บทที่ 3

ระบบโปรแกรมคอมพิวเตอร์

ความก้าวหน้าทางเทคโนโลยี ด้านไมโครคอมพิวเตอร์ในปัจจุบัน ได้อำนวยให้การประมวลผลข้อมูลวิจัยด้วยเครื่องไมโครคอมพิวเตอร์ มีประสิทธิภาพมากยิ่งขึ้น เนื่องจากขนาดของหน่วยความจำมากขึ้น การประมวลผลรวดเร็วขึ้น และราคาของเครื่องลดลง อยู่ในระดับที่หน่วยงานหรือโครงการวิจัยต่าง ๆ สามารถจัดหามาใช้ได้สะดวกขึ้น ไมโครคอมพิวเตอร์ จึงเป็นอุปกรณ์ที่ช่วยให้นักวิจัยสามารถย่นระยะเวลาที่ใช้ในการเก็บ และจัดระเบียบข้อมูลวิเคราะห์แสดงและรายงานผลการทดลองได้เป็นอย่างมาก ดังนั้นจึงมีผู้เขียนโปรแกรมสำเร็จรูปที่ใช้ในการจัดข้อมูล และวิเคราะห์เชิงสถิติมากมาย ซึ่งในงานวิจัยนี้ก็เช่นเดียวกัน ผู้วิจัยได้ทำการพัฒนาโปรแกรมสำเร็จรูป ขึ้นมาใช้โต้ตอบร่วมกับผู้ใช้ โดยสามารถใช้ได้กับไมโครคอมพิวเตอร์สำหรับเครื่อง 16 บิท ได้อย่างมีประสิทธิภาพ

3.1 ประเภทของโปรแกรมสำเร็จรูปทางสถิติ

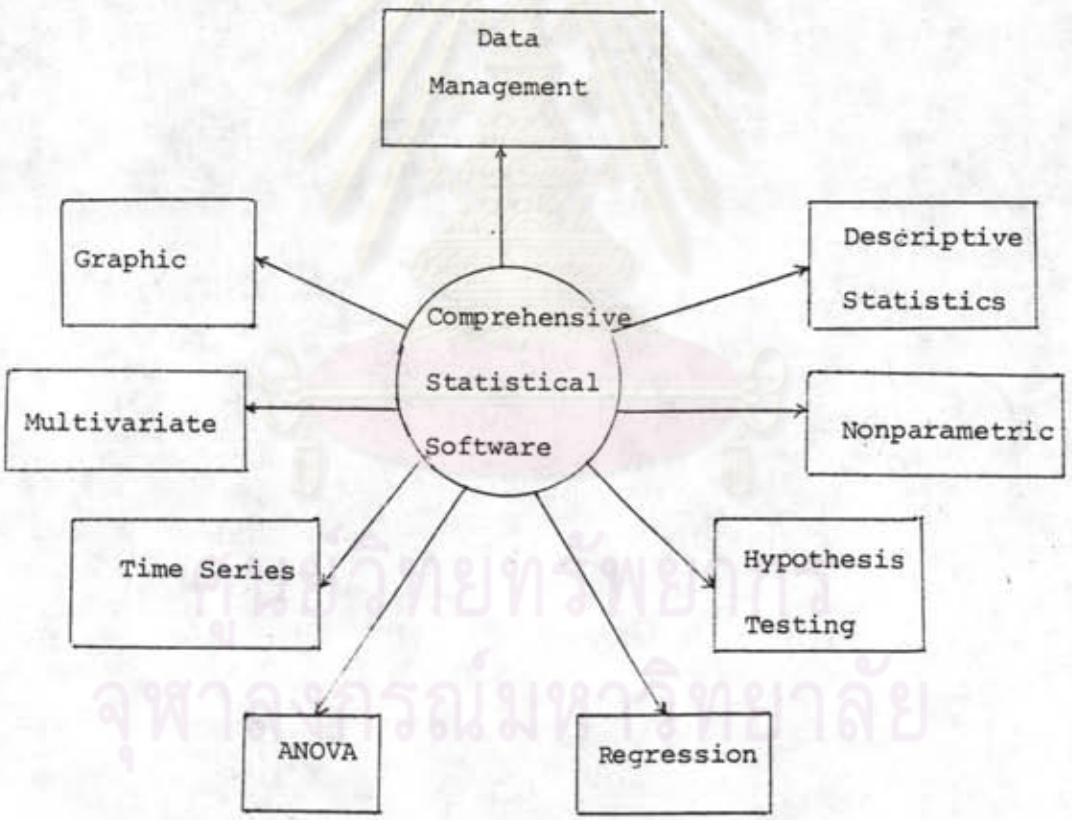
ในปัจจุบัน มีโปรแกรมสำเร็จรูปทางสถิติ (Statistical Software Packages; SSP) อยู่ไม่ต่ำกว่า 270 โปรแกรม ซึ่งอาจจำแนกตามองค์ประกอบของโปรแกรมได้ 2 ประเภทใหญ่ ๆ คือ

3.1.1 โปรแกรมสำเร็จ (Comprehensive Statistical Software)

โปรแกรมประเภทนี้ส่วนใหญ่จะประกอบด้วยโปรแกรมย่อยสำหรับจัดการข้อมูล วิเคราะห์ และแสดงผล (รูปที่ 3.1) จำหน่ายรวมกันเป็นชุด ปัจจุบันมีอยู่ไม่ต่ำกว่า 30 ชนิดในท้องตลาด เนื่องจากโปรแกรมประเภทนี้ เขียนขึ้นสำหรับงานวิเคราะห์ทางสถิติแบบเอนกประสงค์ อาจขาดรายละเอียดบางประการในงานวิเคราะห์บางชนิดไป อย่างไรก็ตาม โปรแกรมนี้มักจะมีขนาดใหญ่และเก็บไว้ในจานเก็บข้อมูลขนาด $5\frac{1}{4}$ นิ้ว จำนวน 1 แผ่น ขึ้นไปถึง 26 แผ่น ตัวอย่างของโปรแกรมเหล่านี้ได้แก่ ABSATAT, Microstat, NWA Statpak, Statpro, SPS, SPSS^{X} และ SYSTAT เป็นต้น

3.1.2 โปรแกรมที่เน้นเฉพาะด้าน (Specially Statistical Software)

เป็นโปรแกรมที่เขียนขึ้น เพื่อเน้นการวิเคราะห์เชิงสถิติประเภทใดประเภทหนึ่งมากกว่าอย่างอื่น จึงอาจให้รายละเอียดของ ผลการวิเคราะห์ประเภทนั้นมากกว่าโปรแกรมสำเร็จ แต่ข้อควรระวัง ในการเลือกใช้โปรแกรมประเภทนี้ คือ ควรพิจารณาอย่างรอบคอบว่า ข้อมูลที่สร้างขึ้นโดยโปรแกรมนี้สามารถนำไปใช้กับโปรแกรมเฉพาะอย่างประเภทอื่นได้หรือไม่ ทั้งนี้เพื่อป้องกันการที่จะต้องสร้างข้อมูลขึ้นมาใหม่อีก ในกรณีที่ต้องการวิเคราะห์เชิงสถิติประเภทอื่นต่อไป อาจแบ่งโปรแกรมเหล่านี้ได้เป็นกลุ่มย่อยคือ



รูปที่ 3.1 ลักษณะโครงสร้างของโปรแกรมวิเคราะห์เชิงสถิติแบบสำเร็จ (Comprehensive Statistical Software) สำหรับไมโครคอมพิวเตอร์

3.1.2.1 โปรแกรมเน้นประมวลผลข้อมูล การสำรวจ ทางเศรษฐกิจและสังคม ประกอบด้วยคำสั่งที่ใช้ในการป้อน ตรวจสอบ แก้ไข เก็บ และเรียกข้อมูลจากการสำรวจมาวิเคราะห์ นอกจากนี้ บางโปรแกรม อาจมีคำสั่งสำหรับวิเคราะห์เชิงสถิติ ชนิด Descriptive Statistics และ Cross Tabulation ตัวอย่างของโปรแกรมประเภทนี้ได้แก่ ISIS, Micro Survey, SNAP , Survey System เป็นต้น

3.1.2.2 โปรแกรมเน้นการพยากรณ์ข้อมูล มักจะเน้นการวิเคราะห์ชนิด Regression, Box-Jenkins และ Time Series ซึ่งใช้ในการพยากรณ์ข้อมูลด้านเศรษฐกิจ ตัวอย่าง เช่น โปรแกรม TWG ARIMA, MICRO-TSP

3.1.2.3 โปรแกรมเน้นด้านวางแผนการทดลอง และวิเคราะห์วาเรียนซ์ โปรแกรมประเภทนี้ในปัจจุบันยังมีจำนวนน้อย และส่วนใหญ่เป็นการวิเคราะห์ผลการทดลองทางสังคมศาสตร์ ตัวอย่างของโปรแกรมนี้ได้แก่ โปรแกรม ANOVA IIแต่สำหรับการวางแผนการทดลอง และวิเคราะห์ในแปลงทดลองทางเกษตรศาสตร์นั้น มีผู้เขียนเพื่อจำหน่ายน้อยมาก ผู้ใช้เครื่องไมโครคอมพิวเตอร์มักจะต้องเขียนโปรแกรมเฉพาะขึ้นใช้

3.1.2.4 โปรแกรมที่เน้นการสร้างกราฟ กราฟที่สร้างขึ้นด้วย โปร - แกรมประเภทนี้ ได้แก่ กราฟเส้น กราฟรูปแท่ง กราฟรูปวงกลมแบ่งส่วนได้ และ Scatter Plot บางโปรแกรมมีคำสั่งสำหรับสร้างเส้น และสมการชนิดเส้นตรงและโพลิโนเมียล จากข้อมูลด้วยคุณภาพและรายละเอียดของกราฟที่สร้างขึ้นจากโปรแกรมเหล่านี้ตลอดจนชนิดของเครื่องพิมพ์กราฟลงบนกระดาษที่โปรแกรมต้องการจะแตกต่างกัน ผู้ใช้จะต้องพิจารณาให้รอบคอบก่อนจะซื้อมาใช้ตัวอย่างของโปรแกรมประเภทนี้คือ Visi Plot, dGRAPH, CURFIT , Microsoft Chart เป็นต้น

3.1.2.5 โปรแกรมที่เน้นการใช้โปรแกรมเชิงเส้น มีผู้เขียน Software สำหรับโปรแกรมเชิงเส้น (Linear Programming) ไว้มากมาย ส่วนใหญ่ราคาไม่แพง และมีจำนวนมาก ตัวอย่างของโปรแกรมประเภทนี้ ได้แก่ Linear Programming และ LP/80 เป็นต้น

3.1.2.6 โปรแกรมที่เน้นเฉพาะด้านอื่น ๆ เป็นโปรแกรมสำเร็จรูปที่เน้นเฉพาะด้านอื่น ๆ ที่ไม่ได้กล่าวถึงอีกมากมายในปัจจุบัน เนื่องจากความเจริญก้าวหน้าทางเทคโนโลยีสูงมาก และทันสมัย ส่วนโปรแกรม ที่เกี่ยวกับการประมาณค่าพารามิเตอร์ของการแจกแจงของผลิตภัณฑ์ยารูดในงานควบคุมคุณภาพ ในงานวิจัยนี้เป็นโปรแกรมประเภทหนึ่งที่เน้นการ

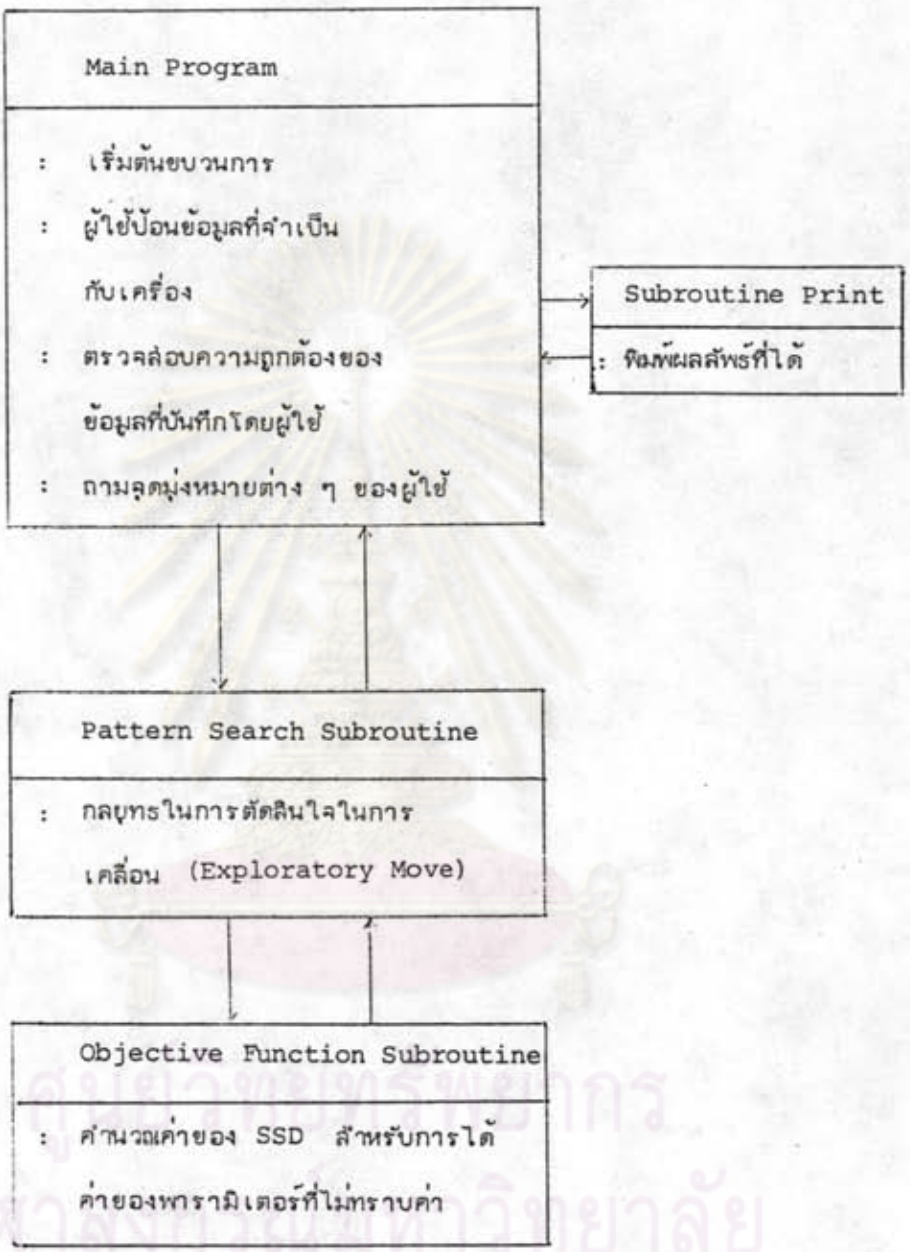
ใช้ในการควบคุมคุณภาพตามโรงงานอุตสาหกรรมที่เกี่ยวข้องกับการผลิตทั่วไป และเป็นโปร - แกรมที่สามารถสั่งทำงานโดยอาศัยเมนู (Menu Driven) มาแสดง ซึ่งจะใช้ได้ง่าย เหมาะสำหรับ ผู้ที่เริ่มใช้เครื่องคอมพิวเตอร์ และไม่มีประสบการณ์ทางภาษาคอมพิวเตอร์อีกด้วย เพราะจะมีคำอธิบายช่วยเหลือโดยตลอด

3.2 ระบบโปรแกรมคอมพิวเตอร์

โปรแกรมคอมพิวเตอร์ของงานวิจัยนี้ ถูกเขียนเป็นภาษาเบสิค ซึ่งประกอบด้วย โปรแกรมหลัก (Main Program) และ 3 โปรแกรมย่อย ประกอบด้วย

- Pattern Search Subroutine
- Objective Function Subroutine
- Subroutine Print

ลำดับของการดำเนินงานของระบบ แสดงให้ดูในรูปที่ 3.2 โดยที่โปรแกรมหลักจะเริ่มด้วยใส่ตัวแปร และโต้ตอบร่วมโดยทันทีกับผู้ใช้ให้ข้อมูลในส่วนที่จำเป็นต่อการวิเคราะห์ตัดสินใจในการกำหนดค่าเริ่มต้นของค่าพารามิเตอร์ของตัวแบบที่เลือกไว้ อีกทั้งยังตรวจสอบความถูกต้องของข้อมูลที่บันทึกเข้าไปโดยผู้ใช้ จากนั้นการควบคุมจะนำส่งไปยัง Pattern Search Subroutine ในการหากลยุทธในทิศทางการเคลื่อนที่ แล้วส่งค่าไปยัง Objective Function Subroutine เพื่อคำนวณค่า SSD (Sum of Squares of Differences) ซึ่งเป็นค่าของผลรวมกำลังสองของผลแตกต่างระหว่างค่าสังเกต หรือข้อมูลที่บันทึกเข้าไปกับค่าจากตัวแบบที่คำนวณได้โดยระบบโปรแกรมคอมพิวเตอร์ แล้วการควบคุมก็ถูกส่งไปยัง Main Program อีกครั้งหนึ่ง แล้วส่งไปยัง Subroutine Print เพื่อพิมพ์ค่าสุดท้ายของพารามิเตอร์ ที่ประมาณขึ้นของการเลือกตัวแบบ Prior Distribution ขั้นต่อไป การควบคุมจะกลับไปสู่ Main Program อีกครั้ง เพื่อถามความมุ่งหมายของผู้ใช้อีก ซึ่งจะดำเนินการอีกตามเมนูหรือจะหยุดงาน



รูปที่ 3.2 ระบบโปรแกรมคอมพิวเตอร์

3.2.1 Pattern Search Subroutine

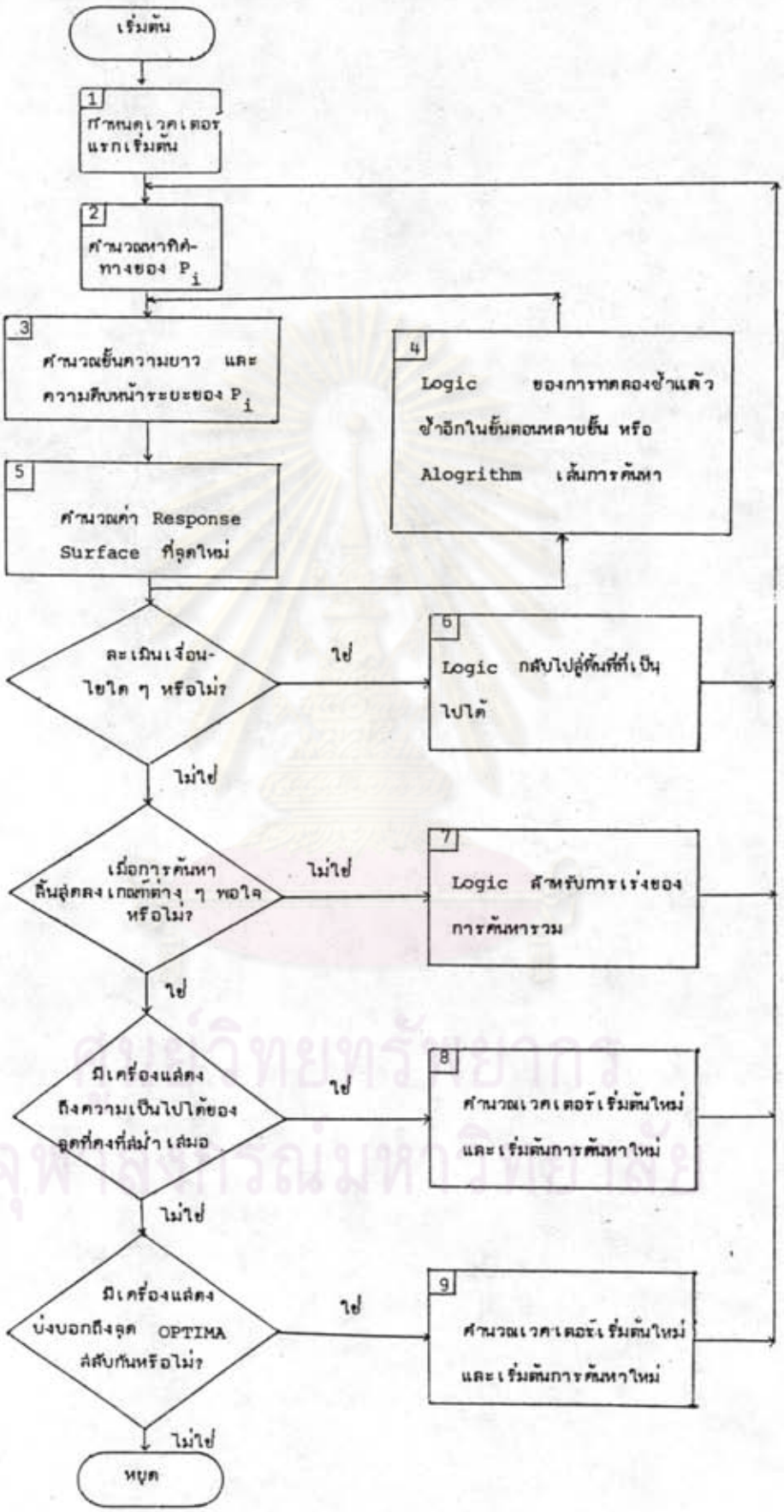
จากระบบโปรแกรมคอมพิวเตอร์ที่กล่าวมาแล้วนั้น จุดสำคัญที่ถือว่าเป็นหัวใจของการแก้ปัญหาทั้งหมดก็คือ การคำนวณ Search Routine ในอดีตหลายสิบปีที่ผ่านมา ได้มีนักวิจัยหลายต่อหลายคน ได้พัฒนาขบวนการดังกล่าวนี้ เรื่อยมาจนกระทั่ง Hooke , R. และ T.A. Jeeves ได้สามารถพัฒนาสำเร็จ เป็นที่ยอมรับของวงการทั่วไป งานที่เขาพัฒนาได้นั้นเป็น Heuristic Program ซึ่งในเนื้อหาของโปรแกรม จะไม่เน้นถึงส่วนละเอียดของการออกแบบ โดยที่ Search Routine ทั้งหมด ที่เขาสร้างแนวการพัฒนาจะตอบคำถามพื้นฐาน 2 คำถาม และสมมติให้เลือกจุดเฉพาะบน Response Surface ดังนั้น 2 คำถาม ก็คือ

- ทิศทางต่อไปของการเคลื่อนที่ของจุดเฉพาะเป็นอะไร
- ระยะมากน้อยแค่ไหน ที่จุดเฉพาะนั้นจะเคลื่อนที่ไปยังทิศทางนั้น

ในการเคลื่อนที่ของจุดเฉพาะบน Response Surface นั้น จัดว่าเป็นเวคเตอร์พื้นฐาน ซึ่งอาจจะเป็นไปในแนวแกนตั้งหรือแนวแกนนอน หรือทิศทางอื่น ๆ โดยสุ่มก็ได้ และในทิศทางที่ได้ถูกตัดสินใจเลือกเคลื่อนที่ไปนั้น อาจจะเป็นหนึ่งขั้นตอนหรือหลายขั้นตอน หรือกำหนดเส้นที่ค้นหาในเชิง 1 มิติมาก็ได้ กรณีใดกรณีหนึ่ง เราอาจพิจารณาได้ในสมการต่อไปนี้

$$D_{i+1} = D_i + \lambda P_i \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (64)$$

สมการข้างต้นนี้เป็นเทอมเชิงปริมาณที่สามารถตอบคำถาม 2 คำถามที่สำคัญได้ โดยที่ D_i เป็นเวคเตอร์ n มิติ ที่มี component คือ $(d_1, d_2, \ldots, d_n)$ แสดงถึงจุดทดลองสำหรับการ Trial ครั้งที่ i และ λ_i เป็นค่าคงที่บวก ส่วน P_i เป็นเวคเตอร์ที่จะตอบคำถามแรกที่ว่าทิศทางต่อไปของการเคลื่อนที่ของจุดเฉพาะเป็นอะไร ส่วนขนาดของ λ_i จะบ่งบอกคำตอบของคำถามที่สองที่ว่า ระยะมากน้อยแค่ไหนที่จุดเฉพาะนั้นจะเคลื่อนที่ไป



รูปที่ 3.3 หน่วยหลัก ๆ ของข้อสรุป Pattern Search Subroutine

จากรูปที่ 3.3 เป็นผังงานที่เป็นจุดหลัก ๆ ของ Pattern Search Subroutine หรือ Search Routine โดยเริ่มต้นจาก Box ที่ 1 ซึ่งเป็นการกำหนดเวคเตอร์เริ่มต้นก่อน ส่วน Box ที่ 2 3 4 5 และ 7 นั้น แต่ละ Box ประกอบไปด้วยส่วนที่เรียกว่า Search Code หรือ Search Algorithm ซึ่งส่วนประกอบเหล่านี้จะเป็นจุดที่ก่อให้เกิดการตัดสินจุดที่ตั้งของจุดเฉพาะต่อไป อีกทั้งยังประมาณค่า Response Surface ที่จุดดังกล่าวด้วย (ในโปรแกรม การทำงานขั้นตอนนี้อยู่ใน Objective Function Subroutine) และจะมีการตัดสินใจทิศทางที่ดีที่สุดเหมาะสมที่สุดของการเคลื่อนที่

ใน Box ที่ 6 จะเป็น Box ที่ประกอบไปด้วย Logicที่เหมาะสม ที่จะทำการตรวจสอบ ถ้า Search Code ได้บ่งบอกถึงการเคลื่อนที่ออกไปนอกเขตพื้นที่ที่เป็นไปได้กล่าวคือ ถ้าตรวจสอบได้ว่า การเคลื่อนที่ดังกล่าวออกนอกเขตพื้นที่อันเป็นการละเมิดเงื่อนไขที่ตั้งไว้มันจะถูกนำกลับไปสู่การเริ่มต้นที่ Box ที่ 2 ตามผังงานอีกครั้ง ในงานวิจัยนี้ตามโปรแกรมได้มีการแปลงโดยการบวกเข้าด้วยกันระหว่าง Penalty function กับ Original Objective Function แล้วทำการ Optimize ฟังก์ชั่นใหม่ที่ได้โดยที่เป็นฟังก์ชั่นที่ไม่มีเงื่อนไขใด ๆ บังคับ (ขบวนการนี้ให้ดูตัวแปร PNLT ซึ่งใช้ใน Objective Function Subroutine (FCTI) และดูในผนวกด้วย)

ใน Box ที่ 8 แสดงให้เห็นถึง Logic ที่นำมาใช้ในการตัดสินใจถ้า Routine นั้น ยึดติดบน สัดส่วนที่สัมพันธ์ของ Response Surface การทดสอบชนิดต่าง ๆ จะรวมอยู่โดยสุ่มหรือ จุดเช็คที่เป็นระบบ ที่หลาย ๆ Trial Point ในส่วนที่ใกล้เคียงของจุดคงที่ที่ได้รับการส่งสัย ถ้าตรวจแก้ไขถูกพบ Routine ก็จะเคลื่อนที่สู่การทดสอบสุดท้าย สำหรับการสลับที่ของ Optima ถ้าจุดที่ดีกว่าถูกค้นพบ การค้นหาใหม่ก็จะเริ่มต้น โดยการใช้ที่ตั้งของจุดใหม่เป็นจุดเริ่มต้น

ใน Box ที่ 9 มีไว้เพื่อทำการทดสอบการสลับกันของ Optima โดยการเริ่มต้นใหม่ของการค้นหาจากที่ตั้งที่แตกต่างกันบน Response Surface นับว่าเป็นสิ่งจำเป็น ตั้งแต่จุดไม่ทราบแน่ชัดของ Search Routine ก็สามารถยืนยันได้แน่นอนว่า จุดที่เหมาะสมที่สุด ทั่วทั้งหมดถูกค้นพบแล้ว ในโปรแกรมของงานวิจัยนี้ Box ที่ 9 เป็นที่รวมเข้าไว้ใน Main Program ที่หลังจากการค้นพบสิ้นสุดลงไป และพิมพ์ผลลัพธ์ของมันที่ได้ออกมา ขั้นต่อไปก็จะถามผู้ใช้ถึงความต้องการที่จะหาความแม่นยำที่ดีกว่าหรือไม่ โดยการใส่จุดเริ่มต้นใหม่ใน Box ที่ 1

3.2.2 การดัดแปลงทำให้ง่ายขึ้นของ Pattern Search

Pattern Search Subroutine ที่ใช้ในโปรแกรมคอมพิวเตอร์ได้ดัดแปลง และพัฒนาจากงานของ W.H. TAUBERT โดยได้ใช้หลักบางส่วนจาก Dr. B.Khoshnevis ผู้ซึ่งมีประสบการณ์ที่เพียบพร้อมในงานด้าน Search Routine และเป็นที่น่าสังเกตว่าการดำเนินของขบวนการ Search Routineนี้ขึ้นอยู่กับ Particular Response Surface อย่างไรก็ตาม Buffa, Elwood S. และ Jeffery G. Miller ได้พัฒนาดัดแปลง งาน Pattern Search ของ W.H. Taubert เล็กน้อย โดยใช้หลักการที่คงเส้นคงวา บนความหลากหลายของ Response Surface

อนึ่ง ในปี ค.ศ. 1965 Weisman, J., C.F. Wood และ L. Rivlin ได้ดัดแปลงและพัฒนา Pattern Search Algorithm โดยการใช้แนวคิดจาก Hooke , R. และ T.A. Jeevesซึ่ง นับว่ามีประโยชน์มาก และในงานวิจัยนี้ก็ได้นำแนวคิดนี้มาร่วมในการสร้างโปรแกรมคอมพิวเตอร์ชุดนี้

สำหรับ Pattern Search เป็นขบวนการที่ค้นหาโดยตรง ซึ่งโดยทั่วไปวิธีการค้นหาโดยตรงนั้นเป็น Heuristic โดยธรรมชาติ และไม่อาจเห็นภาพของ Response Surface ในเทอมของตัวแบบการวิเคราะห์เฉพาะแบบ และไม่ต้องการการประมาณทางจำนวนของอนุพันธ์ ใน Pattern Search นั้น อนุกรมของ Exploratory Move จะเป็นเครื่องบ่งชี้การรู้ถึงของ Objective Function การรู้ถึง ดังกล่าวนั้นเป็นประโยชน์ สู่การตัดสินใจในทิศทางที่น่าจะเป็นไปได้ สำหรับการเคลื่อนที่ที่ประสบผลสำเร็จ ซึ่งในที่นี้จะเรียก "Pattern Move" ซึ่งในแต่ละ Pattern Move นั้นจะเป็นสำดับตามของ Exploratory Move ซึ่งเป็นการแก้ไขทิศทางของ Pattern Move ให้ถูกต้องยิ่งขึ้น ขบวนการนี้จะทำต่อเนื่องไปเรื่อย จนกระทั่ง ค่าของ Objective Function ไม่สามารถทำให้ลดลงไปได้อีก

ข้อแตกต่างระหว่าง Algorithm ของ Weisman และคณะกับ Pattern Search ก็คือ แบบแผนในขนาดของขั้นตอนที่ถูกควบคุมอยู่ ใน Algorithm ของ Weisman และคณะ นั้นมีเป็นระบบสำหรับการควบคุม ขนาดของขั้นตอนของแต่ละตัวแปร ซึ่งประเด็นนี้แตกต่างอย่างสิ้นเชิงกับ Algorithm ของ Hooke ., R. และ T.A. Jeeves ที่ขนาดของขั้นตอน สำหรับตัวแปรอิสระทั้งหมด ถูกเปลี่ยนแปลงที่เวลา

เดียวกัน และโดยอัตราส่วนเดียวกัน แสดงให้เห็นว่า Algorithm ของ Weisman และคณะนั้นมีเป็นระบบ สำหรับการควบคุมขนาดของขั้นตอนของแต่ละตัวแปร ซึ่งประเด็นนี้แตกต่างอย่างสิ้นเชิงกับ Algorithm ของ Hooke , R. และ T.A. Jeeves ที่ขนาดของขั้นตอน สำหรับตัวแปรอิสระทั้งหมด ถูกเปลี่ยนแปลงที่เวลาเดียวกันและโดยอัตราส่วนเดียวกัน แสดงให้เห็นว่า Algorithm ของ Weisman และคณะ เสริมในการค้นหาจะมีประสิทธิภาพ และมีแบบแผนที่เหนือกว่า Algorithmของ Hooke , R. และ T.A. Jeeves

อนึ่ง Taubert ได้พบว่า หลักปฏิบัติแบบแผน Pattern Search Algorithm แนวของ Weisman และคณะนั้นสามารถพิสูจน์ได้โดยการ เปลี่ยนแปลงอย่างมีระบบที่เรียกว่า "Pattern Growth Multiplier" ซึ่งตัวคูณนี้ถูกใช้สำหรับควบคุมคุณภาพความยาวของ Pattern Move ซึ่งสามารถนิยามได้โดยสมการ คือ

$$D_{i+1}(J) = D_i(J) + \lambda_i \left[C_i(I) - D_i(J)\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (65)$$

โดย i = การ Trial ครั้งที่ i

D_{i+1} = จุดฟื้นฐานใหม่ชั่วคราว

D_i = จุดฟื้นฐานเดิมแสดงโดย Exploratory Search

λ = Pattern Growth Multiplier

C_i = ที่ตั้งการค้นหาที่จุดปลายของ Exploratory Search

J = component ที่ j ของแต่ละเวคเตอร์

จากสมการข้างต้นนี้ สามารถเห็นได้ชัดว่า เมื่อค่าของ λ เพิ่มขึ้น Algorithm สามารถทำให้ Pattern Move มีขนาดใหญ่ขึ้น ในขณะเดียวกัน ถ้าค่าที่ลดลงจะมีผลให้ระยะการเคลื่อนที่ก็จะลดลงด้วย การควบคุมของ Logic ที่พัฒนา โดย Taubert ที่เปลี่ยนค่า λ เพื่อที่ว่าทำเลจุดตั้งของ Response Surface และด้วยสาเหตุนี้ทำให้ Algorithm ของ Weisman และคณะ มีประสิทธิภาพ และสามารถพิจารณาได้โดยง่าย ที่ค่าคงที่ของ λ = 2.0 ใน Algorithm

ในโปรแกรมคอมพิวเตอร์ ภาษาเบสิคได้ถูกแสดงไว้โดยละเอียดในภาคผนวก พร้อมรายละเอียดตัวแปร สามารถเข้าใจขบวนการระบบต่าง ๆ ที่กล่าวมาแล้วข้างต้นนี้

3.2.3 The Hooke- Jeeves Pattern Search

วิธีการค้นหาจุดอุดมะ (Optimization Technique) นั้น มีเทคนิคการค้นหามากมาย และ วิธีของ Hooke -Jeeves ก็เป็นวิธีการค้นหาค่าพารามิเตอร์ที่เหมาะสมหรือหาค่าของตัวแปรวิธีการหนึ่งซึ่งเป็นวิธีที่คิดขึ้นในการแก้ปัญหาการค้นหาคำตอบของฟังก์ชั่นหรือสมการที่หาคำตอบไม่ได้โดยตรงจากการหาอนุพันธ์ (Derivative) เพราะฟังก์ชั่น หรือสมการบางอย่างมีรูปแบบที่ซับซ้อนและยุ่งยาก เช่น ฟังก์ชั่น ไคสแควร์ และฟังก์ชั่นความหนาแน่นของการแจกแจงแบบปกติ เป็นต้น

นิยามการค้นหาโดยตรง (Definition of Direct Search) ให้ Σ แทนสเปซของจุด (Space of Points) และ P, Q, R เป็นจุดใด ๆ ใน Σ ($\varepsilon \Sigma$) ถ้าจุด P ดีกว่า จุด Q ใช้สัญญลักษณ์ว่า "$P \subset Q$" ซึ่งการเปรียบเทียบความสัมพันธ์ $\subset$ บนปริภูมิของจุด Σ นั้นจะต้องสอดคล้องกับความสัมพันธ์แบบถ่ายทอด (Transitive Relation) กล่าวคือ $P \subset Q$, $Q \subset R \longrightarrow P \subset R$, มีจุด P^* เป็นจุดที่ ที่ใกล้ที่สุด (Extremal Point) ของปริภูมิของจุด Σ ซึ่งมีคุณสมบัติว่า $P^* \subset P$ จุด P^* นี้แสดงคำตอบของปัญหา ส่วนจุดอื่น ๆ ของปริภูมิของจุด Σ จะแสดงคำตอบที่เป็นไปได้ (Possible Solution)

วิธีการค้นหาโดยตรงจะทำการเปรียบเทียบแบบอันดับ (Squential Comparisions) โดยใช้ความสัมพันธ์ ซึ่งกำหนดดังนี้

3.2.3.1 กำหนดจุดทดลอง (Trial Points) P_r ($r = 1,2,\ldots,N$, $N =$ จำนวนรอบทั้งหมด)

3.2.3.2 ให้ B_o เป็นจุดพื้นฐานเริ่มต้น และ B_r เป็นจุดพื้นฐานใด ๆ ที่ $B_r = P_s$ สำหรับบางค่า $s < r$

3.2.3.3 มีเซตของจำนวนเต็มที่เรียกว่า สภาวะการณ์ (States) ของขบวนการรวมทั้ง สภาวะการณ์เริ่มต้น S_o สภาวะการณ์ S_r จะสอดคล้องกับจุดพื้นฐาน B_r

3.2.3.4 กฎของการหยุด (Stop Rule) สำหรับการสิ้นสุดลงของขบวนการกำหนดให้เป็น N

3.2.3.5 การประมาณคำตอบ (Approximate Solution) หรือประมาณ P^* คือ จุดพื้นฐาน B_N

จุดสำคัญสำหรับการกำหนดขบวนการนี้ถูกนิยามโดยกฎต่อไปนี้

1. จุดพื้นฐานเริ่มต้น B_o และสภาวะการณ์เริ่มต้น $S_o (\neq 0)$ เป็นจุดพื้นฐาน และสภาวะการณ์ใด ๆ

2. จุดทดลองที่เหลือคือ $P_r = h(B_{r-1}, S_{r-i})$ เมื่อ h เป็นฟังก์ชั่น บน Σ และ $S_{r-1} \neq 0$

3. ถ้า $P_r \subset B_{r-1}$ แล้ว $B_r = P_r$ และ $S_r = f(S_{r-1})$ นอกเหนือจากนั้น $B_r = B_{r-1}$ และ $S_r = g(S_{r-1})$

4. เมื่อครั้งแรก สำหรับ $S_i = 0$ ขบวนการจะหยุด เช่น $i = N$

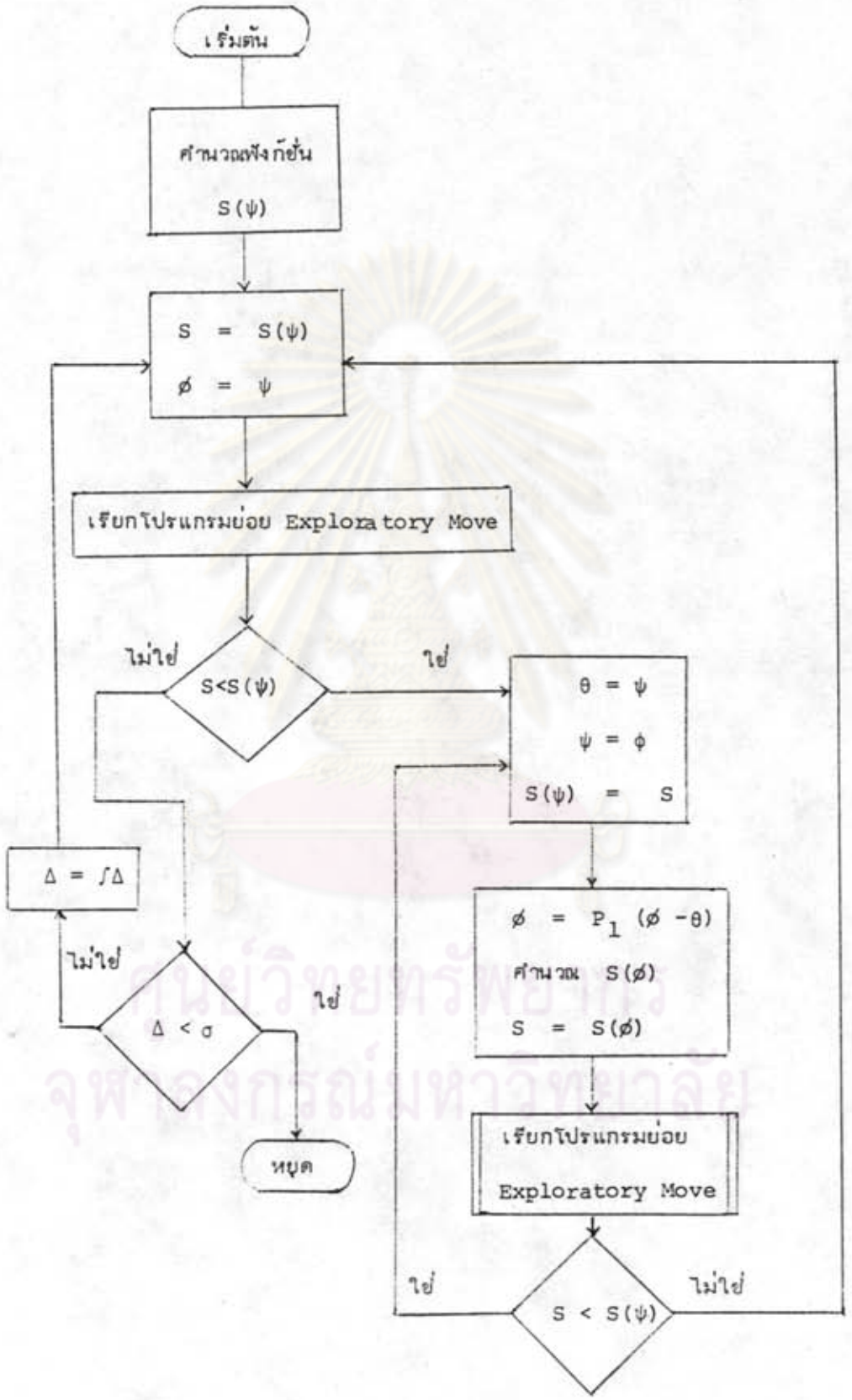
จากนิยาม ดังกล่าว วิธีการ Hooke - Jeeves Pattern Search จะอาศัยนิยามนี้เป็นพื้นฐานในการสร้างขบวนการ การค้นหาค่าของตัวแปรวิธีการนี้มี 2 ขั้นตอนคือ

ขั้นตอนที่ 1 เรียกว่า Exploratory Move ขั้นตอนนี้เราจะกำหนดจุดเริ่มต้น (B_o) หรือการคาดเดาจุดเริ่มต้น (Intitial Guesses) และกำหนดขนาดขั้นเริ่มต้น (Intitial Step Size) คือ a_oของตัวแปรทุกตัวในส่วนการหรือฟังก์ชั่น คำนวณค่าฟังก์ชั่นที่จุดเริ่มต้น (S_o) และที่จุดเริ่มต้นบวกกับขนาดขั้นเริ่มต้น (S_i) การคำนวณค่าฟังก์ชั่นที่จุดทั้งสองนี้จะเป็นลักษณะการเคลื่อนย้ายจุดใด ๆ ในสเปซของจุด (Σ) ไปยังอีกจุดหนึ่ง เรียกว่า Move (จากจุด $B_o \rightarrow B_o + a_o$) การเคลื่อนย้ายนั้นจะเคลื่อนย้ายทีละค่าของตัวแปรถ้าฟังก์ชั่น หรือส่วนการนั้นมีตัวแปรหลาย ๆ ตัว ถ้าค่า S_i "น้อยกว่า" S_o เรียกจุด $B_1 (B_o + a_o)$ ว่าจุดที่สำเร็จ (Success) ถ้าค่า S_i "มากกว่า" S_o เรียกจุดว่าจุดล้มเหลว (Failure) เมื่อสำเร็จ (Success) แล้วก็จะคูณค่ายนาดขั้นเริ่มต้นด้วยค่าปัจจัยที่เพิ่มขึ้น (Step Accelation Factor) คำนวณค่าฟังก์ชั่นที่จุดเริ่มต้นบวกกับขนาดขั้นเริ่มต้นใหม่ (S_2) ทำการเปรียบเทียบกับค่าฟังก์ชั่น S_1 ใหม่ ถ้าเกิดล้มเหลว

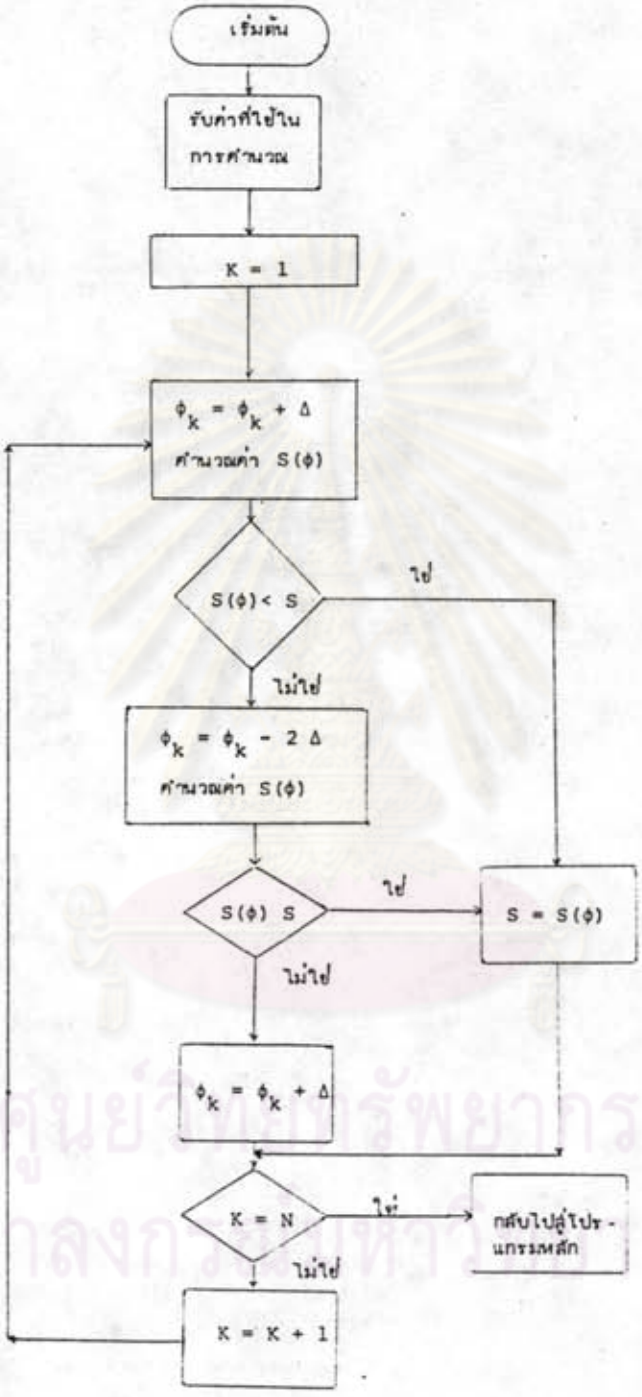
(Failure) ก็จะคูณค่าขนาดเริ่มต้นด้วยค่าปัจจัยที่ลดลง (Step Decelation Factor) แล้วคำนวณค่าฟังก์ชั่นที่จุดเริ่มต้นบวกกับขนาดขั้นเริ่มต้นอีก (S_3) เปรียบเทียบค่า S_3 กับ S_1 อีก ทำเช่นนี้ไปเรื่อย ๆ จนครบทุก ๆ ค่าของตัวแปร (กรณีที่ตัวแปรมีหลายตัว)

ขั้นตอนที่ 2 เรียกว่า Pattern Move เป็นขั้นตอนที่ต่อจาก Exploratory Move คือ มีการกำหนดจุดเริ่มต้นใหม่ โดยใช้จุดสุดท้ายในขั้นตอนของ Exploratory Move ที่สำเร็จ (Success) แล้วมีการเคลื่อนย้ายจุดที่เริ่มต้นนั้นไปยังจุด ๆ หนึ่ง (B_r) โดยที่จุด $B_r = Pl \ X(B_{r-1} - B_{r-2})$ เมื่อ Pl เรียกว่า Pattern Step Factor B_{r-1} และ B_{r-2} เป็นจุดเริ่มต้นในรอบที่ $r - 1$ และ $r - 2$ ที่สำเร็จ (Success) จากนั้นจะใช้จุด B_r นี้เป็นจุดเริ่มต้นในการค้นหาต่อไปโดยใช้ขั้นตอนที่ 1

ตามวิธีการ Hooke-Jeeves Pattern Search จะหยุดค้นหาค่าของตัวแปรที่ต่อเมื่อค่าขนาดขั้นเริ่มต้น (Intial Step Size) ที่คำนวณได้ในขณะนั้น น้อยกว่าค่าที่เราจะยอมรับได้ (Convergence Tolerance Value) ดังผังงาน (Flow Chart) ต่อไปนี้



รูปที่ 3.4 ผังงานของการหาจุดที่ต่ำสุดของฟังก์ชัน โดยวิธี Hooke-Jeeves Pattern Search



รูปที่ 3.5 ผังงานโปรแกรมย่อยของ Exploratory Move

หมายเหตุ θ = จุดพื้นฐานก่อนการเปลี่ยนแปลง

ψ = จุดพื้นฐานในขณะนั้น

ϕ = จุดพื้นฐานที่เป็นผลมาจากการ Move

$S(\psi)$ = ค่าฟังก์ชั่นที่จุดพื้นฐาน ψ

$S(\phi)$ = ค่าฟังก์ชั่นที่ได้จากการ Move

S = ค่าฟังก์ชั่นก่อนการ Move

Δ = ขนาดขั้นเริ่มต้น (Initial Step Size)

δ = ค่าที่เราจะยอมรับได้ (Convergence Tolerance Value)

ρ = ปัจจัยที่ลดลง (Step Decelation Vactory), $\rho < 1$

ψ_k = ค่า Coordinate สำหรับ ψ , $k = 1,2,\ldots, N$

N = จำนวน Coordinate สำหรับจุดนั้น ๆ

P_1 = Pattern Step Factor

3.2.4 Search Routine Parameters

หลังจาก Pattern Search Subroutine ถูกเรียกโดยโปรแกรมหลัก (Main Program) Response Surface ของ Objective Function จะถูกตรวจสอบจนกระทั่งต่างก็ถึงจุดจำกัดบนจำนวนของจุดแตกต่างที่ถูกตรวจสอบ จะถึงที่หมายหรือจุดที่ดีกว่าจะถูกตรวจพบดังนั้น จึงมี 2 กฎเกณฑ์สำหรับการหยุดการค้นหาโดยให้ 2 ตัวแปรคือ LIM และ TOL ถูกใช้ในการกำหนดกฎเกณฑ์คือ

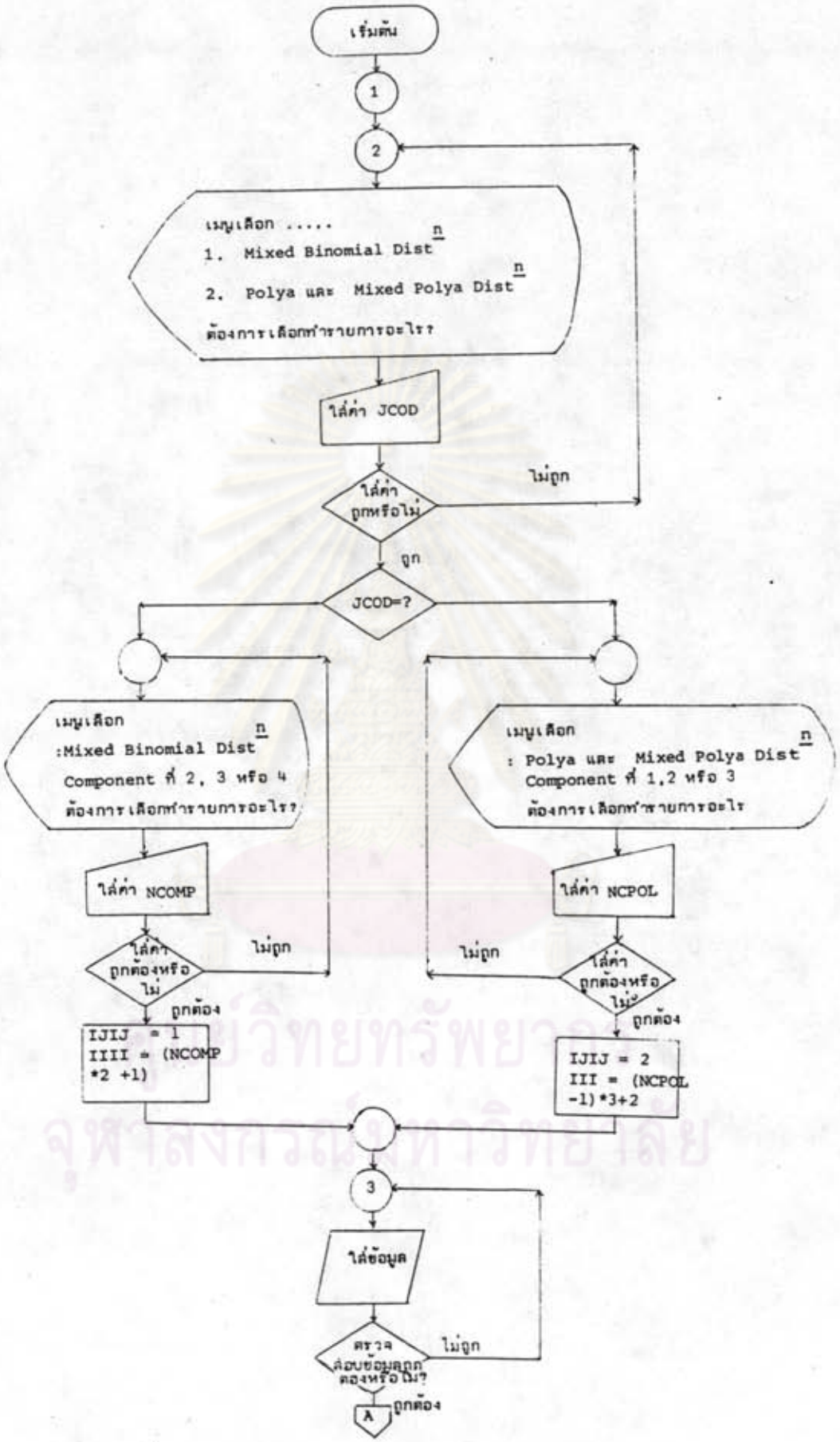
LIM : LIM เป็นค่าประมาณของขอบเขตบน บนจำนวนที่เรียกใน Objective Function ในระหว่างที่การค้นหารายละเอียดของ Pattern Search Subroutine มันเป็นค่าพิเศษที่นับว่าเป็นจำนวนที่สูงของจุดบน Response Function ที่ถูกทดสอบจนกระทั่งค่าต่ำสุดของ Response Function จะถึง

โดยปกติ ค่า LIM นี้จะตั้งไว้ในโปรแกรมคอมพิวเตอร์ระหว่าง 800 - 3000 เนื่องจากประสบการณ์ที่ผ่านมา พบว่าค่า 3000 สำหรับ LIM เป็นค่าที่ดีที่สุด เพราะจะให้ผลลัพธ์ของการประมาณค่าพารามิเตอร์ที่ดีในทางปฏิบัติ

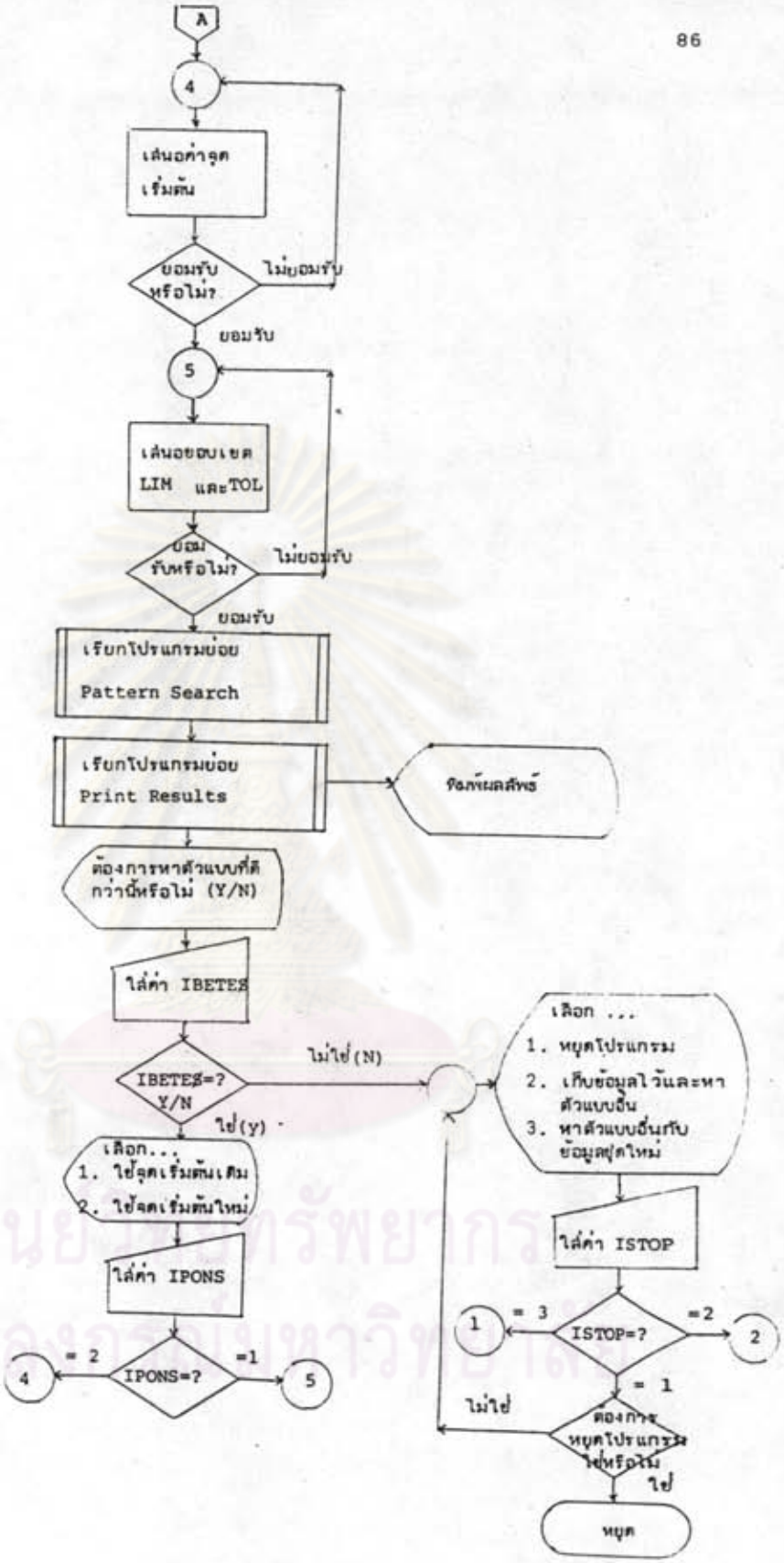
TOL : TOL เป็นปริมาณที่ใช้ในการให้ได้มาซึ่งขอบเขตต่ำสุดบนขนาดขั้นสำหรับ Exploratory Move พบว่าค่าของ 0.001 สำหรับ TOL จะให้ผลเป็นที่น่าพอใจ และถ้าใช้ค่าที่เล็กกว่านี้ เช่น 1.0×10^{-5} หรือน้อยกว่า จะไม่สามารถให้ผลลัพธ์ที่ดีขึ้นอย่างมีนัยสำคัญ

3.3 ผังงานของระบบการคำนวณ

ผังงานที่จะแสดงต่อไปนี้เป็นผังงานที่แสดงระบบการคำนวณตามโปรแกรมที่โต้ตอบร่วมดังนี้



รูปที่ 3.6 ผังงานของระบบการคำนวณโปรแกรมคอมพิวเตอร์



รูปที่ 3.6 ผังงานของระบบการคำนวณโปรแกรมคอมพิวเตอร์ (ต่อ)

จากบทนี้ เป็นส่วนที่เกี่ยวกับระบบโปรแกรมคอมพิวเตอร์ทั้งหมดในงานวิจัยนี้ ส่วนรายละเอียดตัวแปรต่าง ๆ ในโปรแกรมจะกล่าวในภาคผนวกอีกครั้งหนึ่ง