รายการอ้างอิง

ภาษาไทย

กรมชลประทาน. 2536. รายงานการศึกษาความเหมาะสม และผลกระทบสิ่งแวดล้อมโครงการเขื่อนเก็บกักน้ำแม่น้ำป่าสัก จังหวัดสระบุรี และจังหวัดลพบุรี.

กรมชลประทาน. 2546. โครงการศึกษาเพื่อทำแผนหลักรองรับการพัฒนาแหล่งน้ำและปรับปรุงโครงการชลประทานสำหรับแผนฯ9 (ลุ่มน้ำป่าสัก).

กรมชลประทาน. 2546. การศึกษาเกณฑ์การกักเก็บน้ำเขื่อนป่าสักชลสิทธิ์.กลุ่มงานจัดสรรน้ำ สำนักอุทกวิทยาและบริหารน้ำ.

กรมอุตุนิยมวิทยา. 2536. การศึกษาข้อมูลฝนที่ตกในบริเวณพื้นที่ลุ่มแม่น้ำป่าสัก โครงการพระราชดำริตรวจวัดปริมาณน้ำฝนที่ตกในบริเวณพื้นที่ลุ่มแม่น้ำป่าสัก.

กัมปนาท ภักดีกุล. 2544. การประยุกต์ปัญญาประดิษฐ์ทางคอมพิวเตอร์ที่เลียนแบบทฤษฎีการคัดเลือกตามธรรมชาติของ Charles Darwin เพื่อจัดการทรัพยากรน้ำ. วารสารนิเวศวิทยา กันยายน-ธันวาคม 2544 : 22-35.

กัมปนาท ภักดีกุล. 2546. เจเนติกอัลกอริธึ่ม การประยุกต์พันธุกรรมคอมพิวเตอร์เพื่อการจัดการทรัพยากรน้ำ., ชลกร ฉบับ วันชูชาติ 4 มกราคม 2546, 128-144.

การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย. รายงานการศึกษาเรื่องการปฏิบัติการอ่างเก็บน้ำเขื่อนป่าสัก. กองอุตุและอุทกวิทยา ฝ่ายสำรวจ, 2542.

เจษฎา แก้วกัลยา (2535). เอกสารประกอบการเรียนวิชา Water Management, ภาควิชาวิศวกรรมทรัพยากรน้ำ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.

ณัฐวุฒิ เภสัชเวชกิจ.2546. การกำหนดเกณฑ์การจัดสรรน้ำแบบหลายเงื่อนไข สำหรับการจัดการอ่างเก็บน้ำ กรณีศึกษาอ่างเก็บน้ำเขื่อนป่าสักชลสิทธิ์. วิทยานิพนธ์ปริญญามหาบัณฑิต, มหาวิทยาลัยเทคโนโลยีมหานคร, กรุงเทพมหานคร.

ธนัญชัย ลีภักดิ์ปรีดา.2543. การหาค่าเหมาะสมที่สุด หลักการพื้นฐานและขั้นตอนวิธีการ สำนักพิมพ์มหาวิทยาลัยธรรมศาสตร์, กรุงเทพ

บัญชา ขวัญยืน. 2541. การวิเคราะห์ระบบเพื่อการวางแผนและการจัดการโครงการชลประทาน. มหาวิทยาลัย เกษตรศาสตร์ วิทยาเขตกำแพงแสน, นครปฐม.

บุญลือ คงชอบ (2546) . "การจัดการน้ำโครงการเขื่อนป่าสักชลสิทธิ์ ", วิศวกรรมศาสตร์มหาบัณฑิต, มหาวิทยาลัยเกษตรศาสตร์, กรุงเทพมหานคร.

วราวุธ วุฒิวณิชย์.2538. การจัดการน้ำขั้นสูง. ภาควิชาวิศวกรรมชลประทาน, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน, นครปฐม.

วราวุธ วุฒิวณิชย์. 2539. อุทกวิทยาประยุกต์. ภาควิชาวิศวกรรมชลประทาน คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน, นครปฐม.

สมฤทัย ทะสดวก. 2544. การวางแผนพื้นที่เพาะปลูกตามศักยภาพแหล่งน้ำด้วยเจนิติกแอลกอลิทึ่ม. การประชุมวิชาการวิศวกรรมโยธาแห่งชาติ ครั้งที่7 พฤษภาคม 2544 : WRE27 -33.

เสรี ศุภราทิตย์. 2545. การประยุกต์ใช้ Neuro-genetic algorithm ในการพยากรณ์และเตือนภัยน้ำท่วม. =ชลกรณ์ฉบับวันชูชาติ 4 มกราคม 2545 : 74-83.

อรอนงค์ วรรณราช.2546. การบริหารอ่างเก็บน้ำป่าสักชลสิทธิ์ โดยประยุกต์ใช้เทคนิคโครงข่ายใยประสาทเทียม. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมแหล่งน้ำ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2546

ภาษาอังกฤษ

Bhaktikul, K ,. (2001). "The development of a Genetic Algorithm for real time water allocation and water scheduling in complex irrigation systems". PHD thesis School of Civil and Environmental Engineering The University of Edinburgh, Edinburgh, UK.

Darwin, C. (1959) The Origin of species. University of Pennsylvania Press, Philadelphia ,

Fogel,L., A. Owens and M. Walsh. 1966. Artificial Intelligence through Simulated Evolution. Wiley, New York.

Goldberg,D.E.,and Kao,C.H. (1987)"Genetic algorithms in pipeline optimization." J.Cmop. in Civ. Engrg., ASCE, 1(2),128-141.

Goldberg,D.E. (1989). Genetic Algorithms in search optimization & machine learning. Addison-Wesley,Reading,Mass.USA.

Goldberg,D.E.,and K.Deb. (1989). "A comparative analysis of selection schemes used in genetic algorithms". Foundations of genetic algorithms , Morgan Kaufman,Sam Mateo, Calif.,69-93.

Holland,J.H. (1975). Adaptation in natural and artificial systems. MIT Press, Cambridge, Mass.

Koza, J.R. 1992. Genetic Programming, MIT Press, Cambridge,MA.

Michalewicz, Z.(1992). Genetic algorithms + data structures = evolution programs. Springer-Verlag, New York, Inc., New York.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Mitsuo, G. and Runwei C. 1997 Genetic algorithms and engineering design. A Wiley Interscience Publication., New York.

Moreira, Dilvan De Abreu. 1995 Agents: A Distributed Client/Server System For Leaf Cell Generation.Ph.D. Thesis, University of Kent at Canterbury.

Neelakantan, T.R. and N.V. Pundarikanthan. 2000. Neural Network-Based Simulation Optimization Model for Reservoir Operation. Journal of Water Resources Planning and Management, Vol. 126 (2): 57-64.

Rao, S.S 1983. Optimization: theory and applications. 2nd ed. Wiley Eastern, New Delhi.

Rechenberg, I. 1973. Evolutionstrategie: Optimieriung teachischer Systeme nach prinaipien der biologischen Evolution. Frommann-Holzboog, Stuttgart, Germany.

Reddy,J.M., Wilamowski,B,. and Sharmasarkar,F.C. (1999). "Optimal scheduling of irrigation for lateral canals." ICID J., 48(3),1-12

Schwefel,H. 19958. Evolution and Optimization Seeking. Wiley, New York.

Wang,Q.J.(1991). "The genetic algorithm and its application to calibrating comceptual rainfall-runoff models. "Water Resour. Res.,27(9),2467-2471.

Wardlaw,R.B., and Barnes, J.M. (1999) "Optimal allocation of irrigation water supplies in real time." J. Irri. and Drainage Engrg., ASCE 125(6),345-354.

Wardlaw,R., and Bhaktikul, K. (2001) "Application of a genetic algorithms for water allocation in an irrigation system. J. Irrigation and Drainage. ICID,CIID. John Wiley & Sons. 50(159-170).

Wardlaw, R.B., and Sharif,M.(1999)."Evaluation of genetic algorithms for optimal reservoir system operation". J. Water Resour. Plng. And Mgmt., ASCE 125(1),25-33.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

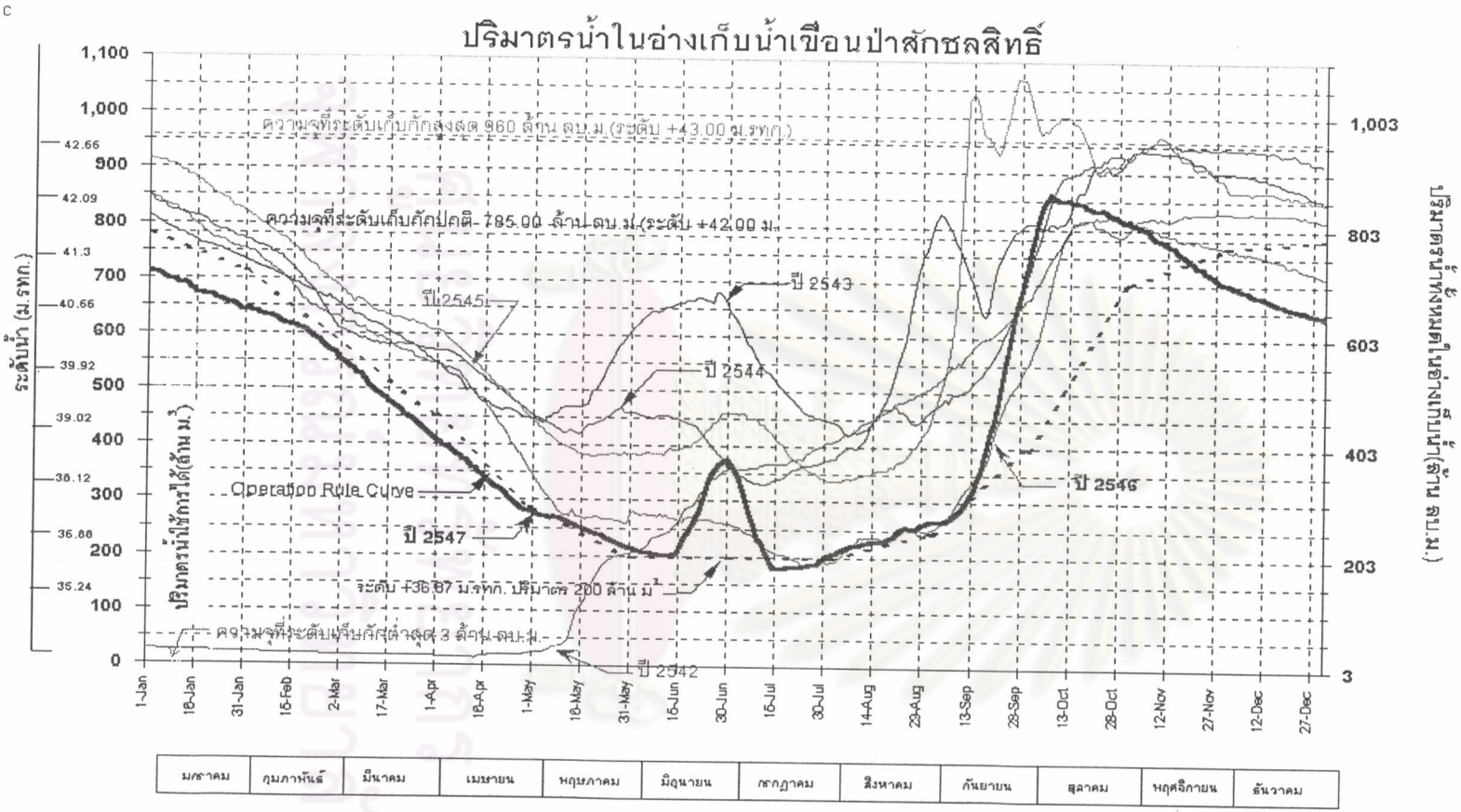
ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ข้อมูลการปฏิบัติการอ่างเก็บน้ำป่าสักชลสิทธิ์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ก- 1 ปริมาตรน้ำในอ่างเก็บน้ำเขื่อนป่าสักชลสิทธิ์ ปี พ.ศ.2542-2547 และ Operation Rule Curve

ที่มา : สำนักชลประทานที่ 10

ตาราง ก-1 เปรียบเทียบปริมาณฯไหลลงอ่างฯ,ปริมาณการระบายและปริมาณน้ำฝนในแต่ละปี

เดือน	ปริมาณน้ำไหลลงอ่างฯ (ล้านลบ.ม.)					ปริมาณการระบายน้ำ (ล้านลบ.ม.)					ปริมาณฝน ณ. หัวงานเขื่อน TS.10 (มม.)				
	ปี 2542	ปี 2543	ปี 2544	ปี 2545	ปี 2546	ปี 2542	ปี 2543	ปี 2544	ปี 2545	ปี 2546	ปี 2542	ปี 2543	ปี 2544	ปี 2545	ปี 2546
ม.ค.	1.25	0.79	26.21	49.44	20.74	5	81	77	113	105	1.30	-	9.30	-	-
ก.พ.	0.95	-	30.57	37.06	20.74	4	80	178	148	125	32.10	5.60	4.20	-	-
มี.ค.	5.27	-	21.75	57.65	10.37	5	95	38	111	102	100.40	12.50	62.50	46.00	34.00
เม.ย.	26.04	47.47	15.59	33.73	3.46	17	122	109	187	192	175.48	392.60	55.90	19.00	16.00
พ.ค.	232.94	294.45	102.39	70.00	22.35	53	84	78	104	104	288.90	281.10	254.30	62.00	60.00
มิ.ย.	152.87	328.86	106.52	120.20	73.66	4	300	183	32	79	100.40	235.22	116.20	57.00	38.00
ก.ค.	55.34	604.42	116.3	55.30	126.52	3	801	85	145	184	133.00	125.80	18.00	29.00	111.00
ส.ค.	147.27	782.94	472.26	162.52	313.11	45	428	377	45	248	124.80	257.10	64.00	190.00	117.00
ก.ย.	642.73	1723.4	442.48	2239.74	603.68	472	1716	163	1778	309	273.30	115.50	68.00	251.00	240.00
ต.ค.	946.14	806.18	332.18	713.26	479.49	824	692	116	710	185	173.20	89.40	90.00	47.00	59.00
พ.ย.	330.10	136.03	135.62	184.81	32.67	301	168	85	229	42	16.10	-	-	17.00	-
ธ.ค.	30.06	31.94	43.23	22.64	31.62	21	79	41	32	31	-	-	-	-	-
รวม	2,570.96	4,756.48	1,845.10	3,746.35	1,738.41	1,753.75	4,646.98	1,528.74	3,633.62	1,706.21	1,418.98	1,514.82	742.40	718.00	675.00

หมายเหตุ - ข้อมูลน้ำจนถึง วันที่ 19 ธันวาคม 2546

- ปี 2545 Inflow สูงสุด วันที่ 10 ก.ย. 2545 = 160.20 ล้าน ลบ.ม./วัน หรือ 1854.20 ลบ.ม./วินาที

  Outflow สูงสุด วันที่ 13 ก.ย. 2545 = 129.60 ล้าน ลบ.ม./วัน หรือ 1500.00 ลบ.ม./วินาที

- ปี 2546 Inflow สูงสุด วันที่ 2 ต.ค. 2546 = 41.48 ล้าน ลบ.ม./วัน หรือ 480.10 ลบ.ม./วินาที

  Outflow สูงสุด วันที่ 25 ก.ย. 2546 = 25.29 ล้าน ลบ.ม./วัน หรือ 300.00 ลบ.ม./วินาที

ที่มา : กรมชลประทาน

ภาคผนวก ข

ข้อมูลนำเข้าของโปรแกรม
และการจัดเรียงข้อมูลลงในโปรแกรม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวกนี้ประกอบไปด้วยข้อมูลนำเข้าของโปรแกรม ได้แก่ ข้อมูลน้ำท่า 33 ปี ตั้งแต่ปี พ.ศ.2313-2545 ดังแสดงในตารางที่ ข-1 ข้อมูลความต้องการน้ำ ตั้งแต่ปี พ.ศ.2313-2545 ดังแสดงในตารางที่ ข-2 โดยแยกออกเป็นข้อมูลนำเข้าของเกณฑ์รวมได้แก่ ข้อมูลน้ำท่า และข้อมูลความต้องการ รวมถึงอัตราการระเหยรายเดือน ความสัมพันธ์ระหว่างพื้นที่ผิว-ความจุอ่างเก็บน้ำ และปริมาณเก็บกักต่ำสุด-สูงสุดของอ่าง ดังแสดงในตารางที่ ข-3

ข้อมูลนำเข้าของเกณฑ์ย่อยได้แก่ ข้อมูลน้ำท่า และข้อมูลความต้องการน้ำของ ปีน้ำน้อย ปีน้ำค่อนข้างน้อย ปีน้ำปกติ ปีน้ำค่อนข้างมาก และปีน้ำมาก รวมถึง อัตราการระเหยรายเดือน ความสัมพันธ์ระหว่างพื้นที่ผิว-ความจุอ่างเก็บน้ำและปริมาณเก็บกักต่ำสุด-สูงสุดของอ่าง ดังแสดงในตารางที่ ข-4

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง ข-1 ปริมาณน้ำท่ารายเดือนเข้าอ่างเก็บน้ำป่าสักชลสิทธิ์

หน่วย ล้าน ลบ.ม.

ปี (พ.ศ.)	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ประเภท
2514	20.3	25.4	31.7	25.7	33.3	22.9	33.2	159.8	418.6	301.5	24.2	14.3	ปีน้ำน้อย
2516	6.4	3.7	6.8	16.0	24.6	14.2	21.9	44.0	344.2	460.9	21.9	11.3	
2522	11.0	6.5	6.4	13.7	65.7	74.6	86.8	94.8	195.9	272.7	22.9	15.8	
2529	23.2	15.3	15.6	25.7	72.8	80.9	43.3	226.2	311.7	126.7	50.0	36.7	
2535	13.7	7.4	6.6	7.1	9.7	38.9	25.6	286.6	208.0	423.3	55.3	31.0	
2536	8.3	7.3	12.1	9.7	17.2	27.9	18.0	54.6	266.1	160.5	13.7	9.8	
2515	33.3	25.4	27.7	34.3	29.0	19.5	23.0	32.6	529.1	567.0	153.8	39.7	ปีน้ำค่อนข้างน้อย
2517	18.3	9.7	12.7	14.4	50.4	19.1	10.7	49.4	201.8	517.6	219.0	47.2	
2520	16.5	8.7	7.3	18.1	50.3	34.4	15.0	50.3	607.7	409.8	31.2	21.2	
2532	12.3	4.6	7.7	11.0	15.8	236.7	82.1	140.0	206.2	317.7	141.2	33.0	
2540	15.2	10.8	4.6	18.0	20.5	15.9	44.2	168.3	428.4	724.0	118.8	38.6	
2541	12.4	10.9	11.4	11.6	18.5	67.6	129.0	148.5	388.4	390.2	67.7	9.6	
2513	26.0	23.3	26.0	29.7	26.9	52.6	149.0	264.1	694.1	401.2	84.1	34.7	ปีน้ำปกติ
2519	34.6	16.6	15.1	15.1	46.0	42.8	63.9	240.4	721.0	991.9	370.6	70.8	
2523	25.1	11.7	18.1	12.4	16.5	77.2	199.0	269.6	637.7	1216.1	120.9	47.7	
2524	27.0	13.1	10.7	21.4	45.3	41.9	210.3	633.5	548.2	378.7	157.1	60.3	
2525	36.0	19.8	16.4	22.2	41.8	61.9	26.8	39.8	1062.4	1094.3	193.1	77.0	
2526	58.5	33.1	29.0	12.8	18.7	32.7	37.2	177.6	414.5	1259.2	299.5	96.7	
2527	37.8	23.2	26.1	29.7	36.8	95.8	163.7	145.0	608.5	1068.0	211.3	75.5	
2531	23.9	12.1	9.9	12.1	207.7	218.3	70.3	146.2	221.4	419.0	278.0	55.0	
2533	13.3	6.1	5.3	3.0	18.7	224.7	143.0	184.0	277.6	853.0	198.4	43.7	
2534	38.3	16.5	10.3	9.4	18.9	79.0	49.0	293.4	1178.1	839.8	51.2	31.9	
2537	10.6	6.8	5.6	7.7	39.7	127.2	254.4	301.1	1030.1	598.4	32.0	33.1	
2542	1.3	1.0	5.3	26.0	232.9	152.9	55.3	147.3	642.7	946.1	330.1	30.1	
2544	26.2	30.6	21.8	15.6	102.4	106.5	116.3	472.3	442.5	332.2	135.6	43.2	
2528	43.3	26.3	22.2	27.0	56.6	71.7	160.4	336.4	975.9	992.0	425.9	99.7	ปีน้ำค่อนข้างมาก
2518	27.2	15.9	15.6	9.7	19.5	64.8	193.2	205.4	986.7	1200.2	128.0	55.2	
2530	31.9	21.6	13.8	7.8	17.1	40.3	35.2	39.7	1261.1	1075.8	155.1	68.1	
2539	20.7	12.2	8.9	13.6	142.8	119.7	121.3	96.9	516.4	1316.9	420.1	70.7	
2521	38.4	22.2	22.0	15.8	35.9	44.5	396.2	741.1	894.1	2436.0	95.2	63.8	ปีน้ำมาก
2538	14.2	8.6	9.4	6.6	23.3	27.6	108.5	845.6	1683.4	766.2	124.8	38.3	
2543	0.8	0.0	0.0	47.5	294.5	328.9	604.4	782.9	1723.2	806.2	136.0	31.9	
2545	49.4	37.1	57.7	33.7	70.0	120.2	55.3	162.5	2239.7	713.3	184.8	22.6	
ค่าเฉลี่ย	23.5	15.0	15.1	17.7	58.2	84.4	113.5	241.8	692.9	738.7	153.1	44.2	
ค่าสูงสุด	58.5	37.1	57.7	47.5	294.5	328.9	604.4	845.6	2239.7	2436.0	425.9	99.7	
ค่าต่ำสุด	0.8	0.0	0.0	3.0	9.7	14.2	10.7	32.6	195.9	126.7	13.7	9.6	

ที่มา 1) ข้อมูล พ.ศ.2513-2541 มาจากการประเมินโดยวิธีสัดส่วนพื้นที่ จากข้อมูลสถานีวัดน้ำท่า S2 และ S9

2) ข้อมูล พ.ศ.2542-2545 มาจากการข้อมูลปริมาณน้ำท่าที่ไหลเข้าอ่างเก็บน้ำจากโครงการป่าสักฯ

STD = 1049.4 ล้าน ลบ.ม.

ตาราง ข-2 ความต้องการน้ำรายเดือนของอ่างเก็บน้ำป่าสักชลสิทธิ์ สภาพการใช้น้ำปัจจุบัน หน่วย ล้าน ลบ.ม.

ปี (พ.ศ.)	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ประเภท
2514	75.3	79.0	131.0	188.9	68.1	3.0	17.3	29.8	15.8	37.1	18.9	5.9	ปีน้ำน้อย
2516	75.1	78.6	131.7	189.9	68.1	3.5	14.8	30.5	15.6	37.1	19.0	5.9	
2522	75.3	77.7	132.8	189.2	68.2	3.0	17.3	36.7	17.4	37.6	19.1	5.9	
2529	75.2	78.8	131.6	188.1	68.3	3.8	14.6	30.3	16.7	34.0	18.8	5.7	
2535	75.3	79.1	131.6	190.4	68.7	3.5	17.3	32.1	14.1	34.9	19.0	5.6	
2536	75.2	79.1	129.8	188.4	68.2	3.3	16.6	29.2	14.7	37.3	19.2	5.9	
2515	75.3	79.0	130.1	187.9	68.9	3.1	18.4	33.9	12.7	35.9	18.3	5.7	ปีน้ำค่อนข้างน้อย
2517	73.8	78.6	130.7	188.0	68.1	3.7	14.6	31.9	13.5	33.7	18.0	5.8	
2520	75.1	76.7	132.0	188.3	68.2	3.5	19.4	32.3	13.3	34.8	19.1	5.6	
2532	75.3	79.1	130.9	189.6	68.3	3.4	19.3	31.4	13.8	35.8	18.6	5.9	
2540	74.9	77.8	133.0	189.4	68.6	4.0	19.8	33.2	16.5	35.5	19.2	5.9	
2541	74.1	78.5	132.5	189.2	68.3	3.5	16.7	29.8	17.5	36.5	18.7	5.9	
2513	75.2	78.4	131.3	189.4	68.5	3.1	15.6	30.3	13.9	35.5	19.0	5.6	ปีน้ำปกติ
2519	75.2	78.7	130.9	190.2	67.9	3.2	13.4	28.4	13.1	34.2	18.7	5.9	
2523	75.2	77.6	132.2	189.2	68.3	3.4	16.4	29.5	13.3	33.7	18.8	5.9	
2524	75.3	79.1	132.0	188.3	68.0	3.7	15.3	32.2	13.8	35.3	17.5	5.9	
2525	75.2	79.1	133.5	188.5	68.5	3.2	13.7	33.8	14.4	35.0	18.4	5.9	
2526	75.0	76.9	131.5	190.3	68.2	3.2	16.3	28.6	16.1	33.3	18.7	5.6	
2527	73.8	78.6	132.7	189.4	68.4	3.6	20.7	30.8	16.1	34.7	18.9	5.9	
2531	74.3	78.7	131.7	188.0	68.0	3.2	15.6	30.8	18.5	34.6	19.3	5.9	
2533	75.2	78.7	131.9	189.5	68.1	3.7	19.9	39.1	14.4	33.5	18.8	5.9	
2534	74.8	78.6	133.3	189.6	68.4	3.3	17.3	31.4	16.2	36.4	19.3	5.5	
2537	75.1	79.1	132.1	189.1	68.3	3.2	19.5	29.9	15.2	37.0	19.3	5.9	
2542	75.2	78.7	132.2	186.6	68.0	3.4	14.3	31.1	14.7	34.0	18.5	5.9	
2544	75.0	78.5	132.1	189.0	68.2	3.3	16.5	31.3	15.0	34.8	18.8	5.8	
2518	75.3	77.8	129.4	189.6	68.2	3.5	15.0	32.0	15.3	36.0	18.5	5.7	ปีน้ำค่อนข้างมาก
2528	75.3	79.1	133.5	189.0	68.3	3.4	15.2	32.8	14.1	34.7	17.8	5.8	
2530	75.2	77.7	132.4	189.4	68.3	3.4	20.4	35.1	13.1	35.8	18.4	5.9	
2539	75.2	78.7	131.6	188.2	68.1	3.4	18.4	35.6	13.7	34.5	17.8	5.9	
2521	74.8	77.8	133.3	189.2	68.2	3.5	13.5	33.7	15.2	34.9	18.9	5.9	ปีน้ำมาก
2538	75.3	77.9	132.7	188.7	68.2	3.3	16.5	28.2	13.1	34.0	18.9	5.9	
2543	75.0	78.8	130.7	188.5	68.1	3.5	18.2	32.3	17.5	34.2	19.0	5.9	
2545	75.0	78.1	132.2	188.8	68.1	3.4	16.1	31.4	15.3	34.4	18.9	5.9	
ค่าเฉลี่ย	75.0	78.4	131.8	189.0	68.3	3.4	16.8	31.8	15.0	35.2	18.7	5.8	
ค่าสูงสุด	75.3	79.1	133.5	190.4	68.9	4.0	20.7	39.1	18.5	37.6	19.3	5.9	
ค่าต่ำสุด	73.8	76.7	129.4	186.6	67.9	3.0	13.4	28.2	12.7	33.3	17.5	5.5	

ที่มา 1.รายงานการศึกษาเกณฑ์การเก็บกักน้ำเขื่อนป่าสักชลสิทธิ์ กรมชลประทาน (พฤษภาคม 2546)

2. อรอนงค์ (2546)

ตารางที่ ข-3 การจัดเรียงข้อมูลในโปรแกรมของเกณฑ์รวม

Data file and Comments

การจัดเรียงข้อมูลน้ำท่าและความต้องการน้ำลงในโปรแกรมของเกณฑ์รวม

Inflow File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
26.05	23.25	26.05	29.74	26.95	52.63	148.98	264.07	694.06	401.24	84.09	34.70
20.3	25.4	31.7	25.7	33.3	22.9	33.2	159.8	418.6	301.5	24.2	14.3
33.35	25.42	27.67	34.34	29.02	19.47	22.98	32.63	529.13	566.98	153.75	39.75
6.4	3.7	6.8	16.0	24.6	14.2	21.9	44.0	344.2	460.9	21.9	11.3
18.30	9.74	12.66	14.39	50.44	19.12	10.74	49.35	201.77	517.64	218.98	47.17
27.22	15.93	15.57	9.74	19.49	64.83	193.22	205.42	986.74	1200.17	128.02	55.18
34.60	16.57	15.11	15.11	45.98	42.80	63.92	240.38	720.96	991.93	370.59	70.84
16.48	8.74	7.28	18.12	50.26	34.42	15.02	50.26	607.69	409.83	31.23	21.22
38.42	22.22	22.03	15.75	35.87	44.53	396.17	741.08	894.14	2436.04	95.24	63.83
11.0	6.5	6.4	13.7	65.7	74.6	86.8	94.8	195.9	272.7	22.9	15.8
25.13	11.75	18.12	12.38	16.48	77.21	199.04	269.61	637.74	1216.11	120.92	47.71
27.04	13.11	10.65	21.40	45.25	41.88	210.33	633.55	548.23	378.69	157.07	60.28
35.97	19.76	16.39	22.22	41.79	61.92	26.77	39.79	1062.41	1094.28	193.12	77.03
58.46	33.05	28.95	12.84	18.67	32.69	37.24	177.55	414.47	1259.18	299.47	96.70
37.79	23.22	26.13	29.68	36.79	95.79	163.71	145.05	608.51	1067.96	211.33	75.48
43.25	26.31	22.22	27.04	56.64	71.66	160.44	336.44	975.91	992.03	425.95	99.70
23.2	15.3	15.6	25.7	72.8	80.9	43.3	226.2	311.7	126.7	50.0	36.7
31.87	21.58	13.84	7.83	17.12	40.34	35.24	39.70	1261.09	1075.79	155.06	68.11
23.86	12.11	9.92	12.11	207.69	218.25	70.29	146.23	221.44	419.03	277.99	55.00
12.29	4.55	7.74	11.02	15.75	236.74	82.13	140.04	206.24	317.68	141.22	32.96
13.29	6.10	5.28	3.00	18.67	224.72	143.04	184.02	277.62	852.99	198.41	43.71
38.33	16.48	10.29	9.38	18.94	79.03	48.99	293.37	1178.14	839.78	51.17	31.87
13.7	7.4	6.6	7.1	9.7	38.9	25.6	286.6	208.0	423.3	55.3	31.0
8.3	7.3	12.1	9.7	17.2	27.9	18.0	54.6	266.1	160.5	13.7	9.8
10.56	6.83	5.65	7.74	39.70	127.20	254.40	301.11	1030.09	598.40	31.96	33.05
14.20	8.56	9.38	6.56	23.31	27.59	108.54	845.61	1683.39	766.21	124.83	38.33
20.67	12.20	8.92	13.57	142.77	119.74	121.28	96.88	516.36	1316.90	420.12	70.75
15.22	10.81	4.63	17.97	20.53	15.91	44.22	168.34	428.39	723.96	118.79	38.64
12.37	10.94	11.35	11.63	18.47	67.56	129.05	148.47	388.38	390.18	67.74	9.59
1.25	0.95	5.27	26.04	232.94	152.87	55.34	147.27	642.73	946.14	330.10	30.06
0.79	0.00	0.00	47.47	294.45	328.86	604.42	782.94	1723.24	806.18	136.03	31.94

26.21	30.57	21.75	15.59	102.39	106.52	116.30	472.26	442.48	332.18	135.62	43.23
49.44	37.06	57.65	33.73	70.00	120.20	55.30	162.52	2239.74	713.26	184.81	22.64

Demand File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
75.2	78.4	131.3	189.4	68.5	3.1	15.6	30.3	13.9	35.5	19.0	5.6
75.3	79.0	131.0	188.9	68.1	3.0	17.3	29.8	15.8	37.1	18.9	5.9
75.3	79.0	130.1	187.9	68.9	3.1	18.4	33.9	12.7	35.9	18.3	5.7
75.1	78.6	131.7	189.9	68.1	3.5	14.8	30.5	15.6	37.1	19.0	5.9
73.8	78.6	130.7	188.0	68.1	3.7	14.6	31.9	13.5	33.7	18.0	5.8
75.3	77.8	129.4	189.6	68.2	3.5	15.0	32.0	15.3	36.0	18.5	5.7
75.2	78.7	130.9	190.2	67.9	3.2	13.4	28.4	13.1	34.2	18.7	5.9
75.1	76.7	132.0	188.3	68.2	3.5	19.4	32.3	13.3	34.8	19.1	5.6
74.8	77.8	133.3	189.2	68.2	3.5	13.5	33.7	15.2	34.9	18.9	5.9
75.3	77.7	132.8	189.2	68.2	3.0	17.3	36.7	17.4	37.6	19.1	5.9
75.2	77.6	132.2	189.2	68.3	3.4	16.4	29.5	13.3	33.7	18.8	5.9
75.3	79.1	132.0	188.3	68.0	3.7	15.3	32.2	13.8	35.3	17.5	5.9
75.2	79.1	133.5	188.5	68.5	3.2	13.7	33.8	14.4	35.0	18.4	5.9
75.0	76.9	131.5	190.3	68.2	3.2	16.3	28.6	16.1	33.3	18.7	5.6
73.8	78.6	132.7	189.4	68.4	3.6	20.7	30.8	16.1	34.7	18.9	5.9
75.3	79.1	133.5	189.0	68.3	3.4	15.2	32.8	14.1	34.7	17.8	5.8
75.2	78.8	131.6	188.1	68.3	3.8	14.6	30.3	16.7	34.0	18.8	5.7
75.2	77.7	132.4	189.4	68.3	3.4	20.4	35.1	13.1	35.8	18.4	5.9
74.3	78.7	131.7	188.0	68.0	3.2	15.6	30.8	18.5	34.6	19.3	5.9
75.3	79.1	130.9	189.6	68.3	3.4	19.3	31.4	13.8	35.8	18.6	5.9
75.2	78.7	131.9	189.5	68.1	3.7	19.9	39.1	14.4	33.5	18.8	5.9
74.8	78.6	133.3	189.6	68.4	3.3	17.3	31.4	16.2	36.4	19.3	5.5
75.3	79.1	131.6	190.4	68.7	3.5	17.3	32.1	14.1	34.9	19.0	5.6
75.2	79.1	129.8	188.4	68.2	3.3	16.6	29.2	14.7	37.3	19.2	5.9
75.1	79.1	132.1	189.1	68.3	3.2	19.5	29.9	15.2	37.0	19.3	5.9
75.3	77.9	132.7	188.7	68.2	3.3	16.5	28.2	13.1	34.0	18.9	5.9
75.2	78.7	131.6	188.2	68.1	3.4	18.4	35.6	13.7	34.5	17.8	5.9
74.9	77.8	133.0	189.4	68.6	4.0	19.8	33.2	16.5	35.5	19.2	5.9
74.1	78.5	132.5	189.2	68.3	3.5	16.7	29.8	17.5	36.5	18.7	5.9
75.2	78.7	132.2	186.6	68.0	3.4	14.3	31.1	14.7	34.0	18.5	5.9
75.0	78.8	130.7	188.5	68.1	3.5	18.2	32.3	17.5	34.2	19.0	5.9
75.0	78.5	132.1	189.0	68.2	3.3	16.5	31.3	15.0	34.8	18.8	5.8
75.0	78.1	132.2	188.8	68.1	3.4	16.1	31.4	15.3	34.4	18.9	5.9

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การจัดเรียงข้อมูลโค้งพื้นที่-ความจุอ่างฯ ค่าระดับเก็บกักต่ำสุดสูงสุด และอัตราการระเหย ของเกณฑ์รวมแสดงได้ดังนี้

MIN-MAX CAPACITY

200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000

AREA-CAPACITY CURVE (km^2 - MCM)

0.00 0.00
3.30 3.30
8.36 14.57
27.00 48.16
59.96 132.96
95.46 287.01
127.15 508.86
148.75 784.48
191.75 1124.08
252.44 1566.88

EVAPORATION (m./month)

0.1084 0.1054 0.1207 0.0817 0.0367 0.0312 0.0181 0.00 0.00 0.0042 0.0842 0.1038

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-4 การจัดเรียงข้อมูลในโปรแกรมของเกณฑ์ย่อย

Data file and Comments

การจัดเรียงข้อมูลน้ำท่าและความต้องการน้ำลงในโปรแกรมของปีน้ำน้อย

Inflow File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
13.75	7.38	6.65	7.10	9.65	38.88	25.59	286.64	207.97	423.31	55.27	30.96
20.28	25.42	31.72	25.69	33.26	22.89	33.17	159.79	418.63	301.47	24.15	14.33
23.22	15.30	15.57	25.68	72.75	80.95	43.34	226.18	311.68	126.75	49.99	36.69
6.37	3.73	6.83	16.00	24.59	14.20	21.85	43.98	344.18	460.91	21.94	11.29
11.02	6.46	6.37	13.66	65.74	74.57	86.77	94.79	195.95	272.70	22.95	15.75
8.29	7.28	12.11	9.74	17.21	27.86	18.03	54.63	266.15	160.53	13.66	9.83

Demand File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
75.3	79.1	131.6	190.4	68.7	3.5	17.3	32.1	14.1	34.9	19.0	5.6
75.3	79.0	131.0	188.9	68.1	3.0	17.3	29.8	15.8	37.1	18.9	5.9
75.2	78.8	131.6	188.1	68.3	3.8	14.6	30.3	16.7	34.0	18.8	5.7
75.1	78.6	131.7	189.9	68.1	3.5	14.8	30.5	15.6	37.1	19.0	5.9
75.3	77.7	132.8	189.2	68.2	3.0	17.3	36.7	17.4	37.6	19.1	5.9
75.2	79.1	129.8	188.4	68.2	3.3	16.6	29.2	14.7	37.3	19.2	5.9

การจัดเรียงข้อมูลน้ำท่าและความต้องการน้ำลงในโปรแกรมของปีน้ำค่อนข้างน้อย

Inflow File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
13.75	7.38	6.65	7.10	9.65	38.88	25.59	286.64	207.97	423.31	55.27	30.96
20.28	25.42	31.72	25.69	33.26	22.89	33.17	159.79	418.63	301.47	24.15	14.33
23.22	15.30	15.57	25.68	72.75	80.95	43.34	226.18	311.68	126.75	49.99	36.69
6.37	3.73	6.83	16.00	24.59	14.20	21.85	43.98	344.18	460.91	21.94	11.29
11.02	6.46	6.37	13.66	65.74	74.57	86.77	94.79	195.95	272.70	22.95	15.75
8.29	7.28	12.11	9.74	17.21	27.86	18.03	54.63	266.15	160.53	13.66	9.83

Demand File

1

12

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

1	2	3	4	5	6	7	8	9	10	11	12
75.3	79.1	131.6	190.4	68.7	3.5	17.3	32.1	14.1	34.9	19.0	5.6
75.3	79.0	131.0	188.9	68.1	3.0	17.3	29.8	15.8	37.1	18.9	5.9
75.2	78.8	131.6	188.1	68.3	3.8	14.6	30.3	16.7	34.0	18.8	5.7
75.1	78.6	131.7	189.9	68.1	3.5	14.8	30.5	15.6	37.1	19.0	5.9
75.3	77.7	132.8	189.2	68.2	3.0	17.3	36.7	17.4	37.6	19.1	5.9
75.2	79.1	129.8	188.4	68.2	3.3	16.6	29.2	14.7	37.3	19.2	5.9

การจัดเรียงข้อมูลน้ำท่าและความต้องการน้ำลงในโปรแกรมของปีน้ำปกติ

Inflow File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
35.97	19.76	16.39	22.22	41.79	61.92	26.77	39.79	1062.41	1094.28	193.12	77.03
25.13	11.75	18.12	12.38	16.48	77.21	199.04	269.61	637.74	1216.11	120.92	47.71
34.60	16.57	15.11	15.11	45.98	42.80	63.92	240.38	720.96	991.93	370.59	70.84
38.33	16.48	10.29	9.38	18.94	79.03	48.99	293.37	1178.14	839.78	51.17	31.87
1.25	0.95	5.27	26.04	232.94	152.87	55.34	147.27	642.73	946.14	330.10	30.06
37.79	23.22	26.13	29.68	36.79	95.79	163.71	145.05	608.51	1067.96	211.33	75.48
58.46	33.05	28.95	12.84	18.67	32.69	37.24	177.55	414.47	1259.18	299.47	96.70
10.56	6.83	5.65	7.74	39.70	127.20	254.40	301.11	1030.09	598.40	31.96	33.05
27.04	13.11	10.65	21.40	45.25	41.88	210.33	633.55	548.23	378.69	157.07	60.28
13.29	6.10	5.28	3.00	18.67	224.72	143.04	184.02	277.62	852.99	198.41	43.71
26.21	30.57	21.75	15.59	102.39	106.52	116.30	472.26	442.48	332.18	135.62	43.23
26.05	23.25	26.05	29.74	26.95	52.63	148.98	264.07	694.06	401.24	84.09	34.70
23.86	12.11	9.92	12.11	207.69	218.25	70.29	146.23	221.44	419.03	277.99	55.00

Demand File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
75.2	79.1	133.5	188.5	68.5	3.2	13.7	33.8	14.4	35.0	18.	5.9
75.2	77.6	132.2	189.2	68.3	3.4	16.4	29.5	13.3	33.7	18.8	5.9
75.2	78.7	130.9	190.2	67.9	3.2	13.4	28.4	13.1	34.2	18.7	5.9
74.8	78.6	133.3	189.6	68.4	3.3	17.3	31.4	16.2	36.4	19.3	5.5
75.2	78.7	132.2	186.6	68.0	3.4	14.3	31.1	14.7	34.0	18.5	5.9
73.8	78.6	132.7	189.4	68.4	3.6	20.7	30.8	16.1	34.7	18.9	5.9
75.0	76.9	131.5	190.3	68.2	3.2	16.3	28.6	16.1	33.3	18.7	5.6
75.1	79.1	132.1	189.1	68.3	3.2	19.5	29.9	15.2	37.0	19.3	5.9
75.3	79.1	132.0	188.3	68.0	3.7	15.3	32.2	13.8	35.3	17.5	5.9
75.2	78.7	131.9	189.5	68.1	3.7	19.9	39.1	14.4	33.5	18.8	5.9

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

75.0	78.5	132.1	189.0	68.2	3.3	16.5	31.3	15.0	34.8	18.8	5.8
75.2	78.4	131.3	189.4	68.5	3.1	15.6	30.3	13.9	35.5	19.0	5.6
74.3	78.7	131.7	188.0	68.0	3.2	15.6	30.8	18.5	34.6	19.3	5.9

การจัดเรียงข้อมูลน้ำท่าและความต้องการน้ำลงในโปรแกรมของปีน้ำค่อนข้างมาก

Inflow File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
43.25	26.31	22.22	27.04	56.64	71.66	160.44	336.44	975.91	992.03	425.95	99.70
27.22	15.93	15.57	9.74	19.49	64.83	193.22	205.42	986.74	1200.17	128.02	55.18
20.67	12.20	8.92	13.57	142.77	119.74	121.28	96.88	516.36	1316.90	420.12	70.75
31.87	21.58	13.84	7.83	17.12	40.34	35.24	39.70	1261.09	1075.79	155.06	68.11

Demand File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
75.3	79.1	133.5	189.0	68.3	3.4	15.2	32.8	14.1	34.7	17.8	5.8
75.3	77.8	129.4	189.6	68.2	3.5	15.0	32.0	15.3	36.0	18.5	5.7
75.2	78.7	131.6	188.2	68.1	3.4	18.4	35.6	13.7	34.5	17.8	5.9
75.2	77.7	132.4	189.4	68.3	3.4	20.4	35.1	13.1	35.8	18.4	5.9

การจัดเรียงข้อมูลน้ำท่าและความต้องการน้ำลงในโปรแกรมของปีน้ำมาก

Inflow File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
38.42	22.22	22.03	15.75	35.87	44.53	396.17	741.08	894.14	2436.04	95.24	63.83
0.79	0.00	0.00	47.47	294.45	328.86	604.42	782.94	1723.24	806.18	136.03	31.94
49.44	37.06	57.65	33.73	70.00	120.20	55.30	162.52	2239.74	713.26	184.81	22.64
14.20	8.56	9.38	6.56	23.31	27.59	108.54	845.61	1683.39	766.21	124.83	38.33

Demand File

1

12

1	2	3	4	5	6	7	8	9	10	11	12
74.8	77.8	133.3	189.2	68.2	3.5	13.5	33.7	15.2	34.9	18.9	5.9
75.0	78.8	130.7	188.5	68.1	3.5	18.2	32.3	17.5	34.2	19.0	5.9
75.0	78.1	132.2	188.8	68.1	3.4	16.1	31.4	15.3	34.4	18.9	5.9
75.3	77.9	132.7	188.7	68.2	3.3	16.5	28.2	13.1	34.0	18.9	5.9

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การจัดเรียงข้อมูลโค้งพื้นที่-ความจุอ่างฯ ค่าระดับเก็บกักต่ำสุดสูงสุด และอัตราการระเหย ของทุกปีน้ำแสดงได้ดังนี้

MIN-MAX CAPACITY

200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000
200.000 600.000

AREA-CAPACITY CURVE (km^2 - MCM)

0.00 0.00
3.30 3.30
8.36 14.57
27.00 48.16
59.96 132.96
95.46 287.01
127.15 508.86
148.75 784.48
191.75 1124.08
252.44 1566.88

EVAPORATION (m./month)

0.1084 0.1054 0.1207 0.0817 0.0367 0.0312 0.0181 0.00 0.00 0.0042 0.0842 0.1038

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

โปรแกรมภาษาซี หาเกณฑ์การปฏิบัติงานอ่างเก็บน้ำ ด้วยวิธี Genetic Algorithms

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวกนี้ประกอบไปด้วยsource code สำหรับการหาRule Curve ของอ่างเก็บน้ำเขื่อนป่าสักชลสิทธิ์ ด้วยวิธีการของ GAs

ตารางที่ ค-1 ส่วนประกอบข้องไฟล์ที่ใช้ในแบบจำลอง GAs

File name	Description
Sp6xtour.c	Name of source code file
Projfile.lst	Name of input project file
Network.net	Name of network file
Demand.dem	Name of demand file
Inflow	Name of Inflow filw
Area_capa	Name of Relation Area-Capacity
Test12n_R	Name of Min-Max capacity
Input2.txt	Name of evaporation
Sp6xtour	Output file

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
/*  PROGRAM FOR RULE CURVE PASAK  */

#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <process.h>
#include <time.h>

#define POPSIZE   120
#define LENGTH    12
#define MaxGEN    40000
#define Pmutate   0.08
#define PXOVER    0.95
#define Accurate  0.001
#define INTERVAL  300
#define DELTA     0.000001
#define NREACH    12
#define NNODE     12
#define YEAR     13
#define PrintMENU 0
#define Modi_muta 0.3

void initial();
float RandReal (float, float);
int  RandInt (int, int);
void Read_inputfiles();
void SwapIntValues(int *, int *);
void SwapRealValues(float *, float *);
void nrerror(char *messg) ;
void objective(int);
void Tournament();
void SelectMate();
void UniSelectMate();
void mutatepop(int);

int    nreach, nnodes, mated[POPSIZE], index, R1, R2;

float  Qinf[YEAR][NNODE],d[YEAR][NNODE],y2[NNODE], upper[NREACH], lower
[NREACH];
float  area[NNODE],volume[NNODE],evap[NNODE] ;
float  L[NNODE],E[NNODE] ;

float  ftnss[POPSIZE+1],Q[NREACH][POPSIZE+1];
float  mated_pop[POPSIZE][LENGTH], DELTA_ftnss,OLD_minftnss;
float  bestfit, sumx,sumd;
int     pofi,ii;

FILE   *f1,*f2,*f3,*f4,*f5;

void main (void)
{
    int   gen;
```

```
    clock_t start,finish;
    float duration;

    if(!(f1 = fopen("sp6xtour.out","w"))) {
        printf("can't open file ga31out.c\n");
        exit(1);
    }



    Read_inputfiles();
    srand(1);
    start = clock();
//  R1 = 50;
//do{

    R1  = 4400;
    R2  =   250;
    initial();

    index = 0;
    OLD_minftnss     = 10000.0;
    DELTA_ftnss      = 1000000.0;
    ftnss[POPSIZE]   = 1000000000.0f;
    bestfit          = 100000000.0f;

for(gen = 0; gen <= MaxGEN; gen++) {
    objective(gen);  if(index == 999) break;
    Tournament();
    SelectMate();
//  UniSelectMate();
    mutatepop(gen);

     if (!((gen+1)%INTERVAL)){
        DELTA_ftnss = (float)fabs((double)((OLD_minftnss - ftnss[POPSIZ
E])/ftnss[POPSIZE]));
        OLD_minftnss = ftnss[POPSIZE];
     }

} /**** End of GENERATED loop ****/
  finish = clock();
  duration = (float) (finish - start) / CLOCKS_PER_SEC;
  fprintf(f1,"Time = %5.2f sec", duration);

//  R1 = R1 + 100;
//}while (R1 <= 6000);

  printf("End of run  ENJOY!!!! ");
  (void) fclose(f3);
  (void) fclose(f1);
}  /**** End of MAIN program ****/

/**********START OF SUBROUTINES **********/
```

```
void Read_inputfiles()
{
    int   ii, jj, y, nyrs,  nno[NNODE],nno2[NNODE],nno3[NNODE],scheme_n
ode[NNODE],
          num_schemes, nchk, nstep;

    float tdem;//  float  dr;
    char  title[80], title2[80], title3[80], title4[80];
    char  networkfile[20], demandsfile[20], inflowsfile[20],evapfile[20
];

    if(!(f2 = fopen("Projfile.lst","r"))) {
      nrerror(" Can not open project file") ;
    }
    /* now read the data file names */
    fgets(networkfile, 20, f2) ;
    fgets(demandsfile, 20, f2) ;
    fgets(inflowsfile, 20, f2) ;
    fgets(evapfile, 20, f2) ;

    for(ii = 0; ii < 20; ii++) {
        if(networkfile[ii] == '\n') networkfile[ii] = '\0' ;
        if(demandsfile[ii] == '\n') demandsfile[ii] = '\0' ;
        if(inflowsfile[ii] == '\n') inflowsfile[ii] = '\0' ;
        if(evapfile[ii] == '\n') evapfile[ii] = '\0' ;
    }
    (void) fclose(f2) ;


    /*************************/
    /* Now read network data */

    if(!(f2 = fopen(networkfile,"r"))) {
       nrerror(" Unable to open network file") ;
    }

    fgets(title, 80, f2);
//  puts(title);


    fscanf(f2," %d", &nnodes) ;
//  printf("%3d\n",nnodes) ;
    tdem = 0 ;

    for(jj = 0; jj < nnodes; jj++) fscanf(f2," %d ", &nno[jj]) ;



    fscanf(f2,"%d",&num_schemes);


    for(jj=0;jj<num_schemes;jj++) fscanf(f2,"%d ", &scheme_node[jj]);


    /****************************************/
```

```
/* Need to read the demand file in here !!*/

    if(!(f2 = fopen(demandsfile, "r")))  {
        nrerror(" can't open demands file") ;
    }

    fgets(title2, 80, f2);

    fscanf(f2," %d", &nstep) ;

    fscanf(f2," %d ", &nchk) ;

    if(nchk != nnodes)  nrerror(" 3#Index and reach no.arrays out of si
nc. ") ;
    for(jj = 0; jj < nnodes; jj++) {
        fscanf(f2," %d ", &nno2[jj]) ;
    }

        for(y = 0; y < YEAR; y++)  {

                for(jj = 0; jj < nnodes; jj++) {
                fscanf(f2," %f ", &d[y][jj]) ;
                fscanf(f2,"\n");
            }

        }

(void)fclose(f2);

   /****************close demand file************/

 /* Start to read inflows data. The flow data file is kept open */

    if(!(f2 = fopen(inflowsfile,"r"))) {
    nrerror("can't open inflows file");
    }
    fgets(title3, 80, f2);

    fscanf(f2," %d ", &nyrs) ;

    fscanf(f2," %d ", &nchk) ;

for(jj = 0; jj < nnodes; jj++)
        fscanf(f2," %d ", &nno3[jj]) ;

    for(y = 0; y < YEAR; y++) {
        for(jj = 0; jj < nnodes; jj++) {
            fscanf(f2," %f ", &Qinf[y][jj]) ;
            fscanf(f2,"\n");
        }
    }


    (void)fclose(f2) ;
```

```
    /*************close inflow file***************/



    /* Start to read evap data.  */

    if(!(f2 = fopen(evapfile,"r"))) {
    nrerror("can't open evap file");
    }
    fgets(title4, 80, f2);

for(jj = 0; jj < nnodes; jj++)
        fscanf(f2," %f ", &evap[jj]) ;


    (void)fclose(f2) ;

    /*************close evap file***************/


}

/************************************/
void initial()
{
    int i,j,ii;

    if(!(f3 = fopen("test12n_R.dat","r"))) {
        printf("can't open test.dat\n");
        exit(1);
    }
    for (j=0; j<NREACH; j++){
        fscanf(f3,"%f %f",&lower[j],&upper[j]);
//      fprintf(f1,"%3.3f %3.3f\n",lower[j],upper[j]);
    }
//  fprintf(f1,"\n");

//  printf("values read\n");
//  getchar();
    (void) fclose(f3);

    if(!(f5 = fopen("area_capa.dat","r"))) {
        printf("can't open areacapa.dat\n");
        exit(1);
    }
    for (ii=0; ii<10; ii++)
        fscanf(f5,"%f %f",&area[ii],&volume[ii]);

  (void) fclose(f5);

    for(i=0;i<POPSIZE;i++){
        for(j=0;j<NREACH;j++){
         Q[j][i] = RandReal(lower[j],upper[j]);    /* Random Real Values
*/
```

```
//        fprintf(f1,"%7.3f",Q[j][i]);
          }
//     fprintf(f1,"\n");
     }
//   fprintf(f1,"\n");

}
/*************************************/
 /* This is to random initial population in Real Number */
float RandReal (float lower,float upper)
{
    float val;
    float nb = ( upper - lower );
    val =  (float)(rand()%1000/1000.0*nb + lower);
    return ((float)val);
}
/*************************************/
 /* This is to random initial population for integers */
int RandInt (int lower,int upper)
{
    int val;
    int nb = (upper - lower );
    val =  (rand()%nb + lower);
    return val;
}
/*********************************************************************/



/* routine to calculate fitness,constraints to objective function */
void  objective(int gen)
{
   int    i,j,jj,y,ii,check1[POPSIZE],check2[POPSIZE],seton;
   float  yy,TATALspill,TOTALshortage,AVERAGEspill,AVERAGEshortage;
   float  sumftnss,minftnss,stringftnss,avgftnss;

   float  x[POPSIZE][YEAR][NNODE];
   float  flood[POPSIZE][YEAR][NNODE],pp[POPSIZE][YEAR][NNODE];

   float  SH[POPSIZE][YEAR][NNODE],SP[POPSIZE][YEAR][NNODE],RE[POPSIZE]
[YEAR][NNODE];
   float  Balnc[POPSIZE][YEAR][NNODE],nodeftnss[POPSIZE][YEAR][NNODE];

   float  s[POPSIZE][NNODE],SUMoutflow[YEAR] ,SUMspill[YEAR],SUMshortag
e[YEAR] ;
   float  sumconstr1[POPSIZE],sumconstr2[POPSIZE];//,yy2[NNODE];

   float  constraint1[POPSIZE][YEAR][NNODE],excess[POPSIZE][YEAR][NNODE
];

   float  pofx[YEAR][NNODE];
   float  pRE[YEAR][NNODE],pSH[YEAR][NNODE],pSP[YEAR][NNODE];

  sumftnss = 0.0f;
```

```
    for(i=0;i<POPSIZE;i++)  {
      sumconstr1[i] = 0.0f;
      sumconstr2[i] = 0.0f;
      stringftnss  = 0.0f;
      check1[i] = 0;
      check2[i] = 0;


      for(jj=0; jj<NNODE; jj++)    s[i][jj] = Q[jj][i];

  for(y=0; y<YEAR; y++){

  for(jj=0; jj<NNODE; jj++){

      //Interpolat evapotran
      if(jj > 0){
          for(ii=0; ii<10; ii++)
              if ( volume[ii] > s[i][jj-1]  ) {
                      yy = (float)(s[i][jj-1]-volume[ii-1])
                           /(float)(volume[ii]-volume[ii-1])
                           *(float)(area[ii]-area[ii-1]);
               //        yy2[jj] = (float)(yy+area[ii-1]);
                      E[jj] =  (float)(evap[jj]*(yy+area[ii-1]));

                             /*  printf("Y = %6.5f\n",  y2+area[ii-1]);
                                 printf("E = %6.5f\n",  E);
                                 printf(f1,"E = %6.5f\n",  E);*/
                      ii = 9;
                  }

      }


      else    {
          for(ii=0; ii<10; ii++)
              if ( volume[ii] > s[i][jj+11]  ) {
                      yy = (float)(s[i][jj+11]-volume[ii-1])
                           /(float)(volume[ii]-volume[ii-1])
                           *(float)(area[ii]-area[ii-1]);
                 //     yy2[jj] = (float)(yy+area[ii-1]);
                      E[jj] =  (float)(evap[jj]*(yy+area[ii-1]));


                      ii = 9;
                  }

      }
      // loss from seepage

  //  L[jj] = (float)0.1/12*s[i][jj-1];
      if(jj > 0) L[jj] = (float)( (s[i][jj-1] + s[i][jj])*0.1/24 );
      else L[jj] = (float)( (s[i][jj+11] + s[i][jj])*0.1/24 );
```

```
    // calculate supply (x) here
    if(jj > 0) x[i][y][jj] = (float)(Qinf[y][jj] + s[i][jj-1] - s[i][jj
] - E[jj] - L[jj] );
    else x[i][y][jj] = (float)(Qinf[y][jj] + s[i][jj+11] - s[i][jj] - E
[jj]- L[jj] );



    if(d[y][jj] >= x[i][y][jj] ) RE[i][y][jj] = x[i][y][jj];
    else if(d[y][jj] >= 26.0 ) RE[i][y][jj] = d[y][jj];
        else RE[i][y][jj] = 26.0f;



    if(d[y][jj] >= 26 ) SH[i][y][jj] =  d[y][jj] - RE[i][y][jj];
        else SH[i][y][jj] = 0.0f;


    SP[i][y][jj] = x[i][y][jj] - RE[i][y][jj];
    if(SP[i][y][jj]<0.0) SP[i][y][jj] = 0.0f;




    if(x[i][y][jj] < 26.0f){ Balnc[i][y][jj] = (float)fabs((double)(26.
0 - x[i][y][jj]));
                            constraint1[i][y][jj] = ((float)pow(Balnc[i][
y][jj],2));
        }
     else constraint1[i][y][jj]  = 0.0f;
      sumconstr1[i] = sumconstr1[i] + constraint1[i][y][jj];



   if(x[i][y][jj] > 1555.2f){ flood[i][y][jj] = (float)fabs((double)(x
[i][y][jj] - 1555.2));
                            excess[i][y][jj] = ((float)pow(flood[i][y][jj
],2));
       }
     else excess[i][y][jj] = 0.0f;
     sumconstr2[i] = sumconstr2[i] + excess[i][y][jj];



    pp[i][y][jj] = (float)fabs((double)(d[y][jj]-x[i][y][jj]));


    if (d[y][jj] >= 26.0 ) nodeftnss[i][y][jj] = (float)(1/d[y][jj])*(
(float)pow(pp[i][y][jj],2));
    else nodeftnss[i][y][jj] = (float)(1/RE[i][y][jj])*((float)pow(pp[
i][y][jj],2));
```

```
      stringftnss = (float)(stringftnss + nodeftnss[i][y][jj]);

      if (x[i][y][jj]  < 26.0) check1[i]++;
      if ( x[i][y][jj] > 1555.2 )       check2[i]++;
 }// end of [jj] loop
 }// end of [y] loop

     ftnss[i] = (float)(stringftnss + R1*sumconstr1[i] + R2*sumconstr2[i
 ]);
    sumftnss = sumftnss + ftnss[i];

 // Approved Fitness Function
        if(bestfit > ftnss[i]){
            bestfit= ftnss[i];
            pofi = i;

         TOTALshortage = 0.0f;
         TATALspill = 0.0f;
     for(y=0; y<YEAR; y++)  {
             SUMoutflow[y] =0.0f;
             SUMshortage[y] = 0.0f;
             SUMspill[y] = 0.0f;

         for(jj=0; jj<NNODE; jj++){


                 pRE[y][jj] = RE[i][y][jj];

                 pofx[y][jj] = x[i][y][jj];
                 SUMoutflow[y] = SUMoutflow[y] + pofx[y][jj];
             //  TOTALoutflow = TOTALoutflow+ pofx[y][jj];


                 pSH[y][jj] = SH[i][y][jj];
                 SUMshortage[y] = SUMshortage[y] + pSH[y][jj];
                 TOTALshortage = TOTALshortage + pSH[y][jj];

                 pSP[y][jj] = SP[i][y][jj];
                 SUMspill[y] = SUMspill[y] + pSP[y][jj];
                 TATALspill = TATALspill + pSP[y][jj];


         } //jj
       }   //y
                 AVERAGEshortage = TOTALshortage/YEAR;
                 AVERAGEspill = TATALspill/YEAR;

       }    //if bestfit
 } // end of i loop

  if(gen == MaxGEN-1 || DELTA_ftnss <= DELTA){
                 index = 999;
                 printf("Terminate!!! Gen%3d \nPOPSIZE %5d\n", gen+1,POP
SIZE);
```

```
                printf("bestfit = %8.3f\n",bestfit);

                fprintf(f1,"ftnss[pofi] = %8.3f check1 = %2d check2 = %
2d\n",ftnss[pofi], check1[pofi], check2[pofi]);
                printf("ftnss[pofi] = %8.3f check1 = %2d check2 = %2d\n
",ftnss[pofi],check1[pofi], check2[pofi]);


    for( y = 0; y < YEAR; y++) {

                fprintf(f1,"Inflow[%d]          ",y+1);
                printf("Inflow[%d]          ",y+1);
                for( jj = 0; jj < nnodes; jj++) {
                    fprintf(f1,"%9.3f",Qinf[y][jj]);
                    printf("%9.3f",Qinf[y][jj]);
                }
                fprintf(f1,"\n");
                printf("\n");


        /*      fprintf(f1,"area                ");
                printf("y2      =");
                for( jj = 0; jj < nnodes; jj++) {
                    fprintf(f1,"%9.4f",yy2[jj]);
                    printf("%9.4f",yy2[jj]);
                }
                fprintf(f1,"\n");
                printf("\n");   */


                fprintf(f1,"Evapo               ");
                printf("Evapo    =");
                for( jj = 0; jj < nnodes; jj++) {
                    fprintf(f1,"%9.3f",E[jj]);
                    printf("%9.3f",E[jj]);
                }
                fprintf(f1,"\n");
                printf("\n");


                fprintf(f1,"Seepage             ");
                printf("Seepage=");
                for( jj = 0; jj < nnodes; jj++) {
                    fprintf(f1,"%9.3f",L[jj]);
                    printf("%9.3f",L[jj]);
                }
                fprintf(f1,"\n");
                printf("\n");


                fprintf(f1,"Outflow[%d]         ",y+1);
                printf("Outflow[%d]         ",y+1);
                for( jj = 0; jj < nnodes; jj++) {
```

```
        fprintf(f1,"%9.3f",pofx[y][jj]);
        printf("%9.3f",pofx[y][jj]);
    }
    fprintf(f1,"\n");
    printf("\n");


    fprintf(f1,"Storage  ");
    printf("Storage  =");
    fprintf(f1,"%9.3f",s[pofi][11]);
    printf("%9.3f",s[pofi][11]);

    for(jj=0;jj < NREACH;jj++){

        fprintf(f1,"%9.3f",s[pofi][jj]);
        printf("%9.3f",s[pofi][jj]);
    }
    fprintf(f1,"\n");
    printf("\n");



    fprintf(f1,"Demand[%d]          ",y+1);
    printf("Demand[%d]          ",y+1);
    for( jj = 0; jj < nnodes; jj++) {
        fprintf(f1,"%9.3f",d[y][jj]);
        printf("%9.3f",d[y][jj]);
    }
    fprintf(f1,"\n");
    printf("\n");



    fprintf(f1,"Release[%d]         ",y+1);
    printf("Release[%d]         ",y+1);
    for( jj = 0; jj < nnodes; jj++) {
        printf("% 9.3f", pRE[y][jj]);
        fprintf(f1,"% 9.3f", pRE[y][jj]);
    }
    fprintf(f1,"\n");
    printf("\n");


    fprintf(f1,"Shortage[%d]        ",y+1);
    printf("Shortage[%d]        ",y+1);
    for( jj = 0; jj < nnodes; jj++) {
        printf("% 9.3f", pSH[y][jj]);
        fprintf(f1,"% 9.3f", pSH[y][jj]);
    }
    fprintf(f1,"\n");
    printf("\n");
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
                fprintf(f1,"Spill[%d]          ",y+1);
                printf("Spill[%d]          ",y+1);
                for( jj = 0; jj < nnodes; jj++) {
                    printf("% 9.3f", pSP[y][jj]);
                    fprintf(f1,"% 9.3f", pSP[y][jj]);
                }


                printf("\n");
                fprintf(f1,"\n");


                printf("Sum_Outflow[%d]=% 9.3f\n",y+1,SUMoutflow[y]);
                fprintf(f1,"Sum_Outflow[%d]=% 9.3f\n",y+1,SUMoutflow[y]
);


                printf("Sum_Shortage[%d]=% 9.3f\n",y+1,SUMshortage[y]);
                fprintf(f1,"Sum_Shortage[%d]=% 9.3f\n",y+1,SUMshortage[
y]);


                printf("Sum_Spill[%d]=% 9.3f\n",y+1,SUMspill[y]);
                fprintf(f1,"Sum_Spill[%d]=% 9.3f\n",y+1,SUMspill[y]);

                printf("\n");
                fprintf(f1,"\n");


}   //y

                printf("TOTAL shortage =% 9.3f\n",TOTALshortage );
                fprintf(f1,"TOTAL shortage =% 9.3f\n",TOTALshortage );

                printf("TATAL Spill    =% 9.3f\n",TATALspill );
                fprintf(f1,"TATAL Spill    =% 9.3f\n",TATALspill );
    fprintf(f1,"\n");
    printf("\n");

                printf("AVERAGEshortage    =% 9.3f\n",AVERAGEshortage )
;
                fprintf(f1,"AVERAGEshortage  =% 9.3f\n",AVERAGEshortage
);

                printf("AVERAGEspill    =% 9.3f\n",AVERAGEspill );
                fprintf(f1,"AVERAGEspill    =% 9.3f\n",AVERAGEspill );

    fprintf(f1,"\n");
    printf("\n");


    fprintf(f1,"Storage  ");
                printf("Storage  =");
                fprintf(f1,"%9.3f",s[pofi][11]);
                printf("%9.3f",s[pofi][11]);
```

```
                for(jj=0;jj < NREACH;jj++){

                    fprintf(f1,"%9.3f",s[pofi][jj]);
                    printf("%9.3f",s[pofi][jj]);
                }
                fprintf(f1,"\n");
                printf("\n");
                fprintf(f1,"\n");
                printf("\n");

    for( y = 0; y < YEAR; y++) {
                fprintf(f1,"Outflow[%d]          ",y+1);
                printf("Outflow[%d]          ",y+1);
                for( jj = 0; jj < nnodes; jj++) {
                    fprintf(f1,"%9.3f",pofx[y][jj]);
                    printf("%9.3f",pofx[y][jj]);
                }
                fprintf(f1,"\n");
                printf("\n");
     }

            fprintf(f1,"\n");
               printf("\n");

    for( y = 0; y < YEAR; y++) {
                fprintf(f1,"Shortage[%d]         ",y+1);
                printf("Shortage[%d]         ",y+1);
                for( jj = 0; jj < nnodes; jj++) {
                    printf("% 9.3f", pSH[y][jj]);
                    fprintf(f1,"% 9.3f", pSH[y][jj]);
                }
                fprintf(f1,"\n");
                printf("\n");
     }

            fprintf(f1,"\n");
               printf("\n");

for( y = 0; y < YEAR; y++) {

                fprintf(f1,"Spill[%d]          ",y+1);
                printf("Spill[%d]          ",y+1);
                for( jj = 0; jj < nnodes; jj++) {
                    printf("% 9.3f", pSP[y][jj]);
                    fprintf(f1,"% 9.3f", pSP[y][jj]);

                }
                fprintf(f1,"\n");
                printf("\n");
    }

            fprintf(f1,"\n");
               printf("\n");
```

```
            for( y = 0; y < YEAR; y++) {
                printf("Sum_Outflow[%d]=% 9.3f\n",y+1,SUMoutflow[y]);
                fprintf(f1,"Sum_Outflow[%d]=% 9.3f\n",y+1,SUMoutflow[y]
);

            }
                    fprintf(f1,"\n");
                    printf("\n");

            for( y = 0; y < YEAR; y++) {
                printf("Sum_Shortage[%d]=% 9.3f\n",y+1,SUMshortage[y]);
                fprintf(f1,"Sum_Shortage[%d]=% 9.3f\n",y+1,SUMshortage[
y]);

            }
                printf("TOTAL shortage =% 9.3f\n",TOTALshortage );
                fprintf(f1,"TOTAL shortage =% 9.3f\n",TOTALshortage );


                printf("AVERAGEshortage    =% 9.3f\n",AVERAGEshortage )
;
                fprintf(f1,"AVERAGEshortage  =% 9.3f\n",AVERAGEshortage
);
                    fprintf(f1,"\n");
                    printf("\n");

            for( y = 0; y < YEAR; y++) {
                printf("Sum_Spill[%d]=% 9.3f\n",y+1,SUMspill[y]);
                fprintf(f1,"Sum_Spill[%d]=% 9.3f\n",y+1,SUMspill[y]);

            }
                printf("TATAL Spill    =% 9.3f\n",TATALspill );
                fprintf(f1,"TATAL Spill    =% 9.3f\n",TATALspill );

                printf("AVERAGEspill    =% 9.3f\n",AVERAGEspill );
                fprintf(f1,"AVERAGEspill    =% 9.3f\n",AVERAGEspill );
                    fprintf(f1,"\n");
                    printf("\n");


    }// end of if loop

  avgftnss = sumftnss/POPSIZE;
  minftnss  = ftnss[0];
  seton = POPSIZE;
  for(i=0; i<POPSIZE; i++) {
     if(ftnss[i]  < minftnss)  minftnss  = ftnss[i];
     if(ftnss[i]  < ftnss[POPSIZE]){
         ftnss[POPSIZE] = ftnss[i];
         seton = i;
     }
  }

  for(j=0; j<LENGTH; j++){
      Q[j][POPSIZE] = Q[j][seton];
```

```
    }

   printf("Gen.%3d %10.3f %10.3f\n",gen+1,ftnss[POPSIZE],bestfit);
    fprintf(f1,"Gen%1d %10.3f %10.3f\n",gen+1,ftnss[POPSIZE],bestfit);
}
/************************************/
void Tournament()
{
    int  ii,jj,j,mem;

    for(mem=0; mem < POPSIZE; mem++){

        ii = RandInt(0,POPSIZE-1);
        jj = RandInt(0,POPSIZE-1);

        if(ftnss[ii] <= ftnss[jj]){
            for(j=0; j<NREACH; j++){
                mated_pop[mem][j] = Q[j][ii];
            }
        }

        if(ftnss[ii] > ftnss[jj]){
            for(j=0; j<NREACH; j++){
                mated_pop[mem][j] = Q[j][jj];
            }
        }
    }
}// end of tournament
/************************************/

/* Routine to crossover population, 1 site */
void XOVER(int i1, int i2)
{
    int j,kk;

    kk = RandInt(0,LENGTH-1);

    for(j=kk;j<LENGTH;j++)  {
          SwapRealValues(&mated_pop[i1][j],&mated_pop[i2][j]);
    }
}
/************************************/
/* Routine to crossover population, 2-site */
void TPXOVER(int i1, int i2)
{
    int j,kk1,kk2;

    kk1 = RandInt(0,LENGTH-1);
    kk2 = RandInt(0,LENGTH-1); /* this is for 2-site XOVER */
    if (kk1>kk2) SwapIntValues(&kk1,&kk2); /* this is to swap 2-site if
 the later is less*/

    for(j=kk1;j<kk2;j++)  {
          SwapRealValues(&mated_pop[i1][j],&mated_pop[i2][j]);
    }
```

```
}
/*********************************/
// Routine to crossover population, 1 site
void UNIXOVER(int i1, int i2)
{
    int j,kk;

    for(j=0;j<LENGTH;j++)  {
        kk= rand()&01;
        if(kk==1) SwapRealValues(&mated_pop[i1][j],&mated_pop[i2][j]);
    }
}
/*********************************/
void UniSelectMate()
{
    int  mem, num_select,one;
    float Prob_X;

    num_select = 0;
    for (mem = 0; mem< POPSIZE ; mem++) {
        Prob_X = rand()%1000/1000.0f;
        if (Prob_X < PXOVER){
            ++num_select;
            if(num_select%2 == 0){
                UNIXOVER(one,mem);
            }
            else one = mem;

        }
    }
}
/*************************/
void SelectMate()
{
    int  i,j,mem, num_select,one;
    float Prob_X;

    num_select = 0;
    for (mem = 0; mem< POPSIZE ; mem++) {
        Prob_X = rand()%1000/1000.0f;
        if (Prob_X < PXOVER){
            ++num_select;
//   printf("Prob_X = %3.2f PXOVER =%3.2f num_select = %3d\n", Prob_X,P
XOVER,num_select);
//   fprintf(f1,"Prob_X = %3.2f PXOVER =%3.2f num_select = %3d\n", Prob
_X,PXOVER,num_select);
//   getchar();
            if(num_select%2 == 0){
//              XOVER(one,mem); //for  1 point xover
                TPXOVER(one,mem); //for  2 point xover
            }
            else one = mem;

        }
    }
```

```
/****print menu******/
 if ( PrintMENU == 1) {
 fprintf(f1,"AFTER CROSSOVER\n");
    for(i=0;i<POPSIZE;i++) {
       for(j=0;j<LENGTH;j++) {
        fprintf(f1,"%7.3f",mated_pop[i][j]);
       }
       fprintf(f1,"\n");
    }
   fprintf(f1,"end\n");
 }
/******/
}
/*************************************/
void mutatepop(int gen)
{
  int i,j,mutant;
  double r;

      for (i=0;i<POPSIZE;i++) {
          for (j=0;j<LENGTH;j++) {
               r = rand()%1000/1000.0;

//this is Modified uniform mutation
               if(r<Pmutate){
               mutant = rand()&01;
                if (mutant==1)   mated_pop[i][j] = mated_pop[i][j] + (
float)Modi_muta;
                else mated_pop[i][j] = mated_pop[i][j] - (float)Modi_m
uta;
                if (mated_pop[i][j] < lower[j]) mated_pop[i][j] = lowe
r[j];
                if (mated_pop[i][j] > upper[j]) mated_pop[i][j] = uppe
r[j];
              }
/*this is non-uniform mutation */

 /*                 if(r<Pmutate){
             mutant = rand()&01;
             if(gen < MaxGEN*0.25){
                 if (mutant==1){
                   mated_pop[i][j] = mated_pop[i][j] + (float)Modi_m
uta;
                 }
                 else mated_pop[i][j] = mated_pop[i][j] - (float)Mod
i_muta;
             }
             else if((gen >= MaxGEN*0.25) && (gen < MaxGEN*0.5)){

                if (mutant==1){
                   mated_pop[i][j] = (float)(mated_pop[i][j] + (floa
t)Modi_muta/2);
                }
                else mated_pop[i][j] = (float)(mated_pop[i][j] - (fl
oat)Modi_muta/2);
```

```
                }
                else if((gen >= MaxGEN*0.5) && (gen < MaxGEN*0.75))
                    if (mutant==1){
                      mated_pop[i][j] = (float)(mated_pop[i][j] + (floa
t)Modi_muta/4);
                    }
                    else mated_pop[i][j] = (float)(mated_pop[i][j] - (f
loat)Modi_muta/4);
                }
                else {
                   if (mutant==1){
                      mated_pop[i][j] = (float)(mated_pop[i][j] + (floa
t)Modi_muta/6);
                   }
                   else mated_pop[i][j] = (float)(mated_pop[i][j] - (fl
oat)Modi_muta/6);
                }  */

                if ( mated_pop[i][j] < 0.0f)    mated_pop[i][j] = 0.0f;
                else if ( mated_pop[i][j] > upper[j]) mated_pop[i][j] =
 upper[j];
                } // close Modified uniform here

                for(j=0;j<LENGTH;j++)  {
                    Q[j][i] = mated_pop[i][j];
                }
      }
/****print menu******/
 if ( PrintMENU == 1) {
 fprintf(f1,"AFTER MUTATION\n");
   for(i=0;i<POPSIZE;i++) {
      for(j=0;j<LENGTH;j++) {
         fprintf(f1,"%7.3f",mated_pop[i][j]);
      }
      fprintf(f1,"\n");
   }
   fprintf(f1,"end\n");
   fprintf(f1,"\n");
 }

}
/*************************************/
 /* Routine to print error message */

void nrerror(char *messg)
{
    puts(messg) ;
    exit(1) ;
}
 /*************************************/
/* Routine to swop value */

void SwapIntValues(int *xx, int *yy)
{
    int temp;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
    temp = *xx;
    *xx=*yy;
    *yy=temp;
}
 /*************************************/
void SwapRealValues(float *xx, float *yy)
{
    float temp;
    temp = *xx;
    *xx=*yy;
    *yy=temp;
}
 /*************************************/
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง

ค่าของRule Curveใน Generation แรก และ Generation สุดท้าย จากโปรแกรม GAs

ใช้ตัวอย่างจากปีน้ำค่อนข้างมาก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวกนี้ได้แสดงตัวอย่างของ Rule Curve ทั้ง 120 ตัวอย่าง (Population Size) ของ generation ที่ 1 และ generation สุดท้าย และ แสดง ค่าของ Rule Curve ที่ดีที่สุดใน แต่ละ Generation โดยแสดงเป็นตัวอย่างในช่วงต้นและช่วงสุดท้ายเท่านั้น จะเห็นแนวโน้มของ ค่า Rule Curve จะให้ค่า Fitness ที่ดีขึ้น นั่นคือ Min.Z มีค่าลดลงตามฟังก์ชันวัตถุประสงค์ ดังแสดงในรูปที่ ง-1

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

แสดงค่าRule Curve ของทั้ง 120 Population Size ใน generation ที่1

231.160 554.920 453.840 580.000 328.440 750.240 563.280 472.080 931.120 552.640 735.800 310.200
413.560 828.520 930.360 573.160 956.200 915.920 828.520 531.360 497.160 659.040 885.520 316.280
421.920 490.320 519.960 744.160 745.680 880.200 539.720 751.760 785.960 608.880 860.440 893.120
706.920 427.240 226.600 879.440 734.280 816.360 444.720 453.080 711.480 704.640 307.160 740.360
392.280 859.680 615.720 689.440 703.120 775.320 228.120 852.840 749.480 763.160 602.040 791.280
440.160 226.600 344.400 839.920 418.880 280.560 230.400 915.920 400.640 692.480 538.960 811.800
876.400 754.040 481.200 466.000 204.560 276.760 498.680 616.480 678.040 673.480 263.840 925.040
774.560 838.400 934.160 485.760 907.560 434.080 917.440 533.640 675.760 445.480 608.120 608.880
289.680 262.320 906.040 611.160 833.080 287.400 685.640 700.080 735.040 906.800 942.520 432.560
711.480 493.360 215.960 766.200 902.240 254.720 405.200 830.040 790.520 635.480 273.720 589.120
949.360 420.400 322.360 683.360 469.800 782.920 697.800 636.240 223.560 239.520 466.000 314.000
915.160 750.240 934.160 526.800 281.320 345.160 205.320 456.120 547.320 418.120 772.280 491.080
918.200 890.840 358.840 776.080 367.960 646.880 520.720 918.960 584.560 222.800 513.880 327.680
884.000 649.160 779.120 697.800 511.600 472.840 674.240 608.120 616.480 567.080 652.200 231.160
657.520 466.000 421.160 835.360 484.240 215.200 652.960 215.960 464.480 351.240 707.680 567.840
413.560 757.840 240.280 959.240 517.680 912.880 884.000 798.880 296.520 554.920 753.280 878.680
692.480 567.080 813.320 519.960 435.600 668.920 817.880 590.640 434.840 668.160 910.600 542.760
656.000 389.240 594.440 622.560 806.480 430.280 370.240 206.080 841.440 662.840 951.640 733.520
348.200 568.600 270.680 460.680 597.480 646.120 438.640 582.280 540.480 352.000 548.080 669.680
640.800 804.960 806.480 413.560 647.640 806.480 206.840 319.320 558.720 672.720 608.880 421.920
228.880 336.040 344.400 699.320 928.080 345.160 819.400 874.880 318.560 588.360 353.520 681.840
406.720 241.800 449.280 690.960 475.120 873.360 865.000 529.080 860.440 307.920 841.440 516.160
869.560 958.480 444.720 694.760 215.960 731.240 623.320 561.760 877.920 495.640 257.000 741.120
656.000 587.600 202.280 860.440 854.360 722.880 504.760 799.640 393.800 521.480 201.520 644.600
338.320 416.600 266.880 523.760 668.920 775.320 832.320 908.320 328.440 317.040 747.960 343.640
941.760 450.040 479.680 725.920 523.000 621.800 529.840 617.240 535.160 589.120 310.200 245.600
745.680 772.280 305.640 521.480 412.040 956.960 722.120 602.040 617.240 532.120 858.160 921.240
346.680 348.200 425.720 516.160 417.360 279.800 570.880 414.320 545.800 757.840 286.640 732.760
440.160 709.960 797.360 399.880 437.880 469.800 340.600 240.280 893.120 814.080 832.320 918.200
437.880 774.560 443.960 624.080 690.960 946.320 565.560 309.440 348.960 368.720 298.040 322.360
606.600 542.000 331.480 554.160 233.440 700.840 421.920 533.640 392.280 218.240 317.040 587.600
766.200 693.240 341.360 437.880 560.240 216.720 327.680 213.680 798.120 887.800 928.080 497.160
353.520 675.000 562.520 514.640 438.640 826.240 453.840 864.240 482.720 320.840 833.080 253.200
570.120 425.720 593.680 334.520 787.480 405.200 779.880 707.680 345.920 948.600 277.520 564.800
361.880 676.520 809.520 275.240 600.520 675.000 612.680 902.240 217.480 938.720 246.360 337.560
202.280 528.320 583.800 650.680 751.000 223.560 573.920 307.920 368.720 417.360 248.640 884.000
342.120 473.600 513.880 940.240 405.200 329.200 378.600 833.080 740.360 777.600 880.960 706.920
416.600 618.000 306.400 727.440 728.200 674.240 214.440 295.000 637.760 727.440 700.080 429.520
481.960 554.160 715.280 650.680 846.760 567.840 213.680 552.640 290.440 315.520 808.000 266.120
245.600 903.760 207.600 775.320 329.200 439.400 637.760 372.520 232.680 776.080 324.640 282.840

870.320 265.360 629.400 570.120 638.520 560.240 675.000 676.520 678.040 905.280 521.480 595.200
885.520 931.120 293.480 652.960 760.120 398.360 348.200 599.000 400.640 397.600 353.520 288.160
222.800 447.760 208.360 785.960 512.360 615.720 316.280 595.200 800.400 902.240 342.880 779.880
914.400 846.760 703.120 830.040 884.000 741.880 928.080 639.280 477.400 205.320 562.520 352.000
244.080 533.640 430.280 777.600 471.320 446.240 562.520 282.080 285.880 874.120 808.760 846.000
549.600 525.280 954.680 491.840 507.800 610.400 284.360 735.040 834.600 470.560 254.720 466.000
825.480 568.600 622.560 364.160 675.760 471.320 599.760 471.320 456.120 405.960 860.440 474.360
880.960 216.720 668.920 285.120 744.920 728.960 644.600 231.160 521.480 298.040 374.040 629.400
624.840 908.320 424.960 849.800 240.280 931.120 643.840 757.840 697.040 938.720 547.320 480.440
604.320 931.880 661.320 567.080 892.360 682.600 250.920 844.480 713.000 912.880 369.480 307.920
773.040 588.360 763.160 333.000 548.840 827.000 367.960 861.200 675.760 909.840 355.800 795.080
846.000 502.480 412.040 732.760 346.680 757.840 684.120 605.840 622.560 954.680 333.760 735.800
931.120 616.480 869.560 428.000 513.880 687.160 849.800 849.800 307.920 551.120 664.360 866.520
522.240 715.280 771.520 536.680 424.960 711.480 230.400 437.880 865.000 254.720 821.680 663.600
212.920 908.320 285.120 728.200 328.440 831.560 230.400 570.880 720.600 268.400 577.720 647.640
952.400 310.200 468.280 438.640 694.760 762.400 233.440 396.080 454.600 776.840 345.920 659.800
400.640 337.560 582.280 830.040 789.000 662.080 421.920 957.720 617.240 622.560 626.360 676.520
554.920 611.160 298.040 382.400 817.880 332.240 656.760 258.520 363.400 719.080 361.880 953.920
826.240 656.760 497.920 776.840 709.200 525.280 220.520 263.840 257.000 797.360 578.480 937.200
418.120 843.720 659.040 582.280 367.960 703.880 736.560 475.880 207.600 329.960 571.640 382.400
324.640 611.920 670.440 893.880 649.160 735.040 821.680 376.320 770.000 355.800 941.000 609.640
430.280 520.720 274.480 387.720 643.840 692.480 937.960 856.640 893.880 257.000 614.200 741.120
614.960 715.280 784.440 399.120 594.440 948.600 419.640 917.440 857.400 610.400 386.200 586.080
441.680 861.200 656.760 445.480 300.320 558.720 315.520 266.120 633.200 779.880 884.760 278.280
521.480 600.520 656.000 936.440 211.400 629.400 221.280 612.680 463.720 266.880 916.680 684.120
510.840 551.880 237.240 717.560 646.880 459.920 662.080 245.600 367.960 776.080 925.040 874.880
310.960 724.400 921.240 840.680 526.800 671.200 768.480 250.920 607.360 795.080 226.600 371.760
340.600 228.880 848.280 678.040 370.240 768.480 901.480 472.840 395.320 782.160 917.440 925.800
441.680 751.760 512.360 219.000 469.800 200.760 617.240 576.960 643.840 591.400 932.640 459.920
257.000 893.880 307.920 348.960 920.480 254.720 523.760 660.560 331.480 526.040 507.040 735.800
675.760 817.120 485.000 270.680 629.400 227.360 759.360 307.160 818.640 955.440 394.560 695.520
911.360 836.880 566.320 469.800 211.400 299.560 374.800 839.160 675.000 208.360 684.120 341.360
724.400 694.000 703.120 681.840 878.680 468.280 516.160 543.520 206.080 399.120 377.080 545.040
430.280 681.840 430.280 394.560 312.480 294.240 440.920 361.880 282.840 221.280 352.000 260.800
441.680 852.080 238.000 317.800 474.360 400.640 886.280 713.760 688.680 890.840 885.520 626.360
571.640 920.480 414.320 696.280 712.240 367.200 505.520 901.480 831.560 480.440 867.280 396.840
206.080 670.440 937.960 202.280 918.200 793.560 583.040 497.920 720.600 437.880 730.480 647.640
748.720 912.880 228.120 511.600 550.360 377.840 586.080 930.360 928.840 574.680 591.400 404.440
912.120 860.440 244.080 732.000 937.960 400.640 288.920 363.400 621.800 819.400 450.800 229.640
361.120 418.880 262.320 925.040 264.600 739.600 567.840 788.240 488.800 819.400 922.760 611.160
287.400 716.040 283.600 882.480 255.480 798.880 942.520 300.320 926.560 723.640 285.880 206.080

915.160 800.400 749.480 475.880 221.280 339.840 791.280 352.000 253.960 872.600 940.240 253.960
453.080 858.920 316.280 424.200 327.680 827.000 713.760 678.040 694.000 654.480 434.840 726.680
721.360 260.800 288.160 389.240 706.920 601.280 716.040 856.640 519.960 507.800 827.760 820.160
592.160 751.760 706.160 266.120 717.560 932.640 458.400 721.360 215.960 703.120 747.960 248.640
434.840 515.400 885.520 863.480 294.240 915.160 766.200 779.120 521.480 603.560 812.560 403.680
441.680 657.520 889.320 433.320 565.560 209.120 303.360 678.800 286.640 814.840 263.840 622.560
420.400 422.680 956.960 315.520 241.040 462.200 738.080 388.480 573.160 741.120 299.560 286.640
533.640 928.080 748.720 735.040 956.200 239.520 404.440 564.040 380.880 521.480 897.680 858.160
700.840 578.480 569.360 348.960 551.120 681.080 320.080 216.720 310.960 497.920 228.120 903.000
691.720 548.080 657.520 813.320 274.480 830.800 421.920 656.000 411.280 807.240 467.520 540.480
870.320 610.400 439.400 637.000 779.120 630.920 455.360 501.720 517.680 881.720 829.280 846.760
820.160 374.800 541.240 903.000 700.080 374.040 595.200 914.400 625.600 311.720 323.120 697.800
713.000 801.920 474.360 773.040 502.480 310.960 742.640 918.960 342.880 632.440 684.880 703.880
257.000 591.400 595.960 561.000 667.400 601.280 377.840 633.200 887.800 552.640 623.320 931.120
322.360 598.240 617.240 556.440 450.800 901.480 466.000 453.080 903.000 891.600 760.120 455.360
456.120 411.280 498.680 683.360 742.640 324.640 649.160 921.240 302.600 583.800 456.120 203.040
456.120 673.480 704.640 937.200 662.080 631.680 413.560 264.600 315.520 483.480 695.520 347.440
865.760 827.760 500.960 634.720 389.240 686.400 332.240 822.440 916.680 664.360 915.160 419.640
518.440 629.400 811.800 644.600 364.160 542.000 667.400 662.840 248.640 326.160 878.680 256.240
586.840 428.000 728.200 635.480 647.640 322.360 330.720 935.680 472.080 223.560 403.680 523.760
587.600 520.720 788.240 792.040 891.600 619.520 338.320 497.160 576.200 780.640 864.240 476.640
885.520 393.800 549.600 560.240 938.720 823.960 292.720 615.720 638.520 799.640 659.800 348.200
651.440 922.000 460.680 773.040 565.560 209.120 710.720 533.640 525.280 893.120 779.120 934.920
510.080 515.400 890.080 369.480 776.840 529.840 355.040 569.360 442.440 928.080 918.200 812.560
326.160 732.000 478.920 725.920 798.120 604.320 622.560 940.240 539.720 215.960 415.080 368.720
451.560 485.760 643.080 920.480 749.480 946.320 213.680 789.760 367.200 284.360 338.320 850.560
572.400 903.000 446.240 569.360 714.520 936.440 688.680 605.840 714.520 707.680 251.680 953.160
348.960 795.080 829.280 752.520 523.760 861.960 729.720 665.120 734.280 220.520 510.080 614.200
586.080 340.600 380.880 380.120 536.680 437.880 580.760 846.000 297.280 284.360 694.000 313.240
345.920 545.040 860.440 717.560 553.400 402.920 741.880 802.680 681.840 558.720 938.720 830.800
884.760 535.920 334.520 866.520 785.200 733.520 476.640 489.560 648.400 825.480 380.120 217.480
336.040 652.200 328.440 448.520 231.920 435.600 338.320 244.080 903.760 570.120 709.200 601.280
694.760 396.080 361.880 853.600 795.080 417.360 763.920 663.600 558.720 297.280 529.840 839.160
745.680 582.280 858.920 857.400 912.880 869.560 395.320 770.000 666.640 654.480 548.080 702.360
247.880 774.560 813.320 411.280 571.640 530.600 477.400 257.000 645.360 493.360 833.080 473.600
450.800 236.480 905.280 573.920 529.080 838.400 782.160 758.600 815.600 655.240 836.120 877.920
946.320 449.280 467.520 480.440 385.440 803.440 662.080 391.520 691.720 528.320 606.600 358.080
400.640 577.720 384.680 693.240 211.400 839.160 343.640 276.000 817.120 692.480 597.480 846.760
560.240 681.080 877.160 352.000 849.040 952.400 729.720 898.440 792.800 639.280 907.560 613.440

Gen1 bestfit > 1000000000.000

แสดงค่าRule Curve ของทั้ง 120 Population Size ใน generation สุดท้าย

854.421 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 249.066 200.000 200.000 201.000 200.000 584.344 959.700 960.000 960.000
854.121 734.198 530.874 248.466 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 248.466 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.121 733.898 530.574 248.466 200.000 200.000 201.000 200.000 584.344 960.000 960.000 960.000
854.121 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.121 734.198 531.174 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 249.066 200.000 200.000 201.300 200.300 584.644 959.700 960.000 960.000
854.121 734.198 530.874 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.721 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.721 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 959.400
854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.274 248.766 200.000 200.000 201.300 200.000 584.944 960.000 960.000 960.000
854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.421 734.198 530.274 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 733.898 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 959.700
853.821 733.898 530.574 248.766 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.274 248.766 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.274 248.766 200.000 200.000 201.300 200.000 584.944 960.000 960.000 960.000
854.121 734.198 530.874 249.066 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.944 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
854.121 733.898 530.574 248.466 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
853.821 733.598 530.574 248.766 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
853.821 733.598 530.574 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.421 734.498 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.121 733.898 530.874 249.066 200.000 200.000 201.600 200.000 584.344 960.000 960.000 960.000
854.121 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 959.700
854.121 733.898 530.574 248.466 200.000 200.000 201.300 200.000 584.344 960.000 959.700 960.000
854.121 734.198 530.874 248.466 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 248.166 200.000 200.000 201.000 200.000 584.344 960.000 960.000 960.000
853.821 734.198 530.574 248.466 200.000 200.000 201.600 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.121 733.898 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.121 733.898 530.574 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.874 248.766 200.000 200.300 201.600 200.000 584.344 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

854.121 734.198 530.874 248.466 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 248.466 200.000 200.000 201.300 200.000 584.944 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.421 734.198 530.874 249.066 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
853.821 734.198 530.574 248.766 200.000 200.000 201.300 200.300 584.344 960.000 960.000 960.000
854.421 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.498 530.574 248.466 200.000 200.000 201.300 200.000 584.044 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
854.421 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
854.721 733.898 530.274 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 959.700
854.121 734.198 530.874 248.766 200.000 200.000 201.600 200.000 584.644 960.000 959.700 960.000
854.121 733.898 530.574 248.766 200.000 200.000 200.700 200.000 584.944 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.044 960.000 960.000 960.000
854.421 734.198 530.874 248.466 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
853.821 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.121 734.498 530.574 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.274 248.766 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.000 200.000 584.344 960.000 960.000 960.000
854.721 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 959.700 960.000
854.121 734.198 530.274 248.766 200.000 200.300 201.600 200.000 584.644 959.700 960.000 960.000
854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 531.174 248.466 200.000 200.000 201.000 200.000 584.344 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.121 734.198 530.274 248.766 200.000 200.000 201.300 200.000 584.644 959.700 960.000 960.000
854.121 733.898 530.274 248.766 200.000 200.000 201.000 200.000 584.944 960.000 960.000 960.000
854.121 734.198 530.274 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.498 530.574 248.466 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.421 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.300 200.000 201.000 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.274 248.766 200.000 200.000 201.600 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000
854.121 734.198 530.574 249.366 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.498 530.574 248.166 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 531.174 248.766 200.000 200.300 201.300 200.000 584.644 960.000 960.000 960.000
854.121 734.198 530.874 249.066 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000

854.721 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

853.821 734.498 530.874 248.466 200.000 200.000 201.600 200.000 584.644 960.000 960.000 960.000

854.121 734.198 530.874 249.066 200.000 200.000 201.300 200.000 584.644 959.700 960.000 960.000

854.121 734.198 530.574 248.766 200.300 200.000 201.000 200.000 584.944 960.000 960.000 960.000

854.121 733.898 530.574 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

854.421 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 959.700

854.121 734.198 530.574 248.466 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

854.121 733.898 530.874 248.766 200.000 200.000 201.000 200.000 584.344 960.000 960.000 960.000

854.421 733.898 530.574 248.466 200.300 200.000 201.300 200.000 584.344 960.000 960.000 960.000

853.821 734.198 530.874 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

853.821 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 959.700

854.121 733.898 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.344 959.700 960.000 960.000

854.121 734.198 530.574 248.466 200.000 200.000 201.000 200.000 584.944 960.000 960.000 960.000

853.821 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.300 584.644 960.000 960.000 960.000

853.821 734.198 530.274 248.766 200.000 200.000 201.600 200.000 584.644 960.000 960.000 960.000

854.121 734.498 530.574 248.766 200.000 200.000 201.300 200.000 584.044 960.000 960.000 960.000

853.821 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

854.121 734.198 530.274 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

854.121 734.198 530.874 248.766 200.000 200.000 201.600 200.000 584.644 960.000 960.000 960.000

854.121 734.198 530.574 248.466 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

854.121 734.198 531.174 248.466 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

854.421 734.198 530.274 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

854.121 734.198 530.574 248.466 200.000 200.000 201.000 200.000 584.344 960.000 960.000 959.700

854.121 733.898 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 959.700 960.000

854.121 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.644 960.000 959.700 960.000

854.121 734.198 531.174 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

854.121 734.198 530.874 248.466 200.000 200.300 201.000 200.000 584.644 960.000 960.000 960.000

854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

854.121 734.198 530.574 248.766 200.000 200.000 200.700 200.000 584.644 960.000 960.000 960.000

854.421 734.198 530.874 248.766 200.000 200.000 201.000 200.000 584.344 960.000 960.000 960.000

853.821 734.498 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 959.700

854.121 734.198 530.874 249.066 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000

854.121 734.198 530.574 249.066 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

854.121 734.198 530.874 248.766 200.000 200.000 201.000 200.000 584.344 960.000 959.700 960.000

854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

Gen3900 bestfit = 136806.344

แสดงการคำนวณหา Rule Curve ที่ดีที่สุดในแต่ละ Generation

Gen1 231.160 554.920 453.840 580.000 328.440 750.240 563.280 472.080 931.120 552.640 735.800 310.200

Gen2 870.320 265.360 629.400 570.120 638.820 560.240 675.000 676.520 678.040 905.580 521.480 595.200

Gen3 870.620 610.400 439.400 637.000 638.820 559.940 675.000 676.820 677.740 905.580 829.280 846.760

Gen4 429.980 681.840 430.280 394.560 312.480 468.280 515.860 543.520 206.380 398.820 695.520 347.440

Gen5 606.300 610.400 439.400 637.000 778.820 741.880 644.600 802.680 282.840 218.240 316.740 587.600

Gen6 346.220 340.300 380.880 394.560 312.480 294.240 440.920 361.880 681.840 558.720 938.720 830.800

Gen7 430.280 520.720 439.400 636.700 779.120 567.840 214.440 295.000 497.160 776.380 829.280 846.760

Gen8 585.780 340.600 380.880 433.320 513.880 216.720 515.860 266.120 297.280 284.360 510.680 613.900

Gen9 348.960 340.600 380.880 523.460 668.920 567.840 214.140 295.000 637.760 727.440 760.120 455.360

Gen10 869.720 694.000 702.820 569.820 312.780 294.240 370.540 264.600 677.740 905.280 829.280 846.760

Gen11 348.960 340.300 429.980 394.260 312.480 430.280 440.920 361.880 454.900 662.840 510.080 613.900

Gen12 348.660 340.600 429.980 394.860 312.780 430.880 440.620 361.880 454.600 893.720 700.980 429.220

Gen13 349.260 340.900 429.980 394.260 312.480 293.940 370.540 361.880 677.740 904.980 829.580 847.060

Gen14 349.560 340.600 381.180 394.560 312.480 430.280 440.920 264.000 638.660 663.140 700.680 429.520

Gen15 346.520 340.600 380.880 394.560 312.180 294.540 340.000 361.880 367.660 776.380 700.380 429.520

Gen16 346.520 340.600 380.580 394.560 312.180 294.540 370.840 205.780 682.140 662.840 700.380 429.520

Gen17 349.260 339.700 380.580 394.260 312.780 294.540 230.700 206.080 681.840 776.380 829.580 846.760

Gen18 349.260 340.000 380.880 395.160 312.480 294.240 370.540 206.080 677.740 904.980 829.880 846.760

Gen19 869.420 694.000 380.580 394.560 312.780 294.840 340.300 361.580 638.660 776.380 829.580 846.760

Gen20 349.260 340.000 380.580 394.260 313.080 294.540 230.400 205.780 681.540 727.440 701.280 428.920

Gen21 348.960 341.200 380.280 394.860 312.180 294.540 230.400 264.900 638.060 776.380 829.580 846.760

Gen22 349.560 340.600 380.880 393.960 312.480 294.540 340.000 205.780 681.840 794.780 829.580 846.760

Gen23 349.260 340.900 380.280 394.260 312.780 294.540 230.400 206.080 681.840 776.080 700.680 428.920

Gen24 349.560 340.900 380.880 393.960 293.940 294.240 230.700 206.080 367.660 776.680 700.980 428.920

Gen25 349.560 341.200 379.980 393.960 312.180 294.540 230.100 206.080 681.540 727.440 700.980 428.920

Gen26 869.720 340.900 380.880 394.260 312.780 294.840 230.700 206.380 681.840 882.020 939.020 831.100

Gen27 349.860 340.900 380.580 394.560 312.780 294.540 230.400 206.380 638.960 881.720 939.320 830.800

Gen28 349.260 341.800 380.280 394.560 313.080 295.140 230.100 206.080 681.240 727.440 700.980 428.920

Gen29 349.560 341.800 380.280 394.560 312.480 294.240 231.000 205.780 367.660 663.440 700.980 429.520

Gen30 349.860 341.800 379.980 394.260 312.780 294.540 231.000 206.080 682.140 776.380 700.680 429.520

Gen31 349.560 341.800 380.880 393.660 312.180 294.540 230.700 205.780 454.600 663.440 700.380 429.520

Gen32 869.420 342.400 380.580 394.560 313.080 295.140 230.700 205.780 681.540 881.720 939.020 830.800

Gen33 349.560 342.100 380.580 393.960 312.480 294.540 230.700 206.080 367.060 881.120 700.380 429.220

Gen34 349.560 342.400 380.580 394.260 312.780 295.140 230.700 206.680 455.200 882.020 939.320 830.500

Gen35 349.260 342.400 380.580 393.960 312.480 294.240 230.400 206.680 682.140 881.720 938.720 831.100

Gen36 869.420 693.400 380.880 394.560 312.780 294.240 231.000 206.080 455.200 882.020 939.020 830.200

Gen37 349.560 342.400 380.580 393.960 312.780 295.140 230.400 206.080 454.900 882.020 700.980 429.520

Gen38 869.120 693.100 381.180 394.560 312.780 294.840 231.000 206.080 455.200 881.420 939.020 830.500

Gen39 869.420 694.000 380.880 393.960 312.780 294.540 230.700 205.780 681.240 881.420 939.020 830.500

แสดงการคำนวณหา Rule Curve ที่ดีที่สุดในแต่ละ Generation (ต่อ)

Gen40 869.720 694.300 380.280 393.660 312.180 294.540 231.300 206.080 454.300 882.020 938.720 831.100

Gen41 869.720 693.700 380.280 393.960 312.480 294.840 230.700 205.780 638.660 881.420 939.620 831.100

Gen42 869.120 693.400 380.880 393.960 312.780 294.540 230.700 205.780 367.360 882.320 939.020 831.100

Gen43 869.120 694.300 381.180 393.960 312.480 295.140 230.400 207.280 368.260 881.120 939.320 831.100

Gen44 869.720 694.000 380.880 393.660 312.780 294.540 230.400 206.080 681.540 881.720 939.020 831.700

Gen45 868.820 693.700 381.180 393.360 312.780 294.840 230.400 206.080 681.240 881.420 939.320 830.500

Gen46 869.120 693.700 380.880 393.060 312.780 295.140 230.400 207.280 367.960 882.320 939.320 830.500

Gen47 868.520 694.000 381.180 393.960 312.780 294.240 230.700 204.880 455.200 881.420 939.620 830.800

Gen48 869.420 693.700 380.880 393.960 312.480 294.840 231.000 206.080 681.240 881.120 939.020 832.000

Gen49 868.520 693.400 380.880 393.360 312.780 294.840 230.400 205.780 455.200 881.720 939.020 831.700

Gen50 868.520 693.100 380.880 393.060 312.480 294.540 230.700 205.780 366.760 881.720 939.020 831.700

Gen51 868.520 693.100 380.880 393.060 312.480 294.540 230.700 205.780 366.760 881.720 939.020 831.700

Gen52 868.820 693.700 380.880 392.760 312.480 293.940 230.700 205.780 681.240 880.820 939.320 831.700

Gen53 868.520 693.400 381.180 392.760 313.380 294.840 230.700 205.780 681.240 882.920 938.720 831.400

Gen54 868.220 694.000 381.180 393.060 312.180 294.540 231.000 206.380 681.240 882.020 939.020 831.700

Gen55 868.520 693.700 381.180 393.060 312.480 294.240 231.000 206.380 682.740 882.020 939.320 831.700

Gen56 867.920 693.700 380.880 393.060 312.780 295.440 230.400 205.780 681.540 882.020 938.720 832.300

Gen57 868.820 693.400 381.480 392.760 312.480 294.840 231.000 206.380 681.540 882.620 938.720 832.300

Gen58 868.520 693.400 380.880 393.060 312.780 294.840 231.000 205.480 680.940 882.020 939.620 832.300

Gen59 867.920 694.300 380.880 393.360 312.480 294.240 230.700 206.080 455.200 882.020 939.020 831.700

Gen60 867.920 694.600 381.480 392.760 311.880 295.440 230.700 205.780 681.240 881.720 939.020 832.600

Gen61 867.920 694.300 381.480 392.760 312.480 293.940 231.000 206.080 681.240 882.320 939.320 833.200

| | | | | | | | | | | | | |

| | | | | | | | | | | | | |

Gen3889 854.721 734.198 530.574 248.466 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

Gen3890 854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

Gen3891 853.821 734.198 530.574 248.766 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000

Gen3892 854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

Gen3893 853.821 734.198 530.874 248.766 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000

Gen3894 854.121 734.498 530.874 248.766 200.000 200.000 201.300 200.000 584.644 960.000 960.000 960.000

Gen3895 854.121 734.198 530.874 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

Gen3896 854.121 734.198 530.874 248.466 200.000 200.000 201.000 200.000 584.644 960.000 960.000 960.000

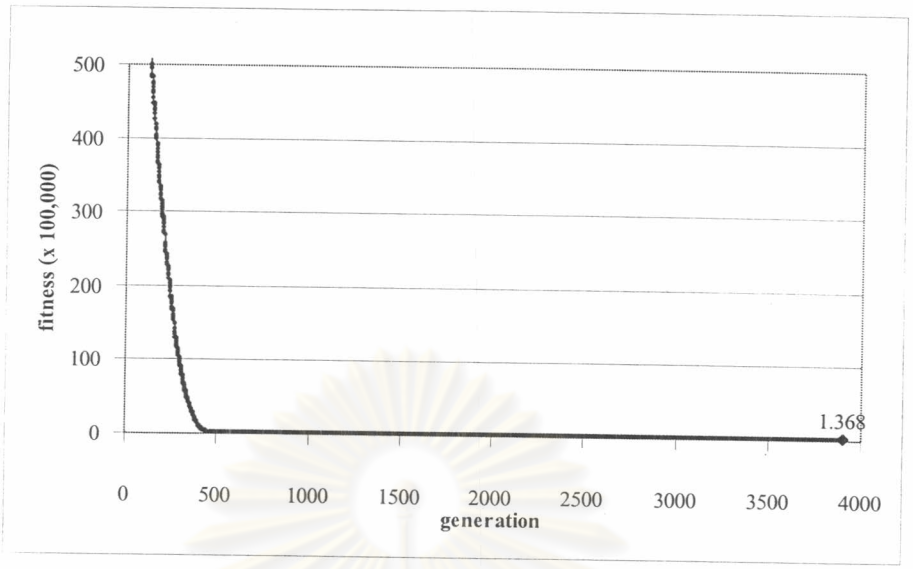
Gen3897 854.721 733.898 530.274 248.766 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

Gen3898 853.821 733.898 530.274 248.766 200.000 200.000 201.300 200.300 584.644 960.000 960.000 960.000

Gen3899 854.121 734.198 530.574 248.766 200.000 200.000 201.300 200.300 584.644 960.000 960.000 960.000

Gen3900 854.121 734.198 530.574 248.466 200.000 200.000 201.300 200.000 584.344 960.000 960.000 960.000

ftnss[pofi] = 136828.6 check1 = 0 check2 = 0



รูปที่ ง-1 แสดงแนวโน้มของค่า Fitness เมื่อ generation สูงขึ้น ของปีน้ำค่อนข้างมาก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

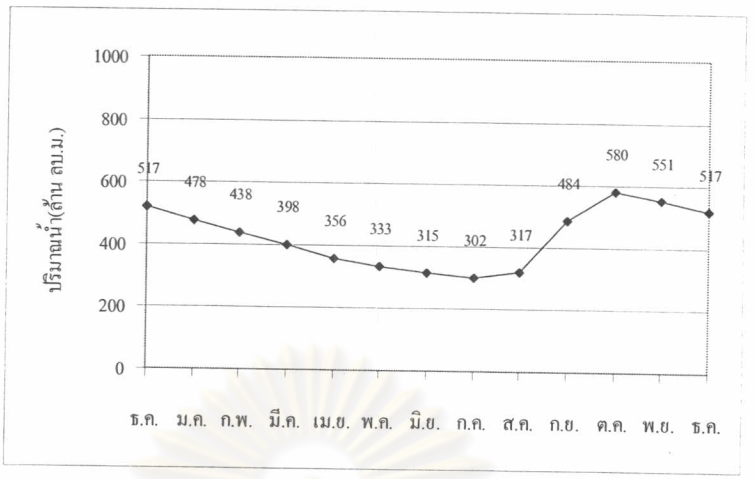
ภาคผนวก จ
ผลการศึกษา

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

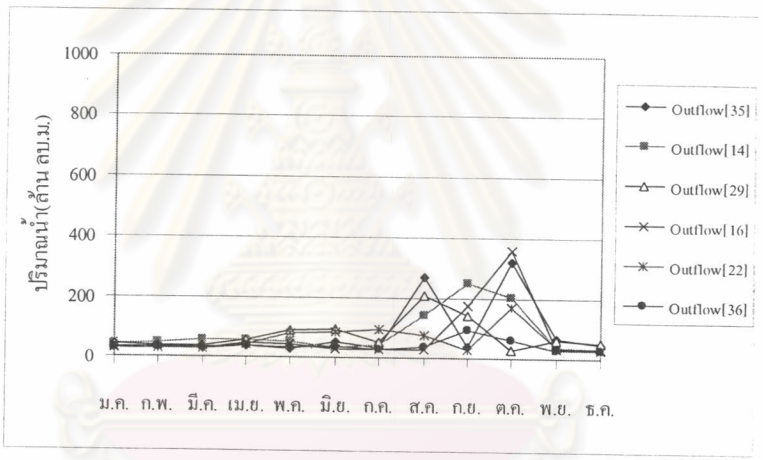
ภาคผนวกนี้ได้แสดงผลการคำนวณของโปรแกรมแสดงเป็นกราฟ ของทั้งเกณฑ์ย่อย และเกณฑ์รวมโดยแสดงดังรายการต่อไปนี้

1. ปริมาณน้ำควบคุมรายเดือนของอ่างฯเขื่อนป่าสักฯ
2. ผลการคำนวณหาอัตราการปล่อยน้ำ
3. ผลการคำนวณหาอัตราการขาดน้ำ
4. ผลการคำนวณหาอัตราการปล่อยน้ำเกินความต้องการ
5. ค่า Fitness Value

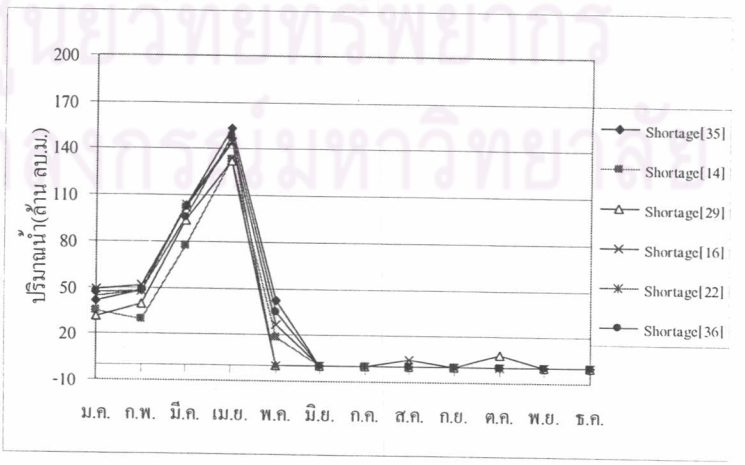
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



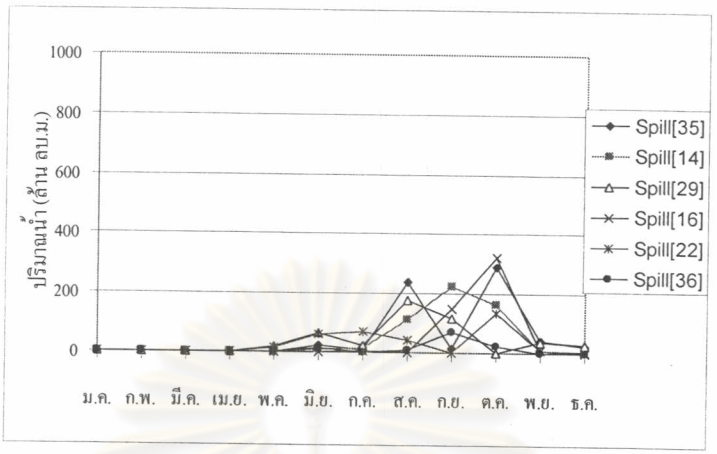
รูปที่ จ-1 ปริมาณน้ำควบคุมรายเดือนของอ่างฯเขื่อนป่าสักฯ ช่วงปีน้ำน้อย



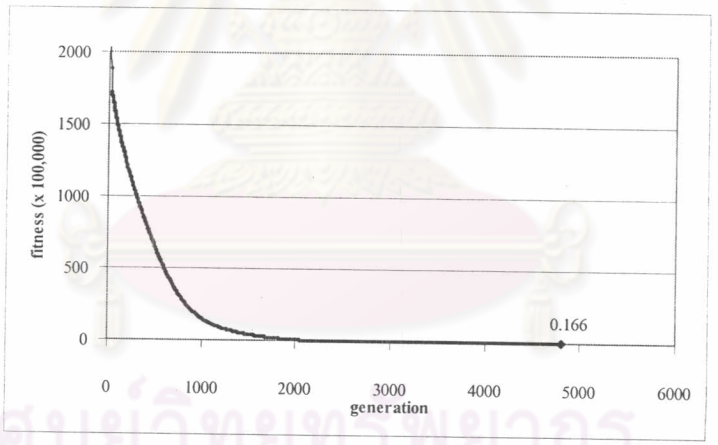
รูปที่ จ-2 ผลการคำนวณหาอัตราการปล่อยน้ำ ช่วงปีน้ำน้อย



รูปที่ จ-3 ผลการคำนวณหาอัตราการขาดน้ำ ช่วงปีน้ำน้อย

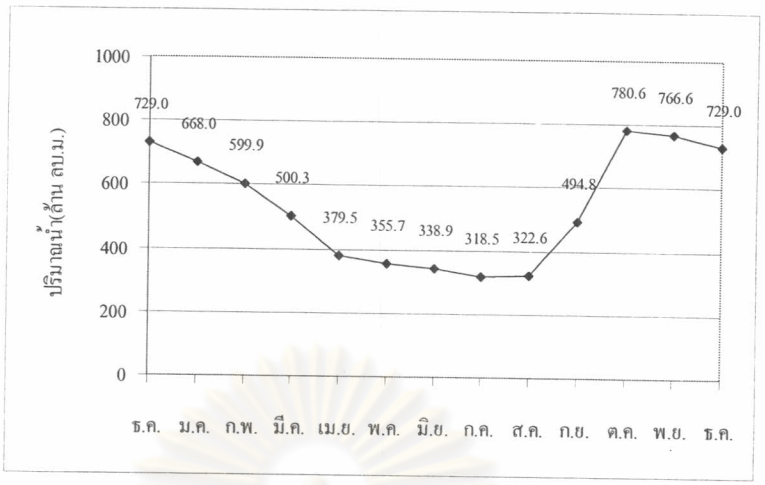


รูปที่ จ-4 ผลการคำนวณหาอัตราการปล่อยน้ำเกินความต้องการ ช่วงปีน้ำน้อย

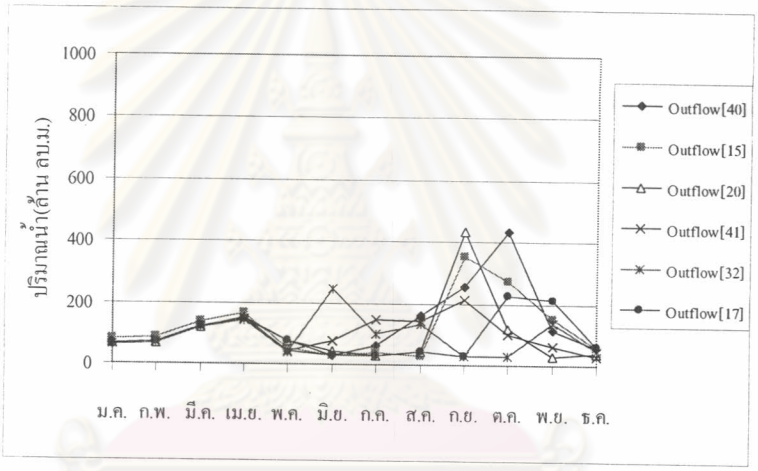


รูปที่ จ-5 ค่า Fitness Value ช่วงปีน้ำน้อย

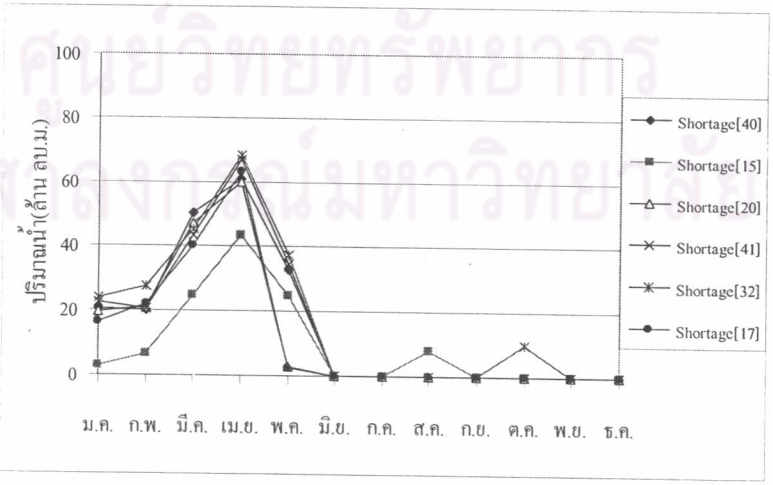
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



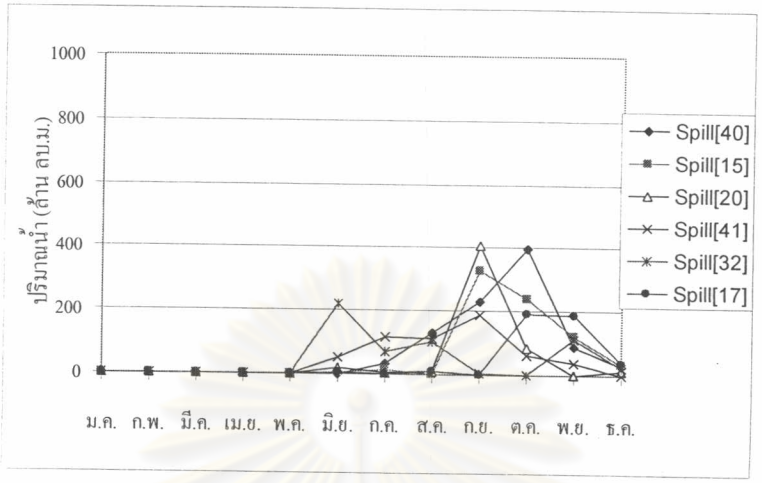
รูปที่ จ-6 ปริมาณน้ำควบคุมรายเดือนของอ่างฯเขื่อนป่าสักฯ ช่วงปีน้ำค่อนข้างน้อย



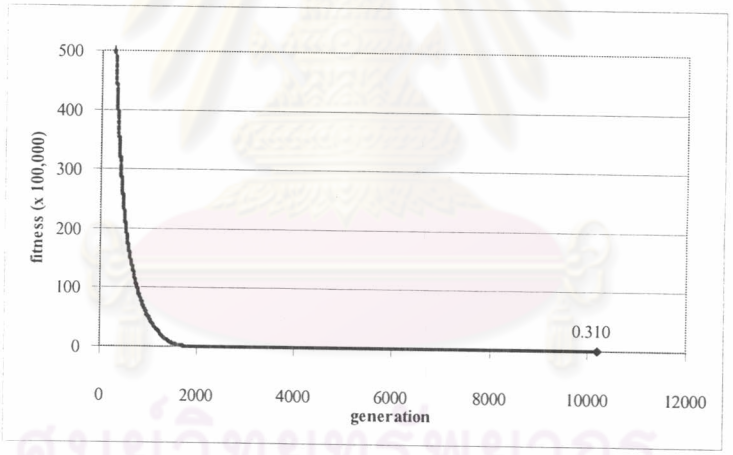
รูปที่ จ-7 ผลการคำนวณหาอัตราการปล่อยน้ำ ช่วงปีน้ำค่อนข้างน้อย



รูปที่ จ-8 ผลการคำนวณหาอัตราการขาดน้ำ ช่วงปีน้ำค่อนข้างน้อย

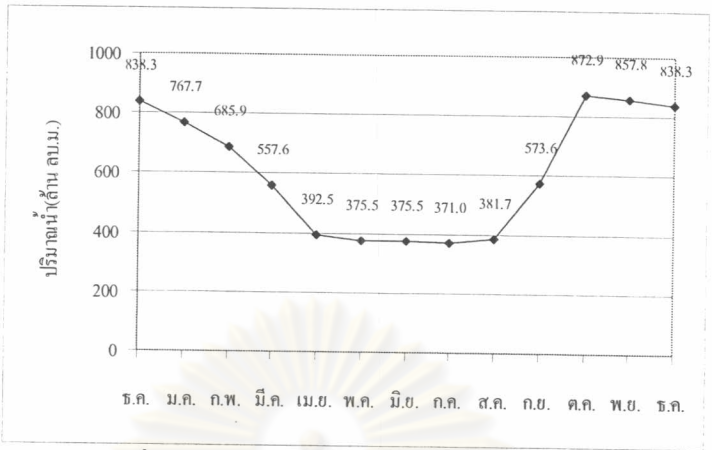


รูปที่ จ-9 ผลการคำนวณหาอัตราการปล่อยน้ำเกินความต้องการ ช่วงปีน้ำค่อนข้างน้อย

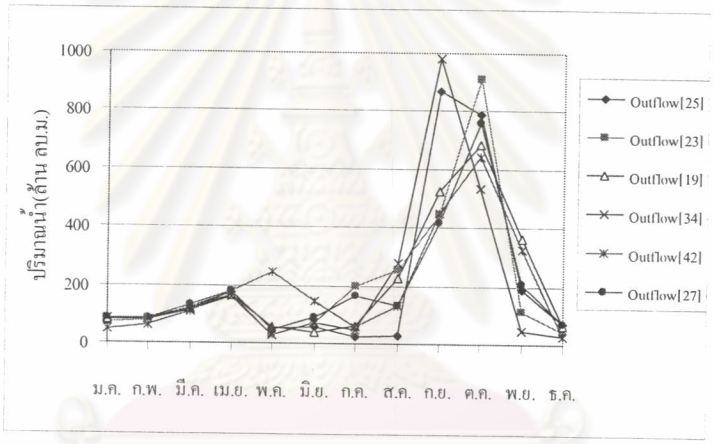


รูปที่ จ-10 ค่า Fitness Value ช่วงปีน้ำค่อนข้างน้อย

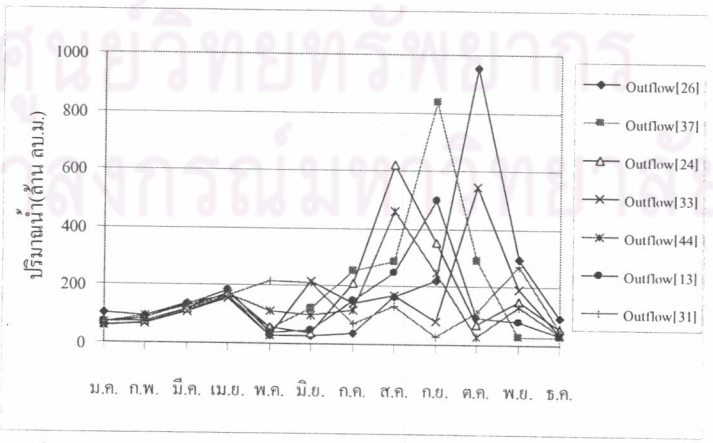
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ จ-11 ปริมาณน้ำควบคุมรายเดือนของอ่างฯเขื่อนป่าสักฯ ช่วงปีน้ำปกติ

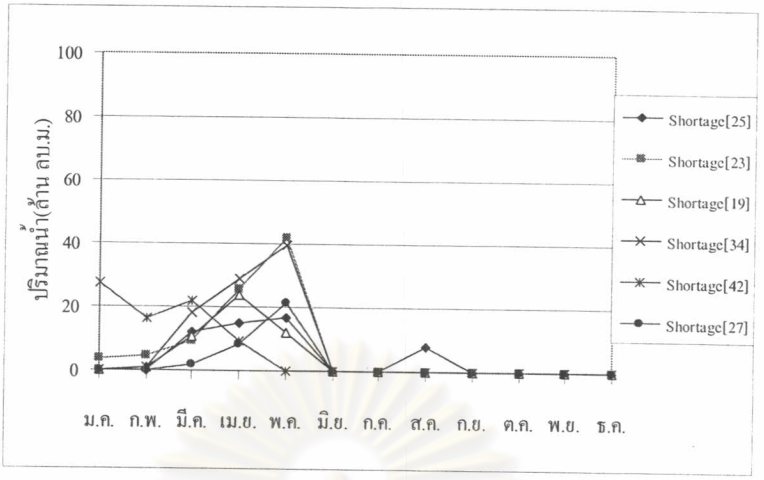


รูปที่ จ-12 ผลการคำนวณหาอัตราการปล่อยน้ำ ช่วงปีน้ำปกติ

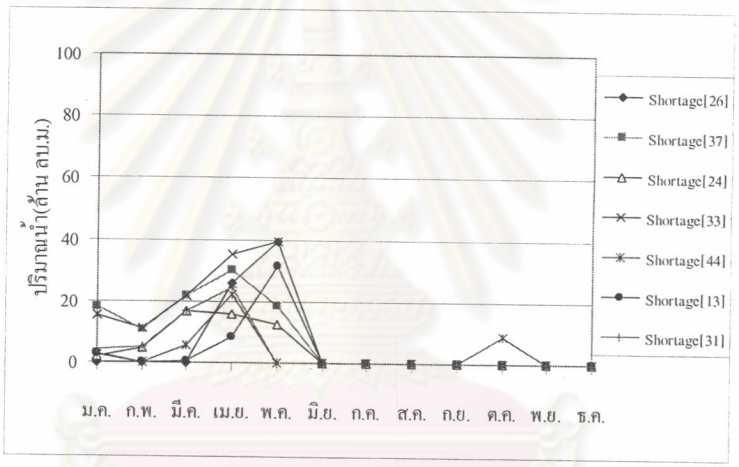


รูปที่ จ-13 ผลการคำนวณหาอัตราการปล่อยน้ำ ช่วงปีน้ำปกติ

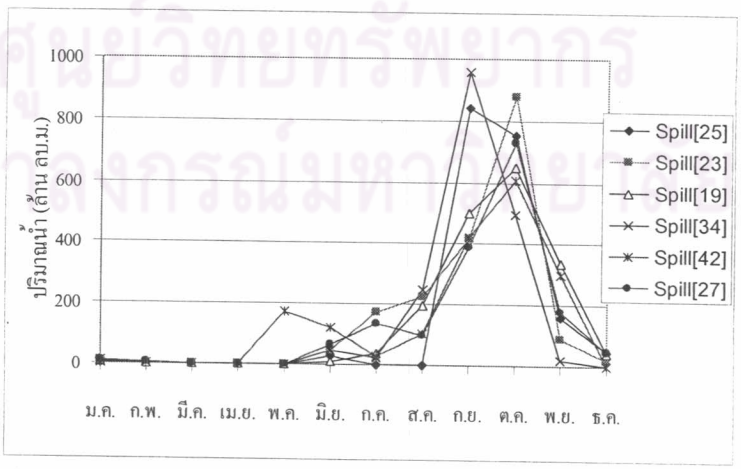
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



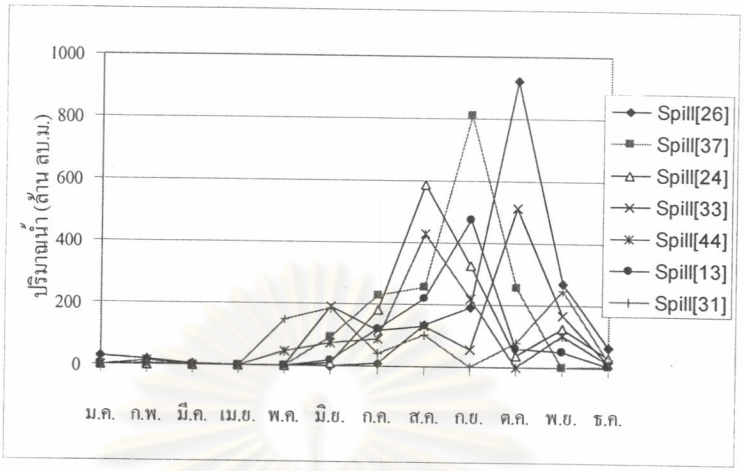
รูปที่ จ-14 ผลการคำนวณหาอัตราการขาดน้ำ ช่วงปีน้ำปกติ



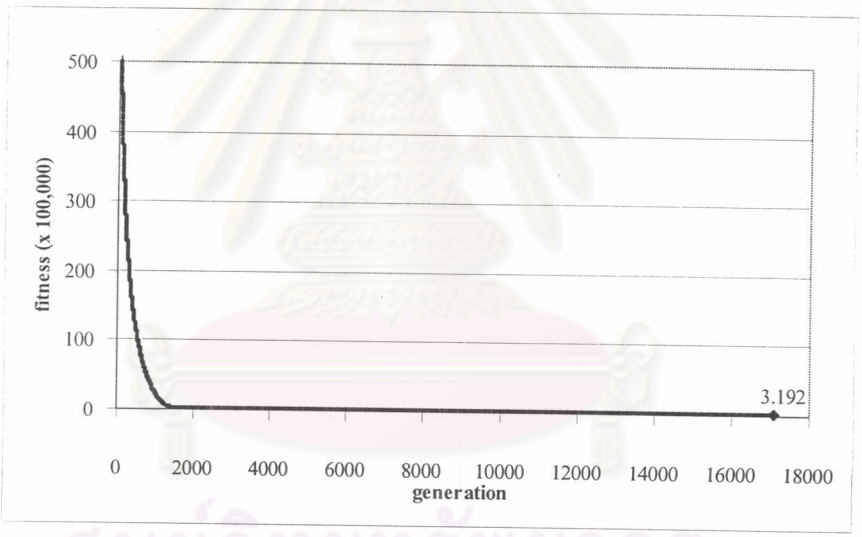
รูปที่ จ-15 ผลการคำนวณหาอัตราการขาดน้ำ ช่วงปีน้ำปกติ



รูปที่ จ-16 ผลการคำนวณหาอัตราการปล่อยน้ำเกินความต้องการ ช่วงปีน้ำปกติ

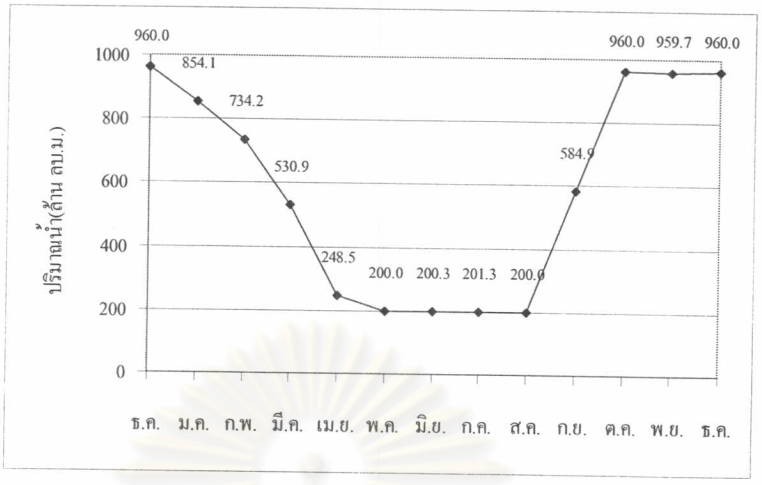


รูปที่ จ-17 ผลการคำนวณหาอัตราการปล่อยน้ำเกินความต้องการ ช่วงปีน้ำปกติ

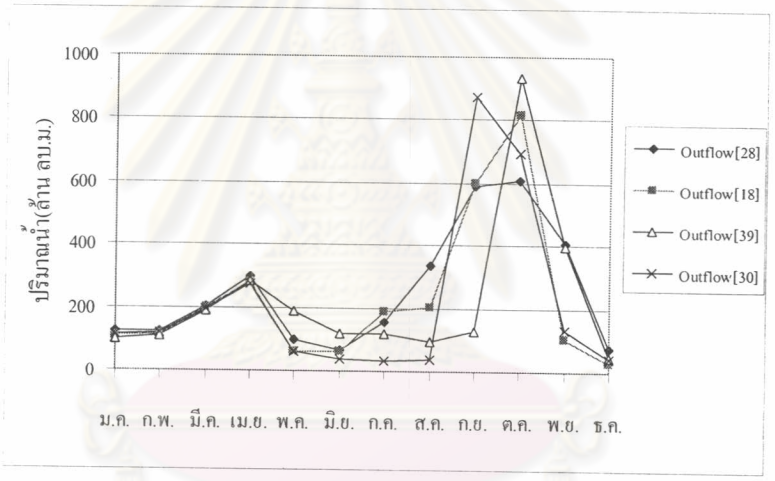


รูปที่ จ-18 ค่า Fitness Value ช่วงปีน้ำปกติ

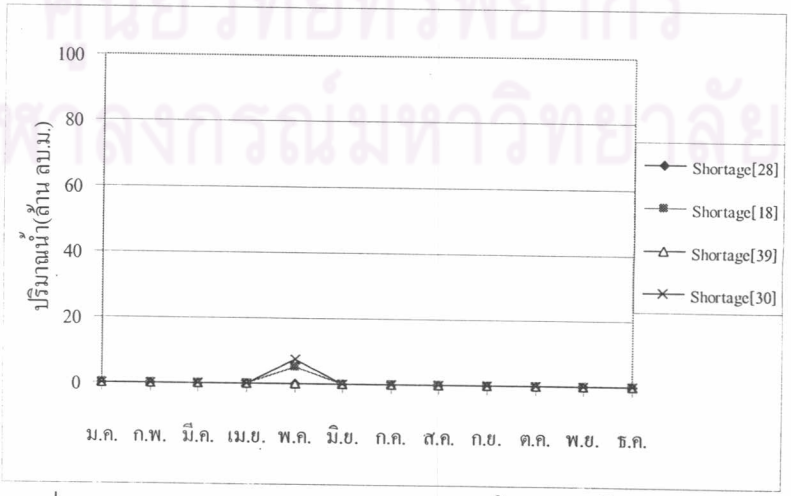
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ จ-19 ปริมาณน้ำควบคุมรายเดือนของอ่างฯเขื่อนป่าสักฯ ช่วงปีน้ำค่อนข้างมาก

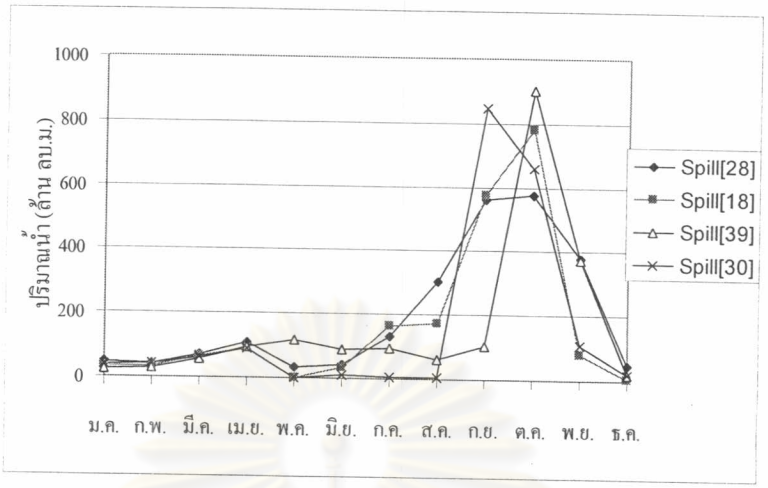


รูปที่ จ-20 ผลการคำนวณหาอัตราการปล่อยน้ำ ช่วงปีน้ำค่อนข้างมาก

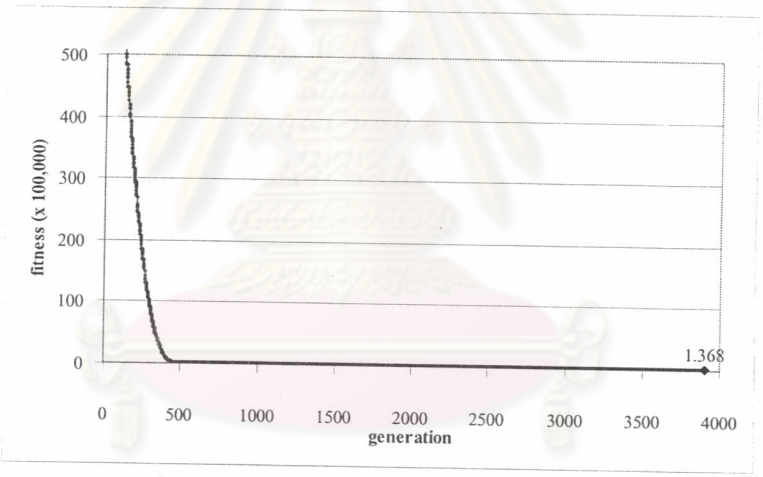


รูปที่ จ-21 ผลการคำนวณหาอัตราการขาดน้ำ ช่วงปีน้ำค่อนข้างมาก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

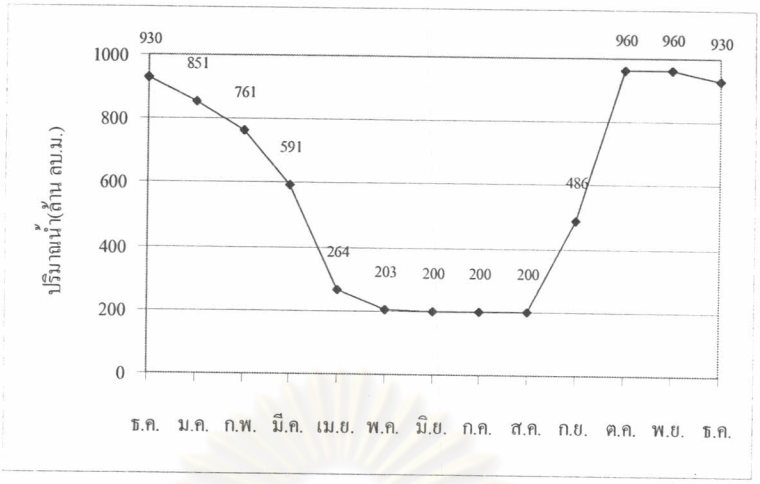


รูปที่ จ-22 ผลการคำนวณหาอัตราการปล่อยน้ำเกินความต้องการ ช่วงปีน้ำค่อนข้างมาก

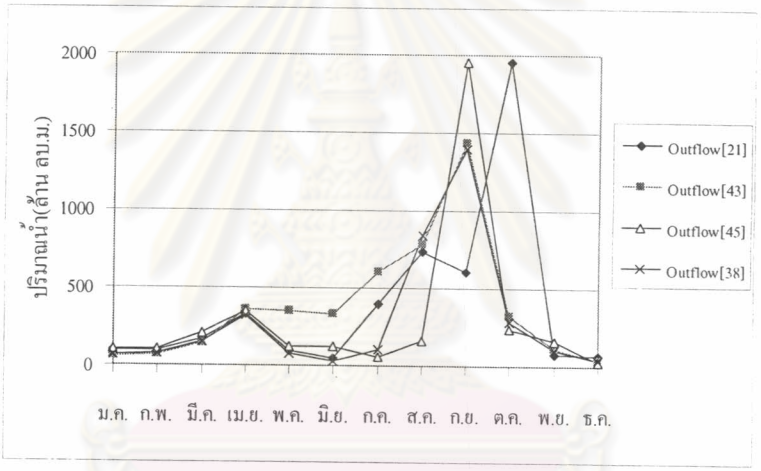


รูปที่ จ-23 ค่า Fitness Value ช่วงปีน้ำค่อนข้างมาก

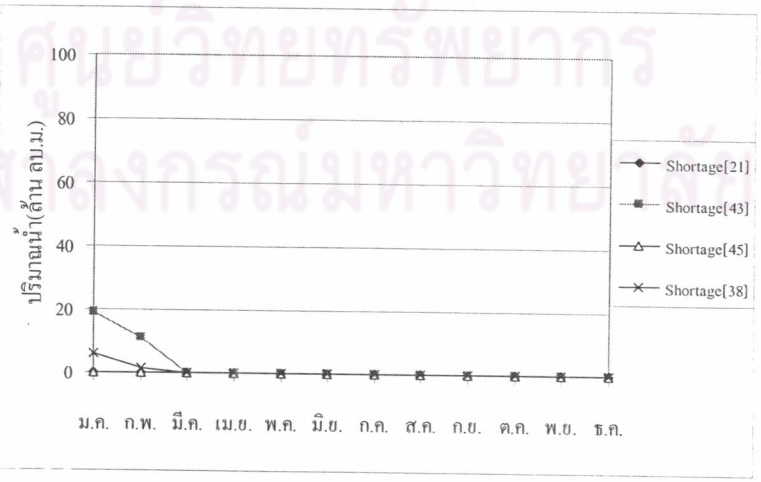
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



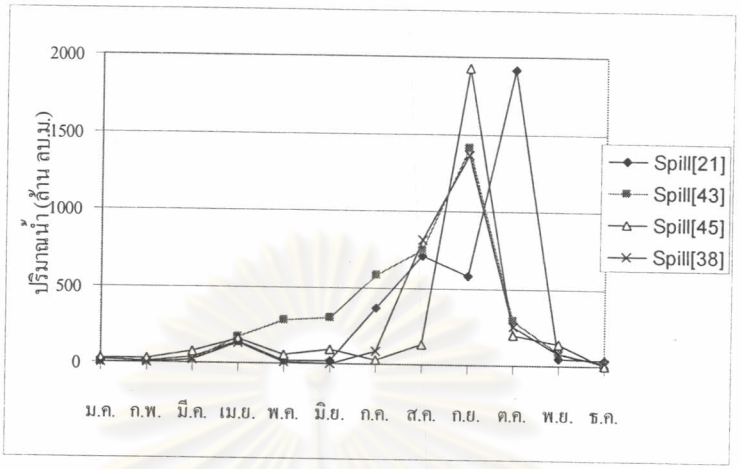
รูปที่ จ-24 ปริมาณน้ำควบคุมรายเดือนของอ่างฯเขื่อนป่าสักฯ ช่วงปีน้ำมาก



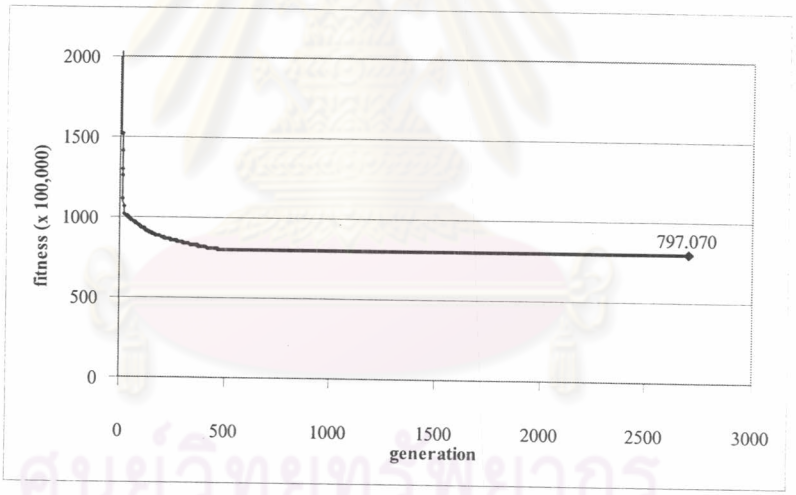
รูปที่ จ-25 ผลการคำนวณหาอัตราการปล่อยน้ำ ช่วงปีน้ำมาก



รูปที่ จ-26 ผลการคำนวณหาอัตราการขาดน้ำ ช่วงปีน้ำมาก

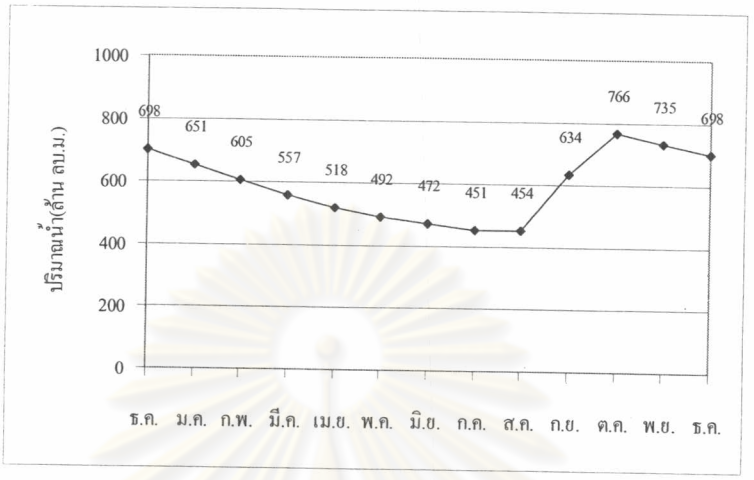


รูปที่ จ-27 ผลการคำนวณหาอัตราการปล่อยน้ำเกินความต้องการ ช่วงปีน้ำมาก

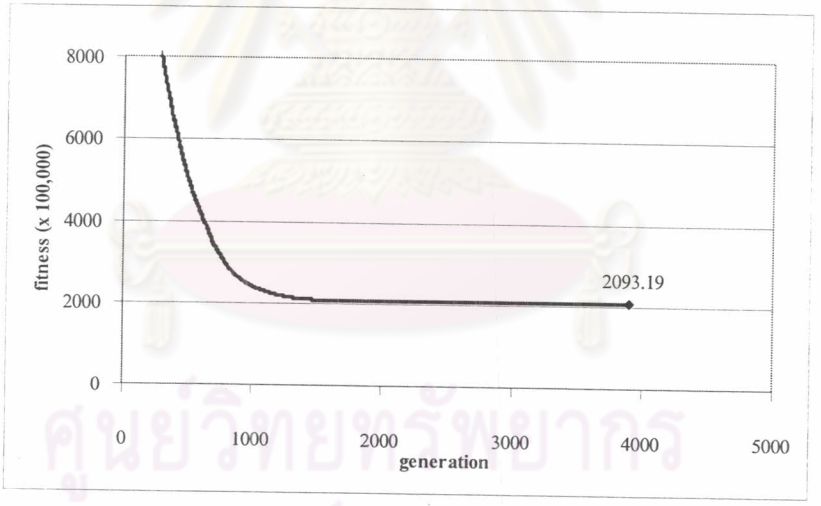


รูปที่ จ-28 ค่า Fitness Value ช่วงปีน้ำมาก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ จ-29 ปริมาณน้ำควบคุมรายเดือนของอ่างฯเขื่อนป่าสักฯ เกณฑ์รวม



รูปที่ จ-30 ค่า Fitness Value เกณฑ์รวม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

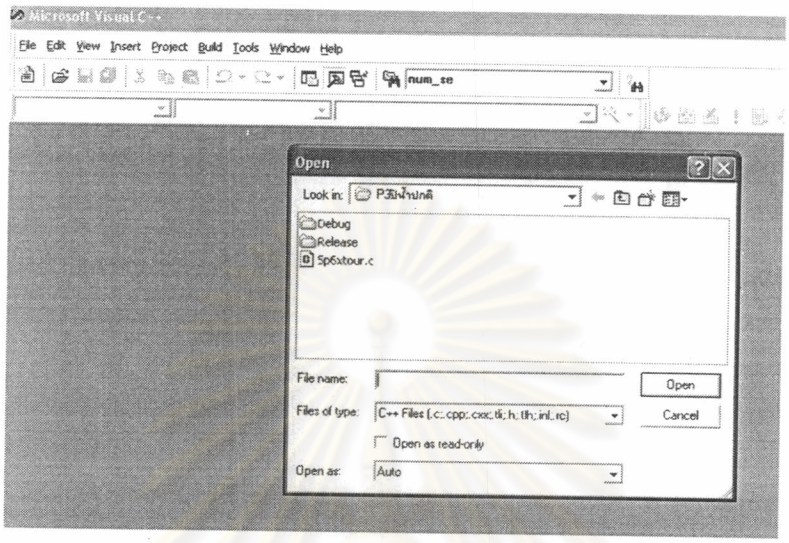
ภาคผนวก ฉ

คู่มือการใช้โปรแกรม GA โดยภาษาซี

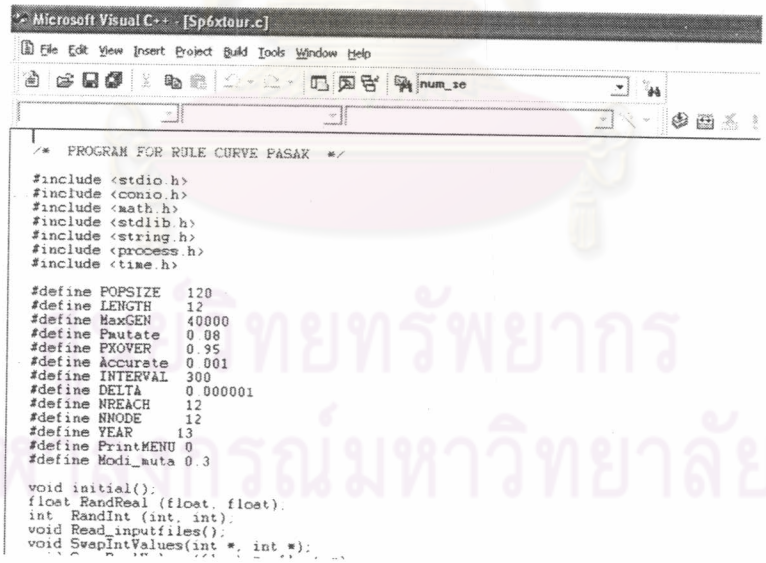
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนการใช้โปรแกรม

1.เปิดโปรแกรมดังรูปที่ ฉ-1 และ ฉ-2

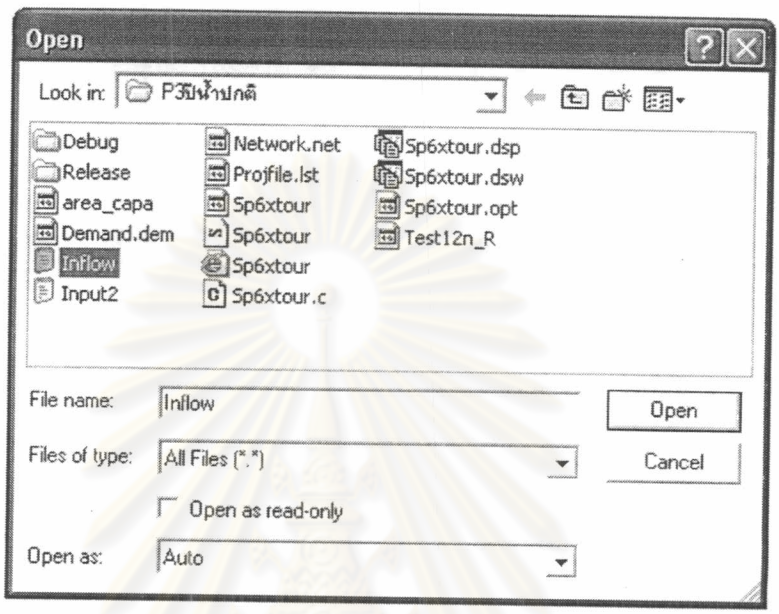


รูปที่ ฉ-1 เปิดโปรแกรม

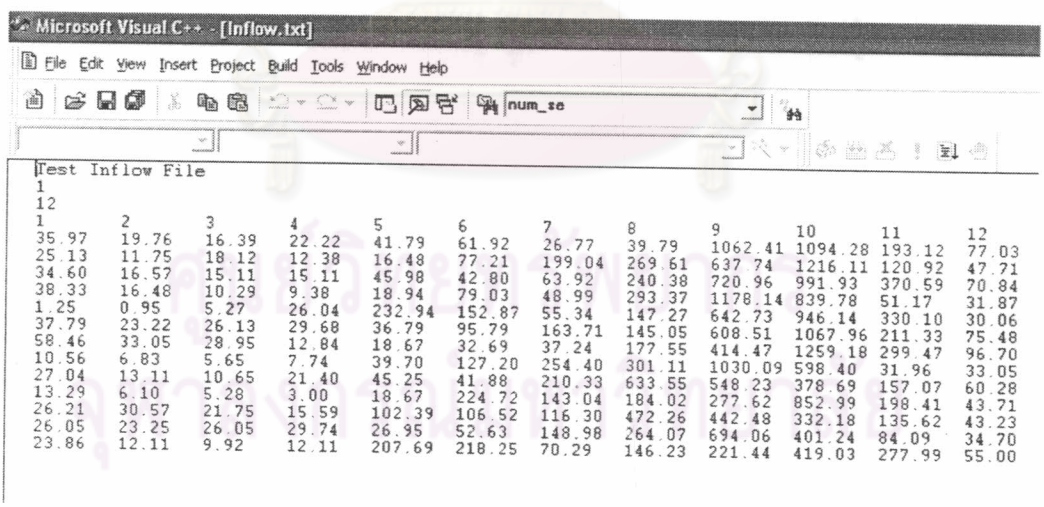


รูปที่ ฉ-2 ส่วนบนของโปรแกรม

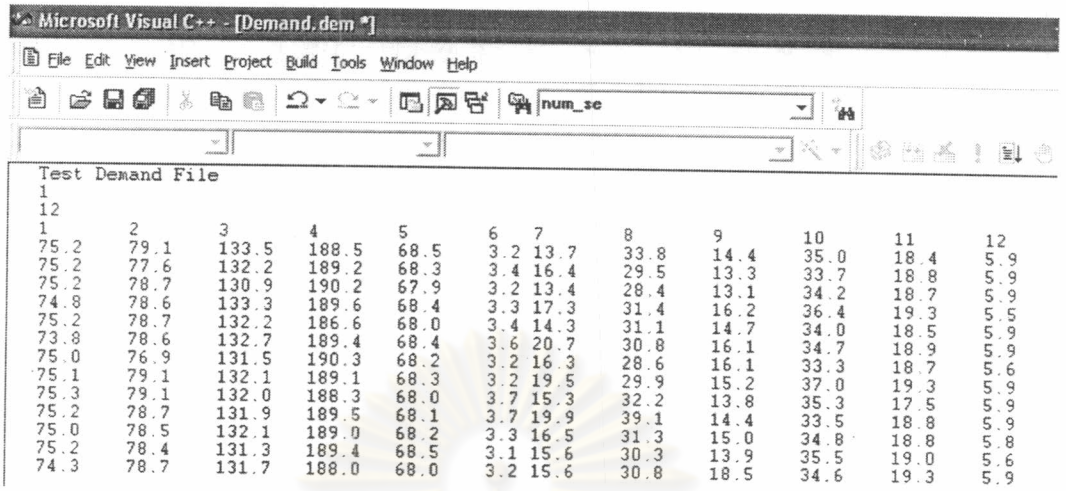
2. เปิดไฟล์ต่างๆเพื่อกรอกข้อมูล น้ำท่า ความต้องการน้ำ ความสัมพันธ์ระหว่างพื้นที่ผิวน้ำ-ความจุอ่าง อัตราการระเหย ปริมาณเก็บกักต่ำสุด-สูงสุดของอ่าง ดังรูปที่ ฉ-3 ถึง ฉ -8



รูปที่ ฉ-3 ไฟล์ต่างๆที่ใช้ในการกรอกข้อมูล

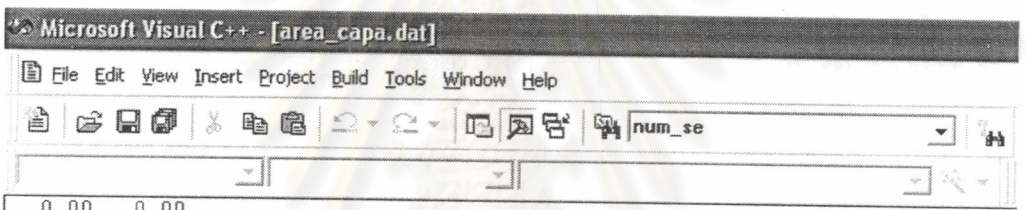


รูปที่ ฉ-4 การกรอกข้อมูลน้ำท่าในไฟล์ Inflow.txt



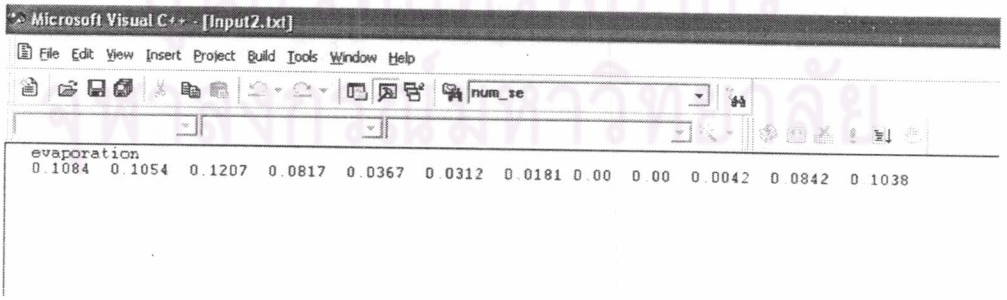

```
Test Demand File
1
12
1       2       3       4       5       6   7   8       9       10      11      12
75.2    79.1    133.5   188.5   68.5    3.2 13.7    33.8    14.4    35.0    18.4    5.9
75.2    77.6    132.2   189.2   68.3    3.4 16.4    29.5    13.3    33.7    18.8    5.9
75.2    78.7    130.9   190.2   67.9    3.2 13.4    28.4    13.1    34.2    18.7    5.9
74.8    78.6    133.3   189.6   68.4    3.3 17.3    31.4    16.2    36.4    19.3    5.5
75.2    78.7    132.2   186.6   68.0    3.4 14.3    31.1    14.7    34.0    18.5    5.9
73.8    78.6    132.7   189.4   68.4    3.6 20.7    30.8    16.1    34.7    18.9    5.9
75.0    76.9    131.5   190.3   68.2    3.2 16.3    28.6    16.1    33.3    18.7    5.6
75.1    79.1    132.1   189.1   68.3    3.2 19.5    29.9    15.2    37.0    19.3    5.9
75.3    79.1    132.0   188.3   68.0    3.7 15.3    32.2    13.8    35.3    17.5    5.9
75.2    78.7    131.9   189.5   68.1    3.7 19.9    39.1    14.4    33.5    18.8    5.9
75.0    78.5    132.1   189.0   68.2    3.3 16.5    31.3    15.0    34.8    18.8    5.8
75.2    78.4    131.3   189.4   68.5    3.1 15.6    30.3    13.9    35.5    19.0    5.6
74.3    78.7    131.7   188.0   68.0    3.2 15.6    30.8    18.5    34.6    19.3    5.9
```

รูปที่ ฉ-5 การกรอกข้อมูลความต้องการน้ำในไฟล์ Demand.dem*



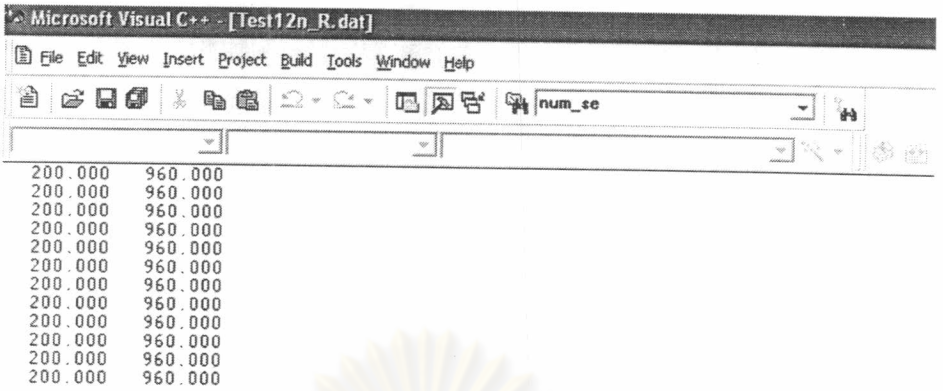

```
0.00    0.00
3.30    3.30
8.36    14.57
27.00   48.16
59.96   132.96
95.46   287.01
127.15  508.86
148.75  784.48
191.75  1124.08
252.44  1566.88
```

รูปที่ ฉ-6 การกรอกข้อมูลความสัมพันธ์ระหว่างพื้นที่ผิวอ่าง-ความจุอ่าง ในไฟล์ area_capa.dat



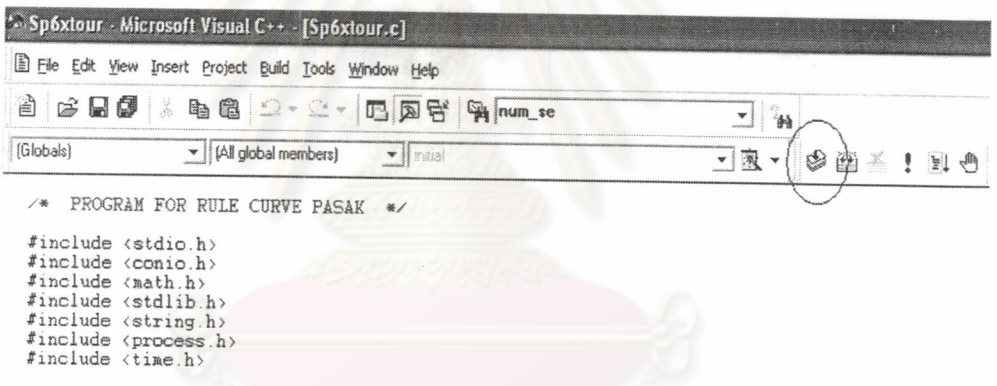

```
evaporation
0.1084  0.1054  0.1207  0.0817  0.0367  0.0312  0.0181  0.00  0.00  0.0042  0.0842  0.1038
```

รูปที่ ฉ-7 การกรอกข้อมูลอัตราการระเหยในไฟล์ Input2.txt



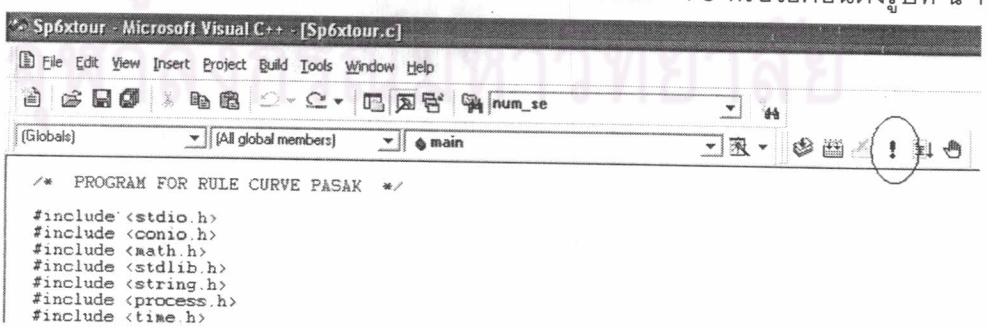
รูปที่ ฉ-8 การกรอกข้อมูลความจุต่ำสุด-สูงสุดของอ่างแต่ละเดือน ในไฟล์ Test12n_R.dat

3. Compile โปรแกรม โดยกด F7 หรือไอคอนดังรูปที่ ฉ-9 เพื่อตรวจสอบความถูกต้องของการเชื่อมโยงภายในโปรแกรม



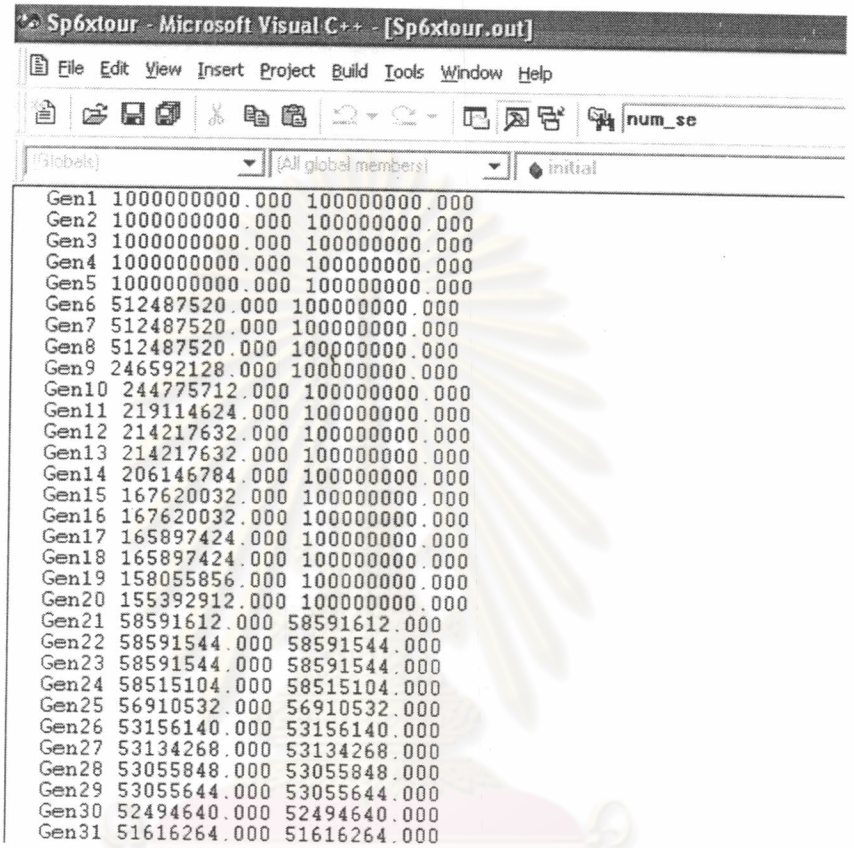
รูปที่ ฉ-9 การCompile โปรแกรม

4.Execute โปรแกรม หรือการรันโปรแกรมโดยการกด Ctrl + F5 หรือไอคอนดังรูปที่ ฉ-10



รูปที่ ฉ-10 การรันโปรแกรม (Execute Program)

5. เปิดไฟล์ผลการรันโปรแกรมชื่อ Sp6xtour.out ดังรูปที่ ฉ-11 และเปิดไฟล์ผลการรันด้วยโปรแกรม EXCEL อีกครั้งหนึ่งเพื่อนำผลลัพธ์ที่มาใช้ได้สะดวกขึ้น



รูปที่ ฉ-11 ผลการรันโปรแกรม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายธนชาติ สุขอนันตวงษ์ เกิดวันที่ 20 มิถุนายน 2518 ที่กรุงเทพมหานคร สำเร็จการศึกษา วิศวกรรมศาสตรบัณฑิต (วศ.บ.) จากวิทยาลัยการชลประทาน ในปีการศึกษา 2540 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต (วศ.ม.) ภาควิชาวิศวกรรมแหล่งน้ำ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยในปีการศึกษา 2544

ปัจจุบันรับราชการ ที่กรมชลประทาน ส่วนวิศวกรรมบริหาร สำนักชลประทานที่ 14 จังหวัด ประจวบคีรีขันธ์

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย