

퍼지 환경하에서 품질수준 확보를 위한 일정계획에 관한 연구

-Bottleneck을 고려한 생산라인에서-

이상완

동아대학교 산업공학과

신대혁

마산전문대학 산업안전관리과

A Bottleneck-Based Production Scheduling
under Fuzzy Environment

Sang Wan Lee

Dept. of Industrial Engineering, Dong-A University

Dae Hyuk Shin

Dept. of Industrial Safety Management, Masan College

Abstract

Job shop scheduling problem is a complex system and an NP-hard problem. Thus it is natural to look for heuristic method. We consider the multi-part production scheduling problem for quality level in a job shop scheduling under the existence of alternative routings. The problem is more complex if the processing time is imprecision. It requires suitable method to deal with imprecision. Fuzzy set theory can be useful in modeling and solving scheduling problems with uncertain processing times. Li-Lee fuzzy number comparison method will be used to compare processing times that evaluated under fuzziness. This study presents heuristic method for quality level in bottleneck-based job shop scheduling under fuzzy environment. On the basis of the proposed method, an example is presented.

1. 서론

대량생산이 제조업의 지상과제였던 시절이 우리나라의 경우 불과 십여년전의 일이고 최근에는 품질좋은 제품을, 저가로, 납기에 맞추기 위해 가능한 빨리 만들어내어야 한다는 목표가 품질경영의 개념에 수용되어 발전하고 있다. 그러나 소비자가 가지는 제품에 대한 기호의 변화와 경제적 환경의 성숙으로 인한 제품선택 기준의 변화로 인해 기업에서는 기존의 대량생산 시스템을 다품종 소량생산 시스템으로 전환시켜야할 시점이 되었다.

대량생산 시스템은 부품이 정해진 가공기계를 따라 흘러가므로 해서 비교적 단순하지만 소량생산 시스템의 경우 주문에 따라 부품이 거치는 루트가 다양하기때문에 단순한 문제가 아니다.

그리고 부품의 공정별 가공시간이 사람이 담당하는 경우와 다르게 기계화로 인해 확정적이라고 볼 수 있지만 작업의 준비시간, 공정간의 이동시간, 예기치 않은 기계고장으로 인한 시간의 지연으로 인해 아직까지는 가공시간을 확정적으로 표현하기가 힘들어서 부품의 가공시간을 확률로 다루어 가공시간의 평균, 분산, 표준편차를 적용한 연구가 있었고 여기서 순서통계량 이론을 접목시킨 연구가 있었다.

또 어떤 부품을 만드는데 있어서 나타날 수 있는 隘路작업(bottleneck operation)이나 隘路기계(bottleneck machine)의 발생은 가공중인 부품에 불량을 야기시킬 수 있고 JIT 생산의 경우 생산라인을 중단시키기도 하는데 이 불량품은 생산라인의 주기를 파괴시키기 때문에 불량품이 생산원가에 미치는 영향은 과소평가할 수 없다.

흐름생산의 일정계획 문제에서 퍼지수를 도입한 연구는 있어왔으나 주문생산의 상황에 도입한 경우는 거의 보이지 않는다. 이는 주문생산이 가지는 엄청난 계산량에다가 퍼지의 계산량을 더할 경우 무한대의 시간이 주어진다 해도 해결하기 힘든 실정이기 때문이다.

따라서 본 연구에서는 애로공정을 가지는 부품을 생산하는 주문생산의 일정계획 문제를 diagraph로 표현하여 Gantt chart로 쉽게 표현하도록 하였으며, 원활한 생산라인을 유지하므로써 제품의 품질 수준을 확보하여 불필요한 품질비용의 발생을 막고, 확실성이 가지는 약점을 보완하기 위해 퍼지개념을 도입하므로써 가공시간의 불확실성을 정량화하여 빠른시간내에 실행가능해를 구하는 휴리스틱 기법을 개발하고자 한다.

2. 퍼지수 비교법

2.1 기호

\tilde{A} : 퍼지수 A

$u_{\tilde{A}}(x)$: \tilde{A} 내에서 x 에 대한 구성함수 값

S : \tilde{A} 의 지지(support) 집합

i : 부품

r : 각 부품의 대체경로 수

l : 각 부품의 작업(operation)수

\tilde{M} : 퍼지 makespan

$\left(\sum_{i=1}^n\right)$: 퍼지 합(summation)
 $\widetilde{\min}$: 퍼지 최소화

2.2 Li-Lee 법[1]

$$m(\tilde{A}) = \frac{\int_{S(\tilde{A})} x \cdot \mu_{\tilde{A}}(x) \cdot f(\tilde{A}) dx}{\int_{S(\tilde{A})} \mu_{\tilde{A}}(x) \cdot f(\tilde{A}) dx}$$

$$s(\tilde{A}) = \frac{\int_{S(\tilde{A})} x^2 \cdot \mu_{\tilde{A}}(x) \cdot f(\tilde{A}) dx}{\int_{S(\tilde{A})} \mu_{\tilde{A}}(x) \cdot f(\tilde{A}) dx} - [m(\tilde{A})]^2$$

$f(\tilde{A})$: \tilde{A} 의 확률밀도함수

삼각퍼지수일 경우 GMVs(Generalized Mean Value)는

$$m(\tilde{A}) = \frac{1}{3} (a + b + c)$$

같은 GMVs를 가지는 작업이 있을 경우

$$s(\tilde{A}) = \frac{1}{18} (a^2 + b^2 + c^2 - ab - ac - bc)$$

범위(spread), $s(\tilde{A})$ 는 같은 GMVs를 가지는 퍼지수에서 $s(\tilde{A})$ 가 더 작은 퍼지수를 작다고 판단한다.

3. 휴리스틱 알고리즘 개발

3.1 FOPSA (Fuzzified One Part Scheduling Algorithm)

단계 0) n : 부품 수, l : 각 부품의 작업(operation)수로 둔다.

단계 1) 각 부품이 가지는 대체경로를 $i_1, i_2, i_3, \dots, i_l$ 로 표현한다.

단계 2) 부품 i 에 대한 \tilde{T}_i 를 계산한다.

$$\tilde{T}_{i_r} = \binom{i}{r} \tilde{P}_{ij}$$

$$\tilde{T}_i = \widetilde{\min} \{ \tilde{T}_{i_r} \}$$

- *단계 3) \tilde{T}_i 값이 큰 순으로 정렬한다.
- *단계 4) 단계3에서 첫번째에 위치한 부품의 가장 짧은 대체경로를 선택한다.
- 단계 5) $q=2$
- 단계 6) 선택된 부품(들)에 선택되지 않은 모든 부품들을 하나씩 도입하여 \tilde{M} 을 구한다
{ FOMSA의 적용 }
- *단계 7) 각각의 \tilde{M} 중 가장 작은 값을 가지는 경로를 선택하여 그 때의 부품을 q 번째 도입
부품으로 정한다.
- 단계 8) 만약 $n=i$ 이면 단계7에서 선택된 경로와 \tilde{M} 을 임시해라 하고 단계 10으로 간다.
그렇지 않으면 $q=q+1$, 단계6로 간다.
- 단계 9) 단계4에서 순서를 도입순서로 하여 \tilde{M} 을 구한다.
- *단계10) 단계9에서의 \tilde{M} 과 임시해를 비교하여 더 작은 값을 가지는 경로와 \tilde{M} 을 최종해
로 정한다.

3.2 FOMSA (Fuzzified One Machine Sequencing Algorithm)

단계 0) U : 스케줄되지 않은 작업(operation)의 집합

t : 시점, $U : \{j/j=1, 2, \dots, n\}$

- *단계 1) U 중에 준비된 작업(operation) j 에서 시작시점이 가장 빠른 작업을 스케줄 한다.
(동률(tie)이 발생한 경우 임의로 한 방향을 선택한다)
- 단계 2) 자원충돌(conflict)이 발생하는 후속 모든 기계에서 단계1을 반복 적용해서 \tilde{M} 을
구한다.
- 단계 3) 단계1에서 동률이 발생했을 때 취하지 않은 나머지 방향에 대해서도 단계2를 적
용시켜 \tilde{M} 을 구한다.

여기서 *는 Li-Lee 퍼지수 비교법이 사용되는 단계이다.

4. 적용예

4.1 모형

본 연구에서 다루는 모형은 3대 기계, 4개 부품, 12개 작업(operation)으로 구성되어 있으며 이의 선후관계는 <Table 1>과 같다. 그리고 각 부품의 작업들이 각 기계에서 처리될 때의 퍼지수 가공시간이 <Table 2>에 있다.

〈 Table 1 〉 The precedence relation of an operation

part	route
I	(1) → (2) → (3)
II	(4) → (5) → (6) → (7)
III	(8) → (9) → (10)
IV	(11) → (12)

〈 Table 2 〉 The processing time of operations at each machine

Machine	operation											
	I			II				III			IV	
	1	2	3	4	5	6	7	8	9	10	11	12
M_1	(5,7,10)		(6,9,11)	(2,3,5)		(3,4,5)			(5,6,7)	(8,10,11)		
M_2		(3,4,7)			(6,8,9)	(5,6,7)	(4,5,7)	(2,4,5)			(3,4,5)	
M_3			(4,5,6)	(6,7,8)				(5,7,9)	(6,8,10)	(7,10,11)		

4.2 적용

상기의 모형으로 FOPSA에 적용시키면 아래와 같다.

단계 0-1) $n=4$, l 은 I에서는 3, II에서는 4, III에서는 3, IV에서는 2이다.

단계 2) 부품 i 에 대한 \hat{T}_i 의 결과는 〈Table 3〉과 같다.

$$\hat{T}_1 = 17, \hat{T}_2 = 20.33, \hat{T}_3 = 16.67, \hat{T}_4 = 13.33$$

단계 3) II → I → III → IV

단계 4) II₁ 선택

< Table 3 > The computational example of an each route by Step 2

alternative route	route	\tilde{T}_i	$m(\tilde{T}_i)$
I ₁	(1, 1) → (2, 2) → (3, 1)	(14, 20, 28)	20.67
I ₂	(1, 1) → (2, 2) → (3, 3)	(12, 16, 23)	17
II ₁	(4, 1) → (5, 2) → (6, 1) → (7, 2)	(15, 20, 26)	20.33
II ₂	(4, 1) → (5, 2) → (6, 2) → (7, 2)	(17, 22, 28)	22.33
II ₃	(4, 1) → (5, 2) → (6, 3) → (7, 2)	(18, 23, 29)	23.33
III ₁	(8, 2) → (9, 1) → (10, 1)	(15, 20, 23)	19.33
III ₂	(8, 2) → (9, 1) → (10, 3)	(12, 17, 21)	16.67
IV ₁	(11, 2) → (12, 3)	(10, 14, 16)	13.33
IV ₂	(11, 3) → (12, 3)	(13, 18, 21)	17.33

단계 5) $q=2$

단계 6) {FOMSA} 적용한 결과는 <Table 4>와 같고 이의 diagraph는 <Figure 1>, <Figure 2>와 같다.

< Table 4 > The computational example of an each route at $q=2$

introduced part	route	\tilde{M}	$m(\tilde{M})$
{II, I}	II ₁ + I ₁ [(4, 1) → (1, 1)]	(17, 24, 32)	24.3
	II ₁ + I ₁ [(1, 1) → (4, 1)]	(21, 29, 40)	30
	II ₁ + I ₂ [(4, 1) → (1, 1)]	(15, 20, 28)	21
	II ₁ + I ₂ [(1, 1) → (4, 1)]	(21, 28, 38)	29
{II, III}	II ₁ + III ₂	(22, 29, 35)	28.67
	II ₁ + III ₂	(15, 21, 26)	20.67
{II, IV}	II ₁ + IV ₁	(16, 21, 26)	21
	II ₁ + IV ₂	(15, 20, 26)	20.33

단계 7) IV를 두번째 도입 부품으로 결정

단계 8) 단계6으로 간다. $q=3$

단계 6) {FOMSA} 적용한 결과는 <Table 5>와 같고 이의 **diagraph**는 <Figure 3> <Figure 4>와 같다.

(Table 5) The computational example of an each route at $q=3$

introduced part	route	\tilde{M}	$m(\tilde{M})$
{ II, IV, I }	$II_1 + IV_2 + I_1 [(4, 1) \rightarrow (1, 1)]$	(17, 24, 32)	24.33
	$II_1 + IV_2 + I_1 [(1, 1) \rightarrow (4, 1)]$	(21, 29, 40)	30
	$II_1 + IV_2 + I_2 [(4, 1) \rightarrow (1, 1)]$	(17, 23, 27)	22.3
	$II_1 + IV_2 + I_2 [(1, 1) \rightarrow (4, 1)]$	(21, 28, 38)	29
{ II, IV, III }	$II_1 + IV_2 + III_1$	(22, 29, 35)	28.67
	$II_1 + IV_2 + III_2$	(18, 25, 30)	24.33

단계 7) I 을 세번째 도입 부품으로 결정

단계 8) 단계6으로 간다. $q=4$

단계 6) {FOMSA} 적용한 결과는 <Table 6>와 같고 이의 **diagraph**는 <Figure 5> <Figure 6>과 같다.

(Table 6) The computational example of an each route at $q=4$

introduced part	route	\tilde{M}	$m(\tilde{M})$
{ II, IV, I, III }	$II_1 + IV_2 + I_2 + III_1 [(4, 1) \rightarrow (1, 1)]$	(23, 30, 38)	30.33
	$II_1 + IV_2 + I_2 + III_1 [(1, 1) \rightarrow (4, 1)]$	(27, 35, 45)	35.67
	$II_1 + IV_2 + I_2 + III_2 [(4, 1) \rightarrow (1, 1)]$	(22, 30, 36)	29.33
	$II_1 + IV_2 + I_2 + III_2 [(1, 1) \rightarrow (4, 1)]$	(22, 30, 36)	29.33

단계 7) III 을 네번째 도입 부품으로 결정

단계 8) 경로 $II_1 \rightarrow IV_2 \rightarrow I_2 \rightarrow III_2$ 와 $\tilde{M}=(22, 30, 36)$ 를 임시해로 두고 단계9로 간다.

단계 9) 단계3의 순서에 따른 결과는 <Table 7>과 같고 이의 **diagraph**는 <Figure 7>과 같다.

단계10)경로 $II_1 + I_2 + III_2 + IV_1 [(11, 2) \rightarrow (8, 2)]$, $\tilde{M}=(21, 28, 37)$ 해로 정한다.

〈 Table 7 〉 The computational example of an each route by Step 9

introduced part	route	\tilde{M}	$m(\tilde{M})$
{II, I}	$\Pi_2 + I_1 [(4, 1) \rightarrow (1, 1)]$	(17, 24, 32)	24.3
	$\Pi_1 + I_1 [(1, 1) \rightarrow (4, 1)]$	(21, 29, 40)	30
	$\Pi_1 + I_2 [(4, 1) \rightarrow (1, 1)]$	(15, 20, 28)	21
	$\Pi_1 + I_2 [(1, 1) \rightarrow (4, 1)]$	(21, 28, 38)	29
{II, I, III}	$\Pi_2 + I_2 + III_1$	(23, 30, 38)	30.33
	$\Pi_1 + I_2 + III_2$	(20, 28, 36)	28
{II, I, III, IV}	$\Pi_1 + I_2 + III_2 + IV_1 [(8, 2) \rightarrow (11, 2)]$	(21, 30, 36)	29
	$\Pi_1 + I_2 + III_2 + IV_1 [(11, 2) \rightarrow (8, 2)]$	(21, 28, 37)	28.67
	$\Pi_1 + I_2 + III_2 + IV_2$	(22, 30, 36)	29.33

5. 결론

다품종 소량 생산 시스템의 일정계획 문제를 수학적 모형으로 나타낼 경우 NP-hard 문제의 형태를 가지게 되어 실시간 내에 해결하기에 어려움을 가지고 있는 현실로 인해 휴리스틱 기법이 많이 활용되고 있다.

그리고 가공시간과 같은 시스템의 시간치들이 준비시간, 공정간의 이동시간, 예기치 않은 기계고장으로 인한 시간지연으로 인해 확정적으로 보기에는 아직 어려움이 있다.

이외에도 어떤 부품을 제조하는 공정에서 발생할 수 있는 애로 작업이나 애로기계로 인해 발생할 수 있는 각 기계에서 사용되는 각종 자원의 상충(conflict)현상이 발생하여 제품의 품질 수준 확보에 문제를 제기할 수 있다.

본 연구는 이와같은 현상들을 모형화하고 이를 해결하는데 퍼지수 개념을 도입하여 생산라인을 운용하는데 필요로 하는 융통성을 부여함으로써 의사결정자가 의사결정을 하는데 있어서의 폭넓은 정보를 제시하였다.

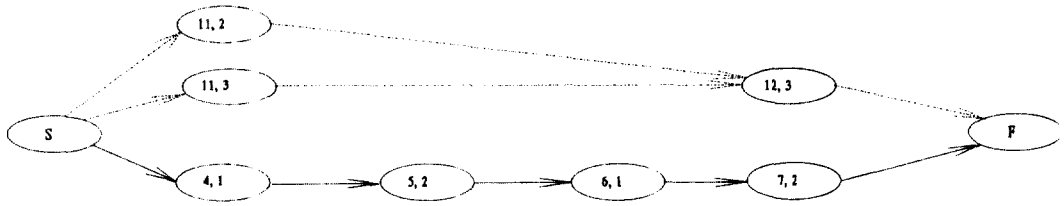
참고문헌

[1] Lee, E. S. and Li, R. J. (1988), "Comparison of Fuzzy Number based on the Probability Measure of Fuzzy Events," *Computers Math. Applic.*, Vol. 15, pp. 887-896.

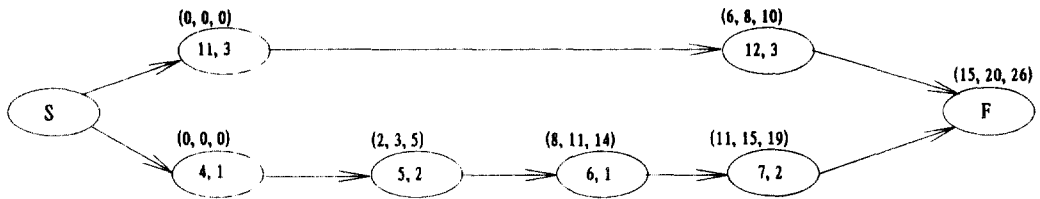
[2] Bell, C. E. and Park, K. (1990), "Solving Resource-Constrained Project

- Scheduling Problem by A* Search," *Naval Research Logistics*, Vol. 37, pp. 61-84.
- [3] Carlier, J. (1982), "The One-Machine Sequencing Problem," *European J. of O. R.*, Vol. 11, pp. 42-47.
- [4] Carlier, J. and Pinson, E. (1989), "An Algorithm for Solving the Job Shop Problem," *Management Science*, Vol. 35, No. 2, pp. 164-176.
- [5] Adams, J., Balas, E. and Zawack, D. (1988), "The Shifting Bottleneck Procedure for Job Shop Scheduling," *Management Science*, Vol. 34, No. 3, pp. 391-401.
- [6] Dauzere-Peres, S. and Lasserre, J. B. (1993), "A Modified Shifting Bottleneck Procedure for Job-Shop Scheduling," *Int. J. of Prod. Res.*, Vol. 31, No. 4, pp. 923-932.

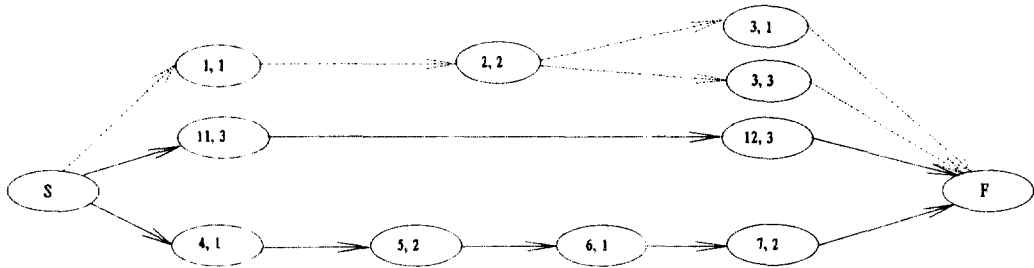
APPENDIX



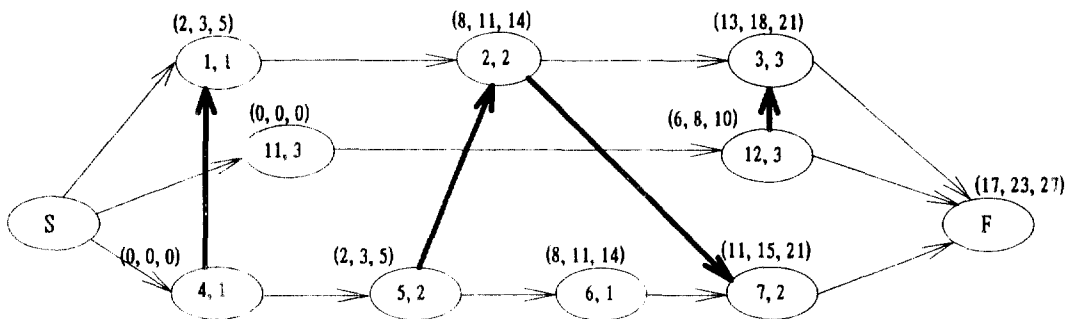
< Figure 1 > The network instance of the FOPSA { II, IV }



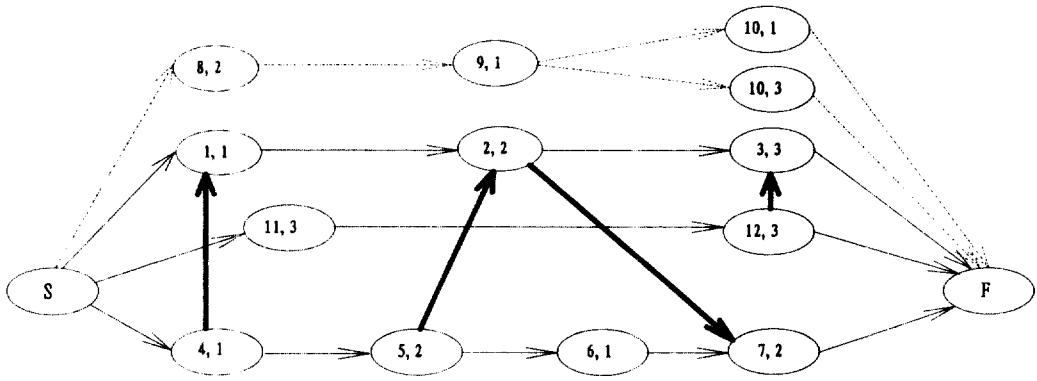
< Figure 2 > The network result of the FOPSA { II, IV }



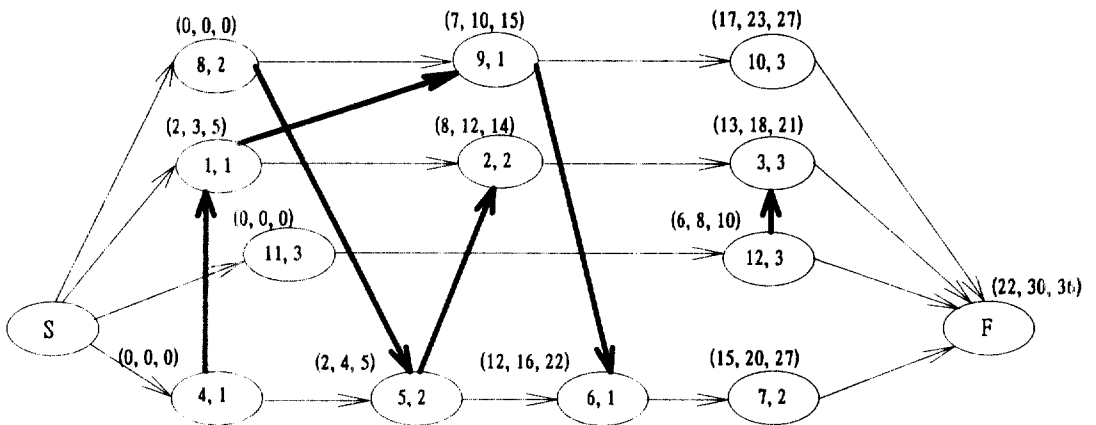
< Figure 3 > The network instance of the FOPSA { II, IV, I }



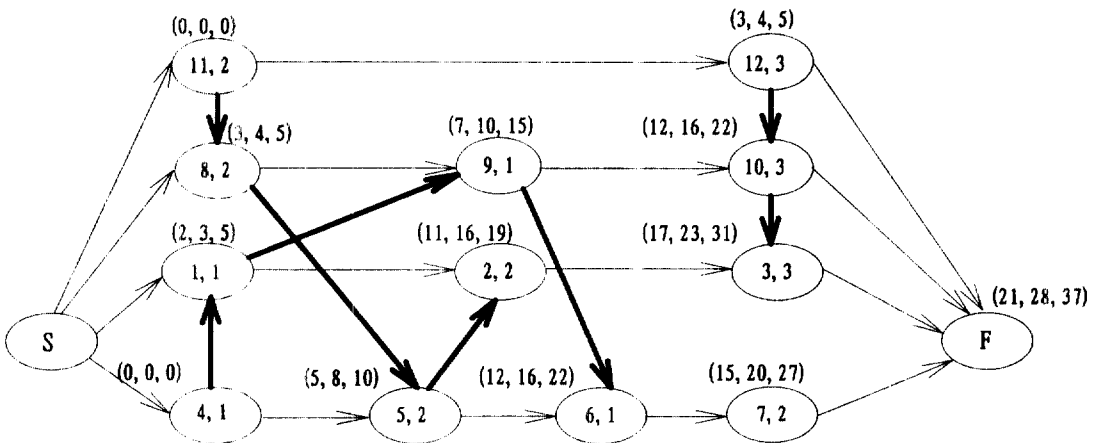
< Figure 4 > The network result of the FOPSA { II, IV, I }



< Figure 5 > The network instance of the FOPSA { II , IV , I , III }



< Figure 6 > The network result of the FOPSA { II , IV , I , III }



< Figure 7 > The network result of the FOPSA { II , I , III , IV }