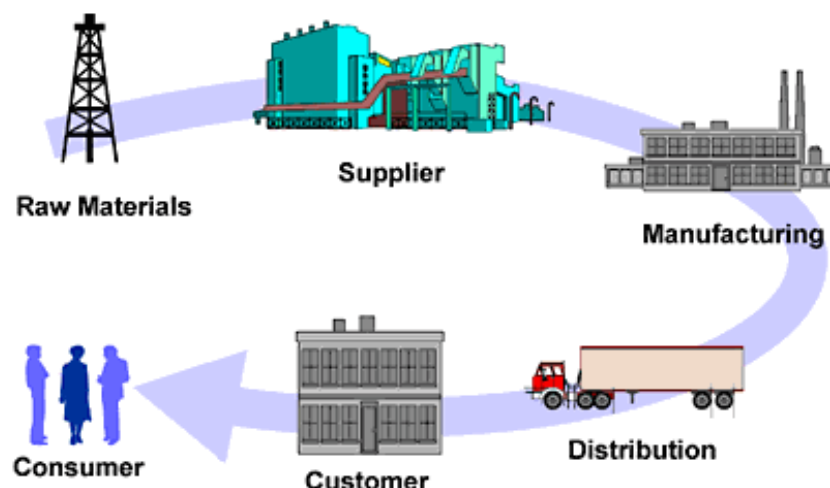


## บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการวิจัยเรื่อง “การปรับปรุงกระบวนการทางธุรกิจด้านการวางแผนผลิตสำหรับอุตสาหกรรมการผลิตยา” ผู้วิจัยได้ศึกษาค้นคว้าเอกสาร ทฤษฎี และทบทวนวรรณกรรมที่เกี่ยวข้องเพื่อนำมาหาวิธีในการปรับปรุงประสิทธิภาพด้านการวางแผนผลิต พร้อมทั้งศึกษาเครื่องมือสนับสนุนการวิจัยในการวิเคราะห์ปัญหาตามกรอบของการจัดการโซ่อุปทาน และปรับปรุงกระบวนการตามแนวทางของวิศวกรรมอุตสาหกรรม โดยได้ประยุกต์นำเทคนิคต่างๆเข้ามาใช้เพื่อพัฒนาประสิทธิภาพการดำเนินงาน ซึ่งเทคนิคที่ว่่านั้นประกอบด้วย การจัดการโซ่อุปทาน (Supply Chain Management) แผนภาพกระบวนการทางธุรกิจ (Business Process Mapping) การเขียนแผนภาพกระบวนการทางธุรกิจ IDEF0 (Integration Definition for Function Modeling) แผนการผลิตรวม (Aggregate Planning) การจัดการตารางการผลิตหลัก (Master Product Schedule) เป็นต้น โดยบทนี้เนื้อหาประกอบด้วยทฤษฎีและงานวิจัยที่เกี่ยวข้อง

### 2.1 การจัดการโซ่อุปทาน (Supply Chain Management)

โซ่อุปทาน (Supply Chain) คือ กระบวนการของวงจรชีวิตผลิตภัณฑ์ซึ่งสนับสนุนการไหลทางกายภาพ สารสนเทศ การเงิน และความรู้ การเคลื่อนย้ายขนถ่ายผลิตภัณฑ์และการบริการจากผู้ส่งมอบวัตถุดิบไปถึงผู้บริโภคขั้นสุดท้าย ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 โซ่อุปทาน

ประจวบ กล่อมจิตร [1] กล่าวเกี่ยวกับโซ่อุปทานว่าประกอบด้วยจุดสำคัญซึ่งแสดงให้เห็นถึงความสัมพันธ์ของการไหลของวัตถุดิบตั้งแต่ต้นทางจนถึงปลายทาง

- ผู้ส่งมอบ (Suppliers) หมายถึง ผู้ที่ส่งวัตถุดิบให้กับโรงงานหรือสถานปฏิบัติการสำหรับกระจายสินค้า
- โรงงานผู้ผลิต (Manufacturers) หมายถึง ผู้ที่ทำหน้าที่ในการแปรรูปวัตถุดิบที่ได้รับจากผู้ส่งมอบ ให้มีคุณค่าสูงขึ้น
- ศูนย์กระจายสินค้า (Distribution Centers) หมายถึง จุดที่ทำหน้าที่ในการกระจายสินค้าไปให้ถึงมือผู้บริโภคหรือลูกค้า ซึ่งศูนย์กระจายสินค้าหนึ่งๆอาจมีสินค้าที่มาจากหลายโรงงานผลิต
- ร้านค้าย่อยและลูกค้าหรือผู้บริโภค (Retailers or Customers) คือ จุดปลายทางของการจัดการโซ่อุปทาน

การจัดการโซ่อุปทาน (Supply Chain Management; SCM) คือ การออกแบบ การวางแผนปฏิบัติการควบคุมติดตามกิจกรรมในโซ่อุปทาน โดยมีวัตถุประสงค์ในการสร้างคุณค่าของการแข่งขัน การยกระดับงานสากล และการปรับปรุงอุปทาน (Supply) ให้สอดคล้องกับอุปสงค์ (Demand) และการวัดการปฏิบัติงาน จัดเป็นกิจกรรมที่เกี่ยวข้องกับการบูรณาการ (Integration) และการจัดการในองค์กร โดยได้มีการนำโซ่อุปทาน กิจกรรมต่างๆในโซ่อุปทาน และเกี่ยวข้องกับกิจกรรมความสัมพันธ์และความร่วมมือในโซ่อุปทาน ซึ่งมีผลกระทบต่อกระบวนการทางธุรกรรมที่จะสร้างเสริมให้มีมูลค่าเพิ่มในสินค้าและบริการ นำมาซึ่งความสามารถในการแข่งขันได้อย่างยั่งยืน ปฏิสัมพันธ์ของการจัดการกระบวนการต่างๆที่เกี่ยวข้องกับอุปทานของสินค้าและบริการ โดยการปฏิสัมพันธ์จะมีลักษณะเชิงบูรณาการ โดยมีเป้าหมายในการที่จะสร้างมูลค่าเพิ่ม และสนองตอบต่อความต้องการของตลาด การผลิต การกระจาย และการส่งมอบสินค้าและยังรวมถึงการสื่อสารสนเทศของข้อมูลและข่าวสาร โดยมีวัตถุประสงค์ที่จะลดต้นทุนรวมของธุรกิจและเพิ่มศักยภาพของการแข่งขัน จะเห็นได้ว่าการจัดการโซ่อุปทานนั้นเกี่ยวข้องกับกระบวนการต่างๆที่เกี่ยวข้องกับความร่วมมือทางธุรกิจ ตั้งแต่แหล่งของวัตถุดิบต้นน้ำ (Upstream Source) จนถึงการส่งมอบสินค้าและบริการปลายน้ำ (Downstream Customers) ซึ่งกระบวนการเหล่านี้ครอบคลุมถึงกระบวนการที่เกี่ยวข้องกับการให้ได้ว่าซึ่งวัตถุดิบกระบวนการส่งเสริมกิจกรรมทางการตลาดและการผลิต รวมถึงกระบวนการเคลื่อนย้ายสินค้าจนถึงมือผู้ต้องการสินค้า ทั้งนี้ กระบวนการต่างๆ จะมีปฏิสัมพันธ์ในลักษณะของบูรณาการ ทั้งนี้ การกิจสำคัญของการจัดการโซ่อุปทานนั้น คือการมุ่งให้ลูกค้าเกิดความพอใจสูงสุด โดยเน้นในเรื่องประสิทธิภาพเชิงต้นทุน และผลตอบแทนทางธุรกิจ โดย Council of Supply Chain Management Professional (2006) สถาบันชื่อดังที่ได้ถูกอ้างอิงจากหนังสือหลายๆเล่ม ได้ให้ความหมายของการจัดการโซ่อุปทาน (Supply Chain Management) ไว้ว่า การจัดการโซ่อุปทานเป็นการรวมกันของการวางแผนและการจัดการในทุกๆ กิจกรรมซึ่งมีส่วนเกี่ยวข้องในการจัดซื้อจัดหา กระบวนการเปลี่ยนแปลงต่างๆ การจัดการโลจิสติกส์ และยังรวมไปถึงการประสานและร่วมมือกัน

ระหว่างสมาชิกในโซ่อุปทาน ซึ่งประกอบไปด้วยซัพพลายเออร์ ลูกค้า หรือผู้ให้บริการลำดับต่างๆ สารสำคัญคือ การจัดการโซ่อุปทานเป็นการจัดการในเรื่องของการจัดหาและความต้องการภายใต้ความสัมพันธ์ระหว่างบริษัทต่างๆ โดยความหมายของการจัดการโซ่อุปทานโดยรวมจะเป็นการจัดการโดยเน้นความสัมพันธ์ของสมาชิกในโซ่อุปทาน ซึ่งมีเป้าหมายเพื่อลดความไม่แน่นอนและความเสี่ยงโดยให้มีระดับสินค้าคงเหลือที่มีผลในทางบวก มีการบริหารวงจร (Cycle time) การบริหารกระบวนการ (Process) และการบริการที่เหมาะสมที่สุดสำหรับลูกค้าหรือผู้บริโภคขั้นสุดท้าย (End Customer Service) ซึ่งการจัดการโซ่อุปทานนั้นมีความสำคัญต่อองค์กรต่างๆ ซึ่งสาเหตุที่ต้องมีการศึกษาและประยุกต์ใช้หลักการของการจัดการโซ่อุปทานเข้าไปไว้ในหน่วยงาน เนื่องจากองค์กรใดที่มีการดำเนินงานหรือเชื่อมโยงติดต่อกับหน่วยงานที่ทำธุรกิจกับลูกค้า ซึ่งจะต้องมีปัญหาเกิดขึ้นไม่มากนักน้อยดังนั้นการจัดการโซ่อุปทานจึงมุ่งเน้นในการแก้ปัญหาในด้านต่างๆ ประสิทธิภาพของระบบโซ่อุปทานจะช่วยสร้างคุณค่าให้แก่ผลิตภัณฑ์ [1, 2]

## 2.2 การจัดทำแผนภาพกระบวนการทางธุรกิจ (Business Process Mapping)

เดือนใจ สมบูรณ์วิวัฒน์ [2] อธิบายถึงกระบวนการทางธุรกิจ เป็นการบ่งบอกให้ทราบถึงสภาพในปัจจุบันของการไหลของข้อมูลและวัตถุดิบระหว่างองค์กร การวิเคราะห์กระบวนการทางธุรกิจสามารถทำได้โดยการทำผังกระบวนการทำงานปัจจุบัน ดังนั้นการศึกษาถึงโซ่อุปทานใดๆ จึงควรจัดทำผังกระบวนการธุรกิจขององค์กรเพื่อให้เกิดความเข้าใจและมองเห็นภาพโครงสร้างขององค์กร และการไหลของข้อมูลและวัตถุดิบระหว่างองค์กรและภายในองค์กรของอุตสาหกรรมนั้น การวิเคราะห์กระบวนการธุรกิจจะเป็นการปรับระบบเพื่อให้เกิดผลลัพธ์ที่ต้องการ โดยมีมาตรวัดผลการปฏิบัติงานที่สำคัญและทันสมัย ซึ่งการวิเคราะห์กระบวนการธุรกิจนั้นแบ่งเป็น 4 ระยะ คือ

- 1) กำหนดขอบเขตของโครงการ (Project Definition) เป็นระยะที่วิเคราะห์องค์กรที่ศึกษาและประเด็นที่สำคัญที่ต้องการศึกษา
- 2) วิเคราะห์กระบวนการในปัจจุบัน (As-is Process Description /Analysis) คือ การวิเคราะห์รูปแบบกระบวนการธุรกิจที่เป็นอยู่ในปัจจุบัน วิเคราะห์หาสาเหตุของปัญหาต่าง ๆ ที่เกิดขึ้นในแต่ละกระบวนการทำงาน
- 3) ออกแบบกระบวนการที่ต้องการให้เกิดขึ้น (To-be Process Design / Validation) ภายหลังจากที่วิเคราะห์กระบวนการและกิจกรรมที่เป็นอยู่ (As-is) ในปัจจุบันโดยสรุปถึงปัญหาที่เกิดขึ้นในแต่ละกิจกรรมแล้วในขั้นตอนนี้จะนำเสนอแนวทางที่ต้องการจะปรับปรุงและออกแบบกิจกรรมที่ควรจะเป็น (To-be) โดยมุ่งเน้นในประเด็นการเชื่อมโยงของต้นทุนในแต่ละกระบวนการ เวลา ความสามารถในการผลิต คุณภาพ ตลอดจนนำเสนอถึงสิ่งที่ระบบต้องการเพิ่มเติมเพื่อให้การทำงานมีประสิทธิภาพมากยิ่งขึ้น

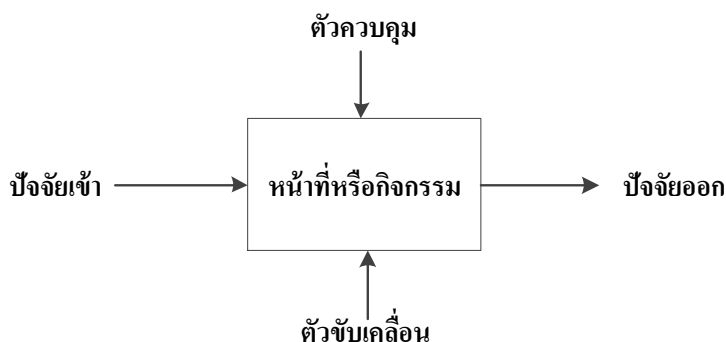
- 4) การวางแผนนำไปประยุกต์ใช้ในธุรกิจ (Implementation Planning) โดยหลังจากได้ปรับปรุงและออกแบบกระบวนการธุรกิจแล้ว ควรมีการวางแผนและหาวิธีที่จะนำกระบวนการที่ออกแบบไว้ไปประยุกต์ใช้ให้เกิดประโยชน์มากที่สุด

ตัวอย่างเครื่องมือที่ใช้ในการจัดทำผังกระบวนการธุรกิจในโซ่อุปทานที่นิยมใช้ในปัจจุบัน เช่น

- ศึกษาการกระทำ กิจกรรมของกระบวนการ การไหลของข้อมูลสารสนเทศและวัสดุ โดยใช้เครื่องมือผัง IDEF0 (Integration Definition for Function Modeling)
- ศึกษากระบวนการไหลกิจกรรมตามหน้าที่ โดยใช้เครื่องมือผัง Functional Flow Chart
- ศึกษากระบวนการทางธุรกิจ กิจกรรมการไหลตามหน้าที่และเวลา โดยใช้เครื่องมือผัง Swim-Lane Diagram

### 2.3 การเขียนแผนภาพกระบวนการทางธุรกิจ IDEF0 (Integration Definition for Function Modeling)

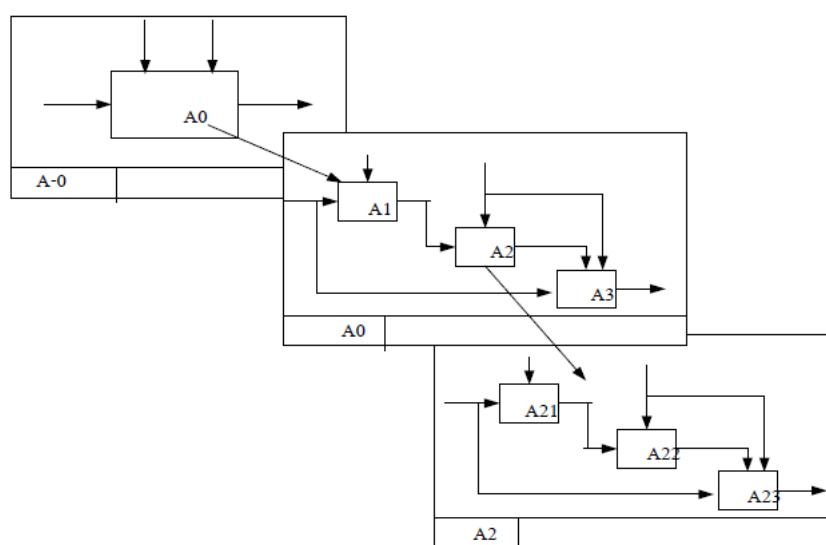
เดือนใจ สมบูรณ์วิวัฒน์ [2] ให้นิยาม IDEF0 (Integration Definition for Function Modeling) คือ เครื่องมือที่ใช้ในการวาดแผนภาพกระบวนการทางธุรกิจเพื่อนำไปสู่การวิเคราะห์และปรับปรุงกระบวนการทางธุรกิจ ซึ่งมุ่งเน้นการบ่งชี้กระบวนการ แสดงถึงรูปร่าง การจัดการธุรกิจการปรับปรุงกระบวนการและระบบที่ซับซ้อน ซึ่ง IDEF0 ใช้สำหรับแสดงกระบวนการธุรกิจและการไหลของข้อมูล เป็นวิธีการที่ใช้ในการจำลองการตัดสินใจ การกระทำ และกิจกรรมขององค์กร แสดงและวิเคราะห์การติดต่อสื่อสารประสานงานจะอยู่ในรูปแบบของกิจกรรมทางธุรกิจ (Activity Modeling) ซึ่งจะบ่งชี้การดำเนินการในแต่ละกิจกรรมที่แสดงให้เห็นถึงความสัมพันธ์ของการดำเนินงานในแต่ละกิจกรรมที่มี ทรัพยากรที่ใช้ และปัจจัยที่เกี่ยวข้องซึ่งกำกับในแต่ละกระบวนการของห่วงโซ่อุปทาน ซึ่งองค์ประกอบในการเขียน IDEF0 แสดงดังรูปที่ 2.2



รูปที่ 2.2 องค์ประกอบของ IDEF0

- 1) หน้าที่หรือกิจกรรม (Activity) คือ สิ่งที่แสดงหน้าที่หรือการกระทำในแต่ละกิจกรรมหรืออาจจะแสดงกระบวนการทั้งหมดด้วยชื่อ ซึ่งกิจกรรมหนึ่งสามารถแบ่งย่อยออกเป็นกิจกรรมอื่น ๆ ได้โดยที่จะมีตัวเลขแสดงลำดับของกิจกรรมไว้ที่มุมขวาล่างของกรอบสี่เหลี่ยม
- 2) ปัจจัยนำเข้า (Input) คือ ส่วนที่แสดงทิศทางการไหลของวัตถุดิบและข้อมูลที่ต้องการใช้เพื่อทำกิจกรรม เช่น วัตถุดิบ คำสั่งซื้อ เป็นต้น
- 3) ตัวควบคุม (Control) คือ ตัวควบคุมการทำงานในกิจกรรมนั้น ๆ เพื่อให้เกิด Output เช่น นโยบาย คำสั่งซื้อ วันกำหนดส่งมอบสินค้า เป็นต้น
- 4) ตัวขับเคลื่อน (Mechanism) คือ ส่วนที่แสดงให้เห็นว่ากิจกรรมที่เกิดขึ้นบรรลุผลสำเร็จด้วยปัจจัยอะไรบ้าง เช่น เงินทุน บุคลากร เป็นต้น
- 5) ปัจจัยออกหรือผลลัพธ์ (Output) คือ ผลลัพธ์ที่เกิดจากการทำกิจกรรมนั้น ๆ เช่น สินค้า ปริมาณการส่งออก เป็นต้น

นอกจากนี้การเขียนตัวแบบ IDEF0 นั้นจะเริ่มจากกระบวนการในระดับบนไปสู่กระบวนการในระดับย่อย แสดงดังรูปที่ 2.3



รูปที่ 2.3 ตัวแบบ IDEF0 จากระดับบนสุดสู่กระบวนการในระดับย่อย

ในขั้นต้นของการวิเคราะห์ควรจะมีการบันทึกรายละเอียดที่ต้องการทั้งหมดในแต่ละกระบวนการเพื่อนำผลที่ได้ไปเขียนเป็นแผนภาพของกระบวนการไหลของข้อมูลสารสนเทศ หลังจากนั้นจะดำเนินการรวบรวมข้อมูลเพื่อใช้ในการวิเคราะห์สู่การกำจัดการกิจกรรมที่ไม่จำเป็น ช่วยลดความสูญเสียและความสำเร็จที่เพิ่มขึ้นในกระบวนการ หรือนำไปสู่การปรับปรุงกระบวนการธุรกิจที่เหมาะสมต่อไป ซึ่งการนำเครื่องมือ IDEF0 ไปประยุกต์ใช้ในการบริหารจัดการร่วมกับเครื่องมืออื่นๆ ในภาคอุตสาหกรรมนั้นเป็นไปอย่างกว้างขวางโดยจะยกตัวอย่างให้เห็นในหัวข้อของงานวิจัยที่เกี่ยวข้องต่อไป

## 2.4 แผนการผลิตรวม (Aggregate Planning)

การผลิตเป็นกระบวนการที่ทำให้เกิดการสร้างสิ่งหนึ่งสิ่งใดขึ้นมา จากการใช้ทรัพยากรหรือปัจจัยการผลิตที่มีอยู่ การดำเนินการผลิตจะเป็นไปตามลำดับขั้นตอนของการกระทำก่อนหลัง ส่วนการวางแผนเป็นขั้นตอนของการวิเคราะห์ข้อมูลที่มีอยู่และวางแผนการใช้ทรัพยากรให้ตรงตามความคาดหมายที่ต้องการและเป็นไปได้ไปอย่างมีประสิทธิภาพ [3]

การวางแผนการผลิตรวม (Aggregate Planning) หมายถึง กิจกรรมเพื่อกำหนดปริมาณในการผลิตของโรงงานอย่างคร่าวๆ ในช่วงของการวางแผน โดยพิจารณาจากความสามารถและปัจจัยการผลิตที่มีอยู่โดยไม่เจาะจงถึงรายละเอียด และจัดเป็นแผนระยะกลาง โดยมีเป้าหมายหลักของการวางแผนการผลิตรวม คือ การวางแผนเพื่อใช้ทรัพยากรที่มีอยู่ทำการผลิตให้มีประสิทธิภาพสูงสุด โดยการกำหนดกลยุทธ์การผลิตเพื่อให้สามารถสนองต่อความต้องการในลักษณะที่มีต้นทุนการผลิตและการปฏิบัติการต่ำที่สุด การวางแผนการผลิตรวมมีขั้นตอนหลักๆ 3 ขั้นตอน [4] ดังนี้

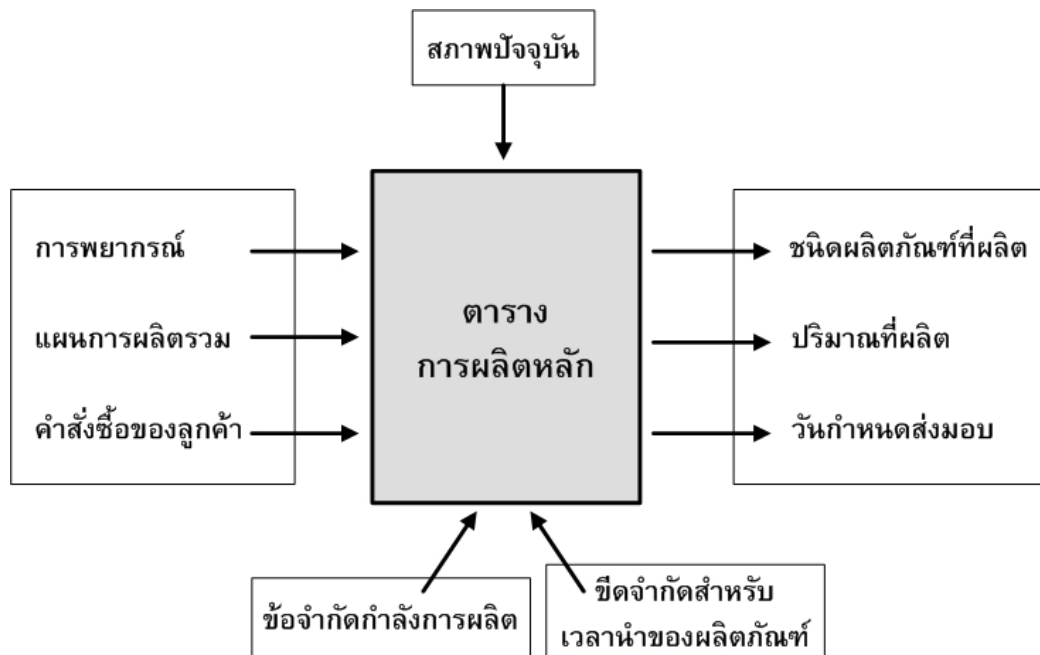
- 1) การพยากรณ์ความต้องการ เป็นขั้นตอนการคาดหมายถึงปริมาณความต้องการของผลิตภัณฑ์ในอนาคต
- 2) การวางแผนการผลิต คือ เมื่อรู้ความต้องการแล้วต่อไปคือการกำหนดกลยุทธ์สำหรับการผลิต
- 3) การกำหนดปริมาณของผลิตภัณฑ์ที่จะผลิต คือ จะทำการผลิตผลิตภัณฑ์ใดปริมาณเท่าใดตามกลยุทธ์ที่ได้วางไว้

แผนสำหรับกำหนดปริมาณผลผลิตและระยะเวลาการปฏิบัติงานในอนาคตล่วงหน้า เพื่อตอบสนองความต้องการด้วยกลยุทธ์ต่าง ๆ ที่สามารถแบ่งออกได้เป็น 2 ลักษณะ ได้แก่

- 1) กลยุทธ์เชิงรับ (Passive strategies) เป็นกลยุทธ์ที่เน้นการปรับกำลังการผลิตขององค์กรโดย
  - ก. การเปลี่ยนแปลงระดับสินค้าคงคลัง
  - ข. การเปลี่ยนแปลงจำนวนพนักงาน
  - ค. การเปลี่ยนแปลงเวลาการทำงาน
  - ง. การจ้างผู้ผลิตรายอื่นแทน
  - จ. การจ้างพนักงานชั่วคราว
- 2) กลยุทธ์เชิงรุก (Active strategies) เป็นกลยุทธ์ที่พยายามสร้างผลกระทบต่อรูปแบบความต้องการของตลาด และปรับความต้องการของลูกค้าโดย
  - ก. การใช้แรงกระตุ้นความต้องการของลูกค้า
  - ข. การค้างส่งสินค้าในช่วงที่มีความต้องการสูง
  - ค. การผลิตสินค้าต่างชนิดในนอกช่วงฤดูกาล

## 2.5 การจัดตารางการผลิตหลัก (Master Product Schedule)

การจัดตารางการผลิตหลัก (Master Production Schedule; MPS) เป็นการจัดทำแผนการผลิตที่ระบุเจาะจงลงไปว่าจะทำการผลิตชิ้นงานอะไร จำนวนเท่าใด และจะต้องเสร็จสมบูรณ์เมื่อใด [5] ช่วงเวลาที่ใช้ในตารางการผลิตหลักอาจมีหน่วยเป็น เดือน สัปดาห์ หรือวันก็ได้ แต่โดยทั่วไปแล้วมักจะจัดทำตารางการผลิตหลักเป็นรายเดือนหรือรายสัปดาห์ ขึ้นอยู่กับความเหมาะสมของการผลิตนั้น ๆ สำหรับการผลิตเพื่อจัดเก็บ (Make to Stock) ตารางการผลิตหลักสร้างขึ้นจากคำพยากรณ์ยอดขายหรือแผนการผลิตรวมที่ได้กำหนดไว้เป็นหลัก แต่สำหรับการผลิตสินค้าตามสั่ง (Make to Order) นั้น การกำหนดตารางการผลิตหลักจะสร้างขึ้นจากคำสั่งซื้อของลูกค้าเป็นหลัก ซึ่งตารางการผลิตหลักบ่งบอกถึงชนิดผลิตภัณฑ์ ปริมาณที่ผลิต และวันกำหนดส่งมอบอย่างชัดเจน ทั้งนี้การจัดทำตารางการผลิตหลักต้องมีความสอดคล้องกับแผนการผลิตรวมที่ได้กำหนดไว้แล้ว และต้องสอดคล้องกับกำลังการผลิตของโรงงานด้วย [3, 4] โดยไม่ควรให้มีจำนวนของผลิตภัณฑ์มากกว่าความสามารถของโรงงานที่จะทำการผลิตได้ ซึ่งความสามารถในการผลิตของโรงงานนี้พิจารณาได้จากเครื่องจักรและแรงงาน หน้าที่ของตารางการผลิตหลัก แสดงดังรูปที่ 2.4



รูปที่ 2.4 หน้าที่ของตารางการผลิตหลัก

ในบางองค์กรอาจแยกความแตกต่างของแผนการผลิตรวม (Aggregate Production Planning; APP) และตารางการผลิตหลักไม่เด่นชัดซึ่งอันที่จริงแล้วตารางการผลิตหลัก เป็นผลสืบเนื่องมาจากแผนการผลิตรวม แต่บอกถึงรายละเอียดมากกว่า ดังตารางที่ 2.1

ตารางที่ 2.1 การจัดตารางการผลิตหลักจากแผนการผลิตรวม

|                                            | มกราคม   |     |     |     | กุมภาพันธ์ |     |     |     |
|--------------------------------------------|----------|-----|-----|-----|------------|-----|-----|-----|
| <u>แผนการผลิตรวม</u>                       | 15001200 |     |     |     |            |     |     |     |
| (แสดงปริมาณปลาหมึกปรุงรสที่ต้องการทั้งหมด) |          |     |     |     |            |     |     |     |
| สัปดาห์                                    | 1        | 2   | 3   | 4   | 5          | 6   | 7   | 8   |
| <u>ตารางการผลิตหลัก</u>                    |          |     |     |     |            |     |     |     |
| (แสดงปริมาณปลาหมึกปรุงรสที่ต้องการทั้งหมด) |          |     |     |     |            |     |     |     |
| อบกรอบ 18x24                               | 100      |     | 100 |     | 100        |     | 100 |     |
| กรอบกระป๋อง 60x12                          |          | 500 |     | 500 |            | 450 |     | 450 |
| บดเผ็ด 10x36                               |          |     | 300 |     |            |     | 100 |     |

เมื่อได้ตารางการผลิตหลักแล้ว ตารางการผลิตนี้จะถูกนำมาใช้ในการจัดทำแผนความต้องการวัสดุคงคลังซึ่งเป็นการกำหนดแผนการสั่งซื้อหรือการสั่งงานเพื่อผลิตวัสดุอะไร จำนวนเท่าไร และเมื่อไร เพื่อให้ได้ผลิตผลตามกำหนดการของตารางการผลิตหลัก นอกจากนี้อาจจะทำแผนกำลังการผลิตควบคู่ไปด้วย เพื่อเตรียมกำลังการผลิตให้เพียงพอกับตารางการผลิตหลัก กำลังการผลิตที่เตรียมไว้ไม่ควรมีมากหรือน้อยเกินไป ถ้ากำลังการผลิตมากเกินไป จะทำให้เกิดค่าใช้จ่ายมากเกินไปและไม่มีความสิ้นเปลือง ในทางตรงกันข้าม ถ้ากำลังการผลิตน้อยเกินไปจะทำให้ไม่สามารถผลิตตามตารางการผลิตหลักได้ โดยมีขั้นตอนที่จำเป็นในการจัดตารางการผลิตหลักประกอบด้วย 6 ขั้นตอนดังนี้

ขั้นตอนที่ 1 เลือกผลิตภัณฑ์ที่จะกำหนดไว้ในตารางการผลิตหลัก

ขั้นตอนที่ 2 กำหนดขอบเขตของเวลาบนตารางการผลิตหลัก

ขั้นตอนที่ 3 หาข้อมูลความต้องการของแต่ละผลิตภัณฑ์ในช่วงเวลาต่าง ๆ ที่กำหนดไว้  
สำหรับการจัดตารางการผลิตหลัก

ขั้นตอนที่ 4 ทดลองจัดตารางการผลิตหลัก

ขั้นตอนที่ 5 วางแผนกำลังการผลิตแบบหยาบ (Rough Cut Capacity Planning: RCCP)  
บนตารางการผลิตหลัก

ขั้นตอนที่ 6 ทบทวนตารางการผลิตหลักเพื่อให้แน่ใจว่ามีกำลังการผลิตเพียงพอ



## 2.6 ทฤษฎีอัลกอริทึม (Theory of Algorithm)

David [6] ได้อธิบายทฤษฎีอัลกอริทึมโดยแบ่งเป็น ความหมายและคุณสมบัติของอัลกอริทึม และเพื่อให้เข้าใจอัลกอริทึมได้มากขึ้นยังได้ยกตัวอย่างของอัลกอริทึมประเภทต่างๆรวมทั้งข้อดีและข้อเสียของอัลกอริทึมแต่ละประเภท สุดท้ายเป็นวิธีการนำเสนอและการเขียนอัลกอริทึม

### 2.6.1 ความหมายของอัลกอริทึม

อัลกอริทึมมักจะถูกนิยามว่า เป็นขั้นตอนการทำงานที่ละเอียดสำหรับใช้แก้ปัญหาเฉพาะประเภท แต่ความหมายที่ถูกต้องของอัลกอริทึม คือ ขั้นตอนการทำงานสำหรับ Turing machine (เครื่องคอมพิวเตอร์ง่าย ๆ แบบหนึ่งซึ่ง A.M.Turing เป็นผู้ออกแบบสร้างขึ้นในทศวรรษ 1930 ว่ากันว่าเครื่องนี้สามารถแก้ปัญหาต่างๆได้ ถ้าส่งปัญหาเข้าไปโดยมีขั้นตอนวิธี (Algorithm) กำหนดให้ชัดเจน) อัลกอริทึมประกอบด้วยขั้นตอนการทำงานที่มีแบบแผนสำหรับแก้ปัญหาเฉพาะอย่างถึงแม้ว่าจะถูกใช้เป็นหลักในวิทยาศาสตร์ด้านคอมพิวเตอร์และวิทยาศาสตร์แขนงอื่นๆ แต่ก็มีเมื่อนำอัลกอริทึมไปใช้งานอย่างกว้างขวางในทุกวงการ

### 2.6.2 คุณสมบัติของอัลกอริทึม

คุณสมบัติที่สำคัญสองประการของอัลกอริทึมคือ เป็นขั้นตอนการทำงานที่ฉลาดและทำให้การทำงานง่ายขึ้น และเหมาะกับการทำงานที่มีการทำซ้ำ ๆ การสร้างอัลกอริทึมนั้นเป็นงานยาก และในขั้นตอนของอัลกอริทึม นั้นจะไม่มีขั้นตอนที่ทำให้เฉพาะบุคคล หรือใส่ขั้นตอนที่ไม่จำเป็นต้องทำซ้ำ หรือใส่ขั้นตอนที่ไม่ จำเป็นต้องทำ กล่าวคือไม่ว่าใครก็สามารถทำตามขั้นตอนนั้นได้ และสามารถทำซ้ำได้โดยให้ผลลัพธ์ออกมาเหมือนกัน ส่วนประกอบของอัลกอริทึมประกอบด้วย ข้อมูล (Input) ตัวแปร (Variables) และกิจกรรม (Activities) คุณสมบัติของอัลกอริทึมจะต้องมีขอบเขตชัดเจน มีความแน่นอน มีประสิทธิภาพ และไม่เฉพาะเจาะจง ดังนี้

1) มีขอบเขตชัดเจน (Finiteness) คือ ต้องมีขั้นตอนสิ้นสุดการทำงาน หลังจากมีขั้นตอนการทำงานที่แน่นอน ซึ่งขั้นตอนการทำงานนี้อาจมีจำนวนมาก (ขึ้นกับขนาดและค่าของข้อมูล) แต่ต้องมีจุดสิ้นสุดการทำงาน ขั้นตอนการทำงานที่ไม่มีคุณสมบัตินี้ อาจเป็นเพียงวิธีการคำนวณ (computing methods) ไม่ใช่อัลกอริทึม

2) มีความแน่นอน (Definiteness) คือ ทุกขั้นตอนการทำงานของอัลกอริทึมจะต้องมีความแน่นอนและอธิบายชัดเจน ต้องระบุให้ชัดเจนว่าต้องทำอะไร อย่างไร และต้องไม่มีขั้นตอนใดที่สามารถแปลความหมายได้มากกว่าหนึ่งความหมาย เนื่องจาก ภาษาพูดโดยทั่วไปนั้นมักไม่มีความแม่นยำและความชัดเจนในการสื่อความหมาย จึงมีการคิดค้นภาษาโปรแกรม (Programming language; ภาษาโปรแกรมหมายถึง ภาษาที่ออกแบบโครงสร้างขึ้นเพื่อใช้ในการเขียนคำสั่งให้

คอมพิวเตอร์ทำงาน ทำนองเดียวกับตัวโน้ตของภาษาคณิต ภาษาโปรแกรมมีตั้งแต่ระดับต่ำสุด คือ ใกล้เคียงกับภาษาเครื่อง (Machine language) มากที่สุด ไปจนถึงภาษาระดับสูง คือ ใกล้เคียงกับ ภาษามนุษย์ (ภาษาอังกฤษธรรมดา ๆ) มากที่สุด) ซึ่งการใช้ภาษาโปรแกรมในการเขียนคำสั่งนั้นจะมีความหมายชัดเจน การสื่อความหมายใน Computational method โดยใช้ภาษาโปรแกรมนั้นเรียกว่า โปรแกรม

3) มีประสิทธิภาพ (Effectiveness) ซึ่งโดยทั่วไปอัลกอริทึมจะต้องมีประสิทธิภาพ กล่าวคือทุกขั้นตอนการทำงานจะต้องเรียบง่ายเพื่อให้สามารถปฏิบัติตามได้

4) ไม่เฉพาะเจาะจง (Generality) คือ อัลกอริทึมที่ถูกต้องจะไม่เพียงแต่แก้ปัญหาเฉพาะเพียงปัญหาเดียว เช่น หาคำตอบของ  $8+9$  แต่จะสามารถหาคำตอบของปัญหาที่คล้ายๆกัน เช่น หาคำตอบของผลรวมระหว่างเลขจำนวนเต็มสองจำนวน

### 2.6.3 ประเภทของอัลกอริทึม

David [6] และวิชฌุ ช้างเนียม [7] ได้อธิบายว่าการแยกประเภทของอัลกอริทึมเหมือนกับการแยกประเภทขั้นตอนการแก้ไขปัญหา โดยอธิบายการทำงานของอัลกอริทึมแต่ละประเภทได้ดังนี้

1) Brute Force Algorithm เป็นการแก้ไขปัญหาโดยสั่งให้ทำงานไปเรื่อย ๆ จนกระทั่งได้คำตอบของทุกปัญหา ซึ่งเป็นวิธีการที่ง่าย แต่การแก้ไขในรูปแบบนี้ไม่เหมาะสมสำหรับแก้ไขปัญหามีข้อมูลจำนวนมาก ตัวอย่างอัลกอริทึมที่ใช้หลักการ Brute Force Algorithm เช่น การจัดเรียงข้อมูลแบบ Bubble sort, Selection sort เป็นต้น

2) Divide and Conquer Algorithm เป็นอัลกอริทึมที่มีหลักการคิดโดยแยกปัญหาออกเป็นสองส่วน คือ ส่วนที่หนึ่งแบ่งปัญหาออกเป็นส่วนเล็ก ๆ แล้วแก้ไขปัญหานั้นก่อน และอีกส่วนนำผลที่ได้จากการแก้ไขปัญหานั้นส่วนเล็ก ๆ กลับมารวมกันใหม่ ตัวอย่างอัลกอริทึมที่ใช้หลักการ Divide and Conquer Algorithm เช่น การจัดเรียงข้อมูลแบบ Quick sort, Merge sort เป็นต้น

3) Decrease and Conquer Algorithm เป็นอัลกอริทึมที่แก้ไขปัญหาลดขนาดของปัญหาลง และเลือกขนาดของกลุ่มปัญหาที่ต้องการแก้ไขปัญหานั้น โดยละเว้นปัญหามางส่วนไว้ก่อนเพื่อจะแก้ปัญหามีขนาดเล็กกว่าเดิม เนื่องจากการแก้ไขปัญหามีขนาดเล็กกว่าจะสามารถแก้ไขปัญหานั้นได้ง่ายกว่า ตัวอย่าง อัลกอริทึมที่ใช้หลักการ Decrease and Conquer Algorithm เช่น การค้นหาข้อมูลแบบไบนารี เป็นต้น

4) Transform and Conquer Algorithm เป็นการแก้ปัญหาลดขนาดของปัญหาลง โดยการเปลี่ยนรูปแบบของปัญหาที่ต้องการแก้ไขให้อยู่ในรูปแบบอื่นก่อน ด้วยคาดหวังว่าเมื่อเปลี่ยนรูปแบบของปัญหาแล้วจะสามารถแก้ไขปัญหานั้นได้ง่าย และรวดเร็วขึ้น ตัวอย่างในการเปลี่ยนโครงสร้างข้อมูลก่อนการแก้ไขปัญหานั้น เช่น การนำข้อมูลที่ต้องการค้นหามาจัดเรียงข้อมูลก่อน เพื่อให้สามารถใช้อัลกอริทึมที่มีประสิทธิภาพค้นหาข้อมูลได้ แทนที่จะค้นหาข้อมูลทีละตัวจากข้อมูลที่กระจัดกระจาย

5) Greedy Algorithm หรืออัลกอริทึมแบบละโมภ เป็นอัลกอริทึมที่มีลักษณะของการแก้ไขปัญหาด้วยการเพิ่มประสิทธิภาพของการแก้ไขปัญหาลงให้เหมาะสมที่สุด (Optimization problems) ซึ่งเป็นรูปแบบอัลกอริทึมที่พิจารณาคำตอบที่ดีที่สุดและคุ้มค่าที่สุดในการแก้ไขปัญหานั้นๆ โดยแบ่งปัญหาเป็นส่วนๆ และหาคำตอบที่ดีที่สุดในแต่ละส่วน เมื่อนำคำตอบมารวมกันก็จะให้วิธีการแก้ปัญหาโดยรวมที่ดีที่สุด แต่เมื่อได้คำตอบแล้ว จะไม่สามารถย้อนกลับไปแก้ไขได้อีก ตัวอย่างอัลกอริทึมที่ใช้หลักการ Greedy Algorithm เช่น ปัญหาการทอนเหรียญคือเลือกทอนเหรียญในหน่วยที่มีขนาดใหญ่ที่สุดก่อน เป็นต้น

6) Dynamic Programming Algorithm หรืออัลกอริทึมโปรแกรมพลวัต เป็นอัลกอริทึมที่มีลักษณะการแก้ไขปัญหาลงด้วยการแบ่งปัญหาเป็นส่วนเล็ก ๆ แล้วนำผลของปัญหาลึก ๆ ที่ดีที่สุดนำมาแก้ไขปัญหาลึก ๆ หรือเรียกว่าการแก้ไขปัญหาลงจากล่างขึ้นบน (Bottom-up approach) ตัวอย่างอัลกอริทึมที่ใช้หลักการ Dynamic Programming Algorithm เช่น การหาค่าตัวเลข Fibonacci เป็นต้น

7) Backtracking Algorithm หรืออัลกอริทึมย้อนรอยถอยหลัง เป็นอัลกอริทึมค้นหาเส้นทางทุกเส้นทางที่เป็นไปได้ เพื่อหาคำตอบของปัญหาที่ละส่วนย่อยว่า คำตอบนั้นเป็นคำตอบที่ถูกต้องหรือไม่ แต่ถ้าคำตอบนั้นไม่ใช่ส่วนหนึ่งของคำตอบจะถอยหลังกลับมาจุดเดิม และยกเลิกคำตอบนั้นแล้วค้นหาคำตอบใหม่ ตัวอย่างอัลกอริทึมที่ใช้หลักการ Backtracking Algorithm เช่น การกำหนดสีให้กับเมืองในแผนที่, การคิดความเป็นไปได้ทั้งหมดของการเดินหมากระดาน เป็นต้น

8) Branch and Bound Algorithm เป็นอัลกอริทึมที่เพิ่มประสิทธิภาพในการแก้ไขปัญหาลงด้วยการนำ โครงสร้างทรี (Tree) มาเก็บปัญหาย่อย ๆ โดยที่ปัญหาลึกจะอยู่ในตำแหน่งบนสุดของทรี คือ โหนดราก (Root node) และในแต่ละโหนดจะแก้ไขปัญหาลงของตัวเอง และถ้าแก้ปัญหาลึกต้องจะใช้ผลนั้นเป็นข้อมูลในการแก้ไขปัญหาลงทั้งหมด แต่ถ้าการแก้ไขปัญหาลึกไม่ถูกต้องให้แบ่งปัญหาลึกออกเป็นสองโหนดย่อย เก็บไว้ในตำแหน่งโหนดลูกของโหนดที่แก้ไขปัญหาลึกไม่ถูกต้อง แล้วกลับไปทำใหม่จนกระทั่ง ทุกโหนดย่อยในทรีสามารถแก้ไขปัญหาลึกได้ทุกโหนด ตัวอย่างอัลกอริทึมที่ใช้หลักการ Branch and Bound Algorithm เช่น ปัญหาในการหาเส้นทางที่เหมาะสมให้กับพนักงานขายสินค้าให้สามารถเดินทางได้ครบทุกที่ได้เร็วที่สุด (Travelling Salesman Problems; TSP) เป็นต้น

9) Recursive Algorithm หรืออัลกอริทึมแบบวนซ้ำ เป็นการแก้ไขปัญหาลงขึ้นพื้นฐานด้วยการเรียกใช้ตัวเองซ้ำ ๆ โดยนำข้อมูลปัญหาลงย่อยของปัญหาลงทั้งหมดกลับมาเป็นข้อมูลในการแก้ไขปัญหาลึก ตัวอย่างอัลกอริทึมแบบเรียกซ้ำ เช่น การหาค่า Factorial, อัลกอริทึมบวกข้อมูลตัวเลขที่อยู่ในกลุ่ม เป็นต้น

10) Randomized Algorithms หรืออัลกอริทึมแบบสุ่ม ใช้หลักการสุ่มข้อมูลแล้วนำข้อมูลที่สุ่มเลือกขึ้นมาได้กระทำกับอัลกอริทึมเพื่อให้ได้ผลตามที่ต้องการ ตัวอย่างอัลกอริทึมที่ใช้หลักการ Randomized Algorithms ไปใช้งาน เช่น การพยายามหาข้อมูลที่สำคัญที่สุดด้วยการเลือกข้อมูลเพื่อใช้ในการ เปรียบเทียบ (Pivot) ในการจัดเรียงข้อมูล เป็นต้น

11) Probabilistic algorithms เป็นอัลกอริทึมที่ประกอบด้วยวิธีการหาคำตอบแบบสุ่ม (randomly) หรือแบบ สุ่มเทียม (pseudo-randomly)

12) Genetic algorithms หรือ Evolutionary algorithms เป็นอัลกอริทึมที่เลียนแบบวิวัฒนาการทางชีววิทยา (biological evolutionary processes) ซึ่งประกอบด้วยวิธีการหาคำตอบที่ดีที่สุดโดยผ่านกระบวนการ Reproduction, Mutation, Recombination และ Crossbreeding ซึ่ง genetic programming นี้จะใช้ โดยตรงกับอัลกอริทึมที่สามารถหาคำตอบที่เป็นไปได้ ซึ่ง Natallia Kokash [8] ได้อธิบายเพิ่มเติมว่าในการหาคำตอบสำหรับ optimization problem หลักการของ Genetic algorithms หรือ Evolutionary algorithms คือ สนใจคำตอบที่เหลืรอดอยู่จากชุดของคำตอบที่เป็นไปได้ ซึ่งเกิดจากกระบวนการประมาณการครั้ง ละเล็กครั้งละน้อยจนได้คำตอบที่เหมาะสม จากนั้นจึงทำซ้ำขั้นตอนนี้ โดยการเทียบคำตอบที่ได้กับสมการวัตถุประสงค์ ซึ่งเรียกว่า fitness for evolutionary algorithms จากนั้นนำคำตอบที่ได้ทั้งหมดมา breeding โดยใช้ operators ที่ได้จากกระบวนการ genetic process วิธีการนี้จะทำให้ได้คำตอบใหม่ (evolution of populations) ที่เหมาะสมกับสถานการณ์กว่าคำตอบแรก (their ancestors)

13) Heuristic algorithms เป็นอัลกอริทึมที่มีวัตถุประสงค์ในการหาคำตอบที่เหมาะสมที่สุดด้วยการประมาณ ค่าที่เหมาะสม (a suitable approximation) ซึ่งจะใช้ในสถานการณ์ที่ทรัพยากรมีจำกัด เช่น เวลาในการหาคำตอบมีจำกัด ทำให้ไม่สามารถหาคำตอบที่ดีที่สุดได้หรือไม่สามารถหาอัลกอริทึมที่ดีที่สุดได้ในขณะนั้น

#### 2.6.4 ขั้นตอนการพัฒนาอัลกอริทึม

วิชญ ช่างเนียม [7] ได้อธิบายขั้นตอนการพัฒนาอัลกอริทึม ประกอบด้วย 9 ขั้นตอนดังนี้

1) ระบุปัญหา (Statement of the problem) การระบุปัญหาที่ชัดเจนมีความสำคัญ เนื่องจากปัญหาใน ชีวิตจริงจะมีความซับซ้อนจึงเป็นการยากที่จะระบุปัญหาให้ชัดเจน แต่หากไม่ระบุปัญหาให้ชัดเจนแล้วก็จะเกิดความสับสนเปล่า (Garbage-in and Garbage-out (GIGO) situation)

2) พัฒนาแบบจำลอง (Development of Model) แบบจำลองเป็นตัวแทนของสถานการณ์จริงเนื่องจาก แบบจำลองทำให้ปัญหาอยู่ในรูปแบบที่เข้าใจได้ แบบจำลองอาจอยู่ในรูปกราฟ เมตริกซ์ สมการ ลำดับของ ตรรกศาสตร์ แบบจำลองทางคณิตศาสตร์ที่ประกอบ ด้วยสมการวัตถุประสงค์ และสมการข้อจำกัด เป็นต้น ตัวอย่างแบบจำลองในปัญหา Travelling salesman problem ก็คือรูป Distance network และ Distance matrix หรือในการหาเส้นทางที่สั้นที่สุด อาจใช้แผนที่แสดงถนนเพื่อหาเส้นทางที่สั้นที่สุดเป็นแบบจำลอง ส่วนปัญหาการจัดการเคลื่อนย้ายวัตถุดิบในบริเวณผลิต อาจใช้แผนผังสามมิติของพื้นที่ในการวางแผน เป็นต้น

3) ออกแบบอัลกอริทึม (Design of algorithm) อัลกอริทึมประกอบด้วยขั้นตอนการทำงานที่ถูกกำหนดอย่างชัดเจน ในหลายปัญหาที่มีการออกแบบอัลกอริทึมให้มีความทำงานแบบวนซ้ำซึ่งจะทวนขั้นตอนเดิมจนกว่าจะได้คำตอบที่ต้องการ หลักการหนึ่งในการออกแบบอัลกอริทึมคือ หลักการ top-down programming โดยจะมีการแบ่งปัญหามาออกเป็นปัญหาย่อยๆ และแบ่งย่อยออกไปเรื่อยๆ จนกว่าจะได้ขั้นตอนการทำงานที่ชัดเจน เมื่อนำขั้นตอนย่อยๆ มารวมกันก็จะสามารถแก้ปัญหาหลักได้ ข้อสำคัญในการออกแบบอัลกอริทึมคือ นักวิจัยควรระลึกถึงวัตถุประสงค์ในการแก้ปัญหา และควรมีขั้นตอนการทำงานเท่าที่จำเป็นให้น้อยที่สุด เพื่อว่าในการทำงานนั้นจะใช้เวลาและหน่วยความจำน้อยที่สุด ส่งผลให้อัลกอริทึมทำงานอย่างมีประสิทธิภาพ

4) ตรวจสอบความถูกต้องของอัลกอริทึม (Correctness of the algorithm) ความถูกต้องของอัลกอริทึม หมายถึงความถูกต้องของตรรกะหรือขั้นตอนของอัลกอริทึม โดยทั่วไปแล้วอัลกอริทึมจะมีขั้นตอนจำนวนมาก มีการแยกขั้นตอนการทำงาน (branching) และมีความทำงานแบบวนซ้ำ (loops) รวมถึงการทำงานแยกหลายกลุ่มย่อย ดังนั้นจึงต้องมีการตรวจสอบความถูกต้องในการทำงานของทุกส่วนว่าให้ผลลัพธ์ออกมาตามที่คาดหมาย โดยการทวนสอบความถูกต้องของอัลกอริทึมด้วยข้อมูลที่ทราบคำตอบอยู่แล้ว

5) นำไปใช้จริง (Implementation) เมื่อตรวจสอบความถูกต้องของอัลกอริทึมแล้ว จะต้องแปลงอัลกอริทึมให้อยู่ในรูปของ code (program) ซึ่งจะสามารถนำไปใช้งานในเครื่องคอมพิวเตอร์ได้ ในขั้นตอนนี้โครงสร้างข้อมูลจริงจะต้องถูกนำมาพิจารณา ที่สำคัญจะต้องตัดสินใจเลือกภาษาที่จะใช้เนื่องจากส่งผลโดยตรงต่อ ประสิทธิภาพของโปรแกรม

6) วิเคราะห์ความซับซ้อนของอัลกอริทึม (Analysis of complexity algorithm) เป็นการหาจำนวนขั้นตอนการทำงาน รวมถึงเวลาที่ต้องใช้ในการแก้ปัญหาที่ยากที่สุด (worst-case behavior) ซึ่งจะทำให้ต้องผ่านขั้นตอนการทำงานทุกขั้นตอนที่ออกแบบไว้ การวิเคราะห์นี้ทำให้ทราบถึงเวลามากที่สุดที่ต้องใช้ในการ แก้ปัญหาคับด้วยอัลกอริทึมหนึ่ง ๆ

7) ทดสอบโปรแกรม (Program testing) เป็นการตรวจสอบความถูกต้องของโปรแกรมอีกชั้นหนึ่ง หลังจากที่ได้มีการตรวจสอบความถูกต้องของอัลกอริทึมแล้ว เนื่องจากอาจเกิดความผิดพลาดในขั้นตอนการแปลงอัลกอริทึมเป็นโปรแกรม

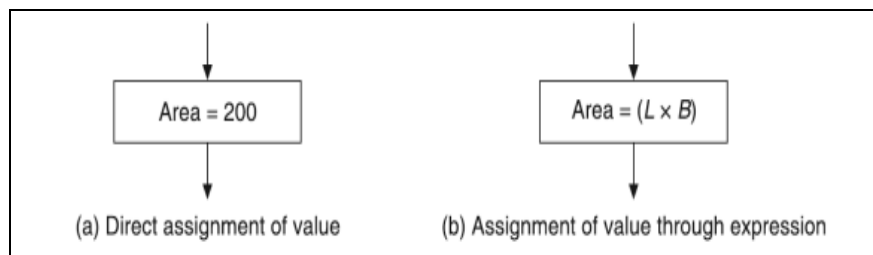
8) เปรียบเทียบคำตอบที่ได้กับอัลกอริทึมที่ดีที่สุดที่มี (Comparison with the best existing algorithm) การพัฒนาอัลกอริทึมมีวัตถุประสงค์เพื่อหาคำตอบที่ดีกว่าเดิม จึงต้องมีการเปรียบเทียบคำตอบที่ได้กับคำตอบ จากอัลกอริทึมเดิมที่ดีที่สุดที่มีอยู่ ถ้าคำตอบที่ได้ใหม่ดีกว่าคำตอบเดิมก็จะนำไปใช้ได้ แต่หากคำตอบที่ได้ไม่ดีกว่าเดิม ก็จะต้องย้อนกลับไปขั้นตอนที่ 4) ถึงขั้นตอน 8) และหากไม่สามารถหาคำตอบที่ดีกว่าเดิมได้เลย นั้นหมายความว่าอัลกอริทึมที่มีอยู่เดิมเป็นอัลกอริทึมที่ดีที่สุดแล้ว

9) การบันทึก (Documentation) เป็นขั้นตอนสุดท้ายในการพัฒนาอัลกอริทึม

### 2.6.5 Flowchart และส่วนประกอบของอัลกอริทึม

Flowchart ของอัลกอริทึม คือ รูปแสดงตรรกะของขั้นตอนการทำงานของอัลกอริทึม ประกอบด้วย Block พื้นฐานที่แบ่งประเภทได้ดังนี้

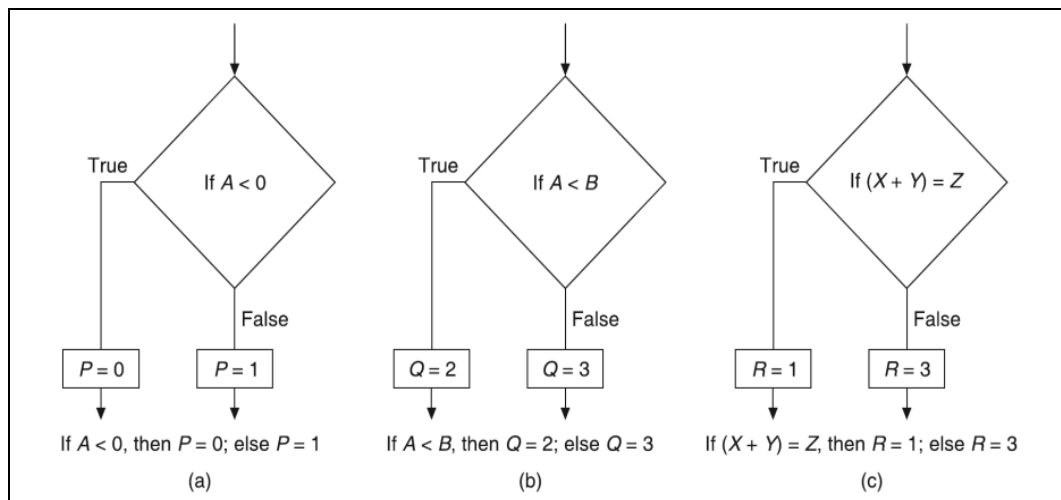
- 1) Function vertex เป็นการประกาศค่า สำหรับตัวแปร โดยอาจแสดงค่าโดยตรงหรือแสดงเป็นสูตรคำนวณ ดังตัวอย่างในรูปที่ 2.5



รูปที่ 2.5 ตัวอย่าง Function vertex

ที่มา : [http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=htm\\_text#v=onepage&q&f=false](http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=htm_text#v=onepage&q&f=false)

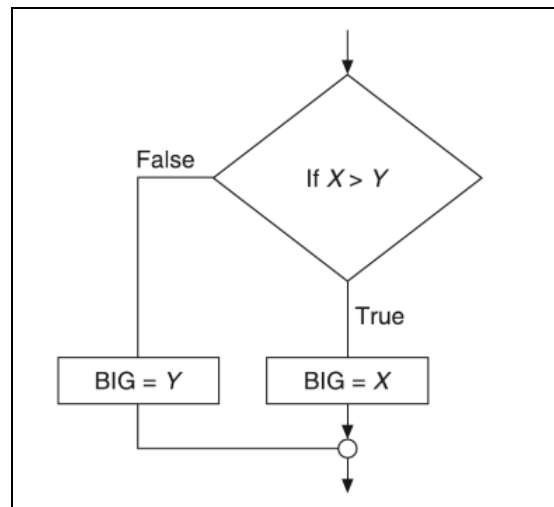
- 2) Predicate vertex เป็น node ในตรรกะของ Flowchart ซึ่งเป็นการตรวจสอบค่าของตัวแปรและให้คำตอบ ออกมาเป็น Boolean conditions (True or False) ดังตัวอย่างในรูปที่ 2.6



รูปที่ 2.6 ตัวอย่าง Predicate vertex

ที่มา [http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=htm\\_text#v=onepage&q&f=false](http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=htm_text#v=onepage&q&f=false)

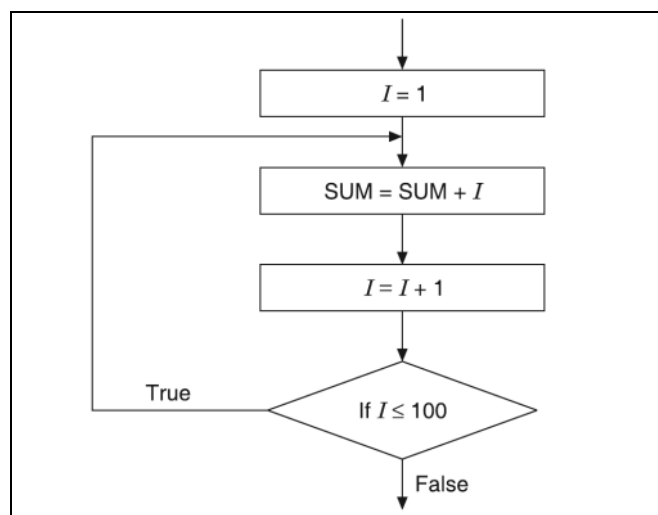
- 3) Collecting vertex เป็น node ในตรรกะของ Flowchart ซึ่งรวมถึงของขั้นตอนก่อนหน้าเข้าด้วยกัน โดย มีลักษณะเป็นวงกลมขนาดเล็ก ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 ตัวอย่าง Collecting vertex

ที่มา [http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=html\\_text#v=onepage&q&f=false](http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=html_text#v=onepage&q&f=false)

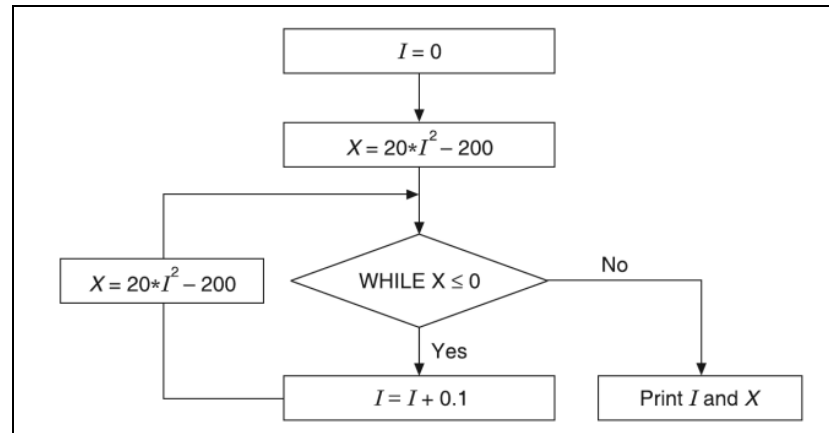
เมื่อนำ block พื้นฐานมาจัดเรียงกัน ก็จะได้ขั้นตอนการทำงานที่เรียกว่า Iteration ซึ่งเป็นการทำงานแบบวนซ้ำ ดังตัวอย่างในรูปที่ 2.8



รูปที่ 2.8 ตัวอย่าง Flowchart แบบ Iteration

ที่มา [http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=html\\_text#v=onepage&q&f=false](http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=html_text#v=onepage&q&f=false)

ส่วนการทำงานแบบวนซ้ำอีกประเภทหนึ่งดังตัวอย่างในรูปที่ 2.9 นั้นเป็นแบบ WHILE Type เป็นการวนซ้ำ จนกว่าจะได้คำตอบที่ต้องการ



รูปที่ 2.9 ตัวอย่าง Flowchart ชนิด WHILE Type

ที่มา [http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=html\\_text#v=onepage&q&f=false](http://books.google.co.th/books?id=ogBlw9LR7BsC&printsec=frontcover&hl=th&output=html_text#v=onepage&q&f=false)

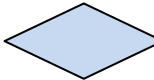
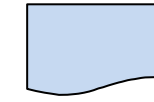
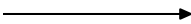
## 2.7 ฟังงาน (Flowchart)

ฟังงาน หรือ Flowchart เป็นเครื่องมือที่ใช้ออกแบบระบบงานด้วยสัญลักษณ์ช่วยให้มีโครงสร้างของระบบงานที่เป็นลำดับขั้นตอนและเข้าใจได้ง่าย การนำฟังงานมาใช้ในการออกแบบโปรแกรม สามารถตรวจสอบได้ว่ามีลำดับขั้นตอนการทำงานถูกต้องหรือไม่ และสามารถเปลี่ยนแปลงแก้ไขข้อผิดพลาดของระบบงานภายในฟังงานได้ง่ายกว่าการหาข้อผิดพลาดที่เกิดจากการเขียนโปรแกรม อีกทั้งยังช่วยลดความสับสนในการพัฒนาโปรแกรมอีกด้วย วัตถุประสงค์ที่สำคัญที่สุดในการนำฟังงานมาเป็นเครื่องมือในการพัฒนาโปรแกรมคือ เพื่อให้บุคคลอื่นสามารถทำความเข้าใจ ถึงลำดับขั้นตอนการทำงานของโปรแกรมและผลลัพธ์ที่ได้จากการทำงานของโปรแกรมที่พัฒนาขึ้นได้

สัญลักษณ์มาตรฐานที่นำมาใช้ในการเขียนฟังงานเรียกว่า สัญลักษณ์ ANSI (American National Standards Institute) มีอยู่ 8 สัญลักษณ์ ดังตารางที่ 2.2



ตารางที่ 2.2 สัญลักษณ์ที่นำมาใช้ในการเขียนผังงาน

| สัญลักษณ์                                                                           | ชื่อเรียก          | ความหมายสัญลักษณ์ในการใช้งานในผังงาน                                |
|-------------------------------------------------------------------------------------|--------------------|---------------------------------------------------------------------|
|    | Terminator         | จุดเริ่มต้นและจุดสิ้นสุดของโปรแกรม                                  |
|    | Process            | การประมวลผลหรือการคำนวณของโปรแกรม                                   |
|    | Data               | รับข้อมูลเข้ามาในโปรแกรม หรือส่งค่าออกไปจากโปรแกรม                  |
|    | Decision           | ตรวจสอบเงื่อนไข แล้วเลือกการทำงานของโปรแกรม                         |
|   | Document           | แสดงผลออกทางเอกสาร                                                  |
|  | On-page reference  | จุดเชื่อมต่อหลายเส้นทางของโปรแกรมให้เหลือการเข้ามาเพียงเส้นทางเดียว |
|  | Off-page reference | ขึ้นหน้าใหม่ในกรณีที่ผังงานมีความยาวเกินกว่าที่จะแสดงพอในหนึ่งหน้า  |
|  | Arrow              | ลูกศรแสดงทิศทางการทำงานของโปรแกรมและการไหลของข้อมูล                 |

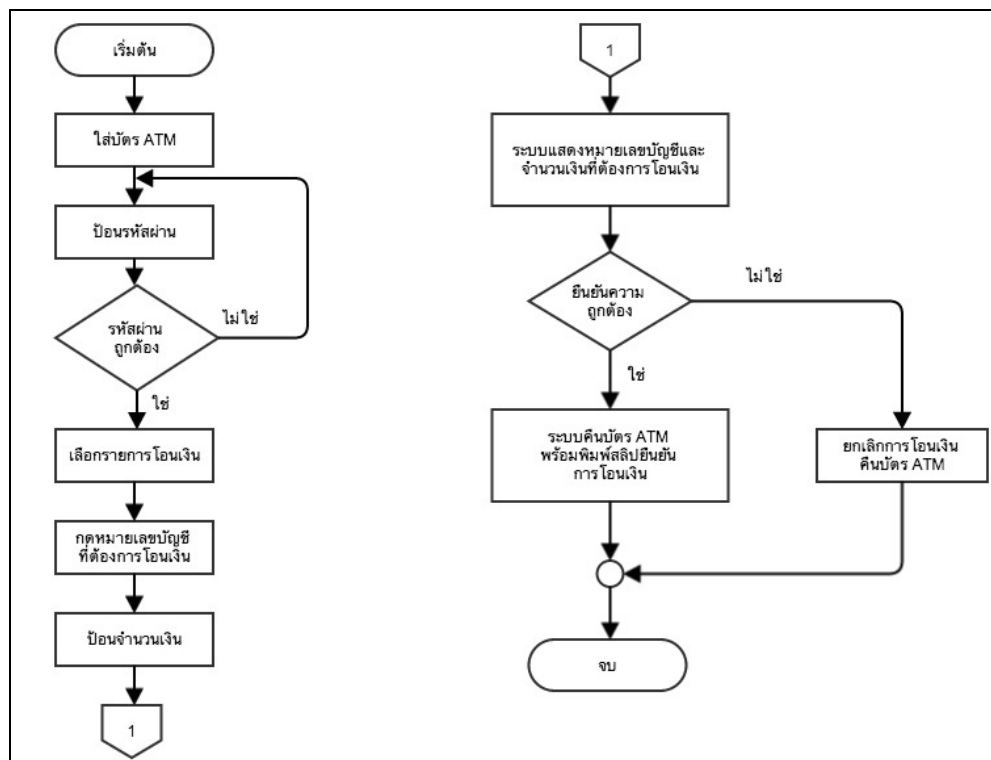
วิชญ์ ช้างเนียม [7] ได้อธิบายพื้นฐานของโครงสร้างข้อมูลและอัลกอริทึม กล่าวคือเป็นโครงสร้างข้อมูล (Data structure) หมายถึง การจัดการข้อมูลในหน่วยความจำภายในเครื่องคอมพิวเตอร์ หรือในบางครั้งเป็นการจัดการข้อมูลในดิสก์ให้มีความสัมพันธ์กันภายในกลุ่มข้อมูล ให้มีรูปแบบและข้อกำหนดที่ชัดเจนในการกำหนดคุณสมบัติเพื่อสร้างความสัมพันธ์ภายในกลุ่มข้อมูล รูปแบบการเก็บข้อมูลที่อยู่ในรูปแบบโครงสร้างข้อมูล เช่น อาร์เรย์ (Array), ลิงคัลลิสต์ (Link-list), สแตก (Stack), ไบนารีทรี (Binary tree) เป็นต้น

อัลกอริทึม (Algorithm) หรือเรียกอีกอย่างว่า ขั้นตอนวิธี เป็นวิธีการแสดงลำดับขั้นตอนในการทำงาน หรือ การแก้ไขปัญหาอย่างใดอย่างหนึ่ง อัลกอริทึมหรือขั้นตอนวิธีเป็นสิ่งที่พบได้ในชีวิตประจำวัน

เช่น ขั้นตอน การใช้งานของตู้กดเงินอัตโนมัติ (ATM) เพื่อทำธุรกรรมทางการเงิน ดังแสดงตัวอย่างในรูปที่ 2.10 เมื่อนำสัญลักษณ์ มาเขียนเป็น Flowchart ของการทำงาน จะได้ Flowchart ดังรูปที่ 2.11

1. ใส่บัตร ATM
2. ป้อนรหัสผ่านของบัตร ATM
3. ในหน้าบริการ เลือกบริการรายการโอนเงิน
4. เลือกรูปแบบการโอนเงินว่าจะโอนเงินเข้าบัญชีอื่นธนาคารเดียวกัน หรือธนาคารอื่น ๆ
5. กดหมายเลขบัญชีที่ต้องการโอนเงินเข้าบัญชี
6. ตรวจสอบเลขที่บัญชีที่ป้อนถูกต้องหรือไม่ ถ้าถูกต้องให้กดตกลง
7. กรอกจำนวนเงินที่ต้องการโอนเงิน แล้วกดตกลง
8. ชื่อบัญชีและจำนวนเงินที่ต้องการโอนจะปรากฏขึ้นเพื่อยืนยันความถูกต้องในการโอนเงิน เมื่อตรวจสอบบัญชีและจำนวนเงินถูกต้องให้กดตกลง เป็นการเสร็จสิ้นการโอนเงิน
9. รับบัตร ATM
10. รับใบสลิปการโอนเงิน

รูปที่ 2.10 อัลกอริทึมหรือขั้นตอนวิธีการใช้ตู้กดเงินอัตโนมัติ (ATM) เพื่อโอนเงิน



รูปที่ 2.11 ฟังก์ชันขั้นตอนการใช้ตู้กดเงินอัตโนมัติ (ATM) เพื่อโอนเงิน

## 2.8 งานวิจัยที่เกี่ยวข้อง

การวิเคราะห์กระบวนการธุรกิจด้านการวางแผนผลิตด้วยการประยุกต์ใช้ Integration Definition for Function Modeling (IDEF0) เป็นเครื่องมือสำคัญที่ใช้ในการทำผังกระบวนการธุรกิจ (Business Process Mapping) สามารถนำไปวิเคราะห์แนวทางการปรับปรุงกระบวนการธุรกิจได้ เช่น

สิทธิพร ฉันท์เฉลิมพร [9] ได้ศึกษาการจัดการโซ่อุปทานของอุตสาหกรรมการผลิตไก่สุกแช่แข็งส่งออกใน บริษัท จี.เอฟ.พี.ที. จำกัด (มหาชน) ซึ่งประสบปัญหาการส่งมอบสินค้าให้กับลูกค้าล่าช้า มีการวิเคราะห์ปัญหาโดยประยุกต์ใช้ Integration Definition for Function Modeling (IDEF0) และแผนภาพการไหลของกระบวนการ จากนั้นวิเคราะห์หาแนวทางการปรับปรุงขั้นตอนการดำเนินงานด้านวางแผนการส่งออก สามารถลดเวลานำในการวางแผน ทำให้ตอบสนองความต้องการของลูกค้าได้อย่างรวดเร็วและเพิ่มสมรรถภาพการส่งมอบสินค้าตรงเวลาได้ดีขึ้น

การพัฒนาฮิวริสติกอัลกอริทึมเป็นแนวทางหนึ่ง ในการหาคำตอบด้วยการประมาณค่าที่เหมาะสม ถูกนำไปใช้ในสถานการณ์ที่ทรัพยากร เช่น เวลา มีจำกัด ทำให้ไม่สามารถหาคำตอบที่ดีที่สุดได้ หรือไม่สามารถหาอัลกอริทึมที่ดีที่สุดได้ในขณะนั้น โดยมีการประยุกต์ใช้กับปัญหาที่หลากหลาย เช่น

นพวรรณ อังกูรคันสนีย [10] พัฒนาฮิวริสติกส์อัลกอริทึมแบบ Dynamic Combination Rule Forward Assignment Heuristic Algorithm เพื่อแก้ปัญหาการจัดรถขนส่งยา เพื่อให้ได้ต้นทุนการขนส่งที่ต่ำที่สุดและพัฒนาระบบสนับสนุนการตัดสินใจสำหรับเป็น เครื่องมือช่วยให้พนักงานสามารถหาคำตอบภายในเวลาอันเหมาะสม

Bilgin และ Azizoglu [11] ได้ศึกษาเกี่ยวกับการจัดการภาระงานและปัญหากำลังการผลิตในระบบการผลิตแบบยืดหยุ่น (Flexible Manufacturing Systems) และอุตสาหกรรมสารกึ่งตัวนำ (Semiconductor Industries) โดยหน่วยการผลิตมีข้อจำกัดในเรื่องชุดอุปกรณ์และเวลา โดยได้พัฒนาฮิวริสติก 2 ตัว และนำ Tabu Search Algorithm มาใช้ในการจัดการภาระงานข้างต้นที่เป็นปัญหาแบบ NP-hard ผลการวิจัยพบว่ากระบวนการ Tabu Search สามารถให้คำตอบที่เหมาะสมอย่างรวดเร็ว

Suporn Engudomnukul [12] นำฮิวริสติกมาใช้ในการปรับปรุงการจัดกำลังการผลิตของสายการผลิตยาเม็ดในองค์การเภสัชกรรม เนื่องจากพบว่าเครื่องจักรในสายการผลิตที่มีอยู่ 5 สายการผลิตมีการรับปริมาณงานที่แตกต่างกันมาก อีกทั้งการจัดกำลังการผลิตใช้เวลานาน และเมื่อมีรายการใหม่ก็ไม่สามารถตัดสินใจได้ว่าจะจัดกำลังการผลิตไปที่สายการผลิตใดจึงเหมาะสม ดังนั้นจึงได้นำ ฮิวริสติกและกระบวนการวิเคราะห์ตามลำดับขั้นมาใช้ในการปรับปรุงการจัดกำลังการผลิต ผลการปรับปรุงพบว่าปริมาณงานในแต่ละสายการผลิตมีความสมดุลและยังทราบกำลังการผลิตที่เหลืออยู่

สำหรับปัญหาด้านการวางแผนผลิตนั้น มีการพัฒนาเกี่ยวกับการจัดตารางการผลิต และการประยุกต์ใช้เครื่องมือการจัดตารางการผลิตหลัก (Master Production Schedule) อย่างกว้างขวางในธุรกิจหลายประเภท เช่น

กฤษฎา ชาวบางพรหม [13] ศึกษาโรงงานที่มีลักษณะธุรกิจเป็นแบบการรับจ้างผลิตสินค้า ซึ่งลูกค้าจะออกแบบและจัดส่งบรรจุภัณฑ์มาให้ และเจ้าหน้าที่วางแผนการผลิตยังจัดตารางการผลิตหลักด้วยตนเองโดยใช้ความสามารถและประสบการณ์ของพนักงานผู้วางแผน ทำให้เกิดปัญหาหลายประการ ได้แก่ การส่งมอบไม่ทันกำหนดของลูกค้า มีการทำงานล่วงเวลาของเจ้าหน้าที่ที่แผนกผลิตจำนวนมาก และ ปริมาณการส่งบรรจุภัณฑ์ไม่ตรงกับความต้องการที่ใช้ในการผลิตจริง ดังนั้นจึงนำเสนอระบบสนับสนุนการตัดสินใจสำหรับการจัดตารางการผลิตหลัก และการวางแผนความต้องการวัสดุคงคลังในส่วนของการบรรจุภัณฑ์แต่ละรายการ ซึ่ง สามารถประยุกต์ใช้ได้อย่างมีประสิทธิภาพในทางปฏิบัติ

เกียรติจิกร วรปรัชญา [14] เสนอการจัดตารางการผลิตใหม่สำหรับกระบวนการหล่อเหล็กแผ่นหลายสายการผลิตโดยกำหนดตัวชี้วัดสมรรถนะที่เหมาะสมในทางปฏิบัติ ด้วยการจัดทั้งลำดับงาน การจัดสรรทรัพยากรให้เหมาะสม และคำนึงถึงเสถียรภาพจากการใช้ตารางการผลิตใหม่ด้วยกลยุทธ์การจัดตารางการผลิตใหม่ให้มีความเหมาะสมในแต่ละกรณี เช่น กรณีคำสั่งผลิตเร่งด่วนและกรณีเครื่องจักรเสียหาย เป็นต้น และได้ใช้ขั้นตอนวิธีทางพันธุกรรมซึ่งเหมาะสมสำหรับกระบวนการหล่อแบบต่อเนื่องซึ่งเป็นระบบไหลเลื่อนแบบยืดหยุ่น มาทำการค้นหาค่าเหมาะสมของตารางการผลิตใหม่

พูลพัฒน์ พัวทวีพงศ์ [15] นำเสนอการเพิ่มสมรรถภาพของโซ่อุปทานเครื่องประดับด้านการส่งมอบ โดยเริ่มจากการศึกษาโครงสร้างโซ่อุปทานเครื่องประดับและการเชื่อมโยงกันระหว่างอุตสาหกรรมมุ่งเน้นด้านการวางแผนผลิต ด้วยการเพิ่มระบบการวางแผนผลิตเข้าสู่อุตสาหกรรมเครื่องประดับเชื่อมโยงกับอุตสาหกรรมพลอยโดยเริ่มจากการวางแผนการผลิตรวมโดยสร้างสมการเชิงเส้น และสร้างขั้นตอนการจัดตารางการผลิตหลักที่ระบุกำหนดการของการผลิตทุกขั้นตอนเพื่อควบคุมให้ส่งมอบได้ทันเวลา

ภัทรภรณ์ สัจจนดำรงค์ [16] พัฒนาเครื่องมือช่วยในการวางแผนการผลิตของโรงงานผลิตเสื้อผ้าสำเร็จรูปแห่งหนึ่งโดยใช้หลักการการหาค่าที่เหมาะสมที่สุดด้วยวิธีการโปรแกรมเชิงเส้นตรง และเรียกเครื่องมือนี้ว่า MPO (Master Planning Optimizer) ซึ่งประมวลผลด้วย Solver Engine ผ่านโปรแกรม Microsoft Excel เพื่อสร้างตารางการผลิตหลักที่ทำให้เกิดต้นทุนรวมต่ำที่สุด บนเงื่อนไขที่จำกัดทั้งในส่วนของการผลิตและจำนวนวัตถุดิบที่มีอยู่ และสามารถนำตารางการผลิตหลักที่ได้นี้ไปใช้วางแผนความต้องการด้านวัสดุต่อไป

อภิชาติ สอนกลิ่น [17] พัฒนาการจัดตารางการผลิตพลาสติกขึ้นรูปของเครื่องจักรแบบขนานที่เหมือนกันภายใต้อัตราการผลิตที่ต่างกัน เพื่อเพิ่มประสิทธิภาพการวางแผน โดยขั้นตอนแรกเป็นการแก้ปัญหาการจัดตารางการผลิตหลัก กำหนดตารางการผลิตในแต่ละขั้นตอนให้ส่งสินค้าได้ทันตามกำหนด และขั้นตอนที่สองเป็นการจัดลำดับงานโดยวิธี Branch and Bound สามารถลดจำนวนรายการที่เลื่อนส่งลูกค้า Split Order และลดต้นทุนในการขนส่งได้

Anders [18] ศึกษาการจัดตารางการผลิตหลักเปรียบเทียบกับการจัดตารางความต้องการของวัสดุ โดยการกำหนดงานของการจัดตารางการผลิตหลักสามารถกระจายความต้องการตามกำลังการผลิตและแผนกำหนดการใช้ได้ในลักษณะการผลิตตามสั่งตามแผนการผลิตระยะยาว

Corominas และคณะ [19] ได้พัฒนา Model ที่รวมทางเลือกต่าง ๆ มาใช้ในการจัดการงานและจัดตารางการผลิตของชุดงานให้กับกลุ่มของคนงาน ซึ่งประสิทธิภาพมีผลต่อความสามารถของคนงานโดยมีวัตถุประสงค์ในการทำให้เวลาในการผลิตสั้นที่สุด

Schwindt และคณะ [20] ได้คิดค้น Priority Rule Based Method ในการแก้ปัญหาการจัดตารางการผลิตสารเคมีแบบเป็น Batch โดยนำข้อจำกัดของความพร้อมของวัตถุดิบและความสามารถในการเก็บมาพิจารณาพร้อมกัน โดยมีวัตถุประสงค์ในการปรับปรุงการจัดตารางการผลิตโดยทำให้ Makespan มีค่าน้อยสุด ผลจากการศึกษาพบว่าสามารถรับมือกับการจัดตารางการผลิตที่เป็น Batch ได้เป็นอย่างดี

ดังนั้นจึงยังไม่พบงานวิจัยที่มีการประยุกต์ใช้เครื่องมือต่าง ๆ ดังกล่าวข้างต้น ในการวางแผนผลิตสำหรับอุตสาหกรรมการผลิตยา ที่ทำให้ทราบลำดับความสำคัญของการเลือกผลิตรายการยาในแต่ละขั้นตอนให้เกิดความสอดคล้องกัน เพื่อควบคุมให้ส่งมอบได้ทันกำหนดเวลา และสามารถคำนวณกำลังการผลิตที่เหลืออยู่ได้