

## n/m 흐름作業의 Heuristic 技法에 관한 研究

(A Study on the Heuristic Algorithm for n/m Flow-Shop Problem)

李 根 富 \*

### Abstract

This paper analyzed and developed flow-shop sequencing heuristic method. The essence of the heuristic approach is in the application of selective routine that reduce the size of a problem. The advantages of this approach are consistency, Speed, endurance and the ability to cope with more data and larger systems than is humanly possible. In recent years many heuristic procedures have been suggested for the flow-shop sequencing problem. Although limited comparisons of these procedures have been made, a full scale test and evaluation have not been reported previously. The maximum flow-time criterion is selected as the evaluation criterion is selected as the evaluation criterion of flow-shop's efficiency. The author evaluated these 3 heuristic method's performance. By the evaluation of the result, we can see that the modified methods produce a shorter maximum flow-time than the original methods.

### 1. 序 論

#### 1.1 研究目的

본 研究는 生産管理의 핵심이라 할 수 있는 Job Shop scheduling을 Heuristic 技法으로 解決하는 방안을 제시한다.

특히 顧客의 注文에 따라 生産하는 形態인 flow shop의 生産性과 効率性을 높이기 위한 注文이나 作業의 順序를 調整함으로써 日程計劃의 目標인 納期遵守 및 더 나아가 納期短縮을 시행함으로써 市場競爭力을 강화하며 企業의 信用度를 높일 수 있는 방안을 經濟적으로 解決하기 위해 最適解는 아니지만 이에 근사한 最適解를 구하는 技法을 연구하였다.

#### 1.2 研究方法

Job shop scheduling의 處理順序의 數는 注文된 作業이 n개인 경우 n!만큼 된다. 이때 각 作業順序의 効率性을 評價하여 그중 가장 좋은 순서를 선택하면 가장 좋은 處理順序를 구할 수 있다. 바로

E.J. Ignall와 L.E. Schrage<sup>1)</sup>가 개발한 Branch & Bound 技法이다.

이 方法은 作業場의 工程數가 많아지거나 注文의 數가 많아지면 最適解를 구하기 어려운 단점이 있다.

본 연구는 이를 보완하기 위해 근사한 解를 쉽게 구할 수 있는 Heuristic 技法을 개발하고 이를 기존 기법과 그 効率을 비교하고자 한다.

본 論文은 N/M/F/F<sub>max</sub> 문제일 경우에 D.G. Dannenbring<sup>2)</sup>이 비교 연구한 여러 技法중에서 Close Order Search와 Extensive Search의 概念을 다른 技法들에도 확장·적용하여 그 有効性을 比較·分析하였다.

1) E. G. Ignall, and L. E. Schrage, "Application of the Branch and Bound Technique to some Flow-shop Scheduling Problems", *Operations Research* 13(3), 1965.

2) D. G. Dannenbring, "An Evaluation of Flow-shop Sequencing Heuristics", *Management Science*, July 1977, Vol. 23, No. 11, pp. 1174 ~ 1182.

\* 淸州大學校 理工大學 專任講師

2. 問題의 Model化

2.1 問題의 假定

문제의 모델(model)化를 위해 아래와 같은 假定들을 한다.

- ① Job shop에 대한 注文은 동일한 priority를 갖는다.
- ② 한 製品은 동시에 다른 工程에서 處理될 수 없다.
- ③ 한 工程은 동시에 둘 이상의 注文을 處理할 수 없다.
- ④ 注文들의 Due time은 고려하지 않는다.
- ⑤ 注文이 각 公程에서 발생되는 set up time은 加工時間에 포함된다.

2.2 比較된 發見的 Heuristic 技法들

$N \times 3$  흐름작업장 문제의 解를 구하는 發見的 技法중 이 연구에서 比較된 技法들은 아래와 같다.

(1) Slope Order Index技法(SI)<sup>3)</sup>

$$SI_i = -\sum_{j=1}^m \{ [m - (2j - 1)] / 2 \} p_{ij}$$

(m : 公程수)

이때  $P_{ij}$  : i 번째 작업을 j 번째 公程에서 처리하는 데 걸리는 시간  $SI_i$ 의 값이 決定되면 큰값의 注文부터 크기순으로 처리한다.

(2) Campell, Dudex and Smith技法(CDS)<sup>4)</sup>

아래 式에 의해(m-1)個의, 즉 2 工程의 문제로 變換시킨후 各各의 解를 구하고 그중 나은 解를 選擇한다.

$$P_{i1}^k = \sum_{j=1}^k P_{ij}$$

$$P_{i2}^k = \sum_{j=m-k+1}^m P_{ij} \quad (m = \text{公程수})$$

(3) Rapid Access Procedure (RA)

3) D. S. Pamer, "Sequencing Jobs through a Multi-stage Process in the Minimum Total Time - A Quick Method of Obtaining a Near Optimum," *Operation Research Quarterly*, Vol. 16, pp. 101 ~ 107.

4) H.G. Campell, R.A. Duck, and M.L. Smith, "A Reuristic Algorithm for the n Job, m Machine Sequencing Problem", *Management Science*, Vol. 16, pp. 214 ~ 221.

아래 式에 의해 그 工程의 문제로 變換시킨후 解를 구한다.

$$P'_{i1} = \sum_{j=1}^m (m-j+1) P_{ij}$$

$$P'_{i2} = \sum_{j=1}^m (j) P_{ij}$$

(4) SICS, CDSCS, RACS技法

이 技法들은 上記한 SO, CDS, RA 技法들에 의해 구한 解에 Close Order Search 절차를 적용해서 解를 구한다.

⑦ 注文이 n個인 경우 이미 구한 解에서 인접하는 注文變들[(n-1)개]을 각각 한번씩 교환하여 (n-1개)의 새로운 解를 구한다.

⑨ 원래의 解와 이 (n-1개)의 새로운 解들의 最大 흐름시간을 比較해서 그중 가장 좋은 解를 選擇한다.

(5) SOES, CDSES, RAES技法

上記한 SO, CDS, RA 技法들에 의해 구한 解에 아래의 Extensive Search 절차를 적용해서 解를 구한다.

⑦ Extensive Search

Close Order Search를 하여 解가 改善이 될 경우에 계속 Close Order Search 절차를 적용해서 점차 나은 解를 구한다. 더 이상 改善되지 않으면 計算을 끝낸다.

2.3 Heuristic 技法들의 比較

D.G. Dannenbrig은 여러 技法들의 Performance 評價基準을 아래와 같이 결정 使用하였다.

(1) Relative error

- ⑦ 最適處理順序 및 豫測值(R\*)
  - ⑨ 구할 수 있는 최선의 處理順序(R<sub>min</sub>)
  - ⑨ 최선의 lower bound (R<sub>max</sub>)
- $$R = 100 [ 1 - (MS / MS^*) ]$$

(2) Consistency : (C)

$$C = R^2$$

(3) Error Potential Ratio (P)

$$EPR = 100 (MS - MS^* / MS - MS^*)$$

(4) 가장 좋은 解를 낳을 確率(P)

$$P = 100 (V / N)$$

(5) 加重된 順位(W)

$$W = \sum_{j=1}^m (j) b_j$$

윗式에서 使用된 各 기호는 아래와 같다.

MS : 각 發見的 技法들에 의해 구한 解의 最大 흐름시간

MS\*: MS 중 가장 좋은 解의 최대흐름시간

V: 각 技法들이 MS\*와 같은 解를 낳은 回數

N: 생성된 문제의 갯수

$b_j$ : j 번째 좋은 해를 갖는 回數(j=1: 최적해) 위의 評價基準들을 사용하여 Dannenbring이 각 Heuristic 技法들의 效率性を 비교한 결과가 아래와 같다.

	R* (%)	C* (%)	EPR	P* (%)	I	S
RAES	0.04	3.18	2.32	75.86	1.33	8.40
RACS	1.30	7.87	4.66	63.13	3.02	10.28
CDS	1.73	11.66	5.68	55.47	3.09	10.50
M	1.74	11.31	5.97	56.88	3.92	11.29
R <sup>1)</sup>	2.03	14.52	7.12	45.00	4.12	11.68

1) 주문의 수가 3개, 4개, 5개, 6개인 경우에 표본의 수를 각각 6개, 10개, 40개, 100개를 취하였다.

上記한 表에 의하면 D.G. Dannenbring이 개발한 RAES 技法에 의한 處理順序가 다른 技法에 의한 處理順序보다 그 效率이 높음을 알 수 있다.

3. 기존 Heuristic 技法의 改善案

3.1 着眼點

n個의 注文時 2個의 注文으로 이루어지는  $nC_2$ 個의 각각 注文變에 대해 흐름시간이 적어지게 處理

順序를 정한 후 전체를 종합하면 2 공정인 경우 Johnson<sup>5)</sup> 技法과 같다. 따라서 2 공정에 적용되는 性質을 3 공정의 경우로 확장하고자 하는 것이 주된 着眼點이다.

그러나  $nC_2$ 個 전체의 各各 흐름시간을 구해 서로 比較하고 종합해야 하므로 Dannenbring가 개발한 RA 技法에 의한 處理順序를 基礎로 인접하는 注文變에 上記한 性質을 적용시켜 順序를 改善한다.

3.2 Rapid Access Procedure (개선한 RA 技法)

이 技法은 각 注文들의 工程時間에 加重値를 주어서 m-공정의 경우를 2-공정의 경우로 變換시킨 후 Johnson 技法을 적용하는 것으로 아래 과정과 같이 注文處理順序를 구한다.

- ① RA 技法에 의해 初期處理順序를 구한다.
  - ② i 번과 i+1 번의 注文을 a, b라 할 때 a-b, b-a의 흐름시간을 비교후 적으면 i 번째에 b를 놓고 그 다음번에 a를 배치하여 처음의 RA處理順序를 變換시킨다.
  - ③ RA 順序가 變化하지 않으면 i+1 번째와 i+2 번을 같은 과정으로 乘換시킨다.
  - ④ 모든 比較후 處理順序의 최대흐름시간과 원래 RA處理順序의 최대흐름을 비교하여 적은 값을 갖는 處理順序를 최종처리순서로 選擇한다.
- 上記한 技法의 Computer Program은 아래와 같다.

FORTRAN-77 304-03

```

PROGRAM VIND;
  DIMENSION A(3,3), B(3,3), C(3,3), D(3,3), E(3,3), F(3,3), G(3,3), H(3,3), I(3,3), J(3,3), K(3,3), L(3,3), M(3,3), N(3,3), O(3,3), P(3,3), Q(3,3), R(3,3), S(3,3), T(3,3), U(3,3), V(3,3), W(3,3), X(3,3), Y(3,3), Z(3,3), AA(3,3), AB(3,3), AC(3,3), AD(3,3), AE(3,3), AF(3,3), AG(3,3), AH(3,3), AI(3,3), AJ(3,3), AK(3,3), AL(3,3), AM(3,3), AN(3,3), AO(3,3), AP(3,3), AQ(3,3), AR(3,3), AS(3,3), AT(3,3), AU(3,3), AV(3,3), AW(3,3), AX(3,3), AY(3,3), AZ(3,3), BA(3,3), BB(3,3), BC(3,3), BD(3,3), BE(3,3), BF(3,3), BG(3,3), BH(3,3), BI(3,3), BJ(3,3), BK(3,3), BL(3,3), BM(3,3), BN(3,3), BO(3,3), BP(3,3), BQ(3,3), BR(3,3), BS(3,3), BT(3,3), BU(3,3), BV(3,3), BW(3,3), BX(3,3), BY(3,3), BZ(3,3), CA(3,3), CB(3,3), CC(3,3), CD(3,3), CE(3,3), CF(3,3), CG(3,3), CH(3,3), CI(3,3), CJ(3,3), CK(3,3), CL(3,3), CM(3,3), CN(3,3), CO(3,3), CP(3,3), CQ(3,3), CR(3,3), CS(3,3), CT(3,3), CU(3,3), CV(3,3), CW(3,3), CX(3,3), CY(3,3), CZ(3,3), DA(3,3), DB(3,3), DC(3,3), DD(3,3), DE(3,3), DF(3,3), DG(3,3), DH(3,3), DI(3,3), DJ(3,3), DK(3,3), DL(3,3), DM(3,3), DN(3,3), DO(3,3), DP(3,3), DQ(3,3), DR(3,3), DS(3,3), DT(3,3), DU(3,3), DV(3,3), DW(3,3), DX(3,3), DY(3,3), DZ(3,3), EA(3,3), EB(3,3), EC(3,3), ED(3,3), EE(3,3), EF(3,3), EG(3,3), EH(3,3), EI(3,3), EJ(3,3), EK(3,3), EL(3,3), EM(3,3), EN(3,3), EO(3,3), EP(3,3), EQ(3,3), ER(3,3), ES(3,3), ET(3,3), EU(3,3), EV(3,3), EW(3,3), EX(3,3), EY(3,3), EZ(3,3), FA(3,3), FB(3,3), FC(3,3), FD(3,3), FE(3,3), FF(3,3), FG(3,3), FH(3,3), FI(3,3), FJ(3,3), FK(3,3), FL(3,3), FM(3,3), FN(3,3), FO(3,3), FP(3,3), FQ(3,3), FR(3,3), FS(3,3), FT(3,3), FU(3,3), FV(3,3), FW(3,3), FX(3,3), FY(3,3), FZ(3,3), GA(3,3), GB(3,3), GC(3,3), GD(3,3), GE(3,3), GF(3,3), GG(3,3), GH(3,3), GI(3,3), GJ(3,3), GK(3,3), GL(3,3), GM(3,3), GN(3,3), GO(3,3), GP(3,3), GQ(3,3), GR(3,3), GS(3,3), GT(3,3), GU(3,3), GV(3,3), GW(3,3), GX(3,3), GY(3,3), GZ(3,3), HA(3,3), HB(3,3), HC(3,3), HD(3,3), HE(3,3), HF(3,3), HG(3,3), HH(3,3), HI(3,3), HJ(3,3), HK(3,3), HL(3,3), HM(3,3), HN(3,3), HO(3,3), HP(3,3), HQ(3,3), HR(3,3), HS(3,3), HT(3,3), HU(3,3), HV(3,3), HW(3,3), HX(3,3), HY(3,3), HZ(3,3), IA(3,3), IB(3,3), IC(3,3), ID(3,3), IE(3,3), IF(3,3), IG(3,3), IH(3,3), II(3,3), IJ(3,3), IK(3,3), IL(3,3), IM(3,3), IN(3,3), IO(3,3), IP(3,3), IQ(3,3), IR(3,3), IS(3,3), IT(3,3), IU(3,3), IV(3,3), IW(3,3), IX(3,3), IY(3,3), IZ(3,3), JA(3,3), JB(3,3), JC(3,3), JD(3,3), JE(3,3), JF(3,3), JG(3,3), JH(3,3), JI(3,3), JJ(3,3), JK(3,3), JL(3,3), JM(3,3), JN(3,3), JO(3,3), JP(3,3), JQ(3,3), JR(3,3), JS(3,3), JT(3,3), JU(3,3), JV(3,3), JW(3,3), JX(3,3), JY(3,3), JZ(3,3), KA(3,3), KB(3,3), KC(3,3), KD(3,3), KE(3,3), KF(3,3), KG(3,3), KH(3,3), KI(3,3), KJ(3,3), KK(3,3), KL(3,3), KM(3,3), KN(3,3), KO(3,3), KP(3,3), KQ(3,3), KR(3,3), KS(3,3), KT(3,3), KU(3,3), KV(3,3), KW(3,3), KX(3,3), KY(3,3), KZ(3,3), LA(3,3), LB(3,3), LC(3,3), LD(3,3), LE(3,3), LF(3,3), LG(3,3), LH(3,3), LI(3,3), LJ(3,3), LK(3,3), LL(3,3), LM(3,3), LN(3,3), LO(3,3), LP(3,3), LQ(3,3), LR(3,3), LS(3,3), LT(3,3), LU(3,3), LV(3,3), LW(3,3), LX(3,3), LY(3,3), LZ(3,3), MA(3,3), MB(3,3), MC(3,3), MD(3,3), ME(3,3), MF(3,3), MG(3,3), MH(3,3), MI(3,3), MJ(3,3), MK(3,3), ML(3,3), MM(3,3), MN(3,3), MO(3,3), MP(3,3), MQ(3,3), MR(3,3), MS(3,3), MT(3,3), MU(3,3), MV(3,3), MW(3,3), MX(3,3), MY(3,3), MZ(3,3), NA(3,3), NB(3,3), NC(3,3), ND(3,3), NE(3,3), NF(3,3), NG(3,3), NH(3,3), NI(3,3), NJ(3,3), NK(3,3), NL(3,3), NM(3,3), NO(3,3), NP(3,3), NQ(3,3), NR(3,3), NS(3,3), NT(3,3), NU(3,3), NV(3,3), NW(3,3), NX(3,3), NY(3,3), NZ(3,3), OA(3,3), OB(3,3), OC(3,3), OD(3,3), OE(3,3), OF(3,3), OG(3,3), OH(3,3), OI(3,3), OJ(3,3), OK(3,3), OL(3,3), OM(3,3), ON(3,3), OO(3,3), OP(3,3), OQ(3,3), OR(3,3), OS(3,3), OT(3,3), OU(3,3), OV(3,3), OW(3,3), OX(3,3), OY(3,3), OZ(3,3), PA(3,3), PB(3,3), PC(3,3), PD(3,3), PE(3,3), PF(3,3), PG(3,3), PH(3,3), PI(3,3), PJ(3,3), PK(3,3), PL(3,3), PM(3,3), PN(3,3), PO(3,3), PP(3,3), PQ(3,3), PR(3,3), PS(3,3), PT(3,3), PU(3,3), PV(3,3), PW(3,3), PX(3,3), PY(3,3), PZ(3,3), QA(3,3), QB(3,3), QC(3,3), QD(3,3), QE(3,3), QF(3,3), QG(3,3), QH(3,3), QI(3,3), QJ(3,3), QK(3,3), QL(3,3), QM(3,3), QN(3,3), QO(3,3), QP(3,3), QQ(3,3), QR(3,3), QS(3,3), QT(3,3), QU(3,3), QV(3,3), QW(3,3), QX(3,3), QY(3,3), QZ(3,3), RA(3,3), RB(3,3), RC(3,3), RD(3,3), RE(3,3), RF(3,3), RG(3,3), RH(3,3), RI(3,3), RJ(3,3), RK(3,3), RL(3,3), RM(3,3), RN(3,3), RO(3,3), RP(3,3), RQ(3,3), RR(3,3), RS(3,3), RT(3,3), RU(3,3), RV(3,3), RW(3,3), RX(3,3), RY(3,3), RZ(3,3), SA(3,3), SB(3,3), SC(3,3), SD(3,3), SE(3,3), SF(3,3), SG(3,3), SH(3,3), SI(3,3), SJ(3,3), SK(3,3), SL(3,3), SM(3,3), SN(3,3), SO(3,3), SP(3,3), SQ(3,3), SR(3,3), SS(3,3), ST(3,3), SU(3,3), SV(3,3), SW(3,3), SX(3,3), SY(3,3), SZ(3,3), TA(3,3), TB(3,3), TC(3,3), TD(3,3), TE(3,3), TF(3,3), TG(3,3), TH(3,3), TI(3,3), TJ(3,3), TK(3,3), TL(3,3), TM(3,3), TN(3,3), TO(3,3), TP(3,3), TQ(3,3), TR(3,3), TS(3,3), TT(3,3), TU(3,3), TV(3,3), TW(3,3), TX(3,3), TY(3,3), TZ(3,3), UA(3,3), UB(3,3), UC(3,3), UD(3,3), UE(3,3), UF(3,3), UG(3,3), UH(3,3), UI(3,3), UJ(3,3), UK(3,3), UL(3,3), UM(3,3), UN(3,3), UO(3,3), UP(3,3), UQ(3,3), UR(3,3), US(3,3), UT(3,3), UJ(3,3), UV(3,3), UW(3,3), UX(3,3), UY(3,3), UZ(3,3), VA(3,3), VB(3,3), VC(3,3), VD(3,3), VE(3,3), VF(3,3), VG(3,3), VH(3,3), VI(3,3), VJ(3,3), VK(3,3), VL(3,3), VM(3,3), VN(3,3), VO(3,3), VP(3,3), VQ(3,3), VR(3,3), VS(3,3), VT(3,3), VJ(3,3), VV(3,3), VW(3,3), VX(3,3), VY(3,3), VZ(3,3), WA(3,3), WB(3,3), WC(3,3), WD(3,3), WE(3,3), WF(3,3), WG(3,3), WH(3,3), WI(3,3), WJ(3,3), WK(3,3), WL(3,3), WM(3,3), WN(3,3), WO(3,3), WP(3,3), WQ(3,3), WR(3,3), WS(3,3), WT(3,3), WJ(3,3), WV(3,3), WW(3,3), WX(3,3), WY(3,3), WZ(3,3), XA(3,3), XB(3,3), XC(3,3), XD(3,3), XE(3,3), XF(3,3), XG(3,3), XH(3,3), XI(3,3), XJ(3,3), XK(3,3), XL(3,3), XM(3,3), XN(3,3), XO(3,3), XP(3,3), XQ(3,3), XR(3,3), XS(3,3), XT(3,3), XU(3,3), XV(3,3), XW(3,3), XX(3,3), XY(3,3), XZ(3,3), YA(3,3), YB(3,3), YC(3,3), YD(3,3), YE(3,3), YF(3,3), YG(3,3), YH(3,3), YI(3,3), YJ(3,3), YK(3,3), YL(3,3), YM(3,3), YN(3,3), YO(3,3), YP(3,3), YQ(3,3), YR(3,3), YS(3,3), YT(3,3), YJ(3,3), YV(3,3), YW(3,3), YX(3,3), YY(3,3), YZ(3,3), ZA(3,3), ZB(3,3), ZC(3,3), ZD(3,3), ZE(3,3), ZF(3,3), ZG(3,3), ZH(3,3), ZI(3,3), ZJ(3,3), ZK(3,3), ZL(3,3), ZM(3,3), ZN(3,3), ZO(3,3), ZP(3,3), ZQ(3,3), ZR(3,3), ZS(3,3), ZT(3,3), ZJ(3,3), ZV(3,3), ZW(3,3), ZX(3,3), ZY(3,3), ZZ(3,3)

```

5) S.M. Johnson, "Optimal Two and Three Stage Production Schedules with Set-up Times Included", *Nav. Res. Log. Quart.* 1, 1954, pp. 61 ~ 68.





```

WARNING * 3 *****
>>> VARIABLE NOT INITIALIZED IN PROGRAM
SYMBOL/LABEL = ASUM

NO ERRORS: F70 R04-00 MAINPROG .MAIN 05/04/82 11:53:46 TABLE SPACE: 4 K
STATEMENT BUFFER: 20 LINES/1321 BYTES STACK SPACE: 164 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
    
```

3·4 Campell, Dudek & Smith (CDS 技法)  
 注文順序를 아래의 式에 의해 K를 1부터 m-1  
 까지 變換시켜 m-1個의 2 공정 흐름작업순서로  
 FORTRAN-VIIO R04-00

變換한 후 最適處理順序를 각각 구하고 그 값이 가  
 장 적은 최대흐름시간을 갖는 處理順序를 擇한다.  
 上記한 技法의 Computer Program은 아래와 같  
 다.

```

FORTRAN VIIO: LICENSED RESTRICTED RIGHTS AS STATED IN LICENSE C-0404

1 00000001 DIMENSION A(20,3),B(20,2),YINDEX(20),ASUM(3)
2 00000002 INTEGER XINDEX(20),IX,IR,P,TINDEX(20),MINDEX(20)
3 00000003 DATA I, /6/
4 00000004 M=10
5 00000005 X=30.707
6 00000006 DO 50 I=1,M
7 00000007 DO 3 J=1,3
8 00000008 DO 3 J=1,3
9 00000009 IX=IX+1220703125
10 00000010 IF (IX+1,2,2)
11 00000011 IX=IX+2147483647 + 1
12 00000012 YFL=IX
13 00000013 YFL=YFL*.4656513E-9
14 00000014 IM=100*YFL
15 00000015 IX=IR
16 00000016 A(I,J)=XM
17 00000017 WRITE(=,1001) ((A(I,J),J=1,3),I=1,M)
18 00000018 ITER = 1
19 00000019 IF (ITER.NE. 1) GO TO 21
20 00000020 DO 5 I=1,M
21 00000021 B(I,1)=A(I,1)
22 00000022 B(I,2)=A(I,3)
23 00000023 YINDEX(I)=0.
24 00000024 5 CONTINUE
25 00000025 F=0
26 00000026 IL=M+1
27 00000027 7 XMIN=1000.
28 00000028 DO 11 I=1,M
29 00000029 IF (YINDEX(I).EQ. -100) GO TO 11
30 00000030 DO 10 J=1,2
31 00000031 IF (B(I,J).GE. XMIN) GO TO 10
32 00000032 XMIN=B(I,J)
33 00000033 IR0W=I
34 00000034 IC0L=J
35 00000035 10 CONTINUE
36 00000036 11 CONTINUE
37 00000037 IF (XMIN.EQ. 1000.) GO TO 50
38 00000038 IF (IC0L.EQ. 2) GO TO 15
39 00000039 F=F+1
40 00000040 XINDEX(F)=IR0W
41 00000041 GO TO 20
42 00000042 15 IL=IL-1
43 00000043 XINDEX(IL)=IR0W
44 00000044 20 YINDEX(IR0W)=-100.
45 00000045 GO TO 7
46 00000046 50 CALL ANSCM,XINDEX,A,XM2)
47 00000047 IF (ITER.NE. 1) GO TO 30
48 00000048 DO 51 I=1,M
49 00000049 TINDEX(I)=XINDEX(I)
50 00000050 YMIN=XM2
51 00000051 ITER=ITER+1
52 00000052 GO TO 4
53 00000053 21 DO 25 I=1,M
54 00000054 B(I,1)=A(I,1)+A(I,2)
55 00000055 B(I,2)=A(I,2)+A(I,3)
56 00000056 YINDEX(I)=0.
57 00000057 25 CONTINUE
58 00000058 F=0
59 00000059 IL=M+1
60 00000060 DO 30 I=1,M
61 00000061 IF (XINDEX(I).GE. YMIN) GO TO 311
62 00000062 DO 33 J=1,M
63 00000063 YINDEX(I)=XINDEX(I)
64 00000064 DO 35 J=1,M
65 00000065 XINDEX(I)=TINDEX(I)
66 00000066 WRITE(=,1020) (XINDEX(I),I=1,M),YMIN
67 00000067 ITER=ITER+1
68 00000068 GO TO 4
69 00000069 30 311
70 00000070 DO 30 I=1,M
71 00000071 IF (I.EQ. 1) GO TO 75
72 00000072 DO 30 J=1,M
73 00000073 IF (J.EQ. 1) GO TO 75
74 00000074 IF (J.EQ. I) GO TO 77
75 00000075 XINDEX(J)=XINDEX(J)
    
```

```

76      00090000      75      30      GO      TO      30
77      00090000      75      30      WINDEX(J)=XINDEX(I)
78      00090000      75      30      GO      TO      30
79      00090000      77      30      WINDEX(J)=XINDEX(IM1)
80      00090000      30      CONTINUE
81      00090000      30      CALL      ANS(CM,WINDEX,1,AM2)
82      00090000      30      IF(CAM2 .GE. YMIN) GO TO 30
83      00090000      30      GO      35      K=1,M
84      00090000      35      TINDEX(K)=WINDEX(K)
85      00090000      35      YMIN=XM2
86      00090000      90      CONTINUE
87      00090000      90      WRITE(TM,1020) (TINDEX(I),I=1,M),YMIN
88      00090000      90      ITER=ITER+1
89      00090000      90      GO      TO      50
90      00090000      300      301      K=1,M
91      00090000      301      XINDEX(I)=TINDEX(I)
92      00090000      723      ICOUNT=0
93      00090000      723      J=190      I=2,M
94      00090000      723      IM1=I-1
95      00090000      723      DO      130      J=1,M
96      00090000      723      IF(CJ .EQ. IM1) GO TO 175
97      00090000      723      IF(CJ .EQ. I) GO TO 177
98      00090000      723      WINDEX(J)=XINDEX(J)
99      00090000      723      GO      TO      130
100     00090000      175     WINDEX(J)=XINDEX(I)
101     00090000      175     GO      TO      30
102     00090000      177     WINDEX(J)=XINDEX(IM1)
103     00090000      180     CONTINUE
104     00090000      180     CALL      ANS(CM,WINDEX,1,AM2)
105     00090000      180     IF(CAM2 .GE. YMIN) GO TO 190
106     00090000      180     GO      185      K=1,M
107     00090000      185     TINDEX(K)=WINDEX(K)
108     00090000      185     YMIN=XM2
109     00090000      185     ICOUNT=ICOUNT+1
110     00090000      190     CONTINUE
111     00090000      190     IF(ICOUNT .EQ. 0) GO TO 100
112     00090000      190     GO      195      K=1,M
113     00090000      195     XINDEX(K)=TINDEX(K)
114     00090000      195     TINDEX(K)=XINDEX(K)
115     00090000      195     WRITE(TM,1020) (TINDEX(I),I=1,M),YMIN
116     00090000      195     ITER=ITER+1
117     00090000      195     GO      TO      50
118     00090000      195     PRINT(30X,F15.1,5X,F15.1,5X,F15.1)
119     00090000      195     PRINT(30X,1317,' ELAPSED TIME=',F15.1)
120     00090000      195     STOP
121     00090000      195     END
*****
**RN: 100 * 9 *****
>>> VARIABLE NOT INITIALIZED IN PROGRAM
SYMBOL/LABEL = NSUM

```

```

NO ERRORS: F70 R04-00 MAINPRG .MAIN 05/04/82 11:33:33 TABLE SPACE:
STATEMENT BUFFER: 20 LINES/1321 BYTES STACK SPACE: 164 WORDS
SINGLE PRECISION FLOATING-PT SUPPORT REQUIRED FOR EXECUTION * KB
05/04/82 11:33:37 PAGE 3
***, SEE DOCUMENTATION PACKAGE, 04-101499.

```

4. 結果의 比較 및 分析

4.1 發見的 技法들의 効率性 比較結果

前章에서 論한 여러 技法들과 効率性을 比較하기 위한 제안 100個의 問題에 各 企業들을 적용해서 Computer를 使用하여 解를 求한 후 이들의 効率性을 比較分析한 結果는 아래와 같다.

발견적 技法들의 효율성 比較결과

	R (%)	C	P (%)	W
CDSSES	0.135	0.235	88.8	1.17
RAES	0.223	0.390	79.6	1.367

CDSCS	1.107	8.787	68.4	1.776
RACS	1.413	6.548	55.1	2.214
SOES	1.413	7.826	51.0	2.092
CDS	2.051	11.50	46.9	2.786
SOCS	2.435	15.524	30.6	2.582
RA	2.791	19.046	32.7	3.327
SO	4.541	39.060	9.2	3.816

4.2 技法들의 우월성 比較결과

上記한 圖表에서 알 수 있듯이 各 評價基準에 對한 各 技法들의 相對的인 우월성은 아래와 같다.

- R : CDSSES > RAES > CDSCS > RACS > SOES > CDS > SOCS > RA > SO
- C : CDSSES > RAES > RACS > SOES > CDSCS > CDS > SOCS > RA > SO
- P : CDSSES > RAES > CDSCS > RACS > SOES > CDS > RA > SOCS > SO
- W : CDSSES > RAES > CDSCS > SOES > RACS > SOCS > CDS > RA > SO

또 Close Order Search나 Extensive Search를 적용할 경우 初期解가 좋은 경우에는 보다 낫은 해를 얻게 된다.

- CDS > RA > SO
- CDS > RACS > SOCS
- CDS > RAES > SOES

5. 結 論

前章에서 論한  $3 \times n$  흐름작업장의 注文處理順序를 決定할 수 있는 改善된 Heuristic 技法과 기존 Heuristic 技法의 效率性을 比較·分析한 結果 아래와 같은 要約을 얻을 수 있었다.

① 최대흐름시간을 관점으로 할 경우 RA보다는 CDS가, SO보다는 RA가 效率性이 높음을 알 수 있다.

② RAES는 注文順序가  $A \geq B \geq C, A \geq C \geq B, C \geq A \geq B$ 인 경우 效率性이 매우 높다.

③ 특히 RA 경우 Computer의 도움없이 쉽게 處理順序를 求할 수 있는 경우의 注文順序가  $A \geq B \geq C, A \geq C \geq B, C \geq A \geq B$ 임을 알 수 있다.

本論文의 경우  $3 \times n$ 의 경우에만 研究·分析되었으나 이 概念을  $m \times n$  흐름작업장 경우 및 다른 일반 業務에 유사한 경우에는 이들 改善된 技法으로 그 效率性을 높일 수 있다고 사료된다.

參 考 文 獻

- 1.) Ashour, Said, "An Experimental Investigation Comparative Evaluation of Flow-shop Sequencing Techniques", *Operation Research*, Vol. pp.541 ~ 549.
- 2.) Burns, F., and J.Rooker, "Extensions and Comments Regarding Special Cases of the Three Machine Flow-shop Problem", *Nav. Res. Log. Quart.* 1975.
- 3.) Campell, H.G., Dudek, R.A., and M.L.Smith, "A Heuristic Algorithm for the n Job, m Machine Sequencing Problem", *Management Science*. Vol. 16, pp.214 ~ 221.
- 4.) Conway, R.W., Maxwell, W.L., and L.W. Miller, "Theory of Scheduling", Addison-Wesley, Reading, Massachusetts, 1967.
- 5.) Dannenbring, D.G., "An Evaluation of Flow-shop Sequencing Heuristics", *Management Science*, July 1977, Vol.23, No.11, pp.1174 ~ 1182.
- 6.) Ingall, E.J., and L.E.Schrage, "Application of the Branch and Bound Technique to some Flow-shop Scheduling Problems", *Operation Research*, 1965.
- 7.) Johnson, S.M., "Optimal Two and Three Stage Production Schedules with Set-up Times Included", *Nav. Res. Log. Quart.* 1954.1, pp.61 ~ 68.
- 8.) Martin, K.Starr, "Systems Management of Operations", Prentice Hall, Inc. 1971, pp.295 ~ 324.
- 9.) Page, E.S., "An Approach to Scheduling Jobs on Machines", *Journal of the Royal Statistical Society* Vol.23, pp.434 ~ 492.
- 10.) Pamer, D.S., "Sequencing Jobs through a Multi-stage Process in the Minimum Total Time - A Quick Method of Obtaining a Near Optimum", *Operational Research Quarterly*, Vol.16, pp.101 ~ 107.