บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

การวางแผนสำหรับโครงงาน (Project Planning)

โครงงานประกอบด้วยงานหลาย ๆ งานซึ่งมีความสัมพันธ์กัน โดยมีลำดับการดำเนินงานที่จะต้องเสร็จสิ้นไปตามลำดับก่อนหลัง กล่าวคือ งานบางอย่างจะเริ่มต้นไม่ได้จนกว่างานอื่นบางงานจะแล้วเสร็จลงก่อน และมีงานอีกหลาย ๆ งานที่อาจทำพร้อมกันไปได้ ในโครงงานที่ประกอบด้วยงานหลาย ๆ งานต่างชนิดกันและมีความซับซ้อน การเตรียมงานและการจัดรูปโครงงานให้เข้าใจง่ายจะทำให้การวางแผนและการควบคุมโครงงานเป็นไปได้อย่างมีประสิทธิภาพ

1. ภูมิหลังของการวางแผนสำหรับโครงงาน

ในระยะเริ่มแรก การวางแผนสำหรับควบคุมโครงงานนิยมใช้ Gantt Chart ที่คิดขึ้นโดย H.L. Gantt ซึ่งใช้วิธีเขียนเป็นเส้นตรงหรือเป็นแถบอยู่ในแนวระนาบแสดงถึงกิจกรรมทั้งหลายของหน่วยงานต่าง ๆ บนหน่วยเวลาเดียวกัน มีการกำหนดเวลาเริ่มต้นและสิ้นสุด ในปัจจุบันนี้ยังคงนิยมใช้ Gantt Chart อยู่มากในงานควบคุมการผลิต

อย่างไรก็ตามแผนภูมิในลักษณะนี้ยังมีข้อบกพร่องอยู่ที่ไม่สามารถแสดงความสัมพันธ์ของงานย่อย ๆ ในโครงงานได้ เช่น ไม่สามารถแสดงให้เห็นชัดเจนว่างานใดจะต้องสิ้นสุดลงก่อนที่อีกงานจะเริ่มต้นได้ หรือจะล่าช้าได้มากแค่ไหนจึงจะไม่ส่งผลให้งานอื่น ๆ ต้องล่าช้าไปด้วย หรืองานใดจะสามารถทำไปพร้อม ๆ กับงานอื่นได้บ้าง ฯลฯ ถ้าโครงงานมีความซับซ้อนมาก การใช้ Gantt Chart ในการควบคุมโครงงานจะประสบกับความยุ่งยากและไม่เหมาะสม จึงเกิดความจำเป็นต้องมีเทคนิคใหม่ที่เป็นระบบ (Systematic) และมีประสิทธิภาพกว่า เพื่อใช้ในการวางแผนและควบคุมโครงงาน

ด้วยเหตุนี้ จึงมีการพัฒนาเทคนิคใหม่ในการวางแผนงาน การกำหนดขั้นตอนการทำงาน และการควบคุมโครงงานขึ้นสองอย่างคือ วิธีสายงานวิกฤติ (Critical Path Method) ซึ่งใช้คำย่อว่า CPM และเทคนิคการประเมินค่าโครงการและประเมินผล (Program Evaluation & Review Technique) ซึ่งใช้คำย่อว่า PERT เทคนิคทั้งสองอย่างนี้ถูกพัฒนาขึ้นโดยหน่วยงานคนละหน่วยงานในระยะเวลาเกือบจะพร้อม ๆ กัน (พ.ศ.2499 – 2501) PERT พัฒนาขึ้นโดยคณะกรรมาธิการโครงงานพิเศษของกองทัพเรือสหรัฐโดยร่วมมือกับบริษัท Booz, Allen & Hamilton ซึ่งเป็นบริษัทที่ปรึกษาเกี่ยวกับงานบริหาร การพัฒนานี้มีจุดมุ่งหมายที่จะนำไปใช้ในการวางแผนและควบคุมโครงการต่อเรือดำน้ำโพลาริส ซึ่งเป็นเรือดำน้ำพลังงานนิวเคลียส์ติดขีปนาวุธ การใช้ PERT ในการควบคุมโครงการนี้ก่อให้เกิดผลดีอย่างมากโดยทำให้สามารถต่อเรือดำน้ำลำนี้เสร็จภายในระยะเวลาเพียง 2 ปี ซึ่งนับว่าเร็วมากสำหรับโครงการใหญ่ ๆ ขนาดนี้ ส่วน CPM พัฒนาขึ้นโดยบริษัท E.I. du Pont de Nemours & Company เพื่อใช้ในการควบคุมโครงงานก่อสร้างซึ่งช่วยให้บริษัทนี้สามารถลดต้นทุนได้ถึงหนึ่งล้านเหรียญสหรัฐฯ หลังจากนั้นมาเทคนิคทั้งสองก็ถูกนำไปใช้ในวงการต่าง ๆ อย่างกว้างขวาง

ถึงแม้ว่าเทคนิคทั้งสองนี้จะพัฒนาขึ้นโดยหน่วยงานคนละแห่ง แต่มีวิธีการส่วนใหญ่เหมือนกัน คือ เป็นวิธีการควบคุมโครงงานด้วยการกำหนดเวลาให้งานต่าง ๆ ในโครงงานแล้วเสร็จภายในเวลาที่กำหนด ข้อแตกต่างของเทคนิคทั้งสองนี้อยู่ที่ข้อมูลที่ใช้ในการวิเคราะห์โครงงานเพื่อวางแผน กล่าวคือ ถ้าสามารถกำหนดเวลาการทำงานของแต่ละงานได้ค่อนข้างแน่นอน (Deterministic) เราเรียกวิธีการที่ใช้ว่า CPM แต่ถ้าเวลาการทำงานของแต่ละงานไม่แน่นอนโดยที่สามารถจะกำหนดหาค่าความเป็นไปได้ของเวลาเหล่านั้นได้(Probabilistic) เราเรียกวิธีการที่ใช้ว่า PERT อย่างไรก็ตาม ในปัจจุบันนี้ PERT และ CPM ถูกถือเป็นเทคนิคเดียวกัน และถูกเรียกรวม ๆ ว่า PERT – CPM

2. ขั้นตอนการดำเนินการของการวางแผนสำหรับโครงงาน

ขั้นตอนการวางแผนสำหรับโครงงานมีดังนี้

2.1 ศึกษาโครงงานที่เกี่ยวข้องเพื่อกำหนดชนิดของงาน ปริมาณงานและขั้นตอนการทำงานทั้งหมดด้วยวิธีการแบ่งแยกงานที่จะทำให้ชัดเจน และเชื่อมโยงความสัมพันธ์ของงานแต่ละงานตามลำดับก่อนหลัง เมื่อแยกขั้นตอนของงานที่จะต้องทำได้เด่นชัดแล้ว ก็นำมาผูกเป็นโครงข่ายของงาน (Job Network) ซึ่งดูแล้วเข้าใจได้ง่าย วิธีการสร้างโครงข่ายของงานนี้จะได้กล่าวภายหลัง

2.2 จากโครงข่ายของงานที่ได้สร้างขึ้นโดยถูกต้องแล้ว ลำดับต่อไปคือการกำหนดระยะเวลาที่ใช้ในการทำงานแต่ละขั้นตอน ซึ่งจะต้องอาศัยข้อมูลในอดีตประสบการณ์และความชำนาญ เมื่อกำหนดเวลาการทำงานของแต่ละขั้นตอนแล้ว ขั้นต่อไปคือการกำหนดหาส่วนของงานที่เป็นงานวิกฤติ (Critical Activities) ซึ่งอยู่ในสายงานวิกฤติ (Critical Path)

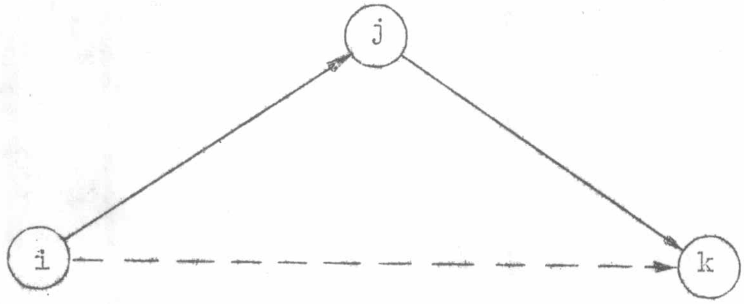
2.3 ขั้นสุดท้ายคือการปรับปรุงโครงงานที่ได้วางไว้แล้ว และควบคุมให้ดำเนินงานได้ตามเป้าหมายภายในช่วงเวลาที่กำหนด การปรับปรุงโครงงานอาจทำได้หลายวิธี โดยมีจุดมุ่งหมายที่จะทำให้โครงงานเสร็จสิ้นไปได้เร็วขึ้น อาจทำได้โดยการทำงานล่วงเวลาหรือการเพิ่มกำลังคนและเครื่องจักรกลให้มากขึ้น

ขั้นตอนการดำเนินงานที่ได้กล่าวมานี้ เป็นขั้นตอนที่ใช้ในการวิเคราะห์การดำเนินงานตามรูปแบบการวางแผนสำหรับโครงงาน ซึ่งใช้ได้ทั้งกับ PERT และ CPM

3. โครงข่ายของงาน (Job Network)

การวางแผนโครงงานโดยใช้เทคนิคนี้ ใช้การสร้างโครงข่าย (Network) ของปัญหาโดยประกอบด้วยไดอาแกรมลูกศร (Arrow Diagram) แทนที่จะใช้รูปแบบทางคณิตศาสตร์เหมือนเทคนิคอื่น ๆ ของการวิจัยการดำเนินงาน (Operations Research) ในโครงข่ายของงานมีองค์ประกอบสำคัญเพียงสามประการคือ

ก. จุดยอด (Nodes) ซึ่งเป็นรูปวงกลมมีตัวเลขใด ๆ อยู่ภายใน เช่น ตัวเลข i แทนจุดแสดงเวลาของเหตุการณ์ (Event) j ซึ่งอาจหมายถึงเวลาเริ่มต้นของงาน j-k หรือเวลาสิ้นสุดของงาน i-j (ดูรูปที่ 2.1)



รูปที่ 2.1 ไดอาแกรมลูกศรอย่างง่าย ๆ ที่ประกอบด้วย จุดยอด เส้นที่มีลูกศรและเส้นประที่มีลูกศร

ข. เส้นที่มีลูกศร (Arrows) แทนงานแต่ละขั้นตอน

ค. เส้นประที่มีลูกศร (Dummy Arrows) แทนงานสมมติ เวลาที่ใช้ในขั้นตอนของงานสมมตินี้จะมีค่าเป็นศูนย์

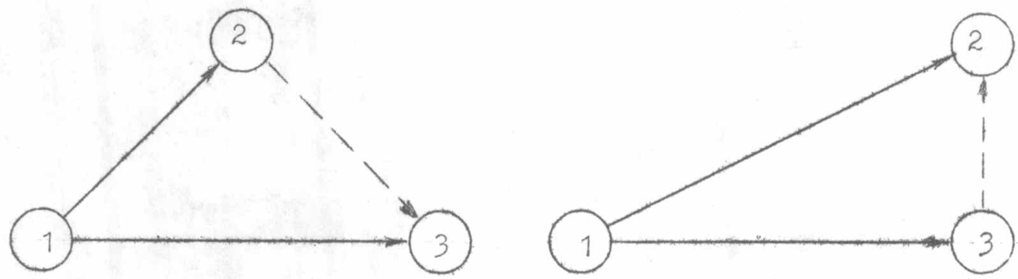
ไดอาแกรมลูกศรจะแสดงให้เห็นความสัมพันธ์และความต่อเนื่องของงานต่าง ๆ ในโครงงาน หัวลูกศรของเส้นที่มีลูกศร (Arrows) ซึ่งแทนความหมายของงานเป็นสิ่งแสดงทิศทางความก้าวหน้าของงาน ความต่อเนื่องของงานต่าง ๆ ในโครงงานถูกกำหนดโดยใช้เหตุการณ์ (Events) ซึ่งแสดงจุดสิ้นสุดของงานและจุดเริ่มต้นของงานใหม่ที่ถัดไป งานที่เริ่มต้นจากเหตุการณ์อันหนึ่งจะเริ่มต้นไม่ได้จนกว่างานที่มาสิ้นสุดที่เหตุการณ์นั้นจะเสร็จสิ้นเรียบร้อยลงเสียก่อน ความยาวของเส้นที่มีลูกศรไม่จำเป็นต้องเป็นส่วนสัดกับระยะเวลาที่ใช้ในการทำงาน และเส้นนั้นไม่จำเป็นต้องเป็นเส้นตรงเสมอไป

ในการเขียนโครงข่ายที่สมบูรณ์ สิ่งที่จะลืมไม่ได้คือ ความสัมพันธ์ของงานแต่ละงานในโครงงาน งานใดที่จะต้องทำก่อน งานใดที่จะต้องทำภายหลัง และงานใดที่สามารถจะทำไปพร้อม ๆ กับงานอื่นได้ การวิเคราะห์อย่างรอบคอบจึงจะทำให้การวางแผนสำหรับโครงงานเป็นไปได้อย่างถูกต้อง

กฎเกณฑ์การสร้างโครงข่ายของโครงงานโดยสรุป มีดังนี้

1. งานแต่ละงานในโครงงานจะแสดงด้วยเส้นที่มีลูกศร เส้นที่มีลูกศรหนึ่งเส้นแทนงานได้เพียงหนึ่งงานเท่านั้น เส้นที่มีลูกศรหนึ่งเส้นจะแทนงานสองงานหรือเส้นที่มีลูกศรสองเส้นจะแทนงานงานเดียวไม่ได้ ในกรณีที่มีงานที่สามารถแบ่งออกเป็นงานย่อย ๆ หลายงานได้ อาจใช้เส้นที่มีลูกศรแสดงงานย่อยแต่ละงานได้โดยเส้นที่มีลูกศรหนึ่งเส้นต่องานย่อยหนึ่งงาน

2. งานสองงานที่เริ่มต้นไปได้พร้อมกันและทำไปได้พร้อมกัน จะใช้เหตุการณ์เริ่มต้นและเหตุการณ์สุดท้ายเดียวกันไม่ได้ ต้องใช้งานสมมติมาช่วยในการเขียนรูปโครงข่ายของงานดังแสดงไว้ในรูปที่ 2.2



รูปที่ 2.2 การเขียนโครงข่ายของงานสองงานที่เริ่มต้นและสิ้นสุดพร้อมกัน

3. ในการเขียนโครงข่ายของงานให้มีลำดับความต่อเนื่องอย่างถูกต้อง จะต้องตอบปัญหาต่อไปนี้ให้ได้ทุกครั้งที่จะเพิ่มงานใด ๆ ในโครงข่าย

ก. งานใดบ้างที่ต้องทำให้เสร็จสิ้นก่อนงานนี้

ข. งานใดบ้างที่จะต้องทำต่อจากงานนี้

ค. งานใดบ้างที่สามารถทำไปพร้อมกับงานนี้

4. การคำนวณเวลาที่ใช้ในการทำงาน

หลังจากสร้างโครงข่ายของงานได้แล้ว ขั้นต่อไปคือการกำหนดระยะเวลาที่ใช้ในการทำงานของงานแต่ละงานในโครงงาน ถ้าเป็นการกำหนดเวลาการทำงานของงาน ซึ่งกำหนดได้แน่นอนโดยรู้แน่ชัดจากข้อมูลในอดีตและประสบการณ์โดยไม่มีทางจะแปรเปลี่ยนได้ ก็สามารถกำหนดเวลาการทำงานของงานแต่ละงานได้ทันที ซึ่งเป็นลักษณะของ CPM แต่โดยส่วนมากแล้ว เวลาการทำงานของงานแต่ละงานมักจะไม่สามารถกำหนดให้แน่นอนลงได้ การคำนวณหาเวลาที่ใช้ในการทำงานจึงต้องใช้วิธีการของ PERT

ถึงแม้ว่าเวลาที่ใช้ในการทำงานโดยทั่ว ๆ ไปจะมีการเปลี่ยนแปร แต่จะพบได้ว่ามีเวลาที่ใช้ในการทำงานอยู่สามค่าที่น่าสนใจ คือ

1. เวลาน้อยที่สุดที่ใช้ในการทำงาน (Most Optimistic Time) ซึ่งเป็นเวลาที่มีโอกาสน้อยที่จะเกิดขึ้น ตัวอย่างเช่น มีโอกาสที่จะเป็นไปได้เท่ากับ 1 ใน 100 เป็นต้น มีโอกาสเกิดขึ้นได้เมื่อทุกสิ่งที่เกี่ยวข้องกับการทำงานอำนวยให้และเป็นไปด้วยดีอย่างยิ่ง

2. เวลามากที่สุดที่ใช้ในการทำงาน (Most Pessimistic Time) ซึ่งเป็นเวลาที่มีโอกาสที่จะเกิดขึ้นน้อย จะเกิดขึ้นได้เมื่อทุกสิ่งที่เกี่ยวข้องกับการทำงานเกิดผิดพลาดอย่างโชคร้ายที่สุดเท่านั้น

3. เวลาโดยส่วนมากที่ใช้ในการทำงาน (Most Likely Time) คือจำนวนเวลาที่ใช้เสมอในการทำงานภายใต้สภาวะการทำงานปกติ

โดยที่งานแต่ละงานมีรูปแบบความน่าจะเป็นไปได้ (Probabilistic Model) ของเวลาการทำงานที่แตกต่างกัน การพิจารณาหารูปแบบความน่าจะเป็นไปได้ของงานทุก ๆ งานเป็นการยุ่งยากและเสียเวลามาก ดังนั้นเพื่อให้การหาค่าเวลาที่จะใช้ในการทำงานแต่ละงานง่ายขึ้น จึงต้องใช้วิธีตั้งสมมติฐานสำหรับรูปแบบความน่าจะเป็นไปได้ของงานต่าง ๆ ขึ้นให้เป็นรูปแบบการกระจายแบบเดียวกัน ซึ่งจะต้องอยู่ในบรรทัดฐานต่อไปนี้

ก. มีความน่าจะเป็นไปได้น้อยที่เวลาในการทำงานจะเป็นเวลาน้อยที่สุดที่ใช้ในการทำงาน (Most Optimistic Time)

ข. มีความน่าจะเป็นไปได้น้อยที่เวลาในการทำงานจะเป็นเวลามากที่สุดที่ใช้ในการทำงาน (Most Pessimistic Time)

ค. มีจุดยอด (Node) เดียว คือ มีการกระจายของข้อมูลที่มีความถี่สูงสุดที่เวลาโดยส่วนมากที่ใช้ในการทำงาน (Most Likely Time) เพียงจุดเดียว และอาจจะอยู่ที่จุดใดก็ได้ระหว่างเวลาที่มากที่สุดและเวลาที่น้อยที่สุดที่ใช้ในการทำงาน

จากบรรทัดฐานดังกล่าวจึงได้สมมติฐานว่า งานทุกงานในโครงงานมีการกระจายแบบเบต้า (Beta Distribution) เพราะการกระจายแบบเบต้ามีคุณสมบัติตามบรรทัดฐานข้างต้นอย่างครบถ้วน ดังนั้นการคำนวณหาค่าเวลาที่คาดว่าจะใช้ในการทำงาน (t_e) และค่าแปรเปลี่ยน (Variance) สามารถทำการคำนวณได้จากสมการต่อไปนี้

$$t_e = \frac{(a + 4m + b)}{6} \qquad \text{...............} \; 2.1$$

$$V = \left(\frac{b - a}{6}\right)^2 \qquad \text{...............} \; 2.2$$

เมื่อ t_e คือ เวลาที่คาดว่าจะใช้ในการทำงาน

a คือ เวลาน้อยที่สุดที่ใช้ในการทำงาน

m คือ เวลาโดยส่วนมากที่ใช้ในการทำงาน

b คือ เวลามากที่สุดที่ใช้ในการทำงาน

V คือ การแปรเปลี่ยน

ต่อจากนั้น ก็สามารถทำการหางานวิกฤติและสายงานวิกฤติได้โดยวิธีของ CPM

5. การคำนวณหาสายงานวิกฤติ

PERT - CPM มีประโยชน์ในการควบคุมงานเพราะแสดงกำหนดวันเริ่มต้นงานและวันเสร็จงานไว้อย่างชัดเจน การคำนวณกำหนดวันดังกล่าวนี้ทำได้โดยใช้วิธีการทางคณิตศาสตร์อย่างง่าย ๆ ผลจากการคำนวณนี้จะทำให้รู้ว่า งานใดเป็นงานวิกฤติ (Critical Activities) และงานใดเป็นงานไม่วิกฤติ (Non-critical Activities) ถ้างานซึ่งเป็นงานวิกฤติล่าช้าไปจะมีผลให้งานทั้งโครงงานล่าช้าไปด้วย ส่วนงานไม่วิกฤติมีช่วงระหว่างกำหนดเวลาเริ่มต้นเร็วที่สุด (Earliest Start) และเวลาสิ้นสุดล่าสุดมากกว่าเวลาที่จะใช้ในการทำงาน จึงทำให้งานไม่วิกฤติมีเวลายืดหยุ่น

การคำนวณเพื่อหางานวิกฤติ แบ่งออกเป็นสองส่วนคือ

1. ส่วนที่เป็นการกำหนดเวลาไปข้างหน้า (Forward Pass) ซึ่งทำการคำนวณจากจุดยอดเริ่มต้น และเลื่อนไปทำที่จุดยอดถัดไปเรื่อย ๆ จนถึงจุดยอดสุดท้าย โดยคำนวณหาเวลาเริ่มต้นเร็วที่สุดของจุดยอดทุกจุดในโครงข่าย ผลจากการคำนวณด้วยการใช้สมการข้างล่าง จะแสดงในรูป □ ที่ทุก ๆ จุดยอดในโครงข่าย

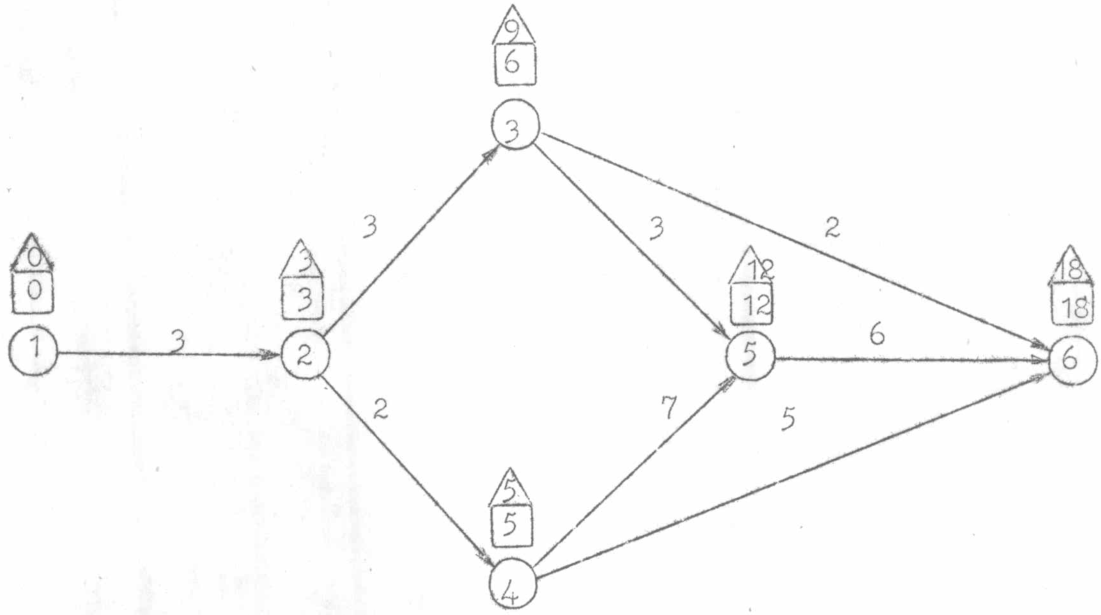
$$ES_j = \max_i \left\{ ES_i + D_{ij} \right\} \quad \cdots\cdots\cdots\cdots \quad 2.3$$

เมื่อ ES_j คือ เวลาเริ่มต้นเร็วที่จุดยอด j

ES_i คือ เวลาเริ่มต้นเร็วที่สุดของจุดยอด i ซึ่งเป็นจุดเริ่มต้นของงานใด ๆ ที่มาสิ้นสุดที่ j

D_{ij} คือ เวลาที่ใช้ในการทำงานของงาน i-j ใด ๆ ที่มาสิ้นสุดที่จุดยอด j

ดังนั้น เวลาเริ่มต้นเร็วที่สุดของจุดยอดใด ๆ จะหมายถึงค่าเวลาสูงสุดที่คิดจากงานทั้งหลายที่ร่วมใช้จุดยอด j เป็นจุดสิ้นสุดของงานโดยคิดรวมเวลาตั้งแต่เริ่มโครงงาน



รูปที่ 2.3 โครงข่ายของงานและการหางานวิกฤติโดยที่เวลาที่ใช้ในการทำงานของแต่ละงานมีค่าเท่ากับตัวเลขที่แสดงไว้บนเส้นที่มีลูกศรแทนงานนั้น ๆ

2. ส่วนที่เป็นการกำหนดเวลาย้อนหลัง (Backward Pass) ซึ่งเริ่มทำการคำนวณจากจุดยอดสุดท้ายของโครงงาน แล้วเลื่อนย้อนมาที่จุดยอดถัดมาเรื่อย ๆ จนกระทั่งถึงจุดยอดเริ่มต้นของโครงงานในที่สุด ผลจากการคำนวณจากสมการ 2.4 เป็นค่าเวลาสิ้นสุดล่าสุดของแต่ละจุดยอด ซึ่งจะแสดงไว้ในรูป △ ที่จุดยอดแต่ละจุด

$$LF_i = \min_j \left\{ LF_j - D_{ij} \right\} \qquad \ldots\ldots\ldots\ldots 2.4$$

เมื่อ LF_i คือ เวลาสิ้นสุดล่าสุดของจุดยอด i

LF_j คือ เวลาสิ้นสุดล่าสุดของจุดยอด j ซึ่งเป็นจุดยอดสิ้นสุดของงานใด ๆ ที่เริ่มต้นจากจุดยอด i ที่กำลังคำนวณ

D_{ij} คือ เวลาที่ใช้ในการทำงานของงาน i-j ใด ๆ ที่เริ่มต้นจากจุดยอด i

เวลาสิ้นสุดล่าสุดของจุดยอดใด ๆ จึงหมายถึงค่าเวลาน้อยที่สุดที่คิดจากงานทั้งหลาย ที่เริ่มต้นจากจุดยอด i โดยคิดลดเวลาตั้งแต่เวลาสิ้นสุดของโครงงาน

ผลจากการคำนวณส่วนที่เป็นการกำหนดเวลาไปข้างหน้าและการกำหนดเวลาย้อนห... นี้ ทำให้สามารถหางานวิกฤติได้ กล่าวคือ งาน i-j ใด ๆ จะเป็นงานวิกฤติซึ่งอยู่ในสา... งานวิกฤติ เมื่อ

1. $ES_i = LF_i$

2. $ES_j = LF_j$

3. $ES_j - ES_i = LF_j - LF_i = D_{ij}$

เมื่อดูโครงข่าย (ดูรูปที่ 2.3) จะเห็นได้โดยง่ายว่าค่าตัวเลขใน □ และ △ ที่เหตุการณ์เริ่มต้นของงานวิกฤติจะมีค่าเท่ากัน เช่นเดียวกับที่ตัวเลขใน □ และ △ ของเหตุการณ์สิ้นสุดมีค่าเท่ากัน และผลต่างระหว่างตัวเลขใน □ (หรือ △) ที่เหตุการณ์ สิ้นสุดกับตัวเลขใน □ (หรือ △) ที่เหตุการณ์เริ่มต้นมีค่าเท่ากับเวลาที่ใช้ในการทำงาน ของงานนั้น ๆ

จุดยอดที่มีค่า ES = LF เรียกว่า "จุดยอดวิกฤติ (Critical Node)"

6. <u>การกำหนดหาความยืดหยุ่นของงาน</u>

เนื่องจากงานวิกฤติเป็นงานที่ไม่อาจเปลี่ยนแปลงกำหนดเวลาการทำงาน ไม่ว่าจะเป็นการเริ่มต้นหรือการสิ้นสุดโดยไม่ส่งผลกระทบต่อโครงงานทั้งหมดได้ งานใน สายงานวิกฤติจึงไม่มีความยืดหยุ่นซึ่งหมายความว่าไม่มีเวลาเหลือสำหรับขยับเวลาเริ่มต้นหรือ สิ้นสุดให้เปลี่ยนแปลงเป็นอื่นได้

ความยืดหยุ่นของงาน (Float) จึงเป็นเวลาส่วนที่งานซึ่งไม่วิกฤติ (Non-Critical Activities) สามารถเลื่อนไปให้ช้าหรือเร็วขึ้นได้ภายในขอบเขตที่เป็นไปได้ ความยืดหยุ่นของงานจึงมีประโยชน์ต่อการวางแผนงานของสายงานไม่วิกฤติกล่าวคือ อาจ นำกำลังคนและเครื่องมือเครื่องจักรของงานที่มีช่วงว่างเพราะความยืดหยุ่นไปใช้ในงานอื่นได้

เป็นการลดเวลาว่างของกำลังคนและเครื่องจักรซึ่งเป็นการลดต้นทุนได้วิธีหนึ่ง

ความยืดหยุ่นของงานแบ่งออกโดยสรุปได้ 4 ประการคือ

1. Total Float
2. Interference Float
3. Free Float
4. Independent Float

การที่จะกำหนดหาความยืดหยุ่นของงานต่าง ๆ ได้ จะต้องสามารถกำหนดเวลาเริ่มต้นล่าสุด (Latest Start) และเวลาสิ้นสุดเร็วสุด (Earliest Finish) ของงานต่าง ๆ ให้ได้เสียก่อนโดยใช้สมการ 2.5 และ 2.6 เพื่อหาค่าเวลาเริ่มต้นล่าสุดและเวลาสิ้นสุดเร็วสุดของงาน i-j ใด ๆ

$$LS_{ij} = LF_j - D_{ij} \qquad \ldots\ldots\ldots\ldots\ 2.5$$

$$EF_{ij} = ES_i + D_{ij} \qquad \ldots\ldots\ldots\ldots\ 2.6$$

เมื่อ LS_{ij} คือ เวลาเริ่มต้นล่าสุด (Latest Start)

EF_{ij} คือ เวลาสิ้นสุดเร็วสุด (Earliest Finish)

LF_j คือ เวลาสิ้นสุดล่าสุด (Latest Finish) ของเหตุการณ์สิ้นสุดของงาน i-j

ES_i คือ เวลาเริ่มต้นเร็วสุด (Earliest Start) ของเหตุการณ์เริ่มต้นของงาน i-j

D_{ij} คือ เวลาที่ใช้ในการทำงานของงาน i-j

สมการ 2.7 - 2.10 ใช้สำหรับการคำนวณค่าความยืดหยุ่นชนิดต่าง ๆ

- Total Float (TF_{ij})

$$TF_{ij} = (LF_j - ES_i) - D_{ij} \qquad \ldots\ldots\ 2.7$$

- Interference Float (IFF_j)

$$IFF_j = LF_j - ES_j \qquad \text{...........2.8}$$

- Free Float (FF_{ij})

$$FF_{ij} = (ES_j - ES_i) - D_{ij} \qquad \text{...........2.9}$$

- Independent Float (IF_{ij})

$$IF_{ij} = (ES_{jk} - LF_{hi}) - D_{ij} \qquad \text{..........2.10}$$

เมื่อ ES_{jk} คือ เวลาเริ่มต้นเร็วสุดของงานต่อไป

LF_{hi} คือ เวลาสิ้นสุดล่าสุดของงานก่อน

การคำนวณหาความยืดหยุ่น (Float) นี้ จะสามารถแสดงว่างานใดเป็นงานวิกฤติได้เช่นกัน โดยงานวิกฤติจะมีค่าความยืดหยุ่นทุกชนิดเป็นศูนย์เสมอ

ในการหาความยืดหยุ่นของงานชนิดต่างจำเป็นจะต้องกำหนดหาค่าเวลาเริ่มต้นและสิ้นสุดของงานโดยมีหลักการดังนี้

1. งานทุกชนิดที่มีจุดเริ่มต้นจากจุดยอดเดียวกันจะมีเวลาเริ่มต้นเร็วสุดเท่ากัน
2. งานทุกงานที่สิ้นสุดที่จุดยอดเดียวกัน จะมีเวลาสิ้นสุดล่าสุดเท่ากัน
3. ค่าเวลาเริ่มต้นเร็วสุดของงานจะต้องไม่มากกว่าค่าเวลาสิ้นสุดล่าสุดที่จุดยอดเดียวกัน
4. ค่าเวลาสิ้นสุดเร็วสุดของงานที่จุดยอดใด ๆ จะต้องไม่มากกว่าค่าเวลาสิ้นสุดล่าสุดของจุดยอดนั้น
5. เวลาเริ่มต้นของงานใด ๆ จะต้องมาก่อนเวลาสิ้นสุดของงานนั้น ๆ
6. ค่า Free Float จะมากกว่า Total Float ไม่ได้
7. ค่า Independent Float จะมากกว่า Free Float ไม่ได้
8. งานวิกฤติจะต้องมีค่าความยืดหยุ่นทุกชนิดเป็นศูนย์เสมอ
9. Free Float เป็นค่าผลต่างระหว่าง Total Float และ Interference Float

7. ความน่าจะเป็นไปได้ของโครงงาน

ความน่าจะเป็นไปได้ของโครงงานสามารถหาได้โดยการคำนวณค่าความน่าจะเป็นไปได้ของเหตุการณ์ ที่จะเกิดขึ้นได้ตามกำหนดในโครงงาน ให้ μ_i เป็นเวลาเริ่มต้นเร็วที่สุดของเหตุการณ์ i เพราะว่าเวลาการทำงานของงานต่าง ๆ ที่รวมกันเป็นตัวแปรสุ่ม (random variables) μ_i จึงเป็นตัวแปรสุ่มด้วย และสมมติว่างานทุกงานในโครงงานเป็นอิสระในทางสถิติ (Statistically Independent)

ถ้ามีเพียงสายงานเดียวระหว่างเหตุการณ์เริ่มต้นและเหตุการณ์ i $E\{\mu_i\}$ หาได้จากผลรวมของ t_e ของงานต่าง ๆ ในสายงานนั้น และ $V\{\mu_i\}$ คือผลรวมของค่าความแปรเปลี่ยน (V) ของงานเหล่านั้น

ถ้ามีมากกว่าหนึ่งสายงานระหว่างเหตุการณ์เริ่มต้นและเหตุการณ์ i $E\{\mu_i\}$ และ $V\{\mu_i\}$ มีค่าเท่ากับผลรวมของ t_e และ V ของงานในสายงานที่ใช้เวลาการทำงานมากที่สุด ถ้ามีสายงานที่มีค่า $E\{\mu_i\}$ เท่ากัน จะเลือกสายงานที่มีค่า $V\{\mu_i\}$ มากที่สุดเพราะสายงานนั้นมีความไม่แน่นอนมากที่สุด

เนื่องจาก μ_i เป็นผลรวมของตัวแปรสุ่มที่เป็นอิสระ และอาศัยทฤษฎีจำกัดเข้าสู่ศูนย์กลาง (Central Limit Theorem) จึงถือว่า μ_i มีการกระจายแบบปกติ (Normal Distribution) โดยมี $E\{\mu_i\}$ เป็นค่าเฉลี่ยและ $V\{\mu_i\}$ เป็นค่าความแปรเปลี่ยน ความน่าจะเป็นไปได้ที่เหตุการณ์ i จะเกิดขึ้นทันตามกำหนดเวลา ST_i (ซึ่งกำหนดโดยผู้กำหนดแผนงาน) สามารถหาได้ดังนี้

$$P\left\{\mu_i \leq ST_i\right\} = P\left\{\frac{\mu_i - E\{\mu_i\}}{\sqrt{V\{\mu_i\}}} \leq \frac{ST_i - E\{\mu_i\}}{\sqrt{V\{\mu_i\}}}\right\} = P\left\{z \leq K_i\right\}$$

เมื่อ Z มีการกระจายแบบปกติมาตรฐาน (Standard Normal Distribution) ที่มีค่าเฉลี่ยเป็นศูนย์และความแปรเปลี่ยนเป็น 1 และ

$$K_i = \frac{ST_i - E\{\mu_i\}}{\sqrt{V\{\mu_i\}}}$$

โปรแกรมพลวัต (Dynamic Programming)

ในบรรดาเทคนิคทางคณิตศาสตร์ทั้งหลายที่นำมาใช้ในการวิจัยการดำเนินงาน (Operations Research) นั้น ดูเหมือนว่าโปรแกรมพลวัตมีแนวความคิดง่ายที่สุด แต่ก็เป็นวิธีการที่นำไปประยุกต์ใช้ได้ยากวิธีหนึ่ง สาเหตุที่ทำให้การนำโปรแกรมพลวัตไปประยุกต์เป็นเรื่องยุ่งยาก ก็คือ ขาดการจัดตั้งรูปแบบ (Model Formulation) ที่แน่นอนและการแก้ปัญหาที่ตรงจุด ผู้ใช้โปรแกรมพลวัตจำเป็นต้องมีความเข้าใจปัญหาและเทคนิคที่เกี่ยวข้องทั้งหมดก่อนที่จะเริ่มต้นแก้ปัญหา ไม่มีกฎตายตัวที่จะทำตามได้ง่าย ๆ อันใดอันหนึ่งที่จะนำไปสู่การจัดตั้งปัญหา (Problem Formulation) ที่ถูกต้องได้เสมอ ดังนั้น วิธีการที่จะอธิบายแนวความคิดของโปรแกรมพลวัตได้ดี คือการแสดงด้วยตัวอย่างการแก้ปัญหาโดยใช้โปรแกรมพลวัตในลักษณะต่าง ๆ

1. แนวความคิดของโปรแกรมพลวัต (Concept of Dynamic Programming)

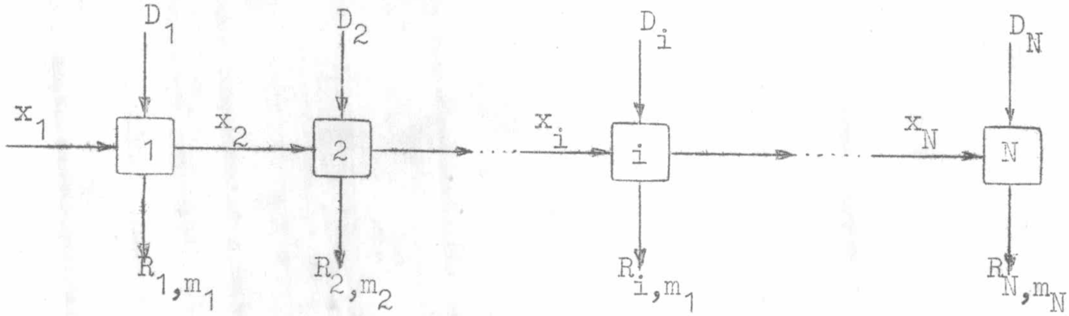
โปรแกรมพลวัตเหมาะสมสำหรับการแก้ปัญหาที่ต้องมีการตัดสินใจที่มีความสัมพันธ์กัน (Interrelated Decisions) เช่น การตัดสินใจที่ต้องทำตามลำดับ (Sequence) ซึ่งการตัดสินใจแต่ละครั้งมีอิทธิพลต่อการตัดสินใจในลำดับต่อ ๆ ไปด้วยเป็นต้น

ตัวอย่างง่าย ๆ ของปัญหาในลักษณะนี้ก็คือ ปัญหาที่เกี่ยวกับการลงทุนของบริษัทแห่งหนึ่งซึ่งมีโรงงานอยู่ N โรงงาน ที่จะต้องพิจารณาปรับปรุงโดยมีงบประมาณสำหรับการนี้ C บาท แผนการปรับปรุงของโรงงาน i (i = 1, 2, ..., N)

มีอยู่ m_i แผนการไม่ปรับปรุงเลยก็นับเป็นแผนการหนึ่งด้วย ค่าใช้จ่ายในการปรับปรุงตามแผนการ m_i (ของโรงงาน i) คือ $C_{i,\, m_i}$ ซึ่งจะทำให้ได้รับผลตอบแทนจากการปรับปรุงเป็นมูลค่า $R_{i,\, m_i}$ การตัดสินใจนี้มีวัตถุประสงค์ที่จะเลือกแผนการที่เป็นไปได้ (Feasible Plan) สำหรับแต่ละโรงงาน i ซึ่งจะทำให้ได้รับผลตอบแทนรวมสูงสุดโดยที่ค่าใช้จ่ายของการลงทุนปรับปรุงจะต้องไม่เกินงบประมาณที่มีอยู่ (C)

เมื่อใช้โปรแกรมพลวัตในการแก้ปัญหานี้ ปัญหาจะถูกแบ่งย่อยออกเป็นปัญหาย่อย (Subproblems) ซึ่งมีขนาดเล็กลงและแก้ปัญหาได้ง่ายกว่า ในโปรแกรมพลวัตเรียกปัญหาย่อยซึ่งเป็นส่วนหนึ่งของปัญหาว่า " ขั้น (Stages) " ในแต่ละขั้นจะมีการตัดสินใจอย่างน้อยหนึ่งครั้ง ตามตัวอย่างการลงทุนข้างต้น จะต้องมีการตัดสินใจเลือกทางเลือก (Alternatives) หนึ่งทางเลือกต่อโรงงานแต่ละโรงงาน ดังนั้น แต่ละโรงงานจึงถือเป็นขั้นหนึ่งของปัญหา รูปแบบ (Model) แทนโปรแกรมพลวัตที่มี N ขั้นแสดงอยู่ในรูปที่ 2.4 ตามปกติกระบวนการหาผลที่ดีที่สุด (Optimization Process) จะเริ่มต้นที่ขั้นที่ 1 เมื่อเสร็จแล้วจึงเลื่อนต่อไปยังขั้นที่ 2 และต่อไปเรื่อย ๆ จนถึงขั้นที่ N

ในแต่ละขั้น จะต้องประกอบด้วยส่วนสำคัญที่ขาดไม่ได้คือ ทางเลือก (Alternative) หรือตัวแปรการตัดสินใจ (Decision Variables) ต่าง ๆ และผลตอบแทน (Return Functions) จากแต่ละทางเลือก ในตัวอย่างเกี่ยวกับการลงทุนที่



D_i คือ ตัวแปรการตัดสินใจ (Dicision Variables) ของขั้น i

R_{i,m_i} คือ ผลตอบแทนจากการตัดสินใจในขั้นที่ i

x_i คือ ภาวะ (State) ของระบบที่ขั้นที่ i

รูปที่ 2.4 การแยกปัญหาใหญ่ออกเป็นปัญหาย่อย N ปัญหา

กล่าวถึง ทางเลือก (Alternatives) คือ แผนการปรับปรุงโรงงานซึ่งมี M_i แผน ส่วนผลตอบแทน (Return Functions) คือ ผลตอบแทน R_{i,m_i} จากทางเลือก m_i ของขั้นที่ i

ภาวะ (State) ของระบบเป็นส่วนสำคัญที่สุดของโปรแกรมพลวัต เพราะเป็นสิ่งแสดงการเชื่อมโยงระหว่างขั้นตอนต่าง ๆ ซึ่งจะทำให้การตัดสินใจที่เป็นผลลัพธ์จากการที่แยกหาผลดีที่สุดของแต่ละขั้นเป็นผลลัพธ์ที่เป็นไปได้ของปัญหาทั้งหมดโดยอัตโนมัติ ยิ่งไปกว่านั้นยังช่วยให้ทำการตัดสินใจเพื่อให้ได้ผลที่ดีที่สุดของแต่ละขั้นได้โดยไม่ต้องตรวจสอบผลอันเกิดจากการตัดสินใจในขั้นนั้นว่าจะส่งผลกระทบกระเทือนต่อการตัดสินใจในขั้นต่าง ๆ ที่ผ่านมาหรือไม่

การกำหนดภาวะในโปรแกรมพลวัตนับว่าเป็นเรื่องที่ยุ่งยากพอสมควร เพราะไม่มีวิธีการง่าย ๆ ที่ตายตัวอันใดที่จะใช้ในการกำหนดภาวะได้ ตามปกติแนวทางที่จะใช้ในการกำหนดภาวะสามารถหาได้โดยอาศัยคำถามต่อไปนี้

1. อะไรคือความสัมพันธ์ที่เชื่อมโยงขั้นต่าง ๆ

2. ข้อมูลใดที่จำเป็นต่อการทำให้การตัดสินใจในแต่ละขั้นเป็นไปได้ (Feasible) โดยไม่ต้องย้อนไปตรวจสอบความเป็นไปได้ของขั้นที่ได้ทำการตัดสินใจไปแล้ว

ในตัวอย่างที่กล่าวถึง สิ่งที่เชื่อมโยงขั้นต่าง ๆ เข้าด้วยกันก็คือ ข้อเท็จจริงที่ว่าโรงงาน (ขั้น) ต่าง ๆ จะต้องได้รับการแบ่งปันงบประมาณจากงบประมาณที่มีอยู่ C บาท จึงเป็นสิ่งที่บ่งชี้ว่า ภาวะควรจะกำหนดในรูปของการจัดสรรงบประมาณ (Capital Allocation)

บางคนอาจให้คำจำกัดความของภาวะในกรณีนี้ว่า "จำนวนงบประมาณที่จัดสรรให้แก่ขั้นที่ " ซึ่งยังไม่ถูกต้อง เพราะภาวะที่กำหนดขึ้นจะต้องช่วยให้ทำการตัดสินใจที่เป็นไปได้ (Feasible Decision) ในขั้นต่าง ๆ ได้โดยไม่ต้องตรวจสอบการตัดสินใจที่ได้ทำไปแล้วในขั้นต่อน ๆ คำจำกัดความข้างต้นเพียงแต่แสดงว่า งบประมาณที่จัดสรรให้ขั้นที่ i สามารถมีค่าได้ตั้งแต่ศูนย์จนถึงจำนวนงบประมาณที่มีอยู่คือ C บาท ซึ่งยังไม่เป็นการเพียงพอที่จะประกันว่า การตัดสินใจในแต่ละขั้นเป็นการตัดสินใจที่เป็นไปได้

สมมติว่าได้ตัดสินใจที่จะแบ่งงบประมาณ 0.4 C ให้แก่ขั้นที่ i ที่กำลังพิจารณาอยู่ เราไม่สามารถจะประกันได้ว่า การตัดสินใจนี้จะเป็นการตัดสินใจที่เป็นไปได้ นอกจากจะต้องตรวจสอบว่างบประมาณที่ได้จัดให้แก่ขั้นต่าง ๆ ที่ผ่านไปแล้วมีจำนวนไม่เกิน C-0.4C = 0.6 C แสดงว่าคำจำกัดความของภาวะข้างต้นไม่เอื้ออำนวยให้ทำการตัดสินใจ เพื่อให้ได้ผลดีที่สุดได้โดยอิสระ ซึ่งไม่ตรงตามความคิดพื้นฐานของโปรแกรมพลวัต

คำจำกัดความของภาวะในขั้น i ที่ถูกต้องในกรณีนี้คือ "จำนวนงบประมาณทั้งหมดที่จัดสรรให้แก่ขั้น 1, 2, ... และขั้น i " ซึ่งทำให้สามารถทำการตัดสินใจที่เป็นไปได้ในขั้นนี้ โดยไม่จำเป็นต้องตรวจสอบการตัดสินใจในขั้นที่ผ่านมาแล้ว จำนวนที่ให้แก่ขั้น i โดยเฉพาะจะมีค่าเท่ากับผลต่างของภาวะในขั้นที่ i และ i-1

การตัดสินใจในแต่ละขั้นเพื่อให้ได้ผลที่ดีที่สุดนั้น มีขึ้นเพื่อให้ได้รับผลตอบแทนตามเป้าหมายให้มากที่สุดเท่าที่จะทำได้ภายในขอบข่ายของภาวะของปัญหา โดยจะเริ่มต้นจากขั้นที่ 1 แล้วต่อไปยังขั้นที่ 2 และต่อไปเรื่อย ๆ จนหมดทุกขั้น

2. สมการรีเคอซีฟ (Recursive Equation)

สมการรีเคอซีฟช่วยในการใช้ขั้นและภาวะในการแยกปัญหาในโปรแกรมพลวัต และยังช่วยให้ทำการหาผลที่ดีที่สุด (Optimize) ของขั้นต่าง ๆ ได้โดยลำพังอีกด้วย ยิ่งไปกว่านั้น ยังช่วยให้สามารถรู้ค่าผลตอบแทนที่ดีที่สุดสะสม (Cumulative Optimal Return) ของขั้นต่าง ๆ ที่ผ่านมาแล้วตลอดเวลา ดังนั้นเมื่อเสร็จการหาผลที่ดีที่สุดของขั้นสุดท้าย ก็จะทราบผลตอบแทนที่ดีที่สุด (Optimal Return) ของปัญหาทั้งหมดได้ทันที

ตัวอย่างการแก้ปัญหาเกี่ยวกับการลงทุนโดยใช้โปรแกรมพลวัตต่อไปนี้ จะอธิบายแนวความคิดของสมการรีเคอซีฟได้ชัดเจนขึ้น

ตัวอย่างที่ 2.1

นักลงทุนผู้หนึ่งมีเงินทุน \$ 6,000 เพื่อใช้ในการลงทุน 3 อย่าง โดยจะต้องลงทุนเป็นจำนวนเต็มของ \$ 1,000 ผลตอบแทนที่ได้รับขึ้นอยู่กับจำนวนเงินที่ลงทุนตามตาราง ที่ 2.1

ตารางที่ 2.1 ผลตอบแทนจากการลงทุนในตัวอย่างที่ 2.1

จำนวนเงินลงทุน (× \$ 1,000)	ผลตอบแทนจากการลงทุน (× \$ 1,000)		
	การลงทุน 1	การลงทุน 2	การลงทุน3
0	0	0	0
1	0.5	1.5	1.2
2	1.0	2.0	2.4
3	3.0	2.2	2.5
4	3.1	2.3	2.6
5	3.2	2.4	2.7
6	3.3	2.5	2.8

จากแนวความคิดของโปรแกรมพลวัต จะเห็นได้ว่าปัญหานี้เป็นปัญหาที่มี 3 ขั้น โดยมีการลงทุน 1,2 และ 3 เป็นแต่ละขั้น ซึ่งต่างก็มีทางเลือก (Alternatives) 7 ทางเลือก โดยมีงบประมาณสูงสุดเท่ากับ 6 ($ 6,000) จากตารางที่ 2.1 สามารถนำมาเขียนใหม่โดยแสดงค่า C_{i,m_i} และ R_{i,m_i} ในตารางที่ 2.2

ตารางที่ 2.2 ค่าใช้จ่ายและผลตอบแทนจากการลงทุนของทางเลือก ต่าง ๆ ในตัวอย่างที่ 2.1

m_i	$i = 1$		$i = 2$		$i = 3$	
	C_{1,m_1}	R_{1,m_1}	C_{2,m_2}	R_{2,m_2}	C_{3,m_3}	R_{3,m_3}
1	0	0	0	0	0	0
2	1	0.5	1	1.5	1	1.2
3	2	1.0	2	2.0	2	2.4
4	3	3.0	3	2.2	3	2.5
5	4	3.1	4	2.3	4	2.6
6	5	3.2	5	2.4	5	2.7
7	6	3.3	6	2.5	6	2.8

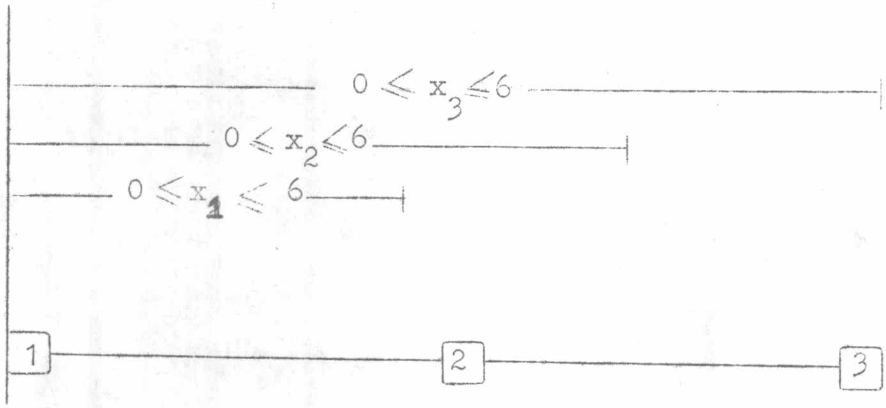
m_i คือ ทางเลือกของการลงทุน i (i=1, 2, 3)

C_{i,m_i} คือ ค่าใช้จ่ายในการลงทุน i ตามทางเลือก m_i

R_{i,m_i} คือ ผลตอบแทนจากการลงทุน i ตามทางเลือก m_i

ให้ $f_i(x_i)$ คือ ผลตอบแทนที่ดีที่สุดสะสมของขั้นที่ 1, 2,... จนถึง i

x_i คือ ตัวแปรภาวะ (State Variables) เมื่อ x_i เป็นเงินลงทุนของขั้น 1, 2... และ i



รูปที่ 2.5 แสดงลักษณะและขอบเขตของตัวแปรภาวะในตัวอย่างที่ 2.1

พิจารณารูปที่ 2.5 ค่า x_1 และ x_2 ไม่ทราบแต่ชัดว่าจะเป็นเท่าใดภายในช่วง $0 \leq x_1 \leq 6$ และ $0 \leq x_2 \leq 6$ ถ้า $x_1 = 0$ หมายความว่าไม่มีการลงทุนในขั้นที่ 1 แต่ถ้า $x_1 = 6$ แสดงว่าเงินทุนทั้งหมดใช้ไปในขั้นที่ 1 และถ้า $x_2 \leq 6$ ก็แสดงว่าเงินของการลงทุน 1 และ 2 มีค่าได้ระหว่าง 0 ถึง 6 และเนื่องจากมีเงินลงทุนทั้งหมด (การลงทุนทั้งสาม) $C = 6$ ทำให้ $x_3 = 6$

วิธีการของโปรแกรมพลวัต เริ่มต้นด้วยการคำนวณหาค่า $f_1(x_1)$ ซึ่งเป็นผลตอบแทนที่ดีที่สุดของขั้น 1 ผลตอบแทนนี้เป็นฟังชั่นของ x_1 ต่อไปก็คำนวณค่า $f_2(x_2)$ อันเป็นผลตอบแทนที่ดีที่สุดสะสมของขั้นที่ 1 และ 2 โดยคำนวณจาก $f_1(x_1)$ ด้วย แล้วท้ายที่สุดก็คำนวณค่า $f_3(x_3)$ จากค่า $f_2(x_2)$

<u>ขั้นที่ 1</u> $(x_1 = 0, 1, 2, 3, 4, 5, 6)$

ถ้า $x_1 = 0$ มีทางเลือกที่เป็นไปได้เพียงทางเลือกเดียว คือ $m_1 = 1$ เพราะมี $C_{1,1} = 0$ (ซึ่งไม่มากกว่า x_1) ทำให้ได้รับผลตอบแทน $R_{1,1} = 0$ ดังนั้น ผลตอบแทนที่ดีที่สุดสะสม $f_1(x_1 = 0) = 0$

$x_1 = 1$ มีทางเลือกที่เป็นไปได้ 2 ทางเลือก คือ $m_1 = 1$ และ $m_1 = 2$ เนื่องจาก $m_1 = 2$ ให้ผลตอบแทน $R_{1,2} = 0.5$ ซึ่งมากกว่า $R_{1,1} = 0$ ทำให้ทางเลือกที่ดีที่สุดคือ $m_1 = 2$ นั่นคือ เมื่อ $x_1 = 1$ $m_1 = 2$ และ $f_1(x_1) = 0.5$

$x_1 = 2$ ทางเลือกที่เป็นไปได้มี 3 ทางเลือกคือ $m_1 = 1, m_1 = 2$ และ $m_1 = 3$ ทางเลือกที่ดีที่สุดคือ $m_1=3$ เพราะ $R_{1,3} = \text{Max}\left\{R_{1,1}, R_{1,2}, R_{1,3}\right\} = \text{Max}\left\{0, 0.5, 1.0\right\} = 1.0$

ถ้า $x_1 = 3$ ก็จะมีทางเลือกที่เป็นไปได้ 4 ทางเลือก คือ $m_1 = 1, m_1 = 2,$ $m_1 = 3$ และ $m_1 = 4$ ทางเลือกที่ดีที่สุด $m_1 = 4$ เพราะ $R_{1,4} = \text{Max}\left\{R_{1,1}, R_{1,2}, R_{1,3}, R_{1,4}\right\} = \text{Max}\left\{0, 0.5, 1.0, 3.0\right\} = 3.0$

เมื่อทำต่อไปด้วยวิธีนี้จนถึง $x_1 = 6$ ก็จะได้ทางเลือกที่ดีที่สุดและผลตอบแทนที่ดีที่สุดของขั้นที่ 1 ตามตารางที่ 2.3

ตารางที่ 2.3 ทางเลือกที่ดีที่สุดและผลตอบแทนที่ดีที่สุดสะสมของขั้นที่ 1 ในตัวอย่างที่ 2.1

x_1	$f_1(x_1)$	m_1
0	0	1
1	0.5	2
2	1.0	3
3	3.0	4
4	3.1	5
5	3.2	6
6	3.3	7

<u>ขั้นที่ 2</u> $(x_2 = 0, 1, 2, 3, 4, 5, 6)$

ถึงแม้ว่าในขั้นที่ 2 นี้จะสามารถทำการหาผลตอบแทนที่ดีที่สุดจากเฉพาะทางเลือกของขั้นนี้ก็ตาม ผลตอบแทนที่ดีที่สุดจะต้องเป็นผลรวมของผลตอบแทนในขั้นที่ 2 และขั้นที่ 1 การใช้ผลตอบแทนที่ดีที่สุดสะสมเป็นเกณฑ์พิจารณาในการหาผลที่ดีที่สุดเช่นนี้ เมื่อทำการคำนวณขั้นสุดท้ายแล้ว ก็จะทราบค่าผลตอบแทนที่ดีที่สุดของทั้งหมดได้ทันที

ผลตอบแทนสะสมของขั้นที่ 2 คำนวณจากผลรวมของผลตอบแทนของทางเลือกในขั้นที่ 2 และ $f_1(x_1)$ เมื่อเป็นเช่นนี้แล้ว ก็หมายความว่า ไม่มีความจำเป็นอันใดที่จะต้องย้อนไปตรวจสอบความเป็นไปได้ของการตัดสินใจในขั้นที่ผ่านมาแล้ว ซึ่งเป็นไปตามหลักเกณฑ์ของภาวะของระบบ

ถ้า $x_2 = 0$ มีทางเลือกที่เป็นไปได้เพียงทางเลือกเดียวคือ $m_2 = 1$ ซึ่งมี $R_{2,1} = 0$ เพราะว่า $C_{2,1} = 0$ เป็นจำนวนเงินที่ใช้ในการลงทุนในขั้นที่ 2 จึงเหลือ $x_1 = x_2 - C_{2,1} = 0-0 = 0$ สำหรับการลงทุนในขั้นที่ 1 ผลตอบแทนที่ดีที่สุดในขั้นที่ 1 เมื่อ $x_1 = 0$ คือ $f_1(0) = 0$ ดังนั้น

เมื่อ $x_2 = 0$ ทำให้ $f_2(x_2 = 0) = R_{2,1} + f_1(x_2 - x_1) = 0+0 = 0$ นั่นคือ $f_2(0) = 0$ โดยมี $m_2^* = 1$

ต่อไป ถ้า $x_2 = 1$ มีทางเลือกที่เป็นไปได้ 2 ทางเลือกคือ $m_2 = 1$ และ $m_2 = 2$ ส่วน $m_2 = 3$ เป็นทางเลือกที่เป็นไปไม่ได้ เพราะ $C_{2,3} = 2$ ซึ่งมากกว่า $x_2 = 1$ ดังนั้น

$$f_2(x_2 = 1) \quad = \max_{m_2 = 1,2} \left\{ R_{2,m_2} + f_1(x_2 - C_{2,m_2}) \right\}$$

$$= \text{Max} \begin{cases} R_{2,1} + f_1(x_2 - C_{2,1}) \\ R_{2,2} + f_1(x_2 - C_{2,2}) \end{cases}$$

$$= \text{Max} \begin{cases} 0 + f_1(1-0) = 0 + 0.5 = 0.5 \\ 1.5 + f_1(1-1) = 1.5 + 0 = \underline{1.5} \end{cases}$$

$$= 1.5$$

โดยมี $m_2^* = 2$

โดยวิธีเดียวกัน เมื่อ $x_2 = 2$

$$f_2(x_2 = 2) = \max_{m_2 = 1,2,3} \left\{ R_{2,m_2} + f_1(x_2 - C_{2,m_2}) \right\}$$

$$= \text{Max} \begin{cases} R_{2,1} + f_1(x_2 - C_{2,1}) = 0 + f_1(2-0) \\ R_{2,2} + f_1(x_2 - C_{2,2}) = 1.5 + f_1(2-1) \\ R_{2,3} + f_1(x_2 - C_{2,3}) = 2.0 + f_1(2-2) \end{cases}$$

$$= \text{Max} \begin{cases} 0 + 1.0 = 1.0 \\ 1.5 + 0.5 = \underline{2.0} \\ 2.0 + 0 = \underline{2.0} \end{cases}$$

$$= 2.0$$

โดย $m_2^* = 2, 3$

เมื่อ $x_2 = 3$

$$f_2(x_2 = 3) = \max_{m_2 = 1,\,2,\,3,\,4} \left\{ R_{2,m_2} + f_1(x_2 - C_{2,m_2}) \right\}$$

$$= \text{Max} \begin{cases} 0 + f_1(3-0) = 0 + 3.0 = \underline{3.0} \\ 1.5 + f_1(3-1) = 1.5 + 1.0 = 2.5 \\ 2.0 + f_1(3-2) = 2.0 + 0.5 = 2.5 \\ 2.2 + f_1(3-3) = 2.2 + 0 = 2.2 \end{cases}$$

$$= 3.0$$

โดย $m_2^* = 1$

เมื่อ $x_2 = 4$

$$f_2(x_2 = 4\ ) = \max_{m_2 = 1,2,3,4,5} \left\{ R_{2,m_2} + f_1(x_1 - C_{2,m_2}) \right\}$$

$$= \text{Max} \begin{cases} 0 + f_1(4-0) = 0 + 3.1 = 3.1 \\ 1.5 + f_1(4-1) = 1.5 + 3.0 = \underline{4.5} \\ 2.0 + f_1(4-2) = 2.0 + 1.0 = 3.0 \\ 2.2 + f_1(4-3) = 2.2 + 0.5 = 2.7 \\ 2.3 + f_1(4-4) = 2.3 + 0 = 2.3 \end{cases}$$

$$= 4.5$$

โดย $m_2^* = 2$

เมื่อ $x_2 = 5$

$$f_2(x_2 = 5) = \underset{m_2 = 1,2 \ldots,6}{\text{Max}} \left\{ R_{2,m_2} + f_1(x_1 - C_{2,m_2}) \right\}$$

$$= \text{Max} \begin{cases} 0 + f_1(5-0) = 0 + 3.2 = 3.2 \\ 1.5 + f_1(5-1) = 1.5 + 3.1 = 4.6 \\ 2.0 + f_1(5-2) = 2.0 + 3.0 = \underline{5.0} \\ 2.2 + f_1(5-3) = 2.2 + 1.0 = 3.2 \\ 2.3 + f_1(5-4) = 2.3 + 0.5 = 2.8 \\ 2.4 + f_1(5-5) = 2.4 + 0 \quad = 2.4 \end{cases}$$

$$= 5.0$$

โดย $m_2^* = 3$

เมื่อ $x_2 = 6$

$$f_2(x_2 = 6) = \underset{m_2 = 1,2,\ldots,7}{\text{Max}} \left\{ R_{2,m_2} + f_1(x_2 - C_{2,m_2}) \right\}$$

$$= \text{Max} \begin{cases} 0 + f_1(6-0) = 0 + 3.3 = 3.3 \\ 1.5 + f_1(6-1) = 1.5 + 3.2 = 4.7 \\ 2.0 + f_1(6-2) = 2.0 + 3.1 = 5.1 \\ 2.2 + f_1(6-3) = 2.2 + 3.0 = \underline{5.2} \\ 2.3 + f_1(6-2) = 2.3 + 1.0 = 3.3 \\ 2.4 + f_1(6-5) = 2.4 + 0.5 = 3.1 \\ 2.5 + f_1(6-6) = 2.5 + 0 \quad = 2.5 \end{cases}$$

$$= 5.2$$

โดย $m_2^* = 4$

ในตารางที่ 2.4 เป็นผลตอบแทนที่ดีที่สุดสะสม ($f_2(x_2)$) และทางเลือกที่ดีที่สุด (m_2*)ของขั้นที่ 2 ซึ่งได้จากการคำนวณข้างต้น

ตารางที่ 2.4 ผลตอบแทนที่ดีที่สุดสะสมและทางเลือกที่ดีที่สุดของขั้นที่ 2 ในตัวอย่างที่ 2.1

x_2	$f_2(x_2)$	m_2^*
0	0	1
1	1.5	2
2	2.0	2, 3
3	3.0	1
4	4.5	2
5	5.0	3
6	5.2	4

<u>ขั้นที่ 3</u> ($x_3 = 6$)

เมื่อ $x_3 = 6$ มีทางเลือกที่เป็นไปได้ทั้งหมด 7 ทางเลือก คือ $m_3 = 1$, 2, 3, 4, 5, 6 และ 7 ดังนั้น

$$f_3(x_3 = 6) = \max_{m_3 = 1,2,\ldots,7} \left\{ R_{3,m_3} + f_2(x_3 - C_{3,m_3}) \right\}$$

$$= \max \begin{cases} 0 + f_2(6-0) = 0 + 5.2 = 5.2 \\ 1.2 + f_2(6-1) = 1.2 + 5.0 = 6.2 \\ 2.4 + f_2(6-2) = 2.4 + 4.5 = \underline{6.9} \\ 2.5 + f_2(6-3) = 2.5 + 3.0 = 5.5 \\ 2.6 + f_2(6-4) = 2.6 + 2.0 = 4.6 \\ 2.7 + f_2(6-5) = 2.7 + 1.5 = 4.2 \\ 2.8 + f_2(6-6) = 2.8 + 0 \quad = 2.8 \end{cases}$$

$$= 6.9$$

โดย $m_3^* = 3$

ในการคำนวณหาค่า $f_3(x_3)$ ขั้นที่ 1 และ 2 ถูกถือเสมือนเป็นขั้นเดียวที่มีข้อมูลรวมกันอยู่ในรูปของ $f_2(x_2)$ เมื่อ $x_2 = x_3 - C_{3,m_3}$ หรืออีกนัยหนึ่งลักษณะของขั้นที่ 1 และขั้นที่ 2 โดยลำพังจะไม่ถูกนำมาพิจารณาในการคำนวณค่า $f_3(x_3)$ ซึ่งก็หมายความว่า ไม่มีการสืบสาวถึงผลการตัดสินใจในขั้นต่าง ๆ ก่อนหน้าที่จะถึงขั้นปัจจุบัน

ผลลัพธ์ที่ดีที่สุดของปัญหาซึ่งหมายถึงการตัดสินใจที่ทำให้ได้ผลตอบแทนที่ดีที่สุด สามารถอ่านได้โดยตรงจากผลการคำนวณเบื้องต้น ด้วยการเริ่มต้นจากขั้นที่ 3 เมื่อ $x_3=6$ ทางเลือกที่ดีที่สุด $m_3 = 3$ ซึ่งมี $C_{3,3} = 2$ แสดงว่าเงินลงทุนในขั้นที่ 3 มีจำนวน \$ 2,000 เหลือ $x_2 = 6-2 = 4$ จากตารางที่ 2.4 เมื่อ $x_2 = 4$ ทางเลือกที่ดีที่สุด $m_2^* = 2$ ซึ่งมีผลให้ $x_1 = x_2 - C_{2,m_2} = 4-1 = 3$ และจากตารางที่ 2.3 เมื่อ $x_1 = 3$ ทางเลือกที่ดีที่สุดคือ $m_1^* = 4$

ดังนั้นการตัดสินใจที่ดีที่สุด (Optimal Decision) สำหรับปัญหานี้ คือ $m_1^* = 4$, $m_2^* = 2$ และ $m_3^* = 3$ โดยได้รับผลตอบแทนที่ดีที่สุด \$ 6,900

สิ่งที่น่าสนใจประการหนึ่งคือ โปรแกรมพลวัตช่วยให้สามารถทำการวิเคราะห์ความไว (Sensitivity Analysis) ได้ทันทีโดยอัตโนมัติ ตัวอย่างเช่น สมมติว่าต้องการศึกษาการเปลี่ยนแปลงผลลัพธ์ที่ดีที่สุด ถ้าเงินทุนทั้งหมด (C) เป็น 5 แทนที่จะเป็น 6 การหาผลลัพธ์ที่ดีที่สุดเมื่อ C เปลี่ยนเป็น 5 ก็เพียงแต่คำนวณหาค่า $f_3(x_3 = 5)$ ดังนี้

$$f_3(x_3 = 5) = \max_{m_3 = 1,2,\ldots\ldots,6} \left\{ R_{3,m_3} + f_2(x_3 - C_{3,m_3}) \right\}$$

$$= \text{Max} \begin{cases} 0 + f_2(5-0) = 0 + 5.0 = 5.0 \\ 1.2 + f_2(5-1) = 1.2 + 4.5 = \underline{5.7} \\ 2.4 + f_2(5-2) = 2.4 + 3.0 = 5.4 \\ 2.5 + f_2(5-3) = 2.5 + 2.0 = 4.5 \\ 2.6 + f_2(5-4) = 2.6 + 1.5 = 4.1 \\ 2.7 + f_2(5-5) = 2.7 + 0 = 2.7 \end{cases}$$

$$= 5.7$$

โดย $m_3^* = 2$

ผลลัพธ์ที่ดีที่สุดก็สามารถหาได้โดย $x_3 = 5 \longrightarrow m_3^* = 2 \longrightarrow x_2 = 5-1 = 4 \longrightarrow m_2^* = 2 \longrightarrow x_1 = 4-1 = 3 \longrightarrow m_1^* = 4$

เพราะว่าการคำนวณในขั้นสุดท้ายไม่ใช่เรื่องยุ่งยาก ดังนั้นจะเป็นประโยชน์ยิ่งขึ้น ถ้าจะทำการคำนวณตั้งแต่ $x_3 = 0, 1, 2, \ldots\ldots, C$ แทนที่จะคำนวณเพราะ $x_3 = C$ เท่านั้น

จากตัวอย่างข้างต้น ข้อมูลที่ใช้ในการหาผลตอบแทนเพื่อนำไปสู่การตัดสินใจที่ดีที่สุดในขั้น i จากขั้นที่ผ่านมาแล้ว i-1 ขั้น รวมอยู่ในรูปของผลที่ดีที่สุดสะสม $f_{i-1}(x_{i-1})$ ผลตอบแทนที่ดีที่สุดสะสมของขั้นที่ i-1 นี้ เป็นข้อมูลเพียงประการเดียวจากขั้นต่าง ๆ ที่ผ่านมา i-1 ขั้น ซึ่งนำมาใช้ในการหาผลที่ดีที่สุดในขั้นที่ i

จะเห็นได้ว่า การคำนวณของโปรแกรมพลวัตเป็นไปตามรูปแบบ (Pattern) พิเศษอันหนึ่ง กล่าวคือ ลำดับในการคำนวณในปัญหาที่มี N ขั้น จะเป็นไปดังนี้

$$f_1,\ f_2,\ \ldots\ldots,\ f_i,\ \ldots\ldots,\ f_n$$

รูปแบบนี้สามารถแสดงโดยใช้สมการรีเคอซีฟทั่วไปโดยสัมพันธ์กับ $f_i(x_i)$ และ $f_{i-1}(x_{i-1})$

สมการรีเคอซีฟของปัญหาที่เกี่ยวข้องกับการลงทุนที่มี N ขั้นในตัวอย่างที่ 2.1 สามารถเขียนได้ดังนี้

$$f_1(x_1) = \max_{\substack{m_1 \\ C_{1,m_1} \le x_1}} \left\{ R_{1,m_1} \right\}$$

$$f_i(x_i) = \max_{\substack{m_i \\ C_{i,m_i} \underset{\le}{\cancel{\ngtr}} x_i}} \left\{ R_{i,m_i} + f_{i-1}(x_i - C_{i,m_i}) \right\}$$

การเปลี่ยนจาก x_{i-1} ไปสู่ x_i เรียกว่า "การแปลงภาวะ (State - Transformation)" ซึ่งแสดงถึงความสัมพันธ์ระหว่าง x_{i-1} และ x_i การแปลงภาวะนี้อาจเป็นไปอย่างง่าย ๆ ดังเช่นในตัวอย่างที่ 2.1 นี้ ($x_{i-1} = x_i - C_{i,m_i}$) แต่ในบางกรณีการแปลงภาวะอาจยุ่งยากกว่านี้ซึ่งจะมีผลต่อประสิทธิภาพของการคำนวณด้วย เพราะสมการการแปลงภาวะนี้เป็นเงื่อนไขขอบข่าย (Constraint) ของปัญหา

สมการกำหนดเป้าหมาย (Objective Function) ของโปรแกรมพลวัตอาจอยู่ในรูปผลรวมของผลตอบแทน (Return) ของขั้นต่าง ๆ หรือผลคูณ หรืออาจจะเป็นผลบวกและผลคูณของผลตอบแทนของขั้นต่าง ๆ ร่วมกันก็ได้ อย่างไรก็ตาม สมการกำหนดเป้าหมายนี้จะต้องสามารถแยกออกเป็นปัญหาย่อย ๆ ได้ ปัญหาย่อยที่แยกออกมานี้จะสามารถวิเคราะห์เพื่อแก้ปัญหาได้ดีกว่าปัญหาใหญ่ แต่ก็ไม่มีสิ่งใดจะประกันได้ว่า ปัญหาย่อย ๆ เหล่านี้จะสามารถแก้ปัญหาได้ด้วยวิธีง่าย ๆ ถึงแม้ว่าอาจจะมีความยากลำบากอยู่บ้างในการคำนวณ แต่โปรแกรมพลวัตก็สามารถจะใช้จัดการกับปัญหาต่าง ๆ หลายชนิดได้อย่างดียิ่ง

3. <u>การคำนวณไปข้างหน้าและย้อนกลับ</u> (Forward & Backward Computation)

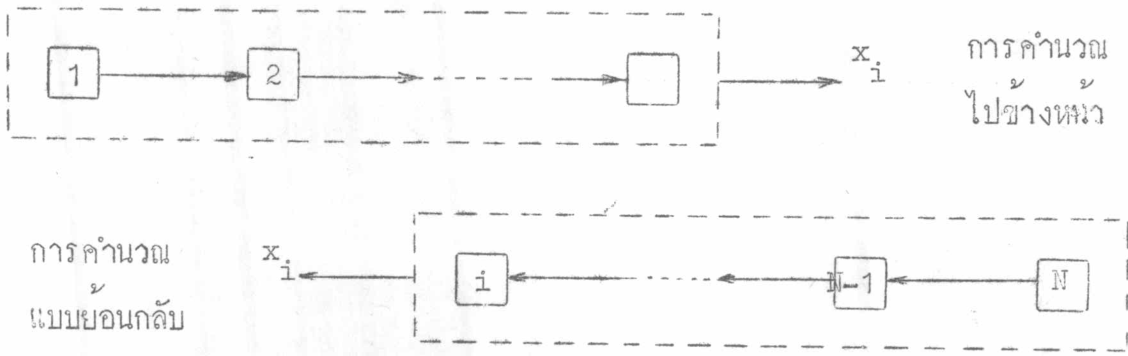
สมการรีเคอซีฟในตัวอย่างที่ 2.1 ทำการหาค่า f_i ตามลำดับ

$$f_1 \longrightarrow f_2 \longrightarrow \cdots\cdots\cdots\cdots \longrightarrow f_n$$

เมื่อ f_1 และ f_n เป็นฟังก์ชั่นแรกและฟังก์ชั่นสุดท้ายของสมการรีเคอซีฟ การคำนวณแบบนี้เรียกว่า "การคำนวณไปข้างหน้า (Forward Computation)" สมการรีเคอซีฟอาจมีการสร้างรูปแบบปัญหาในลักษณะที่แตกต่างกัน ทำให้การหาผลลัพธ์ทำได้โดยการคำนวณจากขั้นสุดท้ายย้อนกลับมาสู่ขั้นแรกได้ ดังนี้

$$R_n \longrightarrow f_{n-1} \longrightarrow \cdots\cdots \longrightarrow f_2 \longrightarrow f_1$$

ในกรณีนี้ เป็นวิธีการคำนวณแบบที่เรียกว่า "การคำนวณแบบย้อนกลับ" (Backward Computation)



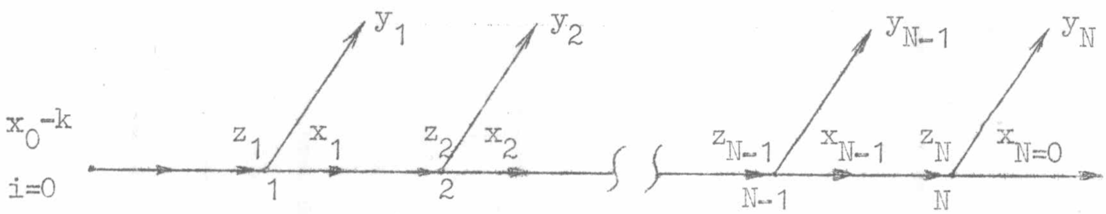
รูปที่ 2.6 การคำนวณไปข้างหน้าและการคำนวณแบบย้อนกลับ

เมื่อพิจารณาปัญหาเกี่ยวกับการลงทุนในตัวอย่างที่ 2.1 ภาวะที่ขั้น i อาจกำหนดได้ว่า " งบประมาณที่จัดสรรให้ขั้นที่ i และขั้นที่ผ่านมาแล้ว i-1 ขั้น " หรือ " งบประมาณที่จัดสรรให้ขั้นที่ i และขั้นต่อ ๆ ไปอีก N-i ขั้น " ก็ได้ รูปที่ 2.6 แสดงความแตกต่างของภาวะในกรณีทั้งสอง ในกรณีแรก การคำนวณเริ่มด้วยการหาค่า f_1 แต่ในกรณีหลังการคำนวณเริ่มด้วยการหาค่า f_n จึงสรุปได้ว่า การกำหนดภาวะจะเป็นสิ่งชี้ว่าจะต้องทำการคำนวณไปข้างหน้าหรือย้อนกลับ

ถึงแม้ว่าการคำนวณทั้งสองวิธีนี้จะให้ผลลัพธ์เหมือนกันในที่สุด แต่โดยทั่วไปแล้วการคำนวณแบบย้อนกลับมักจะสะดวกกว่า ทั้งนี้เป็นเหตุสืบเนื่องมาจากการสร้างการแปลงภาวะในระหว่างขั้นต่าง ๆ ในตัวอย่างที่ 2.1 เป็นปัญหาง่าย ๆ ความแตกต่างของวิธีการทั้งสองจึงไม่ปรากฏให้เห็นเด่นชัด ตัวอย่างที่ 2.2 จะแสดงให้เห็นความแตกต่างของการคำนวณไปข้างหน้าและการคำนวณแบบย้อนกลับ

<u>ตัวอย่างที่ 2.2</u>

ผู้เลี้ยงปศุสัตว์รายหนึ่งมีแกะอยู่ k ตัว ในช่วง N ปีข้างหน้าทุก ๆ สิ้นปีเขาจะต้องพิจารณาขายแกะปีละครั้ง ซึ่งเขาจะต้องตัดสินใจว่าจะขายไปเป็นจำนวนเท่าใดและเหลือเอาไว้เท่าใด ถ้าขายก็จะได้กำไรตัวละ p_i ที่ปีที่ i ส่วนแกะที่ไม่ขายจะมีจำนวนเพิ่มขึ้นเป็นสองเท่าเมื่อสิ้นปีที่ i + 1 คือปีถัดไป และเมื่อครบ N ปีแล้ว เขาจะขายแกะที่เหลือจนหมด



รูปที่ 2.7 รูปแบบของปัญหาในตัวอย่างที่ 2.2

แต่ละปีถือเป็นหนึ่งขั้น เพราะฉะนั้น ขั้นที่ i จึงแทนปีที่ i

พิจารณารูปที่ 2.6 ให้ x_i เป็นจำนวนแกะที่เหลือเอาไว้ในปีที่ i และ y_i เป็นจำนวนที่ขายไปในปีเดียวกัน กำหนดให้ $z_i = x_i + y_i$ และจากเงื่อนไขของปัญหาคือ จำนวนแกะที่เหลืออยู่จะเพิ่มเป็นสองเท่าเมื่อสิ้นปีถัดไป ดังนั้น

$$z_1 = 2\,x_0 = 2k$$

$$z_i = 2\,x_{i-1} \qquad i = 2, \ldots\ldots, N$$

ภาวะของระบบที่ขั้น i อาจจะเป็น z_i ซึ่งเป็นจำนวนแกะที่มีอยู่ที่ขั้น i หรือ x_i ซึ่งเป็นจำนวนแกะที่เหลืออยู่หลังจากตัดสินใจขายไปในขั้นที่ 1, 2,....... และ i แล้ว อย่างใดอย่างหนึ่ง

ถ้าให้ z_i เป็นภาวะและ $f_i(z_i)$ เป็นผลกำไรที่ดีที่สุดสะสมของขั้น i, i + 1,....., N สมการรีเคอชีฟในกรณีนี้ ก็จะสามารถเขียนได้ดังนี้

$$f_N(z_N) = \max_{y_N = z_N \leq 2^N k} \left\{ p_N y_N \right\}$$

$$f_i(z_i) = \max_{0 \leq y_i \leq z_i \leq 2^i k} \left\{ p_i y_i + f_{i+1}(2(z_i - y_i)) \right\}$$

$$i = 1,2,\ldots,N-1$$

y_i และ z_i จะต้องเป็นเลขจำนวนเต็มที่ไม่ติดลบ และ y_i ซึ่งเป็นจำนวนที่ขายไปเมื่อสิ้นสุดปีที่ i จะต้องน้อยกว่าหรือเท่ากับ z_i ค่าสูงสุดที่จะเป็นไปได้ของ y_i คือ $2^i k$ (เมื่อ k เป็นจำนวนเริ่มแรกของฝูงแกะ) ซึ่งจะเกิดขึ้นได้ถ้าไม่มีการขายเลยในขั้นที่ 1, 2, ... และ i-1 เพราะฉะนั้น ทางเลือกในแต่ละขั้นจึงเป็น $y_i = 0,1,2, \ldots\ldots, 2^i k$ สมการรีเคอชีฟข้างต้นทำการคำนวณค่า f_i จาก f_{i+1} ดังนั้น ลำดับของการคำนวณจึงเป็น

$$f_N \longrightarrow f_{N-1} \longrightarrow \cdots\cdots\cdots \longrightarrow f_2 \longrightarrow f_1$$

ซึ่งเป็นการคำนวณแบบย้อนกลับ (Backward Computation)

ถ้าให้ x_i เป็นภาวะของระบบ สมการรีเคอชีฟก็จะต้องแตกต่างออกไปเมื่อให้ $g_i(x_i)$ เป็นค่าผลกำไรที่ดีที่สุดสะสมจากขั้นที่ 1, 2, ... และ i สมการรีเคอชีฟในกรณีหลังนี้ก็จะเป็น

$$g_1(x_1) = \max_{y_1 = 2k - x_1} \left\{ p_1 y_1 \right\}$$

$$g_i(x_i) = \max_{y_i \leq 2^i k - x_i} \left\{ p_i y_i + g_{i-1} \frac{(y_i + x_i)}{2} \right\}$$

$$i = 2, 3, \ldots, n$$

และ $\frac{y_i + x_i}{2}$ เป็น integer

y_i และ x_i เป็นเลขจำนวนเต็มที่ไม่ติดลบ การแก้สมการรีเคอซีฟนี้ หาค่า g_i จาก g_{i-1} ดังนั้น ลำดับของการคำนวณจึงเป็น

$$g_1 \longrightarrow g_2 \longrightarrow \quad \cdots\cdots\cdots \quad \longrightarrow g_n$$

ซึ่งเป็นการคำนวณไปข้างหน้า (Forward Computation)

เมื่อพิจารณาเปรียบเทียบระหว่างวิธีการทั้งสอง จะเห็นได้ว่าการแปลงภาวะของการคำนวณไปข้างหน้าทำให้การคำนวณยุ่งยากกว่าการคำนวณแบบย้อนกลับซึ่งการแปลงภาวะจากขั้นที่ i ไปยังขั้นที่ i + 1 เป็นไปอย่างง่าย ๆ คือ $z_{i+1} = 2(z_i - y_i)$ เป็นผลให้การคำนวณ $f_i(z_i)$ ทำได้ง่ายเช่นกัน แต่ในการคำนวณไปข้างหน้า การแปลงภาวะจากขั้นที่ i ไปยังขั้นที่ i-1 คือ $x_{i-1} = (y_i+x_i)/2$ โดยที่ x_{i-1} ต้องเป็นเลขจำนวนเต็ม ทำให้ต้องเลือกค่า y_i และ x_i ที่จะทำให้ x_{i-1} เป็นเลขจำนวนเต็มที่ไม่ติดลบและจะต้องเป็นไปตามเงื่อนไข $y_i \leq 2^i k - x_i$ อีกด้วย ทำให้เห็นว่า ในปัญหาเช่นนี้การคำนวณไปข้างหน้าจะยุ่งยากกว่าการคำนวณย้อนกลับเป็นอันมาก