

응용논문

자원제약하의 동적 다중 프로젝트 일정계획에
Tabu Search 적용
- A Tabu Search Approach for Resource
Constrained Dynamic Multi-Projects Scheduling -

윤 중 준*
Yun, Joung Jun
이 화 기**
Lee, Hwa Ki

Abstract

Resource Constrained Dynamic Multi-Projects Scheduling(RCDMPS) is intended to minimize the total processing time(makespan) of two or more projects sequentially arriving at the shop under restricted resources.

The aim of this paper is to develop the new Tabu Search heuristic for RCDMPS to minimize makespan.

We propose the insertion method to generate the neighborhood solutions in applying the Tabu Search for the RCDMPS and the diversification strategy to search the solution space diversely. The proposed diversification strategy apply the dynamic tabu list that the tabu list size is generated and renewed at each iteration by the complexity of the project, and change the proposed tabu attribute.

In this paper, We use the dynamic tabu list for the diversification strategy and intensification strategy in the tabu search, and compare with other dispatching heuristic method to verify that the new heuristic method minimize the makespan of the problem.

1. 서론

자원제약하의 다중 프로젝트 문제는 두 개 또는 그 이상의 프로젝트가 연속적으로 발생하여 각 프로젝트를 동시에 수행할 경우 각 프로젝트 네트워크에서 작업의 선행관계와 주어진 자원 이용량을 만족시키면서 각 프로젝트의 수행기간을 최소로 하는 일정계획에 관한 문제이다.

대부분의 현실적인 상황에서 발생하는 프로젝트 일정계획 문제는 두 개 이상의 프로젝트가 동시에 또는 임의의 시간 간격을 두고 발생함으로써 전체 프로젝트 일정을 새로이 설계해야 하는 문제가 발생한다. 따라서 주어진 자원을 효율적으로 사용해서 각각의 프로젝트를 가장 빨리 끝낼 수 있는 일정계획이 요구된다. 지금까지 자원제약하의 다중 프로젝트에 대한 연구는 있었으나 임의의 시간을 두고 발생하는 동적인 상황에서의 다중 프로젝트에 대한 연구는 희박하다.

본 연구의 목적은 자원제약하의 동적인 상황에서 다중 프로젝트 일정계획 문제(Resource Constrained Dynamic Multi-Projects Scheduling. 이하 RCDMPS라 함)에 대하여 해의 개선 알고리즘인 Tabu Search(이하 TS라 함)기법을 적용하여 총 처리시간(makespan)을 최소로 하는 프로젝트 일정계획을 생성하고자 한다.

* 인하대학교 대학원 산업공학과

** 인하대학교 산업공학과

RCDMPS에 TS를 적용함에 있어서 탐색성능을 향상시키기 위해 이웃해 생성방법과 TS의 매개변수에 대한 연구가 요구된다. 따라서 RCDMPS를 위한 TS기법을 개발하기 위해서 이웃해 생성방법을 제시하고, TS 매개변수를 연구하여 RCDMPS에 적합한 단순 TS기법과 TS의 다양화 기법들을 제안한다. 비교분석은 제안한 TS기법들을 비교하고, 기존의 작업배분 휴리스틱인 MINLFT(Dynamic)과 비교한다. 또한 모의 실험과 사용자 편의(GUI 환경)를 위해 제안된 TS기법은 범용 프로그래밍 언어인 Microsoft Visual Basic 6.0을 사용하였다.

2. 이론적 고찰

2.1 자원제약하의 동적 다중 프로젝트 문제

일반적인 프로젝트 일정계획 문제는 프로젝트가 단일이나 다중이나 와 자원이 단일이나 다중이나에 따라 구분된다.

자원제약 다중 프로젝트 문제는 프로젝트가 두 개 또는 그 이상의 숫자로 연속적으로 발생하는 경우 각 프로젝트의 네트워크에서 활동(activity)들의 선행관계와 자원 이용량을 만족시키면서 각 활동의 완료시간을 결정하는 것이다. 여기서 각각의 프로젝트는 여러 개의 활동들로 이루어져 있고 각각의 활동은 여러 종류의 자원을 이용해서 수행되어 진다. 일반적으로 자원제약 다중 프로젝트 일정계획 문제는 크게 주어진 자원의 효율적인 사용을 고려한 자원 평준화(resource leveling)문제와 프로젝트 수행기간(makespan)의 최소화 문제로 나누어진다.

다중 프로젝트 일정계획 문제는 프로젝트의 수가 커지고 이들을 구성하고 있는 활동의 수가 매우 많아지면 해석적 방법에 의해서 해를 구하기가 어렵다. 문제의 최적 해를 얻어내기 어려운 이유로 인해 문제의 해결을 위해 제시되고 있는 방법들이 "heuristic 기법"을 도입한 작업배분 규칙이다. 크기가 큰 프로젝트의 연속적인 발생 시에는 작업수의 증가에 따르는 계산의 양을 효과적으로 줄일 수 있는 heuristic 기법의 사용이 더 유용하다.

Kurtulus and Davis[8]은 단일자원에 대한 다중 프로젝트에 대한 연구에서 어느 한 순간에 필요한 자원의 최대량과 각각의 자원의 종류에 따른 가용 율에 중점을 두고 분석한 결과 납기 지연에 대한 순위와 평균 완료시간에 대한 순위에서 SASP(Shortest Activity from Shortest Project)가 매우 좋은 결과를 생성했다.

그러나 위의 규칙은 모든 프로젝트를 즉시 알고있는 정적상황의 규칙이다. 즉, 정적상황은 시간 축에 자원에 대한 요구 량이 완전히 설정되어 있다. 그러므로 만약 새로운 프로젝트들이 시간간격을 두고 들어온다면 동적 상황이 존재하고 미래의 자원에 대한 요구 량이 불확실하게 된다[10]. 여기에서 동적 상황이란 미래의 자원에 대한 제약이 어떠한 형태로 발생되게 될지 알 수 없으므로 새롭게 추가되는 프로젝트의 진행과정 또한 정적인 경우와 달리 어떠한 형태로 진행되게 될지 알 수 없다. 하승진[2]은 어떤 시간대에서 남아있는 프로젝트들의 마지막 활동들의 끝나는 시간 중 가장 큰 값 중에서 가장 작은 값을 갖는 작업을 먼저 선택하는 규칙인 MINLFT(Dynamic)규칙을 제안하여 동적인 상황에서 적용된 SASP규칙보다 총 처리시간 면에서 우수함을 보였다.

2.2 Tabu Search

Tabu Search(TS)는 어려운 조합문제를 해결하기 위한 새로운 Artificial Intelligence(AI)기법으로 Glover에 의해서 TS가 제안되었으며[7] 외판원 문제, Flow Shop 문제, Job Shop 문제, Quadratic Assignment Problem(QAP), Resource Constrained Project Scheduling Problem(RCPSP)등의 일정계획에서 조합최적화 문제에 효율적으로 적용되고 있다.[1, 4, 5, 6, 12, 13, 14]

TS기법의 기본적인 탐색절차는 초기해 $S_0 \in X$ 를 현재해 s 로 하고, 탐색을 시작하여 이웃해 집단 $N(s)$ 을 생성하고, 그 중에서 목적함수 $F(s)$ 가 최소가 되는 s' 를 선택한다. 이때 $F(s') < F(s)$ 이면 s' 를 현재해로 하여 탐색을 진행하는 절차는 이웃해 탐색방법과 같다. 그러나 TS는 선택한 s' 가 $F(s') < F(s)$ 일지라도 탐색공간을 넓히기 위하여 s' 를 현재해로 하여 탐색을 계속한다. 그리고 해의 사이클링을 방지하기 위해 s' 를 일정기간동안 기억장소(Tabu List) T 에 저장하여 새로 찾아진 s' 가 T 안의 해(Tabu 상태)가 되지 않도록 한다. 또한 더 나은 해를 찾기 위하여 새로 찾아진 s' 가 T 안의 해 일지라도 s' 를 선택할 수 있는 기준(이를 열망기준)을 사용하여 s' 가 이 기준을 만족하면 이를 선택한다. TS는 이상과 같은 탐색절차를 일정 회수동안 반복하여 해를 찾는 방법이다. 이러한 TS를 단순(simple) TS기법이라 부른다[1].

TS기법에서는 현재해의 여러 이웃해를 생성함으로써 해 공간의 탐색을 강화(intensification)하는 한편, tabu 목록에 있는 해로의 이동을 막음으로써 해 공간의 탐색을 확장해 나간다. 또한 단순 TS기법에서는 최근에 일어난 한정된 이동만을 tabu 목록에 기억하는 단기기억(short term memory)을 사용하고 있고, 해 공간을 다양하게 탐색(diversification)하기 위하여 해의 이동에 관한 정보를 처음부터 현재까지 기억하는 장기기억(long term memory)을 이용하는 다양한 TS기법의 여러 연구가 이루어 지고있다.

3. 자원제약하의 동적 다중 프로젝트 일정계획을 위한 Tabu Search 기법

본 연구에서 다루는 RCDMPS은 아래의 가정을 갖는다.

- 첫째, 각 프로젝트는 서로 독립이며, 사용하는 자원은 서로 공통이다.
- 둘째, 각 프로젝트에서 활동 수행시간은 알려져 있고, 비음 정수이다.
- 셋째, 작업준비시간은 무시되고, 수행시간에 포함된 것으로 본다.
- 넷째, 모든 종류의 자원은 재사용 가능한 유형의 자원이다.

본 장에서는 RCDMPS에 TS를 적용하기 위해서 고려되는 초기해 및 평가기준, 이웃해 생성 절차를 제안하고, tabu 매개변수를 다음과 같이 설계한다.

3.1 초기해 및 평가기준

초기해 생성은 단일 프로젝트에 대해서 FCFS 휴리스틱 규칙으로 생성된 일정계획을 초기해로 하여 TS기법을 적용한다. 이어서 임의의 시점에 두 번째 프로젝트가 도착하면 동적 상황이 존재하며 임의의 시점 이전의 일정계획은 TS를 적용하여 진행되어온 단일 프로젝트의 일정계획을 따르고, 임의의 시점 이후의 활동들에 대해서는 다시 FCFS 휴리스틱 기법을 적용하여 생성된 일정계획을 자원제약하의 동적 다중 프로젝트 문제의 초기해로 선정한다. 사용되는 용어는 다음과 같다.

- N : 프로젝트에서 활동(arc)들의 총 수.
- K : 프로젝트에서 마디(node)들의 총 수.
- n_i : 활동 i 의 직후행 활동의 수.
- CC : 프로젝트의 복잡도(complexity), node 수와 arc 수의 비율, 즉 $CC = \frac{N}{K}$
- (i, j) : 활동의 시작 node i , 완료 node j .
- PS_t : 현재해로서 반복에 의해 생성되는 프로젝트 partial schedule을 나타냄.
 $t = 1, 2, \dots, T$.
- $M(PS_t)$: makespan, 즉, PS_t 의 총 처리시간.

제시한 초기해 생성 절차는 다음과 같다.

< 초기해 생성절차 >

절차 1. 초기해를 구한다.

- a) 자원제약하의 단일 프로젝트 일정계획 문제에서는 FCFS 휴리스틱 기법을 적용하여 생성된 일정계획을 초기해로 한다.
- b) 임의의 시간에 프로젝트 p가 도착하는 동적 다중 프로젝트에서는 TS를 적용하여 생성된 단일 프로젝트와 프로젝트 p가 도착한 시간 이후의 활동들에 대해서만 FCFS 휴리스틱 기법을 적용하여 생성된 일정계획을 초기해로 한다.

절차 2. 초기해를 현재해와 최선해로 둔다.

평가기준은 프로젝트의 총 처리시간 즉, makespan 이다.

3.2 이웃해 생성 절차

TS를 RCDMPS 문제에 적용하여 탐색할 때 찾아진 하나의 현재해를 s 라 한다. 현재해 s 에서 이웃해 s' 를 생성하는 이동방법으로 순서문제에서는 일반적으로 두 개의 원소를 서로 교환하는 교환이동(swap move)과 하나의 원소를 다른 위치로 삽입하는 삽입이동(insert move)방법이 가장 많이 사용되고 있다. n 개의 작업에 대해 교환이동으로 발생할 수 있는 순열의 개수는 $n(n-1)/2$ 이고, 삽입이동에 의해 발생할 수 있는 순열의 개수는 $n(n-1)$ 개이다. 교환이동으로 생성된 이웃해는 두 번의 삽입이동으로 생성할 수 있으나 삽입이동으로 생성된 어떤 이웃해는 교환이동에 의해 생성될 수 없는 경우도 있다[1]. 몇몇 연구에서는 교환이동보다 삽입이동이 탐색효율 면에서 더 좋은 것으로 나타났다[11, 12]. 따라서 본 연구에서는 더 많은 이웃해들을 생성함으로써 적극적인 해의 탐색을 위해 삽입이동방법을 사용하고, 문제의 자원제약, 순서제약을 고려하여 이웃해를 생성한다. 사용되는 용어 및 기호는 다음과 같다.

- $st(i, j)$: 활동 (i, j) 의 시작 시간.
- $ft(i, j)$: 활동 (i, j) 의 완료 시간, $ft(i, j) = st(i, j) + d_{ij}$
- $d_{(i, j)}$: 활동 (i, j) 을 수행하는데 걸리는 시간.
- $TGT(i, j)$: 삽입을 위해 선택되는 대상활동.
- $r(TGT(i, j))$: $TGT(i, j)$ 활동이 이용하는 자원.
- AS : $r(TGT(i, j))$ 에서 $TGT(i, j)$ 가 삽입 가능한 가공순서에 있는 활동으로 활동의 완료시간이 LB 보다 크고, UB 보다 작거나 같은 활동의 집합에서 $TGT(i, j)$ 을 뺀 활동의 집합.
- A_{rk} : PT_r 에서 k 번째의 가공순서를 갖는 활동. $k = 1, 2, \dots, K$.
- PT_r : $arrt_p$ 이후의 자원 r 을 이용하는 활동순서의 집합. $r = 1, 2, \dots, R$.
- LB : Lower Bound. $TGT(i, j)$ 가 가장 빨리 시작할 수 있는 시점, $TGT(i, j)$ 의 직전 선행 활동들 중 가장 큰 완료시간과 $arrt_p$ 중 최대값.
- UB : Upper Bound. $TGT(i, j)$ 가 가장 늦게 시작할 수 있는 시점, $TGT(i, j)$ 의 직후행 활동의 시작시간, 후행활동이 없으면 현재 schedule의 makespan이다.
- \emptyset : level. SA와 NSA를 구분하는 기준, $TGT(i, j)$ 가 삽입된 가공 순서를 고려하여 계산한 시작시간 $st(TGT(i, j))$ 과 PS_t 에서 $TGT(i, j)$ 활동의 원래의 시작시간 인 $st^*(TGT(i, j))$ 중에서 최소값으로 한다.

- SA : Scheduled Activity. PS_i 에서 \emptyset 이전에 완료되는 활동의 집합.
 NSA : Non Scheduled Activity. PS_i 에서 \emptyset 이후에 완료되는 활동의 집합,
 $TGT(i, j)$ 는 항상 NSA의 원소.
 CA : Critical Activity. 직전 선행활동이 완료된 직후에 시작되거나 선행활동이 없고 시작 시간이 $arrt_p$ 인 활동.
 NCA : Non Critical Activity. 직전 선행활동이 완료된 직후에 시작하지 않거나, 선행활동이 없고 시작 시간이 $arrt_p$ 이 아닌 활동.
 FA (i, j): First Activity. NSA중 각 자원에서 시작할 수 있는 가장 빠른 활동.

기존의 job shop 문제[1]에서는 이웃해의 효율적 탐색과 공정이동에 따른 효과적인 재 일정 계획을 위하여 삽입공정을 선택하여 각 공정의 삽입이동 범위를 구하였고, 이동 가능한 삽입공정과 그 공정의 삽입위치를 제한하는 방법을 제시하였다.

RCDMPS에서도 현재해에서 이웃해를 구하는 방법은 현재의 해(일정계획)에서 임의의 활동을 선택하여 그 활동을 임의의 다른 위치로 삽입하는 방법이다. 본 연구에서는 이웃해의 효율적 탐색과 이동활동에 따른 효과적인 재 일정계획을 위하여 NCA집합 중에서 이동가능한 대상활동을 구하여, 동일자원을 이용하는 자원에서 삽입이동 범위를 구하고, 범위내의 각 활동들의 위치로 삽입이동 하는 방법이다.

대상활동의 삽입이동범위는 Lower Bound(LB) 와 Upper Bound(UB)로 구분하며, 이동범위 내에서 대상활동을 현재와 다른 모든 위치로 삽입이동 시킨다면, 이때 발생하는 삽입이동의 수는 삽입이동 범위내에 있는 공정의 수에서 하나를 뺀 수가된다. 연구에서 제시한 삽입이동 방법으로 대상활동을 이동한 후, 변경된 각각의 활동순서를 유지하면서 각 활동이 가장 빨리 시작하는(활동을 left-shift 시킨) 일정은 active 일정을 보장하지 못한다. 따라서 active 일정을 위한 재 일정계획이 필요하게 된다.

본 연구에서 제시하는 active 재 일정계획은 대상활동의 삽입으로 active 일정계획이 유지되지 못하는 활동들(NSA)만을 자원의 여유시간을 고려하여 재 일정계획 하는 것이다. 재 일정계획 하는 절차는 이웃해 생성절차8, b)에서의 순서를 따른다.

삽입이동 방법 및 재 일정계획 절차를 이용한 이웃해 생성절차는 다음과 같다.

< 이웃해 생성 절차 >

- 절차 1. 초기해에서 프로젝트 p 의 도착 시간 $arrt_p$ 를 구한다.
 절차 2. 현재해에서 $arrt_p$ 이후의 PT_r , NCA집합을 구한다.
 절차 3. NCA집합에서 가장 늦은 완료시간을 갖는 활동을 $TGT(i, j)$ 로 선택한다. tie가 발생하면 임의로 선택한다.
 절차 4. $TGT(i, j)$ 의 삽입이동 가능한 범위 LB, UB를 구한다.
 절차 5. a) PT_r 에서 삽입이동 가능한 범위사이에 가공되는 활동들의 집합 AS를 구한다.
 b) AS에서 첫 번째 활동 A_{rk} 을 선택하고, AS수정한다.
 절차 6. a) 자원 r 을 이용하는 PT_r 에서 $TGT(i, j)$ 의 순서를 수정한다.

가공순서의 수정은 PT_r 에서 $TGT(i, j)$ 는 선택한 활동 A_{rk} 위치로 삽입이동하고, A_{rk} 는 $TGT(i, j)$ 가 현 가공순서 이전으로 삽입이동 할 때에는 $TGT(i, j)$ 다음의 가공순서를 갖고 뒤로 삽입이동 할 때에는 $TGT(i, j)$ 이전의 가공순서를 갖는다.

- b) 대상활동의 시작시간 $st(TGT(i, j))$ 을 구한다. 시작시간은 대상활동의 직전 선행활동들 중 가장 큰 완료시간과 삽입된 위치 바로 이전활동의 완료시간 중 최대값이다.

절차 7. 대상활동의 시작시간 \emptyset 을 기준으로 SA와 NSA로 구분한다.

절차 8. a) SA의 활동은 현재의 일정계획을 따른다.

b) NSA의 활동은 다음의 순서에 따라 재 일정계획 한다.

순서 1. NSA에서 $FA(i, j)$ 을 선택한다.

순서 2. $FA(i, j)$ 중 $r = r(TGT(i, j))$ 을 이용하는 활동 (i, j) 의 선행제약을 검사한다. (i, j) 활동에 대해 NSA에 선행활동이 있으면 순서 5로 가고, 그렇지 않으면 순서 3으로 간다.

순서 3. $FA(i, j)$ 의 활동 중 자원 r 을 이용하는 (i, j) 에 대하여 가장 빠른 시작시간 $st(i, j)$ 을 구한다.

① SA에서 (i, j) 의 선행활동들 중 완료시간이 가장 큰 시간을 구한다.

② 활동 (i, j) 을 자원 r 의 여유시간을 고려하여 SA에 첨가하고 $FA(i, j)$ 을 수정한다.

③ PT_r 을 수정한다.

순서 4. 활동 (i, j) 의 시작시간을 구하고 SA에 첨가하며, $FA(i, j)$ 을 수정한 후 순서 5로 간다.

순서 5. $FA(i, j) = \{\emptyset\}$ 이면 절차 9로 가고, 그렇지 않으면 $r = r+1$ 로 하여 순서 2로 간다.

절차 9. $AS = \{\emptyset\}$ 아니면, 절차 10으로 간다. 그렇지 않으면 절차 5로 간다.

절차 10. $NCA = \{\emptyset\}$ 이면, 끝낸다. 그렇지 않으면 절차 3으로 간다.

3.3 tabu 매개변수

해의 개선 알고리즘인 TS 기법의 적용단계에서 중요한 것은 주어진 문제 특성에 맞게 tabu 매개변수를 어떻게 활용하는가에 달려 있다. tabu 매개변수를 조정함으로써 최적해나 최적 근사해를 찾는 탐색시간을 줄일 수 있어 현실적인 문제에 적용할 수 있다[1]. 연구에서는 tabu 매개변수를 다음과 같이 고려한다.

3.3.1 해의 이동

해의 이동(move)은 하나의 가능해로부터 생성한 모든 이웃해 집단 중에서 하나의 이웃해로 이동하는 것이다. 본 연구에서 이동이란 삽입이동방법에 의한 이웃해 생성절차를 이용하여 이웃해를 생성하고, tabu 제약과 열망기준을 만족하는 가장 좋은 이웃해로의 변환을 말한다.

3.3.2 tabu 속성

tabu 속성은 해의 이동을 제약(tabu 제약)하는 것으로 어떤 속성을 갖는 해의 이동을 막음으로써 해의 순환(cycling)을 방지하고 탐색속도를 빠르게 한다. 따라서 tabu 제약은 이러한 속성들의 집합을 선택함으로써 만들어지고 tabu 목록에 기록된다.

RCDMPS 모델은 job shop 모델과 유사하므로, [1]에서 제시한 tabu 속성을 본 문제의 특성에 맞게 변형하여 사용한다. 따라서 본 연구에서는 삽입활동으로 선택된 활동(대상활동)과 대상활동이 이용하는 자원의 순서 위치를 기억하고, 삽입하고자 하는 위치의 활동과 순서위치를 고려하는 다중 tabu 속성을 제안한다. 즉 tabu 속성 TA1, TA2, TA3는 다음과 같다.

(1) TA1 : 삽입이동을 고려하는 대상활동과 이용자원만을 나타낸다.

(2) TA2 : TA1에 추가하여 삽입이동 되는 위치의 활동을 나타낸다.

(3) TA3 : TA2에 추가하여 삽입이동 대상활동의 삽입 전후의 가공순서 위치를 나타낸다.

3.3.3 tabu 목록의 크기 $|TL_s|$

tabu 목록의 크기는 tabu 목록에 저장되는 tabu 속성의 개수를 나타내는 것으로, 탐색과정에서 tabu 목록의 크기만큼 해의 순환을 제약할 수 있다. tabu 목록의 크기는 문제의 크기에 따라 다르나, tabu 목록의 크기가 너무 작으면 순환에 빠지기 쉽고, 너무 크면 순환을 방지할 수 있으나 부분최적에 빠지기 쉽다. tabu 목록의 크기는 탐색 중에 정해진 크기만큼 유지된다[1]. 즉, TS는 탐색을 강화(intensification)하거나 다양화(diversification)하기 위해 일반적으로는 여러 가지 방법이 제시되었으나 본 연구에서는 새로운 tabu 속성이 들어오면 가장 먼저 들어온 tabu 속성이 tabu 목록에서 빠져나가게 되는 선입선출 방식을 이용한다.

또한 연구에서는 tabu 목록의 크기를 단순 TS적용 시에는 단기기억을 이용하며, 다양화 기법으로 제안되는 TS에서는 중·장기기억을 이용한다. 즉, 프로젝트 네트워크의 구조, 크기 등의 프로젝트 특성에 따른 복잡성에 의해 각 실행회수 안에서 지역적으로 정해지게 한다. 이러한 개념의 tabu 목록의 크기 연구는 Saihi[13]에 의해서 연구되었다. 본 연구에서는 다음의 수식과 같이 tabu 목록의 크기를 각 실행회수마다 계산하고 목록을 갱신하는 동적 타부목록을 사용한다.

$$|TL_s| = \begin{cases} \text{Max}[30, N/(n_i + n_j)] & \text{if } CC < 1.4 \\ \text{Min}[30, N/(n_i + n_j)] & \text{otherwise} \end{cases}$$

3.3.4 열망기준(aspiration level)

열망기준은 어떤 이동이 tabu 상태이지만 이를 해제하여 해의 이동을 가능케 하는 기준으로서, 연구에서는 현재까지 탐색한 가장 좋은 해, 즉 최선해의 총 처리시간으로 한다.

3.3.5 종료조건(stop criterion)

종료조건으로는 반복 수(실행회수), 탐색 이웃해의 수, 최선해를 개선하지 못한 회수 등이 있으나, 연구에서는 반복 수 및 다양화 전략동안 최선해를 개선하지 못하면 TS기법을 종료한다.

3.4 다양화 기법

해 공간이 좁으면 일반적으로 단순 TS기법만으로 좋은해를 찾을 수 있으나 해 공간이 넓은 어려운 문제에서는 흔히 좋은해를 찾지 못하게 된다. 해의 개선을 위하여 해 공간의 탐색이 다양하게 이루어져야 한다. TS기법에서 해 공간의 탐색을 다양화하는 여러 기법들이 제안되었다 [1, 13]. 가장 단순한 다양화 방법은 단순 TS를 반복 적용하는 방법이고, 장기기억함수를 이용하는 방법 등이 제시되고 있다. 본 연구에서는 첫째, 단순 TS기법을 하나의 초기해에 대해 일정회수(임의 선정) 만큼 이동한 후 새로운 초기해로 다시 시작하는 과정을 종료조건을 만족할 때까지 반복하는 방법과 둘째, tabu 속성을 사용하여 tabu 상태를 더 제약적으로 하며, 장기기억 함수를 이용하는 두 가지 다양화 방법을 사용하고자 한다. 두 번째 TS기법은 이동에 따른 정보(tabu 속성)를 장기기억 함수에 저장하고 이동하여, 미리 지정된 이동회수나, 최선해가 개선되지 못한 회수를 만족하면 tabu 속성을 변경하여 지금까지 탐색된 가장 좋은해를 새로운 초기해로 하고 TS기법을 다시 시작한다. 다양화 전략을 수행하는 절차는 다음과 같다.

< 다양화 전략 수행 절차 >

- 절차 1 : 현재해 $s = Best_PS$ 라 한다.
- 절차 2 : tabu 목록의 크기는 동적 타부목록으로 각 실행회수마다 계산한다.
- 절차 3 : 제시한 tabu 속성 3가지를 순차적으로 적용한다.
 - 순서 1. tabu 속성 TA1을 이용
 - 순서 2. tabu 속성 TA2를 이용

순서 3. tabu 속성 TA3을 이용

절차 4 : 알고리즘을 종료조건이 만족할 때까지 계속 실행한다.

즉, 절차 3의 순서 1에서 현재해의 개선을 일정회수 동안 개선하지 못하면, 순서 2와 순서 3으로 순차적으로 넘어간다. 순차적으로 tabu 속성을 적용할 때 초기해는 지금까지 찾아진 가장 좋은해로 한다. 만약, 다양화 전략동안 해의 개선이 없으면 TS는 종료되고, 새로운 개선된 해를 얻으면 다양화 전략은 순서 1로 되돌아가서 알고리즘을 계속 수행한다.

3.5 제안된 TS기법

본 연구에서 제안된 다양화 전략 적용 TS 기법의 알고리즘 흐름도는 그림 1과 같으며, 사용되는 용어 및 기호 정의는 다음과 같다.

$|TL_s|$: Tabu List size. tabu 목록의 최대 크기이다.

ta : tabu attribute. tabu 속성을 나타낸다.

ac : aspiration criteria. TS의 열망기준.

sc : stop criterion. 종료 조건.

$Best_NS$: 현재해로부터 생성한 모든 이웃해 중 가장 좋은 이웃해이다.

$Best_PS$: 최선해로서 종료조건을 만족 할 때까지 생성한 $Best_NS$ 중 가장 좋은 해이다.

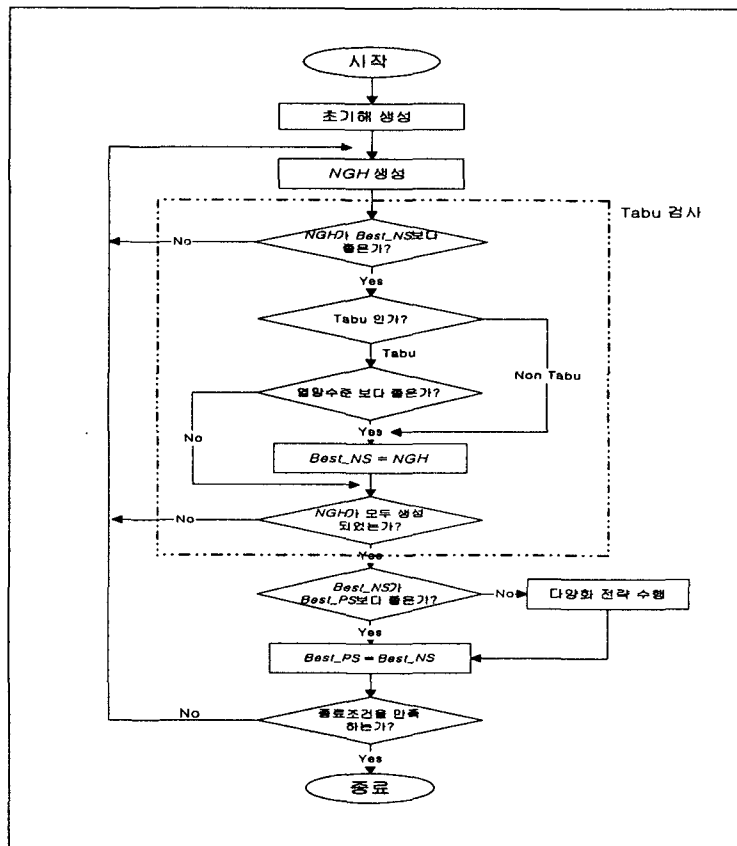


그림 1. 제안된 TS 기법의 알고리즘 흐름도

4. 제안된 Tabu Search 기법 적용 예

4.1 모의실험의 결과 비교

본 절에서는 [2]에서 사용된 여러 가지 형태의 프로젝트 예를 제시하고, 앞장에서 제안한 다양화 전략 적용 TS에 의한 일정계획과 [2]에서 제안한 MINLFT(dynamic)휴리스틱을 비교 실험한다. 모의실험에 사용된 프로젝트는 임의의 3개 프로젝트로 그림 2의 프로젝트 네트워크를 수치 예 1, 그림 3의 프로젝트 네트워크를 수치 예 2, 그림 4의 프로젝트 네트워크를 수치 예 3으로 하여 모의실험 한다. 수치예의 네트워크 상에서 임의의 호에 표시된 영문표기는 필요설비에 대한 약어이고 숫자는 작업시간을 의미한다. 표 1은 MINLFT 휴리스틱 규칙과 제안한 다양화 전략 적용 TS에 의한 일정계획 절차를 모의 실험한 결과이다.

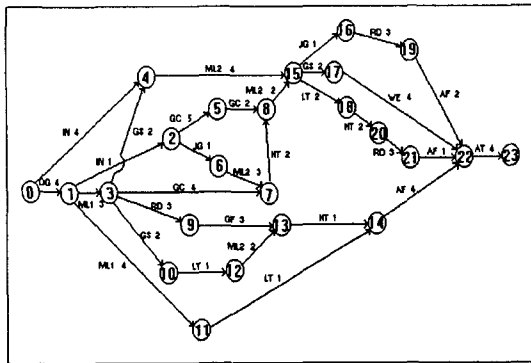


그림 2. 수치 예 1

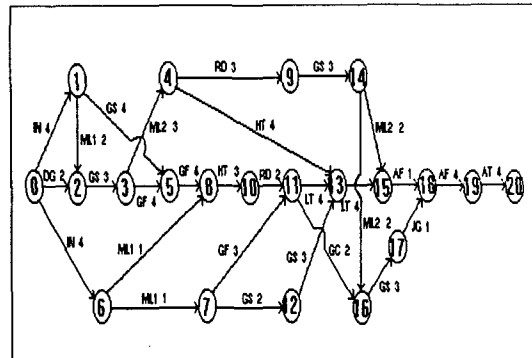


그림 3. 수치 예 2

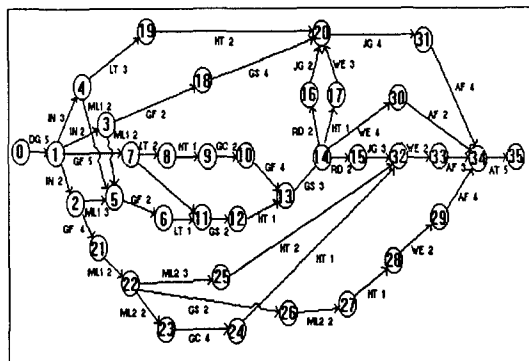


그림 4. 수치 예 3

총 처리시간

구분	도착시간	실행 1	실행 2	실행 3
MINLFT (Dynamic)	15일	54	64	51
	20일	59	69	55
제안한 TS 절차	15일	54	61	49
	20일	59	66	53

표 1. 모의실험 결과 비교

표 1에서 실행 1은 먼저 수치 예 1이 수주된 후에 15일, 20일 시간 간격을 두고 수치 예 2가 수주되었을 경우이고, 실행 2는 수치 예 1과 수치 예 3이 수주되었을 경우, 실행 3은 수치 예 2와 수치 예 1이 동적으로 수주되었을 경우에 대해 모의 실험한 결과이다. 실험에서 보여 주듯이 실행 1은 같은 결과를 실행 2, 3은 총 처리시간(makespan)에서 본 연구에서 제안한 TS에 의한 일정계획 절차가 우수함을 보여주고 있다.

4.2 금형 가공 예

본 연구에 사용된 모델은 인천의 남동 공단에 위치한 금형 및 프레스 가공업체인 S사의 프로젝트들을 이용한다. 이 공장의 금형 가공에 사용되는 기계는 모두 12대이며, 작업을 수행하기 위해 사용되고 있는 실제 프로젝트는 그림 5의 프로젝트 1과 그림 6의 프로젝트 2이다.

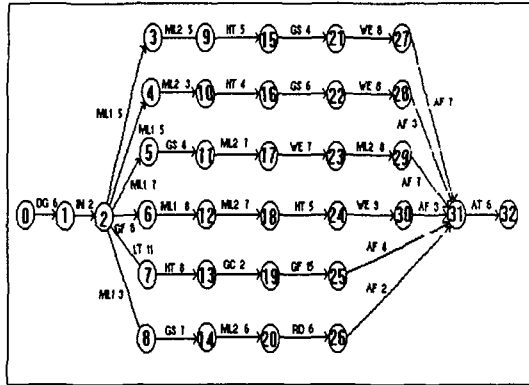


그림 5. 프로젝트 1

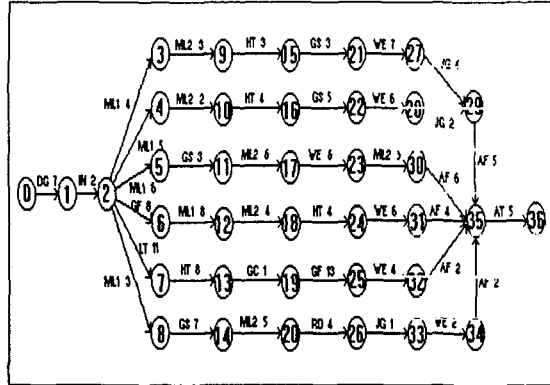


그림 6. 프로젝트 2

프로젝트 네트워크 상에 표시된 것과 같이 각각의 작업들은 영문자로 된 약어이름을 사용하며 표 2와 같다.

예약어 및 내용	
DG:	설계
IN:	재료 입고
ML1:	1차 밀링
ML2:	2차 밀링
GF:	성형 연삭
LT:	선반작업
GS:	평면 연삭
HT:	열처리
GC:	원통 연삭
WE:	와이어 컷트
RD:	레이디얼
JG:	지그볼링
AF:	사상
AT:	조립 및 실험

표 2. 예약어 및 내용

4.3 비교분석

본 절에서는 RCDMPS에서 목적함수가 총 처리시간의 최소화인 주어진 금형 가공 예에 대해 각각 모의실험을 통하여 비교하고자 한다. 모의실험은 프로젝트 1이 수주되어 일정계획된 후 15일 경과 후 프로젝트 2가 수주되었을 경우에 총 처리시간을 최소화하는 일정계획을 생성하는 것이다. 연구에서는 제시된 다양화 기법을 비교하고, 다양화 기법을 적용하여 좋은해를 생성한 기법과 하승진[2]이 제시한 MINLFT(Dynamic)휴리스틱 기법을 비교한다. 비교분석에 들어가기 전, RCDMPS를 위한 TS를 적용한 기법을 사용자 편의를 위해 구현한 프로그램을 설명하면, 그림 7은 RCDMPS 일정계획을 위한 주 메뉴 화면이고, 각각의 file 메뉴를 누르면 연계된 하위 메뉴가 나타나게 된다. 그림 8은 file 메뉴와 연계된 하위 메뉴이다.

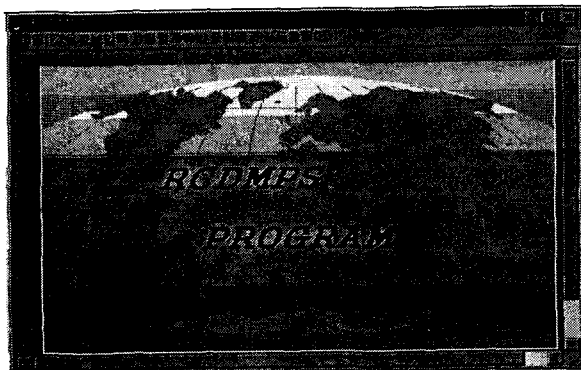


그림 7. 주 메뉴 화면



그림 8. 하위 메뉴

제안한 TS 일정계획의 알고리즘 흐름에 따라 그림 9와 같이 Data file로부터 프로젝트에 대한 정보 입력을 받아 FCFS 휴리스틱을 적용하여 그림 10의 초기해를 생성한다.

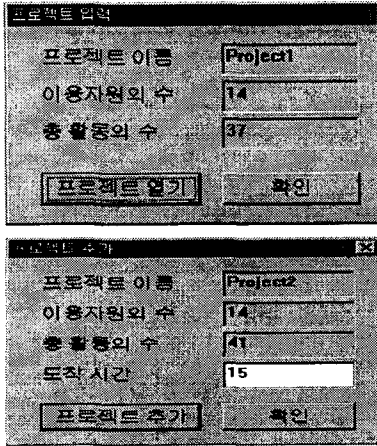


그림 9. 프로젝트 정보 입력화면

Activity	Duration	(Start, End)	Resource
(1, 2)	2	(8, 10)	DS
(1, 3)	2	(9, 11)	EN
(2, 4)	2	(10, 12)	ME.1
(2, 5)	5	(10, 15)	ME.1
(2, 6)	7	(10, 17)	ME.1
(2, 7)	8	(10, 18)	GF
(2, 7)	11	(8, 19)	LT
(2, 8)	3	(25, 28)	ME.1
(3, 9)	5	(18, 23)	ME.2
(4, 10)	3	(18, 21)	ME.2
(5, 11)	4	(25, 29)	CS
(6, 12)	8	(28, 36)	ME.1
(7, 13)	8	(19, 27)	HT
(8, 14)	7	(20, 27)	CS
(9, 15)	5	(27, 32)	HT
(10, 16)	4	(32, 36)	HT
(11, 17)	7	(40, 47)	ME.2
(12, 18)	7	(36, 43)	ME.2
(13, 19)	2	(27, 29)	GC
(14, 20)	6	(58, 64)	ME.2
(15, 21)	4	(36, 40)	CS
(16, 22)	6	(48, 54)	CS
(17, 23)	7	(50, 57)	WC
(18, 24)	5	(49, 54)	HT
(19, 25)	15	(29, 44)	GF
(20, 26)	6	(54, 60)	RD
(21, 27)	8	(57, 65)	WC
(22, 28)	8	(65, 73)	WC
(23, 29)	8	(57, 65)	ME.2
(24, 30)	5	(48, 53)	JC

그림 10. FCFS적용 초기해 생성화면

그림 11은 TS를 적용함에 있어 tabu 매개변수를 설정해주는 화면이고, 그림 12는 TS를 적용하여 생성된 일정계획에 대한 종합정보사항을 간략하게 보여주는 화면이다.

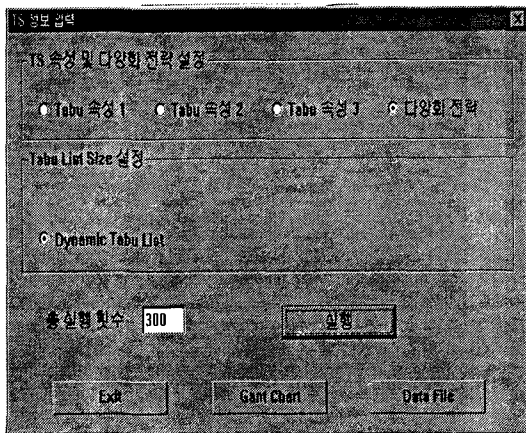


그림 11. TS 정보 입력화면

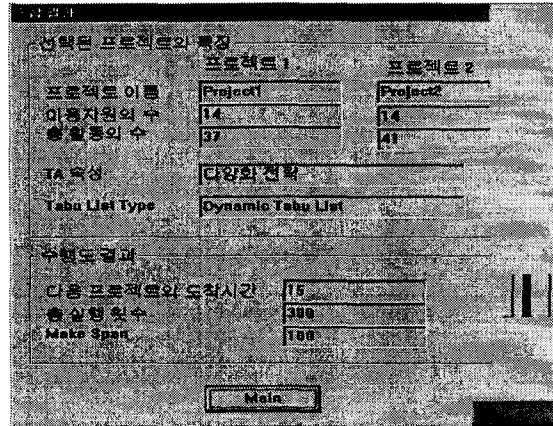




그림 12. 알고리즘 적용 후 종합 결과화면

4.3.1 다양화 기법

앞장에서 제시한 두 가지 다양화 방법을 비교한다. 첫째 방법은 단순 TS를 반복 적용(이후 반복 단순 TS라 함)하는 방법으로 단순히 초기해를 달리하여 단순 TS를 반복하는 방법으로, 하나의 초기해에 대해 일정회수(실험에서는 이웃해 생성회수)만큼 이동한 후 새로운 초기해로 다시 시작하는 과정을 종료조건(실행회수 300회)을 만족할 때까지 반복하는 것으로, 사용되는 tabu 속성은 제시한 속성 중 이동하는 활동만을 고려하는 속성 TA1을 사용한다, 둘째 방법은 장기기억 함수를 이용하여 tabu 속성을 변화시키면서 적용하는 방법(이후 다양화 전략 적용 TS라 함)으로 TS에서 임의의 초기해와 tabu 속성 TA1로부터 시작하여 일정회수(해를 개선하지 못한 회수, 실험에서는 20회)동안 해의 개선이 이루어지지 않으면 지금까지의 가장 좋은 해를 현재해로 하여 이 해로부터 초기해를 구하고 속성을 변화시키면서(속성 TA2, 속성 TA3) 다시 시작하는 방법으로 종료조건(실행회수 300회)이 만족될 때까지 반복하는 방법이다.

실험결과 그림 13은 반복 단순 TS에 의해 생성된 gantt chart이고, 그림 14는 다양화 전략 적용 TS에 의해 생성된 gantt chart이다. 생성된 해의 총 처리시간은 반복 단순 TS는 101일이 고, 다양화 전략 적용 TS는 100일을 생성하여, 다양화 전략 적용 TS 기법이 단순 TS를 반복 적용하는 기법보다는 해를 더 개선하였다. 그림에서 은 프로젝트 1, 은 프로젝트 2이다.

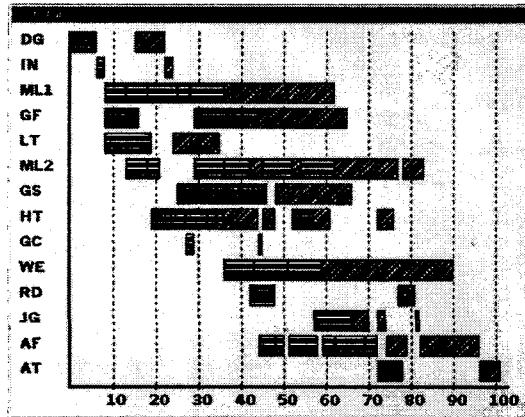


그림 13. 반복 단순 TS

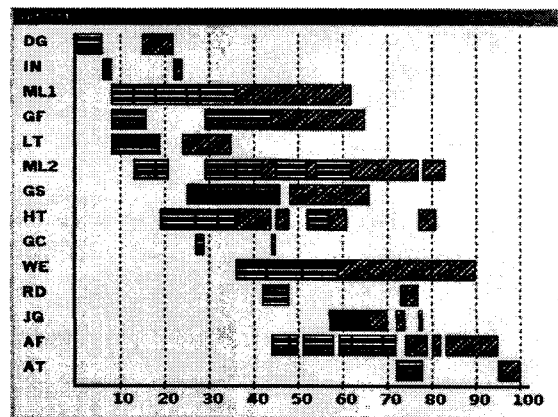
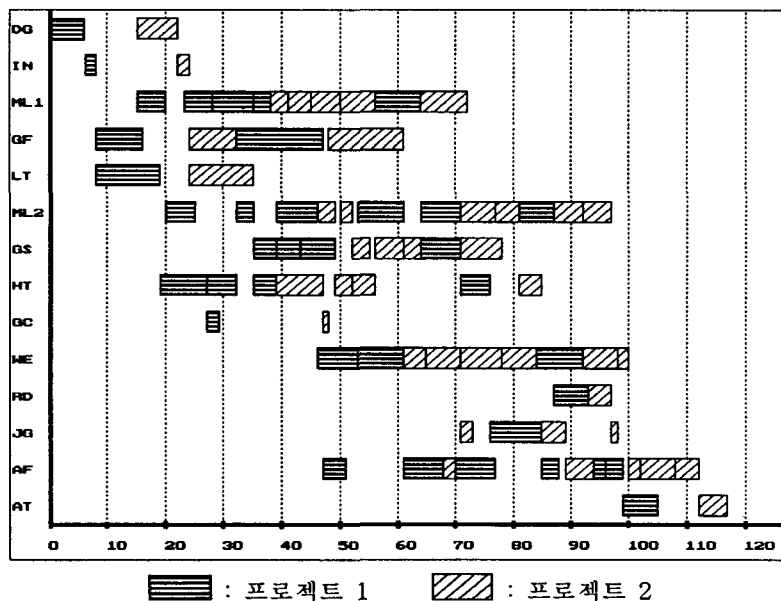


그림 14. 다양화 전략 적용 TS

4.3.2 작업배분 규칙과의 비교

S사의 실제 문제인 프로젝트 1, 2가 동적인 상황에서 수주된 경우 제안한 TS에 의한 일정계획 알고리즘을 적용 시 다양화 전략을 이용한 TS 알고리즘과, MINLFT(Dynamic)규칙에 의해 생성된 일정계획을 비교한다.

모의 실험결과 프로젝트 1이 수주된 후에 프로젝트 2가 15일 경과 후 수주되었을 경우 MINLFT(Dynamic) 규칙으로 생성된 일정계획의 총 처리시간은 117일이 되며, 본 연구에서 제시한 다양화 전략 적용 TS로 생성된 일정계획은 100일로 총 처리시간 최소화 면에서 해의 개선이 이루어 졌다. 그림 15는 MINLFT(Dynamic)규칙에 의해 생성된 gantt chart이다.



 : 프로젝트 1  : 프로젝트 2

그림 15. MINLFT(Dynamic) 휴리스틱 적용 일정계획

5. 결 론

본 연구에서는 프로젝트 네트워크가 동적인 상황에서 다중으로 수주되었을 경우 생성되는 일정계획 문제에서 FCFS 휴리스틱 규칙으로 초기해를 생성하고 해의 개선 알고리즘인 TS를 적용하여 실험한 결과 기존의 MINLFT(Dynamic) 휴리스틱 규칙보다 해의 개선을 이루었다.

또한, TS를 적용하는 단계에서 다양화 전략을 적용하여 해의 다양한 탐색을 유도하였고, 이를 위해 각 실행회수 마다 주어진 프로젝트 문제의 복잡성에 따라 tabu 목록의 크기를 구하고 갱신하여 사용하는 동적 타부목록의 관리를 이용하였다. 그리고 다양한 해의 탐색을 해나가는 다른 방법으로 일정한 회수만큼 해의 개선이 이루어지지 않으면 tabu 제약을 강화하여 해의 탐색을 유도하고자 tabu 속성을 다양하게 적용하였다. 적용결과 총 처리시간의 최소화면에서 효율적인 해의 개선이 이루어짐을 알 수 있었다.

또한, 본 연구와 사용자 편의(GUI 환경)를 위해 범용 프로그래밍 언어인 Microsoft Visual Basic 6.0을 사용하여 구현하였다.

참 고 문 헌

- [1] 이덕성, 배상윤, 김여근, "Job Shop 일정계획을 위한 Tabu Search", 대한산업공학회지, 제 21권 3호, 1995.
- [2] 하승진, "자원제약하의 동적 다중 프로젝트 일정계획에 관한 연구", 인하대학교 석사 학위 논문, 1995.
- [3] Belarmino Adenso-Diaz, "Restricted neighborhood in the tabu search for the flow shop problem", *European Journal of Operational Research*, vol.62, pp.27-37, 1992.
- [4] Fred Glover, "Tabu Search-Part I", *ORSA Journal on Computing*, vol.1, no.3, pp.190-206, 1989.
- [5] Fred Glover, "Tabu Search-Part II", *ORSA Journal on Computing*, vol.2, no.1, pp.4-32, 1990.
- [6] Fred Glover, "Tabu Search : A Tutorial", *Interfaces*, vol.20, no.4, pp.74-94, 1990.
- [7] Fred Glover, "A user's guide to tabu search", *Annals of Operation Research*, vol41, pp.3-28, 1993.
- [8] I. Kurtulus and E. W. Davis, "Multi-Project scheduling : categorization of heuristic rules performance", *Management Science*, vol.28, no.2, pp.161-172, 1982.
- [9] Jae-Kwan Lee and Yeong-Dae Kim, "Search Heuristic for Resource Constrained Project Scheduling", *Journal of the Operational Research Society*, vol47, pp.678-689, 1996.
- [10] John Dumond and Vincent A. Mabert, "Evaluating project scheduling and due date assignment procedures : an experimental analysis", *Management Science*, vol.34, no.1, pp.101-118, 1988.
- [11] J. Wesley Barnes and Manuel Laguna, "A tabu search experience in production scheduling", *Annals of Operation Research*, vol.41, pp.141-156, 1993.
- [12] Laguna. M, Barnes. J. W and Glover. F, "Tabu search methods for a single machine scheduling problem", *Journal of Intelligent Manufacturing*, vol.2, pp.63-74, 1991.
- [13] Paul R. Thomas and Said Salhi, "A Tabu Search Approach for the Resource Constrained Project Scheduling Problem", *Journal of Heuristics*, 4: pp.123-139, 1998.
- [14] R. P. Mohanty and M. K. Siddiq, "Multiple Projects_multiple resources constrained scheduling : some studies", *INT. J. PROD. RES.*, vol27, no2, pp.261-280, 1989.