

# วิธีการเชิงพันธุกรรมแบบขนานสำหรับคำถามที่เหมาะสมที่สุด แบบกระจายของการ JOIN จำนวนมาก

PARALLEL GENETIC ALGORITHMS FOR DISTRIBUTED QUERY OPTIMIZATION OF LARGE JOIN

## พิศาล สุขชี

นักศึกษาระดับปริญญาตรี วิทยาลัยวิศวกรรมศาสตร์ มหาวิทยาลัยศิลปากร  
E-mail : phisan.shukki@gmail.com

## สุนีย์ พงษ์พินิจบุญญ

อาจารย์ประจำภาควิชาคอมพิวเตอร์  
คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร  
E-mail : sunee@su.ac.th

## บทคัดย่อ

งานวิจัยนี้นำเสนอวิธีการสำหรับแก้ปัญหาเรื่องการค้นหาคำถามที่เหมาะสมที่สุดสำหรับการ Join จำนวนมาก (Distributed Query Optimization of Large Join) บนฐานข้อมูลแบบกระจาย (Distributed Database) ซึ่งปัญหานี้จัดเป็นปัญหา NP-Complete Combinatorial Optimization งานวิจัยนี้มีวัตถุประสงค์เพื่อปรับปรุงเวลาที่ใช้ในการค้นหาวิธีเลือกคิวรี่ที่เร็วที่สุดของระบบฐานข้อมูลแบบกระจายโดยใช้เทคนิค Island-based Parallel Genetic Algorithm โดยเทคนิคนี้ถูกนำมาประยุกต์ใช้กับการคำนวณแบบขนานบนระบบคอมพิวเตอร์คลัสเตอร์ โดยเทคนิคที่ถูกนำมาเสนอนี้ไม่เพียงสามารถลดเวลาในการค้นหาได้อย่างมีประสิทธิภาพ แต่ยังสามารถให้คำตอบที่เป็นคำตอบที่มีความเหมาะสมที่สุดได้อย่างรวดเร็ว

**คำสำคัญ :** คำถามที่เหมาะสมที่สุดแบบกระจาย การ Join จำนวนมาก ฐานข้อมูลแบบกระจาย วิธีการเชิงพันธุกรรมแบบขนาน คิวรี่เลือกคิวรี่ที่เร็วที่สุด เอ็นพีบริบูรณ์

## ABSTRACT

This research presents the method for solving a problem area of Distributed Query Optimization of Large Join on Distributed Database which is the NP-complete combinatorial optimization problem. The main aim of this research work is to improve the search time of Distributed Database systems, in finding the best query execution plan by using the well-known search technique, Island-based Parallel Genetic Algorithm which is applied with parallel computing on the PC Computer Clusters. The proposed enhanced technique not only can reduce the search time efficiently but it also can give the fast convergence to optimal solutions

**KEYWORDS :** Distributed query optimization, Large join, Distributed database, Island-based parallel genetic algorithm, Query execution plan, NP-complete

## บทนำ

ข้อมูลที่อยู่ภายในระบบฐานข้อมูลแบบกระจายจะถูกกระจาย ออกไปจัดเก็บยังสถานที่ (Site) สำหรับจัดเก็บข้อมูลที่แตกต่างกันภายในระบบซึ่งถูกเชื่อมต่อกันผ่านระบบเครือข่าย (Network System) ดังนั้นการรวมข้อมูล (Join) ระหว่างรีเลชัน (Relations) ที่เกิดขึ้นภายในระบบฐานข้อมูลแบบกระจายต้องกระทำโดยมีการรับส่งข้อมูลระหว่างแหล่งข้อมูลต่างๆ เพื่อนำข้อมูลไปทำการประมวลผล เพื่อให้ได้ผลลัพธ์ตามคำสั่งสอบถามข้อมูล (Input Query) ดังนั้นสิ่งสำคัญสิ่งหนึ่งที่จะต้องคำนึงถึงสำหรับการรวมข้อมูล คือเวลาที่จะต้องสูญเสียไปสำหรับการดำเนินการ ซึ่งเป็นผลรวมระหว่างเวลาที่จะต้องเสียไปจากการประมวลผลข้อมูลและเวลาที่จะต้องเสียไปจากการรับส่งข้อมูลระหว่างแหล่งข้อมูลต่างๆ ผ่านระบบเครือข่าย

การรวมข้อมูลแต่ละครั้งหากมีการดำเนินการโดยมีการจัดลำดับ การรวมข้อมูลระหว่างคู่ของรีเลชันที่ไม่เหมาะสมแล้วอาจก่อให้เกิดผลลัพธ์ชั่วคราว (Intermediate Result) ที่มีขนาดใหญ่โดยไม่จำเป็น ซึ่งจะต้องสูญเสียเวลาที่ใช้ในการรับส่งข้อมูลและการประมวลผลเพิ่มขึ้น ทำให้ประสิทธิภาพในการดำเนินการด้อยลงไป ดังนั้นการคัดเลือกคิวรีเอ็กคิวทีพืชนั้น (Query Execution Plan: QEP) ที่แสดงถึงลำดับการดำเนินการเพื่อรวมข้อมูลระหว่างรีเลชันที่ดีที่สุดเพื่อใช้ในการดำเนินการประมวลผลเพื่อให้ได้ผลลัพธ์ในแต่ละครั้งจึงเป็นเรื่องที่จำเป็น

แต่อย่างไรก็ตามการคัดเลือกคิวรีเอ็กคิวทีพืชนั้นที่ที่สูงสุดในกรณีที่มีการรวมข้อมูลระหว่างรีเลชันตั้งแต่ 10 รีเลชันขึ้นไป นับว่าเป็นปัญหา NP-Complete Combinatorial Optimization อีกรูปแบบหนึ่ง เพราะมีจำนวนของคิวรีเอ็กคิวทีพืชนั้นที่เป็นไปได้ จำนวนมหาศาล ซึ่งต้องใช้เวลาในการดำเนินการคัดเลือกที่นานด้วยเช่นกัน และที่สำคัญอย่างยิ่งการค้นหาคิวรีเอ็กคิวทีพืชนั้นที่มีความเหมาะสมที่สุดจะต้องทำการคำนวณและค้นหาเป็นแบบครั้งต่อครั้ง สำหรับแต่ละคำสั่งสอบถามข้อมูลที่ถูกส่งเข้ามา เพราะระบบไม่สามารถที่จะเก็บคำตอบที่ได้ก่อนหน้าเอาไว้ใช้เป็นคำตอบสำหรับคำสั่งสอบถามข้อมูลอื่นๆ ได้ เนื่องจากระบบฐานข้อมูลแบบกระจายนั้นมีปัจจัยแวดล้อมที่สามารถเปลี่ยนแปลงได้ตลอดเวลา เช่น อัตราการขนส่งข้อมูลระหว่างแหล่งข้อมูลผ่านระบบเครือข่าย (Data Transfer Rate) หรือจำนวน

แถวข้อมูลของแต่ละรีเลชัน (Cardinality) ที่มีการเปลี่ยนแปลงเนื่องจากการใช้งานของผู้ใช้

ในปัจจุบันมีแนวโน้มว่าโปรแกรมประยุกต์ หลายๆ ประเภทที่ใช้งานระบบฐานข้อมูลแบบกระจายสำหรับจัดการบริหารข้อมูลอย่างเช่น ระบบช่วยสนับสนุนการตัดสินใจ (Decision Support System) หรือระบบคลังข้อมูล (Data Warehouse) ซึ่งอาจต้องมีการดำเนินการในลักษณะที่ทำให้เกิดการรวมข้อมูลของรีเลชันจำนวนมาก (Large Join Query) เกิดขึ้น และสำหรับในเรื่องของปัญหาที่ต้องการการคำนวณที่มีความซับซ้อน และต้องใช้เวลานานในการคำนวณเพื่อให้ได้คำตอบสำหรับปัญหานั้น ปัจจุบันได้มีการทดลองแก้ปัญหาโดยใช้วิธีการคำนวณแบบขนาน (Parallel Computing) เข้ามาประยุกต์เพื่อแก้ปัญหา ซึ่งประสบความสำเร็จและได้รับการยอมรับเป็นอย่างมาก

ดังนั้นเพื่อให้การแก้ปัญหาเรื่องการค้นหาคำถามที่เหมาะสมที่สุดสำหรับการรวมข้อมูลระหว่างรีเลชันจำนวนมากเป็นไปอย่างสอดคล้องกับการพัฒนาในปัจจุบัน ผู้วิจัยจึงได้นำวิธีการ Island-based PGA ซึ่งเป็นวิธีการค้นหาโดยใช้การคำนวณแบบขนานเข้ามาทดสอบเพื่อแก้ปัญหา โดยมีจุดประสงค์ที่จะพัฒนากระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจาย (Distributed Query Optimization) บนระบบฐานข้อมูลแบบกระจายให้สามารถตอบสนองต่อลักษณะการทำงานในปัจจุบันได้ดียิ่งขึ้น

สำหรับการดำเนินงานทดลอง ผู้วิจัยได้ทดสอบเพื่อแก้ปัญหาโดยการจำลอง (Simulated) การทำงานของกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายให้มีการทำงานอยู่ภายใต้ปัจจัยแวดล้อมที่ผู้วิจัยกำหนดขึ้น โดยในการวัดประสิทธิภาพของผลการทดลองได้ทำการเปรียบเทียบประสิทธิภาพการทำงานระหว่างกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายแบบที่ใช้ Island-base PGA กับแบบที่ใช้ Sequential GA ทั้งในด้านความรวดเร็วในการค้นหาคิวรีเอ็กคิวทีพืชนั้นที่มีความเหมาะสมที่สุด และด้านของคุณภาพของคิวรีเอ็กคิวทีพืชนั้นที่ได้จากการค้นหา และสำหรับเทคนิคที่ใช้เพื่อการคำนวณแบบขนาน ผู้วิจัยใช้มาตรฐาน Message Passing Interface version 2 (MPI2) บนระบบคอมพิวเตอร์คลัสเตอร์ (Computer Cluster) สำหรับดำเนินการทดลอง

## งานวิจัยที่เกี่ยวข้อง

Swami และ Gupta (1988), Ioannidis และ Kang (1990) นำเสนอการแก้ปัญหาเรื่องการค้นหาคำตอบที่เหมาะสมที่สุด (Query Optimization) ในกรณีที่เกิดการรวมข้อมูลระหว่างรีเลชันเป็นจำนวนมากโดยการนำ Iterative Improvement Algorithm และ Simulated Annealing Algorithm ทำการเปรียบเทียบประสิทธิภาพในการทำงาน แต่ได้มีการนำเสนอวิธีการทดลองและข้อสรุปที่แตกต่างกัน

Sunee (1996) นำเสนอการแก้ปัญหาที่เน้น ฐานข้อมูลแบบกระจายโดยพัฒนาอัลกอริทึมชื่อว่า Two-Stage Simulated Annealing Algorithm เป็นการทำงานร่วมกันระหว่างวิธีการแบบฮิวริสติก (Heuristic) กับวิธีการอบเหนียวจำลอง (Simulated Annealing)

Bennett, Ferris และ Ioannidis (1991) นำเสนอวิธีการแก้ปัญหาโดยใช้ GA ซึ่งการทดลองได้จำกัดจำนวนของรีเลชันไว้เพียง 16 รีเลชัน และทำการแบ่งพื้นที่การค้นหา (search space) ออกเป็นสองประเภทคือ ประเภทแรกประกอบด้วย Left-Deep Tree เพียงอย่างเดียว และประเภทที่สองประกอบไปด้วยทั้ง Left-Deep Tree ผสมกับ Bushy Tree

Stillger และ Spiliopoulou (1996) นำเสนอวิธีการแก้ปัญหาโดยใช้โปรแกรมเชิงพันธุกรรม (Genetic Programming: GP) แทนเพื่อเป็นการลดขั้นตอนและความซับซ้อนที่เกิดจาก GA

Dong และ Liang (2007) นำเสนอการแก้ปัญหานี้โดยใช้ GA แต่มุ่งประเด็นศึกษาที่การทำงานของ GA บน Query Models ในแต่ละแบบ Island-based Parallel Genetic Algorithm (Island-based PGA)

เป็นอีกหนึ่งเทคนิคสำหรับวิธีการเชิงพันธุกรรมแบบขนาน (Parallel Genetic Algorithm) ซึ่งได้รับแรงบันดาลใจจากความเป็นจริงในธรรมชาติ โดยพิจารณาว่าประชากรกลุ่มใหญ่ประกอบไปด้วยประชากรกลุ่มย่อยๆ (Sub-Population) ชนิดเดียวกันหลายกลุ่ม แต่ถูกแบ่งแยกจากกันด้วยข้อจำกัดของขนาดพื้นที่หรือสภาพแวดล้อม และประชากรแต่ละกลุ่มนั้นมีทิศทาง ในการวิวัฒนาการเป็นของตัวเอง ไม่ได้ขึ้นอยู่กับประชากรกลุ่มอื่นๆ ซึ่งจุดนี้เองที่ทำให้เกิดความหลากหลายมากยิ่งขึ้นในการวิวัฒนาการ และในบางกรณีประชากรกลุ่มย่อยๆ อาจมีการแลกเปลี่ยนประชากรที่มีคุณภาพดีระหว่างกลุ่มกันได้

ทำให้แต่ละกลุ่มสามารถนำส่วนที่ดีที่สุดไปใช้ในการวิวัฒนาการร่วมกัน การทำงานของ Island-based PGA สำหรับงานวิจัยนี้ เริ่มต้นโดยแต่ละหน่วยประมวลผล (Processor) บนระบบคอมพิวเตอร์ คลัสเตอร์ สุ่มเลือกประชากรกลุ่มย่อยของตนเองขึ้นมาตามขนาดของประชากรที่กำหนด และดำเนินกระบวนการ Sequential GA กับกลุ่มประชากรของตนเอง เมื่อแต่ละกลุ่มได้ทำการวิวัฒนาการจนมีจำนวนรุ่นของการวิวัฒนาการ (Generation) ตามจำนวนที่ได้กำหนดไว้ แต่ละหน่วยประมวลผล จะทำการแลกเปลี่ยน (Migration) กลุ่มสมาชิกที่ดีที่สุดจำนวนหนึ่งกับกลุ่มประชากรย่อยอื่นๆ และดำเนินกระบวนการในรูปแบบนี้ต่อไปเรื่อยๆ เมื่อกระบวนการทั้งหมดสิ้นสุดลง จะทำการคัดเลือกผลลัพธ์ที่ดีที่สุดจากประชากรกลุ่มย่อยทั้งหมดเพื่อใช้เป็นผลลัพธ์ในกระบวนการค้นหาคำตอบที่เหมาะสมที่สุดแบบกระจาย

## ขั้นตอนสำหรับ Island-base PGA

สามารถบรรยายถึงรายละเอียดของ Island-based PGA ที่ประยุกต์ใช้ในงานวิจัยนี้ได้ดังต่อไปนี้

**Initialize sub-population** แต่ละหน่วยประมวลผลบนระบบ จะทำการสร้างกลุ่มประชากรต้นแบบของตัวเองขึ้นมา โดยใช้วิธีการสุ่มเลือก จนได้กลุ่มประชากรย่อย (Sub-Population Size) ที่มีขนาดตามที่กำหนดไว้

**Fitness Function** แต่ละหน่วยประมวลผลจะทำการหาค่าความเหมาะสมของสมาชิกแต่ละตัวภายในกลุ่มประชากรย่อยของตนเอง ซึ่งวิธีสำหรับการคิดค่าความเหมาะสมสำหรับประชากรนั้นจะอยู่บนพื้นฐาน การคำนวณโดยใช้โมเดลการคำนวณค่าใช้จ่ายแบบกระจาย (Distributed Cost Model) ที่กำหนดไว้

**Selection** ใช้วิธีการคัดเลือกแบบทัวร์นาเมนต์ (Tournament) ซึ่งจะดำเนินการโดยสุ่มเลือกสมาชิกในกลุ่มประชากรออกมาครั้งละ 10 คิวรีเอ็กซีคิวชันแพลนแล้วเปรียบเทียบเอา คิวรีเอ็กซีคิวชันแพลนที่มีค่าความเหมาะสมดีที่สุด หรือกล่าวได้ว่ามีค่าใช้จ่ายที่ต้องสูญเสีย (Cost) ที่น้อยที่สุดซึ่งได้จากการคำนวณในฟังก์ชันหาค่าความเหมาะสม (Fitness Function) เพื่อนำไปใช้สำหรับดำเนินการในขั้นตอน การพัฒนาทางสายพันธุ์ต่อไป

**Genetic Operators** การเกิดครอสโอเวอร์ (Crossover) และมิวเทชัน (Mutation) เพื่อสร้างประชากรรุ่นถัดไปนั้นจะถูก

ควบคุมโดยค่าความน่าจะเป็น และวิธีการสำหรับครอสโอเวอร์ เป็นแบบ Sub-tree Crossover และสำหรับวิธีการมิวเทชันเป็นแบบ Swap Node Mutation

**Migration** เป็นการแลกเปลี่ยนกลุ่มสมาชิกของประชากรในแต่ละหน่วยประมวลผลกับ หน่วยข้างเคียง โดยการแลกเปลี่ยนแต่ละครั้งได้กำหนดให้กลุ่มสมาชิกที่ถูกแลกเปลี่ยน (Migrants) จะถูกคัดเลือกจากคิวรีเอ็กซีคิวทีวชันแพลนที่มีค่าน้อยที่สุด ลดหลั่นลงมาจนครบจำนวนที่ได้กำหนดไว้ เมื่อมีการรับส่งข้อมูลกันแล้วแต่ละหน่วยประมวลผลจะนำกลุ่มสมาชิกใหม่ที่ได้รับไปแทนที่กลุ่มสมาชิกเดิมที่อยู่ในเกณฑ์ที่อ่อนแอ หรือกลุ่มของคิวรีเอ็กซีคิวทีวชันแพลนที่มีค่าใช้จ่ายที่ต้องสูญเสียมากที่สุดตามลำดับ และรูปแบบการรับส่งข้อมูลที่เกิดขึ้นในระบบคอมพิวเตอร์คลาสเตอร์เป็นแบบวงแหวน (Ring Topology)

### การกำหนดปัจจัยสำหรับการทดลอง

สามารถบรรยายถึงปัจจัยสำหรับการทดลองที่ได้กำหนดขึ้น ในงานวิจัยนี้ได้ดังต่อไปนี้

#### 1. Query Model

งานวิจัยนี้เลือกทำการทดลองกับ Query Model แบบ Linear Query Model ที่ประกอบไปด้วยการรวมข้อมูลแบบ Equality Join ซึ่งจะสามารถแสดงความสัมพันธ์และเงื่อนไขได้ในรูปของกราฟแสดงความสัมพันธ์ของการรวมข้อมูล (Join Graph) ที่เป็นเส้นตรง และจะให้พื้นที่การค้นหาที่มีขนาดเล็กกว่าแบบ Star query model (Sunee, 1996)

#### 2. Search Space

สำหรับกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายจะใช้กราฟแสดงความสัมพันธ์ของการรวมข้อมูลเป็นข้อมูลตั้งต้น (Input) สำหรับดำเนินการกระบวนการเพื่อค้นหาคิวรีเอ็กซีคิวทีวชันแพลนที่ดีที่สุดและพื้นที่การค้นหาที่ใช้ในการดำเนินการทดลองนั้น เกิดขึ้นจากการรวมข้อมูลของรีเลชัน ระหว่าง 10-40 รีเลชัน และคิวรีเอ็กซีคิวทีวชันแพลนที่เป็นสมาชิกในแต่พื้นที่การค้นหานั้นจะมีรูปร่างทั้งแบบ Linear Tree และ Bushy Tree

#### 3. การสร้างคิวรีเอ็กซีคิวทีวชันแพลน

ในขั้นตอนการสร้างกลุ่มประชากรต้นแบบ (Initialize Population) จะทำการสร้างคิวรีเอ็กซีคิวทีวชันแพลนขึ้นมาใหม่จากการสุ่มโดยมีข้อมูลเริ่มต้นจากกราฟแสดงความสัมพันธ์ของการ

รวมข้อมูลและขั้นตอนครอสโอเวอร์และมิวเทชันนั้นจะทำการสร้างคิวรีเอ็กซีคิวทีวชันแพลนขึ้นมาใหม่จากสมาชิกเดิมภายในพื้นที่การค้นหาโดยการสร้างคิวรีเอ็กซีคิวทีวชันแพลนในกระบวนการต่างๆ จะต้องหลีกเลี่ยงการสร้างคิวรีเอ็กซีคิวทีวชันแพลนที่มีลำดับการรวมข้อมูลซึ่งก่อให้เกิดผลคูณคาร์ทีเซียน (Cartesian Product) ซึ่งจะเป็นการลดจำนวนของคิวรีเอ็กซีคิวทีวชันแพลนที่ไม่สามารถใช้งานได้

รูปร่างของคิวรีเอ็กซีคิวทีวชันแพลนที่ถูกสร้างนั้นมีโอกาสเป็นไปได้อย่างหลากหลายไม่ว่าจะเป็น Linear Tree และ Bushy Tree สำหรับ Bushy Tree เป็นรูปร่างที่มีความซับซ้อนต้องใช้เวลาสำหรับการคิดคำนวณค่าใช้จ่ายที่ต้องสูญเสีย (Cost) ที่จะเกิดขึ้นนานกว่าและการดำเนินการตามกระบวนการต่างๆ ของ GA ทำได้ยากและช้ากว่าแบบ Linear Tree แต่อย่างไรก็ตามในระบบฐานข้อมูลแบบกระจายคิวรีเอ็กซีคิวทีวชันแพลนที่มีรูปร่างแบบ Bushy Tree สามารถที่จะนำไปทำการประมวลผลเพื่อหาคำตอบในรูปแบบขนานได้ซึ่งจะทำให้สามารถลดเวลาประมวลผลจริง (Response Time) ที่จะเกิดขึ้นลงไปได้อีก ดังนั้นการพิจารณาคิวรีเอ็กซีคิวทีวชันแพลนที่มีรูปร่างแบบ Bushy Tree จึงเป็นเรื่องที่จะต้องให้ความสำคัญ

#### 4. Distributed Cost Model

ขั้นตอนฟังก์ชันหาค่าความเหมาะสมของ GA ที่ใช้ในงานวิจัยนี้ใช้วิธีการคำนวณ โดยใช้โมเดลการคำนวณค่าใช้จ่ายแบบกระจาย ซึ่งก็คือสูตรคำนวณที่ใช้สำหรับการคำนวณเวลารวม (Total Time) เพื่อทำนายเวลาที่จะต้องเสียไปหากมีการนำคิวรีเอ็กซีคิวทีวชันแพลนนั้นไปดำเนินการจริงๆ ซึ่งงานวิจัยนี้ใช้โมเดล การคำนวณค่าใช้จ่าย แบบกระจายแบบทั่วไปที่ไม่ได้อาศัยอยู่กับวิธีการรวมข้อมูลแบบใดแบบหนึ่ง (Sunee, 1996), (Ozsu และ Valduriez, 1991)

$$\text{Total\_Cost} = \text{Local\_Processing} + \text{Transfer\_cost} \quad (1)$$

โดยสามารถอธิบายสมการที่ 1 ได้ดังนี้

**Local Processing** คือ เวลารวมที่เกิดจากเวลาที่ต้องสูญเสียจากการประมวลผลคำสั่งต่างๆ ในระดับ หน่วยประมวลผล และเวลาที่ต้องสูญเสียไปจากการเข้าถึงข้อมูลในระดับอินพุต/เอาต์พุต

**Transfer Cost** คือเวลารวมทั้งหมดที่ต้องสูญเสียไปในการรับส่งข้อมูลระหว่างแหล่งข้อมูลผ่านระบบเครือข่าย ซึ่งเป็นผลรวมของขนาดของข้อมูลคูณกับอัตราการขนส่งข้อมูลผ่านระบบเครือข่าย

อย่างไรก็ตามหากพิจารณาถึงข้อมูลในปัจจุบัน ประสิทธิภาพของเครื่องคอมพิวเตอร์ต่างๆ นั้นไม่ค่อยมีความแตกต่างกันมากนักไม่ว่าจะเป็นประสิทธิภาพในการคำนวณ หรือประสิทธิภาพในการเข้าถึงข้อมูลในหน่วยความจำ หรือการทำงานด้าน อินพุต/เอาท์พุต ดังนั้นสำหรับงานวิจัยนี้ในส่วนของโมเดลการคำนวณค่าใช้จ่ายแบบกระจายจึงพิจารณาตัดการคำนวณส่วนของ Local Processing ออกไปเนื่องจากพิจารณาว่าในแหล่งข้อมูลต่างๆ นั้นมีประสิทธิภาพการคำนวณระดับ Local Processing เท่ากัน (Ozsu และ Valduriez, 1991) (Stillger, 1996) ดังนั้น โมเดลการคำนวณค่าใช้จ่ายแบบกระจายในงานวิจัยนี้จึงมุ่งเน้นที่จะทำการคำนวณหาค่าเวลารวมทั้งหมดที่เกิดจากการขนส่งข้อมูลผ่านระบบเครือข่าย โดยมีปัจจัยที่สำคัญสำหรับการคำนวณคือ ขนาดของข้อมูลที่จะต้องรับส่งและอัตราการขนส่งข้อมูล (Data Transfer Rate) ระหว่างแหล่งข้อมูลต่างๆ

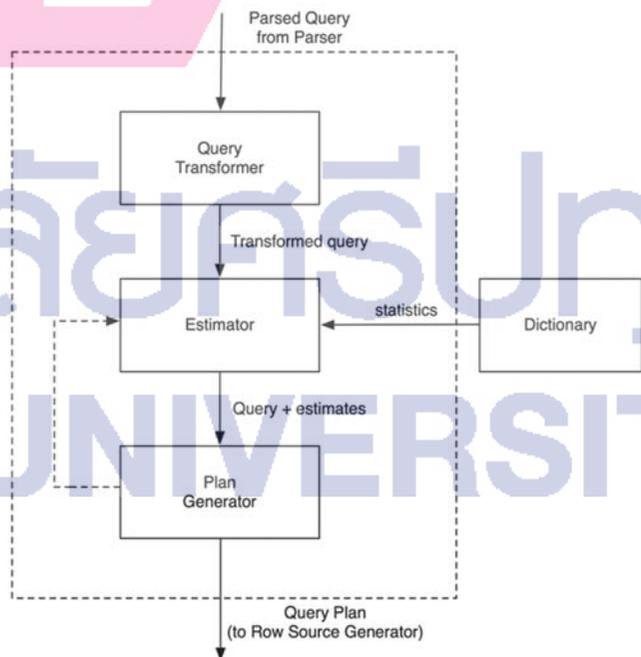
### 5. จำนวนหน่วยประมวลผล

การกำหนดจำนวนหน่วยประมวลผลเป็นอีกหนึ่งปัจจัยการทดลองที่สำคัญสำหรับ Island-based PGA ซึ่งจะมีการแบ่งการทำงานออกเป็นโปรเซสย่อยๆ ทำงานอยู่บนหน่วยประมวลผลต่างๆ ถึงแม้แต่ละโปรเซสจะทำงานอย่างเป็นอิสระกัน แต่จำเป็นต้องมีการแลกเปลี่ยนข้อมูลเพื่อทำให้มีการวิวัฒนาการที่ดียิ่งขึ้นโดยใช้ส่วนใดร่วมกัน การสื่อสารข้อมูลระหว่างกันจะต้องแลกเปลี่ยนข้อมูลผ่านสายสัญญาณเครือข่ายภายในของระบบคลัสเตอร์ ซึ่งหากมีการสื่อสารข้อมูลเป็นจำนวนมากพร้อมๆ กันอาจทำให้รับส่งข้อมูลระหว่างกันล่าช้าตามไปด้วย และอาจเป็นสาเหตุที่ทำให้กระบวนการต่างๆ ของ Island-based PGA ล่าช้าตามไปดังนั้นผู้วิจัยจึงได้เลือกที่จะทำการใช้จำนวนของหน่วยประมวลผลเพียง 4 หน่วยประมวลผล และสำหรับการทำงานของ Island-based PGA จะแบ่งการทำงานออกเป็น 4 โปรเซสย่อยต่อการทำงานหนึ่งครั้ง

### การทดลองและผลการทดลอง

ผู้วิจัยได้ดำเนินการทดลองโดยการจำลองแบบการทำงานของกระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจายขึ้น ซึ่งโครงสร้างของกระบวนการที่ผู้วิจัยได้ทำการจำลองขึ้น สามารถแสดงได้รับภาพที่ 1 (Oracle, 2009)

การทดลองที่เกิดขึ้นได้ดำเนินการบนระบบคอมพิวเตอร์คลัสเตอร์ ซึ่งประกอบไปด้วยเครื่อง Front-end 1 เครื่อง และเครื่อง Compute node 3 เครื่อง ทั้งหมดเป็นเครื่อง Intel Xeon 2.80GHz ขนาดของ Cache 2M และขนาด Memory 4Gb สำหรับการพัฒนาในด้านโปรแกรมผู้วิจัยได้ใช้เครื่องมือหลักที่เป็นโอเพนซอร์สสำหรับการพัฒนา ดังนี้คือ JSqlParser (2008) สำหรับพัฒนาในส่วนของการวิเคราะห์และตรวจสอบความถูกต้องของคำสั่งสอบถามข้อมูล, JAXB (2008) ในขั้นตอนการรับส่งข้อมูลสถิติของฐานข้อมูล (Database Statistics) และข้อมูลพื้นฐานของฐานข้อมูล (Database Profile) ในรูปแบบของ XML จากพจนานุกรมข้อมูล (Data Dictionary) และสำหรับส่วนของขั้นตอนสำหรับ GA ผู้วิจัยได้ทำการแก้ไขดัดแปลงการทำงานจาก TinyGP (Riccardo และคณะ, 2008)



ภาพที่ 1 กระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจาย

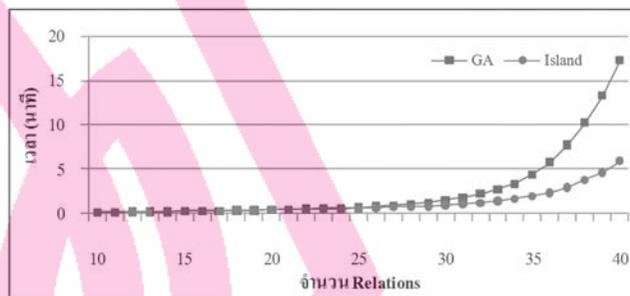
สำหรับการทดลองเพื่อวัดประสิทธิภาพนั้น ผู้วิจัยได้ทำการทดสอบในเชิงเปรียบเทียบผลของการทำงานระหว่างกระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจายแบบที่ใช้ Island-based PGA กับแบบที่ใช้ GA สำหรับค้นหา คิวรีเอ็กซีคิวทีชั้นแพลนที่มีความเหมาะสมที่สุด โดยผู้วิจัยได้เลือกทำการเปรียบเทียบการทำงานระหว่างด้านความเร็วในการทำงาน และคุณภาพของผลลัพธ์ที่ได้

สำหรับวิธีการในการทดสอบความรวดเร็วในการทำงานของกระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจายนั้น ผู้วิจัยได้ทำการออกแบบการทดลองโดยการให้กระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจายทั้งสองแบบทำการค้นหาค่าตอบของคำสั่งสอบถามข้อมูลที่มีการรวมข้อมูลระหว่างรีเลชันขนาดต่างๆ ตั้งแต่ 10-40 รีเลชัน และเพื่อให้ได้เวลาเฉลี่ยของการค้นหาค่าตอบ ผู้วิจัยได้ทำการทดลองโดยกำหนดให้กระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจายแต่ละแบบทำการค้นหาค่าตอบ และบันทึกเวลาทั้งหมด 10 ครั้งในแต่ละคำสั่งสอบถามข้อมูล

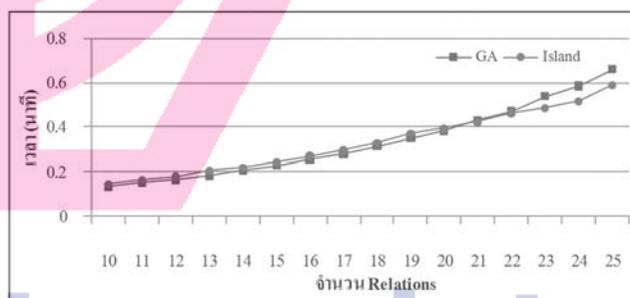
โดยผลการทดลองเพื่อวัดประสิทธิภาพด้านความเร็วสามารถแสดงได้ดังภาพที่ 2 โดยหากพิจารณาในช่วงที่มีการรวมข้อมูลระหว่างรีเลชันในช่วง 23-40 รีเลชันแสดงให้เห็นว่ากระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจายที่ใช้ Island-based PGA นั้นทำงานได้รวดเร็วกว่า แต่หากพิจารณาในช่วงที่มีการรวมข้อมูลระหว่างรีเลชันตั้งแต่ 10-22 รีเลชันปรากฏว่าแบบที่ใช้ GA สามารถทำงานได้อย่างรวดเร็วกว่าแบบที่ใช้ Island-based PGA ซึ่งแสดงผลการทดลองได้ดังภาพที่ 3

สำหรับวิธีในการทดสอบวัดประสิทธิภาพในด้านคุณภาพของผลลัพธ์ที่ได้จากกระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจายทั้งสองแบบผู้วิจัยได้ทำการออกแบบการทดลองโดยให้กระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจายทั้งสองทำการค้นหาค่าตอบบนพื้นที่การค้นหาเดียวกัน โดยใช้ข้อมูลพื้นฐานของฐานข้อมูลและข้อมูลสถิติของฐานข้อมูลเดียวกัน และทำการเปรียบเทียบคุณภาพของผลลัพธ์ที่ได้ โดยการรวมข้อมูลระหว่างรีเลชันแต่ละขนาด ผู้วิจัยได้ทำการทดสอบโดยเปลี่ยนข้อมูลพื้นฐานของฐานข้อมูล และข้อมูลสถิติของฐานข้อมูล ทั้งหมด 5 ครั้งและทำการบันทึกข้อมูลความถี่ของคุณภาพผลลัพธ์ที่ดีที่สุดที่ได้จากการทำงาน

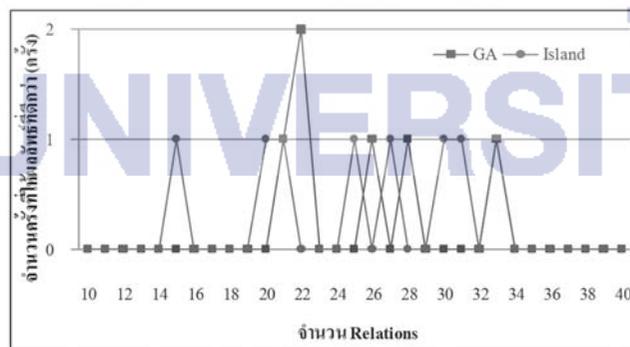
โดยผลการทดลองเพื่อวัดประสิทธิภาพด้านคุณภาพของผลลัพธ์ สามารถแสดงได้ดังภาพที่ 4 ซึ่งเป็นกราฟแสดงความถี่ของจำนวนครั้งที่ให้ผลลัพธ์ที่ดีกว่า สำหรับการค้นหาค่าตอบในช่วงจำนวนการ รวมข้อมูลของรีเลชันระหว่าง 10 - 40 รีเลชัน ซึ่งปรากฏว่ากระบวนการค้นหาค่าถามที่เหมาะสมที่สุดแบบกระจายที่ใช้ Island-based PGA นั้นให้ผลลัพธ์ที่มีคุณภาพที่ดีกว่า



ภาพที่ 2 ผลการเปรียบเทียบประสิทธิภาพด้านเวลาระหว่าง Island based PGA และ GA

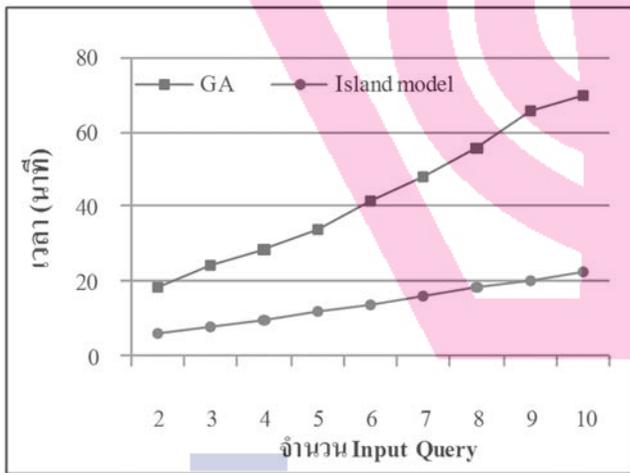


ภาพที่ 3 ผลการเปรียบเทียบประสิทธิภาพด้านเวลาในช่วง 10 ถึง 25 รีเลชัน



ภาพที่ 4 ผลการเปรียบเทียบคุณภาพของผลลัพธ์ระหว่าง GA และ Island based PGA

สำหรับการทดลองสุดท้ายที่ผู้วิจัยได้กำหนดขึ้นเพื่อเป็นตัวบ่งชี้ถึงความคุ้มค่าต่อการนำเอาการคำนวณแบบขนานเข้ามาช่วยในการแก้ปัญหา คือ การวัดประสิทธิภาพด้านการรองรับปริมาณงานจำนวนมากต่อการทำงาน โดยผู้วิจัยได้เลือกที่จะทำการทดสอบในกรณีที่เป็นกรณีที่เลวร้ายที่สุด (Worst Case) คือ ในกรณีที่มีการคำนวณค้นหาคำถามที่เหมาะสมที่สุดแบบกระจาย ต้องทำการคำนวณเพื่อหาคำตอบของคำสั่งสอบถามข้อมูลหลายๆ คำสั่งในเวลาที่ยาวนาน และแต่คำสั่งสอบถามข้อมูลจะเป็นการรวมข้อมูล 40 รีเลชัน ทั้งหมด ซึ่งจะต้องใช้การคำนวณและทรัพยากรอย่างมาก โดยผู้วิจัยทำการทดลองส่งคำสั่งสอบถามข้อมูลเข้าไปพร้อมๆ กันตั้งแต่ 2-10 คำสั่ง แล้วทำการหาค่าเวลาโดยเฉลี่ยในการทำงานของกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายซึ่งสามารถแสดงผลของการทดลองได้ดังนี้



ภาพที่ 5 เปรียบเทียบประสิทธิภาพด้านการรองรับปริมาณงานจำนวนมากระหว่าง GA และ Island-based PGA

จากภาพที่ 5 แสดงให้เห็นว่าเมื่อกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายต้องการประมวลผลเพื่อหาคำตอบสำหรับคำสั่งสอบถามข้อมูลหลายๆ คำสั่งพร้อมๆ กันเป็นจำนวนเพิ่มขึ้นเรื่อยๆ ประสิทธิภาพของกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายที่ใช้ GA ซึ่งเป็นการประมวลผลแบบเรียงลำดับ มีแนวโน้มว่าจะมีประสิทธิภาพที่ตกลงเรื่อยๆ อย่างชัดเจนมากกว่าแบบที่ใช้ Island-based PGA ซึ่งใช้การคำนวณแบบขนาน ดังนั้นผลการทดลองในภาพที่ 5 แสดงให้เห็นว่ากระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายที่ใช้

Island-based PGA นั้นสามารถรองรับการทำงานที่ต้องใช้การคำนวณที่สูงๆ ได้ดีกว่า

## สรุปผลการทดลอง

การพัฒนากระบวนการค้นหาคำถามที่เหมาะสมที่สุดสำหรับระบบฐานข้อมูลแบบกระจายนั้นสิ่งที่จะต้องคำนึงถึงอย่างยิ่ง คือ ประสิทธิภาพในการประมวลผลสอบถามแบบ Real-time ซึ่งจะต้องสามารถตอบสนองการทำงานของผู้ใช้งานได้อย่างรวดเร็วที่สุด และเวลาการทำงานเพื่อหาผลลัพธ์ของคำสั่งสอบถามข้อมูลจะต้องอยู่ในขอบเขตที่ผู้ใช้งานสามารถที่จะรอคอยผลลัพธ์ของการทำงานได้

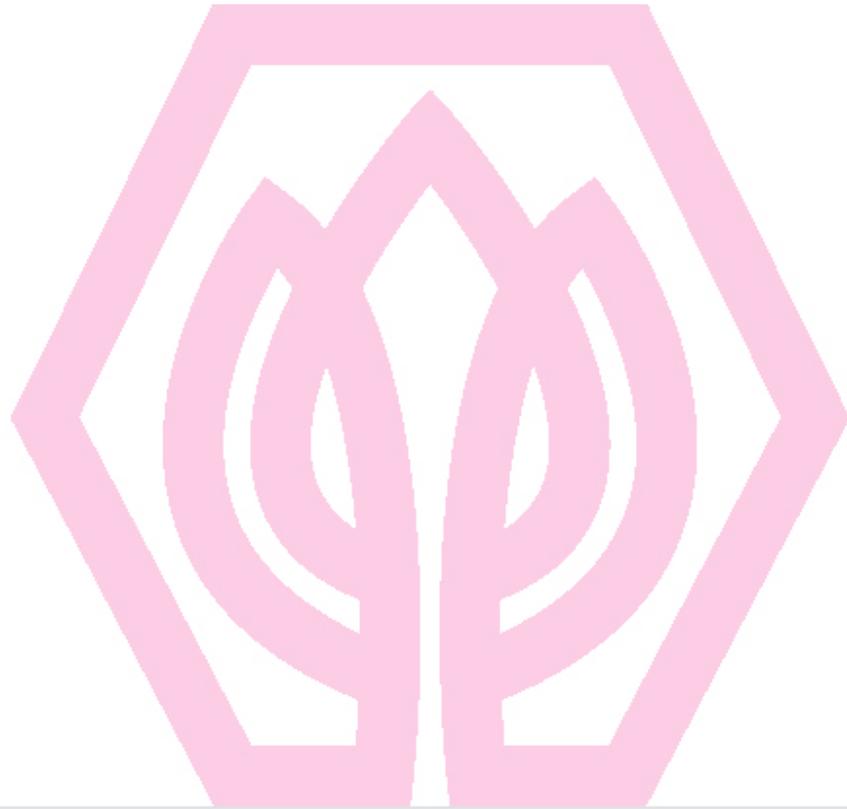
กระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจาย เป็นกระบวนการที่ใช้สำหรับค้นหาวิธีการประมวลผลสอบถามข้อมูลที่มีความเหมาะสมที่สุด ซึ่งจะถูกนำไปใช้สำหรับประมวลผลสอบถามเพื่อให้ได้คำตอบตามที่ผู้ใช้งานต้องการ โดยกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายนี้จะต้องมีการทำงานแบบ "คำนวณแบบครั้งต่อครั้ง" สำหรับแต่ละคำสั่งสอบถามข้อมูลที่เข้ามา โดยจะไม่สามารถที่จะเก็บคำตอบของกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายจากคำสั่งสอบถามข้อมูลก่อนหน้ามาใช้เป็นคำตอบสำหรับ คำสั่งสอบถามข้อมูลต่อไปได้ เนื่องจากกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายนั้นมีปัจจัยแวดล้อมหลายอย่าง ซึ่งสามารถแปรผันได้ตลอดเวลา เช่น อัตราการขนส่งข้อมูลผ่านระบบเครือข่าย, จำนวนแถวข้อมูลของแต่ละรีเลชัน ที่อาจมีการเปลี่ยนแปลงได้ตลอดจากการทำงานของผู้ใช้งานอื่นๆ

ดังนั้นสำหรับงานวิจัยนี้ ผู้วิจัยมีวัตถุประสงค์ที่จะทำการพัฒนากระบวนการ ค้นหาคำถามที่เหมาะสมที่สุดแบบกระจาย ที่สามารถทำงานได้อย่างรวดเร็ว และตอบสนองต่อการทำงานของผู้ใช้งานได้อย่างดีที่สุด โดยการนำวิธีการค้นหาที่ใช้การคำนวณแบบขนาน คือ Island-based PGA มาเป็นวิธีที่ใช้สำหรับค้นหาวิธีการในการประมวลผลข้อมูลที่มีความเหมาะสมที่สุดในกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจาย โดยการวัดประสิทธิภาพที่ได้จากการพัฒนานั้นผู้วิจัยได้ทำการทดลองเพื่อบ่งชี้ประสิทธิภาพใน 2 ด้านคือความเร็วในการทำงาน และคุณภาพของผลลัพธ์ที่ได้จากการทำงาน

จากการทดลองผู้วิจัยสามารถสรุปได้ว่ากระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายที่ใช้วิธีการค้นหาแบบ Island-based PGA นั้นสามารถทำงานรวดเร็วกว่าแบบที่ใช้ GA ธรรมดา และในเรื่องของคุณภาพของผลลัพธ์นั้นกระบวนการค้นหาคำถามที่เหมาะสมที่สุดแบบกระจายที่ใช้วิธีการค้นหาแบบ Island-based PGA ให้คุณภาพของผลลัพธ์ที่ดีกว่าแบบที่ใช้ GA และในด้านการรองรับปริมาณงานจำนวนมากสามารถสรุปได้ว่า Island-base PGA สามารถรองรับคำสั่งสอบถามข้อมูลพร้อมๆ กันจำนวนมากได้ดีกว่าแบบที่ใช้ GA

## เอกสารอ้างอิง

- Bennett K., Michael C. Ferris, and Yannis E. Ioannidis. 1991. "A Genetic Algorithm for Database Query Optimization." **Proceedings of the Fourth International Conference on Genetic Algorithms.** 400-407.
- Dong H., and Liang Y. 2007. "Genetic Algorithms for Large Join Query Optimization." **GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation.** 1211-1218.
- Ioannidis Y.E. 1996. "Query Optimization." **ACM Computing Surveys.** 103-114.
- Ioannidis Y.E., and Kang Y.C. 1990. "Randomized Algorithms for Optimization Large Join Queries." **Proc. of the 1990 ACM SIGMOD Conf.** 312-321.
- Java Architecture for XML Binding (JAXB), Retrieved December, 1 2008 from [http://java.sun.com/ developer/technicalArticles/WebServices/jaxb/](http://java.sun.com/developer/technicalArticles/WebServices/jaxb/)
- JSqlParser, Retrieved December, 1 2008 from <http://jsparser.sourceforge.net/>
- Oracle Database Performance Tuning Guide. (n.d.). Retrieved September, 1 2009 from [http://downloadeast.oracle.com/docs/cd/B14117\\_01/server.101/b10752/optimops.htm](http://downloadeast.oracle.com/docs/cd/B14117_01/server.101/b10752/optimops.htm)
- Ozsu M.T., and Valduriez P. 1991. **Principles of Distributed Database Systems.** New Jersey: Prentice-Hall.
- Poli, R., William B. Langdon, and Nicholas F. McPhee. 2008. **A Field Guide to Genetic Programming.** United States.
- Sangkyu Rho. 1997. "Optimizing distributed join queries: A genetic algorithm approach." **J.C. Baltzer AG, Science Publishers.** 199-228.
- Stillger M., Myra Spiliopoulou and Freytag JC. 1996. "Parallel Query Optimization: Exploiting Bushy and Pipeline Parallelism with Genetic Programms." Retrieved January 15, 2009 from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.2987>
- Stillger M., and Spiliopoulou M. 1996. "Genetic Programming in Database Query Optimization." In **Proc First Annu. Conf. Genetic Programming, Stanford, CA.** 388-393.
- Sunee Pongpinigpinyo. 1996. **Distributed Query Optimization Using Two Stages Simulated Annealing.** Master Thesis, Computer Science. University of Tasmania.
- Swam A. 1990. "Optimization of large Join Queries: Combining Heuristic and Combinatorial techniques." **Proc. of 1989 ACM SIGMOD Conf.** 367-376.
- Swami A., and Gupta A. 1988. "Optimization of large Join Queries." **Proc. ACM-SIGMOD Conf.** 8-7.



### >> พิศาล สุขขี

จบการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร และกำลังศึกษาหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

ปัจจุบันทำงานในตำแหน่ง ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร พระราชวังสนามจันทร์ จังหวัดนครปฐม



### >> สุนีย์ พงษ์พินิจบุญ

จบการศึกษาหลักสูตรวิทยาศาสตรดุษฎีบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี และหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาสถิติ มหาวิทยาลัยศิลปากร

ปัจจุบันทำงานในตำแหน่ง เลขานุการภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร พระราชวังสนามจันทร์ จังหวัดนครปฐม ผลงานทางวิชาการ เช่น การคัดเลือกคุณลักษณะของภาพพอยน์น้ำจืดวงศ์ Thiaridae โดยใช้โครงข่ายประสาทเทียม และระบบผู้เชี่ยวชาญสำหรับจำแนกประเภทพอยน์น้ำจืดวงศ์ Thiaridae