^d)^m\right)$ to specify a nonlinear mapping within the framework of the generally accepted approach to the development of the MPC algorithms [1]–[3].

5 Conclusion

For the first time the exponentiation operations in vector finite fields of characteristic two have been proposed to implement nonlinear mappings in the MPC algorithms, the public key being formed as a superposition of two different nonlinear mappings, which is given as a set of power polynomials of the fourth degree. The introduced approach seems promising for the development of practical post-quantum algorithms, including digital signature algorithms for possible submission to the NIST competition in framework of the call for additional proposals [4].

References

[1] J. Ding and A. Petzoldt, "Current State of Multivariate Cryptography," *IEEE Security and Privacy Magazine*, vol. 15, no. 4, pp. 28–36, 2017.

[2] Q. Shuaiting, H. Wenbao, Li Yifa, and J. Luyao, "Construction of Extended Multivariate Public Key Cryptosystems," *International Journal of Network Security*, vol. 18, no. 1, pp. 60–67, 2016.

[3] Y. Hashimoto, "Recent Developments in Multivariate Public Key Cryptosystems," in *International Symposium on Mathematics, Quantum Theory, and Cryptography* (Mathematics for Industry, vol. 33), T. Takagi, M. Wakayama, K. Tanaka, N. Kunihiro, K. Kimoto, and Y. Ikematsu, Eds. Singapore: Springer, 2021, pp. 209–229. https://doi.org/10.1007/978-981-15-5191-8_16.

[4] Post-Quantum Cryptography: Digital Signature Schemes, 2022. [Online]. Available: https://csrc.nist.gov/Projects/pqc-dig-sig/standardization/call-for-proposals.

[5] N. A. Moldovyan and P. A. Moldovyanu, "Vector Form of the Finite Fields $GF(p^m)$," *Bulletin of Academy of Sciences of Moldova. Mathematics*, vol. 3, no. 61, pp. 57–63, 2009.

[6] J.-C. Faugére, “A new efficient algorithm for computing Grőbner basis (F4),” *J. Pure Appl. Algebra*, vol. 139, no. 1–3, pp. 61–88, 1999.

[7] J.-C. Faugére, “A new efficient algorithm for computing Grőbner basis without reduction to zero (F5),” in *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, 2002, pp. 75–83.

[8] Rainbow Signature. One of three NIST Post-quantum Signature Finalists, 2021. [Online]. Available: https://www.pqcrainbow.org/.

[9] GeMSS: A Great Multivariate Short Signature. [Online]. Available: https://www-polsys.lip6.fr/Links/NIST/GeMSS.html.

A. A. Moldovyan[1], N. A. Moldovyan[2]

Received February 1, 2023
Revised February 6, 2023
Accepted September 20, 2023

[1,2]St. Petersburg Federal Research Center of
the Russian Academy of Sciences (SPC RAS),
St. Petersburg Institute for Informatics and
Automation of the Russian Academy of Sciences
14 Liniya, 39, St.Petersburg, 199178
Russia

[1]Alexandr Moldovyan
ORCID: https://orcid.org/0000-0001-5480-6016
E–mail: maa1305@yandex.ru

[2]Nikolay Moldovyan
ORCID: https://orcid.org/0000-0002-4483-5048
E–mail: nmold@mail.ru

Computer Science Journal of Moldova, vol.**32**, no.1(94), 2024

Optimizing Cervical Cancer Classification with SVM and Improved Genetic Algorithm on Pap Smear Images

S. Umamaheswari, Y. Birnica, J. Boobalan, V. S. Akshaya

Abstract

This study presents an approach to optimize cervical cancer classification using Support Vector Machines (SVM) and an improved Genetic Algorithm (GA) on Pap smear images. The proposed methodology involves preprocessing the images, extracting relevant features, and employing a genetic algorithm for feature selection. An SVM classifier is trained using the selected features and optimized using the genetic algorithm. The performance of the optimized model is evaluated, demonstrating improved accuracy and efficiency in cervical cancer classification. The findings hold the potential for assisting healthcare professionals in early cervical cancer diagnosis based on Pap smear images.

Keywords: SVM, Pap smear images, Cervical cancer, GA, Healthcare.

MSC 2020: 62H35.

1 Introduction

Cervical cancer is a serious health problem that affects women all over the world and has a significant impact on morbidity and mortality rates [1]. Early detection through screening plays a serious role in reducing the burden of this disease. However, the interpretation of Pap smear images, a commonly used screening tool, can be subjective and lead to varying levels of diagnostic accuracy. In recent years, there has been an increasing interest in using sophisticated computational methods like

doi:10.56415/csjm.v32.05

support vector machines and genetic algorithms to optimize cervical cancer classification based on Pap smear images. This study aims to address the challenges associated with cervical cancer classification by proposing an approach that combines SVM and an improved genetic algorithm. By integrating these techniques, the goal is to develop a robust and efficient model capable of accurately analyzing Pap smear images and providing reliable classification outcomes. The optimization process involves preprocessing the images, extracting relevant features, selecting informative features using a genetic algorithm, and training an SVM classifier with optimized parameters. The resulting model holds the potential to enhance the accuracy and efficiency of cervical cancer diagnosis, assisting healthcare professionals in making timely and accurate decisions for improved patient care.

2 Methodology

The proposed methodology for optimizing cervical cancer classification with SVM and Improved Genetic Algorithm on Pap smear images involves several steps:
1. *Pre-processing:* The Pap smear images are first pre-processed to enhance the contrast and eliminate noise. This involves techniques such as histogram equalization and Gaussian filtering.
2. *Segmentation:* The pre-processed images are then segmented to extract the region of interest (ROI), which contains the cervical cells. This involves using a super pixel-based Markov random field segmentation algorithm.
3. *Feature extraction:* Various textures and patterns are extracted from the segmented ROI. These include Gabor filters, Haralick features, and Zernike moments.
4. *Feature selection:* The most appropriate features that improve classification accuracy are determined using an improved genetic algorithm.
5. *Classification:* The selected features are used to train an SVM classifier. The SVM classifier is optimized using the Improved Genetic Algorithm to find the best hyperparameters that maximize the classification accuracy.
6. *Performance evaluation:* Parameters including accuracy, precision,

recall, and F1-Measure are used to assess the proposed system's performance [2]. The results are compared with existing benchmark methods to determine the effectiveness of the proposed approach.

3 Proposed System

3.1 Image pre-processing

To optimize Pap smear images for cervical cancer analysis, image preprocessing [3] is essential. To improve image quality, decrease noise, and extract important information, many techniques are used. Among these methods are scaling the images to a uniform dimension, boosting contrast to improve visibility, lowering noise through smoothing or filtering, normalizing intensity or color values, extracting the region of interest (such as the cervix), segmenting various structures within the image, and registering multiple images for precise comparisons. The quality and appropriateness of Pap smear pictures are enhanced through the use of various pre-processing techniques, enabling more precise and trustworthy analysis and classification of cervical cancer.

Image pre-processing methods encompass a range of techniques used to enhance and optimize images before further analysis or classification. In the context of medical imaging, including Pap smear images for cervical cancer analysis, the following methods are commonly employed: 1. Image resizing, 2. Contrast enhancement, 3. Noise reduction, 4. Image normalization, 5. Edge detection, 6. Image segmentation, 7. Morphological operations. This article mainly focuses on Image normalization and Image segmentation techniques.

3.1.1 Image Normalization

Image normalization is a pre-processing technique used to standardize the intensity or color values of an image. It ensures that images from different sources or under different conditions can be compared and analyzed accurately. By rescaling pixel values to a consistent range or distribution, normalization improves the reliability and comparability of image features, reducing the impact of variations caused by factors such as lighting or imaging settings. It enhances the quality and consistency of images, making them more suitable for subsequent analysis or classification tasks.

3.2 Image Segmentation

The Region Of Interest (ROI) containing the cervical cells is extracted from the segmented Pap smear images that have undergone pre-processing. Super pixel-based Markov Random Field (MRF) [4] segmentation technique is used to complete this segmentation process.

3.2.1 Super pixel generation

Super pixels are compact and perceptually meaningful image regions that group pixels with similar characteristics together. They provide a higher level of abstraction compared to individual pixels and can facilitate more efficient and accurate segmentation. Super pixels are generated by dividing the image into compact, regular regions while preserving image boundaries.

3.2.2 Markov Random Field (MRF) Modelling

MRF is a probabilistic graphical model broadly used in image segmentation. It models the relationship between neighboring pixels by incorporating contextual information to improve the accuracy.

Let's denote the pre-processed image as I and the corresponding super pixels as S. Each super pixel S consists of a set of pixels $S = p_1, p_2, ..., p_n$. The goal is to assign a label (foreground or background) to each superpixel. The MRF formulation is as follows,

$$E(S) = \sum \left(D(S) + V(S)\right), \tag{1}$$

where $E(S)$ is the energy function, D(S) represents the data term, and $V(S)$ denotes the regularization term.

3.2.3 Data Term

The data term measures the compatibility between each super pixel and the estimated class labels. It is based on the colour and texture features of the super pixels. The data term is defined as follows,

$$D(S) = \sum (D_{data(S)}), \tag{2}$$

where $D_{data(S)}$ represents the data cost for each super pixel S.

3.2.4 Regularization Term

The regularization term encourages spatial coherence and smoothness in the segmentation results. It penalizes sharp transitions between neighboring super pixels. The regularization term is defined as follows,

$$V(S) = \sum (V_{reg(S)}), \tag{3}$$

where $V_{reg(S)}$ represents the regularization cost for each super pixel S.

3.2.5 Optimization

The goal is to find the optimal labeling configuration S* that minimizes the energy function $E(S)$. This optimization problem is typically solved using optimization techniques such as graph cuts or belief propagation.

The pre-processed pictures are segmented into areas that correspond to the cervical cells, which form the ROI using the super pixel-based MRF. This segmentation step is crucial for isolating the cells and providing accurate input for the subsequent stages of the classification process.

3.3 Feature Extraction

Feature extraction is the process that takes the segmented ROI from the previous module and extracts texture and shape features from it. These qualities offer crucial details regarding the traits and patterns visible in cervical cell images which can be utilized for classification. Gabor filters, Haralick features, and Zernike moments are used to extract texture and shape features. Gabor filters are spatial frequency filters that capture texture information at different scales and orientations, while Haralick features describe the statistical properties of pixel intensities in an image. Zernike moments capture the shape characteristics of the segmented ROI, including its boundary and internal structure.

The extraction of these features is performed using AlexNET's pretrained CNN model. The extracted features form a feature vector representing the unique characteristics of the segmented ROI, which serves as input data for the subsequent classification module. Machine learning algorithms or classifiers can be trained to distinguish between different types of cervical cells and classify them accordingly.

The feature extraction module plays a vital role in capturing relevant information from segmented ROIs and transforming it into a suitable format for classification. By utilizing Gabor filters, Haralick features, and Zernike moments, the system effectively captures both texture and shape characteristics, enabling accurate classification and diagnosis of cervical cancer cells.

3.4 Feature Selection

The fourth module of the proposed cervical cancer classification project is featuring selection. In this module, an Improved Genetic Algorithm (IGA) is employed to identify and select the most related features from the feature vector obtained in the previous module. The goal of feature selection is to keep just the most useful features that considerably improve classification accuracy while reducing the dimensionality of the feature space.

Feature selection using an Improved Genetic Algorithm involves the following steps:

1. Initial Population: A population of potential feature subsets is randomly generated. Everyone in the population represents a candidate feature subset, where each feature is represented by a binary value (0 or 1) indicating its inclusion or exclusion in the subset.

2. Fitness Evaluation: The fitness of everyone in the population is obtained using a Fitness Function (FF). The FF assesses the classification accuracy achieved by using the corresponding feature subset. This

evaluation is typically done by training and testing a classifier (SVM) on the selected features and measuring its performance using metrics like accuracy, precision, recall, or F1-Measure.

$$fitness(individual) = performance\ (classifier,\ selected\ features) \tag{4}$$

3. Selection: A selection process is applied to select individuals with higher fitness values. The individuals with better performance (higher classification accuracy) have a higher probability of being selected for reproduction. Common selection techniques include tournament selection, roulette wheel selection, or rank-based selection [5]. In this research, Roulette wheel selection is used because it gives better results compared with other methods.

4. Crossover: Crossover is performed by selecting pairs of individuals from the selected population and combining their feature subsets to create new offspring. This process simulates the genetic recombination that occurs in natural evolution. Different crossover techniques can be used, such as 1-point crossover, 2-point crossover, or uniform crossover.

5. Mutation: Mutation introduces random changes in the feature subsets of the offspring. This facilitates the exploration of new search space areas while preventing an early convergence. Mutation randomly flips the binary values (0 to 1 or 1 to 0) of certain features in the offspring.

6. Fitness Evaluation (Offspring): The fitness of the newly created offspring is evaluated using the fitness function, similar to the initial population. This step determines the classification accuracy achieved by the offspring using their modified feature subsets.

7. Replacement: The offspring are selected to replace individuals in the population based on their fitness values. This process ensures the survival of individuals with higher fitness and discards those with lower fitness, maintaining the population size.

8. Termination: The Improved Genetic Algorithm iteratively performs the steps of selection, crossover, mutation, and fitness evaluation. This can be a fixed number of generations, reaching a specific fitness threshold, or convergence of the algorithm.

3.5 Classification Using SVM

The fifth module of the cervical cancer classification project uses features from previous modules to train the SVM classifier. The classifier is optimized using the Improved Genetic Algorithm to find optimal hyperparameters for maximum accuracy. SVM is a widely used supervised learning algorithm in medical image analysis. There are six steps involved in classification.

1. Data Preparation: The feature vector, consisting of the selected features obtained from the feature selection module, serves as the input data for the SVM classifier. The feature vector is typically represented as a matrix, where each row represents an instance or sample, and each column represents a feature.

2. SVM Training: Using the labelled training data, the SVM classifier is trained. The feature vectors with corresponding class labels (such as ”positive” or ”negative” for cervical cancer) make up the training data. Finding the best hyperplane to maximally separate the positive and negative instances in the feature space is the goal of the SVM method.

3. Hyper parameter Optimization: IGA is involved in optimizing the hyper parameters of the SVM classifier. The hyper parameters are settings that control the behavior and performance of the SVM algorithm. Examples of SVM hyper parameters include the kernel type and regulation parameter (C). The Improved Genetic Algorithm investigates several hyperparameter combinations to determine which ones produce the best classification accuracy.

4. Fitness Evaluation: The fitness function in the Improved Genetic Algorithm evaluates the classification accuracy achieved by the SVM classifier using the selected hyper parameters. The classification accuracy is typically measured using metrics such as accuracy, precision, recall, or F1-Measure [6]. The Improved Genetic Algorithm chooses individuals for reproduction in the next generation who have higher fitness values.

5. Crossover and Mutation: Crossover and mutation procedures are carried out using the Improved Genetic Algorithm to produce new candidate solutions in the population. Crossover involves integrating

the hyper parameters of chosen individuals to produce a new generation with various combinations of hyper parameters. For exploring new areas of the search space and preventing premature convergence, mutation introduces random modifications to the hyperparameters.

6. Replacement and Termination: The offspring with their modified hyper parameters replace individuals in the population based on their fitness values. This process ensures the survival of individuals with better classification accuracy. The Improved Genetic Algorithm iteratively performs the steps of fitness evaluation, crossover, mutation, replacement, and termination.

3.6 Performance Evaluation

Performance evaluation is a crucial step in assessing the effectiveness of the proposed cervical cancer classification system. It involves the use of various metrics to quantitatively measure the system's performance and compare it with existing benchmark methods.

1. Accuracy: Accuracy measures the overall correctness of the classification system and is defined as the ratio of correctly classified instances to the total number of instances [6]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{5}$$

2. Precision: Precision is the ability of the system that measures how well the system can pick out positive examples from the sum of instances that are projected to be positive. It is calculated as [6]:

$$Precision = \frac{TP}{TP + FP}. \tag{6}$$

3. Recall: The capacity of the system to accurately identify positive cases out of all real positive instances is measured by recall, also known as sensitivity or true positive rate. It is defined as [6]:

$$Recall = \frac{TP}{TP + FN}. \tag{7}$$

4. F1-Measure: The harmonic mean of precision and recall is the F1-Measure, which offers a single measure to balance them both. It is

calculated as [6]:

$$F1 - Measure = 2 * \frac{Precision * Recall}{Precision + Recall}. \quad (8)$$

By considering both precision and recall, the F1-Measure provides a comprehensive evaluation of the system's performance, especially when dealing with imbalanced datasets.

The performance assessment is normally carried out using an appropriate dataset that contains labeled samples of various cervical cell types. The dataset is divided into training and testing sets, with the former being used to train the classification model and the latter to assess the model's efficacy.

The proposed system's performance is compared with recent techniques by applying the same evaluation metrics to their results. The comparison helps determine whether the proposed approach achieves superior or comparable performance compared to other methods.

4 Dataset

Dataset Name: Cervical Cancer Pap Smear Images Dataset. There are four classes available in this data set, and each class has its own image samples as listed below [6].

1. Carcinoma In Situ (200 samples)
2. Light Dysplastic (300 samples)
3. Moderate Dysplastic (250 samples)
4. Severe Dysplastic (350 samples)

Carcinoma In Situ

Number of Samples: 200

Description: Carcinoma in situ shown in Figure 1 states the abnormal cells that are found only in the innermost lining of the cervix. An aberrant cell cluster known as a carcinoma in situ is one that has not yet spread from the site where it initially developed, yet it has the potential to do so in the future to transform into cancer. Carcinoma in situ is considered a pre-cancerous condition and is an early stage of cervical cancer.

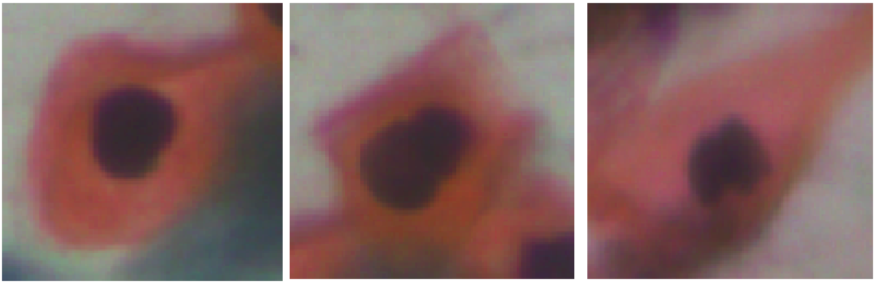


Figure 1. Carcinoma In Situ

Light Dysplastic

Number of Samples: 300

Light dysplastic in Figure 2 states the presence of mildly anomalous cells in the cervical tissue. These cells show some changes but are not considered cancerous. Light dysplastic cells may indicate the primary phases of cervical cancer development and require further monitoring and treatment.

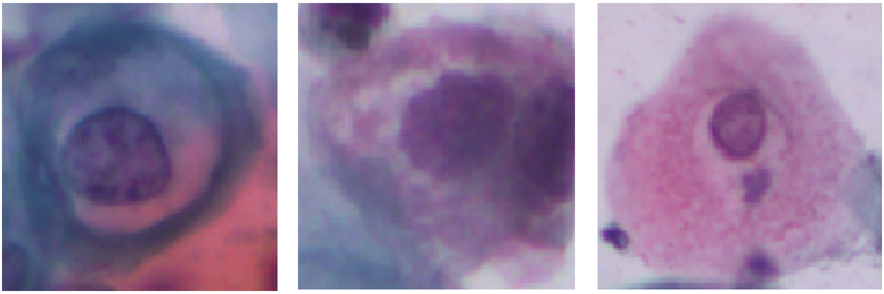


Figure 2. Light Dysplastic

Moderate Dysplastic

Number of Samples: 250

Moderate dysplastic shown in Figure 3 describes the presence of moderately anomalous cells in the cervical tissue. These cells exhibit more pronounced changes than light dysplastic cells but are still not classified as cancerous. Moderate dysplastic cells indicate a higher risk of cervical cancer and may require closer monitoring and treatment.

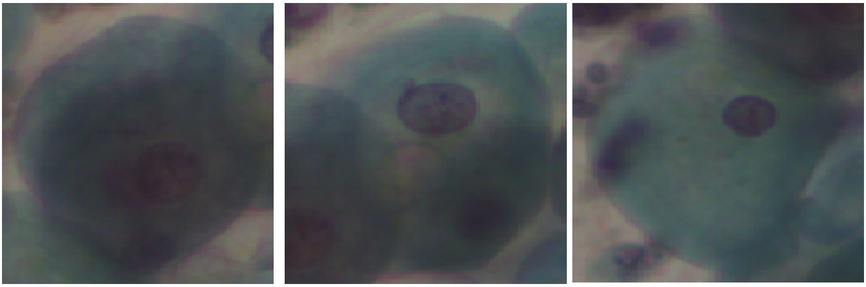


Figure 3. Moderate Dysplastic

Severe Dysplastic
Number of Samples: 350
Severe dysplastic in Figure 4 shows significantly abnormal cells in the cervical tissue. These cells show significant changes and have a higher possibility of developing cervical cancer if left untreated. Severe dysplastic cells require prompt medical intervention and treatment.

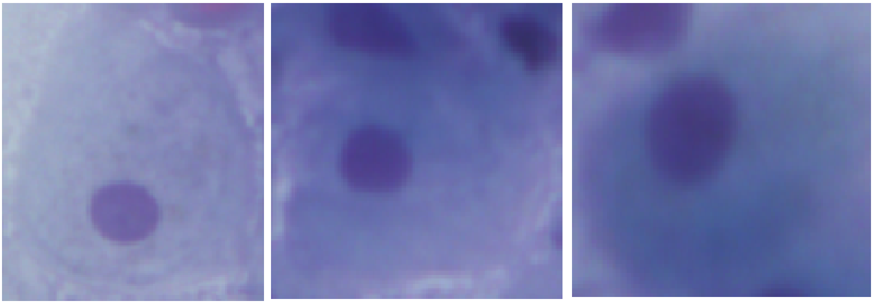


Figure 4. Severe Dysplastic

Data split: 70% (770 samples) of the samples from the data set are being used for training, 15% (165 samples) of the data is used for validation, and the remaining 15% (165 samples) is used for testing [7].

5 Results and Discussions

The graphical user interface of the segmentation process in the proposed cervical cancer classification system is presented in Figure 5. It provides users with a user-friendly platform to interact with the system and perform image segmentation on the selected Pap smear image.

1. Segmentation Options: The GUI includes controls related to segmentation techniques available in the system.

2. Segmentation Visualization: The GUI includes a visualization area where users can see the segmented regions overlaid on the original image. This allows users to visually inspect the segmentation results and assess the quality of the segmentation before proceeding to further analysis.

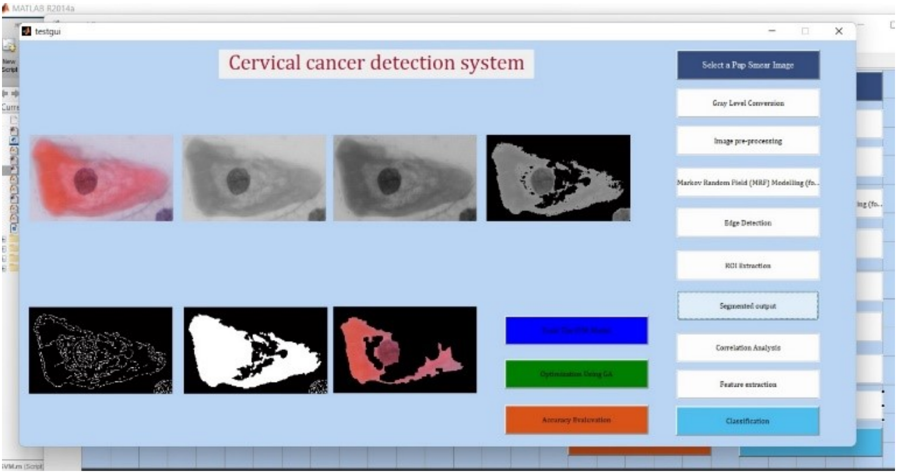


Figure 5. Segmentation process of the proposed cervical cancer classification system

The classification efficiency comparison for different methods in Cervical Cancer Classification is presented. The proposed method, IGA-SVM, is compared with existing machine learning algorithms such as Linear Regression, Logistic Regression, Decision Trees, Random Forests, and Naive Bayes. The evaluation metrics used for comparison include Accuracy, Precision, Recall, and F1-Measure [25].

The proposed method, IGA-SVM obtains the optimum Accuracy of 97.14% compared to other machine learning algorithms. It also demonstrates superior Precision (99.50 %) and F1-Measure (97.90%), indicating the ability to correctly classify the Carcinoma In Situ class with high precision and balance between precision and recall. The Recall (96.30%) is slightly lower compared to Naive Bayes, Random Forests, and Decision Trees, but still at a high level. Classification efficiency comparison for Carcinoma In Situ with other learning models is shown in Figure 6.

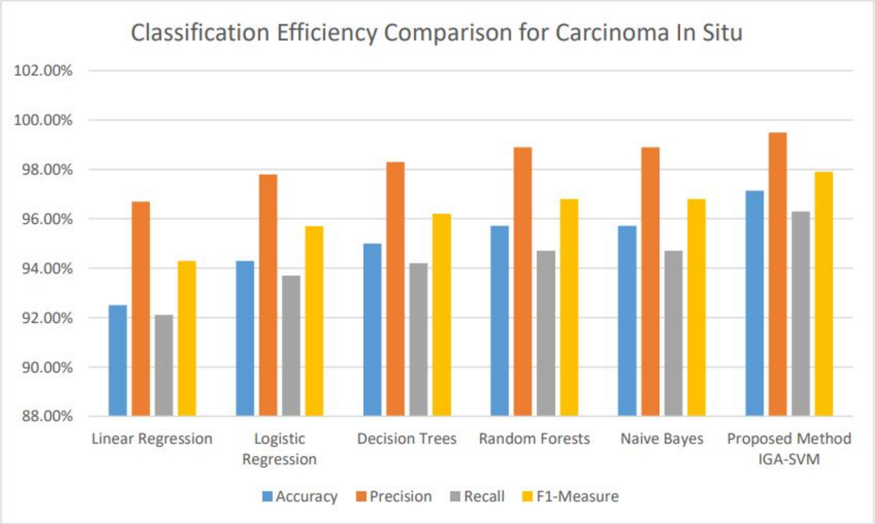


Figure 6. Classification Efficiency Comparison for Carcinoma In Situ Classification with machine learning models

Classification Efficiency Comparison for Light Dysplastic Classification with machine learning models is shown in Figure 7. According to the results, the IGA-SVM outperforms all other methods with an Accuracy of 98.1%. It also achieves high Precision (98.0%) and F1-Measure (98.7%), indicating the model's ability to accurately classify the Light Dysplastic class.

Additionally, the Recall (99.4%) is the highest among all methods, suggesting that the proposed method effectively captures the positive instances of the Light Dysplastic class.

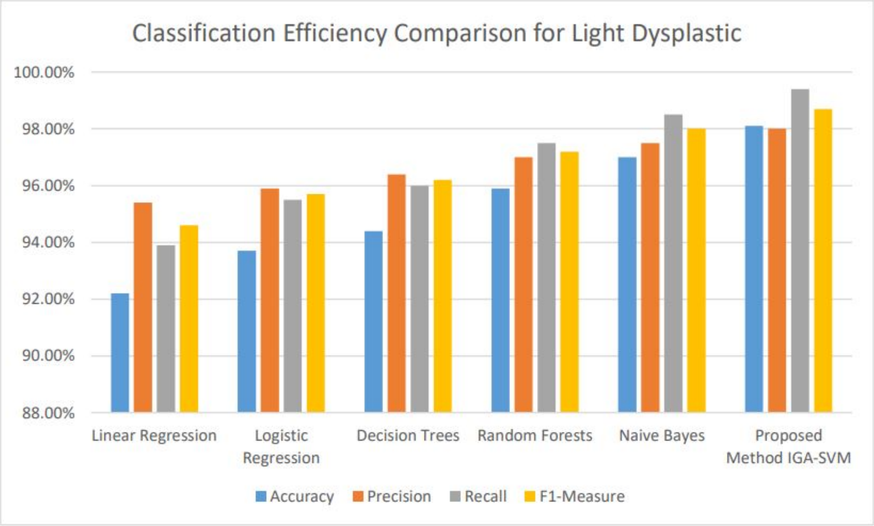


Figure 7. Classification Efficiency Comparison for Light Dysplastic Classification with machine learning models

Classification Efficiency Comparison for Moderate Dysplastic Classification with machine learning models is presented in Figure 8. Similar to previous cases, IGA-SVM achieves the highest Accuracy (98.9%), Precision (100%), and F1-Measure (99.2%) among all methods. It shows perfect Precision, indicating no false positive predictions in the Moderate Dysplastic class. The Recall (98.4%) is slightly lower compared to Naive Bayes but still at an excellent level.

IGA-SVM attains the highest Accuracy (96.1%) and F1-Measure (96.9%) among all methods in Severe Dysplastic classification model as depicted in Figure 9. It demonstrates high-precision (96.7%) and Recall (97.2%), indicating its ability to effectively classify the Severe Dysplastic class. It outperforms the other methods consistently in terms of Accuracy and F1-Measure.

According to the results, the proposed method, IGA-SVM, consistently achieves the highest or competitive classification performance across all dysplastic classes. It outperforms other machine learning al-

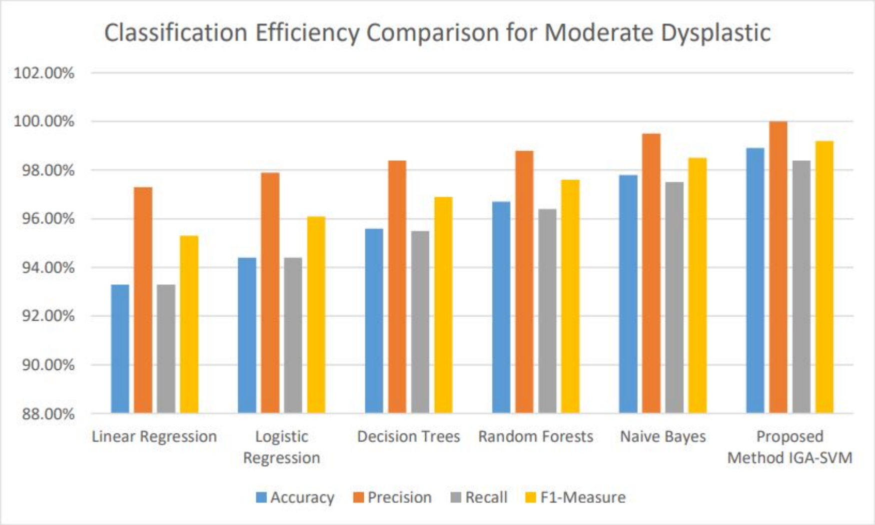


Figure 8. Classification Efficiency Comparison for Moderate Dysplastic Classification with machine learning models

gorithms in terms of Accuracy, Precision, Recall, and F1-Measure in most cases. This suggests that the combination of Improved Genetic Algorithm for feature selection and Support Vector Machine for classification provides an effective framework for cervical cancer classification based on Pap smear images.

6 Related Works

A CAD framework that classifies cytology images uses an ensemble of three standard CNN-based classifiers. The proposed ensemble model generates ranks of the classifiers using two non-linear functions which help to take into account the confidence in predictions of the base learners [8]. A CNN-based ThinPrep cytologic test (TCT) cervical cancer screening model was established through a retrospective study of multicenter TCT images. This model shows improved speed and accuracy for cervical cancer screening, and helps overcome the shortage of med-

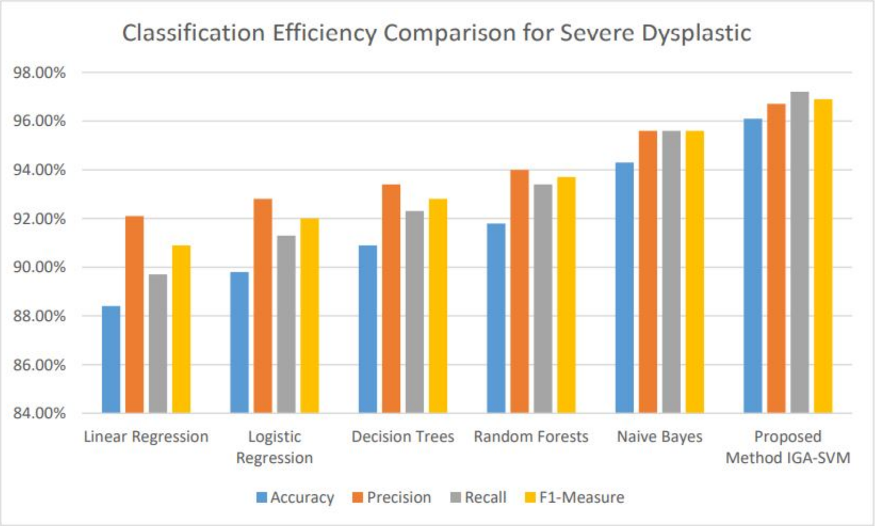


Figure 9. Classification Efficiency Comparison for Severe Dysplastic Classification with machine learning models

ical resources required for cervical cancer screening [9]. An automatic cervical cancer screening system using convolutional neural network was proposed in [10]. A novel deep learning method named AttFPN as an automated detection model for abnormal cervical cells in cervical cancer screening was proposed in [11]. The model was guided by clinical knowledge and attention mechanism, consisting of a multi scale feature fusion structure and an attention module. The proposed method outperformed the related stat-of-the-art deep learning methods and was comparable to a pathologist with 10 years of experience. DGCA-RCNN framework for the detection of abnormal cervical cells in Pap smear images was presented in [12]. It is an extended version of the Faster RCNN-FPN model by introducing deformable convolution layers into FPN to improve scalability and adding a GCA module alongside RPN to enhance the spatial context information. Cervical cell image generation model based on taming transformers (CCGtaming transformers) to provide high quality cervical cancer datasets with sufficient samples and balanced weights was developed [13].

The structure of the encoder was improved by introducing SEblock and MultiRes-block to improve the ability to extract information from cervical cancer cells images. An efficient and totally segmentation-free method for automated cervical cell screening that utilizes a modern object detector to directly detect cervical cells or clumps, without the design of a specific handcrafted feature was proposed in [14] to investigate the presence of unreliable annotations and cope with them by smoothing the distribution of noisy labels. A new framework based on a strong feature Convolutional Neural Networks (CNN) Support Vector Machine (SVM) model was proposed to accurately classify the cervical cells. A method fusing the strong features extracted by Gray-Level Cooccurrence Matrix (GLCM) and Gabor with abstract features from the hidden layers of CNN was conducted, meanwhile the fused ones were input into the SVM for classification. An effective dataset amplification method was designed to improve the robustness of the model [15].

The technique used in hospital laboratories involves the manual numeration of blood cells victimization using a device referred to as Haemocytometer. Using this method can be monotonous, sometimes produces inaccurate results and also is a time consuming process. So as to beat the complications, this analysis presents an absolute automatic systemized system to associate with nursing platelet cells within the blood samples and to classify various forms of Leukaemia [16]. The Pap smear test is a manual screening procedure that is used to detect precancerous changes in cervical cells based on color and shape properties of their nuclei and cytoplasms. Automating this procedure is still an open problem due to the complexities of cell structures. An unsupervised approach for the segmentation and classification of cervical cells was proposed. The segmentation process involves automatic thresholding to separate the cell regions from the background, a multi-scale hierarchical segmentation algorithm to partition these regions based on homogeneity and circularity, and a binary classifier to finalize the separation of nuclei from cytoplasm within the cell regions [17].

Cervical cancer is one of the leading causes of cancer death in females worldwide. The disease can be cured if the patient is diagnosed in the pre-cancerous lesion stage or earlier. A common physical examination technique widely used in the screening is Papanicolaou test or Pap test. In this automated cervical cancer cell segmentation and classification method [18], a single-cell image is segmented into the nucleus, cytoplasm, and background, using the fuzzy C-means (FCM) clustering technique.

Accurate classification of Pap smear images becomes the challenging task in medical image processing. This can be improved in two ways. One way is by selecting suitable well defined specific features and the other is by selecting the best classifier. To enhance accuracy, the earlier detection and, for diverse perspectives, the proposed research analyzed the techniques used in [19], [21], and [23] to apply in medical imaging analysis, including the classification of Pap smear images for cervical cancer detection. A nominated texture-based cervical cancer (NTCC) classification system which classifies the Pap smear images into any one of the seven classes was presented in [20]. This can be achieved by extracting well defined texture features and selecting best classifier. Pap smear test has been broadly used for detection of cervical cancer. However, the conventional Pap smear test has several shortcomings including: subjective nature (dependent on individual interpretation), low sensitivity (i.e., ability to detect abnormal changes), and the need for frequent retesting. There has been a great effort to automate Pap smear tests, and it is one of the critical fields of medical image processing [22]. A method for detecting overlapping cell nuclei in Pap smear samples was presented in [24]. The extraction of overlapping cell nuclei is a critical issue in automated diagnosis systems. Due to the similarities between overlapping and malignant nuclei, misclassification of the overlapped regions can affect the automated systems final decision.

7 Conclusions

A novel approach for optimizing cervical cancer classification using SVM and Improved Genetic Algorithm on Pap smear images has been proposed. The methodology involves image pre-processing, segmentation, feature extraction, feature selection, classification, and performance evaluation. The results show high accuracy, precision, recall, and F1-Measure in classifying cervical cancer cells. The Improved Genetic Algorithm is effective for the SVM classifier's feature selection and hyper-parameter tuning. Future work will extend the approach to other imaging modalities and explore deep-learning techniques for improved cervical cancer classification. Additional research is required to confirm the methodology's efficacy on larger data sets and more clinical investigations.

References

[1] M.A. Yalda, I. Y. Abdulmalek, and H. R. M. Ali, "Women's Knowledge Regarding Pap Smear and Cervical Cancer in Duhok City in Respect to Related Educational Perspective Session," *History of Medicine*, vol. 9, no. 1, pp. 960–967, 2023.

[2] Cheon et al., "Feature Importance Analysis of a Deep Learning Model for Predicting Late Bladder Toxicity Occurrence in Uterine Cervical Cancer Patients," *Cancers*, vol. 15, no. 13, Article no. 3463, Jul. 2023, DOI: 10.3390/cancers15133463.

[3] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evol. Intel.*, vol. 15, no. 1, pp. 1–22, Mar. 2022, DOI: 10.1007/s12065-020-00540-3.

[4] H. Yao et al., "Semantic Segmentation for Remote Sensing Image Using the Multigranularity Object-Based Markov Random Field With Blinking Coefficient," *IEEE Trans. Geosci. Remote Sensing*, vol. 61, pp. 1–22, 2023, DOI: 10.1109/TGRS.2023.3301494.

[5] Z. Alyafeai and L. Ghouti, "A fully-automated deep learning pipeline for cervical cancer classification," *Expert Systems with Applications*, vol. 141, Article no. 112951, 2020. DOI: 10.1016/j.eswa.2019.112951.

[6] M. Alsalatie et al., "A New Weighted Deep Learning Feature Using Particle Swarm and Ant Lion Optimization for Cervical Cancer Diagnosis on Pap Smear Images," *Diagnostics*, vol. 13, no. 17, Article no. 2762, Aug. 2023, DOI: 10.3390/diagnostics13172762.

[7] M. Sumathi and S. P. Raja, "Machine learning algorithm-based spam detection in social networks," *Soc. Netw. Anal.*, vol. 13, Article no. 104, 2023, https://doi.org/10.1007/s13278-023-01108-6.

[8] A. Manna, R. Kundu, D. Kaplun, A. Sinitca, and R. Sarkar, "A fuzzy rank-based ensemble of CNN models for classification of cervical cytology," *Sci. Rep.*, vol. 11, Article no. 14538, 2021. https://doi.org/10.1038/s41598-021-93783-8.

[9] X. Tan et al. "Automatic model for cervical cancer screening based on convolutional neural network: a retrospective, multicohort, multicenter study," *Cancer Cell Int*, vol. 21, Article no. 35, 2021, DOI: 10.1186/s12935-020-01742-6.

[10] Aziz-ur-Rehman, Nabeel Ali, Imtiaz. A. Taj, Muhammad Sajid, Khasan S. Karimov, "An Automatic Mass Screening System for Cervical Cancer Detection Based on Convolutional Neural Network", *Mathematical Problems in Engineering*, vol. 2020, Article ID 4864835, 14 pages, 2020. https://doi.org/10.1155/2020/4864835.

[11] Lei Cao et al, "A novel attention-guided convolutional network for the detection of abnormal cervical cells in cervical cancer screening," *Medical Image Analysis*, vol. 73, Article no. 102197, October 2021, DOI: 10.1016/j.media.2021.102197.

[12] Xia Li, Zhenhao Xu, Xi Shen, Yongxia Zhou, Binggang Xiao and Tie-Qiang Li, "Detection of Cervical Cancer Cells in Whole Slide Images Using Deformable and Global Context Aware Faster RCNN-FPN," *Curr. Oncol.*, vol. 28, no. 5, pp. 3585–3601, 2021 Sep 16, DOI: 10.3390/curroncol28050307. PMID: 34590614; PMCID: PMC8482136.

[13]] Chen Zhao, Renjun Shuai, Li Ma, Wenjia Liu, and Menglin Wu, "Improving cervical cancer classification with imbalanced datasets combining taming transformers with T2T-ViT," *Multimed. Tools Appl.*, vol. 81, pp. 24265–24300, 2022. https://doi.org/10.1007/s11042-022-12670-0.

[14] Yao Xiang et al, "A novel automation-assisted cervical cancer reading method based on convolutional neural network", *Biocybernetics and Biomedical Engineering*, vol. 40, no. 2, pp. 611–623, April–June 2020.

[15] A. Dongyao Jia, B. Zhengyi Li, and C. Chuanwang Zhang, "Detection of cervical cancer cells based on strong feature CNN-SVM

network," *Neurocomputing*, vol. 411, pp. 112–127, 21 October 2020, https://doi.org/10.1016/j.neucom.2020.06.006.

[16] K. Balakumar, Anand T. Gokul, G. Naveenkumar, and S. Umamaheswari, "Improving the Performance of Leukemia Detection using Machine Learning Techniques," in *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, (Coimbatore, India), 2022, pp. 867–872, DOI: 10.1109/ICESC54411.2022.9885461.

[17] A. Gençtav, S. Aksoy, and S. Önde, "Unsupervised segmentation and classification of cervical cell images," *Pattern Recognit*, vol. 45, no. 12, pp. 4151–4168, 2012.

[18] T. Chankong, N. Theera-Umpon, and S. Auephanwiriyakul, "Automatic cervical cell segmentation and classification in pap smears," *Comput Meth Prog Bio*, vol. 113, no. 2, pp. 539–556, 2014.

[19] B. V. Dharani Krishna, C. Kavin Prabhu, S. Harish, and S. Umamaheswari, "Towards Building of a Robust Organic Fruit Tester," in *2022 8th International conference on Advanced computing and Communication systems(ICACCS)*, 2022, pp. 631–634.

[20] Edwin Jayasingh Mariarputham and Allwin Stephen, "Nominated Texture based Cervical Cancer Classification," *Computational and Mathematical Methods in Medicine*, Article no. 586928, 2015, DOI: 10.1155/2015/586928.

[21] S. Umamaheswari, S. Aartisha, J. Kanimozhi, and R. Suhashini, "Building accurate legal case outcome prediction models," in *2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 16 June 2023, DOI: 10.1109/ICAECA56562.2023.10200651.

[22] T. A. Sajeena and A. S. Jereesh, "Automated cervical cancer detection through RGVF segmentation and SVM classification," in *2015 International Conference on Computing and Network Communications (CoCoNet)*, (Trivandrum, India), pp. 663–669, 2015.

[23] S. Umamaheswari, K. Harikumar, and D. Allinjoe, "Customer Relationship Management using Sentimental Analysis," in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 2021, DOI: 10.1109/ICAECA52838.2021.9675766.

[24] M. Guven, C. Cengizler, "Data cluster analysis-based classification of overlapping nuclei in Pap smear samples," *Biomed Eng Online*, vol. 13, Article no. 159, 2014, DOI: 10.1186/1475-925X-13-159.

[25] A. Rehman, T. Saba, M. Mujahid, F. S. Alamri, and N. ElHakim, "Parkinson's Disease Detection Using Hybrid LSTM-GRU Deep Learning Model," *Electronics*, vol. 12, no. 13, Article no. 2856, Jun. 2023.

S. Umamaheswari, Y. Birnica,
J. Boobalan, V. S. Akshaya

Received October 11, 2023
Revised February 2, 2024
Accepted February 4, 2024

S. Umamaheswari
ORCID: https://orcid.org/0000-0001-6590-2521
Associate Professor, Dept.of ECE, Kumaraguru College of Technology
Coimbatore, Tamilnadu, India.
E–mail: umamaheswari.s.ece@kct.ac.in

Y. Birnica
Dept. Dept.of ECE, Kumaraguru College of Technology
Coimbatore, Tamilnadu, India.
E–mail: birnica.21mco@kct.ac.in

J. Boobalan
ORCID: https://orcid.org/0000-0002-9655-5435
Assistant Professor, Dept.of ECE, Kumaraguru College of Technology
Coimbatore, Tamilnadu, India.
E–mail: boobalan.j.ece@kct.ac.in

V. S. Akshaya
ORCID: https://orcid.org/0000-0001-7120-3006
Professor, Dept. of. CSE, Sri Eshwar College of Engineering
Coimbatore, Tamilnadu, India.
E–mail: vsakshaya@gmail.com

A Binary Grey Wolf Optimizer with Mutation for Mining Association Rules

KamelEddine Heraguemi, Nadjet Kamel,
Majdi M. Mafarja

Abstract

In this decade, the internet becomes indispensable in companies and people life. Therefore, a huge quantity of data, which can be a source of hidden information such as association rules that help in decision-making, is stored. Association rule mining (ARM) becomes an attractive data mining task to mine hidden correlations between items in sizeable databases. However, this task is a combinatorial hard problem and, in many cases, the classical algorithms generate extremely large number of rules, that are useless and hard to be validated by the final user. In this paper, we proposed a binary version of grey wolf optimizer that is based on sigmoid function and mutation technique to deal with ARM issue, called BGWOARM. It aims to generate a minimal number of useful and reduced number of rules. It is noted from the several carried out experimentations on well-known benchmarks in the field of ARM, that results are promising, and the proposed approach outperforms other nature-inspired algorithms in terms of quality, number of rules, and runtime consumption.

Keywords: Association rules mining, ARM, Grey Wolf Optimizer, support, confidence.

MSC2020: 62H20.

1 Introduction

Nowadays, the huge number of connected devices to INTERNET become a relevant source of data. As a consequence, the saved data needs to be explored and used in many other fields such as Marketing,

doi:10.56415/csjm.v32.06

Engineering, and Medical [1]. Due to this huge amount of data, automated processing becomes an interesting research area for academic researchers. The data mining field includes a huge number of techniques that process data and attempt to collect accurate, relevant, engaging, and comprehensible knowledge from huge databases. One of the most attractive tasks in data mining is Association Rule Mining (ARM) [2]. It seeks to define correlations among items in a transactional dataset.

Agrawal et al. [3] introduced the Association rule (AR) concept in 1993. Since that, ARM has been widely and successfully applied in many hypersensitive domains such as healthcare, market analysis, electric engineering, and web recommendation systems [4]. Basically, ARM aims to identify important dependencies between items in a given dataset in the form of an IF-THEN statement: IF $<$ some conditions are satisfied $>$ THEN $<$ some values of other attributes$>$. Conditions in the IF statement are called Antecedents, and those within the THEN clause are called Consequences. Obviously, numerous relations of this kind can be extracted from a dataset, but only the useful relations in the real life need to be selected.

Indeed, discovering ARs in a wide transactional dataset is an NP-Hard problem [4]. In a database with n items, there exists $2n$ itemsets, which generate a maximum number of $2k-2$ association rules, where k is the length of itemsets. This proves that the time consumption exponentially increases with the increase of the number of items. Moreover, the increase of items affects memory consumption too, especially nowadays, with huge stored data. This case makes traditional algorithms, such as Apriori [3] and FP-Growth [4], require a considerable execution time. In order to overcome this drawback, many studies take direction to evolutionary and bio-inspired algorithms, such as genetic algorithms [5], particle swarm algorithms [6], bat algorithm [7], and recently whale optimization algorithm [8], to select the most useful and interesting ARs within a reasonable time and less hardware consumption. Generally, for intelligent algorithms, the database is considered as a search space, and the algorithm – as an exploration strategy that aims to explore the search space and define the rules that maximize/ minimize an earlier defined fitness function that evaluates the rule quality based on its measures. Moreover, many researches deal with ARM as a multi-objective optimization problem [9], [10]. This idea has been motivated by the huge number of rules' quality measures introduced in various objective functions. Almost all the time, optimization algorithms prove their robustness and efficiency to solve ARM issues within an acceptable runtime and less hardware consumption.

Grey Wolf Optimization algorithm (GWO) is one of the most well-known nature-inspired optimization approaches published recently [11]. It mimics the social hierarchy of the grey wolves in nature. GWO confirmed its efficiency in various real-life applications such as electrical engineering [12] and feature selection [13]. GWO confirmed its competitivity compared to other swarm-inspired metaheuristics, such as Particle swarm optimization (PSO), Bat Algorithm (BA), and Bee swarm optimization algorithm (BSO) in terms of exploitation and exploration. Furthermore, GWO beats other metaheuristics in terms of the number of variables that need to be initialized. In GWO, only one variable has to be initialized. With this in mind, and motivated by the success of GWO in various domains, we propose in this paper a new binary version of GWO based on sigmoid function and mutation technique to deal with ARM issue, namely BGWOARM. We use a new bitmap database representation, and an updated wolf's position updating algorithm is introduced to generate candidate rules. Afterward, a mutation operator is applied to get the fittest rule. To evaluate the efficiency of the proposed approach, deep experimentations are carried out on various famous benchmarks in the field of ARM defers in size and item number. Also, a comparative study in terms of runtime and rule quality is made with recently published method in the domain of ARM. The computational results of BGWOARM are promising and prove its efficiency.

The rest of this paper is organized as follows: Section 2 provides a literature review that shows the recently published works in the field of ARM. The section that immediately follows introduces a general background on association rule mining and the original grey wolf optimizer. Section 4 presents the details of our proposed method to solve ARM issue based on a binary grey wolf optimizer. Furthermore, the results of our proposal are outlined. Finally, we conclude and outline our future work and improvements.

2 Related Works

Since its inception in 1993 by Agrawal et al. [2], the problem of ARM got a lot of attention. In literary research, there are a remarkable number of researchers that may be divided into two approaches; exact and optimization. The first approach seeks to retrieve all the relationships between objects that exist across the database, while the second aims to produce essential and relevant rules. ARM has been dominated by two major methods: 1) Apriori, a well-known traditional method, identifies all associations depending on the minimal support specified by the expert [3]; 2) FP-growth, which was established to solve Apriori shortcomings, notably multiple dataset scans, in which the entire dataset is only scanned twice [14]. These techniques now have to deal with a lot of data, which makes them slower and memory eaters.

Afterward, studies have been made to deal with data mining problems as optimization problems, even for ARM which is considered as an NP-hard problem. Thus, several researches started in applying genetic algorithms (GA) to extract ARs from transactional databases [15]. GAs are evolutionary algorithms based on the natural reproduction of DNA's. Mainly, the application of such algorithms to tackle ARM has three main tasks, which are: rule encoding, fitness function definition, and generating new rules from the dataset. Yang et al. in [5] proposed an approach based on GA for identifying the ARs. This method didn't use any user specified minimum thresholds. Whereas, the authors utilized a relative minimum confidence as objective function to pick the best rules. In 2014, Drias in [16] declared that most of the optimization algorithms for ARM have two disadvantages: they generate false rules and extract low support and confidence rules as a high-quality rule. To cope with these drawbacks, the authors proposed two GA-based approaches, the first one named IARMGA and the other based on a Memetic algorithm, named IARMMA. In this work, the authors described a new technique called delete and decomposition strategy, that aimed to obtain rules with higher fitness. Their test results demonstrate that IARMMA offers greater solution quality. Whereas, IARMMA has increased processing time relative to IARMGA, especially when the data growth.

Recently, in [17] a modified GA was proposed with the aim to extract interesting and non-redundant relationships between items in a dataset. In order to concretize their objective, the authors consider four different quality measures, support, confidence, comprehensibility, and interestingness, to evaluate the rules. Also, they controlled the redundant rules by designing a novel rules filter method. The results obtained by the experimental study proves the efficiency of their idea.

With the development of bio-inspired techniques, numerous swarm-based algorithms, such as the PSO algorithm, Bat algorithm (BA), and Firefly algorithm (FA), etc., are recommended to cope with ARM problem. In the work presented in [18], PSO has been used to discover ARs in a transactional database. In this work, there were two phases: preprocessing and mining phase. The first one was to evaluate the objective function, while, the other one was to generate the rules based on PSO algorithm. An enhanced technique based on PSO was designed in [19]. This study proposed a Boolean variant of PSO to extract ARs named (BPSO), whereby it obtains the best rules without imposing any measurement criteria.

More recently, the authors in [6] proposed a technique, based on a new binary PSO for detecting unseen correlations among both machine abilities and product characteristics without specified minimum limits. At the same time, they provided a unique overlapping measure indicator to remove less quality regulations. Derouiche et al., presented in [20] an application of Chemical Reaction Optimization metaheuristic (CRO) for solving ARM problem, namely CRO-ARM. Many experiments were carried out on two datasets and compared to Apriori, FP-growth, and BSPO. The outcomes were promising; however, this approach needs to be tested on larger datasets.

There are several further papers in the literature that focus on Bee swarm optimization techniques and provide an approach called BSO-ARM to mine ARs [21]. Tests showed that BSO-ARM enjoys much better results than genetic algorithms. As well, an additional study was published using three processes to determine each bee’s study area (Modulo, next, syntactic). Moreover, and based on the Penguin Search Engine Optimization (Pe-ARM) method, the authors suggested an association rule miner [22]. This approach is distinguished by thorough exploration of the search space. The efficiency of this proposal is proved by numerous tests carried out on different biological data-sets.

As the first investigation of the ARM problem with the BAT algorithm, an algorithm called BAT-ARM was proposed in [23]. The bat algorithm mimics the echolocation behavior of microbats, where they move toward the prey based on the processing of the echolocation. In BAT-ARM, a new formulation of bat movements was introduced according to ARM problem. In order to prove the efficiency of their proposal, the authors carried out several tests and compared the results to those of Apriori and FP-Growth algorithms. The major drawback of this approach was those bats in the populations don't share their information about the preys. Consequently, the search space for exploration is reduced. In the continuity of their work, the same authors proposed an improvement for BAT-ARM by introducing communication strategies, namely: master/slave [24], ring, and Hybrid [7]. All strategies proved their superiority against BAT-ARM and other recently published works in terms of runtime and rules quality [7]. Along with our work on the bat algorithm, in [9] we suggested a multi-objective BA to optimize 4 quality measurements in the field of ARs which are: interestingness, comprehensibility, support, and confidence. Results show the superiority of multi-objective approach to extract the best and useful rules to the final user.

The approach proposed in [25] uses a sigmoid function binary cuckoo search, and it was applied to extract categorical ARs. Recently, Whale optimization algorithm (WOA) was adopted to extract relationships between items in a dataset [8]. The researchers look into the excellent trade-off between intensity and diversity that characterized the classic whale optimizer, which was founded on an encircling methodology, a spiral-shaped pathway, and a hunt strategy. In terms of execution speed, excellence, and memorial consumption, WOA-ARM technique outperformed other works. A new review, that summarizes several evolutionary-based computation methods used for solving an ARM problem, was presented in [4].

A review of the works conducted on ARM detection from a large-scale dataset shows the key role optimization algorithms to accelerate the mining process. In most cases, SI algorithms suffer from the large number of parameters, which makes the process to choose the best ones hard for a final user. The main difference between the proposed method of this paper and the existing methods is updating a binary version of GWO to deal with ARM, which has only one parameter that needs to be chosen by the final user. Also, a mutation technique is introduced in order to improve the generated rules.

3 Background

In this section, some fundamental specifications of the proposed methodology are described. First, we explain the basics of ARM. Then, we present the GWO algorithm.

3.1 Preliminaries on Association rule mining

ARM problem was presented by Agrawal et al. in 1993, with the goal of helping in decision-making and assisting grocery administrators in designing discounts and placing products in the store to achieve maximized profitability. These decisions depend on connections that have been created from a massive portion of previous transactions gathered by the sellers [3].

Definition 1. *"Formally, the association rule problem is defined as follows: Let* $I = i_1, i_2, ..., i_n$ *be a set of literals called items; let* $Dt_1, t_2, ..., t_m$ *be a transactional database, where each transaction* t *contains a set of items. An association rule is an implication like* $X \Rightarrow Y$*, where* $X, Y \in I$ *and* $X \bigcap Y = \Phi$*" [7], where* X, Y *are called antecedent (If statement) and consequent (then statement), respectively."*

To calculate the quality of the generated patterns from whichever datasets collected and in order to determine the highest notable instances for the decision maker, several objective and subjective [26] measurements are created and published over the time, that can be used to judge ARs. Objective measures were utilized to examine the produced rules in this research. Because of the huge number of frequent item sets collected from a large scale dataset, a discovered pattern is allowed as an AR only if its support and its confidence are equal or greater than the minimal limit imposed by the user, and disallowed if they are not. Support and confidence are two measures that aim to determine rules quality, which is defined as follows:

Definition 2. *"Support is the proportion of transactions in D that contains X, to the total of records in database. Support of item X is calculated using equation 1 and the support of an association rule $X \Rightarrow Y$ is the support of $X \cup Y$" [7].*

$$support(X) = \frac{(Number\ of\ transactions\ containing\ X)}{(Total\ Number\ of\ transactions)}. \tag{1}$$

Definition 3. *"Confidence is the proportion of transactions covering X and Y, to the total of records containing X. When the percentage exceeds a threshold of confidence, an interesting association rule can be generated" [7].*

For instance, a rule $X \Rightarrow Y$ with a confidence level of 0.8 states that 80 percent of the transactions containing X also include Y. The confidence can be formulated as follows:

$$confidence(X \Rightarrow Y) = \frac{support(X \bigcup Y)}{support(X)}. \tag{2}$$

3.2 Grey Wolf Optimization algorithm (GWO)

GWO algorithm is one of the well-known nature-inspired optimization approaches that were published recently [11]. GWO was differentiated from other swarm intelligence (SI) algorithms by a number of features. There is just one variable that may be adjusted in this method. Furthermore, with GWO, a suitable balance between diversity and intensity may be established. Consequently, this proposed method has shown promising convergence in dealing with a wide range of engineering challenges. Additionally, GWO is a basic approach which may be simply applied and implemented. It mimics the hierarchical structure of wolves' pack and its collective hunting strategy in wildlife. Usually, wolves desire being in a community with something like a consistent hierarchy.

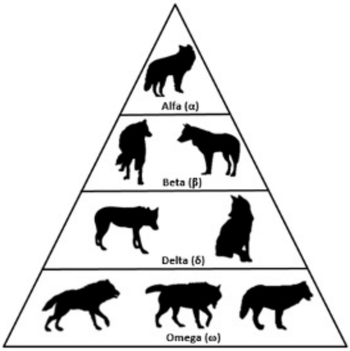


Figure 1. A graphic illustrating the hierarchical structure of wolves' pack

Figure 1 provides an illustration of wolf hierarchy, in which α wolves always considered as the most significant hunters and γ wolves have the lowest strength in face to other classes. Mathematically the wolves' hierarchy structure in the pack is presented by Mirjalili et al. [11]. The remaining search agents have been declared as ω, which is driven and guided by α, β, and γ to the search space which is full of promising solutions in hope to find the best optimal solution. Essentially, the mathematical formalism of GWO is stated in 3 key stages defined as follows: surrounding the victim, hunting the prey, and attacking the prey [12], and they are stated as follows:

3.2.1 Surrounding the victim

When a prey is found, the iteration begins ($t = 1$). Hence, α, β, and γ wolves drive the ω group to chase and ultimately surround the victim, this Gray wolf's strategy can be formulated mathematically as:

$$X(t\vec{+}1) = X_p\vec{(}t) + \vec{A}.\vec{D}, \tag{3}$$

where $\vec{X}$ is the wolves' actual location, $\vec{X_p}$ referes to the prey's locality, t presents the actual iteration, and $\vec{A}$ is an array of coefficients. Whereas, $\vec{D}$ is defined as follows:

$$\vec{D} =\mid \vec{D}.X_p\vec{(}t) - X\vec{(}t) \mid . \tag{4}$$

The parameters $\vec{A}$ and $\vec{C}$ are combinations of controlling parameters which can be calculated as follows:

$$\vec{A} = 2\vec{a}.\vec{r_1} - \vec{a}, \tag{5}$$

$$\vec{C} = 2.\vec{r_2}, \tag{6}$$

where $\vec{a}$ are elements gradually reduced from 2 to 0 throughout the optimization process, and $\vec{r_1}, \vec{r_2}$ are random arrays in [0,1].

3.2.2 Hunting the prey

The grey wolf is hunting by shifting the location of every wolf in the pack by moving toward the prey; this habit is theoretically expressed in the form: α is the leader with best position, β and γ are supposed to have extra details regarding prey's possible places. Thus, the ω group will follow them and be forced to move in light of the leaders within the next iterations. The location changing or hunting behavior is stated as follows:

$$\vec{D_\alpha} =\mid \vec{C_1^t}.\vec{X_\alpha^t} - X^t \mid,\ \vec{D_\beta} =\mid \vec{C_1^t}.\vec{X_\beta^t} - X^t \mid,\ \vec{D_\gamma} =\mid \vec{C_1^t}.\vec{X_\gamma^t} - X^t \mid, \tag{7}$$

$$\vec{X_1^t} = \vec{X_\alpha^t} - A_1^t.\vec{D_\alpha^t},\ \vec{X_2^t} = \vec{X_\beta^t} - A_2^t.\vec{D_\beta^t},\ \vec{X_3^t} = \vec{X_\gamma^t} - A_3^t.\vec{D_\gamma^t}, \tag{8}$$

$$X^{t+1} = \frac{X_1^t + X_2^t + X_3^t}{3}. \tag{9}$$

3.2.3 Attacking the prey

The parameter $\vec{a}$ governs the attacking procedure, updates the values of $\vec{A}$, and guides the ω group to pursue / leave the victim (solution). Theoretically, if $\mid \vec{A} \mid > 1$, wolves are on the lookout for a new strategy to expand their search area. Otherwise, they go toward their dominants, implying that omega wolves would follow the leaders who take advantage of the limited search area. $\vec{a}$ are carried on:

$$\vec{a} = 2(1 - t/N), \tag{10}$$

where N is the maximum iteration number, and t refers to the actual iteration. The bounds of $\vec{A}$ will be inside $[-2a, 2a]$. Hence, the search agents can touch any region between their position and the location of the quarry when the $\vec{A}$ is in the interval of $[-1, 1]$ by decreasing the weighting values of $\vec{a}$ from 2 to 0. The motion rule when $| \vec{A} |< 1$ or $| \vec{A} |> 1$ is vividly shown in Fig. 2.

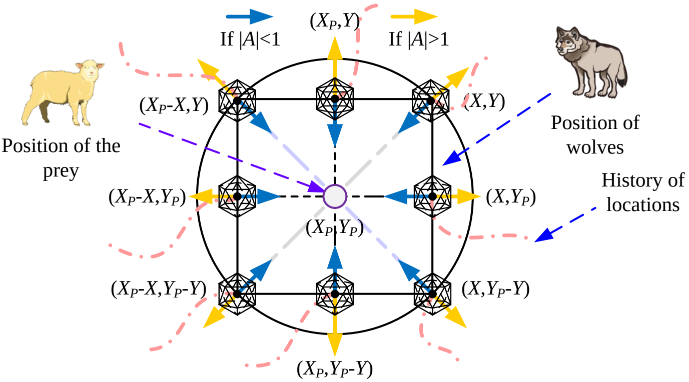
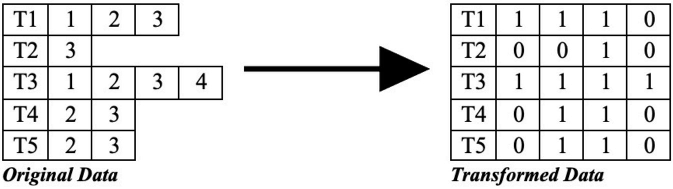


Figure 2. Impact of $\vec{A}$ on the direction of motion in GWO

4 Proposed Algorithm

4.1 Dataset and rule representation

In data mining, the data preprocessing task is one of the most irrelevant tasks. This task can influence directly on the model results. With this in mind, as well as to avoid the multi-database scans that also affect calculation time and memory consumption, datasets are transformed to bitmap representation [2] which simplifies the process of support and confidence computing. Let us consider, as shown in Figure 3, that there are 5 transactions ***T1*** to ***T5*** in transactional database which contains 4 Items. All transactions are transformed to binary form. For more illustration, consider ***T4*** in which the consumer purchased 2 products ***(I2, I3)***. Therefore, for ***B4***, the rows under ***I2*** and ***I3*** will contain the value '1'. Whereas, ***I1*** and ***I4*** will contain the value '0' because these two items don't exist in the transaction.



T1	1	2	3	
T2	3			
T3	1	2	3	4
T4	2	3		
T5	2	3		

Original Data

T1	1	1	1	0
T2	0	0	1	0
T3	1	1	1	1
T4	0	1	1	0
T5	0	1	1	0

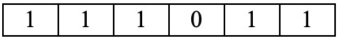
Transformed Data

Figure 3. Database representation

Besides, in order to use nature-inspired algorithms, which are mainly designed for continuous optimization problems, to solve an ARM problem, rules need to be represented in a structural form that can be used by the proposed algorithms. Indeed, the literature contains two main representations for ARM which are the Pittsburgh method and Michigan method [27]. The first supposed that a set of rules is considered as a single individual; however, the other considers every rule as one individual in the population. Our proposal opts for the second one, where each rule (solution) is presented by an array of ***2k*** items, where ***k*** is the number of items in the database. The vector is coded as follows:

- $R[]i] = 1$ if the i^{th} item exists in the rule, and 0 otherwise.
- $R[i+1] = 0$ if the i^{th} item exists in antecedent of the rule, and 1 if it appears in the consequence part.

For instance: let $I = i_1, i_2, i_3$ be a set of items: the rule $i_2 \rightarrow i_1, i_3$. It is coded as $X1 = 1, 1, 1, 0, 1, 1$. Figure 4 shows a rule encrypted.



1	1	1	0	1	1

Figure 4. Rule Encoding

4.2 Fitness function

The fitness function examines the quality of solutions in nature-inspired algorithms. To get the optimum results, it must be maximized. Indeed, to formulate a good objective function, that rewards the proper kind of solutions, is important. As previously stated, an association rule is approved if its support and confidence levels meet the user's requirements. The fitness function of a rule R may be formulated as follows:

$$fit(R) = \begin{cases} \alpha.Support(R) + \beta.Confidence(R) & \text{if R is accepted} \\ -1 & \text{otherwise} \end{cases} \tag{11}$$

4.3 Binary Grey Wolf Optimizer for ARM

This section describes the whole process of our binary GWO algorithm to deal with the ARM problem. Our approach preserves the same philosophy and process as in the original GWO, from which the described algorithm has inherited its performance and advantages. The modifications take the three main steps of GWO, Encircling, Hunting the prey, and attacking, in relation to our application problem (ARM).

As mentioned above, our proposal is based on Pittsburgh encoding which means that any individual in the population is a solution (rule). Thus, the wolves in the pack in the BGWOARM algorithm represent the generated rules. The initialization step consists of creating and assigning to each wolf a random rule from the search space [11], evaluating the initial fitness values, and selecting the leaders, $(X_\alpha = best\ agent, X_\beta = Second\ best\ agent, X_\gamma = Third\ best\ agent)$, that will lead the pack through the search space and guide the pack to the prey. **Algorithm 1** shows the Binary Grey Wolf Optimizer for ARM.

When the prey is detected, the encircling task starts by $i = 1$. The leader leads the rest of the agents toward the prey based on the new proposed rule generation algorithm presented in **Algorithm 2**. The algorithm consists of calculating the distance of the new rule based on two steps according to rule encoding (Item existence and Item Positions). In order to evaluate each distance, equation 4 is used. After-

Algorithm 1 Binary Grey Wolf Optimizer for ARM

```
Input: Number of MaxIte, number of wolves in Population, minSup, MinConf
Output: Set of valid Association rules (ValidRules)
Initialize the swarm Xi (i = 1, 2, . . ., n),
Evaluate the initial fitness for all agents,
Select $X_\alpha, X_\beta, X_\gamma$                              ▷ The Fittest wolves
$i \leftarrow 1$
while $i <= MaxIte$ do
    for each wolf in the pack do
                                             ▷ Update positions by Algorithm 2
        X1 = Generation of new rule ($X_\alpha$, $X_i, C_i$);
        X2 = Generation of new rule ($X_\beta$, $X_i, C_i$);
        X3 = Generation of new rule ($X_\gamma$, $X_i, C_i$);
      ▷ Generate new position (Rule) based on Mutation Algorithm 3
        NewRule= Mutation ($X_1, X_2, X_3$);
    end for
    Update a, A and C,
    if NewRule is accepted then
        Evaluate the fitness function Eq. (11)
        Add NewRule to the set of rules ValidRules
        Update $X_\alpha, X_\beta, X_\gamma$
    end if
    $i \leftarrow i + 1$
end while
Return ValidRules
```

ward, a sigmoid function, equation 12, is applied to convert the results to binary, either for the item's existence or appearance position. When the sigmoid of the distance, in relation to the item, is less or equal to a random value, the item exists and not otherwise. Whereas, if the sigmoid of the distance, in relation to the position, is less or equal to a random value, the item appears in the consequence, and in the antecedent otherwise.

$$sigmoid(x) = \frac{1}{1 + e^{-10(A*D-0.5)}}, \tag{12}$$

where A is the actual coefficient for the actual wolf and D is the distance between the agent and the prey, which is divided into two values that are $D_{item}, D_{Position}$. The first refers to the item that exists or is not in the rule, whereas, the second defines where the item appears, in the antecedent or consequence of the rule.

Afterward, the hunting behavior is the process in which each wolf in the pack has to change its position with the aim of approaching the prey. In the mathematical formulation of the original GWO which is appropriate for continuous optimizations, the hunting is presented by the equations 7,8, and 9, in which the ω pack, that can move continuously in the search space, is obliged to pursuit the leaders (α, β, and γ). Thus, it is impossible to use the same hunting process for solving the ARM problem. With this in mind, a crossover algorithm is proposed to make the ω pack obliged to pursue the leaders in the hunting process. To resume, three new rules are generated in relation to α, β, and γ actual positions. Afterward, the crossover is applied between them to generate the new position. **Algorithm 3** shows the crossover algorithm.

Moreover, the generated rule is evaluated against the user thresholds (MinSup, MinConf); when the rule is accepted, it is added to a set of valid rules, and its fitness is evaluated. Finally, the leaders are updated based on the new fitness values. This search will be repeated until the maximum number of iterations is reached.

Algorithm 2 Generation of new rule

```
Input: $Rule_x$, $Rule_i$, $C : coefficient$
Output: Rule
$i \leftarrow 1$
while $i <= 2k$ do
                                        ▷ Calculate the prey Distance
    $D_{item} =\mid C * Rule_x(i) - Rule(i) \mid$
    $D_{Position} =\mid C * Rule_x(i+1) - Rule(i+1) \mid$
                                        ▷ Identifying prey's position
    if sigmoid $(D_{item}) <= RAND$ then
        Rule(i) = 1
    else
        Rule(i)=0
    end if
    if sigmoid $(D_{Position}) <= RAND$ then
        Rule(i+1) = 1
    else
        Rule(i+1)=0
    end if
    $i \leftarrow i + 1$
end while
Return $Rule$
```

Algorithm 3 Mutation Algorithm

```
Input: Rules (X1, X2, X3)
Output: NewRule (X)
$i \leftarrow 1$
while $i <= 2k$ do
                                                  ▷ Mutate the rules
    if $(rand < 0.33)$ then
        X(i) = X1(i)
        X(i+2) = X1(i+2)
    else
        if $(rand < 0.66)$ then
            X(i) = X2(i)
            X(i+2) = X2(i+2)
        else
            X(i) = X3(i)
            X(i+2) = X3(i+2)
        end if
    end if
    $i \leftarrow i + 2$
end while
Return NewRule
```

5 Results and discussions

In order to show the efficiency of the presented proposal, several experiments were carried out on different and well-known datasets in the field, which are described in the next section. After that, we present a comparative study in-face-of recently developed methods. To make the comparison totally fair, all algorithms are written in Java and executed on Intel Core I5 machine with 4 GB of memory running under Linux Ubuntu. Also, each algorithm is used with its best parameters recommended in the original paper.

5.1 Benchmark and setup description

With the aim to test and compare our proposed algorithm BGWORM, we use seven well-known benchmarks, that are frequently used in data mining community, from numerous and well-known sources in data mining field, such as Frequent and mining dataset repository [28] and Bilkent University function approximation repository [29]. Table 1 shows the datasets utilized in our experiments. Moreover, we observe from Table 1 that benchmarks vary in terms of transactions number and elements in each one. For example, connect dataset has 100,000 records with 999 items, whereas BMS-WebView-1 has fewer transactions and items.

Table 1. Benchmarks description

Dataset	Transactions size	Item size
IBM-Stand	1 000	20
Quack	2 178	4
Chess	3 196	37
Mushroom	8 124	119
Pumbs-star	40 385	7 116
BMS-WebView-1	59 602	497
Connect	100 000	999

5.2 Stability study

As mentioned above, one of the most known advantages of Grey Wolf Optimizer is the minimum parameters number which are mainly, number of wolves in the pack and number of iterations. In order to choose the best parameters, the comparison study is presented in the section, which will prove the efficiency of our proposal. With this in mind and to analyze the behavior of our method as a stochastic evolutionary algorithm, in this section, we mainly aim to look into our algorithm (BGWOARM) stability and how the algorithm deals with the objective function and CPU time when we vary numbers of wolves in the pack and iterations. In these tests, we utilize five datasets (IBM-Stand, Quack, Chess, Mushroom, Connect).

Table 2 presents the results attained by the execution of BGWOARM with varying wolves' numbers in packs regularly from 10 to 50. Results in Table 2 were obtained with a fixed number of iterations equal to 200. It is observed that the best results are achieved with 10 wolves in a pack in most datasets. Whereas, with Mushroom and Connect datasets, the best fitness becomes acceptable starting from 30 wolves. This observation can be clarified by the transactions number and items in these datasets. On the other hand, we can note that CPU time grows with the iterations increment, which is a natural behavior of each swarm-based algorithm.

Table 2. Evaluation of the GWOARM with several numbers of Wolves

Dataset	**Wolves**	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
IBM-Stand	**Fitness**	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96
	CPU time	0,48	0,65	0,85	1.11	1,30	1,52	1,70	1,91	2,11
Quack	**Fitness**	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	CPU time	0,37	0,46	0,60	0,69	0,84	0,96	1,12	1,25	1,37
Chess	**Fitness**	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99
	CPU time	1,02	1,42	1,86	2,24	2,81	3,21	3,84	4,53	4,78
Mushroom	**Fitness**	0,76	0,80	0,82	0,86	0,92	0,90	0,83	0,97	0,92
	CPU time	1,54	2,35	3,46	4,17	5,33	5,45	6,24	6,94	8,48
Connect	**Fitness**	0,72	0,81	0,85	0,80	0,96	0,95	0,93	0,96	0,94
	CPU time	27	42	57	72	85	103	111	126	155

On the other hand, the number of iterations is a substantial parameter for Grey Wolf Optimizer, which has an impact on algorithm stability and execution time. On this basis, we repeated our tests by fixing the number of wolves to 30 and varying the number of iterations from 100 to 900, regularly. Table 3 illustrates the results achieved by our algorithm. From the outcomes, the best values of the objective function were obtained starting from 300 iterations with all the datasets. This can be a sign that the best parameters for our algorithm are 30 and 300 for the number of wolves and iterations, respectively.

Table 3. Evaluation of the BGWOARM with several numbers of Iterations

Dataset	**Iterations**	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**
IBM-Stand	**Fitness**	0,77	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98
	CPU time	0,83	1,57	2,28	2,96	3,67	4,42	5,26	5,73	6,55
Quack	**Fitness**	1	1	1	1	1	1	1	1	1
	CPU time	0,42	0,47	1,08	1,42	180	2,16	2,57	288	3,22
Chess	**Fitness**	0,84	0,92	0,96	0,96	0,97	0,98	0,96	0,99	0,98
	CPU time	1,18	2,27	3,52	4,68	5,97	7,11	8,43	9,34	10,65
Mushroom	**Fitness**	0,55	0,76	0,69	0,73	0,77	0,64	0,73	0,78	0,76
	CPU time	2	5	7	10	12	14	17	19	22
Connect	**Fitness**	0,62	0,75	0,95	0,89	0,97	0,97	0,98	0,95	0,97
	CPU time	44	91	144	178	223	268	308	353	402

5.3 Comparison against similar approaches

In order to prove the effectiveness of our approach, a series of comparisons were carried out with recently developed algorithms in the field of rule mining by fixing the algorithm parameters to the best values detected from the stability study, 30 and 300 for the number of wolves and iterations, respectively.

The outcomes from the binary gray wolf optimizer for ARM were compared against the following algorithms: Whale Optimization Algorithm for ARM (WO-ARM) [8], Bat algorithm for ARM (BAT-ARM) [23], Bees swarm optimization algorithm for ARM (BSO-ARM) [30], Penguins Search Optimization Algorithm for ARM (Pe-ARM) [22], and multi-swarm bat algorithm for ARM (MSB-ARM) [7].

The outcomes illustrate the overage obtained by 20 executions for every algorithm. Table 4 presents the results obtained in our tests by each algorithm in terms of CPU time consumption with four middle size benchmarks. It is clearly noted that BGWOARM outclasses the other algorithms. The exception is with IBM-Stand datasets where the Whale Optimization algorithm outguesses our proposal with 0.3 seconds, which is negligible. From the previous section, we observe that the number of iterations can affect on the CPU time in direct proportion, which is the case for all swarm-based algorithms. Based on this, a comparison study for our proposal in the face of BAT-ARM and MSB-ARM was carried out. Results are presented in Figure 5, in which the evolution of CPU time in terms of iteration number is shown. Results prove the efficiency of BGWOARM. Also, we can note the reduced time consumed by the proposed algorithm over all datasets.

Table 4. Comparing our approach to existing approaches w.r.t Time (sec)

Dataset	**Pe-ARM**	**BSO-ARM**	**MSB-ARM**	**BAT-ARM**	**WO-ARM**	**BGWO-ARM**
IBM-Stand	1.68	1.92	13	19	**1.2**	1.57
Quack	3.35	4.5	40	76	2.3	**1.08**
Chess	4.92	5.1	13	141	7.7	**3.52**
Mushroom	10.68	9.1	144	341	10	**7**

Actually, CPU time is an important fact to judge an evolutionary algorithm, but it’s not enough. The optimal value of the fitness function is also critical which describes the solution quality, in our tests, the fitness function aims to maximize two main measures in the ARM field which are support and confidence. With this in mind, we compare our results in the face of other swarm-based algorithms in terms of maximum fitness function values. The outcomes illustrate the superiority of BGWOARM against other algorithms with all the datasets.

Table 5 presents the outcomes of our comparison in terms of fitness function values. On another side, swarm-based algorithm needs to explore the search space to extract the best rules, which needs to extract the maximum number of rules from the dataset that satisfied the minimum threshold support and minimum threshold confidence introduced by the final user. So, another comparison is accomplished.

Figure 6 and Figure 7 summarize the evolution of the number of generated rules from five different datasets in terms of minimum sup-

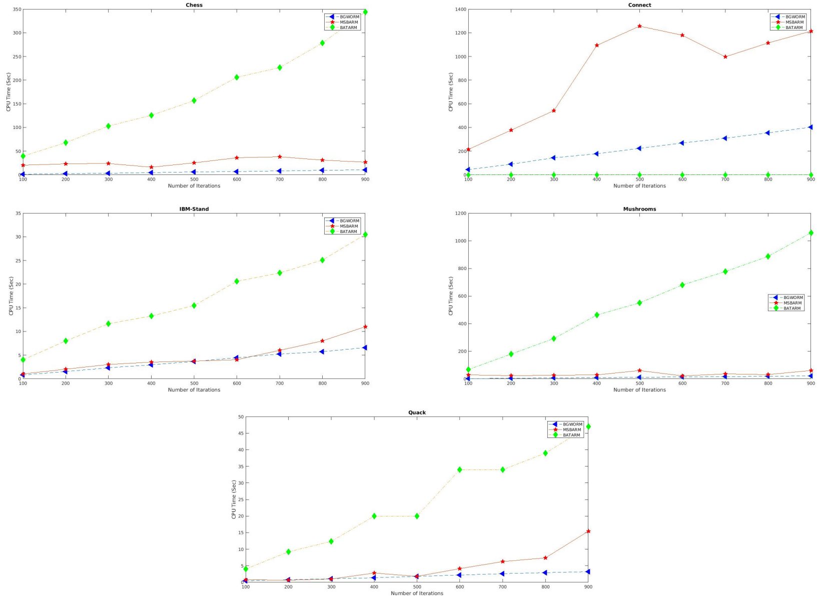


Figure 5. CPU time performance in terms of the iterations number.

port and minimum confidence, respectively. It can be noticed that the proposed algorithm outperforms the other algorithms in the majority of cases. This superiority can be argued by the great exploration of the search space extended from the original gray wolf algorithm.

Table 5. Comparing our approach to existing approaches, w.r.t Fitness

Dataset	**Pe-ARM**	**BSO-ARM**	**MSB-ARM**	**BAT-ARM**	**WO-ARM**	**BGWO-ARM**
IBM-Stand	0.92	0.93	0.84	0.41	0.94	**0.98**
Quack	0.91	1	1	0.52	1	1
Chess	0.89	0.88	0.97	0.92	0.99	**0.99**
Mushroom	0.88	0.75	0.68	0.93	0.93	**0.97**

Therefore, as it is shown in these results, the binary gray wolf optimizer for ARM generated rules with competitive measures in terms of support and confidence compared to similar algorithms. Also, the results prove the superiority of the proposed algorithm in terms of CPU time against the other algorithms. These results were obtained thanks

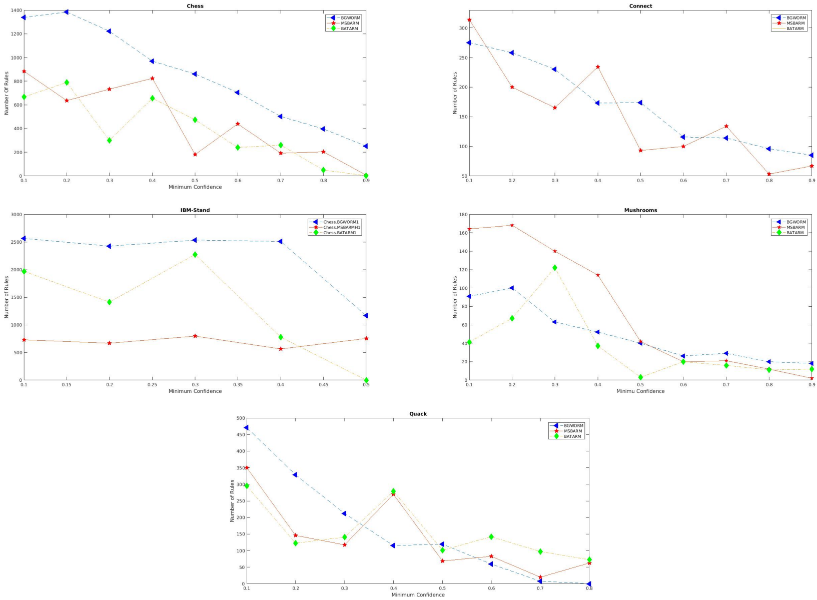


Figure 6. Number of generated rules performance in terms of minimum support.

to the newly introduced binary encoding of the rules, which reduces the algorithm complexity. Moreover, the original gray wolf optimizer algorithm simplicity, power exploitation, and exploration have been bequeathed to our algorithm.

5.4 Comparison against exact approaches

With the aim to discover the coverage rate of our proposal on the datasets, we calculate the proportion of valid rules generated (PVR). This proportion is calculated based on the full number of valid rules generated by the exact exhaustive algorithms (Apriori [3], FP-Growth [14]). PVR is defined by the following rule:

$$PVR = 100 * \frac{Number\ of\ generated\ rules}{Total\ number\ of\ Valid\ rules}. \qquad (13)$$

The obtained results are the overage of twenty executions on

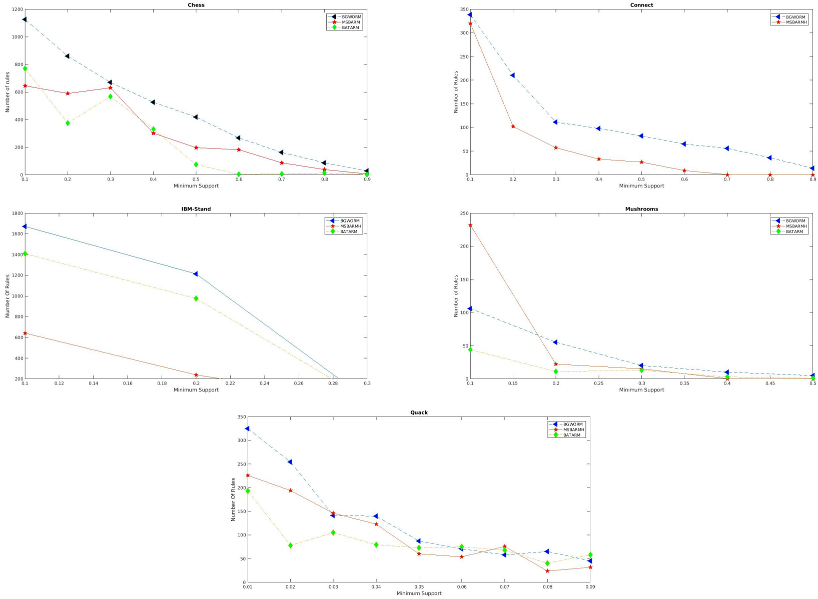


Figure 7. Number of generated rules performance in terms of minimum confidence

medium size datasets that have more than 40 000 transactions. Table 6 presents how the CPU time varies w.r.t different datasets. Again, it is clearly observed that BGWOARM surpasses the exact algorithms in terms of CPU time, thanks again to the fast search mechanism of the gray wolf algorithm and the new binary encoding proposed for the rules. Table 7 illustrates the percentage of valid rules relative to diverse benchmarks. We noticed that our method proves its superiority against MSBARM and BSO-ARM in terms of PVR, which exceeds 70% for all the datasets. Furthermore, the CPU time consumption of the proposed method is very small in the face of exhaustive approaches, whereas the PVR is not less than 70%. These results prove the power and the necessity of optimization approaches instead of exact ones.

According to the obtained results, it can be noted the BGWO-ARM superiority against other approaches. This outperforming in terms of CPU time and number of generated rules can be explained by the

Table 6. Comparing our approach to exact approaches w.r.t CPU time (sec)

Datasets	**BGWO-ARM**	**MSB-ARM**	**BSO-ARM**	**FP-Growth**	**Apriori**
Pumbs-star	307	315	**300**	600	500
BMS-WebView-1	**272**	348	400	850	1100
Connect	**402**	1094	950	2900	2600

Table 7. Comparing our approach to exact approaches w.r.t PVR (%)

Datasets	**BGWO-ARM**	**MSB-ARM**	**BSO-ARM**	**FP-Growth**	**Apriori**
Pumbs-star	**70**	54	60	100	100
BMS-WebView-1	**81**	46	62	100	100
Connect	**77**	49	55	100	100

low complexity of our approach, which comes from the original GWO. Moreover, the new generation method based on the sigmoid function can lead our mining process to the best rules. Also, the mutation operator has an important role in the generated rule quality based on three fittest rules α, β, and γ. On the other hand, we can observe that BGWO-ARM generates maximum of valid rules thanks to the good exploration of the search space, inherited from GWO, and the exploitation to extract the best local rules.

6 Conclusion and Future Works

In this paper, a new binary grey wolf optimizer with mutation for mining ARs in large database, called BGWOARM, has been presented. The proposed algorithm used a bitmap representation for the database, which reduces runtime and simplifies rule measures calculation. Moreover, a new rule generation method based on the sigmoid function is introduced, which produces a powerful rule generator. Afterward, the mutation algorithm is applied to generate the fittest candidate rule. The BGWOARM performances have been compared to five similar approaches recently published in the field of ARM in terms of quality, number of rules, and run time. Results proved the efficiency of the proposal, and that it outperformed these methods within most experiments. Moreover, our results are compared in the face of the classic methods in terms of rule validity, which shows the efficiency of the method in search space exploration. The technique must be upgraded and evaluated with a massive database. We also plan to further parallelize the technique and implement it on a GPU to improve both the quality of the solution and its execution time.

References

[1] I. Bose and R. K. Mahapatra, “Business data mining—a machine learning perspective,” *Information and management*, vol. 39, pp. 211–225, 2001.

[2] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques.* Elsevier, 2011.

[3] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” vol. 22, 1993, pp. 207–216.

[4] A. Telikani, A. H. Gandomi, and A. Shahbahrami, “A survey of evolutionary computation for association rule mining,” *Information Sciences*, vol. 524, pp. 318–352, 2020. [Online]. Available: https://doi.org/10.1016/j.ins.2020.02.073

[5] X. Yan, C. Zhang, and S. Zhang, “Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support,” *Expert Systems with Applications*, vol. 36, pp. 3066–3076, 2009.

[6] Z. Kou and L. Xi, “Binary particle swarm optimization-based association rule mining for discovering relationships between machine capabilities and product features,” 2018. [Online]. Available: https://doi.org/10.1155/2018/2456010

[7] K. E. Heraguemi, N. Kamel, and H. Drias, “Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies,” *Applied Intelligence*, vol. 45, pp. 1021–1033, Dec 2016.

[8] K. Heraguemi, H. Kadrii, and A. Zabi, “Whale optimization algorithm for solving association rule mining issue,” *International*

Journal of Computing and Digital Systems, vol. 10, pp. 2210–142, 2021. [Online]. Available: http://journals.uob.edu.bh

[9] K. E. Heraguemi, N. Kamel, and H. Drias, "Multi-objective bat algorithm for mining numerical association rules," *International Journal of Bio-Inspired Computation*, vol. 11, pp. 239–248, 2018.

[10] E. V. Altay and B. Alatas, "Performance analysis of multi-objective artificial intelligence optimization algorithms in numerical association rule mining," *Journal of Ambient Intelligence and Humanized Computing*, 2019. [Online]. Available: https://doi.org/10.1007/s12652-019-01540-7

[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar 2014.

[12] Z. Abderrahim, K. E. Herraguemi, and M. Sabir, "A new improved variable step size mppt method for photovoltaic systems using grey wolf and whale optimization technique based pid controller," *Journal Europeen des Systemes Automatises*, vol. 54, pp. 175–185, Feb 2021.

[13] Q. Al-Tashi, H. M. Rais, S. J. Abdulkadir, H. Alhussian, and S. Mirjalili, "A review of grey wolf optimizer-based feature selection methods for classification," pp. 273–286, 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-32-9990-0_13

[14] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM sigmod record*, vol. 29, no. 2, pp. 1–12, 2000.

[15] S. M. Ghafari and C. Tjortjis, "A survey on association rules mining using heuristics," *WIREs Data Mining and Knowledge Discovery*, vol. 9, Jul 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1307

[16] H. Drias, "Genetic algorithm versus memetic algorithm for association rules mining," *2014 6th World Congress on Nature and Biologically Inspired Computing, NaBIC 2014*, pp. 208–213, Oct 2014.

[17] A. Derouiche, A. Layeb, and Z. Habbas, "Mining interesting

association rules with a modified genetic algorithm," *Pattern Recognition and Artificial Intelligence*, vol. 1322, p. 274, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7972016/

[18] R. J. Kuo, C. M. Chao, and Y. T. Chiu, "Application of particle swarm optimization to association rule mining," *Applied Soft Computing*, vol. 11, pp. 326–336, 2011.

[19] K. Sarath and V. Ravi, "Association rule mining using binary particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 1832–1840, 2013.

[20] A. Derouiche, A. Layeb, and Z. Habbas, "Chemical reaction optimization metaheuristic for solving association rule mining problem," vol. 2017-Octob. IEEE Computer Society, 3 2018, pp. 1011–1018.

[21] Y. Djenouri, H. Drias, and Z. Habbas, "Bees swarm optimisation using multiple strategies for association rule mining," *International Journal of Bio-Inspired Computation*, vol. 6, pp. 239–249, 2014.

[22] Y. Gheraibia, A. Moussaoui, Y. Djenouri, S. Kabir, and P. Y. Yin, "Penguins search optimisation algorithm for association rules mining," *CIT. Journal of Computing and Information Technology*, vol. 24, pp. 165–179, 2016.

[23] K. E. Heraguemi, N. Kamel, and H. Drias, "Association rule mining based on bat algorithm," *Journal of Computational and Theoretical Nanoscience*, vol. 12, no. 7, pp. 1195–1200, 2015.

[24] K. E. Heraguemi, N. Kamel, and H. Drias, "Multi-population cooperative bat algorithm for association rule mining," in *Computational collective intelligence*, 2015, pp. 265–274.

[25] U. Mlakar, M. Zorman, and I. Fister, "Modified binary cuckoo search for association rule mining," vol. 32, 2017, pp. 4319–4330.

[26] P. N. Tan, V. Kumar, and J. Srivastava, "Selecting the right objective measure for association analysis," vol. 29, 2004, pp. 293–313.

[27] A. A. Freitas, *Data mining and knowledge discovery with evolutionary algorithms.* Springer Science & Business Media, 2002.

[28] B. Goethls and M. J. Zaki, "Frequent itemset mining dataset repository," 2003. [Online]. Available: http://fimi.ua.ac.be/data/

[29] H. A. Guvenir and I. Uysal, "Bilkent university function approximation repository," 2000. [Online]. Available: http://funapp.cs.bilkent.edu.tr/DataSets/

[30] Y. Djenouri, H. Drias, Z. Habbas, and H. Mosteghanemi, "Bees swarm optimization for web association rule mining," vol. 3, 2012, pp. 142–146.

KamelEddine Heraguemi, Nadjet Kamel,
Majdi M. Mafarja

Received December 10, 2023
Accepted January 26, 2024

KamelEddine Heraguemi
ORCID: https://orcid.org/0000-0001-6992-5536
The Networks & Distributed Systems Laboratory.
National School of Artificial Intelligence
Algiers, Algeria
E–mail: kameleddine.heraguemi@ensia.edu.dz

Nadjet Kamel
ORCID: https://orcid.org/0000-0003-3608-8895S
The Networks & Distributed Systems Laboratory.
University Setif1 Ferhat Abbas.
Sétif, Algeria
E–mail: nkamel@univ-setif.dz

Majdi M. Mafarja
ORCID: https://orcid.org/0000-0002-0387-8252
Department of Computer Science, Faculty of Engineering and Technology, Birzeit University
Birzeit, Palestine
E–mail: mmafarja@birzeit.edu

A Coloured Petri Net-based approach and Genetic Algorithms for improving services in the Emergency Department

Zouaoui Louhab, Fatma Boufera

Abstract

The Emergency Department (ED) plays an important role in the healthcare field, due to the nature of the services it provides, especially for patients with urgent cases. Therefore, good management of ED is very important in improving the quality of services. Good management depends on the effective use of material and human resources. One of the most common problems that the ED suffers from is the long waiting period and the length of the patient's stay. Many researchers have proposed many solutions to reduce waiting time and length of stay (LOS). One of the best solutions for resource optimization is modeling and simulation based on inputs such as patient length of stay and door-to-doctor time (DTDT). In this study, the ED was modeled using a Coloured Petri Net, and to determine the number of resources needed, genetic algorithms were used. This study was conducted in the ED of Hassani Abdelkader Hospital in Sidi Bel Abbes, and several simulation models were evaluated, which reduced the waiting time and the length of stay for the patient.

Keywords: emergency department (ED), system modelling, Coloured Petri Net, genetic algorithms, optimization.

MSC 2020: 68T50.

1 Introduction

Health is the basic asset of human beings, and the healthcare system usually meets the health needs of the population of any country.

doi:10.56415/csjm.v32.07

Hospitals are the most important component of the healthcare service network, and the emergency department (ED) is an important component of the hospital [1]. As the years pass, the population increases, new diseases appear with different symptoms, the difficulty of predicting critical situations, and the limited resources, all these obstacles stand in the way of decision-makers [2]. The patient arrives at the ED in different cases, from critical, non-critical, and semi-critical, to other cases. The models for classifying patients change from one country to another, and the organizational structure may be variable from one country to another, but it shares several characteristics, such as staff nurses, treatment rooms, and medical and laboratory equipment. The process of patient flow is as follows: registration, triage Medical examinations, and additional tests [3].

Among the most common problems is the long waiting time, this problem leads to dissatisfaction among patients, as the patient is always looking for the quality of service. Time is a valuable asset for the patient in seeking treatment in any center, whether private or public, especially for patients with special cases [5].

The ED has recently faced several problems, which negatively affect the workflow. Among the problems that the ED suffers from is the random flow of patients, which causes overcrowding, and this leads to the length of the patient's stay in the ED [6]. The ED is the main entrance to the hospital; despite this, it suffers from several limitations, for example, limited resources and budget. For this purpose, those in charge of the ED work to increase the capacity of the ED to accommodate the largest number of patients by increasing the number of material and human resources [7].

All these factors attracted the attention of many researchers in the field of healthcare and led to a reconsideration of the work policy and the structure of the ED [8]. To improve the quality of service in the ED, the researchers relied on many approaches, where different techniques were used such as demand management, mapping, queuing systems, agent-based systems, and simulation [9].

The ED depends on employing highly qualified doctors and nurses, who are able to face any emergency that may occur at any time because the time factor plays a major role in saving lives [10]. The outbreak of the COVID-19 virus led to many problems for the ED, as it greatly affected the workflow of the medical staff, which led to a review of the ED management policy [11].

Nowadays, simulation is the most widely used method in healthcare management. Simulations have been successfully used in many different areas such as the medical sector, manufacturing, system services, supply chain, transportation, etc. In addition, the simulation method is one of the best ways for decision-makers to evaluate, analyze, and review any operating systems from the simplest to the most complex in order to solve the problems they contain [5]. The most important key element of the simulation is the selection of a key performance indicator (KPI). Although there are no metrics that define a KPI, the most commonly used in this field are average waiting times, length of patient stay (LOS), and the number of patients who leave the ED without medical and nursing advice [3].

In this study, we do a case study of the ED of Sidi Bel Abbes hospital. System modeling is required due to the complexity of the study system. For this purpose, we have relied on a Coloured Petri Net, which is more used to model complex and synchronous systems. Coloured Petri Net tools are used in the simulation in order to simulate the flow of patients and make adjustments to the system before applying it in its real environment. This research is organized as follows. In Section 2, we look at some of the literature studies conducted in emergency departments. In Section 3, the paths that patients take in the ED are explained. In Section 4, we present a research methodology that includes an explanation of KPIs, a brief definition of Coloured Petri Net, a presentation of simulation model, and the use of genetic algorithms to identify resources in the emergency department. In subsection 4.5, simulation results for the proposed models are explained and discussed. In the last section, we present a general conclusion of the research with proposing future solutions.

2 Background and Literature review

Nowadays, several studies have been established based on simulation [11, 12, 13], especially in the field of healthcare. Simulation has become a wide field of research, especially with the ED [14]. Simulations are of great help in several issues such as process identification, process modeling, identification of patient profiles, planning, mastering and optimizing flows, and trial-by-performance [15].

For optimal resource planning in emergency departments, Yousefi et al. (2018) [4] relied on genetic algorithms; in order to solve the problem of overcrowding and patient length of stay, agent-based simulation and machine learning were relied on. Scheduling patient admission is a major problem for the emergency department, and Ceschia and Schaerf (2012) [16] worked on this problem by proposing solutions while considering many real-world features, such as uncertainty in length of stay, presence of emergency patients, and the possibility of late admission. A metaphysical approach is proposed that solves both versions Static and dynamic, depending on the complex neighborhood structure and simulated annealing. This approach gave good results, as it was possible to schedule and accept a large number of patients.

In another work carried out by Ceschia and Schaerf (2016) [17] for the patient's admission process, the scheduling problem was reconsidered in order to be suitable for practical applications. The restrictions imposed on the use of operating rooms for critically ill patients, who require surgery, were taken into consideration. A new model is proposed that includes a flexible planning horizon, new components of the objective function, and a complex concept of patient delay. The local search method is based on the use of a search space based on a complex neighborhood. Statistical distributions and real-world data are used to compile challenging and realistic case studies. Allocating beds in emergency departments requires careful scheduling from the medical staff.

Demeester et al. (2010) [18] have developed a hybrid algorithm based on Tabu search algorithms, the aim of which is to appropriately allocate beds in emergency departments, taking into account the medical needs of patients. Simulation modeling is one of the best methods used for improvement in ED. This method is relied on by Nas and Koyuncu (2019) [1] for improvement. The aim is to optimally determine the number of beds. ED data is analyzed and arrival times and features related to the patient's arrival at the hospital are studied. To analyze this data, ten algorithms for machine learning (ML) were used. The proposed model gave good results, reduced LOS by 7%, and improved the number of beds in ED.

There are a variety of methods and techniques used for improvement in healthcare. Among the methods used is a simulation; the latter has been relied upon by many researchers, including Kittipittayakorn and Ying (2016) [19]. A simulation model was created with the aim of reducing the long waiting time of the patient, the patient's behaviors were studied and modeled to be incorporated into the simulation, and huge data was used. This study gave good results that led to a significant reduction in waiting time.

Simulation has a major role in improving. Gül and Guneri (2012) [20] created a simulation model for discrete events for the emergency department, through which all daily operations were modeled and analyzed. The goal of this work is the optimal use of human and material resources. Another work done by Lamiri et al. (2009) [21] uses mixed integer programming with Monte Carlo simulation. This work aims to organize the surgical operations in the emergency department well.

In a study conducted by Yousefi et al. (2018) [4] in the ED, using agent-based simulation modeling, medical staff allocate human resources to different departments based on experience, or depending on decision support tools. In this study, all staff agents participate in the decision-making process. In modeling, all the components belonging to the ED take part, including patients, doctors, and staff. To assess the behavior of the system, several scenarios were evaluated, and in each scenario, the KPIs were evaluated. Agent-based simulations led to well-organized operations in the emergency department, reducing waiting time.

To study the relationship between the ED and other departments, Yousefi et al. (2017) [3] proposed an integrated approach based on Discrete Event Simulation (DES) and System Dynamics (SD), through which the complexities and interactions between sub-units and emergency departments are revealed. This simulation is known as hybrid simulation, which helps to better understand the ED operations.

3 Patient Management Flowchart

In Figure 1, we show all the paths that the patient follows in the emergency department. Based on this diagram, we build a simulation model. The patient arrives at the ED either by his own means or by ambulance. The patient undergoes the triage process by the triage nurse. Based on the patient's condition, two cases are classified, critical and non-critical. In this study, we rely on the emergency severity index (ESI); the critical condition is classified in the first level, and non-critical cases are classified in other levels (5, 4, 3, and 2).

Patients in critical condition are transferred directly to the operating room. If the surgery is successful, the patient remains in the recovery room and then in the short stay unit for a period of time. It does not exceed 24 hours, then the patient is transferred to another department. Some patients in a non-serious condition need a nursing consultation, while the majority of patients are referred to a normal medical consultation; some cases need a special medical consultation. During a private medical consultation, most cases need additional tests. After obtaining the results of the additional tests, the specialist Doctor decides to either direct the patient to another department of the hospital or to be discharged from the emergency department.

4 Methodology

In this section, we will explain the methodology in detail. After a good study of the behavior and characteristics of the system and using the expertise of the medical staff, we model the ED through a Coloured Petri Net and develop a genetic algorithm that helps us greatly to determine the number of resources used in the emergency department by entering new resources into the model and comparing the results.

4.1 Key performance indicator

To evaluate any system, we use KPIs, which the system has as its own characteristics. In this study, we evaluate the ED using door-to-doctor time (DTDT) and length of stay (LOS). These are the most commonly used KPIs in healthcare. Below we explain these indicators:

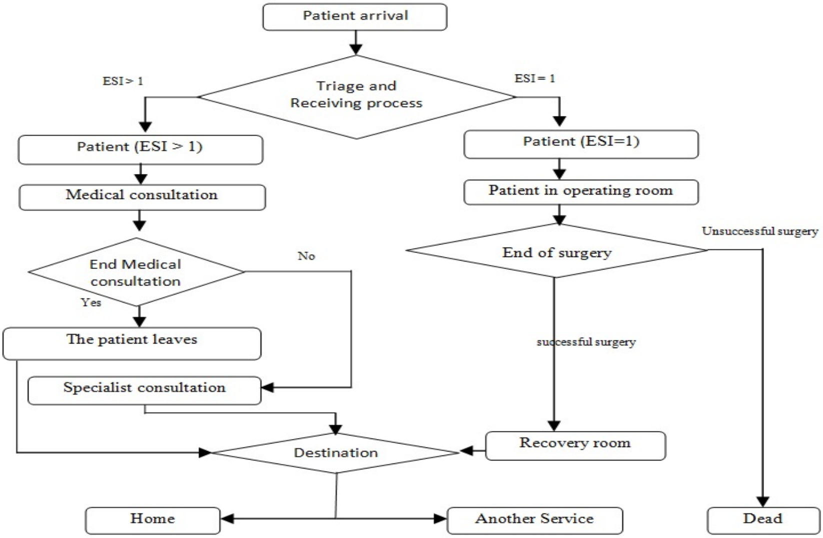


Figure 1. Patient path in the emergency department

- Length of stay (LOS): The time a patient spends in the emergency department, from admission to departure at home, or to another section of the hospital.

- Door-to-doctor time (DTDT): The amount of time that a patient spends from entering the ED to the first medical consultation.

4.2 Simulation model

The simulation model was created by our study of the behavior of the system, and with the help of medical and administrative staff. Figure 2 shows the basic elements of the model. The most important basic elements of the model are the triage process, general and specialist medical consultation, additional tests, surgery, and patient orientation. Through the model, each place represents the situation in which the patient may be. The paths that the patient takes vary according to the condition, from critical to non-critical, nursing, medical and special consultation. Table 1 shows the places of the simulation model, Table

2 shows the transitions. Patient access to the ED is designed by a token in the place the patient has accessed. This place is characterized by a set of PATIENT colors. The latter includes a set of attributes for calculating the duration of operations for all stages, waiting times, LOS and DTDT. In this model, we use P and P2 as variables of a PATIENT type. These variables are used as inputs and outputs for the transitions' functions. Patient access is modeled by an exponential distribution function with a mean of 8.5, this parameter is calculated based on the data collected. ESI is used on each patient in the triage process.

The simulation model includes several transitions; each transition is characterized by four properties, the name of the transition, the delay expression, the guard, and the code segment expression. After recording the patient's data, he undergoes the triage process to be classified into two categories, critical and non-critical cases. We use several functions in the model such as the triage function (Reception_time), and the function of calculating the duration of medical advice (MC_Time). The functions are used to calculate times at different stages as well as update the characteristics of a patient's color set. The tokens during their transitions may be subject to a guard function, for example, according to ESI values, the patient is graded. The model is validated by sharing the opinions of clinicians and staff and comparing the output of the model Simulation with real data.

4.3 Resource identification using the genetic algorithm model

In this section, we use genetic algorithms to determine the number of appropriate human resources in the ED. We rely on calculating the average period of time that the patient spends in the following four stages: nursing consultation, medical consultation, specialized medical consultation, and additional tests. The average total time period for the patient is the sum of the four time periods mentioned above. The initial population represents the number of human resources for each emergency phase. Figure 3 shows an example of the initial population:

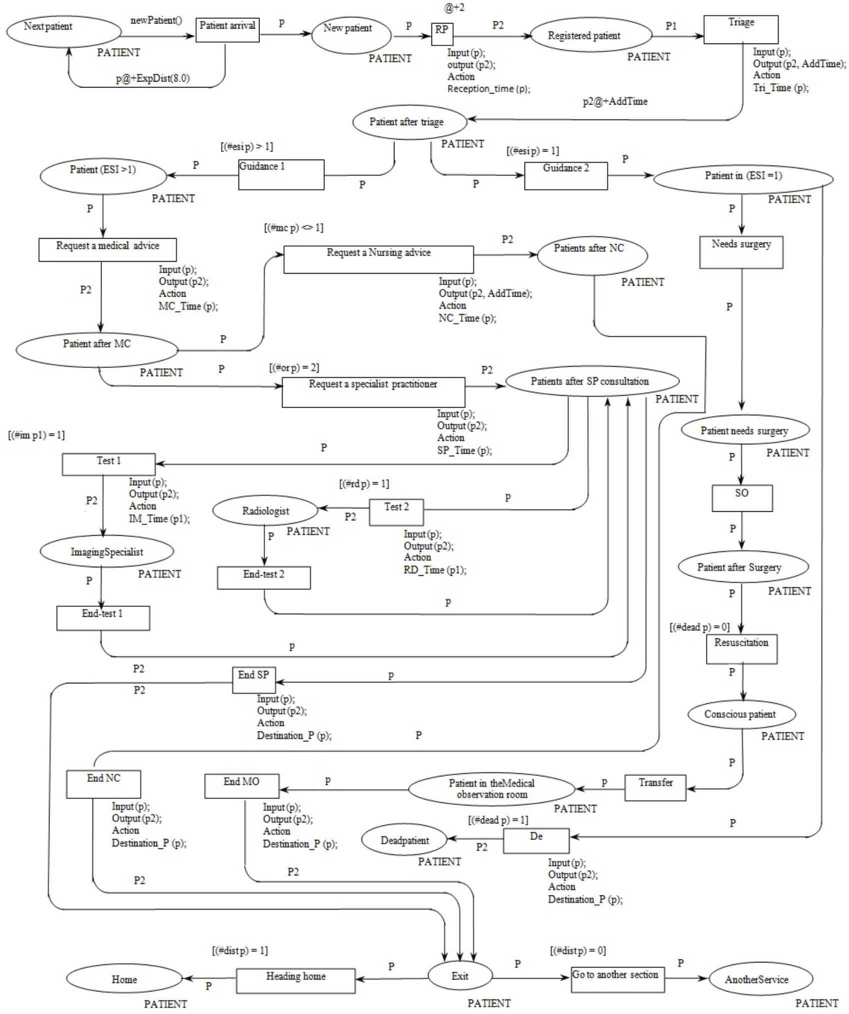


Figure 2. Simulation model using Coloured Petri Nets for ED

Table 1. Place description of the simulation model

Place	Description
New patient	Patient coming to the emergency department
Registered patient	A patient registered in the reception register
patient after triage	The patient after the screening process with his classification according to the case
Patient (ESI 1)	A patient in a non-critical condition
Patient in (ESI =1)	A patient is in a critical condition
Patient after MC	Patient after medical consultation
Patients after NC	Patient after nursing consultation
Radiologist	Patient at the radiologist
Imaging Specialist	Patient at the Imaging Specialist
Patient needs surgery	Patient in the operating room
Patient after Surgery	Patient in the first recovery room
conscious patient	A patient in the second recovery room
Patient in theMedical observation room	Patient in the medical observation room
Dead patient	A patient died after a failed surgery
Exit	A patient finished his visit to the emergency department
Home	The patient goes home
Another Service	The patient goes to another section

Table 2. Description of simulation model transitions

Transition	Description
Patient arrival	Admission of the patient to the emergency department
RP	Registration of the patient in the emergency department
Triage	Triage process
Guidance 1, Guidance 1	Directing the patient to the various sections in the emergency department
Request a Nursing advice	
Request a specialist practitioner	The beginning of a specialized medical consultation
needs surgery	Beginning of the nursing consultation
Test 1, Test 1	Additional exams start
End-test 1, End-test 2	End of additional exams
SO	Surgery process
Resuscitation	Resuscitation process
Transfer	Transfer the patient to the medical observation room
End NC, End SP, End MO	The end of medical and nursing consultations
De	The death of the patient

$$Min\ LOS = \sum_{i=1}^{N} A\_los(x(i)). \tag{1}$$

- N : Number of Resources (For the case studied N=4).
- i : Index of a Resource; i = 1; 2; : : : N;
- x(i) : Resources available at every ED stage, for example: if x(1) = 2, we have two nurses in the nursing consultation stage. For the case studied, x(1) represents the number of nurses, x(2) – the number of general doctors, x(3) – the number of specialist doctors, and x(4) – the number of radiologists.
- *A_los*: A function that calculates the average length of time a patient spends in the resource i. The goal of this function is to find the lowest value of LOS, with the appropriate number of Human Resources. We run the genetic algorithms three times, and each time we obtain new solutions. Tables 4, 5, and 6 represent the results obtained by implementing the genetic algorithms. We modify the number of Human Resources obtained in the simulation model, run the simulation model again, and compare Results every time.

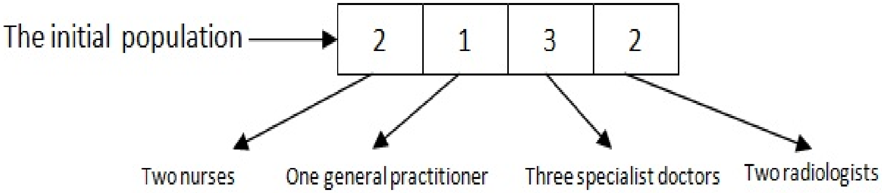


Figure 3. Example of the initial population

4.4 Simulation Results and Discussion

During the study of the behavior of the system, we noted that the most exploited resources in the ED are general and private medical advice, nursing, and additional tests. As for the reception and sorting resource, the time period remains constant and small. The goal of this study is to reduce LOS and DTDT. After developing the prototype by means of a Coloured Petri Net, we ran it for the first time and recorded the initial data. The Coloured Petri Net model is run by CPN tools, which are tools used to simulate Coloured Petri Net models. We run genetic algorithms three times, and each time we get new data. The goal of running genetic algorithms is to obtain the lowest LOS values with an appropriate number of Human Resources.

Table 3. Simulation results for the proposed models

	Benchmark model	First simulation model	Second simulation model	Third simulation model
Waiting time for a nursing consultation	76.5	79.5	52.2	55.8
Waiting time for a medical consultation	62.8	69.3	42.4	42.5
Waiting time for a specialist consultation	79.5	30.5	85.3	85.5
Waiting time for additional tests	59	36.7	27.4	37
Nursing consultation	16.3	17.0	15.6	15.1
Medical consultation	14.0	13.5	13.6	13.9
Specialist consultation	17.9	16.2	17.1	14.8
Additional tests	12.5	14.0	13.4	14.5
LOS	344.3	281.5	267.8	279.9
DTDT	78.5	82.4	57.5	59.7

Tables 4, 5, and 6 represent the results obtained by running genetic algorithms. The first four columns of Tables 4, 5, and 6 are the number of human resources obtained, the fifth column represents the

Table 4. First execution of the GA algorithm

Nurse	General Practitioner	Specialist Practitioner	Additional tests	Min Los
1.5e+00	1.1e+00	2.7e+00	2.2e+00	1.5e+02
1.5e+00	1.2e+00	2.8e+00	2.1e+00	1.53+02
1.5e+00	1.3e+00	2.6e+00	2.2e+00	1.77+02
1.5e+00	1.4e+00	2.9e+00	2.3e+00	1.52e+02
1.5e+00	1.2e+00	2.6e+00	2.4e+00	1.51e+02
1.5e+00	1.1e+00	2.8e+00	2.3e+00	1.53e+02

Table 5. Second execution of the GA algorithm

Nurse	General Practitioner	Specialist Practitioner	Additional tests	Min Los
1.6e+00	1.3e+00	1.0e+00	1.8e+00	1.9e+02
2.5e+00	2.3e+00	3.6e+00	1.6e+00	1.3e+02
2.1e+00	1.8e+00	1.0e+00	2.9e+00	1.8e+02
2.1e+00	1.8e+00	1.00e+00	3.0e+00	1.5e+02
2.1e+00	1.8e+00	1.00e+00	3.0e+00	1.8e+02
2.1e+00	1.8e+00	1.1e+00	3.0e+00	1.5e+02

Table 6. Third execution of the GA algorithm

Nurse	General Practitioner	Specialist Practitioner	Additional tests	Min Los
1.8e+00	2.1e+00	1.5e+00	1.6e+00	1.5e+02
1.8e+00	2.1e+00	1.5e+00	1.6e+00	1.3e+02
1.2e+00	1.5e+00	1.4e+00	1.5e+00	1.9e+02
1.6e+00	2.1e+00	1.4e+00	1.7e+00	1.6e+02
1.6e+00	2.1e+00	1.3e+00	1.6e+00	1.9e+02
1.8e+00	2.1e+00	1.5e+00	1.6e+00	1.4e+02

calculated objective function. At the beginning, we pointed out that in the Benchmark model, each stage of the ED contains only one person, for example, one doctor, one nurse, etc. The first simulation model is built based on the results obtained in Table 4, while the second simulation model is built based on the results obtained in Table 5, and the third simulation model is built based on the results obtained in Table 6. Regarding the results obtained in Tables 4, 5, and 6, we take the approximate values of these results when entering them into the simulation models. We run the three models and compare the results with the benchmark model. In the second column of Table 3, we find the simulation results for the benchmark model. The third column of Table 3 shows the simulation results for the first model. The fourth column of Table 3 shows the simulation results for the second model. In the fifth column of Table 3, we find the simulation results for the third model.

Based on the results obtained in Table 3, we note that the LOS value decreased by 18.24% for the first simulation model compared to the Benchmark model. This is due to a decrease in waiting times for additional tests and specialized medical consultation, and this is due to the results obtained in Table 4. We notice in Table 4 the increase in the number of specialist doctors and radiologists. As for DTDT, we notice a slight increase, as human resources were not modified in the nursing consultation stage.

Regarding the second simulation model, we note that the LOS value decreased by 22.22% compared to the standard model. This is due to a decrease in waiting times for nursing consultation and additional tests, and this is due to the results obtained in Table 5, where we note the increase in the number of nurses and radiologists. As for DTDT, it decreased by 26.75%, due to the adjustment that occurred in the nursing consultation stage.

Regarding the third simulation model, we note that the LOS value decreased by 18.7% compared to the benchmark model, due to a decrease in waiting times in most stages; this is due to the results obtained in Table 6, where we note the increase in the number of general doctors, nurses, and radiologists. As for DTDT, it decreased by 23.94%, due to the adjustment in the number of human resources in each stage.

5 Conclusion

There have been many researches in the field of healthcare in recent years, especially in emergency departments. To determine resources in the emergency department, this study presented an approach based on modeling and optimization. The emergency department was modeled using a Coloured Petri Net, which is an effective tool widely used for systems modeling complex. Genetic algorithms are run several times in order to determine the appropriate amount of resources, each time new solutions are obtained. This study gave good results, through which several models are proposed, in each case both LOS and DTDT are reduced. This approach allows decision-makers in emergency departments to quickly find appropriate solutions, as this approach allows suggesting several models, which helps decision-makers to choose the appropriate model. In this study, we focused on one goal, which is to reduce LOS and DTDT. In future studies, we will add other objectives in order to improve the quality of service in emergency departments.

References

[1] S. Nas and M. Koyuncu, “Emergency Department Capacity Planning: A Recurrent Neural Network and Simulation Approach,” *Computational and Mathematical Methods in Medicine*, vol. 2019, pp. 1–13, Nov. 2019, DOI: 10.1155/2019/4359719.

[2] O. Derni, F. Boufera, and M. F. Khelfi, “An Advanced Heuristic Approach for the Optimization of Patient Flow in Hospital Emergency Department,” *International Journal of Intelligent Systems and Applications*, vol. 11, no. 9, pp. 29–39, Sep. 2019, DOI: 10.5815/ijisa.2019.09.04.

[3] M. Yousefi and R. P. M. Ferreira, “An agent-based simulation combined with group decision-making technique for improving the performance of an emergency department,” *Brazilian Journal of Medical and Biological Research*, vol. 50, no. 5, 2017, DOI: 10.1590/1414-431x20175955.

[4] M. Yousefi, M. Yousefi, R. P. M. Ferreira, J. H. Kim, and F. S. Fogliatto, “Chaotic genetic algorithm and Adaboost ensemble metamodeling approach for optimum resource planning in emergency departments,” *Artificial Intelligence in Medicine*, vol. 84, pp. 23–33, Jan. 2018, DOI: 10.1016/j.artmed.2017.10.002.

[5] A.F. Najmuddin, I.M. Ibrahim, and S.R. Ismail, “A Simulation Approach: Improving Patient Waiting Time for Multiphase Patient Flow of Obstetrics and Gynecology Department (O&G Department) in Local Specialist Centre,” *WSEAS Transactions on Mathematics*, vol. 9, n. 10, pp. 778–790, 2010.

[6] B. J. Buckley, E. M. Castillo, J. P. Killeen, D. A. Guss, and T. C. Chan, “Impact of an Express Admit Unit on Emergency Department Length of Stay,” *The Journal of Emergency Medicine*, vol. 39, no. 5, pp. 669–673, Nov. 2010, DOI: 10.1016/j.jemermed.2008.11.022.

[7] J. YEH and W. LIN, “Using simulation technique and genetic algorithm to improve the quality care of a hospital emergency department,” *Expert Systems with Applications*, vol. 32, no. 4, pp. 1073–1083, May 2007, DOI: 10.1016/j.eswa.2006.02.017.

[8] M. Zeinalnezhad, A. G. Chofreh, F. A. Goni, J. J. Klemeš, and E. Sari, “Simulation and Improvement of Patients’ Workflow in Heart Clinics during COVID-19 Pandemic Using Timed Coloured Petri Nets,” *International Journal of Environmental Research and Public Health*, vol. 17, no. 22, Article No. 8577, Nov. 2020, DOI: 10.3390/ijerph17228577.

[9] K. Salimifard, S.Y. Hosseini, and M.S. Moradi, “Improving Emergency Department Processes Using Coloured Petri Nets,” in *CEUR Workshop Proceedings, Vol. 989*, pp. 335–349, 2013.

[10] J. Jihene, A. El Mhamedi, and H. Chabchoub, “Simulationmodel of Emergency Department,” in *2007 International Conference on Service Systems and Service Management*, Jun. 2007, DOI: 10.1109/icsssm.2007.4280152.

[11] *Applications and Theory of Petri Nets* (Lecture Notes in Computer Science, vol. 5606), G. Franceschinis and K. Wolf, Eds. Berlin, Heidelberg: Springer, 2009.

[12] N. Hamza, M. A. Majid, and F. Hujainah, "SIM-PFED: A Simulation-Based Decision Making Model of Patient Flow for Improving Patient Throughput Time in Emergency Department," *IEEE Access*, vol. 9, pp. 103419–103439, 2021, DOI: 10.1109/access.2021.3098625.

[13] M. Laskowski, B. C. P. Demianyk, J. Witt, S. N. Mukhi, M. R. Friesen, and R. D. McLeod, "Agent-Based Modeling of the Spread of Influenza-Like Illness in an Emergency Department: A Simulation Study," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 6, pp. 877–889, Nov. 2011, DOI: 10.1109/titb.2011.2163414.

[14] J. Wang, J. Li, K. Tussey, and K. Ross, "Reducing Length of Stay in Emergency Department: A Simulation Study at a Community Hospital," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 42, no. 6, pp. 1314–1322, Nov. 2012, DOI: 10.1109/tsmca.2012.2210204.

[15] R. Konrad et al., "Modeling the impact of changing patient flow processes in an emergency department: Insights from a computer simulation study," *Operations Research for Health Care*, vol. 2, no. 4, pp. 66–74, Dec. 2013, DOI: 10.1016/j.orhc.2013.04.001.

[16] S. Ceschia and A. Schaerf, "Modeling and solving the dynamic patient admission scheduling problem under uncertainty," *Artificial Intelligence in Medicine*, vol. 56, no. 3, pp. 199–205, Nov. 2012, DOI: 10.1016/j.artmed.2012.09.001.

[17] S. Ceschia and A. Schaerf, "Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays," *Journal of Scheduling*, vol. 19, no. 4, pp. 377–389, Nov. 2014, DOI 10.1007/s10951-014-0407-8.

[18] P. Demeester, W. Souffriau, P. De Causmaecker, and G. Vanden Berghe, “A hybrid tabu search algorithm for automatically assigning patients to beds,” *Artificial Intelligence in Medicine*, vol. 48, no. 1, pp. 61–70, Jan. 2010, DOI: 10.1016/j.artmed.2009.09.001.

[19] C. Kittipittayakorn and K.-C. Ying, “Using the Integration of Discrete Event and Agent-Based Simulation to Enhance Outpatient Service Quality in an Orthopedic Department,” *Journal of Healthcare Engineering*, vol. 2016, pp. 1–8, 2016, DOI: 10.1155/2016/4189206.

[20] [20] M. Gül and A. F. Guneri, “ A computer simulation model to reduce patient length of stay and to improve resource utilization rate in an emergency department service system,” *International Journal of Industrial Engineering – theory Applications and Practice*, vol. 19, 2012.

[21] M. Lamiri, F. Grimaud, and X. Xie, “Optimization methods for a stochastic surgery planning problem,” *International Journal of Production Economics*, vol. 120, no. 2, pp. 400–410, Aug. 2009, DOI: 10.1016/j.ijpe.2008.11.021.

Zouaoui Louhab, Fatma Boufera

Received May 14, 2023
Revised September 4, 2023
Accepted September 8, 2023

Zouaoui Louhab
ORCID: `https://orcid.org/0000-0003-3231-9385`
Department of Computer Science, University Mustapha Stambouli,
Mascara, Algeria
E–mail: `zouaoui.louhab@univ-mascara.dz`

Fatma Boufera
ORCID: `https://orcid.org/0000-0002-5733-586X`
Department of Computer Science, University Mustapha Stambouli,
Mascara, Algeria
E–mail: `fboufera@univ-mascara.dz`

Computer Science Journal of Moldova, vol.**32**, no.1(94), 2024

Efficient GPU Power Management through Advanced Framework Utilizing Optimization Algorithms

Ramesha Rehman, Mashood Ul Haq Chishti,
Hamza Yamin

Abstract

The rapid rise in power usage by GPUs due to advances in machine and deep learning has led to an increase in power consumption of GPUs in Deep Learning workloads. To address this issue, a novel research project focuses on integrating Particle Swarm Optimization into a model training optimization framework to effectively reduce GPU power consumption during machine learning and deep learning training workloads. By utilizing the Particle Swarm Optimization (PSO)[1] algorithm within the proposed framework, we show the effectiveness of PSO in creating a more efficient power management strategy while also maintaining the performance. Upon evaluation of the proposed framework, it shows a reduction of 15.8% to 75.8% in power consumption across multiple workloads, with little to no performance loss.

Keywords: GPU, power reduction, machine learning, Particle Swarm Optimization.

MSC2020: 68-XX, 68Txx, 68T07.

1 Introduction

Over recent years, Graphics Processing Units (GPUs) have emerged as pivotal components across various domains such as gaming, computational graphics, machine learning, and scientific simulations [2]. The catalyst behind their prominence is their parallel processing capabilities, revolutionizing high-performance computing by enabling more

doi:10.56415/csjm.v32.08

efficient and rapid calculations. This progression in GPU technology has ushered in an era of heightened computing power, enabling the concurrent execution of numerous tasks. This parallelism contrasts with the sequential nature of central processing units (CPUs), conferring a significant advantage in handling computationally intensive applications. This multifaceted computation ability enhances speed and efficacy across diverse fields. However, the remarkable advancement in GPU computational capacity is accompanied by a significant escalation in energy consumption. The intrinsic parallel processing potential of GPUs, due to their multitude of processing cores, necessitates substantial electricity usage. The escalating power consumption of GPUs has raised pertinent concerns about environmental sustainability and energy conservation.

The elevated energy consumption of GPUs can be attributed to their specific design and architecture, which are optimized for parallel processing. Unlike CPUs, GPUs are engineered to execute tasks concurrently, resulting in a substantial core count to handle multiple calculations simultaneously. While this architectural choice boosts performance, it concurrently demands a greater power supply to accommodate the augmented workload. In response to these concerns, GPU manufacturers are actively exploring diverse avenues to mitigate power consumption and enhance energy efficiency [13]. This pursuit includes the development of more power-efficient architectures that optimize the balance between computational prowess and power usage. Additionally, software improvements play a pivotal role in enhancing energy efficiency. Software optimizations enable the effective distribution of computing tasks across available cores, thereby maximizing parallel processing capabilities and minimizing superfluous power consumption.

Clock gating [17] is a technique that selectively blocks the clock signal to individual GPU component when the component in question is not executing tasks. Power consumption can be substantially reduced by switching off the clock in dormant or seldom utilized components. Energy savings can be obtained by reducing wasted power consumption in GPU units that are not actively engaged in computations.

Another hardware-based technique called dynamic voltage and frequency scaling (DVFS) [16] adjusts the GPU's operating voltage and frequency in accordance with workload demands. Energy usage can be optimized by dynamically adjusting these parameters to the ideal efficiency level [5]. In periods of reduced processing activity, DVFS permits the GPU to operate at lower voltages and frequencies, thus reducing power consumption while maintaining performance. While these hardware-driven optimization methods have shown some degree of effectiveness in lowering GPU power usage, there is still room for improvement. Researchers keep looking at new concepts and ways to improve energy efficiency.

This article delves into the potential of leveraging machine learning (ML) models to curtail GPU power consumption, concentrating on optimal batch size and power limit configurations [8], an aspect often overlooked in the pursuit of model performance enhancement. Notably, ML's transformative impact on energy efficiency has surpassed its immediate domain, with far-reaching consequences. The astronomical energy consumption exhibited by large-scale models like GPT-3, which consumes as much as 1,287 megawatt-hours (MWh) of electricity, exemplifies this critical issue. This consumption equates to the energy utilization of an average U.S. home over a span of 120 years. While commendable strides have been made to reduce operational power footprints, the unceasing growth of artificial intelligence processing requirements poses potential energy challenges.

Given the evolving landscape of GPUs in machine learning and deep neural networks, it is imperative to address the research and innovation gap concerning energy efficiency [14] [12]. This research project aims to fill this void by comprehensively investigating the intricate interplay between energy consumption and processing capacity in the context of GPUs. The endeavor seeks to redefine the role of GPUs in energy-efficient AI and inspire researchers and industry professionals alike. By cultivating an in-depth understanding of energy-efficient GPU utilization, we aspire to foster a collaborative endeavor that amalgamates technological innovation with sustainable resource management in the dynamic realm of machine learning and deep neural networks.

In response to the existing gap in model training optimization, our research introduces a framework [11] aimed at reducing the power consumption of a workload [4] [9] [6]. This framework achieves optimization by adjusting the batch size and establishing an optimal power limit for a given workload [7]. To enhance batch size optimization, we employ a Multi Armed Bandit with GS-MOPSO strategy, allowing exploration of the search space to determine the most suitable batch size for the workload [10]. Simultaneously, power optimization is addressed through the utilization of a Just-in-Time profiler. This profiler leverages pre-calibrated configurations if optimization has been previously conducted for a similar workload. For novel workloads, the profiler measures throughput and average power consumption across various power limits, ultimately selecting the configuration that maximizes throughput while minimizing average power consumption. [15]

2 Methodology

2.1 Experimental Methodology

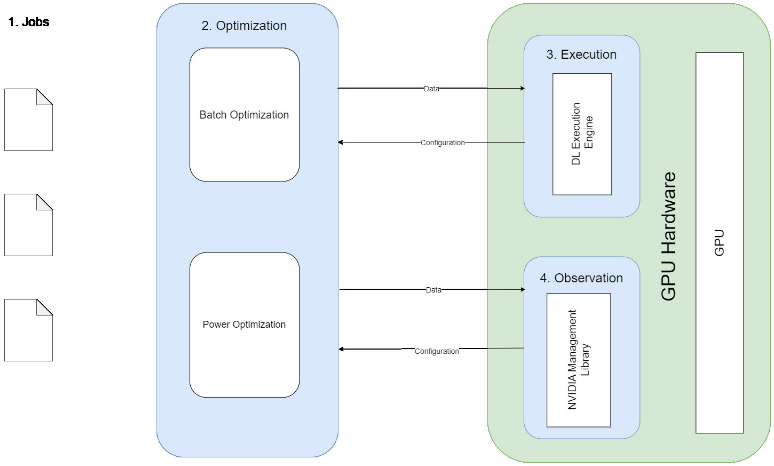


Figure 1. Proposed Methodology architecture

The proposed methodology, illustrated in Figure 1, aims to enhance efficiency through a multi-stage process. Tasks or jobs are channeled into the optimization framework (1), which unfolds as follows:

Optimization Stage: The initial phase employs Particle Swarm Optimization (PSO) to ascertain the optimal configuration for batch size and power limits (2). Subsequently, the training procedure commences employing the established configuration (3).

Monitoring and Feedback: Throughout the training process, the optimization framework gathers continuous statistics and information as feedback from the model's progression (4).

Our Proposed Framework works on step 2 (Optimization) of the proposed methodology, where it interacts with the GPU hardware to set the optimal configuration for the incoming jobs.

Adaptive Learning: Leveraging the collected feedback, the framework engages in adaptive learning, iteratively refining its configuration settings. This iterative loop persists until the predefined performance measure set by the user is attained or until the model's performance does not become more efficient within a specified time duration.

The iterative learning process empowers the framework to dynamically adapt its configuration based on feedback data, maintaining a trajectory towards the defined performance target or until saturation in model improvement is detected. Notably, the framework bifurcates the challenges associated with batch size and power limits, two parameters substantially influencing GPU performance and energy consumption. This strategic separation enables independent determination of the optimal power limit for any given batch size. Additionally, due to this decoupling, our exploration space is effectively confined to diverse batch sizes that harmonize with the optimal power limit. This focused exploration approach expedites the search for the optimum batch size, obviating the need to exhaustively assess every conceivable configuration.

2.2 Theoretical methodology

Our framework places a strong emphasis on the cost metric, which is a crucial component of our approach, hence we propose the following metric

$$Cost = \alpha.E_{tar}(s, w) + (1 - \alpha).P_{max}.T_{tar}(s, w). \tag{1}$$

Here α is the significance factor – if it is 0, then the framework optimizes for time efficiency, and if it is 1, then it optimizes for energy efficiency; P_{max} is the maximum power allowed by the GPU; and E_{tar} and T_{tar} are energy consumed and time taken to reach the target metric with configuration batch size s and power limit w.

In conclusion, our cost metric is a flexible tool that enables you to customise the optimisation method for your framework in accordance with your unique goals. You may efficiently strike a balance between time and energy efficiency by modifying the importance factor, ensuring that your framework fits your intended objectives and limits.

2.2.1 Choosing Ideal Cost

Expanding the equation (1), we get

$$Cost = (\alpha.P_{avg}(s,w) + (1-\alpha).P_{max}).T_{tar}(s,w). \tag{2}$$

In order to get the best cost, one must navigate a huge search space that is bounded by different batch sizes (s) and power restrictions (w). Additionally, because of the inherent variability in neural network training and the various hardware configurations of GPUs, it can be difficult to estimate both the average power consumption ($P_{avg}(s,w)$) and the time needed to obtain the goal metric ($T_{tar}(s,w)$).
We've put in place a dual approach to deal with these complications and uncertainties:

1. **Just-in-Time Profiler:** A Just-in-Time profiler has been incorporated into our system. During the execution of neural network training, this profiler is essential for dynamically measuring and monitoring the performance traits of various (s,w) configurations. We can make better judgments since it offers real-time information about the time and power needs [20], [19].

2. **Multi-Armed Bandit with GS-MOPSO:** We've used a Multi-Armed Bandit approach along with GS-MOPSO (Multi-Objective Particle Swarm Optimisation) to effectively explore the enormous search area and adapt to the stochastic nature of neural network training. Through careful consideration of the trade-offs between energy and time efficiency, this combination enables us

to deploy resources in a sensible manner. It continually modifies the (s, w) settings depending on performance data from the past, assisting us in more efficiently converging towards the ideal solution.

Essentially, our strategy makes use of real-time profiling and sophisticated optimisation techniques to address the problems brought on by the expansive and unpredictable (s, w) search space, as well as the inherent variability in GPU hardware and neural network training. This allows us to ultimately find the best cost-effective solutions.

2.2.2 Optimising power

Utilising the capabilities of our Just-in-Time profiler to increase power efficiency is fundamental to our optimisation approach. Within our system, when a job is started, the profiler activates and adheres to the following protocol:

1. **Batch Size Check:** The profiler first determines if the job's batch size has been calculated and optimised. It uses the pre-calibrated and optimised power limit for that specific batch size if optimisation has already been done for this batch size. With this strategy, we can rapidly and effectively distribute the right power resources for batches of known sizes.

2. **First Epoch Profiling:** When the batch size is used for the first time, our profiler intervenes to collect crucial information. During the first epoch of the work, it measures throughput and average power consumption (PAVG) for various power restrictions. We can create a baseline understanding of how various power restrictions effect performance for this particular batch size thanks to this thorough profiling.

2.2.3 Optimising batch size

We use Multi Armed Bandit with GS-MOPSO as shown in the algorithms below, to improve our batch size optimisation process. Together, these algorithms enable us to effectively calculate the optimal batch size based on the provided power restriction and then to carry out the training task using the identified power limit (w) and batch size (s).

Algorithm 1.

```
Input: Batch Sizes B, belief posterior $mean_b$ and $std_b^2$
Output: Batch size to run $b^*$
Function: $Predict(B, mean_b, std_b^2)$:
Initialize random population P
Initialize Personal Best Set $S_{pb}$ and External set $S_e$
For a=1:B do
    $S_{pb}\{a\}$=P(a)
$S_e$=P
For i=1:B do
    Assign rank to each particle in $P_i$ according to fast
    non-dominated sort
    For k=1:B
        $P_{best}$ = first particle in sorted $S_{pb}\{k\}$
        $N_{best}$= particle closest to $k_{th}$ particle in $S_e$
        Update $P_i(k)$ to $P_{i+1}(k)$
        if rand > $Rank_i/maxRank$
            if rand > $i/B$
                Sample=$[P_{best}, N_{best}]$
                $P_{i+1}(k)$=$N(mean_b, std_b^2)$
            else
                Divide $[P, S_{pb}, S_e]$ into N clusters using
                K-means
                Identify the cluster, to which particle $P_i(k)$
                belongs and assign it to Sample
                $P_{i+1}$=$N(mean_b, std_b^2)$
Update $S_e$
$b^* \longleftarrow argmin S_e$
```

Algorithm 2.

```
***Input:*** *Batch Size b and Total cost* $C_t$*,*
*Previous cost* $C_p$
*belief posterior* $mean_b$ *and* $std_b^2$
***Output:*** *updated belief posterior* $mean_b$ *and* $std_b^2$
***Function:*** $Observe(b, C_t, C_p, mean_b, std_b^2)$*:*
$C_p \longleftarrow C_p \bigcup \{C_t\}$
$std^2 \longleftarrow Variance(C_t)$
$std_b^2 \longleftarrow (\frac{1}{std_0^2} + \frac{|C_p|}{std^2})$
$mean_b \longleftarrow std_b^2(\frac{mean_0}{std_0^2} + \frac{\sum C_p}{std^2})$
```

Algorithm 3.

```
***Input:*** *Batch Size b*
*belief posterior* $mean_0$ *and* $std_0^2$
*while t<T do*
$b^* \longleftarrow Predict(B, mean_b, std_b^2 \forall b \in B)$
*Run job with b\* and add to cost C*
$mean_b$,$std_b^2 \longleftarrow Observe(b, C, mean_0, std_0^2)$
$t \longleftarrow t + 1$
```

Our novel optimisation framework stands at the forefront in enhancing computational efficiency in GPU-centric deep learning tasks. It successfully integrates three essential algorithms to dynamically modify batch sizes and minimize power usage. The sophisticated Algorithm 1 (Predict) at the heart of this technique incorporates the Multi Armed Bandit with GS-MOPSO strategy. Using belief posterior parameters like meanb and std2b, this strategy systematically explores and determines the ideal batch size. By proactively adjusting to the changing subtleties of the workload, it makes sure that the batch sizes have been carefully selected to match the deep learning model's exact specifications.

After the first prediction, Algorithm 2 (Observe) plays an essential role in coordinating an adaptive learning process at the end of every task run. This technique improves the model's understanding of the computation environment by iteratively updating the belief posterior with observed batch sizes, total cost (Ct), and the cumulative history of past costs (Cp). This continuous learning loop increases our optimization framework's versatility so that it can dynamically adjust to changing computing needs. Lastly, Algorithm 3 handles the whole iterative procedure, including belief posterior updates, task execution, and prediction in a smooth manner across a predefined number of iterations, T. The integrated approach's comprehensive nature highlights its effectiveness in managing power consumption and computing efficiency while also adeptly navigating the dynamic nature of deep learning workloads. With the help of these methods, our system shows itself to be a comprehensive and flexible approach, well suited to maximize GPU-driven deep learning's performance in the always changing field of computational difficulties.

We can speed up the batch size selection process and make sure that it perfectly matches the determined power limit(w) by combining these approaches. With this strategy, we may dynamically adjust and optimise the batch size(s) in response to shifting circumstances and limitations. As a consequence, we are able to control and carry out the training workloads with the best possible balancing of batch size and power limit.

3 Experiment/Results

In our evaluation, we considered prominent Deep Neural Network (DNN) models such as DeepSpeech2, ResNet-50, and NeuMF and others as shown in Table 1. Our investigation revolved around comparing training time and energy utilization across three methodologies: the default strategy, grid search exploration, and our novel approach on system specification as shown in Table 2. The outcomes of our research unveiled significant insights, outlined as follows:

3.1 Energy Consumption Reduction:

Figure 2 depicts the graphical representation of performance for the most recent five iterations of our approach, grid search, and the default baseline. Our methodology showcases a remarkable reduction in energy consumption, ranging from 15.3% to an impressive 75.8%.

Table 1. Models and datasets used in our evaluation of the respective configurations

Task	Dataset	Model	Optimizer	Default batch size	Target Metric
Speech Recognition	LibriSpeech	DeepSpeech2	Adam	176	WER: 55
Question answers	Sentiment140	BERT(QA)	Adam	44	F1:88.0
Sentiment Analysis	ImageNet	BERT(SA)	Adam	122	Accuracy:86%
Image Classification	CIFAR -100	RESNET-50, ShuffleNet-v2	Adadelta	232, 1000	Accuracy: 68%, 65%
Recommendation	MovieLens-1M	NeuMF	Adam	1000	NDCG: 0.52

This is further supported with Figure 5 and Table 3. This substantial decrease in energy usage is achieved while imposing minimal impact on training time.

3.2 Balanced Time-Consumption Trade-offs:

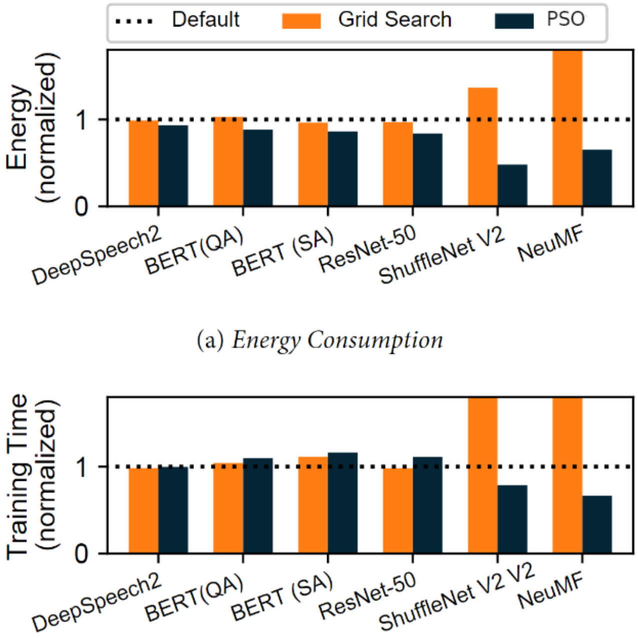
Within the context of training time, our strategy demonstrates a balanced trade-off with energy efficiency. Figure 2b elucidates this aspect by illustrating the duration of the last five rounds for both our methodology and grid search, compared to the default baseline. Notably, our methodology achieves a training time reduction of up to 60.1% for specific workloads, alongside a modest 12.8% increase for a distinct group of workloads. This finding underscores the delicate equilibrium between optimizing time and enhancing energy efficiency.

Table 2. GPU and System Specifications used in evaluation of our framework.

GPU Specifications	System Specification
Model	A40
CPU	AMD EPYC 7513
VRAM	48GB
RAM	512GB DDR4
Architecture	Ampere
Disk	960 GB NVMe SSD
Model	V100
CPU	AMD EPYC 7542
VRAM	32GB
RAM	512GB DDR4
Architecture	Volta
Disk	2TB HDD
Model	RTX 6000
CPU	Xeon Gold 6126
VRAM	24GB
RAM	192GB DDR4
Architecture	Turing
Disk	256GB SSD
Model	P100
CPU	Xeon E5-2670 v3
VRAM	16GB
RAM	128 GB DDR4
Architecture	Pascal
Disk	1TB HDD

Table 3. Improvements of different methodologies employed on DL training workloads w.r.t. Default NVIDIA strategy [18]

Method Employed	Least	Best
DVFS [16]	8.7%	23.1%
Clock Gating [17]	11.2%	60%
GPOEO [4]	8%	29.5%
Our Framework with PSO	15.3%	75.8%



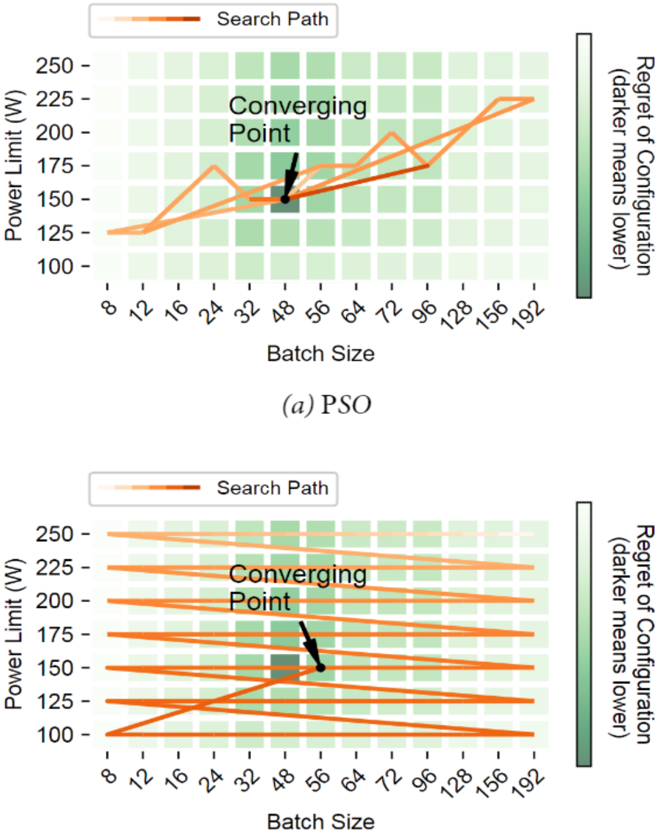
(a) *Energy Consumption*

(b) *Training Time*

Figure 2. Energy consumption and training time of last 5 iterations of Grid Search and Our framework with PSO w.r.t. baseline NVIDIA strategy

3.3 Progressive Regret and Resource Efficiency:

While our methodology and grid search exhibit comparable performance, our approach showcases superior resource efficiency for convergence. As illustrated in Figure 3, the increasing regret trajectories for DeepSpeech2 and ResNet-50 models underscore our methodology's ability to attain similar outcomes with significantly fewer resources compared to grid search, this is also supported by Figure 6.



(a) PSO

(b) Grid Search

Figure 3. Progressive Regret of our framework with both Grid Search and PSO

3.4 GPU Model-Independent Energy Savings:

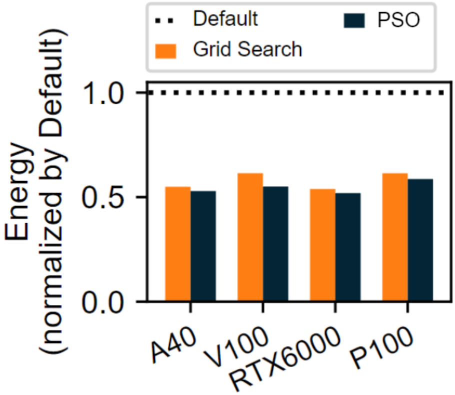


Figure 4. Energy consumption for DL workloads over multiple GPUs using our framework with Grid Search and PSO w.r.t default NVIDIA strategy

An essential discovery from our study is the independence of our methodology's effectiveness from GPU models as shown in Figure 4. Across four NVIDIA GPU generations, our approach consistently achieves substantial energy consumption reductions. This energy-saving capability exceeds 50% compared to the default baseline across multiple GPU models.

In summation, our comprehensive analysis demonstrates the efficacy of our approach in mitigating energy consumption without imposing considerable time penalties. This balanced strategy emerges as a viable avenue to enhance the training of Machine Learning and Deep Neural Network models across diverse workloads and GPU architectures.

4 Conclusion

In summation, our approach has showcased its efficacy through notable reductions in energy consumption, all while maintaining the unimpaired performance of GPUs. This achievement in diminishing GPU energy usage is attributed to the framework's adeptness in delicately balancing energy preservation and optimization of training time.

The promising outcomes of our technological endeavor underscore its potential to instigate a paradigm shift in the domain of energy-efficient Deep Neural Network (DNN) training. The approach's precision in quantifying the intricate interplay between training time and energy consumption has yielded substantial advancements in enhancing GPU energy efficiency.

Anticipating the future, we envision the widespread adoption of this methodology across diverse sectors, transcending its current scope. Innovations such as TinyML [3] open new vistas, wherein the implementation of our strategy in compact devices like mobile devices and embedded systems holds immense promise. Foreseeing a consequential enhancement in the efficiency of such devices through our methodology, we expect this trend to pave the way toward a technological landscape characterized by sustainability and heightened energy consciousness.

References

[1] A. A. Nagra, F. Han, Q. -H. Ling, M. Abubaker, F. Ahmad, S. Mehta, and A. T. Apasiba, "Hybrid self-inertia weight adaptive particle swarm optimization with local search using C4.5 decision tree classifier for feature selection problems," *Connection Science*, vol. 32, no. 1, pp. 16–36, DOI: 10.1080/09540091.2019.1609419.

[2] A. A. Nagra, I. Mubarik, M. M. Asif, K. Masood, M. A. A. Ghamdi, and S. H Almotiri, "Hybrid GA-SVM Approach for Post-operative Life Expectancy Prediction in Lung Cancer Patients," *Appl. Sci*, vol. 12, Article No. 10927, 2022.

[3] M. Shafique, A. Marchisio, R. V. Wicaksana Putra, and M. A. Hanif, “Towards Energy-Efficient and Secure Edge AI: A Cross-Layer Framework ICCAD Special Session Paper,” in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2021, pp. 1–9, DOI: 10.1109/ICCAD51958.2021.9643539.

[4] F. Wang, W. Zhang, S. Lai, M. Hao, and Z. Wang, “Dynamic GPU Energy Optimization for Machine Learning Training Workloads,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2943–2954, 1 Nov. 2022, DOI: 10.1109/TPDS.2021.3137867.

[5] S. Ilager, R. Muralidhar, K. Rammohanrao, and R. Buyya, “A Data-Driven Frequency Scaling Approach for Deadline-aware Energy Efficient Scheduling on Graphics Processing Units (GPUs),” in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud, and Internet Computing (CCGRID)*, 2020, pp. 579–588, DOI: 10.1109/CCGrid49817.2020.00-35.

[6] M. Endrei, C. Jin, M. N. Dinh, D. Abramson, H. Poxon, L. DeRose, and B. R. de Supinski, “Statistical and machine learning models for optimizing energy in parallel applications,” *The International Journal of High-Performance Computing Applications*, vol. 33, no. 6, pp. 1079–-1097, 2019.

[7] S. Ramesh, S. Perarnau, S. Bhalachandra, A. D. Malony, P. Beckman, *et al.*, “Understanding the impact of dynamic power capping on application progress”, in *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE*, 2019, pp. 793–-804.

[8] Y. Wang, W. Zhang, M. Hao, and Z. Wang, “Online power management for multi-cores: A reinforcement learning based approach,” *IEEE Transactions on Parallel and Distributed Systems*, 2021.

[9] K. Fan, B. Cosenza, and B. Juurlink, “Predictable GPUs frequency scaling for energy and performance,” in *Proceedings of the 48th International Conference on Parallel Processing*, 2019, pp. 1–-10.

[10] Y. Wen, Z. Wang, and M. F. O'Boyle, "Smart multi-task scheduling for OpenCL programs on CPU/GPU heterogeneous platforms," in *2014 21st International conference on high performance computing (HiPC). IEEE*, 2014, pp. 1–-10.

[11] V. Kandiah *et al.* "AccelWattch: A power modeling framework for modern gpus," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, DOI: 10.1145/3466752.3480063.

[12] Keskar, N. Shirish, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," arXiv preprint arXiv:1609.04836, 2016.

[13] M. Hodak, M. Gorkovenko, A. Dholakia, "Towards power efficiency in deep learning on Data Center Hardware," in *2019 IEEE International Conference on Big Data (Big Data) [Preprint]*, DOI: 10.1109/bigdata47090.2019.9005632.

[14] J. Chen *et al.*, "Closing the generalization gap of adaptive gradient methods in training deep neural networks," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence [Preprint]*, DOI: 10.24963/ijcai.2020/452.

[15] S. Hong and H. Kim, "An integrated GPU power and Performance Model", in *Proceedings of the 37th annual international symposium on Computer architecture [Preprint]*, DOI: 10.1145/1815961.1815998.

[16] Z. Tang *et al.*, "The impact of GPU DVFS on the energy and performance of Deep Learning," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems [Preprint]*, DOI: 10.1145/1815961.1815998.

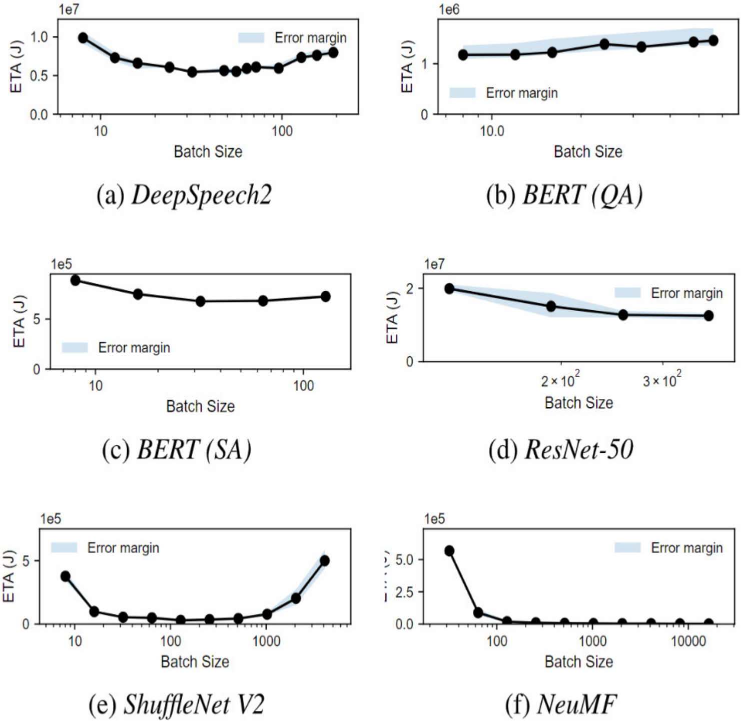
[17] J. Leng *et al.*, "GPUWattch," *ACM SIGARCH Computer Architecture News*, vol. 41, no. 3, pp. 487–-498, 2013, DOI: 10.1145/2508148.2485964. (in Romanian)

[18] Z. Jia, M. Maggioni, B. Staiger, and D. Paolo Scarpazza, "Dissecting the NVIDIA Volta GPU Architecture via Microbenchmarking," *CoRR* abs/1804.06826, April, 2018.

[19] S. Hong and H. Kim, “An Integrated GPU Power and Performance Model,” in *Proceedings of the 37th Annual International Symposium on Computer Architecture (ISCA ’10)*, 2010, pp. 280–289.

[20] J. Guerreiro, A. Ilic, N. Roma, and P. Tomas, “GPGPU Power Modeling for Multi-domain Voltage-Frequency Scaling,” in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2018.

Supplementary Information



(a) *DeepSpeech2*

(b) *BERT (QA)*

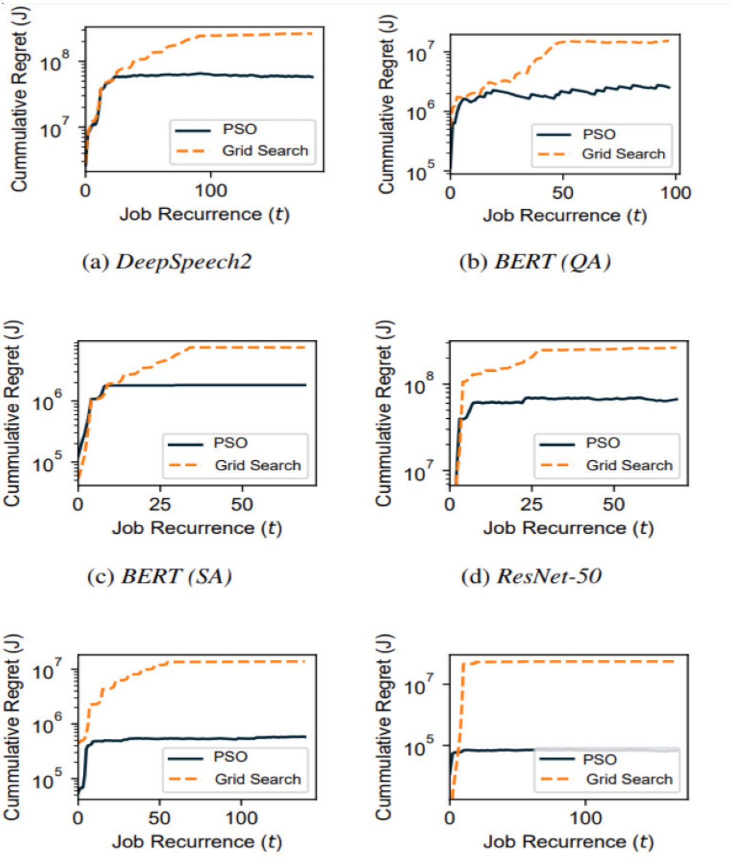
(c) *BERT (SA)*

(d) *ResNet-50*

(e) *ShuffleNet V2*

(f) *NeuMF*

Figure 5. Energy consumed w.r.t. batch size of training workloads. The blue shade represents error margins on multiple runs.



(a) *DeepSpeech2*

(b) *BERT (QA)*

(c) *BERT (SA)*

(d) *ResNet-50*

(e) *ShuffleNet V2*

(f) *NeuMF*

Figure 6. Progressive regret of PSO and Grid search on all training workloads

Ramesha Rehman[1], Mashood Ul Haq Chishti[2],
Hamza Yamin[3]

Received August 30, 2023
Revised January 28, 2024
Accepted January 31, 2024

[1,2,3]Lahore Garrison University
Phase 6, Block C, Defence Housing Authority,
Lahore, Punjab, Pakistan 54810

Ramesha Rehman
ORCID: https://orcid.org/0009-0009-3322-117X
E–mail: ramesharehman@lgu.edu.pk

Mashood Ul Haq Chishti
ORCID: https://orcid.org/0009-0000-4024-2360
E–mail: fa19-bscs-048@lgu.edu.pk

Hamza Yamin
ORCID: https://orcid.org/0009-0008-9556-2730
E–mail: fa19-bscs-084@lgu.edu.pk