김일성종합대학학보 JOURNAL OF **KIM IL SUNG** UNIVERSITY
(자연과학) (NATURAL SCIENCE)
주체104(2015)년 제61권 제10호 Vol. 61 No. 10 JUCHE104(2015).

거리행렬계산의 가속화에 의한 MRCPSP/max의 풀이알고리듬의 성능개선

문경호, 김정훈

경애하는 **김정은**동지께서는 다음과 같이 말씀하시였다.

《과학기술을 확고히 앞세우고 과학기술과 생산을 밀착시키며 경제건설에서 제기되는 모든 문제들을 과학기술적으로 풀어나가는 기풍을 세워 나라의 경제발전을 과학기술적으로 확고히 담보하여야 합니다.》

쌍을 이루는 활동들사이의 최소 및 최대시간지연이 선정된 방식들에 의하여 변할수 있는 일반화된 선후관계를 가지는 다중방식 프로젝트일정작성문제(MRCPSP/max)[1, 2]에 대하여 가상문제에 기초한 풀이방법[1]이 있다.

론문에서는 가상문제의 특성에 기초하여 시간실행가능한 가상일정방안의 존재를 판단하는데 리용되는 거리행렬계산을 가속화함으로써 MRCPSP/max를 푸는 기본알고리듬의 성능을 개선하는 한가지 방법에 대하여 론의한다.

1. 가상문제[1]와 시간해석

MRCPSP/max는 망 $N=\{V,\ E;\ \Delta\}$으로 표현되는데 $V=\{0,\ 1,\ \cdots,\ n,\ n+1\}$은 마디점들의 모임, E는 호들의 모임, $\Delta=(\delta_{ijm_im_j})_{m_i\in M_i,\ m_j\in M_j}$는 호 $(i,\ j)$의 무게행렬이다.

MRCPSP/max 의 어떤 실체 (P)와 준위 $l-1$의 가상일정방안 $(\hat{M},\ \hat{S})_{l-1}$이 주어지고 또 활동 $l(l=\arg\min_i \hat{s}_i)$이 실제방식 m_l로 수행될 때 활동 l의 실제 시작시간 s_l의 최소화문제는 다음과 같은 가상문제 $(\hat{P})_l$로 정의된다.

$$\min \hat{s}_{n+1} \tag{1}$$

$$\text{s.t.}\quad \left.\begin{aligned} &\hat{s}_i+\delta_{ij\hat{m}_i\hat{m}_j}\le \hat{s}_j,\ \ \forall(i,\ j)\in E,\ \ i,\ \ j\in\overline{V} \\ &\hat{s}_i+\delta_{ijm_i\hat{m}_j}\le \hat{s}_j,\ \ \forall(i,\ j)\in E,\ \ i\in V\setminus\overline{V},\ \ j\in\overline{V} \\ &\hat{s}_i+\delta_{ijm_im_j}\le \hat{s}_j,\ \ \forall(i,\ j)\in E,\ \ i,\ \ j\in V\setminus\overline{V} \end{aligned}\right\} \tag{2}$$

$$r^{\rho}_{lm_lk}\le R^{\rho}_k-\sum_{i\in A(t)} r^{\rho}_{im_ik},\quad \forall k\in R^{\rho},\ \ t\ge s_{l-1}, \tag{3}$$

$$r^{v}_{lm_lk}\le R^{v}_k-\left(\sum_{i\in\overline{V}} r^{v}_{i\hat{m}_ik}+\sum_{i\in V\setminus\overline{V}\setminus\{l\}} r^{v}_{im_ik}\right),\quad \forall k\in R^{v}, \tag{4}$$

$$\hat{s}_i=s_i,\ \ \forall i\in V\setminus\overline{V}\setminus\{l\} \tag{5}$$

여기서 m_i, s_i, $\hat{m}_i$, $\hat{s}_i$ 은 각각 활동 i의 실제방식, 실제시작시간, 가상방식, 가상시작시간이며 $\overline{V}=\{\overline{l+1,\ n+1}\}$ 는 가상방식으로 수행되는 활동들의 모임이고 $A(t)=\{i\,|\,s_i \le t < s_i + p_{im_i},\ \forall i \in V \setminus \overline{V} \setminus \{l\}\}$ 는 가상일정방안 $(\hat{M},\ \hat{S})_{l-1}$ 에 대하여 시간 t에서 수행된 활동들의 모임이다. 또한 R_k^ρ 는 재생가능한 자원 $k \in R^\rho$ 의 량, R_k^ν 는 재생불가능한 자원 $k \in R^\nu$ 의 량이며 $r_{im_ik}^\rho$ 는 활동 i가 방식 m_i 에서 수행될 때 요구하는 재생가능한 자원 k의 량, $r_{im_ik}^\nu$ 는 활동들이 소비하는 재생불가능한 자원 k의 전체 량이다.

목적은 제한 (1)－(5)가 만족되고 $\hat{s}_{n+1}$ 이 최소화되게

$$\hat{M}=(m_0,\ \cdots,\ m_l,\ \hat{m}_{l+1},\ \cdots,\ \hat{m}_{n+1})$$

이고 $\hat{S}=(s_0,\ \cdots,\ s_l,\ \hat{s}'_{l+1},\ \cdots,\ \hat{s}'_{n+1})$ 인 새로운 가상일정방안 $(\hat{M},\ \hat{S})_l$ 을 결정하는것이다. $(\hat{P})_l$ 의 망을 $\hat{N}=\{V,\ E;\ \hat{\Delta}\}$ 으로 표기한다.

식 (1)－(5)로 표현되는 가상문제 $(\hat{P})_l$ 은 잘 알려진 전형적인 MRCPSP/max[3]이며 따라서 이것에 대한 시간해석의 수법 즉 시간－실행가능한 최속시작일정방안을 계산함으로써 풀수 있다.

식 (1), (2), (5)를 만족시키는 최속시작시간 $\hat{s}_i(\forall i \in V)$ 는 최속시작시간벡토르 $\hat{E}\hat{S}=(s_0,\ \cdots,\ s_l,\ \hat{e}\hat{s}_l,\ \cdots,\ \hat{e}\hat{s}_{n+1})$ 와 시간－실행가능한 가상적인 최속시작일정방안 $(\hat{M},\ \hat{E}\hat{S})_l$ 을 형성한다. 따라서 $\hat{E}\hat{S}$ 의 계산은 $(\hat{M},\ \hat{E}\hat{S})_l$ 의 존재에 대한 검사와 관련된다.

한편 활동 i의 최속시작시간 $\hat{e}\hat{s}_i$ 는 $(\hat{P})_l$ 에 대응하는 망 $\hat{N}$ 우의 절점 0으로부터 절점 i까지의 최대경로를 찾아서 계산할수 있다. $\hat{N}$ 이 길이가 정인 주기를 가지지 않을 때에만 $\hat{N}$ 에 대한 시간－실행가능한 가상적인 일정방안이 존재[6]한다는것은 알려져있다. 그러므로 최대경로길이행렬 $\hat{D}=(\hat{d}_{ij})_{i\in V,\ j\in V}$ 를 계산한다면 절점 i 로부터 자기자체로 돌아오는 정의 경로길이는 길이가 정인 주기의 존재를 나타내며 결과적으로 시간－실행가능한 가상일정방안의 비존재를 나타낸다. 여기서 $\hat{d}_{ij}$ 는 $\hat{N}$ 의 최대경로길이를 나타냄과 동시에 활동 i 와 j 의 최대거리를 나타낸다.

$\hat{D}$ 은 시간복잡성이 $O(|V|^3)$ 인 다음과 같은 플로이드－와르샬알고리듬[3]에 의하여 계산할수 있다.

$$\hat{D}(0)=(\hat{d}_{ij}(0))_{i\in V,\ j\in V},$$

$$\hat{d}_{ij}(0)=\begin{cases} 0, & i=j \\ \delta_{ij\hat{m}_i\hat{m}_j}, & \forall(i,\ j)\in E \\ -\infty, & \text{그렇지 않을 때} \end{cases}$$

$$\hat{d}_{ij}(k)=\max\{\hat{d}_{ij}(k-1),\ \ \hat{d}_{ik}(k-1)+\hat{d}_{kj}(k-1)\},\ \ \forall i,\ j,\ k\in V,$$

$$\hat{D}=\hat{D}(n+1)$$

다른 한편 만일 $\forall i\in V$ 에 대하여 $\hat{d}_{ii}=0$ ($\hat{D}$ 의 대각선요소들의 값)이고 $\forall i\in\{\overline{0,\ \ l-1}\}$ 에 대하여 $\hat{d}_{0j}=s_j$ 라면 시간－실행가능한 가상일정방안 $(\hat{M},\ \hat{E}\hat{S})_l$ 이 존재한다.

이때 $\hat{E}\hat{S}$는 $\hat{D}$의 첫행에 있는 값들로 주어진다. 즉

$$\hat{E}\hat{S} = (\hat{d}_{00},\ \hat{d}_{01},\ \cdots,\ \hat{d}_{0n},\ \hat{d}_{0n+1}).$$

그리고 $\hat{d}_{0j} \neq s_j$이거나 각이한 두 활동 i와 j에 대하여 $\hat{d}_{ij} + \hat{d}_{ji} > 0$이 성립한다면 $\hat{N}$ 안에 정의 길이의 주기가 생기며 결국 그 어떤 시간－실행가능한 가상적인 일정방안도 존재하지 않는다.

2. 기본알고리듬

단계 1 초기화

① $\overline{M}_j = M_j,\ \forall j \in V$

② $R_k^{\rho}(t) = R_k^{\rho},\ \forall k \in R^{\rho},\ t \geq 0;\ \hat{R}_k^{\nu} = \sum_{j \in V} \min_{m \in M_j} r_{jmk}^{\nu},\ \forall k \in R^{\nu}$

③ $\exists k \in R^{\nu},\ \hat{R}_k^{\nu} \geq R_k^{\nu}$이면 그때 알고리듬의 실행을 중지한다.

④ $\hat{M} = (\hat{m}_0,\ \hat{m}_1,\ \cdots,\ \hat{m}_{n+1});\ \hat{S} = (0,\ 0,\ \cdots,\ 0);\ (\hat{M},\ \hat{S})$;

$(\hat{M},\ \hat{S})$에 기초하는 망 $\hat{N}$의 $\hat{D}$을 계산한다.

⑤ $\exists j \in V,\ \hat{d}_{jj} > 0$이면 그때 알고리듬의 실행을 중지한다.

⑥ $l = 0;\ i = 0;\ m_0 = 1;\ s_0 = 0;\ UB = \infty$

$J(0) = \phi;\ a(0) = 0;\ \overline{V} = V \setminus \{0\};\ \hat{D}(0) = \hat{D}$

단계 2 적합한 모임의 계산

① $l = l + 1$

② $l = n + 1$이면 $UB = \hat{d}_{0,\ n+1}(n)$으로 하고 단계 7로 간다.

③ $J(l) = \{j \mid \min_{j \in \overline{V}} [\hat{d}_{0j}(l-1)]\}$

단계 3 적합한 활동의 선택

① $J(l) = \phi$이면 단계 7로 간다.

② $i \in J(l)$을 선택한다.

③ $J(l) = J(l) \setminus \{i\};\ a(l) = i;\ \overline{V} = \overline{V} \setminus \{i\}$

단계 4 방식의 선택

① $\overline{M}_i = \phi$이면 $\overline{V} = \overline{V} \cup \{i\};\ \overline{M}_i = M_i$로 하고 단계 3으로 간다.

② $m_i \in \overline{M}_i$를 선택한다.

③ $\overline{M}_i = \overline{M}_i \setminus \{m_i\}$

단계 5 재생불가능한 자원 및 시간실행가능성의 평가

① 만일 $\exists k \in R^{\nu},\ R_k^{\nu} - \min_{m \in M_i} r_{imk}^{\nu} + r_{im_ik}^{\nu} \geq R_k^{\nu}$이면 단계 4로 돌아간다.

② $\hat{D}$를 계산한다.

③ 만일 $\exists j \in (V \setminus \overline{V}) \cup \{i\},\ \hat{d}_{0j} > s_j$ 혹은 $\exists j \in V,\ \hat{d}_{jj} > 0$ 혹은 $LB > UB$이면 단계 4로 돌아간다.

단계 6 가장 빠른 시간 및 자원실행가능한 시작시간의 계산과 평가 그리고 자원용량들의 조정

① $s_i = \min\{s_i \mid r^{\rho}_{im_ik} \le R^{\rho}_k(t),\ s_i \le t < s_i + p_{im_i},\ s_i \ge \hat{d}_{0i}, \forall k \in R^{\rho}\}$

② $\delta_{0i1m_i} = s_i$

③ $\hat{D}$ 를 갱신한다.

④ 만일 $\exists j \in (V \setminus \overline{V}) \cup \{i\}$, $\hat{d}_{0j} > s_j$ 혹은 $\exists j \in V, \hat{d}_{jj} > 0$ 이거나 $LB > UB$ 이면 $\delta_{0i1m_i} = rd_i$ 로 하고 단계 4로 돌아간다.

⑤ 자원들의 용량을 조정한다.

⑥ $\hat{D}(l) = \hat{D}$

⑦ 단계 2로 돌아간다.

단계 7 론의되지 않은 활동들과 방식들에로의 되돌이

① $l = l - 1$

② $l = 0$ 이면 알고리듬의 실행을 중지한다.

③ $i = a(l)$

④ 활동 i 와 관련되는 자원들의 용량을 회복한다.

⑤ $\delta_{0i1m_i} = rd_i$

⑥ 단계 4로 돌아간다.

알고리듬에서 l은 준위를 나타내는 첨수, i는 현재 준위 l 에서 선정된 활동, m_i 는 현재 준위 l 에서 선정된 활동 i의 방식, $\overline{V}$ 는 실제방식에서 일정화되지 않은 활동들의 모임, $\overline{M_i}$ 는 아직 론의되지 않은 활동 i의 방식들의 모임, M_i 는 활동 i의 초기방식들의 모임, $J(l)$은 준위 l 에서 적합한 활동들의 모임, $a(l)$은 준위 l 에서 선택된 활동, $\hat{D}(l)$ 은 준위 l의 가상문제 $(\hat{P})_l$ 에 대응하는 망 $\hat{N}$ 의 거리행렬, $\hat{d}_{ij}(l)$ 은 $\hat{D}(l)$ 의 요소, 준위 l에서 활동 i와 j사이의 거리, $\hat{D}$ 은 망 $\hat{N}$ 의 림시거리행렬, $\hat{d}_{ij}$ 은 $\hat{D}$ 의 요소, 두 활동 i와 j사이의 거리, $R^{\rho}_k(t)$ 는 t에서 재생가능한 자원 k의 량, R^{ν}_k 는 재생불가능한 자원 k의 소비량이다.

3. 거리행렬계산의 가속화

기본알고리듬의 단계 5에서 림시거리행렬의 계산은 $o(|V|^3)$ 의 복잡성을 가지는데 이것은 지나치게 많은 시간을 소비한다.

보조정리 기본알고리듬에서, 준위 l에서 론의되는 활동과 관련되는 호들의 무게는 준위 $l-1$에서의 활동과 관련되는 호들의 무게보다 항상 작지 않다.

증명 활동 i 가 준위 $l-1$ 에서는 론의되지 않고 준위 l 에서 적합한 활동으로 선택되였다고 하자. 가상문제의 정의로부터 활동 i 는 준위 $l-1$ 에서는 가상방식 $\hat{m}_i$ 로 수행되고 준위 l 에서는 실제방식 m_i 로 수행된다. 한편 $(i, j) \in E$ 와 $(j, i) \in E$ 가 활동 i 에 관련되는 호라고 하자. 그러면 가상방식의 정의로부터 호 (i, j) 의 무게는 준위 $l-1$ 에서 $\delta_{ij\hat{m}_i\hat{m}_j}$ 이고 준위 l 에서는 $\delta_{ijm_i\hat{m}_j}$ 이다. 류사하게 호 (j, i) 의 무게는 준위 $l-1$ 에서 $\delta_{jim_j\hat{m}_i}$ 이고 준

위 l 에서는 $\delta_{jim_jm_i}$ 이다. 이때 $\delta_{ij\hat{m}_i\hat{m}_j} \le \delta_{ijm_i\hat{m}_j}$ 즉 $\min_{m_i \in M_i} \min_{m_j \in M_j} \delta_{ijm_im_j} \le \min_{m_j \in M_j} \delta_{ijm_im_j}$ 이고 $\delta_{jim_j\hat{m}_i} \le \delta_{jim_jm_i}$ 즉 $\min_{m_i \in M_i} \delta_{jim_jm_i} \le \delta_{jim_jm_i}$ 가 성립하는것은 명백하다. 이것은 단계 4에서 새롭게 생성된 가상문제인 $(\hat{P})_l$ 에 대응하는 망 $\hat{N}$ 의 호무게들이 준위 $l-1$ 의 $(\hat{P})_{l-1}$ 에 대응하는 망 $\hat{N}$ 의 호무게들보다 항상 작지 않다는것을 의미한다. 호무게의 변화는 단계 6에서도 일어나는데 $rd_i \le s_i$ 이므로 이것은 더이상 론의할 필요가 없다.(증명끝)

정리 1 기본알고리듬의 준위 l 에서 림시거리행렬 $\hat{D}$ 의 계산은

$$0 \sim o\left(|V|^2 \left(\sum_{(i,\ j)\in E} 1 + \sum_{(j,\ i)\in E} 1 \right)\right)$$

의 시간복잡성을 가진다.

증명 준위 $l-1$ 의 망 $\hat{N}$ 의 거리행렬 $\hat{D}(l-1)$ 이 미리 주어지고 활동 i 가 준위 l 의 적합한 활동으로 선택되였다고 한다. 이때 $\hat{D}(l-1)$ 을 리용하여 $\hat{D}$ 을 계산할수 있다. 만약 $(i,\ j)\in E$ 와 $(j,\ i)\in E$ 가 준위 l 의 활동 i 에 관련되는 호라고 한다면 보조정리로부터 이러한 호들의 무게 $\delta_{ijm_i\hat{m}_j}$ 와 $\delta_{jim_jm_i}$ 는 항상 감소하지 않으며 이것은 림시거리행렬 $\hat{D}$ 의 계산이 $\hat{d}_{ij}(l-1)$ 와 $\delta_{ijm_i\hat{m}_j}$ 그리고 $\hat{d}_{ji}(l-1)$ 와 $\delta_{jim_jm_i}$ 의 값들의 크기관계에 의존한다는것을 암시해준다. 여기서 $\hat{d}_{ij}(l-1)$ 은 $\hat{D}(l-1)$ 의 요소 즉 준위 $l-1$ 의 두 활동 i 와 j 사이의 거리이다.

먼저 $\hat{d}_{ij}(l-1) = \delta_{ijm_i\hat{m}_j}$ 이라면 이때 $\hat{D}$ 을 계산할수 있는데 계산시간은 0과 같다. 만일 $\hat{d}_{ij}(l-1) > \delta_{ijm_i\hat{m}_j}$ 이라면 $\hat{D}$ 을 계산할 필요가 없는데 이것은 준위 l의 망 $\hat{N}$ 의 두 절점 i 와 j 사이의 거리 $\hat{d}_{ij}$ 이 여전히 $\hat{d}_{ij}(l-1)$ 과 같기때문이며 따라서 $\hat{D}$ 의 다른 모든 요소들은 변화되지 않는다. 그러나 만일 $\hat{d}_{ij}(l-1) < \delta_{ijm_i\hat{m}_j}$ 라면 그때는 $\hat{D}$ 의 모든 요소들이 $o(|V|^2)$ 의 시간복잡성을 가지는 다음과 같은 재귀관계에 의하여 계산되여야 하는데 이것은 $\hat{d}_{ij}$ 이 $\hat{d}_{ij}(l-1)$ 보다 더 작게 되지 않기때문이다.

한편 준위 l의 활동 i 에 관련되는 호들의 총수가 $\left(\sum_{(i,\ j)\in E} 1 + \sum_{(j,\ i)\in E} 1 \right)$ 이므로 결국 $\hat{D}$ 의 계산시간이 $0 \sim o\left(|V|^2 \left(\sum_{(i,\ j)\in E} 1 + \sum_{(j,\ i)\in E} 1 \right)\right)$ 이라는것을 알수 있다.(증명끝)

정리 2 기본알고리듬에서 가상일정방안에 기초하는 프로젝트망 $\hat{N}$ 에 대하여 길이가 정인 주기의 존재성검사에 의하여 시간실행가능성을 평가하는것은 타당하다.

증명 이것을 증명하는것은 준위 l 의 $\hat{N}$ 에 길이가 정인 주기가 존재할 때 준위 $l+1$ 의 $\hat{N}$ 에도 여전히 길이가 정인 주기가 존재한다는것을 보여주는것에 귀착된다.

$C :< u,\ \cdots,\ v-1,\ v,\ v+1,\ \cdots,\ u >$ 를 준위 l 의 $\hat{N}$ 에 있는 길이가 정인 주기이고 2개의 경로 $uv :< u,\ \cdots,\ v-1,\ v >$ 와 $vu :< v,\ v+1,\ \cdots,\ u >$ 로 이루어지며 일반성을 잃지 않고

$$(d_{uv})_l > 0, (d_{vu})_l \le 0$$

이라고 하자. 여기서 $(d_{uv})_l$ 과 $(d_{vu})_l$ 은 각각 준위 l 의 $\hat{N}$ 에 있는 경로 uv 와 vu 의 길이이다. 그러면 부등식 $(d_{uv})_l + (d_{vu})_l > 0$ 이 성립한다.

i 를 준위 l 에서는 가상방식 $\hat{m}_i$ 로 일정화되였고 준위 $l+1$ 에서는 실제방식으로 일정화되는 활동이라고 하자. 활동 i 가 $(\hat{P})_{l+1}$ 의 $\hat{N}$ 에 있는 경로 uv 우의 요소라면 이때 보조정리와 정리 1로부터 활동 i 에 관련되는 활동들사이의 호들의 무게와 길이(거리)는 작아지지 않으며 따라서 $(d_{uv})_{l+1} \ge (d_{uv})_l$ 이 항상 성립한다.

마찬가지로 활동 i 가 경로 vu 우의 요소일 때에도 $(d_{vu})_{l+1} \ge (d_{vu})_l$ 즉

$$|(d_{vu})_{l+1}| \le |(d_{vu})_l|$$

이 항상 성립한다.

$(d_{uv})_{l+1} + (d_{vu})_{l+1} > 0$ 이 성립하기때문에 결국 준위 $l+1$ 의 $\hat{N}$ 에 길이가 정인 주기가 여전히 존재한다고 볼수 있다.(증명끝)

4. 계산실험 및 결과분석

시험문제로 문제서고[1]에 있는 270개의 MRCPSP/max실체들로 하였는데 매 실체는 100개의 활동들로 이루어져있다.

론문에서 제기한 알고리듬(IES-PT)과 TS_{DRH98} [4], PR_{H01} [5], TS_{NI03} [6]들과의 성능을 비교한 결과는 표와 같다.

표. 론문에서 제기한 알고리듬과 선행한 알고리듬들과의 성능비교

알고리듬	최대시간한계/s	콤퓨터속도	실행가능풀이	평균분산/%
TS_{DRH98}	30	PC 333 MHz	95/270	284.07
PR_{H01}	30	PC 333 MHz	270/270	180.05
TS_{NI03}	10	PC 1.0 GHz	269/270	95.88
TS_{NI03}	200	PC 1.0 GHz	270/270	25.28
IES-PT	4	PC 1.4 GHz	270/270	37.31
IES-PT	100	PC 1.4 GHz	270/270	22.05

표에서 실행가능풀이는 알고리듬이 실행가능한 풀이들을 찾은 실체들의 수를 나타내며 평균분산은 주경로로 주어진 하계에 관하여 알고리듬의 평균분산이다.

4s나 100s 한계안에서 IES-PT는 다른 알고리듬들보다 성능이 좋다. 차이가 고속알고리듬만큼 크지 않지만 통계학적으로 보다 성능이 좋으며 시간한계와 처리기속도에서 론문의 알고리듬이 다른 알고리듬들보다 더 적은 시간을 요구한다.

맺 는 말

현재 준위에서의 활동과 관련되는 호들의 무게가 앞준위에서의 활동과 관련되는 호들의 무게보다 항상 작지 않다는 가상문제의 특성에 기초하여 시간실행가능한 가상일정방안의 존재를 판단하는데 리용되는 거리행렬계산을 가속화함으로써 MRCPSP/max를 푸는 기본알고리듬의 성능을 개선하였다.

참 고 문 헌

[1] **김일성**종합대학학보(자연과학), 61, 8, 37, 주체104(2015).

[2] Agustin Barrios et al.; Computers & Operations Research, 38, 33, 2011.

[3] M. Bartuch et al.; Annals of Operations Research, 16, 201, 1998.

[4] Bert De Reyck et al.; European Journal of Operational Research, 111, 152, 1998.

[5] R. Heilmann; OR Spektrum, 23, 335, 2001.

[6] K. Nonobe et al.; In: MIC2003: The 15th Metaheuristics International Conference, 55, 2003.

주체104(2015)년 6월 5일 원고접수

Improvement of the Performance of Solution Algorithm for MRCPSP/max with Acceleration of the Distance Matrix Computation

Mun Kyong Ho, Kim Jong Hun

We speeded up the solution algorithm by reducing the computation time of the distance matrix based on the property of the virtual subproblem that the weights of arcs on current level in basic algorithm don't become smaller than those on previous level.

Key words: multi-mode project scheduling, minimal time delay