บทที่ 6

โปรแกรมทรานซิท (TRANSYT)

6.1 บทนำ

TRANSYT เป็นวิธีการคำนวณหาจังหวะ เวลาสัญญาณไฟจราจรแบบ fixed – time ให้ได้ค่าที่ดีที่สุด (optimum) ในการควบคุมการจราจรโดยคำนึงให้การไหลของการจราจรในโครงข่ายสัญญาณไฟจราจร ก่อเกิดความล่าช้าน้อยที่สุด วิธีการนี้ได้ถูกจัดทำเป็นโปรแกรมคอมพิวเตอร์ช่วยในการคำนวณคำว่า TRANSYT ยอมาจาก " TRAFFIC NETWORK STUDY TOOL"

TRANSYT ได้คิดและพัฒนาขึ้นมาโดย TRRL (Transportation Road Research Laboratory , ประเทศอังกฤษ) โปรแกรมนี้ได้ถูกนำไปใช้ในการจัดทำแผนการควบคุมการจราจรในหลายแห่งทั่วโลก ซึ่งให้ผลเป็นที่น่าพอใจและเป็นที่ยอมรับในหลายประเทศ กรุงเทพมหานครก็ได้เลือกเอา TRANSYT มาใช้ในการจัดทำแผนควบคุมจราจรเช่นกัน

6.2 หลักการสำคัญของ TRANSYT (Principle of TRANSYT)

โปรแกรมทรานซิท ประกอบด้วยส่วนสำคัญ ๆ 2 ส่วน คือ

6.2.1 รูปจำลองทางการจราจร (Traffic Model)

ใช้ในการคำนวณหาค่า performance index(P.I) ของโครงข่ายสัญญาณไฟ สำหรับค่าจังหวะ เวลาสัญญาณไฟจราจรหนึ่ง ๆ

6.2.2 ขบวนการ Hill – Climbing optimization

เป็นขบวนการคำนวนการเปลี่ยนแปลงจังหวะ เวลาสัญญาณไฟจราจร เพื่อให้ค่า performance index ที่ต่ำที่สุด

6.3 รูปจำลองทางการจราจร (Traffic model)

6.3.1 สมมติฐานเบื้องต้น

โปรแกรมทรานซิทมีสมมติฐานสำคัญ ดังนี้

ก. ทุกทางแยกในโครงข่าย (Network) จะต้องมีการควบคุมด้วยสัญญาณไฟจราจร (หรือโดย priority rule)

ข. ทุกทางแยกจะต้องมีรอบเวลาสัญญาณไฟ (Cycletime) ที่เท่ากันหรือเป็นครึ่งหนึ่ง

ค. ปริมาณการจราจรที่เข้าสู่โครงข่าย จะกำหนดด้วยอัตราคงที่

ง. สัดส่วนของปริมาณยวดยานที่เลี้ยวซ้าย – ขวา ในทุกทางแยกจะคงที่ทุกรอบเวลาสัญญาณไฟ

6.3.2 Performance index

ในการหาจังหวะ เวลาสัญญาณไฟจราจรที่ดีที่สุด จะมีค่า performance index เป็นตัวตัดสินว่า จังหวะเวลาสัญญาณไฟจราจรนี้จำเป็นต้องมีการปรับต่อไปอีกหรือไม่ โดยที่จังหวะเวลาสัญญาณไฟจราจรที่ให้ค่า performance index ที่ต่ำสุด จะเป็นค่าที่ดีที่สุด ในโปรแกรมทรานซิทได้กำหนดค่า performance index ในรูปค่าความล่าช้า ดังนี้

$$\text{Performance index} = \sum_{i=1}^{i=n} (d_i + Kc_i)$$

เมื่อ

d_i – เป็นค่าความล่าช้าเฉลี่ยของ Link ที่ i ในโครงข่าย มีหน่วยเป็น pcu – ชม./ชม.

c_i – เป็นค่าจำนวนรถหยุดเฉลี่ยของ Link ที่ i ในโครงข่ายมีหน่วยเป็น pcu (หยุด)/วินาที

K – เป็นค่า Weighting factor

ถ้าค่า $K = 0$ ค่า performance index จะเท่ากับค่าความล่าช้ารวม

เมื่อ $K \neq 0$ จะมีค่าความล่าช้าเนื่องจากรถหยุดไปคิดด้วย

6.3.3 โครงข่ายสัญญาณไฟจราจร (Signal Network)

ในโครงข่าย ทางแยกที่มีสัญญาณไฟจราจรจะถูกแทนไว้ด้วย Node ทิศทางของการจราจรที่ทางแยกจะถูกกำหนดด้วย Link และปริมาณการ

จราจรในทิศทางต่าง ๆ (ซ้าย,ตรง,ขวา) จะจัดสัมพันธ์กับหมายเลขที่กำหนดไว้

6.3.4 รูปแบบของการจราจร (Traffic Pattern)

ในการคำนวณจะแบ่งรอบเวลา (Cycle time) ออกเป็น 50 ส่วน (steps) การแบ่งรอบเวลานี้จะช่วยการคำนวณได้ละเอียดและแม่นยำขึ้น การคำนวณทุกค่าในโปรแกรมทรานซิทจะขึ้นอยู่กับค่าอัตราการไหลของการจราจร กับ ปริมาณรถที่หยุดรอสัญญาณไฟ (queue) ซึ่งคาดว่าจะเกิดขึ้นในแต่ละหน่วยเวลา

รูปแบบของการไหลของการจราจรเฉลี่ย ที่ผ่านจุด ๆ หนึ่งในโครงข่าย ในการวิเคราะห์แสดงให้เห็นในรูปกราฟ

6.3.5 พฤติกรรมของการจราจรในแต่ละ Link

การคำนวณหาพฤติกรรมของการจราจรในแต่ละ Link จะมีการพิจารณาจากรูปแบบของการไหล (flow pattern) ดังต่อไปนี้

ก. รูปแบบ เข้า (in pattern) เป็นรูปแบบของการจราจรที่เข้ามาที่เส้นหยุด (stop line) ที่ปลาย Link

ข. รูปแบบ ออก (out pattern) เป็นรูปแบบของการจราจรที่ออกจาก Link

ค. รูปแบบ ไป (go pattern) เป็นรูปแบบของการจราจรที่ออกจากเส้นหยุด โดยมีปริมาณการไหลของการจราจรอิ่มตัว (saturated) ตลอดไฟเขียว

ปริมาณการจราจรในแต่ละ Link หาได้จากสัดส่วนของปริมาณการจราจรที่มาจาก Upstream Link อื่น ๆ รูปแบบของการจราจรที่เข้ามาในแต่ละ Link จะเข้ามาด้วยความเร็วที่แตกต่างกัน ซึ่งจะทำให้บางส่วนของกลุ่มการจราจร (platoon) มีการกระจายกันออกไป การเกิดการกระจายของกลุ่มยวดยานที่เข้ามานี้อธิบายได้ในรูปแบบของ Exponential Smoothing ค่าองศาของความ

สม่ำเสมอ (degree of smoothing) จะเป็นตัวแปรกับค่าระยะเวลาเดิน (journey time) ใน Link และค่าระยะเวลาเดินทางนี้จะขึ้นอยู่กับการไหลของการจราจรที่เข้ามา (inflow source) เศษส่วนของความสม่ำเสมอ (smoothed fraction) ของรูปแบบออก (out pattern) จะเป็นตัวกำหนดรูปแบบของการจราจรเข้า (in pattern) ค่าของรูปแบบเข้าในแต่ละ Link เขียนเป็นสมการทางคณิตศาสตร์ได้ดังนี้

$$IN_i = \sum F_{ij}(p_{ij} \cdot OUT_j)$$

เมื่อ

OUT_j – รูปแบบเข้าของ link ที่ i

p_{ij} – สัดส่วนของ OUT_j ที่ให้กับ link ที่ i

F_{ij} – smoothing process ของการไหลจาก link ที่ i ไป j

ค่าจำนวนยวดยาน (m_i) ที่เส้นหยุด ในระหว่างช่วงเวลาจะคำนวณได้จากสูตรความสัมพันธ์

$$m_i = m_{i-1} + q_i - S_i$$

เมื่อ

q_i – จำนวนยวดยานที่เข้ามาในช่วงเวลา i (กำหนดโดย IN pattern)

S_i – จำนวนยวดยานที่ออกไปได้ในช่วงเวลา i (กำหนดโดย GO pattern)

การคำนวณจะเริ่มจากเมื่อจำนวนรถหยุดรอสัญญาณไฟ (queue) มีค่าเป็นศูนย์ไปจนถึงรอบเวลาหนึ่งที่ได้สภาวะคงที่ (steady state condition) ซึ่งค่านี้จะใช้เป็นรูปแบบ ในการทำนายการเกิด queue และความล่าช้า (delay) ในรอบเวลาสัญญาณไฟต่อ ๆ ไป

เพราะฉะนั้น จำนวนยวดยานที่ออกในช่วงเวลา i คือ

$m_{i-j} + q_i - m_i$ (เป็นการกำหนด OUT pattern)

ในการคำนวณหาค่าความล่าช้าเฉลี่ยต่อหนึ่งหน่วยเวลาจะประกอบด้วยการคำนวณค่า 2 ค่ารวมกันคือ ค่าแรก เป็นค่าความล่าช้าของจำนวนยวดยานที่มีรูปแบบของจำนวนรถที่เข้ามาคงที่ทุกรอบเวลา ซึ่งจะให้ค่า queue length เฉลี่ยในหนึ่งรอบเวลา ค่าที่สอง เป็นค่าความล่าช้าของจำนวนยวดยานที่มีรูปแบบการเข้ามา ในแต่ละรอบเวลาเป็นแบบ random ซึ่งจะขึ้นอยู่กับค่าองศาความอิ่มตัวเฉลี่ย (average degree of saturation) ที่เส้นหยุดรอสัญญาณไฟ

ส่วนค่าจำนวนรถหยุดเฉลี่ยในหนึ่งรอบเวลา หาได้จากจำนวนรถทั้งหมดที่เข้ามาหยุดรอสัญญาณไฟ โดยสมมติว่า รถทุกคันมาหยุดรอสัญญาณไฟที่เส้นหยุด

6.3.6 ทางแยกที่ไม่มีสัญญาณไฟ

ในโครงข่ายถนนโดยทั่วไปแล้วจะมีจำนวนหลายทางแยกที่ไม่มีสัญญาณไฟจราจร ปัญหาของจุดตัดของการจราจรใน 2 ทิศทางแบบนี้ จะใช้ priority rule เป็นตัวกำหนด ในตัวโปรแกรมทรานซิท ก็มีการนำเอาทางแยกเหล่านี้มาพิจารณาด้วย

6.4 วิธีการ optimization ของโปรแกรมทรานซิท

6.4.1 ขบวนการ Hill-climbing

เป็นขบวนที่ใช้คำนวณหาค่า จังหวะเวลาสัญญาณไฟที่ดีที่สุดในการควบคุมการจราจร โดยเริ่มคำนวณหาค่า performance index ของจังหวะเวลาสัญญาณไฟเริ่มต้น (initial setting) ก่อน จากนั้นจะทำการเปลี่ยน offset ของทางแยกใดทางแยกหนึ่ง แล้วคำนวณหาค่า P.I ของโครงข่ายใหม่ ถ้าค่า P.I ใหม่ที่ได้ลดลงก็จะเปลี่ยนค่า Offset อีก (ในทิศทางเดียวกับค่าแรก) โดยใช้ค่าจำนวนหน่วยรอบเวลาเท่าเดิมกระทำไปจนได้ค่า P.I ต่ำสุด แต่ถ้าในขั้นแรกที่ทำได้ค่า P.I เพิ่มขึ้น ค่า offset ที่ใช้คำนวณต่อไปจะเป็นค่าในทิศทางตรงข้ามกับค่าแรกที่ใช้ ซึ่งจะเปลี่ยนแปลงไปจนได้ P.I ต่ำสุด

6.4.2 การกำหนดจังหวะเวลาไฟเขียวใหม่

ในขบวนการ Hill-climbing นอกจากจะทำการ optimization ค่า offsetแล้ว โปรแกรมทรานซิทยังสามารถ optimization ค่า split หรือจังหวะเวลาของไฟเขียวของแต่ละทางแยกได้ด้วย ในการเปลี่ยนแปลงสัดส่วนไฟเขียว (split) จะกระทำเมื่อการเปลี่ยนแปลงนั้น ทำให้ค่า performance index ลดลง

6.4.3 iteration ของ steps size

การคำนวณในขบวนการ Hill-climb จะเริ่มต้นจากค่าจังหวะเวลาที่กำหนดให้ก่อน (initial setting) ดังนั้นเพื่อลดโอกาสการมีค่า initial setting ที่ไม่เหมาะสม ในขบวนการ Hill-climb จึงจำเป็นต้องมีค่าขนาดขั้นตอนของการเปลี่ยนแปลง (steps size) ทั้งค่า มากและน้อย เพื่อที่จะได้ครอบคลุมค่าที่ควรเปลี่ยนแปลงของการตั้งจังหวะเวลาสัญญาณไฟทั้งหมด ซึ่งจะสามารถช่วยการจัดจังหวะเวลาสัญญาณไฟ ในแต่ละทางแยกได้ค่าที่ดีที่สุด ค่าขนาดของการเปลี่ยนแปลงที่ใช้อยู่ทั่วไปมี 8 ขั้น (steps) ดังแสดงในตารางข้างล่าง

ขั้นที่	ขนาดของการเปลี่ยนแปลง ($\frac{1}{50}$ รอบเวลา)	ชนิดของการ
1	7	เฉพาะ offset
2	20	" offset
3	− 1	" split
4	7	เฉพาะ offset
5	20	" offset
6	− 1	" split
7	1	เฉพาะ offset
8	1	" offset

6.4.4 การจัดกลุ่มของทางแยก

การจัดรวมทางแยกเป็นกลุ่มจะทำเมื่อต้องให้มีการเปลี่ยนแปลง offset พร้อมกันทั้งกลุ่มทางแยก ไม่ใช่เฉพาะทางแยกใดทางแยกหนึ่งจะใช้เมื่อต้องการ optimization พื้นที่ข้างเคียงโดยไม่ต้องการให้มีการเปลี่ยนแปลงค่า offset ของทางแยกที่เป็นเขตเชื่อมต่อระหว่าง 2 พื้นที่นั้น

6.5 การเลือกรอบเวลาสัญญาณไฟจราจร

แม้ว่าในขบวนการ optimization ของโปรแกรมทรานซิท จะไม่มีการ optimize ค่ารอบเวลาสัญญาณไฟจราจรก็ตาม แต่ก็จะสามารถหาค่า P.I ที่น้อยที่สุดของโครงข่ายสัญญาณไฟได้ โดยการเปลี่ยนแปลงค่ารอบเวลาสัญญาณไฟจราจรด้วยค่าต่าง ๆ เพื่อที่จะหาค่ารอบเวลาสัญญาณไฟจราจรที่เหมาะสมที่สุด ซึ่งให้ค่า P.I ต่ำสุดด้วย

หอสมุดคณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

6.6 โปรแกรมคอมพิวเตอร์ ทรานซิท

โปรแกรมทรานซิทเขียนขึ้นด้วยภาษา " Fortran IV " ใช้ภาษาและคำสั่งที่ง่าย ๆ และใช้ได้ทั่วไปกับเครื่องคอมพิวเตอร์แบบต่าง ๆ ในโปรแกรมจะมีคำสั่งประมาณ 1600 คำสั่ง เครื่องคอมพิวเตอร์ที่มีความจุขนาด 16,384 words จะสามารถทำการศึกษาโครงข่ายสัญญาณไฟได้ถึง 165 link หรือ เฉลี่ยประมาณ $3\ ^{1}/2$ link ต่อ 1 ทางแยก (node) ซึ่งคิดเป็นจำนวนทางแยกสูงสุดที่โปรแกรมทรานซิทสามารถทำได้คือ ประมาณ 50 ทางแยก

เวลาเฉลี่ยที่ใช้ในการคำนวณค่าความล่าช้าและการหยุด สำหรับ link ประมาณ 16 มิลิวินาที สำหรับเวลาที่ใช้ในการคำนวณทั้งหมด (T_s) ประมาณได้จากผลคูณของจำนวน link (L) ด้วย จำนวนทางแยก (node, N) ดังค่าประมาณจากสูตร

$$T_s \approx 0.008\ NL$$

ในปัจจุบันโปรแกรมทรานซิท ที่ปรับปรุงครั้งล่าสุด คือ " TRANSYT/7 " ได้ปรับปรุงตัวโปรแกรมให้มีการใช้เวลาคำนวณโดยคอมพิวเตอร์ ให้น้อยลงกว่าโปรแกรมเดิม (TRANSYT/6)

6.7 Version ของโปรแกรมทรานซิท

หลังจากที่ TRRL ได้นำโปรแกรมทรานซิทไปใช้งาน ก็ได้มีการปรับปรุงโปรแกรมอยู่เรื่อยมา ดังนี้คือ

TRANSYT/1 (1967)

เป็นตัวโปรแกรมเริ่มแรก เขียนขึ้นเพื่อใช้กับ Myraid computer ที่เมือง Glasgow

TRANSYT/2 (1968)

ปรับปรุง TRANSYT/1 เขียนเป็นภาษา FortranIV และสามารถใช้ได้กับทางแยกสัญญาณไฟที่มีมากกว่า 3 จังหวะ

TRANSYT/3 (1970)

ปรับปรุงระบบการให้และตรวจสอบข้อมูล (input data)

TRANSYT/4 (1971)

ปรับปรุงให้มีการคำนวณค่า initial setting และมีการเขียนกราฟแสดงรูปแบบการไหลของการจราจร

TRANSYT/5 (1972)

ปรับปรุงเกี่ยวกับด้าน bus ให้มีการคิด bus progression speed และ bus stopped time ของแต่ละ link

TRANSYT/6 (1975)

ปรับปรุง TRANSYT/5 เกี่ยวกับรูปจำลองของการหยุด และพยายามลดเวลาที่ใช้ในการคำนวณด้วยเครื่องคอมพิวเตอร์ให้น้อยลง

TRANSYT/7 (1978)

เป็น version ล่าสุด ที่มีการลดเวลาที่ใช้คำนวณให้น้อยลงอีก สำหรับโครงข่ายขนาด 130 links สามารถเวลาการคำนวณลงถึง 70% (เปรียบเทียบกับใช้ TRANSYT/6)