NASA CR-172094

AN INVESTIGATION OF TRANSITIONAL MANAGEMENT PROBLEMS FOR THE NSTS AT NASA

CONTRACT 9-BC4-19-6-1P

QUARTERLY REPORT (APR. 15, 1988 - JUL. 15, 1988)

BY

PRINCIPAL INVESTIGATOR : JOHN HUNSUCKER

(NASA-CR-172094) AN INVESTIGATION OF TRANSITIONAL MANAGEMENT PROBLEMS FOR THE NSTS AT NASA Quarterly Report, 15 Apr. - 15 Jul. 1988 (Houston Univ.) 78 p CSCL 05A

N89-15770

Unclas

G3/81 0179696

DEPARTMENT OF INDUSTRIAL ENGINEERING

UNIVERSITY OF HOUSTON - UNIVERSITY PARK

JULY 15, 1988

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

I. INTRODUCTION

The intent of this report is to satisfy the contractual obligation of a quarterly report and to provide NASA with an interim overview of the results of the University of Houston team to date. Another objective of this report is to provide a resting place or summary document, if you will, for the ideas and concepts developed with the collaboration and support of the Management Integration Offices of NASA. In addition it is hoped that this report will help to stimulate the healthy problem solving process already present at NASA.

This report is the second quarterly report in the fourth year of the research contract. The main thrust of the work is to assist NSTS in finding ways and means of moving into a truly operational era in the sense of routine timely production of flights. This work is a continuation of the effort of the first three years. The reader who seeks a full understanding of the concepts presented is encouraged to read the final reports of the last three years.

1.0 STRATEGY AND FORMAT

The overall strategy of this effort is to 1) search the literature for applications of transition management and other related issues, 2) conduct investigations into the experiences of the industries with the transition management, and 3) to adapt the information found in 1) and 2) above into a form useful to NASA while at the same time applying industrial engineering and engineering management expertise to problems and issues as they emerge.

The strategy discussed above provides the format for the remaining parts of the report with the industrial adaptation being covered in Chapter II, a discussion of the branch and bound algorithm for a flow shop with multiple processors being discussed in Chapter III, and the contractual effort being presented in Chapter IV.

CHAPTER II

INDUSTRIAL ADAPTATION

1.0 INTRODUCTION

2.0 IMMEDIATE CONCERNS

3.0 SOUTH TEXAS NUCLEAR PROJECT INTERVIEW

4.0 VERIFICATION AND EXPANSION OF KNOWLEDGE THROUGH THE SUBMISSION OF PAPERS AND PRESENTATIONS

APPENDICES:

II A. MEETING ANALYSIS FOR 1987 DEPUTY DIRECTOR NSTS

II B. FIELD NOTES OF INTERVIEW WITH HL/P SOUTH TEXAS NUCLEAR PROJECT (STNP)

II C. PUBLICATION/PRESENTATION OF RESEARCH

II. INDUSTRIAL ADAPTATION

1.0 INTRODUCTION

This quarter's work in the adaptation area has been divided into three sections. The first section deals with issues we felt to be of immediate concern. The second section deals with an interview with managers of the South Texas Nuclear Project in the area of R&D to operations transition. The last section discusses verification and expansion of transition management knowledge through presentations and publications.

2.0 IMMEDIATE CONCERNS:

A significant amount of effort has been devoted this quarter doing an agenda analysis of the Deputy Director of the program office. The intent of this analysis is two-fold: to determine how loaded the Deputy Director is as well as how his time is spent and to compare his work effort with a similar analysis done three years ago on Glen Lunney when he was the head of the shuttle program. The second third of this effort is presented in Appendix II A as a meeting analysis of the Deputy Director during 1987. The tentative conclusions reached in this report are that dealing with HQ takes a significant amount of time and this results in long meetings. Another is that very little future planning is being done. The Deputy Director also spends a large amount of time dealing with technical matters. While this has perhaps been caused by the reflight issues, it does seem large for a top level manager.

3.0 SOUTH TEXAS NUCLEAR PROJECT INTERVIEW

There are many similarities between the shuttle program and the building of a nuclear power plant. The plants are highly complex, costly, in the public eye, and represent fairly new technology. There are also some major differences. One is that the NRC applies very stiff controls on the plants and this predicates much of the safety / documentation / production system. Another difference is that there are more than three power plants in existence, unlike the shuttle, and there is a large collective data base that is used to support design and operation.

There are numerous specific comments in the field notes presented in Appendix II B and they are worth reading. One of the major points in this interview was that a very complex documentation and document control system is required for the plant to go operational. This system included design morphology of the construction and design of the plant. When one considers that this plant is going to be handed over from the design company to the operational company, the reasons for the completeness and complexity of this system become evident.

Another major point is that they use extensive top-down communication. This has helped them to build, what they think, is a strong team to bring the plant on line. Also, as an aside, if NASA decides to cross train any of this staff in production techniques, the nuclear industry would be a good candidate for the temporary assignment of staff.

4.0 VERIFICATION AND EXPANSION OF KNOWLEDGE THROUGH THE SUBMISSION OF PAPERS AND PRESENTATIONS

Part of the process of acquiring and verifying knowledge involves sharing ideas and concepts with fellow researchers and practitioners. There are numerous highly qualified researchers in academe and industry, and the intellectual input of such colleagues is very important for the growth and development of the research activity. Therefore, it is very important that the researchers exchange their work in order to simplify and substantiate their research efforts.

Conferences are one of the principal meeting places for the exchange of ideas and thoughts by researchers. So far this year, one paper has been accepted for presentation at the national level in order to publicize the research work done on this grant and gain valuable response from different areas of the academic and professional communities. Another channel of verification of theoretical and practical concepts is by means of publication in reputable journals. This mode of presentation usually covers a wider segment of researchers and professionals involved in similar activities. Moreover, most prestigious journals have an elaborate process whereby the submitted paper is scrutinized by several prominent people (known as referees) before it is cleared for publication. Such extensive exploration by the referees improves the quality of the paper, and usually provides good direction for future research. Currently, one paper is undergoing the review process, and two more are being prepared for submission to refereed journals in the area of engineering management.

A summary of the presentations and publications of the research is contained in Appendix II C.

APPENDIX II A

MEETING ANALYSIS FOR 1987

DEPUTY DIRECTOR NSTS

MEETING ANALYSIS FOR 1987
DEPUTY DIRECTOR
NSTS PROGRAM
JLH 8 JULY 88

INTRODUCTION: The following charts and information were taken from the 1987 agenda of the Deputy Director of the NSTS Program Office. Each meeting was categorized in four ways: the level of the meeting, the temporal time frame of the meeting, the location of the prime attendant of the meeting other than the Deputy Director, and the subject. The following gives the classifications that were used for each category:

LEVEL	TEMPORAL	LOCATION	SUBJECT
DOWN	NOW	JSC	MANAGEMENT (M)
UP	PAST	NASA OTHER (NO)	TECHNICAL (T)
ACROSS	FUTURE	OTHER (O)	BUDGET (B)
		DOD	PERSONAL (P)
		HQ	

Level refers to whether the meeting dealt with an individual of approximately equal, less, or greater status. The temporal category refers to whether the subject of the meeting was current, from the past, or a future issue. To be classified as future, roughly, a two year time frame was used. The location category is self explanatory. In the subject category, a meeting was a technical meeting if it required technical or engineering expertise on the Deputy Director's part. The personal classification refers to items such as handing out awards, meeting individuals, giving interviews to the press and others. It did not include any personal time such as doctor's appointments or leave.

The results of this analysis are contained in the eight charts and the first of the three tables in the back of this report. The last table contains classifications of various meetings that occurred frequently.

RESULTS: There were 1073 meetings taking a total of 1525 hours for an average of 1.42 hours per meeting.
Level: The majority of the meetings were down both by time and number. While the up and across categories were essentially tied by number, the up meeting took more time and in fact had a greater average time per meeting (3.31 hours per meeting) than any other classification in any category with the exception of the HQ classification for the location category.
Temporal: Almost all meetings were classified as now with virtually none as past and only a few as future. Most of the future meeting were related to the budget.
Location: Most of the meetings were classified as JSC with NO, O, HQ, and DOD following in that order. However, by time, NO led followed by JSC, HQ, O, and DOD in that order.

Note that the HQ classification for this category had the longest average time of all classification of all categories (4.52 hours per meeting).

Subject: The order for both number and time was technical, management, budget, and personal. Roughly half of the time and half of the number of meeting was spent on technical subjects.

The top five meetings:

BY NUMBER					BY TIME				
NUMBER	CLASSIFICATION				TIME	CLASSIFICATION			
246	D	N	JSC	M	449.50	D	N	NO	T
231	D	N	NO	T	200.00	U	N	HQ	M
194	D	N	JSC	T	167.50	D	N	JSC	M
48	A	N	JSC	M	163.75	D	N	JSC	T
44	U	N	JSC	M	125.25	U	N	HQ	T

DISCUSSION: Several issues stand out from the analysis. One is that dealing with HQ takes a lot of time and results in long meetings. Another is that very little future planning is being done. The Deputy Director also spends a large amount of time dealing with technical matters. While this has perhaps been caused by the reflight issues, it does seem large for a top level manager.

An interesting result is that very little time (7.25 hours) is spent on personal matters with JSC staff. One of the results of previous work on operational environments is that a large amount of time is spent by top level management in this area.

LEVEL
NUMBER OF MEETINGS

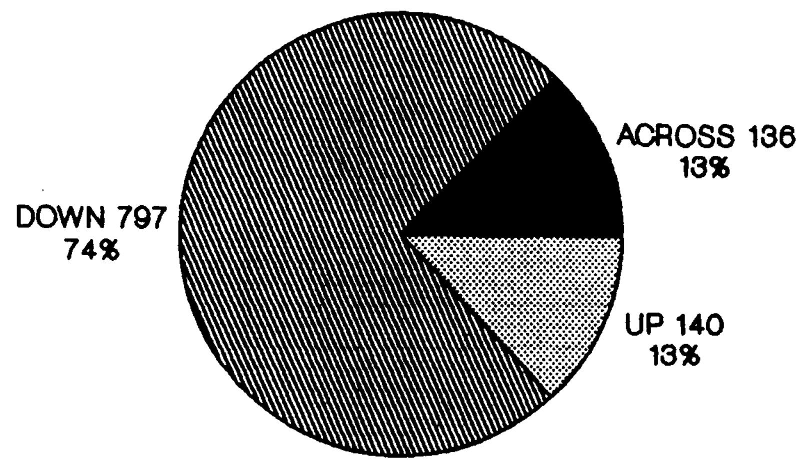


CHART 1

TIME OF MEETINGS

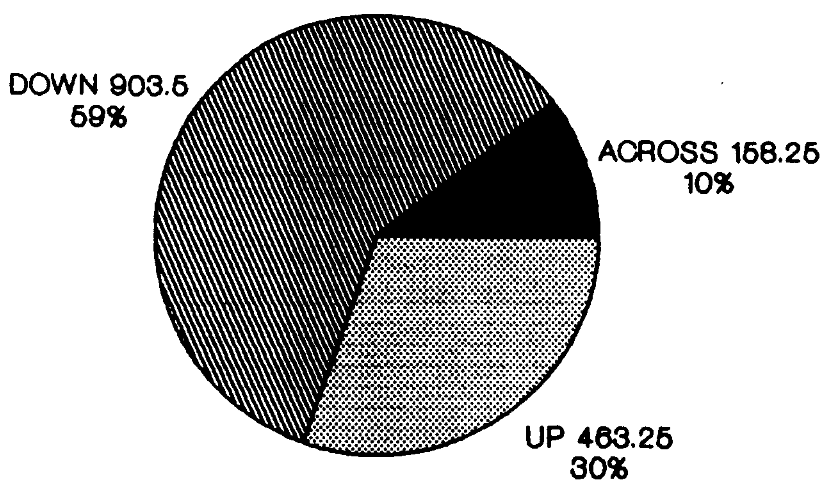


CHART 2

TIME FRAME
NUMBER OF MEETINGS

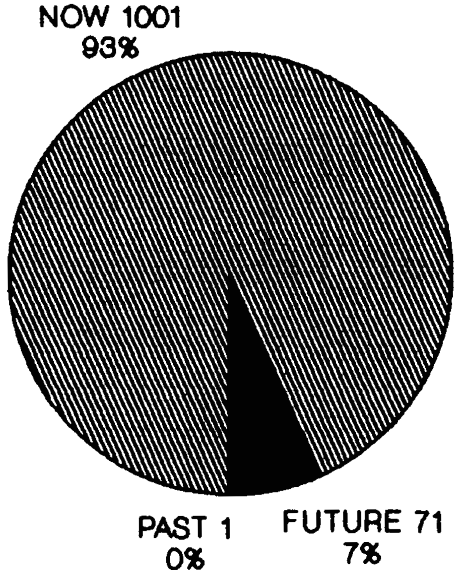


CHART 3

TIME OF MEETINGS

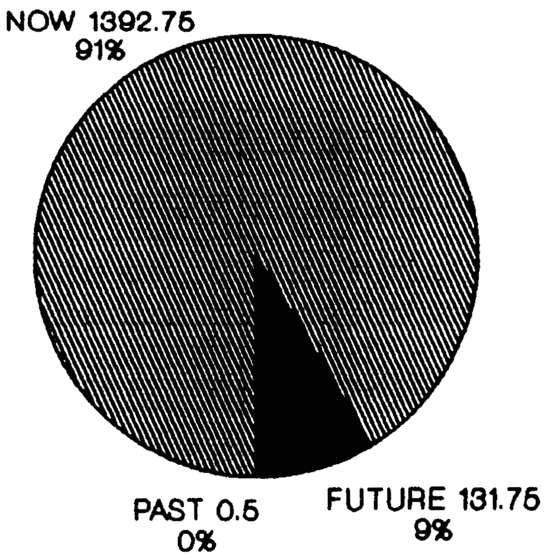


CHART 4

LOCATION
NUMBER OF MEETINGS

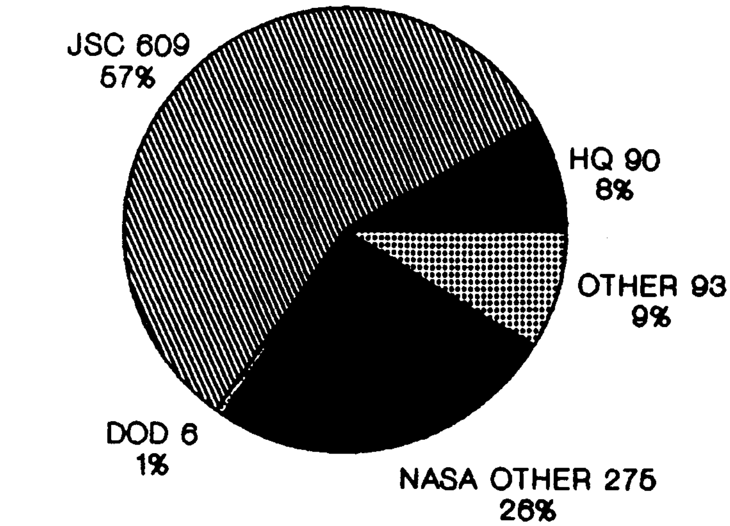


CHART 5

TIME OF MEETINGS

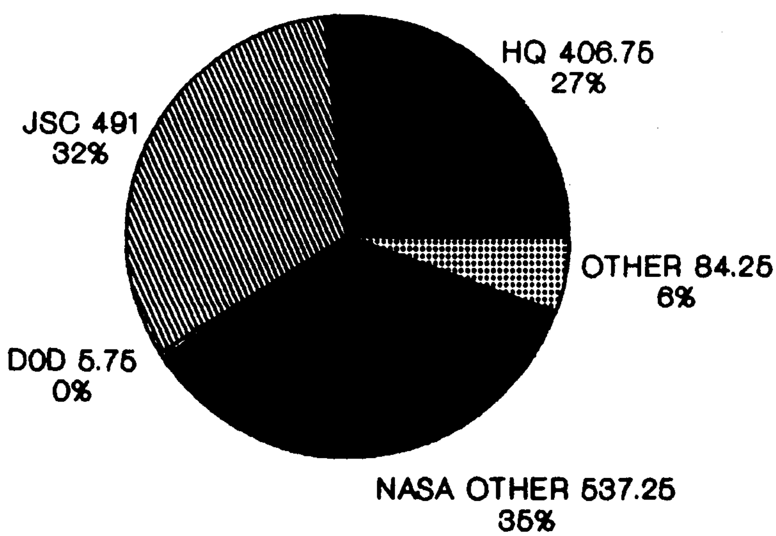


CHART 6

SUBJECT
NUMBER OF MEETINGS

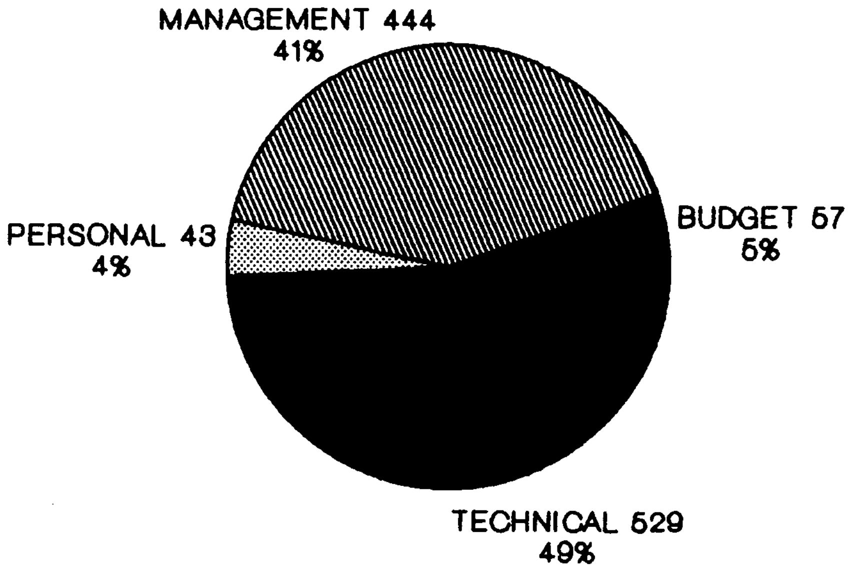


CHART 7

TIME OF MEETINGS

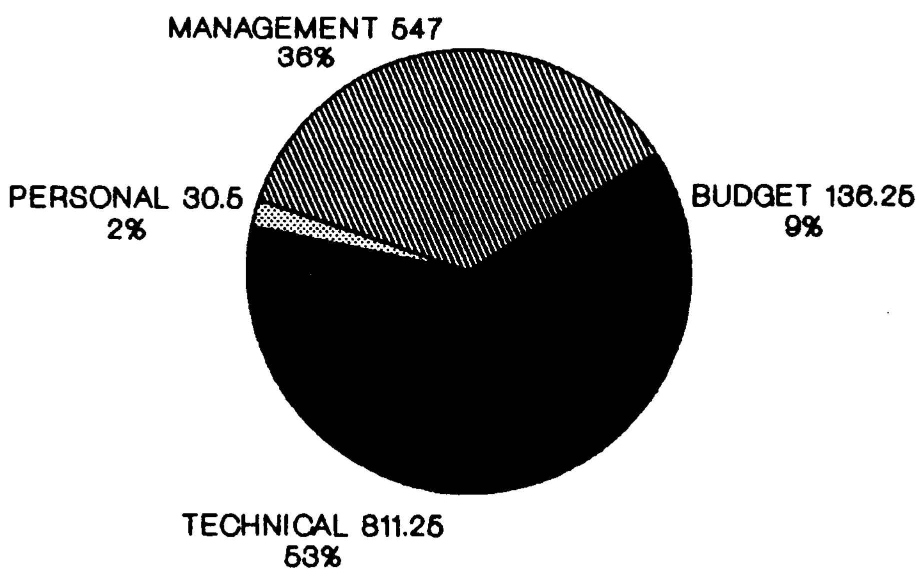


CHART 8

AGENDA SUMMARY CHART
BY MAJOR CATEGORY
1987

NUMBER OF OBSERVATIONS	1073
TOTAL TIME	1525 HOURS
AVG. TIME/OBSERVATION	1.42 HOURS

LEVEL

	NUMBER		TIME		AVG. TIME
ACROSS	136	12.7%	158.25	10.4%	1.16
DOWN	797	74.3%	903.50	59.2%	1.13
UP	140	13.0%	463.25	30.4%	3.31

TIME FRAME

	NUMBER		TIME		AVG. TIME
FUTURE	71	6.6%	131.75	8.6%	1.86
NOW	1001	93.3%	1392.75	91.3%	1.39
PAST	1	0.1%	0.50	0.0%	0.50

LOCATION

	NUMBER		TIME		AVG. TIME
DOD	6	0.6%	5.75	0.4%	0.96
HQ	90	8.4%	406.75	26.7%	4.52
JSC	609	56.8%	491.00	32.2%	0.81
NASA OTH.	275	25.6%	537.25	35.2%	1.95
OTHER	93	8.7%	84.25	5.5%	0.91

SUBJECT

	NUMBER		TIME		AVG. TIME
BUDGET	57	5.3%	136.25	8.9%	2.39
MGMT.	444	41.4%	547.00	35.9%	1.23
PERSONAL	43	4.0%	30.50	2.0%	0.71
TECH.	529	49.3%	811.25	53.2%	1.53

	NO.	LEVEL	TEMP.	LOC.	SUBJ.	TIME	T/N	% BY N	% BY T
1	246	D	N	JSC	M	167.50	0.68	22.9%	11.0%
2	231	D	N	NO	T	449.50	1.95	21.5%	29.5%
3	194	D	N	JSC	T	163.75	0.84	18.1%	10.7%
4	48	A	N	JSC	M	46.00	0.96	4.5%	3.0%
5	44	U	N	JSC	M	42.25	0.96	4.1%	2.8%
6	34	U	N	HQ	T	125.25	3.68	3.2%	8.2%
7	32	U	N	HQ	M	200.00	6.25	3.0%	13.1%
8	26	D	N	O	M	28.00	1.08	2.4%	1.8%
9	23	D	F	JSC	B	24.75	1.08	2.1%	1.6%
10	19	A	N	JSC	T	17.50	0.92	1.8%	1.1%
11	17	A	N	O	P	11.50	0.68	1.6%	0.8%
12	15	A	N	NO	T	14.00	0.93	1.4%	0.9%
13	15	D	N	O	T	12.50	0.83	1.4%	0.8%
14	12	U	F	HQ	B	55.75	4.65	1.1%	3.7%
15	11	D	N	JSC	P	7.25	0.66	1.0%	0.5%
16	11	D	N	O	P	7.25	0.66	1.0%	0.5%
17	10	D	N	NO	M	10.25	1.03	0.9%	0.7%
18	8	D	N	JSC	B	5.00	0.63	0.7%	0.3%
19	6	A	N	O	M	6.75	1.13	0.6%	0.4%
20	6	A	F	O	M	5.00	0.83	0.6%	0.3%
21	5	A	N	NO	M	4.75	0.95	0.5%	0.3%
22	4	A	F	NO	B	20.50	5.13	0.4%	1.3%
23	4	D	F	JSC	T	4.25	1.06	0.4%	0.3%
24	4	D	F	O	M	3.50	0.88	0.4%	0.2%
25	3	A	F	JSC	M	4.00	1.33	0.3%	0.3%
26	3	A	N	DOD	M	3.75	1.25	0.3%	0.2%
27	3	D	N	HQ	T	3.50	1.17	0.3%	0.2%
28	3	U	N	JSC	T	3.25	1.08	0.3%	0.2%
29	3	U	F	HQ	M	3.25	1.08	0.3%	0.2%
30	2	A	N	NO	B	16.00	8.00	0.2%	1.0%
31	2	D	N	HQ	M	9.00	4.50	0.2%	0.6%
32	2	U	N	NO	M	9.00	4.50	0.2%	0.6%
33	2	A	N	JSC	B	2.50	1.25	0.2%	0.2%
34	2	A	N	O	T	2.50	1.25	0.2%	0.2%
35	2	A	F	NO	M	2.50	1.25	0.2%	0.2%
36	2	U	F	HQ	T	2.00	1.00	0.2%	0.1%
37	1	U	N	NO	T	8.00	8.00	0.1%	0.5%
38	1	U	N	HQ	B	7.00	7.00	0.1%	0.5%
39	1	U	N	O	P	2.00	2.00	0.1%	0.1%
40	1	U	N	O	T	2.00	2.00	0.1%	0.1%
41	1	D	N	NO	B	1.25	1.25	0.1%	0.1%
42	1	U	F	JSC	B	1.00	1.00	0.1%	0.1%
43	1	D	N	DOD	T	1.00	1.00	0.1%	0.1%
44	1	U	F	NO	B	1.00	1.00	0.1%	0.1%
45	1	D	N	O	B	1.00	1.00	0.1%	0.1%
46	1	D	F	JSC	M	1.00	1.00	0.1%	0.1%
47	1	U	N	HQ	P	1.00	1.00	0.1%	0.1%
48	1	D	F	O	P	1.00	1.00	0.1%	0.1%
49	1	D	F	O	T	0.75	0.75	0.1%	0.0%
50	1	D	N	DOD	M	0.50	0.50	0.1%	0.0%
51	1	D	N	DOD	P	0.50	0.50	0.1%	0.0%
52	1	D	F	NO	T	0.50	0.50	0.1%	0.0%
53	1	U	F	O	T	0.50	0.50	0.1%	0.0%
54	1	A	F	JSC	T	0.50	0.50	0.1%	0.0%
55	1	A	P	JSC	B	0.50	0.50	0.1%	0.0%

SUMMARY CHART ALL MEETINGS 1987

SAMPLE CLASSIFICATIONS AND GROUND RULES

SAMPLE CLASSIFICATIONS:

STANDUP	D	N	JSC	M
GA STAFF	D	N	JSC	M
STATUS TO COHEN	U	N	JSC	M
SR STAFF	A	N	JSC	M
PRCB (I/II)	D	N	NO	T
FRF	D	N	NO	T
SPRCB	D	N	NO	T
PDMR	U	N	HQ	M
SDRB OR SDR	D	N	NO	T
CLRB	U	N	HQ	T
MGT COUNC	U	N	HQ	M
COSTELLO/PROG CONT/ POP	D	F	JSC	B
FMEA/CIL OR CIL	D	N	NO	T
STRAT. PLANNING	?	F	?	M
LEV I PRCB	U	N	HQ	T
CREW ESCAPE	D	N	JSC	T
VLS EQ LOAN	D	N	NO	T
CIR	D	N	O	M
PEB	U	N	JSC	M
LAUNCH SIT FLOW REV	D	N	NO	T
GMSR	U	N	HQ	M
INTERVIEWS	D	N	O	P
MSFC/KSC POP REV	A	N	NO	B

GROUND RULES:

1. Noon board = 0.5 hrs.
2. Standup = 0.5 hrs.
3. Assume Deputy Director chairs both of above unless direct conflict with other meetings or travel.
4. PDMR/Mgmt Council = 8 hours.
5. FMEA/CIL are classified as down since they are a first time presentation to the Deputy Director.
6. Weekend/holiday meetings with start time only listed are classed as time = 1.0 hours.

APPENDIX II B

FIELD NOTES OF INTERVIEW WITH HL/P

SOUTH TEXAS NUCLEAR PROJECT (STNP)

FIELD NOTES
INTERVIEW WITH HL/P
SOUTH TEXAS NUCLEAR PROJECT (STNP)
ON 24 MAY 88
JLH-25 MAY 88

1. Attending from the NASA team were George Studor from the Program office, Randall Sitton, a research associate from the University of Houston, and John Hunsucker from the University of Houston.
2. Attending from the company were Jim Westermeier, general manager of the project from HL/P and Ken Hess, the project manager for Bechtel.
3. HL/P is serving as the overall project manager for themselves and three other owners: Central Power and Light, the City of San Antonio, and the City of Austin. There was some indication that Austin is going back out of the project.
4. JW works for the nuclear Group VP, Jerry Goldberg and reports to him. (See the org chart for more information.)
5. HL/P's role is to monitor the performance of the contractors and to direct and correct as required.
6. Bechtel reports to JW. They are the A/E firm and the engineer of record for the project. Bechtel assumed this role from Brown/Root. Bechtel is also the construction manager. (See the org chart for more information.) Ebasco is serving as the constructor.
7. JW stays current on engineering and makes the final decision changes on configuration changes. The Design engineer can make interim changes subject to the subsequent formal approval of the JW.
8. STNP has two units. Unit one is now on-line and HL/P is now the engineer of record on Unit one.
9. In their risk analysis they have 29 volumes. They refer to their document as a living document.
10. They use quantitative methods in their hazard analysis.
11. In statistical decision making, they use their own judgement and a staff statistician. In addition, they force presentors to reduce presentations to understandable terms.
12. Their primary hazard analysis system is that required by the Nuclear Regulatory Commission (NRC). To this they have overlaid a significant amount of their own systems.
13. One of their primary documents in hazard control is a Non Conformance Report (NCR). This can be filed by anyone at any level.
14. In general, a contractor fills out an NCR. This must be validated within 24 hours by both Q/A and safety. It is reviewed for safety implications for this plant and for other plants and entered into the national data base if necessary.
15. Typically, an NCR comes from the engineering department or maintenance and goes to the design engineer. It is very rare to have one go from the engineering department to the plant manager to the VP of ops to the GM down to the design engineer.
16. They have around 200 people on site to deal with NCRs.

17. I first described in rough terms the seal problem with Challenger and the meeting at Thiokol. Then I asked why something similar could not happen to them. Both KH and JW were adamant about the fact that an NCR would have been filed and that equipment is not operated when an NCR is filed against it. KW went on to describe a dry firing on Unit one. They activated Unit one with no fuel present and pressurized all lines and boilers. They brought the operating temperature up to operational level. At some point during this process, the contractor discovered that some of the material was substandard. An NCR was filed and KH gave his troops two hours to discover answers before he called off the firing. He also called JW immediately.
18. A non-conforming component can not be used-this is inviolate.
19. Q/A or engineering management can stop work.
20. To be effective, an NCR program must have both a lot of teeth and a lot of discipline.
21. In addition, for the NCR program to be effective, you must stand behind your managers.
22. Almost out of the blue, but perhaps based on comments made in the interview but more likely based on outside information, JW commented that the shuttle program needed to be pulled together stronger.
23. The responsibility to be the engineer of record will pass from Bechtel to HL/P. Then HL/P must decide whether they wish to do it or contract this activity out.
24. In order to pass the responsibility of being the engineer of record from Bechtel to HL/P on Unit one they had a formal decision process consisting of a series of reviews. They started with the design process to insure that design/decision considerations were not lost. Everything was taken back to basic assumptions, documented, and cross referenced. This document is a living document. As changes to design are made, the change and the rationale for the change are included in the document.
25. They refer to this process as the "Modification Program". Emphasized again that all rationale is included in the package.
26. In the modification program there is no substitute for discipline and detail.
27. One of the major attributes of the STNP project is a far reaching, complex document control process. This cost a lot up front but has paid for itself many times over.
28. The documentation system is one of the hardest but most important steps in going operational.
29. (My thoughts---One reason they have to have such a tightly controlled system on documentation is because the responsibility for being the design engineer changes hands. In order to run a plant, you have to know how/why things were done the way they were.)
30. NRC tests their documentation program by sending them the names/descriptions of 12 components which are safety related. Two weeks later, NRC then shows up on site and

expects to see the complete documentation on the 12. In addition on the day they arrive they give the plant the names/descriptions of 6 more components and expect to see the documentation within 24 hours.

31. HL/P has a fairly small (200 or so) people on the design side of the house. These will, for the most part, be absorbed into the operating staff once the design process is over.

32. They have intentionally used mostly local people for entry level jobs.

33. They have a fairly strong educational incentive program. They have a training facility already in place. They have a contract with Wharton Jr. College to teach lower level courses at the facility. They have another contract with the University of Maryland to finish of the training with a BS in nuclear science. About 40 employees per year are allowed in the program.

34. They also use salary considerations and employee clubs as incentives. They do not use quality circles.

35. A large number of the Bechtel and Embasco employees are hired away be HL/P.

36. According to JW, the best motivator is good leadership. To emphasize this point KH pointed out that even though JW had both his office and home in Houston, he stayed on site and had an apartment nearby.

37. JW made the additional point that technical areas tend to be over managed and under led. Upper level management must provide clear direction and guidance.

38. When they finally go on line, they will have about 1200 in operations and 300 in support areas.

39. They do not have a formal program for the fast tracking of rising stars. They do have an effective informal program.

40. Comments on going operational:

1) A major problem is the consistent tendency to under estimate the size and complexity of the problem and to over estimate abilities.

2) Going operational on Unit one was a major test of their people. This process brought to the surface the real players.

3) There was a tremendous excitement in going operational and crossing the finish line.

4) Their stress level is very high but went up as they went operational.

41. There is a major amount of pride involved with the job. You have to get the people both emotionally and personally involved so that they have pride of ownership.

42. They had a real problem at first in overcoming the separate corporate identities of all the corporations involved: HL/P, Bechtel, Ebasco, Westinghouse, etc. They changed this so that people identified with STNP as opposed to their individual corporations. They used a little symbolic reorientation here by changing the logo on the hard hats to reflect STNP. Now all hard hats have this logo and are (I believe) the same color) as opposed to each

corporation having an individual hard hat.

43. At some point previously, they slimmed down the organization and removed many of the marginal performers. This was probably around the time they changed to the STNP identity.

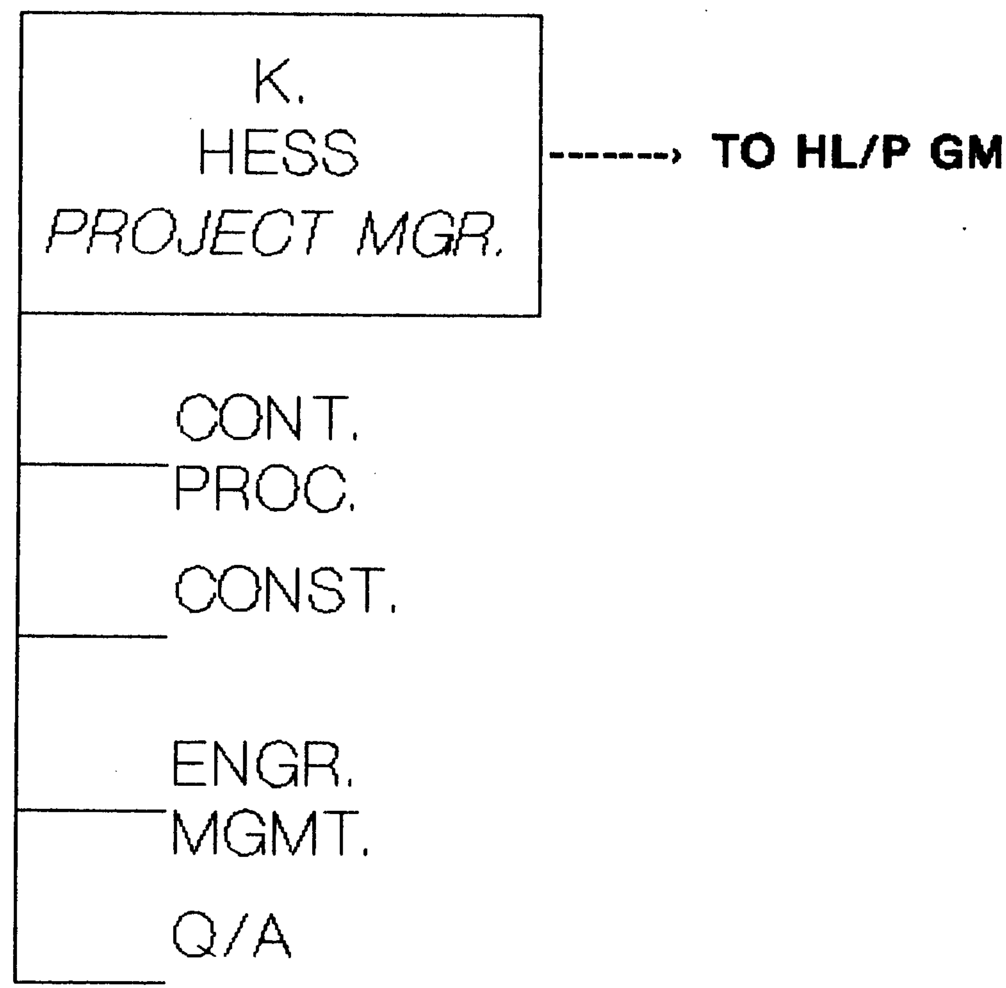
44. Their scheduling is open to everybody and is very public. Major milestones go all the way down to the crafts level. Everyone is aware of these and works towards them.

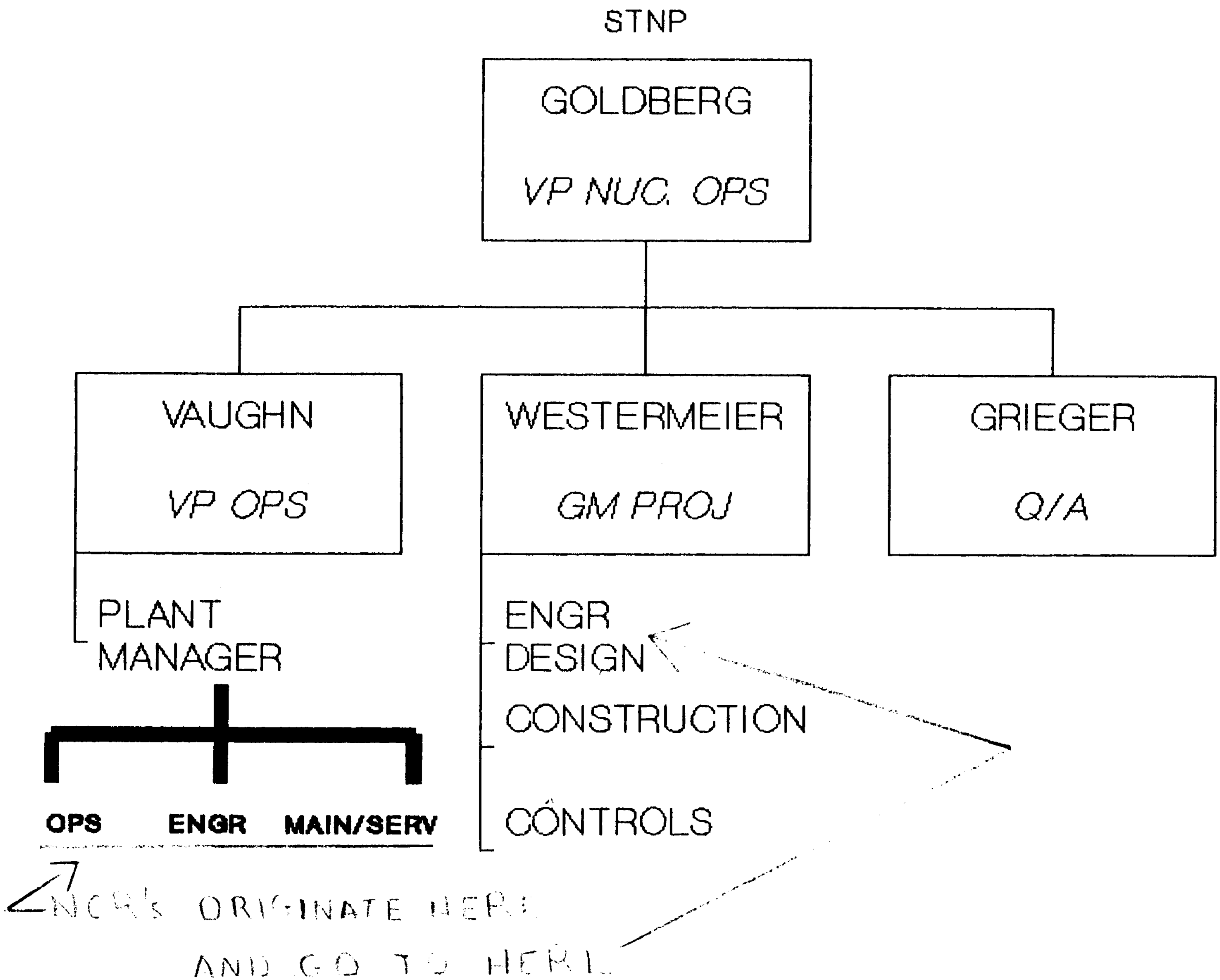
45. They implied that they use a significant amount of top down communication to keep employees informed and aware. (This is, of course, a significant part of establishing ownership.)

46. They have a very detailed scheduling system and can produce schedules with any level of detail.

47. They have schedule and cost people assigned to each office now. There was an implication that this will continue when they leave construction and go to operations.

BECHTEL




STNP
GOLDBERG
VP NUC. OPS
VAUGHN
VP OPS
WESTERMEIER
GM PROJ
GRIEGER
Q/A
PLANT MANAGER
OPS
ENGR
MAIN/SERV
ENGR DESIGN
CONSTRUCTION
CONTROLS
NCR's ORIGINATE HERE
AND GO TO HERE

APPENDIX II C

PUBLICATION/PRESENTATION OF RESEARCH

APPENDIX II C

PUBLICATION/PRESENTATION OF RESEARCH

1. TRANSITION LIFE CYCLE - AN R&D TO OPERATIONS PERSPECTIVE

   - Modified And Resubmitted For Publication In The IEEE Transactions on Engineering Management.

2. R&D TO OPERATIONS TRANSITION MANAGEMENT

   - Presented At The National Decision Science Institute Annual Meeting In Honolulu, Hawaii, Nov. 23-25, 1986.

   - Working Paper, University of Houston, Houston, Texas.

3. TRANSITION MANAGEMENT - A STRUCTURED PERSPECTIVE

   - Published In The Proceedings of The International Conference on Engineering Management: Theory and Application, Swansea, England, (September 15-19, 1986).

   - Accepted For Publication In The IEEE Transactions On Engineering Management.

4. TRANSITION MANAGEMENT - A PERSPECTIVE

   - Published In The Proceedings Of The 24th Annual Southern Management Association Meeting at Atlanta, Georgia, November 12-15, 1986.

5. TRANSITION MANAGEMENT OF AN ORGANIZATION

   - Working Paper, University of Houston, Houston, Texas.

6. AN INDUSTRIAL INSIGHT INTO THE MANAGEMENT OF CHANGE

   - Submitted For Publication in the Long Range Planning Journal.

7. DISASTER ON FLIGHT 51-L: AN IE PERSPECTIVE ON THE CHALLENGER ACCIDENT

   - Published in Industrial Management, Vol. 28, No. 5, 1986.

8. OPERATIONAL ARM FOR THE SPACE SHUTTLE PROGRAM : A PERSPECTIVE DIRECTION

   - Modified And Resubmitted For Publication to the Journal of Society of Logistics Engineers.

9. AN ANALYSIS OF THE FLIGHT RATE CAPABILITY OF NASA'S NSTS PROGRAM

   - Published in the Logistics Spectrum, Vol 21, No. 3, 1987.

10. OPTIMAL SCHEDULING IN AN M-STAGE FLOW SHOP WITH MULTIPLE PROCESSORS

    - Presented at the Joint National Meeting of TIMS/ORSA at New Orleans, Louisiana, May 4-6, 1987.

    - Submitted for Publication in the IIE Transactions.

11. AN ENGINEERING MANAGEMENT PERSPECTIVE ON TRANSITION MANAGEMENT

    - To Be Presented At The Ninth Annual Conference Of The American Society For Engineering Managment In October, 1988.

    - To Be Submitted To The American Society Of Civil Engineers' Journal Of Management In Engineering

12. THE EFFECT OF MULTIPLE FACTORS ON TRANSITION MANAGEMENT STRATEGY

    - To Be Submitted To The American Society Of Civil Engineers' Journal Of Management In Engineering

13. BRANCH AND BOUND ALGORITHM FOR A FLOW SHOP WITH MULTIPLE PROCESSORS

    - Presented at the Joint National Meeting of TIMS/ORSA at Washington, DC, April 25-27, 1988.

    - Submitted for Publication to the European Journal of Operations Research, 1988.

CHAPTER III

BRANCH AND BOUND ALGORITHM FOR A FLOW SHOP WITH MULTIPLE PROCESSORS

III. BRANCH AND BOUND ALGORITHM FOR A FLOW SHOP WITH MULTIPLE PROCESSORS

1.0 INTRODUCTION

A flow shop sequencing problem is characterized as the processing of n jobs on m machines. The machines are laid out in a unidirectional flow pattern and each job is processed identically in the fixed ordering of the machines. The objective of job scheduling can be that of minimizing the maximum completion time required to complete the processing of all of the jobs on all of the machines (makespan), the average time to complete all of the jobs (mean flow time), or any other regular measure of performance. More detailed work could involve the optimization of multiple objectives, or goals.

The sequencing of a flow shop with multiple processors (FSMP) at each stage is a generalization of the flow shop problem. It involves sequencing of n jobs in a flow shop where, for at least one stage, the processor has more than one identical machine. Stated another way, the problem is a special case of a general job shop problem in which all jobs to be scheduled follow the same machine sequence and there is more than one machine for at least one stage. The problem was first identified by Salvador (1973). He suggested a branch and bound approach to solve the problem for the permutation FSMP. Wittrock reports more work on the development of periodic (1985a) and non-periodic (1985b) scheduling heuristic algorithm. He calls the problem as flexible flow lines and proposes to solve it by decomposing into primarily two subproblems; the first one consists of machine allocation, and the other is that of sequencing jobs on each machine. The two authors also points out numerous real life applications of the problem. Kochlar and Morris (1987) report work on the development of the heuristics which considers setup times, finite buffers, blocking and starvation, machine down time, and current and subsequent state of the line. The heuristics developed try to minimize the effect of setup times and blocking. Further work has been reported by Brah and Hunsucker (1987) in the development of mathematical formulation, primarily useful for small size problems. However, much work still remains to be done and there is a need for an in depth study to determine methods of solving widespread problems.

The purpose of this paper is to present a branch and bound algorithm to solve scheduling problem of minimizing the makespan in a FSMP. The lower bounds and elimination rules developed in this paper for the makespan criteria are based upon the generalization of the flow shop problem. They have substantially helped to exhibit the usefulness of the algorithm for much larger problem size. Furthermore, a computational algorithm, along with results, is presented to demonstrate the working of the solution method. The branch and bound algorithm can also be used to optimize other measures of performance.

2.0 BACKGROUND

An important aspect when dealing with the scheduling problems is that even the simplistic case of a static flow shop minimizing the makespan belongs to the family of combinatorial problems. The complexity of the problem is further increased by the fact that unlike the single machine case, the inserted idle time may be advantageous. Further, it has also been shown that the three or more machine permutation flow shop and job shop problems are NP-complete problems (Gonzalez and Sahni 1978). Therefore the complexity of the problem strongly suggest that an exact polynomial bounded method for solution is highly unlikely. Further discussion on the complexity of the scheduling problems, among others, is contained in Garey et al. (1976), Garey and Johnson (1979), King (1979), and Cho and Sahni (1981).

3.0 PROBLEM DESCRIPTION

The problem of FSMP scheduling can be presented graphically as in Figure 3.1. There is a main queue of incoming jobs, and each job can advance to any one of the M_1 machines at stage 1. As can be seen in Figure 3.2, there is a queue at each stage of the flow shop processing, and theoretically all of the jobs can be routed to any one of the M_j machines ($1 \leq j \leq m$) at stage j. When the job has been processed through the last stage m, using one of the M_m machines, it is complete and at that point can leave the system. As is shown by Brah and Hunsucker (1987), the jobs

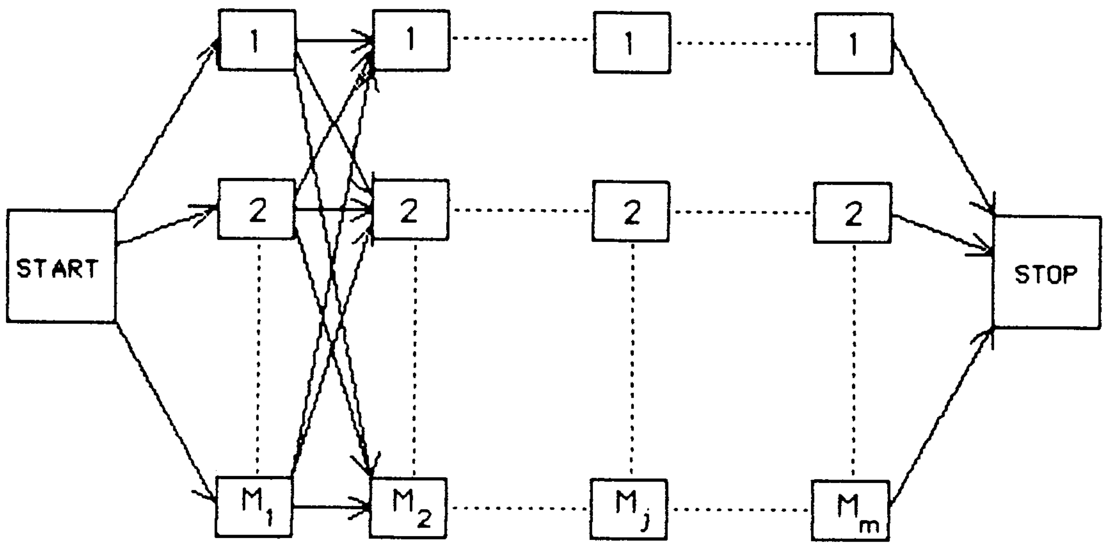


FIGURE 3.1 SCHEMATIC REPRESENTATION OF A FLOW SHOP WITH MULTIPLE PROCESSORS.

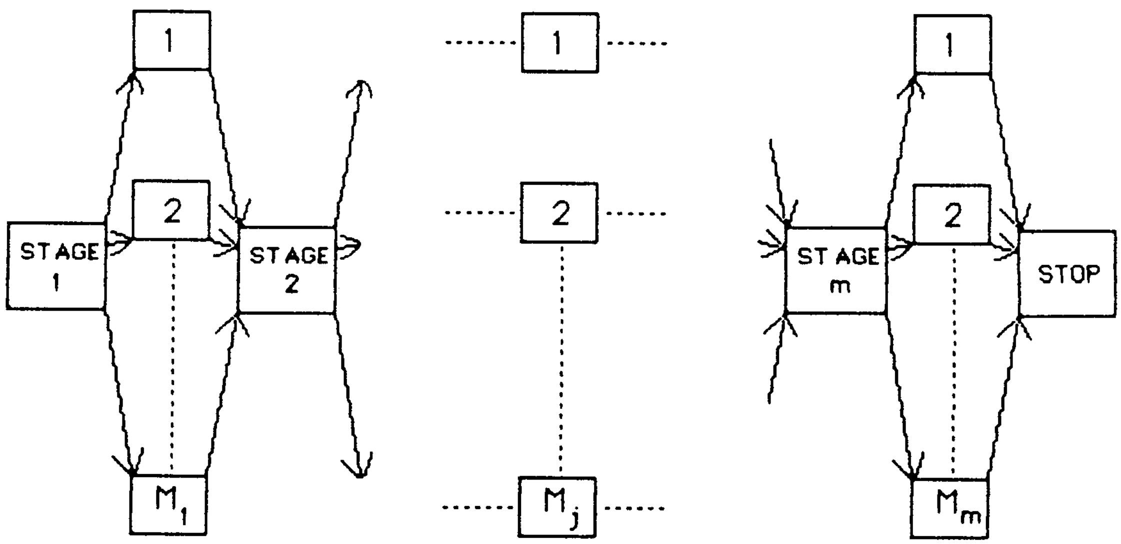


FIGURE 3.2 QUEUING REPRESENTATION OF A FLOW SHOP WITH MULTIPLE PROCESSORS.

can take $\prod_{j=1}^{m} \binom{n-1}{M_j-1} \frac{n!}{M_j!}$ possible sequence combinations, or paths for a schedule.

Before an effort is undertaken to understand the sequencing process, it will be wise to limit the study to reasonable bounds by making some assumptions. In order to achieve the limiting of the varieties of arrangements, the following assumptions are therefore made:

- o Each job is an entity, even though the job is composed of distinct operations, no two operations of the same job may be processed simultaneously.
- o The number of jobs is known and fixed. No job may be cancelled before completion.
- o The arrival time, or release time, of the jobs is known and fixed.
- o The processing times of the jobs are known and constant.
- o Setup time is considered a part of processing time.
- o Setup time is independent of the job sequence.
- o All jobs follow the same machine sequence.
- o No job may be split or pre-empted.
- o The flow shop consists of $m \geq 2$ stages or levels.
- o Each level or stage has $M_j \geq 1$ machines; $j= 1,\ldots,m$; with inequality holding for at least one M_j.
- o All machines are available at the beginning and never breakdown during the scheduling period.

- No machine may process more than one job at a time.
- Machines may remain idle.
- In-process inventory is allowed.

4.0 APPLICATIONS OF THE PROBLEM

The application of this type of problem occurs more often than one would imagine. Many high volume production facilities have several independent flow shops. The process in such facilities is such that machines are interchangeable at each stage and are therefore practically similar. Salvador (1973) first recognized the problem in the polymer, chemical, process and petrochemical industries where there are several parallel plants which can be considered as flow shops, and the jobs can practically be processed at any one of the plants at each stage of the processing. Assembly lines, in which more than one product is manufactured and each work station has multiple machines, is also an obvious application of this problem. Similarly, the situation where a parallel machine(s) is (are) added at one or more stages of the flow shop to ease the pressure on bottle neck facilities, and/or to increase the production capacities can be viewed as an application of the suggested problem.

Similarly, there are situations analogous to production systems where the similarity of a FSMP can be established. Consider for example the running of a program on a computer for a language like FORTRAN. The three steps of compiling, linking and running are performed in a fixed sequence. If there are multiple jobs (computer programs) requiring all of these facilities (steps), each having multiple processors (softwares), the process resembles that of a FSMP. There are similar examples in computers, telecommunications, group technology applications, flexible manufacturing systems, and others. The objective function in all of these functions could be the optimization of any one or more regular measures of performance.

5.0 BRANCH AND BOUND PROCEDURE

The absence of algorithms to solve most real life scheduling problems has given rise to the effort to use general purpose optimization methodologies such as mathematical and dynamic programming, and branch and bound techniques. These methods, however, require quite extensive computations in order to find an optimum solution for large scale problems. Other efforts have been concentrated on developing near optimal solutions by way of useful heuristics. In most studies involving heuristics, the optimal solution though branch and bound techniques have been most widely used to examine their performance.

Basically, the branch and bound methods are related to dynamic programming in the sense that both are enumeration techniques that are expected to perform partial enumeration in most of the cases. Both branch and bound and dynamic programming are optimizing techniques which apply to a much larger class of problems than just those in production scheduling. They explore the decision tree in an intelligent fashion and in essence, use an implicit enumeration method to determine on route which branches need to be fully explored. Further, the efficiency of the branch and bound algorithm depends upon the selection of lower and upper bounds and elimination rules, which in turn establishes the breadth of the search tree.

The branch and bound methods in flow shop scheduling have been widely used for finding optimal or near optimal solution methods. Ignall and Schrage (1965), Lomnicki (1965), McMahon and Burton (1967), Ashour (1970), Gupta (1970), Lageweg et al. (1978), and Bansal (1979) among others have developed different branch and bound methods for various measures of performance like makespan, mean flow time, mean tardiness and maximum tardiness. The difference and the efficiencies of the branch and bound algorithms is in the choice of the lower bound and elimination rules. The strong bounds and elimination rules eliminate relatively more nodes of the search tree which very often brings in more computation requirements as well. If such needs are excessively large, it may become advantageous to search through larger nodes using a weaker, but fast computable lower bound. However, the advantages of stronger bounds and elimination rules are more substantial in large scale problems (Baker 1975).

The branch and bound algorithm of a FSMP consist of three basic steps; the calculation of lower bounds, branching, and node elimination. The branching procedure can take place through several selection rules like the least lower bound, first come first served, or depth first least lower bound rule, etc. (Kohler and Steiglitz 1976). The nodes exploring process can take advantage for computational techniques like parallel processing. It can also use different search procedures such as a filtered beam search technique (Ow and Morton 1988). In any situation, as soon as the lower bound of the node equals or exceeds the upper bound of the complete problem, the node is eliminated from further consideration. Naturally, a characteristic function like makespan, mean completion time, or any other measure of performance can be used to eliminate a partial permutation which does not have a feasible and/or optimal solution.

To begin with, some notation is needed. Let:

n = Number of jobs;

m = Number of stages;

i = The job number, $i = 1, \ldots, n$;

j = The machine stage number, $j = 1, \ldots, m$;

M_j = The number of parallel machines at stage j;

p_{ij} = Processing time of job i at stage j;

N = A set containing all jobs;

A = A set of some jobs such that $A \subset N$;

A' = A set of jobs containing all jobs in the set A, and a job q, where $q \notin A$.

6.0 DETERMINATION OF LOWER BOUNDS

In order to solve the problem of optimal, or near optimal, scheduling in a FSMP using the branch and bound method, a related sub-problem must be solved. This problem involves finding a lower bound on each node for the desired performance measure. To find such a lower bound at each branching node, two contiguous partial schedules must be considered. Let the first of these partial schedules (i.e. the partial sequence at the start of the schedule) involving all jobs on all machines through stage j-1, along with the sequence of job set A, at stage j, be represented by $S_j(A)$. Also let A' represent the augmentation of an unscheduled job q at stage j to the set of jobs A, such that $q \notin A$. Then, $S_j(A')$ represents a schedule formed by appending job q to $S_j(A)$. The second schedule, $S_j'(N-A')$, will consist of the remaining jobs not contained in the schedule $S_j(A')$ at stage j, and all jobs beyond stage j in an arbitrary sequence. The notation $S_j(A')S_j'(N-A')$ will then be used to represent a complete schedule of jobs at stage j and all subsequent stages.

For a given partial sequence $S_j(A)$, let $C[S_j(A), k]$, represent the completion time of the partial sequence on machine k belonging to one of the M_j parallel machines at stage j. The equations involving completion times of the partial sequence $S_j(A')$ on each machine k can be calculated recursively as follows:

$$C[S_j(A'), k] = \begin{cases} \max\{C[S_j(A), k]_{q\in k}, C[S_{j-1}(A'), k_1]_{q\in k_1}\} + p_{qj} \\ \qquad \text{If q is processed on k, at stage j.} \\ C[S_j(A), k] \qquad \text{Otherwise.} \end{cases} \tag{3.1}$$

where

$$C[S_0(A), 0] = C[S_j(\phi), k] = 0 \quad \text{for all j and A.}$$

and

$C[S_0(A), 0]$ = Completion or arrival time of all jobs at the start of processing;

$C[S_j(\phi), k]$ = Completion time of the empty set at stage j.

Thus in order to minimize the maximum completion time, $\max_k \{C[S_m(N), k]\}$, must be minimized. Here, $S_m(N)$ is the complete sequence of all jobs at the last stage. Similarly in order to minimize the mean completion time of the jobs, $\{\sum_{k=1}^{M_m} C[S_m(N), k] / M_m\}$, needs to be minimized.

Several researchers have developed branch and bound formulations of the flow shop problem. The major difference in the approaches has been in the calculation of the lower bounds. A variety of lower bounds for minimizing the makespan have been developed which can generally be classified as machine based bounds, job based bounds, and composite bounds. These lower bounds for the flow shop are discussed by Gupta (1970) and Baker (1975). Their results are used in this research as a basis for the development of lower bounds which is presented below for a FSMP. Salvador (1973) has also developed machine based bounds for the permutation FSMP. The machine based bounds developed here, however, are generalized lower bounds for the FSMP problem which considers permutation and other schedules. Moreover, it turns out that the computation requirements of the machine based bounds developed here are much less, since only a subset of jobs are explored. Besides, it also results in making them stronger lower bounds, therefore considerably decreasing the number of nodes searched. Furthermore, job based lower bound and elimination rules proposed here also serves to reduce the number of nodes explored in the branching tree.

6.1 MACHINE BASED BOUNDS

If a job q is being considered for augmentation to a partial schedule $S_j(A)$ at stage j, then for a FSMP scheduling problem, the unprocessed work load at any stage can be utilized in obtaining a lower bound for minimizing the makespan on that stage. Let the average completion time and processing requirement for stage j be represented as,

$$ACT[S_j(A')] = \sum_{k=1}^{M_j} C[S_j(A'), k] \ / \ M_j + \sum_{i \in (N-A')} p_{ij} \ / \ M_j.$$

The terms on the right hand side of the above equation are:

- o The average interval over which the machines are already committed after scheduling job q at stage j;
- o And the remaining average work load of unprocessed jobs required of machines M_j, at stage j.

First we will show that $ACT[S_j(A')]$ is a lower bound on the completion time of the jobs through stage j if this was the last stage of processing. Then we will develop the complete lower bound for the branching node. As defined above, $ACT[S_j(A')]$ is the average completion time of the jobs formed from the set of scheduled jobs A' and the remaining set of jobs N-A' in an arbitrary sequence on stage j. By definition $ACT[S_j(A')] \leq \max_k \{C[S_j(N), k]\}$, where $S_j(N)$ is the composite schedule of all jobs. Moreover, the jobs in N-A' must be assigned to some processors at stage j, which means that,

$$ACT[S_j(N)] = \sum_{k=1}^{M_j} C[S_j(N), k] \ / \ M_j.$$

Since the average is less than the maximum, $ACT[S_j(A')]$ is the lower bound on the completion time up to the stage j.

Further, let the maximum completion time for a scheduled workload be represented as,

$$MCT[S_j(A')] = \max_k \{C[S_j(A'), k]\} .$$

Note that $MCT[S_j(A')]$ is also a lower bound if stage j was the last stage of processing. Now, if it were possible to determine which job finished last on the stage j, then adding the remaining work load of the job will provide a lower bound on the makespan. However, the best that may be possible is to determine the conditions which predicate the set from which the last job comes. In order to compute the lower bound of the branching node at stage j, consider the following situation. If $ACT[S_j(A')]$ is greater than or equal to $MCT[S_j(A')]$, then obviously, in all cases, one of the remaining unscheduled jobs will be the last job processed at stage j, i.e., the last job at stage j comes from the set of jobs N-A'. Otherwise, if $ACT[S_j(A')]$ is less than $MCT[S_j(A')]$, then the last job may come from either the set of jobs in A' or N-A'. Nevertheless, the jobs in N-A' will dominate all other jobs.

Once we know the set of jobs from which the last job comes, the job in that set with least work remaining could provide the best possible results for minimizing the makespan. This gives the machine based lower bound for the branching node for stage j as follows,

$$
LBM[S_j(A')] = \begin{cases} ACT[S_j(A')] + \min\limits_{i \in N-A'} \sum\limits_{j'=j+1}^{m} p_{ij'} & \text{If } ACT[S_j(A')] \geq MCT[S_j(A')] \\ MCT[S_j(A')] + \min\limits_{i \in A'} \sum\limits_{j'=j+1}^{m} p_{ij'} & \text{Otherwise.} \end{cases} \tag{3.2}
$$

6.2 JOB BASED BOUNDS

The calculations for a job based bound focuses on the remaining processing required of each unscheduled job at each stage j. In a flow shop, there is only one route available for the jobs to process, which is not the case in a FSMP. Meaning, a job based bound for a FSMP cannot be as strong due to the presence of alternate routes for the other jobs in the set. Gupta (1970), and Baker (1975) give the following lower bound for the flow shop problem where only a single processor is permitted at each stage of processing,

$$
LBJ[S_j(A')] = C[S_j(A), k] + \max_{i \in N-A} \sum_{j'=j}^{m} p_{ij'} + \sum_{r \in N-A'} \min [p_{rj}, p_{rm}].
$$

The last term of the above equation holds only if there is only one processor at each stage. A modification of the above job based bound can be constructed by considering the unscheduled jobs in the set N-A' at stage j. All of the jobs in this set have to be scheduled and completed both on stage j and the rest of the processing stages. Therefore, if the maximum of these times is added to the shortest completion time of $S_j(A')$, the job based bound is determined. This gives the lower bound for the problem as,

$$LBJ[S_j(A')] = \min_k C[S_j(A'), k] + \max_{i \in N-A'} \sum_{j'=j}^{m} p_{ij'} \quad (3.3)$$

The advantages of the job based bound will become apparent when the number of jobs is close to the number of parallel processors at each stage. A reasonable assumption is that the dominance of the job in establishing a lower bound is more profound when there are less jobs for each parallel machine. Based upon a similar rationale, the usefulness of the job based bound in a FSMP is expected to be effective towards the end of the schedule at each stage. Also, the conditions which makes the bounds weaker are unexpected forced idle time on the machines and waiting times on the job. The job based bounds are generally more sensitive to such conditions and their effect is greater when the number of jobs and/or stages is large in a FSMP. Baker (1975) reports that job based bounds do not appear to be very effective for a flow shop problem. He suggests that they can be effective, if used in conjunction with the machine based bounds. This conjecture also seems to apply to a FSMP.

6.3 COMPOSITE BOUNDS

If we combine the job based bound with the machine based bound for computing the lower bound for a FSMP, we obtain a composite lower bound. McMahon and Burton (1967) have also suggested a similar composite lower bound based on the jobs and the machines for a pure flow shop. Therefore, the composite bound for a FSMP for the branching node at stage j ($1 \leq j \leq m$) is as follows,

$$LBC[S_j(A')] = \max \{LBM[S_j(A')], LBJ[S_j(A')]\}. \quad (3.4)$$

7.0 ELIMINATION METHODS

Elimination methods for the flow shop scheduling problem have been investigated by several authors. Szwarc (1971) presents a review of the successes and failures of elimination procedures and derives some properties. Baker (1975) discusses these methods and presents results which suggests that elimination strategies are not very useful by themselves. However, when elimination procedures are used in conjunction with lower bounds, they have been shown to be quite effective especially for large size problems. Nevertheless, the elimination strategies developed by Szwarc (1971, 1978), and further evaluated by Baker (1975) are primarily designed for permutation flow shop. They have their best utilization in the special case of a permutation FSMP, where the number of parallel processors at each stage is the same, meaning the machine allocation and sequencing decision is only made at the first stage.

Furthermore, the dominance conditions developed by Gupta (1975), Szwarc (1977), and Gupta et al. (1987) for the flow shop problem are applicable to the FSMP problem provided the jobs being compared use the same processors at all stages of processing. This is to say, that the set of jobs which are assigned to a particular processor at stage one will be assigned together to some processor at each subsequent stage, so jobs in some sense are grouped together. In this situation, there exists a flow shop inside the general problem of a FSMP for that subset of jobs. The best known dominance conditions as proposed by the above authors are briefly discussed here. Their use in the general case is rather limited. Nonetheless, the insight provided by them can be helpful for a FSMP.

7.1 KNOWN DOMINANCE CONDITIONS FOR THE FLOW SHOP PROBLEM

In order to explain the dominance conditions, let us consider $S_j(A)$ and $S^*_j(A)$ as permutations of the same jobs through the same set of processors at all stages of processing upto stage j. In general, the sequence $S_j(A)$ is said to dominate $S^*_j(A)$ (see Gupta 1971, Szwarc 1973) if,

$$C[S_j(A)] \leq C[S^*_j(A)] \qquad \text{for each } 1 \leq j \leq m.$$

Further, consider $S_j(A")$ which is different than $S_j(A')$ in that it contains a job r which precedes job q, and such that neither r nor q is in A. According to Szwarc (1978), the best known job dominance condition for any partial sequence $S_j(A'')$ over $S_j(A')$ is said to hold if,

$$C[S_j(A'')] - C[S_j(A')] \leq p_{rk} \quad \text{for all } (1 \leq j \leq k \leq m).$$

Further improvements on the job dominance conditions of the flow shop in terms of being less restrictive are presented by Gupta et al. (1987).

7.2 SOME EXTENSIONS FOR FLOW SHOP WITH MULTIPLE PROCESSORS

The following are some of the other obvious guidelines which can be used for the FSMP problem:

Recall that A' is the augmentation of job q to A. Now, consider A" as the augmentation of job r to A on the same processor as job q on stage j. Then the node $S_j(A'')$ may be eliminated from further consideration if,

$$C[S_j(A')_{q \in k_j}, k_j] \quad \leq \quad C[S_{j-1}(A'')_{r \in k_{j-1}}, k_{j-1}];$$

Here, $q \in k_j$ means that q was processed on processor k_j at stage j. The above relationship implies that if job q can finish processing at stage j before job r becomes available for processing at the same stage, then it is sufficient to consider a sequence on a processor k_j in which job r follows job q.

Also, if the augmentation of any job to A at stage j yields a lower bound which equals or exceeds the upper bound of the complete problem, then the node emanating from augmentation need not be considered. The upper bound of the problem is the best value of the complete schedule computed so far. As an initialization step, the upper bound of the problem would be set equal to a large number (larger than any possible schedule value) at the start.

Further, some other guidelines are presented in the form of the following two theorems. The first theorem is an extension of the flow shop results and is applicable in special situations as explained in its definition. The second one is a generalized theorem showing that for the maximum completion time criteria, it is sufficient to consider the nondelay schedules for the jobs going to a common processor at the last stage of processing in a FSMP.

<u>THEOREM 3.1</u>: Suppose there exists two jobs r and q such that r directly preceeds q on a common processors k_1 at stage 1 of a FSMP. Further assume that jobs r and q also use a common processor k_2 at stage 2. Then among the set of schedules with this property, for any regular measure of performance, it is sufficient to consider schedules in which the same processing sequence for r and q is followed on k_1 and k_2.

<u>PROOF</u>: Consider a schedule which has job r directly preceding job q on a processor k_1 at stage 1, and r following q, with perhaps some intervening jobs, on a processor k_2 at stage 2. On stage 1, we can exchange the

order of processing of q and r without increasing the starting time of any other jobs on k_2. Therefore, this exchange cannot increase the completion time or any regular measure of performance of such jobs.

As a direct consequence of Theorem 3.1, the following corollary holds.

COROLLARY 3.1: Suppose there exists a set of jobs J which uses a common processor k_1 at stage 1 and k_2 at stage 2 of a FSMP. Then among the set of schedules with this property, for any regular measure of performance, it is sufficient to consider schedules in which the same processing sequence for the jobs in J is followed on k_1 and k_2.

THEOREM 3.2: Suppose there exists jobs r and q in a FSMP that use a common processors k_m at stage m. Then among the set of schedules with this property, for the maximum completion time criteria, it is sufficient to consider schedules in which the processing sequence for r and q on k_m is the same as the arrival sequence from stage m-1.

PROOF: Consider a schedule which has job r finishing before job q on stage m-1, and has r following q, with perhaps some intervening jobs, on the same processor at stage m. Suppose we move job r immediately ahead of job q on k_m. Job r can then start no later than the previous starting time of job q on k_m since it finished before q on stage m-1. The most that can happen to job q and the jobs that may have been between r and q is that their completion times get increased by p_{rm}. Nevertheless, the processing time on the processor k_m can only be expedited, therefore, the maximum completion time cannot increase by the adjustment.

As a direct consequence of Theorem 3.2, the following corollary holds.

COROLLARY 3.2: Suppose there exists a set of jobs J which uses common processors k_{m-1} at stage m-1 and k_m at stage m of a FSMP. Then among the set of schedules with this property, for the maximum completion time criteria, it is sufficient to consider schedules in which the same processing sequence for jobs in set J is followed on k_{m-1} and k_m.

8.0 THE ENUMERATION OF ALL SEQUENCES

There are two decision activities which occur at each stage of the scheduling problem. The first decision is the assignment of the jobs to a specific machine k from M_j parallel machines, at stage j, and the second is the scheduling of jobs on every machine at that stage. The two decisions are closely linked and both of them effect the quality of the scheduling result. The enumeration method of Bratley et al. (1975) for scheduling on parallel machines has been used with some modification for the FSMP problem.

The enumeration of the problem is accomplished by generating a tree which contains two types of nodes. If the path passes through node (i), then the candidate job i is scheduled on the current machine. While, if the path passes through node $\boxed{i}$, then this job i is scheduled on a new machine, which now becomes the current machine. The number of $\square$ nodes on each branch establishes the number of parallel machines used by that branch, and obviously that must be less than or equal to the number of parallel processors M_j, at stage j. However, if the processing time and the cost of processing for all parallel machines $k \in M_j$ at stage j is the same, and the number of jobs is greater than or equal to the number of parallel processors M_j, for all j, then for any regular measure of performance it is not advantageous to keep one of the parallel machine idle for the entire duration, while the others are processing the jobs. Using this, the number of possible branches at each stage j, as established by Brah and Hunsucker (1987), would be,

$$N(n,M_j) = \binom{n-1}{M_j-1} \frac{n!}{M_j!} . \qquad (3.5)$$

This means, for an optimization problem of a flow shop with M_j processors at each stage j, the total number of possible end nodes equals,

$$S(n,m,M_j) = \prod_{j=1}^{m} \binom{n-1}{M_j-1} \frac{n!}{M_j!} . \qquad (3.6)$$

In order to construct a tree that has been discussed above for the stated problem, some definitions and rules at each stage j are necessary. Let the level O_j represent the root node at stage j, and 1_j, 2_j, ... , n_j represent different levels of the stage, with n_j being the last, or the terminal level of stage j. Since there are n jobs and m stages, the total number of levels will be n*m. The last level of the whole tree will be n_m corresponding to the terminal level of the last stage. The following are the necessary rules for the algorithm to develop the branching tree of the problem under consideration.

<u>RULE 1</u>: Level O_j contains only the dummy root node of stage j of the problem ($1 \leq j \leq m$).

<u>RULE 2</u>: Level 1_j contain the nodes $\boxed{1}$, $\boxed{2}$, .. , $\boxed{x}$, where $x = n - M_j + 1$.

<u>RULE 3</u>: A path from level O_j to level i_j, [$(1 \leq i < n)$ & $(1 \leq j \leq m)$] may be extended to the level $(i+1)_j$ by any of the nodes $\boxed{1}$, $\boxed{2}$, ..., $\boxed{n}$, 1, 2, ..., n provided the rules 4 to 7 are observed.

<u>RULE 4</u>: If $\boxed{k}$ or k has previously appeared as a node at level i_j, then k may not used to extend the path at that level.

<u>RULE 5</u>: $\boxed{k}$ may not be used to extend a path, at level i_j, which already contains some node $\boxed{r}$ with $r > k$.

<u>RULE 6</u>: No path may be extended in such a manner that it contains more than M_j square nodes at each stage j.

<u>RULE 7</u>: No path may terminate in such a manner that it contains less than M_j square nodes at each stage j unless the number of jobs is less than M_j.

Rule (1) is simply an indicator of the starting of a new stage. Rule (2) says that the first level of a stage j can only have x square nodes, where x is the index of jobs whose value is equal to $(n - M_j + 1)$. Any number larger than x will violate some of the other rules, specifically rules (5) and (7), and thus cannot be used to generate a square node at the first level. Rule (3) simply states that all unscheduled jobs at stage j are candidates for square and circle nodes as long as they do not violate any other rules, namely rules (4) to (7). Rule (4) is necessary to assure that no job is sequenced twice at one stage. Rule (5) is to avoid duplicate generation of sequences in the branching tree. The number of square nodes in the branching tree establishes the number of processors used in the sequence, and rule (5) guarantees that no more than M_j processors are used at stage j. Finally, as discussed before, there is no advantage in keeping a processor idle when the cost of processing is the same for all of the processors, thus rule (7).

Figure 3.3 gives a sample tree representation of a four job two parallel machine scheduling problem. The branching tree has thirty six end nodes. In seeking an optimal schedule, all of these end nodes can serve as a starting point for the next stage, which is O_{j+1} $(j < m)$. Now, all of the nodes at subsequent stages may not be candidates due to their higher value of lower bounds. Therefore, not all of the nodes need to be explored. Incidentally, it may be

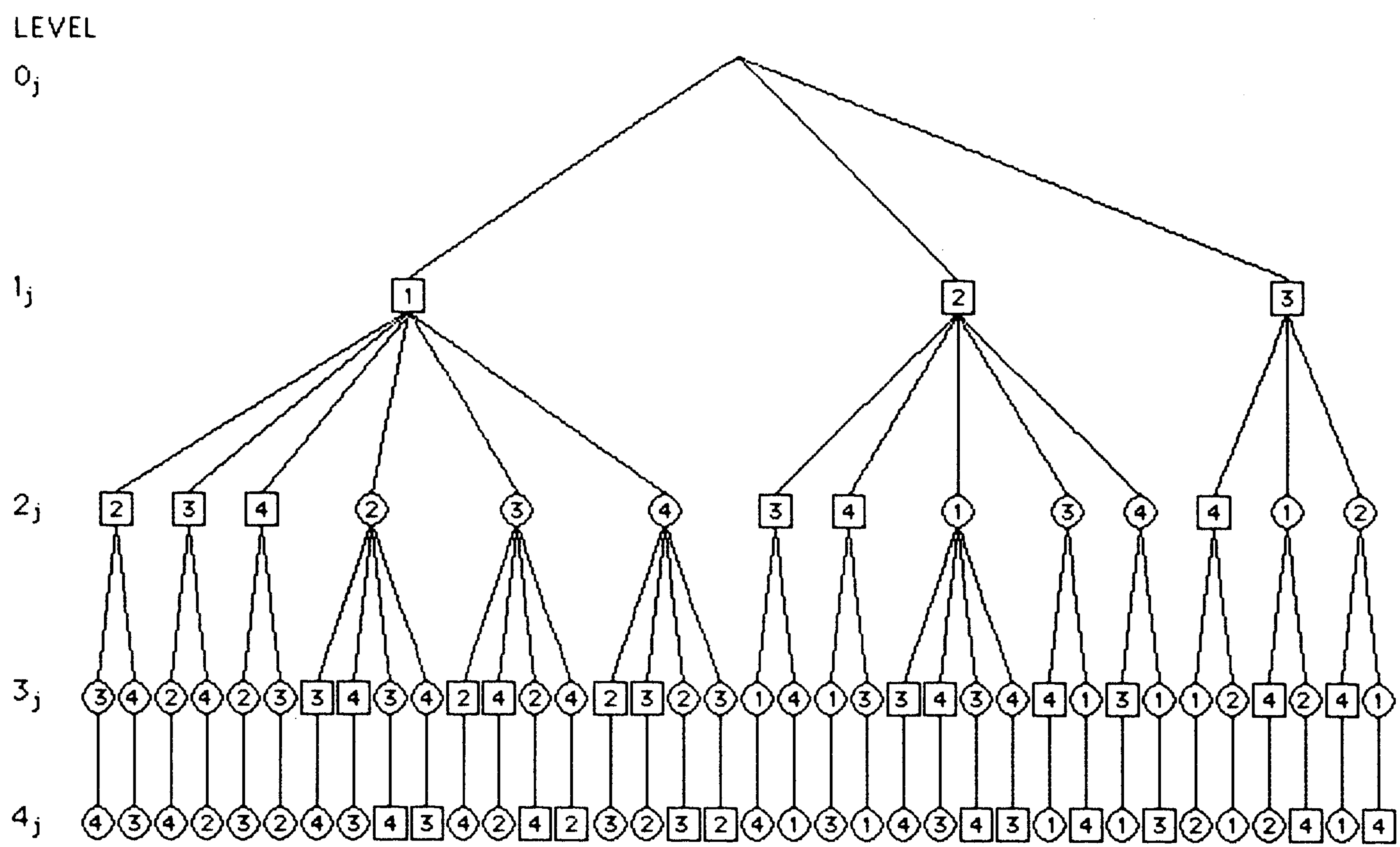


FIGURE 3.3 TREE REPRESENTATION OF FOUR JOBS ON TWO PARALLEL PROCESSORS.

observed that all of the jobs at stage j will not be readily available at the next stage and consequently inserted idle time will increase their lower bound and thus possibly remove them from further consideration. In other words, the sequencing pattern from stage to stage is not expected to deviate considerably in most real life situations, unless the data is so structured. This situation will help to reduce the span of the search tree. Moreover, the requirement of processing times on individual jobs and the difference in the number of parallel processors at each stage, etc., will further establish the breadth of the search tree.

In addition to the above, if the interest is in the subclass of the active schedules called nondelay schedules, then the number of search nodes could be further reduced. Nondelay schedules are defined as those in which no machine is kept idle when it could start processing some operation. The use of nondelay schedules does not necessarily provide an optimum solution. Nonetheless, the decrease in the number of the nodes searched provides a strong empirical reason to generate such schedules (French, 1982). Such procedures could be useful for large size problems, where the speed of computation becomes critical.

9.0 DEVELOPMENT OF A COMPUTATIONAL ALGORITHM

The selection of a search method for the branch and bound algorithm is a function of several factors of which the most significant ones are the available memory size of the computation machine and the problem dimensions. Based upon these considerations, the branch and bound algorithm for a FSMP developed here uses a variation of the depth first least lower bound search technique. Knowing the constraint on the memory size, this allows a fairly large problem size to be solved using this method. Furthermore, the computation speed of the algorithm has been observed to be consistently fast even for problems of modest size, although no comparisons are available to justify the claim. The branch and bound algorithm for generating a solution for optimizing makespan is as follows:

> STEP 1: Generate 1, ..., $(n-M_1+1)$ square nodes at stage 1, and compute their lower bounds. Encode the necessary information about the nodes, and add them to the list of unprocessed nodes. Also, initialize a node for the termination of the computational algorithm.
>
> STEP 2: Remove a node from the list of unprocessed nodes with the priority given to the deepest one in the branching tree with the least lower bound. Break ties arbitrarily.
>
> STEP 3: Procure all information about the retrieved node. If this is one of the end nodes of the branching tree go to step 5, while if this is the last node of the unprocessed nodes list then go to step 6, otherwise move to the next step.
>
> STEP 4: Generate branches from the retrieved node using the algorithm for node generation and compute their respective lower bounds. Discard the nodes with the lower bound value larger than the complete solution. Add the remaining nodes to the list of unprocessed nodes and go to step 2.
>
> STEP 5: Save the current complete branching path, or schedule, as the best solution of the problem. If this is the last branch of the branching tree, or if the limit on the number of iterations and/or

computation time has reached, then proceed to the next step, otherwise go to step 2.

STEP 6: Print the results and stop.

The flow diagram of the branch and bound algorithm for a FSMP is presented in Figure 3.4. The algorithm, coded in FORTRAN, consists of three major parts; the branching tree generation, the lower bound computing, and the list processing part. The branching tree generation and the lower bound computation part use the algorithms developed earlier in this paper. Basically, the job and machine based bounds, with a slight modification to the procedures of computing the lower bound, are used for the computation of lower bounds. This modification in computing the lower bound arises due to the structure of the branching tree generation algorithm. In the branching tree generation algorithm, a square node on the branching tree indicates the end of use for the last processor and the start of processing of jobs on a new processor. So if this branch is to be followed, the remaining unscheduled jobs at this stage must be scheduled only on the leftover processors. This information makes the lower bound more effective since the processing time at stage j of the unscheduled jobs need only be divided by the number of remaining processors. Further, because of the depth first least lower bound search method used in the development of the computational algorithm, it is simple to keep track of all the jobs until that point of the branching tree. The added information makes it possible

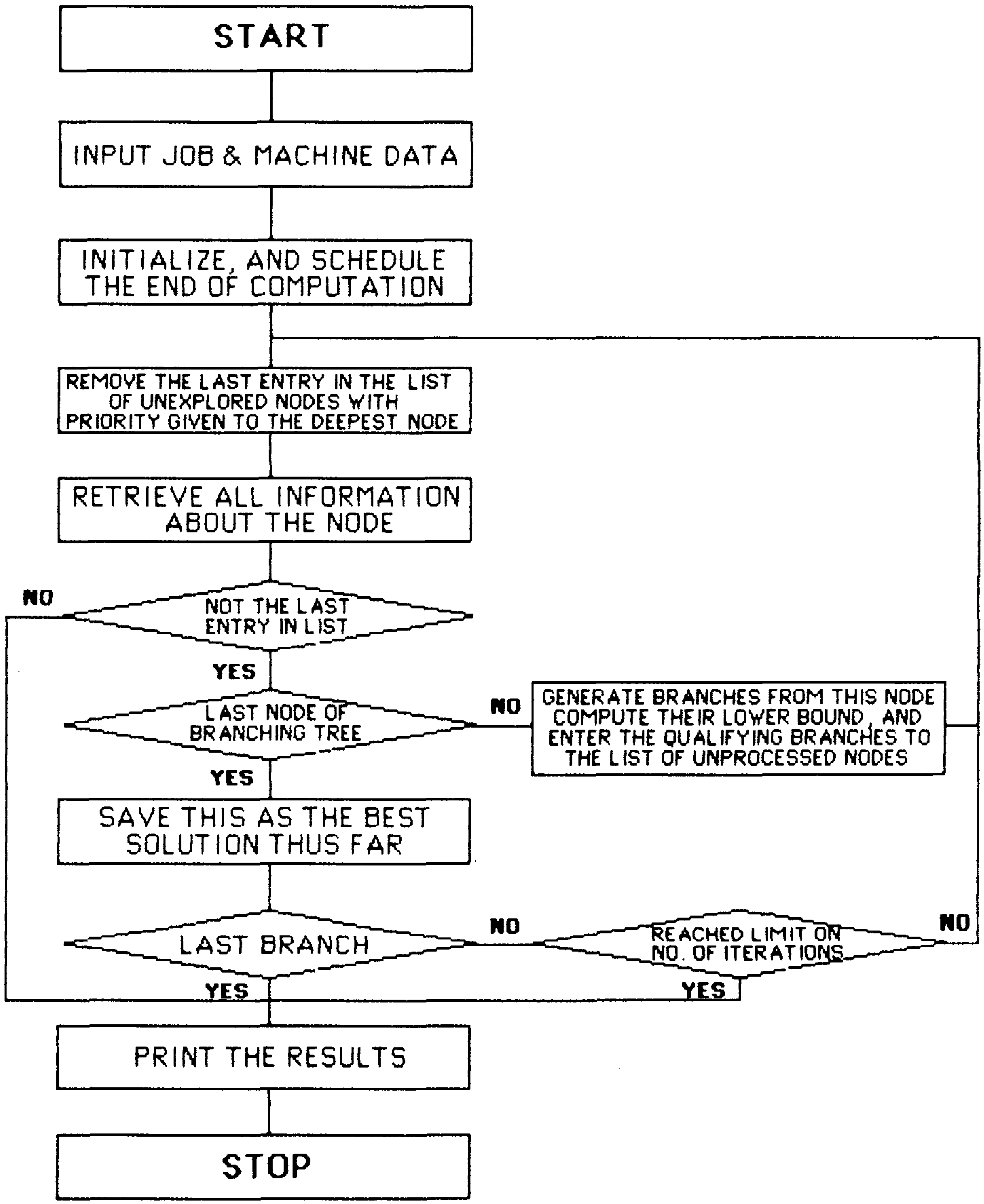


FIGURE 3.4 THE FLOW DIAGRAM FOR THE SOLUTION METHOD OF THE BRANCH AND BOUND ALGORITHM.

to search through a relatively small set of jobs for establishing the lower bound of the branching node. The third part of the algorithm is list processing of the nodes. For the list processing part, the information is first coded for each branching node. If the lower bound on this branch is better than the best available lower bound of a complete solution, provided it is available at the moment, the branching node is stored in the list of unprocessed nodes. The following is the information stored for each one of the branching nodes:

KODE = NPR x 1000000 + NPS x 10000 + LSN x 100 + JOB.

LBND = NS x 10000000 + NSCH x 100000 + LB.

where

JOB = The index of job.

NS = The index of stage.

NSCH = The number in processing sequence.

LB = Lower bound of the branching node.

NPR = The processor number in use.

NPS = Sequence Number on this processor.

LSN = Last square node, or the index of the first job on the processor used by this job.

The stage and the level numbers, are coded in the diametrically opposite manner to their position in the tree. This is arranged so that the deepest node in the search tree has the least value. The list processing part, with this coding method, stores the deepest node on top and therefore makes it available to be retrieved first. In case two or more nodes are at the same stage and level, the one with the least lower bound is retrieved first and processed. Once the node is retrieved, the information on the node is decoded and compared against the last processed node data. Now, if the node has gone down a step in the branching tree, the necessary information, like sequence position and completion time of the job on the retrieved node, is established and recorded. On the other hand, if the retrieved node is at a higher or the same level as of the previous node, the working sequence and completion time matrix of the nodes lower than the present level and upto the level of the last node are re-initialized. The lower bound is then compared against the best known lower bound, provided it is available, and is either eliminated or branched except when this is the last node of the branching tree. Now, if this is not one of the last node of the branching tree, then branches are generated using the tree generation algorithm. The qualifying nodes are stored in the list of unprocessed nodes following the deepest node with the least lower bound first rule. However, in case it is the last node of the branching tree, and if it satisfies the lower bound comparison test, the working sequence position and job completion time matrix along with the completion time of the schedule is saved as the best known solution.

9.1 TESTING OF THE ALGORITHM

A question most frequently asked in an optimization study, like the one performed over here, is concerning the validation of the algorithm. The authentication process of the branch and bound algorithm for a FSMP developed here consists of two parts. The first part consists of the proof that the branching algorithm generates all possible paths and that the bounding procedure does not eliminate an optimal end node of the branching tree. The proof of this component has been successfully demonstrated in earlier sections of this paper. The second part of the validation process consists of the correctness of the computer program developed to solve the problem through the use of a algorithm. It is indeed no secret that the proof of correctness of a computer program of any complicated algorithm, like the one developed here, is fairly difficult. However, in order to satisfy this requirement, the branching and bounding subroutines of the computer program were extensively tested for completeness and correctness. Furthermore, the results of the branch and bound algorithm for a FSMP were compared against a simple nondelay schedule generator of n! possible schedules. The optimal solution of the branch and bound algorithm tested successfully against the best solution of the n! nondelay schedules. Out of the fifty tests performed for comparative study, the branch and bound algorithm for a FSMP outperformed in twenty percent of the cases for the optimal makespan, and in all cases for the computation time.

9.2 AN EXAMPLE

Consider a two stage flow shop ($m = 2$) with two parallel processors at each stage of processing ($M_1 = M_2 = 2$). Further, let the processing time of each job i, at stage j of processing be given as in the processing time matrix of Table 3.1. The release time, and the travel time between stages is assumed to be zero. The problem at hand is that of scheduling four jobs ($n = 4$) in such a shop so as to minimize the maximum completion time.

TABLE 3.1 PROCESSING TIME DATA FOR THE EXAMPLE PROBLEM.

	JOBS				
	j \ i	1	2	3	4
STAGES	1	10	25	10	20
	2	20	20	30	10

The number of possible nodes at each stage j of a FSMP can be computed from equation (3.5) as follows,

$$N(n,j) = \binom{n-1}{M_j-1} \frac{n!}{M_j!} = \frac{(3!)(4!)}{(1!)(2!)(2!)} = 36.$$

Which gives the total number of possible nodes from equation (3.6) as,

$$S(n,m,M_j) = \prod_{j=1}^{m} \binom{n-1}{M_j-1} \frac{n!}{M_j!} = 36^2 = 1,296.$$

Now, if the interest was to generate a nondelay schedule, the problem has a feasible schedule (not generated by the algorithm), as presented in Figure 3.5, with a makespan of sixty time units. However, the optimal schedule, as presented in Figure 3.6, has a maximum completion time of fifty five time units.

9.3 RESULTS OF THE ALGORITHM

The branch and bound algorithm developed in this research, generates optimal schedules for the maximum completion time criteria. The algorithm explored only two end nodes out of the twelve hundred and ninety six possible nodes for the example problem. The CPU time on an IBM-XT for solving this problem is 0.69 seconds. Some other computation time data for various problem sizes is presented in Table 3.2. The processing time data for the study is generated from a uniform distribution between 0 and 100.

10.0 FURTHER EXTENSIONS

The computational algorithm developed in this research uses the bounding procedures to discard the nodes which are known to have a lower bound larger than a complete solution. Given the exponential nature of the problem, the algorithm

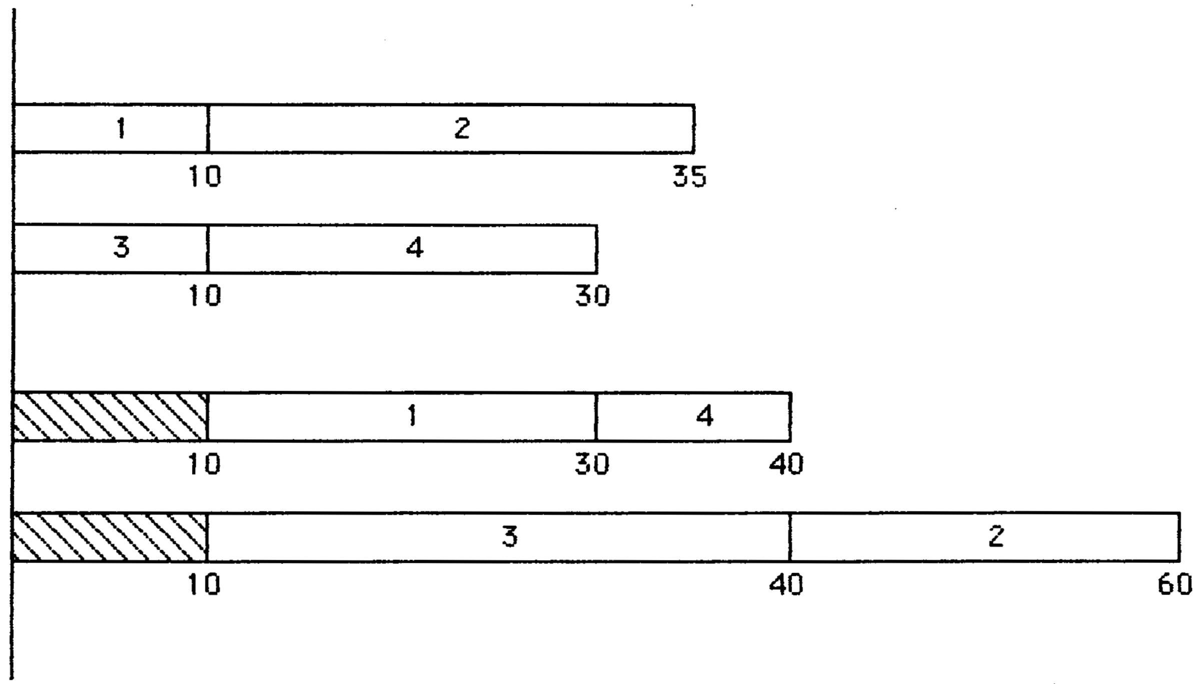


FIGURE 3.5 NONDELAY SCHEDULE FOR MINIMIZING MAKESPAN IN A FLOW SHOP WITH PARALLEL PROCESSORS.

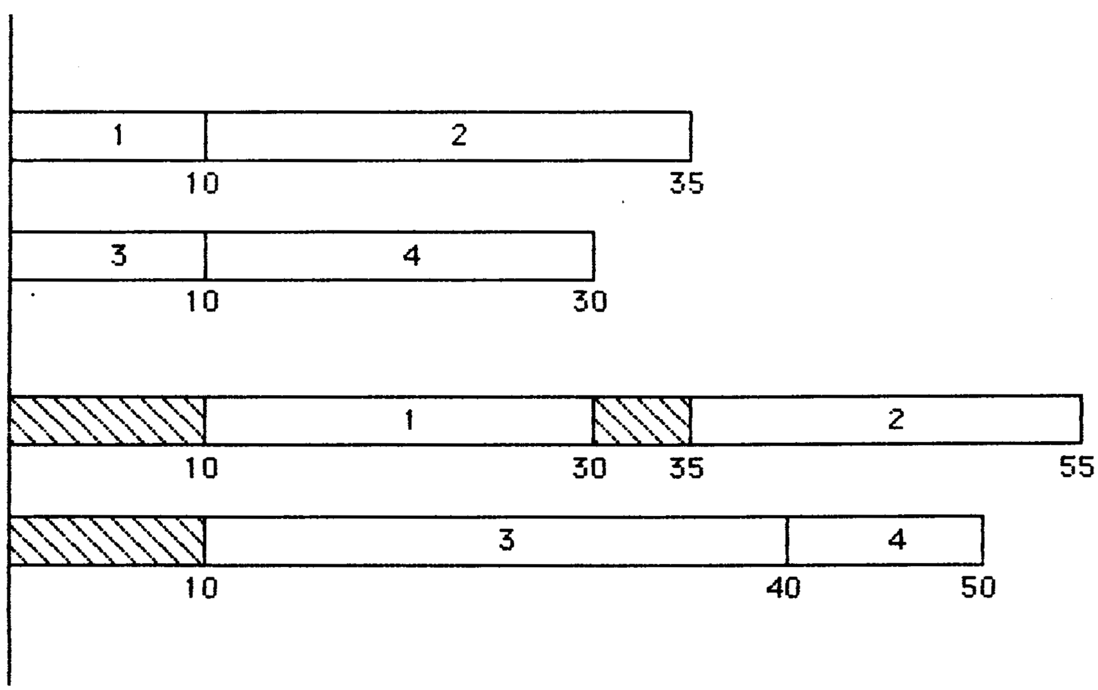


FIGURE 3.6 OPTIMAL SCHEDULE FOR MINIMIZING MAKESPAN IN A FLOW SHOP WITH PARALLEL PROCESSORS.

TABLE 3.2 COMPUTATION TIME RESULTS OF THE BRANCH AND BOUND ALGORITHM.

PROBLEM SIZE			SAMPLE SIZE	NUMBER OF POSSIBLE END NODES	AVERAGE COMP. TIME ON IBM-XT	AV. NO. OF END NODES SEARCHED
n	m	M_j, j=1,m.				
					HR:MN:SEC	
4	2	2,2	10	1.296×10^3	00:00:00.60	1.6
4	5	2,2,2,2,2	10	6.047×10^7	00:01:16.27	4.5
6	2	2,2	10	3.240×10^6	00:00:42.52	8.0
6	3	2,2,2	10	5.832×10^9	00:06:12.70	10.9
6	5	2,2,3,2,2	10	1.260×10^{16}	12:07:19.76	22.6
8	2	3,3	10	1.992×10^{10}	00:06:46.91	8.4

is observed to be consistently working with a fair amount of computation speed. Nevertheless, in order to improve the computation speed for large size problems, the elimination rules developed in this research can be used in conjunction with the lower bounds. For example, if jobs q and r follows an arrangement resembling the pattern b or c of Figure 3.7 as a part of a branching node of the tree at stage 1. Then due to Theorem 3.1, for any regular measure of performance a branching node which contains any one of the three patterns d, e, and f, will be eliminated from further consideration at stage 2. Similarly because of Corollary 3.2, for the makespan criteria, the elimination of nodes containing one of the patterns d, e, or f, will result at stage m if the branching tree at stage m-1 has a partial node resembling a pattern a, b, or c. In similar pursuit, Theorem 3.2 and other elimination rules developed here will further reduce the search tree.

The branch and bound algorithm developed here for optimizing the makespan of a FSMP can also be used to optimize other measures of performance. The only difference will be in computation of lower bounds of the branching nodes. Lower bounds for the measures of performance other than makespan, however, are not known to exist at this time and research is recommended in such direction.

Further efforts can be expanded for the development of useful heuristics, particularly for a combinatorial problem

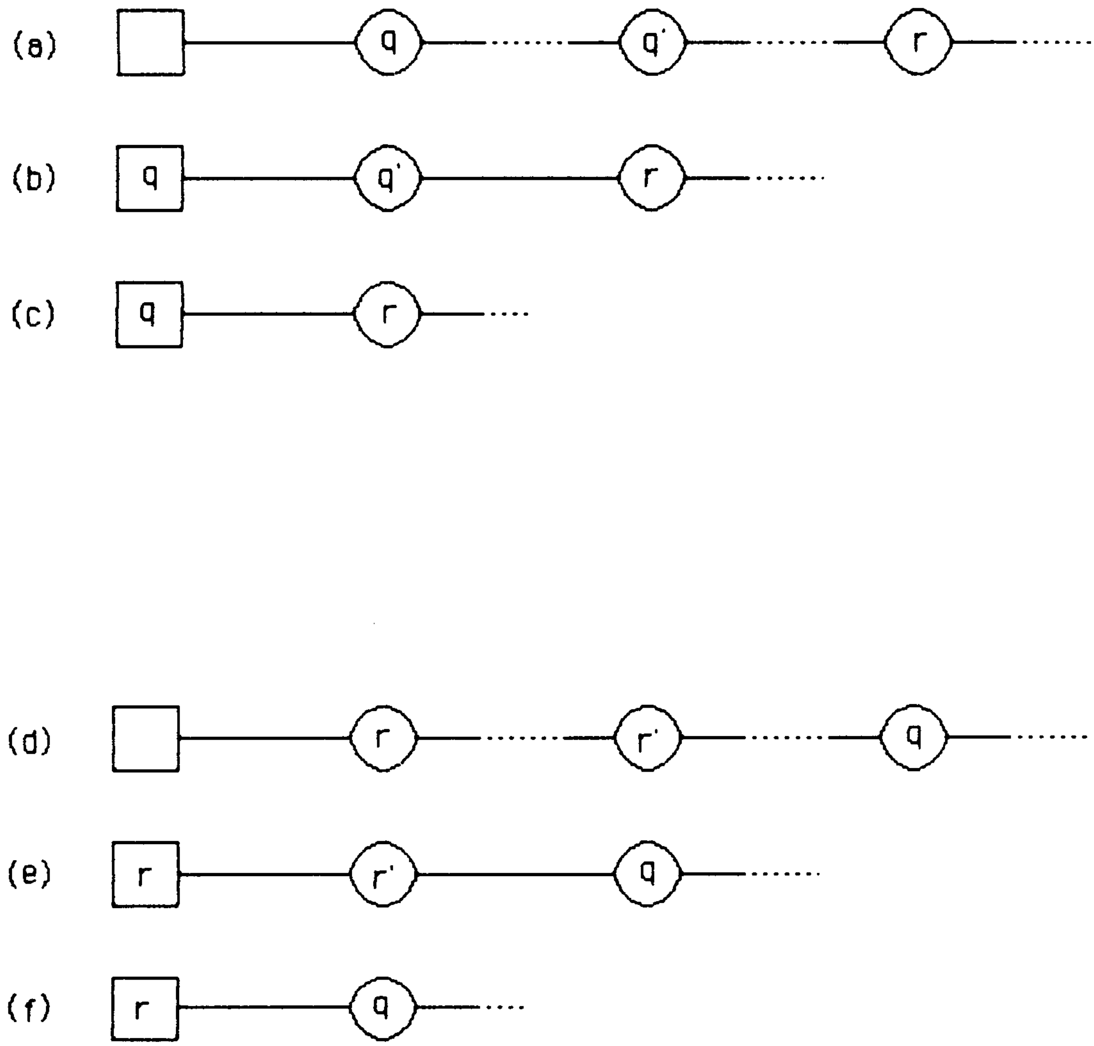


FIGURE 3.7 PARTIAL PATTERNS OF BRANCHING NODES.

like the one of a FSMP. To begin with, there are several variations of branch and bound algorithms which have been usefully employed in the literature. Some of these variations are discussed here and they can be used for an adaptation to the branch and bound algorithm for a FSMP.

- Set up a counter on the number of nodes (and/or end nodes) to be fully explored by the algorithm.
- Set up a percentage improvement index on each new feasible solution generated by the algorithm. This means that if the percentage improvement from one feasible solution to the other is less than that index, further exploration is stopped.
- A combination of the above two variations, etc.

The adaptation of such simple variations is expected to improve the computation speed of the branch and bound algorithm developed here for a FSMP. However, this increased speed will not come without a cost, which is the possibility of missing an optimal solution.

11.0 SUMMARY

The flow shop with multiple processors scheduling problem has been studied before by several researchers. The solution methodologies available in literature ranges from the mathematical formulation for the small size problems to heuristic algorithms for large size problems. This paper presents a branch and bound algorithm and solution method for the optimal solution of the makespan problem of a FSMP. The computational results of the algorithm are fairly encouraging for solving problems of medium size. Several extensions are also proposed for optimal or near optimal solution methods of large scale problems.

BIBLIOGRAPHY

[1] Ashour, S., "A Branch and Bound Algorithm for Flow Shop Scheduling Problems," AIIE Transactions, 2(2)(1970) 172-176.

[2] Baker, Kenneth R., "A Comparative Study of Flow Shop Algorithms," Operations Research, 23(1) (1975) 62-73.

[3] Bansal, S. P., "Minimizing the Sum of Completion Times of n Jobs Over m Machines in a Flow Shop - A Branch and Bound Approach," AIIE Transactions, 9(3) (1977) 306-311.

[4] Bansal, S. P., "On Lower Bounds in Permutation Flow Shop Problems," International Journal of Production Research, 17(4) (1979) 411-418.

[5] Brah, Shaukat A., and Hunsucker, John L.,"Optimal Scheduling in a Flow Shop with Multiple Processors," Paper Presented at the TIMS/ORSA Joint National Meeting in New Orleans, Louisiana, May 4-6, 1987.

[6] Bratley, Paul, Florian, Michael, and Robillard, Pierre, "Scheduling with Earliest Start and Due Date Constraints on Multiple Machines," Naval Research Logistics Quarterly, 22(1) (1975) 165-173.

[7] Cho, Yookun, and Sahni, Sartaj, "Preemptive Scheduling of Independent Jobs with Release and Due Times on Open, Flow, and Job Shops," Operations Research, 29(3) (1981) 511-522.

[8] Garey, M. R., and Johnson, D. S., Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman Press, San Francisco, 1979.

[9] Garey, M. R., Johnson, D. S., and Sethi, Ravi, "The Complexity of Flowshop and Jobshop Scheduling," Management Science, 1(2) (1976) 117-129.

[10] French, Simon, Sequencing and Scheduling: An Introduction to the Mathematics of Job-Shop, John Wiley & Sons, 1982.

[11] Gonzalez, Teofilo, and Sahni, Sartaj, "Flowshop and Jobshop Schedules: Complexity and Approximation," Operations Research, 26(1) (1978) 36-52.

[12] Gupta, J. N. D., "M-Stage Flowshop Scheduling by Branch and Bound," Opsearch, 7(1) (1970) 37-43.

[13] Gupta, J. N. D., "Optimal Schedules for Special Structure Flowshops," Naval Research Logistics Quarterly, 22(2) (1975) 255-269.

[14] Gupta, J. N. D., Shanthikumar, J. G., and Szwarc, W., "Generating Improved Dominance Conditions for the Flowshop Problem," Computers and Operations Research, 14(1) (1987) 41-45.

[15] Ignall, E., and Schrage, L. E., "Application of the Branch and Bound Technique to Some Flow-Shop Problems," Operations Research, 13(4) (1965) 400-412.

[16] King, J. R., "Scheduling and the Problem of Computational Complexity," Omega, The International Journal of Management Science, 7(3) (1979) 233-240.

[17] Kochlar, Sandeep, and Morris, Robert J. T., "Heuristic Methods for Flexible Flow Line Scheduling," Journal of Manufacturing Systems, 6(4) (1987) 299-314.

[18] Kohler, W. H., and Steiglitz, K., "Enumerative and Iterative Computational Approaches," In Computer and Job-Shop Scheduling Theory by E. G. Coffman, Jr. (ed.), John Wiley and Sons, New York, 1976.

[19] Lageweg, B. J., Lenstra, J. K., and Rinnooy Kan, A. H. G., "A General Building Scheme for the Permutation Flow-Shop Problem," Operations Research, 26(1) (1978) 53-67.

[20] Lomnicki, Z., "A Branch and Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem," Operations Research Quarterly, 16(1) (1965) 89-100.

[21] McMahon, G. B., and Burton, P.G., "Flow-Shop Scheduling With the Branch and Bound Method," Operations Research, 15(3) (1967) 473-481.

[22] Ow, Peng S., and Morton, Thomas E., "Filtered beam search in Scheduling," International Journal of Production Research, 26(1) (1988) 35-62.

[23] Salvador, M. S., "A Solution To a Special Case of Flow Shop Scheduling Problems," In Symposium of the Theory of Scheduling and Applications by S. E. Elmaghraby (ed.), Spring-Verlag, New York, 1973.

[24] Szwarc, Wlodzimierz, "Elimination Method in the m x n Sequencing Problem," Naval Research Logistics Quarterly, 18(3) (1971) 295-305.

[25] Szwarc, Wlodzimierz, "Special Cases of the Flow-Shop Problem," Naval Research Logistics Quarterly, 25(3) (1977) 483-492.

[26] Szwarc, Wlodzimierz, "Permutation Flow-Shop Theory Revisited," Naval Research Logistics Quarterly, 25(3) (1978) 557-570.

[27] Wittrock, Robert J., "Scheduling Algorithms for Flexible Flow Lines," IBM Journal of Research and Development, 29(4) (1985a) 401-412.

[28] Wittrock, Robert J., "An Adaptable Scheduling Algorithms for Flexible Flow Lines," Working Paper, IBM Thomas J. Watson Research Center, New York, 1985b.

CHAPTER IV

CONTRACTUAL EFFORT

IV. CONTRACTUAL EFFORT

The research work undertaken by our team has been generally on target with respect to the estimated timeline for the proposed study (Figure 4.1) given in the Statement of Work. The major part of the industrial investigation is industrial interviews. Work is in progress to schedule interviews in the next quarter. The work on flow shop scheduling and related heuristics has also been extended. Efforts are being made to identify scheduling criteria and solution methodologies for the space shuttle scheduling problem. Finally, the progress on the adaptation of industrial and theoretical techniques for consideration of the NSTS is also satisfactory. The principal investigator has made frequent presentations to address the major issues facing NASA on the future direction of the NSTS program.

Analysis work on the subject of Transition Management has been continued based on the results of the last three years of research efforts. Other analysis tools are also being investigated to provide input into the successful implementation of NSTS's transition management program.

We anticipate that the research work will continue to progress smoothly in the upcoming quarter, with all tasks being on schedule. As we enter the third quarter of the research grant work, the emphasis is on continuing the analysis and development of concepts and models that can be adapted to NASA's needs.

```
  0                                                                  12
  !                                                                   !
  !    1     2     3     4     5     6     7     8     9     10    11   !
  !----!-----!-----!-----!-----!-----!-----!-----!-----!-----!-----!-----!
  !                                                                   !
START-UP       QUARTERLY       QUARTERLY          QUARTERLY        FINAL
                REPORT          REPORT             REPORT          REPORT


  !---------- INDUSTRIAL INVESTIGATION --------!

  !---- STUDY OF FLOW SHOP SCHEDULING ----!
        FOR SPACE SHUTTLE APPLICATION

       !------------ LITERATURE SEARCHES -----------!

                           !-- ANALYSIS OF LIT. SEARCH --!
                               AND INTERVIEW PROCESS

                               !-ADAPTATION OF ANALYSIS-!
                                  TO THE NSTS PROGRAM

  !------ ADAPTATION OF INDUSTRIAL AND THEORETICAL ------!
              TECHNIQUES FOR CONSIDERATION OF NSTS

                                                               !----!
                                                               FINAL
                                                               REPORT
```

FIGURE 4.1. ESTIMATED TIMELINE FOR THE PROPOSED STUDY