รายการอ้างอิง

ภาษาไทย

โครงการอบรมเพื่อพัฒนาอุตสาหกรรมไฟฟ้าและอิเล็กทรอนิกส์. การใช้งานโปรแกรมประยุกต์ MATLAB. มหาวิทยาลัยเทคโนโลยีมหานคร, 2542.

ชนะ เยี่ยงกมลสิงห์. การประยุกต์ใช้เจนเนติกอัลกอริทึมในการออกแบบผังโรงงาน. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2541.

พิภพ เล้าประจง. ระบบการวางแผนและควบคุมการผลิต. พิมพ์ครั้งที่ 3, สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น), กรุงเทพมหานคร, 2541. หน้า 459-479.

ศิริจันทร์ ทองประเสริฐ. การจำลองแบบปัญหา. กรุงเทพมหานคร: โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2537.

สุธรรม ศรีเกษม, สง่า ศุภปรีดา, กิติ ศรีนุชศาสตร์, ปรีชา วงษ์ษา. MATLAB เพื่อการแก้ปัญหาทางวิศวกรรมศาสตร์. มหาวิทยาลัยรังสิต, 2521.

ภาษาอังกฤษ

Arcus, A.L., COMSOAL: A Computer Method of Sequencing Operations for Assembly Lines. International Journal of Production Research. 4,4, 1966.

Bramlette, M.F., Initialization, mutation and selection methods in genetic algorithms for function optimization Proc. third Int. Conf. On Genetic Algorithms. George Mason University, 1989: pp.100-107.

Boh, D.B., A neural network approach for line balancing problem: a case study. AIT Thesis No.ISE-96-5, 1996.

Chu, P.C., Beasley J.E., A genetic algorithm for the generalized assignment problem, Computer Operation Research. Vol.24, No.1, 1997: pp.17-23.

Chutima, P., and Yiangkamolsing, C.,Genetic algorithms with improvement heuristic for plant layout problems, Research and Development Journal of the Engineering Institute of Thailand. Vol.10, No.2, 1999: pp.62-72.

Coello, C.A. and Christiansen A.D., MOSES: A multiobjective Optimization Tool for Engineering Design. Tulane University, LA, USA. (n.d.).

Falkanauer, E. and Delchambre, A., A genetic algorithms for bin packing and line balancing, Proceedings of the 1992 IEEE International Conference on Robotics an Automation, 1992.

Fonseca, C.M. and Flemming P.J., Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. http://www.shef.ac.uk/uni/projects/qaipp/mogas.html. (n.d.).

Gen, M., and Chen, R., Genetic algorithm and engineering design. New York: John Wiley & Sons, Inc., 1997.

Goldberg, D.E., Genetic Algorithms in Search,Optimization, & Machine Learning. MA: Addison-Wesley, 1989.

Gosh S., Gagnon R., A comprehensive literature review and analysis of the design, balancing and scheduling of assembly system, Int. J. Prod. Res. Vol 27, No.4, 1989: pp. 637-670.

Hoffmann, T.R., Assembly line balancing: a set of chalenging problems, Int. J. Prod. Res. Vol.28, NO.10, 1990: pp.1807-1815.

Holland, J.H., Adaptation in Natural and Artificial System. University of Michican Press, Ann Arbor, 1975.

Horn, J., Nafpliotis,N., Goldberg, D.E., A niched pareto genetic algorithm for multiobjective optimization, Proc. 1st IEEE Conf. Evolutionary Computation. IEEE, 1994: pp.82-87.

Kim,Y.K., Kim, Y.J., Kim, Y., Genetic algorithm for assembly line balancing with various objectives, Computers and Industrial Engineering. Vol.30, No.3, 1996: pp.397-409.

Kubota, N. et.al, Virus-Evolutionary Genetic Algorithm for a Self-Organizing Manufacturing System. Computer Industrial Engineer. Vol.30, No.4, 1996: pp. 1015-1026.

Leu,Y.Y., Matheson L.A. and Rees, L.P., Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria, Decision Sciences. Vol.25, 1991: pp.581-606.

Mapfaira, H. and Byrne, M., A genetic algorithms approach for assembly line balancing, 15th International Conference on Production Research. 1999.

Michalewicz, Z., Genetic Algorithms+Data Structures=Evolution Programs. 2nd edition, Springer-Verlag, NY, USA, 1992.

Montgomery, D.C., Design and Analysis of Experiments. New York, John Wiley & Sons, 1997.

Murata,T., Ishibuchi,H. and Tanaka, H., Multi-objective genetic algorithms and its applications to flowshop scheduling, Computers and Industrial Engineering. Vol.30, No.4, 1996: pp.957-968

Rachamadugu, R.V., Talbot, F.B., Assemble line balancing with dual criteria, Recent Development in Production Research. Elsevier Science Publisher B.V. (n.d.).

Rattanawilaiwan, T., Line balancing in small diesel engine assembly plant with consideration on the local content government regulations. AIT Thesis , No. IE-82-6, 1982.

Rubinovitz, J. and Levitin, G., Genetic Algorithms for Assembly Line Balancing. International Journal of Abstract in Operation Research, 1996.

Schaffer, J.D., and Eshelman, L.J. On Crossover as an Evolutionarily Viable Strategy. Proc. third Int. Conf. On Genetic Algorithms. George Mason University. (n.d.).

Starkweather, T., Mcdaniel, S., Mathias, K., Whitley, D., A comparison of genetic Sequencing Operators, Colorado State University, Fort Collins, CO. (n.d.).

Suresh, G., Vinod, V.V., Sahu, S., A genetic algorithm for assembly line balancing, Production Planning & Control. Vol.7, No.1, 1996: pp.38-46.

Talbot, F.B. and Patterson, J.H., A Comparative Evaluation of Heuristic Line Balancing Techniques. Management Sciences. 32,4, 1984: pp.430-454.

The MathWorks Inc., Getting started with MATLAB, MATLAB:The language of technical Computing. 1998.

Winston, W.L., Operation Research: Applications and Algorithms. Duxbury, Boston, 1987.

Yogathasan, M., Application of neural network in assembly line balancing. AIT Thesis No.ISE-96-4, 1996.

ภาคผนวก

ภาคผนวก ก

ปัญหา NP-hard

ปัญหา NP-hard คือปัญหาที่ใช้เวลาในการหาคำตอบยาวนานและเวลาในการหาคำตอบจะเพิ่มมากขึ้นเป็นแบบเอ็กซ์โปเนนเชียลเมื่อขนาดของปัญหาเพิ่มขึ้น ซึ่งไม่เหมาะกับการหาคำตอบด้วยวิธีการแบบตรงไปตรงมาในทางปฏิบัติ และโดยทั่วไปแล้วจะใช้ฮิวริสติกในการแก้ปัญหาประเภทนี้เพื่อให้ได้คำตอบที่ดีถึงแม้ว่าจะไม่ใช่คำตอบที่ดีที่สุดก็ตาม

ลักษณะของปัญหาแบบ NP-hard จะอยู่ในรูปของ $f(v)$ (Time Complexity Function) ซึ่งเป็นฟังก์ชั่นที่ใช้แสดงถึงเวลาสูงสุดของปัญหาที่มีขนาด v ตัวอย่างของเวลาในการคำนวณแสดงได้ดังตารางที่ ก.1 เช่น เวลาที่ใช้ในการคำนวณของรูปแบบปัญหาที่มีฟังก์ชั่น $f(v)=v$ โดยกำหนดให้ v ขนาดเท่ากับ 10 และกำหนดให้เวลาที่ใช้ในการคำนวณในแต่ละขั้นตอนเท่ากับ 1 ไมโครวินาทีดังนั้นเวลาทั้งหมดที่ใช้ในการคำนวณทั้งหมดเท่ากับ 10 ไมโครวินาที (1x10) แต่ถ้าปัญหามีขนาดใหญ่ขึ้น เวลาที่ใช้ก็จะเพิ่มมากขึ้นเป็นแบบเส้นตรง แต่ถ้าปัญหาที่มีค่าของ $f(v)$ เป็น 2^v 3^v และ $v!$ เวลาที่ใช้จะเป็นแบบเอ็กซ์โปเนนเชียล

ตารางที่ ก.1 เวลาในการคำนวณที่อยู่ในรูป Time Complexity Function โดยมีสมมติฐานว่าการคำนวณในแต่ละครั้งใช้เวลา 1 ไมโครวินาที

Time Complexity Function	v					
$f(v)$	10	20	30	40	50	60
v	0.00001 sec	0.00002 sec	0.00003 sec	0.00004 sec	0.00005 sec	0.00006 sec
v^2	0.001 sec	0.0004 sec	0.0009 sec	0.0016 sec	0.0025 sec	0.0036 sec
v^5	0.1 sec	3.2 sec	24.3 sec	1.7 min	5.2 min	13 min
v^{10}	2.7 hr	118.5 days	18.7 yrs	3.3 centuries	30.9 centuries	192 centuries
2^v	0.001 sec	1.0 sec	17.9 min	12.7 days	35.7 yrs	366 centuries
3^v	0.59 sec	58 min	6.5 yrs	3855 centuries	$2*10^8$ centuries	$1.3*10^{13}$ centuries
$v!$	3.6 sec	770 centuries	$8.4*10^{16}$ yrs	$2.5*10^{32}$ centuries	$9.6*10^{48}$ centuries	$2.6*19^{66}$ centuries

สมมติให้มีเครื่องคอมพิวเตอร์ที่มีความเร็วสูงกว่าเครื่องคอมพิวเตอร์จากตัวอย่างที่ผ่านมา 1,000 เท่า ถ้าปัญหาไม่มีความซับซ้อนมากนักและให้ระยะเวลาในการคำนวณเท่ากับเครื่องคอมพิวเตอร์จากปัญหาที่ผ่านมา ถ้าปัญหาที่มีฟังก์ชัน v ก็สามารถทำให้เวลาในการคำนวณเร็วขึ้น 1,000 เท่า แต่ถ้าปัญหามีความซับซ้อนมากคอมพิวเตอร์ที่มีความเร็วสูงก็สามารถช่วยในการคำนวณได้เร็วขึ้นในระดับหนึ่ง เช่นปัญหาที่มีฟังก์ชั่นเป็น $v!$ เครื่องคอมพิวเตอร์ที่มีการคำนวณเร็วกว่า 1,000 เท่า ช่วยให้การคำนวณได้เร็วขึ้นเล็กน้อย ดังตัวอย่างในตารางที่ ก.2

ตารางที่ ก.2 ขนาดของปัญหาในการคำนวณของคอมพิวเตอร์ที่มีความเร็วสูงกว่า 1000 เท่า

Time Complexity Function	ขนาดของปัญหาที่ถูกแก้		
	คอมพิวเตอร์ธรรมดา	คอมพิวเตอร์ที่มีความเร็วสูงกว่า 1000 เท่า	
n	N_1	$1000N_1$	
n^2	N_2	$31.62N_2$	
n^5	N_3	$3.98N_3$	
n^{10}	N_4	$1.99N_4$	
2^n	N_5	N_5+10	
3^n	N_6	N_6+6	
$n!$	N_7	N_7+3	$N_7 \leq 10$
		N_7+2	$10 < N_7 \leq 30$
		N_7+1	$30 < N_7 \leq 1000$

ปัญหา NP-hard เป็นปัญหาที่ใช้ระยะเวลาในการหาคำตอบยาวนาน ดังนั้นการหาคำตอบด้วยวิธีการแบบตรงไปตรงมาจึงเป็นไปได้ลำบาก และถึงแม้จะมีเครื่องคอมพิวเตอร์ที่มีความเร็วสูงมาช่วยในการคำนวณก็สามารถช่วยได้ในระดับหนึ่ง วิธีการหาคำตอบของปัญหารูปแบบนี้ได้แก่การใช้ฮิวริสติก หรือ อัลกอลิทึมต่างๆมาช่วยใช้ในการหาคำตอบ

ภาคผนวก ข
รายละเอียดของปัญหาตัวอย่างที่ใช้

1. ปัญหาตัวอย่างขนาด 11 งาน

ตารางที่ ก.1 รายละเอียดของปัญหาตัวอย่างขนาด 11 งาน

ขั้นงานที่	เวลาทำงาน (นาที)	ขั้นงานก่อนหน้า
1	6	-
2	2	1
3	2	2
4	6	3
5	5	4
6	5	1
7	7	1
8	1	1
9	3	6, 7, 8
10	5	9
11	4	11
รวม	46	

รอบเวลาการผลิตสูงสุดเท่ากับ 10 นาที

2. ปัญหาตัวอย่างขนาด 31 งาน

ตารางที่ ก.2 รายละเอียดของปัญหาตัวอย่างขนาด 31 งาน

ขั้นงานที่	เวลาทำงาน (นาที)	ขั้นงานก่อนหน้า
1	10	-
2	60	-
3	18	-
4	5	-
5	35	1, 2, 3, 4
6	10	5

ตารางที่ ก.2 รายละเอียดของปัญหาตัวอย่างขนาด 31 งาน (ต่อ)

ขั้นงานที่	เวลาทำงาน (วินาที)	ขั้นงานก่อนหน้า
7	7	6
8	20	7
9	55	7
10	20	8, 9, 12
11	15	10
12	5	7
13	20	5
14	5	5
15	4	5
16	6	5
17	45	5
18	45	17
19	25	5
20	15	5
21	15	5
22	30	11, 13, 14, 15, 16, 18, 19, 20, 21
23	10	22
24	40	23
25	15	24
26	10	24
27	25	24
28	70	25, 27
29	70	28
30	70	29
31	30	26, 30
รวม	810	

รอบเวลาการผลิตสูงสุดเท่ากับ 100 วินาที

3. ปัญหาตัวอย่างขนาด 39 งาน

ตารางที่ ก.3 รายละเอียดของปัญหาตัวอย่างขนาด 39 งาน

ขั้นงานที่	เวลาทำงาน (นาที)	ขั้นงานก่อนหน้า
1	18	-
2	60	-
3	16	-
4	30	-
5	18	-
6	20	5
7	20	1, 2, 3
8	18	4
9	28	6
10	25	8
11	17	4, 6, 7
12	8	6
13	20	10
14	25	11
15	8	11
16	12	13
17	6	14
18	45	12
19	30	1, 16
20	8	14
21	8	17
22	8	19
23	4	22
24	28	11, 21
25	22	18
26	20	23
27	20	5, 21
28	24	24, 25
29	8	26
30	30	27
31	10	28, 30
32	18	29
33	45	29
34	20	31

ตารางที่ ก.3 รายละเอียดของปัญหาตัวอย่างขนาด 39 งาน (ต่อ)

ขั้นงานที่	เวลาทำงาน (นาที)	ขั้นงานก่อนหน้า
35	20	32
36	24	33
37	8	36
38	22	37
39	20	34, 36, 38
รวม	801	

รอบเวลาการผลิตสูงสุด 134 วินาที

4. ปัญหาตัวอย่างขนาด 54 งาน

ตารางที่ ก.4 รายละเอียดของปัญหาตัวอย่างขนาด 54 งาน

ขั้นงานที่	เวลาทำงาน (วินาที)	ขั้นงานก่อนหน้า
1	20	-
2	20	3
3	25	1
4	30	3
5	40	2, 4
6	15	5
7	8	6
8	40	6
9	10	6
10	3	9
11	20	6
12	8	6
13	50	8
14	35	6
15	40	31
16	10	9
17	20	6
18	7	31
19	7	6
20	7	6
21	15	11

ตารางที่ ก.4 รายละเอียดของปัญหาตัวอย่างขนาด 54 งาน (ต่อ)

ขั้นงานที่	เวลาทำงาน (นาที)	ขั้นงานก่อนหน้า
22	20	9
23	40	12, 14,17, 31, 38
24	10	43
25	15	22, 23
26	30	6
27	25	15, 32
28	7	27
29	7	27
30	7	27
31	25	13
32	15	25
33	25	6
34	15	7
35	15	34
36	20	6
37	80	21, 26, 32, 33, 36, 42
38	20	10, 16
39	7	35
40	7	35
41	7	27
42	7	18, 19, 20, 28, 29, 30, 39, 40, 41
43	30	37
44	10	24
45	10	48
46	10	48
47	15	48
48	10	44
48	5	48
50	5	49
51	20	45, 46, 47, 50
52	10	51
53	30	52
54	50	53
รวม	1039	

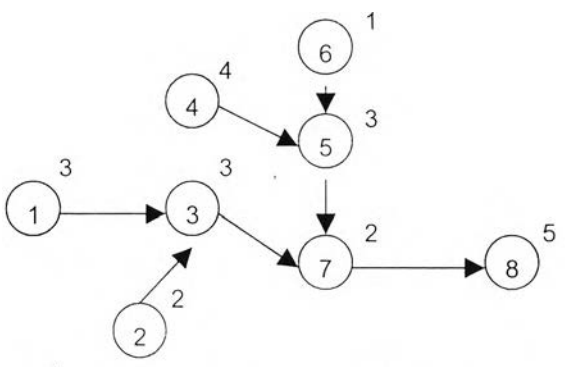
รอบเวลาการผลิตสูงสุด 100 วินาที

ภาคผนวก ค-1
การทดสอบความถูกต้องของโปรแกรมและตัวอย่างการแก้ปัญหาการจัดสมดุลของสายงานการประกอบโดยเจเนติกอัลกอริทึม

เนื่องจาก วิธีการของเจนเนติกอัลกอริทึมที่ได้เสนอไป จะถูกนำไปเขียนเป็นโปรแกรมคอมพิวเตอร์โดยใช้โปรแกรมประยุกต์ MATLAB 5.3.0 ดังนั้นจึงควรมีการทดสอบความถูกต้องของโปรแกรมที่สร้างขึ้นว่าให้ผลที่ถูกต้องตามวิธีการที่เสนอไว้หรือไม่โดยการนำเอาปัญหาที่สมมติขึ้นมาทดลองรันโปรแกรมเพื่อหาคำตอบ การทดสอบจะแบ่งออกเป็น 3 ส่วนคือ

1) การทดสอบความถูกต้องของโปรแกรมในการดำเนินงานแต่ละขั้นตอน การทดลองในส่วนนี้ จะทำโดยการรันโปรแกรมทีละขั้นตอนตามวิธีการของ GAs ผลที่ได้จากแต่ละขั้นตอนจะต้องถูกพิจารณาและมีการทดลองคำนวณหาค่าด้วยมือเพื่อตรวจสอบความเป็นไปได้และความถูกต้องของผลลัพธ์ก่อนที่จะเข้าสู่กระบวนการถัดไป
2) การทดสอบความถูกต้องของโปรแกรมในการลู่เข้าหาคำตอบที่ดีที่สุด ทำโดยการรันโปรแกรมจนกว่าจะครบจำนวนเจนเนอเรชั่นสูงสุด แล้วนำค่าที่ดีที่สุดที่ได้ในทุกเจนเนอเรชั่นมาเขียนกราฟเพื่อดูแนวโน้มของคำตอบ มีการลู่เข้าหาค่าที่ดีที่สุดค่าหนึ่งตามหลักการของ GAs หรือไม่
3) การทดสอบความถูกต้องของโปรแกรมในการหาคำตอบที่เป็นไปได้ที่ดีที่สุด ทำโดยการนำเอาคำตอบที่ดีที่สุดที่ได้ภายหลังจากที่การรันโปรแกรมเสร็จสิ้นลง มาตรวจสอบดูว่าขัดกับข้อจำกัดต่างๆ หรือไม่

ปัญหาสมมติที่ใช้ในการทดสอบมีขนาด 8 งาน มีรายละเอียดดังรูปที่ ค1.1 โดยจะใช้จำนวนประชากรเบื้องต้น 6 ตัว ใช้วิธีการครอสโอเวอร์แบบ PMX ร่วมกับการซ่อมแซมคำตอบ มีค่า Pc=0.7 และ Pm=0.2 จำนวนสถานีสูงสุดที่ยอมรับได้



รูปที่ ค1.1 ไดอะแกรมสายงานการประกอบขนาด 8 งาน

1. การทดสอบความถูกต้องของโปรแกรมในการดำเนินงานแต่ละขั้นตอน

1) การสร้างสตริงคำตอบเบื้องต้น

จากข้อมูลที่ได้ สามารถสร้าง Precedence Matrix ได้ดังรูปที่ ค1.1 และสามารถการสร้างสตริงคำตอบตัวแรกทำได้ดังตารางที่ ค1.1

ก่อน \ หลัง	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	1	0	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	0	0	0	0	0	1	0
6	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0

รูปที่ ค1.1 *เมตริกซ์แสดงความสัมพันธ์ของขั้นงานของปัญหาตัวอย่างขนาด 8 งาน*

ตารางที่ ค1.1 การสร้างสตริงคำตอบเบื้องต้น

ตำแหน่งที่	งานที่ไม่มีงานก่อนหน้า	งานที่สุ่มเลือก	ตำแหน่งที่	งานที่ไม่มีงานก่อนหน้า	งานที่สุ่มเลือก
1	{1 2 4 6}	**2**	5	{6}	**6**
2	{1 4 6}	**1**	6	{5}	**5**
3	{3 4 6}	**3**	7	{7}	**7**
4	{4 6}	**4**	8	{8}	**8**

จะได้สตริงคำตอบตัวแรก คือ [2 1 3 4 6 5 7 8] จากนั้นทำการสร้างสตริงคำตอบตัวอื่นๆจนครบ 6 ตัว ได้ผลดังนี้

$str1$ = [2 1 3 4 6 5 7 8]

$str2$ = [4 1 6 2 3 5 7 8]

$str3$ = [2 4 1 3 6 5 7 8]

$str4$ = [6 1 2 4 5 3 7 8]

$str5$ = [4 6 2 1 3 5 7 8]

$str6$ = [2 1 3 6 4 5 7 8]

จากการตรวจสอบสตริงคำตอบที่สร้างขึ้นทั้ง 6 ตัวกับ Precedence Matrix พบว่าสตริงทั้ง 6 ตัวเป็นสตริงคำตอบที่เป็นไปได้ ซึ่งหมายความว่าสตริงทุกตัวมีการจัดลำดับขั้นงานที่ไม่ขัดกับหลักความสัมพันธ์ก่อนหลังของงาน และไม่มีขั้นงานซ้ำภายในสตริงตัวเดียวกัน นอกจากนี้สตริงคำตอบทั้ง 6 ตัวยังเป็นสตริงที่แตกต่างกัน ไม่มีสตริงที่ซ้ำ ดังนั้นจึงได้ว่าวิธีการสร้างสตริงคำตอบมีความถูกต้อง

2) การถอดรหัสคำตอบ

จากสตริงคำตอบ 6 ตัวนำมาจัดให้กับสถานีงานตามลำดับ โดยกำหนดให้รอบเวลาการผลิตเท่ากับ 6 ได้ผลดังตารางที่ ค1.2 ซึ่งจะเห็นได้ว่าเวลาทำงานของทุกสถานีทำงานไม่เกิน 6 นาที (รอบเวลาการผลิต) ไม่มีขั้นงานใดถูกจัดให้กับหลายสถานี และทุกขั้นงานถูกจัดให้กับสถานีทำงานใดสถานีทำงานหนึ่ง จำนวนสถานีที่ได้จากสตริงคำตอบแต่ละตัวก็ไม่เกิน 6 ซึ่งเป็นจำนวนสถานีสูงสุดที่ยอมรับได้

ตารางที่ ค1.2 ผลการถอดรหัสสตริงคำตอบเบื้องต้น

String No.	String Order	Workstation Order	n	Ti	Real ct
1	[2 1 3 4 6 5 7 8]	[1 1 2 3 3 4 4 5]	5	5 3 5 5 5	5
2	[4 1 6 2 3 5 7 8]	[1 2 2 2 3 3 4 5]	5	4 6 6 2 5	6
3	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	4	6 6 6 5	6
4	[6 1 2 4 5 3 7 8]	[1 1 1 2 3 3 4 5]	5	6 4 6 2 5	6
5	[4 6 2 1 3 5 7 8]	[1 1 2 2 3 3 4 5]	5	5 5 6 2 5	6
6	[1 4 6 2 3 5 7 8]	[1 2 2 3 3 4 4 5]	5	3 5 5 5 5	5

3) การประเมินค่า

ในขั้นตอนนี้จะทำการหาค่า Workload Variance ค่าประสิทธิภาพสายการผลิต ค่าเวลาว่างงานรวม และค่า Fitness ของสตริงคำตอบแต่ละตัว ผลที่ได้จากการคำนวณด้วยโปรแกรม และผลที่ได้จากการคำนวณด้วยมือเพื่อเปรียบเทียบความถูกต้องได้ผลดังตารางที่ ค1.3

จากตารางที่ ค1.3 จะเห็นได้ว่าค่าที่ได้จากโปรแกรม กับค่าที่ได้จากการคำนวณเท่ากัน แสดงให้เห็นถึงความถูกต้องในการคำนวณของโปรแกรม

ตารางที่ ค1.3 ผลการประเมินค่าสตริงคำตอบแต่ละตัว

String	wv		Total Idle Time (min.)		Line Eff. (%)		Fitness	
No.	program	manual	program	manual	program	manual	program	manual
1	0.6400	0.6400	2	2	92.00	92.00	7.1475	7.1475
2	2.2400	2.2400	7	7	76.67	76.67	5.5475	5.5475
3	0.1875	0.1875	1	1	95.83	95.83	7.6000	7.6000
4	2.2400	2.2400	7	7	76.67	76.67	5.5475	5.5475
5	1.8400	1.8400	7	7	76.67	76.67	5.9475	5.9475
6	0.6400	0.6400	2	2	92.00	92.00	7.1475	7.1475

4) การเก็บค่าที่ดีที่สุดเบื้องต้น

จากสตริงคำตอบทั้ง 6 ตัว โปรแกรมได้ทำการเก็บสตริงตัวที่ 3 ไว้เป็นสตริงคำตอบตัวที่ดีที่สุดที่จะต้องเก็บไว้ ซึ่งมีรายละเอียดของสตริงดังนี้

สตริงคำตอบ = [2 4 1 3 6 5 7 8]
สถานีงาน = [1 1 2 2 3 3 3 4]
ค่า Fitness = 7.6000
wv = 0.1875
จำนวนสถานีงาน = 4
รอบเวลาการผลิตจริง = 6
เวลาว่างงานรวม =1
ประสิทธิภาพสายการผลิต= 95.83 %

จากการพิจารณาค่าที่ได้จากการประเมินในตารางที่ ค1.3 ก็จะเห็นได้ว่าสตริงตัวที่ 3 มีค่า Workload Variance ที่ต่ำกว่าสตริงตัวอื่น จึงเป็นคำตอบที่ดีที่สุดที่ต้องเก็บไว้จริง

5) การคัดเลือกสตริงคำตอบ

ในขั้นตอนนี้โปรแกรมจะนำค่า Fitness ที่ได้มาสร้างวงล้อรูเล็ต ได้ดังตารางที่ ค1.4

ตารางที่ ค1.4 การสร้างวงล้อรูเล็ต

String No.	Fitness	*pi*	*qi*
1	7.1475	0.1836	0.1836
2	5.5475	0.1425	0.3260
3	7.6000	0.1952	0.5212
4	5.5475	0.1425	0.6637
5	5.9475	0.1527	0.8164
6	7.1475	0.1836	1.0000
รวม	38.938	1.0000	

จากนั้นจะสุ่มเลือกสตริงคำตอบ 2 ตัวจากวงล้อรูเล็ตแล้วนำมาเปรียบเทียบกัน เพื่อคัดเลือกสตริงคำตอบที่มีค่า Fitness มากกว่าเข้าสู่ Mating Pool ผลการคัดเลือกได้สตริงทั้ง 6 ตัว คือ สตริงหมายเลข 2 3 3 1 6 3 ดังตารางที่ ค1.5 ซึ่งจะกลายเป็นสตริงหมายเลข 1-6 ตามลำดับต่อไป

ตารางที่ ค1.5 ผลการสุ่มเลือกสตริงคำตอบโดยวิธี Tournament Selection

No.	String 1				String 2				Selected String No.
	r_1	*qi* > r1	String No.	Fitness	r_2	*qi* > r2	String No.	Fitness	
1	0.6020	0.6637	4	5.5475	0.2536	0.3260	2	5.5475	2
2	0.5134	0.5212	**3**	**7.6**	0.7327	0.8164	5	5.9475	3
3	0.4222	0.5212	**3**	**7.6**	0.9614	1.0000	6	7.1475	3
4	0.0721	0.1836	**1**	**7.1475**	0.5534	0.6637	4	5.5475	1
5	0.2920	0.3260	2	5.5475	0.8580	1.0000	**6**	**7.1475**	6
6	0.3358	0.5212	**3**	**7.6**	0.6802	0.8164	5	5.9475	3

การทดสอบความถูกต้องในการคัดเลือกสตริงคำตอบทำได้โดยพิจารณาตรวจสอบค่า Fitness ของสตริง 2 ตัวที่สุ่มเลือกขึ้นมาว่าตัวใดมีค่ามากกว่า จากตารางที่ ค1.5 จะเห็นได้ว่าสตริงคู่แรกที่สุ่มมา คือสตริงหมายเลข 4 และ 2 ซึ่งมีค่า Fitness เท่ากัน ดังนั้นจะเลือกตัวใดก็ได้ สำหรับคู่อื่นๆ สตริงที่มีค่า Fitness มากกว่าคือสตริงหมายเลข 3 3 1 6 และ 3 ตามลำดับ ตรงกับที่ได้จากโปรแกรม

6) การครอสโอเวอร์

ในขั้นตอนนี้ โปรแกรมจะทำการสุ่มเลือกสตริงคำตอบมาทำการครอสโอเวอร์ โดยพิจารณาจากสตริงคำตอบที่มีค่าสุ่ม r น้อยกว่าค่า Pc ผลการสุ่มค่าและคัดเลือกแสดงได้ดังตารางที่ ค1.6

ตารางที่ ค1.6 *การเลือกสตริงคำตอบเพื่อนำไปครอสโอเวอร์*

String No.	String Sequence	*ri*	*ri < 0.7 (Pc)*
1	[4 1 6 2 3 5 7 8]	0.9501	
2	[2 4 1 3 6 5 7 8]	0.2311	●
3	[2 4 1 3 6 5 7 8]	0.6068	●
4	[2 1 3 4 6 5 7 8]	0.4860	●
5	[2 1 3 6 4 5 7 8]	0.8913	
6	[2 4 1 3 6 5 7 8]	0.7621	

หมายเหตุ ● หมายถึงสตริงตัวนั้นถูกเลือกไปทำการครอสโอเวอร์

เนื่องจากสตริงที่ถูกเลือกไปครอสโอเวอร์มีเพียง 3 ตัวคือสตริงหมายเลข 2 3 4 ซึ่งไม่สามารถจับคู่ได้จึงต้องทำการลดหรือเพิ่มสตริงคำตอบโดยการสุ่มเลข 0 หรือ 1 ซึ่งโปรแกรมสุ่มได้เลข 1 นั่นหมายความว่าต้องเพิ่มสตริงคำตอบเข้าไปอีก 1 ตัวโดยเลือกจากสตริงคำตอบที่เหลือ ซึ่งในที่นี้เลือกได้สตริงคำตอบตัวที่ 1 ก็จะได้ว่า สตริงคำตอบที่จะนำไป ครอสโอเวอร์คือ 1 2 3 4 ซึ่งสามารถจับคู่ได้เป็น 1-2 และ 3-4 จากนั้นเมื่อให้โปรแกรมทำการครอสโอเวอร์ จะได้ตำแหน่งสุ่มสำหรับสตริงคู่แรกเป็น 4-6 และตำแหน่งสุ่มของสตริงคู่ที่สองเป็น 1-2 ได้สตริงรุ่นลูกของสตริงพ่อแม่คู่แรกเป็น

o_1 = [4 1 2 3 6 5 7 8]

o_2 = [6 4 1 2 3 5 7 8]

และสตริงรุ่นลูกของสตริงพ่อแม่คู่ที่สองเป็น

o_1 = [2 1 4 3 6 5 7 8]

o_2 = [2 4 3 1 6 5 7 8]

จากการพิจารณาผลที่ได้เพื่อทดสอบความถูกต้องของโปรแกรม พบว่า การครอสโอเวอร์ด้วย Pc=0.7 ทำให้สามารถคาดเดาได้ว่าน่าจะมีสตริงคำตอบ 70 % หรือ 0.7*6 = 4.2 ≈ 4 ตัว ที่จะถูกครอสโอเวอร์ ซึ่งจำนวนนี้ตรงกับจำนวนสตริงที่โปรแกรมสุ่มมาครอสโอเวอร์ และจากการพิจารณาค่าสุ่มที่ได้ในตารางที่ ค1.6 ก็จะเห็นได้ว่าตัวที่มีค่าสุ่มน้อยกว่า 0.7 คือสตริงหมายเลข 2 3 และ 4 ดังนั้นจึงต้องเพิ่มสตริงอีก 1 ตัวคือสตริงหมายเลข 1

เมื่อทดลองครอสโอเวอร์สตริงคู่แรกโดยวิธีการครอสโอเวอร์แบบ PMX และใช้ตำแหน่งการครอสโอเวอร์ตามที่โปรแกรมใช้ (ตำแหน่งที่ 4 ถึง 6) จะสตริงพ่อแม่และช่วงการครอสโอเวอร์ ดังนี้

p_1 = [4 1 6 |2 3 5| 7 8]

p_2= [2 4 1 |3 6 5| 7 8]

เมื่อทำการสลับสตริงที่อยู่ในตำแหน่งที่ 4 ถึง 6 (หรือที่อยู่ระหว่างเครื่องหมาย | |) จะได้สตริงรุ่นลูกเป็น

o_1 = [4 1 2 3 6 5 7 8]

o_2 = [6 4 1 2 3 5 7 8]

จากนั้น ตรวจสอบสตริงรุ่นลูกที่ได้ว่าขัดกับหลักความสัมพันธ์ก่อนหลังของงานหรือไม่ ซึ่งในที่นี้พบว่าสตริงรุ่นลูกทั้งสองไม่ขัดกับหลักความสัมพันธ์ดังกล่าว จึงไม่ต้องทำการซ่อมแซมคำตอบต้องทำการซ่อมแซมคำตอบ

การครอสโอเวอร์สตริงคู่ที่สองด้วยวิธี PMX โดยสุ่มตำแหน่งการครอสโอเวอร์ได้ที่ตำแหน่ง 1 และ 2 จะสตริงพ่อแม่และช่วงการครอสโอเวอร์ ดังนี้

p_1 = [| 2 4 | 1 3 6 5 7 8]

p_2= [| 2 1 | 3 4 6 5 7 8]

เมื่อทำการสลับสตริงที่อยู่ในตำแหน่งที่ 3 ถึง 5 (หรือที่อยู่ระหว่างเครื่องหมาย | |) จะได้สตริงรุ่นลูกเป็น

o_1 = [2 1 4 3 6 5 7 8]

o_2 = [2 4 3 1 6 5 7 8]

จากนั้น ตรวจสอบสตริงรุ่นลูกที่ได้ว่าขัดกับหลักความสัมพันธ์ก่อนหลังของงานหรือไม่ ซึ่งในที่นี้พบว่าสตริงรุ่นลูกตัวที่ 2 ขัดกับหลักความสัมพันธ์ดังกล่าว จึงต้องทำการซ่อมแซมคำตอบ โดยนำเอา Precedence Matrix ที่ได้ในตอนแรกมาใช้ในการหางานที่ไม่มีงานก่อนหน้า การซ่อมแซมคำตอบแสดงได้ดังตารางที่ ค1.7 ซึ่งจะได้สตริงรุ่นลูกตัวที่ 2 เป็น [2 4 1 3 6 5 7 8]

เมื่อพิจารณาสตริงที่ได้เทียบกับผลที่ได้จากโปรแกรมพบว่าได้ผลเหมือนกัน แสดงว่าโปรแกรมสามารถทำการครอสโอเวอร์และซ่อมแซมสตริงคำตอบได้อย่างถูกต้อง และสตริงที่ได้ทุกตัวเป็นสตริงคำตอบที่เป็นไปได้

ตารางที่ ค1.7 ผลการซ่อมแซมคำตอบคู่ที่สองที่ได้จากการครอสโอเวอร์

Iteration	Available Set	Infeasible Offspring	Repaired String
1	{1 2 4 6}	[2 4 1 3 6 5 7 8]	[2]
2	{1 4 6}	[2 4 1 3 6 5 7 8]	[2 4]
3	{1 6}	[2 4 1 3 6 5 7 8]	[2 4 1]

ตารางที่ ค1.7 *ผลการซ่อมแซมคำตอบคู่ที่สองที่ได้จากการครอสโอเวอร์ (ต่อ)*

Iteration	Available Set	Infeasible Offspring	Repaired String
4	{3 6}	[2 4 1 3 6 5 7 8]	[2 4 1 3]
5	{6}	[2 4 1 3 6 5 7 8]	[2 4 1 3 6]
6	{5}	[2 4 1 3 6 5 7 8]	[2 4 1 3 6 5]
7	{7}	[2 4 1 3 6 5 7 8]	[2 4 1 3 6 5 7]
8	{8}	[2 4 1 3 6 5 7 8]	[2 4 1 3 6 5 7 8]

หลังจากการครอสโอเวอร์ และซ่อมแซมคำตอบจนได้สตริงรุ่นลูกทั้ง 2 คู่แล้ว โปรแกรมจะนำสตริงคำตอบที่ได้ไปรวมกับสตริงตัวที่ไม่ได้ถูกนำไปครอสโอเวอร์ จากนั้นนำสตริงคำตอบทั้งหมดไปถอดรหัส และประเมินค่า ซึ่งในที่นี้จะหาเพียงค่า Workload Variance และค่า Fitness เท่านั้น จากการถอดรหัสและประเมินค่าโดยโปรแกรมและผลที่ได้จากการทดลองทำด้วยมือเพื่อทดสอบความถูกต้องของโปรแกรม จะได้ผลเหมือนกัน ซึ่งผลที่ได้แสดงไว้ในตารางที่ ค1.8

ตารางที่ ค1.8 *ผลการประเมินค่าสตริงคำตอบที่ได้ภายหลังการครอสโอเวอร์*

String No.	String Order	Workstation Order	Ti	wv	Fitness
1	[4 1 2 3 6 5 7 8]	[1 2 2 3 3 4 4 5]	4 5 4 5 5	0.2400	2.6950
2	[6 4 1 2 3 5 7 8]	[1 1 2 2 3 3 4 5]	5 5 6 2 5	1.8400	1.0950
3	[2 1 4 3 6 5 7 8]	[1 1 2 3 3 4 4 5]	5 4 4 5 5	0.2400	2.6950
4	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	6 6 6 5	0.1875	2.7475
5	[2 1 3 6 4 5 7 8]	[1 1 2 2 3 4 4 5]	5 4 4 5 5	0.2400	2.6950
6	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	6 6 6 5	0.1875	2.7475

7) การเก็บค่าที่ดีที่สุดหลังการครอสโอเวอร์

จากการพิจารณาสตริงที่ได้ภายหลังการครอสโอเวอร์ทั้ง 6 ตัวในตารางที่ ค1.8 จะเห็นว่าสตริงตัวที่ 4 และ 6 เป็นตัวที่ดีที่สุด ซึ่งทั้ง 2 ตัวนี้ให้คำตอบเหมือนกัน จึงเลือกตัวใดตัวหนึ่งไปเปรียบเทียบกับ Elite Preserve Solution ผลปรากฏว่าเป็นตัวเดียวกัน จึงไม่ต้องเก็บค่าใหม่

เมื่อพิจารณาผลที่ได้จากโปรแกรมพบว่า ภายหลังขั้นตอนการเก็บค่าที่ดีที่สุดภายหลังการครอสโอเวอร์ โปรแกรมคอมพิวเตอร์ให้ค่าสตริงคำตอบที่ดีที่สุดเป็น

สตริงคำตอบ = [2 4 1 3 6 5 7 8]
สถานีงาน = [1 1 2 2 3 3 3 4]
ค่า Fitness = 7.6000
wv = 0.1875
จำนวนสถานีงาน = 4
รอบเวลาการผลิตจริง = 6
เวลาว่างงานรวม =1
ประสิทธิภาพสายการผลิต= 95.83 %

จะเห็นได้ว่าสตริงคำตอบที่โปรแกรมเลือกเก็บไว้คือสตริงคำตอบตัวเดิม เช่นกัน แสดงว่าขั้นตอนนี้มีความถูกต้อง

8) การมิวเตชั่น

ในขั้นตอนนี้ โปรแกรมจะทำการสุ่มเลือกสตริงคำตอบมาทำการมิวเตชั่นโดยพิจารณาจากสตริงคำตอบที่มีค่าสุ่ม r น้อยกว่าค่า Pm ผลการสุ่มค่าและคัดเลือกแสดงได้ดังตารางที่ ค1.9

ตารางที่ ค1.9 การเลือกสตริงคำตอบไปมิวเตชั่น

String No.	*ri*	*ri < 0.2 (Pm)*
1	0.4103	
2	0.8936	
3	0.3579	
4	0.0529	●
5	0.8132	
6	0.2028	

หมายเหตุ ● หมายถึงสตริงตัวนั้นถูกเลือกไปทำการมิวเตชั่น

จากนั้นเมื่อให้โปรแกรมทำการมิวเตชั่น จะได้ตำแหน่งสุ่มสำหรับมิวเตชั่นเป็น 1 ถึง 2 ซึ่งได้สตริงรุ่นลูกออกมาเป็น

o=[2 4 6 5 1 3 7 8]

เมื่อทำการพิจารณาเพื่อตรวจสอบความถูกต้องของโปรแกรม พบว่าเมื่อกำหนดให้ Pm=0.2 ทำให้สามารถคาดเดาได้ว่าน่าจะมีสตริงคำตอบ 20 % หรือ 0.2*6 =

1.2 ≈ 1 ตัว ที่จะถูกมิวเตชั่น จำนวนสตริงที่คาดหมายว่าจะถูกมิวเตชั่นตรงกับจำนวนสตริงที่โปรแกรมคัดเลือกมา เมื่อพิจารณาค่าสุ่มดังตารางที่ ค1.9 ก็จะพบว่าสตริงตัวที่มีค่าสุ่มน้อยกว่า 0.2 คือสตริงหมายเลข 4 ตรงตามที่โปรแกรมคัดเลือกไว้

จากนั้นทดลองทำการมิวเตชั่นที่ตำแหน่งที่ 1 ถึง 2 เช่นเดียวกับที่โปรแกรมใช้ จะได้ว่าสตริงตัวใหม่จะมีค่าในตำแหน่งที่1-2 เหมือนสตริงตัวเดิม

o = [2 4 x x x x x x]

ในตำแหน่งที่เหลือจะใช้วิธีกำหนดงานเหมือนกับการสร้างประชากรเบื้องต้น โดยในตอนนี้มีงานที่ถูกจัดไปแล้ว 2 งาน ในตำแหน่งที่ 1-2 การจัดงานในตำแหน่งที่เหลือแสดงได้ดังตารางที่ ค1.10

จากการทดลองทำการมิวเตชั่นจะได้สตริงรุ่นลูกเป็น [2 4 6 5 1 3 7 8] ซึ่งตรงกับผลที่ได้จากโปรแกรม แสดงว่าโปรแกรมสามารถทำการมิวเตชั่นได้อย่างถูกต้อง และสตริงที่ได้เป็นสตริงคำตอบที่เป็นไปได้

ตารางที่ ค1.10 *การหาสตริงคำตอบตัวใหม่จากการมิวเตชั่น*

Iteration	Available Set	Random Selected String	String Sequence
1	-	-	-
2	-	-	[2 4]
3	{1 6}	6	[2 4 6]
4	{1 5}	5	[2 4 6 5]
5	{1}	1	[2 4 6 5 1]
6	{3}	3	[2 4 6 5 1 3]
7	{7}	7	[2 4 6 5 1 3 7]
8	{8}	8	[2 4 6 5 1 3 7 8]

หลังจากการมิวเตชั่น โปรแกรมจะนำสตริงตัวใหม่ที่ไปแทนที่สตริงตัวเดิม ในตำแหน่งสตริงที่ 4

9) การเก็บค่าที่ดีที่สุดหลังการมิวเตชั่น

ภายหลังการมิวเตชั่นจะได้สตริงคำตอบชุดใหม่ซึ่งจะถูกใช้เป็นสตริงพ่อแม่ในเจนเนอเรชั่นต่อไป แต่ก่อนที่จะเข้าสู่เจนเนอเรชั่นต่อไป สตริงคำตอบทุกตัวจะต้องถูกถอดรหัสและประเมินค่า Workload Variance และค่า Fitness เสียก่อน จากการถอดรหัสและประเมินค่าโดยโปรแกรมและผลที่ได้จากการทดลองทำด้วยมือเพื่อทดสอบความถูกต้องของโปรแกรม จะได้ผลเหมือนกัน ซึ่งผลที่ได้แสดงไว้ในตารางที่ ค1.11

ตารางที่ ค1.11 *การประเมินค่าสตริงคำตอบภายหลังการมิวเตชั่น*

String No.	String Order	Workstation Order	Ti	wv	Fitness
1	[4 1 2 3 6 5 7 8]	[1 2 2 3 3 4 4 5]	4 5 4 5 5	0.2400	4.7475
2	[6 4 1 2 3 5 7 8]	[1 1 2 2 3 3 4 5]	5 5 6 2 5	1.8400	3.1475
3	[2 1 4 3 6 5 7 8]	[1 1 2 3 3 4 4 5]	5 4 4 5 5	0.2400	4.7475
4	[2 4 6 5 1 3 7 8]	[1 1 2 2 3 3 4 5]	6 4 6 2 5	2.2400	2.7475
5	[2 1 3 6 4 5 7 8]	[1 1 2 2 3 4 4 5]	5 4 4 5 5	0.2400	4.7475
6	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	6 6 6 5	0.1875	4.8000

จากนั้นสตริงคำตอบที่ดีที่สุดจากการมิวเตชั่นซึ่งในที่นี้ คือสตริงตัวที่ 6 จะต้องถูกนำมาเปรียบเทียบกับ Elite Preserve Solution ซึ่งจากการพิจารณาตารางที่ ค1.11 จะเห็นว่า สตริงที่ดีที่สุดภายหลังการมิวเตชั่นคือสตริงหมายเลข 6 เมื่อนำไปเปรียบเทียบกับสตริงที่ดีที่สุดที่เก็บไว้พบว่าเท่ากัน จึงไม่ต้องทำการเก็บค่าหรือแทนที่สตริงคำตอบ เมื่อพิจารณาผลที่ได้จากโปรแกรม จะได้ว่าภายหลังการพิจารณาเปรียบเทียบในขั้นตอนนี้ สตริงคำตอบที่ดีที่สุดที่เก็บไว้ คือ

สตริงคำตอบ = [2 4 1 3 6 5 7 8]
สถานีงาน = [1 1 2 2 3 3 3 4]
ค่า Fitness = 7.6000
wv = 0.1875
จำนวนสถานีงาน = 4
รอบเวลาการผลิตจริง = 6
เวลาว่างงานรวม =1
ประสิทธิภาพสายการผลิต= 95.83 %

สตริงที่โปรแกรมเลือกเก็บไว้เป็นไปตามที่เราได้พิจารณาในข้างต้น แสดงว่าโปรแกรมสามารถเลือกเก็บค่าที่ดีที่สุดภายหลังการมิวเตชั่นได้อย่างถูกต้อง

หมายเหตุ สมมติว่าสตริงคำตอบที่ดีที่สุดไม่ใช่ตัวที 6 แต่เป็นตัวที่ 5 เมื่อนำไปเปรียบเทียบกับคำตอบที่ดีที่สุดที่เก็บไว้ซึ่งมี wv เท่ากับ 0.1875 จะได้ว่าคำตอบที่เก็บไว้ดีกว่า จึงต้องแทนที่สตริงคำตอบตัวที่แย่ที่สุด ซึ่งคือสตริงคำตอบตัวที่ 4 ด้วยคำตอบที่เก็บไว้ สตริงคำตอบที่ได้ทั้งหมดจะกลายเป็นสตริงคำตอบในเจนเนอเรชั่นถัดไป

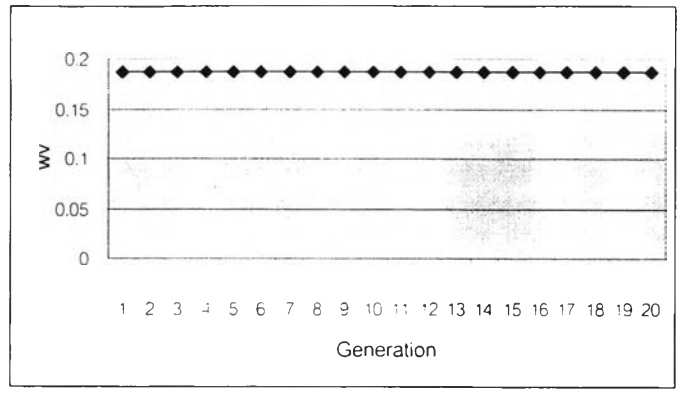
จากการทดสอบที่ผ่านมา สามารถสรุปได้ว่าโปรแกรมสามารถดำเนินงานตามขั้นตอนของ GAs ได้อย่างถูกต้อง

2. การทดสอบความถูกต้องของโปรแกรมในการลู่เข้าหาคำตอบที่ดีที่สุด

เมื่อจบกระบวนการในเจนเนอเรชั่นที่ 1 สตริงคำตอบทั้ง 6 ตัวในตาราง 5.14 จะกลายเป็นสตริงคำตอบเบื้องต้นในเจนเนอเรชั่นต่อไป ซึ่งเมื่อปล่อยให้โปรแกรมหาคำตอบในเจนเนอเรชั่นต่อๆไป จนถึงเจนเนอเรชั่นที่ 20 ซึ่งเป็นเจนเนอเรชั่นสูงสุดที่กำหนดให้ จะได้สตริงคำตอบที่ดีที่สุดในแต่ละเจนเนอเรชั่นดังตารางที่ ค1.12 และกราฟแสดงค่า Workload Variance ซึ่งเป็นค่าวัตถุประสงค์ที่เราพิจารณาแสดงได้ดังรูปที่ ค1.2

ตารางที่ ค1.12 *สตริงคำตอบที่ดีที่สุดของเจนเนอเรชั่นที่ 1 ถึง 20*

Generaion	String Order	Station order	WV
1	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
2	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
3	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
4	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
5	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
6	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
7	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
8	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
9	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
10	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
11	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
12	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
13	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
14	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
15	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
16	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
17	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
18	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
19	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875
20	[2 4 1 3 6 5 7 8]	[1 1 2 2 3 3 3 4]	0.1875



รูปที่ ค1.2 กราฟแสดงค่า Workload Variance ของสตริงที่ดีที่สุดเจนเนอเรชั่นที่ 1 - 20

จากรูปที่ ค1.2 จะเห็นได้ว่าค่า workload variance จะมีแนวโน้มคงที่ แต่ก็ยังคงถือว่าโปรแกรมสามารถให้คำตอบที่ลู่เข้าหาคำตอบที่ดีที่สุดได้ ซึ่งในที่นี้จะลู่เข้าเร็วมาก (ได้คำตอบที่ดีที่สุดตั้งแต่เจนเนอเรชั่นแรก) เนื่องจากเป็นปัญหาที่มีขนาดเล็ก

3. การทดสอบความถูกต้องของโปรแกรมในการหาคำตอบที่เป็นไปได้ที่ดีที่สุด

ภายหลังจากการรันโปรแกรมเสร็จสิ้นลง จะได้คำตอบสำหรับปัญหาการจัดสมดุลของสายงานการประกอบสำหรับปัญหาตัวอย่างนี้ ดังตารางที่ ค1.13 และตารางที่ ค1.14

***ตารางที่ ค1.13** ผลการจัดสมดุลสายงานการประกอบของปัญหา 8 งานโดย GAs*

สถานีที่	ขั้นงาน	เวลาของสถานี
1	2, 4	6
2	1, 3	6
3	6, 5, 7	6
4	8	5
รวม		34

***ตารางที่ ค1.14** ค่า Measure of Performance ต่าง ๆของปัญหา 8 งานที่ได้จากGAs*

จำนวนสถานี (สถานี)	รอบเวลาผลิตจริง (นาที)	Workload Variance	เวลาว่างงานรวม (นาที)	ประสิทธิภาพสายการประกอบ (%)
8	5	0.188	6	85

จากตารางที่ ค1.13 จะเห็นได้ว่าผลที่ได้ไม่ขัดกับหลักความสัมพันธ์ก่อนหลังของงานและขั้นงานทุกงานถูกจัดให้กับสถานีทำงานใดสถานีทำงานหนึ่ง ไม่มีขั้นงานใดที่ถูกจัดซ้ำลงในหลายๆสถานีทำงาน นอกจากนี้เวลาทำงานของแต่ละสถานีก็ไม่เกิน 6 นาทีซึ่งเป็นรอบเวลาการผลิตที่กำหนดไว้ และจำนวนสถานีก็น้อยกว่า 6 ซึ่งเป็นจำนวนสถานีสูงสุดที่ยอมรับได้ ดังนั้นจึงสรุปได้ว่าโปรแกรมสามารถให้คำตอบที่เป็นไปได้

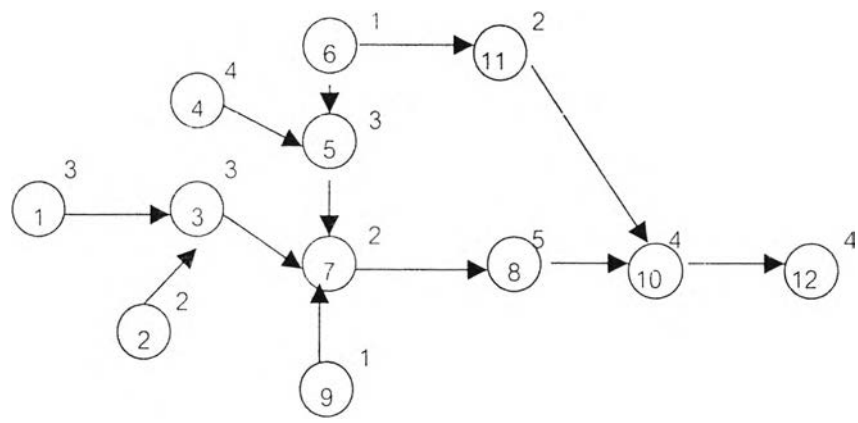
จากการทดสอบทั้ง 3 ส่วนจะเห็นได้ว่าโปรแกรมมีความถูกต้องทั้งในด้านการดำเนินการตามวิธีการของ Gas การลู่เข้าหาคำตอบ และการหาคำตอบที่เป็นไปได้ที่ดีที่สุด

ภาคผนวก ค-2

การทดสอบความถูกต้องของโปรแกรมและตัวอย่างการแก้ปัญหาการจัดสมดุลของสายงานการประกอบแบบหลายวัตถุประสงค์โดยเจเนติกอัลกอริทึม

เนื่องจาก วิธีการของเจนเนติกอัลกอริทึมแบบหลายวัตถุประสงค์ที่ได้เสนอไป จะถูกนำไปเขียนเป็นโปรแกรมคอมพิวเตอร์เช่นเดียวกับวิธีการของเจนเนติกอัลกอริทึมแบบวัตถุประสงค์เดียวในบทที่ 5 ดังนั้นจึงควรมีการทดสอบความถูกต้องของโปรแกรมที่สร้างขึ้นโดยมีการทดสอบ 3 ส่วนเช่นกัน

ปัญหาตัวอย่างที่ใช้ในการทดสอบมีขนาด 12 งาน ซึ่งมีรายละเอียดดังรูปที่ ค2.1 โดยใช้จำนวนประชากรเบื้องต้น 10 ตัว ใช้วิธีการครอสโอเวอร์แบบ OX ร่วมกับการซ่อมแซมคำตอบ มีค่า Pc=0.3 และ Pm=0.2 จำนวนสถานีงานสูงสุดเท่ากับ 10 สถานี



รูปที่ ค2.1 ไดอะแกรมสายงานการประกอบขนาด 12 งาน

1. การทดสอบความถูกต้องของโปรแกรมในการดำเนินงานแต่ละขั้นตอน

1) การสร้างสตริงคำตอบเบื้องต้น

จากข้อมูลที่ได้ สามารถสร้าง Precedence Metrix ได้ดังรูป ค2.2 และการสร้างสตริงคำตอบตัวแรกทำได้ดังตารางที่ ค2.1

ก่อน \ หลัง	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	1	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	1	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	1	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่ ค2.2 เมตริกซ์แสดงความสัมพันธ์ของขั้นงานของปัญหาตัวอย่างขนาด 12 งาน

ตารางที่ ค2.1 ตัวอย่างการสร้างสตริงคำตอบเบื้องต้นของปัญหา 12 งาน

ตำแหน่งที่	งานที่ไม่มีงานก่อนหน้า	งานที่สุ่มเลือก	ตำแหน่งที่	งานที่ไม่มีงานก่อนหน้า	งานที่สุ่มเลือก
1	{1 2 4 6 9}	**9**	7	{3 11}	**3**
2	{1 2 4 6}	**1**	8	{7 11}	**11**
3	{2 4 6}	**4**	9	{7}	**7**
4	{2 6}	**2**	10	{8}	**8**
5	{3 6}	**6**	11	{10}	**10**
6	{3 5 11}	**5**	12	12	**12**

จะได้สตริงคำตอบตัวแรก คือ [9 1 4 2 6 5 3 11 7 8 10 12] จากนั้นทำการสร้างสตริงคำตอบตัวอื่นๆจนครบ 10 ตัว ได้ผลดังนี้

*str*1 = [9 1 4 2 6 5 3 11 7 8 10 12]

*str*2 = [1 4 9 6 2 11 3 5 7 8 10 12]

*str*3 = [4 1 6 9 5 2 11 3 7 8 10 12]

*str*4 = [2 6 9 4 1 11 5 3 7 8 10 12]

*str*5 = [2 6 4 1 9 5 3 7 8 11 10 12]

*str*6 = [6 1 11 2 3 9 4 5 7 8 10 12]

*str*7 = [2 4 1 6 9 5 3 7 11 8 10 12]

$str8$ = [4 2 1 9 6 11 5 3 7 8 10 12]

$str9$ = [1 4 9 2 6 5 3 7 8 11 10 12]

$str10$ = [4 1 6 2 9 5 3 7 8 11 10 12]

จากการตรวจสอบสตริงคำตอบที่สร้างขึ้นทั้ง 10 ตัวกับ Precedence Metrix พบว่าสตริงทั้ง 10 ตัวเป็นสตริงคำตอบที่เป็นไปได้ ซึ่งหมายความว่าสตริงทุกตัวมีการจัดลำดับขั้นงานที่ไม่ขัดกับหลักความสัมพันธ์ก่อนหลังของงาน และไม่มีขั้นงานซ้ำภายในสตริงตัวเดียวกัน นอกจากนี้สตริงคำตอบทั้ง 10 ตัวยังเป็นสตริงที่แตกต่างกัน ไม่มีสตริงที่ซ้ำ ดังนั้นจึงได้ว่าวิธีการสร้างสตริงคำตอบมีความถูกต้อง

2) การถอดรหัสคำตอบ

สตริงคำตอบ 10 ตัวจะถูกนำมาหารอบเวลาการผลิตใหม่ จากนั้นจึงนำงานมาจัดให้กับสถานีงานตามลำดับ ซึ่งได้ผลดังตารางที่ ค.2 จะเห็นได้ว่าเวลาทำงานของทุกสถานีทำงานไม่เกินรอบเวลาการผลิตที่กำหนด ไม่มีขั้นงานใดถูกจัดให้กับหลายสถานี และทุกขั้นงานถูกจัดให้กับสถานีทำงานใดสถานีทำงานหนึ่ง จำนวนสถานีที่ได้จากสตริงคำตอบแต่ละตัวก็ไม่เกิน 9 ซึ่งเป็นจำนวนสถานีสูงสุดที่ยอมรับได้ แสดงว่าโปรแกรมสามารถถอดรหัสสตริงคำตอบได้อย่างถูกต้อง

ตารางที่ ค2.2 *ผลการถอดรหัสสตริงคำตอบเบื้องต้นของปัญหา 12 งาน*

String No.	String Order	Cal. ct	Workstation Order	n	Ti	Real ct
1	[9 1 4 2 6 5 3 11 7 8 10 12]	6.30	[1 1 2 2 3 3 4 4 5 6 7 8]	8	4 6 4 5 2 5 4 4	6
2	[1 4 9 6 2 11 3 5 7 8 10 12]	5.00	[1 2 2 3 3 3 4 5 5 6 7 8]	8	3 5 5 3 5 5 4 4	5
3	[4 1 6 9 5 2 11 3 7 8 10 12]	5.15	[1 2 2 2 3 3 4 4 5 6 7 8]	8	4 5 5 5 2 5 4 4	5
4	[2 6 9 4 1 11 5 3 7 8 10 12]	7.00	[1 1 1 2 3 3 4 4 5 6 7 8]	8	4 4 5 6 2 5 4 4	6
5	[2 6 4 1 9 5 3 7 8 11 10 12]	8.70	[1 1 1 2 2 2 3 3 4 4 5 5]	5	7 7 5 7 8	8
6	[6 1 11 2 3 9 4 5 7 8 10 12]	7.00	[1 1 1 2 2 2 3 3 4 4 5 6]	6	6 6 7 7 4 4	7
7	[2 4 1 6 9 5 3 7 11 8 10 12]	7.53	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6 5 6 4 5 4 4	6
8	[4 2 1 9 6 11 5 3 7 8 10 12]	6.00	[1 2 2 3 3 3 4 5 5 6 7 8]	8	4 5 4 3 5 5 4 4	5
9	[1 4 9 2 6 5 3 7 8 11 10 12]	7.44	[1 1 2 2 2 2 3 3 4 4 5 6]	6	7 7 5 7 4 4	7
10	[4 1 6 2 9 5 3 7 8 11 10 12]	6.00	[1 2 2 2 3 3 4 4 5 6 6 7]	7	4 6 4 5 5 6 4	6

3) การประเมินค่า

ในขั้นตอนนี้จะทำการหาค่า Workload Variance ค่าประสิทธิภาพสายการผลิต ค่าเวลาว่างงานรวม และค่า Fitness ของสตริงคำตอบแต่ละตัว โดยให้น้ำหนักกับวัตถุประสงค์ต่างๆ เป็น 0.6 0.2 และ 0.2 ผลที่ได้จากการคำนวณด้วยโปรแกรม และผลที่ได้จากการคำนวณด้วยมือเพื่อเปรียบเทียบความถูกต้องได้ผลดังตารางที่ ค2.3

ตารางที่ ค2.3 ผลการประเมินค่าสตริงคำตอบแต่ละตัวของปัญหา 12 งาน

String	wv		Total Idle Time (min.)		Line Eff. (%)		Fitness	
No.	Program	Manual	Program	Manual	Program	Manual	Program	Manual
1	1.1875	1.19	14	14	70.83	70.83	9.0812	9.08
2	0.6875	0.69	6	6	85.00	85.00	10.7409	10.74
3	0.9375	0.94	6	6	85.00	85.00	9.9473	9.95
4	1.1875	1.19	14	14	70.33	70.33	9.0812	9.08
5	0.9600	0.96	6	6	85.00	85.00	12.1762	12.18
6	1.5556	1.56	8	8	80.95	80.95	10.1581	10.16
7	0.6939	0.69	8	8	80.95	80.95	11.0676	11.07
8	0.4375	0.44	6	6	85.00	85.00	12.4414	12.44
9	1.8889	1.89	8	8	80.95	80.95	9.9260	9.93
10	0.6939	0.69	8	8	80.95	80.95	11.0676	11.07

จากตารางที่ ค2.3 จะเห็นได้ว่าค่าที่ได้จากโปรแกรม กับค่าที่ได้จากการคำนวณเท่ากัน แสดงให้เห็นถึงความถูกต้องในการคำนวณของโปรแกรม

4) การเก็บค่าที่ดีที่สุดเบื้องต้น

จากสตริงคำตอบทั้ง 10 ตัว โปรแกรมได้ทำการเก็บสตริงคำตอบที่ 8 5 8 และ 8 ไว้เป็นชุดของสตริงคำตอบที่ดีที่สุด มีรายละเอียดของชุดสตริงคำตอบที่ดีที่สุดที่เก็บไว้ดังตารางที่ ค2.4

ตารางที่ ค2.4 ชุดของคำตอบที่ดีที่สุดที่เก็บไว้เบื้องต้น

Elite Preserve Solution	Criteria	Preserved String No.	String Sequence	Fitness	n	Real ct	wv
1st	Max. Fitness	8	[4 2 1 9 6 11 5 3 7 8 10 12]	**12.4414**	8	5	0.4375
2nd	Min. n	5	[2 6 4 1 9 5 3 7 8 11 10 12]	12.1762	**5**	8	0.9600
3rd	Min. real_ct	8	[4 2 1 9 6 11 5 3 7 8 10 12]	12.4414	8	**5**	0.4375
4th	Min wv.	8	[4 2 1 9 6 11 5 3 7 8 10 12]	12.4414	8	5	**0.4375**

เมื่อทำการตรวจสอบความถูกต้องในการเลือกเก็บค่าของโปรแกรม โดยพิจารณาสตริงคำตอบทั้ง 10 ตัวในตารางที่ ค2.3 จะเห็นได้ว่าสตริงตัวที่ 8 มีค่า Fitness ดีที่สุด จึงกำหนดให้เป็นสตริงคำตอบที่เก็บไว้ตัวแรก และสตริงคำตอบตัวที่ 5 8 8 เป็นสตริงตัวที่มีจำนวนสถานี รอบเวลาการผลิตจริง และค่า wv น้อยที่สุดตามลำดับ ถูกควรถูกนำไปเก็บเป็นสตริงคำตอบที่เก็บไว้ตัวที่ 2 3 และ 4 เช่นเดียวกับที่โปรแกรมเลือกเก็บไว้ ดังนั้นจึงได้ว่าโปรแกรมสามารถเลือกเก็บค่าที่ดีที่สุดไว้ได้อย่างถูกต้อง

5) การคัดเลือกสตริงคำตอบ

ในขั้นตอนนี้โปรแกรมนำค่า Fitness ที่ได้มาสร้างวงล้อรูเล็ต ได้ดังตารางที่ ค2.5 จากนั้นโปรแกรมจะสุ่มตัวเลขระหว่าง 0-1 มา 10 ค่า ดังตารางที่ ค2.5 จากนั้นจะคัดเลือกสตริงโดยวิธี Roulette Wheel Selection ซึ่งจะสุ่มค่าขึ้นมาแล้วดูว่าตกอยู่ในช่วงค่า Fitness สะสมของสตริงตัวใดก็เลือกสตริงตัวนั้น ผลที่ได้จากโปรแกรมพบว่าสตริงที่ถูกคัดเลทอกคือสตริงหมายเลข 4 8 3 6 10 5 7 5 9 และ 8 ตามลำดับ

ตารางที่ ค2.5 การสร้างวงล้อรูเล็ตของปัญหา 12 งาน

String No.	Fitness	pi	qi
1	9.0812	0.0859	0.0859
2	10.7409	0.1016	0.1876
3	9.9473	0.0941	0.2817
4	9.0812	0.0859	0.3676
5	12.1762	0.1152	0.4828
6	10.1581	0.0961	0.5789

ตารางที่ ค2.5 *การสร้างวงล้อรูเล็ตของปัญหา 12 งาน (ต่อ)*

String No.	Fitness	pi	qi
7	11.0676	0.1047	0.6836
8	12.4414	0.1177	0.8014
9	9.9260	0.0939	0.8953
10	11.0676	0.1047	1.0000
รวม	105.688	1.0000	

การตรวจสอบความถูกต้องของโปรแกรมในขั้นตอนนี้โดยการพิจารณาว่าค่าสุ่มที่โปรแกรมสุ่มขึ้นมาตกอยู่ในช่วงของสตริงตัวใด (ดูจากตารางที่ ค2.5) ได้ผลดังตารางที่ ค2.6

ตารางที่ ค2.6 *ผลการสุ่มเลือกสตริงคำตอบโดยวิธี Roulette Wheel Selection*

Iteration	r	qi > r	Selected String No.	Iteration	r	qi > r	Selected String No.
1	0.3511	0.3676	4	6	0.4565	0.4828	**5**
2	0.7864	0.8014	**8**	7	0.6068	0.6836	**7**
3	0.2121	0.2817	**3**	8	0.4818	0.4828	**5**
4	0.5455	0.5789	**6**	9	0.8913	0.8953	**9**
5	0.9501	1.0000	**10**	10	0.7621	0.8014	**8**

ตารางที่ ค2.6 แสดงให้เห็นว่าสตริงทั้ง 10 ตัวที่เลือกไว้คือสตริงหมายเลข 4 8 3 6 10 5 7 5 9 และ 8 เช่นเดียวกับผลที่ได้จากโปรแกรม แสดงว่าโปรแกรมสามารถคัดเลือกสตริงคำตอบได้อย่างถูกต้อง

สตริงที่เลือกมาทั้ง 10 ตัวจากตารางที่ ค2.6 จะกลายเป็นสตริงหมายเลข 1-10 ที่พร้อมจะเข้าสู่กระบวนการครอสโอเวอร์ต่อไป

6) การครอสโอเวอร์

ในขั้นตอนนี้ โปรแกรมจะทำการสุ่มเลือกสตริงคำตอบมาทำการครอสโอเวอร์ โดยพิจารณาจากสตริงคำตอบที่มีค่าสุ่ม r น้อยกว่าค่า Pc ผลการสุ่มค่าและคัดเลือกแสดงได้ดังตารางที่ ค2.7

ตารางที่ ค2.7 *การเลือกสตริงคำตอบของปัญหา 12 งานเพื่อนำไปครอสโอเวอร์*

ตารางที่ ค2.7 *การเลือกสตริงคำตอบของปัญหา 12 งานเพื่อนำไปครอสโอเวอร์*

String No.	Original No.	String Sequence	*ri*	*ri < 0.7 (Pc)*
1	4	[2 6 9 4 1 11 5 3 7 8 10 12]	0.8185	
2	8	[4 2 1 9 6 11 5 3 7 8 10 12]	0.3214	●
3	3	[4 1 6 9 5 2 11 3 7 8 10 12]	0.7447	
4	6	[6 1 11 2 3 9 4 5 7 8 10 12]	0.9154	
5	10	[4 1 6 2 9 5 3 7 8 11 10 12]	0.6919	●
6	5	[2 6 4 1 9 5 3 7 8 11 10 12]	0.5218	●
7	7	[2 4 1 6 9 5 3 7 11 8 10 12]	0.2382	●
8	5	[2 6 4 1 9 5 3 7 8 11 10 12]	0.8763	
9	9	[1 4 9 2 6 5 3 7 8 11 10 12]	0.0355	●
10	8	[4 2 1 9 6 11 5 3 7 8 10 12]	0.2169	●

หมายเหตุ ● หมายถึงสตริงตัวนั้นถูกเลือกไปทำการครอสโอเวอร์

สตริงที่ถูกทั้ง 6 ตัวจะถูกนำมาจับคู่ตามลำดับ คือ สตริงตัวที่ 2-5 6-7 และ 9-10 แล้วถูกนำไปครอสโอเวอร์ โดยมีตำแหน่งการครอสโอเวอร์สำหรับสตริงพ่อแม่แต่ละคู่เป็น 4-7 3-6 และ 7-9 ตามลำดับ จากโปรแกรมจะได้สตริงรุ่นลูกจากสตริงพ่อแม่คู่ที่ 1 เป็น

o_1 = [4 6 1 2 9 5 3 11 7 8 10 12]

o_2 = [4 1 2 9 6 3 5 7 8 11 10 12]

สตริงรุ่นลูกจากสตริงพ่อแม่คู่ที่ 2 เป็น

o_1 = [2 4 1 6 9 5 3 7 8 11 10 12]

o_2 = [2 6 4 1 9 5 3 7 11 8 10 12]

และสตริงรุ่นลูกจากสตริงพ่อแม่คู่ที่ 3 เป็น

o_1 = [1 4 9 2 6 8 5 3 7 11 10 12]

o_2 = [4 2 1 9 6 11 3 7 8 5 10 12]

ในการทดสอบความถูกต้องของขั้นตอนนี้ พบว่าถ้ากำหนดให้ Pc=0.7 จำนวนสตริงที่จะถูกครอสโอเวอร์น่าจะมีประมาณ 70% ของสตริงทั้งหมด หรือเท่ากับ 0.7*10 = 7 ตัว ซึ่งจากโปรแกรมจะเห็นได้ว่าจำนวนสตริงที่ถูกสุ่มมาทำการครอสโอเวอร์มี 6 ตัวซึ่งถือได้ว่าใกล้เคียงกัน เมื่อพิจารณาสตริงคำตอบที่มีค่าสุ่ม r น้อยกว่าค่า Pc ก็จะได้ว่าสตริงดังกล่าวได้แก่ สตริงหมายเลข 2 5 6 7 9 และ 10 จากนั้นทดลองนำสตริงคู่แรกไปครอสโอเวอร์ด้วยวิธี OX โดยใช้ตำแหน่งการครอสโอเวอร์เป็น 4 และ 7 เหมือนกับที่โปรแกรมใช้ จะได้สตริงพ่อแม่และช่วงที่จะครอสโอเวอร์ ดังนี้

p_1 = [4 2 1 | 9 6 11 5 | 3 7 8 10 12]

p_2= [4 1 6 | 2 9 5 3 | 7 8 11 10 12]

เมื่อทำการสลับสตริงที่อยู่ในตำแหน่งที่ 4 ถึง 7 (หรือที่อยู่ระหว่างเครื่องหมาย | |) และครอสโอเวอร์ตามวิธี OX จะได้สตริงรุ่นลูกเป็น

o_1 = [4 6 1 2 9 5 3 11 7 8 10 12]

o_2 = [4 1 2 9 6 3 5 7 8 11 10 12]

จากนั้น ตรวจสอบสตริงรุ่นลูกที่ได้ว่าขัดกับหลักความสัมพันธ์ก่อนหลังของงานหรือไม่ ซึ่งในที่นี้พบว่าสตริงรุ่นลูกทั้งสองไม่ขัดกับหลักความสัมพันธ์ดังกล่าว จึงไม่ต้องทำการซ่อมแซมคำตอบต้องทำการซ่อมแซมคำตอบ

สตริงคู่ที่สองจะทำการครอสโอเวอร์ด้วยวิธี OX ที่ตำแหน่งที่ 3 และ 6 เช่นเดียวกับการครอสโอเวอร์โดยโปรแกรม ซึ่งจะได้สตริงพ่อแม่และช่วงที่จะครอสโอเวอร์ ดังนี้

p_1 = [2 6 | 4 1 9 5 | 3 7 8 11 10 12]

p_2= [2 4 | 1 6 9 5 | 3 7 11 8 10 12]

เมื่อทำการสลับสตริงที่อยู่ในตำแหน่งที่ 3 ถึง 6 (หรือที่อยู่ระหว่างเครื่องหมาย | |) จะได้สตริงรุ่นลูกเป็น

o_1 = [2 4 1 6 9 5 3 7 8 11 10 12]

o_2 = [2 6 4 1 9 5 3 7 11 8 10 12]

จากนั้น ตรวจสอบสตริงรุ่นลูกที่ได้ว่าขัดกับหลักความสัมพันธ์ก่อนหลังของงานหรือไม่ ซึ่งในที่นี้พบว่าสตริงรุ่นลูกทั้ง 2 ตัวไม่ขัดกับหลักความสัมพันธ์ดังกล่าว

สตริงคู่สุดท้ายจะถูกครอสโอเวอร์ ที่ตำแหน่งที่ 7 และ 9 ซึ่งจะได้สตริงพ่อแม่และช่วงที่จะครอสโอเวอร์ ดังนี้

p_1 = [1 4 9 2 6 5 | 3 7 8 | 11 10 12]

p_2 = [4 2 1 9 6 11 | 5 3 7 | 8 10 12]

ทำการสลับสตริงที่อยู่ในตำแหน่งที่ 3 ถึง 6 (หรือที่อยู่ระหว่างเครื่องหมาย | |)จะได้สตริงรุ่นลูกเป็น

o_1 = [1 4 9 2 6 8 5 3 7 11 10 12]

o_2 = [4 2 1 9 6 11 3 7 8 5 10 12]

จะได้ว่าสตริงรุ่นลูกทั้งสองเป็นคำตอบที่เป็นไปไม่ได้ จึงต้องทำการซ่อมแซมคำตอบโดยนำเอา Precedence Metrix ที่ได้ในตอนแรกมาใช้ในการหางานที่ไม่มีงานก่อนหน้าตัวอย่างการซ่อมแซมคำตอบรุ่นลูกตัวแรก แสดงได้ดังตารางที่ ค2.8 ซึ่งจะได้สตริงคำตอบรุ่นลูกตัวใหม่เป็น [1 4 9 2 6 5 3 7 8 11 10 12] การซ่อมแซมคำตอบรุ่นลูกตัวที่สองก็ใช้วิธีเดียวกัน จะได้สตริงรุ่นลูกตัวที่สองตัวใหม่เป็น [4 2 1 9 6 11 3 5 7 8 10 12]

ตารางที่ ค2.8 *ผลการซ่อมแซมคำตอบรุ่นลูกตัวแรกที่ได้จากการครอสโอเวอร์สตริงคู่ที่ 3*

Iteration	Available Set	Infeasible Offspring	Repaired String
1	{1 2 4 6 9}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1]
2	{ 2 4 6 9}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4]
3	{2 6 9}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9]
4	{2 6}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9 2]
5	{3 6}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9 2 6]
6	{3 5 11}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9 2 6 5]
7	{3 11 }	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9 2 6 5 3]
8	{7 11}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9 2 6 5 3 7]
9	{8 11}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9 2 6 5 3 7 8]
10	{11}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9 2 6 5 3 7 8 11]
11	{10}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9 2 6 5 3 7 8 11 10]
12	{12}	[1 4 9 2 6 8 5 3 7 11 10 12]	[1 4 9 2 6 5 3 7 8 11 10 12]

เมื่อพิจารณาสตริงที่ได้เทียบกับผลที่ได้จากโปรแกรมพบว่าได้ผลเหมือนกัน แสดงว่าโปรแกรมสามารถทำการครอสโอเวอร์และซ่อมแซมสตริงคำตอบได้อย่างถูกต้อง และสตริงที่ได้ทุกตัวเป็นสตริงคำตอบที่เป็นไปได้

เมื่อทำการครอสโอเวอร์ และซ่อมแซมคำตอบจนได้สตริงรุ่นลูกทั้ง 3 คู่แล้ว ก็ให้นำสตริงรุ่นลูกที่ได้ไปทำการถอดรหัส และประเมินค่า โดยจะต้องทำการหาค่า *cal_ct* ที่จะใช้กำหนดความสามารถของสถานีทำงานก่อน แล้วจึงหาค่าจำนวนสถานี รอบเวลาการผลิตจริง ค่า *wv* และ ค่า Fitness ของสตริงคำตอบทุกตัว สตริงรุ่นลูกที่ถอดรหัสและประเมินค่าแล้วจะถูกนำไปแทนที่สตริงพ่อแม่ แล้วนำไปรวมกับสตริงที่ไม่ได้ถูกนำไปครอสโอเวอร์ จากการถอดรหัสและประเมินค่าโดยโปรแกรมและผลที่ได้จากการทดลองทำด้วยมือเพื่อทดสอบความถูกต้องของโปรแกรม จะได้ผลเหมือนกัน ซึ่งผลที่ได้แสดงไว้ในตารางที่ ค2.9 และ ค2.10

ตารางที่ ค2.9 *ผลการถอดรหัสสตริงคำตอบของปัญหา 12 งานที่ได้ภายหลังการครอสโอเวอร์*

String No.	String Order	Cal. ct	Workstation Order	n	Ti	Real ct
1	[2 6 9 4 1 11 5 3 7 8 10 12]	6.30	[1 1 1 2 3 3 4 4 5 6 7 8]	8	4 4 5 6 2 5 4 4	6
2	[4 6 1 2 9 5 3 11 7 8 10 12]	5.20	[1 1 2 2 3 3 4 4 5 6 7 8]	8	5 5 4 5 2 5 4 4	5
3	[4 1 6 9 5 2 11 3 7 8 10 12]	5.15	[1 2 2 2 3 3 4 4 5 6 7 8]	8	4 5 5 5 2 5 4 4	5

ตารางที่ ค2.9 ผลการถอดรหัสสตริงคำตอบของปัญหา 12 งานที่ได้ภายหลังการครอสโอเวอร์ (ต่อ)

String No.	String Order	Cal. ct	Workstation Order	n	Ti	Real ct
1	[2 6 9 4 1 11 5 3 7 8 10 12]	6.30	[1 1 1 2 3 3 4 4 5 6 7 8]	8	4 4 5 6 2 5 4 4	6
2	[4 6 1 2 9 5 3 11 7 8 10 12]	5.20	[1 1 2 2 3 3 4 4 5 6 7 8]	8	5 5 4 5 2 5 4 4	5
3	[4 1 6 9 5 2 11 3 7 8 10 12]	5.15	[1 2 2 2 3 3 4 4 5 6 7 8]	8	4 5 5 5 2 5 4 4	5
4	[6 1 11 2 3 9 4 5 7 8 10 12]	7.00	[1 1 1 2 2 2 3 3 4 4 5 6]	6	6 6 7 7 4 4	7
5	[4 1 2 9 6 3 5 7 8 11 10 12]	5.70	[1 2 2 3 3 3 4 4 5 6 7 8]	8	4 5 5 5 5 2 4 4	5
6	[2 4 1 6 9 5 3 7 8 11 10 12]	6.30	[1 1 2 2 2 3 3 4 4 5 5 6]	6	6 5 6 7 6 4	6
7	[2 6 4 1 9 5 3 7 11 8 10 12]	8.44	[1 1 1 2 2 2 3 3 3 4 5 5]	5	7 7 7 5 8	8
8	[2 6 4 1 9 5 3 7 8 11 10 12]	6.00	[1 1 2 3 3 4 4 5 6 7 7 8]	8	3 4 4 6 2 5 6 4	6
9	[1 4 9 2 6 5 3 7 8 11 10 12]	8.12	[1 1 1 2 2 2 3 3 4 4 5 5]	5	8 6 5 7 8	8
10	[4 2 1 9 6 11 3 5 7 8 10 12]	7.60	[1 1 2 2 2 2 3 3 4 4 5 6]	6	6 7 6 7 4 4	6

ตารางที่ ค2.10 ผลการประเมินค่าสตริงคำตอบของปัญหา 12 งานที่ได้ภายหลังการครอสโอเวอร์

String No.	wv	Fitness	String No.	wv	Fitness
1	1.1875	9.1890	6	0.8889	11.5683
2	0.9375	10.1416	7	0.9600	12.2116
3	0.9375	10.1416	8	1.6875	8.5898
4	1.5555	10.1153	9	1.3600	11.4758
5	0.9375	10.1416	10	1.5556	10.4104

จากตารางที่ ค2.9 จะเห็นได้ว่าเวลาทำงานของทุกสถานีทำงานไม่เกินรอบเวลาการผลิตที่กำหนด ไม่มีขั้นงานใดถูกจัดให้กับหลายสถานี และทุกขั้นงานถูกจัดให้กับสถานีทำงานใดสถานีทำงานหนึ่ง จำนวนสถานีที่ได้จากสตริงคำตอบแต่ละตัวก็ไม่เกิน 9 ซึ่งเป็นจำนวนสถานีสูงสุดที่ยอมรับได้ ทำให้ยืนยันได้ว่าโปรแกรมสามารถถอดรหัสสตริงคำตอบได้อย่างถูกต้อง

7) การเก็บค่าที่ดีที่สุดหลังการครอสโอเวอร์

หลังการถอดรหัสและประเมินค่าสตริงคำตอบที่ได้จากการครอสโอเวอร์แล้ว โปรแกรมจะทำการหาสตริงคำตอบที่ดีที่สุด 4 ตัวตามเกณฑ์ต่างๆ แล้วนำไปเปรียบเทียบกับชุดของสตริงคำตอบที่ดีที่สุดที่เก็บไว้ ผลการเปรียบเทียบจะได้ชุดของค่าที่ดีที่สุดที่เก็บไว้เหมือนเดิม

จากการพิจารณาหาชุดของสตริงที่ดีที่สุดภายหลังกระบวนการครอสโอเวอร์ โดยไม่ใช้โปรแกรม จะได้สตริงที่ดีที่สุดในแต่ละเกณฑ์ดังตารางที่ ค2.11

ตารางที่ ค2.11 *สตริงคำตอบภายหลังการครอสโอเวอร์ที่ดีที่สุดในแต่ละเกณฑ์การพิจารณา*

	Criteria	Best String	String Sequence	Fitness	n	Real ct	wv
1	Max. Fitness	7	[2 6 4 1 9 5 3 7 11 8 10 12]	**12.2116**	5	8	0.96
2	Min. n	7	[2 6 4 1 9 5 3 7 11 8 10 12]	12.2116	**5**	8	0.96
3	Min. real_ct	2	[4 6 1 2 9 5 3 11 7 8 10 12]	10.1416	8	**5**	0.9375
3	Min wv.	6	[2 4 1 6 9 5 3 7 8 11 10 12]	11.5683	6	6	**0.8889**

สตริงที่ดีที่สุดทั้ง 4 ตัวนี้จะถูกนำไปเปรียบเทียบกับชุดของสตริงคำตอบที่เก็บไว้เป็นคู่ๆตามเกณฑ์การพิจารณา

ในการเปรียบเทียบค่า Fitness ระหว่างสตริงที่ได้จากการครอสโอเวอร์และสตริงที่เก็บไว้ตัวแรก ต้องนำจำนวนสถานีทำงาน รอบเวลาผลิตจริงและค่า wv ของสตริงทั้งสองมาคำนวณหาค่า Fitness ระหว่างสตริงเพียงสองตัวนี้ก่อน ซึ่งจะได้ค่า Fitness ใหม่ของสตริงจากการครอสโอเวอร์และสตริงที่เก็บเป็น 12.1762 และ 12.44141761 ตามลำดับ จากนั้นจึงนำมาเปรียบเทียบกัน ผลการเปรียบเทียบและแทนที่สตริงคำตอบทั้งหมด แสดงได้ดังตารางที่ ค2.12

ตารางที่ ค2.12 *ผลการเปรียบเทียบและแทนที่สตริงคำตอบที่ดีที่สุด*

คู่ที่	Criteria	Elite Preserve Solution	Best from X	Result
1	Max. Fitness	12.4414	12.1762	**Elite Preserve Solution**
2	Min. n	5	5	**Elite Preserve Solution**
3	Min. ct	5	5	**Elite Preserve Solution**
4	Min. wv	0.4375	0.8889	**Elite Preserve Solution**

จากตารางที่ ค2.12 จะได้ว่า Elite Preserve Solution จะให้ค่าที่ดีกว่าหรือเท่ากับสตริงคำตอบที่ดีที่สุดได้จากการครอสโอเวอร์ จึงให้คงชุดของสตริงคำตอบที่ดีที่สุดไว้ตามเดิม ซึ่งจะผลที่ได้เหมือนกับผลที่ได้จากการคำนวณด้วยโปรแกรม แสดงว่าโปรแกรมมีความถูกต้อง

8) การมิวเตชั่น

ในขั้นตอนนี้ โปรแกรมจะทำการสุ่มเลือกสตริงคำตอบมาทำการมิวเตชั่น โดยพิจารณาจากสตริงคำตอบที่มีค่าสุ่ม r น้อยกว่าค่า Pm ผลการสุ่มค่าและคัดเลือกแสดงได้ดังตารางที่ ค2.13

ตารางที่ ค2.13 *ผลการเลือกสตริงคำตอบไปมิวเตชั่น*

String No.	*ri*	*ri < 0.2 (Pm)*	String No.	*ri*	*ri < 0.2 (Pm)*
1	0.4103		6	0.6548	
2	0.8936		7	0.5105	
3	0.0579	●	8	0.3529	
4	0.4924		9	0.0099	●
5	0.4266		10	0.8416	

หมายเหตุ ● หมายถึงสตริงตัวนั้นถูกเลือกไปทำการมิวเตชั่น

จะได้ว่าสตริงที่จะนำมามิวเตชั่น คือ สตริงตัวที่ 3 และ 9 ซึ่งมีลำดับเป็น [4 1 6 9 5 2 11 3 7 8 10 12] และ [1 4 9 2 6 5 3 7 8 11 10 12] จากโปรแกรมจะได้สตริงรุ่นลูกจากสตริงทั้ง 2 เป็น[4 1 2 6 9 5 3 11 7 8 10 12] และ [1 4 9 2 6 3 5 7 8 11 10 12] ตามลำดับ

เมื่อทำการพิจารณาเพื่อตรวจสอบความถูกต้องของโปรแกรม พบว่าเมื่อกำหนดให้ Pm=0.2 ทำให้สามารถคาดเดาได้ว่าน่าจะมีสตริงคำตอบ 20 % หรือ 0.2*10 =2 ตัว ที่จะถูกมิวเตชั่น ซึ่งจำนวนสตริงที่คาดหมายว่าจะถูกมิวเตชั่นตรงกับจำนวนสตริงที่โปรแกรมคัดเลือกมา เมื่อพิจารณาค่าสุ่มดังตารางที่ ค2.13 ก็จะพบว่าสตริงตัวที่มีค่าสุ่มน้อยกว่า 0.2 คือสตริงหมายเลข 3 และ 9 ตรงตามที่โปรแกรมคัดเลือกไว้

จากนั้นทดลองทำการมิวเตชั่นที่ตำแหน่งที่ 1 ถึง 2 เช่นเดียวกับที่โปรแกรมใช้ จะได้ว่าสตริงตัวใหม่จะมีค่าในตำแหน่งที่1-2 เหมือนสตริงตัวเดิม

$$p_1 = [4\ 1\ x\ x\ x\ x\ x\ x\ x\ x\ x\ x]$$

ในตำแหน่งที่เหลือจะใช้วิธีกำหนดงานเหมือนกับการสร้างประชากรเบื้องต้น โดยในตอนนี้มีงานที่ถูกจัดไปแล้ว 2 งาน ในตำแหน่งที่ 1-2 การจัดงานในตำแหน่งที่เหลือแสดงได้ดังตารางที่ ค2.14

ตารางที่ ค2.14 การหาสตริงคำตอบตัวใหม่จากการมิวเตชั่น

Iteration	Available Set	Random Selected String	String Sequence
2	-	-	[4 1]
3	{2 6 9}	2	[4 1 2]
4	{3 6 9}	6	[4 1 2 6]
5	{3 5 9 11}	9	[4 1 2 6 9]
6	{3 5 11}	5	[4 1 2 6 9 5]
7	{3 11}	3	[4 1 2 6 9 5 3]
8	{7 11}	11	[4 1 2 6 9 5 3 11]
9	{7}	7	[4 1 2 6 9 5 3 11 7]
10	{8}	8	[4 1 2 6 9 5 3 11 7 8]
11	{10}	10	[4 1 2 6 9 5 3 11 7 8 10]
12	{12}	12	[4 1 2 6 9 5 3 11 7 8 10 12]

สำหรับสตริงตัวที่ 9 ที่จะทำการมิวเตชั่นที่ตำแหน่งที่ 5 ดังนั้นสตริงตัวใหม่จะมีค่าในตำแหน่งที่1-5 เหมือนสตริงตัวเดิม

$$p_1 = [1\ 4\ 9\ 2\ 6\ x\ x\ x\ x\ x\ x]$$

การกำหนดงานในตำแหน่งที่เหลือจะใช้วิธีเหมือนเดิม ซึ่งจะได้สตริงตัวใหม่เป็น [1 4 9 2 6 3 5 7 8 11 10 12]

ผลการทดลองทำการมิวเตชั่นโดยไม่ใช้โปรแกรมจะให้สตริงคำตอบรุ่นลูกเหมือนกับสตริงคำตอบรุ่นลูกที่ได้จากโปรแกรม แสดงว่าโปรแกรมสามารถทำการมิวเตชั่นได้อย่างถูกต้อง และสตริงคำตอบที่ได้เป็นสตริงคำตอบที่มีความเป็นไปได้

ภายหลังการมิวเตชั่น สตริงตัวใหม่จะต้องถูกนำมาถอดรหัสและประเมินค่าแล้วเอาไปรวมกับสตริงที่เหลือ จากการถอดรหัสและประเมินค่าโดยโปรแกรมและผลที่ได้จากการทดลองทำด้วยมือเพื่อทดสอบความถูกต้องของโปรแกรม จะได้ผลเหมือนกัน ซึ่งผลที่ได้แสดงได้ดังตารางที่ ค2.15 และ ค2.16

ตารางที่ ค2.15 ผลการถอดรหัสสตริงคำตอบที่ได้ภายหลังการมิวเตชั่น

String No	String Order	Cal ct	Workstation Order	n	Ti	Real ct
1	[2 6 9 4 1 11 5 3 7 8 10 12]	6.30	[1 1 1 2 3 3 4 4 5 6 7 8]	8	4 4 5 6 2 3 4 4	6
2	[4 6 1 2 9 5 3 11 7 8 10 12]	5.20	[1 1 2 2 3 3 4 4 5 6 7 8]	8	5 5 4 5 2 3 4 4	5
3	[4 1 2 6 9 5 3 11 7 8 10 12]	8.00	[1 1 2 2 2 2 3 3 3 4 5 5]	5	7 7 7 5 3	8
4	[6 1 11 2 3 9 4 5 7 8 10 12]	7.00	[1 1 1 2 2 2 3 3 4 4 5 6]	6	6 6 7 7 4 4	7

ตารางที่ ค2.15 ผลการถอดรหัสสตริงคำตอบที่ได้ภายหลังการมิวเตชั่น (ต่อ)

String No.	String Order	Cal. ct	Workstation Order	n	Ti	Real ct
5	[4 1 2 9 6 3 5 7 8 11 10 12]	5.70	[1 2 2 3 3 3 4 4 5 6 7 8]	8	4 5 5 5 5 2 4 4	5
6	[2 4 1 6 9 5 3 7 8 11 10 12]	6.30	[1 1 2 2 2 3 3 4 4 5 5 6]	6	6 5 6 7 6 4	6
7	[2 6 4 1 9 5 3 7 11 8 10 12]	8.44	[1 1 1 2 2 2 3 3 3 4 5 5]	5	7 7 7 5 8	8
8	[2 6 4 1 9 5 3 7 8 11 10 12]	6.00	[1 1 2 3 3 4 4 5 6 7 7 8]	8	3 4 4 6 2 5 6 4	6
9	[1 4 9 2 6 3 5 7 8 11 10 12]	7.00	[1 1 2 2 2 2 3 3 4 4 5 6]	6	7 7 5 7 4 4	7
10	[4 2 1 9 6 11 3 5 7 8 10 12]	7.60	[1 1 2 2 2 2 3 3 4 4 5 6]	6	6 7 6 7 4 4	6

ตารางที่ ค2.16 ผลการประมาณค่าสตริงคำตอบที่ได้ภายหลังการมิวเตชั่น

String No.	wv	Fitness	String No.	wv	Fitness
1	1.1875	9.1056	6	0.8889	11.4350
2	0.9375	10.0716	7	0.9600	12.0216
3	0.9375	12.0816	8	1.6875	8.5064
4	1.5555	9.9724	9	1.3600	10.1944
5	0.9375	10.0716	10	1.5556	10.2771

จากตารางที่ ค2.15 จะเห็นได้ว่าเวลาทำงานของทุกสถานีทำงานไม่เกินรอบเวลาการผลิตที่กำหนด ไม่มีขั้นงานใดถูกจัดให้กับหลายสถานี และทุกขั้นงานถูกจัดให้กับสถานีทำงานใดสถานีทำงานหนึ่ง จำนวนสถานีที่ได้จากสตริงคำตอบแต่ละตัวก็ไม่เกิน 9 ซึ่งเป็นจำนวนสถานีสูงสุดที่ยอมรับได้ ทำให้ยืนยันได้ว่าโปรแกรมสามารถถอดรหัสสตริงคำตอบได้อย่างถูกต้อง

9) การเก็บค่าที่ดีที่สุดหลังการมิวเตชั่น

จากสตริงคำตอบทั้ง 10 ตัวที่ได้ภายหลังการมิวเตชั่น โปรแกรมจะทำการหาสตริงคำตอบที่ดีที่สุด 4 ตัวตามเกณฑ์ต่างๆ แล้วนำไปเปรียบเทียบกับชุดของสตริงคำตอบที่ดีที่สุดที่เก็บไว้ ชุดของสตริงคำตอบที่ดีที่สุดชุดใหม่จากโปรแกรม แสดงได้ดังตารางที่ ค2.17

ตารางที่ ค2.17 *ชุดของสตริงคำตอบที่ดีที่สุดชุดใหม่ภายหลังการมิวเตชั่น*

Elite Preserve Solution	Criteria	String Sequence	Fitness	n	Real ct	wv
1st	Max. Fitness	[4 2 1 9 6 11 5 3 7 8 10 12]	**12.4414**	8	5	0.4375
2nd	Min. n	[2 6 4 1 9 5 3 7 8 11 10 12]	12.1762	**5**	8	0.9600
3rd	Min. real_ct	[4 2 1 9 6 11 5 3 7 8 10 12]	12.4414	8	**5**	0.4375
4th	Min wv.	[4 2 1 9 6 11 5 3 7 8 10 12]	12.4414	6	6	**0.4375**

จากการพิจารณาหาชุดของสตริงที่ดีที่สุดภายหลังกระบวนการมิวเตชั่นโดยไม่ใช้โปรแกรม จะได้สตริงที่ดีที่สุดในแต่ละเกณฑ์ดังตารางที่ ค2.18

ตารางที่ ค2.18 *สตริงคำตอบภายหลังการมิวเตชั่นที่ดีที่สุดในแต่ละเกณฑ์การพิจารณา*

	Criteria	Best String	String Sequence	Fitness	n	Real ct	wv
1	Max. Fitness	3	[4 1 2 6 9 5 3 11 7 8 10 12]	**12.0816**	5	8	0.9375
2	Min. n	3	[4 1 2 6 9 5 3 11 7 8 10 12]	12.0816	**5**	8	0.9375
3	Min. real_ct	2	[4 6 1 2 9 5 3 11 7 8 10 12]	10.0716	8	**5**	0.9375
3	Min wv.	6	[2 4 1 6 9 5 3 7 8 11 10 12]	11.4350	6	6	**0.8889**

สตริงที่ดีที่สุดทั้ง 4 ตัวนี้จะถูกนำไปเปรียบเทียบกับชุดของสตริงคำตอบที่เก็บไว้เป็นคู่ๆตามเกณฑ์การพิจารณา โดยเริ่มจากเกณฑ์ที่มี Priority สูงไปต่ำที่สุด

ในที่นี้คำนวณหาค่า Fitness ใหม่ ของสตริงที่ดีที่สุดจากการมิวเตชั่นและสตริงคำตอบที่เก็บไว้ได้เป็น 12.4414 และ 12.1762 ตามลำดับ ผลการเปรียบเทียบสตริงคำตอบ แสดงไว้ในตารางที่ ค2.19 ซึ่งจะได้ว่าสตริงคำตอบที่เก็บไว้ จะให้ค่า Fitness ที่ดีกว่าสตริงหมายเลข 3 ที่ได้จากการมิวเตชั่น จึงนำเอาสตริงที่เก็บไว้ไปแทนที่สตริงหมายเลข 3

สำหรับการเปรียบเทียบโดยดูจากจำนวนสถานีทำงานที่น้อยที่สุด จะได้ว่าสตริงทั้งสองให้ค่าเท่ากัน จึงต้องพิจารณาจาก Fitness ซึ่งก็ให้ค่าเท่ากันอีก ดังนั้นจึงคงคำตอบใน Elite Preserve Solution ไว้ดามเดิม

รอบเวลาการผลิตจริงของสตริงทั้งสองตัวก็มีค่าเท่ากัน แต่เมื่อพิจารณาจากค่า Fitness จะได้ว่าสตริงคำตอบที่เก็บไว้ตัวเดิมจะให้ค่าที่ดีกว่า จึงนำเอาสตริงที่เก็บไว้ไปแทนที่สตริงหลังการมิวเตชั่นตัวที่มีรอบเวลาการผลิตสูงที่สุด (สตริงตัวที่ 7)

เมื่อพิจารณาการเปรียบเทียบคู่ที่ 4 จะได้ว่าสตริงคำตอบที่เก็บไว้ดีกว่า จึงต้องนำเอาสตริงที่เก็บไว้มาแทนที่สตริงหมายเลข 8 ซึ่งมีค่า *wv* สูงที่สุด

ชุดของสตริงคำตอบที่ดีที่สุดชุดใหม่ที่ได้เหมือนกับการคำนวณโดยโปรแกรม ซึ่งแสดงว่าโปรแกรมสามารถเปรียบเทียบและเลือกเก็บชุดของสตริงคำตอบที่ดีที่สุดภายหลังกระบวนการมิวเตชั่นได้อย่างถูกต้อง

ตารางที่ ค2.19 *ผลการเปรียบเทียบสตริงคำตอบที่ดีที่สุด*

คู่ที่	First Criteria	Elite Preserve Solution	Best from Mut.	Result	Second Criteria	Elite Preserve Solution	Best from Mut.	Result
1	Max. Fitness	12.4414	12.1762	**Elite Preserve Solution**	-			
2	Min. n	5	5	-				
					Max. Fitness	2	2	**Elite Preserve Solution**
3	Min. ct	5	5	-				
					Max. Fitness	2.2286	1.8933	**Elite Preserve Solution**
4	Min. wv	0.4375	0.8889	**Elite Preserve Solution**	-			

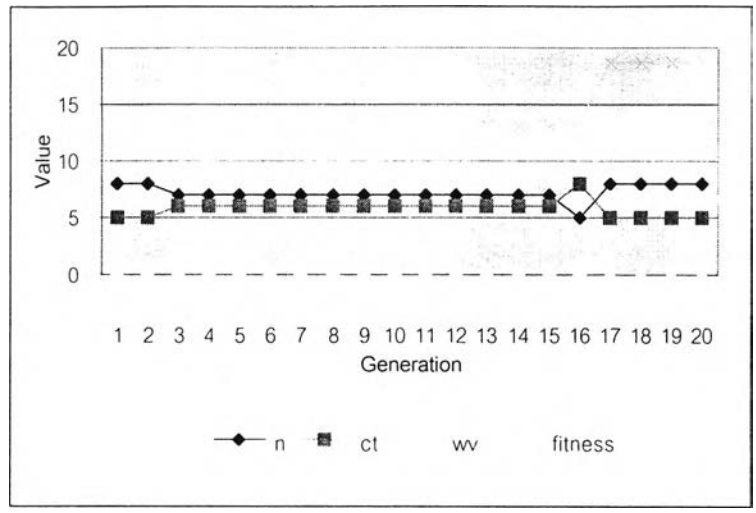
สตริงที่ได้ภายหลังการแทนที่จะกลายเป็นสตริงคำตอบรุ่นพ่อแม่และเข้าสู่กระบวนการของ GAs ในเจนเนอเรชั่นถัดไป

2. *การทดสอบความถูกต้องของโปรแกรมในการลู่เข้าหาคำตอบที่ดีที่สุด*

เมื่อจบกระบวนการในเจนเนอเรชั่นที่ 1 สตริงคำตอบทั้ง 10 ตัวที่ได้จะกลายเป็นสตริงคำตอบเบื้องต้นในเจนเนอเรชั่นต่อไป ซึ่งเมื่อปล่อยให้โปรแกรมหาคำตอบในเจนเนอเรชั่นต่อๆ ไป จนถึงเจนเนอเรชั่นที่ 20 ซึ่งเป็นเจนเนอเรชั่นสูงสุดที่กำหนดให้ จะได้สตริงคำตอบที่ดีที่สุดในแต่ละเจนเนอเรชั่นดังตารางที่ ค2.20 และกราฟแสดงค่าวัตถุประสงค์ต่างๆแสดงได้ดังรูปที่ 8.4

ตารางที่ ค2.20 สตริงคำตอบที่ดีที่สุดของเจนเนอเรชั่นที่ 1 ถึง 20

Gen.	String Order	Station order	n	ct	wv	Fitness
1	[6 9 1 4 2 11 5 3 7 8 10 12]	[1 1 1 2 3 3 4 5 5 6 7 8]	8	5	0.438	12.4415
2	[6 9 1 4 2 11 5 3 7 8 10 12]	[1 1 1 2 3 3 4 5 5 6 7 8]	8	5	0.438	12.4415
3	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
4	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
5	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
6	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
7	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
8	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
9	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
10	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
11	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
12	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
13	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
14	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
15	[9 4 2 6 5 11 1 3 7 8 10 12]	[1 1 2 2 2 3 3 4 4 5 6 7]	7	6	0.408	13.1312
16	[6 1 11 2 3 9 4 5 7 8 10 12]	[1 1 2 2 3 3 4 5 5 6 7 8]	5	8	0.56	13.6985
17	[6 1 11 2 3 9 4 5 7 8 10 12]	[1 1 2 2 3 3 4 5 5 6 7 8]	8	5	0.188	18.6768
18	[6 1 11 2 3 9 4 5 7 8 10 12]	[1 1 2 2 3 3 4 5 5 6 7 8]	8	5	0.188	18.6768
19	[6 1 11 2 3 9 4 5 7 8 10 12]	[1 1 2 2 3 3 4 5 5 6 7 8]	8	5	0.188	18.6768
20	[6 1 11 2 3 9 4 5 7 8 10 12]	[1 1 2 2 3 3 4 5 5 6 7 8]	8	5	0.188	18.6768



รูปที่ ค2.3 กราฟแสดงค่าวัตถุประสงค์ต่าง ๆ ของสตริงที่ดีที่สุดเจนเนอเรชั่นที่ 1 - 20

จากรูปที่ ค2.3 จะเห็นได้ว่าค่า Fitness หรือค่าวัดถุประสงค์รวมจะมีแนวโน้มเพิ่มขึ้นเรื่อย ๆ และลู่เข้าหาค่ามากค่าหนึ่ง ซึ่งผลที่ได้มีความถูกต้องตามทฤษฎี ค่าวัดถุประสงค์รวมบางช่วงจะคงที่เนื่องจากค่าวัตถุประสงค์อื่นคงที่ แต่เมื่อค่าวัดถุประสงค์อื่นมีการเปลี่ยนแปลงค่าวัดถุประสงค์รวมก็จะเพิ่มขึ้น แสดงให้เห็นว่าโปรแกรมสามารถดำเนินการตามกระบวนการของ MOGA ได้อย่างถูกต้อง กล่าวคือสามารถหาคำตอบที่ดีขึ้นเรื่อย ๆ ได้

3. การทดสอบความถูกต้องของโปรแกรมในการหาคำตอบที่เป็นไปได้ที่ดีที่สุด

ภายหลังจากการรันโปรแกรมเสร็จสิ้นลง จะได้คำตอบสำหรับปัญหาการจัดสมดุลของสายงานการประกอบสำหรับปัญหาตัวอย่างนี้ ดังตารางที่ ค2.21 และตารางที่ ค2.22

ตารางที่ ค2.21 *ผลการจัดสมดุลสายงานการประกอบของปัญหา 12 งานโดย MOGA*

สถานีที่	ขั้นงาน	เวลาของสถานี
1	6,1	4
2	11, 2	4
3	3, 9	4
4	4	4
5	5, 7	5
6	8	5
7	10	4
8	12	4
รวม		34

ตารางที่ ค2.22 ค่า Measure of Performance ต่างๆของปัญหา 12 งานที่ได้จากMOGA

จำนวนสถานี (สถานี)	รอบเวลาผลิตจริง (นาที)	Workload Variance	เวลาว่างงานรวม (นาที)	ประสิทธิภาพสายการประกอบ (%)
8	5	0.188	6	85

จะเห็นได้ว่าคำตอบที่ได้ไม่ขัดกับหลักความสัมพันธ์ก่อน-หลังของงาน ไม่มีงานใดที่ถูกจัดให้กับสถานีทำงานมากกว่า 1 สถานี และทุกงานถูกจัดให้กับสถานีทำงานใดสถานีทำงานหนึ่ง นอกจากนี้ จำนวนสถานีทำงานที่ได้ก็น้อยกว่า 9 ซึ่งเป็นจำนวนสถานีทำงานสูงสุดที่ยอมรับได้ เวลาการทำงานของแต่ละสถานีน้อยกว่า 9 ซึ่งเป็นรอบเวลาการผลิตสูงสุดที่กำหนด ดังนั้นคำตอบที่ได้จากโปรแกรมจึงเป็นคำตอบที่เป็นไปได้

เมื่อพิจารณาตารางที่ ค2.20 ก็จะเห็นได้ว่าคำตอบที่ดีที่สุดคือคำตอบที่ได้ในเจนเนอเรชั่นที่ 16 ซึ่งตรงกับคำตอบที่ดีที่สุดที่ได้จากโปรแกรม ดังนั้นจะได้ว่าโปรแกรมสามารถให้คำตอบที่เป็นคำตอบที่ดีที่สุดได้

จากการทดสอบทั้ง 3 ส่วนจะเห็นได้ว่าโปรแกรมมีความถูกต้องทั้งในด้านการดำเนินการตามวิธีการของ GAs การลู่เข้าหาคำตอบ และการหาคำตอบที่เป็นไปได้ที่ดีที่สุด

ภาคผนวก ง-1

ข้อมูลการทดลองเพื่อทดสอบค่าพารามิเตอร์ของ GAs

1. ปัญหา 11 งาน

ตารางที่ ง-1.1 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 11 งาน replication ที่ 1

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
1	10	1	0.7	0.1	0.5556	
2				0.2	0.5556	
3				0.3	0.5556	
4				0.4	0.5556	
5			0.8	0.1	0.8889	
6				0.2	0.5556	
7				0.3	0.5556	
8				0.4	0.5556	
9			0.9	0.1	0.5556	
10				0.2	0.5556	
11				0.3	0.5556	
12				0.4	0.5556	
13			1.0	0.1	0.8889	
14				0.2	0.5556	
15				0.3	0.5556	
16				0.4	0.5556	
17		2	0.7	0.1	0.5556	
18				0.2	0.5556	
19				0.3	0.5556	
20				0.4	0.5556	
21			0.8	0.1	0.8889	
22				0.2	0.5556	
23				0.3	0.5556	
24				0.4	0.5556	
25			0.9	0.1	0.5556	
26				0.2	0.5556	
27				0.3	0.5556	
28				0.4	0.5556	
29			1.0	0.1	0.8889	
30				0.2	0.5556	
31				0.3	0.5556	
32				0.4	0.5556	
33	10	3	0.7	0.1	0.5556	
34				0.2	0.5556	
35				0.3	0.5556	
36				0.4	0.5556	
37			0.8	0.1	0.8889	
38				0.2	0.5556	
39				0.3	0.5556	
40				0.4	0.5556	
41			0.9	0.1	0.5556	
42				0.2	0.5556	
43				0.3	0.5556	
44				0.4	0.5556	
45			1.0	0.1	0.8889	
46				0.2	0.5556	
47				0.3	0.5556	
48				0.4	0.5556	
49		4	0.7	0.1	0.5556	
50				0.2	0.5556	
51				0.3	0.5556	
52				0.4	0.5556	
53			0.8	0.1	0.8889	
54				0.2	0.5556	
55				0.3	0.5556	
56				0.4	0.5556	
57			0.9	0.1	0.5556	
58				0.2	0.5556	
59				0.3	0.5556	
60				0.4	0.5556	
61			1.0	0.1	0.8889	
62				0.2	0.5556	
63				0.3	0.5556	
64				0.4	0.5556	
65	10	5	0.7	0.1	0.5556	
66				0.2	0.5556	
67				0.3	0.5556	
68				0.4	0.5556	
69			0.8	0.1	0.8889	
70				0.2	0.5556	
71				0.3	0.5556	
72				0.4	0.5556	
73			0.9	0.1	0.5556	
74				0.2	0.5556	
75				0.3	0.5556	
76				0.4	0.5556	
77			1.0	0.1	0.8889	
78				0.2	0.5556	
79				0.3	0.5556	
80				0.4	0.5556	
81		6	0.7	0.1	0.5556	
82				0.2	0.5556	
83				0.3	0.5556	
84				0.4	0.5556	
85			0.8	0.1	0.8889	
86				0.2	0.5556	
87				0.3	0.5556	
88				0.4	0.5556	
89			0.9	0.1	0.5556	
90				0.2	0.5556	
91				0.3	0.5556	
92				0.4	0.5556	
93			1.0	0.1	0.8889	
94				0.2	0.5556	
95				0.3	0.5556	
96				0.4	0.5556	

ตารางที่ ง-1.1 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 11 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
97	15	1	0.7	0.1	0.556	
98				0.2	0.556	21
99				0.3	0.556	23
100				0.4	0.556	33
101			0.8	0.1	0.560	
102				0.2	0.560	1
103				0.3	0.556	96
104				0.4	0.556	2
105			0.9	0.1	0.556	
106				0.2	0.556	26
107				0.3	0.556	1
108				0.4	0.556	2
109			1.0	0.1	0.556	
110				0.2	0.556	2
111				0.3	0.556	1
112				0.4	0.556	2
113		2	0.7	0.1	0.560	
114				0.2	0.556	1
115				0.3	0.556	85
116				0.4	0.556	8
117			0.8	0.1	0.560	
118				0.2	0.556	11
119				0.3	0.556	4
120				0.4	0.556	1
121			0.9	0.1	0.556	
122				0.2	0.556	14
123				0.3	0.556	7
124				0.4	0.556	39
125			1.0	0.1	0.556	
126				0.2	0.556	47
127				0.3	0.556	1
128				0.4	0.556	2

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
129	15	3	0.7	0.1	0.556	
130				0.2	0.556	
131				0.3	0.556	
132				0.4	0.556	
133			0.8	0.1	0.556	
134				0.2	0.556	
135				0.3	0.556	
136				0.4	0.556	
137			0.9	0.1	0.556	
138				0.2	0.556	
139				0.3	0.556	
140				0.4	0.556	
141			1.0	0.1	0.556	
142				0.2	0.556	
143				0.3	0.556	
144				0.4	0.556	
145		4	0.7	0.1	0.556	
146				0.2	0.556	42
147				0.3	0.556	25
148				0.4	0.556	11
149			0.8	0.1	0.560	
150				0.2	0.556	24
151				0.3	0.556	11
152				0.4	0.556	96
153			0.9	0.1	0.556	
154				0.2	0.560	27
155				0.3	0.556	27
156				0.4	0.556	5
157			1.0	0.1	0.556	
158				0.2	0.560	124
159				0.3	0.556	1
160				0.4	0.556	47

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
161	15	5	0.7	0.1	0.560	
162				0.2	0.556	4
163				0.3	0.556	41
164				0.4	0.556	1
165			0.8	0.1	0.556	
166				0.2	0.556	15
167				0.3	0.556	4
168				0.4	0.556	1
169			0.9	0.1	0.560	
170				0.2	0.556	8
171				0.3	0.556	1
172				0.4	0.556	3
173			1.0	0.1	0.556	
174				0.2	0.560	4
175				0.3	0.556	34
176				0.4	0.556	12
177		6	0.7	0.1	0.556	
178				0.2	0.556	5
179				0.3	0.556	12
180				0.4	0.556	3
181			0.8	0.1	0.556	
182				0.2	0.556	27
183				0.3	0.556	4
184				0.4	0.556	19
185			0.9	0.1	0.556	
186				0.2	0.556	3
187				0.3	0.556	54
188				0.4	0.556	1
189			1.0	0.1	0.560	
190				0.2	0.556	2
191				0.3	0.556	2
192				0.4	0.556	2

ตารางที่ ง-1.1 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 11 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
193	20	1	0.7	0.1	0.556	
194				0.2	0.556	18
195				0.3	0.556	1
196				0.4	0.556	2
197			0.8	0.1	0.556	
198				0.2	0.556	34
199				0.3	0.556	6
200				0.4	0.556	14
201			0.9	0.1	0.556	
202				0.2	0.556	1
203				0.3	0.556	18
204				0.4	0.556	6
205			1.0	0.1	0.556	
206				0.2	0.556	26
207				0.3	0.556	15
208				0.4	0.556	1
209		2	0.7	0.1	0.556	
210				0.2	0.556	2
211				0.3	0.556	1
212				0.4	0.556	70
213			0.8	0.1	0.556	
214				0.2	0.556	37
215				0.3	0.556	13
216				0.4	0.556	1
217			0.9	0.1	0.556	
218				0.2	0.556	1
219				0.3	0.556	1
220				0.4	0.556	4
221			1.0	0.1	0.556	
222				0.2	0.556	6
223				0.3	0.556	9
224				0.4	0.556	3

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
225	20	3	0.7	0.1	0.560	
226				0.2	0.556	
227				0.3	0.556	
228				0.4	0.556	
229			0.8	0.1	0.556	
230				0.2	0.556	
231				0.3	0.556	
232				0.4	0.556	
233			0.9	0.1	0.560	
234				0.2	0.556	
235				0.3	0.556	
236				0.4	0.556	
237			1.0	0.1	0.556	
238				0.2	0.556	
239				0.3	0.556	
240				0.4	0.556	
241		4	0.7	0.1	0.556	
242				0.2	0.556	25
243				0.3	0.556	18
244				0.4	0.556	42
245			0.8	0.1	0.556	
246				0.2	0.556	1
247				0.3	0.556	3
248				0.4	0.556	7
249			0.9	0.1	0.556	
250				0.2	0.556	6
251				0.3	0.556	19
252				0.4	0.556	45
253			1.0	0.1	0.556	
254				0.2	0.556	62
255				0.3	0.556	1
256				0.4	0.556	2

No.	Pop size	X type	Pc	Pm	wv	No.
257	20	5	0.7	0.1	0.556	
258				0.2	0.556	25
259				0.3	0.556	1
260				0.4	0.556	7
261			0.8	0.1	0.556	
262				0.2	0.556	3
263				0.3	0.556	1
264				0.4	0.556	5
265			0.9	0.1	0.556	
266				0.2	0.556	19
267				0.3	0.556	2
268				0.4	0.556	8
269			1.0	0.1	0.556	
270				0.2	0.556	1
271				0.3	0.556	2
272				0.4	0.556	1
273		6	0.7	0.1	0.556	
274				0.2	0.556	169
275				0.3	0.556	2
276				0.4	0.556	11
277			0.8	0.1	0.556	
278				0.2	0.556	29
279				0.3	0.556	12
280				0.4	0.556	1
281			0.9	0.1	0.556	
282				0.2	0.556	4
283				0.3	0.556	9
284				0.4	0.556	1
285			1.0	0.1	0.556	
286				0.2	0.556	4
287				0.3	0.556	6
288				0.4	0.556	1

ตารางที่ ง-1.2 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 11 งาน replication ที่ 2

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
289	10	1	0.7	0.1	0.556	
290				0.2	0.556	
291				0.3	0.556	
292				0.4	0.556	
293			0.8	0.1	0.556	
294				0.2	0.556	
295				0.3	0.556	
296				0.4	0.556	
297			0.9	0.1	0.556	
298				0.2	0.556	
299				0.3	0.556	
300				0.4	0.556	
301			1.0	0.1	0.556	
302				0.2	0.556	
303				0.3	0.556	
304				0.4	0.556	
305		2	0.7	0.1	0.556	
306				0.2	0.556	
307				0.3	0.556	
308				0.4	0.556	
309			0.8	0.1	0.556	
310				0.2	0.556	
311				0.3	0.556	
312				0.4	0.556	
313			0.9	0.1	0.889	
314				0.2	0.556	
315				0.3	0.556	
316				0.4	0.556	
317			1.0	0.1	0.556	
318				0.2	0.556	
319				0.3	0.556	
320				0.4	0.556	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
321	10	3	0.7	0.1	0.556	
322				0.2	0.556	
323				0.3	0.556	
324				0.4	0.556	
325			0.8	0.1	0.560	
326				0.2	0.556	
327				0.3	0.556	
328				0.4	0.556	
329			0.9	0.1	0.889	
330				0.2	0.556	
331				0.3	0.556	
332				0.4	0.556	
333			1.0	0.1	0.556	
334				0.2	0.556	
335				0.3	0.556	
336				0.4	0.556	
337		4	0.7	0.1	0.556	
338				0.2	0.556	
339				0.3	0.556	
340				0.4	0.556	
341			0.8	0.1	0.560	
342				0.2	0.556	
343				0.3	0.556	
344				0.4	0.556	
345			0.9	0.1	0.556	
346				0.2	0.556	
347				0.3	0.556	
348				0.4	0.556	
349			1.0	0.1	0.556	
350				0.2	0.560	
351				0.3	0.556	
352				0.4	0.556	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
353	10	5	0.7	0.1	0.556	
354				0.2	0.556	
355				0.3	0.556	
356				0.4	0.556	
357			0.8	0.1	0.556	
358				0.2	0.556	
359				0.3	0.556	
360				0.4	0.556	
361			0.9	0.1	0.556	
362				0.2	0.556	
363				0.3	0.556	
364				0.4	0.556	
365			1.0	0.1	0.556	
366				0.2	0.556	
367				0.3	0.556	
368				0.4	0.556	
369		6	0.7	0.1	0.560	
370				0.2	0.556	
371				0.3	0.556	
372				0.4	0.556	
373			0.8	0.1	0.556	
374				0.2	0.556	
375				0.3	0.556	
376				0.4	0.556	
377			0.9	0.1	0.556	
378				0.2	0.556	
379				0.3	0.556	
380				0.4	0.556	
381			1.0	0.1	0.556	
382				0.2	0.556	
383				0.3	0.556	
384				0.4	0.556	

ตารางที่ ง-1.2 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 11 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
385	15	1	0.7	0.1	0.556	
386				0.2	0.556	1
387				0.3	0.556	53
388				0.4	0.556	67
389			0.8	0.1	0.556	
390				0.2	0.556	1
391				0.3	0.556	18
392				0.4	0.556	6
393			0.9	0.1	0.556	
394				0.2	0.556	5
395				0.3	0.556	15
396				0.4	0.556	6
397			1.0	0.1	0.556	
398				0.2	0.556	3
399				0.3	0.556	8
400				0.4	0.556	31
401		2	0.7	0.1	0.560	
402				0.2	0.556	1
403				0.3	0.556	2
404				0.4	0.556	1
405			0.8	0.1	0.556	
406				0.2	0.556	11
407				0.3	0.556	1
408				0.4	0.556	57
409			0.9	0.1	0.556	
410				0.2	0.556	6
411				0.3	0.556	1
412				0.4	0.556	1
413			1.0	0.1	0.556	
414				0.2	0.556	20
415				0.3	0.556	114
416				0.4	0.556	4

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
417	15	3	0.7	0.1	0.556	
418				0.2	0.556	
419				0.3	0.556	
420				0.4	0.556	
421			0.8	0.1	0.556	
422				0.2	0.556	
423				0.3	0.556	
424				0.4	0.556	
425			0.9	0.1	0.556	
426				0.2	0.556	
427				0.3	0.556	
428				0.4	0.556	
429			1.0	0.1	0.556	
430				0.2	0.556	
431				0.3	0.556	
432				0.4	0.556	
433		4	0.7	0.1	0.556	
434				0.2	0.556	132
435				0.3	0.556	23
436				0.4	0.556	102
437			0.8	0.1	0.556	
438				0.2	0.556	33
439				0.3	0.556	2
440				0.4	0.556	6
441			0.9	0.1	0.556	
442				0.2	0.556	123
443				0.3	0.556	5
444				0.4	0.556	45
445			1.0	0.1	0.556	
446				0.2	0.556	19
447				0.3	0.556	37
448				0.4	0.556	6

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
449	15	5	0.7	0.1	0.556	
450				0.2	0.556	1
451				0.3	0.556	9
452				0.4	0.556	51
453			0.8	0.1	0.560	
454				0.2	0.556	62
455				0.3	0.556	1
456				0.4	0.556	4
457			0.9	0.1	0.556	
458				0.2	0.556	1
459				0.3	0.556	1
460				0.4	0.556	1
461			1.0	0.1	0.556	
462				0.2	0.556	2
463				0.3	0.556	8
464				0.4	0.556	1
465		6	0.7	0.1	0.556	
466				0.2	0.556	22
467				0.3	0.556	3
468				0.4	0.556	45
469			0.8	0.1	0.556	
470				0.2	0.556	24
471				0.3	0.556	14
472				0.4	0.556	12
473			0.9	0.1	0.556	
474				0.2	0.556	12
475				0.3	0.556	26
476				0.4	0.556	1
477			1.0	0.1	0.556	
478				0.2	0.556	2
479				0.3	0.556	1
480				0.4	0.556	1

ต้นฉบับ หน้าขาดหาย

2. ปัญหา 31 งาน

ตารางที่ ง-1.3 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 31 งาน replication ที่ 1

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
1	20	1	0.7	0.1	35.556	
2				0.2	38.889	
3				0.3	35.111	
4				0.4	35.333	
5			0.8	0.1	38.667	
6				0.2	35.556	
7				0.3	40.667	
8				0.4	35.111	
9			0.9	0.1	38.667	
10				0.2	38.667	
11				0.3	35.333	
12				0.4	35.333	
13			1.0	0.1	35.333	
14				0.2	35.556	
15				0.3	35.111	
16				0.4	35.333	
17		2	0.7	0.1	35.333	
18				0.2	35.111	
19				0.3	35.111	
20				0.4	35.111	
21			0.8	0.1	36.667	
22				0.2	34.444	
23				0.3	40.889	
24				0.4	35.111	
25			0.9	0.1	35.111	
26				0.2	38.889	
27				0.3	37.111	
28				0.4	34.889	
29			1.0	0.1	36	
30				0.2	35.556	
31				0.3	35.111	
32				0.4	34.444	
33	20	3	0.7	0.1	37	
34				0.2	46.444	
35				0.3	42.222	
36				0.4	36	
37			0.8	0.1	61.111	
38				0.2	41.778	
39				0.3	38.667	
40				0.4	48.444	
41			0.9	0.1	46.222	
42				0.2	60.444	
43				0.3	43.111	
44				0.4	41.778	
45			1.0	0.1	36.667	
46				0.2	35.111	
47				0.3	42.222	
48				0.4	37.778	
49		4	0.7	0.1	35.333	
50				0.2	35.111	247
51				0.3	36	48
52				0.4	34.444	266
53			0.8	0.1	35.111	
54				0.2	35.111	114
55				0.3	35.333	101
56				0.4	35.333	20
57			0.9	0.1	35.333	
58				0.2	35.333	17
59				0.3	35.111	88
60				0.4	35.111	12
61			1.0	0.1	35.111	
62				0.2	35.333	10
63				0.3	35.333	467
64				0.4	34.444	241
65	20	5	0.7	0.1	38.667	
66				0.2	35.111	187
67				0.3	35.333	99
68				0.4	35.333	53
69			0.8	0.1	35.333	
70				0.2	34.444	32
71				0.3	34.444	109
72				0.4	35.111	71
73			0.9	0.1	40.889	
74				0.2	35.556	399
75				0.3	35.111	236
76				0.4	35.333	48
77			1.0	0.1	41.778	
78				0.2	35.333	3
79				0.3	35.333	254
80				0.4	35.111	294
81	20	6	0.7	0.1	40.667	
82				0.2	35.111	124
83				0.3	34.444	17
84				0.4	35.111	376
85			0.8	0.1	41.778	
86				0.2	35.111	21
87				0.3	35.333	31
88				0.4	44.889	266
89			0.9	0.1	35.111	
90				0.2	35.333	453
91				0.3	35.111	240
92				0.4	35.556	170
93			1.0	0.1	35.556	
94				0.2	34.444	40
95				0.3	35.111	131
96				0.4	35.111	360

ตารางที่ ง-1.3 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 31 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
97	30	1	0.7	0.1	35.556	
98				0.2	35.333	
99				0.3	35.333	
100				0.4	34.444	
101			0.8	0.1	46.667	
102				0.2	35.556	
103				0.3	36.667	
104				0.4	35.111	
105			0.9	0.1	35.556	
106				0.2	35.111	
107				0.3	35.111	
108				0.4	35.333	
109			1.0	0.1	40.444	
110				0.2	39.556	
111				0.3	35.111	
112				0.4	35.111	
113		2	0.7	0.1	35.111	
114				0.2	35.111	
115				0.3	35.333	
116				0.4	35.111	
117			0.8	0.1	37.111	
118				0.2	35.111	
119				0.3	34.444	
120				0.4	35.111	
121			0.9	0.1	41.778	
122				0.2	38.667	
123				0.3	34.444	
124				0.4	35.333	
125			1.0	0.1	37.778	
126				0.2	35.333	
127				0.3	34.444	
128				0.4	35.111	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
129	30	3	0.7	0.1	47.778	
130				0.2	40.667	
131				0.3	40.889	
132				0.4	42.889	
133			0.8	0.1	46.444	
134				0.2	48.889	
135				0.3	38.667	
136				0.4	43.111	
137			0.9	0.1	40.889	
138				0.2	35.333	
139				0.3	40.889	
140				0.4	38.667	
141			1.0	0.1	35.556	
142				0.2	36	
143				0.3	46.222	
144				0.4	35.333	
145		4	0.7	0.1	34.444	
146				0.2	34.444	
147				0.3	35.111	
148				0.4	35.111	
149			0.8	0.1	34.444	
150				0.2	35.333	
151				0.3	35.111	
152				0.4	35.333	
153			0.9	0.1	35.111	
154				0.2	35.333	
155				0.3	35.111	
156				0.4	35.111	
157			1.0	0.1	35.111	
158				0.2	35.333	
159				0.3	35.111	
160				0.4	35.333	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
161	30	5	0.7	0.1	35.111	
162				0.2	35.333	
163				0.3	34.444	
164				0.4	35.111	
165			0.8	0.1	38.222	
166				0.2	35.333	
167				0.3	34.444	
168				0.4	35.111	
169			0.9	0.1	34.444	
170				0.2	34.444	
171				0.3	35.111	
172				0.4	34.889	
173			1.0	0.1	35.111	
174				0.2	35.333	
175				0.3	35.111	
176				0.4	34.444	
177		6	0.7	0.1	35.111	
178				0.2	35.333	
179				0.3	35.111	
180				0.4	35.111	
181			0.8	0.1	35.333	
182				0.2	35.111	
183				0.3	34.444	
184				0.4	35.111	
185			0.9	0.1	35.111	
186				0.2	35.111	
187				0.3	34.444	
188				0.4	35.556	
189			1.0	0.1	35.333	
190				0.2	34.444	
191				0.3	35.111	
192				0.4	35.111	

ตารางที่ ง-1.3 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 31 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
193	40	1	0.7	0.1	35.111	
194				0.2	35.111	
195				0.3	34.444	
196				0.4	35.111	
197			0.8	0.1	35.333	
198				0.2	35.111	
199				0.3	35.111	
200				0.4	34.444	
201			0.9	0.1	35.111	
202				0.2	35.111	
203				0.3	35.111	
204				0.4	34.444	
205			1.0	0.1	35.333	
206				0.2	35.111	
207				0.3	34.444	
208				0.4	34.444	
209		2	0.7	0.1	35.111	
210				0.2	35.111	
211				0.3	34.444	
212				0.4	34.444	
213			0.8	0.1	38.222	
214				0.2	35.111	
215				0.3	34.444	
216				0.4	35.111	
217			0.9	0.1	40.889	
218				0.2	34.444	
219				0.3	34.444	
220				0.4	34.444	
221			1.0	0.1	35.333	
222				0.2	35.111	
223				0.3	34.444	
224				0.4	34.444	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
225	40	3	0.7	0.1	41.778	
226				0.2	42.222	
227				0.3	37.111	
228				0.4	40.667	
229			0.8	0.1	37.111	
230				0.2	38.667	
231				0.3	38.222	
232				0.4	36	
233			0.9	0.1	43.111	
234				0.2	35.556	
235				0.3	35.556	
236				0.4	35.111	
237			1.0	0.1	38.667	
238				0.2	35.333	
239				0.3	43.778	
240				0.4	38.666	
241		4	0.7	0.1	34.444	
242				0.2	34.444	
243				0.3	35.111	
244				0.4	35.111	
245			0.8	0.1	34.444	
246				0.2	34.444	
247				0.3	35.111	
248				0.4	34.444	
249			0.9	0.1	34.444	
250				0.2	34.444	
251				0.3	34.444	
252				0.4	35.333	
253			1.0	0.1	34.444	
254				0.2	35.111	
255				0.3	35.111	
256				0.4	34.444	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
257	40	5	0.7	0.1	35.333	
258				0.2	35.111	
259				0.3	34.444	
260				0.4	34.444	
261			0.8	0.1	37.556	
262				0.2	35.111	
263				0.3	35.111	
264				0.4	35.111	
265			0.9	0.1	35.111	
266				0.2	34.444	
267				0.3	34.444	
268				0.4	35.111	
269			1.0	0.1	35.333	
270				0.2	35.111	
271				0.3	34.444	
272				0.4	35.111	
273		6	0.7	0.1	35.556	
274				0.2	34.444	
275				0.3	34.444	
276				0.4	34.444	
277			0.8	0.1	34.444	
278				0.2	35.111	
279				0.3	35.111	
280				0.4	34.444	
281			0.9	0.1	35.111	
282				0.2	34.444	
283				0.3	34.444	
284				0.4	35.111	
285			1.0	0.1	38.667	
286				0.2	34.444	
287				0.3	34.444	
288				0.4	34.444	

ตารางที่ ง-1.4 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 31 งาน replication ที่ 2

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
289	20	1	0.7	0.1	46.444	
290				0.2	40.889	
291				0.3	38.222	
292				0.4	35.333	
293			0.8	0.1	50.444	
294				0.2	44.444	
295				0.3	40.889	
296				0.4	34.444	
297			0.9	0.1	45.556	
298				0.2	35.556	
299				0.3	35.111	
300				0.4	35.111	
301			1	0.1	40.889	
302				0.2	35.333	
303				0.3	35.111	
304				0.4	38.667	
305		2	0.7	0.1	41.778	
306				0.2	35.333	
307				0.3	35.111	
308				0.4	35.333	
309			0.8	0.1	35.556	
310				0.2	38.222	
311				0.3	35.556	
312				0.4	35.333	
313			0.9	0.1	35.333	
314				0.2	35.333	
315				0.3	34.444	
316				0.4	35.111	
317			1	0.1	58.889	
318				0.2	34.444	
319				0.3	35.556	
320				0.4	35.333	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
321	20	3	0.7	0.1	40.889	
322				0.2	40.889	
323				0.3	46.444	
324				0.4	42.889	
325			0.8	0.1	44.889	
326				0.2	39.556	
327				0.3	38.667	
328				0.4	40	
329			0.9	0.1	46.222	
330				0.2	55.111	
331				0.3	46.222	
332				0.4	38.667	
333			1	0.1	42.667	
334				0.2	42.222	
335				0.3	34.444	
336				0.4	38.667	
337		4	0.7	0.1	35.333	
338				0.2	35.333	
339				0.3	36	
340				0.4	34.444	
341			0.8	0.1	35.111	
342				0.2	35.111	
343				0.3	35.333	
344				0.4	35.333	
345			0.9	0.1	35.333	
346				0.2	35.333	
347				0.3	35.111	
348				0.4	35.111	
349			1	0.1	35.111	
350				0.2	35.333	
351				0.3	35.333	
352				0.4	34.444	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
353	20	5	0.7	0.1	37.778	
354				0.2	40.889	
355				0.3	35.111	
356				0.4	34.444	
357			0.8	0.1	38.222	
358				0.2	35.111	
359				0.3	34.444	
360				0.4	35.333	
361			0.9	0.1	34.444	
362				0.2	38.222	
363				0.3	35.333	
364				0.4	35.111	
365			1	0.1	37.111	
366				0.2	35.111	
367				0.3	35.111	
368				0.4	36	
369		6	0.7	0.1	42.889	
370				0.2	35.333	
371				0.3	37.111	
372				0.4	35.111	
373			0.8	0.1	43.778	
374				0.2	37.778	
375				0.3	34.444	
376				0.4	34.444	
377			0.9	0.1	39.556	
378				0.2	34.444	
379				0.3	35.111	
380				0.4	35.111	
381			1	0.1	40	
382				0.2	34.444	
383				0.3	35.556	
384				0.4	34.889	

ตารางที่ ง-1.4 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 31 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
385	30	1	0.7	0.1	37.778	
386				0.2	35.111	
387				0.3	35.333	
388				0.4	35.111	
389			0.8	0.1	34.889	
390				0.2	35.556	
391				0.3	35.111	
392				0.4	35.111	
393			0.9	0.1	35.444	
394				0.2	35.333	
395				0.3	35.111	
396				0.4	35.111	
397			1	0.1	35.333	
398				0.2	35.111	
399				0.3	35.111	
400				0.4	34.444	
401		2	0.7	0.1	35.556	
402				0.2	35.333	
403				0.3	35.111	
404				0.4	34.444	
405			0.8	0.1	43.111	
406				0.2	35.111	
407				0.3	34.444	
408				0.4	35.111	
409			0.9	0.1	38.667	
410				0.2	35.111	
411				0.3	34.444	
412				0.4	35.111	
413			1	0.1	40.889	
414				0.2	34.444	
415				0.3	35.111	
416				0.4	34.444	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
417	30	3	0.7	0.1	42.222	
418				0.2	40.667	
419				0.3	37.111	
420				0.4	35.333	
421			0.8	0.1	40.444	
422				0.2	40.889	
423				0.3	40.889	
424				0.4	35.333	
425			0.9	0.1	35.333	
426				0.2	35.111	
427				0.3	40.889	
428				0.4	38.222	
429			1	0.1	44.444	
430				0.2	35.333	
431				0.3	39.556	
432				0.4	35.333	
433		4	0.7	0.1	35.111	
434				0.2	34.444	
435				0.3	35.111	
436				0.4	35.333	
437			0.8	0.1	35.111	
438				0.2	35.333	
439				0.3	35.333	
440				0.4	35.333	
441			0.9	0.1	35.111	
442				0.2	35.111	
443				0.3	35.111	
444				0.4	35.111	
445			1	0.1	35.111	
446				0.2	35.111	
447				0.3	34.444	
448				0.4	35.333	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
449	30	5	0.7	0.1	35.333	
450				0.2	35.111	
451				0.3	35.333	
452				0.4	35.111	
453			0.8	0.1	40.889	
454				0.2	34.889	
455				0.3	34.444	
456				0.4	34.444	
457			0.9	0.1	41.778	
458				0.2	35.556	
459				0.3	35.111	
460				0.4	34.444	
461			1	0.1	35.111	
462				0.2	34.889	
463				0.3	34.444	
464				0.4	35.333	
465		6	0.7	0.1	35.111	
466				0.2	38.222	
467				0.3	34.444	
468				0.4	35.111	
469			0.8	0.1	37.778	
470				0.2	35.111	
471				0.3	35.111	
472				0.4	35.556	
473			0.9	0.1	35.111	
474				0.2	35.333	
475				0.3	35.111	
476				0.4	35.111	
477			1	0.1	34.444	
478				0.2	35.111	
479				0.3	35.111	
480				0.4	35.111	

ตารางที่ ง-1.4 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 31 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
481	40	1	0.7	0.1	37.778	
482				0.2	37.111	
483				0.3	35.111	
484				0.4	34.444	
485			0.8	0.1	36.667	
486				0.2	35.111	
487				0.3	34.444	
488				0.4	35.111	
489			0.9	0.1	35.556	
490				0.2	35.111	
491				0.3	34.889	
492				0.4	35.111	
493			1	0.1	38.667	
494				0.2	35.111	
495				0.3	34.444	
496				0.4	34.444	
497		2	0.7	0.1	35.333	
498				0.2	37.111	
499				0.3	35.111	
500				0.4	34.444	
501			0.8	0.1	34.444	
502				0.2	35.111	
503				0.3	35.111	
504				0.4	34.444	
505			0.9	0.1	35.333	
506				0.2	35.333	
507				0.3	35.333	
508				0.4	34.444	
509			1	0.1	35.111	
510				0.2	34.444	
511				0.3	34.444	
512				0.4	34.444	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
513	40	3	0.7	0.1	50.667	
514				0.2	37.778	
515				0.3	40.889	
516				0.4	36	
517			0.8	0.1	35.556	
518				0.2	34.444	
519				0.3	35.556	
520				0.4	36	
521			0.9	0.1	39.556	
522				0.2	34.444	
523				0.3	36	
524				0.4	39.556	
525			1	0.1	42.222	
526				0.2	40.889	
527				0.3	35.556	
528				0.4	36	
529		4	0.7	0.1	35.111	
530				0.2	34.444	328
531				0.3	35.111	129
532				0.4	35.556	19
533			0.8	0.1	34.444	
534				0.2	35.333	79
535				0.3	34.444	72
536				0.4	34.444	423
537			0.9	0.1	34.889	
538				0.2	34.444	32
539				0.3	35.333	56
540				0.4	34.444	354
541			1	0.1	35.111	
542				0.2	35.333	49
543				0.3	35.111	167
544				0.4	34.444	318

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
545	40	5	0.7	0.1	35.111	
546				0.2	35.111	17
547				0.3	35.111	256
548				0.4	34.444	244
549			0.8	0.1	35.333	
550				0.2	34.444	304
551				0.3	34.889	144
552				0.4	35.111	35
553			0.9	0.1	34.889	
554				0.2	35.111	37
555				0.3	34.444	22
556				0.4	34.444	159
557			1	0.1	35.333	
558				0.2	34.444	366
559				0.3	34.444	96
560				0.4	34.444	38
561		6	0.7	0.1	34.444	
562				0.2	35.111	227
563				0.3	34.444	83
564				0.4	34.444	219
565			0.8	0.1	35.333	
566				0.2	34.444	36
567				0.3	35.111	61
568				0.4	34.444	33
569			0.9	0.1	35.111	
570				0.2	35.111	7
571				0.3	35.111	165
572				0.4	34.444	88
573			1	0.1	35.111	
574				0.2	35.111	156
575				0.3	34.444	511
576				0.4	34.444	305

3. ปัญหา 39 งาน

ตารางที่ ง-1.5 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 39 งาน replication ที่ 1

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
1	20	1	0.7	0.1	20.245	
2				0.2	17.959	
3				0.3	6.8163	
4				0.4	2.5306	
5			0.8	0.1	15.674	
6				0.2	1.9592	
7				0.3	3.102	
8				0.4	5.9592	
9			0.9	0.1	23.674	
10				0.2	17.388	
11				0.3	1.3878	
12				0.4	4.2449	
13			1.0	0.1	20.245	
14				0.2	16.816	
15				0.3	1.3878	
16				0.4	1.3878	
17		2	0.7	0.1	31.674	
18				0.2	18.531	
19				0.3	4.2449	
20				0.4	3.102	
21			0.8	0.1	12.816	
22				0.2	12.816	
23				0.3	10.531	
24				0.4	3.6735	
25			0.9	0.1	20.245	
26				0.2	6.5306	
27				0.3	6.5306	
28				0.4	3.6735	
29			1.0	0.1	14.531	
30				0.2	2.5306	
31				0.3	3.102	
32				0.4	8.8163	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
33	20	3	0.7	0.1	5.9592	
34				0.2	23.102	
35				0.3	8.2449	
36				0.4	11.102	
37			0.8	0.1	26.245	
38				0.2	6.5306	
39				0.3	4.2449	
40				0.4	11.102	
41			0.9	0.1	10.245	
42				0.2	27.674	
43				0.3	11.674	
44				0.4	7.6735	
45			1.0	0.1	6.8163	
46				0.2	14.531	
47				0.3	9.3878	
48				0.4	9.3878	
49		4	0.7	0.1	2.5306	
50				0.2	4.5306	
51				0.3	5.3878	
52				0.4	7.9592	
53			0.8	0.1	9.9592	
54				0.2	13.959	
55				0.3	2.5306	
56				0.4	5.9592	
57			0.9	0.1	11.388	
58				0.2	4.8163	
59				0.3	7.9592	
60				0.4	2.5306	
61			1.0	0.1	7.3878	
62				0.2	5.3878	
63				0.3	6.5306	
64				0.4	3.6735	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
65	20	5	0.7	0.1	8.8163	
66				0.2	8.8163	
67				0.3	26.531	
68				0.4	2.5306	
69			0.8	0.1	23.102	
70				0.2	4.8163	
71				0.3	15.674	
72				0.4	2.5306	
73			0.9	0.1	17.388	
74				0.2	4.2449	
75				0.3	15.674	
76				0.4	3.102	
77			1.0	0.1	18.531	
78				0.2	24.816	
79				0.3	7.6735	
80				0.4	4.8163	
81		6	0.7	0.1	15.674	
82				0.2	2.5306	
83				0.3	4.2449	
84				0.4	5.3878	
85			0.8	0.1	8.2449	
86				0.2	6.5306	
87				0.3	6.2449	
88				0.4	4.2449	
89			0.9	0.1	39.102	
90				0.2	2.5306	
91				0.3	4.8163	
92				0.4	9.9592	
93			1.0	0.1	37.102	
94				0.2	3.6735	
95				0.3	2.5306	
96				0.4	8.8163	

ตารางที่ ง-1.5 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 39 งาน replication ที่ 1(ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
97	30	1	0.7	0.1	9.959	
98				0.2	11.102	
99				0.3	4.245	
100				0.4	3.102	
101			0.8	0.1	4.245	
102				0.2	3.102	
103				0.3	3.102	
104				0.4	3.673	
105			0.9	0.1	2.531	
106				0.2	3.102	
107				0.3	3.673	
108				0.4	2.531	
109			1.0	0.1	9.388	
110				0.2	2.531	
111				0.3	3.673	
112				0.4	4.816	
113		2	0.7	0.1	22.531	
114				0.2	4.245	
115				0.3	2.531	
116				0.4	5.959	
117			0.8	0.1	16.245	
118				0.2	11.102	
119				0.3	1.388	
120				0.4	4.245	
121			0.9	0.1	3.959	
122				0.2	16.816	
123				0.3	4.245	
124				0.4	2.531	
125			1.0	0.1	3.673	
126				0.2	3.102	
127				0.3	2.531	
128				0.4	4.245	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
129	30	3	0.7	0.1	22.531	
130				0.2	8.531	
131				0.3	7.959	
132				0.4	8.816	
133			0.8	0.1	8.816	
134				0.2	17.388	
135				0.3	8.245	
136				0.4	7.102	
137			0.9	0.1	25.388	
138				0.2	12.816	
139				0.3	4.245	
140				0.4	1.388	
141			1.0	0.1	3.673	
142				0.2	16.816	
143				0.3	4.245	
144				0.4	9.388	
145		4	0.7	0.1	2.531	
146				0.2	7.959	
147				0.3	6.531	75
148				0.4	3.102	224
149			0.8	0.1	7.673	
150				0.2	8.816	
151				0.3	7.959	24
152				0.4	8.245	12
153			0.9	0.1	2.531	
154				0.2	2.531	
155				0.3	7.959	325
156				0.4	4.816	344
157			1.0	0.1	1.388	
158				0.2	4.816	
159				0.3	4.245	71
160				0.4	3.673	148

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
161	30	5	0.7	0.1	31.674	
162				0.2	5.959	
163				0.3	4.816	
164				0.4	4.245	
165			0.8	0.1	18.531	
166				0.2	15.102	
167				0.3	7.673	
168				0.4	4.245	
169			0.9	0.1	23.388	
170				0.2	3.102	
171				0.3	5.959	
172				0.4	13.102	
173			1.0	0.1	10.531	
174				0.2	20.816	
175				0.3	9.388	
176				0.4	0.583	
177		6	0.7	0.1	7.673	
178				0.2	8.245	
179				0.3	1.388	
180				0.4	9.388	
181			0.8	0.1	28.245	
182				0.2	9.959	
183				0.3	8.816	
184				0.4	1.388	
185			0.9	0.1	14.531	
186				0.2	8.245	
187				0.3	2.531	
188				0.4	3.673	
189			1.0	0.1	2.531	
190				0.2	5.959	
191				0.3	5.387	
192				0.4	1.388	

ตารางที่ ง-1.5 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 39 งาน replication ที่ 1(ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
193	40	1	0.7	0.1	25.388	
194				0.2	3.102	
195				0.3	2.531	
196				0.4	5.959	
197			0.8	0.1	21.388	
198				0.2	17.959	
199				0.3	1.388	
200				0.4	4.245	
201			0.9	0.1	11.102	
202				0.2	12.816	
203				0.3	6.816	
204				0.4	3.102	
205			1.0	0.1	5.959	
206				0.2	2.531	
207				0.3	2.531	
208				0.4	7.673	
209		2	0.7	0.1	15.674	
210				0.2	2.531	
211				0.3	1.959	
212				0.4	1.388	
213			0.8	0.1	7.102	
214				0.2	2.531	
215				0.3	2.531	
216				0.4	2.531	
217			0.9	0.1	18.531	
218				0.2	1.959	
219				0.3	6.531	
220				0.4	2.531	
221			1.0	0.1	14.531	
222				0.2	2.531	
223				0.3	4.245	
224				0.4	5.388	
225	40	3	0.7	0.1	11.102	
226				0.2	5.388	
227				0.3	15.674	
228				0.4	3.102	
229			0.8	0.1	21.388	
230				0.2	5.102	
231				0.3	1.959	
232				0.4	2.531	
233			0.9	0.1	4.245	
234				0.2	5.388	
235				0.3	8.816	
236				0.4	1.388	
237			1.0	0.1	5.388	
238				0.2	4.816	
239				0.3	2.531	
240				0.4	6.816	
241		4	0.7	0.1	5.102	
242				0.2	2.531	
243				0.3	5.388	398
244				0.4	2.531	261
245			0.8	0.1	2.531	
246				0.2	9.388	
247				0.3	5.959	283
248				0.4	2.531	50
249			0.9	0.1	5.388	
250				0.2	5.388	
251				0.3	4.245	27
252				0.4	2.531	58
253			1.0	0.1	2.531	
254				0.2	1.959	
255				0.3	3.102	362
256				0.4	2.531	475
257	40	5	0.7	0.1	19.674	
258				0.2	7.102	
259				0.3	4.245	
260				0.4	4.245	
261			0.8	0.1	6.531	
262				0.2	11.102	
263				0.3	4.245	
264				0.4	5.959	
265			0.9	0.1	9.959	
266				0.2	1.388	
267				0.3	8.245	
268				0.4	8.245	
269			1.0	0.1	5.388	
270				0.2	3.102	
271				0.3	5.388	
272				0.4	4.245	
273		6	0.7	0.1	25.674	
274				0.2	4.245	
275				0.3	3.673	
276				0.4	5.959	
277			0.8	0.1	13.388	
278				0.2	2.531	
279				0.3	4.245	
280				0.4	4.245	
281			0.9	0.1	19.388	
282				0.2	5.898	
283				0.3	2.531	
284				0.4	5.388	
285			1.0	0.1	24.816	
286				0.2	10.531	
287				0.3	4.245	
288				0.4	2.531	

ตารางที่ ง-1.6 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 39 งาน replication ที่ 2

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
289	20	1	0.7	0.1	5.3878	
290				0.2	4.8163	
291				0.3	4.2449	
292				0.4	8.2449	
293			0.8	0.1	21.959	
294				0.2	21.959	
295				0.3	4.2449	
296				0.4	10.245	
297			0.9	0.1	20.816	
298				0.2	9.3878	
299				0.3	2.5306	
300				0.4	2.5306	
301			1	0.1	25.388	
302				0.2	9.3878	
303				0.3	4.2449	
304				0.4	2.5306	
305		2	0.7	0.1	5.3878	
306				0.2	6.5306	
307				0.3	6.5306	
308				0.4	2.5306	
309			0.8	0.1	8.8163	
310				0.2	4.2449	
311				0.3	4.2449	
312				0.4	5.9592	
313			0.9	0.1	16.245	
314				0.2	3.6735	
315				0.3	18.531	
316				0.4	7.102	
317			1	0.1	17.388	
318				0.2	7.6735	
319				0.3	1.3878	
320				0.4	3.102	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
321	20	3	0.7	0.1	23.674	
322				0.2	17.959	
323				0.3	25.388	
324				0.4	13.388	
325			0.8	0.1	35.674	
326				0.2	23.674	
327				0.3	9.3878	
328				0.4	12.816	
329			0.9	0.1	9.3878	
330				0.2	2.5306	
331				0.3	4.2449	
332				0.4	5.9592	
333			1	0.1	6.5306	
334				0.2	15.374	
335				0.3	11.102	
336				0.4	8.2449	
337		4	0.7	0.1	9.6735	
338				0.2	7.3878	
339				0.3	5.3878	
340				0.4	4.2449	
341			0.8	0.1	5.102	
342				0.2	2.5306	
343				0.3	4.2449	
344				0.4	12.245	
345			0.9	0.1	6.8163	
346				0.2	2.5306	
347				0.3	7.9592	
348				0.4	3.3878	
349			1	0.1	7.3878	
350				0.2	2.5306	
351				0.3	2.5306	
352				0.4	2.5306	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
353	20	5	0.7	0.1	6.8163	
354				0.2	4.8163	
355				0.3	22.531	
356				0.4	4.2449	
357			0.8	0.1	16.816	
358				0.2	20.816	
359				0.3	9.3878	
360				0.4	6.8163	
361			0.9	0.1	12.245	
362				0.2	3.6735	
363				0.3	12.245	
364				0.4	7.102	
365			1	0.1	7.102	
366				0.2	5.9592	
367				0.3	3.102	
368				0.4	10.531	
369		6	0.7	0.1	13.674	
370				0.2	2.5306	
371				0.3	3.102	
372				0.4	4.2449	
373			0.8	0.1	16.816	
374				0.2	3.102	
375				0.3	2.5306	
376				0.4	12.245	
377			0.9	0.1	3.102	
378				0.2	23.102	
379				0.3	6.5306	
380				0.4	5.9592	
381			1	0.1	16.816	
382				0.2	12.816	
383				0.3	3.3878	
384				0.4	7.6735	

ตารางที่ ง-1.6 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 39 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No
385	30	1	0.7	0.1	11.388	
386				0.2	9.388	
387				0.3	1.388	
388				0.4	5.959	
389			0.8	0.1	9.959	
390				0.2	6.531	
391				0.3	1.388	
392				0.4	3.673	
393			0.9	0.1	9.388	
394				0.2	3.102	
395				0.3	9.673	
396				0.4	3.102	
397			1	0.1	4.245	
398				0.2	9.959	
399				0.3	11.102	
400				0.4	2.531	
401		2	0.7	0.1	3.102	
402				0.2	3.673	
403				0.3	4.245	
404				0.4	3.102	
405			0.8	0.1	26.531	
406				0.2	6.531	
407				0.3	2.531	
408				0.4	7.673	
409			0.9	0.1	1.959	
410				0.2	2.531	
411				0.3	9.102	
412				0.4	2.531	
413			1	0.1	17.959	
414				0.2	5.959	
415				0.3	7.673	
416				0.4	5.959	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
417	30	3	0.7	0.1	10.531	
418				0.2	7.673	
419				0.3	4.816	
420				0.4	5.959	
421			0.8	0.1	20.816	
422				0.2	10.531	
423				0.3	7.388	
424				0.4	9.388	
425			0.9	0.1	14.531	
426				0.2	6.816	
427				0.3	5.959	
428				0.4	9.959	
429			1	0.1	20.816	
430				0.2	7.102	
431				0.3	7.102	
432				0.4	3.102	
433		4	0.7	0.1	6.816	
434				0.2	4.816	
435				0.3	7.388	287
436				0.4	7.959	88
437			0.8	0.1	7.959	
438				0.2	4.245	
439				0.3	5.959	98
440				0.4	7.959	396
441			0.9	0.1	2.531	
442				0.2	6.531	
443				0.3	6.245	121
444				0.4	7.673	241
445			1	0.1	2.531	
446				0.2	3.673	
447				0.3	4.245	375
448				0.4	2.531	348

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
449	30	5	0.7	0.1	25.959	
450				0.2	13.388	
451				0.3	8.245	
452				0.4	5.959	
453			0.8	0.1	2.531	
454				0.2	4.245	
455				0.3	4.245	
456				0.4	2.531	
457			0.9	0.1	19.674	
458				0.2	5.388	
459				0.3	5.959	
460				0.4	5.959	
461			1	0.1	19.102	
462				0.2	7.673	
463				0.3	5.959	
464				0.4	3.102	
465		6	0.7	0.1	15.102	
466				0.2	4.245	
467				0.3	6.531	
468				0.4	5.959	
469			0.8	0.1	5.388	
470				0.2	13.959	
471				0.3	9.388	
472				0.4	2.531	
473			0.9	0.1	3.673	
474				0.2	5.959	
475				0.3	5.959	
476				0.4	9.388	
477			1	0.1	5.388	
478				0.2	3.102	
479				0.3	4.245	
480				0.4	5.959	

ตารางที่ ง-1.6 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 39 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
481	40	1	0.7	0.1	21.959	
482				0.2	5.102	
483				0.3	1.388	
484				0.4	7.673	
485			0.8	0.1	11.674	
486				0.2	7.673	
487				0.3	1.388	
488				0.4	0.583	
489			0.9	0.1	4.245	
490				0.2	3.102	
491				0.3	4.245	
492				0.4	1.388	
493			1	0.1	4.245	
494				0.2	7.102	
495				0.3	4.245	
496				0.4	4.816	
497		2	0.7	0.1	5.959	
498				0.2	5.388	
499				0.3	2.531	
500				0.4	3.102	
501			0.8	0.1	27.673	
502				0.2	3.673	
503				0.3	2.531	
504				0.4	6.531	
505			0.9	0.1	5.388	
506				0.2	1.388	
507				0.3	4.245	
508				0.4	3.102	
509			1	0.1	7.102	
510				0.2	4.245	
511				0.3	1.959	
512				0.4	4.816	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
513	40	3	0.7	0.1	21.388	
514				0.2	11.674	
515				0.3	2.531	
516				0.4	11.102	
517			0.8	0.1	24.531	
518				0.2	7.673	
519				0.3	8.245	
520				0.4	11.102	
521			0.9	0.1	14.531	
522				0.2	3.102	
523				0.3	2.531	
524				0.4	4.816	
525			1	0.1	18.531	
526				0.2	17.388	
527				0.3	9.388	
528				0.4	4.245	
529		4	0.7	0.1	2.531	
530				0.2	2.531	
531				0.3	5.388	19
532				0.4	5.388	468
533			0.8	0.1	4.245	
534				0.2	3.102	
535				0.3	4.816	213
536				0.4	5.388	312
537			0.9	0.1	4.816	
538				0.2	6.245	
539				0.3	4.245	153
540				0.4	3.673	56
541			1	0.1	3.673	
542				0.2	5.388	
543				0.3	4.816	415
544				0.4	1.959	372

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
545	40	5	0.7	0.1	6.531	
546				0.2	5.959	
547				0.3	1.388	
548				0.4	4.816	
549			0.8	0.1	2.449	
550				0.2	5.959	
551				0.3	2.531	
552				0.4	4.816	
553			0.9	0.1	20.245	
554				0.2	5.388	
555				0.3	8.245	
556				0.4	3.102	
557			1	0.1	16.816	
558				0.2	9.388	
559				0.3	6.531	
560				0.4	17.388	
561		6	0.7	0.1	22.531	
562				0.2	5.388	
563				0.3	7.959	
564				0.4	4.816	
565			0.8	0.1	7.959	
566				0.2	6.531	
567				0.3	2.531	
568				0.4	5.959	
569			0.9	0.1	5.388	
570				0.2	5.388	
571				0.3	3.102	
572				0.4	2.531	
573			1	0.1	4.816	
574				0.2	6.531	
575				0.3	2.531	
576				0.4	2.531	

4. ปัญหา 54 งาน

ตารางที่ ง-1.7 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 54 งาน replication ที่ 1

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
1	20	1	0.7	0.1	109.45	
2				0.2	125.92	
3				0.3	100.6	
4				0.4	125.92	
5			0.8	0.1	109.03	
6				0.2	121.6	
7				0.3	125.3	
8				0.4	100.45	
9			0.9	0.1	129.74	
10				0.2	118.31	
11				0.3	124.38	
12				0.4	104.31	
13			1.0	0.1	114.17	
14				0.2	132.53	
15				0.3	106.17	
16				0.4	110.74	
17		2	0.7	0.1	126.1	
18				0.2	124.2	
19				0.3	96.3	
20				0.4	124.2	
21			0.8	0.1	126.5	
22				0.2	125.1	
23				0.3	100.6	
24				0.4	102.5	
25			0.9	0.1	128.2	
26				0.2	96.6	
27				0.3	112.3	
28				0.4	106.3	
29			1.0	0.1	94.5	
30				0.2	106.7	
31				0.3	92.5	
32				0.4	100.0	
33	20	3	0.7	0.1	135.9	
34				0.2	109.5	
35				0.3	124.5	
36				0.4	127.5	
37			0.8	0.1	134.4	
38				0.2	126.8	
39				0.3	110.5	
40				0.4	126.4	
41			0.9	0.1	129.5	
42				0.2	129.8	
43				0.3	102.7	
44				0.4	125.9	
45			1.0	0.1	118.7	
46				0.2	127.6	
47				0.3	107.3	
48				0.4	121.9	
49		4	0.7	0.1	77.6	
50				0.2	91.5	
51				0.3	89.1	
52				0.4	90.5	
53			0.8	0.1	83.6	
54				0.2	89.0	
55				0.3	85.9	
56				0.4	86.7	
57			0.9	0.1	129.5	
58				0.2	78.2	
59				0.3	77.6	
60				0.4	82.2	
61			1.0	0.1	85.9	
62				0.2	87.5	
63				0.3	86.1	
64				0.4	88.2	
65	20	5	0.7	0.1	96.5	
66				0.2	96.6	
67				0.3	107.6	
68				0.4	108.9	
69			0.8	0.1	101.5	
70				0.2	125.9	
71				0.3	112.2	
72				0.4	104.2	
73			0.9	0.1	101.5	
74				0.2	109.9	
75				0.3	97.9	
76				0.4	105.7	
77			1.0	0.1	100.0	
78				0.2	115.2	
79				0.3	93.6	
80				0.4	102.0	
81		6	0.7	0.1	125.3	
82				0.2	116.2	
83				0.3	106.3	
84				0.4	104.9	
85			0.8	0.1	125.6	
86				0.2	125.5	
87				0.3	104.3	
88				0.4	98.7	
89			0.9	0.1	124.5	
90				0.2	127.8	
91				0.3	104.2	
92				0.4	97.2	
93			1.0	0.1	125.0	
94				0.2	101.9	
95				0.3	119.5	
96				0.4	121.9	

ตารางที่ ง-1.7 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 54 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
97	40	1	0.7	0.1	136.5	
98				0.2	105.7	
99				0.3	115.3	
100				0.4	121.5	
101			0.8	0.1	127.6	
102				0.2	116.5	
103				0.3	109.2	
104				0.4	99.3	
105			0.9	0.1	116.5	
106				0.2	114.2	
107				0.3	91.3	
108				0.4	107.7	
109			1.0	0.1	97.7	
110				0.2	104.9	
111				0.3	95.3	
112				0.4	100.6	
113		2	0.7	0.1	100.6	
114				0.2	97.5	
115				0.3	98.2	
116				0.4	115.3	
117			0.8	0.1	131.5	
118				0.2	90.7	
119				0.3	124.5	
120				0.4	93.3	
121			0.9	0.1	99.3	
122				0.2	124.8	
123				0.3	99.2	
124				0.4	104.6	
125			1.0	0.1	100.6	
126				0.2	122.5	
127				0.3	118.7	
128				0.4	95.0	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
129	40	3	0.7	0.1	129.8	
130				0.2	106.5	
131				0.3	125.9	
132				0.4	110.6	
133			0.8	0.1	117.2	
134				0.2	119.3	
135				0.3	114.3	
136				0.4	101.5	
137			0.9	0.1	128.5	
138				0.2	102.9	
139				0.3	126.2	
140				0.4	96.3	
141			1.0	0.1	125.9	
142				0.2	102.3	
143				0.3	107.3	
144				0.4	106.6	
145		4	0.7	0.1	78.7	
146				0.2	80.8	
147				0.3	79.8	
148				0.4	86.1	
149			0.8	0.1	83.6	
150				0.2	83.8	
151				0.3	82.5	
152				0.4	88.2	
153			0.9	0.1	76.5	
154				0.2	82.7	
155				0.3	83.3	
156				0.4	85.5	
157			1.0	0.1	75.6	
158				0.2	85.9	
159				0.3	87.6	
160				0.4	83.9	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
161	40	5	0.7	0.1	99.9	
162				0.2	98.6	
163				0.3	93.3	
164				0.4	107.0	
165			0.8	0.1	127.5	
166				0.2	102.3	
167				0.3	122.5	
168				0.4	101.5	
169			0.9	0.1	124.1	
170				0.2	91.0	
171				0.3	90.7	
172				0.4	117.3	
173			1.0	0.1	103.7	
174				0.2	121.2	
175				0.3	103.5	
176				0.4	105.2	
177		6	0.7	0.1	125.0	
178				0.2	112.2	
179				0.3	102.5	
180				0.4	107.9	
181			0.8	0.1	93.5	
182				0.2	95.0	
183				0.3	100.5	
184				0.4	107.3	
185			0.9	0.1	98.7	
186				0.2	121.5	
187				0.3	100.6	
188				0.4	99.0	
189			1.0	0.1	124.2	
190				0.2	101.2	
191				0.3	106.7	
192				0.4	100.9	

ตารางที่ ง-1.7 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 54 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
193	60	1	0.7	0.1	118.0	
194				0.2	95.0	
195				0.3	90.5	
196				0.4	96.9	
197			0.8	0.1	108.2	
198				0.2	96.6	
199				0.3	114.2	
200				0.4	106.2	
201			0.9	0.1	106.6	
202				0.2	113.5	
203				0.3	97.0	
204				0.4	118.7	
205			1.0	0.1	107.9	
206				0.2	110.7	
207				0.3	107.9	
208				0.4	108.6	
209		2	0.7	0.1	116.9	
210				0.2	106.0	
211				0.3	103.6	
212				0.4	90.3	
213			0.8	0.1	95.9	
214				0.2	115.0	
215				0.3	95.6	
216				0.4	103.7	
217			0.9	0.1	126.2	
218				0.2	124.8	
219				0.3	97.7	
220				0.4	93.9	
221			1.0	0.1	95.5	
222				0.2	107.7	
223				0.3	98.2	
224				0.4	105.7	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
225	60	3	0.7	0.1	128.2	
226				0.2	101.5	
227				0.3	106.0	
228				0.4	104.0	
229			0.8	0.1	122.6	
230				0.2	110.0	
231				0.3	110.6	
232				0.4	100.7	
233			0.9	0.1	102.7	
234				0.2	106.6	
235				0.3	103.7	
236				0.4	107.0	
237			1.0	0.1	126.2	
238				0.2	112.2	
239				0.3	113.0	
240				0.4	106.5	
241		4	0.7	0.1	80.8	
242				0.2	87.8	
243				0.3	77.9	649
244				0.4	85.9	274
245			0.8	0.1	85.9	
246				0.2	83.5	
247				0.3	86.7	
248				0.4	81.3	
249			0.9	0.1	77.3	
250				0.2	82.1	
251				0.3	83.3	103
252				0.4	85.9	958
253			1.0	0.1	77.9	
254				0.2	76.4	
255				0.3	80.8	269
256				0.4	87.5	755

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
257	60	5	0.7	0.1	123.0	
258				0.2	94.5	
259				0.3	95.0	
260				0.4	106.6	
261			0.8	0.1	125.1	
262				0.2	115.9	
263				0.3	102.9	
264				0.4	102.9	
265			0.9	0.1	101.5	
266				0.2	97.6	
267				0.3	93.6	
268				0.4	98.2	
269			1.0	0.1	103.3	
270				0.2	92.5	
271				0.3	96.3	
272				0.4	97.9	
273		6	0.7	0.1	91.6	
274				0.2	89.3	
275				0.3	101.3	
276				0.4	106.7	
277			0.8	0.1	100.3	
278				0.2	103.3	
279				0.3	92.7	
280				0.4	89.3	
281			0.9	0.1	102.2	
282				0.2	97.2	
283				0.3	98.7	
284				0.4	102.2	
285			1.0	0.1	94.0	
286				0.2	97.6	
287				0.3	103.2	
288				0.4	100.3	

ตารางที่ ง-1.8 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 54 งาน replication ที่ 2

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
289	20	1	0.7	0.1	824	
290				0.2	79	
291				0.3	310	
292				0.4	982	
293			0.8	0.1	596	
294				0.2	852	
295				0.3	909	
296				0.4	367	
297			0.9	0.1	870	
298				0.2	73	
299				0.3	694	
300				0.4	136	
301			1	0.1	150	
302				0.2	212	
303				0.3	947	
304				0.4	945	
305		2	0.7	0.1	109	
306				0.2	892	
307				0.3	669	
308				0.4	46	
309			0.8	0.1	338	
310				0.2	234	
311				0.3	768	
312				0.4	530	
313			0.9	0.1	527	
314				0.2	861	
315				0.3	722	
316				0.4	443	
317			1	0.1	205	
318				0.2	161	
319				0.3	868	
320				0.4	408	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
321	20	3	0.7	0.1	657	
322				0.2	480	
323				0.3	86	
324				0.4	50	
325			0.8	0.1	488	
326				0.2	564	
327				0.3	859	
328				0.4	910	
329			0.9	0.1	810	
330				0.2	112	
331				0.3	824	
332				0.4	288	
333			1	0.1	911	
334				0.2	881	
335				0.3	167	
336				0.4	172	
337		4	0.7	0.1	158	
338				0.2	450	
339				0.3	487	
340				0.4	177	
341			0.8	0.1	800	
342				0.2	482	
343				0.3	913	
344				0.4	246	
345			0.9	0.1	810	
346				0.2	890	
347				0.3	565	
348				0.4	491	
349			1	0.1	273	
350				0.2	929	
351				0.3	542	
352				0.4	294	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
353	20	5	0.7	0.1	908	
354				0.2	691	
355				0.3	567	
356				0.4	235	
357			0.8	0.1	236	
358				0.2	872	
359				0.3	328	
360				0.4	267	
361			0.9	0.1	177	
362				0.2	958	
363				0.3	327	
364				0.4	698	
365			1	0.1	16	
366				0.2	502	
367				0.3	419	
368				0.4	621	
369		6	0.7	0.1	583	
370				0.2	905	
371				0.3	597	
372				0.4	994	
373			0.8	0.1	8	
374				0.2	977	
375				0.3	229	
376				0.4	879	
377			0.9	0.1	400	
378				0.2	812	
379				0.3	573	
380				0.4	936	
381			1	0.1	946	
382				0.2	670	
383				0.3	66	
384				0.4	459	

ตารางที่ ง-1.8 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 54 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
385	40	1	0.7	0.1	3	
386				0.2	765	
387				0.3	49	
388				0.4	881	
389			0.8	0.1	741	
390				0.2	920	
391				0.3	871	
392				0.4	837	
393			0.9	0.1	576	
394				0.2	740	
395				0.3	954	
396				0.4	901	
397			1	0.1	959	
398				0.2	124	
399				0.3	639	
400				0.4	925	
401		2	0.7	0.1	681	
402				0.2	937	
403				0.3	760	
404				0.4	82	
405			0.8	0.1	752	
406				0.2	378	
407				0.3	764	
408				0.4	257	
409			0.9	0.1	791	
410				0.2	890	
411				0.3	427	
412				0.4	982	
413			1	0.1	609	
414				0.2	540	
415				0.3	511	
416				0.4	684	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
417	40	3	0.7	0.1	390	
418				0.2	205	
419				0.3	771	
420				0.4	424	
421			0.8	0.1	796	
422				0.2	764	
423				0.3	876	
424				0.4	123	
425			0.9	0.1	756	
426				0.2	890	
427				0.3	290	
428				0.4	486	
429			1	0.1	585	
430				0.2	324	
431				0.3	241	
432				0.4	824	
433		4	0.7	0.1	471	
434				0.2	63	
435				0.3	936	
436				0.4	277	
437			0.8	0.1	942	
438				0.2	219	
439				0.3	410	
440				0.4	894	
441			0.9	0.1	874	
442				0.2	822	
443				0.3	519	
444				0.4	457	
445			1	0.1	579	
446				0.2	34	
447				0.3	976	
448				0.4	503	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
449	40	5	0.7	0.1	24	
450				0.2	866	
451				0.3	731	
452				0.4	543	
453			0.8	0.1	968	
454				0.2	572	
455				0.3	983	
456				0.4	972	
457			0.9	0.1	24	
458				0.2	935	
459				0.3	826	
460				0.4	621	
461			1	0.1	854	
462				0.2	911	
463				0.3	909	
464				0.4	220	
465		6	0.7	0.1	9	
466				0.2	822	
467				0.3	75	
468				0.4	995	
469			0.8	0.1	865	
470				0.2	761	
471				0.3	102	
472				0.4	387	
473			0.9	0.1	197	
474				0.2	95	
475				0.3	863	
476				0.4	98	
477			1	0.1	244	
478				0.2	871	
479				0.3	719	
480				0.4	966	

ตารางที่ ง-1.8 ผลที่ได้จากการทดลองใช้ GAs หาคำตอบของปัญหา 54 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
481	60	1	0.7	0.1	668	
482				0.2	491	
483				0.3	176	
484				0.4	954	
485			0.8	0.1	630	
486				0.2	658	
487				0.3	895	
488				0.4	593	
489			0.9	0.1	937	
490				0.2	929	
491				0.3	885	
492				0.4	522	
493			1	0.1	477	
494				0.2	732	
495				0.3	993	
496				0.4	119	
497		2	0.7	0.1	839	
498				0.2	765	
499				0.3	961	
500				0.4	455	
501			0.8	0.1	349	
502				0.2	801	
503				0.3	859	
504				0.4	291	
505			0.9	0.1	860	
506				0.2	994	
507				0.3	980	
508				0.4	261	
509			1	0.1	32	
510				0.2	418	
511				0.3	561	
512				0.4	231	

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
513	60	3	0.7	0.1	720	
514				0.2	624	
515				0.3	600	
516				0.4	788	
517			0.8	0.1	218	
518				0.2	225	
519				0.3	336	
520				0.4	588	
521			0.9	0.1	173	
522				0.2	145	
523				0.3	694	
524				0.4	96	
525			1	0.1	992	
526				0.2	169	
527				0.3	670	
528				0.4	474	
529		4	0.7	0.1	971	
530				0.2	777	
531				0.3	143	143
532				0.4	296	296
533			0.8	0.1	953	
534				0.2	747	
535				0.3	725	
536				0.4	858	
537			0.9	0.1	286	
538				0.2	587	
539				0.3	420	420
540				0.4	963	963
541			1	0.1	239	
542				0.2	502	
543				0.3	105	105
544				0.4	555	555

No.	Pop size	X type	Pc	Pm	wv	Gen. No.
545	60	5	0.7	0.1	421	
546				0.2	785	
547				0.3	47	
548				0.4	278	
549			0.8	0.1	678	
550				0.2	822	
551				0.3	867	
552				0.4	163	
553			0.9	0.1	454	
554				0.2	243	
555				0.3	751	
556				0.4	854	
557			1	0.1	650	
558				0.2	956	
559				0.3	278	
560				0.4	813	
561		6	0.7	0.1	503	
562				0.2	798	
563				0.3	803	
564				0.4	79	
565			0.8	0.1	51	
566				0.2	715	
567				0.3	619	
568				0.4	684	
569			0.9	0.1	783	
570				0.2	785	
571				0.3	953	
572				0.4	396	
573			1	0.1	57	
574				0.2	958	
575				0.3	279	
576				0.4	655	

ภาคผนวก ง-2

ข้อมูลการทดลองเพื่อทดสอบค่าพารามิเตอร์ของ MOGA

1. ปัญหาขนาด 11 งาน

ตารางที่ ง-2.1 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 1

No	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
1	10	1	0.7	0.1	6	9	0.5556	579.599	
2				0.2	6	9	0.5556	579.599	
3				0.3	6	9	0.5556	579.599	
4				0.4	6	9	0.5556	579.599	
5			0.8	0.1	6	9	0.8889	535.055	
6				0.2	6	9	0.5556	579.599	
7				0.3	6	9	0.5556	579.599	
8				0.4	6	9	0.5556	579.599	
9			0.9	0.1	6	9	0.5556	579.599	
10				0.2	6	9	0.5556	579.599	
11				0.3	6	9	0.5556	579.599	
12				0.4	6	9	0.5556	579.599	
13			1	0.1	6	9	0.8889	535.055	
14				0.2	6	9	0.5556	579.599	
15				0.3	6	9	0.5556	579.599	
16				0.4	6	9	0.5556	579.599	

No	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
17	10	2	0.7	0.1	6	9	0.5556	579.599	
18				0.2	6	9	0.5556	579.599	
19				0.3	6	9	0.5556	579.599	
20				0.4	6	9	0.5556	579.599	
21			0.8	0.1	6	9	0.5556	579.599	
22				0.2	6	9	0.5556	579.599	
23				0.3	6	9	0.5556	579.599	
24				0.4	6	9	0.5556	579.599	
25			0.9	0.1	6	9	0.5556	579.599	
26				0.2	6	9	0.5556	579.599	
27				0.3	6	9	0.5556	579.599	
28				0.4	6	9	0.5556	579.599	
29			1	0.1	6	9	0.5556	579.599	
30				0.2	6	9	0.5556	579.599	
31				0.3	6	9	0.5556	579.599	
32				0.4	6	9	0.5556	579.599	

No	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
33	10	3	0.7	0.1	6	9	0.8889	535.055	
34				0.2	6	9	0.8889	535.055	
35				0.3	6	9	0.5556	579.599	
36				0.4	6	9	0.5556	579.599	
37			0.8	0.1	6	9	0.8889	535.055	
38				0.2	6	9	0.5556	579.599	
39				0.3	6	9	0.5556	579.599	
40				0.4	6	9	0.5556	579.599	
41			0.9	0.1	6	9	0.8889	535.055	
42				0.2	6	9	0.5556	579.599	
43				0.3	6	9	0.5556	579.599	
44				0.4	6	9	0.5556	579.599	
45			1	0.1	6	9	0.8889	535.055	
46				0.2	6	9	0.5556	579.599	
47				0.3	6	9	0.5556	579.599	
48				0.4	6	9	0.5556	579.599	

ตารางที่ ง-2.1 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
49	10	4	0.7	0.1	6	9	0.5556	579.599	
50				0.2	6	9	0.5556	579.599	
51				0.3	6	9	0.5556	579.599	
52				0.4	6	9	0.5556	579.599	
53			0.8	0.1	6	9	0.5556	579.599	
54				0.2	6	9	0.5556	579.599	
55				0.3	6	9	0.5556	579.599	
56				0.4	6	9	0.5556	579.599	
57			0.9	0.1	6	9	0.5556	579.599	
58				0.2	6	9	0.5556	579.599	
59				0.3	6	9	0.5556	579.599	
60				0.4	6	9	0.5556	579.599	
61			1	0.1	6	9	0.5556	579.599	
62				0.2	6	9	0.5556	579.599	
63				0.3	6	9	0.5556	579.599	
64				0.4	6	9	0.5556	579.599	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
65	10	5	0.7	0.1	6	9	0.8889	535.055	
66				0.2	6	9	0.5556	579.599	
67				0.3	6	9	0.5556	579.599	
68				0.4	6	9	0.5556	579.599	
69			0.8	0.1	6	9	0.5556	579.599	
70				0.2	6	9	0.5556	579.599	
71				0.3	6	9	0.5556	579.599	
72				0.4	6	9	0.5556	579.599	
73			0.9	0.1	6	9	0.5556	579.599	
74				0.2	6	9	0.5556	579.599	
75				0.3	6	9	0.5556	579.599	
76				0.4	6	9	0.5556	579.599	
77			1	0.1	6	9	0.5556	579.599	
78				0.2	6	9	0.5556	579.599	
79				0.3	6	9	0.5556	579.599	
80				0.4	6	9	0.5556	579.599	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
81	10	6	0.7	0.1	6	9	0.5556	579.599	
82				0.2	6	9	0.5556	579.599	
83				0.3	6	9	0.5556	579.599	
84				0.4	6	9	0.5556	579.599	
85			0.8	0.1	6	9	0.8889	535.055	
86				0.2	6	9	0.5556	579.599	
87				0.3	6	9	0.5556	579.599	
88				0.4	6	9	0.5556	579.599	
89			0.9	0.1	6	9	0.5556	579.599	
90				0.2	6	9	0.5556	579.599	
91				0.3	6	9	0.5556	579.599	
92				0.4	6	9	0.5556	579.599	
93			1	0.1	6	9	0.8889	535.055	
94				0.2	6	9	0.5556	579.599	
95				0.3	6	9	0.5556	579.599	
96				0.4	6	9	0.5556	579.599	

ตารางที่ ง-2.1 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
97	15	1	0.7	0.1	6	9	0.5556	579.599	
98				0.2	6	9	0.5556	579.599	
99				0.3	6	9	0.5556	579.599	
100				0.4	6	9	0.5556	579.599	
101			0.8	0.1	6	9	0.8889	535.055	
102				0.2	6	9	0.5556	579.599	
103				0.3	6	9	0.5556	579.599	
104				0.4	6	9	0.5556	579.599	
105			0.9	0.1	6	9	0.5556	579.599	
106				0.2	6	9	0.5556	579.599	
107				0.3	6	9	0.5556	579.599	
108				0.4	6	9	0.5556	579.599	
109			1	0.1	6	9	0.5556	579.599	
110				0.2	6	9	0.5556	579.599	
111				0.3	6	9	0.5556	579.599	
112				0.4	6	9	0.5556	579.599	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
113	15	2	0.7	0.1	6	9	0.5556	579.599	
114				0.2	6	9	0.5556	579.599	65
115				0.3	6	9	0.5556	579.599	85
116				0.4	6	9	0.5556	579.599	8
117			0.8	0.1	6	9	0.5556	579.599	
118				0.2	6	9	0.5556	579.599	11
119				0.3	6	9	0.5556	579.599	4
120				0.4	6	9	0.5556	579.599	16
121			0.9	0.1	6	9	0.5556	579.599	
122				0.2	6	9	0.5556	579.599	14
123				0.3	6	9	0.5556	579.599	55
124				0.4	6	9	0.5556	579.599	39
125			1	0.1	6	9	0.5556	579.599	
126				0.2	6	9	0.5556	579.599	47
127				0.3	6	9	0.5556	579.599	28
128				0.4	6	9	0.5556	579.599	78

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
129	15	3	0.7	0.1	6	9	0.8889	535.055	
130				0.2	6	9	0.5556	579.599	
131				0.3	6	9	0.5556	579.599	
132				0.4	6	9	0.5556	579.599	
133			0.8	0.1	6	9	0.5556	579.599	
134				0.2	6	9	0.5556	579.599	
135				0.3	6	9	0.5556	579.599	
136				0.4	6	9	0.5556	579.599	
137			0.9	0.1	6	9	0.5556	579.599	
138				0.2	6	9	0.5556	579.599	
139				0.3	6	9	0.5556	579.599	
140				0.4	6	9	0.5556	579.599	
141			1	0.1	6	9	0.5556	579.599	
142				0.2	6	9	0.8889	535.055	
143				0.3	6	9	0.5556	579.599	
144				0.4	6	9	0.5556	579.599	

ตารางที่ ง-2.1 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
145	15	4	0.7	0.1	6	9	0.5556	579.599	
146				0.2	6	9	0.5556	579.599	9
147				0.3	6	9	0.5556	579.599	65
148				0.4	6	9	0.5556	579.599	25
149			0.8	0.1	6	9	0.5556	579.599	
150				0.2	6	9	0.5556	579.599	46
151				0.3	6	9	0.5556	579.599	16
152				0.4	6	9	0.5556	579.599	35
153			0.9	0.1	6	9	0.5556	579.599	
154				0.2	6	9	0.5556	579.599	24
155				0.3	6	9	0.5556	579.599	18
156				0.4	6	9	0.5556	579.599	15
157			1	0.1	6	9	0.5556	579.599	
158				0.2	6	9	0.5556	579.599	26
159				0.3	6	9	0.5556	579.599	14
160				0.4	6	9	0.5556	579.599	44

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
161	15	5	0.7	0.1	6	9	0.5556	579.599	
162				0.2	6	9	0.5556	579.599	54
163				0.3	6	9	0.5556	579.599	41
164				0.4	6	9	0.5556	579.599	36
165			0.8	0.1	6	9	0.5556	579.599	
166				0.2	6	9	0.5556	579.599	15
167				0.3	6	9	0.5556	579.599	45
168				0.4	6	9	0.5556	579.599	15
169			0.9	0.1	6	9	0.5556	579.599	
170				0.2	6	9	0.5556	579.599	38
171				0.3	6	9	0.5556	579.599	25
172				0.4	6	9	0.5556	579.599	84
173			1	0.1	6	9	0.5556	579.599	
174				0.2	6	9	0.5556	579.599	44
175				0.3	6	9	0.5556	579.599	34
176				0.4	6	9	0.5556	579.599	12

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
177	15	6	0.7	0.1	6	9	0.5556	579.599	
178				0.2	6	9	0.5556	579.599	92
179				0.3	6	9	0.5556	579.599	45
180				0.4	6	9	0.5556	579.599	24
181			0.8	0.1	6	9	0.5556	579.599	
182				0.2	6	9	0.5556	579.599	27
183				0.3	6	9	0.5556	579.599	14
184				0.4	6	9	0.5556	579.599	19
185			0.9	0.1	6	9	0.5556	579.599	
186				0.2	6	9	0.5556	579.599	13
187				0.3	6	9	0.5556	579.599	54
188				0.4	6	9	0.5556	579.599	22
189			1	0.1	6	9	0.5556	579.599	
190				0.2	6	9	0.5556	579.599	45
191				0.3	6	9	0.5556	579.599	48
192				0.4	6	9	0.5556	579.599	33

ตารางที่ ง-2.1 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
193	20	1	0.7	0.1	6	9	0.5556	579.599	
194				0.2	6	9	0.5556	579.599	
195				0.3	6	9	0.5556	579.599	
196				0.4	6	9	0.5556	579.599	
197			0.8	0.1	6	9	0.5556	579.599	
198				0.2	6	9	0.5556	579.599	
199				0.3	6	9	0.5556	579.599	
200				0.4	6	9	0.5556	579.599	
201			0.9	0.1	6	9	0.5556	579.599	
202				0.2	6	9	0.5556	579.599	
203				0.3	6	9	0.5556	579.599	
204				0.4	6	9	0.5556	579.599	
205			1	0.1	6	9	0.5556	579.599	
206				0.2	6	9	0.5556	579.599	
207				0.3	6	9	0.5556	579.599	
208				0.4	6	9	0.5556	579.599	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
209	20	2	0.7	0.1	6	9	0.5556	579.599	
210				0.2	6	9	0.5556	579.599	104
211				0.3	6	9	0.5556	579.599	65
212				0.4	6	9	0.5556	579.599	74
213			0.8	0.1	6	9	0.5556	579.599	
214				0.2	6	9	0.5556	579.599	38
215				0.3	6	9	0.5556	579.599	74
216				0.4	6	9	0.5556	579.599	64
217			0.9	0.1	6	9	0.5556	579.599	
218				0.2	6	9	0.5556	579.599	54
219				0.3	6	9	0.5556	579.599	29
220				0.4	6	9	0.5556	579.599	6
221			1	0.1	6	9	0.5556	579.599	
222				0.2	6	9	0.5556	579.599	22
223				0.3	6	9	0.5556	579.599	44
224				0.4	6	9	0.5556	579.599	25

No.	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
225	20	3	0.7	0.1	6	9	0.8889	535.055	
226				0.2	6	9	0.5556	579.599	
227				0.3	6	9	0.5556	579.599	
228				0.4	6	9	0.5556	579.599	
229			0.8	0.1	6	9	0.5556	579.599	
230				0.2	6	9	0.5556	579.599	
231				0.3	6	9	0.5556	579.599	
232				0.4	6	9	0.5556	579.599	
233			0.9	0.1	6	9	0.8889	535.055	
234				0.2	6	9	0.5556	579.599	
235				0.3	6	9	0.5556	579.599	
236				0.4	6	9	0.5556	579.599	
237			1	0.1	6	9	0.5556	579.599	
238				0.2	6	9	0.5556	579.599	
239				0.3	6	9	0.5556	579.599	
240				0.4	6	9	0.5556	579.599	

ตารางที่ ง-2.1 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
241	20	4	0.7	0.1	6	9	0.5556	579.599	
242				0.2	6	9	0.5556	579.599	25
243				0.3	6	9	0.5556	579.599	18
244				0.4	6	9	0.5556	579.599	42
245			0.8	0.1	6	9	0.5556	579.599	
246				0.2	6	9	0.5556	579.599	21
247				0.3	6	9	0.5556	579.599	45
248				0.4	6	9	0.5556	579.599	15
249			0.9	0.1	6	9	0.5556	579.599	
250				0.2	6	9	0.5556	579.599	54
251				0.3	6	9	0.5556	579.599	74
252				0.4	6	9	0.5556	579.599	45
253			1	0.1	6	9	0.5556	579.599	
254				0.2	6	9	0.5556	579.599	62
255				0.3	6	9	0.5556	579.599	42
256				0.4	6	9	0.5556	579.599	12

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
257	20	5	0.7	0.1	6	9	0.5556	579.599	
258				0.2	6	9	0.5556	579.599	55
259				0.3	6	9	0.5556	579.599	11
260				0.4	6	9	0.5556	579.599	55
261			0.8	0.1	6	9	0.5556	579.599	
262				0.2	6	9	0.5556	579.599	3
263				0.3	6	9	0.5556	579.599	21
264				0.4	6	9	0.5556	579.599	45
265			0.9	0.1	6	9	0.5556	579.599	
266				0.2	6	9	0.5556	579.599	65
267				0.3	6	9	0.5556	579.599	31
268				0.4	6	9	0.5556	579.599	45
269			1	0.1	6	9	0.5556	579.599	
270				0.2	6	9	0.5556	579.599	54
271				0.3	6	9	0.5556	579.599	77
272				0.4	6	9	0.5556	579.599	34

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No
					n	ct	wv	fitness	
273	20	6	0.7	0.1	6	9	0.5556	579.599	
274				0.2	6	9	0.5556	579.599	114
275				0.3	6	9	0.5556	579.599	55
276				0.4	6	9	0.5556	579.599	11
277			0.8	0.1	6	9	0.5556	579.599	
278				0.2	6	9	0.5556	579.599	29
279				0.3	6	9	0.5556	579.599	22
280				0.4	6	9	0.5556	579.599	15
281			0.9	0.1	6	9	0.5556	579.599	
282				0.2	6	9	0.5556	579.599	54
283				0.3	6	9	0.5556	579.599	32
284				0.4	6	9	0.5556	579.599	46
285			1	0.1	6	9	0.5556	579.599	
286				0.2	6	9	0.5556	579.599	89
287				0.3	6	9	0.5556	579.599	12
288				0.4	6	9	0.5556	579.599	88

ตารางที่ ง-2.2 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 2

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
289	10	1	0.7	0.1	6	9	0.8889	534.530	
290				0.2	6	9	0.8889	534.530	
291				0.3	6	9	0.5556	578.760	
292				0.4	6	9	0.5556	578.760	
293			0.8	0.1	6	9	0.8889	534.530	
294				0.2	6	9	0.5556	578.760	
295				0.3	6	9	0.5556	578.760	
296				0.4	6	9	0.5556	578.760	
297			0.9	0.1	6	9	0.5556	578.760	
298				0.2	6	9	0.5556	578.760	
299				0.3	6	9	0.5556	578.760	
300				0.4	6	9	0.5556	578.760	
301			1	0.1	6	9	0.8889	534.530	
302				0.2	6	9	0.5556	578.760	
303				0.3	6	9	0.5556	578.760	
304				0.4	6	9	0.5556	578.760	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
305	10	2	0.7	0.1	6	9	0.5556	578.760	
306				0.2	6	9	0.5556	578.760	
307				0.3	6	9	0.5556	578.760	
308				0.4	6	9	0.5556	578.760	
309			0.8	0.1	6	9	0.5556	578.760	
310				0.2	6	9	0.5556	578.760	
311				0.3	6	9	0.5556	578.760	
312				0.4	6	9	0.5556	578.760	
313			0.9	0.1	6	9	0.5556	578.760	
314				0.2	6	9	0.5556	578.760	
315				0.3	6	9	0.5556	578.760	
316				0.4	6	9	0.5556	578.760	
317			1	0.1	6	9	0.5556	578.760	
318				0.2	6	9	0.5556	578.760	
319				0.3	6	9	0.5556	578.760	
320				0.4	6	9	0.5556	578.760	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
321	10	3	0.7	0.1	6	9	0.5556	578.760	
322				0.2	6	9	0.5556	578.760	
323				0.3	6	9	0.5556	578.760	
324				0.4	6	9	0.5556	578.760	
325			0.8	0.1	6	9	0.5556	578.760	
326				0.2	6	9	0.5556	578.760	
327				0.3	6	9	0.5556	578.760	
328				0.4	6	9	0.5556	578.760	
329			0.9	0.1	6	9	0.8889	534.530	
330				0.2	6	9	0.5556	578.760	
331				0.3	6	9	0.5556	578.760	
332				0.4	6	9	0.5556	578.760	
333			1	0.1	6	9	0.8889	534.530	
334				0.2	6	9	0.5556	578.760	
335				0.3	6	9	0.5556	578.760	
336				0.4	6	9	0.5556	578.760	

ตารางที่ ง-2.2 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
337	10	4	0.7	0.1	6	9	0.5556	578.760	
338				0.2	6	9	0.5556	578.760	
339				0.3	6	9	0.5556	578.760	
340				0.4	6	9	0.5556	578.760	
341			0.8	0.1	6	9	0.5556	578.760	
342				0.2	6	9	0.5556	578.760	
343				0.3	6	9	0.5556	578.760	
344				0.4	6	9	0.5556	578.760	
345			0.9	0.1	6	9	0.5556	578.760	
346				0.2	6	9	0.5556	578.760	
347				0.3	6	9	0.5556	578.760	
348				0.4	6	9	0.5556	578.760	
349			1	0.1	6	9	0.5556	578.760	
350				0.2	6	9	0.5556	578.760	
351				0.3	6	9	0.5556	578.760	
352				0.4	6	9	0.5556	578.760	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
353	10	5	0.7	0.1	6	9	0.5556	578.760	
354				0.2	6	9	0.5556	578.760	
355				0.3	6	9	0.5556	578.760	
356				0.4	6	9	0.5556	578.760	
357			0.8	0.1	6	9	0.5556	578.760	
358				0.2	6	9	0.5556	578.760	
359				0.3	6	9	0.5556	578.760	
360				0.4	6	9	0.5556	578.760	
361			0.9	0.1	6	9	0.5556	578.760	
362				0.2	6	9	0.5556	578.760	
363				0.3	6	9	0.5556	578.760	
364				0.4	6	9	0.5556	578.760	
365			1	0.1	6	9	0.5556	578.760	
366				0.2	6	9	0.5556	578.760	
367				0.3	6	9	0.5556	578.760	
368				0.4	6	9	0.5556	578.760	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
369	10	6	0.7	0.1	6	9	0.5556	578.760	
370				0.2	6	9	0.5556	578.760	
371				0.3	6	9	0.5556	578.760	
372				0.4	6	9	0.5556	578.760	
373			0.8	0.1	6	9	0.5556	578.760	
374				0.2	6	9	0.5556	578.760	
375				0.3	6	9	0.5556	578.760	
376				0.4	6	9	0.5556	578.760	
377			0.9	0.1	6	9	0.8889	534.530	
378				0.2	6	9	0.5556	578.760	
379				0.3	6	9	0.5556	578.760	
380				0.4	6	9	0.5556	578.760	
381			1	0.1	6	9	0.8889	534.530	
382				0.2	6	9	0.5556	578.760	
383				0.3	6	9	0.5556	578.760	
384				0.4	6	9	0.5556	578.760	

ตารางที่ ง-2.2 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
385	15	1	0.7	0.1	6	9	0.8889	534.530	
386				0.2	6	9	0.5556	578.760	
387				0.3	6	9	0.5556	578.760	
388				0.4	6	9	0.5556	578.760	
389			0.8	0.1	6	9	0.8889	534.530	
390				0.2	6	9	0.5556	578.760	
391				0.3	6	9	0.5556	578.760	
392				0.4	6	9	0.5556	578.760	
393			0.9	0.1	6	9	0.8889	534.530	
394				0.2	6	9	0.5556	578.760	
395				0.3	6	9	0.5556	578.760	
396				0.4	6	9	0.5556	578.760	
397			1	0.1	6	9	0.5556	578.760	
398				0.2	6	9	0.5556	578.760	
399				0.3	6	9	0.5556	578.760	
400				0.4	6	9	0.5556	578.760	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
401	15	2	0.7	0.1	6	9	0.5556	578.760	
402				0.2	6	9	0.5556	578.760	128
403				0.3	6	9	0.5556	578.760	10
404				0.4	6	9	0.5556	578.760	84
405			0.8	0.1	6	9	0.5556	578.760	
406				0.2	6	9	0.5556	578.760	116
407				0.3	6	9	0.5556	578.760	33
408				0.4	6	9	0.5556	578.760	21
409			0.9	0.1	6	9	0.5556	578.760	
410				0.2	6	9	0.5556	578.760	17
411				0.3	6	9	0.5556	578.760	9
412				0.4	6	9	0.5556	578.760	14
413			1	0.1	6	9	0.5556	578.760	
414				0.2	6	9	0.5556	578.760	99
415				0.3	6	9	0.5556	578.760	84
416				0.4	6	9	0.5556	578.760	4

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
417	15	3	0.7	0.1	6	9	0.8889	534.530	
418				0.2	6	9	0.5556	578.760	
419				0.3	6	9	0.5556	578.760	
420				0.4	6	9	0.5556	578.760	
421			0.8	0.1	6	9	0.5556	578.760	
422				0.2	6	9	0.5556	578.760	
423				0.3	6	9	0.5556	578.760	
424				0.4	6	9	0.5556	578.760	
425			0.9	0.1	6	9	0.5556	578.760	
426				0.2	6	9	0.5556	578.760	
427				0.3	6	9	0.5556	578.760	
428				0.4	6	9	0.5556	578.760	
429			1	0.1	6	9	0.5556	578.760	
430				0.2	6	9	0.5556	578.760	
431				0.3	6	9	0.5556	578.760	
432				0.4	6	9	0.5556	578.760	

ตารางที่ ง-2.2 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
433	15	4	0.7	0.1	6	9	0.5556	578.760	
434				0.2	6	9	0.5556	578.760	44
435				0.3	6	9	0.5556	578.760	23
436				0.4	6	9	0.5556	578.760	84
437			0.8	0.1	6	9	0.5556	578.760	
438				0.2	6	9	0.5556	578.760	45
439				0.3	6	9	0.5556	578.760	337
440				0.4	6	9	0.5556	578.760	16
441			0.9	0.1	6	9	0.5556	578.760	
442				0.2	6	9	0.5556	578.760	44
443				0.3	6	9	0.5556	578.760	49
444				0.4	6	9	0.5556	578.760	45
445			1	0.1	6	9	0.5556	578.760	
446				0.2	6	9	0.5556	578.760	19
447				0.3	6	9	0.5556	578.760	37
448				0.4	6	9	0.5556	578.760	6

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
449	15	5	0.7	0.1	6	9	0.5556	578.760	
450				0.2	6	9	0.5556	578.760	1
451				0.3	6	9	0.5556	578.760	9
452				0.4	6	9	0.5556	578.760	51
453			0.8	0.1	6	9	0.5556	578.760	
454				0.2	6	9	0.5556	578.760	45
455				0.3	6	9	0.5556	578.760	13
456				0.4	6	9	0.5556	578.760	44
457			0.9	0.1	6	9	0.5556	578.760	
458				0.2	6	9	0.5556	578.760	32
459				0.3	6	9	0.5556	578.760	11
460				0.4	6	9	0.5556	578.760	18
461			1	0.1	6	9	0.8889	534.530	
462				0.2	6	9	0.5556	578.760	88
463				0.3	6	9	0.5556	578.760	45
464				0.4	6	9	0.5556	578.760	2

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
465	15	6	0.7	0.1	6	9	0.5556	578.760	72
466				0.2	6	9	0.5556	578.760	50
467				0.3	6	9	0.5556	578.760	45
468				0.4	6	9	0.5556	578.760	
469			0.8	0.1	6	9	0.5556	578.760	
470				0.2	6	9	0.5556	578.760	64
471				0.3	6	9	0.5556	578.760	24
472				0.4	6	9	0.5556	578.760	12
473			0.9	0.1	6	9	0.5556	578.760	
474				0.2	6	9	0.5556	578.760	72
475				0.3	6	9	0.5556	578.760	26
476				0.4	6	9	0.5556	578.760	87
477			1	0.1	6	9	0.5556	578.760	
478				0.2	6	9	0.5556	578.760	99
479				0.3	6	9	0.5556	578.760	24
480				0.4	6	9	0.5556	578.760	45

ตารางที่ ง-2.2 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 2 (ต่อ)

No	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
481	20	1	0.7	0.1	6	9	0.5556	578.760	
482				0.2	6	9	0.5556	578.760	
483				0.3	6	9	0.5556	578.760	
484				0.4	6	9	0.5556	578.760	
485			0.8	0.1	6	9	0.5556	578.760	
486				0.2	6	9	0.5556	578.760	
487				0.3	6	9	0.5556	578.760	
488				0.4	6	9	0.5556	578.760	
489			0.9	0.1	6	9	0.5556	578.760	
490				0.2	6	9	0.5556	578.760	
491				0.3	6	9	0.5556	578.760	
492				0.4	6	9	0.5556	578.760	
493			1	0.1	6	9	0.5556	578.760	
494				0.2	6	9	0.5556	578.760	
495				0.3	6	9	0.5556	578.760	
496				0.4	6	9	0.5556	578.760	

No	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
497	20	2	0.7	0.1	6	9	0.5556	578.760	
498				0.2	6	9	0.5556	578.760	124
499				0.3	6	9	0.5556	578.760	37
500				0.4	6	9	0.5556	578.760	45
501			0.8	0.1	6	9	0.5556	578.760	
502				0.2	6	9	0.5556	578.760	98
503				0.3	6	9	0.5556	578.760	45
504				0.4	6	9	0.5556	578.760	24
505			0.9	0.1	6	9	0.5556	578.760	
506				0.2	6	9	0.5556	578.760	44
507				0.3	6	9	0.5556	578.760	41
508				0.4	6	9	0.5556	578.760	12
509			1	0.1	6	9	0.5556	578.760	
510				0.2	6	9	0.5556	578.760	88
511				0.3	6	9	0.5556	578.760	45
512				0.4	6	9	0.5556	578.760	21

No	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
513	20	3	0.7	0.1	6	9	0.5556	578.760	
514				0.2	6	9	0.5556	578.760	
515				0.3	6	9	0.5556	578.760	
516				0.4	6	9	0.5556	578.760	
517			0.8	0.1	6	9	0.8889	534.530	
518				0.2	6	9	0.5556	578.760	
519				0.3	6	9	0.5556	578.760	
520				0.4	6	9	0.5556	578.760	
521			0.9	0.1	6	9	0.5556	578.760	
522				0.2	6	9	0.8889	534.530	
523				0.3	6	9	0.5556	578.760	
524				0.4	6	9	0.5556	578.760	
525			1	0.1	6	9	0.5556	578.760	
526				0.2	6	9	0.5556	578.760	
527				0.3	6	9	0.5556	578.760	
528				0.4	6	9	0.5556	578.760	

ตารางที่ ง-2.2 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 11 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
529	20	4	0.7	0.1	6	9	0.5556	578.760	
530				0.2	6	9	0.5556	578.760	9
531				0.3	6	9	0.5556	578.760	15
532				0.4	6	9	0.5556	578.760	4
533			0.8	0.1	6	9	0.5556	578.760	
534				0.2	6	9	0.5556	578.760	18
535				0.3	6	9	0.5556	578.760	5
536				0.4	6	9	0.5556	578.760	6
537			0.9	0.1	6	9	0.5556	578.760	
538				0.2	6	9	0.5556	578.760	21
539				0.3	6	9	0.5556	578.760	19
540				0.4	6	9	0.5556	578.760	11
541			1	0.1	6	9	0.5556	578.760	
542				0.2	6	9	0.5556	578.760	32
543				0.3	6	9	0.5556	578.760	15
544				0.4	6	9	0.5556	578.760	13

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
545	20	5	0.7	0.1	6	9	0.5556	578.760	
546				0.2	6	9	0.5556	578.760	44
547				0.3	6	9	0.5556	578.760	45
548				0.4	6	9	0.5556	578.760	21
549			0.8	0.1	6	9	0.5556	578.760	
550				0.2	6	9	0.5556	578.760	65
551				0.3	6	9	0.5556	578.760	88
552				0.4	6	9	0.5556	578.760	12
553			0.9	0.1	6	9	0.5556	578.760	
554				0.2	6	9	0.5556	578.760	54
555				0.3	6	9	0.5556	578.760	32
556				0.4	6	9	0.5556	578.760	20
557			1	0.1	6	9	0.5556	578.760	
558				0.2	6	9	0.5556	578.760	77
559				0.3	6	9	0.5556	578.760	12
560				0.4	6	9	0.5556	578.760	3

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
561	20	6	0.7	0.1	6	9	0.5556	578.760	
562				0.2	6	9	0.5556	578.760	101
563				0.3	6	9	0.5556	578.760	66
564				0.4	6	9	0.5556	578.760	15
565			0.8	0.1	6	9	0.5556	578.760	
566				0.2	6	9	0.5556	578.760	74
567				0.3	6	9	0.5556	578.760	65
568				0.4	6	9	0.5556	578.760	45
569			0.9	0.1	6	9	0.5556	578.760	
570				0.2	6	9	0.5556	578.760	65
571				0.3	6	9	0.5556	578.760	44
572				0.4	6	9	0.5556	578.760	9
573			1	0.1	6	9	0.5556	578.760	
574				0.2	6	9	0.5556	578.760	68
575				0.3	6	9	0.5556	578.760	60
576				0.4	6	9	0.5556	578.760	14

2. ปัญหาขนาด 31 งาน

ตารางที่ ง-2.3 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 1

No	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
1	20	1	0.7	0.1	9	100	40.4444	570.18	
2				0.2	9	100	34.4444	589.23	
3				0.3	9	100	35.5556	585.22	
4				0.4	9	100	37.7778	577.90	
5			0.8	0.1	9	100	40	571.39	
6				0.2	9	100	35.5556	585.22	
7				0.3	9	100	35.5556	585.22	
8				0.4	9	100	35.3333	586.00	
9			0.9	0.1	9	100	35.5556	585.22	
10				0.2	9	100	40	571.39	
11				0.3	9	100	35.5556	585.22	
12				0.4	9	100	36.6667	581.45	
13			1	0.1	9	100	35.3333	586.00	
14				0.2	9	100	35.1111	586.79	
15				0.3	9	100	34.4444	589.23	
16				0.4	9	100	34.4444	589.23	

No	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
17	20	2	0.7	0.1	9	100	41.1111	568.40	
18				0.2	9	100	37.1111	580.00	
19				0.3	9	100	38.8889	574.55	
20				0.4	9	100	34.4444	589.23	
21			0.8	0.1	9	100	35.3333	586.00	
22				0.2	9	100	34.4444	589.23	
23				0.3	9	100	35.1111	586.79	
24				0.4	9	100	37.1111	580.00	
25			0.9	0.1	9	100	38.8889	574.55	
26				0.2	9	100	34.8889	587.60	
27				0.3	9	100	37.7778	577.90	
28				0.4	9	100	35.3333	586.00	
29			1	0.1	9	100	35.3333	586.00	
30				0.2	9	100	67.1111	526.72	
31				0.3	9	100	39.5556	572.64	
32				0.4	9	100	35.1111	586.79	

No	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
33	20	3	0.7	0.1	9	100	100.667	504.74	
34				0.2	9	100	59.7778	534.80	
35				0.3	9	100	43.1111	563.41	
36				0.4	9	100	43.1111	563.41	
37			0.8	0.1	9	100	67.5556	526.28	
38				0.2	9	100	48.4444	552.12	
39				0.3	9	100	51.1111	547.35	
40				0.4	9	100	46.4444	556.05	
41			0.9	0.1	9	100	47.5556	553.82	
42				0.2	9	100	66.2222	527.60	
43				0.3	9	100	48.4444	552.12	
44				0.4	9	100	61.7778	532.41	
45			1	0.1	9	100	76	519.01	
46				0.2	9	100	36	583.68	
47				0.3	9	100	51.1111	547.35	
48				0.4	9	100	46.4444	556.05	

ตารางที่ ง-2.3 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
49	20	4	0.7	0.1	9	100	35.3333	586.00	
50				0.2	9	100	35.1111	586.79	
51				0.3	9	100	35.3333	586.00	
52				0.4	9	100	40.8889	568.99	
53			0.8	0.1	9	100	35.3333	586.00	
54				0.2	9	100	35.5556	585.22	
55				0.3	9	100	37.7778	577.90	
56				0.4	9	100	38.6667	575.21	
57			0.9	0.1	9	100	35.3333	586.00	
58				0.2	9	100	35.3333	586.00	
59				0.3	9	100	35.3333	586.00	
60				0.4	9	100	35.1111	586.79	
61			1	0.1	9	100	35.3333	586.00	
62				0.2	9	100	35.3333	586.00	
63				0.3	9	100	35.1111	586.79	
64				0.4	9	100	35.1111	586.79	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
65	20	5	0.7	0.1	9	100	43.7778	561.85	
66				0.2	9	100	36	583.68	
67				0.3	9	100	43.7778	561.85	
68				0.4	9	100	35.1111	586.79	
69			0.8	0.1	9	100	35.1111	586.79	
70				0.2	9	100	35.1111	586.79	
71				0.3	9	100	37.1111	580.00	
72				0.4	9	100	36.2222	582.93	
73			0.9	0.1	9	100	35.3333	586.00	
74				0.2	9	100	35.5556	585.22	
75				0.3	9	100	34.4444	589.23	
76				0.4	9	100	35.3333	586.00	
77			1	0.1	9	100	35.5556	585.22	
78				0.2	9	100	37.1111	580.00	
79				0.3	9	100	37.1111	580.00	
80				0.4	9	100	35.3333	586.00	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
81	20	6	0.7	0.1	9	100	45.1111	558.86	
82				0.2	9	100	38.6667	575.21	
83				0.3	9	100	37.1111	580.00	
84				0.4	9	100	38.6667	575.21	
85			0.8	0.1	9	100	35.1111	586.79	
86				0.2	9	100	40.8889	568.99	
87				0.3	9	100	38.6667	575.21	
88				0.4	9	100	37.1111	580.00	
89			0.9	0.1	9	100	35.3333	586.00	
90				0.2	9	100	35.3333	586.00	
91				0.3	9	100	35.1111	586.79	
92				0.4	9	100	35.3333	586.00	
93			1	0.1	9	100	34.4444	589.23	
94				0.2	9	100	36	583.68	
95				0.3	9	100	35.1111	586.79	
96				0.4	9	100	35.5556	585.22	

ตารางที่ ง-2.3 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
97	30	1	0.7	0.1	9	100	35.1111	586.79	
98				0.2	9	100	35.3333	586.00	
99				0.3	9	100	35.5556	585.22	
100				0.4	9	100	42.2222	565.57	
101			0.8	0.1	9	100	76	519.01	
102				0.2	9	100	35.1111	586.79	
103				0.3	9	100	35.5556	585.22	
104				0.4	9	100	38.6667	575.21	
105			0.9	0.1	9	100	35.1111	586.79	
106				0.2	9	100	35.1111	586.79	
107				0.3	9	100	38.6667	575.21	
108				0.4	9	100	35.3333	586.00	
109			1	0.1	9	100	35.3333	586.00	
110				0.2	9	100	35.3333	586.00	
111				0.3	9	100	35.5556	585.22	
112				0.4	9	100	37.7778	577.90	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
113	30	2	0.7	0.1	9	100	34.4444	589.23	
114				0.2	9	100	37.1111	580.00	
115				0.3	9	100	38.8889	574.55	
116				0.4	9	100	35.3333	586.00	
117			0.8	0.1	9	100	35.1111	586.79	
118				0.2	9	100	35.1111	586.79	
119				0.3	9	100	35.1111	586.79	
120				0.4	9	100	40.6667	569.58	
121			0.9	0.1	9	100	35.1111	586.79	
122				0.2	9	100	35.1111	586.79	
123				0.3	9	100	35.3333	586.00	
124				0.4	9	100	35.1111	586.79	
125			1	0.1	9	100	34.4444	589.23	
126				0.2	9	100	37.1111	580.00	
127				0.3	9	100	35.1111	586.79	
128				0.4	9	100	35.3333	586.00	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
129	30	3	0.7	0.1	9	100	66	527.83	
130				0.2	9	100	38.6667	575.21	
131				0.3	9	100	44.2222	560.83	
132				0.4	9	100	43.1111	563.41	
133			0.8	0.1	9	100	50.8889	547.73	
134				0.2	9	100	48.8889	551.29	
135				0.3	9	100	36	583.68	
136				0.4	9	100	39.5556	572.64	
137			0.9	0.1	9	100	44.8889	559.35	
138				0.2	9	100	36	583.68	
139				0.3	9	100	46.4444	556.05	
140				0.4	9	100	44.8889	559.35	
141			1	0.1	9	100	56.6667	538.87	
142				0.2	9	100	67.5556	526.28	
143				0.3	9	100	47.1111	554.70	
144				0.4	9	100	43.1111	563.41	

ตารางที่ ง-2.3 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
145	30	4	0.7	0.1	9	100	35.3333	586.00	
146				0.2	9	100	35.3333	586.00	
147				0.3	9	100	34.4444	589.23	241
148				0.4	9	100	35.5556	585.22	420
149			0.8	0.1	9	100	35.3333	586.00	
150				0.2	9	100	35.3333	586.00	
151				0.3	9	100	35.5556	585.22	254
152				0.4	9	100	34.4444	589.23	267
153			0.9	0.1	9	100	35.3333	586.00	
154				0.2	9	100	36.6667	581.45	
155				0.3	9	100	34.4444	589.23	247
156				0.4	9	100	35.1111	586.79	187
157			1	0.1	9	100	35.5556	585.22	
158				0.2	9	100	35.3333	586.00	
159				0.3	9	100	37.1111	580.00	246
160				0.4	9	100	35.5556	585.22	270

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
161	30	5	0.7	0.1	9	100	37.1111	580.00	
162				0.2	9	100	40.4444	570.18	
163				0.3	9	100	34.4444	589.23	257
164				0.4	9	100	37.1111	580.00	451
165			0.8	0.1	9	100	34.4444	589.23	
166				0.2	9	100	34.4444	589.23	
167				0.3	9	100	35.3333	586.00	243
168				0.4	9	100	35.1111	586.79	379
169			0.9	0.1	9	100	34.4444	589.23	
170				0.2	9	100	34.4444	589.23	
171				0.3	9	100	35.3333	586.00	264
172				0.4	9	100	35.1111	586.79	265
173			1	0.1	9	100	34.4444	589.23	
174				0.2	9	100	35.1111	586.79	
175				0.3	9	100	35.3333	586.00	295
176				0.4	9	100	35.5556	585.22	315

No.	Pop size	X type	Pc	Pm	Objective value				Gen No
					n	ct	wv	fitness	
177	30	6	0.7	0.1	9	100	35.1111	586.79	
178				0.2	9	100	34.4444	589.23	
179				0.3	9	100	40.8889	568.99	
180				0.4	9	100	40.4444	570.18	
181			0.8	0.1	9	100	34.4444	589.23	
182				0.2	9	100	38.2222	576.54	
183				0.3	9	100	35.5556	585.22	
184				0.4	9	100	35.1111	586.79	
185			0.9	0.1	9	100	34.4444	589.23	
186				0.2	9	100	34.4444	589.23	
187				0.3	9	100	34.8889	587.60	
188				0.4	9	100	36	583.68	
189			1	0.1	9	100	35.5556	585.22	
190				0.2	9	100	38.2222	576.54	
191				0.3	9	100	36	583.68	
192				0.4	9	100	34.8889	587.60	

ตารางที่ ง-2.3 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
193	40	1	0.7	0.1	9	100	37.1111	580.00	
194				0.2	9	100	35.3333	586.00	
195				0.3	9	100	36	583.68	
196				0.4	9	100	35.3333	586.00	
197			0.8	0.1	9	100	34.8889	587.60	
198				0.2	9	100	35.1111	586.79	
199				0.3	9	100	35.3333	586.00	
200				0.4	9	100	35.3333	586.00	
201			0.9	0.1	9	100	34.4444	589.23	
202				0.2	9	100	35.5556	585.22	
203				0.3	9	100	35.1111	586.79	
204				0.4	9	100	38.6667	575.21	
205			1	0.1	9	100	35.1111	586.79	
206				0.2	9	100	35.1111	586.79	
207				0.3	9	100	40.8889	568.99	
208				0.4	9	100	38.2222	576.54	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
209	40	2	0.7	0.1	9	100	34.4444	589.23	
210				0.2	9	100	35.1111	586.79	
211				0.3	9	100	35.3333	586.00	
212				0.4	9	100	34.4444	589.23	
213			0.8	0.1	9	100	34.4444	589.23	
214				0.2	9	100	37.7778	577.90	
215				0.3	9	100	39.5556	572.64	
216				0.4	9	100	35.3333	586.00	
217			0.9	0.1	9	100	35.1111	586.79	
218				0.2	9	100	35.3333	586.00	
219				0.3	9	100	35.5556	585.22	
220				0.4	9	100	34.4444	589.23	
221			1	0.1	9	100	35.1111	586.79	
222				0.2	9	100	40.8889	568.99	
223				0.3	9	100	35.3333	586.00	
224				0.4	9	100	36	583.68	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
225	40	3	0.7	0.1	9	100	40.6667	569.58	
226				0.2	9	100	44.4444	560.33	
227				0.3	9	100	42.6667	564.48	
228				0.4	9	100	38.6667	575.21	
229			0.8	0.1	9	100	47.1111	554.70	
230				0.2	9	100	56.4444	539.17	
231				0.3	9	100	43.1111	563.41	
232				0.4	9	100	38.2222	576.54	
233			0.9	0.1	9	100	72.8889	521.49	
234				0.2	9	100	43.3333	562.89	
235				0.3	9	100	42.6667	564.48	
236				0.4	9	100	35.3333	586.00	
237			1	0.1	9	100	45.1111	558.86	
238				0.2	9	100	50.4444	548.50	
239				0.3	9	100	40.8889	568.99	
240				0.4	9	100	55.5556	540.43	

ตารางที่ ง-2.3 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
241	40	4	0.7	0.1	9	100	35.5556	585.22	
242				0.2	9	100	35.5556	585.22	
243				0.3	9	100	35.3333	586.00	189
244				0.4	9	100	34.4444	589.23	246
245			0.8	0.1	9	100	35.3333	586.00	
246				0.2	9	100	35.5556	585.22	
247				0.3	9	100	38.6667	575.21	187
248				0.4	9	100	38.6667	575.21	257
249			0.9	0.1	9	100	34.4444	589.23	
250				0.2	9	100	35.1111	586.79	
251				0.3	9	100	35.3333	586.00	287
252				0.4	9	100	35.3333	586.00	246
253			1	0.1	9	100	34.4444	589.23	
254				0.2	9	100	35.3333	586.00	284
255				0.3	9	100	35.3333	586.00	311
256				0.4	9	100	37.1111	580.00	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
257	40	5	0.7	0.1	9	100	35.1111	586.79	
258				0.2	9	100	35.3333	586.00	
259				0.3	9	100	35.1111	586.79	184
260				0.4	9	100	35.3333	586.00	376
261			0.8	0.1	9	100	34.4444	589.23	
262				0.2	9	100	34.4444	589.23	
263				0.3	9	100	35.1111	586.79	104
264				0.4	9	100	35.1111	586.79	379
265			0.9	0.1	9	100	35.1111	586.79	
266				0.2	9	100	34.4444	589.23	
267				0.3	9	100	35.1111	586.79	225
268				0.4	9	100	35.3333	586.00	341
269			1	0.1	9	100	35.5556	585.22	
270				0.2	9	100	35.1111	586.79	
271				0.3	9	100	34.4444	589.23	221
272				0.4	9	100	35.5556	585.22	253

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
273	40	6	0.7	0.1	9	100	34.4444	589.23	
274				0.2	9	100	35.5556	585.22	
275				0.3	9	100	35.3333	586.00	
276				0.4	9	100	40.6667	569.58	
277			0.8	0.1	9	100	35.3333	586.00	
278				0.2	9	100	35.1111	586.79	
279				0.3	9	100	34.4444	589.23	
280				0.4	9	100	34.8889	587.60	
281			0.9	0.1	9	100	35.1111	586.79	
282				0.2	9	100	35.1111	586.79	
283				0.3	9	100	42.2222	565.57	
284				0.4	9	100	37.7778	577.90	
285			1	0.1	9	100	35.1111	586.79	
286				0.2	9	100	35.1111	586.79	
287				0.3	9	100	35.1111	586.79	
288				0.4	9	100	38.2222	576.54	

ตารางที่ ง-2.4 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 2

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
289	20	1	0.7	0.1	9	100	40.4444	570.18	
290				0.2	9	100	37.7778	577.90	
291				0.3	9	100	36	583.68	
292				0.4	9	100	36	583.68	
293			0.8	0.1	9	100	44.4444	560.33	
294				0.2	9	100	46.4444	556.05	
295				0.3	9	100	42.6667	564.48	
296				0.4	9	100	35.3333	586.00	
297			0.9	0.1	9	100	40.8889	568.99	
298				0.2	9	100	36	583.68	
299				0.3	9	100	37.7778	577.90	
300				0.4	9	100	35.3333	586.00	
301			1	0.1	9	100	40.8889	568.99	
302				0.2	9	100	38.2222	576.54	
303				0.3	9	100	36	583.68	
304				0.4	9	100	35.5556	585.22	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
305	20	2	0.7	0.1	9	100	40.2222	570.78	
306				0.2	9	100	38.8889	574.55	
307				0.3	9	100	38.8889	574.55	
308				0.4	9	100	37.1111	580.00	
309			0.8	0.1	9	100	40.4444	570.18	
310				0.2	9	100	36	583.68	
311				0.3	9	100	35.3333	586.00	
312				0.4	9	100	35.3333	586.00	
313			0.9	0.1	9	100	38.2222	576.54	
314				0.2	9	100	37.6667	578.24	
315				0.3	9	100	35.1111	586.79	
316				0.4	9	100	35.1111	586.79	
317			1	0.1	9	100	37.7778	577.90	
318				0.2	9	100	36	583.68	
319				0.3	9	100	36	583.68	
320				0.4	9	100	36	583.68	

No	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
321	20	3	0.7	0.1	9	100	66	527.83	
322				0.2	9	100	44.44	560.34	
323				0.3	9	100	43.11	563.42	
324				0.4	9	100	38.6667	575.21	
325			0.8	0.1	9	100	50.8888	547.73	
326				0.2	9	100	48.8889	551.29	
327				0.3	9	100	39.5556	572.64	
328				0.4	9	100	36	583.68	
329			0.9	0.1	9	100	66	527.83	
330				0.2	9	100	43.3333	562.89	
331				0.3	9	100	35.3333	586.00	
332				0.4	9	100	35.3333	586.00	
333			1	0.1	9	100	55.5556	540.43	
334				0.2	9	100	50.4444	548.50	
335				0.3	9	100	40.8889	568.99	
336				0.4	9	100	38.2222	576.54	

ตารางที่ ง-2.4 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
337	20	4	0.7	0.1	9	100	37.7778	577.90	
338				0.2	9	100	35.3333	586.00	
339				0.3	9	100	35.1111	586.79	
340				0.4	9	100	35.1111	586.79	
341			0.8	0.1	9	100	37.7778	577.90	
342				0.2	9	100	38.6667	575.21	
343				0.3	9	100	34.4444	589.23	
344				0.4	9	100	34.4444	589.23	
345			0.9	0.1	9	100	35.7776	584.45	
346				0.2	9	100	35.7776	584.45	
347				0.3	9	100	35.3333	586.00	
348				0.4	9	100	35.1111	586.79	
349			1	0.1	9	100	35.8887	584.06	
350				0.2	9	100	37.1111	580.00	
351				0.3	9	100	35.3333	586.00	
352				0.4	9	100	34.4444	589.23	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
353	20	5	0.7	0.1	9	100	41.1111	568.40	
354				0.2	9	100	38.8889	574.55	
355				0.3	9	100	37.1111	580.00	
356				0.4	9	100	34.4444	589.23	
357			0.8	0.1	9	100	35.3333	586.00	
358				0.2	9	100	37.1111	580.00	
359				0.3	9	100	35.1111	586.79	
360				0.4	9	100	35.1111	586.79	
361			0.9	0.1	9	100	35.5556	585.22	
362				0.2	9	100	35.5556	585.22	
363				0.3	9	100	35.3333	586.00	
364				0.4	9	100	35.3333	586.00	
365			1	0.1	9	100	37.1111	580.00	
366				0.2	9	100	37.1111	580.00	
367				0.3	9	100	35.5556	585.22	
368				0.4	9	100	35.3333	586.00	

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
369	20	6	0.7	0.1	9	100	41.1111	568.40	
370				0.2	9	100	38.8889	574.55	
371				0.3	9	100	37.1111	580.00	
372				0.4	9	100	34.4444	589.23	
373			0.8	0.1	9	100	37.1111	580.00	
374				0.2	9	100	35.3333	586.00	
375				0.3	9	100	35.1111	586.79	
376				0.4	9	100	34.4444	589.23	
377			0.9	0.1	9	100	38.8889	574.55	
378				0.2	9	100	38.8889	574.55	
379				0.3	9	100	37.7778	577.90	
380				0.4	9	100	35.3333	586.00	
381			1	0.1	9	100	38.8889	574.55	
382				0.2	9	100	37.7776	577.90	
383				0.3	9	100	36	583.68	
384				0.4	9	100	36	583.68	

ตารางที่ ง-2.4 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
385	30	1	0.7	0.1	9	100	47.1111	554.70	
386				0.2	9	100	43.1111	563.41	
387				0.3	9	100	38.2222	576.54	
388				0.4	9	100	37.1111	580.00	
389			0.8	0.1	9	100	46.4444	556.05	
390				0.2	9	100	44.8889	559.35	
391				0.3	9	100	38.8889	574.55	
392				0.4	9	100	37.1111	580.00	
393			0.9	0.1	9	100	38.8889	574.55	
394				0.2	9	100	36	583.68	
395				0.3	9	100	35.3333	586.00	
396				0.4	9	100	35.1111	586.79	
397			1	0.1	9	100	37.6667	578.24	
398				0.2	9	100	38.2222	576.54	
399				0.3	9	100	36	583.68	
400				0.4	9	100	35.3333	586.00	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
401	30	2	0.7	0.1	9	100	37.1111	580.00	
402				0.2	9	100	35.3333	586.00	
403				0.3	9	100	35.5556	585.22	
404				0.4	9	100	35.3333	586.00	
405			0.8	0.1	9	100	40.4444	570.18	
406				0.2	9	100	37.1111	580.00	
407				0.3	9	100	36	583.68	
408				0.4	9	100	34.4444	589.23	
409			0.9	0.1	9	100	38.8889	574.55	
410				0.2	9	100	38.8889	574.55	
411				0.3	9	100	35.3333	586.00	
412				0.4	9	100	35.1111	586.79	
413			1	0.1	9	100	40.2222	570.78	
414				0.2	9	100	38.8889	574.55	
415				0.3	9	100	35.5556	585.22	
416				0.4	9	100	35.3333	586.00	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
417	30	3	0.7	0.1	9	100	67.5556	526.28	
418				0.2	9	100	56.4444	539.17	
419				0.3	9	100	42.6667	564.48	
420				0.4	9	100	38.2222	576.54	
421			0.8	0.1	9	100	56.4444	539.17	
422				0.2	9	100	44.2222	560.83	
423				0.3	9	100	39.5556	572.64	
424				0.4	9	100	35.1111	586.79	
425			0.9	0.1	9	100	46.4444	556.05	
426				0.2	9	100	44.8889	559.35	
427				0.3	9	100	44.8889	559.35	
428				0.4	9	100	36	583.68	
429			1	0.1	9	100	67.5556	526.28	
430				0.2	9	100	47.1111	554.70	
431				0.3	9	100	56.6667	538.87	
432				0.4	9	100	43.1111	563.41	

ตารางที่ ง-2.4 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
433	30	4	0.7	0.1	9	100	37.7778	577.90	
434				0.2	9	100	35.5556	585.22	
435				0.3	9	100	35.3333	586.00	211
436				0.4	9	100	35.3333	586.00	330
437			0.8	0.1	9	100	35.5556	585.22	
438				0.2	9	100	35.5556	585.22	
439				0.3	9	100	35.3333	586.00	284
440				0.4	9	100	35.3333	586.00	241
441			0.9	0.1	9	100	36.6667	581.45	
442				0.2	9	100	37.7778	577.90	
443				0.3	9	100	35.5556	585.22	214
444				0.4	9	100	35.1111	586.79	355
445			1	0.1	9	100	37.1111	580.00	
446				0.2	9	100	36.6667	581.45	
447				0.3	9	100	35.3333	586.00	188
448				0.4	9	100	34.4444	589.23	211

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
449	30	5	0.7	0.1	9	100	37.1111	580.00	
450				0.2	9	100	37.1111	580.00	
451				0.3	9	100	37.1111	580.00	241
452				0.4	9	100	35.5556	585.22	411
453			0.8	0.1	9	100	35.5556	585.22	
454				0.2	9	100	35.5556	585.22	
455				0.3	9	100	35.5556	585.22	274
456				0.4	9	100	35.3333	586.00	347
457			0.9	0.1	9	100	35.1111	586.79	
458				0.2	9	100	35.3333	586.00	
459				0.3	9	100	35.5556	585.22	254
460				0.4	9	100	35.1111	586.79	346
461			1	0.1	9	100	35.5556	585.22	
462				0.2	9	100	35.1111	586.79	
463				0.3	9	100	35.5556	585.22	341
464				0.4	9	100	35.1111	586.79	254

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
465	30	6	0.7	0.1	9	100	38.8889	574.55	
466				0.2	9	100	35.1111	586.79	
467				0.3	9	100	37.1111	580.00	
468				0.4	9	100	36	583.68	
469			0.8	0.1	9	100	39.5556	572.64	
470				0.2	9	100	37.1111	580.00	
471				0.3	9	100	35.3333	586.00	
472				0.4	9	100	35.1111	586.79	
473			0.9	0.1	9	100	37.7778	577.90	
474				0.2	9	100	38.8889	574.55	
475				0.3	9	100	37.7777	577.90	
476				0.4	9	100	35.1111	586.79	
477			1	0.1	9	100	37.1111	580.00	
478				0.2	9	100	35.3333	586.00	
479				0.3	9	100	35.1111	586.79	
480				0.4	9	100	35.5556	585.22	

ตารางที่ ง-2.4 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
481	40	1	0.7	0.1	9	100	40.6667	569.58	
482				0.2	9	100	38.2222	576.54	
483				0.3	9	100	38.2222	576.54	
484				0.4	9	100	36	583.68	
485			0.8	0.1	9	100	37.2222	579.65	
486				0.2	9	100	36	583.68	
487				0.3	9	100	35.5556	585.22	
488				0.4	9	100	35.1111	586.79	
489			0.9	0.1	9	100	35.5556	585.22	
490				0.2	9	100	35.5556	585.22	
491				0.3	9	100	35.1111	586.79	
492				0.4	9	100	34.4444	589.23	
493			1	0.1	9	100	38.2222	576.54	
494				0.2	9	100	37.7778	577.90	
495				0.3	9	100	38.2222	576.54	
496				0.4	9	100	36	583.68	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
497	40	2	0.7	0.1	9	100	38.2222	576.54	
498				0.2	9	100	37.1111	580.00	
499				0.3	9	100	35.5556	585.22	
500				0.4	9	100	35.3333	586.00	
501			0.8	0.1	9	100	39.5556	572.64	
502				0.2	9	100	36	583.68	
503				0.3	9	100	35.5556	585.22	
504				0.4	9	100	35.1111	586.79	
505			0.9	0.1	9	100	36	583.68	
506				0.2	9	100	37.1111	580.00	
507				0.3	9	100	35.1111	586.79	
508				0.4	9	100	34.4444	589.23	
509			1	0.1	9	100	36	583.68	
510				0.2	9	100	36	583.68	
511				0.3	9	100	37.1111	580.00	
512				0.4	9	100	34.4444	589.23	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
513	40	3	0.7	0.1	9	100	56.4444	539.17	
514				0.2	9	100	44.4444	560.33	
515				0.3	9	100	40.6667	569.58	
516				0.4	9	100	38.2222	576.54	
517			0.8	0.1	9	100	43.1111	563.41	
518				0.2	9	100	47.1111	554.70	
519				0.3	9	100	42.6667	564.48	
520				0.4	9	100	36	583.68	
521			0.9	0.1	9	100	47.5556	553.82	
522				0.2	9	100	43.3333	562.89	
523				0.3	9	100	35.3333	586.00	
524				0.4	9	100	35.1111	586.79	
525			1	0.1	9	100	55.5556	540.43	
526				0.2	9	100	50.4444	548.50	
527				0.3	9	100	42.6667	564.48	
528				0.4	9	100	35.1111	586.79	

ตารางที่ ค-2.4 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 31 งาน replication ที่ 2 (ต่อ)

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
529	40	4	0.7	0.1	9	100	35.5556	585.22	
530				0.2	9	100	35.5556	585.22	
531				0.3	9	100	35.3333	586.00	241
532				0.4	9	100	34.4444	589.23	265
533			0.8	0.1	9	100	35.5556	585.22	
534				0.2	9	100	35.3333	586.00	
535				0.3	9	100	35.1111	586.79	198
536				0.4	9	100	35.1111	586.79	265
537			0.9	0.1	9	100	35.3333	586.00	
538				0.2	9	100	35.6667	584.83	
539				0.3	9	100	34.4444	589.23	245
540				0.4	9	100	35.1111	586.79	314
541			1	0.1	9	100	35.3333	586.00	
542				0.2	9	100	35.3333	586.00	
543				0.3	9	100	35.1111	586.79	234
544				0.4	9	100	35.1111	586.79	384

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
545	40	5	0.7	0.1	9	100	36.6667	581.45	
546				0.2	9	100	37.1111	580.00	
547				0.3	9	100	35.1111	586.79	241
548				0.4	9	100	35.1111	586.79	327
549			0.8	0.1	9	100	35.5556	585.22	
550				0.2	9	100	35.1111	586.79	
551				0.3	9	100	35.3333	586.00	166
552				0.4	9	100	34.4444	589.23	354
553			0.9	0.1	9	100	37.1111	580.00	
554				0.2	9	100	38.8889	574.55	
555				0.3	9	100	35.1111	586.79	107
556				0.4	9	100	35.1111	586.79	253
557			1	0.1	9	100	35.5556	585.22	
558				0.2	9	100	35.5556	585.22	
559				0.3	9	100	35.1111	586.79	247
560				0.4	9	100	34.4444	589.23	341

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
561	40	6	0.7	0.1	9	100	36	583.68	
562				0.2	9	100	37.1111	580.00	
563				0.3	9	100	36	583.68	
564				0.4	9	100	35.3333	586.00	
565			0.8	0.1	9	100	40.4444	570.18	
566				0.2	9	100	38.2222	576.54	
567				0.3	9	100	37.1111	580.00	
568				0.4	9	100	35.1111	586.79	
569			0.9	0.1	9	100	37.7778	577.90	
570				0.2	9	100	42.2222	565.57	
571				0.3	9	100	38.8889	574.55	
572				0.4	9	100	36	583.68	
573			1	0.1	9	100	37.7778	577.90	
574				0.2	9	100	40.8889	568.99	
575				0.3	9	100	35.3333	586.00	
576				0.4	9	100	35.3333	586.00	

3. ปัญหาขนาด 39 งาน

ตารางที่ ค-2.5 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 1

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
1	20	1	0.7	0.1	8	102	4.60938	632.81	
2				0.2	8	103	2.85938	704.97	
3				0.3	10	82	2.69	660.44	
4				0.4	10	85	8.29	517.65	
5			0.8	0.1	9	92	3.3333	643.82	
6				0.2	12	70	4.52083	544.21	
7				0.3	10	83	5.29	558.27	
8				0.4	9	92	2.4444	703.92	
9			0.9	0.1	7	117	3.67347	709.11	
10				0.2	12	70	7.35417	497.26	
11				0.3	11	76	7.78512	506.41	
12				0.4	10	84	5.69	549.46	
13			1	0.1	11	76	6.14876	525.24	
14				0.2	8	102	3.60938	665.93	
15				0.3	11	76	4.33058	562.86	
16				0.4	9	92	5.5556	577.71	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
17	20	2	0.7	0.1	8	104	7.60938	583.74	
18				0.2	7	119	6.81633	638.47	
19				0.3	11	78	7.42149	506.39	
20				0.4	11	78	7.42149	506.39	
21			0.8	0.1	10	83	7.69	525.77	
22				0.2	8	105	7.85938	580.49	
23				0.3	7	120	6.53061	641.29	
24				0.4	10	83	5.09	562.36	
25			0.9	0.1	7	119	7.67347	629.45	
26				0.2	10	84	8.89	514.61	
27				0.3	7	117	2.53061	776.83	
28				0.4	11	76	5.60331	533.96	
29			1	0.1	8	103	2.60938	723.43	
30				0.2	10	84	8.09	520.74	
31				0.3	10	84	7.49	526.19	
32				0.4	8	104	7.60938	583.74	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
33	20	3	0.7	0.1	9	94	7.7778	546.99	
34				0.2	10	84	6.29	540.23	
35				0.3	9	92	6.4444	564.04	
36				0.4	9	93	5.5556	576.50	
37			0.8	0.1	9	92	2.4444	703.92	
38				0.2	9	92	3.7778	624.38	
39				0.3	9	92	3.3333	643.82	
40				0.4	10	84	8.29	519.10	
41			0.9	0.1	9	93	6.4444	562.83	
42				0.2	11	76	6.33058	522.67	
43				0.3	10	82	3.49	613.50	
44				0.4	9	94	8	545.02	
45			1	0.1	9	92	4.6667	596.60	
46				0.2	7	120	6.53061	641.29	
47				0.3	9	93	4.6667	595.39	
48				0.4	9	92	3.3333	643.82	

ตารางที่ ค-2.5 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
49	20	4	0.7	0.1	10	82	3.09	633.93	
50				0.2	11	76	5.05785	544.57	
51				0.3	11	76	4.87603	548.63	
52				0.4	8	103	2.85938	704.97	
53			0.8	0.1	12	72	7.52083	491.49	
54				0.2	7	119	4.2449	687.43	
55				0.3	9	93	6.4444	562.83	
56				0.4	9	91	4.2222	610.26	
57			0.9	0.1	8	104	4.35938	637.72	
58				0.2	10	83	3.09	632.41	
59				0.3	10	83	3.09	632.41	
60				0.4	10	82	1.89	747.13	
61			1	0.1	10	82	2.49	676.89	
62				0.2	11	76	4.33058	562.86	
63				0.3	8	103	2.85938	704.97	
64				0.4	13	66	7.00592	488.81	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
65	20	5	0.7	0.1	9	92	4.4444	602.50	
66				0.2	9	92	3.7778	624.38	
67				0.3	13	64	4.39053	540.55	
68				0.4	9	92	6.8889	558.52	
69			0.8	0.1	10	84	7.49	526.19	
70				0.2	9	92	5.1111	586.34	
71				0.3	10	83	8.49	519.02	
72				0.4	8	102	2.60938	724.42	
73			0.9	0.1	9	94	9.1111	536.62	
74				0.2	8	106	6.60938	592.82	
75				0.3	8	104	7.35938	586.20	
76				0.4	10	83	2.29	694.69	
77			1	0.1	9	92	3.3333	643.82	
78				0.2	10	82	4.89	568.30	
79				0.3	8	103	2.85938	704.97	
80				0.4	10	83	5.89	547.66	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
81	20	6	0.7	0.1	10	83	9.89	509.83	
82				0.2	11	78	7.42149	506.39	
83				0.3	8	103	4.35938	638.68	
84				0.4	13	65	4.23669	542.62	
85			0.8	0.1	11	78	7.96694	501.31	
86				0.2	12	71	8.52083	484.92	
87				0.3	8	102	2.35938	746.79	
88				0.4	10	84	7.09	530.34	
89			0.9	0.1	8	102	3.60938	665.93	
90				0.2	13	65	3.31361	578.84	
91				0.3	13	65	3.00592	595.86	
92				0.4	9	92	2.4444	703.92	
93			1	0.1	11	76	7.05785	513.70	
94				0.2	11	78	6.14876	521.75	
95				0.3	10	83	6.29	541.71	
96				0.4	7	117	4.81633	673.52	

ตารางที่ ค-2.5 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
97	30	1	0.7	0.1	11	76	5.60331	533.96	
98				0.2	10	83	3.89	595.75	
99				0.3	9	91	1.5556	833.92	
100				0.4	10	83	3.49	611.98	
101			0.8	0.1	11	76	9.05785	496.47	
102				0.2	11	76	4.69421	553.00	
103				0.3	10	84	3.49	610.49	
104				0.4	11	76	4.69421	553.00	
105			0.9	0.1	11	76	4.14876	568.43	
106				0.2	7	119	5.95918	650.10	
107				0.3	10	83	4.49	576.82	
108				0.4	9	92	3.3333	643.82	
109			1	0.1	13	65	3.00592	595.86	
110				0.2	8	104	3.60938	663.98	
111				0.3	11	76	4.14876	568.43	
112				0.4	10	83	2.69	658.92	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
113	30	2	0.7	0.1	10	83	6.69	536.47	
114				0.2	8	103	2.30938	750.86	
115				0.3	8	102	2.30935	751.85	
116				0.4	7	119	7.10204	635.22	
117			0.8	0.1	7	119	5.95918	650.10	
118				0.2	8	103	4.85938	625.68	
119				0.3	7	119	7.67347	629.45	
120				0.4	8	103	4.60938	631.83	
121			0.9	0.1	9	95	6.8889	554.97	
122				0.2	8	104	7.85938	581.44	
123				0.3	8	102	4.60938	632.81	
124				0.4	7	119	7.67347	629.45	
125			1	0.1	11	78	6.33059	519.18	
126				0.2	8	103	2.60938	723.43	
127				0.3	7	117	3.38776	721.75	
128				0.4	8	103	2.35938	745.80	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
129	30	3	0.7	0.1	9	94	7.7778	546.99	
130				0.2	9	91	3.7778	625.61	
131				0.3	9	92	3.3333	643.82	
132				0.4	13	66	9.31361	469.33	
133			0.8	0.1	9	92	3.7778	624.38	
134				0.2	9	92	5.1111	586.34	
135				0.3	11	76	5.05785	544.57	
136				0.4	9	93	3.3333	642.61	
137			0.9	0.1	10	85	7.49	524.75	
138				0.2	8	106	7.10938	586.96	
139				0.3	9	94	8.6667	539.72	
140				0.4	9	91	2	755.23	
141			1	0.1	9	92	2.4444	703.92	
142				0.2	9	91	2.8889	670.48	
143				0.3	11	77	8.87603	495.95	
144				0.4	7	117	4.53061	680.73	

ตารางที่ ค-2.5 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
145	30	4	0.7	0.1	9	94	5.1111	583.94	409
146				0.2	8	102	0.85938	1154.34	264
147				0.3	11	76	4.33058	562.86	126
148				0.4	12	70	3.52083	578.82	357
149			0.8	0.1	8	103	3.85938	655.05	318
150				0.2	10	82	2.69	660.44	315
151				0.3	12	70	6.85417	502.72	83
152				0.4	7	118	8.53061	622.97	385
153			0.9	0.1	12	70	8.52083	487.00	226
154				0.2	10	82	3.49	613.50	431
155				0.3	9	94	7.3333	551.28	119
156				0.4	11	76	4.69421	553.00	401
157			1	0.1	8	102	1.35938	918.55	261
158				0.2	11	76	4.14876	568.43	280
159				0.3	7	118	3.10204	735.98	213
160				0.4	13	65	3.31361	578.84	287

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
161	30	5	0.7	0.1	11	76	5.23967	540.79	
162				0.2	9	94	6.8889	556.13	
163				0.3	11	76	5.78512	530.87	
164				0.4	7	117	3.10204	736.73	
165			0.8	0.1	8	103	2.35938	745.80	
166				0.2	13	65	4.39053	538.06	
167				0.3	9	94	4.6667	594.21	
168				0.4	9	94	3.3333	641.43	
169			0.9	0.1	9	92	2.6667	685.13	
170				0.2	11	76	8.14876	503.25	
171				0.3	9	94	9.1111	536.62	
172				0.4	10	82	2.29	696.21	
173			1	0.1	9	92	4.2222	609.03	
174				0.2	13	64	3.15976	589.42	
175				0.3	13	65	3.31361	578.84	
176				0.4	8	104	8.60938	575.33	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
177	30	6	0.7	0.1	9	92	5.1111	586.34	
178				0.2	11	76	3.96694	574.52	
179				0.3	9	92	6.2222	567.09	
180				0.4	9	92	3.7778	624.38	
181			0.8	0.1	10	82	2.29	696.21	
182				0.2	9	92	3.3333	643.82	
183				0.3	9	92	2.4444	703.92	
184				0.4	11	76	4.33058	562.86	
185			0.9	0.1	7	116	2.53061	777.60	
186				0.2	8	103	2.85938	704.97	
187				0.3	7	118	4.81633	672.77	
188				0.4	11	76	3.78512	581.19	
189			1	0.1	8	106	5.85938	603.49	
190				0.2	8	104	2.60938	722.47	
191				0.3	8	103	6.85938	592.62	
192				0.4	9	94	5.7778	571.51	

ตารางที่ ค-2.5 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
193	40	1	0.7	0.1	7	116	2.53061	777.60	
194				0.2	9	92	4.6667	596.60	
195				0.3	7	119	7.10204	635.22	
196				0.4	10	82	2.69	660.44	
197			0.8	0.1	13	66	4.54438	531.41	
198				0.2	10	82	3.09	633.93	
199				0.3	10	82	4.09	590.34	
200				0.4	12	70	4.52083	544.21	
201			0.9	0.1	9	92	4.6667	596.60	
202				0.2	8	102	1.60938	855.60	
203				0.3	12	71	7.6875	491.93	
204				0.4	11	76	6.69421	517.94	
205			1	0.1	10	82	1.89	747.13	
206				0.2	10	82	2.29	696.21	
207				0.3	10	83	2.89	644.75	
208				0.4	9	92	2.8889	669.25	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
209	40	2	0.7	0.1	11	76	6.33058	522.67	
210				0.2	7	119	4.81633	672.03	
211				0.3	13	64	3.15976	589.42	
212				0.4	8	102	3.35938	677.28	
213			0.8	0.1	8	103	1.60938	854.62	
214				0.2	7	120	5.95918	649.37	
215				0.3	8	103	2.60938	723.43	
216				0.4	8	103	2.35938	745.80	
217			0.9	0.1	10	84	7.09	530.34	
218				0.2	10	83	5.69	550.95	
219				0.3	9	92	5.1111	586.34	
220				0.4	7	120	5.95918	649.37	
221			1	0.1	8	103	3.85938	655.05	
222				0.2	7	117	2.53061	776.83	
223				0.3	10	83	6.89	534.08	
224				0.4	10	84	3.89	594.26	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
225	40	3	0.7	0.1	9	92	5.1111	586.34	
226				0.2	8	103	2.85938	704.97	
227				0.3	9	92	2.4444	703.92	
228				0.4	10	83	6.29	541.71	
229			0.8	0.1	10	82	2.69	660.44	
230				0.2	11	76	5.05785	544.57	
231				0.3	8	103	2.85938	704.97	
232				0.4	13	66	4.39053	535.65	
233			0.9	0.1	11	76	5.96694	527.97	
234				0.2	11	77	8.5124	498.60	
235				0.3	7	119	5.95918	650.10	
236				0.4	11	76	4.69421	553.00	
237			1	0.1	11	76	9.23967	495.27	
238				0.2	8	103	6.85938	592.62	
239				0.3	7	117	4.2449	688.92	
240				0.4	13	66	5.6213	508.18	

ตารางที่ ค-2.5 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
241	40	4	0.7	0.1	7	118	3.38776	721.00	224
242				0.2	9	91	2.4444	705.16	466
243				0.3	8	102	1.85938	809.58	83
244				0.4	7	116	1.38776	956.87	253
245			0.8	0.1	11	76	3.96694	574.52	342
246				0.2	10	82	2.29	696.21	295
247				0.3	8	105	4.8593	623.77	106
248				0.4	8	102	1.35938	918.55	279
249			0.9	0.1	10	82	2.69	660.44	105
250				0.2	10	84	3.09	630.93	265
251				0.3	11	76	4.5124	557.73	350
252				0.4	7	116	2.53061	777.60	316
253			1	0.1	10	83	3.49	611.98	267
254				0.2	7	118	4.81633	672.77	182
255				0.3	9	90	0.8889	1100.80	210
256				0.4	11	76	4.33058	562.86	53

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
257	40	5	0.7	0.1	10	84	7.09	530.34	
258				0.2	7	118	4.81633	672.77	
259				0.3	10	83	4.69	571.59	
260				0.4	9	92	3.3333	643.82	
261			0.8	0.1	11	76	3.96694	574.52	
262				0.2	11	76	6.5124	520.24	
263				0.3	10	83	5.09	562.36	
264				0.4	9	94	6.8889	556.13	
265			0.9	0.1	11	76	4.69421	553.00	
266				0.2	8	103	2.35938	745.80	
267				0.3	11	76	7.60331	508.10	
268				0.4	7	120	6.53061	641.29	
269			1	0.1	7	117	4.2449	688.92	
270				0.2	10	82	3.49	613.50	
271				0.3	7	118	4.2449	688.17	
272				0.4	10	82	4.89	568.30	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
273	40	6	0.7	0.1	9	91	1.1111	975.60	
274				0.2	7	118	5.38776	660.64	
275				0.3	9	92	2.4444	703.92	
276				0.4	9	91	5.1111	587.57	
277			0.8	0.1	8	102	3.60938	665.93	
278				0.2	8	103	2.85938	704.97	
279				0.3	9	94	4.6667	594.21	
280				0.4	9	92	2.8889	669.25	
281			0.9	0.1	10	83	2.69	658.92	
282				0.2	11	76	6.5124	520.24	
283				0.3	10	82	3.49	613.50	
284				0.4	9	92	3.3333	643.82	
285			1	0.1	13	66	5.92899	503.10	
286				0.2	10	82	2.69	660.44	
287				0.3	11	76	3.96694	574.52	
288				0.4	13	64	4.23669	545.11	

ตารางที่ ค-2.6 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 2

No	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
289	20	1	0.7	0.1	11	76	5.9669	527.973	
290				0.2	10	83	5.4900	554.473	
291				0.3	13	66	5.4676	510.937	
292				0.4	10	82	3.4900	613.497	
293			0.8	0.1	9	91	2.0000	755.235	
294				0.2	11	76	5.9669	527.973	
295				0.3	10	82	4.8900	568.305	
296				0.4	12	70	5.0208	532.071	
297			0.9	0.1	10	82	2.6900	660.441	
298				0.2	10	82	5.0900	563.878	
299				0.3	13	66	7.9290	479.659	
300				0.4	12	70	9.5208	480.211	
301			1	0.1	13	66	8.0828	478.336	
302				0.2	8	105	6.8594	590.712	
303				0.3	11	76	4.8760	548.629	
304				0.4	10	83	4.6900	571.589	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
305	20	2	0.7	0.1	10	83	5.0900	562.359	
306				0.2	11	76	5.9669	527.973	
307				0.3	7	119	2.3878	788.371	
308				0.4	8	102	2.3594	746.789	
309			0.8	0.1	10	84	8.0900	520.740	
310				0.2	8	103	3.8594	655.054	
311				0.3	11	76	8.8760	497.714	
312				0.4	8	105	5.1094	618.220	
313			0.9	0.1	7	118	4.8163	672.772	
314				0.2	10	84	8.0900	520.740	
315				0.3	10	84	8.0900	520.740	
316				0.4	11	76	6.6942	517.942	
317			1	0.1	8	103	3.3594	676.300	
318				0.2	10	84	8.0900	520.740	
319				0.3	7	117	3.1020	736.731	
320				0.4	8	103	2.3594	745.804	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
321	20	3	0.7	0.1	9	94	7.7778	546.99	
322				0.2	10	84	6.29	540.23	
323				0.3	9	92	6.4444	564.04	
324				0.4	9	93	5.5556	576.50	
325			0.8	0.1	9	92	2.4444	703.92	
326				0.2	9	92	3.7778	624.38	
327				0.3	9	92	3.3333	643.82	
328				0.4	10	84	8.29	519.10	
329			0.9	0.1	9	93	6.4444	562.83	
330				0.2	11	76	6.33058	522.67	
331				0.3	10	82	3.49	613.50	
332				0.4	9	94	8	545.02	
333			1	0.1	9	92	4.6667	596.60	
334				0.2	7	120	6.53061	641.29	
335				0.3	9	93	4.6667	595.39	
336				0.4	9	92	3.3333	643.82	

ตารางที่ ค-2.6 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
337	20	4	0.7	0.1	9	92	4.6667	596.60	
338				0.2	9	94	5.5556	575.32	
339				0.3	10	82	1.89	747.13	
340				0.4	9	92	2.4444	703.92	
341			0.8	0.1	11	76	5.60331	533.96	
342				0.2	10	83	3.89	595.75	
343				0.3	12	71	8.35417	486.21	
344				0.4	10	82	2.29	696.21	
345			0.9	0.1	11	78	6.69421	514.45	
346				0.2	11	76	4.87603	548.63	
347				0.3	13	65	3.92899	552.80	
348				0.4	10	82	3.89	597.27	
349			1	0.1	10	82	2.69	660.44	
350				0.2	12	70	3.1875	595.18	
351				0.3	8	102	3.60938	665.93	
352				0.4	9	92	2.4444	703.92	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
353	20	5	0.7	0.1	10	83	5.89	547.66	
354				0.2	11	76	7.23967	511.74	
355				0.3	9	92	2.4444	703.92	
356				0.4	11	76	4.5124	557.73	
357			0.8	0.1	10	83	8.49	519.02	
358				0.2	9	92	2.8889	669.25	
359				0.3	8	106	5.85938	603.49	
360				0.4	10	84	9.29	511.94	
361			0.9	0.1	8	103	2.85938	704.97	
362				0.2	8	104	5.35938	614.14	
363				0.3	10	82	3.09	633.93	
364				0.4	10	84	6.29	540.23	
365			1	0.1	9	94	6.4444	561.64	
366				0.2	10	82	2.29	696.21	
367				0.3	12	70	8.35417	488.29	
368				0.4	10	85	6.69	533.54	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
369	20	6	0.7	0.1	9	91	2	755.23	
370				0.2	11	76	5.42149	537.26	
371				0.3	8	106	8.60938	573.46	
372				0.4	10	82	3.89	597.27	
373			0.8	0.1	9	94	5.5556	575.32	
374				0.2	11	78	6.5124	516.75	
375				0.3	8	106	8.35938	575.37	
376				0.4	13	65	3.77515	558.52	
377			0.9	0.1	9	92	5.5556	577.71	
378				0.2	11	76	4.69421	553.00	
379				0.3	10	84	10.29	506.18	
380				0.4	10	84	7.09	530.34	
381			1	0.1	11	76	4.69421	553.00	
382				0.2	13	65	3.6213	564.72	
383				0.3	10	84	4.69	570.11	
384				0.4	9	92	5.5556	577.71	

ตารางที่ ค-2.6 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
385	30	1	0.7	0.1	9	92	6.4444	564.04	
386				0.2	9	92	3.3333	643.82	
387				0.3	7	116	2.53061	777.60	
388				0.4	12	70	3.85417	565.28	
389			0.8	0.1	8	102	0.60938	1417.33	
390				0.2	11	76	5.23967	540.79	
391				0.3	9	92	2.4444	703.92	
392				0.4	10	83	2.69	658.92	
393			0.9	0.1	10	82	2.69	660.44	
394				0.2	9	90	0.8889	1100.80	
395				0.3	7	117	4.2449	688.92	
396				0.4	10	82	4.69	573.11	
397			1	0.1	8	102	2.35938	746.79	
398				0.2	13	65	4.85207	526.13	
399				0.3	10	83	3.49	611.98	
400				0.4	7	116	1.38776	956.87	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
401	30	2	0.7	0.1	13	66	8.08284	478.34	
402				0.2	10	83	7.69	525.77	
403				0.3	10	83	3.69	603.42	
404				0.4	10	84	8.09	520.74	
405			0.8	0.1	7	117	5.38776	661.39	
406				0.2	10	83	6.09	544.59	
407				0.3	10	84	8.09	520.74	
408				0.4	8	103	3.10938	689.48	
409			0.9	0.1	7	119	5.38776	659.90	
410				0.2	8	102	2.35938	746.79	
411				0.3	8	105	7.10938	587.89	
412				0.4	8	105	7.85938	580.49	
413			1	0.1	8	102	1.85938	809.58	
414				0.2	11	76	11.0579	485.47	
415				0.3	8	106	5.60938	607.68	
416				0.4	8	104	7.60938	583.74	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
417	30	3	0.7	0.1	9	92	6.8889	558.52	
418				0.2	10	83	6.69	536.47	
419				0.3	10	82	3.49	613.50	
420				0.4	10	82	3.49	613.50	
421			0.8	0.1	7	119	7.10204	635.22	
422				0.2	8	106	6.60938	592.82	
423				0.3	8	103	5.10938	620.13	
424				0.4	9	94	5.5556	575.32	
425			0.9	0.1	8	104	7.85938	581.44	
426				0.2	8	105	6.10938	600.57	
427				0.3	8	105	4.35938	636.77	
428				0.4	9	92	3.3333	643.82	
429			1	0.1	10	83	6.69	536.47	
430				0.2	7	117	3.10204	736.73	
431				0.3	8	102	1.85938	809.58	
432				0.4	8	103	6.60938	595.66	

ตารางที่ ค-2.6 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
433	30	4	0.7	0.1	11	78	6.33058	519.18	240
434				0.2	11	76	3.96694	574.52	165
435				0.3	10	82	5.49	555.99	119
436				0.4	8	103	3.35938	676.30	264
437			0.8	0.1	10	82	2.69	660.44	411
438				0.2	7	117	4.2449	688.92	437
439				0.3	10	83	4.29	582.54	308
440				0.4	7	116	1.38776	956.87	219
441			0.9	0.1	7	116	1.95918	841.09	308
442				0.2	13	65	3.15976	586.94	350
443				0.3	10	82	3.89	597.27	129
444				0.4	10	82	2.69	660.44	264
445			1	0.1	9	91	2	755.23	443
446				0.2	9	92	2.44	704.33	128
447				0.3	8	102	1.85938	809.58	180
448				0.4	10	82	3.09	633.93	337

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
449	30	5	0.7	0.1	13	65	4.85207	526.13	
450				0.2	11	76	4.69421	553.00	
451				0.3	11	78	7.23967	508.25	
452				0.4	13	65	5.00592	522.64	
453			0.8	0.1	8	103	2.35938	745.80	
454				0.2	10	82	5.09	563.88	
455				0.3	13	65	3.31361	578.84	
456				0.4	8	103	2.85938	704.97	
457			0.9	0.1	9	90	0.8889	1100.80	
458				0.2	11	78	7.60331	504.61	
459				0.3	7	119	4.81633	672.03	
460				0.4	9	92	4.2222	609.03	
461			1	0.1	9	92	2.4444	703.92	
462				0.2	8	102	0.60938	1417.33	
463				0.3	8	105	4.35938	636.77	
464				0.4	9	92	5.5556	577.71	

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
465	30	6	0.7	0.1	11	78	6.5124	516.75	
466				0.2	13	66	5.31361	513.86	
467				0.3	10	82	5.09	563.88	
468				0.4	9	92	5.5556	577.71	
469			0.8	0.1	10	83	6.29	541.71	
470				0.2	11	76	3.96694	574.52	
471				0.3	11	76	4.69421	553.00	
472				0.4	8	103	2.53938	729.25	
473			0.9	0.1	10	84	7.89	522.47	
474				0.2	8	103	6.85938	592.62	
475				0.3	9	92	5.5556	577.71	
476				0.4	9	90	1.1111	976.86	
477			1	0.1	10	82	5.09	563.88	
478				0.2	10	83	6.29	541.71	
479				0.3	10	83	3.29	621.57	
480				0.4	10	84	5.09	560.88	

ตารางที่ ค-2.6 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
481	40	1	0.7	0.1	12	70	4.02083	559.36	
482				0.2	12	70	8.02083	491.03	
483				0.3	11	76	3.96694	574.52	
484				0.4	9	92	2.4444	703.92	
485			0.8	0.1	11	76	4.87603	548.63	
486				0.2	9	92	2.6667	685.13	
487				0.3	13	65	6.23669	500.92	
488				0.4	11	76	4.87603	548.63	
489			0.9	0.1	10	82	3.29	623.09	
490				0.2	8	103	2.35938	745.80	
491				0.3	11	75	4.5124	559.55	
492				0.4	7	117	3.67347	709.11	
493			1	0.1	7	118	4.2449	688.17	
494				0.2	13	65	3.00592	595.86	
495				0.3	11	76	4.14876	568.43	
496				0.4	7	119	4.2449	687.43	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No
					n	ct	wv	fitness	
497	40	2	0.7	0.1	10	84	7.49	526.19	
498				0.2	8	105	7.10938	587.89	
499				0.3	7	119	8.81633	620.14	
500				0.4	7	119	4.2449	687.43	
501			0.8	0.1	7	117	2.53061	776.83	
502				0.2	11	76	4.69421	553.00	
503				0.3	8	103	4.85938	625.68	
504				0.4	9	94	6.4444	561.64	
505			0.9	0.1	8	103	4.60938	631.83	
506				0.2	11	76	5.60331	533.96	
507				0.3	11	76	5.60331	533.96	
508				0.4	8	102	1.85938	809.58	
509			1	0.1	8	104	7.60938	583.74	
510				0.2	7	118	4.81633	672.77	
511				0.3	8	103	2.35938	745.80	
512				0.4	7	117	2.53061	776.83	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
513	40	3	0.7	0.1	11	76	6.69421	517.94	
514				0.2	7	117	3.67247	709.15	
515				0.3	11	76	7.60331	508.10	
516				0.4	9	91	1.5556	833.92	
517			0.8	0.1	10	83	8.29	520.58	
518				0.2	10	83	4.49	576.82	
519				0.3	11	76	5.05785	544.57	
520				0.4	7	120	7.67347	628.72	
521			0.9	0.1	10	82	3.29	623.09	
522				0.2	8	105	4.35938	636.77	
523				0.3	8	103	5.35938	615.10	
524				0.4	10	83	5.49	554.47	
525			1	0.1	9	94	7.3333	551.28	
526				0.2	9	92	2.4444	703.92	
527				0.3	7	117	3.10204	736.73	
528				0.4	10	84	5.89	546.17	

ตารางที่ ค-2.6 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 39 งาน replication ที่ 2 (ต่อ)

No	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
529	40	4	0.7	0.1	11	76	5.05785	544.57	258
530				0.2	11	76	4.5124	557.73	33
531				0.3	13	64	1.92899	700.67	146
532				0.4	8	116	1.38776	898.03	224
533			0.8	0.1	12	70	3.52083	578.82	341
534				0.2	10	82	3.09	633.93	232
535				0.3	11	76	3.96694	574.52	91
536				0.4	7	120	6.53061	641.29	63
537			0.9	0.1	8	102	1.85938	809.58	396
538				0.2	11	76	5.05785	544.57	279
539				0.3	10	82	1.49	825.38	122
540				0.4	9	91	3.3333	645.06	103
541			1	0.1	10	82	2.29	696.21	347
542				0.2	13	65	4.39053	538.06	78
543				0.3	8	102	0.85938	1154.34	221
544				0.4	10	82	3.49	613.50	83

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
545	40	5	0.7	0.1	10	84	7.89	522.47	
546				0.2	8	102	2.60938	724.42	
547				0.3	8	105	4.35938	636.77	
548				0.4	8	102	3.10938	690.47	
549			0.8	0.1	9	92	5.1111	586.34	
550				0.2	13	65	3.31361	578.84	
551				0.3	10	83	6.29	541.71	
552				0.4	10	83	6.69	536.47	
553			0.9	0.1	8	103	2.35938	745.80	
554				0.2	9	91	2	755.23	
555				0.3	10	82	3.09	633.93	
556				0.4	9	91	1.5556	833.92	
557			1	0.1	13	65	4.39053	538.06	
558				0.2	13	64	4.23669	545.11	
559				0.3	12	70	5.1875	528.55	
560				0.4	8	104	5.10938	619.17	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
561	40	6	0.7	0.1	9	76	4.33058	562.86	
562				0.2	7	118	5.38776	660.64	
563				0.3	9	92	2.4444	703.92	
564				0.4	9	92	3.7778	624.38	
565			0.8	0.1	9	92	2	754.00	
566				0.2	10	83	6.29	541.71	
567				0.3	13	65	4.39053	538.06	
568				0.4	11	76	7.60331	508.10	
569			0.9	0.1	7	118	5.38776	660.64	
570				0.2	10	83	5.89	547.66	
571				0.3	11	76	7.42149	509.88	
572				0.4	7	118	5.95918	650.84	
573			1	0.1	13	64	2.39053	645.53	
574				0.2	10	83	8.29	520.58	
575				0.3	13	65	5.00592	522.64	
576				0.4	9	94	6	567.98	

4. ปัญหาขนาด 54 งาน

ตารางที่ ค-2.7 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 1

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
1	20	1	0.7	0.1	14	80	48.3112	540.61	
2				0.2	14	80	92.7679	506.35	
3				0.3	15	80	43.3956	525.11	
4				0.4	15	80	41.6622	528.42	
5			0.8	0.1	15	80	41.2622	529.23	
6				0.2	14	80	47.8622	541.28	
7				0.3	15	80	39.6622	532.60	
8				0.4	15	80	42.8622	526.10	
9			0.9	0.1	15	80	43.3956	525.11	
10				0.2	15	80	42.3289	527.12	
11				0.3	14	80	32.8827	574.15	
12				0.4	15	80	43.5289	524.87	
13			1	0.1	14	80	48.0255	541.03	
14				0.2	15	80	40.3289	531.16	
15				0.3	15	80	40.5956	530.60	
16				0.4	15	80	42.1956	527.37	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
17	20	2	0.7	0.1	15	80	40.0622	531.73	
18				0.2	15	80	39.5289	532.90	
19				0.3	14	80	12.8827	737.22	
20				0.4	15	80	40.7289	530.32	
21			0.8	0.1	15	80	40.5956	530.60	
22				0.2	15	80	38.7289	534.70	
23				0.3	14	80	47.1684	542.34	
24				0.4	15	80	39.5289	532.90	
25			0.9	0.1	15	80	44.0622	523.91	
26				0.2	15	80	40.8622	530.05	
27				0.3	14	80	9.5694	830.04	
28				0.4	15	80	38.7289	534.70	
29			1	0.1	15	80	38.7289	534.70	
30				0.2	14	80	20.3112	639.16	
31				0.3	14	80	20.3112	639.16	
32				0.4	14	80	20.3112	639.16	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
33	20	3	0.7	0.1	15	80	41.2622	529.23	
34				0.2	14	80	48.1684	540.82	
35				0.3	14	80	16.0255	684.64	
36				0.4	15	80	45.9289	520.72	
37			0.8	0.1	14	80	47.4541	541.90	
38				0.2	15	80	40.3289	531.16	
39				0.3	14	80	15.4541	692.61	
40				0.4	14	80	22.4541	622.93	
41			0.9	0.1	15	80	49.3956	515.44	
42				0.2	14	80	48.1684	540.82	
43				0.3	14	80	30.0255	584.15	
44				0.4	15	80	39.9289	532.02	
45			1	0.1	14	80	50.0255	538.16	
46				0.2	14	80	47.0255	542.56	
47				0.3	14	80	13.1684	731.40	
48				0.4	15	80	41.3956	528.96	

ตารางที่ ค-2.7 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54งาน replication ที่ 1 (ต่อ)

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
49	20	4	0.7	0.1	14	80	27.0255	596.91	
50				0.2	14	80	20.1684	640.37	
51				0.3	14	80	15.1684	696.82	
52				0.4	14	80	17.3112	668.63	
53			0.8	0.1	14	80	19.3112	647.97	
54				0.2	14	80	17.3112	668.63	
55				0.3	14	80	10.5969	795.05	
56				0.4	14	80	14.8827	701.19	
57			0.9	0.1	14	80	16.3112	680.86	
58				0.2	14	80	16.4541	679.02	
59				0.3	14	80	11.5969	766.94	
60				0.4	14	80	15.1684	696.82	
61			1	0.1	14	80	16.4541	679.02	
62				0.2	14	80	15.7398	688.55	
63				0.3	14	80	12.5969	743.30	
64				0.4	14	80	12.7398	740.22	

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
65	20	5	0.7	0.1	15	80	38.7289	534.70	
66				0.2	15	80	40.8622	530.05	
67				0.3	14	80	47.0255	542.56	
68				0.4	15	80	40.4622	530.88	
69			0.8	0.1	15	80	44.5956	522.97	
70				0.2	14	80	47.4541	541.90	
71				0.3	14	80	25.7398	603.30	
72				0.4	14	80	13.1684	731.40	
73			0.9	0.1	15	80	43.3956	525.11	
74				0.2	14	80	47.0255	542.56	
75				0.3	14	80	21.1684	632.28	
76				0.4	14	80	47.1684	542.34	
77			1	0.1	15	80	39.2622	533.49	
78				0.2	15	80	39.1289	533.79	
79				0.3	14	80	20.0255	641.59	
80				0.4	14	80	47.3112	542.12	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
81	20	6	0.7	0.1	15	80	39.5289	532.90	
82				0.2	15	80	12.8827	713.62	
83				0.3	14	80	47.168	542.34	
84				0.4	14	80	42.3289	550.71	
85			0.8	0.1	15	80	20.0255	617.99	
86				0.2	15	80	39.5289	532.90	
87				0.3	14	80	40.9956	553.36	
88				0.4	15	80	39.2622	533.49	
89			0.9	0.1	15	80	47.1684	518.74	
90				0.2	15	80	39.2622	533.49	
91				0.3	14	80	21.5969	629.04	
92				0.4	14	80	47.1684	542.34	
93			1	0.1	14	80	50.7398	537.18	
94				0.2	15	80	40.5956	530.60	
95				0.3	14	80	47.1684	542.34	
96				0.4	14	80	47.0255	542.56	

ตารางที่ ค-2.7 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
97	40	1	0.7	0.1	15	80	39.9289	532.02	
98				0.2	15	80	39.9289	532.02	
99				0.3	14	80	31.4541	578.92	
100				0.4	14	80	37.8827	560.29	
101			0.8	0.1	15	80	40.8622	530.05	
102				0.2	15	80	39.2622	533.49	
103				0.3	14	80	26.8827	597.59	
104				0.4	15	80	41.7956	528.16	
105			0.9	0.1	14	80	47.1684	542.34	
106				0.2	14	80	47.7398	541.46	
107				0.3	15	80	39.5289	532.90	
108				0.4	14	80	47.0255	542.56	
109			1	0.1	14	80	47.3112	542.12	
110				0.2	15	80	40.3289	531.16	
111				0.3	14	80	26.3112	600.38	
112				0.4	15	80	44.8622	522.51	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
113	40	2	0.7	0.1	15	80	38.7289	534.70	
114				0.2	15	80	41.5289	528.69	
115				0.3	14	80	13.1684	731.40	
116				0.4	14	80	12.8827	737.22	
117			0.8	0.1	15	80	39.2622	533.49	
118				0.2	14	80	23.0255	619.12	
119				0.3	14	80	20.5969	636.80	
120				0.4	14	80	20.0255	641.59	
121			0.9	0.1	15	80	39.2622	533.49	
122				0.2	15	80	38.7289	534.70	
123				0.3	14	80	13.4541	725.83	
124				0.4	14	80	12.8827	737.22	
125			1	0.1	15	80	38.7289	534.70	
126				0.2	15	80	39.7956	532.31	
127				0.3	14	80	19.7398	644.08	
128				0.4	14	80	20.0255	641.59	

No	Pop size	X type	Pc	Pm	Objective value				Gen. No
					n	ct	wv	fitness	
129	40	3	0.7	0.1	14	80	15.1684	696.82	
130				0.2	14	80	13.3112	728.59	
131				0.3	14	80	12.7398	740.22	
132				0.4	14	80	30.8827	580.95	
133			0.8	0.1	14	80	47.8827	541.25	
134				0.2	14	80	27.1684	596.24	
135				0.3	14	80	14.3112	710.46	
136				0.4	14	80	15.4541	692.61	
137			0.9	0.1	14	80	20.4541	637.97	
138				0.2	14	80	14.0255	715.37	
139				0.3	14	80	14.0255	715.37	
140				0.4	14	80	15.7398	688.55	
141			1	0.1	14	80	23.4541	616.38	
142				0.2	14	80	47.7398	541.46	
143				0.3	14	80	23.5969	615.48	
144				0.4	14	80	13.1684	731.40	

ตารางที่ ค-2.7 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 1 (ต่อ)

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
145	40	4	0.7	0.1	14	80	19.1684	649.30	
146				0.2	14	80	15.0255	698.98	
147				0.3	14	80	10.5969	795.05	786
148				0.4	14	80	10.5969	795.05	
149			0.8	0.1	14	80	17.3112	668.63	
150				0.2	14	80	16.1684	682.73	
151				0.3	14	80	13.7398	720.49	
152				0.4	14	80	14.3112	710.46	
153			0.9	0.1	14	80	18.1684	659.22	
154				0.2	14	80	10.7398	790.71	
155				0.3	14	80	10.3112	804.08	722
156				0.4	14	80	10.7398	790.71	
157			1	0.1	14	80	16.0255	684.64	
158				0.2	14	80	14.8827	701.19	
159				0.3	14	80	9.88265	818.60	811
160				0.4	14	80	13.1684	731.40	

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
161	40	5	0.7	0.1	14	80	19.7398	644.08	
162				0.2	15	80	43.3956	525.11	
163				0.3	14	80	13.4541	725.83	
164				0.4	14	80	20.8827	634.51	
165			0.8	0.1	15	80	39.3956	533.19	
166				0.2	14	80	20.0255	641.59	
167				0.3	14	80	20.0255	641.59	
168				0.4	14	80	47.4541	541.90	
169			0.9	0.1	14	80	47.0255	542.56	
170				0.2	15	80	38.7289	534.70	
171				0.3	14	80	12.8827	737.22	
172				0.4	14	80	14.7398	703.44	
173			1	0.1	15	80	39.2622	533.49	
174				0.2	15	80	42.3289	527.12	
175				0.3	14	80	12.5969	743.30	
176				0.4	15	80	39.2622	533.49	

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
177	40	6	0.7	0.1	14	80	47.1684	542.34	
178				0.2	15	80	39.5289	532.90	
179				0.3	14	80	13.7398	720.49	
180				0.4	14	80	13.1684	731.40	
181			0.8	0.1	14	80	13.4541	725.83	
182				0.2	14	80	22.4541	622.93	
183				0.3	14	80	13.7398	720.49	
184				0.4	14	80	22.7398	621.00	
185			0.9	0.1	14	80	19.7398	644.08	
186				0.2	15	80	38.7289	534.70	
187				0.3	14	80	12.5969	743.30	
188				0.4	15	80	39.2622	533.49	
189			1	0.1	15	80	39.2622	533.49	
190				0.2	14	80	9.16837	845.83	
191				0.3	14	80	12.8827	737.22	
192				0.4	14	80	20.0255	641.59	

ตารางที่ ค-2.7 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
193	60	1	0.7	0.1	14	80	47.1684	542.34	
194				0.2	15	80	40.8622	530.05	
195				0.3	14	80	22.3112	623.92	
196				0.4	15	80	41.5289	528.69	
197			0.8	0.1	14	80	48.0255	541.03	
198				0.2	15	80	42.1956	527.37	
199				0.3	15	80	42.0622	527.63	
200				0.4	14	80	48.0255	541.03	
201			0.9	0.1	15	80	39.2622	533.49	
202				0.2	14	80	43.9289	547.74	
203				0.3	14	80	22.1684	624.92	
204				0.4	15	80	40.0622	531.73	
205			1	0.1	15	80	39.5289	532.90	
206				0.2	15	80	39.2622	533.49	
207				0.3	14	80	24.1684	612.02	
208				0.4	15	80	40.7289	530.32	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
209	60	2	0.7	0.1	15	80	38.7289	534.70	
210				0.2	14	80	20.3112	639.16	
211				0.3	14	80	12.5969	743.30	
212				0.4	14	80	19.7398	644.08	
213			0.8	0.1	15	80	39.6622	532.60	
214				0.2	14	80	20.3112	639.16	
215				0.3	14	80	12.5969	743.30	
216				0.4	14	80	20.0255	641.59	
217			0.9	0.1	14	80	19.7398	644.08	
218				0.2	14	80	21.1684	632.28	
219				0.3	14	80	8.7398	864.30	
220				0.4	14	80	20.7398	635.65	
221			1	0.1	14	80	19.7398	644.08	
222				0.2	14	80	20.0255	641.59	
223				0.3	14	80	20.3112	639.16	
224				0.4	15	80	38.8622	534.40	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
225	60	3	0.7	0.1	14	80	47.4541	541.90	
226				0.2	14	80	31.3112	579.42	
227				0.3	14	80	14.0255	715.37	
228				0.4	14	80	15.1684	696.82	
229			0.8	0.1	14	80	47.0541	542.52	
230				0.2	14	80	30.0255	584.15	
231				0.3	14	80	24.7398	608.72	
232				0.4	15	80	43.6622	524.62	
233			0.9	0.1	15	80	45.3956	521.60	
234				0.2	15	80	39.9289	532.02	
235				0.3	14	80	13.1684	731.40	
236				0.4	14	80	13.1684	731.40	
237			1	0.1	14	80	47.1684	542.34	
238				0.2	15	80	40.7289	530.32	
239				0.3	14	80	47.7398	541.46	
240				0.4	15	80	39.2622	533.49	

ตารางที่ ค-2.7 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 1 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
241	60	4	0.7	0.1	14	80	22.8827	620.05	
242				0.2	14	80	17.0255	671.98	
243				0.3	14	80	15.0255	698.98	854
244				0.4	14	80	15.8827	686.58	
245			0.8	0.1	14	80	16.3112	680.86	
246				0.2	14	80	12.8827	737.22	
247				0.3	14	80	12.0255	756.33	
248				0.4	14	80	11.8827	759.78	
249			0.9	0.1	14	80	15.5969	690.56	
250				0.2	14	80	15.0255	698.98	
251				0.3	14	80	12.1684	752.95	702
252				0.4	14	80	12.7398	740.22	
253			1	0.1	14	80	16.5969	677.22	
254				0.2	14	80	15.4541	692.61	
255				0.3	14	80	10.7398	790.71	738
256				0.4	14	80	13.5969	723.13	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
257	60	5	0.7	0.1	15	80	38.7289	534.70	
258				0.2	15	80	39.6622	532.60	
259				0.3	14	80	19.7398	644.08	
260				0.4	15	80	39.5289	532.90	
261			0.8	0.1	15	80	41.5289	528.69	
262				0.2	15	80	39.5289	532.90	
263				0.3	14	80	15.0255	698.98	
264				0.4	15	80	39.2622	533.49	
265			0.9	0.1	15	80	39.2622	533.49	
266				0.2	15	80	38.7289	534.70	
267				0.3	14	80	9.59694	829.01	
268				0.4	14	80	14.3112	710.46	
269			1	0.1	15	80	38.7289	534.70	
270				0.2	15	80	39.2622	533.49	
271				0.3	14	80	20.3112	639.16	
272				0.4	14	80	20.0255	641.59	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
273	60	6	0.7	0.1	15	80	45.3956	521.60	
274				0.2	14	80	14.0255	715.37	
275				0.3	14	80	13.7398	720.49	
276				0.4	15	80	39.3956	533.19	
277			0.8	0.1	15	80	41.3956	528.96	
278				0.2	14	80	13.1684	731.40	
279				0.3	14	80	12.8827	737.22	
280				0.4	14	80	20.3112	639.16	
281			0.9	0.1	15	80	39.6622	532.60	
282				0.2	15	80	40.4622	530.88	
283				0.3	14	80	21.7398	627.99	
284				0.4	15	80	42.1956	527.37	
285			1	0.1	15	80	41.5289	528.69	
286				0.2	14	80	47.1684	542.34	
287				0.3	14	80	12.8827	737.22	
288				0.4	14	80	20.3112	639.16	

ตารางที่ ค-2.8 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 2

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
289	20	1	0.7	0.1	14	80	47.3112	542.12	
290				0.2	14	80	48.0255	541.03	
291				0.3	14	80	43.3112	548.86	
292				0.4	14	80	47.1684	542.34	
293			0.8	0.1	14	80	47.3112	542.12	
294				0.2	14	80	47.4541	541.90	
295				0.3	14	80	19.7398	644.08	
296				0.4	15	80	41.5289	528.69	
297			0.9	0.1	14	80	47.3112	542.12	
298				0.2	15	80	35.1289	543.84	
299				0.3	14	80	13.8827	717.90	
300				0.4	15	80	39.5289	532.90	
301			1	0.1	15	80	39.2622	533.49	
302				0.2	14	80	47.8827	541.25	
303				0.3	14	80	47.3112	542.12	
304				0.4	15	80	39.6622	532.60	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
305	20	2	0.7	0.1	14	80	47.1684	542.34	
306				0.2	15	80	39.3956	533.19	
307				0.3	14	80	17.3112	668.63	
308				0.4	14	80	13.7398	720.49	
309			0.8	0.1	15	80	38.7289	534.70	
310				0.2	14	80	47.1684	542.34	
311				0.3	14	80	47.1684	542.34	
312				0.4	15	80	41.3956	528.96	
313			0.9	0.1	15	80	39.5289	532.90	
314				0.2	15	80	39.2622	533.49	
315				0.3	14	80	20.5969	636.80	
316				0.4	14	80	19.7398	644.08	
317			1	0.1	14	80	47.0255	542.56	
318				0.2	14	80	25.3112	605.57	
319				0.3	14	80	20.3112	639.16	
320				0.4	15	80	39.1289	533.79	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
321	20	3	0.7	0.1	15	80	41.5289	528.69	
322				0.2	15	80	42.7289	526.35	
323				0.3	14	80	48.3112	540.61	
324				0.4	14	80	50.3112	537.76	
325			0.8	0.1	14	80	49.1684	539.36	
326				0.2	15	80	47.9289	517.58	
327				0.3	14	80	25.7398	603.30	
328				0.4	14	80	48.5969	540.19	
329			0.9	0.1	15	80	40.4622	530.88	
330				0.2	15	80	44.7289	522.74	
331				0.3	14	80	48.5969	540.19	
332				0.4	14	80	49.1684	539.36	
333			1	0.1	14	80	48.3112	540.61	
334				0.2	14	80	14.5969	705.73	
335				0.3	14	80	22.3112	623.92	
336				0.4	14	80	23.8827	613.73	

ตารางที่ ค-2.8 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
337	20	4	0.7	0.1	14	80	17.3112	668.63	
338				0.2	14	80	16.3112	680.86	
339				0.3	14	80	15.0255	698.98	
340				0.4	14	80	21.3112	631.18	
341			0.8	0.1	14	80	12.4541	746.44	
342				0.2	14	80	16.4541	679.02	
343				0.3	14	80	8.7398	864.30	
344				0.4	14	80	17.4541	667.00	
345			0.9	0.1	14	80	15.1684	696.82	
346				0.2	14	80	16.0255	684.64	
347				0.3	14	80	12.0255	756.33	
348				0.4	14	80	16.3112	680.86	
349			1	0.1	14	80	23.0255	619.12	
350				0.2	14	80	15.1684	696.82	
351				0.3	14	80	12.1684	752.95	
352				0.4	14	80	14.1684	712.89	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
353	20	5	0.7	0.1	15	80	39.2622	533.49	
354				0.2	14	80	47.1684	542.34	
355				0.3	14	80	12.8827	737.22	
356				0.4	14	80	47.1684	542.34	
357			0.8	0.1	15	80	38.9956	534.09	
358				0.2	15	80	32.8622	550.62	
359				0.3	14	80	48.8827	539.77	
360				0.4	14	80	47.1684	542.34	
361			0.9	0.1	14	80	47.0255	542.56	
362				0.2	15	80	38.8622	534.40	
363				0.3	14	80	20.0255	641.59	
364				0.4	14	80	47.1684	542.34	
365			1	0.1	15	80	40.0622	531.73	
366				0.2	14	80	47.0255	542.56	
367				0.3	14	80	13.1684	731.40	
368				0.4	14	80	19.7398	644.08	

No	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
369	20	6	0.7	0.1	14	80	47.1684	542.34	
370				0.2	15	80	38.7289	534.70	
371				0.3	14	80	47.1684	542.34	
372				0.4	15	80	38.7289	534.70	
373			0.8	0.1	14	80	47.1684	542.34	
374				0.2	14	80	20.0255	641.59	
375				0.3	14	80	13.7398	720.49	
376				0.4	14	80	48.0255	541.03	
377			0.9	0.1	15	80	38.7289	534.70	
378				0.2	14	80	47.1684	542.34	
379				0.3	14	80	12.7398	740.22	
380				0.4	15	80	39.5289	532.90	
381			1	0.1	14	80	47.1684	542.34	
382				0.2	14	80	47.1684	542.34	
383				0.3	14	80	12.8827	737.22	
384				0.4	14	80	47.0255	542.56	

ตารางที่ ค-2.8 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
385	40	1	0.7	0.1	14	80	48.3112	540.61	
386				0.2	15	80	39.2622	533.49	
387				0.3	14	80	47.0255	542.56	
388				0.4	15	80	43.1289	525.60	
389			0.8	0.1	15	80	41.2622	529.23	
390				0.2	15	80	39.9289	532.02	
391				0.3	14	80	47.4541	541.90	
392				0.4	15	80	43.3956	525.11	
393			0.9	0.1	14	80	47.1684	542.34	
394				0.2	15	80	41.1289	529.50	
395				0.3	14	80	47.1684	542.34	
396				0.4	15	80	41.6622	528.42	
397			1	0.1	15	80	41.9289	527.89	
398				0.2	15	80	44.0622	523.91	
399				0.3	15	80	41.5289	528.69	
400				0.4	15	80	44.7289	522.74	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
401	40	2	0.7	0.1	15	80	39.2622	533.49	
402				0.2	14	80	20.0255	641.59	
403				0.3	14	80	12.8827	737.22	
404				0.4	15	80	40.0622	531.73	
405			0.8	0.1	15	80	40.5956	530.60	
406				0.2	15	80	40.0622	531.73	
407				0.3	14	80	12.5969	743.30	
408				0.4	15	80	40.0622	531.73	
409			0.9	0.1	15	80	39.2622	533.49	
410				0.2	15	80	39.5289	532.90	
411				0.3	14	80	13.8827	717.90	
412				0.4	14	80	21.1684	632.28	
413			1	0.1	15	80	39.2622	533.49	
414				0.2	15	80	38.7289	534.70	
415				0.3	14	80	12.5969	743.30	
416				0.4	14	80	21.1684	632.28	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
417	40	3	0.7	0.1	15	80	41.7956	528.16	
418				0.2	15	80	43.9289	524.14	
419				0.3	14	80	13.3112	728.59	
420				0.4	14	80	22.5969	621.96	
421			0.8	0.1	15	80	41.7956	528.16	
422				0.2	15	80	39.5289	532.90	
423				0.3	14	80	47.7398	541.46	
424				0.4	15	80	41.1289	529.50	
425			0.9	0.1	14	80	22.1684	624.92	
426				0.2	14	80	13.1684	731.40	
427				0.3	14	80	12.8827	737.22	
428				0.4	14	80	14.1684	712.89	
429			1	0.1	14	80	49.4541	538.95	
430				0.2	15	80	44.0622	523.91	
431				0.3	14	80	47.3112	542.12	
432				0.4	15	80	40.3289	531.16	

ตารางที่ ค-2.8 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
433	40	4	0.7	0.1	14	80	13.5969	723.13	
434				0.2	14	80	13.4541	725.83	
435				0.3	14	80	13.7398	720.49	796
436				0.4	14	80	15.8827	686.58	
437			0.8	0.1	14	80	15.5969	690.56	
438				0.2	14	80	15.8827	686.58	
439				0.3	14	80	10.1684	808.78	
440				0.4	14	80	16.3112	680.86	
441			0.9	0.1	14	80	16.5969	677.22	
442				0.2	14	80	16.5969	677.22	
443				0.3	14	80	13.8827	717.90	714
444				0.4	14	80	17.1684	670.29	
445			1	0.1	14	80	14.3112	710.46	
446				0.2	14	80	14.3112	710.46	
447				0.3	14	80	10.1684	808.78	766
448				0.4	14	80	14.7398	703.44	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
449	40	5	0.7	0.1	14	80	14.0255	715.37	
450				0.2	14	80	50.7398	537.18	
451				0.3	14	80	12.8827	737.22	
452				0.4	14	80	15.0255	698.98	
453			0.8	0.1	15	80	39.9289	532.02	
454				0.2	14	80	47.1684	542.34	
455				0.3	14	80	12.8827	737.22	
456				0.4	14	80	47.1684	542.34	
457			0.9	0.1	15	80	40.0622	531.73	
458				0.2	15	80	41.2622	529.23	
459				0.3	14	80	23.5969	615.48	
460				0.4	14	80	20.3112	639.16	
461			1	0.1	15	80	38.7289	534.70	
462				0.2	15	80	41.6622	528.42	
463				0.3	14	80	12.8827	737.22	
464				0.4	14	80	20.3112	639.16	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
465	40	6	0.7	0.1	14	80	47.4541	541.90	
466				0.2	15	80	38.7289	534.70	
467				0.3	14	80	14.0255	715.37	
468				0.4	15	80	38.9956	534.09	
469			0.8	0.1	15	80	39.2622	533.49	
470				0.2	15	80	40.4622	530.88	
471				0.3	14	80	47.4541	541.90	
472				0.4	15	80	39.5289	532.90	
473			0.9	0.1	14	80	13.1684	731.40	
474				0.2	15	80	40.0622	531.73	
475				0.3	14	80	12.8827	737.22	
476				0.4	14	80	13.1684	731.40	
477			1	0.1	14	80	13.1684	731.40	
478				0.2	14	80	19.7398	644.08	
479				0.3	14	80	13.1684	731.40	
480				0.4	15	80	42.5956	526.61	

ตารางที่ ค-2.8 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
481	60	1	0.7	0.1	15	80	42.0622	527.63	
482				0.2	15	80	39.5289	532.90	
483				0.3	14	80	47.3112	542.12	
484				0.4	14	80	47.4541	541.90	
485			0.8	0.1	15	80	38.7289	534.70	
486				0.2	15	80	44.8622	522.51	
487				0.3	14	80	47.7398	541.46	
488				0.4	15	80	44.8622	522.51	
489			0.9	0.1	14	80	47.0255	542.56	
490				0.2	14	80	47.1684	542.34	
491				0.3	14	80	26.8827	597.59	
492				0.4	15	80	41.3956	528.96	
493			1	0.1	15	80	39.2622	533.49	
494				0.2	15	80	40.8633	530.04	
495				0.3	14	80	37.0255	562.40	
496				0.4	14	80	47.4541	541.90	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
497	60	2	0.7	0.1	15	80	40.0622	531.73	
498				0.2	15	80	39.5289	532.90	
499				0.3	14	80	15.7398	688.55	
500				0.4	15	80	39.9289	532.02	
501			0.8	0.1	14	80	20.3112	639.16	
502				0.2	14	80	13.7398	720.49	
503				0.3	14	80	11.5969	766.94	
504				0.4	15	80	38.7289	534.70	
505			0.9	0.1	14	80	19.7398	644.08	
506				0.2	14	80	31.8827	577.45	
507				0.3	14	80	19.7398	644.08	
508				0.4	14	80	20.0255	641.59	
509			1	0.1	15	80	39.2622	533.49	
510				0.2	14	80	20.0255	641.59	
511				0.3	14	80	13.7398	720.49	
512				0.4	14	80	19.7398	644.08	

No.	Pop size	X type	Pc	Pm	Objective value				Gen No.
					n	ct	wv	fitness	
513	60	3	0.7	0.1	15	80	36.1956	540.94	
514				0.2	14	80	22.3112	623.92	
515				0.3	14	80	13.4541	725.83	
516				0.4	14	80	14.4541	708.07	
517			0.8	0.1	14	80	47.0255	542.56	
518				0.2	14	80	29.5969	585.81	
519				0.3	14	80	13.1684	731.40	
520				0.4	14	80	30.7398	581.47	
521			0.9	0.1	14	80	13.7398	720.49	
522				0.2	14	80	47.4541	541.90	
523				0.3	14	80	13.7398	720.49	
524				0.4	15	80	40.7289	530.32	
525			1	0.1	14	80	15.5969	690.56	
526				0.2	14	80	12.8827	737.22	
527				0.3	14	80	13.3112	728.59	
528				0.4	15	80	39.2622	533.49	

ตารางที่ ค-2.8 ผลที่ได้จากการทดลองใช้ MOGA หาคำตอบของปัญหาขนาด 54 งาน replication ที่ 2 (ต่อ)

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
529	60	4	0.7	0.1	14	80	16.8827	673.70	
530				0.2	14	80	17.5969	665.39	
531				0.3	14	80	11.7398	763.32	744
532				0.4	14	80	12.1684	752.95	
533			0.8	0.1	14	80	16.4541	679.02	
534				0.2	14	80	18.8827	652.03	
535				0.3	14	80	12.8827	737.22	
536				0.4	14	80	19.3112	647.97	
537			0.9	0.1	14	80	15.8827	686.58	
538				0.2	14	80	13.8827	717.90	
539				0.3	14	80	10.0255	813.62	658
540				0.4	14	80	17.0255	671.98	
541			1	0.1	14	80	17.5969	665.39	
542				0.2	14	80	12.1684	752.95	
543				0.3	14	80	8.7398	864.30	801
544				0.4	14	80	17.7398	663.81	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
545	60	5	0.7	0.1	14	80	12.8827	737.22	
546				0.2	15	80	40.5956	530.60	
547				0.3	14	80	12.8827	737.22	
548				0.4	14	80	33.0255	573.70	
549			0.8	0.1	14	80	13.1684	731.40	
550				0.2	14	80	20.3112	639.16	
551				0.3	14	80	9.31122	840.05	
552				0.4	14	80	12.8827	737.22	
553			0.9	0.1	15	80	39.2622	533.49	
554				0.2	14	80	19.7398	644.08	
555				0.3	14	80	13.5969	723.13	
556				0.4	14	80	20.0255	641.59	
557			1	0.1	15	80	39.6622	532.60	
558				0.2	15	80	41.2622	529.23	
559				0.3	14	80	14.0255	715.37	
560				0.4	14	80	13.1684	731.40	

No.	Pop size	X type	Pc	Pm	Objective value				Gen. No.
					n	ct	wv	fitness	
561	60	6	0.7	0.1	9	80	39.7958	532.31	
562				0.2	14	80	20.8827	634.51	
563				0.3	14	80	12.5969	743.30	
564				0.4	15	80	42.7289	526.35	
565			0.8	0.1	14	80	20.3112	639.16	
566				0.2	14	80	20.0255	641.59	
567				0.3	14	80	12.5969	743.30	
568				0.4	14	80	20.3112	639.16	
569			0.9	0.1	15	80	40.0622	531.73	
570				0.2	15	80	39.2622	533.49	
571				0.3	14	80	20.0255	641.59	
572				0.4	15	80	39.2622	533.49	
573			1	0.1	15	80	38.5956	535.01	
574				0.2	15	80	39.2622	533.49	
575				0.3	15	80	39.5289	532.90	
576				0.4	15	80	38.7289	534.70	

ภาคผนวก จ-1

Duncan's Multiple Range Test (GAs)

11task problem

Response = Workload Variance

population size

MSE= 0.00135 n= 192 S= 0.00265

Order	Popsize	mean	Select
1	20	0.55604	/
2	15	0.55627	/
3	10	0.56843	

VS.	diff.	R	Result
3 vs. 1	0.01239	0.00774	S
3 vs 2	0.01216	0.00734	S
2 vs 1	0.00023	0.00734	-

population size = 20 15

Crossover type

MSE= 0.00135 n= 96 S= 0.00375

Order	Ctype	mean	Select
1	1	0.55608	/
2	6	0.55613	/
3	4	0.55625	/
4	2	0.55964	/
5	5	0.55964	/
6	3	0.57376	

VS.	diff.	R	Result
6 vs.1	0.01768	0.01180	S
6 vs.2	0.01764	0.01158	S
6 vs.3	0.01751	0.01132	S
6 vs.4	0.01413	0.01094	S
6 vs.5	0.01413	0.01038	S
5 vs 1	0.00355	0.01158	-
5 vs. 2	0.00351	0.01132	-
5 vs. 3	0.00338	0.01094	-

VS.	diff.	R	Result
5 vs.4	0	0.01038	-
4 vs.1	0.003552	0.01132	-
4 vs.2	0.00351	0.01094	-
4 vs.3	0.003385	0.01038	-
3 vs.1	0.000167	0.01094	-
3 vs.2	0.000125	0.01038	-
2 vs.1	4.2E-05	0.01038	-

crossover type = 1 6 4 2 5

Pm

MSE= 0.00135 n= 144 S= 0.0031

Order	Pm	mean	Select
1	0.4	0.556	/
2	0.3	0.55608	/
3	0.2	0.55625	/
4	0.1	0.57266	

VS.	diff.	R	Result
4 vs.1	0.01666	0.00924	S
4 vs.2	0.01658	0.00893	S
4 vs.3	0.01641	0.00848	S

VS.	diff.	R	Result
3 vs.1	0.00025	0.008934	-
3 vs.2	0.000167	0.008475	-
2 vs.1	0.01666	0.008475	-

Pm = 0.4, 0.3, 0.2

Response = No. of Generation Founding Best Solution

population size

Mse= 651.513 n= 80 Syi= 2.854

Order	Popsize	mean	Select
1	20	12.5333	/
2	15	20.025	

VS.	diff.	R	Result
2 vs. 1	7.4917	0.00774	S

population size = 20

Crossover type

MSE= 0.00135 n= 96 S= 0.00375

Order	Ctype	mean	Select
1	5	9.2292	/
2	6	12.2917	/
3	2	13.8125	/
4	1	17.3542	/
5	4	28.7083	

VS.	diff.	R	Result
5 vs 1	19.4791	10.3157	S
5 vs. 2	16.4166	10.8683	S
5 vs. 3	14.8958	11.2368	S
5 vs.4	11.3541	11.4946	-
4 vs.1	8.125	10.8683	-

VS.	diff.	R	Result
4 vs.2	5.0625	11.23675	-
4 vs.3	3.5417	11.49464	-
3 vs.1	4.5833	11.23675	-
3 vs.2	1.5208	11.49464	-
2 vs.1	3.0625	11.49464	-

crossover type = 5 6 2 1

Pm

MSE= 0.00135 n= 144 S= 0.0031

Order	Pm	mean	Select
1	0.4	0.556	/
2	0.3	0.55608	/
3	0.2	0.55625	/
4	0.1	0.57266	

VS.	diff.	R	Result
4 vs.1	0.01666	0.00924	S
4 vs.2	0.01658	0.00893	S
4 vs.3	0.01641	0.00848	S

VS.	diff.	R	Result
3 vs.1	0.00025	0.008934	-
3 vs.2	0.000167	0.008475	-
2 vs.1	0.01666	0.008475	-

Pm = 0.4, 0.3, 0.2

31 task problem

Response = Workload Variance

population size

Mse= 5.5722 n= 192 Syi= 0.17

Order	Pop_size	mean	Select
1	40	35.5717	/
2	30	36.2633	
3	20	37.5908	

VS.	diff.	R	Result
3 vs. 1	2.0191	0.49745	S
3 vs 2	1.3275	0.47189	S
2 vs 1	0.6916	0.47189	S

population size = 40

Crossover type

Mse= 5.5722 n= 96 Syi= 0.241

Order	Ctype	mean	Select
1	4	35.0185	/
2	5	35.5069	/
3	6	35.5972	/
4	2	35.9259	
5	1	36.4132	
6	3	40.39	

VS.	diff.	R	Result
6 vs.1	5.3715	0.75891	S
6 vs.2	4.8831	0.74445	S
6 vs.3	4.7928	0.72759	S
6 vs.4	4.4641	0.70349	S
6 vs.5	3.9768	0.66736	S
5 vs 1	1.3947	0.74445	S
5 vs. 2	0.9063	0.72759	S
5 vs. 3	0.816	0.70349	S

VS.	diff.	R	Result
5 vs.4	0.4873	0.667356	-
4 vs.1	0.9074	0.727587	S
4 vs.2	0.419	0.703494	-
4 vs.3	0.3287	0.667356	-
3 vs.1	0.5787	0.703494	-
3 vs.2	0.0903	0.667356	-
2 vs.1	0.4884	0.667356	-

crossover type = 4 5 6

Pm

Mse= 5.5722 n= 144 Syi= 0.197

Order	Pm	mean	Select
1	0.4	35.6327	/
2	0.3	35.966	/
3	0.2	36.358	/
4	0.1	37.9444	

VS.	diff.	R	Result
4 vs.1	2.3117	0.59407	S
4 vs.2	1.9784	0.5744	S
4 vs.3	1.5864	0.54489	S

VS.	diff.	R	Result
3 vs.1	0.7253	0.574401	-
3 vs.2	0.392	0.544894	-
2 vs.1	0.3333	0.544894	-

Pm = 0.4, 0.3, 0.2

Population Size * Crossover

Mse= 5.5722 n= 32 Syi= 0.417

Order	Ctype	mean	Select
1	4	34.7916	/
2	6	34.868	/
3	5	34.9514	/
4	2	35.1736	/
5	1	35.2499	/
6	3	38.3958	

VS.	diff.	R	Result
6 vs.1	3.6042	1.31446	S
6 vs.2	3.5278	1.28943	S
6 vs.3	3.4444	1.26022	S
6 vs.4	3.2222	1.21849	S
6 vs.5	3.1459	1.15589	S
5 vs 1	0.4583	1.28943	-
5 vs. 2	0.3819	1.26022	-
5 vs. 3	0.2985	1.21849	-

VS.	diff.	R	Result
5 vs.4	0.0763	1.155894	-
4 vs.1	0.382	1.260217	-
4 vs.2	0.3056	1.218488	-
4 vs.3	0.2222	1.155894	-
3 vs.1	0.1598	1.218488	-
3 vs.2	0.0834	1.155894	-
2 vs.1	0.0764	1.155894	-

Fix pop 40 crossovertype= 4 6 5 2 1

Crossover*Pm

a. fix crosstype

Mse= 5.5722 n= 24 Syi= 0.482

crosstype = 4

Order	Pm	mean	Select
1	0.1	34.9444	/
2	0.4	34.9537	/
3	0.2	35.0185	/
4	0.3	35.1578	/

VS.	diff.	R	Result
4 vs.1	0.2134	1.45517	-
4 vs.2	0.2041	1.40699	-
4 vs.3	0.1393	1.33471	-

VS.	diff.	R	Result
3 vs.1	0.0741	1.406989	-
3 vs.2	0.0648	1.334712	-
2 vs.1	0.0093	1.334712	-

crosstype = 5

Order	Pm	mean	Select
1	0.3	34.8333	/
2	0.4	34.9537	/
3	0.2	35.3981	/
4	0.1	36.8426	

VS.	diff.	R	Result
4 vs.1	2.0093	1.45517	S
4 vs.2	1.8889	1.40699	S
4 vs.3	1.4445	1.33471	S

VS.	diff.	R	Result
3 vs.1	0.5648	1.406989	-
3 vs.2	0.4444	1.334712	-
2 vs.1	0.1204	1.334712	-

crosstype = 6

Order	Pm	mean	Select
1	0.3	34.9444	/
2	0.2	35.1667	/
3	0.4	35.3426	/
4	0.1	36.9352	

VS.	diff.	R	Result
4 vs.1	1.9908	1.45517	S
4 vs.2	1.7685	1.40699	S
4 vs.3	1.5926	1.33471	S

VS.	diff.	R	Result
3 vs.1	0.3982	1.406989	-
3 vs.2	0.1759	1.334712	-
2 vs.1	0.2223	1.334712	-

When fix Crossover type Pm= 0.2, 0.3, 0.4

b. fix Pm

Mse= 5.5722 n= 24 Syi= 0.482

Pm= 0.2

Order	Ctype	mean	Select
1	4	35.0185	/
2	6	35.1666	/
3	5	35.3981	/
4	2	35.5555	/
5	1	36.4352	/
6	3	40.5741	

VS.	diff.	R	Result
6 vs.1	5.5556	1.51781	S
6 vs.2	5.4075	1.4889	S
6 vs.3	5.176	1.45517	S
6 vs.4	5.0186	1.40699	S
6 vs.5	4.1389	1.33471	S
5 vs 1	1.4167	1.4889	-
5 vs. 2	1.2686	1.45517	-
5 vs. 3	1.0371	1.40699	-

VS.	diff.	R	Result
5 vs.4	0.8797	1.334712	-
4 vs.1	0.537	1.455173	-
4 vs.2	0.3889	1.406989	-
4 vs.3	0.1574	1.334712	-
3 vs.1	0.3796	1.406989	-
3 vs.2	0.2315	1.334712	-
2 vs.1	0.1481	1.334712	-

Pm= 0.3

Order	Ctype	mean	Select
1	5	34.8333	/
2	6	34.9444	/
3	4	35.1574	/
4	2	35.185	/
5	1	35.6852	/
6	3	39.9907	

VS.	diff.	R	Result
6 vs.1	5.1574	1.51781	S
6 vs.2	5.0463	1.4889	S
6 vs.3	4.8333	1.45517	S
6 vs.4	4.8057	1.40699	S
6 vs.5	4.3055	1.33471	S
5 vs 1	0.8519	1.4889	-
5 vs. 2	0.7408	1.45517	-
5 vs. 3	0.5278	1.40699	-

VS.	diff.	R	Result
5 vs.4	0.5002	1.334712	-
4 vs.1	0.3517	1.455173	-
4 vs.2	0.2406	1.406989	-
4 vs.3	0.0276	1.334712	-
3 vs.1	0.3241	1.406989	-
3 vs.2	0.213	1.334712	-
2 vs.1	0.1111	1.334712	-

Pm= 0.4

Order	Ctype	mean	Select
1	2	34.861	/
2	4	34.9537	/
3	5	34.9537	/
4	1	35.0833	/
5	6	35.3426	/
6	3	38.6018	

VS.	diff.	R	Result
6 vs.1	3.7408	1.51781	S
6 vs.2	3.6481	1.4889	S
6 vs.3	3.6481	1.45517	S
6 vs.4	3.5185	1.40699	S
6 vs.5	3.2592	1.33471	S
5 vs 1	0.4816	1.4889	-
5 vs. 2	0.3889	1.45517	-
5 vs. 3	0.3889	1.40699	-

VS.	diff.	R	Result
5 vs.4	0.2593	1.334712	-
4 vs.1	0.2223	1.455173	-
4 vs.2	0.1296	1.406989	-
4 vs.3	0.1296	1.334712	-
3 vs.1	0.0927	1.406989	-
3 vs.2	0	1.334712	-
2 vs.1	0.0927	1.334712	-

When fix Pm Crosstype = 1 2 4 5 6

39 task problem

Response = Workload Variance

population size

Mse= 27.1469 n= 192 Syi= 0.376

Order	Pop_size	mean	Select
1	40	6.6879	/
2	30	7.4833	/
3	20	9.6466	

VS.	diff.	R	Result
3 vs. 1	2.9587	1.13558	S
3 vs 2	2.1633	1.09797	S
2 vs 1	0.7954	1.04157	

population size = 40 30

Crossover type

Mse= 27.1469 n= 96 Syi= 0.532

Order	Ctype	mean	Select
1	4	5.1526	/
2	2	7.1378	
3	1	7.5163	
4	6	7.9288	
5	5	9.3011	
6	3	10.5989	

VS.	diff.	R	Result
6 vs.1	5.4463	1.67508	S
6 vs.2	3.4611	1.64317	S
6 vs.3	3.0826	1.60595	S
6 vs.4	2.6701	1.55277	S
6 vs.5	1.2978	1.47301	
5 vs 1	4.1485	1.64317	S
5 vs. 2	2.1633	1.60595	S
5 vs. 3	1.7848	1.55277	S

VS.	diff.	R	Result
5 vs.4	1.3723	1.473005	-
4 vs.1	2.7762	1.605948	S
4 vs.2	0.791	1.552771	-
4 vs.3	0.4125	1.473005	-
3 vs.1	2.3637	1.552771	S
3 vs.2	0.3785	1.473005	-
2 vs.1	1.9852	1.473005	S

crossover type = 4

Pm

Mse= 27.1469 n= 144 Syi= 0.434

Order	Pm	mean	Select
1	0.4	5.329	/
2	0.3	5.8143	/
3	0.2	7.7662	
4	0.1	12.8475	

VS.	diff.	R	Result
4 vs.1	7.5185	1.2027	S
4 vs.2	7.0332	1.26783	S
4 vs.3	5.0813	1.31125	S

VS.	diff.	R	Result
3 vs.1	2.4372	1.267832	S
3 vs.2	1.9519	1.311251	S
2 vs.1	0.4853	1.311251	

Pm = 0.4, 0.3

Crossover*Pm

fix crosstype = 4

Mse= 5.5722 n= 24 Syi= 0.482

Order	Pm	mean	Select
1	0.1	34.9444	/
2	0.4	34.9537	/
3	0.2	35.0185	/
4	0.3	35.1578	/

VS.	diff.	R	Result
4 vs.1	0.2134	1.45517	-
4 vs.2	0.2041	1.40699	-
4 vs.3	0.1393	1.33471	-

VS.	diff.	R	Result
3 vs.1	0.0741	1.406989	-
3 vs.2	0.0648	1.334712	-
2 vs.1	0.0093	1.334712	-

54 task problem

Response = Workload Variance

population size

Mse= 73.243 n= 192 Syi= 0.618

Order	Pop_size	mean	Select
1	60	100.789	/
2	40	104.308	
3	20	109.557	

VS.	diff.	R	Result
3 vs. 1	8.76738	1.86526	S
3 vs 2	5.24843	1.8035	S
2 vs 1	3.51895	1.71085	S

population size = 60

Crossover type

Mse= 73.243 n= 96 Syi= 0.873

Order	Ctype	mean	Select
1	4	84.331	/
2	2	105.172	
3	1	105.335	
4	6	107.347	
5	5	111.693	
6	3	115.43	

VS.	diff.	R	Result
6 vs.1	31.099	2.75143	S
6 vs.2	10.258	2.69902	S
6 vs.3	10.095	2.63788	S
6 vs.4	8.083	2.55053	S
6 vs.5	3.737	2.41951	S
5 vs 1	27.362	2.69902	S
5 vs. 2	6.521	2.63788	S
5 vs. 3	6.358	2.55053	S

VS.	diff.	R	Result
5 vs.4	4.346	2.419509	S
4 vs.1	23.016	2.637876	S
4 vs.2	2.175	2.550529	-
4 vs.3	2.012	2.419509	-
3 vs.1	21.004	2.550529	S
3 vs.2	0.163	2.419509	-
2 vs.1	20.841	2.419509	S

crossover type = 4

Pm

Mse= 73.243 n= 144 Syi= 0.713

Order	Pm	mean	Select
1	0.3	102.186	/
2	0.4	103.065	/
3	0.2	104.136	
4	0.1	110.151	

VS.	diff.	R	Result
4 vs.1	7.965	2.15382	S
4 vs.2	7.086	2.0825	S
4 vs.3	6.015	1.97552	S

VS.	diff.	R	Result
3 vs.1	1.95	2.082498	S
3 vs.2	1.071	1.975521	
2 vs.1	0.879	1.975521	

Pm = 0.3, 0.4

Population Size * Crossover

Fix population Size = 60

Mse= 73.243 n= 32 Syi= 1.513

Order	Ctype	mean	Select
1	4	81.7873	/
2	6	99.1897	/
3	5	103.023	/
4	1	104.742	/
5	2	105.048	/
6	3	110.944	

VS.	diff.	R	Result
6 vs.1	29.1568	4.76561	S
6 vs.2	11.7544	4.67484	S
6 vs.3	7.9211	4.56894	S
6 vs.4	6.2019	4.41765	S
6 vs.5	5.8957	4.19071	S
5 vs 1	23.2611	4.67484	S
5 vs. 2	5.8587	4.56894	S
5 vs. 3	2.0254	4.41765	

VS.	diff.	R	Result
5 vs.4	0.3062	4.190712	-
4 vs.1	22.9549	4.568935	S
4 vs.2	5.5525	4.417646	S
4 vs.3	1.7192	4.190712	-
3 vs.1	21.2357	4.417646	S
3 vs.2	3.8333	4.190712	-
2 vs.1	17.4024	4.190712	S

When fix pop 60 crossovertype= 4

Crossover*Pm

Fix crosstype = 4

Mse= 5.5722 n= 24 Syi= 0.482

Order	Pm	mean	Select
1	0.2	83.5838	/
2	0.3	84.3915	/
3	0.1	84.4556	/
4	0.4	84.8915	/

VS.	diff.	R	Result
4 vs.1	1.3077	5.27575	-
4 vs.2	0.5	5.10106	-
4 vs.3	0.4359	4.83902	-

VS.	diff.	R	Result
3 vs.1	0.8718	5.101058	-
3 vs.2	0.0641	4.839018	-
2 vs.1	0.8077	4.839018	-

When fix crossover type = 4 Pm= 0.1, 0.2, 0.3, 0.4

Pc*Pm

Mse= 73.243 n= 36 Syi= 1.426

Fix Pm=0.2

Order	Pc	mean	Select
1	0.8	101.52	/
2	0.9	101.733	/
3	1	103.507	/
4	0.7	105.502	/

VS.	diff.	R	Result
4 vs.1	3.9829	4.30763	-
4 vs.2	3.7696	4.165	-
4 vs.3	1.9954	3.95104	-

VS.	diff.	R	Result
3 vs.1	1.9875	4.164997	-
3 vs.2	1.7742	3.951041	-
2 vs.1	0.2133	3.951041	-

Fix Pm= 0.3

Order	Pc	mean	Select
1	0.7	101.839	/
2	0.9	102.383	/
3	0.1	105.646	/
4	0.8	106.675	/

VS.	diff.	R	Result
4 vs.1	4.8353	4.30763	S
4 vs.2	4.2911	4.165	S
4 vs.3	1.0282	3.95104	-

VS.	diff.	R	Result
3 vs.1	3.8071	4.164997	-
3 vs.2	3.2629	3.951041	-
2 vs.1	0.5442	3.951041	-

When fix Pm Pc = 0.7, 0.8, 0.9, 1.0

Response = No. of Generation Founding Best Solution

Pm

Mse= 35328 n= 4 Syi= 93.98

Order	Pm	mean	Select
1	0.3	281.5	/
2	0.4	633.5	

VS.	diff.	R	Result
2 vs.1	352	325.166	S

Pm = 0.3

ภาคผนวก จ-2

Duncan's Multiple Range Test (MOGA)

11task problem

Response = ค่าวัตถุประสงค์รวม

population size

MSE= 55.076 n= 192 Syi= 0.53559

Order	Popsize	mean	Select
1	10	575.017	
2	15	577.332	/
3	20	578.255	/

VS.	diff.	R	Result
3 vs. 1	3.238	1.563916	S
3 vs 2	0.923	1.483578	-
2 vs 1	2.315	1.483578	S

population size = 20 15

Crossover type

MSE= 55.076 n= 96 Syi= 0.75744

Order	Ctype	mean	Select
1	3	572.70	
2	1	574.56	
3	6	577.33	/
4	5	578.26	/
5	2	579.18	/
6	4	579.18	/

VS.	diff.	R	Result
6 vs.1	6.48	2.385921	S
6 vs.2	4.617	2.340475	S
6 vs.3	1.85	2.287455	-
6 vs.4	0.925	2.211711	-
6 vs.5	0	2.098096	-
5 vs 1	6.48	2.34048	S
5 vs. 2	4.617	2.28745	S
5 vs. 3	1.85	2.21171	-

VS.	diff.	R	Result
5 vs.4	0.925	2.09810	-
4 vs.1	5.555	2.28745	S
4 vs.2	3.692	2.21171	S
4 vs.3	0.925	2.09810	-
3 vs.1	4.63	2.21171	S
3 vs.2	2.767	2.09810	S
2 vs.1	1.863	2.09810	-

crossover type = 4 2 5 6

Pm

MSE= 55.076 n= 144 Syi= 0.61844

Order	Pm	mean	Select
1	0.1	571.17	
2	0.2	577.95	/
3	0.3	579.18	/
4	0.4	579.18	/

VS.	diff.	R	Result
4 vs.1	8.015	1.867699	S
4 vs.2	1.233	1.805855	-
4 vs.3	0	1.713088	-

VS.	diff.	R	Result
3 vs.1	8.015	1.805855	S
3 vs.2	1.233	1.713088	-
2 vs.1	8.015	1.713088	S

Pm = 0.4, 0.3, 0.2

Population size*crossover type

Mse= 55.076 n= 32 Syi= 1.31192

fix popsize=15

Order	Ctype	mean	Select
1	1	573.641	
2	3	575.01	
3	5	577.8	/
4	2	579.18	/
5	6	579.18	/
6	4	579.18	/

VS.	diff.	R	Result
6 vs.1	5.539	4.132537	S
6 vs.2	4.17	4.053822	S
6 vs.3	1.38	3.961988	-
6 vs.4	0	3.830796	-
6 vs.5	0	3.634009	-
5 vs 1	5.539	4.05382	S
5 vs. 2	4.17	3.96199	S
5 vs. 3	1.38	3.83080	-

VS.	diff.	R	Result
5 vs.4	0	3.63401	-
4 vs.1	5.539	3.96199	S
4 vs.2	4.17	3.83080	-
4 vs.3	1.38	3.63401	-
3 vs.1	4.159	3.83080	-
3 vs.2	2.79	3.63401	-
2 vs.1	1.369	3.63401	-

When population size = 15 crossover type=2 4 5 6

fix popsize=20

Order	Ctype	mean	Select
1	3	573.631	
2	1	579.18	/
3	5	579.18	/
4	2	579.18	/
5	6	579.18	/
6	4	579.18	/

VS.	diff.	R	Result
6 vs.1	5.549	4.132537	S
6 vs.2	0	4.053822	-
6 vs.3	0	3.961988	-
6 vs.4	0	3.830796	-
6 vs.5	0	3.634009	-
5 vs 1	5.549	4.05382	S
5 vs. 2	0	3.96199	-
5 vs. 3	0	3.83080	-

VS.	diff.	R	Result
5 vs.4	0	3.63401	-
4 vs.1	5.549	3.96199	S
4 vs.2	0	3.83080	-
4 vs.3	0	3.63401	-
3 vs.1	5.549	3.83080	-
3 vs.2	0	3.63401	-
2 vs.1	0	3.63401	-

When population size = 15 crossover type=2 4 5 6

Population size * Pm

MSE= 55.076 n= 144 Syi= 1.07118

fix population size=15

Order	Pm	mean	Select
1	0.1	572.720	
2	0.2	578.250	/
3	0.3	579.180	/
4	0.4	579.180	/

VS.	diff.	R	Result
4 vs.1	6.46	3.23495	S
4 vs.2	0.93	3.127832	-
4 vs.3	0	2.967156	-

VS.	diff.	R	Result
3 vs.1	6.46	3.127832	S
3 vs.2	0.93	2.967156	-
2 vs.1	6.46	2.967156	S

When population size = 15 Pm=0.2 0.3 0.4

fix population size=20

Order	Pm	mean	Select
1	0.1	576.4	
2	0.2	578.258	
3	0.3	579.18	
4	0.4	579.18	

VS.	diff.	R	Result
4 vs.1	2.78	3.23495	-
4 vs.2	0.922	3.127832	-
4 vs.3	0	2.967156	-

VS.	diff.	R	Result
3 vs.1	2.78	3.127832	-
3 vs.2	0.922	2.967156	-
2 vs.1	1.858	2.967156	-

all levels of Pm are not significant at popsize = 20

Crossover type*Pm

MSE= 55.076 n= 144 Syi= 1.07118

fix crosstype=6

Order	Pm	mean	Select
1	0.1	571.871	
2	0.2	579.18	/
3	0.3	579.18	/
4	0.4	579.18	/

VS.	diff.	R	Result
4 vs.1	7.309	3.23495	S
4 vs.2	0	3.127832	-
4 vs.3	0	2.967156	-

VS.	diff.	R	Result
3 vs.1	7.309	3.127832	-
3 vs.2	0	2.967156	-
2 vs.1	7.309	2.967156	-

When crossover type=6 Pm=0.2 0.3 0.4

Response = No. of Generation Founding Best Solution

Crossover type

MSE= 714.745 n= 96 Syi= 3.85882

Order	Ctype	mean	Select
1	4	29.3542	/
2	5	36.8958	
3	6	46.8542	
4	2	47.75	

VS.	diff.	R	Result
4 vs.1	18.3958	11.76941	S
4 vs.2	10.8542	11.38353	-
4 vs.3	0.8958	10.80471	-

VS.	diff.	R	Result
3 vs.1	17.5	11.38353	S
3 vs.2	9.9584	10.80471	-
2 vs.1	7.5416	10.80471	-

crossover type = 4

Pm

MSE= 714.745 n= 144 Syi= 3.34184

Order	Pm	mean	Select
1	0.4	30.9375	/
2	0.3	36.8906	/
3	0.2	52.8125	/

VS.	diff.	R	Result
3 vs.1	21.875	9.858426	S
3 vs.2	15.9219	9.35715	S
2 vs.1	5.9531	9.35715	-

Pm = 0.4, 0.3

31 task problem

Response = ค่าวัตถุประสงค์รวม

population size

Mse= 122.1 n= 192 Syi= 0.79746

Order	Pop_size	mean	Select
1	20	576.23	
2	30	578.41	/
3	40	580.05	/

VS.	diff.	R	Result
3 vs. 1	3.825	2.328574	S
3 vs 2	1.641	2.208956	-
2 vs 1	2.184	2.208956	-

population size = 40 30

Crossover type

Mse= 122.1 n= 96 Syi= 1.12777

Order	Ctype	mean	Select
1	3	561.12	
2	2	578.94	
3	6	580.17	
4	1	580.28	/
5	4	584.42	/
6	5	584.44	

VS.	diff.	R	Result
6 vs.1	23.324	3.552489	S
6 vs.2	5.501	3.484823	S
6 vs.3	4.269	3.405879	S
6 vs.4	4.163	3.293101	S
6 vs.5	0.021	3.123935	-
5 vs 1	23.303	3.484823	S
5 vs. 2	5.48	3.405879	S
5 vs. 3	4.248	3.293101	S

VS.	diff.	R	Result
5 vs.4	4.142	3.123935	S
4 vs.1	19.161	3.405879	S
4 vs.2	1.338	3.293101	-
4 vs.3	0.106	3.123935	-
3 vs.1	19.055	3.293101	S
3 vs.2	1.232	3.123935	-
2 vs.1	17.823	3.123935	S

crossover type = 5, 4

Pm

Mse= 122.1 n= 144 Syi= 0.92082

Order	Pm	mean	Select
1	0.1	573.82	/
2	0.2	576.83	/
3	0.3	579.91	/
4	0.4	582.36	

VS.	diff.	R	Result
4 vs.1	8.543	2.780888	S
4 vs.2	5.534	2.688806	S
4 vs.3	2.459	2.550682	-

VS.	diff.	R	Result
3 vs.1	6.084	2.688806	S
3 vs.2	3.075	2.550682	S
2 vs.1	3.009	2.550682	S

Pm = 0.4, 0.3

Crossover*Pc

Mse= 122.1 n= 24 Syi= 2.25555

fix crosstype = 4

Order	Pc	mean	Select
1	1	583.153	
2	0.8	583.847	
3	0.7	584.946	
4	0.9	585.743	

VS.	diff.	R	Result
4 vs.1	2.5897	6.811757	-
4 vs.2	1.8964	6.586202	-
4 vs.3	0.797	6.24787	-

VS.	diff.	R	Result
3 vs.1	1.7927	6.586202	-
3 vs.2	1.0994	6.24787	-
2 vs.1	0.6933	6.24787	-

all levels of Pc are not significant at crossover type=4

fix crosstype = 5

Order	Pc	mean	Select
1	0.7	582.818	-
2	1	583.145	-
3	0.8	585.632	-
4	0.9	586.175	-

VS.	diff.	R	Result
4 vs.1	3.357	6.811757	-
4 vs.2	3.03	6.586202	-
4 vs.3	0.543	6.24787	-

VS.	diff.	R	Result
3 vs.1	2.814	6.586202	-
3 vs.2	2.487	6.24787	-
2 vs.1	0.327	6.24787	-

all levels of Pc are not significant at crossover type=5

Crossover*Pm

Mse= 122.1 n= 24 Syi= 2.25555

fix crosstype = 4

Order	Pm	mean	Select
1	0.1	583.429	-
2	0.2	584.044	-
3	0.4	585.068	-
4	0.3	585.148	-

VS.	diff.	R	Result
4 vs.1	1.719	6.811757	-
4 vs.2	1.104	6.586202	-
4 vs.3	0.08	6.24787	-

VS.	diff.	R	Result
3 vs.1	1.639	6.586202	-
3 vs.2	1.024	6.24787	-
2 vs.1	0.615	6.24787	-

all levels of Pm are not significant at crossover type=4

fix crosstype = 5

Order	Pm	mean	Select
1	0.1	583.38	
2	0.2	583.895	
3	0.3	584.368	
4	0.4	586.127	

VS.	diff.	R	Result
4 vs.1	2.747	6.811757	-
4 vs.2	2.232	6.586202	-
4 vs.3	1.759	6.24787	-

VS.	diff.	R	Result
3 vs.1	0.988	6.586202	-
3 vs.2	0.473	6.24787	-
2 vs.1	0.515	6.24787	-

all levels of Pm are not significant at crossover type=5

39 task problem

Response = ค่าวัตถุประสงค์รวม

population size

Mse= 10842 n= 192 Syi= 7.51457

Order	Pop_size	mean	Select
1	20	589.675	
2	40	634.463	/
3	30	637.161	/

VS.	diff.	R	Result
3 vs. 1	47.486	21.94254	S
3 vs 2	2.698	20.81536	-
2 vs 1	44.788	20.81536	S

population size = 40 30

Crossover type

Mse= 10842 n= 96 Syi= 10.6272

Order	Ctype	mean	Select
1	3	597.597	
2	6	603.056	
3	5	616.609	
4	2	621.337	
5	1	624.525	
6	4	659.473	/

VS.	diff.	R	Result
6 vs.1	61.876	33.4757	S
6 vs.2	56.417	32.83807	S
6 vs.3	42.864	32.09416	S
6 vs.4	38.136	31.03144	S
6 vs.5	34.948	29.43736	S
5 vs 1	26.928	32.83807	-
5 vs. 2	21.469	32.09416	-
5 vs. 3	7.916	31.03144	-

VS.	diff.	R	Result
5 vs.4	3.188	29.43736	-
4 vs.1	23.74	32.09416	-
4 vs.2	18.281	31.03144	-
4 vs.3	4.728	29.43736	-
3 vs.1	19.012	31.03144	-
3 vs.2	13.553	29.43736	-
2 vs.1	5.459	29.43736	-

crossover type = 4

Response = No. of Generation Founding Best Solution

Pm

Mse= 10726.2 n= 16 Syi= 25.8918

Order	Pm	mean	Select
1	0.3	162.875	/
2	0.4	243.000	
3	0.2	262.500	
4	0.1	307.875	

VS.	diff.	R	Result
4 vs.1	145	80.78255	S
4 vs.2	64.875	78.7112	-
4 vs.3	45.375	74.82742	-

VS.	diff.	R	Result
3 vs.1	99.625	78.7112	S
3 vs.2	19.5	74.82742	-
2 vs.1	80.125	74.82742	S

Pm = 0.3

54 task problem

Response = ค่าวัตถุประสงค์รวม

population size

Mse= 3238.1 n= 192 Syi= 4.10671

Order	Pop_size	mean	Select
1	20	588.754	
2	40	623.07	/
3	60	623.343	/

VS.	diff.	R	Result
3 vs. 1	34.589	11.99161	S
3 vs 2	0.273	11.3756	-
2 vs 1	34.316	11.3756	S

population size = 60 40

Crossover type

Mse= 3238.1 n= 96 Syi= 5.80777

Order	Ctype	mean	Select
1	1	543.192	
2	6	602.073	
3	3	602.663	
4	5	602.887	
5	2	608.962	
6	4	710.555	/

VS.	diff.	R	Result
6 vs.1	167.363	18.29448	S
6 vs.2	108.482	17.94601	S
6 vs.3	107.892	17.53947	S
6 vs.4	107.668	16.95869	S
6 vs.5	101.593	16.08753	S
5 vs 1	65.77	17.94601	S
5 vs. 2	6.889	17.53947	-
5 vs. 3	6.299	16.95869	-

VS.	diff.	R	Result
5 vs.4	6.075	16.08753	-
4 vs.1	59.695	17.53947	S
4 vs.2	0.814	16.95869	-
4 vs.3	0.224	16.08753	-
3 vs.1	59.471	16.95869	S
3 vs.2	0.59	16.08753	-
2 vs.1	58.881	16.08753	S

crossover type = 4

Pm

Mse= 3238.1 n= 144 Syi= 4.74203

Order	Pm	mean	Select
1	0.1	577.094	
2	0.2	589.024	
3	0.4	604.172	
4	0.3	676.599	/

VS.	diff.	R	Result
4 vs.1	99.505	14.32092	S
4 vs.2	87.575	13.84671	S
4 vs.3	72.427	13.13541	S

VS.	diff.	R	Result
3 vs.1	27.078	13.84671	S
3 vs.2	15.148	13.13541	S
2 vs.1	11.93	13.13541	-

Pm = 0.3

Population Size * Crossover

Mse= 3238.1 n= 32 Syi= 10.0594

Fix crossover type = 4

Order	popsize	mean	Select
1	20	697.723	-
2	60	712.072	-
3	40	721.869	-

VS.	diff.	R	Result
3 vs. 1	24.146	29.37332	-
3 vs 2	9.797	27.86441	-
2 vs 1	14.349	27.86441	-

all levels of Population size are not significant at considering levels of crossover type

Population size*Pc

Mse= 3238.1 n= 144 Syi= 8.21343

Fix Population size = 40

Order	Pc	mean	Select
1	0.8	603.108	
2	1	620.295	/
3	0.7	632.01	/
4	0.9	637.94	/

VS.	diff.	R	Result
4 vs.1	34.832	24.80456	S
4 vs.2	17.645	23.98321	-
4 vs.3	5.93	22.7512	-

VS.	diff.	R	Result
3 vs.1	28.902	23.98321	S
3 vs.2	11.715	22.7512	-
2 vs.1	17.187	22.7512	-

When fix population size=40 Pc=0.7 0.9 1.0

Fix Population size = 60

Order	Pc	mean	Select
1	1	614.252	
2	0.7	617.319	/
3	0.9	628.007	/
4	0.8	632.702	/

VS.	diff.	R	Result
4 vs.1	18.45	24.80456	-
4 vs.2	15.383	23.98321	-
4 vs.3	4.695	22.7512	-

VS.	diff.	R	Result
3 vs.1	13.755	23.98321	-
3 vs.2	10.688	22.7512	-
2 vs.1	3.067	22.7512	-

all levels of Pc are not significant at considering level of population size

Crossover*Pm

Mse= 5.5722 n= 24 Syi= 0.48185

Fix crosstype = 4

Order	Pm	mean	Select
1	0.1	673.846	
2	0.2	695.027	
3	0.4	704.75	
4	0.3	768.594	/

VS.	diff.	R	Result
4 vs.1	94.748	5.275752	-
4 vs.2	73.567	5.101058	-
4 vs.3	63.844	4.839018	-

VS.	diff.	R	Result
3 vs.1	30.904	5.101058	-
3 vs.2	9.723	4.839018	-
2 vs.1	21.181	4.839018	-

When fix crossover type = 4 Pm= 0.3

Response = No. of Generation Founding Best Solution

Pc

Mse= 679.58 n= 4 Syi= 13.0344

Order	Pc	mean	Select
1	0.9	711.25	/
2	1.0	779	
3	0.7	795	

VS.	diff.	R	Result
3 vs. 1	83.75	46.66307	S
3 vs 2	16	45.09895	-
2 vs 1	67.75	45.09895	S

Pc = 0.9

ภาคผนวก ฉ-2
โปรแกรมสำหรับเจเนติกอัลกอริทึมแบบหลายวัตถุประสงค์

1. การใช้งานโปรแกรม

โปรแกรมที่สร้างขึ้น เขียนโดยโปรแกรมประยุกต์ MATLAB 5.3.0 ซึ่งในการใช้งานจำเป็นต้องมีโปรแกรมดังกล่าวเสียก่อน เมื่อเข้าสู่โปรแกรมแล้ว ให้พิมพ์คำว่า ALB เพื่อเข้าสู่ Main Program

Main Program (ALB.m) เป็น โปรแกรมที่สั่งให้ดำเนินงานเพื่อหาคำตอบของปัญหา ใน Main Program จะประกอบไปด้วย 4 ส่วนหลักคือ

1. ***Input data*** เป็นโปรแกรมสำหรับรับข้อมูลเข้าต่างๆ ซึ่งได้แก่
   - จำนวนขั้นงานทั้งหมดของปัญหา
   - รายละเอียดของแต่ละขั้นงาน ประกอบด้วย
     1) เวลาทำงานของขั้นงาน
     2) ขั้นงานที่ต้องทำก่อนหน้า ในการกรอกข้อมูลถ้าขั้นงานนั้นไม่มีงานก่อนหน้าให้กด "Enter" ผ่านไปได้ทันที แต่ถ้ามีงานก่อนหน้ามากกว่า 1 งานให้เขียนขั้นงานที่อยู่ก่อนหน้า ภายใน เครื่องหมาย [ ]

เมื่อกรอกข้อมูลของแต่ละขั้นงานเสร็จแล้ว โปรแกรมจะถามว่าข้อมูลที่ป้อนเข้าไปถูกต้องหรือไม่ ถ้าต้องการแก้ไขข้อมูลให้ตอบ 'n'

- รอบเวลาการผลิตสูงสุดที่กำหนด
- จำนวนสถานีทำงานยอมเผื่อให้

หลังจากกรอกข้อมูลเสร็จแล้ว โปรแกรมจะถามอีกครั้งว่าข้อมูลที่ป้อนเข้ามีความถูกต้องหรือไม่

2. ***Parameter Setting (paraset.m)*** เป็นโปรแกรมสำหรับกำหนดค่าพารามิเตอร์ที่จะใช้ ซึ่งประกอบไปด้วย

   - จำนวนประชากร
   - วิธีการครอสโอเวอร์ ให้เลือกวิธีการที่ 1 – 6
   - ความน่าจะเป็นในการการครอสโอเวอร์
   - ความน่าจะเป็นในมิวเตชั่น
   - จำนวนเจนเนอเรชั่นสูงสุด
   - น้ำหนักของแต่ละวัตถุประสงค์ จะต้องเขียนภายในเครื่องหมาย [ ]

   ภายหลังจากที่ทำการกำหนดพารามิเตอร์แล้ว โปรแกรมจะถามว่าต้องการแก้ไขหรือไม่ ถ้าตอบ 'n' โปรแกรมจะเข้าสู่ขั้นตอนถัดไปโดยอัตโนมัติ

3. ***GAs (GA.m)*** เป็นโปรแกรมสำหรับการดำเนินการหาคำตอบโดยวิธีการของ MOGA ซึ่งภายในโปรแกรมนี้จะประกอบไปด้วยคำสั่งหรือโปรแกรมย่อยต่างๆ โปรแกรมจะรันจนกว่าจะถึงเจนเนอเรชั่นสูงสุด แล้วจึงเข้าสู่กระบวนการต่อไปโดยอัตโนมัติ
4. ***Final Output (Finaloutput.m)*** เป็นโปรแกรมแสดงคำตอบที่หาได้ โดยจะแสดงบนหน้าจอของโปรแกรม Matlab
5. ***Reporting (GArun.m)*** เป็นโปรแกรมแสดงผลออกมา และเก็บไว้ในไฟล์ชื่อ "ALBanswer.m" สำหรับเก็บไว้เพื่อเปิดอ่านในภายหลัง

MAIN PROGRAM (ALB.m)

```
clc
clear all
disp('=================================================================================');
disp('            Welcome to ALB Solving Programme based on GAs                        ');
disp('                        By. Kannika  Silanon                                     ');
disp('=================================================================================');

disp(' ')
disp('  This programme used for solving ALB problem with multi criteria.');
disp('Before starting the programme please see the instruction below.');
disp(' ');
disp('Press anykey to continue...');
pause
disp(' ');
disp(' ');
disp('Instruction: ');
disp(' ');
disp('The programme will execute on 4 step .-');
disp(' ');

disp('Step 1: Input data');
disp(' ')
disp('   1.Input the following data');
disp('    - Total number of task');
disp('    - Each task detail');
disp('        Processing time of task');
disp('        Predecessor of task (No. of predecessor  have to type ');
disp('        in blanket [].If there is no predecessor task,type only [] )');
disp('    - Determined cycle time ');
disp('    - Maximun allowance number of station  the max no. which will ');
disp('       add to theoritical min n in order to set up max n) ');
disp('   2.After input all data,You will be asked to check correctness of data.If there
are incorrect,you have to edit them before go to next step')

disp(' ');
disp(' ');
disp('Step 2: Parameter setting')
disp(' ');
disp('   1. Setting the following parameters');
disp('    - Population size');
disp('    - Crossover type (choose 1-6)');
disp('    - Probability of crossover  between 0-1)');
disp('    - Probability of mutation (between 0-1 ');
disp('    - Maximun number of generations');
disp('    - Weight of each objective (total weight must not exceed 1)');
disp('   2. After setting parameter, Check correctness before starting GAs');
disp('   3. You will be asked to execute programme, If you type "y".,You ');
disp('     will go to Step 3.If not,You will be asked to quit the programme');
disp(' ');
disp(' ');
disp('Step 3: Execute the programme');
disp(' ');
disp('      The programme will execute automatically after you type "y"  when ');
disp('  asked in the last step,Please wait until the programme finish.');
disp(' ');
disp(' ');
disp('Step 4: Display solution');
disp(' ');
disp('      When the programme stop, You will see a solution of the problem.');
disp('  The solution is also saved in ALBanswer.m file ')
disp('      After completing all step, you will out of the programme automatically.');
disp('  To solve the next problem, type "ALB" to access the programme');
disp('');
disp('');
disp('=================================================================');
disp(' ')
stop='n';
```

```
while stop=='n'
   disp('Do you want to start the programme? (y/n)');
   logic=input(' ','s');

   if logic=='y'
      clc
      disp('=============================================================');
      disp('                ALB Solving Programme based on GAs           ');
      disp('=============================================================');
      disp(' ')
      disp('Step 1: Input data');
      disp(' ');
      inputdata2


      clc
      disp('=============================================================');
      disp('                ALB Solving Programme based on MOGA          ');
      disp('=============================================================');
      disp(' ')
      disp('Step 2: Input parameter');
      disp(' ')
      paraset

      clc
      disp('=============================================================');
      disp('                ALB Solving Programme based on MOGA          ');
      disp('=============================================================');
      disp(' ')
      disp('Step 3: Programme execute');
      disp(' ')
      disp('       Please wait while programme is running.....');
      GArun

      clc
      disp('=============================================================');
      disp('                ALB Solving Programme based on MOGA          ');
      disp('=============================================================');
      disp(' ')
      disp('Step 4: Display solution');
      disp(' ')
      finaloutput
      stop='y';
   else
      clc
      disp('   Are you sure to quit the programme?(y/n)');
      stop=input(' ','s');
      if stop=='y'
         clc
         disp('    Good bye')
      end
   end

end
```

GA program

```
%=========================================================================%
%                              GA script                                  %
%                  use for operating MOGA approach                        %
%                     last modified 5/1/42                                %
%=========================================================================%

gen=0;
time(1)=cputime;


% create initial population
createpopulation;

%evaluate initial population
[string]=evaluate(string,w,gen,time);
[sumn,sumct,sumwv]=setsum(string);
[string]=fitness(string,weight,sumn,sumct,sumwv);
[string]=allrank(string);
[keepset]=keepstring(string,weight);

gen=0;
%show graph
fit=NaN*ones(maxGen,1);
NoN=NaN*ones(maxGen,1);
WV=NaN*ones(maxGen,1);
CT=NaN*ones(maxGen,1);

fit(gen+1)=keepset(1).fitness;
NoN(gen+1)=keepset(1).Nostation;
CT(gen+1)=keepset(1).ct;
WV(gen+1)=keepset(1).wv;


subplot(2,2,1),plot((fit),'gs-');
xlabel('generation');ylabel('f(x)');
title('Fitness of Multi-objective GA');
text(0.5,0.95,['Best=  ',num2str(fit(gen+1))],'Unit','normalized');


subplot(2,2,2),plot((NoN),'r*-');
xlabel('generation');ylabel('No. of workstation');
title('No of workstation');
text(0.5,0.95,['min n=  ',num2str(NoN(gen+1))],'Unit','normalized');

subplot(2,2,3),plot((CT),'bo-');
xlabel('generation');ylabel('cycle time');
title('cycle time');
text(0.5,0.95,['min ct=  ',num2str(CT(gen+1))],'Unit','normalized');

subplot(2,2,4),plot((WV),'c+-');
xlabel('generation');ylabel('workload variance');
title('workload variance');
text(0.5,0.95,['min wv=  ',num2str(WV(gen+1))],'Unit','normalized');


%=========================================================================
% GA Loop

while gen<maxGen
   %selection (own method)
   [selchrom]=wheelselect(string,weight);

   %crossover
   crosschrom=crossover(selchrom,crosstype,Pc,%);

[chrom,bestcross]=subeval(crosschrom,task,w,lower_n,upper_n,weight,gen,time,ct_old,sumn,s
umct,sumwv);
```

```
    [keepset]=compare(bestcross,keepset,weight,sumn,sumct,sumwv);

    %mutation
    newchromosome=mutation(crosschrom,x,Pm);

    %evaluate

[mutchrom,bestmut]=subeval(newchromosome,task,w,lower_n,upper_n,weight,gen,time,ct_old,su
mn,sumct,sumwv);
    [string,keepset]=newpop(keepset,bestmut,mutchrom,weight,sumn,sumct,sumwv);

    clear newchromosome crosschrom mutchrom chrom
    % increment generational counter
    gen=gen+1;

    %update display convergence
    fit(gen+1)=keepset(1).fitness;
    NoN(gen+1)=keepset(1).Nostation;
    CT(gen+1)=keepset(1).ct;
    WV(gen+1)=keepset(1).wv;

    subplot(2,2,1),plot((fit),'gs-');
    xlabel('generation');ylabel('f(x)');
    title('Fitness of Multi-objective GA');
    text(0.5,0.95,['Best=  ',num2str(fit(gen+1))],'Unit','normalized');

    subplot(2,2,2),plot((NoN),'r*-');
    xlabel('generation');ylabel('No. of workstation');
    title('No of workstation');
    text(0.5,0.95,['min n=  ',num2str(NoN(gen+1))],'Unit','normalized');

    subplot(2,2,3),plot((CT),'bo-');
    xlabel('generation');ylabel('cycle time');
    title('cycle time');
    text(0.5,0.95,['min ct=  ',num2str(CT(gen+1))],'Unit','normalized');

    subplot(2,2,4),plot((WV),'c+-');
    xlabel('generation');ylabel('workload variance');
    title('workload variance');
    text(0.5,0.95,['min wv=  ',num2str(WV(gen+1))],'Unit','normalized');
    drawnow;

    %display result
    disp(' ')
    fprintf('**************** Generation:%g **************** \n',gen);


end
time(2)=cputime;
totaltime=time(2)-time(1);
```

Data input program

```
%============================================================%
%                    inputdata  script                       %
%           This script use for input initial input data      %
%                 last modified 5/1/42                        %
%============================================================%

clear all
editdata='on';
while editdata=='on'
   %input total task in assembly line
   disp(' ')
   m=input('    Enter total number of task:');
   disp(' ')

   %input data of each task
   condition='n';      %set initial condition for while-loop
   for i=1:m
      while condition=='n'
         task(i).name=i;
         disp(' ')
         fprintf('    Enter processing time of task %g:',task(i).name);
         task(i).time=input(' ');
         fprintf('    Enter predecessor of task %g:',i);
         task(i).predecessor=input('  ');

         % check correct of input data
         disp('                                          ')
         disp('Please check input data..');
         data=task(i)

         % Test-loop
         checkans='anykey';
         while checkans=='anykey'
            fprintf('     Is it correct?(y/n)\n');
            check=input(' ','s');
            if check=='n'
               disp('Please re-input ');
               checkans=0;
               condition='n';%if input data is wrong goto while-loop for re-input
               clc
            elseif check=='y'
               checkans=0;
               condition='n';
               clc
               break;
            elseif isempty(check)
               checkans='anykey';
               condition=1;
               disp(' ')
            else
               checkans='anykey';
               condition=1;
               disp(' ')
            end
         end
         if check=='y'
            break
            clc
         end
      end
   end


   % find total processing time
   w=0;

   for i=1:m
```

```
      w=w+task(i).time;
   end

% find cycle time

   pt=[task.time];      % matrix of processing time for all task
   disp(' ')
   disp(' ')
   iniCT=input('   Enter determined cycle time:\n');
   c=[iniCT max(pt)];
   ct_old=max(c);  %cycle time is max value between determined ct & max processing time

   %find lower and upper no. of station
   bound=input('   Enter maximum allowance of n:\n');
   lower_n=ceil(w/ct_old);     %minimum no. of station
   upper_n=lower_n+bound; %maximum acceptable no. of station

   clc
   disp('-----------------------------------------------------------------')
   disp('                      Summary of input data')
   disp('-----------------------------------------------------------------')
   fprintf('   1.Total task            : %g\n',m);
   fprintf('   2.Total processing time : %g\n',w);
   fprintf('   3.Cycle time            : %g\n',ct_old);
   fprintf('   4.Minimum no. of station: %g\n',lower_n);
   fprintf('   5.Maximum no. of station: %g\n',upper_n);
   disp('-----------------------------------------------------------------')

   checkans2='anykey';
   while checkans2=='anykey'
      disp(' ')
      disp('Do you want to edit input data?(y/n)');
      logic=input(' ','s');
      if logic=='n'
         editdata=0;
         checkans2=0;
      elseif logic=='y'
         editdata='on';
         checkans2=0;
         clc
         disp('Re-input data');
      elseif isempty(logic)
         checkans2=='anykey';
      else
         checkans2=='anykey';
      end
   end
end


% create precedence matrix

x=zeros(m,m);
for i=1:1:m
   bt=[task(i).predecessor];
   if ~isempty(bt)
      for a=1:length(bt)
         h=bt(a);
         x(h,i)=1;
      end
   end
end

% clear temporary variable
clear a h bound  c bt pt  i  n number answer condition predecessor check data j checkans
checkans2  editdata
```

Final output program

```
%=============================================================================%
%                                finaloutput script                           %
%                            used for reporting the solution                  %
%=============================================================================%

disp(' ');
disp('*************************************************************');
disp('     Measure of Performance')
disp('*************************************************************');
disp(' ')
fprintf('  Total Required Number of Workstation  :   %g\n',keepset(1).Nostation);
fprintf('  Cycle Time of Line          :   %g\n',keepset(1).ct);
fprintf('  Workload variance of Line   :   %g\n',keepset(1).wv);
fprintf('  Total Idle Time             :   %g\n',keepset(1).totalIdle);
fprintf('  Line Efficiency             :   %g %%\n',keepset(1).lineEff);
disp(' ')
disp('  Press "Enter" to continue...')
pause


disp(' ')
disp('*************************************************************');
disp('     Solution of Each Workstation'
disp('*************************************************************');

point=[];
taskinstation=[];

for i=1:keepset(1).Nostation
   disp(' ');
   clear taskinstation point
   point=find(([keepset(1).stationNO])==i );
   for j=1:length(point)
      taskinstation(j)=keepset(1).sequence point(j));
   end
   timeUsed=keepset(1).station(i).time;

   fprintf('   Workstation # %g\n',i);
   disp(' ');
   fprintf('       Tasks in Workstation   = %s\n',num2str(taskinstation));
   fprintf('       Workstation Time   = %g\n',timeUsed);
   disp(' ')
   pause
end
disp('         ************************************')
```

Reporting Program

```
fid=fopen('ALBanswer.m','w');


fprintf(fid,'==========================================================================\n'
);
   fprintf(fid,'              SUMMARY OF ALB SOLUTION\n');

fprintf(fid,'==========================================================================\n'
);
   %fprintf(fid,' \n');
   fprintf(fid,'   1.Total task            : %g\n',m);
   fprintf(fid,'   2.Total processing time : %g\n',w);
   fprintf(fid,'   3.Cycle time            : %g\n',ct_old);
   fprintf(fid,'   4.Minimum no. of station: %g\n',lower_n);
   fprintf(fid,'   5.Maximum no. of station: %g\n',upper_n);
   %fprintf(fid,' \n');
   fprintf(fid,'--------------------------------------------------------------------\n');
   fprintf(fid,'                      MOGA parameter\n');
   fprintf(fid,'--------------------------------------------------------------------\n');
   %fprintf(fid,' \n');
   fprintf(fid,'       Problem size   :%g tasks\n',m);
   fprintf(fid,'       Population size    :%g\n',pop_size);
   fprintf(fid,'       Crossover type :%g\n',crosstype);
   fprintf(fid,'       Prob. of crossover :%g\n',Pc);
   fprintf(fid,'       Prob. of mutation  :%g\n',Pm);
   fprintf(fid,'       No.of generation   :%g\n',maxGen);
   fprintf(fid,'       Weight  :%s\n',num2str(weight));
   %fprintf(fid,' \n');
   fprintf(fid,'--------------------------------------------------------------------\n');


      GA;


  fprintf(fid,'                              ALB Solution\n');
  fprintf(fid,'--------------------------------------------------------------------\n');
  %fprintf(fid,' \n');
  fprintf(fid,'Total No. of workstation      =%g\n',keepset(1).Nostation);
  fprintf(fid,'Cycle time     =%g\n',keepset(1).ct);
  fprintf(fid,'Workload variance      =%g\n',keepset(1).wv);
  fprintf(fid,'Total Idle Time        =%g\n',keepset(1).totalIdle);
  fprintf(fid,'Line efficiency        =%g\n',keepset(1).lineEff);
  fprintf(fid,'sequence       =%s\n',num2str(keepset(1).sequence));
  fprintf(fid,'station        =%s\n',num2str(keepset(1).stationNO));
  %fprintf(fid,' \n');
  fprintf(fid,'No.of Generation finding best solution       =%g\n',keepset(1).gen);
  fprintf(fid,'Time used for finding best solution(sec.)    =%g\n',keepset(1).cputime);
  fprintf(fid,'Total Time used (sec.)     =%g\n',totaltime);
  %fprintf(fid,' \n');
  fprintf(fid,'==========================================================\n ');

  fclose(fid);
```

Parameter Setting Program

```
% paraset.m is a script file used for setting GAs parameter


editsetting='on';
while editsetting=='on'

% define population size
disp(' ')
pop_size=input('   Enter population size      :  ');



crosstype=[];
while isempty(crosstype)
    disp(' ')
    disp('  Choose crossover type ')
    disp('      type 1 : MOX ')
   disp('       type 2 : PMX with repair method ')
   disp('       type 3 : CX with repair method')
   disp('       type 4 : OX with repair method'
   disp('       type 5 : PBX with repair method')
   disp('       type 6 : OBX with repair method')
   disp('  ')
   crosstype=input('       type: ');
end
switch(crosstype)
case 1
   ctype='MOX';
case 2
   ctype='PMX';
case 3
   ctype='CX';
case 4
   ctype='OX';
case 5
   ctype='PBX';
case 6
   ctype='OBX';
end



Pc=[];
while isempty(Pc)| Pc>1
    disp(' ')
    Pc=input(' Enter crossover probability   : ');% crossover prob.
end

Pm=[];
while isempty(Pm)| Pm>1
    disp(' ')
    Pm=input(' Enter mutation probability: ');   %mutation prob.
end

maxGen=[];
while isempty(maxGen)
    disp(' ')
    maxGen=input(' Enter maximum generations : ');
end

disp(' ')
u=[];
while isempty(u) | u>1
   weight(1)=input('   Enter weight of objective1: Min no. of workstation   :  ');
   u=weight(1);
end
```

```
v=[];
while isempty(v) | v> (1-u)
   weight(2)=input('   Enter weight of objective2: Min cycle time       :  ');
   v=weight(2);
end

weight(3)=1-weight(1)-weight(2);
fprintf('  Weight of objective3: Min workload variance      :  %g\n',weight(3));
disp(' ')
disp('Press "Enter" to continue...')
pause


clc
    disp('-----------------------------------------------------------------')
   disp('                      Summary of parameter setting')
   disp('-----------------------------------------------------------------')
   fprintf('       1.Population size  : %g\n',pop_size);
   fprintf('       2.Crossover type   : %s\n',ctype);
   fprintf('       3.Prob. of crossover   : %g\n',Pc);
   fprintf('       4.Prob. of mutation    : %g\n',Pm);
   fprintf('       5.Maximum Generation   : %g\n',maxGen);
   fprintf('       6.Weight of objective  : %s\n',num2str(weight));
   disp('-----------------------------------------------------------------')

   checkans2='anykey';
   while checkans2=='anykey'
      disp(' ')
      disp('Do you want to edit parameter setting?(y/n)');
      logic=input(' ','s');
      if logic=='n'
         editsetting=0;
         checkans2=0;
      elseif logic=='y'
         editsetting='on';
         checkans2=0;
         clc
         disp('reset parameter');
      else checkans2=='anykey';
      end
   end
end

clear logic checkans2 editsetting u v
```

Sub-program

```
%=========================================================================%
%                              allrank function                           %
%          used for creating different initial string solutions           %
%                              modified 19/9/42                           %
%=========================================================================%

function [string]=allrank(string)


beforerank=[string.Nostation];
afterrank=ranking2(beforerank,0);

for i=1:length(beforerank);
   string(i).nrank=afterrank(i);
end

beforerank=[string.ct];
afterrank=ranking2(beforerank,0);
for i=1:length(beforerank)
   string(i).ctrank=afterrank(i);
end

beforerank=[string.wv];
afterrank=ranking2(beforerank,0);
for i=1:length(beforerank)
   string(i).wvrank=afterrank(i);
end

beforerank=[string.fitness];
afterrank=ranking2(beforerank,1);
for i=1:length(beforerank)
   string(i).fitnessrank=afterrank(i);
end


%=========================================================================%
%                              assign function                            %
%                    use for represent solution into string form          %
%                              last modified 26/8/42                      %
%=========================================================================%
function solution=assign(m,x)
% Assign : create task sequence for string solution.


solution=zeros(1,m);
premetrix=x;          %x is precedence metrix
index=sum(x);
% assign task to one string
for i=1:m

   % find set of no predecessor
   nop=find(index==0);

   %assign task without predecessor to string

   k=round(rand*length(nop));
   while k==0
      k=round(rand*length(nop));
   end
   solution(i)=nop(k);

   %find new set of task without predecessor

   premetrix(:,solution(i))= NaN;   %delete task which was assigned
   premetrix(solution(i),:)=0;
   index=sum(premetrix);
end
```

```
%=========================================================================%
%                     checkn function for multi-objective                 %
%              used for create different initial string solutions         %
%                              modified 19/9/42                           %
%=========================================================================%

function [b,a]=checkN(stringNo,string,upper_n)

if string(stringNo).Nostation>upper_n
   b=0;
   a=0;
   break
else
   b=1;
   a=checksolution(stringNo,string);  %call checksolution function
end


%=========================================================================%
%                           checksolution  function                       %
%                              modified 5/11/42                           %
%=========================================================================%

function a=checksolution(stringNo,string)

for before=1:(stringNo-1)
   if string(stringNo).Nostation==string(before).Nostation
      y=zeros(1,string(stringNo).Nostation);
      for i=1:string(stringNo).Nostation
         currentsol=sort(string(stringNo).station(i).task);
         beforesol=sort(string(before).station(i).task);
         y(i)=isequal(currentsol,beforesol);
      end

      if all(y==1)
         a=0;
         break
      else a=2;
      end
   else a=2;
   end
end


%=========================================================================%
%                      compare function for multi-objective               %
%   used for comparing best string from crossover and elite preserve solution %
%                              modified 19/9/42                           %
%=========================================================================%
function [keepset]=compare(bestcross,keepset,weight)

dual(1)=bestcross(1);
dual(2)=keepset(1);
[dual]=fitness(dual,weight);

if dual(1).fitness > dual(1).fitness
   keepset(1)=bestcross(1);
end

if bestcross(2).Nostation<keepset(2).Nostation
   keepset(2)=bestcross(2);
end

if bestcross(3).ct<keepset(3).ct
   keepset(3)=bestcross(3);
end

if bestcross(4).wv<keepset(4).wv
   keepset(4)=bestcross(4);
end
```

```
%=========================================================================%
%                createpopulation script for multi-objective              %
%             used for create different initial string solutions          %
%                          modified 19/9/42                               %
%=========================================================================%

% create the first string

stringNo=1;
string(stringNo).sequence=assign(m,x) ;
sequence=string(stringNo).sequence;
b=0;
while b==0;
    %disp('random n for string1')
    [string(stringNo).ct]=randNoSt(lower_n,upper_n,w,task);
    [string]=decode(sequence,task,string(stringNo).ct);
    if string(stringNo).Nostation>upper_n
      b=0;
      break
    else b=1;
    end
end


%create other different strings
for stringNo=2:pop_size
   a=0;
   while a==0
      c=0;
      b=0;
      string(stringNo).sequence=assign(m,x);
      sequence=string(stringNo).sequence;
      while b==0
         c=c+1;
         if c<7
            [string(stringNo).ct]=randNoSt(lower_n,upper_n,w,task);
         else
            [string(stringNo).ct]=ct_old;
         end
         % call findN  for finding total no. of station

[string(stringNo).stationNO,string(stringNo).station,string(stringNo).Nostation,string
(stringNo).stationtime]=findN(sequence,task,string(stringNo).ct);
         [b,a]=checkN(stringNo,string,upper_n);
      end
   end
end

clear  a b  sequence stringNo


%=========================================================================%
%                              cross function                             %
%             used for performing Modified one-point crossover (type 1)   %
%                              modified 19/9/42                           %
%=========================================================================%

function [pair]=cross(pair)
%cross function used to perform specific crossover method(type 1)

[pair]=xpoint(pair,1);     %use xpoint function to find crossover point

for j=1:size(pair,2)
    parent11=pair(j).parent1;
    parent22=pair(j).parent2;
    pair(j).offspring1=zeros(1,length(parent11));
    pair(j).offspring2=zeros(1,length(parent22));


   for i=1:pair(j).xp
      % head of offspring1 comes from parent1
```

```
    pair(j).offspring1(i)=pair(j).parent1(i);
    del1=find(pair(j).parent2==pair(j).offspring1(i));
    parent22(del1)=0;
      % head of offspring2 comes from parent2
    pair(j).offspring2(i)=pair(j).parent2(i);
    del2=find(pair(j).parent1==pair(j).offspring2(i));
    parent11(del2)=0;
   end

   % tail of offspring1 comes from remain task in parent2
   [x,y,tail1]=find(parent22);
   % tail of offspring2 comes from remain task in parent1
   [xx,yy,tail2]=find(parent11);

   % create complete offspring
    for c=1:length(tail1)
      pair(j).offspring1(pair(j).xp+c)=tail1(c);
      pair(j).offspring2(pair(j).xp+c)=tail2(c);
   end

end


%=========================================================================%
%                          crossover function
%                       for operating crossover
%                       last modified 21/10/42
%=========================================================================%

function crosschrom=crossover(selchrom,crosstype,Pc,x)
% crossover function used for crossover performing(multi-objective)

%call function paired
[pair,remstring]=paired(selchrom,Pc);

%perform crossover
switch (crosstype)
case 1
   [pair]=cross(pair);
case 2
   [pair]=PMX(pair);
case 3
   [pair]=CX(pair);
case 4
   [pair]=OX(pair);
case 5
   [pair]=PBaseX(pair);
case 6
   [pair]=OBaseX(pair);
end


for  a=1:size(pair,2)
   crosschrom((2*a)-1,:)=pair(a).offspring1;
   crosschrom(2*a,:)=pair(a).offspring2;
end

[Nchro,Lind]=size(crosschrom);
for b=1:length(remstring)
      crosschrom((Nchro+b),:)=selchrom(remstring(b)).sequence;
end

%check feasibility of sequence from  standard crossover operator
if crosstype==2| crosstype==3 | crosstype==4 | crosstype==5 |crosstype==6
    for k=1:Nchro
        solution=crosschrom(k,:);
        a=feasible(solution,x);                  %use feasible.m function
        while a==0
            infeasible=solution;
            solution=repair(infeasible,x);   % use repair.m function
            a=feasible(solution,x);
```

```
        end
        crosschrom(k,:)=solution;
    end

end

clear selchrom


%=========================================================================%
%                                CX function                              %
%              used for performing cycle crossover (type 3)               %
%                                modified 3/11/42                         %
%=========================================================================%

function [pair]=CX(pair)
%CX function used to perform standard crossover operator "Cycle crossover"

Npair=size(pair,2);

for j=1:Npair
    parent1=pair(j).parent1;
    parent2=pair(j).parent2;

    pair(j).offspring1=zeros(1,length(parent1));
    pair(j).offspring2=zeros(1,length(parent1));
    position=1;
    value=parent2(1);

    while  value~=parent1(1)
       pair(j).offspring1(position)=parent1(position);
       pair(j).offspring2(position)=parent2(position);
       [position]=find(parent1==value);
       value=parent2(position);
    end
    pair(j).offspring1(position)=parent1(position);
    pair(j).offspring2(position)=parent2(position);

    [pos]=find(pair(j).offspring1==0);
    for i=1:length(pos)
       pair(j).offspring1(pos(i))=parent2(pos(i));
       pair(j).offspring2(pos(i))=parent1(pos(i));
    end
 end



%=========================================================================%
%                           data script                                   %
%                  to input initial input data                            %
%                      last modified 5/11/42                              %
%=========================================================================%

clear all

%input total task in assembly line
m=input('Enter total number of task:');

%input data of each task
condition='n';      %set initial condition for while-loop
for i=1:m
   while condition=='n'
   task(i).name=i;
   fprintf('    Enter processing time of task %g:',task(i).name);
   task(i).time=input(' ');
   fprintf('    Enter number of predecessor of task %g:',i);
   number=input(' ');
   if number~=0
      predecessor=zeros(1,number);
```

```
      for j=1:number
         fprintf(       'Enter predecessor# %g of task %g:',j,i);
         predecessor(j)=input(' ');
      end
      task(i).predecessor=predecessor;
   else
      task(i).predecessor=[];
   end

   % check correct of input data
   disp('                                          ')
   data=task(i)
   fprintf('     Is it correct?(y/n)\n');
   check=input(' ','s' ;
   if check=='n'
      condition='n';       %if input data is wrong goto while-loop for re-input
   elseif check=='y'
      break
   end
   end

end


% find total processing time
w=0;
for i=1:m
   w=w+task(i).time;
end

% find cycle time
pt=[task.time];     % matrix of processing time for all task
iniCT=input('   Enter determined cycle time:\n');
c=[iniCT max(pt)];
ct_old=max(c);  %cycle time is max value between determined ct & max processing time

%find lower and upper no. of station
bound=input(' Enter maximum allowance of n:\n');
lower_n=w/ct_old;      %minimum no. of station
upper_n=lower_n+bound; %maximum acceptable no. of station

disp('-----------------------------------------------------------------')
disp('                      Summary of input data')
disp('-----------------------------------------------------------------')
fprintf('   Total task is %g\n',m);
fprintf('   Total processing time is %g\n',w);
fprintf('   Cycle time is:%g\n',ct_old);
fprintf('   Minimum no. of station:%g\n',lower_n);
fprintf('   Maximum no. of station:%g\n',upper_n);

% create precedence matrix

x=zeros(m,m);
for i=1:1:m
   bt=[task(i).predecessor];
   if ~isempty(bt)
      for a=1:length(bt)
         h=bt(a);
         x(h,i)=1;
      end
   end
end
 %show precedence metrix
 x

% clear temporary variable
clear a h bound  c bt pt  i  n number answer condition predecessor check data j
```

```
%=========================================================================%
%                              decode function                            %
%          to calculate  No. of workstation for all string in generation1-maxGen
%                            last modified 27/8/42                        %
%=========================================================================%

function [string]=decode(popsequence,task,ct)
% findNst function used for calculating No. of workstation for generation1-maxGen

[Nind,Lind]=size(popsequence);

for stringNo=1:Nind
   % set initial value of variable
   c=1;n=1;
   time=0;
   sequence=popsequence(stringNo,:);
   string(stringNo).sequence=sequence;
   string(stringNo).ct=ct;
   string(stringNo).stationNO=zeros(1,Lind);

   % calculate total time of each station
   while c<= Lind
      i=sequence(c);              % find task name
      string(stringNo).station(n).time=time+task(i).time;   %add time to station time
      time=string(stringNo).station(n).time;
      %check whether stationtime is less than cycle time
      if string(stringNo).station(n).time<=ct
         string(stringNo).stationNO(c)=n;
         c=c+1;        % goto while loop again
      else
         %cal station time for station n
         string(stringNo).station(n).time=time-task(i).time;

         % prepare new station for assign task #C
         n=n+1;
         time=0;
      end %end of if-loop
   end         %goto while-loop for assign task#c to station
   string(stringNo).Nostation=n;
   for k=1:string(stringNo).Nostation
      string(stringNo).stationtime(k)=string(stringNo).station(k).time;
   end
   %find task in each station
   for h=1:string(stringNo).Nostation
      position=find(string(stringNo).stationNO==h);
      string(stringNo).station(h).task=sequence(position);
   end
end


%=========================================================================%
%                              diffRand function                          %
%             to  random different value in definite range                %
%                              modified 19/9/42                           %
%=========================================================================%
function diffInt=diffRand(maxvalue,Nrand)
% diffRand:create different random number in definite range

for i=1:maxvalue
   number(i)=i;
end

for j=1:Nrand
    pos=0;
    while pos==0
      pos=round(rand*length(number));
   end
   diffInt(j)=number(pos);
   number(pos)=0;
   [x1,y1,number]=find(number);
end
```

```
%=========================================================================================%
%                                  evaluate function                                      %
%                           to calculate measure of performance of each string            %
%                                  modified 19/9/42                                       %
%=========================================================================================%

function [string]=evaluate(string,w,gen,time)
% evaluate: calculate ct,wv, line efficiency of each solution


pop_size=size(string,2);
for stringNo=1:pop_size
   realCT=max([string(stringNo).station.time]);
   for n=1:string(stringNo).Nostation
      string(stringNo).station(n).idle=realCT-string(stringNo).station(n).time;
      string(stringNo).station(n).eff=string(stringNo).station(n).time*100/realCT;
   end

   %cal cycle time
   string(stringNo).ct= realCT;

   % cal workload variance
   string(stringNo).wv=wvariance(string,stringNo,w);  % call wvariance function

   %cal total idle time
   string(stringNo).totalIdle=sum([string(stringNo).station.idle]);

   %cal line efficiency

string(stringNo).lineEff=sum([string(stringNo).station.eff])/string(stringNo).Nostation;
    string(stringNo).gen=gen+1;
   clear time(3)
   time(3)=cputime;
   foundtime=time(3)-time(1);
   string(stringNo).cputime=foundtime;

end


%=========================================================================================%
%                                  feasible function                                      %
%                           to check feasibility of solution                              %
%                                  modified 19/9/42                                       %
%=========================================================================================%

function a=feasible(solution,x)
% Feasible function use for checking feasibility of solution

for i=2:length(solution)
   for j=1:(i-1)
      if x(solution(i),solution(j))==1
         a=0;  %whenever  some task break precedence constraint,stop checking & exit
function
         break
      else
         a=1;  % if  solution string is feasible,exit this function  with a=0
      end
   end

   if a==0     % if  solution string is infeasible,exit this function  with a=0
      break
   end
end



%=========================================================================================%
```

```
%                              findN function
%                           used for decoding string
%                           last modified 27/8/42
%=============================================================================

function [stationNO,station,Nostation,stationtime]=findN(sequence,task,ct)
%FindN function used for calculating number of workstations

% set initial value of variable
c=1;n=1;
time=0;
stationNO=zeros(1,length(sequence));

% calculate total time of each station
while c<= length(sequence)
   i=sequence(c);              % find task name
   station(n).time=time+task(i).time; %add time to station time
   time=station(n).time;

   %check whether stationtime is less than cycle time
   if station(n).time<=ct
      stationNO(c)=n;
      c=c+1;        % goto while loop again
   else
      %cal station time for station n
      station(n).time=time-task(i).time;

      % prepare new station for assign task #C
      n=n+1;
      time=0;
   end %end of if-loop
end            %goto while-loop for assign task#c to station

Nostation=n;
for k=1:Nostation
   stationtime(k)=station(k).time;
end

%find task in each station

for h=1:Nostation
   position=find(stationNO==h);
   station(h).task=sequence(position);
end


%=============================================================================%
%                              fitness function                               %
%                        to calculate fitness value of string                 %
%                              modified 10/12/42                              %
%=============================================================================%

function [string]=fitness(string,weight)
%Fitness: used for calculating fitness value of string

alln=[string.Nostation];
sumn=sum(alln);

allct=[string.ct];
sumct=sum(allct);

allwv=[string.wv];
sumwv=sum(allwv);

popsize=size(string,2);
for i=1:popsize
   string(i).fitness=(weight(1)*sumn/string(i).Nostation)+(weight(2)*sumct/string(i).ct)+
(weight(3)*sumwv/string(i).wv ;
end
```

```
%=========================================================================%
%                          fitness2 function                              %
%                  to calculate fitness value of compared string          %
%                          modified 10/12/42                              %
%=========================================================================%

function [keepset]=fitness2(keepset,SUMn,SUMct,SUMwv,weight)

old=keepset(1).fitness
keepset(1).fitness=(weight(1)*SUMn/keepset(1).Nostation)+(weight(2)*SUMct/keepset(1).ct)+
(weight(3)*SUMwv/keepset(1).wv);
new=keepset(1).fitness
```

```
%=========================================================================%
%                          keepset function                               %
%                  to find best string from crossover or mutation         %
%                          modified 19/12/42                              %
%=========================================================================%

function keep=keepbest(stringrank,string)
%Keepset: used for finding  new best string from crossover or mutation

best=find(stringrank==1);
if length(best)==1
   keep=string(best);
else
   for i=1:length(best)
      fit(i)=string(best(i)).fitnessrank;
   end
   position=find(fit==min(fit));
   if length(position)>1
      xx=0;
      while xx==0
         xx=round(rand*length(xx));
      end
      dummy=position(xx);
      clear position
      position=dummy;
   end
   keep=string(best(position));
end
```

```
%=========================================================================%
%                          keepstring function                            %
%                  to create set of elite preserve solution               %
%                          modified 10/12/42                              %
%=========================================================================%

function [keepset]=keepstring(string,weight)
%Keepstring: to create set of elite preserve solution

bestfit=find([[string.fitnessrank]]==1);
if length(bestfit)==1
   keepset(1)=string(bestfit);
else
   ObjRank=ranking2(weight,1);
   ObjNo=find(ObjRank==1);
   if length(ObjNo)>1
      point=0;
      while point==0
         point=round(rand*length(ObjNo));
      end
```

```
      dummy=ObjNo(point);
      clear ObjNo
      ObjNo=dummy;
   end
   for k=1:length(bestfit)
      switch(ObjNo)
      case 1
         fit(k)=string(bestfit(k)).nrank;
      case 2
         fit(k)=string(bestfit(k)).ctrank;
      case 3
         fit(k)=string(bestfit(k)).wvrank;
      end
   end

   bestfit2=find(fit==min(fit));
   if length(bestfit2)==1
      keepset(1)=string(bestfit(bestfit2));
   else
      yy=0;
      while yy==0
         yy=round(rand*length(bestfit2));
      end
      keepset(1)=string(bestfit(bestfit2(yy)));
   end
end


Nind=size(string,2);
stringrank=NaN*ones(1,Nind);


for k=1:3
   switch(k)
   case 1
      stringrank=[string.nrank];
   case 2
      stringrank=[string.ctrank];
   case 3
      stringrank=[string.wvrank];
   end
   keepset(k+1)=keepbest(stringrank,string);
end


%=========================================================================================%
%                                  mutation function                                      %
%                                 to operate mutation                                     %
%                                  modified 10/12/42                                      %
%=========================================================================================%

function newchromosome=mutation(crosschrom,x,Pm)
%Mutation: to mutate string by random-sequence mutation

[Nind,Lind]=size(crosschrom);
for i=1:Nind
   if rand<=Pm
      mutstatus(i)=i;
   else
      mutstatus(i)=0;
   end
end
mutchrom=find(mutstatus~=0);%mutchrom show chromosome No. which will be performed
mutation

% create head of mutated chromosome
originalx=x;
newchromH=0;
for k=1:length(mutchrom)
   x=originalx;  %set original precedence metrix
   clear newchromH
```

```
   newchrom(k).oldsequence=crosschrom(mutchrom(k),:);
   individual=crosschrom(mutchrom(k),:);

 % random to find mutation point
   if round(rand)==0
      newchrom(k).point=0;
   else
      newchrom(k).point=round(rand*Lind);
       while  newchrom(k).point==Lind
           newchrom(k).point=round(rand*Lind);
       end
   end
   if newchrom(k).point~=0
      newchromH(1:newchrom(k).point)=individual(1:newchrom(k).point);
      % delete assigned task from precedence metrix
      for j=1:length(newchromH)
         x(:,newchromH(j))=NaN;
         x(newchromH(j),:)=0;
      end

      %prepare variable for assign function
      m=Lind-newchrom(k).point;

      %create tail of mutated chromosome
      newchromT=assign(m,x);
      newchrom(k).newsequence=[newchromH newchromT];% mutated chromosome
   elseif newchrom(k).point==0
      m=Lind;
      newchrom(k).newsequence=assign(m,x);
   end
end

%replace oldsequence with newsequence
newchromosome=crosschrom;
for a=1:length(mutchrom)
   newchromosome(mutchrom(a),:)=newchrom(a).newsequence;
end


%=================================================================================%
%                                  newpop function                                %
%    to compare and replace best string from mutation and elite preserve solution %
%                                  modified 10/12/42                              %
%=================================================================================%

function [string,keepset]=newpop(keepset,bestmut,mutchrom,weight)
 %if max fitness from mutation is better than current best fitness,replace current best
fitness with the better one

 ObjRank=ranking2(weight,1);
 ObjNo=find(ObjRank==1);
 if length(ObjNo)>1
    point=0;
    while point==0
       point=round(rand*length(ObjNo));
    end
    dummy=ObjNo(point);
    clear ObjNo
    ObjNo=dummy;
 end


dual(1)=bestmut(1);
dual(2)=keepset(1);
[dual]=fitness(dual,weight);

 if dual(1).fitness>dual(2).fitness
    keepset(1)=bestmut(1);
    replaced=[];
 else
    replacefit=find([mutchrom.fitness]==min([mutchrom.fitness]));
```

```
   if length(replacefit)==1
      mutchrom(replacefit)=keepset(1);
      replaced(1)=replacefit;
   else
      for i=1:length(replacefit)
         switch(ObjNo)
         case 1
            value(i)=mutchrom(replacefit(i)).Nostation;
         case 2
            value(i)=mutchrom(replacefit(i)).ct;
         case 3
            value(i)=mutchrom(replacefit(i)).wv;
         end
      end
      position=find(value==max(value));
      if length(position)>1
         a=0;
         while a==0
            a=round(rand*length(position));
         end
         dummy=position(a);
         clear position
         position=dummy;
      end
      replaced(1)=replacefit(position);
      mutchrom(replaced(1))=keepset(1);
   end
end


%replacement by objective value
set=0;
for i=1:3
    Noreplace=length(replaced);
   if i==1
      Obj=ObjNo;
   else
      Obj=find(ObjRank==min(ObjRank));
      if length(Obj)>1
         point=0;
         while point==0
            point=round(rand*length(Obj));
         end
         dummy=Obj(point);
         clear Obj
         Obj=dummy;
      end
   end

   switch(Obj)
   case 1
      new=bestmut(2).Nostation;
      old=keepset(2).Nostation;
   case 2
      new=bestmut(3).ct;
      old=keepset(3).ct;
   case 3
      new=bestmut(4).wv;
      old=keepset(4).wv;
   end

   if new==old
      xdual(1)=bestmut(Obj+1);
      xdual(2)=keepset(Obj+1);
      [xdual]=fitness(xdual,weight);
      if xdual(1).fitness>=xdual(2).fitness
         a=1;
      else
         a=0;
      end
```

```
    elseif new<old
       a=1;
    else
       a=0;
    end



    if a==1
       keepset(Obj+1)=bestmut(Obj+1);
    elseif a==0
       replacepoint=replacement(replaced,mutchrom,Obj);
       if replacepoint~=0
          if length(replacepoint)==1
             mutchrom(replacepoint)=keepset(Obj+1);
             replaced(Noreplace+1)=replacepoint;
          else
             clear set
             for k=1:length(replacepoint)
                set(k)=mutchrom(replacepoint(k)).fitnessrank;
             end
             point=find(set==max(set));
             if length(point)>1
                pos=0;
                while pos==0
                   pos=round(rand*length(point));
                end
                dummy=point(pos);
                clear point
                point=dummy;
             end
             replaced(Noreplace+1)=replacepoint(point);
             mutchrom(replacepoint(point))=keepset(Obj+1);
          end
       end
    end
    ObjRank(Obj)=NaN;
 end

 Nind=size(mutchrom,2);
 for h=1:Nind
    string(h)=mutchrom(h);
 end

 clear mutchrom


%=========================================================================%
%                              OBaseX function                            %
%          used for performing order-based crossover (type 6)             %
%                              modified 3/11/42                           %
%=========================================================================%

function [pair]=OBaseX(pair)
% OBaseX function used to perform standard crossover operator "Order Based crossover"

Npair=size(pair,2);
Lind=length(pair(1).parent1);

for j=1:Npair
   Npoint=0;pos1=0;pos2=0;pos=0;
   parent11=pair(j).parent1;
   parent22=pair(j).parent2;
   clear pos1 pos pos2
    while Npoint==0 | Npoint==Lind
        Npoint=round(rand*Lind);
    end

    pos1=sort(diffRand(Lind,Npoint));
    for l=1:length(pos1)
      pos(l)=find(pair(j).parent2==pair(j).parent1(pos1(l)));
```

```
    end
    pos2=sort(pos);
    pair(j).offspring1=zeros(1,Lind);
    pair(j).offspring2=zeros(1,Lind);

    for i=1:length(pos1)
       pair(j).offspring1(pos2(i))=pair(j).parent1(pos1(i));
       [del2]=find(pair(j).parent2==pair(j).parent1(pos1(i)));
       parent22(del2)=0;
       pair(j).offspring2(pos1(i))=pair(j).parent2(pos2(i));
       [del1]=find(pair(j).parent1==pair(j).parent2(pos2(i)));
       parent11(del1)=0;
    end

    [blank1]=find(pair(j).offspring1==0);
    [blank2]=find(pair(j).offspring2==0);
    [xx,yy,remain1]=find(parent11);
    [xx,yy,remain2]=find(parent22);

    for k=1:length(blank1)
       pair(j).offspring1(blank1(k))=remain2(k);
       pair(j).offspring2(blank2(k))=remain1(k);
    end
end


%=========================================================================%
%                                OX function                              %
%              used for performing order crossover (type 4)               %
%                                modified 3/11/42                         %
%=========================================================================%

function  [pair]=OX(pair)
%OX function used to perform standard crossover operator "OX"

Npair=size(pair,2);
Lind=length(pair(1).parent1);
[pair]=xpoint(pair,2); %use xpoint function for finding 2 cross point

for j=1:Npair
    parent11=pair(j).parent1;
    parent22=pair(j).parent2;


   %perform crossover
   for i=pair(j).xp1:pair(j).xp2
       del1=find(pair(j).parent1==pair(j).parent2(i));
    parent11(del1)=0;
    del2=find(pair(j).parent2==pair(j).parent1(i));
    parent22(del2)=0;
    end

   parent11=repmat(parent11,1,2);
   parent22=repmat(parent22,1,2);

   [xx,yy,doublep1]=find(parent11(pair(j).xp1:pair(j).xp1+Lind-1));
   [xx,yy,doublep2]=find(parent22(pair(j).xp1:pair(j).xp1+Lind-1));

   add=1;
   for i=1:Lind
      if i>=pair(j).xp1 & i<=pair(j).xp2
         pair(j).offspring1(i)=pair(j).parent2(i);
         pair(j).offspring2(i)=pair(j).parent1(i);
      else
         pair(j).offspring1(i)=doublep1(add);
         pair(j).offspring2(i)=doublep2(add);
         add=add+1;
      end
   end
end
```

```
%=========================================================================
%                              paired function                           %
%                         to group two strings in pair                   %
%                              modified 2/12/42                          %
%=========================================================================

function   [pair,remstring]=paired(selchrom,Pc)
%Pair: to select strings which will be crossovered and group them in pair

condition=0;
Nind=size(selchrom,2);
while condition==1
   for i=1:Nind
      if rand<=Pc
         crossstatus(i)=i;
      else
         crossstatus(i)=0;
      end
   end   % crossstatus show individual no. that will be performed a crossover

   if rem(length(find(crossstatus)),2)==1
      checkstatus=find(crossstatus==0);
      if round(rand)==0    | isempty(checkstatus)          %cut off
         position=find(crossstatus~=0);
         pointdel=0;
         while pointdel==0
            pointdel=round(rand*length(position));
         end
         crossstatus(position(pointdel))=0;
      else
         position=find(crossstatus==0);
         pointadd=0;
         while pointadd==0
            pointadd=round(rand*length(position));
         end
         crossstatus(position(pointadd))=position(pointadd);
      end   %crosschro show individual no. that will be performed a crossover(even only)

   end
   crosschro=find(crossstatus);
   if length(crosschro)<2
      condition=0;
      clear crossstatus crosschro
   else
      condition=1;
   end
end

remstring=find(crossstatus==0);


%random task position for pairing
xchrom=crosschro;
clear crosschro
for h=1:length(xchrom)
   selpoint=0;
   while selpoint==0
      selpoint=round(rand*length(xchrom));
   end
   crosschro(h)=xchrom(selpoint);
   xchrom(selpoint)=0;
   [x,y,xchrom]=find(xchrom);
end

for k=1:2:length(crosschro)-1
   pair((k+1)/2).parent1=selchrom(crosschro(k)).sequence;
   pair((k+1)/2).parent2=selchrom(crosschro(k+1)).sequence;
```

```
end


%=========================================================================
%                          PBaseX function
%          used for performing position-based crossover (type 5)
%                          modified 3/11/42
%=========================================================================

function [pair]=PBaseX(pair)
% PBaseX function used to perform standard crossover operator "Position Based crossover"

Npair=size(pair,2);
Lind=length(pair(1).parent1);

for j=1:Npair
   Npoint=0;pos1=0;pos2=0;
   parent11=pair(j).parent1;
   parent22=pair(j).parent2;
   clear pos1 pos2
    while Npoint==0 | Npoint==Lind
        Npoint=round(rand*Lind);
    end

    pos1=diffRand(Lind,Npoint);
    for l=1:length(pos1)
      pos2(l)=find(pair(j).parent2==pair(j).parent1(pos1(l)));
   end

    pair(j).offspring1=zeros(1,Lind);
   pair(j).offspring2=zeros(1,Lind);

   for i=1:length(pos1)
      pair(j).offspring1(pos1(i))=pair(j).parent1(pos1(i));
      parent22(pos2(i))=0;
      pair(j).offspring2(pos2(i))=pair(j).parent2(pos2(i));
      parent11(pos1(i))=0;
   end

   [blank1]=find(pair(j).offspring1==0);
   [blank2]=find(pair(j).offspring2==0);
   [xx,yy,remain1]=find(parent11);
   [xx,yy,remain2]=find(parent22);

   for k=1:length(blank1)
      pair(j).offspring1(blank1(k))=remain2(k);
      pair(j).offspring2(blank2(k))=remain1(k);
   end
end


%=========================================================================
%                          PMX function
%          used for performing partial-mapped crossover (type 2)
%                          modified 3/11/42
%=========================================================================
function [pair]=PMX(pair)
%PMX function used to perform standard crossover operator "PMX"

Npair=size(pair,2);
Lind=length(pair(1).parent1);
[pair]=xpoint(pair,2);     %use xpoint function for finding 2 cross point

for j=1:Npair
   for i=1:Lind
       if i>=pair(j).xp1 & i<=pair(j).xp2
           pair(j).offspring1(i)=pair(j).parent2(i);
           pair(j).offspring2(i)=pair(j).parent1(i);

       else
           value1=pair(j).parent1(i);
```

```
            x1=find(pair(j).parent2(pair(j).xp1:pair(j).xp2)==value1);
            while ~isempty(x1)
                value1=pair(j).parent1(x1+pair(j).xp1-1);
                x1=find(pair(j).parent2(pair(j).xp1:pair(j).xp2)==value1);
            end
            if isempty(x1)
                pair(j).offspring1(i)=value1;
            end


            value2=pair(j).parent2(i);
            x2=find(pair(j).parent1(pair(j).xp1:pair(j).xp2)==value2);
            while ~isempty(x2)
                value2=pair(j).parent2(x2+pair(j).xp1-1);
                x2=find(pair(j).parent1(pair(j).xp1:pair(j).xp2)==value2);
            end
            if isempty(x2)
                pair(j).offspring2(i)=value2;
            end
        end
    end
end
%=============================================================================%
%                          popselect function                                 %
%       to calculate probability that each string will be selected            %
%                          last modified 5/8/42                               %
%=============================================================================%

function pselect=popselect(string,obj)
 %Popselect: calculate probability that each string will be selected


 pop_size=size(string,2);
 if obj==1
    for stringNo=1:pop_size
       pselect(stringNo)=string(stringNo).fitness/sum([string.fitness]);
    end

 else
    switch (obj)
    case 2
       value=[string.Nostation];
    case 3
       value=[string.ct];
    case 4
       value=[string.wv];
    end

    for i=1:pop_size
       diffvalue(i)=sum(value')-value(i);
    end
    alldiff=sum(diffvalue');

    for stringNo=1:pop_size
       pselect(stringNo)=diffvalue(stringNo)/alldiff;
    end

 end


%=============================================================================%
%                             randNost function                               %
%              used for random number of workstation                          %
%=============================================================================%

function [ct]=randNoSt(lower_n,upper_n,w,task)

NoSt=0;
secondUpper=upper_n-1;
while NoSt<lower_n | NoSt>secondUpper
   NoSt=(rand*secondUpper);
```

```
    if NoSt<lower_n
       temp=secondUpper-NoSt;
       NoSt=NoSt+(rand*temp);
    end
end

calCT=w/NoSt;
% matrix of processing time for all task
pt=[task.time];
c=[calCT max(pt)];
ct=max(c); %cycle time is max value between determined ct & max processing time


%=================================================================
%                          ranking2 function
%                           used for ranking
%=================================================================

function afterrank=ranking2(beforerank,condition)
% Ranking function used for ranking value
% beforerank is a set of value which need to be ranked
% condition  (0 or 1) is parameter which identified what condition of ranking is
%         if condition=0   the value will be ranked from min to max (for minimizing
objective--more less value ,more better it is
%         if condition=1   the value will be ranked from max to min


afterrank=zeros(1,length(beforerank));
prerank=sort(beforerank);

if condition==0
   i=1;rank=1;
   while i<=length(beforerank)
      order=find(beforerank==prerank(i));
      if length(order)==1
         afterrank(order)=rank;
         i=i+1;
      else
         for j=1:length(order)
           afterrank(order(j))=rank;
         end
         i=i+length(order);
      end
      rank=rank+1;
   end

elseif condition==1
   i=length(beforerank);
   rank=1;
     while i>=1
       order=find(beforerank==prerank(i));
       if length(order)==1
          afterrank(order)=rank;
          i=i-1;
       else
         for j=1:length(order)
            afterrank(order(j))=rank;
         end
         i=i-length(order);
        end
        rank=rank+1;
     end
else
   disp('error')
   isempty(afterrank);
end


%=================================================================
%                          repair function
%             used for reparing infeasible string
```

```
%=====================================================================%

function solution=repair(infeasible,x)
% Repair function used to repair infeasible solution

% set initial variable
solution=zeros(1,length(infeasible));
premetrix=x;
index=sum(x);
position=0;

for i=1:length(infeasible)
   nop=find(index==0);% find task without predecessor
   clear position
   for j=1:length(nop)
      position(j)=find(infeasible==nop(j));
   end

%assign task in nop to repaired solution string (follow the order in original sequence)
    solution(i)=infeasible(min(position));

%update precedence metrix
   premetrix(:,solution(i))=NaN;
   premetrix(solution(i),:)=0;
   index=sum(premetrix);
end


%=====================================================================%
%                          replacement function
%            used for finding string No. which would be replaced
%=====================================================================%


function solution=repair(infeasible,x)
% Repair function used to repair infeasible solution

% set initial variable
solution=zeros(1,length(infeasible));
premetrix=x;
index=sum(x);
position=0;

for i=1:length(infeasible)
   nop=find(index==0);% find task without predecessor
   clear position
   for j=1:length(nop)
      position(j)=find(infeasible==nop(j));
   end

%assign task in nop to repaired solution string (follow the order in original sequence)
    solution(i)=infeasible(min(position));

%update precedence metrix
   premetrix(:,solution(i))=NaN;
   premetrix(solution(i),:)=0;
   index=sum(premetrix);
end


function replacepoint=replacement(replaced,mutchrom,Obj)


replacepoint=[];x=0;
switch(Obj)
case 1
   x=max([mutchrom.nrank])+1;
case 2
   x=max([mutchrom.ctrank])+1;
case 3
   x=max([mutchrom.wvrank])+1;
```

```
end

while isempty(replacepoint)
   x=x-1;
   if x>0
      switch(Obj)
      case 1
         replacepoint=find(([mutchrom.nrank])==x);
      case 2
         replacepoint=find(([mutchrom.ctrank])==x);
      case 3
         replacepoint=find(([mutchrom.wvrank])==x);
      end

      for i=1:length(replaced
         for j=1:length(replacepoint)
            if replacepoint(j)==replaced(i)
               replacepoint(j)=0;
            end
         end
         [x,y,replacepoint]=find replacepoint);
      end
   else
      replacepoint=0;
   end
end


%=============================================================================%
%                          roulette function                                  %
%                      to select suitable string                              %
%                          last modified 5/8/42                               %
%=============================================================================%

function stringselected=roulette(string,pop_size,obj)


pselect=popselect(string,obj);    %call popselect function
upperbound=upperb(pselect);   %call upperb function

for i=1:pop_size
   select=find(upperbound>=rand );
   stringselected(i)=select(1);
end

%stringselected is matrix of selected string No.


%=============================================================================%
%                          setsum function                                    %
%          to calculate summation of each string from generation1             %
%                          last modified 5/8/42                               %
%=============================================================================%

function [sumn,sumct,sumwv]=setsum(string,weight)


alln=[string.Nostation];
sumn=sum(alln);

allct=[string.ct];
sumct=sum(allct);

allwv=[string.wv];
sumwv=sum(allwv);




%=============================================================================%
```

```
%                        subeval function                                %
%                  to decode and evaluate new string                     %
%                        last modified 5/10/42                           %
%=========================================================================%

function
[chrom,bestchrom]=subeval(crosschrom,task,w,lower_n,upper_n,weight,gen,time,ct_old)


[Nind,Lind]=size(crosschrom);
for k=1:Nind
   chrom(k).sequence=crosschrom(k,:);
end

for i=1:Nind;
   b=0;
   c=0;
   while b==0
      c=c+1;
      sequence=chrom(i).sequence;
      if c<7
         [chrom(i).ct]=randNo3t_lower_n,upper_n,w,task);
      else
         [chrom(i).ct]=ct_old;
      end

      [chrom(i).stationNO,chrom(i).station,chrom(i).Nostation,chrom(i).stationtime]=findN
(sequence,task,chrom(i).ct);
      b=checkNstation(i,chrom,upper_n);
   end

end


[chrom]=evaluate(chrom,w,gen,time);
[chrom]=fitness(chrom,weight);

[chrom]=allrank(chrom);
[bestchrom]=keepstring(chrom,weight);


%=========================================================================%
%                        upperb function                                 %
%              to prepare upper bound in roulette wheel                  %
%                        last modified 20/8/42                           %
%=========================================================================%

function upperbound=upperb(pselect)
% Upperb: prepare upper bound in roulette wheel

%disp('upperbound')
pop_size=length(pselect);
oldupper=0;
for stringNo=1:pop_size
   upperbound(stringNo)=oldupper+pselect(stringNo);
   oldupper=upperbound(stringNo);
end


%=========================================================================%
%                  wheelselect  function                                  %
%          to operate roulette wheel selection                            %
%                        modified 27/8/99                                 %
%=========================================================================%
function [selchrom]=wheelselect(string,weight)


pop_size=size(string,2);
y=1;selsize=0;
half=ceil(pop_size/2);
remain=pop_size-half;
```

```
for obj=1:4
   switch(obj)
   case 1
      Nostring(obj)=half;
   case 2
      Nostring(obj)=round(remain*weight(obj-1));
   case 3
      Nostring(obj)=round(remain*weight(obj-1));
   case 4
      Nostring(obj)=pop_size-sum(Nostring');
   end
   stringselected=roulette(string,Nostring(obj),obj);
   for i=1:length(stringselected)
      selchrom(selsize+i)=string(stringselected(i)');
   end
   selsize=size(selchrom,2);
end
```

```
%=================================================================%
%                        wvariance function                       %
%              to calculate workload variance of each string      %
%                          modified 5/8/42                        %
%=================================================================%

function wv=wvariance(string,stringNo,w)
%Wvariance: cal workload variance for each string

    variance=zeros(1,string(stringNo).Nostation);
    for i=1:string(stringNo).Nostation
        variance(i)=(string(stringNo).station(i).time-(w/string(stringNo).Nostation))^2;
    end
   wv=sum(variance')/string(stringNo).Nostation;
```

```
%=================================================================%
%                         xpoint function                         %
%                      to random crossover point                  %
%                          modified 5/8/42                        %
%=================================================================%

function [pair]=xpoint(pair,Npoint)

%xpoint function used to random crossover point
%pair = structure array from paired function
%Npoint = number of crossover point (1 or 2)

Lind=length(pair(1).parent1);
for j=1:size(pair,2)

    if Npoint==1
      pair(j).xp=0;
        while pair(j).xp==0 |pair(j).xp==Lind
        pair(j).xp=round(rand*Lind);
      end
   else
      pair(j).xp1=0;
        pair(j).xp2=0;
        while pair(j).xp1==pair(j).xp2 |pair(j).xp1==0 |(pair(j).xp1==1&pair(j).xp2==Lind)
            pair(j).xp1=round(rand*Lind);
            pair(j).xp2=round(rand*Lind);
        if pair(j).xp1 > pair(j).xp2
            dummy=pair(j).xp1;
            pair(j).xp1=pair(j).xp2;
            pair(j).xp2=dummy;
         end
      end
   end
end
```

ประวัติผู้ทำการวิจัย

นางสาวกรรณิกา ศิลานนท์ เกิดเมื่อวันที่ 20 เมษายน พ.ศ 2519 ที่อำเภอเมือง จังหวัดเชียงใหม่ สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตรบัณฑิต(เกียรตินิยมอันดับหนึ่ง) สาขาวิศวกรรมอุตสาหการ จากมหาวิทยาลัยเชียงใหม่ เมื่อปีพ.ศ 2541 จากนั้นได้เข้าศึกษาต่อในระดับบัณฑิตศึกษาที่ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยในปีเดียวกัน