

SSMS에서 동적 공구할당을 고려한 부품투입 알고리즘의 시뮬레이션 분석

Simulation Analysis of Part Release Algorithm under Dynamic Tool Allocation in SSMS

이충수*, 김성식**, 노형민*

Choong Soo Lee, Sung Shick Kim, Hyung Min Rho

Abstract

Recently in manufacturing environment, manufacturing order is characterized by unstable market demand, shorter product life cycle, a variety of product and shorter production lead time. In order to adapt this manufacturing order, flexible manufacturing system(FMS) in manufacturing technology advances into the direction that machines become further versatile functionally and that tools are controlled by fast tool delivery device. Unlike conventional FMS to mainly focus on part flow, it is important to control tool flow in single-stage multimachine systems(SSMS), consisting of versatile machines and fast tool delivery device. This research is motivated by the thought that exact estimation of tool competition at part release in SSMS enhances the system performance. In this paper, in SSMS under dynamic tool allocation strategy to share tools among machines, we propose real-time part release and tool allocation algorithms which can apply real factory and which can improve system performance.

* 한국과학기술연구원 CAD/CAM 연구센터

** 고려대학교 산업공학과

1. 서론

최근 제조회장은 다양한 소비자의 요구, 제품의 짧은 수명 주기, 치열한 국내외 경쟁 등으로 특징 지워진다. 이러한 제조회장에 적응하기 위하여 FMS(flexible manufacturing systems)에서 기계는 다기능 공작기계(versatile machine)로 구성되고, 공구는 빠른 공구 이동장치(tool delivery system)에 의하여 제어되는 추세에 있다. 1980년대 말부터 다양한 작업을 수행할 수 있는 다기능 공작기계의 개념이 도입되었다[2, 6, 10]. 이 다기능 공작기계는 다양한 작업을 수행할 수 있고, 빠른 공구 이동장치에 의한 지원을 받기 때문에 어떤 부품을 가공하기 위하여 매우 적은 셋업을 필요로 한다. 다기능 공작기계로 구성된 시스템은 병렬기계(parallel machines)[12, 13], 이산 제조 시스템(random manufacturing systems)[7], SSMS(single-stage multimachine system)[11] 등의 다른 이름으로 불리워졌다.

SSMS는 동일한 기능의 다기능 기계들로 구성되기 때문에 각 기계는 필요한 공구만 공급된다면 다양한 작업을 수행할 수 있다. 부품이 필요한 공구를 보유하고 있는 기계로 이동하면서 가공하던 기존 FMS와 달리, SSMS에서는 한 기계에서 필요한 공구를 공급 받아 부품의 가공을 완료할 수 있다. 따라서 SSMS에서는 기존 FMS에서 중요시되던 부품 흐름보다 신속한 공구 이동장치를 이용하여 공구 흐름을 제어하는 것이 중요한 문제이다. Kashyap과 Khator[9]는 FMS에서 총 운영비의 20~30%는 공구에 관련된 비용이라고 언급했다. Jones 등[8]은 효율적인 공구할당(tool allocation)에 의하여 공구재고를 40%까지 줄일 수 있다고 주장했다. 공구할당은 정적전략(static strategy)과 동적전략(dynamic strategy)으로 분류할 수 있다. 정적전략에서 생산계획 기간에 필요한 공구는 가공 전에 해당 기계에 할당되며, 할당된 공구는 그 기간 동안에 이동되지 않는다[3, 5, 14]. 이러한 정적전략들은 공구

의 제어가 용이한 반면, 높은 공구 재고가 소요된다. 정적전략을 채택했을 때, 다기능기계임에도 불구하고 필요한 공구가 없어 일부 작업을 수행할 수 없는 경우가 생길 수 있다. 동적전략에서는 생산계획 기간에 필요한 공구는 초기에 미리 할당되지 않고, 가공중에 필요할 때마다 할당된다. 또한 빠른 공구 이동장치를 이용하여 시스템내의 다른 기계가 보유하고 있는 공구를 공유함으로써 가상적으로 공구 매거진의 크기를 증가시킴으로써 정적전략보다 공구 재고를 줄일 수 있는 반면, 공구의 제어를 복잡하게 한다.

본 연구에서는 동적전략을 채택한 SSMS 형태의 실제 공장에 적용하여 시스템의 성능을 개선할 수 있는, 부품투입과 공구할당을 동시에 수행하는 알고리즘을 제안하고자 한다. 여기서 부품투입이란 기계에 부품을 할당한다는 의미이다. SSMS에서는 한 기계에서 부품의 가공을 완료하기 때문에 부품투입은 부품이 시스템에 도착했을 때 한 기계이상이 유휴상태인 경우 기계를 선택(machine selection)하는 상황과 한 부품의 가공을 완료했을 때 시스템 버퍼(buffer)에 한 부품 이상이 있는 경우 부품을 선택(part selection)하는 상황으로 분류된다.

SSMS에 동적전략을 사용하여 공구할당만을 다룬 기존 연구로써, Cantamessa 등[1]은 두 기계를 대상으로 작업장의 공구 수를 최소화하는 알고리즘을 제안하였다. 그들의 성능척도(performance measure)는 공구 유휴시간(tool idle time)과 공구 교환횟수를 최소로 하는 것이었다. Kashyap과 Khator[9]는 RSR(request selection rule)과 TSR(tool selection rule)의 효과를 분석하였다. 그들은 총작업 소요시간(makespan)을 성능척도로 시물레이션을 수행한 결과, RSR이 TSR보다 큰 영향을 준다는 사실을 입증하였다. Grieco 등[4]은 빠른 공구 이동장치를 이용하여 공구할당을 하여 시스템의 보유 공구 수를 줄이고도 줄이지 않았을 때와 유사한 성능을 유지할 수 있다는 것을 입증하

었다. 그들은 공구 유희시간을 성능척도로 이용하였다. 상기의 연구들은 부품투입(part release)에 관한 연구가 아닌, 공구할당만을 수행한 연구이기 때문에 공구 유희시간, 공구 교환횟수 등과 같은 국지적 성능척도(local performance)만을 개선하였다. 동적 공구할당 전략하의 SSMS에서 공구할당과 부품투입을 동시에 고려한다면 납기지연 부품수, 생산량 등과 같은 시스템의 전역적 성능척도(global performance)를 개선할 수 있을 것이다.

부품투입시 공구 경쟁(tool competition)의 정도를 고려하지 않는다면 같은 시기에 특정 공구를 사용하려는 과부하가 심해져 시스템의 성능을 저하시킬 수 있다. 부품투입에 공구 경쟁을 고려한 연구로써 Koo[11]는 다른 기계에서 가공중인 부품에 필요로 하는 공구와 투입하려는 후보 부품의 가공에 필요한 공구의 경쟁 정도를 표현한 TA(tool availability)를 계산하였다. Koo는 부품투입을 위하여 부품선택시, 납기 등 여러 요소를 고려하여 긴급도를 계산한 작업 우선순위 규칙 URG(highest urgency)와 TA를 조합하여 부품을 선택하였고, 기계선택시 적은 공구 이동횟수(tool movement frequency)가 계산된 기계를 선택하였다. Koo는 성능척도로 생산율(throughput rate)과 납기지연 부품(tardy part) 등을 이용하였다.

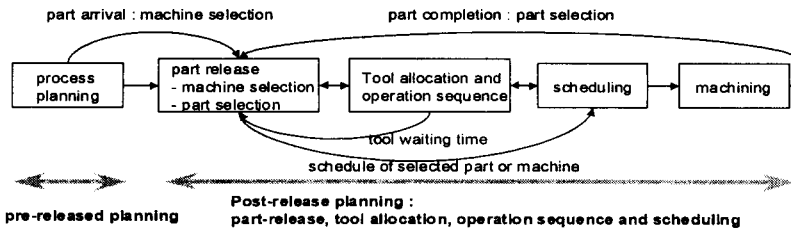
상기 기존 연구들은 공구할당과 부품투입의 두 측면에서 알고리즘을 개선할 필요가 있다. 첫째, 상기의 모든 연구들[1, 4, 9, 11]은 한 공구에 의하여 가공을 완료한 직후에 그 상황에서 선행관계를 만족하는 최적의 다음 공구를 할당하였다. 그러나 공구할당 시점을 공구의 가공완료 직후가 아닌 부품의 가공완료 직후로 변경한다면, 선행관계를 이용하여 부품의 가능한 여러 작업순서를 고려함으로써 공구의 과부하를 줄이는 공구할당을 수행할 수 있다. 그러므로 부품 전체의 가능한 작업순서 중에서 공구 과부하에 의한 공구 대기시간(tool waiting time)을 최소로 하는 동적 공구할당 알고리즘에 대한 연구가 필요하다. 둘째, Koo가

부품투입시 공구 경쟁을 고려하기 위하여 제안한 TA와 공구 이동횟수로는 다른 기계에서 가공중인 공구와 필요한 공구가 경쟁을 피할 수 있는지/없는지 또는 공구 대기시간이 얼마나 발생하는지를 정확히 파악할 수 없다. 따라서 동적 공구할당 전략하의 시스템에서는 공구 경쟁에 의한 공구 대기시간을 정확히 계산하여 반영하는 새로운 부품투입 규칙에 대한 연구가 필요하다.

본 연구에서는 동적 공구할당 전략하의 SSMS에서 실시간으로 다른 기계의 공구 일정계획을 참조함으로써 공구의 상태를 고려하여 공구 대기시간을 줄이는 방향으로 작업순서와 공구할당 알고리즘을 동시에 수행하고, 이 알고리즘에서 계산된 공구 대기시간을 이용한 실시간(real-time) 부품투입 알고리즘을 제안한다. 또한 본 연구에서 제안한 공구할당을 고려한 부품투입 알고리즘의 우수성을 시뮬레이션 실험을 통하여 입증하고자 한다. 실험의 목적은 공구 경쟁의 정도를 정확히 예측한 알고리즘이 시스템에서 같은 공구수를 유지하는 상태에서 다른 알고리즘보다 시스템 성능을 높일 수 있다는 것을 입증하는 것과 작업순서를 고려한 공구할당 알고리즘이 한 공구에 의하여 가공을 완료한 직후에 다음 공구를 선택하는 알고리즘보다 공구 대기시간을 줄일 수 있다는 것을 입증하는 것이다. 이를 위한 시뮬레이션 실험에서 전자의 성능척도는 납기지연 부품수와 생산량(throughput)을 이용했고, 후자의 성능척도는 공구 대기시간을 이용했다.

2. 부품투입 및 공구할당

본 연구의 시스템의 기능은 <그림 1>과 같이 전 처리계획(pre-release planning)과 후 처리계획(post-release planning)으로 구성된다. 전 처리계획은 부품이 시스템에 도착하기 전에 셋업, 작업, 공구, 가공시간 등을 결정하는 공정설계를 수행하고, 후 처리계획은 부품이



<그림 1> 부품투입 전 계획과 부품투입 후 계획

시스템에 도착한 후 작업순서, 공구할당, 부품 투입, 일정계획 등을 실시간으로 수행한다. 본 논문에서는 후 처리계획만을 다룬다.

기존 생산 시스템과 달리 SSMS 에서 부품투입시 정확한 공구 경쟁 즉 공구 대기시간을 고려하는 것이 매우 중요하다. 본 연구에서는 다른 기계의 일정계획을 참조하여 공구 대기시간을 최소화하는 방향으로 작업순서와 공구할당을 수행하고, 계산된 공구 대기시간을 부품투입에 이용하였다. SSMS에서는 한 기계에서 부품의 가공을 완료하기 때문에 부품투입 알고리즘을 수행하는 시점은 부품이 시스템에 도착했을 때 한 기계이상인 유휴상태인 경우인 기계선택과 한 부품의 가공을 완료했을 때 시스템 버퍼에 한 부품 이상이 있는 경우인 부품선택이다. 이 두 경우에 후보 부품 또는 후보 기계에 대하여 공구 대기시간을 최소화 하는 공구할당, 작업순서를 동시에 수립한다. 그 후, 공구 대기시간을 고려한 부품투입 알고리즘에 의하여 부품 또는 기계를 선택한 후, 해당 기계의 일정계획을 수립한다. 이때, 각 기계별로 한 부품의 일정계획만을 수립하며, 수립된 부품의 가공이 완료된 직후 다음 선택된 부품의 일정계획을 수립한다. 부품투입과 공구할당 알고리즘을 설명하기 위하여 필요한 기호는 다음과 같다.

<notation>

- p_{ij} : processing time for machining operation j of part i ($p_i = \sum_j p_{ij}$)
- tw_{ij} : tool waiting time of the required tool for operation j of part i ($tw_i = \sum_j tw_{ij}$)
- ttw_{ij} : tool transporter waiting time of the required

- tool t for operation j of part i
- ct_i : the time at which the required tool t is completed on a machine
- t_{ijk} : the time at which the operation j of part i is completed on current machine k
- mt_{ijk} : tool moving time from another machine to current machine k
- S_i : slack time of part i
- SS : set of parts which has small slack time
- O_i : set of operations required for part i
- D_i : due-date of part i
- $tnow$: current decision time
- LM : the limit of movement toward the left side in operation sequence
- RM : the limit of movement toward the right side in operation sequence
- FPOS : operation set that can be placed on the first position of operation sequence
- STOS : operation set that use same tool as previous operation and that satisfy LM and RM
- WOS : operation set generating tool waiting time except for tool changing time

2.1 부품투입

알고리즘에서 부품선택의 경우, 해당 기계에 대하여 가장 적은 여유시간(slack time) $S_i = \{(D_i - tnow) - (p_i + tw_i)\}$ 을 갖는 부품을 선택한다. 기계선택의 경우, 해당 부품에 대하여 가장 적은 공구 대기시간 tw_i 를 제공하는 기계를 선택한다. 이 부품투입 알고리즘은 다음과 같다.

Step 1. 부품투입 상황이 부품선택이면 Step 2 로 이동. 기계선택이면 Step 3 으로 이동.

Step 2. 부품선택

- (1) 부품을 여유시간 $\{(D_i - t_{now}) - p_{ij}\}$ 이 적은 순서로 나열한 후 SS 를 생성
- (2) SS 의 각 부품에 대하여 공구할당 알고리즘을 수행하여 tw_i 계산 (2.2 절)
- (3) SS 중에서 가장 적은 S_i 를 갖는 부품을 선택한 후, Step 4 로 이동

Step 3. 기계선택

- (1) 모든 유휴기계에 대하여 공구할당 알고리즘을 수행하여 tw_i 를 계산(2.2 절)
- (2) 가장 적은 tw_i 를 제공하는 기계를 선택, Step 4 로 이동

Step 4. 매핑된 부품-기계의 일정계획 수립

2.2 공구할당

어떤 부품의 가공을 위하여 필요한 공구가 해당 기계에 없을 경우, 다른 기계로부터 공구를 가져온다. 이때 그 공구를 사용중이어서 즉시 가져올 수 없는 경우, 공구 대기시간이 일어난다. 이 경우, 선행관계를 만족하는 범위내에서 그 작업의 순서를 바꿈으로써 공구 대기시간을 줄일 수 있다. 이 알고리즘을 수행하는 절차는 다음과 같다.

Phase I. LM 과 RM 에 의하여 변동 가능한 작업순서를 결정

Step 1. 임시 작업순서 결정

- (1) $P(i, j)$ 생성
- (2) $\text{sum}(j) = \sum_i P(i, j)$ 계산
- (3) $\text{sum}(j) = 0$ 인 작업을 작업순서에 추가한 후, $P(i, j)$ 에서 그 작업을 제거
- (4) $P(I, I)$ 이면 Step 2 로 이동, 그렇지 않으면 Step 1 의 (2)로 이동

Step 2. 각 작업에 대하여 LM 과 RM 을 계산, Phase II 로 이동

Phase II. 공구 대기시간을 최소화하는 공구할

당 및 작업순서 결정

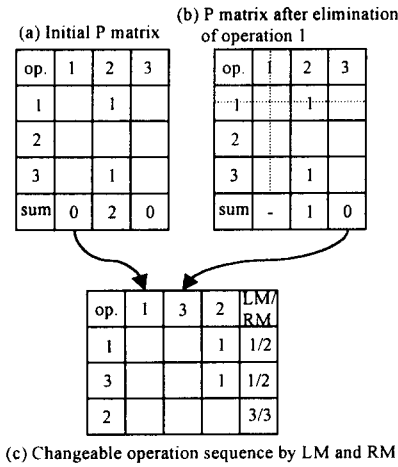
Step 1. 공구할당과 작업순서 결정

- (1) FPOS 생성
- (2) FPOS 에서 한 작업을 선택한 후 (3)으로 이동. FPOS 에 더 이상 선택할 작업이 없으면 Phase II 의 Step 2 로 이동
- (3) 선택된 작업을 작업순서에 고정시킨 후, LM 과 RM 을 변경
- (4) STOS 를 생성, STOS 가 생성되면 (5)로 이동, 그렇지 않으면 (6)으로 이동
- (5) STOS 에 유휴한 공구를 사용하는 작업이 있으면 작업순서에 고정. STOS 에 유휴한 공구를 사용하는 없으면 LM 과 RM 을 만족하는 나머지 작업중 가장 적은 tw_{ij} 를 갖는 작업 j 를 작업순서에 고정. LM 과 RM 을 변경.(7)로 이동
- (6) LM 과 RM 을 만족하는 나머지 작업중 가장 적은 tw_{ij} 가 계산되는 공구를 갖는 작업 j 를 작업순서에 고정시킨 후, LM 과 RM 을 변경
- (7) 작업순서에 고정하지 않은 작업이 하나 남았다면 현재의 작업순서(cur_seq), 일정계획(cur_sch), 공구 이동장치의 일정계획(cur_tt_sch)의 tw_i 계산한 후 (8)로 이동. 그렇지 않으면 (4)로 이동
- (8) tw_i 가 현재 최적 공구 대기시간(tw^*)보다 적으면 $tw^* = tw_i$, $seq^* = cur_seq$, $sch^* = cur_sch$, $tt_sch^* = cur_tt_sch$ 로 변경한 후, (2)로 이동

Step 2. tw_i 를 최소화하는 단계

- (1) WOS 생성
- (2) WOS 를 생성하면 (3)으로 이동. 그렇지 않으면 종료
- (3) WOS 에서 한 작업을 선택
- (4) 그 작업의 mp(moving position) = LM
- (5) 그 작업을 mp 로 이동한 후, tw_i 계산
- (6) tw_i 가 tw^* 보다 적다면 $tw^* = tw_i$, $seq^* = cur_seq$, $sch^* = cur_sch$, $tt_sch^* = cur_tt_sch$
- (7) mp = RM 이면 (8)로 이동. 그렇지 않으면

- mp = mp+1 로 변경 후, (5)로 이동
 (8) 작업이 WOS 에서 마지막이면 종료. 그렇지 않으면 (3)으로 이동



<그림 2> 선행관계 행렬에서 LM 과 RM 에 의해 변경 가능한 작업순서 결정

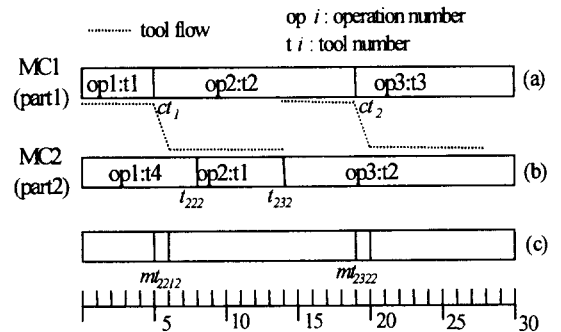
설계정보의 기하형상과 공차에 의하여 작업간에는 반드시 만족되어야 하는 선행관계가 생성된다. Phase I 에서 선행관계를 만족하는 작업순서를 결정하기 위하여 <그림 2>의 (a)와 같은 선행관계 행렬(P matrix: precedence matrix) P(i, j)를 이용했다. <그림 2> (a)의 초기 선행관계 행렬에서 P(1, 2)=1로 표시되어 있다. 이것은 작업 1(op1)이 작업 2(op2)보다 먼저 가공되어야 한다는 것을 의미한다. 이 초기 선행관계 행렬에서 열 j 의 선행 작업수의 합 sum(j)를 계산하여 0인 작업을 <그림 2> (c)의 작업순서에 고정시킨다. sum(j)=0는 작업 j보다 먼저 가공해야 할 작업이 없다는 의미이다. <그림 2>의 (a)에서 sum(j)가 0인 작업은 작업 1과 작업 3이며, 여기서는 작업 1을 임의로 작업순서에 고정시켰다. <그림 2>의 (b)에서는 작업 1을 제거한 후, sum(j)를 다시 계산하여 나머지 작업들을 작업순서에 고정시켰다. 어떤 작업 j의 LM(left movement)은 작업순서에서

왼쪽에서의 이동 허용범위를 의미하고, RM(right movement)은 오른쪽에서의 이동 허용범위를 의미한다[15,16]. <그림 2>의 (c)에는 LM 과 RM 에 의하여 변동 가능한 작업순서가 나타나 있다. 여기서 작업 1과 작업 3의 LM 과 RM 은 1과 2이다. 따라서 작업 1과 3은 작업순서에서 첫번째 또는 두번째에 위치할 수 있다. Phase II 에서는 <그림 2>의 (c)에서 결정된 LM 과 RM 사이에서 변동 가능한 여러 작업순서에서 다른 기계의 일정계획을 참조하여 공구 상태, 공구 이동시간, 공구 이동장치 대기시간 등을 고려하여 공구 대기시간을 최소로 하는 공구할당과 작업순서를 결정한다. <그림 2>의 예에서는 다른 기계의 일정계획을 참조하여 가능한 작업순서 op1 → op3 → op2 과 op3 → op1 → op2 중, 공구 대기시간을 적게 일으키는 작업순서를 결정한다.

공구 대기시간 tw_{ij} 는 공구 경쟁에 의한 대기시간($ct_i - t_{ijk}$), 공구 이동시간(mt_{ijk}), 공구 이동장치의 대기시간(ttw_{ij})으로 구성된다. 공구할당 시점에 필요한 공구가 다른 기계에서 사용중이라면 공구 대기시간은 0 이상일 것이다. 그러나 공구할당시점 전부터 유희해서 미리 이동했다면 공구 대기시간은 0일 것이다. 이 공구 대기시간의 계산은 다음과 같다.

$$tw_{ij} = \sum_t \sum_k \{ \max(0, ct_i - t_{ijk} + mt_{ijk} + ttw_{ji}) U_{ijk} \} \text{ for all } i, j \in O_i$$

$$U_{ijk} = \begin{cases} 1 & \text{if operation } j \text{ of part } i \text{ is processed with tool } t \text{ on machine } k \\ 0 & \text{otherwise} \end{cases}$$



<그림 3> 공구 대기시간의 계산

예를 들어 <그림 3>의 (a)와 (b)같이 기계 1 과 2 에 다른 부품이 스케줄된 경우, 공구 이동장치의 대기시간은 0 라고 가정했을 때, <그림 3> 의 (c)처럼 공구 t1 과 t2 에서 공구 이동시간(mt_{2212} , mt_{2322})이 발생했음에도 불구하고 부품 2 의 작업 2(op2)에 필요한 공구 t1 의 공구 대기시간은 0 이고, 작업 3 에 필요한 공구 t2 의 공구 대기시간은 6 이다. 이의 계산식은 아래와 같다.

$$tw_{22} = \max(0, ct_1 - t_{222} + mt_{2212} + ttw_{221})$$

$$= \max(0, 5 - 8 + 1 + 0) = 0$$

$$tw_{23} = \max(0, ct_2 - t_{232} + mt_{2322} + ttw_{222})$$

$$= \max(0, 19 - 14 + 1 + 0) = 6$$

위와 같은 작업순서 및 공구할당 결정문제는 NP 난도문제(NP-hard problem)이다. 본 연구에서는 선행관계를 만족하면서 공구 경쟁을 줄이는 경험적 알고리즘으로 NP 난도문제를 풀었으며, 이 알고리즘으로부터 도출된 결과는 근사 최적해이다.

3. 시뮬레이션 실험

2 절에서 제안한 부품투입과 공구할당을 동시에 수행하는 알고리즘의 성능척도에 큰 영향을 주는 요인을 분석하기 위하여 2^k 요인 실험(2^k factorial design)을 도입하였다. 제안된 부품투입 알고리즘의 우수성을 입증하기 위하여 성능척도에 큰 영향을 주는 요인을 변화시키면서 다른 알고리즘과 비교하였다. 또한 부품투입을 고려하지 않은 공구할당 알고리즘만을 다른 공구할당 알고리즘과 비교하였다. 시뮬레이션 언어는 AweSim 과 C++을 이용하였다.

3.1 공구할당을 고려한 부품투입 알고리즘의 분석

3.1.1 2^k 요인실험

이 시뮬레이션 실험의 성능척도는 납기지

연 부품수를 최소화하는 것과 생산량을 최대화하는 것이다. SSMS 에서는 공구흐름을 적절히 제어하면서 많은 종류의 부품을 생산하기 때문에 공구흐름과 부품흐름의 속도에 관련된 평균 공구 중복율(tool duplication rate), 부품 종류수, 부품의 평균 작업수, 공구 이동장치수, 부품 이동장치수, 부품의 도착 간격 등이 성능 척도에 큰 영향을 줄 수 있는 요인일 것으로 생각했다. 이 중 부품 이동장치수는 기계간 이동을 하지 않고 시스템 버퍼와 기계사이만을 이동하므로 과부하가 걸리지 않을 것으로 예상되어 실험에서 제외하였다. 부품 도착 간격은 시스템의 성능척도에 큰 영향을 주나, 도착 간격이 작으면 시스템의 성능을 저하시키고 크면 성능이 좋아질 것으로 예상되어 실험에서 제외시켰다. 공구 중복율은 시스템에 있는 각 공구 종류별 공구수를 기계수로 나눈 수치이다. 실험은 기계수는 4 대, 공구 종류수 25 개로 이루어졌다. 이때, 평균 공구 중복율 0.5 는 각 공구 종류가 시스템에 2 개정도 존재한다는 것을 의미한다. 이 실험에서 공구 종류수, 부품 종류수, 평균 작업수는 실제 현장의 1/5 정도로 축소하였다. 2^k 요인실험을 위하여 <그림 4>과 같이 각 요인의 고수준과 저수준을 정의했다. <그림 5>는 각 실험점(design point)에서 요인의 조건을 나타내고 있으며 R_i는 성능 척도를 나타낸다.

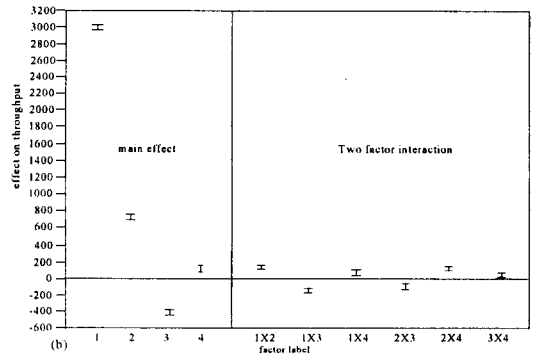
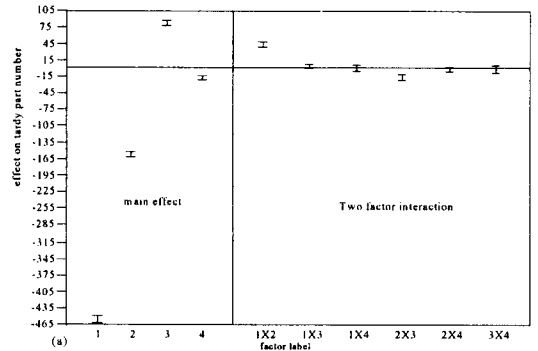
factor	factor name	- (low level)	+ (high level)
1	tool duplication rate	U(0.25, 0.4)	U(0.75, 0.9)
2	part type number	U(4, 5)	U(12, 15)
3	operation number/part	U(5, 7)	U(15, 20)
4	tool transporter number	1	2

<그림 4> 요인의 수준 정의

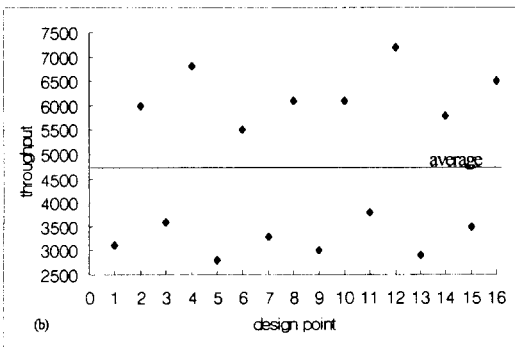
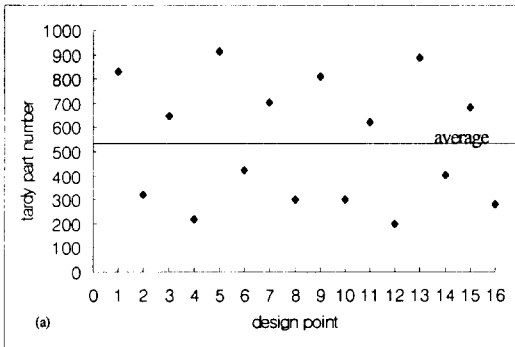
<그림 6>의 (a)는 5000 개의 부품을 완성할 때 측정된 납기지연 부품수이며, <그림 6>의 (b)는 400,000 분을 시뮬레이션하여 측정된 생산량이다. 시뮬레이션 초기에 각 기계는 비어있는 상태에서 시작하기 때문에 처음 10000 분의 통계량은 버렸다.

design point	factor				Responce
	1	2	3	4	
1	-	-	-	-	R ₁
2	+	-	-	-	R ₂
3	-	+	-	-	R ₃
4	+	+	-	-	R ₄
5	-	-	+	-	R ₅
6	+	-	+	-	R ₆
7	-	+	+	-	R ₇
8	+	+	+	-	R ₈
9	-	-	-	+	R ₉
10	+	-	-	+	R ₁₀
11	-	+	-	+	R ₁₁
12	+	+	-	+	R ₁₂
13	-	-	+	+	R ₁₃
14	+	-	+	+	R ₁₄
15	-	+	+	+	R ₁₅
16	+	+	+	+	R ₁₆

<그림 5> 2⁴ 요인실험에서 실험점의 요인 조건



<그림 7> 납기지연 부품수(a)와 생산량(b) 대한 주효과와 2원 상호작용효과



<그림 6> 실험점들의 납기지연 부품수(a)와 생산량(b)

<그림 6>의 (a)와 (b)에서 각 실험점의 성능척도는 10 번의 시뮬레이션을 수행한 평균이다. <그림 7>의 (a)와 (b)에는 납기지연 부품수와 생산량에 대한 요인의 주효과와 요인간의 2원 상호작용효과를 계산하여 정리하였다. 3 원이상의 상호작용효과를 계산하였으나 0에 근사했기 때문에 표시하지 않았다. 실제로 <그림 7>의 (a)와 (b)에서 2원 상호작용 효과는 주효과와 비교할 때 무시할 정도임을 알 수 있다. 이로부터 시뮬레이션 실험에 가장 큰 영향을 주는 요인은 평균 공구 중복율이고, 그 다음으로 부품 종류수, 평균 작업수, 공구 이동장치수임을 알 수 있다. 그리고 최적의 실험 조건은 요인 1, 2, 4가 고수준이고, 요인 3이 저수준인 경우임을 알 수 있다.

3.1.2 다른 부품투입 알고리즘과 시뮬레이션 비교 분석

실험의 목적은 공구 경쟁을 정확히 예측한 부품투입 알고리즘이 시스템에서 같은 공구수를 유지하는 상태에서 다른 알고리즘보다 시스템 성능을 높일 수 있다는 것을 입증하는 것이다. 이를 위하여 제안된 부품투입 알고리즘의 성능척도를 다른 알고리즘들의 성능척도와 비교하였다. 비교대상은 부품투입시 공구 경쟁을 전혀 고려하지 않은 알고리즘(I)과 TA와 공구 이동횟수를 고려한 알고리즘(II)이다. 이 알고리즘들은 다음과 같다.

알고리즘 I: 부품투입시 공구 경쟁을 고려하지 않고 EDD로 기계에 부품을 할당한 후, 다음 단계에서 그 부품의 가공을 위하여 필요한 공구를 할당하는 알고리즘이다. 공구할당 알고리즘은 작업순서를 고려하지 않고 한 공구를 사용하여 가공을 완료한 직후에 선행관계에 의하여 가능한 공구 중 적절한 다음 공구를 선택한다.

알고리즘 II: 부품선택시, EDD와 TA를 조합하여 부품을 선택하고, 기계선택시 적은 공구 이동횟수가 계산된 기계를 선택하여 부품투입을 수행하는 알고리즘이다. 이 알고리즘은 Koo[11]의 부품투입 알고리즘과 유사한 것이다. 이 알고리즘에서는 부품선택시 Koo의 URG/TA 규칙을 EDD/TA 규칙으로 수정하여 부품을 선택하였고, 기계선택시 공구 이동횟수가 적은 기계를 선택하였다. 여기서 TA는 다음의 식으로 계산된다.

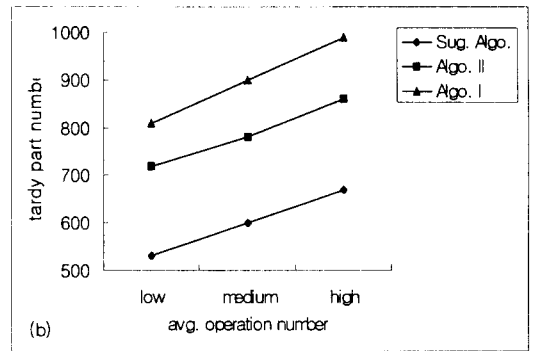
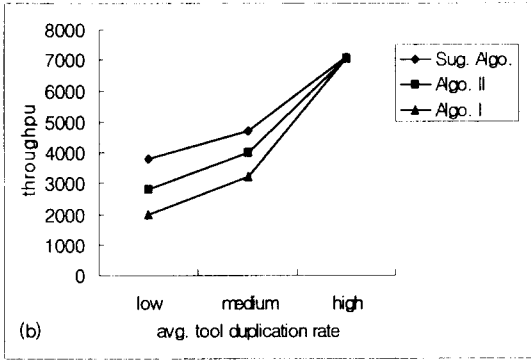
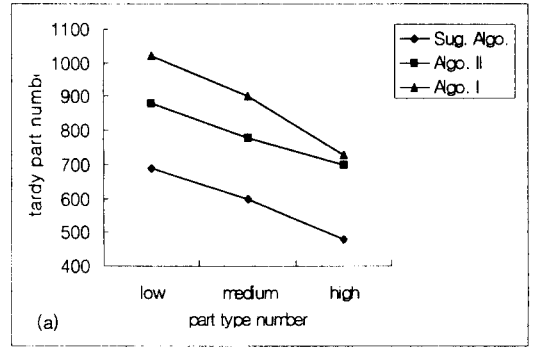
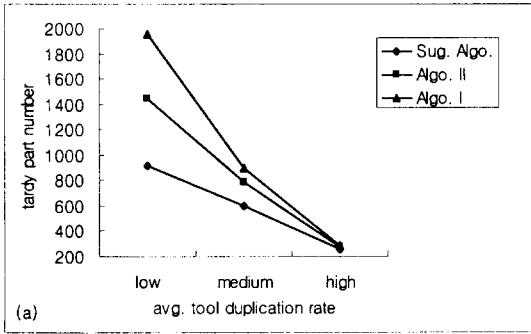
$$TA = \sum_j^n (1-TC_j) / n$$

공구 경쟁(TC_j)은 후보 부품을 가공하는 기간에 필요한 공구가 다른 기계에서 다른 부품의 가공을 위하여 같은 공구를 필요로 할 때 발생할 수 있다. 이때 그 기간에 공구의 수가 공구 경쟁을 일으키는 작업의 수 보다 많

다면 공구 대기시간이 발생하지 않으므로 작업 j 의 $TC_j = 0$ 이다. 그렇지 않으면 공구 경쟁이 일어나 공구 대기시간이 발생할 확률이 많으므로 작업 j 의 $TC_j = 1$ 이다. 식에서 n 은 후보 부품의 작업수이다. 공구 이동횟수는 후보 부품의 가공을 위하여 필요한 공구와 후보 기계에 보유하고 있는 공구의 중복 정도를 고려하여 계산한다. 알고리즘 II에서 공구할당은 알고리즘 I과 마찬가지로 작업순서를 고려하지 않고 한 공구를 사용하여 가공을 완료한 직후에 선행관계에 의하여 가능한 공구 중 적절한 다음 공구를 선택한다.

3.1.1 절의 2^k 요인실험에서 시스템의 성능척도에 큰 영향을 주는 요인이 평균 공구 중복율, 부품 종류수, 평균 작업수임을 알았다. 본 절에서는 이 요인들의 값을 변화시키면서 제안한 부품투입 알고리즘과 알고리즘 I, II의 성능을 비교하고자 한다. 공구 이동장치수는 작은 영향을 주므로 생략했다. 아래의 각 실험에서 변화시킨 요인 외의 요인은 중수준의 값을 이용했다.

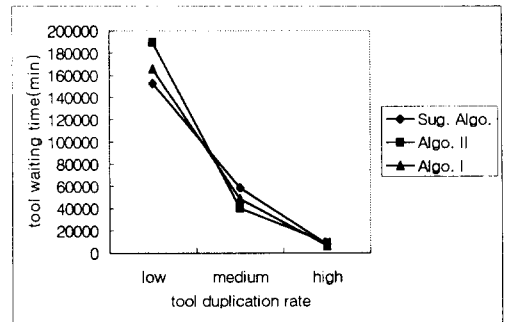
<그림 8>의 (a)는 평균 공구 중복율의 변화에 따른 납기 지연 부품수를, <그림 8>의 (b)는 생산량을 알고리즘 I, II와 비교한 그래프이다. <그림 8>의 (a)와 (b)에서 평균 공구 중복율이 고수준일 때, 세 알고리즘의 차이는 거의 없었다. 평균 공구 중복율이 저수준과 중수준일 때 제안된 알고리즘, 알고리즘 II, 알고리즘 I의 순서로 성능이 좋았다. 특히 평균 공구 중복율이 저수준일 때 제안된 알고리즘과 다른 알고리즘의 성능과 큰 차이가 있었다. <그림 9>의 (a)와 (b)에서는 부품 종류수와 평균 작업수의 변화에 따른 납기 지연 부품수를 나타냈다. 여기에서도 제안된 알고리즘, 알고리즘 II, 알고리즘 I의 순서로 납기 지연 부품이 작았다. 부품 종류수와 평균 작업수의 변경하면서 생산량에 대해서도 실험을 수행한 결과, 제안된 알고리즘, 알고리즘 II, 알고리즘 I의 순서로 많았다.



<그림 8> 평균 공구 중복율의 변화에 따른 납기 지연 부품수(a)와 생산량(b)

<그림 9> 평균 공구 중복율(a)과 평균 부품 종류수(b)의 변화에 따른 납기 지연 부품수

이 실험에서 흥미로운 것은 비교한 세 부품투입 알고리즘에서의 공구 대기시간의 변화였다. 제안된 알고리즘의 성능이 알고리즘 I, II 보다 좋았음에도 불구하고, <그림 10>과 같이 공구 중복율의 변화에 따른 실험에서 제안된 알고리즘의 총 공구 대기시간은 항상 제일 작지 않았다. 부품 종류수와 평균 작업수를 변화시킨 실험에서도 제안된 알고리즘의 공구 대기시간이 항상 작지는 않았다. 그 이유는 부품투입시 공구 대기시간만을 이용한 것이 아니고 여유시간 $(D_i - t_{now}) - (p_i + t_w)$ 을 이용했기 때문에 제안된 부품투입 알고리즘의 공구 대기시간이 반드시 작으라는 보장이 없다. 본 연구에서 사용한 공구할당 알고리즘 자체만의 실험은 3.2 절에서 다루었다.



<그림 10> 부품투입시 공구중복율의 변화에 따른 총 공구 대기시간

이 모든 실험으로부터 부품투입시 공구 경쟁의 정도인 공구 대기시간을 정확히 고려할 수록 시스템의 성능을 높일 수 있다는 것

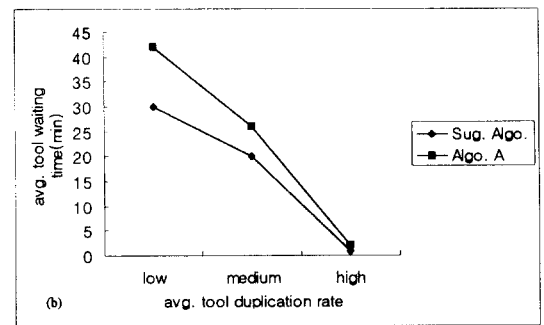
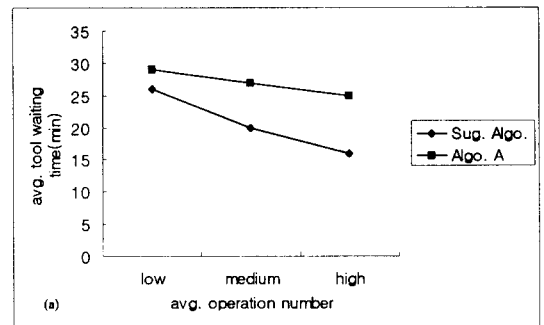
을 입증했다. 또한 <그림 8>로부터 같은 평균 공구 중복율을 유지하는 상태에서 제안된 알고리즘이 다른 알고리즘보다 낫기지면 부품수 및 생산량을 줄일 수 있음을 입증했다. 특히 <그림 8>에서 시스템의 평균 공구 중복율이 적을 때 제안된 알고리즘은 다른 알고리즘보다 월등히 성능을 개선하였다. 즉, 공구의 상태를 참조하여 계산한 공구 대기시간을 부품 투입에 이용함으로써 가공중에 공구 경쟁으로 인한 공구의 과부하를 줄여 시스템의 성능을 높일 수 있었다.

3.2 공구할당 알고리즘의 분석

본 연구의 공구할당 시점은 부품의 가공 완료 직후로써, 선행관계를 이용하여 부품의 가능한 여러 작업순서를 고려함으로써 공구 대기시간을 최소화 하는 공구할당을 수행하였다. 본 연구의 공구할당 알고리즘과 비교하려는 알고리즘 A는 3.1.2 절의 알고리즘 I과 II에서 사용했던 공구의 가공완료 직후에 선행관계를 만족하는 후보 공구중 공구 대기시간이 가장 적은 공구를 할당하는 알고리즘이다. 공구할당 알고리즘 자체만을 분석하려는 이 실험에서의 성능척도는 평균 공구 대기시간이다.

제안한 공구할당 알고리즘과 알고리즘 A를 비교하기 위하여 부품의 평균 작업수가 저수준, 중수준, 고수준인 부품을 각 10 개씩 선정하였다. 공구할당 알고리즘 자체만의 성능을 비교하기 위하여 기계별 일정계획, 기계별 보유 공구 등의 시뮬레이션 환경을 동일하게 설정한 후, 이 30 개의 부품에 대하여 한 부품씩만을 투입하여 공구할당 알고리즘을 수행하여 공구 대기시간을 측정하였다. 여기서 30 개의 부품을 연속 투입한다면 첫번째 부품 이후에는 공구의 상태가 달라지므로 두 공구할당 알고리즘을 비교하는 것이 의미가 없기 때문에 한 부품만을 투입하였다. 부품의 평균 작업수 저수준, 중수준, 고수준의 30 개 부품에 대하여

한 부품씩 공구할당 알고리즘을 수행하여 공구 대기시간을 측정하여 평균을 표시한 그래프를 <그림 11>의 (a)에 정리하였다. <그림 11>의 (a)로부터 부품의 평균 작업수가 많을수록 알고리즘 A와 공구 대기시간의 차이가 커짐을 알 수 있다. 그 이유는 작업수가 많을 때, LM과 RM의 변동폭이 커지므로 작업의 순서를 변경하여 공구의 과부하를 피할 수 있기 때문이다. <그림 11>의 (b)는 부품의 평균 작업수가 중수준인 상태에서 평균 공구 중복율의 변화에 따른 평균 공구 대기시간을 정리하였다. 여기서는 평균 공구 중복률이 작아질수록 알고리즘 A와 큰 차이를 보였다. 본 연구의 공구할당 알고리즘과 알고리즘 A를 비교한 결과, <그림 11>의 (a), (b)와 같이 공구 대기시간을 약 17% 정도 줄일 수 있었다.



<그림 11> 공구할당시 평균 작업수(a)와 평균 공구 중복율(b)의 변화에 따른 평균 공구 대기시간

공구할당 알고리즘 자체만을 비교했을 때, <그림 11>과 같이 제안된 공구할당 알고리즘이 알고리즘 A 보다 공구 대기시간을 약 17% 정도 줄였으나, <그림 8>과 <그림 9>에서 보는 바와 같이 제안된 부품투입 알고리즘이 알고리즘 I보다 약 31% 정도 성능을 개선하였고, 알고리즘 II보다 약 19% 정도 성능을 개선하였다.

4. 결론

본 연구는 SSMS 에서 동적 공구할당 전략하의 부품투입과 공구할당 문제를 언급했다. 이 연구의 동기는 공구의 상태를 참조하여 계산한 공구 대기시간을 부품투입에 고려함으로써 가공 중에 공구 경쟁으로 인한 공구의 과부하를 줄여 시스템의 성능을 높일 수 있을 것이라는 사고로부터 시작되었다.

본 연구에서는 부품투입시, 공구 경쟁의 정도를 정확히 고려하여 시스템의 성능을 개선하는 알고리즘을 제안하였다. 이를 위하여 부품투입 알고리즘의 수행 시점을 실시간으로 이동하였으며, 공구 대기시간을 최소로 하는 작업순서와 공구할당을 수행한 후, 계산된 공구 대기시간을 부품투입 알고리즘에서 이용하였다. 부품투입과 공구할당을 동시에 수행하는 본 연구의 내용과 결과를 요약하면 다음과 같다.

(1) 동적 공구할당 전략하에서 작업순서를 적절히 변경함으로써 공구 대기시간을 최소

로 하는 실시간 공구할당 알고리즘을 제안하였다.

(2) 계산된 공구 대기시간으로부터 여유시간을 정확히 계산하여, 이를 이용한 부품투입 알고리즘을 제안하였다.

(3) 시뮬레이션 실험을 통하여 (1)의 공구할당 알고리즘을 작업순서를 고려하지 않고 한 공구의 가공을 완료한 직후에 다음 공구를 할당하는 알고리즘과 비교한 결과, 공구 대기시간을 약 19% 정도 줄일 수 있음을 입증하였다. 또한 부품투입시 공구 경쟁의 정도를 정확히 예측한 부품투입 알고리즘이 다른 알고리즘보다 시스템의 성능을 높일 수 있다는 것을 입증하였다. 특히 평균 공구 중복율이 적은 상태에서는 다른 알고리즘보다 월등히 성능을 개선하였다. 즉, 공구의 상태를 참조하여 계산한 공구 대기시간을 부품투입에 이용함으로써 가공중에 공구 경쟁으로 인한 공구의 과부하를 줄여 시스템의 성능을 높일 수 있었다.

SSMS 에서 생산하려는 부품 종류수와 부품 비율에 대하여 필요한 공구 종류별 최적량을 보유하고 있다면 시스템의 성능은 개선될 것이다. 향후 연구로써 제안한 알고리즘의 결과를 이용하여 민감도 분석(sensitivity analysis)을 수행함으로써 시스템 성능을 높이기 위한 공구 종류별 적정 공구량을 계산하는 연구를 수행할 것이다.

Reference

- [1] Cantamessa, M. and Lombardi, F., "Tool flow planning in a flexible manufacturing system", *Computer Integrated Manufacturing Systems*, Vol. 6(2), pp. 93-102, 1993.
- [2] Cariapa, V., "Multi-mode machine tools - a concept that improves operations of flexible manufacturing systems", *International Journal of Production Research*, Vol. 29(5), pp. 1069-1079, 1991.
- [3] Cuppan, B., "Tool management considerations for cells and FMS", 3rd Biennial International Machine Tool Technical Conference, pp. 6.71-6.86, 1986.
- [4] Grieco, A., Semararo, Q., Tolio, T. and Toma, S., "Simulation of tool and part flow in FMS", *International Journal of Production Research*, Vol. 33(3), pp. 643-658, 1995.
- [5] Hankins, S. and Rovito, V., "The impact of tooling on flexible manufacturing systems", Second Biennial International Machine Tool Technical Conference, pp. 193-199, Illinois, 1984.
- [6] Hollingum, J., "Renishaw probes new manufacturing method", *The FMS Magazine*, Vol. 6(2), pp. 75-77, 1988.
- [7] Iwata, K. and Onosato, M., "Random manufacturing system: a new concept of manufacturing systems for production to order", *Annals of the CIRP*, Vol. 43(1), pp. 379-383, 1994.
- [8] Jones, T., How the big(and little) guys do it, tooling and production, pp. T7-T10, 1994.
- [9] Kashap, A. S. and Khator, S. K., "Control rules for tool sharing in flexible manufacturing systems", *Computers and Industrial Manufacturing*, Vol. 25(1-4), pp. 507-510, 1993.
- [10] Kellock, B., "One-hit machining cuts bar work cost", *Machinery and Production Engineering systems*, Vol. 143(3682), pp. 38-39, 1985.
- [11] Koo, P-H., "Flow planning and control of single-stage multimachines systems", Ph.D Thesis, Purdue university, 1996.
- [12] Levin, J. B. and Dutta, D., "Computer aided process planning for parallel machines", *Journal of Manufacturing Systems*, Vol. 11(2), pp. 79-92, 1991.
- [13] MacCarthy, B. L. and Liu, J., "Addressing the gap in scheduling research : a review of optimization and heuristic methods in production scheduling", *International Journal of Production Research*, Vol. 31(1), pp. 59-79, 1993.
- [14] Mason, F., "Computerized cutting tool management", *American Machinist and Automated Manufacturing*, pp. 105-120, 1986.
- [15] Rho, H. M., Geelink, R., van 't Erve, A. H., Kals, H. J. J., "An Integrated Cutting Tool Selection and Operation Sequencing Method", *Annals of the CIRP*, Vol. 41(1), pp.517-520, 1992.
- [16] Wang, E., Kim, Y. S., Lee, C. S., and Rho, H. M., "Feature-Based Machining Precedence Reasoning and Sequence Planning", *Proc. ASME Computers in Engineering Conference*, 1998.

● 저자소개 ●

이충수



1987년 한양대학교 산업공학 학사
 1989년 한양대학교 산업공학 석사
 현 재 한국과학기술연구원 CAD/CAM 센터 연구원
 관심분야 : 공정설계, 일정계획, 생산정보 통합화

김성식



1972년 고려대학교 기계공학 학사
 1974년 고려대학교 산업공학 석사
 1976년 S. M. U. 산업공학 석사
 1979년 S. M. U. 산업공학 박사
 현 재 고려대학교 산업공학과 교수
 관심분야 : 생산공학, 시뮬레이션, 분산환경에서의 ERP,
 객체지향 CIM 시스템

노형민



1977년 서울대학교 기계설계학 학사
 1979년 한국과학기술원 생산공학 석사
 1985년 펜실바니아 주립대 산업공학 석사
 현 재 한국과학기술연구원 CAD/CAM 센터 책임연구원
 관심분야 : 생산정보 통합화, 공정설계