ภาคผนวก (ก)
การใช้งานคลาสที่ใช้ในการคำนวณโดยหลักการเจเนติกอัลกอริทึม

1. คลาส BinaryGenome

1.1 วิธีการ (Method)

- BinaryGenome()

คอนสตรัคเตอร์ของคลาส

- BinaryGenome(int PopSize)

คอนสตรัคเตอร์ของคลาสเมื่อต้องการตั้งค่าจำนวนประชากรใหม่

- void InitPopulation()

ฟังก์ชันที่ใช้ในการสร้างประชากรเริ่มต้น

- void InitPopulation(int PopSize)

ฟังก์ชันที่ใช้ในการสร้างประชากรเริ่มต้นและตั้งค่าจำนวนประชากรใหม่ผ่านตัวแปร PopSize

- void AddChromosome(int Length)

ฟังก์ชันที่ใช้ในการเพิ่มจำนวนโครโมโซมและกำหนดความยาวของโครโมโซมผ่านตัวแปร Length

- virtual double FitnessFunction(int PopIndex)

ฟังก์ชันเวอร์ชวล (Virtual Function) ที่ใช้ในการคำนวณค่าความเหมาะสมซึ่งจะแปรเปลี่ยนไปตามลักษณะของปัญหา โดยที่ตัวแปร PopIndex คือ ตำแหน่งของโครโมโซมในประชากรที่ต้องการคำนวณค่าความเหมาะสม

- void CalculateFitnessValue()

ฟังก์ชันที่ใช้ในการคำนวณค่าความเหมาะสมของประชากร

- double ConverseBitToDec(double Lowerlimit,double Upperlimit,int PopIndex, int DNAIndex)

ฟังก์ชันที่คืนค่าจำนวนจริงจากการถอดรหัสโครโมโซมเลขฐานสองเป็นจำนวนจริงที่ถูกเข้ารหัสไว้ โดยที่ตัวแปร LowerLimit, Upperlimit, PopIndex และ DNAIndex คือ ค่าต่ำสุด, ค่าสูงสุดของจำนวนจริงที่จะถอดรหัส, ตำแหน่งของโครโมโซมในประชากร และค่าตำแหน่งของโครโมโซมบน DNA ตามลำดับ

- double FitnessValue(int PopIndex)

ฟังก์ชันที่คืนค่าความเหมาะสมโครโมโซมแต่ละตัว โดยที่ตัวแปร PopIndex คือ ตำแหน่งของโครโมโซมในประชากรที่ต้องการคำนวณค่าความเหมาะสม

- void RecordPopToTempPop(int PopIndex,int TempPopIndex)

ฟังก์ชันที่ใช้บันทึกโครโมโซมจากกลุ่มประชากรจริงลงในกลุ่มประชากรชั่วคราว PopIndex และ TempPopIndex คือ ตำแหน่งของโครโมโซมในประชากรจริง และ ตำแหน่งของโครโมโซมในประชากรชั่งคราว ตามลำดับ

- void RecordTempPopToPop(int PopIndex,int TempPopIndex)

ฟังก์ชันที่ใช้บันทึกโครโมโซมจากกลุ่มประชากรชั่วคราวลงในกลุ่มประชากรจริง

1.2 โครงสร้าง (Structure)

- struct Chromosome

โครงสร้างข้อมูลที่เก็บคุณสมบัติของโครโมโซมแต่ละตัว

- struct Population

โครงสร้างข้อมูลที่เก็บคุณสมบัติของประชากรทั้งหมด

1.3 ตัวแปร

- Population *Pop

ตัวแปรพอยเตอร์ของโครงสร้าง Population ที่ใช้ในการจองหน่วยความจำสำหรับประชากรจริง

- Poppulation *TempPop

ตัวแปรพอยเตอร์ของโครงสร้าง Population ที่ใช้ในการจองหน่วยความจำสำหรับประชากรชั่วคราว

- int PopulationSize

จำนวนโครโมโซมของประชากร

- int CountDNANo

จำนวนโครโมโซมบน DNA

- double SumFitValue

ผลรวมค่าความเหมาะสมของประชากรทั้งหมด

2. คลาส GeneticAlgorithm

2.1 วิธีการ (Method)

- GA()

คอนสตรัคเตอร์ของคลาส

- void Elitize(BinaryGenome& g)

ฟังก์ชันที่ใช้ในการทำตัวดำเนินการกลยุทธ์การคัดสรร

- void Selection(BinaryGenome& g,int SelectType)

ฟังก์ชันที่ใช้ในการทำการคัดเลือก โดยตัวแปร SelectType คือ ค่าจำนวนเต็มที่แทนชนิดในการคัดเลือก ได้แก่ RW_SELECT และ US_SELECT ซึ่งใช้แทนการคัดเลือกชนิดวงล้อรูเลต และการคัดเลือกชนิด Stochastic Universal Sampling ตามลำดับ

- void CrossOver(BinaryGenome& g,int CrossType,int nPoint=1,int Width=0,int Length=0)

ฟังก์ชันที่ใช้ในการทำการผสมข้ามพันธุ์ โดยตัวแปร CrossType คือ ค่าจำนวนเต็มที่แทนชนิดในการผสมข้ามพันธุ์ ได้แก่ NP_CROSS, UF_CROSS และ TOD_CROSS ซึ่งใช้แทนการผสมข้ามพันธุ์ชนิดที่กำหนดจำนวนจุดให้เกิดการผสมข้ามพันธุ์, การผสมข้ามพันธุ์ชนิด Uniform และการผสมข้ามพันธุ์ชนิดที่กำหนดจำนวนจุดให้เกิดการผสมข้ามพันธุ์แบบสองมิติ ตามลำดับ และตัวแปร nPoint คือ จำนวนจุดในการผสมข้ามพันธุ์ ตัวแปร Width และ Length คือ ตัวแปรที่ใช้แทนความกว้างและยาวของโครโมโซมในสองมิติ ตามลำดับ

- void RouletteWheelSelection(BinaryGenome& g)

ฟังก์ชันที่ใช้ในการทำการคัดเลือกชนิดวงล้อรูเลต

- void UniversalSamplingSelection(BinaryGenome& g)

ฟังก์ชันที่ใช้ในการทำการคัดเลือกชนิด Stochastic Universal Sampling

- void nPointCrossover(BinaryGenome& g,int nPoint=1)

ฟังก์ชันที่ใช้ในการทำการผสมข้ามพันธุ์ชนิดที่กำหนดจำนวนจุดให้เกิดการผสมข้ามพันธุ์

- void UniformCrossover(BinaryGenome& g)

ฟังก์ชันที่ใช้ในการทำการผสมข้ามพันธุ์ชนิด Uniform

- void CrossOver2D(BinaryGenome& g,int nPoint,int Width,int Length)

ฟังก์ชันที่ใช้ในการทำการผสมข้ามพันธุ์ชนิดที่กำหนดจำนวนจุดให้เกิดการผสมข้ามพันธุ์แบบสองมิติ

- void Mutation(BinaryGenome& g)

ฟังก์ชันที่ใช้ในการทำตัวดำเนินการการกลายพันธุ์

- void Mating(BinaryGenome& g)

ฟังก์ชันที่ใช้ในการจับคู่โครโมโซมเพื่อการผสมข้ามพันธุ์

2.2 ตัวแปร

- int EliteSize

ขนาดของการคัดสรร

- int GenerationSize

จำนวนรุ่นในการคำนวน

- double CrossOverProb

ค่าความน่าจะเป็นในการผสมข้ามพันธุ์

- double MutationProb

ค่าความน่าจะเป็นในการกลายพันธุ์

3. วิธีการใช้

การใช้งานคลาส BinaryGenome และ GA ในการคำนวณตามหลักการเจเนติกอัลกอริทึมจะเริ่มต้นด้วยการประกาศตัวแปรคลาส ตัวอย่างเช่น ga และ genomeให้เป็นของทั้งสองคลาส GA และ BinaryGenome ตามลำดับ คือ

```
GA ga;
BinaryGenome genome;
```

จากนั้นจึงทำการสร้างประชากรเริ่มต้นโดยการผ่านค่าจำนวนประชากรเริ่มต้นให้กับฟังก์ชัน InitPopulation(PopSize) ของคลาส BinaryGenome

```
genome.InitPopulation(100);
```

เมื่อสร้างประชากรเริ่มต้นแล้วจะต้องกำหนดจำนวนโครโมโซมที่ใช้ในการคำนวณโดยเรียกใช้ฟังก์ชัน genome.AddChromosome(Length) ของคลาส BinaryGenome ตาม

จำนวนโครโมโซมที่จะใช้ ตัวอย่างเช่นหากต้องการใช้โครโมโซมในการคำนวณ 2 โครโมโซม และแต่ละโครโมโซมมีความยาว 10 และ 9 ตามลำดับ จะได้ว่า

```
genome.AddChromosome(10);
genome.AddChromosome(9);
```

จากนั้นจึงทำการคำนวณค่าความเหมาะสมของประชากรด้วยฟังก์ชัน CalculateFitnessValue() ของคลาส BinaryGenome แล้วกระทำตัวดำเนินการทางพันธุกรรมกับประชากร โดยการผ่านค่าตัวแปรคลาสของคลาส BinaryGenome ให้กับฟังก์ชันของต่างๆ ของคลาส GA จนได้จำนวนรอบเท่ากับจำนวนรอบที่ต้องการคำนวณ (Generations) ดังนี้

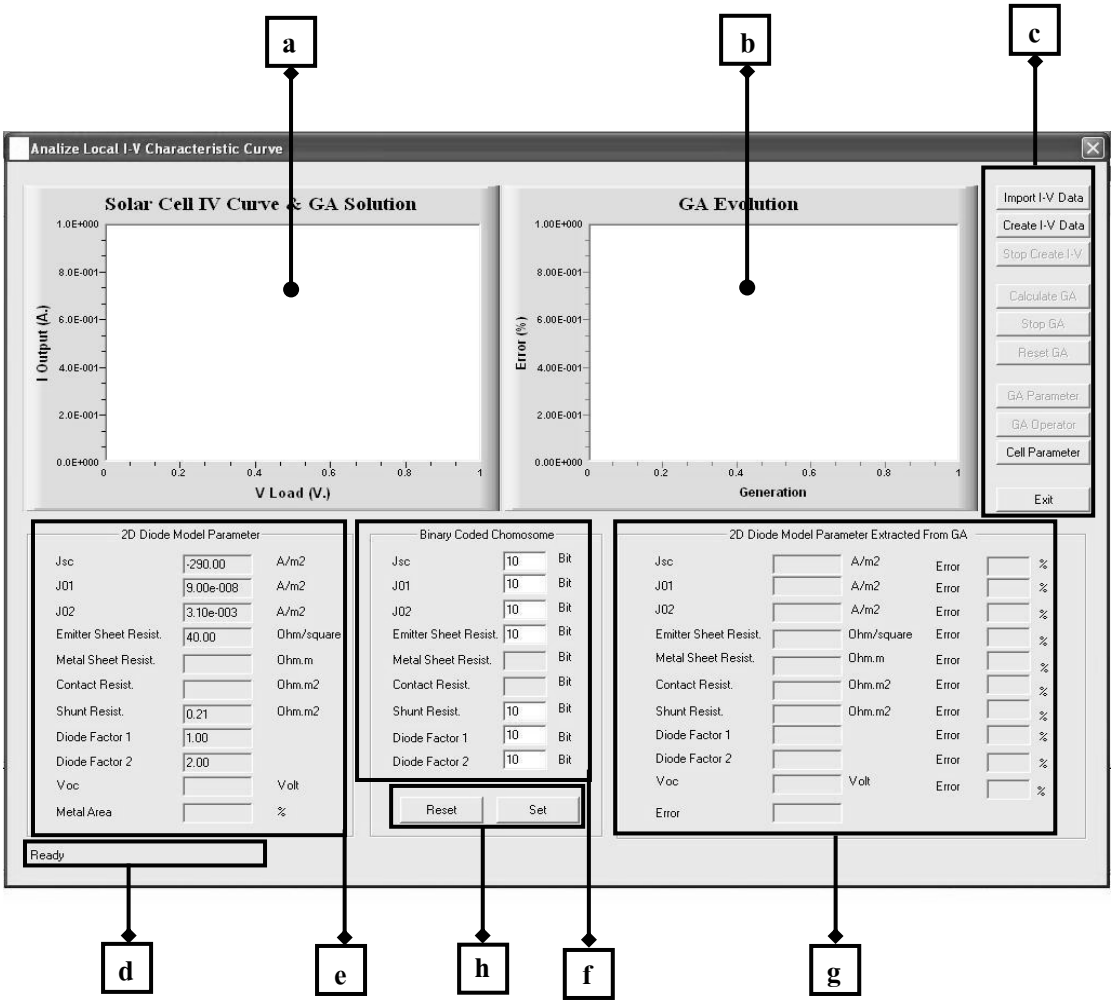
```
For(I = 0;I < Generation;I++)
{
        genome.CalculateFitnessValue();
        ga.Elitize(genome);
        ga.Selection(genome,SelectType);
        ga.CrossOver(genome,CrossType,2);
        ga.Mutation(genome);
}
```

ภาคผนวก (ข)

วิธีการใช้งานโปรแกรมฟิตกราฟเพื่อหาค่าพารามิเตอร์ของเซลล์แสงอาทิตย์และโปรแกรมออกแบบรูปแบบขั้วของเซลล์แสงอาทิตย์โดยใช้เจเนติกอัลกอริทึม

1. การใช้งานโปรแกรมฟิตกราฟเพื่อหาค่าพารามิเตอร์ของเซลล์แสงอาทิตย์โดยใช้เจเนติกอัลกอริทึม

โปรแกรมฟิตกราฟเพื่อหาค่าพารามิเตอร์ของเซลล์แสงอาทิตย์โดยใช้เจเนติกอัลกอริทึมมีลักษณะเป็นไดอะล็อกซึ่งมีส่วนของกราฟและกล่องข้อความในการแสดงผลการคำนวณ ซึ่งการใช้งานโปรแกรมนี้สามารถทำได้ด้วยการรันไฟล์ FitIVCurve.exe โดยส่วนประกอบหลักของไดอะล็อกของโปรแกรมเป็นดังนี้



a. กราฟแสดงกราฟลักษณะเฉพาะกระแสและศักย์ของเซลล์แสงอาทิตย์ที่ต้องการหาค่าพารามิเตอร์และกราฟที่ได้จากค่าพารามิเตอร์ที่ค้นหาด้วยเจเนติกอัลกอริทึม

b. กราฟแสดงค่า RMSE จากผลเฉลยที่ดีที่สุดในแต่ละรุ่นของการคำนวณด้วยเจเนติกอัลกอริทึม

c. ปุ่มที่ใช้ในการสั่งงานโปรแกรม

d. แถบแสดงสถานะของโปรแกรม

e. กล่องข้อความแสดงค่าพารามิเตอร์ของเซลล์ที่กำหนดขึ้นเพื่อจำลองกราฟลักษณะเฉพาะของเซลล์เพื่อใช้ในการทดลองค้นหาด้วยเจเนติกอัลกอริทึม

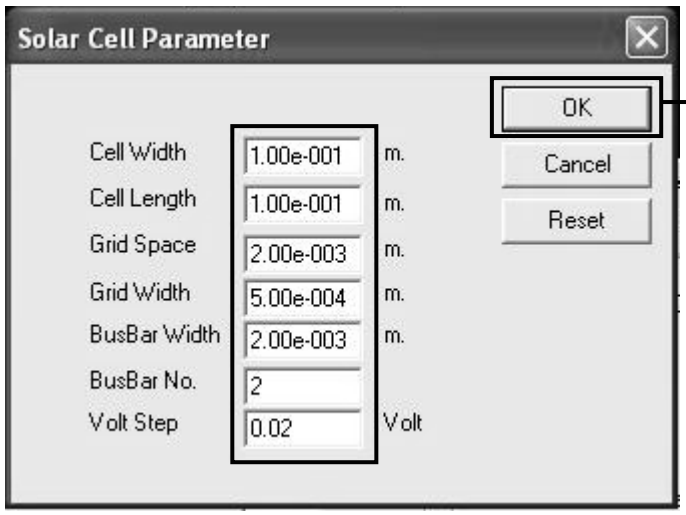
f. จำนวนบิตของเลขฐานสองที่ใช้ในการเข้ารหัสค่าพารามิเตอร์แต่ละตัว

g. กล่องข้อความแสดงค่าพารามิเตอร์ของเซลล์ที่ได้จากการค้นหาด้วยเจเนติกอัลกอริทึมและค่าความผิดพลาดของค่าพารามิเตอร์ที่ได้กับค่าพารามิเตอร์ที่กำหนดขึ้น

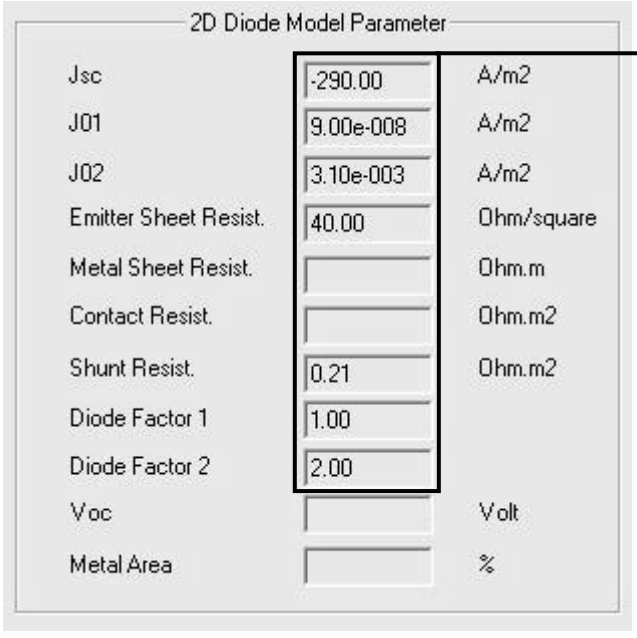
h. ปุ่มกดที่ใช้ในการยืนยันหรือยกเลิกการกำหนดค่าในกล่องข้อความ

1.1 การเลือกหรือสร้างกราฟลักษณะเฉพาะกระแสและศักย์เพื่อใช้ในการค้นหาค่าพารามิเตอร์ด้วยเจเนติกอัลกอริทึม

ในการนำกราฟกระแสและศักย์ของเซลล์แสงอาทิตย์มาเพื่อนำมาหาค่าพารามิเตอร์ด้วยเจเนติกอัลกอริทึมนั้น สามารถกระทำได้สองกรณี คือ การสร้างกราฟกระแสและศักย์จากค่าพารามิเตอร์และรูปแบบของขั้วของเซลล์แสงอาทิตย์ที่กำหนดขึ้นเอง และการอ่านค่าผ่านไฟล์ (Text File) ซึ่งกระทำได้ดังนี้

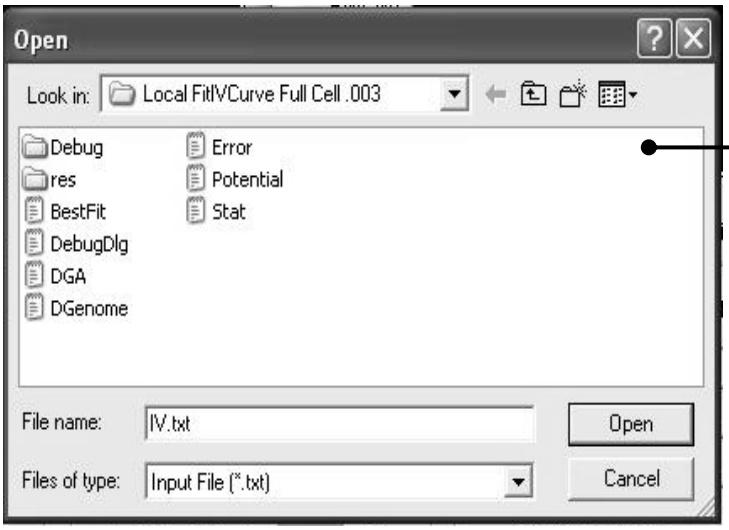


1. เริ่มต้นด้วยการกำหนดค่าขนาดของเซลล์ในกรณีที่ต้องการสร้างกราฟกระแสและศักย์จากค่าพารามิเตอร์ที่กำหนดขึ้นเอง โดยคลิกที่ปุ่ม Cell Parameter ซึ่งจะปรากฏไดอะล็อก Solar Cell Parameter ขึ้นมาให้ทำการกำหนดค่าต่างๆ แล้วกดปุ่ม OK เพื่อยืนยันการกำหนดค่า

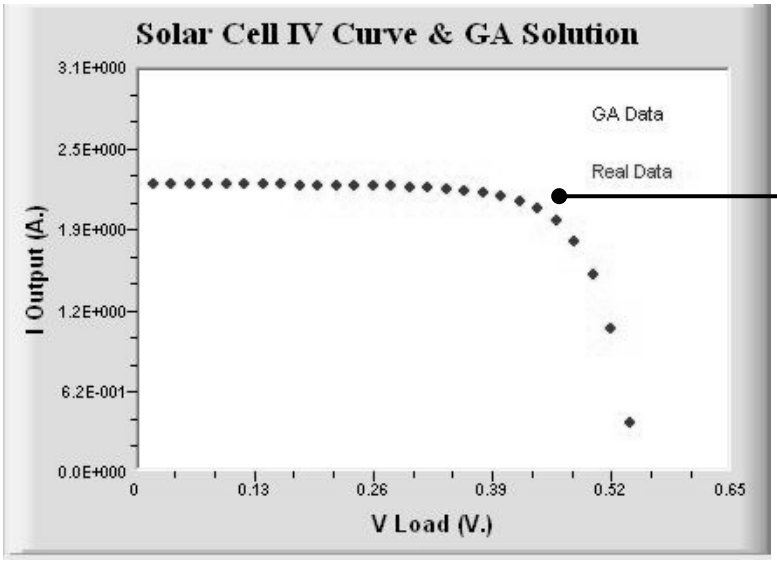


2. ในกรณีที่ต้องการสร้างกราฟกระแสและศักย์จากค่าพารามิเตอร์ที่กำหนดขึ้นเอง สามารถกำหนดค่าพารามิเตอร์ของเซลล์ได้ผ่านกล่องข้อความ และยืนยันการกำหนดค่าโดยการกดปุ่ม Set ในส่วนล่างของไดอะล็อก

3. กดปุ่ม Import I-V Data หรือ Create I-V Data เพื่อเลือกกราฟลักษณะเฉพาะของเซลล์แสงอาทิตย์จากไฟล์หรือสร้างขึ้นเองจากค่าพารามิเตอร์ที่กำหนดไว้ข้างต้น



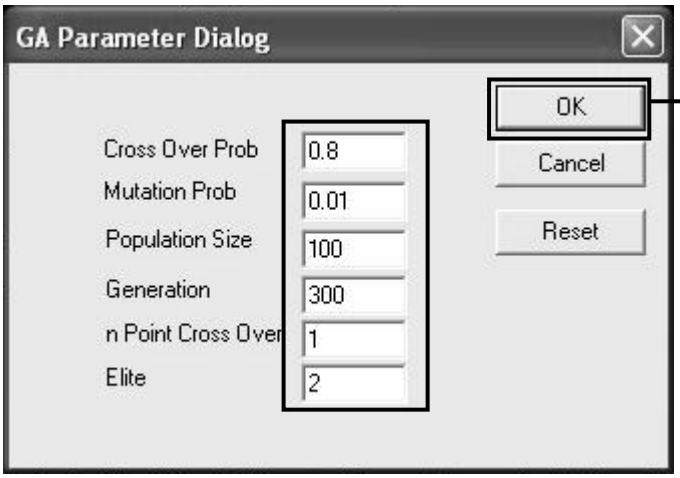
4. เมื่อ กดปุ่ม Import I-V Data หรือ Create I-V Data แล้วจะปรากฏไดอะล็อกให้เลือกไฟล์ที่ต้องการจะอ่านหรือเก็บค่ากระแสและศักย์ของเซลล์ ตามลำดับ ให้เลือกไฟล์ที่ต้องการแล้วกดปุ่ม Open เพื่อยืนยัน



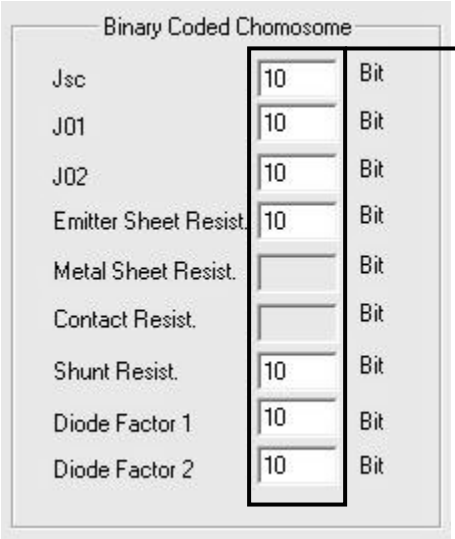
5. จากนั้นจะปรากฏกราฟที่ต้องการหาค่าพารามิเตอร์ขึ้น

1.2 การกำหนดค่าและตัวดำเนินการทางพันธุกรรมที่ใช้ในการคำนวณด้วยเจเนติกอัลกอริทึม

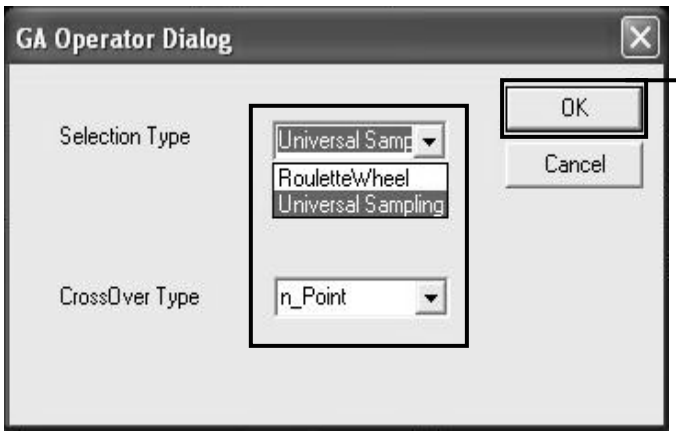
ก่อนที่จะเริ่มคำนวณด้วยเจเนติกอัลกอริทึมต้องทำการกำหนดค่าและตัวดำเนินการทางพันธุกรรมที่ต้องการเลือกใช้ในการคำนวณก่อน ซึ่งสามารถกระทำได้ดังนี้



1. กดปุ่ม GA Parameter เพื่อทำการกำหนดค่าพารามิเตอร์ที่ใช้ในการคำนวณด้วยเจเนติกอัลกอริทึม ซึ่งจะปรากฏไดอะล็อกเพื่อให้กำหนดค่าพารามิเตอร์ต่างๆ หลังจากการกำหนดค่าแล้วให้กดปุ่ม OK เพื่อยืนยันการกำหนดค่า



2. กำหนดจำนวนบิตของเลขฐานสองที่ใช้ในการเข้ารหัสค่าพารามิเตอร์ของเซลล์ ซึ่งจำนวนบิตที่ใช้จะขึ้นอยู่กับความแม่นยำที่ต้องการเข้ารหัส เมื่อกำหนดค่าแล้วให้กดปุ่ม Set เพื่อยืนยันการกำหนดค่า

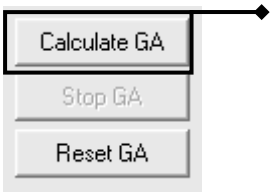


3. กดปุ่ม GA Operator ซึ่งจะปรากฏไดอะล็อกเพื่อเลือกให้ตัวดำเนินการทางพันธุกรรมที่ต้องการ แล้วกดปุ่ม OK เพื่อยืนยันตัวดำเนินการที่เลือกใช้

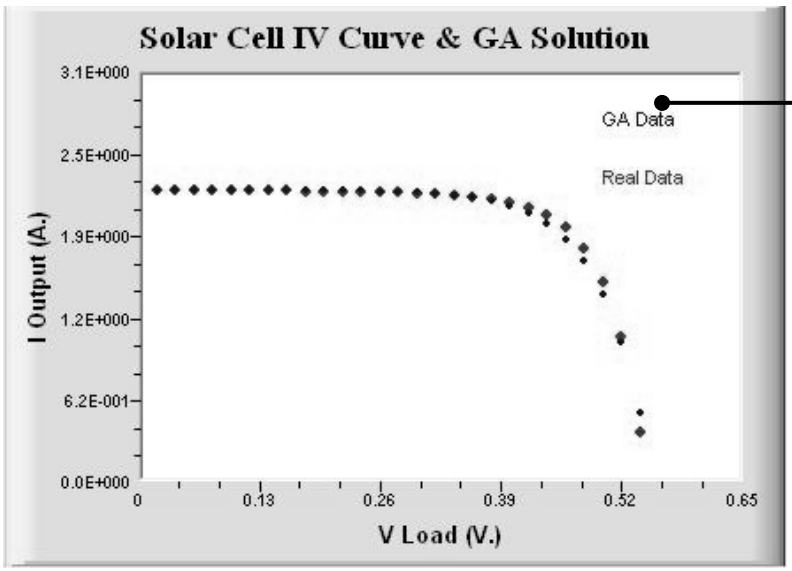
<u>คำเตือน</u> หากมีการเปลี่ยนแปลงค่าจำนวนประชากรหรือจำนวนบิตในการคำนวณด้วยเจเนติกอัลกอริทึม ต้องกดปุ่ม Reset GA ก่อนเริ่มต้นการคำนวณทุกครั้ง เพราะโปรแกรมต้องทำการจองหน่วยความจำใหม่

1.3 การคำนวณหาค่าพารามิเตอร์ของเซลล์แสงอาทิตย์ด้วยเจเนติกอัลกอริทึม

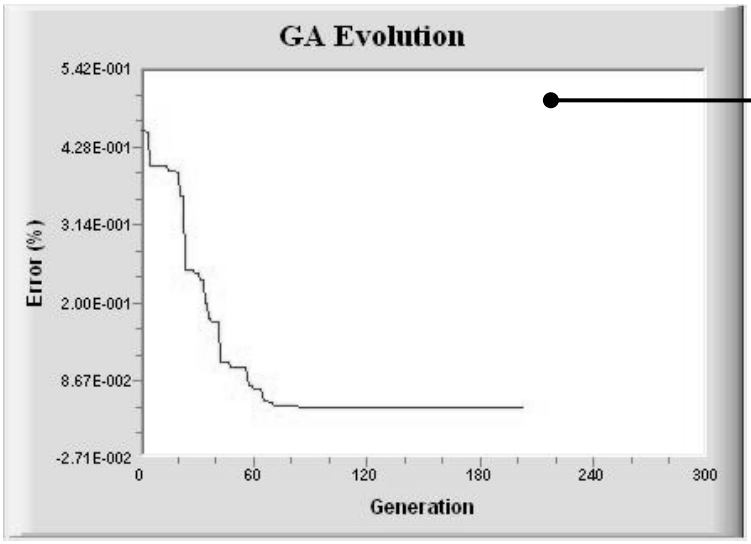
ในการคำนวณหาค่าพารามิเตอร์ของเซลล์แสงอาทิตย์ด้วยเจเนติกอัลกอริทึม สามารถกระทำได้ดังนี้



1. เริ่มต้นด้วยการกดปุ่ม Calculate GA เพื่อเริ่มการคำนวณ



2. เมื่อเริ่มการคำนวณแล้ว จะปรากฏกราฟกระแสและศักย์ที่ได้จากการหาค่าพารามิเตอร์ด้วยเจเนติกอัลกอริทึมขึ้นมาซ้อนกับกราฟที่ต้องการหาค่าพารามิเตอร์ที่มีอยู่ก่อนแล้ว

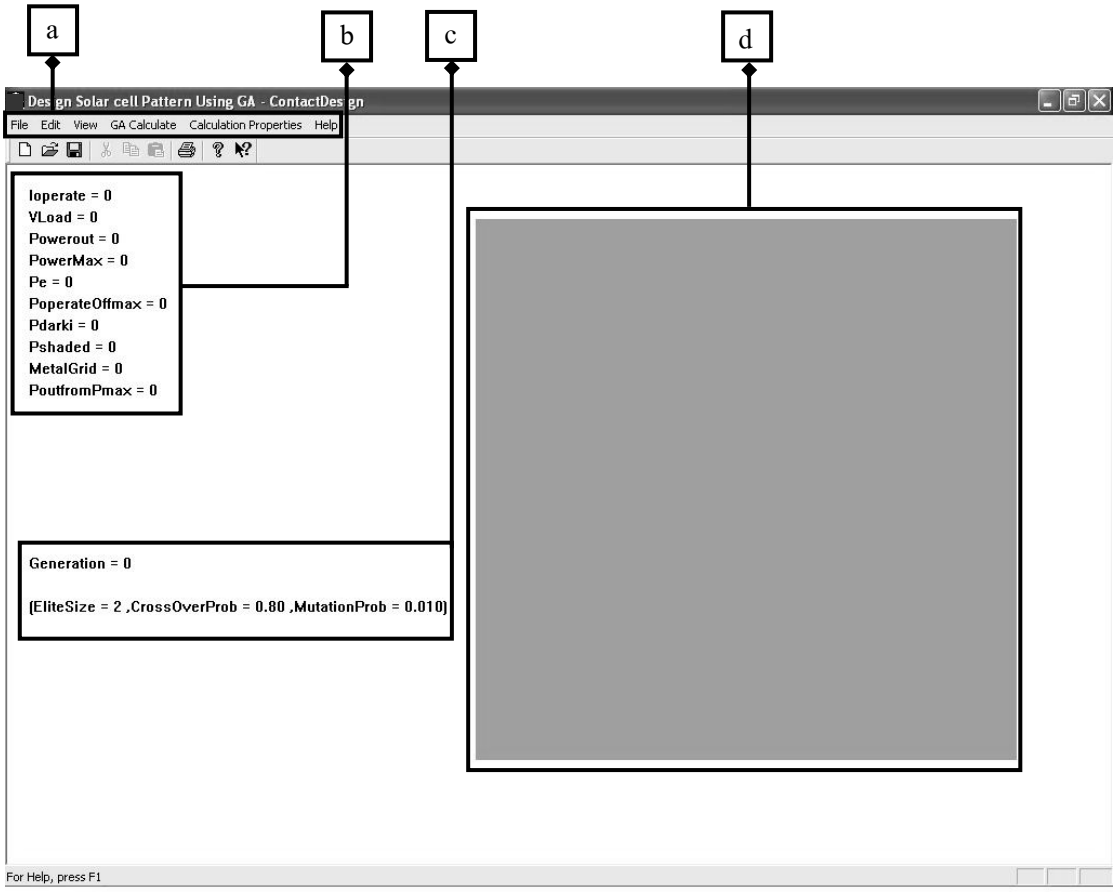


3. ในขณะที่ทำการคำนวณกราฟที่แสดงค่า RMSE ก็จะแสดงการพัฒนาการของค่า RMSE ที่ได้จากการคำนวณด้วยเจเนติกอัลกอริทึม ซึ่งมีผลต่อการตัดสินใจเพื่อหยุดการคำนวณหากไม่มีการพัฒนาของค่า RMSE

4. หากต้องการหยุดการคำนวณให้กดปุ่ม Stop GA แล้วกดปุ่ม Yes เพื่อยืนยัน และหากต้องการคำนวณต่อจากเดิมให้กดปุ่ม Calculate GA เพื่อคำนวณต่อได้เลย แต่หากต้องการคำนวณใหม่ให้กดปุ่ม Reset GA แล้วกดปุ่ม Calculate GA เพื่อเริ่มการคำนวณใหม่ โดยเมื่อมีการหยุดการคำนวณจะสามารถปรับเปลี่ยนค่าพารามิเตอร์หรือตัวดำเนินการของเจเนติกอัลกอริทึมได้

2. โปรแกรมการออกแบบรูปแบบขั้วของเซลล์แสงอาทิตย์โดยใช้เจเนติกอัลกอริทึม

โปรแกรมการออกแบบรูปแบบขั้วของเซลล์แสงอาทิตย์โดยใช้เจเนติกอัลกอริทึมมีลักษณะเป็นหน้าต่างที่มีส่วนแสดงผลการคำนวณทั้งค่าคุณสมบัติของรูปแบบขั้วและรูปแบบขั้วที่ดีที่สุดที่ได้จากการออกแบบ ซึ่งสามารถควบคุมการทำงานของโปรแกรมผ่านเมนู ซึ่งการใช้งานโปรแกรมนี้สามารถทำได้ด้วยการรันไฟล์ ContactDesign.exe โดยส่วนประกอบหลักของหน้าต่างของโปรแกรมเป็นดังนี้



a. เมนูควบคุมการทำงานของโปรแกรม

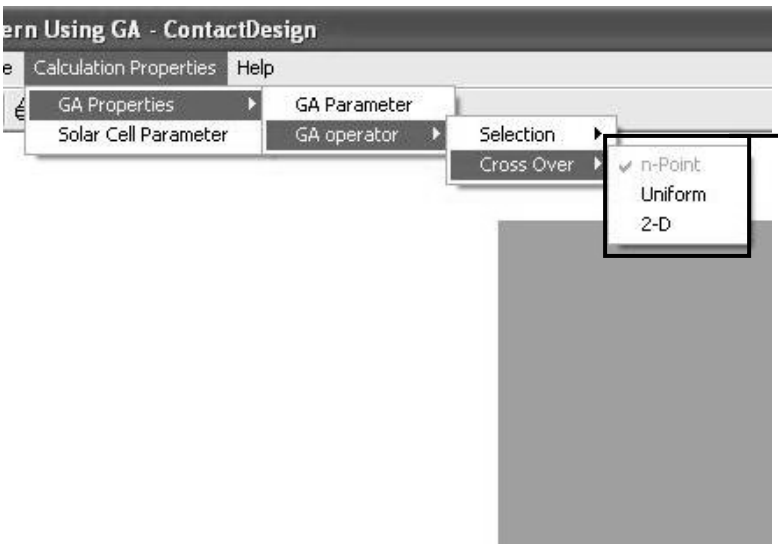
b. ส่วนที่แสดงค่าคุณสมบัติของรูปแบบขั้วที่ได้จากการออกแบบในแต่ละรุ่นของการคำนวณ

c. ส่วนที่แสดงรอบของการคำนวณและค่าพารามิเตอร์ของเจเนติกอัลกอริทึมที่ใช้ในการคำนวณ

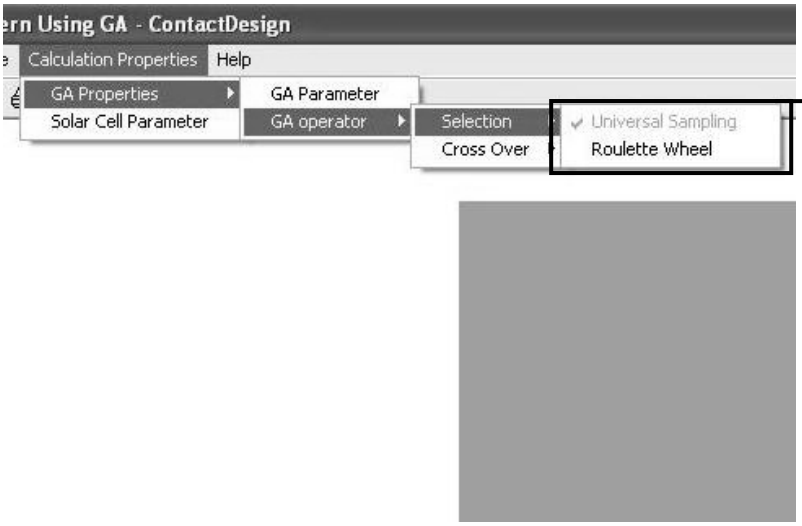
d. ส่วนที่แสดงรูปแบบขั้วที่ดีที่สุดที่ได้ในแต่ละรุ่นของการคำนวณ

2.1 การกำหนดค่าพารามิเตอร์และตัวดำเนินการทางพันธุกรรมที่ใช้ในการคำนวณ

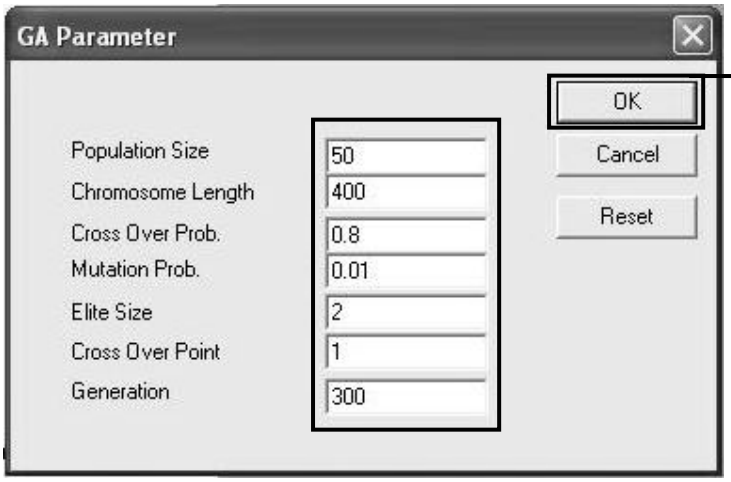
ในการกำหนดค่าพารามิเตอร์และตัวดำเนินการทางพันธุกรรมที่ใช้ในการคำนวณสามารถกระทำได้ผ่านเมนู Calculation Properties ซึ่งสามารถกระทำได้ดังนี้



1. กำหนดชนิดของตัวดำเนินการ Crossover สามารถกระทำได้โดยการเลือกเมนู Calculation Properties → GA Properties → GA operator → Cross Over แล้วเลือกตัวดำเนินการ Crossover ที่ต้องการ

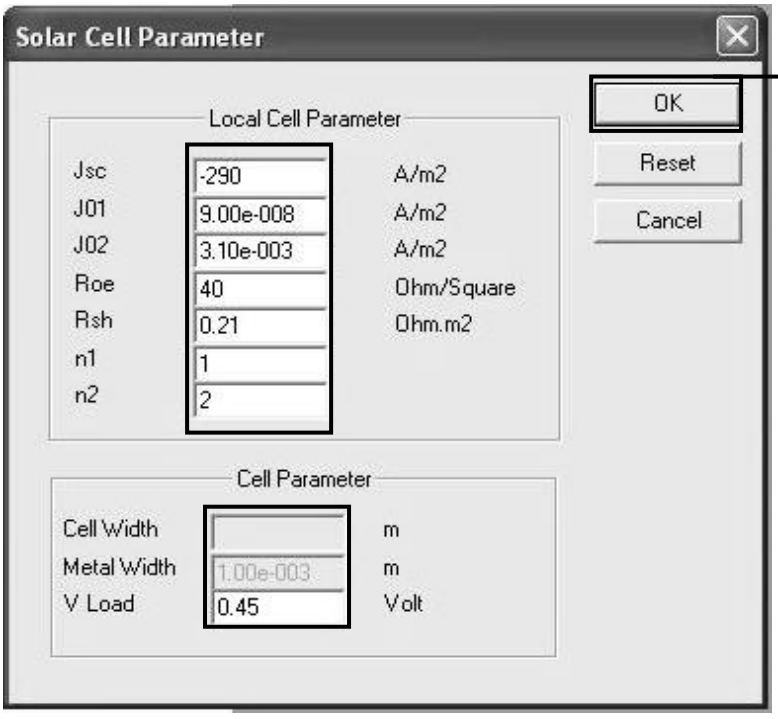


2. กำหนดชนิดของการ Selection สามารถกระทำได้โดยการเลือกเมนู Calculation Properties → GA Properties → GA operator → Selection แล้วเลือกการ Selection ที่ต้องการ



3. กำหนดค่าพารามิเตอร์ในการคำนวณด้วยเจเนติกอัลกอริทึมสามารถกระทำได้ด้วยการเลือกเมนู Calculation Properties → GA Parameter โดยจะปรากฏไดอะล็อก GA Parameter ขึ้นมาให้กำหนดค่าพารามิเตอร์ และสามารถยืนยันการกำหนดค่าด้วยการกดปุ่ม OK

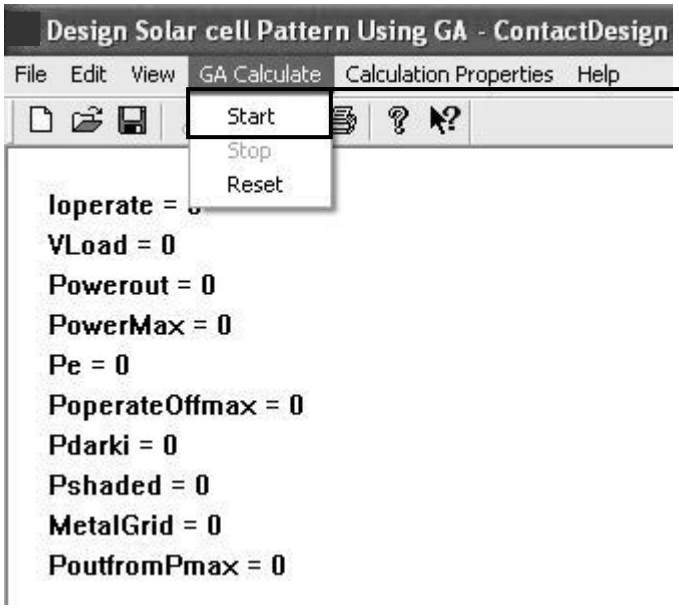
<u>คำเตือน</u> หากมีการเปลี่ยนแปลงค่าจำนวนประชากรหรือจำนวนบิตในการคำนวณด้วยเจเนติกอัลกอริทึม ต้องเลือกเมนู GA Calculate → Reset ก่อนเริ่มต้นการคำนวณทุกครั้ง เพราะโปรแกรมต้องทำการจองหน่วยความจำใหม่



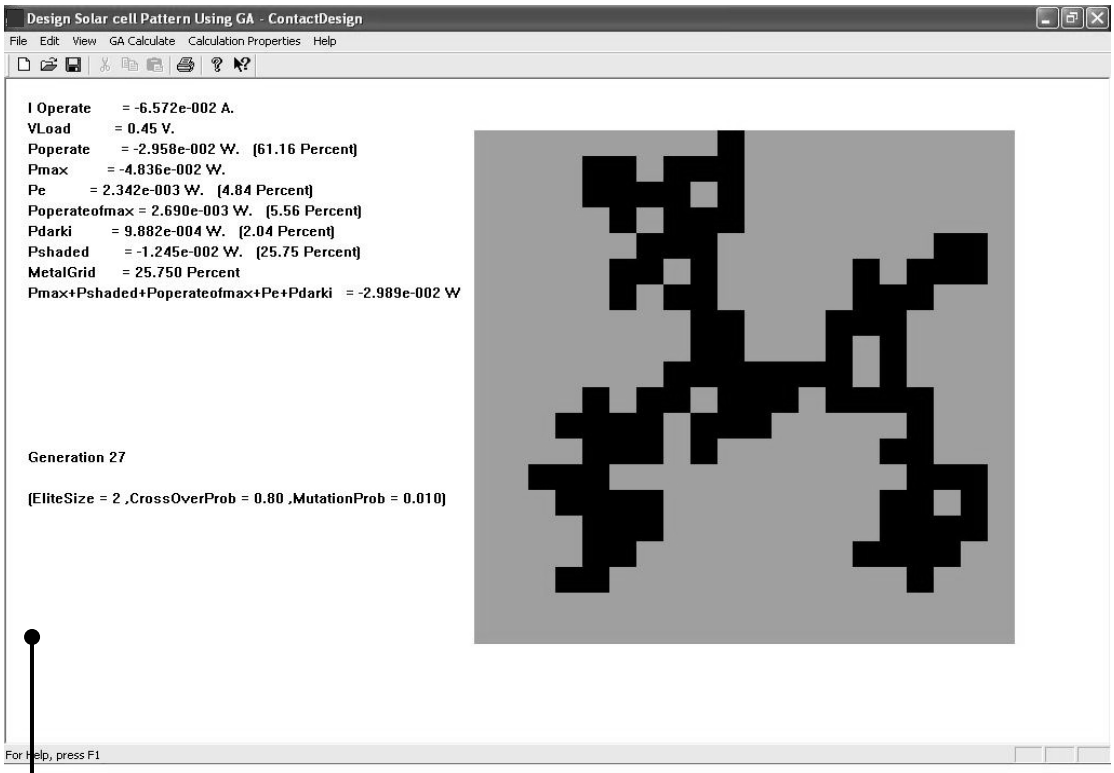
4. กำหนดค่าพารามิเตอร์ของเซลล์แสงอาทิตย์กระทำได้โดยการเลือกเมนู Calculation Properties → Solar Cell Parameter ซึ่งจะปรากฏไดอะล็อก Solar Cell Parameter ให้กำหนดค่าแล้วกด OK เพื่อยืนยันค่าที่กำหนด

2.2 การออกแบบรูปแบบขั้วโดยใช้หลักการเจเนติกอัลกอริทึม

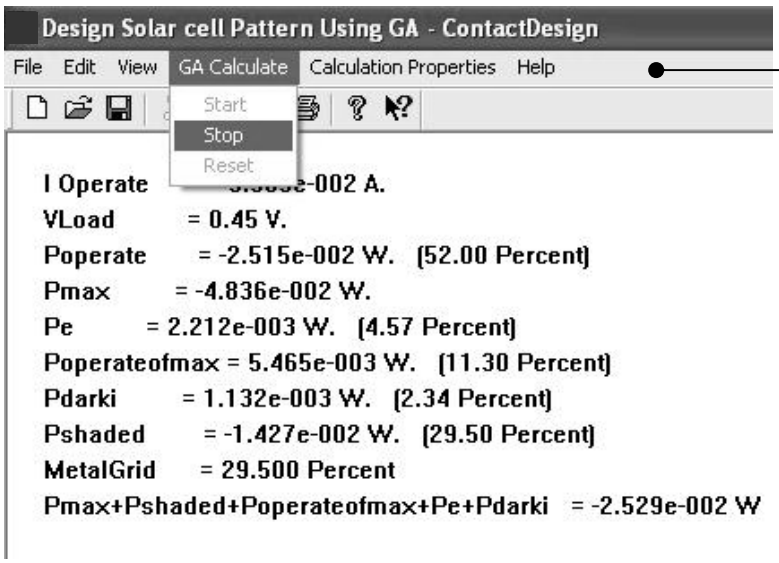
ในการออกแบบรูปแบบขั้วโดยใช้หลักการเจเนติกอัลกอริทึม สามารถกระทำได้ด้วยเมนู GA Calculate ดังนี้

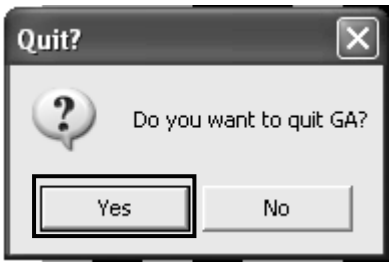


1. เมื่อต้องการเริ่มต้นการคำนวณให้เลือกเมนู GA Calculate → Start



2. เมื่อเริ่มการคำนวณแล้วหน้าต่างของโปรแกรมจะแสดงผลการคำนวณ ทั้งค่าคุณสมบัติของรูปแบบขั้วและรูแบบขั้วที่ดีที่สุดในแต่ละรุ่นของการคำนวณ





3. หากต้องการหยุดการคำนวณเพื่อปรับค่าพารามิเตอร์ในการคำนวณทำได้ด้วยการเลือดเมนู GA Calculate → Stop จะปรากฏไดอะล็อกเพื่อยืนยันการหยุดการคำนวณให้เลือก Yes เพื่อหยุดการคำนวณ และหาก ต้องการคำนวณต่อไปให้เลือกเมนู GA Calculate → Start เพื่อทำการคำนวณต่อไปได้เลย แต่หากต้องการเริ่มการคำนวณใหม่ให้เลือกเมนู GA Calculate → Reset ก่อน

ภาคผนวก (ค)

Manuscripts

Designing General Contact Patterns for Solar Cells using Genetic Algorithm

Meemak P. and Aiyarak P.
Innovation in Physics (I^2P) Research Unit, Department of Physics, Faculty of Science, Prince of Songkhla University, Songkhla 90112 Thailand.
Corresponding E-mail: pitipol.meemak@gmail.com

Abstract

The design of top contact grids of solar cells there is a trade-off between shadowing loss and electrical power loss in semiconductor. Power generated in solar cell can be maximized by sufficient coverage of the grid. Genetic algorithms were employed to minimize the total losses. We tested a number of different genetic algorithm parameters and operators. Experiment with 50 population size shows that the appropriate crossover and mutation probabilities are 0.4 and 0.01, respectively and 2D n-point crossover operators performed better than 1D n-point crossover. The genetic algorithms with the appropriate condition produce a general contact pattern that has the efficiency better than an optimized conventional pattern with the same size of metal width.

Keywords: Genetic Algorithm; Top Contact Patterns; Solar Cell

1. Introduction

Minimization of the losses caused by front-contact in a solar cell, i.e. shadowing loss and electrical power losses is one of the main contributions for its optimization. The electrical losses itself are actually made up of several components such as contact resistance power loss, metal grid resistance power loss and upper semiconductor resistance power loss. However, the emitter, at the top of the semiconductor, resistance is a function of its resistivity, thickness, and the geometry of the contact grid. As optimum cell performance requires a minimum of shading by the grid, a compromise must be reached between the opacity of the grid and its current collection efficiency.

Many published papers consider the minimum total loss for a particular geometry such as conventional contact pattern that is comprised of fingers and bus bars contact grid (Burgers and Eikelboom, 1997), (Cuevas, *et al.*, 1990), (Ganggopadyay, *et al*,. 2002), multi-layer contact grid (Flat and Milnes, 1979) and edge frame contact grid (König and Ebest, 2003). Although there is a research which can simulate the general geometries by consider the effect of the variation in emitter potential on the local current collection (Burgers, *et al*., 1993). How ever, this does not calculate the actual general geometries of contact pattern.

In this paper, we propose a novel technique based on genetic algorithm for designing general geometries contact grid. A current collection by top contact in silicon solar cell model (Burgers, *et al*., 1993), (Wyeth, 1977) is considered as a fitness function.

2. A Current Collection Model

Important quantities in a current collection model are the potential of the emitter $V_e(x, y)$ and the potential of the metal covering the solar cell $V_m(x, y)$. To simplify the model, the distribution of V_m will be neglected. A calculation of V_e is presented here with the following assumptions:

1. At a point (x,y) current is collected under influence of illumination as a function of the emitter potential according to a local 2-diode model.

$$J_e(V_e) = J_{01}\left(\exp\left(\frac{V_e}{V_b}\right)-1\right) + J_{02}\left(\exp\left(\frac{V_e}{2V_b}\right)-1\right) + \frac{V_e}{R_P} - J_{sc} \qquad (1)$$

It is assumed that the dark currents J_{01} and J_{02} are constant across the solar cell surface. V_b ,i.e. $\frac{kT}{q}$, is the Boltzmann voltage, while V_e, R_p and J_{sc} are the potential of emitter, the shunt resistivity and the short circuit current density, respectively.

2. The thickness of the emitter (t) is very much smaller than the lateral dimensions of the cell. Thus current flow in the emitter layer is assumed as shown in Figure 1 (Wyeth, 1977).

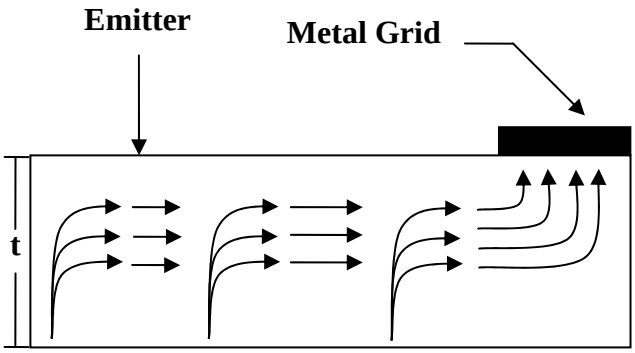


Figure 1(a) Schematic diagram of the actual current flow in the emitter layer.

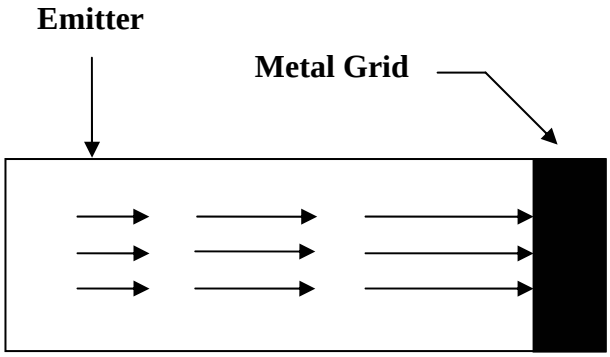


Figure 1(b) Schematic diagram of the assumed current flow in the emitter layer.

3. The resistance of metal grid (ρ_m) is much less than the sheet resistance of the emitter layer ($\rho_{e,\square}$). Hence, we can assume that the metal grid potential (V_m) is equal to the external load potential (V_{Load}).

$$V_{Load} = V_m \tag{2}$$

4. The current flow in the emitter layer is Ohmic.

The potential distribution in emitter can be described by a Poisson equation which is derived from the previous assumptions and the sheet resistance ($\rho_{e,\square}$) is assumed as constant across the cell thus we have:

$$\nabla^2 V_e(x, y) = -\rho_{e,\square} J_e(V_e) \tag{3}$$

For the emitter potential, two boundary conditions are applied. At the metal grid the potential is fixed at V_{Load} and current cannot flow across the boundary of the solar cell.

$$V_m = V_{Load} \tag{4}$$

$$\frac{\partial V_e}{\partial n} = 0 \tag{5}$$

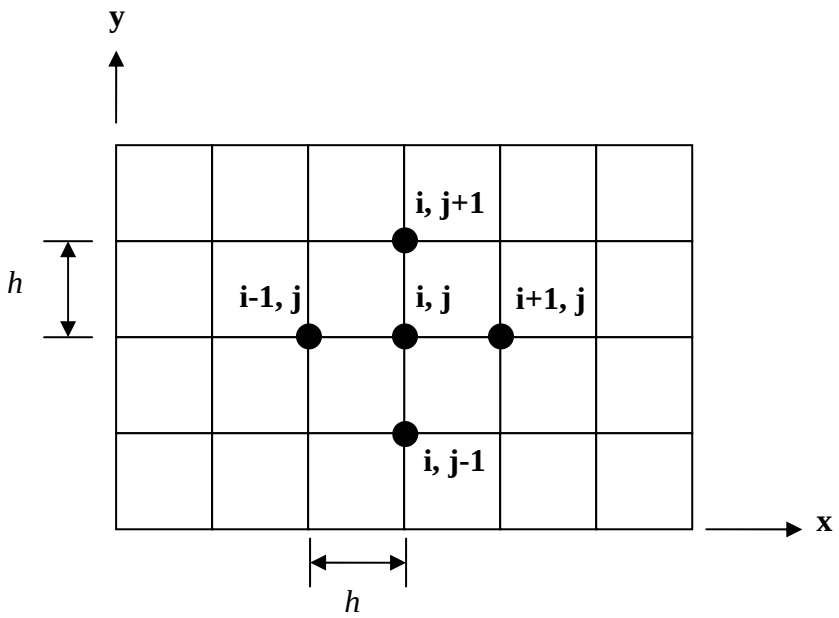


Figure 2 Rectangular arrays of grid points which represent the emitter layer.

We consider only rectangular cells since they are commonly used and they can be closely packed on a solar cell panel. The continuous equation in (3) has to be discretised as a network of resistors with current sources at the nodes so a rectangular array of grid points in Figure 2 is used. A finite difference method on a

rectangular grid is employed to discretise the Poisson equation thus we obtain the emitter potential at each grid point:

$$V_{i,j} = \frac{1}{4}\left(h^2 \rho_{i,j} J_{i,j}(V_{i,j}) + V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1}\right) \quad (6)$$

Every grid point on emitter is considered, leading to systems of non-linear equations. These can be solved by Newton iterations. Hence, the emitter distributed potential was found. Also the total collected current I_{out} can be computed.

$$I_{out} = \sum_{i,j} J_{i,j}(V_{i,j}).h^2 \quad (7)$$

Variation in emitter profiles and bulk diffusion lengths is reflected in the emitter sheet resistance, the dark current densities J_{01} and J_{02} and the short circuit current density J_{sc} .

3. Genetic algorithms

3.1 Principle

The Genetic Algorithms (GAs) are searching process based on the laws of nature selection and genetics (Coley, 1999). GAs are typically black-box methods that use fitness information. They do not require gradient information or other internal knowledge of the problem. GAs are population-based search techniques that maintain populations of potential solutions during searches.

In order to apply a genetic algorithm to a given problem, solutions of this problem must first be encoded as chromosomes (typically, bit strings). The population is composed of a group of chromosomes from which candidates can be selected for the solution of the problem. Each chromosome is made up of a number of subcomponents called genes. In order to evaluate each potential solution, GAs need an objective function that assigns a fitness value to a particular solution. A particular group of chromosomes is selected from the population to generate offspring by the defined genetic operations. The fitness of the offspring is evaluated in a similar fashion to their parents. The chromosomes in the current population are then replaced by their offspring.

A simple GAs usually consists of three operations: selection or reproduction, genetic operations and replacement (Coley, 1999).

- Selection. Methods of selecting individuals for reproduction are numerous and include roulette wheel sampling, stochastic universal sampling, elitism etc.

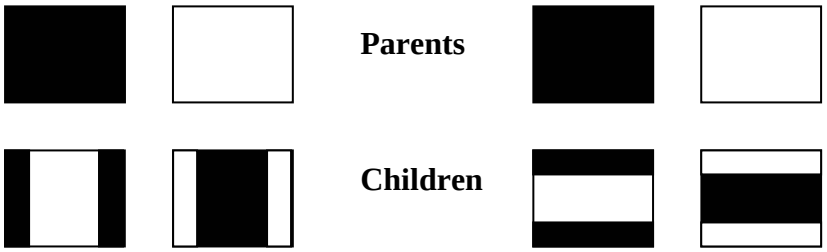
- Crossover. There are a number of variations of crossover such as 1D n-point crossover, 2D n-point crossover and uniform crossover. The 2D crossover operator is present in Figure 3 (Li, *et al*., 1995). We flip a coin to decide whether to do crossover vertically or horizontally. If we do crossover vertically, we randomly choose n-different numbers which smaller than the number of bits in vertical direction. A similar approach is used for the horizontal direction.

- Mutation. Mutation involves randomly flipping some of the bits in a string (chromosome). A very small probability is usually attached to occurrence of mutation

at each bit location. This operation is performed to ensure that new areas of the solution are explored

A GAs cycle is repeated for a fixed number of generations or until no more improvement is observed. The best chromosome is generated during the search is the final result of the GAs. Figure 4 illustrates a simple GAs flowchart.

The objective function is a main source to provide the mechanism for evaluating the status of each chromosome. This is an important link between the GAs and the problem.



(a) Vertical crossover **(b) Horizontal crossover**

Figure 3 The 2D n-point crossover operators (a) crossover along the horizontal direction and (b) the vertical direction.

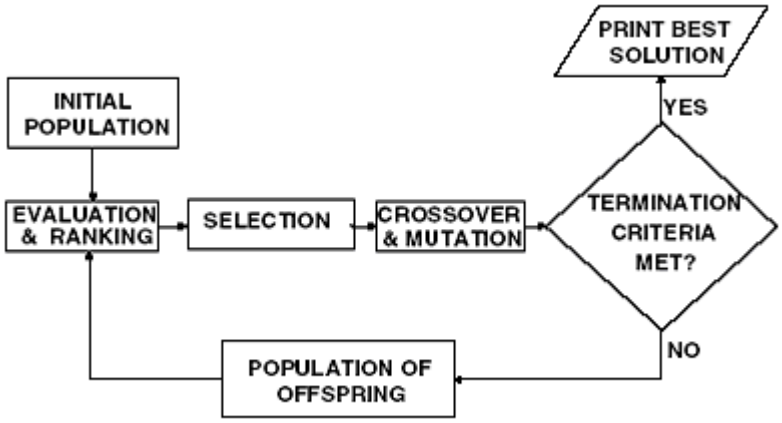


Figure 4 A Simple GAs Flowchart.

3.2 GAs Implementation for Designing Contact Patterns

A GAs library class was developed to design contact patterns for increasing the efficiency of solar cells. An implementation of the GAs for designing contact patterns is based on the following.

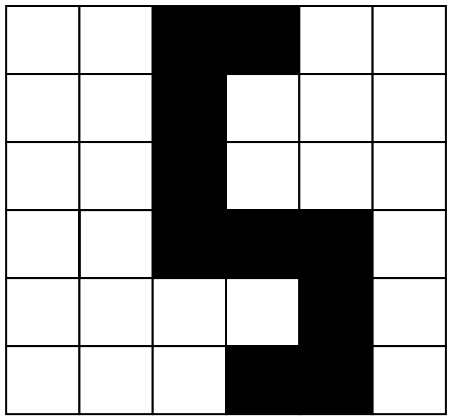
- Solution encoding: 400 bits of 2D binary chromosome represent a contact geometry of 2x2 cm. with 1 mm. of metal width. Figure 5 gives an example of 36 bits of 2D binary chromosome.

- Evaluation function: collected current I_{out} from equation (7).

- Initial population generation: randomly generated fifty chromosomes.

- Selection: stochastic universal sampling selection and elitism.
- Crossover: 1D n-point crossover or 2D n-point crossover.
- Mutation: Bit flipping.
- Maximum number of generations: 300

A set of contact grid geometry acts as the input data for GAs. A contact pattern is designed by maximization of a generated current from cell. We will test a number of different genetic algorithm parameters and operators i.e. crossover and mutation probabilities, crossover operator and number of crossover site. We will design both a conventional contact grid and a general geometry contact grid which have the same size of metal grid width to compare their efficiency.



(a)

0	0	1	1	0	0
0	0	1	0	0	0
0	0	1	0	0	0
0	0	1	1	1	0
0	0	0	0	1	0
0	0	0	1	1	0

(b)

Figure 5 (a) A contact grid geometry. (black and white represents metal and emitter area, respectively.) (b) A 36 bits of 2D binary chromosome representation. (‘1’ and ‘0’ represents metal and emitter area respectively.)

4. Results

We designed the contact pattern of a 2x2 cm^2 cell with 1 mm metal grid width and following characteristics: J_{sc} = 290 A/m^2, J_{01} = $0.9x10^{-7}$ A/m^2, J_{02} = $3.1x10^{-3}$ A/m^2, $\rho_{e,\square} = 40\,\Omega/\square$, $R_p = 0.21\,\Omega.m^2$, n_1 = 1 and n_2 = 2. At first we tested different probabilities of crossover and mutation with 50 population size in 300 generations. We found that choosing 0.4 as the probability of crossover and 0.01 as the probability of mutation provided good results.

We investigated GA performance with two different crossover operators (1D n-point and 2D n-point crossover) and number of crossover sites was considered. Figure 6 shows the effect of the number of crossover site and crossover operators. We found that 2D n-point crossover operators performed better than 1D n-point crossover and a larger number of crossover sites tend to worsen results. Hence 2D n-point crossover with one crossover site provides the best result.

In order to design the contact patterns we increased the generations to 2000 generations to ensure that we could obtain the best contact pattern. Figure 7 shows the best general contact pattern of a 2x2 cm^2 cell with 1 mm metal grid width. The extracted current from this cell and its contact efficiency is $7.75x10^{-2}$ Amp. and 72.13%, respectively. Furthermore we optimized the conventional contact pattern with the same size of metal grid width. The result shows that the extracted current from this cell and its contact efficiency is $7.56x10^{-2}$ and 70.37%, respectively. This result shows that the use of GAs for designing general contact patterns significantly provide the efficiency better than the optimized conventional pattern with the same size of metal width and hence improves the efficiency of the solar cell

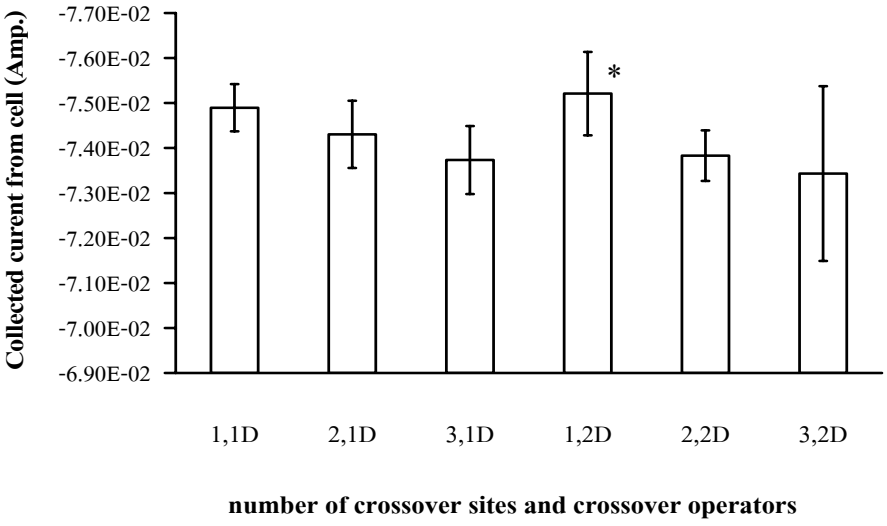


Figure 6 The effect of number of crossover site and crossover operators (1D n-point and 2D n-point crossover) (* is the best collected current condition)

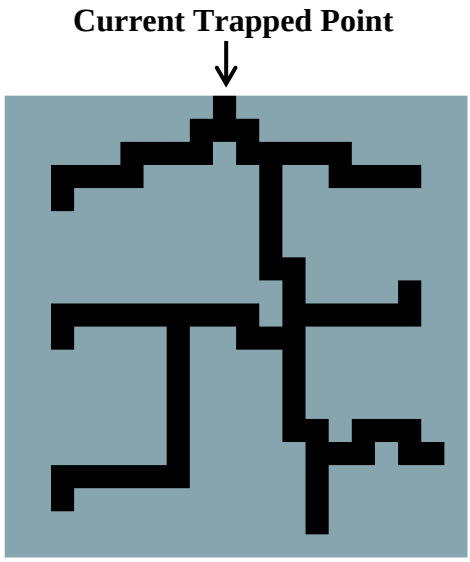


Figure 7 The best contact pattern of a 2x2 cm^2 cell with 1 mm. metal grid width that was designed by using GAs. The extracted current from this cell and the efficiency of cell is 7.75 x 10^{-2} Amp. and 72.13 %, respectively.

5. Conclusions

GAs with elite selection and 2D n-point crossover with one crossover site are employed to design a general contact pattern. The quality of best contact patterns obtained from GAs increases if the number of crossover site is decreased. To produce a good contact pattern, 2D n-point crossover operators is better than 1D n-point crossover. The results obtained, when compared with the optimized conventional pattern, indicate that the GAs are possible to design the general contact pattern to significantly increase the efficiency of solar cell.

Acknowledgment

We express special thank to Assoc. Prof. Boonlua Phongdara and Asst. Prof. Dr. Teparksorn Pengpan my adviser for suggestion about the computer programming and GAs method. And thank to Mr. Lersviriyanantakul Chaiwat for consultation about the program. Also thanks to DPST.

References

Burgers, A.R., *et al*. 1993. "2-D Numerical Analysis of Current Collection in Silicon Solar Cells", In Photovoltaic Specialists Conference, 1993., Conference Record of the Twenty Third IEEE 10-14 May 1993. pp. 340-346

Burgers, A.R. and Eikelboom, J.A. 1997. "Optimizing Metallization Patterns for Yearly Yield", Photovoltaic Specialists Conference, 1997., Conference Record of the Twenty-Sixth IEEE 29 Sept.-3 Oct. 1997. pp. 219-222

Coley, D.A. 1999. An Introduction to Genetic Algorithms for Scientists and Engineers. Singapore : World Scientific Publishing Co. Pte. Ltd.

Cuevas, A., *et al.* 1990. "26-percent efficient point-junction concentrator solar cells with a front metal grid", Electron Device Letters. 11(1990), 6-8.

Ganggopadyay, U., *et al.* 2002. "Front Grid Design for Plated Contact Solar Cells", In Photovoltaic Specialists Conference, 2002. Conference Record of the Twenty-Ninth IEEE 19-24 May 2002. pp. 399-402

Flat, A, and Milnes A.G. 1979. "Optimization of multi-layer front contact grid patterns for solar cell", Solar Energy. 23(1979), 289-299

König, D. and Ebest, G. 2003. "New Contact Frame Design for Minimizing Losses Due to Edge Recombination and Grid-Induced Shading", Solar Energy Materials & Solar Cells. 75(2003), 381-386.

Li, L.; Louis, S.J. and Brune, J. N. 1995. "Application of Genetic Algorithms to 2d Velocity of Seismic Refraction Data", In Proceedings of the Third Golden West International Conference on Intelligent Systems. Kluwer Academic Press.

Wyeth, N.C. 1977. "Sheet Resistance Component of Series Resistance in a Solar Cell as a Function of Grid Geometry", Solid-State Electronics. 20(1977), 629-634.