Scheduler

James Marshallsea

We are very mindful of the number of commercial users of the CPC464 system. A good number have chosen the CPC464 as it can span home and business applications more readily than most of the alternatives. Many such users have the machine at home to develop their programming skills and write their own applications, so here's a very interesting application from one of our first competition entrants that puts work study efficiency theory into practice.

The specific application may not be exactly tuned to your requirements, but you may well be able to tailor the format to suit a wide variety of planning applications in commercial (and private) situations.

This program is for use in workstudy/production control. Being implemented on a micro, it should be applicable to the practically minded manager who would normally be intimidated by trying to get to grips with a large company computer terminal.

The program will provide an efficient method for determining a sequence for processing a set of jobs, or maybe a customer, either of which will place demands on organisational resources. Under certain conditions the problem will be solved, or at least an optimum solution will be found.

Sequencing

Sequencing is the order in which jobs are placed for processing. Sequencing jobs involves the ordering of jobs through one or more processes, so that specific performances, (optimum performances), such as minimal idle time, total machine time, and time delivery dates are reached, variations of which can produce significant results in costs and productivity.

The sequential ordering of the start and finish of jobs is called a schedule. This schedule is only formed when the processing sequence has been established, noting that a job does not start until the previous job has finished. The machines provide the process by which the jobs are completed, and machines need not be mechanical, they could be human, (visual quality check, and so on). It is also important to note that all jobs must go through the same process.

The process time is the amount of time that a machine will need to complete that process, the times for these processes are then collected. These times are then manipulated to obtain the configuration requiring the least amount of processing time: maximum efficiency. Assuming that the least amount of processing time is the result required!

Total facility processing time is the total machine time plus the machine idle time. Therefore reducing the total facility processing time also reduces the machine idle time.

The method used to solve this problem was developed by S.A. Johnson and later extended by W.A. Stark, (number of jobs - 1 method). These two algorithms are used in the program to obtain the results.

The Johnson algorithm finds the shortest processing time for two machines, the Stark algorithm is then used to get the shortest processing time from the total number of jobs. By using the Johnson algorithm on the first two times and the last two times, another sequence is obtained. Stark's algorithm is repeatedly applied to the Johnson algorithm, (1 to number jobs-1).

This procedure will develop several sequences, allowing some discretion as to the best sequence for one's purposes, considering that holidays and maintenance must be accounted for.

The program itself

The program will ask for the number of jobs you have to process, say 4 cars for a service at a garage - or four items for assembly into a final product.

Then the program asks for the number of operations: the example garage operations are:
service, oil change, wash/clean, polish [Really? You must let me have the name of your garage: ed]

The assembly operations are:

Assemble and cable into roverpoint, seal roverpoint, attach plug and test. Next enter a brief description of the operations, service, oil change, wash/clean and polish. Now enter the job name: for the garage it might be a CORTINA.

Then enter the times the cortina has to spend in each operation.

The computer then asks for the units of time you are using.

The computer now presents a summary of the data input, together with the finish times. Another prompt invites you to proceed to check times for the initial sequence.

Next comes the schedule itself, and you can list all the options, or select only the best ones.

Having decided how you want to view the results, the program goes through its paces to list the best results for you to examine, and then finally to determine the recommended schedule, sequence and the time involved.

Please note the following listing has been compressed to 40 columns. DO NOT terminate line ends with [ENTER] unless the next line on this listing starts with a valid line number! (Follow the sequence incrementing in 10's)

```
100 MODE 2:INK 1,0:INK 0,13:BORDER 13
110 DIM m(20,20),j$(20),m$(20),a(20)
120 DIM b(20),c(20),seq(20,20),PASS(20)
130 DIM mt(20),t1(20,20),t2(20,20)
140 DIM it(20),kl(20)
150 GOTO 410
160 FOR I=1 TO 500:NEXT I
170 PRINT,,,,"PRESS ANY KEY FOR MORE"
180 IF INKEY$="" THEN 180
190 RETURN
200 REM *** CALCULATE IN/OUT SCHEDULE
210 t1(1,1)=0
220 FOR I=1 TO nm
230 t2(1,I)=t1(1,I)+m(c(1),I)
240 IF I= nm THEN 260
250 t1(1,I+1)=T2(1,I)
260 NEXT I
270 TI=T2(1,1)
280 FOR J=2 TO NJ
290 T1(J,1)=T2(J-1,1)
300 T2(J,1)=T1(J,1)+M(C(J),1)
310 NEXT J
320 REM
330 FOR J=2 TO NJ
340 FOR I=2 TO NM
350 T1(J,I)=T2(J,I-1)
360 IF T2(J-1,I)>T2(J,I-1) THEN
    T1(J,I)=T2(J-1,I)
370 T2(J,I)=T1(J,I)+M(C(J),I)
380 NEXT I
390 NEXT J
400 RETURN
410 CLS:PRINT:PRINT
420 SPEED INK 20,20
430 INK 2,0,13
440 MODE 1:INK 1,0:INK 0,13:BORDER
    13:PAPER 0:PEN 1
450 PRINT "M-MACHINE SCHEDULING"
460 PRINT:PEN 2
470 PRINT TAB(6)"by J.P.MARSHALLSEA":PEN
    1:PRINT
480 PRINT:PRINT"THE PROGRAM CAN BE USED
FOR UP TO 20"
490 PRINT "JOBS AND 20 MACHINE
    OPERATIONS.":PRINT
500 PEN 3:PRINT,"NOTE"
510 PRINT STRING$(40,45)
520 PRINT ,"COMPLETE ALL DATA ENTRIES
    BY"
530 PRINT ,"HITTING THE RETURN KEY"
540 PRINT STRING$(40,45)
550 PEN 1
560 GOSUB 160
570 MODE 2:INK 1,0:INK 0,13:BORDER 13
    :CLS
580 DISPLAY=0
590 INPUT "Type in the number of jobs
    ";NJ
600 PRINT
610 INPUT "Type in the number of machine
    operations";NM
620 CLS
630 PRINT "Begin by describing the
    machine operations."
640 PRINT "These operations are assumed
    to be sequential."
650 PRINT
660 FOR I=1 TO nm
670 PRINT:PRINT
680 PRINT "Type a description of machine
    operation";I
690 PRINT "(use up to 7 characters)."
700 INPUT m$(I)
710 IF LEN(m$(I))<=7 THEN 750
720 PRINT
730 PRINT"Use only 7 characters please"
740 PRINT:GOTO 680
750 NEXT I
760 CLS
770 PRINT "Now briefly,(6 characters or
    less) describe each job,and the"
780 PRINT "time in each machine
    operation for that job."
790 PRINT
800 FOR J=1 TO NJ
810 PRINT "Type a description of job ";j
820 INPUT j$(j)
830 PRINT
840 PRINT "For this job,enter the time
    (to the nearest tenth unit) required"
850 PRINT "for each of the machine
    operations listed below."
860 PRINT
870 PRINT "(Use the same time units for
    all operations.)"
880 PRINT
890 FOR I=1 TO nm
900 PRINT "Time in operation ";m$(I);" =
    ";
910 INPUT M(J,I)
920 PRINT
930 NEXT I
940 CLS
950 NEXT j
960 REM
970 PRINT "Input complete"
980 PRINT
990 PRINT "Type in the units of time,
    (E.G.,Minutes,Hours,ETC.)"
1000 INPUT UNIT$
1010 ns=1
1020 nf=ns+3
1030 IF nf>nm THEN nf=nm
1040 CLS
1050 IF ns=1 GOTO 1080
1060 PRINT, ,"MACHINE OPERATIONS (CONT."
1070 GOTO 1090
1080 PRINT ,,"MACHINE OPERATIONS"
1090 PRINT ,,STRING$(18,45);
1100 PRINT,"(TIMES IN ";UNIT$;")"
1110 PRINT ,"JOB";;;
1120 FOR I=ns TO nf
1130 PRINT m$(I);,;
1140 NEXT I
1150 PRINT STRING$(80,45)
1160 FOR j=1 TO nj
1170 PRINT J;".";;;j$(j);,;
1180 FOR I=ns TO nf
1190 PRINT M(J,I); ,;
1200 NEXT I
1210 PRINT
1220 NEXT j
1230 PRINT:PRINT"Do you wish to change
    any of the data"
1240 a$=INKEY$:IF a$="" THEN 1240
1250 IF a$="Y" THEN GOSUB 2730 :GOTO
    1040
1260 IF a$<>"N" THEN 1240
1270 PRINT "Do you wish to re-arrange
    your schedule."
```

```
1280 A$=INKEY$:IF A$="" THEN 1280
1290 IF A$="Y" THEN GOSUB 3820 :GOTO
1010
1300 IF A$<>"N" THEN 1280
1310 IF NF=NM THEN 1330
1320 NS=NF+1:GOTO 1020
1330 FOR j=1 TO nj:c(j)=j:NEXT j
1340 GOSUB 2990
1350 GOSUB 200
1360 CLS
1370 PRINT: PRINT"INITIAL SEQUENCE"
1380 PRINT:PRINT"ORDER          JOB NAME
     F/TIME"
1390 PRINT
1400 FOR j=1 TO nj
1410 PRINT j;,;j$(j);,;t2(j,nm)
1420 NEXT j
1430 GOSUB 160
1440 kk=0
1450 GOSUB 3090
1460 LET bt=nm*t2(nj,nm)
1470 display=0
1480 flag=1
1490 GOSUB 3390
1500 GOSUB 160
1510 IF nm=2 THEN GOTO 1600
1520 PRINT "To veiw all loading
     sequences as they are determined,"
1530 PRINT "type an <A>,to view only the
best schedule(s),"
1540 PRINT "type a <B>."
1550 PRINT "You may return later to
     review all schedules."
1560 PRINT:PRINT"A or B"
1570 A$=INKEY$:IF A$="" THEN 1570
1580 IF A$="B" THEN display=1:GOTO 1600
1590 IF A$<>"A" THEN GOTO 1570
1600 FOR KK=1 TO nm-1
1610 rpt=0
1620 GOSUB 1950
1630 GOSUB 2070
1640  IF rpt<> 1 THEN GOSUB 2560
1650 GOSUB 200
1660 IF flag=0 THEN 1700
1670 IF NM <>2 THEN 1680
1680 GOSUB 2420
1690 GOSUB 3090
1700 GOSUB 3390
1710 IF flag=0 THEN 1740
1720 GOSUB 160
1730 IF rpt=1 THEN GOTO 1870
1740 NEXT kk
1750 IF nm=2 THEN 1870
1760 GOSUB 3640
1770 FOR LL=0 TO lm
1780 kk=kl(ll)
1790 display=0
1800 CLS
1810 PRINT "A good job sequence is:"
1820 PRINT
1830 PRINT "ORDER          JOB NAME
     F/TIME"
1840 IF nm=2 THEN RETURN
1850 LET rpt=1:GOTO 1620
1860 NEXT ll
1870 GOSUB 2840
1880 ON FLAG GOTO 1910,1930
1890 PRINT:PRINT "THEN GOODBYE!":PRINT
1900 GOTO 1940
1910 flag=0:CLS
1920 GOTO 1010
1930 GOTO 580
1940 END
1950 REM **** COMBINE DATA FOR J&J ****
1960 FOR j=1 TO nj
1970 a(j)=0:b(j)=0
1980 NEXT j
1990 FOR I=1 TO kk
2000 FOR j=1 TO nj
2010 a(J)=a(J)+M(J,I)
2020 b(j)=b(J)+m(j,nm-I+1)
2030 NEXT j
2040 NEXT I
2050 RETURN
2060 REM **** J&J ALGORITHM ****
2070 nf=0:nl=NJ:l=0
2080 GOSUB 2360
2090 FOR j=1 TO nj
2100 c(j)=nj+1
2110 NEXT j
2120 sm=tt
2130 FOR j=1 TO nj
2140 fl=0
2150 REM **** ELIMINATE DONE JOBS ****
2160 FOR jJ=1 TO nj
2170 IF j=c(jJ) THEN fl=1
2180 NEXT jJ
2190 IF fl=1 THEN GOTO 2220
2200 IF b(j)<= sm THEN sm=b(j):il=j:l=2
2210 IF a(j)<= sm THEN sm=a(j):il=j:l=0
2220 NEXT j
2230 IF l=2 THEN GOTO 2260
2240 c(nf)=il
2250 nf=nf+1:GOTO 2280
2260 c(nl)=il
2270 nl=nl-1
2280 IF NL >= NF THEN 2120
2290 REM ***** GENERATE AN ARRAY FOR
2300 REM ***** FILTER.
2310 FOR J=1 TO NJ
2320 SEQ(KK,J)=C(J)
2330 NEXT J
2340 RETURN
2350 REM *** FIND MAXIMUM ELEMENT ***
2360 tt=A(1)
2370 FOR J=1 TO NJ
2380 tt=MAX(a(j),tt)
2390 tt=MAX(b(j),tt)
2400 NEXT J
2410 RETURN
2420 REM *** PRINT JOB SEQUENCE
2430 IF RPT=1 THEN GOTO 2500
2440 CLS
2450 IF NM<>2 THEN GOTO 2470
2460 PRINT "AN OPTIMAL SEQUENCE
IS:":GOTO 2480
2470 PRINT "A POSSIBLE SEQUENCE IS:"
2480 PRINT
2490 PRINT"ORDER            NAME
     F/TIME"
2500 FOR j=1 TO nj
2510 PRINT j;,;j$(c(J));,;t2(j,nm)
2520 NEXT j
2530 GOSUB 160
2540 RETURN
2550 REM **** FILTER SUBROUTINE ****
2560 flag=1
2570 IF nm=2 THEN RETURN
2580 IF kk<2 THEN RETURN
2590 FOR k=1 TO kk-1
2600 IF nflag=0 THEN GOTO 2670
2610 nf=0
2620 FOR j=1 TO nj
2630 IF nf=1 THEN GOTO 2650
2640 IF c(J)<> seq(k,j) THEN nf=1
2650 NEXT j
```

```
2660 IF nf=0 THEN LET flag=0
2670 NEXT k
2680 REM *** FLAG=1 IF SEQUENCE DIFFERS
2690 REM *** FROM PREVIOUS SEQUENCE
2700 REM
2710 RETURN
2720 REM **** CORRECTIONS ****
2730 PRINT:INPUT "Enter the job number
     ";j
2740 IF j<1 OR j>nj THEN PRINT
     "ERROR-RE-ENTER PLEASE":GOTO 2730
2750 PRINT:PRINT"Now enter the new job
     description"
2760 INPUT w$
2770 IF LEN (w$) >7 THEN w$=LEFT$(w$,7)
2780 j$(j)=w$
2790 FOR I=1 TO nm
2800 PRINT "Enter the new times for
     operation ";m$(i);
2810 INPUT m(j,i)
2820 NEXT i
2830 RETURN
2840 LET FLAG=0
2850 PRINT
2860 PRINT"Do you wish to schedule more
     ?"
2870 PRINT:PRINT" Y or N"
2880 INPUT AN$
2890 IF an$="N" THEN RETURN
2900 IF an$<>"Y" THEN GOTO 2880
2910 PRINT:PRINT"Do you wish to modify
     existing data ?"
2920 PRINT" Y or N"
2930 INPUT an$
2940 IF an$="Y" THEN LET flag=1:RETURN
2950 IF an$<>"N" THEN GOTO 2930
2960 PRINT"Then a new problem is
     asssumed"
2970 flag=2
2980 RETURN
2990 REM *** CALCULATE MACHINE TIME
3000 FOR I=1 TO nm+1 :mt(I)=0:NEXT I
3010 FOR I=1 TO nm
3020 FOR j=1 TO nj
3030 mt(I)=mt(I)+m(j,I)
3040 NEXT j
3050 MT(NM+1)=MT(NM+1)+MT(I)
3060 NEXT I
3070 RETURN
3080 REM *** PRINT SCHEDULE ***
3090 IF display=1 THEN RETURN
3100 ns=1
3110 PRINT "JOB";;
3120 IF ns=1 THEN GOTO 3150
3130 PRINT" SCHEDULING (CONT.)"
3140 GOTO 3160
3150 PRINT "SCHEDULE"
3160 PRINT STRING$(79,45)
3170 nf=ns+3
3180 IF nf>nm THEN nf=nm
3190 FOR I=ns TO nf
3200 PRINT ,;"   ";m$(I);"   ";
3210 NEXT I
3220 PRINT
3230 FOR I=ns TO nf
3240 PRINT ,;"IN       OUT ";
3250 NEXT I
3260 PRINT
3270 FOR J=1 TO NJ
3280 PRINT C(J);".";" ";
3290 FOR I=NS TO NF
3300 PRINT ,INT(T1(J,I));"
     ";INT(T2(J,I));
3310 NEXT I
3320 PRINT
3330 NEXT J
3340 IF NF=NM THEN GOTO 3360
3350 NS=NF+1:GOTO 3110
3360 GOSUB 160
3370 RETURN
3380 REM ***** IDLE TIME *****
3390 OT=NM*T2(NJ,NM)
3400 IF ot>bt THEN bt=ot
3410 IT(KK)=INT(OT)-INT(MT(NM+1))
3420 IF flag=0 THEN RETURN
3430 PRINT
3440 PRINT"   PERFORMANCE CHARACTERISTICS
     FOR";
3450 IF kk<>0 THEN GOTO 3480
3460 PRINT " INITIAL SEQUENCE"
3470 GOTO 3520
3480 IF nm<>2 THEN GOTO 3510
3490 PRINT" OPTIMAL SEQUENCE"
3500 GOTO 3520
3510 PRINT " THIS SEQUENCE"
3520 PRINT STRING$(79,45)
3530 PRINT"TOTAL FACILITY PROCESSING
     TIME ";
3540 PRINT  "= ";ROUND(OT,2);",in units
     of ";UNIT$
3550 PRINT
3560 PRINT "TOTAL MACHINE PROCESSING
     TIME ";
3570 PRINT "= ";ROUND(MT(NM+1),2);",in
     units of ";UNIT$
3580 PRINT
3590 PRINT "IDLE MACHINE TIME ";
3600 PRINT "= ";ROUND(IT(KK),2);",in
     units of ";UNIT$
3610 PRINT
3620 RETURN
3630 REM *** SHORTEST SEQUENCE ***
3640 FOR I=0 TO nm-1
3650 BT=MIN(IT(I),BT)
3660 NEXT I
3670 is=-1
3680 kk=0
3690 IF it(kk)=bt THEN is=kk
3700 IF kk>=(nm-1) THEN GOTO 3720
3710 kk=kk+1:GOTO 3690
3720 kl(0)=is
3730 l=1
3740 FOR I= 0 TO is-1
3750 IF it(I)<>bt THEN GOTO 3780
3760 kl(l)=n
3770 l=l+1
3780 NEXT I
3790 lm=l-1
3800 RETURN
3810 REM ******* ALTER SCHEDULES
     *********
3820 PRINT:PRINT "Enter the machine
     number you wish to move."
3830 PRINT "Then enter the new location
     in your schedule (X,Y)"
3840 INPUT fs,ms
3850 PAS$=M$(FS):M$(FS)=M$(MS)
     :M$(MS)=PAS$
3860 FOR I=1 TO nJ
3870 PASS(i)=m(I,fs)
3880 m(I,fs)=m(I,ms)
3890 m(I,ms)=PASS(i)
3900 NEXT i
3910 RETURN
```

■ CPC464 USER