การสร้างพอร์ทหุ้นด้วยวิธีคำนวณเชิงพันธุกรรม

ดร.อาณัติ ลีมัคเดช[ℵ]

บทคัดย่อ

การสร้างพอร์ทหุ้น คือ การคำนวณหาน้ำหนักที่เหมาะสมในการลงทุนหุ้นแต่ละตัว ซึ่งวิธีการของ Markowitz (1952) เป็นวิธีที่ใช้แพร่หลายที่สุด โดยวิเคราะห์ปัญหาว่าผู้สร้างพอร์ทต้องการสร้างพอร์ทหุ้นที่มีความเสี่ยงต่ำสุด ภายใต้อัตราผลตอบแทนคาดการณ์ที่กำหนดค่าหนึ่ง สมการนี้สามารถแก้ได้ด้วยหลักการ Quadratic Programming อย่างไรก็ตามจุดอ่อนของวิธี Markowitz อยู่ที่การกำหนดอัตราผลตอบแทนคาดการณ์ในสมการ ซึ่งมักมีความผิดพลาดเพราะในทางปฏิบัติใช้ค่าเฉลี่ยอัตราผลตอบแทนจากอดีตแทนการคาดการณ์ Xia *et al* (2000) ได้เสนอให้ใช้การคาดการณ์อันดับของอัตราผลตอบแทนคาดการณ์แทน แม้ว่าวิธีนี้จะทำให้ความผิดพลาดในการจัดพอร์ทน้อยกว่าวิธี Markowitz แต่เครื่องมือคณิตศาสตร์ดั้งเดิม ไม่สามารถแก้สมการของ Xia *et al* ได้ จึงจำเป็นต้องใช้วิธีคำนวณเชิงพันธุกรรม (Genetic Algorithm) การศึกษานี้อธิบายวิธีคำนวณเชิงพันธุกรรม และวิธีใช้งานโปรแกรมฟรีแวร์ GOAL ในการวิเคราะห์การสร้างพอร์ทหุ้นแบบใหม่

บทนำ

นักลงทุนทุกคนทราบดีว่าสามารถลดความเสี่ยงของการลงทุนในหุ้นด้วยการลงทุนในหุ้นหลายตัว เพราะหากหุ้นตัวใดตัวหนึ่งมีราคาลดลง นักลงทุนยังมีหุ้นตัวอื่นที่มีกำไรมาชดเชยได้ อย่างไรก็ตามหลักการกระจายความเสี่ยงโดยลงทุนในพอร์ทของหุ้น (Portfolio) มิได้มีการอธิบายกลไกการทำงานอย่างเป็นระบบจนกระทั่ง Markowitz (1952) ได้นำเสนอแบบจำลองการสร้างพอร์ทขึ้น เพื่อคำนวณว่านักลงทุนควรจัดสัดส่วนการลงทุนในหุ้นแต่ละตัวอย่างไร Sharpe (1963) ได้เสนอวิธีการคำนวณแบบ Diagonal Principal เพื่อแก้สมการในแบบจำลอง Markowitz รวดเร็วขึ้น ทำให้เกิดการใช้แบบจำลอง Markowitz เพื่อสร้างพอร์ทหุ้นในกลุ่มบริษัทจัดการกองทุนอย่างแพร่หลาย อย่างไรก็ตามจุดอ่อนสำคัญของแบบจำลอง Markowitz คือ การกำหนดอัตราผลตอบแทนคาดการณ์ในสูตร เพราะการคำนวณมักนำข้อมูลอัตราผลตอบแทนในอดีตมาคำนวณค่าเฉลี่ยและใช้เป็นตัวแทนอัตราผลตอบแทนคาดการณ์ในทางปฏิบัติ

Xia *et al* (2000) เห็นว่าข้อมูลที่สะท้อนอัตราผลตอบแทนคาดการณ์ของหุ้นได้ดีกว่าค่าเฉลี่ยในอดีต คือ ข้อมูลการพยากรณ์ราคาหุ้นจากนักวิเคราะห์ Xia *et al* ได้เสนอแบบจำลองการจัดพอร์ท โดยกำหนดให้อัตราผลตอบแทนของหุ้นแต่ละตัวเป็นหนึ่งในสมการเงื่อนไขของระบบ และใช้การจัดอันดับอัตราผลตอบแทนซึ่งนักลงทุนสามารถคาดการณ์ได้ง่ายกว่ามาช่วยในการสร้างพอร์ท ข้อดีของแบบจำลองของ Xia *et al* คือ สามารถแปลงให้อยู่ในระบบสมการที่ "คล้าย" กับแบบจำลอง Markowitz ที่ผู้ใช้งานส่วนใหญ่มีความคุ้นเคย อย่างไรก็ตามระบบสมการของ Xia *et al* ไม่สามารถแก้สมการได้ด้วยวิธีการดั้งเดิม เพราะผลลัพธ์ที่ได้อาจเป็นเพียงค่าท้องถิ่น (Local Optimal

---

[ℵ] ผู้ช่วยศาสตราจารย์ประจำภาควิชาการเงิน คณะพาณิชยศาสตร์และการบัญชี มหาวิทยาลัยธรรมศาสตร์

Solution) Xia *et al* จึงเสนอให้ใช้วิธีคำนวณเชิงพันธุกรรม (Genetic Algorithm) เพื่อแก้ปัญหาในระบบสมการนี้ เนื่องจากการคำนวณเชิงพันธุกรรมเป็นวิธีการคำนวณแบบใหม่ บทความนี้จึงมีจุดมุ่งหมายในการอธิบายวิธีคำนวณเชิงพันธุกรรม พร้อมทั้งแสดงวิธีการใช้งานโปรแกรมฟรีแวร์ GOAL ในการสร้างพอร์ทหุ้นตามแบบจำลอง Xia *et al*

บทความนี้แบ่งเป็น 5 ส่วน โดยเริ่มจากการอธิบายวิธีจัดพอร์ทหุ้นแบบ Markowitz และจุดอ่อนของวิธีนี้ จากนั้นส่วนที่สองนำเสนอการสร้างพอร์ทหุ้นด้วยวิธีจัดอันดับอัตราผลตอบแทนของ Xia *et al* และอธิบายวิธีคำนวณเชิงพันธุกรรมในส่วนถัดมา ส่วนที่สี่แสดงวิธีการใช้งานโปรแกรม GOAL เพื่อแก้สมการสร้างพอร์ทหุ้นแบบใหม่ และข้อเสนอแนะในส่วนสุดท้าย

1. วิธีสร้างพอร์ทหุ้นแบบ Markowitz และจุดอ่อน

Harry Markowitz ขณะศึกษาปริญญาเอกทางเศรษฐศาสตร์ที่มหาวิทยาลัยชิคาโก ได้เสนอแนวความคิดในการวัดอัตราผลตอบแทนและความเสี่ยงจากการลงทุนอย่างเป็นระบบ โดยมองว่าอัตราผลตอบแทนของการลงทุนในหุ้นในไม่แน่นอน แต่เราสามารถคาดการณ์ได้โดยอาศัยค่าเฉลี่ยของอัตราผลตอบแทนในอดีต ถ้าการกระจายของอัตราผลตอบแทนนี้มีลักษณะเป็นการกระจายแบบปกติ (Normal Distribution) เราสามารถวัดความเสี่ยงของการลงทุนในหุ้นโดยคำนวณส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation) ของอัตราผลตอบแทน ถ้าค่านี้มากแสดงว่าความเสี่ยงของการลงทุนในหุ้นนั้นมากตามไปด้วย เพราะแสดงว่าโอกาสที่ผลตอบแทนที่เกิดขึ้นจริงจะแตกต่างจากค่าเฉลี่ยหรืออัตราผลตอบแทนคาดการณ์นั้นมีอยู่สูง

Markowitz ได้แสดงให้เห็นว่าถ้าเรานำสินทรัพย์ที่มีความเสี่ยง เช่น หุ้นตั้งแต่ 2 ตัวขึ้นไป มาประกอบกันเป็นพอร์ท (Portfolio) แล้ว เราจะพบว่าเราสามารถสร้างกลยุทธการลงทุนที่ให้อัตราผลตอบแทนคาดการณ์ค่อนข้างสูง และมีความเสี่ยงซึ่งวัดจาก Standard Deviation ของพอร์ทนั้นต่ำลงได้ ทั้งนี้เนื่องจากความสัมพันธ์ระหว่างหุ้น ยกตัวอย่างเช่น หุ้น A และ B ต่างเป็นหุ้นที่มีอัตราผลตอบแทนสูงและมีความเสี่ยงสูง แต่ทิศทางการเปลี่ยนแปลงของราคาหุ้น A และ B มักจะสวนทางกัน ถ้าเราทุ่มเงินทั้งหมดซื้อหุ้น A ตัวเดียว และราคาหุ้น A เกิดลง ผลตอบแทนของเราก็จะลดลงอย่างมาก หรือหากซื้อหุ้น B ตัวเดียวก็เสี่ยงต่อการที่ราคาหุ้น B ลดลง แต่ถ้าเราแบ่งเงินไปลงทุนในหุ้นทั้งสองตัวอย่าง “เหมาะสม” ตามแบบจำลองที่ Markowitz เสนอ เราจะสามารถลดความเสี่ยงจากการลงทุนไปได้มาก เพราะความสัมพันธ์ที่สวนทางกันของหุ้นทั้งสองทำให้กลุ่มหลักทรัพย์ที่มีหุ้นสองตัวมีความเสี่ยงน้อย เช่น ในกรณีที่ราคาหุ้น A ลดลง การขาดทุนนี้จะชดเชยด้วยกำไรจากราคาหุ้น B ที่สูงขึ้น

คำว่าแบ่งเงินไปลงทุนหุ้นจำนวน *n* หุ้นอย่าง “เหมาะสม” นั้น Markowitz ได้สร้างเป็นแบบจำลองโดยกำหนดอัตราผลตอบแทนคาดการณ์ของพอร์ทหุ้น และการวัดความเสี่ยงด้วยส่วนเบี่ยงเบนมาตรฐาน ดังแสดงในสมการที่ (1) และ (2) ตามลำดับ

$$E(R_p) = \sum_{i=1}^{n} w_i E(R_i) \qquad (1)$$

โดยที่ $E(R_i)$ คือ อัตราผลตอบแทนคาดการณ์ของหุ้น i โดย i = 1 ถึง n

w_i คือ สัดส่วนของการลงทุนในหุ้นแต่ละตัว

$E(R_p)$ คือ อัตราผลตอบแทนคาดการณ์ของพอร์ทหุ้น

สมการที่ (1) แสดงว่าอัตราผลตอบแทนคาดการณ์ของพอร์ทคือ ค่าเฉลี่ยถ่วงน้ำหนักของอัตราผลตอบแทนหุ้นแต่ละตัวนั่นเอง

$$\sigma_p^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{i,j} \qquad (2)$$

โดยที่ $\sigma_{i,j}$ คือ ความแปรปรวนร่วม (Covariance) ระหว่างหุ้น i และหุ้น j

σ_p^2 คือ ความแปรปรวนของพอร์ทหุ้น

สมการที่ (2) แสดงการวัดความแปรปรวนของอัตราผลตอบแทน ซึ่งสามารถใช้คำนวณหาส่วนเบี่ยงเบนมาตรฐานโดยถอดรากที่สองของสมการนี้ การแก้สมการของ Markowitz เราสามารถใช้สมการความแปรปรวนหรือส่วนเบี่ยงเบนมาตรฐานก็ได้ จะได้คำตอบเดียวกัน แต่การใช้สมการความแปรปรวนจะทำให้การแก้สมการง่ายกว่าจึงเป็นที่นิยม

จากสมการทั้งสอง Markowitz นำมาสร้างแบบจำลองพอร์ทหุ้น โดยมองว่านักลงทุนจะจัดพอร์ทเพื่อทำให้ตนเองได้รับอัตราผลตอบแทนคาดการณ์สูงสุด ภายใต้ความเสี่ยงที่กำหนดไว้ค่าหนึ่งได้ ดังนั้นแบบจำลองของ Markowitz จึงกำหนดให้สมการที่ 1 เป็นสมการเป้าหมาย ภายใต้ข้อจำกัดที่กำหนดโดยสมการที่ 2 ดังแสดงในระบบสมการ (3)

$$\underset{w_i}{Maximize} \quad E(R_p) \qquad (3.1)$$

subject to

$$\sigma_p^2 = k \qquad (3.2)$$

$$\sum_{i=1}^{n} w_i = 1 \qquad (3.3)$$

$$w_i \geq 0 \qquad (3.4)$$

สมการที่ (3.1) เป็นสมการเป้าหมายที่นักลงทุนต้องการทำให้อัตราผลตอบแทนคาดการณ์ของพอร์ทสูงสุด โดยมีเครื่องมือคือ การแบ่งสัดส่วนการลงทุนในหุ้นแต่ละตัวอย่างเหมาะสม (w_i)

สมการที่ (3.3) เป็นเงื่อนไขเพิ่มเติมว่าสัดส่วนการลงทุนทั้งหมดเมื่อรวมกันแล้วต้องมีค่าเป็น 1 และอสมการที่ (3.4) ใช้ในกรณีที่นักลงทุนไม่ต้องการ Short Sell หุ้น[1] ดังนั้นน้ำหนักการลงทุนในหุ้นแต่ละตัวต้องมีค่าเป็นบวกเท่านั้น หากนักลงทุนสามารถทำ Short Sell หุ้นได้ ก็จะตัดเงื่อนไขนี้ออกจากระบบสมการ

สมการที่ (3.2) แสดงเงื่อนไขโดยกำหนดให้ค่าของความแปรปรวนพอร์ทที่แสดงโดยสมการที่ (2) มีค่าเท่ากับค่าคงที่ค่าหนึ่ง ดังนั้นผลลัพธ์ในการแก้ระบบสมการนี้ซึ่งเป็นค่า w_i ชุดหนึ่งนั้นจึงเป็นค่าเจาะจงเมื่อความแปรปรวนของพอร์ทถูกกำหนดให้เท่ากับ k เท่านั้น

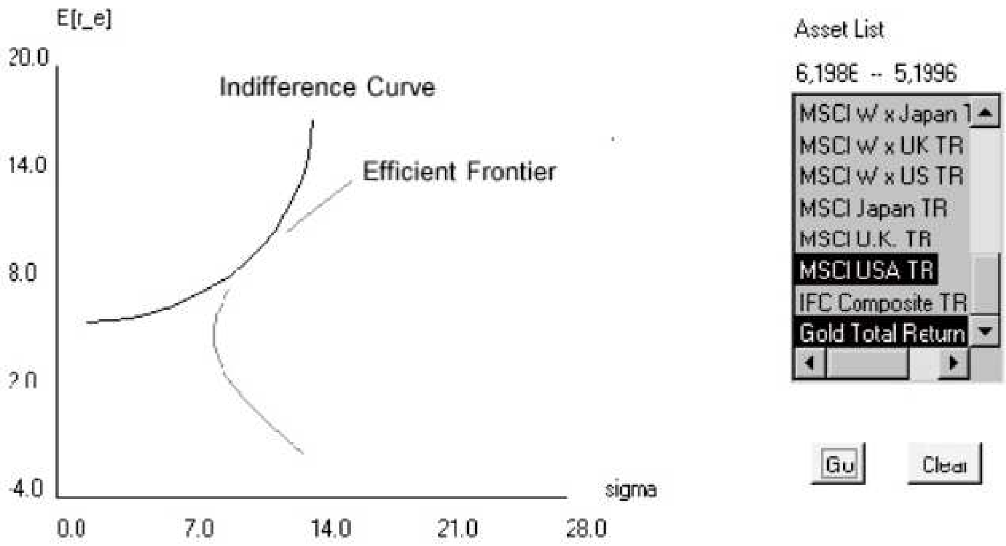
หากทำการแก้ระบบสมการ (3) ซ้ำโดยเปลี่ยนค่าคงที่ของความแปรปรวนเป็นค่าต่าง ๆ เราจะได้กลุ่มของค่า w_i ซึ่งแสดงสัดส่วนการลงทุนที่ดีที่สุด ที่ความเสี่ยงระดับต่าง ๆ หากเรานำผลการคำนวณอัตราผลตอบแทนคาดการณ์

---

[1] การ Short Sell คือ การขายหุ้นโดยผู้ขายไม่มีหุ้น Short Sell เป็นกลยุทธ์เก็งกำไรเมื่อนักลงทุนคาดว่าราคาหุ้นจะลดลง จึงขายในราคาตลาดขณะนั้น (โดยยืมหุ้นจากบริษัทหลักทรัพย์) และซื้อกลับเมื่อราคาลดลงตามที่คาดไว้ กลยุทธ์นี้มีความเสี่ยงสูง เพราะหากราคาหุ้นไม่ลดลงตามที่คาด จะก่อให้เกิดหนี้แทน

และความเสี่ยงของพอร์ท ณ สัดส่วนการลงทุนที่ดีที่สุดมาแสดงบนกราฟ โดยมีแกนตั้งเป็นอัตราผลตอบแทนคาดการณ์และแกนนอนแสดงความเสี่ยงแล้ว เราจะได้เห็นทางเลือกของการลงทุนต่าง ๆมีลักษณะเป็นเส้นขอบที่เรียกว่า Efficient Frontier

นักลงทุนจะทำการลงทุนโดยตัดสินใจตามระดับความกลัวความเสี่ยง (Risk Aversion) ของตนซึ่งสะท้อนผ่านความพอใจของการลงทุนด้วยเส้น Indifference Curve ตามหลักการของทฤษฎีเศรษฐศาสตร์จุลภาคทั่วไป นัก ลงทุนจะเลือกลงทุนในพอร์ทหุ้นที่เส้น Indifference Curve ของเขาสัมผัสกับเส้น Efficient Frontier ดังแสดงในรูปที่ 1



รูปที่ 1 การเลือกพอร์ทหุ้นตามแบบจำลอง Markowitz

ต่อมา Levy and Markowitz (1974) ได้ย่อปัญหาการสร้างพอร์ทให้กระชับขึ้น โดยนำตัวแปรระดับความกลัวความเสี่ยงของนักลงทุน เข้ามาใส่ในสมการเป้าหมายเลย เพื่อลดขั้นตอนการคำนวณ สมการเป้าหมายของระบบสมการที่ (3) จึงเปลี่ยนเป็นสมการแสดงความพอใจของนักลงทุนดังแสดงในระบบสมการที่ (4)

$$\underset{w_i}{Maximize} \quad (1-\gamma)E(R_p) - \gamma\sigma_p^2 \tag{4.1}$$

subject to

$$\sum_{i=1}^{n} w_i = 1 \tag{4.2}$$

$$w_i \geq 0 \tag{4.3}$$

โดยที่ γ คือ ตัวแปรวัดระดับความกลัวความเสี่ยง

การแก้ระบบสมการที่ (3) และ (4) เป็นการแก้ปัญหาเชิงคณิตศาสตร์ที่เรียกว่า Quadratic Programming with Linear Constraints[2] ซึ่งปัจจุบันมีโปรแกรมที่สามารถนำมาใช้งานได้อย่างหลากหลาย เช่น โปรแกรม Add-in Solver ที่ใช้ร่วมกับโปรแกรมสเปรดชีท Microsoft Excel สามารถแก้ไขปัญหาลักษณะนี้ได้

สมมติให้นักลงทุนกำลังพิจารณาการจัดสัดส่วนการลงทุนในหุ้น 3 ตัว คือ A B และ C ซึ่งมีค่าที่เกี่ยวข้องในการคำนวณดังตารางที่ 1 และค่าความกลัวความเสี่ยงของนักลงทุน (γ) ที่กำลังพิจารณาสร้างพอร์ทเท่ากับ 0.5[3]

ตารางที่ 1 ค่าสมมุติเพื่อใช้ทดลองสร้างพอร์ทการลงทุนแบบ Markowitz

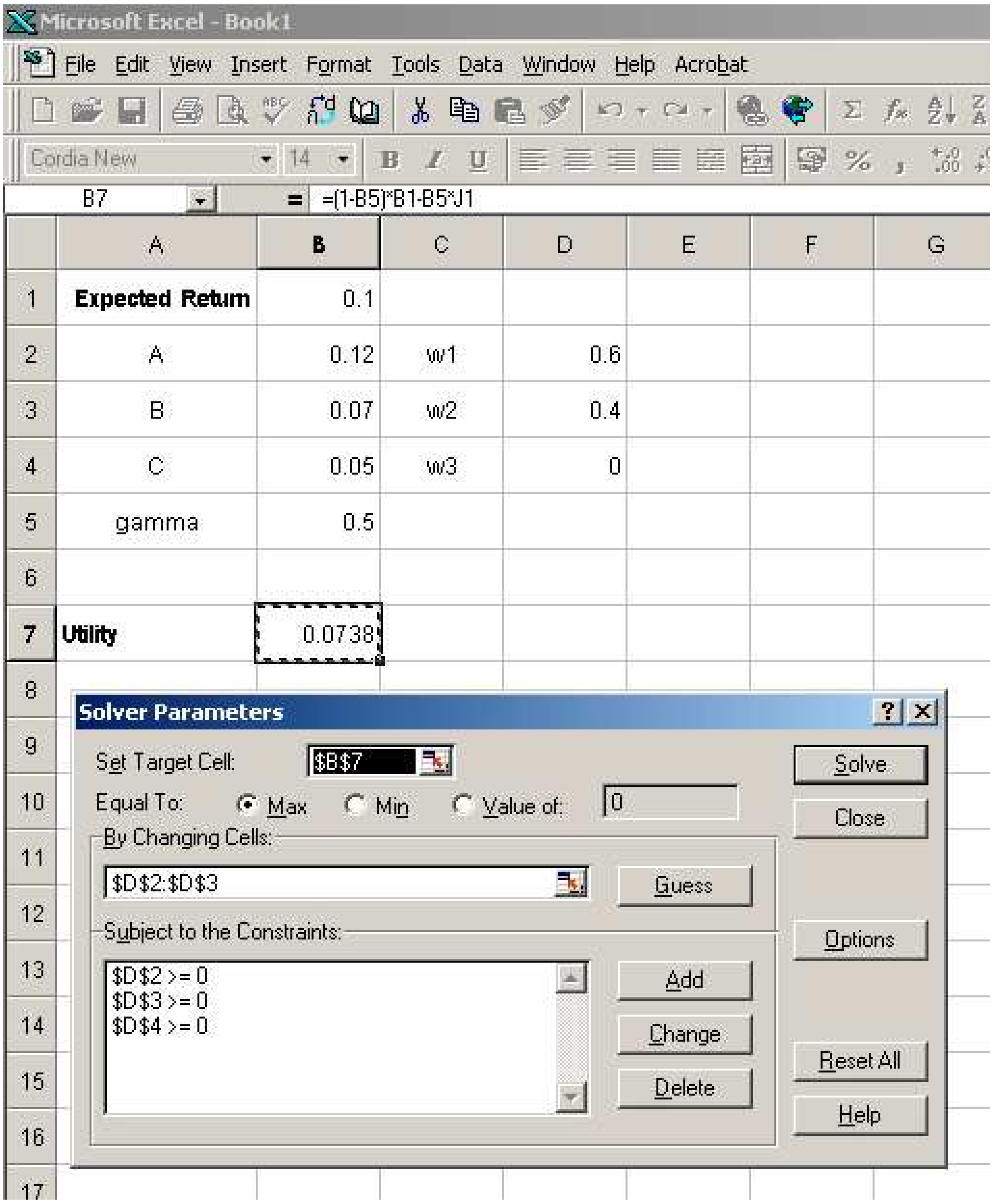
	หุ้น A	**หุ้น B**	**หุ้น C**
อัตราผลตอบแทนคาดการณ์	12%	7%	5%
ความแปรปรวนร่วม			
หุ้น A	9%	-23%	5%
หุ้น B		4%	-7.5%
หุ้น C			3.5%

รูปที่ 2 แสดงหน้าจอของ Solver ในการแก้ปัญหาตามแบบจำลอง Markowitz และรูปที่ 3 แสดงผลลัพธ์การจัดพอร์ทหุ้น 3 ตัวนี้

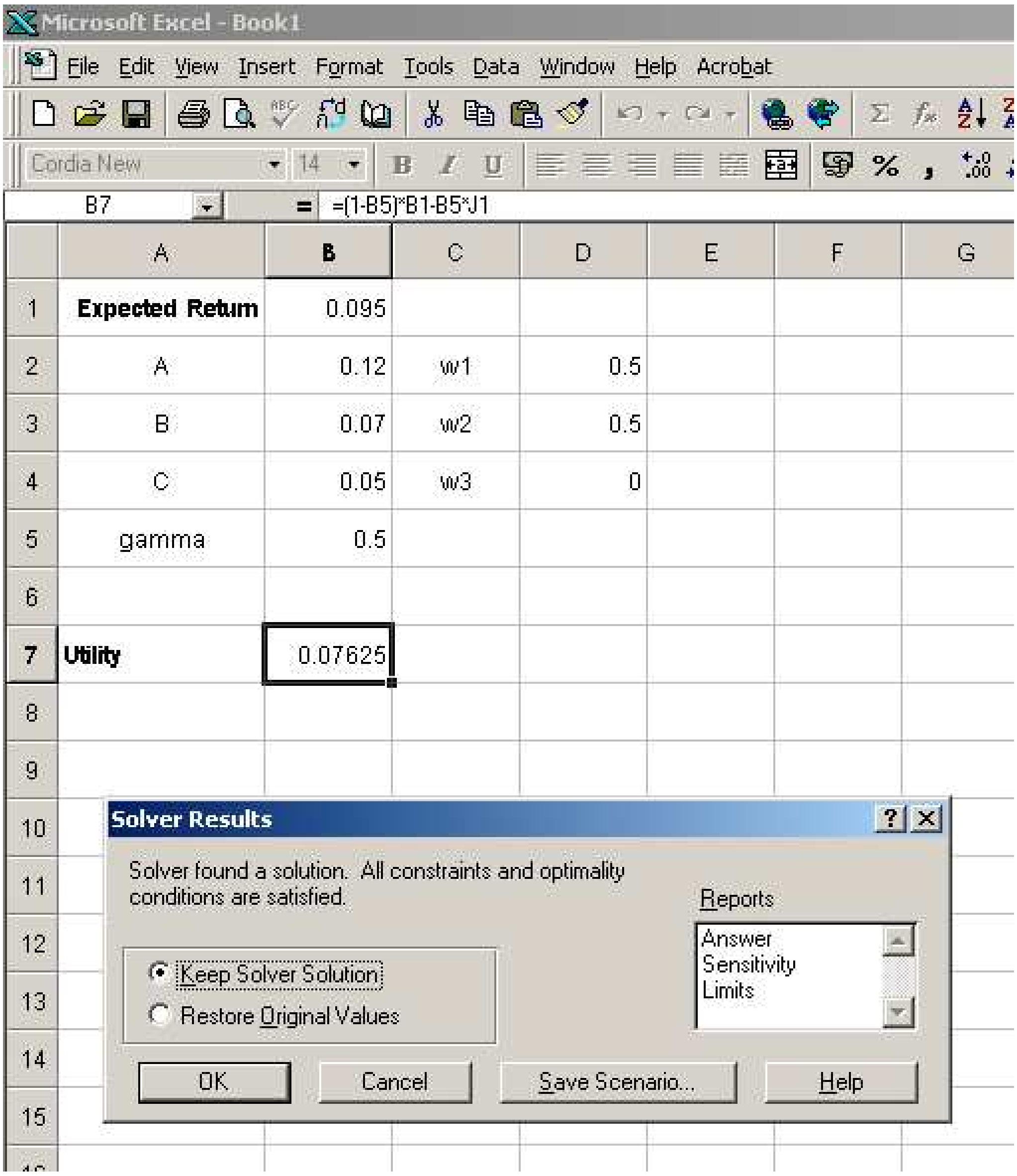
---

2 การแก้ปัญหาลักษณะ Optimization เช่นระบบสมการที่ (3) จะสามารถแปลงระบบสมการเป็นปัญหาการหาค่าต่ำสุดได้ (Minimization) โดยสลับสมการความแปรปรวนขึ้นมาเป็นสมการเป้าหมายและให้สมการอัตราผลตอบแทนคาดการณ์เป็นสมการเงื่อนไข พร้อมระบุค่าคงที่ค่าหนึ่งเป็นอัตราผลตอบแทนที่กำหนดในสมการเงื่อนไข ผลลัพธ์จะเหมือนเดิม การแปลงปัญหาในระบบสมการที่ (3) วิธีนี้ทำให้สมการเป้าหมายมีลักษณะเป็น Quadratic และสมการเงื่อนไขกลายเป็นสมการเชิงเส้น (Linear) จึงสามารถใช้วิธีแก้ปัญหาแบบ Quadratic Programming ได้

3 γ จะมีค่าระหว่าง 0 ถึง 1 โดยค่าที่เท่ากับ 0 แสดงว่านักลงทุนไม่คำนึงถึงความเสี่ยงเลยและจะสร้างพอร์ทด้วยการลงทุนในหุ้นที่ให้อัตราผลตอบแทนสูงสุดเพียงตัวเดียว ในขณะที่เมื่อมีค่าเป็น 1 นักลงทุนจะคำนึงถึงความเสี่ยงมากที่สุด และสร้างพอร์ทหุ้น โดยกระจายความเสี่ยงในลักษณะที่ทำให้ความเสี่ยงต่ำสุด



รูปที่ 2 การตั้งระบบสมการเพื่อแก้ปัญหาโดยโปรแกรม Solver



รูปที่ 3 ผลลัพธ์ของ Solver

ผลลัพธ์ของ Solver เสนอให้แบ่งการลงทุนในหุ้นเพียงสองตัว คือ A และ B ในสัดส่วน 50% เท่ากัน ซึ่งจะทำให้สมการเป้าหมายมีค่าสูงสุดที่ 0.076 แม้ว่าการแก้สมการตามแบบจำลอง Markowitz จะสามารถทำได้ง่ายและรวดเร็วด้วยเทคโนโลยีปัจจุบัน แต่จุดอ่อนที่สำคัญของแบบจำลองนี้คือ การกำหนดค่าอัตราผลตอบแทนคาดการณ์ในแบบจำลอง เนื่องจากค่านี้ไม่สามารถสังเกตได้ ในทางปฏิบัติจึงใช้ค่าอัตราผลตอบแทนเฉลี่ยที่คำนวณจากข้อมูลในอดีต ทำให้เกิดการผิดพลาดได้มาก จุดอ่อนในด้านนี้มีผู้พยายามแก้ไขโดยใช้เทคโนโลยีการคำนวณสมัยใหม่ ดังสรุปในตารางที่ 2

ตารางที่ 2 งานวิจัยเพื่อแก้ปัญหาการใช้อัตราผลตอบแทนคาดการณ์

ผู้วิจัย	ลักษณะการคำนวณ
Rustem *et al* (2000)	แสดงวิธีปรับพอร์ทหุ้น โดยจำลองค่าสถานการณ์ต่าง ๆที่เป็นไปได้ทั้งร้ายและดี เพื่อให้นักลงทุนสามารถประเมินภาพอัตราผลตอบแทนที่อาจเกิดขึ้น หากสถานการณ์เลวร้ายที่สุด
Xia *et al* (2000)	ใช้การคาดการณ์อันดับของอัตราผลตอบแทนแทนการระบุอัตราผลตอบแทนคาดการณ์ และแก้สมการด้วยวิธีคำนวณเชิงพันธุกรรม (Genetic Algorithm)
Avramov (2002)	ใช้วิธี Bayesian ในการพยากรณ์อัตราผลตอบแทน เพื่อสร้างพอร์ทหุ้น
Gaivoronski and Stella (2003)	ใช้การปรับพอร์ทผ่านระบบซื้อขายหุ้นออนไลน์ จึงสามารถใช้ข้อมูลราคาหุ้นล่าสุดในการคาดการณ์อัตราผลตอบแทนได้ตลอดเวลา ผู้วิจัยแสดงว่าแม้รวมต้นทุนการซื้อขายแล้ว นักลงทุนยังคงได้อัตราผลตอบแทนสูงกว่าปกติ

2. วิธีสร้างพอร์ทหุ้นด้วยวิธีจัดอันดับอัตราผลตอบแทน

เมื่อเปรียบเทียบงานวิจัยต่าง ๆที่พยายามแก้จุดอ่อนด้านการหาอัตราผลตอบแทนคาดการณ์ในแบบจำลอง Markowitz งานของ Xia *et al* (2000) มีความน่าสนใจเมื่อเทียบกับวิธีอื่น เพราะมีการใช้ข้อมูลจากการพยากรณ์จากนักวิเคราะห์เข้ามาประกอบ และมีความแม่นยำในการคาดการณ์มากกว่า เพราะใช้การคาดการณ์ “อันดับ” ของอัตราผลตอบแทน นอกจากนี้ Xia *et al* นำข้อมูลการพยากรณ์ของนักวิเคราะห์มาช่วยกำหนดเป็นช่วงของอัตราผลตอบแทนที่เป็นไปได้ในแบบจำลอง ที่สำคัญคือ แบบจำลองของ Xia *et al* มีโครงสร้างที่คล้ายคลึงกับแบบจำลอง Markowitz ที่ผู้ใช้งานส่วนใหญ่คุ้นเคยแล้ว ทำให้สร้างความเข้าใจได้ง่าย

Xia *et al* มองว่าเป็นการยากที่นักลงทุนหรือนักวิเคราะห์จะคาดการณ์อัตราผลตอบแทนในอนาคตได้อย่างแม่นยำ แต่เป็นการง่ายกว่าที่นักลงทุนจะคาดการณ์ “อันดับ” ของอัตราผลตอบแทน เช่น เรียงอันดับได้ว่า อัตราผลตอบแทนหุ้น A จะมากกว่าหุ้น B และ หุ้น B มากกว่าหุ้น C ด้วยความแม่นยำมากกว่าการระบุอัตราผลตอบแทนคาดการณ์ เช่นตัวอย่างในตารางที่ 1

จากระบบสมการที่ (4) Xia *et al* ได้ปรับระบบสมการของ Markowitz เป็นระบบสมการที่ (5)

$$\underset{w_i, E(R_i)}{Maximize} \quad (1-\gamma)E(R_p) - \gamma\sigma_p^2 \tag{5.1}$$

subject to

$$\sum_{i=1}^{n} w_i = 1 \tag{5.2}$$

$$E(R_i) \geq E(R_{i+1}), \quad i = 1,\dots,n-1 \tag{5.3}$$

$$a_i \leq E(R_i) \leq b_i, \quad i = 1,\dots,n \tag{5.4}$$

$$w_i \geq 0 \tag{5.5}$$

การคำนวณอัตราผลตอบแทนคาดการณ์ของพอร์ทหุ้น $E(R_p)$ ในระบบสมการนี้และการคำนวณความแปรปรวนยังคงเป็นไปตามแบบจำลอง Markowitz แต่อัตราผลตอบแทนคาดการณ์ไม่ได้เป็นตัวแปรภายนอก หากถูกกำหนดจากภายในระบบสมการด้วยตามเงื่อนไขในอสมการ 2 ชุด คือ อสมการ (5.3) เป็นอสมการ กำหนดอันดับของอัตราผลตอบแทนที่นักลงทุนคาดการณ์ มีจำนวนทั้งสิ้นเท่ากับ n-1 อสมการ และอสมการ (5.4) เป็นการกำหนดช่วงของอัตราผลตอบแทนหุ้นแต่ละตัวที่เป็นไปได้ มีจำนวนเท่ากับจำนวนหุ้นที่กำลังพิจารณาสร้างพอร์ท

Xia *et al* ได้ให้หลักการในการกำหนดอสมการที่ (5.3) และ (5.4) ดังนี้

การเรียงอันดับอัตราผลตอบแทน ได้กำหนดให้พิจารณาจากค่าเฉลี่ยถ่วงน้ำหนักของอัตราผลตอบแทนคาดการณ์ 3 ด้าน คือ

1. อัตราผลตอบแทนคาดการณ์จากค่าเฉลี่ยของข้อมูลในอดีต หรือ $E(R_{ai})$ เช่นเดียวกับวิธีของ Markowitz
2. แนวโน้มของอัตราผลตอบแทน หรือ $E(R_{ti})$ โดยพิจารณาจากแนวโน้มของข้อมูลว่ามีทิศทางเพิ่มขึ้นหรือลดลง แล้วพยากรณ์อัตราผลตอบแทนคาดการณ์จากแนวโน้มนั้น
3. การพยากรณ์อัตราผลตอบแทนจากข้อมูลทางบัญชี หรือ $E(R_{fi})$ นักวิเคราะห์อาจพยากรณ์อัตราผลตอบแทนของหุ้นจากการพยาการณ์กำไรต่อหุ้น ถ้าคาดว่าผลดำเนินการของบริษัทในอนาคตจะดีกว่าในอดีต อัตราผลตอบแทนคาดการณ์จากข้อมูลทางบัญชีจะอยู่สูงกว่า $E(R_{ai})$

เมื่อได้อัตราผลตอบแทนคาดการณ์ทั้ง 3 ด้านแล้ว สามารถคำนวณอัตราผลตอบแทนคาดการณ์ของหุ้นใด ๆ เพื่อจัดอันดับได้ตามสมการที่ (6)

$$E(R_i) = h_1 E(R_{ai}) + h_2 E(R_{ti}) + h_3 E(R_{fi}) \tag{6}$$

โดยที่น้ำหนักสำหรับหุ้นแต่ละด้าน (h_i) อาจได้มาจากการกำหนดของนักวิเคราะห์ตามการคาดการณ์สภาวะตลาดในช่วงนั้น

ส่วนการสร้างช่วงอันดับของอัตราผลตอบแทนคาดการณ์ของหุ้นใด ๆ นั้น สามารถนำการคาดการณ์ของนักวิเคราะห์มาประกอบด้วย เช่น หากนักวิเคราะห์ส่วนใหญ่เห็นว่าราคาหุ้นมีแนวโน้มเพิ่มขึ้น เราอาจกำหนดให้อัตราผลตอบแทนขั้นต่ำหรือ b_i ของหุ้นนั้นมีค่าเท่ากับอัตราผลตอบแทนเฉลี่ย และอัตราผลตอบแทนขั้นสูง หรือ a_i มีค่าเท่ากับค่าสูงสุดของอัตราผลตอบแทนคาดการณ์ทั้ง 3 ด้าน ในทางตรงข้ามหากคาดว่าหุ้นนั้นมีแนวโน้มของราคาลดลงจะกำหนดให้อัตราผลตอบแทนขั้นสูงมีค่าเท่ากับอัตราผลตอบแทนเฉลี่ย และอัตราผลตอบแทนขั้นต่ำมีค่าเท่ากับค่าต่ำสุดของอัตราผลตอบแทนทั้ง 3 ด้าน

ระบบสมการที่ (5) ไม่สามารถใช้วิธีการคำนวณดั้งเดิมคือ Quadratic Programming แก้สมการเช่นเดียวกับแบบจำลอง Markowitz เนื่องจากมีโครงสร้างที่ซับซ้อน การหาค่าสูงสุดของสมการเป้าจึงอาจได้ค่าท้องถิ่น (Local

Optimal Solution) แทนที่จะเป็นค่าสูงสุดที่แท้จริงได้ (Global Optimal Solution) Xia *et al* ได้เสนอให้ใช้วิธีคำนวณเชิงพันธุกรรม (Genetic Algorithm) เพื่อแก้สมการนี้ เนื่องจากวิธีการคำนวณเชิงพันธุกรรมเป็นวิธีคำนวณที่ค่อนข้างใหม่ ในส่วนถัดไปจะอธิบายแนวคิดของการคำนวณวิธีนี้

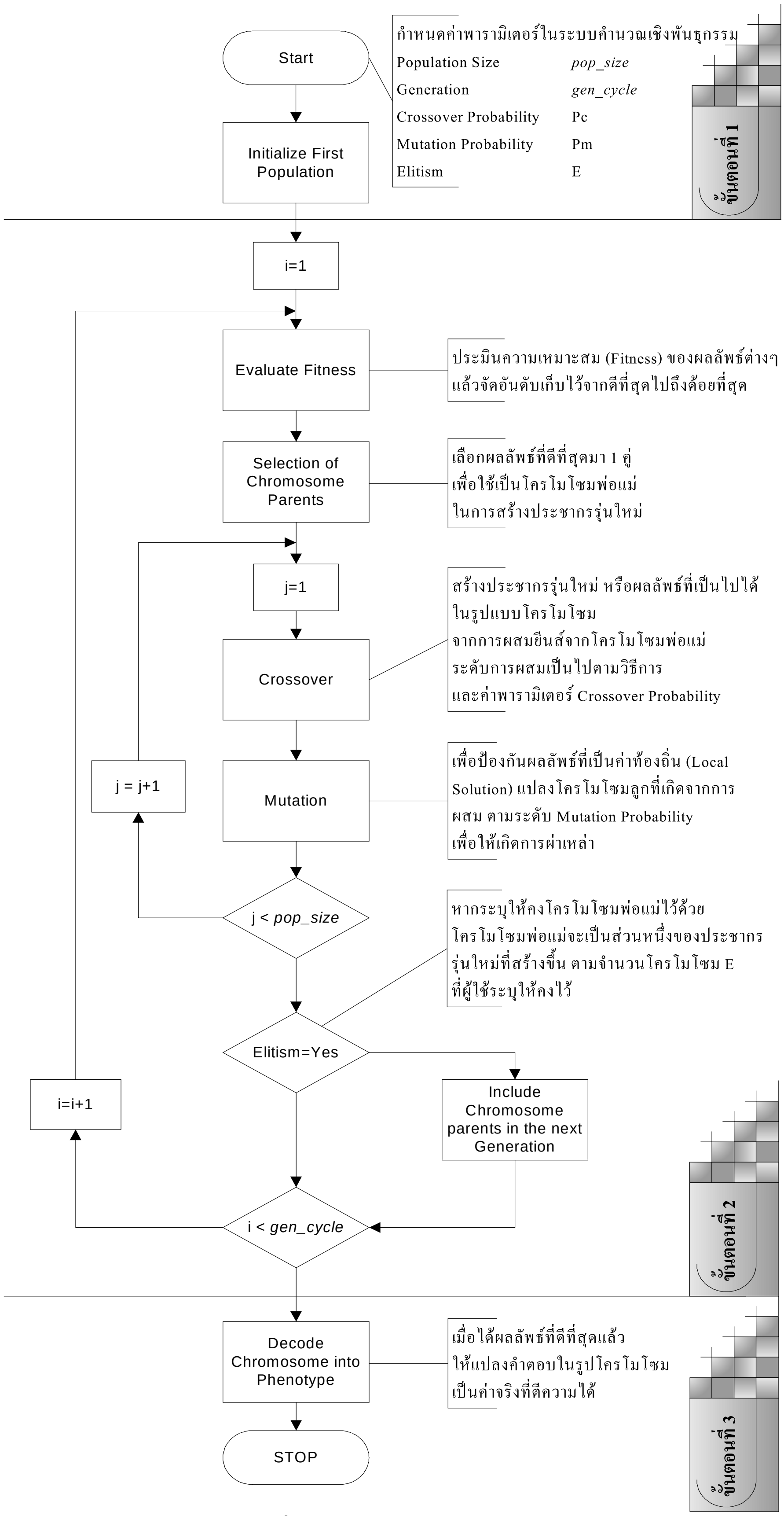
3. วิธีคำนวณเชิงพันธุกรรม (Genetic Algorithm)

Holland (1975) เสนอแนวคิดในการคำนวณโดยใช้หลักการทางชีววิทยาของทฤษฎีวิวัฒนาการว่า สิ่งมีชีวิตทุกชนิดประกอบด้วยเซลล์ ซึ่งแต่ละเซลล์จะมีกลุ่มของโครโมโซม (Chromosomes) ที่เหมือนกัน โครโมโซมประกอบด้วยยีนส์ (Genes) ซึ่งจะสะท้อนลักษณะเฉพาะของสิ่งมีชีวิตนั้น เช่น สีตา สีผม โดยยีนส์แต่ละตัวจะมีตำแหน่งของตนเองที่เรียกว่า Locus

เมื่อนำโครโมโซมทั้งหมดมาประกอบกันจะเรียกว่า Genome และเรียกกลุ่มของยีนส์ใน Genome นี้ว่า Genotype หรือรหัสพันธุกรรม ซึ่ง Genotypes นี้จะแปลงไปเป็นอวัยวะของสิ่งมีชีวิตต่อไปเรียกว่า Phenotype ตามทฤษฎีวิวัฒนาการสิ่งมีชีวิตที่แข็งแรงที่สุดจะมีโอกาสสืบทอดสายพันธุ์ที่แข็งแรงต่อไป การผสมยีนส์ของพ่อและแม่จะทำให้เกิดลูกซึ่งคัดลอกยีนส์ของพ่อแม่ที่ผสมกัน ทำให้เกิดสายพันธุ์ที่แข็งแรงยิ่งขึ้น บางครั้งการคัดลอกยีนส์ของพ่อแม่ไม่สมบูรณ์ อาจทำให้เกิดการผ่าเหล่า (Mutation) ในรุ่นลูก การผ่าเหล่าทำให้สิ่งมีชีวิตมีโอกาสพัฒนาสายพันธุ์ใหม่ที่ยิ่งเข้มแข็งขึ้น หากการผ่าเหล่าทำให้เกิดสายพันธุ์ด้อย สายพันธุ์ด้อยจะไม่สืบทอดต่อ

การแก้ปัญหาทางคณิตศาสตร์ที่ซับซ้อน เช่น การหาค่าสูงสุดหรือต่ำสุดของสมการเป้าหมายภายใต้เงื่อนไขต่างๆ เช่น แบบจำลองสร้างพอร์ทหุ้นในระบบสมการที่ (6) มักใช้กระบวนการค้นหาคำตอบที่ดีที่สุดจากการลองผิดลองถูกของกลุ่มคำตอบที่เป็นไปได้ ซึ่งมีกระบวนการค้นหาคำตอบหลายวิธี เช่น Hill Climbing, Tabu Search, Simulated Annealing รวมทั้งการคำนวณเชิงพันธุกรรมหรือ Genetic Algorithm

กระบวนการคำนวณเชิงพันธุกรรมสามารถอธิบายขั้นตอนได้ตามรูปที่ 4 ซึ่งเกี่ยวข้องกับกระบวนการที่สำคัญ 3 ขั้นตอนหลักได้แก่ หนึ่ง สุ่มคำตอบที่เป็นไปได้ สอง สร้างประชากร (Population) รุ่นใหม่และประเมินคุณภาพของคำตอบด้วยฟังก์ชันการประเมินความเหมาะสม (Fitness) กระบวนการนี้จะวนรอบตามจำนวนรุ่นประชากร และสาม แปลงคำตอบกลับมาอยู่ในรูปค่าจริงที่ตีความได้



รูปที่ 4 กระบวนการคำนวณเชิงพันธุกรรม

ในขั้นตอนที่หนึ่ง ระบบจะแปลง (Encode) ผลลัพธ์ที่เป็นไปได้ของปัญหาคณิตศาสตร์ให้อยู่ในรูปโครโมโซม ตัวแปรที่ต้องการค้นหาหมายถึง ยีนส์หนึ่งยีนส์เมื่อนำยีนส์มาประกอบกันจึงได้โครโมโซมหรือ Genotype การแปลงค่าทำได้หลายวิธี เช่น วิธี Binary, Permutation, หรือ Value Encoding ตารางที่ 3 แสดงตัวอย่างการแปลงในรูปแบบต่าง ๆ การเลือกใช้การแปลงแบบใดขึ้นอยู่กับลักษณะของปัญหา เช่น การแปลงแบบ Binary เหมาะสำหรับการคำนวณหาค่าสูงสุดหรือต่ำสุดตามเงื่อนไข การแปลงแบบ Permutation เหมาะกับปัญหาการจัดอันดับ เช่น การวางแผนเดินทางตัวเลขที่แสดงหมายถึง ลำดับของจุดที่เดินทาง ในขณะที่วิธี Value มีการแปลงได้หลายแบบ เช่น เป็นค่าจริง (Real Number) ตัวอักษร หรือกลุ่มวัตถุ ก็ได้ การแปลงชนิดนี้เหมาะกับปัญหาลักษณะพิเศษต่าง ๆ เช่น การแปลงแบบค่าจริงอาจใช้ในการค้นหาน้ำหนักที่เหมาะสมสำหรับการคำนวณฟังก์ชันหนึ่งได้ เมื่อกำหนดวิธีการแปลงแล้ว ระบบจะสุ่มสร้างกลุ่มโครโมโซมนี้ขึ้นมาเป็นประชากร จำนวนโครโมโซมในประชากรแต่ละรุ่นขึ้นกับค่าพารามิเตอร์ที่กำหนด ความยาวของโครโมโซมขึ้นอยู่กับจำนวนตัวแปรที่ต้องการหาค่าในระบบสมการ และมีผลต่อการกำหนดพารามิเตอร์จำนวนประชากร เพื่อหาคำตอบที่ดีที่สุดด้วย

ตารางที่ 3 ตัวอย่างการ Encode โครโมโซมแบบต่าง ๆ

วิธีการแปลง (Encoding)	ตัวอย่างโครโมโซม
Binary	101100101100101011100101
Permutation	1 5 3 2 6 4 7 9 8
Value	1.2324 5.3243 0.4556 2.3293 2.4545 ABDJEIFJDHDIERJFDLDFLFEGT (back), (back), (right), (forward), (left)

ระบบจะประเมินความเหมาะสม (Fitness) ของโครโมโซมในประชากรแต่ละชุด โดยพิจารณาจากสมการเป้าหมาย จากนั้นจะทำการคัดเลือกโครโมโซมที่เหมาะสมที่สุด 2 ชุดให้เป็นพ่อและแม่เพื่อใช้สืบทอดสายพันธุ์ (Reproduction) ให้แก่ประชากรรุ่นใหม่ อีกนัยหนึ่งประชากรรุ่นใหม่ที่จะสร้างขึ้นคือ ลูกที่เกิดจากการผสมยีนส์ที่ดีที่สุดของประชากรรุ่นก่อนนั่นเอง

การคัดเลือกนี้สามารถทำได้หลายวิธี เช่น Roulette Wheel, Boltzman, Tournament, Rank หรือ Steady State Selection หลักการกว้าง ๆของการคัดเลือกนี้คือ โครโมโซมที่เหมาะสมที่สุดมีโอกาสที่จะได้รับการคัดเลือกมากกว่าโครโมโซมที่ด้อยกว่า ยกตัวอย่างเช่นวิธี Roulette Wheel เปรียบเสมือนการนำโครโมโซมทุกตัวของประชากรรุ่นนั้นมาเขียนเป็นช่องเหมือนกับ Roulette ความกว้างในแต่ละช่องจะไม่เท่ากัน แต่เป็นสัดส่วนกับความเหมาะสมของโครโมโซมนั้น แล้วทอดลูกเต๋าหากตกช่องไหน จะเลือกโครโมโซมนั้น เนื่องจากโครโมโซมที่เหมาะสมกว่าจะมีช่องขนาดใหญ่กว่า ดังนั้นโอกาสถูกเลือกจึงมีมากกว่า อย่างไรก็ตามในบางกรณีที่มีความแตกต่างของความเหมาะสมในโครโมโซมมากเกินไป โดยโครโมโซมที่ดีที่สุดอาจมีสัดส่วนถึง 90% ของพื้นที่ทั้งหมด ซึ่งจะทำให้โครโมโซมอื่นไม่มีโอกาสถูกเลือกเลย เราอาจเปลี่ยนวิธีคัดเลือกเป็นวิธีอื่น เช่น Rank Selection

หลังจากคัดเลือกโครโมโซมพ่อแม่ได้แล้ว กระบวนการสร้างประชากรรุ่นใหม่จะเกิดขึ้นตามลำดับต่อไปนี้

- ผสมยีนส์ (Crossover) จากโครโมโซมพ่อและแม่ จากตัวอย่างในตารางที่ 4 โครโมโซม 1 และ 2 เป็นพ่อและแม่ ระบบจะกำหนดจุดแบ่ง (Crossover Point) ซึ่งแสดงด้วยสัญลักษณ์ | เพื่อแบ่งโครโมโซมเป็นสองส่วน จากนั้นโครโมโซมลูกจะถูกสร้างขึ้นใหม่จากการผสมยีนส์พ่อและแม่ การผสมยีนส์เกิดจาก

การกำหนดพารามิเตอร์ **Crossover Probability** ซึ่งมีค่าตั้งแต่ 0-100% โดย 0 หมายถึง การนำคัดลอกยีนส์พ่อและแม่มาทั้งส่วน ในขณะที่ 100% คือ การสร้างโครโมโซมของลูกขึ้นมาใหม่เลย

ตารางที่ 4 ตัวอย่างการสร้างโครโมโซมลูกจากการผสมยีนส์พ่อและแม่

Chromosome 1 (พ่อ)	**11011** I 00100110110
Chromosome 2 (แม่)	11001 I **11000011110**
Offspring 1 (ลูก)	11011 I 11000011110
Offspring 2 (ลูก)	11001 I 00100110110

ผู้ใช้ยังสามารถกำหนดจุดแบ่ง (Crossover Point) ในการผสมยีนส์ให้มีความซับซ้อนขึ้นอีกได้ ซึ่งจะทำให้เกิดผลลัพธ์ที่เป็นไปได้หลากหลายขึ้น เช่น Two Points Crossover ทำให้เกิดการผสมยีนส์เป็นโครโมโซมลูกดังนี้ **11** I 0010 I **11** + 11 I **0111** I 11 = **11011111**

- การผ่าเหล่า (Mutation) เมื่อเกิดการผสมยีนส์จากพ่อและแม่เพื่อถ่ายไปสู่ลูก จะทำให้โครโมโซมของรุ่นลูกกระจุกอยู่ในกลุ่มเดียวกับพ่อแม่ ในเชิงคณิตศาสตร์ผลลัพธ์ที่ได้อาจเป็นค่าสูงสุดหรือต่ำสุดเฉพาะในขอบเขตจำกัดหนึ่งเท่านั้น หรือเรียกว่าค่าท้องถิ่น (Local Optimal Solution) มิใช่ค่าสูงสุดหรือต่ำสุดที่แท้จริง (Global Optimal Solution) เพื่อป้องกันปัญหานี้ จึงควรกำหนด **Mutation Probability** ให้ยีนส์ของลูกมีการผ่าเหล่าได้ เช่น ในกรณี Binary Encoding หากเราได้โครโมโซมลูกมาเป็น 110**1**11100 แทนที่จะใช้รวมเป็นหนึ่งโครโมโซมของประชากรรุ่นลูก เราจะปรับยีนส์บางตำแหน่ง (Locus) ให้เป็นโครโมโซมใหม่ เช่น 110**0**11100 แล้วใช้ค่านี้แทน การทำเช่นนี้เพื่อทำให้การคำนวณเชิงพันธุกรรมมีโอกาสได้ค่าผลลัพธ์ที่ไม่ใช่ค่าท้องถิ่น ค่า Mutation Probability กำหนดได้ตั้งแต่ 0-100% โดย 0 หมายถึง ไม่มีการผ่าเหล่าเลย และ 100% หมายถึง มีการผ่าเหล่ายีนส์ทุกตัวในโครโมโซมลูก
- Elitism บางครั้งการสร้างประชากรลูกโดยผสมยีนส์จากโครโมโซมพ่อและแม่ซึ่งเป็นผลลัพธ์ที่ดีที่สุดจากประชากรรุ่นก่อน อาจทำให้สูญเสียผลลัพธ์ที่ดีอยู่แล้วไปได้ การกำหนด Elitism จึงเป็นการให้คงโครโมโซมพ่อและแม่ให้เป็นส่วนหนึ่งของประชากรรุ่นลูก เพื่อให้มีโอกาสสร้างสายพันธุ์ใหม่ในประชากรรุ่นถัดไปด้วย

กระบวนการสร้างประชากรรุ่นใหม่จะถูกทำซ้ำจนได้จำนวนโครโมโซมหรือผลลัพธ์เท่ากับที่กำหนดในพารามิเตอร์ **Population** การคำนวณจะย้อนกลับไปสู่การประเมินผลลัพธ์ของประชากรรุ่นใหม่ด้วยฟังก์ชันคณิตศาสตร์เพื่อคัดเลือกโครโมโซมที่ดีที่สุดมาสร้างประชากรรุ่นใหม่ต่อไป การคำนวณจะวนเป็นรอบจนกระทั่งได้ค่าคำตอบที่ดีที่สุด หรือครบจำนวนรุ่นของประชากรที่กำหนดในพารามิเตอร์ **Generation**

ตารางที่ 5 ได้สรุปคำศัพท์ทางพันธุกรรมและเชื่อมโยงกับความหมายทางการคำนวณเพื่อให้ผู้อ่านทบทวนแนวคิดและสร้างความเข้าใจการคำนวณเชิงพันธุกรรมได้ง่ายขึ้น

ตารางที่ 5 เปรียบเทียบศัพท์ทางพันธุกรรมและการคำนวณทางพันธุกรรม

ศัพท์ทางพันธุกรรม	ศัพท์การคำนวณทางพันธุกรรม
ยีนส์ (Gene)	ตัวแปรที่ต้องการค้นหาค่า
โครโมโซม (Chromosome) หรือ Genotype เป็นผลจากการนำยีนส์มารวมกัน	กลุ่มผลลัพธ์ที่เป็นไปได้ของตัวแปรต่าง ๆในระบบในรูปแบบโครโมโซม เช่น หากใช้วิธีแปลงแบบ Binary กลุ่มผลลัพธ์ที่เป็นไปได้อาจอยู่ในรูป 10010111 โดยตัวเลขในหลักที่ 1 ถึง 4 สะท้อนค่าตัวแปรที่ 1 และตัวเลขหลักที่ 5 ถึง 8 สะท้อนค่าตัวแปรตัวที่ 2
Phenotype	การแปลงกลุ่มผลลัพธ์ที่เป็นไปได้ของตัวแปรต่าง ๆ ในรูปโครโมโซมให้ออกเป็นค่าจริงของตัวแปรนั้น
จำนวนประชากร (Population)	จำนวนผลลัพธ์ที่เป็นไปได้สำหรับการทดสอบหาค่าที่ดีที่สุดในแต่ละรุ่นของประชากร
จำนวนรุ่นของประชากร (Generation)	จำนวนการทดลองทำซ้ำเพื่อค้นหาผลลัพธ์ที่ดีที่สุด
การคัดเลือกทางธรรมชาติ (Natural Selection)	กำหนดฟังก์ชันคณิตศาสตร์ด้วยวิธีต่าง ๆ เช่น Roulette Wheel เพื่อประเมินความเหมาะสมของผลลัพธ์ที่ได้
การสืบทอดเผ่าพันธุ์ (Reproduction) และการคัดลอกสายพันธุ์	กำหนดจุดแบ่ง (Crossover Point) ในโครโมโซมพ่อและแม่ (จากผลลัพธ์ที่ถูกคัดเลือกของประชากรรุ่นก่อน) เพื่อสร้างโครโมโซมใหม่ของลูกหรือผลลัพธ์กลุ่มใหม่
ผ่าเหล่า (Mutation)	วิธีปรับผลลัพธ์ที่เป็นไปได้เพื่อหลีกเลี่ยงการได้ผลลัพธ์ที่เป็นค่าท้องถิ่น (Local Optimal Solution) ของสมการเป้าหมาย
Elitism	การรักษาผลลัพธ์ที่ดีที่สุดจากประชากรรุ่นก่อนให้เป็นทางเลือกของการสร้างประชากรรุ่นใหม่ด้วย

4. โปรแกรมการคำนวณเชิงพันธุกรรม GOAL

ปัจจุบันมีผู้ผลิตโปรแกรมคำนวณเชิงพันธุกรรมขึ้นมาจำนวนมาก การศึกษานี้ให้ความสนใจโปรแกรมฟรีแวร์เพื่อให้สามารถใช้ประโยชน์ในการเรียนการสอนได้ นอกเหนือจากการใช้ในงานวิจัย โปรแกรม GOAL เป็นโปรแกรมฟรีแวร์ที่พัฒนาโดย Angel Martin ตั้งแต่ ปี 2002 ปัจจุบันรุ่นที่เผยแพร่คือ รุ่นที่ 2 ซึ่งผู้ใช้งานสามารถดาวน์โลดได้ฟรีจากเวบไซต์ http://www.geocities.com/geneticoptimization

เมื่อติดตั้งแล้ว ผู้ใช้สามารถดูรายละเอียดการใช้งานได้ด้วยเมนู Help ในส่วนนี้จะอธิบายการใช้โปรแกรม GOAL เพื่อสร้างพอร์ทหุ้น โดยสมมติให้มีหุ้น 3 ตัว คือ A, B และ C โดยเราได้คำนวณอันดับของหุ้นทั้งสามตัวนี้จากอัตราผลตอบแทนคาดการณ์ตามวิธีการที่เสนอโดย Xia *et al* (2000) แล้ว สมมติค่าความแปรปรวนร่วมของหุ้นทั้งสามตัวให้เหมือนกับตารางที่ 1 เราจะสามารถสร้างระบบสมการได้ดังนี้

$$\underset{w_i, E(R_i)}{Maximize} \quad (1-\gamma)E(R_p) - \gamma\sigma_p^2 \tag{7.1}$$

subject to

$$\sum_{i=1}^{n} w_i = 1 \tag{7.2}$$

$$E(R_A) \geq E(R_B) \tag{7.3}$$

$$E(R_B) \geq E(R_C) \tag{7.4}$$

$$0.05 \leq E(R_A) \leq 0.12 \tag{7.5}$$

$$0.05 \leq E(R_B) \leq 0.09 \tag{7.6}$$

$$0.01 \leq E(R_C) \leq 0.05 \tag{7.7}$$

$$w_i \geq 0 \tag{7.8}$$

สร้างไฟล์คำสั่งของ GOAL ตามระบบสมการที่ 7 โดยเปิดดูโครงสร้างจาก Help -> Advanced topics -> Structure of a text file generated by GOAL อ่านคำอธิบายซึ่งชี้ว่าโครงสร้างไฟล์คำสั่งประกอบด้วย 4 ส่วน คือ

ส่วนที่ 1 PARAMETERS เป็นส่วนที่กำหนดตัวแปรที่ต้องการค้นหาในแบบจำลอง พร้อมทั้งขอบเขตของค่าที่เป็นไปได้ ในกรณีนี้ตัวแปรคือ น้ำหนักในการลงทุนหุ้นทั้งสามตัว และอัตราผลตอบแทนคาดการณ์ เรายังสามารถกำหนดช่วงอัตราผลตอบแทนในอสมการที่ (7.6)-(7.8) ในส่วนนี้ได้ด้วย

ส่วนที่ 2 MODEL กำหนดสมการเป้าหมายที่เราต้องการหาค่า โดยสามารถใช้ภาษา Visual Basic ในการเขียนโปรแกรมคำนวณได้ ในส่วนนี้เราจะเริ่มด้วยการเตรียมข้อมูลของค่าคงที่ต่าง ๆ ได้แก่ ระดับการกลัวความเสี่ยง (γ) ความแปรปรวนร่วมก่อน แล้วจึงกำหนดสมการเป้าหมายตาม (7.1) ตัวแปรสำคัญที่เราต้องการค้นหาจากระบบสมการนี้คือ น้ำหนักที่ควรลงทุนในหุ้นแต่ละตัว ซึ่งมี 3 หุ้น อย่างไรก็ตามผลรวมของน้ำหนักเหล่านี้จะต้องเท่ากับ 1 เพื่อประหยัดเวลาในการคำนวณเราอาจไม่ต้องกำหนดตัวแปรน้ำหนักหุ้นตัวที่สามในสมการ แต่กำหนดให้น้ำหนักส่วนนี้เท่ากับ 1 ลบด้วยผลรวมของน้ำหนักหุ้นสองตัวแรก

ส่วนที่ 3 CONSTRAINTS เป็นการระบุสมการข้อจำกัดในระบบสมการนี้ ในที่นี้คือ การใส่อันดับของอัตราผลตอบแทนหุ้นต่าง ๆเปรียบเทียบกัน นอกจากนี้ เนื่องจากเราไม่ได้กำหนดตัวแปรน้ำหนักของหุ้นตัวที่สามในระบบสมการ เพื่อป้องกันไม่ให้น้ำหนักของหุ้นตัวที่สามติดลบ (ซึ่งหมายถึงการ Short Sell) เราจึงสร้างอสมการเงื่อนไขเพิ่มขึ้นมาอีกโดยกำหนดให้ผลรวมของตัวแปรน้ำหนักหุ้นตัวที่หนึ่งและสองต้องน้อยกว่า 1

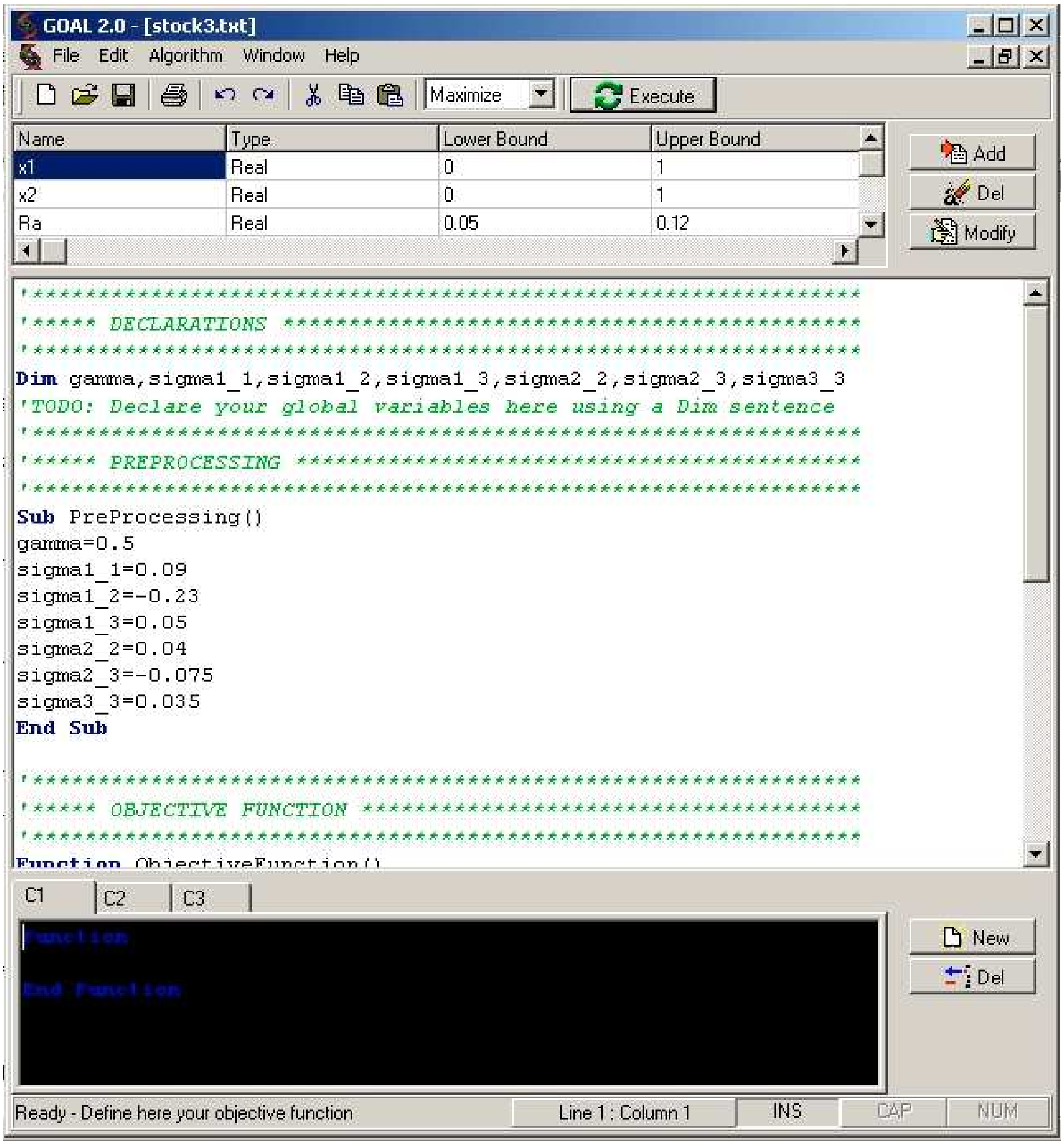
ส่วนที่ 4 ALGORITHM เป็นการกำหนดค่าพารามิเตอร์ในกระบวนการคำนวณเชิงพันธุกรรมที่อธิบายความหมายในส่วนที่แล้ว ซึ่งเราจะทดลองใช้ตามระบบที่ตั้งไว้ก่อน

เราสามารถคัดลอกข้อมูลโครงสร้างไฟล์คำสั่งมาตกแต่งเพิ่มเติมด้วยโปรแกรม Text Editor ทั่วไป เช่น Notepad ได้ ผลที่ได้รับคือ Text File ที่มีรายละเอียดดังแสดงในรูปที่ 5

```
PARAMETERS
{{
REAL: x1[0;1]
REAL: x2[0;1]
REAL: Ra[0.05;0.12]
REAL: Rb[0.05;0.09]
REAL: Rc[0.01;0.05]
}}

MODEL
{{

'***** DECLARATIONS ********************************************
Dim gamma,sigma1_1,sigma1_2,sigma1_3,sigma2_2,sigma2_3,sigma3_3
'TODO: Declare your global variables here using a Dim sentence

'***** PREPROCESSING *******************************************
Sub PreProcessing()
gamma = 0.5
sigma1_1=0.09
sigma1_2=-0.23
sigma1_3=0.05
sigma2_2=0.04
sigma2_3=-0.075
sigma3_3=0.035
End Sub

'***** OBJECTIVE FUNCTION **************************************
Function ObjectiveFunction()
      ObjectiveFunction = (1-gamma)*Return(x1,x2)-gamma*Variance(x1,x2)
End Function
Function Return(a,b)
      Return = x1*Ra+x2*Rb+(1-x1-x2)*Rc
End Function
Function Variance(a,b)
        covar1=2*x1*x2*sigma1_2+2*x1*x3*sigma1_3
        covar2=2*x2*x3*sigma2_3
        Variance=x1^2*sigma1_1+x2^2*sigma2_2+(1-x1-x2)^2*sigma3_3+covar1+covar2
End Function

'***** POSTPROCESSING ******************************************
Sub PostProcessing()

End Sub
}}

CONSTRAINTS
{{
#C1:
Function Constr_1()
      Constr_1 = Ra>=Rb
End Function
#C2:
Function Constr_2()
      Constr_2 = Rb>=Rc
End Function
#C3:
Function Constr_3()
      Constr_3 = x1+x2 < 1
End Function

}}

ALGORITHM
{{
POPULATION: 30
GENERATIONS: 100
REPRODUCTION_TYPE: 1
REPRODUCTION_PROBABILITY: 0.85
SELECTION_TYPE: 2
SELECTION_PROBABILITY: 0.85
MUTATION: 0.005
ELITISM: 1
REFRESH_PERIOD: 10
REFRESH_PERCENT: 10
}}
```

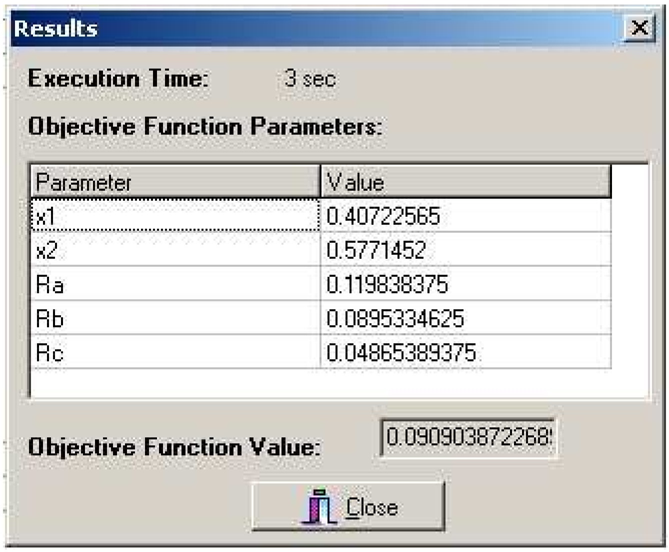
รูปที่ 5 ไฟล์คำสั่งเพื่อแก้ปัญหาระบบสมการที่ (7)

ใช้เมนู File -> Open ในโปรแกรม GOAL เพื่ออ่านไฟล์ที่สร้างนี้ โปรแกรมจะอ่านไฟล์และแสดงหน้าจอเพื่อเตรียมคำนวณดังแสดงในรูปที่ 6



รูปที่ 6 หน้าจอ GOAL หลังจากรับไฟล์คำสั่ง

กดปุ่ม Execute ในเมนูด้านบน โปรแกรมจะเริ่มการคำนวณ โดยระยะเวลาในการคำนวณจะขึ้นอยู่กับความซับซ้อนของปัญหา ระหว่างคำนวณโปรแกรมจะแสดงหน้าจอเล็กขึ้นมาเพื่อรายงานให้ทราบผลการคำนวณที่ดีที่สุดในแต่ละรุ่นของประชากร เมื่อคำนวณเสร็จแล้วจะแสดงหน้าจอผลลัพธ์ดังรูปที่ 7



รูปที่ 7 หน้าจอ GOAL เมื่อคำนวณเสร็จแล้ว

รูปที่ 7 แสดงว่าน้ำหนักที่นักลงทุนควรให้กับหุ้นตัวที่ 1 และ 2 คือ ประมาณ 41 และ 58 เปอร์เซ็นต์ของสัดส่วนเงินลงทุนทั้งหมดตามลำดับ ส่วนที่เหลือเพียง 1 เปอร์เซ็นต์จึงลงทุนในหุ้นตัวที่สามซึ่งเป็นหุ้นที่ให้อัตราผลตอบแทนต่ำสุด อัตราผลตอบแทนของหุ้นแต่ละตัวกลายเป็นค่าที่ถูกกำหนดจากระบบสมการด้วยซึ่งทุกค่าสอดคล้องกับพิสัยที่กำหนดไว้ ในขณะที่สมการเป้าหมายที่ต้องการค่าสูงสุดจะได้ค่าประมาณ 0.091 เราอาจทดลองเปลี่ยนค่าระดับความกลัวความเสี่ยงของนักลงทุน เพื่อทดลองดูสัดส่วนการลงทุนในหุ้นที่เปลี่ยนไปได้

นอกจากผลลัพธ์ที่แสดงหน้าจอแล้ว โปรแกรม GOAL จะทำการสร้างไฟล์รายงานผลลัพธ์โดยละเอียดให้ด้วย โดยจะมีชื่อเหมือนกับไฟล์คำสั่งแต่ต่อท้ายด้วย _report.txt ไฟล์นี้สามารถเปิดดูด้วยโปรแกรม Text Editor ทั่วไป ดังแสดงในรูปที่ 8

(อยู่ด้านท้าย)

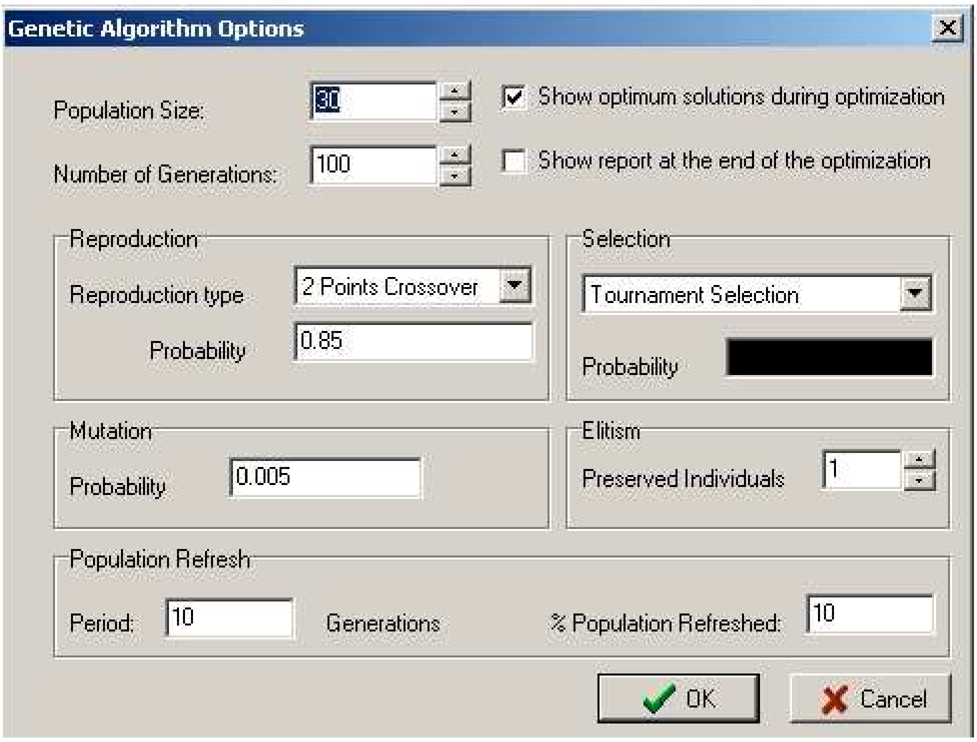
รูปที่ 8 รายงานผลการคำนวณโดยละเอียดของโปรแกรม

เพื่อความสะดวกในการอ่าน รายงานผลการคำนวณในรูปที่ 8 ได้ถูกดัดแปลงเล็กน้อย โดยตัดผลลัพธ์ในประชากรรุ่นที่ 13 ถึง 97 ออก นอกจากนี้ในส่วนของ Optimal Genotype ซึ่งมีความยาวถึง 160 ตัวอักษร ได้ถูกตัดทอนลงมาให้ดูพอเข้าใจ

ผลการคำนวณที่แสดงในรูปที่ 8 เริ่มจากข้อมูลเกี่ยวกับพารามิเตอร์ที่กำหนดในวิธีคำนวณ ซึ่งได้อธิบายความหมายของพารามิเตอร์แต่ละตัวไว้ในส่วนที่ 3 จากนั้นผลของการคำนวณจะแสดงรายละเอียดของผลลัพธ์ที่ดีที่สุดในแต่รุ่นประชากร เริ่มต้นจากการ “เดา” ค่าผลลัพธ์ที่เป็นไปได้ในประชากรรุ่นที่ 0 แสดงค่าเฉลี่ยของสมการเป้าหมายและส่วนเบี่ยงเบนมาตรฐานจากผลลัพธ์ทั้งหมดของประชากรรุ่นนั้น จำนวนผลลัพธ์ที่ผ่าเหล่า และผสมยีนส์ Optimum genotype เป็นผลลัพธ์ที่ดีที่สุดซึ่งแสดงในรูปแบบโครโมโซม ในที่นี้ใช้วิธีการแปลงค่าแบบ Binary และแปลงจากรูปโครโมโซมเป็นค่าที่อ่านได้ใน Optimum Phenotype ซึ่งสังเกตว่าจะมีอยู่ 5 ค่าใช้แทนตัวแปรทั้ง 5

ที่ต้องการหาค่าจากระบบสมการนั่นเอง คอลัมน์สุดท้ายแสดงค่าสูงสุดของสมการเป้าหมาย เราจะสังเกตได้ว่าในแต่ละรุ่น ค่าสมการเป้าหมายจะถูกปรับปรุงให้สูงขึ้น จนกระทั่งค่าเริ่มจะนิ่งที่ระดับ 0.0909 แสดงว่าเราไม่สามารถปรับปรุงค่าให้ดีกว่านี้แล้ว ดังนั้นการกำหนดจำนวนรุ่นประชากรให้เท่ากับ 100 ในตัวอย่างนี้จึงสมควรแล้ว หากเราพบว่าค่าสมการเป้าหมายยังคงมีแนวโน้มจะเปลี่ยนแปลงได้อีก เราอาจจะต้องเพิ่มจำนวนรุ่นของประชากรให้มากขึ้น

นอกจากการตั้งค่าพารามิเตอร์ของวิธีคำนวณผ่านไฟล์คำสั่งแล้ว ผู้ใช้ยังสามารถกำหนดผ่านเมนู Algorithm -> Options ของโปรแกรมได้ ดังแสดงหน้าจอในรูปที่ 9



รูปที่ 9 การกำหนดค่าพารามิเตอร์วิธีคำนวณ

5. ข้อเสนอแนะ

บทความนี้เป็นเพียงจุดเริ่มต้นของการนำวิธีคำนวณเชิงพันธุกรรมเข้ามาแก้ไขปัญหาทางการเงิน เช่น การสร้างพอร์ทหุ้น การแก้ปัญหาด้วยโปรแกรมฟรีแวร์ เช่น GOAL ทำให้นักวิจัยมีโอกาสนำวิธีคำนวณชนิดใหม่นี้ไปประยุกต์กับปัญหาทางการจัดการต่าง ๆ ได้อีก เช่น การวางแผนการผลิต การออกแบบระบบจัดส่งสินค้า

ในส่วนการสร้างงานวิจัยทางการเงินนั้น วิธีคำนวณเชิงพันธุกรรมทำให้การนำแบบจำลองที่ซับซ้อนมาใช้งานจริงมีความเป็นไปได้มากขึ้น เช่น การวางแผนทางการเงินที่พิจารณาโครงสร้างหลายช่วงเวลา แม้กระทั่งในส่วนการสร้างพอร์ทหุ้นเองก็ยังมีโอกาสขยายผลอีกมาก เพราะงานศึกษาเปรียบเทียบของ Xia *et al* (2000) ยังไม่สมบูรณ์ เนื่องจากเป็นการเปรียบเทียบวิธีการใหม่และวิธีการดั้งเดิมของ Markowitz เท่านั้น ยังไม่ได้มีการศึกษากับข้อมูลเชิงประจักษ์ว่าวิธีการใหม่จะให้อัตราผลตอบแทนที่สูงกว่าหรือความเสี่ยงต่ำกว่าวิธีของ Markowitz

นอกจากนี้ การประเมินค่าความเสี่ยงด้วยวิธีการอื่นนอกเหนือจากการใช้ส่วนเบี่ยงเบนมาตรฐานเพื่อสร้างพอร์ทหุ้น เช่น Value at Risk กำลังเริ่มเป็นที่นิยม การนำการสร้างพอร์ทหุ้นด้วยการจัดอันดับมาผสมกับการเปลี่ยนวิธีการวัดความเสี่ยงยังคงเป็นงานที่ท้าทายความสามารถของนักวิจัยอยู่

วิธีคำนวณเชิงพันธุกรรมยังสามารถใช้ทดแทนการประมาณค่าแบบจำลองเศรษฐมิติที่ซับซ้อนซึ่งเคยใช้วิธีประมาณค่าแบบ Maximum Likelihood ได้ เช่น แบบจำลองราคาหุ้น (Asset Pricing Model) หรือการคำนวณพารามิเตอร์โครงสร้างตลาดในการศึกษา Market Microstructure เพราะสามารถหาค่าได้รวดเร็วกว่า และสามารถหลีกเลี่ยงปัญหาผลลัพธ์ที่เป็นค่าท้องถิ่นได้

บรรณานุกรม

Avramov, D. (2002), "Stock Return Predictability and Model Uncertainty," *Journal of Financial Economics* 64, 423-458.

Gaivoronski, A., and F. Stella (2003), "On-line Portfolio Selection Using Stochastic Programming," *Journal of Economic Dynamics & Control* 27, 1013-1043.

Holland, J. (1975), Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press.

Levy, H., and H. Markowitz (1979), "Approximating Expected Utility by a Function of Mean and Variance," *American Economic Review* 69, 308-319.

Markowitz, H. (1952), "Portfolio Selection," *Journal of Finance* 7, 77-91.

Rustem, B., R.G. Becker, and W. Marty (2000), "Robust Min-max Portfolio Strategies for Rival Forecast and Risk Scenarios," *Journal of Economic Dynamics & Control* 24, 1591-1621.

Sharpe, W. (1963), "A Simplified Model of Portfolio Analysis," *Management Science*, 277-293.

Xia, Y., B. Liu, S. Wang, and K.K. Lai (2000), "A Model for Portfolio Selection with Order of Expected Returns," *Computers & Operations Research* 27, 409-422.