บทที่ 6

การจำลองการใช้ลิฟต์ของผู้โดยสารในอาคาร

6.1. ลักษณะหรือพฤติกรรมการใช้ลิฟต์ของผู้โดยสาร

กลุ่มคนที่อยู่ในอาคารขนาดใหญ่ ย่อมต้องการใช้ลิฟต์เพื่อเดินทางจากชั้นหนึ่งไปยังอีกชั้นหนึ่งอยู่เสมอ ลักษณะหรือพฤติกรรมการใช้ลิฟต์ของผู้โดยสารคนหนึ่ง ๆ โดยปกติมีขั้นตอนเป็นดังนี้ [10]

- เดินทางมาที่บริเวณหน้าลิฟต์
- กดเรียกลิฟต์ที่ปุ่ม hall call up หรือ hall call down ในกรณีที่ต้องการไปยังชั้นที่สูงกว่าหรือต่ำกว่า ตามลำดับ ในกรณีที่มีผู้โดยสารอื่นกดเรียกอยู่ก่อนแล้ว ก็ไม่กดเรียกซ้ำอีก
- รอลิฟต์มารับ ในกรณีที่มีลิฟต์หลายตัว ผู้โดยสารจะสังเกตที่แผงไฟบอกตำแหน่งลิฟต์และรอลิฟต์ตัวกำลังเดินทางมารับ หรือในกรณีที่ไม่มีแผงไฟบอกตำแหน่ง ผู้โดยสารจะรอจนกว่ามีสัญญาณเตือนการจอด (hall lantern) และมีสัญญาณไฟบอกให้ทราบว่าลิฟต์ตัวใดจะมารับ จึงจะเดินไปยังหน้าลิฟต์ตัวนั้น
- เมื่อลิฟต์มาถึงและประตูเปิดออก ผู้โดยสารเข้าไปในลิฟต์ กดปุ่มบังคับให้ประตูปิดหรือรอจนกว่าประตูปิดเอง และกดปุ่ม car call ภายในตัวลิฟต์ที่ชั้นที่ต้องการจะไป โดยปกติชั้น car call ที่กดจะเป็นชั้นที่สูงกว่าหรือต่ำกว่า ขึ้นอยู่กับว่าผู้โดยสารกดเรียกด้วย hall call up หรือ hall call down
- เมื่อลิฟต์เคลื่อนไปหยุดที่ชั้นตรงกับ car call ที่กดไว้ ประตูเปิดออก ผู้โดยสารจะเดินออกนอกตัวลิฟต์ เป็นการจบสิ้นการใช้ลิฟต์ของผู้โดยสารคนหนึ่ง

จากที่กล่าวผ่านมา จะเห็นได้ว่า ถ้าต้องการจะจำลองการใช้ลิฟต์ของผู้โดยสารควรจะให้ความสนใจข้อมูลเหล่านี้

1) เวลาที่ผู้โดยสารมาถึงหน้าลิฟต์ และกดเรียก hall call
2) ชั้นที่ผู้โดยสารมาถึง หรือ ชั้นที่กด hall call
3) ผู้โดยสารเดินทางขึ้นหรือลง (hall call up หรือ hall call down)
4) ชั้นที่ผู้โดยสารต้องการจะไป หรือชั้นที่กด car call
5) ช่วงเวลาที่กด car call นับจากลิฟต์เดินทางมารับแล้ว

(เวลาการรอลิฟต์ของผู้โดยสาร ไม่ได้ถือเป็นข้อมูลพฤติกรรมการใช้ลิฟต์ของผู้โดยสาร เพราะขึ้นกับการทำงานของระบบลิฟต์เอง)

ตัวอย่างข้อมูลการใช้ลิฟต์ของผู้โดยสารอาจเป็นดังนี้

Passenger	Arriving Time	Direction	Origin Floor	Destination Floor	Delay
#1	11:23:34	Up	1	6	15
#2	11:23:38	Down	9	7	6
. .	. . . . .	. . .	. .	. .	. .

ข้อมูลการใช้ลิฟต์ของผู้โดยสาร 1 คนเรียกว่าเป็น 1 record ข้อมูลที่บรรจุอยู่ใน 1 record ได้แก่ เวลาการมาถึง (arriving time), ทิศทางขึ้นหรือลง (direction), ชั้นเริ่มต้น (origin floor), ชั้นเป้าหมาย (destination floor), และช่วงเวลาที่กด car call (delay) ความสัมพันธ์ระหว่าง record เหล่านี้ จะแสดงถึงความคับคั่ง และปริมาณการใช้ลิฟต์ที่ชั้นต่างๆ ว่ามีมากน้อยเพียงไร เรียกว่า ทราฟฟิก (traffic) ซึ่งจะได้กล่าวถึงในหัวข้อถัดไป การกำหนดค่าตัวแปร ได้แก่ ค่าเวลา หมายเลขชั้น ต่าง ๆ ให้ใกล้เคียงลักษณะการใช้ลิฟต์ของอาคารในช่วงหนึ่ง จะต้องใช้หลักการและสมการทางสถิติเป็นตัวกำหนดค่า

6.2. ทราฟฟิกของลิฟต์ (Elevator Traffic)

ทราฟฟิกของลิฟต์ คือ ลักษณะความหนาแน่น ความคับคั่ง หรืออัตราการใช้ลิฟต์ของกลุ่มผู้โดยสารในอาคารในช่วงเวลาหนึ่ง โดยปกติจะแบ่งลักษณะของทราฟฟิกออกเป็น 4 ชนิด [2] ดังนี้

1) Incomming Traffic หรือ Up Peak Traffic เป็นทราฟฟิกที่มีการกดเรียกลิฟต์ที่ชั้น 1 มาก เพื่อต้องการเดินทางขึ้นไปยังชั้นต่างๆ โดยปกติสำหรับอาคารที่เป็นที่ตั้งของสำนักงานธุรกิจ จะเกิดขึ้นเวลาเช้าก่อนเวลาเข้าทำงาน

2) Outgoing Traffic หรือ Down Peak Traffic เป็นทราฟฟิกที่มีการกดเรียกจากชั้นต่างๆ เพื่อต้องการลงมาที่ชั้น 1 มาก โดยปกติสำหรับอาคารที่เป็นที่ตั้งของสำนักงานธุรกิจ จะเกิดขึ้นเวลาเย็นซึ่งเป็นเวลาเลิกงาน

3) Quiet Traffic เป็นทราฟฟิกที่มีปริมาณการใช้ลิฟต์น้อย โดยมากมักเป็นเวลากลางคืนหรือวันหยุด ระบบลิฟต์ที่มีความสามารถพิเศษจะมีส่วนจัดการดับไฟแสงสว่างและพัดลมในตัวลิฟต์ เมื่อเครื่องควบคุมตรวจพบว่าเกิด Quiet Traffic ขึ้น เป็นการประหยัดพลังงานไฟฟ้า

4) Two-Way Traffic หรือ Normal Day Traffic เป็นทราฟฟิกปกติที่เกิดขึ้นตลอดทั้งวัน ยกเว้นช่วงเวลาที่เกิดทราฟฟิกต่างๆ ที่กล่าวมาแล้ว Two-way Traffic แบ่งเป็น Two-way Lobby Traffic เป็น Traffic ที่มีการขึ้น-ลงที่ชั้นห้องโถง (หรือชั้นที่ตั้งของที่รับประทานอาหารกลางวัน) มาก ได้แก่ เวลาพักเพื่อรับประทานอาหารกลางวัน เป็นต้น และ Two-way Interfloor Traffic เป็น Traffic ที่มีการขึ้น-ลงจากชั้นหนึ่งไปอีกชั้นหนึ่ง โดยไม่จำเพาะที่ชั้นห้องโถง

การออกแบบวิธีการเลือกส่งลิฟต์ในเครื่องควบคุมกลุ่มลิฟต์ จะต้องมีความสามารถขั้นพื้นฐานในการบริการ Two-way Traffic ได้เป็นอย่างดี ส่วนสำหรับทราฟฟิกอื่นๆ เป็นความสามารถพิเศษเฉพาะตัวของเครื่องควบคุมในระบบลิฟต์ ซึ่งจะต้องมีความสามารถตรวจสอบอยู่ตลอดเวลาว่าเกิดทราฟฟิกชนิดใดขึ้นแล้วหรือไม่ และจะปรับวิธีการเลือกส่งลิฟต์อย่างไร เพื่อตอบสนองการเปลี่ยนแปลงของทราฟฟิกที่เกิดขึ้น

ดังนั้น ในการพัฒนาวิธีการเลือกส่งลิฟต์ที่จะต้องคำนึงถึงทราฟฟิกชนิดต่างๆ จึงจำเป็นจะต้องมีเครื่องมือที่สามารถสร้างข้อมูลการใช้ลิฟต์เลียนแบบหรือจำลองแบบการใช้ลิฟต์ในอาคารจริงได้ เพื่อป้อนให้กับโปรแกรมจำลองระบบลิฟต์ (ELSIM) ใช้ตรวจสอบวิธีการเลือกส่งลิฟต์ที่พัฒนาขึ้น ในโครงงานวิทยานิพนธ์นี้ ได้ทำการสร้างโปรแกรมสร้างชุดข้อมูลการใช้ลิฟต์ คือ TFGEN เพื่อใช้ในการนี้

6.3. โปรแกรมสร้างชุดข้อมูลการใช้ลิฟต์ (Traffic Generator Program)

โปรแกรมสร้างชุดข้อมูลการใช้ลิฟต์ TFGEN (Traffic Generator) เป็นโปรแกรมทำหน้าที่สร้างไฟล์ข้อมูลการใช้ลิฟต์ที่จะป้อนให้โปรแกรมจำลองระบบลิฟต์ ข้อมูลที่เก็บไว้จะแบ่งเป็น record แต่ละ record แทนพฤติกรรมการใช้ลิฟต์ของผู้โดยสารลิฟต์ 1 คน ซึ่งประกอบด้วย

1) arriving time เวลาเข้าสู่ระบบ คือ เวลาที่ผู้โดยสารมาถึงหน้าลิฟต์และกดเรียก hall call
2) direction ชนิดของ hall call ที่กดเรียก ได้แก่เรียกขึ้น (Up) หรือ เรียกลง (Down)
3) arriving hall ชั้นที่กดเรียก คือชั้นที่กด hall call
4) delay ช่วงเวลาตั้งแต่เวลาที่ลิฟต์มาถึงชั้นที่กดเรียก และเวลาที่ผู้โดยสารที่เข้าไปในลิฟต์แล้วกด car call
5) destination hall ชั้นเป้าหมาย หมายถึงชั้นที่กด car call เมื่อผู้โดยสารเข้าไปในลิฟต์แล้ว

ตารางที่ 6.1 แสดงตัวอย่างข้อมูลที่ได้จากโปรแกรมสร้างชุดข้อมูล สำหรับ record แรก อธิบายได้ว่า มีผู้ใช้ลิฟต์มาถึงหน้าลิฟต์ที่ชั้น 13 เวลา 00:00:03 น. ผู้ใช้ลิฟต์นี้ต้องการลงไปชั้น 4 โดยถ้าลิฟต์มารับแล้ว ผู้ใช้ลิฟต์จะใช้เวลาช่วงหนึ่งก่อนที่จะกดเรียก car call ชั้น 4

6.3.1. ข้อมูลที่ป้อนให้โปรแกรมสร้างชุดข้อมูลการใช้ลิฟต์

การใช้โปรแกรมสร้างชุดข้อมูลการใช้ลิฟต์ ผู้ใช้จะต้องป้อนข้อมูลเพื่อกำหนดลักษณะของชุดข้อมูลที่ต้องการสร้าง ดังนี้

1) ชื่อไฟล์ที่เก็บข้อมูล

2) จำนวนครั้งของการเรียกลิฟต์ ซึ่งก็คือ จำนวน reccord ทั้งหมด

3) จำนวนชั้นในระบบลิฟต์

4) เปอร์เซ็นหรืออัตราส่วนการใช้ลิฟต์เดินทางขึ้นจากชั้น 1 เทียบกับการเรียกใช้ลิฟต์ของทุกชั้น ตัวแปรที่ใช้เก็บค่า คือ a

5) เปอร์เซ็นหรืออัตราส่วนการใช้ลิฟต์เดินทางลงยังชั้น 1 เทียบกับการลงลิฟต์ของทุกชั้น ตัวแปรที่ใช้เก็บค่า คือ b

6) เปอร์เซ็นหรืออัตราส่วนการใช้ลิฟต์เดินทางขึ้น-ลงที่ไม่เกี่ยวข้องกับชั้น 1 เทียบกับการลงลิฟต์ของทุกชั้น ตัวแปรที่ใช้เก็บค่า คือ c (ค่านี้คำนวนได้จาก c = 100 - a - b)

7) ค่าช่วงเวลาเฉลี่ยของการเรียกลิฟต์ของผู้โดยสาร (วินาที/ครั้ง) ตัวแปรที่ใช้ คือ λ

ตารางที่ 6.1 ตัวอย่างข้อมูลที่ได้จากโปรแกรมสร้างชุดข้อมูล

ARRIVE TIME	CALL DIRECTION	ARRIVE HALL	2ND CALL DELAY	DESTINATION HALL
00:00:03	down	13	1	4
00:00:05	down	3	3	0
00:00:07	up	3	2	9
00:00:10	up	2	6	5
00:00:13	down	12	3	11
00:00:15	down	6	5	0
00:00:18	down	7	3	1
00:00:20	down	3	1	0
00:00:24	up	0	0	8
00:00:26	down	13	3	3
00:00:31	down	4	5	1
00:00:32	down	15	4	13
00:00:38	up	7	2	13
00:00:40	up	10	3	11
00:00:42	down	11	1	9
00:00:43	up	7	4	11
00:00:45	up	2	3	4
00:00:47	up	3	2	4
00:00:49	up	2	2	7
00:00:53	up	6	2	11
00:00:53	down	14	3	2
00:00:57	up	3	5	10
00:01:00	up	1	3	13
00:01:04	up	11	2	14

6.3.2. การคำนวนเบื้องต้น

1) การคำนวนการกระจายของเวลา

การคำนวนเวลา arriving time และ delay time ที่ใช้ในโปรแกรมนี้ ใช้หลักการสร้างข้อมูลสุ่มเป็นการกระจายแบบปัวส์ซอง [21] (Poisson's Distribution) ดังนี้

ความน่าจะเป็นที่ผู้โดยสาร n คนจะมาในช่วงเวลา T [10]

$$= (\lambda T)^n e^{-\lambda T}/n! \qquad \text{.....(6.1)}$$

โดย λ = อัตราการมาถึง (จำนวนผู้โดยสาร/เวลา)

จากสมการที่ 6.1 จะเห็นว่า การกระจายของเวลาเป็นแบบ Negative Exponential โดยมีค่า λ เป็นค่าเฉลี่ยของเวลาระหว่างการมาถึงของผู้โดยสาร จากสมการที่ 6.1 นี้ ถ้าแปลงให้อยู่ในรูปของเวลาการมาถึงของผู้โดยสารแต่ละคนจะได้

$$t_o = 0$$
$$t_i = t_{i-1} + \{ -\ln(r)/\lambda \} \quad i = 1,2,3,.. \qquad \text{.....(6.2)}$$

โดยที่ t_i = เวลาการมาถึงของผู้โดยสารคนที่ i

r = ค่าสุ่ม (random) ที่อยู่ในช่วง (0.000, 1.000)

สมการ 6.2 เป็นสมการที่นำค่า λ กระจายเป็นเวลาการมาถึงออกเป็น t_o, t_1, t_2, ... โดยใช้ตัวแปรสุ่ม (Random)

2) การคำนวนความหนาแน่นของการใช้ลิฟต์

ที่กล่าวผ่านมาเป็นการคำนวนการกระจายของเวลาการมาใช้ลิฟต์ ซึ่งยังไม่ได้เกี่ยวข้องหรือระบุถึงปริมาณการใช้ลิฟต์ที่ชั้นต่าง ๆ ซึ่งจะมีความหนาแน่นแตกต่างกันไป การอธิบายถึงความหนาแน่นของชั้นที่ผู้โดยสารขึ้น-ลง สามารถอธิบายด้วย Origin Density Vector และ Origin-Destination Matrix [10] ดังนี้

Origin Density Vector หรือเรียกย่อว่า OV คือ vector ที่อธิบายถึงความหนาแน่นของการขึ้นลิฟต์ที่แต่ละชั้น โดย element ที่ i ของ OV มีค่าเป็นเปอร์เซ็นต์ที่ผู้โดยสารจะขึ้นลิฟต์ที่ i เทียบกับทุกชั้น ตัวอย่าง OV ในรูปที่ 6.1 แสดงว่าความหนาแน่นของผู้ใช้ลิฟต์จะขึ้นจากชั้น 1, 2, 3 และ 4 เป็น 40%, 24%, 12% และ 24% ตามลำดับ

Origin-Destination Matrix หรือเรียกย่อว่า ODM คือ matrix ที่อธิบายถึงความหนาแน่นของการเดินทางจากชั้นหนึ่งไปยังอีกชั้นหนึ่ง โดย element ที่ i-j ของ ODM มีค่าเป็นเปอร์เซ็นต์ที่ผู้โดยสารจะขึ้นลิฟต์จากชั้น i และลงที่ชั้น j เทียบกับการขึ้นลิฟต์ที่ชั้น i ทุกครั้ง ตัวอย่าง ODM ดังรูป 6.2

การหาค่า OV และ ODM จะต้องทราบปริมาณการขึ้น-ลงที่ชั้นต่างๆ ซึ่งสามารถแบ่งได้เป็น 3 ชนิดด้วยกัน คือ

a) ผู้โดยสารที่เดินทางจากชั้น 1
b) ผู้โดยสารที่เดินทางลงไปยังชั้น 1
c) ผู้โดยสารที่เดินทางขึ้น-ลงระหว่างชั้นอื่นๆ ที่ไม่เกี่ยวข้องกับชั้น 1

และตัวแปรที่ใช้มีความหมายดังนี้

	i = 1	i = 2	i = 3	i = 4
ขึ้นลิฟท์จากชั้น i:	40%	24%	12%	24%

รูปที่ 6.1 - Origin Density Vector (OV)

ไปยังชั้น

		j = 1	j = 2	j = 3	j = 4
ขึ้นลิฟท์จากชั้น	i = 1	0	40	20	40
	i = 2	67	0	11	22
	i = 3	67	16	0	17
	i = 4	67	22	11	0

รูปที่ 6.2 - Origin-Destination Matrix (ODM)

a = % ของผู้โดยสารทั้งหมดที่เป็นชนิด (a)

b = % ของผู้โดยสารทั้งหมดที่เป็นชนิด (b)

c = % ของผู้โดยสารทั้งหมดที่เป็นชนิด (c)

และ a + b + c = 100

ov[i] = % ของผู้โดยสารที่ขึ้นลิฟต์จากชั้น i

odm[i][j] = % ของผู้โดยสารที่ขึ้นลิฟต์จากชั้น i และลงลิฟต์ที่ชั้น j

เมื่อกำหนดให้ N เป็นจำนวนชั้นทั้งหมดของอาคาร และมีหมายเลขชั้นเป็น 1, 2, 3,.., N จะหา Origin Density Vector (OV) และ Origin-destination Matrix (ODM) [10] ได้ดังนี้

$$ov[1] = a$$

$$ov[i] = (b+c) / (N-1) \qquad \text{เมื่อ } i = 2,3,..,N$$

$$odm[1][j] = \begin{cases} 0 & \text{เมื่อ } j = 1 \\ 100 / (N-1) & \text{เมื่อ } j = 2,3,..,N \end{cases}$$

$$odm[i][1] = \begin{cases} 0 & \text{เมื่อ } i = 1 \\ 100\ b / (b+c) & \text{เมื่อ } i = 2,3,..,N \end{cases}$$

$$odm[i][j] = \begin{cases} 0 & \text{เมื่อ } i = j \\ 100\ c / (b+c)(N-2) & \text{เมื่อ } i \neq j \end{cases}$$

ค่าของ OV และ ODM ที่ได้ดังรูปที่ 6.1 และ 6.2 ได้จากการกำหนดให้ a = 40, b = 40, c = 20 สำหรับอาคาร 4 ชั้น ที่มีประชากรที่ชั้น 1 ถึงชั้น 4 เป็น 50, 200, 100 และ 200 ตามลำดับ

Origin Density Vector (OV) และ Origin-destination Matrix (ODM) ของอาคารหนึ่ง จะมีค่าไม่คงที่ ทั้งนี้เนื่องจากปริมาณการใช้ลิฟต์ที่ชั้นต่างในช่วงเวลาตลอด 1 วันมีการเปลี่ยนแปลงไป ในการสร้างข้อมูลจำลองการใช้ลิฟต์ ถ้าต้องการให้ได้ข้อมูลของทราฟฟิกในช่วงเวลาต่างๆ ทำได้โดยการกำหนด พารามิเตอร์ a, b และ c ให้ตรงกับทราฟฟิกในช่วงเวลานั้นๆ ตัวอย่างการกำหนดพารามิเตอร์ a, b และ c ในช่วงเวลาต่างๆ อาจเป็นดังนี้

ช่วงเวลา	A	B	C
เช้า	90	5	5
สาย	45	45	10
กลางวัน (ช่วงแรก)	20	60	20
กลางวัน (ช่วงหลัง)	70	10	20
เย็น	5	90	5

6.3.3. ขั้นตอนของโปรแกรม

ในรูปที่ 6.3 แสดงโปรแกรมสร้างชุดข้อมูลการใช้ลิฟต์ที่พัฒนาด้วยภาษา C โดยย่อ โครงสร้างของโปรแกรมแบ่งเป็นส่วนๆ ได้แก่ initial(), from_user(), cons_mtx(),gen_dat() และ complete() อธิบายได้ดังนี้

เริ่มต้นด้วยการทำส่วน initial() ได้แก่ initial ตัวแปร และล้างจอภาพ ต่อจากนั้น จะให้ผู้ใช้ป้อนชื่อไฟล์ที่ต้องการเก็บข้อมูลและถามความต้องการพิมพ์ข้อมูลออกเครื่องพิมพ์ ในกรณีที่ต้องการพิมพ์ออกเครื่องพิมพ์จะทำการพิมพ์ส่วนหัวของข้อมูลออกเครื่องพิมพ์

ขั้นตอนต่อไป คือ from_user() เป็นการให้ผู้ใช้กำหนดลักษณะของแบบข้อมูลที่ต้องสร้าง ข้อมูลที่รับได้แก่

- จำนวน Calls
- จำนวนชั้น
- a : สัดส่วนของประชากรที่ขึ้นลิฟต์ที่ชั้น 1
- b : สัดส่วนของประชากรที่ลงลิฟต์ที่ชั้น 1
- c : สัดส่วนของประชากรขึ้น-ลงชั้นอื่นๆ (c = 100 - a - b)
- λ : ช่วงเวลาการกดเรียกเฉลี่ย (mean_arrive)

ส่วนฟังก์ชั่น cons_mtx() จะใช้ข้อมูล a, b และ c ที่ได้รับ คำนวน OV และ ODM ตามวิธีการคำนวนที่ได้กล่าวมาแล้ว ต่อจากนั้นปรับปรุง OV และ ODM เป็น OVX และ ODMX โดย

```
ovx[1] = ov[1]
ovx[i] = ovx[i-1] + ov[i]          i = 2, 3, 4, ...
```

```
odmx[i][1] = odm[i][1]
odmx[i][j] = odmx[i][j-1] + odmx[i][j]      j = 2, 3, 4, ...
```

ค่า ovx[i] และ odmx[i][j] ต่างเป็นค่าสะสมของ ov[i] และ odm[i][j] ที่ตำแหน่งข้อมูลน้อยกว่าหรือเท่ากับ i ตามลำดับ

ขั้นตอนต่อไปคือ generate() ทำหน้าที่สร้าง เก็บ และพิมพ์ข้อมูล การทำงานจะเป็นโปรแกรมวงรอบ ทำการสร้างข้อมูลตั้งแต่ record 0 ถึง record ที่ต้องการ ในแต่ละวงรอบจะเรียกฟังก์ชั่น gen_data() เพื่อสร้างข้อมูล, print_data() เมื่อมีการสั่งให้พิมพ์ข้อมูลออกเครื่องพิมพ์ และ save_data() สำหรับเก็บข้อมูลใส่ไฟล์ที่กำหนดไว้

ข้อมูล OVX และ ODMX ที่สร้างได้ และข้อมูลเวลาการกดเรียกเฉลี่ย (mean_arrive) จะถูกใช้ในฟังก์ชั่น gen_data() เพื่อสร้างข้อมูลการใช้ลิฟต์ ได้แก่ เวลาการกดเรียก (arr_time), ชั้นเริ่มต้น (org_floor), ชั้นเป้าหมาย (des_floor), เวลาระหว่างลิฟต์จอดรับกับเวลาการกด car call (des_delay) และชนิดของการเรียกหรือทิศทาง (direction) วิธีการสร้างข้อมูลแต่ละตัวเป็นตามขั้นตอนของฟังก์ชั่น generate() ซึ่งสามารถอธิบายได้ดังนี้

arr_time ใช้การเรียกฟังชั่น poisson(mean_arrive) เพื่อหาค่าสุ่มแบบที่มีการกระจายแบบปัวซ์ซองที่มีค่าเฉลี่ยเป็น mean_arrive เมื่อได้ค่าแล้วนำไปบวกกับ arrive_time ของวงรอบที่แล้ว

org_floor ใช้การเรียกฟังชั่น random เพื่อหาค่าสุ่มมีทศนิยม 3 ตำแหน่ง แล้วนำไปหาว่าค่าที่ได้อยู่ในช่วงค่าที่ i เท่าใด ใน Origin density vector OVX ค่า i นั้นคือ org_floor

des_floor ใช้การเรียกฟังชั่น random เพื่อหาค่าสุ่มมีทศนิยม 3 ตำแหน่ง แล้วนำไปหาว่าค่าที่ได้อยู่ในช่วงค่าที่ j เท่าใด ใน Origin-destination matrix ODMX ที่ค่า i คือ org_floor ค่า j นั้นคือ des_floor

des_delay ใช้การเรียกฟังชั่น poisson(mean_delay) เพื่อหาค่าสุ่มแบบที่มีการกระจายแบบปัวซ์ซองที่มีค่าเฉลี่ยเป็น mean_delay ค่าที่ได้คือ des_delay

direction ใช้การเปรียบเทียบค่า org_floor กับ des_floor ถ้า org_floor น้อยกว่า des_floor ค่าที่ได้ คือ 0 (กดขึ้น) ถ้ามากกว่า ค่าที่ได้ คือ 1 (กดลง)

เมื่อได้ค่าทั้งหมดแล้ว ก็จะทำการเก็บใส่ไฟล์ด้วยฟังก์ชั่น save_data() และในกรณีที่มีการกำหนดให้พิมพ์ออกเครื่องพิมพ์ ข้อมูล 1 record นี้ ก็จะพิมพ์ออกเครื่องพิมพ์ และวกกลับไปทำในวงรอบถัดไป จนได้ record ครบตามที่กำหนดไว้

ส่วนสุดท้ายของโปรแกรม คือ complete() ทำหน้าที่ปิดไฟล์ และพิมพ์ข้อความแจ้งการทำงานเสร็จสิ้น

```
.....
.....
.....

/* User's input data */
int a,b,c,max_record,max_floor,mean_arrive,mean_delay,print_switch;

/* Origin density vector & Origin-destination matrix */
float ov[NORM_FLOOR],odm[NORM_FLOOR][NORM_FLOOR],ovx[NORM_FLOOR],
     odmx[NORM_FLOOR][NORM_FLOOR];

/* Data to save */
long arr_time,des_delay;
int  direction,org_floor,des_floor;


main()
{
   initial();          /* Initial */

   from_user();        /* Get data from user */

   cons_mtx();         /* Make Origin Vector
                          & Origin Destination Matrix */
   gen_dat();          /* Generate, print and save data into file */

   complete();         /* Before end program */

}


/* Initial program */
initial()
{
   .....
   .....
}


/* Get input data from user */
from_user()
{
   .....
   .....
}
```

รูปที่ 6.3 – โปรแกรมสร้างชุดข้อมูลการใช้ลิฟต์ TFGEN.C

```
/* Construct Origin Vector (ovx[]) & Origin-Destination Matrix (odmx[]) */
cons_mtx()
{
   float a,b,c,last,last2;
   int i,j;
   a*=10.0; b*=10.0; c*=10.0;
   ov[0]=a;
   for (i=1; i<max_floor; ++i)
        ov[i]=(b+c)/(max_floor-1);
   odm[0][0]=0.0;
   for (j=1; j<max_floor; ++j)
        odm[0][j]=1000.0/(max_floor-1);
   for (i=1; i<max_floor; ++i)
        odm[i][0]=1000.0*(b/(b+c));
   for (i=1; i<max_floor; ++i) {
        for (j=1; j<max_floor; ++j) {
             if (i==j)
                  odm[i][j]=0.0;
             else
                  odm[i][j]=1000*c/((b+c)*(max_floor-2));
        }
   }
   last=0.0;
   for (i=0; i<max_floor; ++i) {
       ovx[i]=last+ov[i];
       last=ovx[i];
       last2=0.0;
       for (j=0; j<max_floor; ++j) {
            odmx[i][j]=last2+odm[i][j];
            last2=odmx[i][j];
       }
       odmx[i][max_floor-1]=1000.0;
   }
   ovx[max_floor-1]=1000.0;
}


/* Generate, print and save data into file */
gen_dat()
{
   for (record=0; record<max_record; ++record) {
        gen_data();                  /* Generate data */
        print_data(device);          /* Print data on console or printer */
        save_data();                 /* Save one record of data */
   }
}
```

รูปที่ 6.3 (ต่อ)

```
/* Close file & Display some messages */
complete()
{
   close(s_file);                    /* Close file */
   printf("Complete in generating of %d passenger calls.\n",max_record);
   printf("Traffic data store in file ' %s '.",f_name);
}


/* Generate data */
generate()
{
   float x;
   int ps_ran(),i,j;
   arr_time+=ps_ran(mean_arrive);
   x=random(1000);
   i=0;
   while (x>ovx[i])
         ++i;
   org_floor=i;
         x=random(1000);
   j=0;
   while (x>odmx[org_floor][j])
         ++j;
   des_floor=j;
   if (org_floor<des_floor)
         direction=0;
   else
         direction=1;
   des_delay=ps_ran(belay);
}
```

รูปที่ 6.3 (ต่อ)

```
/* Use  Poisson's formula to generate arrival time */
int ps_ran(l)
int l;
{
   int output,r1,i;
   float x,s,r,lamda;
   lamda=l;
   x=exp(-lamda);
   s=1.0;
   for (output=0; s>=x;  ++output) {
         r1=random(1000);
         r=r1/1000.0;
         s*=r;
   }
   return(output-1);
}


/* Print data */
print_data(dev)
FILE *dev;
{
   ..........................
   ..........................
}


/* Save data */
save_data()
{
   ..........................
   ..........................
}
```

รูปที่ 6.3 (ต่อ)

6.3.4. ตัวอย่างการใช้งานโปรแกรมสร้างชุดข้อมูลการใช้ลิฟต์

ข้อกำหนด : ต้องการสร้างข้อมูลการใช้ลิฟต์สำหรับอาคาร 16 ชั้น ที่มีจำนวนประชากรหนาแน่นเท่ากันทุกชั้นในช่วงเวลาหนึ่ง สำหรับผู้โดยสาร 100 คน ที่มีช่วงเวลาการกดเรียกลิฟต์เป็น 3 วินาที เวลาการกด car call เมื่อลิฟต์มารับแล้วเป็น 3 วินาที และมีปริมาณการใช้ลิฟต์ดังนี้

มีการขึ้นลิฟต์จากชั้น 1 10%
มีการลงลิฟต์ที่ชั้น 1 10%
มีการขึ้นลงลิฟต์ที่ชั้นอื่นๆ นอกเหนือจากชั้น 1 80%

ขั้นตอนการใช้โปรแกรม TFGEN :

1) เรียกโปรแกรม TFGEN

```
A>tfgen
```

2) บนจอภาพจะขึ้นข้อความเพื่อถามความต้องการต่าง ๆ ตามรูปที่ 6.4 และตามตัวอย่างนี้ จะต้องป้อนข้อมูลเป็นดังที่ขีดเส้นใต้ไว้ ได้แก่

ชื่อไฟล์ที่ต้องการเก็บชุดข้อมูล คือ TEST1.TRF
ต้องการพิมพ์ชุดข้อมูลออกเครื่องพิมพ์ด้วย
จำนวนการกดเรียกหรือ จำนวน record คือ 100
จำนวนชั้นอาคาร คือ 16 ชั้น
เวลาการกดเรียกเฉลี่ย 3 วินาที
ช่วงเวลาก่อนกด car call เฉลี่ย 3 วินาที
สัดส่วนของการใช้ลิฟต์ที่ขึ้นจากชั้น 1 (a) 10%
สัดส่วนของการใช้ลิฟต์ที่ลงไปชั้น 1 (b) 10%

สัดส่วนการใช้ลิฟต์ที่นอกเหนือชั้น 1 หรือ ค่า c ไม่จำเป็นต้องป้อน เพราะ ค่า c คือ 100 - a - b

3) เพื่อป้อนข้อมูลครบตามที่กำหนดแล้ว โปรแกรมจะแสดง Orgin density vector และ Origin-destination metrix ทางจอภาพ จากนั้นจะสร้างข้อมูลการใช้ลิฟต์ และเก็บข้อมูลลงไฟล์ที่กำหนดไว้ ในที่นี้ กำหนดให้พิมพ์ออกเครื่องพิมพ์ ดังนั้นข้อมูลการใช้ลิฟต์จึงพิมพ์ออกเครื่องพิมพ์ได้ตามรูปที่ 6.5 เป็นอันจบสิ้นการทำงานของโปรแกรม

```
Name of save file : TEST.TRF
Do you want print out ? (if answere 'yes',press 'y') : Y
Number of calls [20,500] : 100
Number of halls [4,16] : 16
Mean arrive time (3-30) : 3
Mean delay time (1-10) : 3
a: Up travelling from main floor [0,100]: 10
b: Down travelling to main floor [0,100-U]: 10
c: Up and down travelling above main floor [100-U-D]: 80
```

รูปที่ 6.4 - การป้อนข้อมูลกำหนดลักษณะชุดข้อมูลการใช้ลิฟท์

PASSENGER NO.	ARRIVE TIME	CALL DIRECTION	ARRIVE HALL	2ND CALL DELAY	DESTINATION HALL
001	00:00:03	down	15	4	4
002	00:00:04	up	1	2	15
003	00:00:07	down	14	6	1
004	00:00:10	down	14	1	4
005	00:00:13	up	4	3	10
006	00:00:15	up	0	4	8
007	00:00:18	up	0	2	11
008	00:00:20	up	1	2	6
009	00:00:24	down	13	5	5
010	00:00:28	up	3	0	14
011	00:00:32	up	0	3	13
012	00:00:35	down	14	7	0
013	00:00:37	up	0	2	15
014	00:00:39	down	8	5	0
015	00:00:43	up	5	4	11
...	.....	....	..	...	..
...	.....	....	..	...	..
...	.....	....	..	...	..
...	.....	....	..	...	..
495	00:24:56	down	12	1	7
496	00:25:01	up	5	4	11
497	00:25:04	up	5	4	11
498	00:25:04	up	7	5	14
499	00:25:06	down	14	3	4
500	00:25:10	up	1	5	11

รูปที่ 6.5 - ข้อมูลการใช้ลิฟท์ที่พิมพ์ออกทางเครื่องพิมพ์