



## การพัฒนาแบบจำลองวงจรตรรกะสังเคราะห์บนยีนด้วยขั้นตอนวิธีเชิงพันธุกรรม

จักกริช พฤษการ\*

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์

\* ผู้นิพนธ์ประสานงาน โทรศัพท์ 09 1094 6001 อีเมล: chakrit.p@ku.ac.th DOI: 10.14416/j.kmutnb.2018.11.002

รับเมื่อ 8 พฤษภาคม 2561 แก้ไขเมื่อ 13 กรกฎาคม 2561 ตอรับเมื่อ 3 สิงหาคม 2561 เผยแพร่ออนไลน์ 16 พฤศจิกายน 2561

© 2019 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

### บทคัดย่อ

บทความนี้ได้กล่าวถึงการนำวิธีการทางวิศวกรรมมาประยุกต์ใช้ในการศึกษาและพัฒนาแบบจำลองของกระบวนการแสดงออกของยีน ซึ่งการแสดงออกของยีนนั้นจะอยู่ในรูปของโปรตีน โดยการนำแนวคิดของกระบวนการลอกรหัสของ *lac operon* ที่มีกระบวนการทำงานที่เหมือนกับวงจรอิเล็กทรอนิกส์ ซึ่งสามารถพิจารณาด้วยสถานะของการทำงานแบบ On และ Off ได้ โดยมีตัวกระตุ้นหนึ่งหรือสองตัวที่เปรียบเสมือนการป้อนค่าอินพุตให้กับวงจร แบบจำลองการทำงานนี้ใช้ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) มาช่วยในการค้นหาค่าพารามิเตอร์ที่มีความเหมาะสมที่สุด เพื่อให้การคำนวณปริมาณการเกิดผลผลิตโปรตีนของกระบวนการลอกรหัสที่มีความเข้มข้นของตัวกระตุ้นที่แตกต่างกันนั้นได้ผลลัพธ์ที่มีความถูกต้อง ผลของการออกแบบเกตตรรกะสังเคราะห์พื้นฐานเหล่านี้สามารถนำไปขยายเป็นวงจรตรรกะสังเคราะห์ที่มีความซับซ้อนมากขึ้นได้

**คำสำคัญ:** กระบวนการแสดงออกของยีน, ขั้นตอนวิธีเชิงพันธุกรรม, วงจรตรรกะ, ชีววิทยาสังเคราะห์



## Developing a Model of Synthesis Logic Circuits Based on Gene with Genetic Algorithm

Chakkrit Preuksakarn\*

Department of Computer Engineering, Faculty of Engineering at Kamphaengsaen, Kasetsart University, Nakhon Pathom, Thailand

\* Corresponding Author, Tel. 09 1094 6001, E-mail: chakkrit.p@ku.ac.th DOI: 10.14416/j.kmutnb.2018.11.002

Received 8 May 2018; Revised 13 July 2018; Accepted 3 August 2018; Published online: 16 November 2018

© 2019 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

### Abstract

This article mentions about applying the engineering method for studying and developing more complex gene expression by defining the terms of protein expression. Using the idea of transcription process of *lac* operon which is similar to the electronic circuit due to the process that we can assume as the state ON and OFF by the inputs of one or two inducers to the circuit. By using the genetic algorithm for searching the most optimal parameter to calculate the amount of protein product of the transcription process in various concentration of inducers. The resulting optimal design of these biological logic gates can be extended to synthesis advanced sequential logic circuits.

**Keywords:** Gene Expression, Genetic Algorithm, Logic Circuit, Synthesis Biology

## 1. บทนำ

ชีววิทยาสังเคราะห์ เป็นความก้าวหน้าทางเทคโนโลยีสมัยใหม่ที่ถูกพัฒนามาจากการจำลองกระบวนการต่างๆ ที่เกิดขึ้นตามธรรมชาติของสิ่งมีชีวิต โดยใช้หลักการทางวิศวกรรมในออกแบบและสร้างระบบชีวภาพเทียม ทั้งนี้เพื่อช่วยส่งเสริมงานวิจัยด้านชีววิทยาให้มีความก้าวหน้ามากยิ่งขึ้น การออกแบบวงจรตรรกะทางชีวภาพในเชิงวิศวกรรมนั้น มีความคล้ายกันกับวงจรตรรกะทางอิเล็กทรอนิกส์ เนื่องจากปฏิกิริยาทางชีวเคมีต่างๆ ที่ถูกควบคุมโดยเอนไซม์นั้นเป็นผลผลิตมาจากยีน และกลไกที่ควบคุมการแสดงออกของยีนเกี่ยวข้องกับ การควบคุมการเปิด (Turn On) และปิด (Turn Off) ยีนในระดับการถอดรหัส (Transcription) และการแปลรหัส (Translation) วงจรตรรกะทางชีวภาพจึงเปรียบเสมือนการเลียนแบบพฤติกรรมของกระบวนการชีวเคมีในสิ่งมีชีวิต ซึ่งสามารถนำไปประยุกต์ใช้งานได้ในอนาคตกับหลายสาขา เช่น ด้านการแพทย์ ด้านเทคโนโลยีชีวภาพ ด้านมลพิษชีวภาพ และด้านพลังงานชีวภาพ เป็นต้น [1]-[3]

ในทางการแพทย์ปัจจุบัน มีโรคหลายชนิดที่เกี่ยวข้องกับความผิดปกติพันธุกรรมและโครโมโซม [4] และการรักษา ยังไม่สามารถตอบสนองต่อความต้องการของผู้ป่วยได้ เนื่องจากข้อจำกัดหลายประการในการทดลองการควบคุมการแสดงออกของยีน เพราะการแสดงออกของยีนนั้นเป็นกระบวนการทางชีวเคมีที่มีความซับซ้อนและเกี่ยวข้องกับหลายพารามิเตอร์ [5] อย่างไรก็ตาม การสร้างแบบจำลองทางคณิตศาสตร์ สามารถที่จะอธิบายปฏิกิริยาเคมีที่มีหลายพารามิเตอร์เหล่านี้ได้ และการพัฒนาวงจรทางตรรกะทางชีวภาพ [6], [7] มีความจำเป็นอย่างมาก เพื่อให้มนุษย์สามารถทำการค้นคว้าศึกษาและทดลองควบคุมยีนให้ทำงานในเงื่อนไขที่ซับซ้อนได้มากขึ้นในอนาคต

ในแง่ของระบบทางชีววิทยานั้น สิ่งมีชีวิตเป็นเครื่องคำนวณเชิงโมเลกุลโดยธรรมชาติอยู่แล้ว ดังนั้นการควบคุมยีน จึงมีพฤติกรรมคล้ายกับคอมพิวเตอร์โมเลกุล ไม่ว่าจะเป็นฮาร์ดแวร์ซอฟต์แวร์ อินพุต เอาต์พุต ซึ่งก็คือโมเลกุลที่ติดต่อกับสารอื่นๆ ผ่านทางปฏิกิริยาเคมีที่โปรแกรมเอาไว้แล้ว ฮาร์ดแวร์และซอฟต์แวร์เป็นโมเลกุลชีวภาพที่สามารถทำงานทางเคมี

ที่กำหนดไว้ล่วงหน้าได้ อินพุตของระบบจะเป็นโมเลกุลหนึ่ง และสามารถทำตามกฎ (ซอฟต์แวร์) ที่กำหนดให้ได้ และเอาต์พุตของการคำนวณเชิงเคมี ก็จะเป็นอีกโมเลกุลหนึ่ง [8] ระบบการทำงานในลักษณะนี้สามารถที่จะนำมาสร้างเป็นแบบจำลองวงจรทางตรรกะได้ (Logic Circuits) เนื่องจากทั้งอินพุต และเอาต์พุตของการควบคุมในระดับยีนนั้น มีสถานะ On-Off เหมือนกับเกตตรรกะ (Logic Gate) ซึ่งเป็นตัวดำเนินการทางตรรกศาสตร์ที่รับข้อมูลเข้าอย่างน้อยหนึ่งตัว มาคำนวณและส่งออกข้อมูลออกหนึ่งตัว [9], [10]

ด้วยหลักการนี้ เราสามารถที่จะออกแบบกระบวนการแสดงออกของยีนบนพื้นฐานของการคำนวณทางคณิตศาสตร์ได้ โดยการสร้างแบบจำลองวงจรตรรกะสังเคราะห์ ซึ่งมีพฤติกรรมการทำงานเหมือนกับวงจรตรรกะอิเล็กทรอนิกส์ เช่น NOT, AND, OR, NAND, NOR และ XOR เป็นต้น วงจรตรรกะสังเคราะห์นี้จะคำนวณปฏิกิริยาทางเคมีและสร้างสมดุลเพื่อให้เกิดความเสถียรระหว่างอินพุตและเอาต์พุต อย่างไรก็ตาม การหาค่าพารามิเตอร์ที่เหมาะสมสำหรับการคำนวณนั้นยังมีหลายประเด็นที่ต้องศึกษา ดังนั้นในงานวิจัยนี้จะศึกษาถึงการสร้างแบบจำลองวงจรตรรกะสังเคราะห์ และการหาพารามิเตอร์ที่เหมาะสมสำหรับใช้ในการคำนวณ โดยจะศึกษาการเปิดและปิดยีนในระดับการถอดรหัส เนื่องจากการควบคุมการแสดงออกของยีนส่วนใหญ่จะเป็นการควบคุมในขั้นตอนนี้ และในขั้นตอนนี้ถือว่าเป็นขั้นตอนที่สามารถควบคุมการแสดงออกของยีนได้อย่างมีประสิทธิภาพที่สุด

## 2. วิธีการศึกษาและออกแบบระบบ

### 2.1 การควบคุมการแสดงออกของยีน (Regulation of Gene Expression)

การแสดงออกของยีน เป็นการเปลี่ยนถ่ายข้อมูลพันธุกรรมที่เก็บไว้ในรูปแบบของดีเอ็นเอ โดยส่วนใหญ่จะเป็นการสังเคราะห์โปรตีนและเอนไซม์ต่างๆ เพื่อตอบสนองต่อความต้องการของเซลล์และการอยู่รอดของสิ่งมีชีวิตเมื่อสภาพแวดล้อมเปลี่ยนแปลงไป ในการแสดงออกของยีนนั้นสามารถควบคุมได้หลายระดับที่แตกต่างกัน เช่น ระหว่างการถอดรหัส (Transcription) การตัดแปลงโมเลกุลและการย่อยสลาย

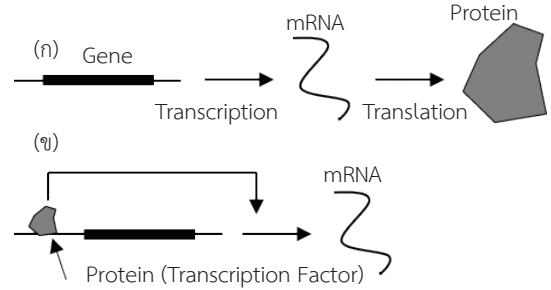
mRNA (mRNA Processing) การแปลรหัส (Translation) และการรักษาระดับของโปรตีนให้คงที่ อย่างไรก็ตาม จุดเริ่มต้นของการแสดงออกของยีน จะเริ่มจากระดับของการถอดรหัส ซึ่งเป็นขั้นตอนที่สำคัญในการศึกษาการควบคุมการแสดงออกของยีน

จากการศึกษากระบวนการถอดรหัสของ [11] พบว่าแบบจำลองของการคำนวณทางคณิตศาสตร์โดยส่วนใหญ่จะเกิดขึ้นที่บริเวณ *cis*-Regulatory Region (CRR) ของยีน แต่ละตัวที่เชื่อมโยงกันเป็นโครงข่าย ซึ่งอินพุตของ CRR นั้นจะเป็นตัวควบคุมอัตราการถอดรหัสของยีน ดังนั้นเราสามารถคำนวณอัตราการถอดรหัสได้จากฟังก์ชันของอินพุตของ *cis*-regulatory (*cis*-Regulatory Input Function; CRIF) และฟังก์ชันนี้สามารถอธิบายได้ด้วยแบบจำลองของเกตทางตรรกะ เนื่องจากมีการพบว่าจากการศึกษาแบบจำลอง *Lac* Operon ซึ่งเป็นแบบจำลองกลไกการควบคุมแสดงออกของยีนของ *Escherichia coli* (*E. coli*) นั้น มีพฤติกรรมคล้ายกับ AND-gate และ OR-gate เพราะมีตัวควบคุม 2 ตัว คือ cAMP Receptor Protein (CRP) และ LacI ซึ่งเปรียบเสมือนอินพุตฟังก์ชันของเกตทางตรรกะ และการแสดงออกของยีนของ *E. coli* นั้น จะแตกต่างกันออกไปขึ้นอยู่กับ การเปลี่ยนแปลงของฟังก์ชันอินพุตนี้ [12], [13]

## 2.2 การออกแบบวงจรตรรกะสำหรับกระบวนการถอดรหัส

การถอดรหัสบนดีเอ็นเอให้ออกมาเป็น mRNA นั้น จะถูกควบคุมโดยกลุ่มโปรตีน และปัจจัยในการควบคุมกระบวนการทางชีวเคมีโดยกลุ่มโปรตีนนี้เรียกว่า Transcription Factor (TF) ดังรูปที่ 1 [14] แม้จะมีหลายแบบจำลองที่อธิบายกระบวนการนี้ด้วยสมการดิฟเฟอเรนเชียล [15] แต่ยังเป็น การยากที่จะกำหนดค่าพารามิเตอร์ต่างๆ ที่เหมาะสม ดังนั้นในงานวิจัยนี้ จึงใช้วิธีการทางลอจิกเพื่อแสดงถึงการแสดงออกของยีน โดยใช้ฟังก์ชันตรรกะสำหรับอธิบายถึงปฏิสัมพันธ์ต่างๆ ของกระบวนการทางชีวเคมีในการถอดรหัส

จากแบบจำลองทางคณิตศาสตร์ที่แสดงถึงผลผลิตจากกระบวนการทางชีวเคมีในการแสดงออกของยีน [16] โดยที่การแสดงออกของยีนแต่ละตัวนั้นจะขึ้นอยู่กับความเข้มข้น



รูปที่ 1 (ก) กระบวนการถอดรหัสและการแปลรหัส

(ข) กระบวนการแสดงออกของยีนระดับการถอดรหัส

ของ mRNA และโปรตีน ดังนั้นถ้ากำหนดให้โครงข่ายของยีนมีจำนวน  $L$  ยีน โดยที่  $i = 1, 2, \dots, L$  แล้ว

$$m_i^* = \alpha_i f_i(\tilde{x}_i) - \gamma_{m_i} m_i + \alpha_{i,0} \quad (1)$$

$$p_i^* = \beta_i m_i - \gamma_{p_i} p_i \quad (2)$$

เมื่อ  $m_i$  คือ ความเข้มข้นของ mRNA ของยีน

$i, p_i$  คือ ความเข้มข้นของโปรตีนของยีน  $i, \gamma_{m_i}$

$\gamma_{p_i}$  คือ ค่าคงที่อัตราการเสื่อมสลายของ mRNA และโปรตีน

$\alpha_i$  คือ อัตราการถอดรหัสของ mRNA

$\beta_i$  คือ อัตราการสังเคราะห์โปรตีน

$\alpha_{(i,0)}$  คือ ค่าตั้งต้นของอัตราการถอดรหัสของ mRNA

$f_i(\cdot)$  คือ ฟังก์ชันปฏิกิริยาเคมีที่ตำแหน่งโพรโมเตอร์ของสายดีเอ็นเอ

$\tilde{x}_i$  คือ ความเข้มข้นของกลุ่มโปรตีนที่เป็นปัจจัยในการควบคุมกระบวนการถอดรหัส (TF)

ฟังก์ชันปฏิกิริยาทางเคมีนี้เป็นพีชคณิตบูลีนที่มีขอบเขตของสมาชิก  $\{0,1\}$  ดังนั้น ถ้ากำหนดให้  $i = 1, 2, \dots, L$  แล้ว

$$\tilde{x}_i = \begin{cases} 0 & \text{ถ้า } x_i < \theta_i \\ 1 & \text{ถ้า } x_i \geq \theta_i \end{cases} \quad (3)$$

เมื่อ  $\theta_i$  คือ ค่า Threshold ของความเข้มข้นของโปรตีน

แบบจำลองดังกล่าวแสดงให้เห็นถึงกระบวนการถอด



รหัสที่เปลี่ยนดีเอ็นเอเป็น mRNA ซึ่งกระบวนการนี้ถูกควบคุมด้วยปัจจัยของกระบวนการถอดรหัส (Transcription Factor) อัตราการเกิดผลิตภัณฑ์ที่ได้จากยีนนั้นจะขึ้นอยู่กับความเข้มข้นของปัจจัยของกระบวนการถอดรหัสที่แตกต่างกัน ดังนั้นจึงสามารถสร้างฟังก์ชันตรรกะสังเคราะห์ เพื่ออธิบายถึงการแสดงออกของยีนในกระบวนการถอดรหัสได้ดังนี้

การแสดงออกของยีนเมื่อมีปัจจัยของกระบวนการถอดรหัส 1 ตัว สามารถสร้างแบบจำลองที่มีลักษณะเหมือนกับฟังก์ชันตรรกะสังเคราะห์ของเกต NOT และ BUFFER ได้ ดังสมการที่ (4) [17]

$$f(X_1) = \frac{\left[\left(\frac{R}{K_R}\right)^{h_1}\right]^{m_1} + \left[\left(\frac{X_1}{K_1}\right)^{h_2}\right]^{m_2}}{1 + \left(\frac{R}{K_R}\right)^{h_1} + \left(\frac{X_1}{K_1}\right)^{h_2}} \quad (4)$$

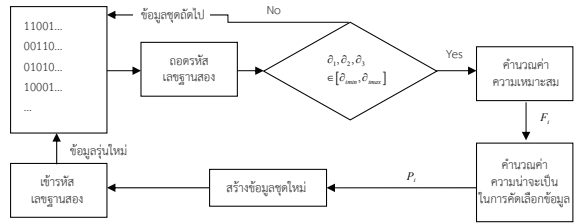
การแสดงออกของยีนเมื่อมีปัจจัยของกระบวนการถอดรหัส 2 ตัว สร้างแบบจำลองที่มีลักษณะเหมือนกับฟังก์ชันตรรกะสังเคราะห์ของเกต AND, OR, NAND, NOR, และ XOR ได้ ดังสมการที่ (5)

$$f(X_1, X_2) = \frac{\left[\left(\frac{R}{K_R}\right)^{h_1}\right]^{m_1} + \left[\left(\frac{X_1}{K_1}\right)^{h_2}\right]^{m_2} + \left[\left(\frac{X_2}{K_2}\right)^{h_3}\right]^{m_3} + \left[\left(\frac{X_1}{K_1}\right)^{h_2} \left(\frac{X_2}{K_2}\right)^{h_3}\right]^{m_4}}{1 + \left(\frac{R}{K_R}\right)^{h_1} + \left(\frac{X_1}{K_1}\right)^{h_2} + \left(\frac{X_2}{K_2}\right)^{h_3} + \left(\frac{X_1}{K_1}\right)^{h_2} \left(\frac{X_2}{K_2}\right)^{h_3}} \quad (5)$$

เมื่อ  $X_1, X_2$  คือ ความเข้มข้นของ Transcription Factor (TF)  
 $R$  คือ ความเข้มข้นของ RNA Polymerase  
 $K_R$  คือ ค่าคงที่สมดุลของ RNA Polymerase  
 $K_1, K_2$  คือ ค่าคงที่สมดุลของ Transcription Factor  
 $h_1, h_2, h_3$  คือ ค่าสัมประสิทธิ์ของ Hill (Hill Coefficients)  
 $m_1, m_2, m_3$  คือ พารามิเตอร์ที่ใช้ควบคุม

**2.3 การประมาณค่าพารามิเตอร์ด้วยขั้นตอนวิธีเชิงพันธุกรรม**

จากสมการที่ (4) และ (5) มีหลายพารามิเตอร์สำหรับการออกแบบเกตตรรกะสังเคราะห์ (NOT, BUFFER, AND, OR, NAND, NOR, XOR) ดังนั้นการหาค่าพารามิเตอร์



**รูปที่ 2** แฝงผังกระบวนการทำงานในการค้นหาพารามิเตอร์ให้กับแบบจำลองโดยใช้ขั้นตอนวิธีเชิงพันธุกรรม

ที่เหมาะสมสำหรับแบบจำลองทางคณิตศาสตร์ เพื่อให้เกตตรรกะสังเคราะห์นี้ สามารถทำงานได้เหมือนกับเกตตรรกะทางไฟฟ้าได้ ดังตารางที่ 1 จึงเป็นสิ่งที่มีความจำเป็นมาก

**ตารางที่ 1** ตารางค่าความจริงแสดงคุณสมบัติของเกตตรรกะสังเคราะห์

INPUTS		OUTPUTS						
$X_1$	$X_2$	NOT	BUFFER	AND	OR	NAND	NOR	XOR
0	0	1	0	0	0	1	1	0
0	1	1	0	0	1	1	0	1
1	0	0	1	0	1	1	0	1
1	1	0	1	1	1	0	0	0

ในงานวิจัยนี้ใช้ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) สำหรับการหาพารามิเตอร์ที่เหมาะสมให้กับแบบจำลองดังกล่าว การใช้ขั้นตอนวิธีเชิงพันธุกรรมนั้นชุดข้อมูลเริ่มต้นที่ต้องใช้ในการคำนวณจะอยู่ในรูปแบบของไบนารีสตริง (Binary String) ดังรูปที่ 2 จะแสดงให้เห็นถึงแผนผังการทำงานของอัลกอริทึมในการค้นหาค่าพารามิเตอร์ให้กับแบบจำลอง

**2.4 การเข้ารหัสและถอดรหัสในรูปแบบเลขฐานสอง**

กำหนดให้พารามิเตอร์แต่ละตัว (เลขฐานสิบ) ที่ต้องการหา คือ  $\theta$ , เมื่อเข้ารหัสหรือแปลงค่าพารามิเตอร์ให้อยู่ในรูปของไบนารีสตริง (เลขฐานสอง) ที่มีความยาว  $B$ , บิต จะสามารถหาความสัมพันธ์ระหว่างความยาวของไบนารีสตริง ( $B_i$ ) กับค่า

**ตารางที่ 2** ตัวอย่างของพารามิเตอร์ในรูปแบบของเลขฐานสองและฐานสิบ

$d_1$	code	$d_2$	code	$d_3$	code
0	0000000000	0	0000000000	0	0000000000
0.01	0000000001	0.01	0000000001	0.01	0000000001
...	...	...	...	...	...
...	...	...	...	...	...
10	1111111111	10	1111111111	10	1111111111

ความละเอียดการเพิ่มค่าพารามิเตอร์ ( $\varnothing$ ) ได้จากสมการที่ (6)

$$\varnothing = \frac{d_{imax} - d_{imin}}{2^{B_i} - 1} \quad (6)$$

เมื่อ  $d_{imax}$ ,  $d_{imin}$  คือ ขอบเขตสูงสุดและขอบเขตต่ำสุดของแต่ละพารามิเตอร์ และ  $d_i \in [d_{imin}, d_{imax}]$

ในงานวิจัยนี้ กำหนดให้  $(d_{imin}, d_{imax}) = (0, 10)$  และจำนวนบิตของพารามิเตอร์  $B_i = 10$  ดังนั้น ความละเอียดของการเพิ่มค่าพารามิเตอร์ ( $\varnothing$ ) จะเท่ากับ 0.01 จากนั้นสามารถเข้ารหัสจากเลขฐานสิบเป็นเลขฐานสอง ได้จากสมการที่ (7)

$$\delta_i = \frac{(d_i - d_{imin})\omega}{d_{imax} - d_{imin}} \quad (7)$$

เมื่อ  $d_i$  คือ ค่าเลขฐานสอง

$d_i$  คือ เลขฐานสิบ

$\omega$  คือ ค่าสูงสุดของเลขฐานสอง โดยที่

$$\omega = \sum_{j=0}^{B_i-1} 2^j \quad (8)$$

จากตารางที่ 2 แสดงให้เห็นถึงตัวอย่างกลุ่มของพารามิเตอร์ในระบบเลขฐานสอง โดยที่  $(d_1, d_2, d_3)$  คือ ค่าของพารามิเตอร์ตัวที่ 1, 2 และ 3 ตามลำดับ Code คือ ค่าเลขฐานสองของพารามิเตอร์ ถ้าสมมติให้  $(d_1, d_2, d_3)$  เท่ากับ (0.01, 0, 10) แล้ว จะได้ลำดับของเลขฐานสองเป็นดังนี้ 0000000001 0000000000 1111111111

สำหรับกระบวนการถอดรหัสจากเลขฐานสองเป็นเลขฐานสิบนั้น จะเป็นการทำงานแบบย้อนกลับของการเข้ารหัส ดังสมการที่ (9)

$$d_i = d_{imin} + \frac{\delta_i}{\omega} (d_{imax} - d_{imin}) \quad (9)$$

## 2.5 การคัดเลือกชุดข้อมูล

ในกระบวนการการคัดเลือกชุดข้อมูลนี้ จะใช้หลักการของการเลือกข้อมูลที่มีค่าความเหมาะสมที่สูง เพื่อสร้างเป็นชุดข้อมูลใหม่ที่เข้าใกล้คำตอบของการหาพารามิเตอร์ที่เหมาะสมให้แบบจำลองได้มากยิ่งขึ้น โดยอัลกอริทึมจะทำการค้นหาพารามิเตอร์  $(d_1, d_2, \dots, d_n)$  ที่ทำให้เกิดค่าใช้จ่ายน้อยที่สุด ดังนั้น

ถ้ากำหนดให้  $F$  คือฟังก์ชันค่าความเหมาะสม (Fitness Function) และ  $C_{total}$  คือฟังก์ชันค่าใช้จ่าย แล้ว

$$F = \frac{1}{C_{total}} \quad (10)$$

จากสมการที่ (4) และ (5) ค่าเป้าหมายของการคำนวณเป็นค่าทางตรรกะ ดังนั้นการหาค่าใช้จ่ายในสมการที่ (10) เพื่อให้ได้พารามิเตอร์ใกล้เคียงกับค่าเป้าหมายนั้น จะแบ่งเป็น 2 ส่วน คือ ค่าความถูกต้องของผลลัพธ์ทางตรรกะ ( $C_a$ ) และค่าความแตกต่างระหว่างลอจิกสูงกับต่ำ ( $C_d$ ) โดยกำหนดฟังก์ชันค่าใช้จ่าย ดังสมการที่ (11)

$$C_{total} = \rho C_a + (1 - \rho) C_d \quad (11)$$

เมื่อ  $p \in [0,1]$  คือค่าปัจจัยน้ำหนักของความสัมพันธ์ระหว่าง  $C_d$  และ  $C_a$  ซึ่ง  $C_d$  ได้จากสมการที่ (12) และ  $C_a$  ได้จากสมการที่ (13) โดยที่  $\max(f_i)$ ,  $\min(f_i)$  คือค่าสูงสุดและต่ำสุดของผลลัพธ์ที่ได้จากการป้อนชุดพารามิเตอร์ในสมการการแสดงผลของยีน [สมการที่ (4), (5)]

$$C_d = 1 - \frac{\max(f_i) - \min(f_i)}{\max(f_i)} \quad (12)$$

$$C_a = \begin{cases} \max(f_i) - f_i(X_1, X_2), & \text{สำหรับ ลอจิก 1} \\ 1 - [\max(f_i) - f_i(X_1, X_2)], & \text{สำหรับ ลอจิก 0} \end{cases} \quad (13)$$

ดังนั้นเมื่อ  $f(X_1, X_2)$  เป็นฟังก์ชันตรรกะสังเคราะห์ของการแสดงออกของยีน และ  $(X_1, X_2) = (0,0), (0,1), (1,0), (1,1)$  โดย 0 เป็นค่าลอจิกต่ำ และ 1 เป็นค่าลอจิกสูง แล้ว  $C_d$  จะมีค่าน้อยก็ต่อเมื่อ  $\max(f_i)$  และ  $\min(f_i)$  มีความแตกต่างกันมาก

หลังจากประเมินความเหมาะสมของพารามิเตอร์จากฟังก์ชันความเหมาะสมแล้ว จะทำการคัดเลือกชุดข้อมูลเพื่อนำไปสร้างเป็นชุดข้อมูลรุ่นถัดไป โดยการคัดเลือกนี้จะใช้วิธีการคัดเลือกแบบ Roulette Wheel [18] คือ การสุ่มเลือกด้วยการกำหนดความน่าจะเป็นในการถูกคัดเลือกตามสัดส่วนของคะแนนความเหมาะสมของชุดข้อมูลจากผลรวมคะแนนทั้งหมด ดังสมการที่ (14)

$$P_i = \frac{F_i}{\sum F_i} \quad (14)$$

## 2.6 การสร้างข้อมูลชุดใหม่

เมื่อได้คัดเลือกชุดพารามิเตอร์จากหลักการของความน่าจะเป็นแล้ว ในกระบวนการนี้จะทำการสร้างข้อมูลชุดใหม่ขึ้นมา โดยอาศัยหลักการพื้นฐานสามขั้นตอน คือ ขั้นแรกจะทำการเลือกชุดพารามิเตอร์มาครั้งละคู่จากจำนวนทั้งหมด ขั้นที่สองจะทำการสุ่มตำแหน่งบิตของข้อมูลที่จะทำการเปลี่ยนแปลง โดยใช้วิธีการสุ่มเลือกแบบหนึ่งจุด (Single Point Crossover) [19] เนื่องจากให้ผลลัพธ์ที่ดีกว่าการสุ่ม

แบบหลายจุด ตัวอย่างเช่น สมมติให้ชุดพารามิเตอร์สองชุดเป็นดังนี้  $(d_1, d_2, d_3)_A$  และ  $(d_1, d_2, d_3)_B$  โดยที่พารามิเตอร์แต่ละตัวมีขนาด 10 บิต เมื่อมีการสุ่มเลือกบิตที่ต้องการเปลี่ยนแปลง จะได้ดังนี้

$$(d_1, d_2, d_3)_A = 1100000000[1]111111111000000000$$

$$(d_1, d_2, d_3)_B = 1010101011[0]1111111110000011111$$

ขั้นที่สาม ทำการสลับตำแหน่งของข้อมูลในตำแหน่งที่ถูกเลือกมา และจะได้ชุดข้อมูลใหม่เป็นดังนี้

$$(d_1, d_2, d_3)_A = 1100000000[0]1111111110000000000$$

$$(d_1, d_2, d_3)_B = 1010101011[1]1111111110000011111$$

ถึงแม้ว่าการสร้างข้อมูลพารามิเตอร์ชุดใหม่นั้น จะใช้หลักการสุ่มตำแหน่งเพื่อแลกเปลี่ยนข้อมูลก็ตาม แต่วิธีการนี้เมื่อใช้ร่วมกับวิธีการคัดเลือกข้อมูลที่เหมาะสมก่อนหน้าแล้วสามารถที่จะทำให้การคำนวณค่าพารามิเตอร์มีความเหมาะสมมากยิ่งขึ้นได้

อย่างไรก็แล้ว แม้ว่าจะได้ชุดข้อมูลใหม่ที่ดีขึ้นกว่าเดิมก็ตาม แต่กระบวนการดังกล่าวยังเป็นการเปลี่ยนแปลงโดยอาศัยชุดข้อมูลเดิมที่มีอยู่แล้วเท่านั้น และอาจจะไม่สามารถค้นพบชุดข้อมูลใหม่ที่ดีกว่าชุดข้อมูลเดิม ดังนั้นเพื่อเป็นการช่วยเสริมความสมบูรณ์ของอัลกอริทึมก่อนหน้า จึงเพิ่มวิธีการเลียนแบบจากธรรมชาติที่เรียกว่า การกลายพันธุ์ (Mutation) เข้าไปด้วย โดยวิธีการนี้จะใช้การสุ่มตำแหน่งบิตของข้อมูลที่ต้องการเปลี่ยนแปลง เมื่อได้ตำแหน่งแล้ว จะทำการเปลี่ยนบิตข้อมูลจาก 0 เป็น 1 หรือ 1 เป็น ตัวอย่างเช่น

ก่อนการกลายพันธุ์

$$(d_1, d_2, d_3) = 110[0]00000001111111110000000000$$

หลังการกลายพันธุ์

$$(d_1, d_2, d_3) = 110[1]00000001111111110000000000$$

ในกระบวนการนี้จะใช้อัตราสุ่มที่ต่ำ เพื่อไม่ให้เกิดการเปลี่ยนแปลงข้อมูลมากเกินไป ดังนั้น ถ้าให้  $p$  คือความน่าจะเป็นของการสุ่มข้อมูล และ  $l$  คือความยาวของบิตข้อมูล จะสามารถเขียนสมการได้ดังนี้

$$p = \frac{1}{l} \quad (15)$$



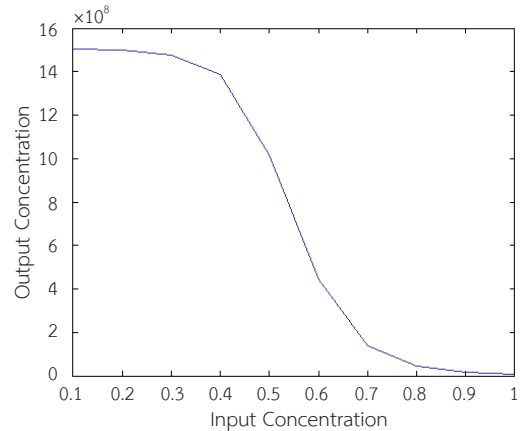
### 3. ผลการศึกษา

ในงานวิจัยนี้ได้แบ่งการทดลองออกเป็น 2 ส่วน จากสมการที่ (4) และ (5) ซึ่งเป็นแบบจำลองสำหรับการแสดงออกของยีนเมื่อมีปัจจัยของการลอกรหัส 1 และ 2 ตัว โดยเกิดตรรกะสังเคราะห์ที่มีปัจจัยของการลอกรหัส 1 ตัว คือ NOT และ BUFFER ส่วนเกิดตรรกะสังเคราะห์ที่มีปัจจัยของการลอกรหัส 2 ตัว คือ AND, OR, NAND, NOR และ XOR โดยในทุกประเภทจะทำการสุ่มกลุ่มของพารามิเตอร์จำนวน 100 ชุด แต่ละพารามิเตอร์มีจำนวนบิต ( $B_i$ ) 10 บิต จากนั้นจะทำการคัดเลือกชุดข้อมูลด้วยวิธีการคัดเลือกแบบ Roulette Wheel โดยมีอัตราการคัดเลือก 50% ของจำนวนพารามิเตอร์ทั้งหมด เพื่อสร้างเป็นกลุ่มพารามิเตอร์รุ่นแรกสำหรับป้อนเข้าสู่กระบวนการทำงานในการค้นหาพารามิเตอร์ที่เหมาะสมและในกระบวนการค้นหาดังกล่าวจะกำหนดให้มีอัตราการกลายพันธุ์ 5%

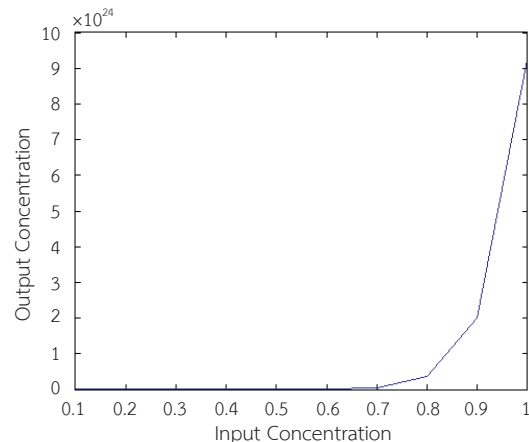
สำหรับเกิดตรรกะสังเคราะห์ NOT และ BUFFER อัลกอริทึมจะทำการค้นหาพารามิเตอร์ที่เหมาะสมที่สุดจำนวน 7 ตัว ( $R, K_R, K_1, h_1, h_2, m_1, m_2$ ) ส่วนเกิดตรรกะสังเคราะห์ AND, OR, NAND, NOR และ XOR อัลกอริทึมจะหาพารามิเตอร์จำนวน 11 ตัว ( $R, K_R, K_1, K_2, h_1, h_2, m_1, m_2, m_3, m_4$ ) โดยอัลกอริทึมจะหาพารามิเตอร์ที่ให้ผลลัพธ์ที่ตรงกับตารางค่าความจริงที่พบทั้งหมด 10 ครั้ง แล้วพิจารณาเลือกพารามิเตอร์ที่เหมาะสมที่สุดจากฟังก์ชันค่าใช้จ่ายที่น้อยที่สุด จากการทดลองได้ผลลัพธ์ดังนี้

#### 3.1 เกิดตรรกะสังเคราะห์ “NOT”

จากผลการทดลองหาค่าพารามิเตอร์สำหรับเกิดตรรกะสังเคราะห์ NOT ที่ทำให้เกิดค่าใช้จ่ายที่น้อยที่สุดนั้น ได้ผลลัพธ์เป็นดังนี้  $R = 5.76, K_R = 0.4, K_1 = 0.6, h_1 = 1.50, h_2 = 9.37, m_1 = 6.68, m_2 = 0$  จากรูปที่ 3 จะสังเกตเห็นได้ว่าในช่วงแรกที่ไม่มีการป้อนค่า Transcription Factor เข้าไประบบจะมีการสร้างผลผลิต (โปรตีน) ออกมา และเมื่อมีการเพิ่มความเข้มข้นของ Transcription Factor โปรตีนจะถูกต่อต้านจนมีการสร้างที่ลดลง



รูปที่ 3 ผลลัพธ์ของเกิดตรรกะสังเคราะห์ NOT



รูปที่ 4 ผลลัพธ์ของเกิดสังเคราะห์ BUFFER

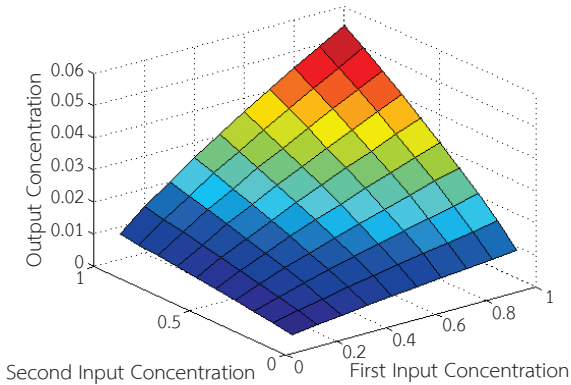
#### 3.2 เกิดตรรกะสังเคราะห์ “BUFFER”

เกิดชนิดนี้จะให้ผลลัพธ์ที่ตรงกันข้ามกับ NOT คือ ผลผลิต (โปรตีน) จะไม่ถูกผลิตออกมาในช่วงแรกเมื่อความเข้มข้นของ Transcription Factor เป็นศูนย์หรือมีค่าน้อย แต่เมื่อเพิ่มค่าขึ้น จะมีการผลิตโปรตีนออกมามากขึ้น ซึ่งค่าพารามิเตอร์ที่เหมาะสมคือ  $R = 4.956, K_R = 0.28807, K_1 = 0.01955, h_1 = 1.87, h_2 = 3.60, m_1 = 0.00488, m_2 = 2.69$  จากการทดลองโดยใช้ค่าพารามิเตอร์เหล่านี้ จะได้ผลลัพธ์ดังรูปที่ 4

#### 3.3 เกิดตรรกะสังเคราะห์ “AND”

เกิดนี้จะมีอินพุต 2 ตัว โดยมีพารามิเตอร์ที่ทำให้มี





รูปที่ 5 ผลลัพธ์ของเกตสังเคราะห์ AND

ผลลัพธ์ที่ได้ถูกต้องคือ  $R = 4.13$ ,  $K_R = 9.58$ ,  $K_1 = 7.03$ ,  $K_2 = 5.10$ ,  $h_1 = 7.67$ ,  $h_2 = 8.12$ ,  $h_3 = 7.74$ ,  $m_1 = 2.37$ ,  $m_2 = 3.20$ ,  $m_3 = 7.28$ ,  $m_4 = 0.10$

จากรูปที่ 5 จะเห็นได้ว่า ด้วยพารามิเตอร์ดังกล่าว โปรตีนจะถูกกระตุ้นและมีการผลิตโปรตีนเพิ่มมากขึ้นเมื่อ Transcription Factor ทั้งสองตัวมีค่าเพิ่มขึ้น และจะถูกต่อต้านทำให้โปรตีนไม่ถูกผลิตออกมา หรือถูกผลิตในปริมาณน้อยเมื่อ Transcription Factor ทั้งสองมีค่าลดลง

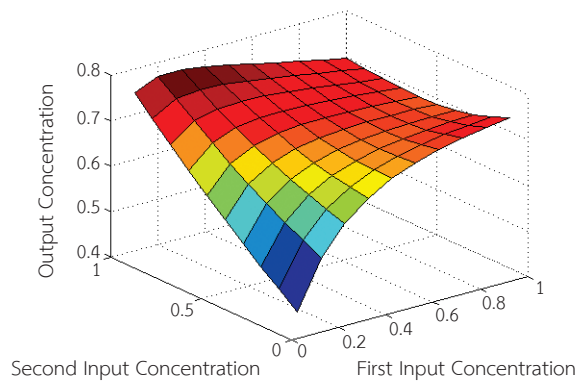
### 3.4 เกตตรรกะสังเคราะห์ “OR”

ผลที่ได้จากการทดลองของการทดสอบการแสดงออกของเกตตรรกะสังเคราะห์ OR เป็นดังรูปที่ 6 โดยมีพารามิเตอร์ดังนี้  $R = 1.78$ ,  $K_R = 3.23$ ,  $K_1 = 0.45$ ,  $K_2 = 1.03$ ,  $h_1 = 2.78$ ,  $h_2 = 1.12$ ,  $h_3 = 0.74$ ,  $m_1 = 1.09$ ,  $m_2 = 0.88$ ,  $m_3 = 2.08$ ,  $m_4 = 0.40$

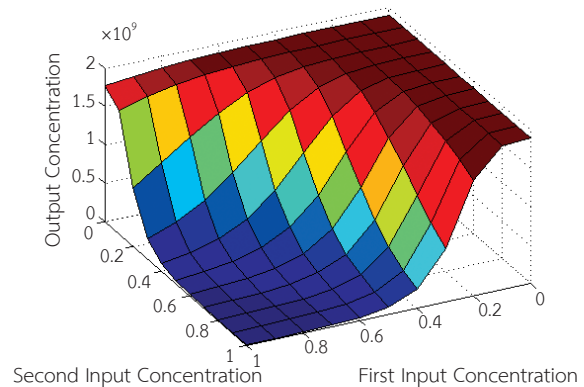
ค่าพารามิเตอร์ดังกล่าว จะทำให้ไม่มีการกระตุ้นให้มีการผลิตโปรตีนเมื่อ Transcription Factor ทั้งสองมีค่าเป็น 0 แต่จะผลิตโปรตีนเมื่อ Transcription Factor ตัวใดตัวหนึ่งหรือทั้งสองตัวมีค่าเป็น 1

### 3.5 เกตตรรกะสังเคราะห์ “NAND”

เกตชนิดนี้ เป็นส่วนเติมเต็มของเกต AND กล่าวคือ เมื่อความเข้มข้นของ Transcription Factor ตัวใดตัวหนึ่งหรือทั้งสองตัวมีค่าเป็น 0 โปรตีนจะถูกกระตุ้นให้มีการสร้างผลผลิต



รูปที่ 6 ผลลัพธ์ของเกตสังเคราะห์ OR

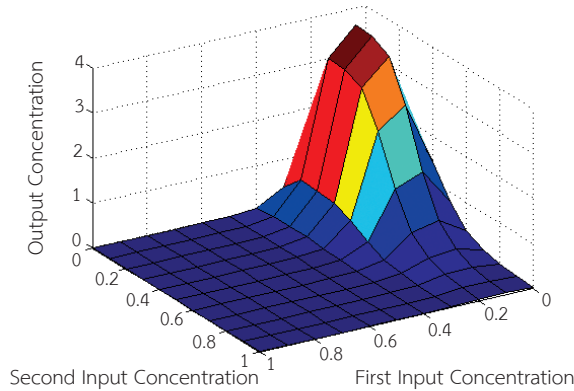


รูปที่ 7 ผลลัพธ์ของเกตสังเคราะห์ NAND

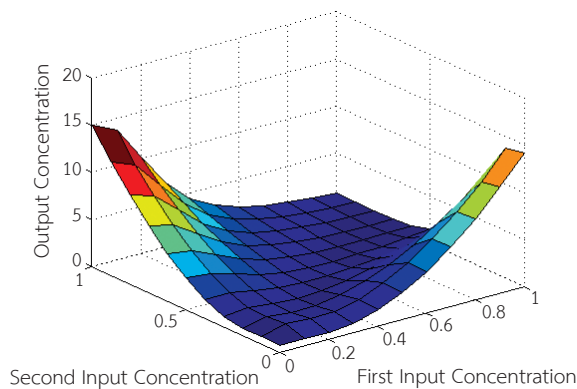
เพิ่มขึ้น และเมื่อมีการเพิ่มความเข้มข้นของ Transcription Factor ทั้งสองจนถึง 1 จะส่งผลให้การผลิตโปรตีนถูกยับยั้ง จะไม่สามารถผลิตได้ ดังรูปที่ 7 โดยค่าพารามิเตอร์ที่เหมาะสมสำหรับการสร้างเกตนี้ คือ  $R = 3.56$ ,  $K_R = 0.05$ ,  $K_1 = 0.05$ ,  $K_2 = 0.09$ ,  $h_1 = 3.95$ ,  $h_2 = 4.75$ ,  $h_3 = 3.74$ ,  $m_1 = 2.32$ ,  $m_2 = 0$ ,  $m_3 = 0$ ,  $m_4 = 0$

### 3.6 เกตตรรกะสังเคราะห์ “NOR”

เกตชนิดนี้ เป็นส่วนเติมเต็มของเกต OR กล่าวคือ ในช่วงแรกที่มีความเข้มข้นของ Transcription Factor ทั้งสองเป็น 0 นั้น โปรตีนจะถูกกระตุ้นให้มีการผลิต แต่เมื่อมีการเพิ่มความเข้มข้นของ Transcription Factor ทั้งสองจนถึง 1 โปรตีนผลผลิตจะถูกยับยั้งไม่ให้มีการผลิตออกมา ดังรูปที่ 8 โดยค่า



รูปที่ 8 ผลลัพธ์ของเกตสังเคราะห์ NOR



รูปที่ 9 ผลลัพธ์ของเกตสังเคราะห์ XOR

พารามิเตอร์ที่เหมาะสมสำหรับการสร้างเกตนี้ คือ  $R = 2.17$ ,  $K_R = 7.97$ ,  $K_1 = 0.25$ ,  $K_2 = 0.47$ ,  $h_1 = 8.4$ ,  $h_2 = 7.6$ ,  $h_3 = 6$ ,  $m_1 = 0.0013$ ,  $m_2 = 0.0023$ ,  $m_3 = 0.0019$ ,  $m_4 = 0.0007$

### 3.7 เกตตรรกะสังเคราะห์ “XOR”

ค่าพารามิเตอร์ที่เหมาะสมที่ได้จากอัลกอริทึมสำหรับการสร้างเกตนี้คือ  $R = 1.4$ ,  $K_R = 3.58$ ,  $K_1 = 0.27$ ,  $K_2 = 0.26$ ,  $h_1 = 1.78$ ,  $h_2 = 2.3$ ,  $h_3 = 3.46$ ,  $m_1 = 0.0035$ ,  $m_2 = 1.88$ ,  $m_3 = 1.58$ ,  $m_4 = 0.0024$

จากรูปที่ 9 จะเห็นว่า โปรตีนจะไม่มีการผลิตออกมาเมื่อค่าของ Transcription Factor ทั้งสองมีค่าเป็น 0 หรือ 1 พร้อมกัน แต่จะผลิตโปรตีนออกมาเมื่อค่าของ Transcription Factor ทั้งสองมีค่าต่างกัน หรือเมื่อมีตัวใดตัวหนึ่งเป็น 1 เท่านั้น

## 4. อภิปรายผลและสรุป

ในงานวิจัยนี้ได้ทำการศึกษา ทดลอง และนำเสนอแบบจำลองของเกตตรรกะสังเคราะห์สำหรับการลอกรหัสโดยใช้กระบวนการขั้นตอนวิธีเชิงพันธุกรรมในการหาค่าพารามิเตอร์ที่เหมาะสม ซึ่งจากการทดลองพบว่าเกตตรรกะสังเคราะห์ NOT, BUFFER, AND, OR, NAND, NOR และ XOR ที่ถูกออกแบบจากพารามิเตอร์เหล่านี้ สามารถแสดงผลลัพธ์ได้เสมือนเกตตรรกะอิเล็กทรอนิกส์ โดยเมื่อมีการป้อนค่าปัจจัยของการลอกรหัส (Transcription Factor) ที่แตกต่างกันออกไปให้กับเกตตรรกะสังเคราะห์นี้แล้ว ผลลัพธ์จากการจำลองการทำงานนั้นสอดคล้องตารางค่าความจริง

แบบจำลองนี้สามารถแสดงผลลัพธ์ของการแสดงออกของยีนเมื่อมีปัจจัยมากระตุ้นได้ โดยอ้างอิงกับหลักการการทำงานของเกตตรรกะพื้นฐาน และจากอัลกอริทึมที่พัฒนาขึ้นมาสามารถหาค่าพารามิเตอร์ได้อย่างเหมาะสม

ในอนาคต เมื่อเทคโนโลยีมีความก้าวหน้ามากขึ้นสามารถขยายองค์ความรู้ทางด้านนี้ ด้วยการสร้างวงจรสังเคราะห์เหล่านี้ขึ้นมา แล้วนำมาประยุกต์ใช้ร่วมกันหรือทดแทนเซลล์ของสิ่งมีชีวิตได้ เพื่อให้สามารถสร้างผลผลิตหรือกำหนดการแสดงออกของยีน ซึ่งจะเกิดประโยชน์เป็นอย่างมากกับสิ่งมีชีวิต

## 5. กิตติกรรมประกาศ

ขอแสดงความขอบคุณ Professor Chun-Liang Lin, Department of Electrical Engineering, National Chung Hsing University, Taiwan ที่ให้คำปรึกษาและข้อแนะนำที่เป็นประโยชน์สำหรับงานวิจัยนี้ และขอขอบคุณคณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน ที่ให้การสนับสนุนงานวิจัยนี้จนสำเร็จลุล่วงไปได้ด้วยดี

## เอกสารอ้างอิง

- [1] E. Andrianantoandro, S. Basu, D. K. Karig, and R. Weiss, “Synthetic biology: New engineering rules for an emerging discipline,” *Molecular*



- Systems Biology*, vol. 2, pp. 1–14, 2006.
- [2] G. Cuccato, G. D. Gatta, and D. di Bernardo, “Systems and synthetic biology: Tackling genetic networks and complex diseases,” *Heredity*, vol. 102, no. 6, pp. 527–532, 2009.
- [3] A. S. Khalil and J. J. Collins, “Synthetic biology: Applications come of age,” *Nature Reviews Genetics*, vol. 11, no. 5, pp. 367–379, 2010.
- [4] Y. Naasse, H. Charoute, B. E. Houate, C. Elbekkay, L. Razoki, A. Malki, A. Barakat, and H. Rouba, “Chromosomal abnormalities and Y chromosome microdeletions in infertile men from Morocco,” *BMC Urology*, vol. 95, no. 15, pp. 2–6, 2015.
- [5] L. Bintu, N. E. Buchler, H. G. Garcia, U. Gerland, T. Hwa, J. Kondev, T. Kuhlman, and R. Phillips, “Transcriptional regulation by the numbers: Applications,” *Current Opinion in Genetics & Development*, vol. 15, no. 2, pp. 125–135, 2005.
- [6] M. N. Win and C. D. Smolke, “Higher-order cellular information processing with synthetic RNA devices,” *Science*, vol. 322, no. 5900, pp. 456–460, 2008.
- [7] S. Ausländer, D. Ausländer, M. Müller, M. Wieland, and M. Fussenegger, “Programmable single-cell mammalian biocomputers,” *Nature*, vol. 487, pp. 123–127, 2012.
- [8] K. Hattori, “Advanced biological computer developed,” ScienceDaily, American Technion Society, Rockville, MD, May 2013.
- [9] S. Konur and M. Gheorghe, “Design and analysis of genetically constructed logic gates,” in *Proceedings of The University of Sheffield Engineering Symposium Conference*, 2014, pp. 31–32.
- [10] A. Wong, H. Wang, C. L. Poh, and R. I. Kitney, “Layering genetic circuits to build a single cell, bacterial half adder,” *BMC Biology*, vol. 13, no. 1, pp. 40, 2015.
- [11] A. E. Mayo, Y. Setty, S. Shavit, A. Zaslaver, and U. Alon, “Plasticity of the cis-regulatory input function of a gene,” *PLoS Biology*, vol. 4, no. 4, 2006.
- [12] C. H. Yuh, H. Bolouri, and E. H. Davidson, “Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene,” *Science*, vol. 279, no. 5358, pp. 1896–1902, 1998.
- [13] N. E. Buchler, U. Gerland, and T. Hwa, “On schemes of combinatorial transcription logic,” *PNAS*, vol. 100, no. 9, pp. 5136–5141, 2003.
- [14] N. Ichinose, T. Yada, O. Gotoh, and K. Aihara, “Reconstruction of transcription–translation dynamics with a model of gene networks,” *Journal of Theoretical Biology*, vol. 255, no. 4, pp. 378–386, 2008.
- [15] Y. Maki, Y. Takahashi, Y. Arikawa, S. Watanabe, K. Aoshima, Y. Eguchi, T. Ueda, S. Aburatani, S. Kuhara, and S. Okamoto, “An integrated comprehensive workbench for inferring genetic networks: Voyagene,” *Journal of Bioinformatics and Computational Biology*, vol. 2, no. 3, pp. 533, 2004.
- [16] R. S. Cox, M. G. Surette, and M. B. Elowitz, “Programming gene expression with combinatorial promoters,” *Molecular Systems Biology*, vol. 3, pp. 145, 2007.
- [17] C. H. Chuang, C. L. Lin, Y. C. Chang, T. Jennawasin, and P. K. Chen, “Design of synthetic biological logic circuits based on evolutionary algorithm,”



- IET Systems Biology*, vol. 7, no. 4, pp. 89–105, 2013.
- [18] T. Pencheva, K. Atanassov, and A. Shannon, “Modelling of a roulette wheel selection operator in genetic algorithms using generalized nets,” *International Journal Bioautomation*, vol. 13, no. 4, pp. 257–264, 2009.
- [19] J. M. Mendes, “A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem,” *WSEAS Transactions on Computers*, vol. 12, no. 4, pp. 164–173, 2013.