經營科學 第12卷 第2號
1995年 8月

작업준비시간을 고려한 총작업완료시간 최소화 해법에 관한 연구

안상형*

A Study on Algorithms to Minimize Makespan of Sequence-dependent Jobs

Sang Hyung Ahn*

ABSTRACT

In this paper we develop an efficient heuristic algorithm for the problem of scheduling n sequence-dependent jobs on a basic processor to minimize makespan. Efficient solution methods are already known for the sequence-independent case. But for the sequence-dependent case, this problem belongs to a set of strong NP-complete problems.

We present a heuristic which is similar to shortest setup time heuristic but opportunity cost of setup time rather than shortest setup time is used for choosing next job. This heuristic algorithm has same computational complexity and worst case ratio as the shortest setup time heuristic. We used Wilcoxon signed rank test to show that our heuristic is superior to nearest setup time heuristic in term of average behavior.

1. 序言

日程計劃(scheduling)을 수립하는 데 있어서의 기본적인 문제중의 하나는 처리 기계상에서 總作業完了時間(makespan)을 최소화하는 문제이다. 여기서 기본적인 문제라 함은 1회 가공(single operation)을 요하는 다수의 작업들이 시스템 내에 대기하고 있을 때 일정한 작업 처리순서를 정하여 특정 성과측도를 최소화하는 경우의 문제를 의미한다. 이 경우에 작업준비시간(setup time)

* 서울대학교 경영대학 교수

은 작업 처리순서에 의해 영향을 받지 않는 것으로 가정되고 있다. 이러한 기본적인 문제의 확장으로 작업준비시간(setup time)이 작업 처리순서에 따라 변하는 경우에 총작업완료시간을 최소화하는 문제가 있다. 그리고 여기서 성과측도가 되는 총작업완료시간은 최초작업의 착수로부터 마지막 작업이 완료되는 시점까지의 가공소요시간을 일컫는다.

이 문제는 생산현장에서 범용기계로 일군의 작업들을 처리하거나, 하나의 기계로 다양한 물건을 생산할 때 최소 시간 내에 처리나 생산을 끝마치기 위한 처리순서 또는 생산순서를 결정하는 문제로 나타난다. 지금까지 생산관리 분야에서 이에 대한 많은 연구가 진행되어 왔으나, 이 문제가 가지고 있는 기본적인 속성, 즉 NP-Completeness의 속성으로 말미암아 효과적으로 최적해를 구하기가 지극히 힘들다. 때문에 대부분의 연구는 準最適解를 구하는 휴리스틱해법의 개발이나, 準最適解의 엄밀성을 분석하는 最惡限界分析, 平均形態分析, 確率的分析등이 주류를 이루고 있다.

작업준비시간을 고려한 總作業完了時間 최소화 문제를 해결하기 위해 개발된 휴리스틱 알고리즘은 크게 세 範疇로 분류된다. 첫째는 모든 작업의 처리순서를 순차적으로 결정하는 알고리즘, 둘째는 모든 작업의 처리순서가 주어졌을 때 처리순서를 變形시킴으로써 總作業完了時間을 단축시키는 알고리즘, 셋째는 모든 작업의 처리순서를 순차적으로 결정하는 알고리즘과 처리순서를 改善하는 알고리즘을 混合한 알고리즘이다.

最短 作業準備時間法(Nearest Setup Time Algorithm)은 모든 작업의 처리순서를 순차적으로 결정하는 알고리즘으로 計算의 複雜度가 낮으면서도 最適解에 近似한 效率的인 알고리즘으로 알려져 있다.

본 논문에서는 機會費用의 概念을 導入하여 最短 作業準備時間法보다 計算의 複雜度가 높지 않으면서 遂行結果를 改善시킨 휴리스틱 알고리즘을 제시하고자 한다. 또한, 이 알고리즘의 효율성과 개선정도를 통계적 처리에 의한 비교분석을 통하여 입증하고자 한다.

2. 문제의 정의 및 모형의 구성

2.1 문제의 개요

작업준비시간이 작업순서에 따라 달라질 때 總作業完了時間의 최소화 문제에 대한 보다 명확한 정의를 위하여 어떤 시스템 내에 처리시간이 t_i인 n개의 작업(i=1, 2, …, n)과 처리기계가 대기하고 있다고 가정한다. 이때 처리순서가 i번째인 작업의 처리시간을 $t_{[i]}$, i번째 작업을 끝마치고 j번째 작업을 처리하기 위한 준비시간을 $s_{[i][j]}$, 처리순서가 i번째인 작업이 시스템 내에 있는 시간을 $F_{[i]}$라 하자.

처리순서에 따른 $F_{[i]}$는 다음과 같다.

$$
\begin{aligned}
F_{[1]} &= s_{0,[1]}+t_{[1]} \\
F_{[2]} &= s_{[1],[2]}+t_{[2]}+F_{[1]} \\
F_{[n-1]} &= s_{[n-2],[n-1]}+t_{[n-1]}+F_{[n-2]} \\
F_{[n]} &= s_{[n-1],[n]}+t_{[n]}+F_{[n-1]}
\end{aligned}
$$

總作業完了時間은 최종 처리되는 n번째 작업이 시스템을 떠나는 시간이므로 종료상태를 n+1이라 하면 總作業完了時間을 M이라 하면 다음과 같다.

$$
\begin{aligned}
M &= F_{[n]}+s_{[n],[n+1]} \\
&= \sum_{j=1}^{n+1} s_{[j-1],[j]} + \sum_{j=1}^{n} t_j
\end{aligned}
$$

두 번째 합계는 모든 작업의 처리시간을 합한

것으로서 일정하므로 總作業完了時間의 최소화는 첫 합계를 최소화함으로써 얻을 수 있다.

이러한 總作業完了時間 최소화 문제의 형태는 종종 해밀턴經路問題(Hamiltonian Path), 또는 販賣員 巡廻問題(Traveling Salesman Problem)로 해석되기도 한다. 전통적인 설명에서는 판매원이 n개 도시에 거주하는 고객 모두를 반드시 한 번씩만 방문해야 하며, 이때 판매원은 도시들 간의 거리를 고려하여 다시 원래의 자리로 돌아 올 수 있는 최소의 巡廻經路를 택해야 한다. 日程問題의 $s_{[i][j]}$는 販賣員 巡廻問題의 도시 i에서 도시 j까지의 거리에 해당된다.

2.2 수리모형

앞 절에서 보인 것과 같이 작업준비시간을 고려한 경우의 總作業完了時間 최소화 문제는 해밀턴經路問題로 정형화된다. 이 경우 해밀턴經路問題는 方向이 있는 그래프 G=(V, E)에 의해 表現된다. 여기서 마디의 集合 V는 처리해야 할 작업의 集合이며, 작업 i가 작업 j보다 먼저 처리될 수 있을 경우에만 가지 (i, j)가 存在한다. 해밀턴經路는 그래프 위의 모든 마디를 重複이나 빠짐없이 巡廻하는 經路를 의미한다. 해밀턴經路는 $T=(i_1, i_2, \cdots, i_n)$으로 나타내고 s_{ij}는 (i, j)의 距離, e(t)는 해밀턴經路 t에 속하는 마디의 集合을 나타낸다. 변수 X_{ij}를 작업 i가 처리순서에서 작업 j를 선행하면 $X_{ij}=1$, 선행하지 않으면 $X_{ij}=0$으로 정의하면 작업준비시간을 고려한 경우의 總作業完了時間 問題는 다음과 같이 해밀턴經路問題로 定式化된다.

$$\text{Min } Z(t) = \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} x_{ij}$$

$$\sum_{j=1}^{n} x_{ij}=1, \ \forall\ i=1, \cdots, n$$

$$\sum_{i=1}^{n} x_{ij}=1, \ \forall\ j=1, \cdots, n$$

$$\sum_{i\in Q} \sum_{j\in Q'} X_{ij} \ge 1, \qquad Q \subseteq V,\ Q \neq \phi,\ (\text{단, } Q'=V-Q)$$

$$x_{ij} \in \{0, 1\} \qquad i, j=1, \cdots, n$$

2.3 휴리스틱

작업준비시간을 고려한 總作業完了時間 최소화 문제는 NP-complete문제로서 아직까지 多項時間 알고리즘은 存在하지 않는 것으로 여겨진다. 따라서 작업준비시간을 고려한 總作業完了時間 최소화 문제를 해결하기 위해 최적해를 보장하지는 못하지만 만족할만한 적정 해를 보장해 주는 다항시간 알고리즘인 휴리스틱 알고리즘이 많이 사용되고 있다. 작업시간을 고려한 總作業完了時間 최소화 문제 해결을 위한 휴리스틱 알고리즘은 크게 세 範疇로 분류된다. 첫째는 모든 작업의 처리순서를 순차적으로 결정하는 알고리즘, 둘째는 모든 작업의 처리순서가 주어졌을 때 처리순서를 變形시킴으로써 總作業完了時間을 단축하시키는 알고리즘, 셋째는 모든 작업의 처리순서를 순차적으로 결정하는 알고리즘과 처리순서를 改善하는 알고리즘을 混合한 알고리즘이다.

첫 번째 범주의 알고리즘으로는 任意的 挿入方法, Convex hull 挿入方法, 最大 角度 挿入 方法

등이 있으며 本 硏究에서 比較分析할 最短 作業準備時間法과 새로운 휴리스틱 알고리즘도 이 범주에 속한다.

둘째 범주의 알고리즘은 주어진 작업의 처리순서를 변형시킴으로써 단축된 총작업완료시간을 가지는 작업처리순서로 개선시키는 알고리즘이다. 가장 대표적인 알고리즘으로는 작업 交換方法이 있으며 한번에 몇 개의 마디를 交換하느냐에 따라 2-Opt, 3-Opt 등으로 區分된다.

세 번째는 위의 두 가지 알고리즘을 結合하여 먼저 모든 작업의 처리순서를 순차적으로 결정한 뒤 만들어진 작업 처리순서를 改善시켜 나가는 알고리즘으로 가장 多樣하고 效率性도 높은 편이다.

本 硏究에서 比較分析할 最短 作業準備時間法과 새로운 휴리스틱은 첫 번째 범주의 알고리즘으로 부분 작업 처리순서에 아직까지 처리순서가 정해지지 않은 새로운 작업을 포함시켜 가면서 전체 작업을 포함하는 작업 처리순서를 완성한다. 이때 부분 작업 처리순서에 포함될 작업의 선정 기준으로 最短 作業準備時間法은 最短 작업준비시간을 사용하지만 새 휴리스틱에서는 다음과 같은 機會費用을 선정 기준으로 사용한다.

휴리스틱의 演算過程에서 모든 작업은 아래의 네 가지 범주 중 하나에 속하게 된다. 첫째는 작업 처리순서에서 앞 작업과 뒤 작업이 확정되어 이미 부분 작업 처리 순서의 內部에 속하는 작업들이며, 둘째는 앞 작업은 확정되어 있으나 뒤 작업은 확정되지 않은 작업들이며, 셋째는 反對로 앞 작업은 확정되어 있지 않으나 뒤 작업은 확정되어 있는 작업들이고, 넷째는 앞 작업 및 뒤 작업도 확정되지 않은 작업들로서 부분 작업 처리순서에 包含되어 있지 않은 작업들이다.

첫 번째 경우는 이미 부분 작업 처리순서가 완료되었으므로 고려의 대상에서 제외되며 나머지 경우에 대해서만 고려된다.

작업 j가 둘째 경우에 속한다고 假定하자. 작업 k와 작업 k'가 j뒤에 처리될 수 있는 작업 중에서 첫 번째, 두 번째로 작업 준비시간이 짧다고 하자. 그러면 작업 j에 작업 준비시간이 가장 짧은 작업을 連結하지 않음으로 해서 發生되는 機會費用은 $OL(j)=S_{jk'}-S_{jk}$로 計算된다.

이번에는 작업 j가 세 번째 境遇에 속한다고 하자. 이때 작업 i와 작업 i'가 작업 j앞에 배치될 수 있는 작업 중에서 첫 번째, 두 번째로 짧은 작업 준비시간을 가지고 있다면 작업 j를 最短 작업 준비시간을 가지는 작업 뒤에 배치하지 않음으로서 發生하게 되는 機會費用은 $OL(j)=S_{i'j}-S_{ij}$로 나타낸다.

작업 j가 네 번째 境遇에 속한다면 작업 j의 앞이나 뒤에 작업을 배치할 수 있다. 따라서 最短 작업 준비시간을 가지는 작업과 連結하지 않음으로 인해 發生하는 機會費用은 $OL(j)=\max\{S_{i'j}-S_{ij},\ S_{jk'}-S_{jk}\}$로 나타낼 수 있다.

2.4 最短 作業準備時間法(nearest setup time method)

最短 作業準備時間法은 현재 부분 완성된 작업 처리순서를 擴張시켜 나갈 때 아직까지 처리순서가 확정되지 않은 작업 중에서 작업 준비시간이 가장 짧은 작업을 包含시켜나감으로써 모든 작업의 처리순서를 결정하는 방법이다. 이 방법은 Karg와 Thompson에 의해 最初로 開發되었으며 다른 휴리스틱 알고리즘에 비해 상당히 單純한 알고리즘 構造를 가지고 있으나 演算結果는 그다지 양호한 편은 아니다.

1) 알고리즘

작업이 n개인 문제의 경우 各 작업을 첫 처리

작업으로 하여 작업준비시간이 가장 작은 작업을 다음 처리작업으로 連結시켜나감으로써 모든 작업의 처리순서를 形成하고 形成된 n개의 모든 작업의 처리순서 중 최소의 총작업완료시간을 가지는 처리 순서를 選擇한다. m번째 反復演算에서 形成된 부분 작업 처리순서를 T_m이라 하고 $i(T_m)$, $j(T_m)$을 各各 작업 처리순서 T_m이 시작되는 작업과 끝나는 작업이라 하며 $M(T_m)$을 부분 작업 처리순서 T_m의 總作業完了時間이라 하면 알고리즘은 다음과 같다.

初期化段階 : $m := 0$, $\min := \infty$

1段階 : $m := m+1$

$T_m : \phi$, $i(T_m) := m$, $j(T_m) := m$

2段階로 간다.

2段階 : $M(e) := \min\{s(x, y) : x \notin T_m,\ y = i(T_m)$, 또는 $y \notin T_m,\ x = j(T_m)\}$

$T_m := T_m \cup \{e\}$

3段階로 간다.

3段階 : $|T_m| < n-1$이면, 2段階로 간다.

아니면, $T_m := T_m \cup \{j(T_m),\ i(T_m)\}$

4段階로 간다.

4段階 : $M(T_m) < \min$이면, $t := T_m$, $\min := M(T_m)$, 5段階로 간다.

아니면, 5段階로 간다.

5段階 : $m < n$이면, 1段階로 간다.

아니면 종료한다.

2) 時間複雜度函數

작업의 수가 n인 問題의 境遇에 각 단계의 時間複雜度는 다음과 같다.

1, 2段階 : $0(8n^2+4n)$

3段階 : 2段階를 $(n-1)$번 反復演算

4段階 : $0(2(n-1)+1)$

5段階 : 1段階부터 4段階까지 n번 反復演算

그러므로 時間複雜度函數는 $0(8n^4-11n^3-1)$이다.

時間複雜度函數는 $0(8n^4-11n^3-1)$이나 n이 충분히 큰 경우 $0(n^4)$으로 漸近的으로 支配되므로 最短 作業準備時間法의 時間複雜度函數는 $0(n^4)$이다.

3) 最惡의 境遇 比率(worst case ratio)

問題가 n개의 작업으로 이루어져 있고 $s_{(i,j)}$가 작업 i를 마친 후 작업 j를 시작하기 전의 작업준비시간이라고 할 때 다음과 같은 작업준비시간을 가지는 例題를 통해 最惡의 境遇 比率을 구하도록 한다.

$s_{(i,i+1)} = 1$, $\quad i \in N\text{-}\{2\}$

$s_{(2,1)} = 1$

$s_{(2,3)} = n\text{-}1$

$s_{(i,2)} = 1$, $\quad i \in N\text{-}\{2\}$

$s_{(3,i)} = 1$, $\quad i \in N\text{-}\{3\}$

$s_{(i,i-1)} = n$, $\quad i \in N\text{-}\{2, 3\}$

$s_{(i,i-2)} = n$, $\quad i \in N\text{-}\{3, 4\}$

이 問題이 最適 작업처리순서는 $T^*(1, 2, 3, \cdots, n, 1)$이고 이때의 最小 總作業完了時間은 $M(T^*)=2n-2$이다.

이 問題를 最短 作業準備時間法에 의해서 풀면 $m \notin \{1, 2, 3\}$이면 最善의 처리순서는 $T_m=(3, 2, m, 1, n, n-1, \cdots, m+1, \cdots, 4, 3)$이고 이때의 總作業完了時間은 $M(T_m)=n(n-3)+3$이 된다.

$m \in \{1, 2, 3\}$인 경우 最善의 처리순서는 $T_m=(3, 4, 2, 1, n-1, n-2, \cdots, 5, 4, 3)$이고 이때의 總作業完了時間 $M(T_m)=n(n-3)+3$이다.

그러므로 最短 作業準備時間法의 最惡의 境遇 比率 $R_n \geq (n(n-3)+3)/(2n-2)$이다. 따라서 n이 충분히 큰 경우, n/2에 근사됨을 알 수 있다.

3. 새로운 휴리스틱

앞에서 살펴보았듯이 最短 作業準備時間法은 알고리즘의 構造가 簡單한 장점을 가지고 있으나 演算結果는 그다지 좋지 못한 短點이 있다. 本 硏究에서는 基本構造는 最短 作業準備時間法의 單純性을 유지하면서 부분 작업처리순서를 擴張시킬때 작업준비시간이 가장 작은 작업을 連結시켜 나가는 방식을 止揚하고 절 2.3에서 도입한 機會費用 槪念을 도입함으로써 작업준비시간의 機會費用이 가장 큰 작업을 부분 작업처리순서에 포함시키는 方式을 통하여 演算結果를 改善시키고자 한다.

1.1 알고리즘

基本構造는 最短 作業準備時間法과 큰 차이가 없으나 부분 작업처리순서를 擴張시킬 때 連結되는 작업의 選擇基準을 단순히 最短 작업준비시간이 아니라 最大 작업준비시간의 機會費用으로 한다. 알고리즘에 사용되는 용어는 最短 作業準備時間法과 同一하다.

初期化段階 : $m := 0$, $\min := \infty$

1段階 : $m := m+1$

$T_m : \phi$, $i(T_m) := m$, $j(T_m) := m$

2段階로 간다.

2段階 : $Loss(e) := \min\{OL(k) : k \in i(T_m)$, 또는 $k \in j(T_m)\}$

$T_m := T_m \cup \{e\}$

3段階로 간다.

3段階 : $|T_m| < n-1$이면, 2段階로 간다.

아니면, $T_m := T_m \cup \{j(T_m), i(T_m)\}$

4段階로 간다.

4段階 : $M(T_m) < \min$이면, $T := T_m$, $\min := M(T_m)$, 5段階로 간다.

아니면, 5段階로 간다.

5段階 : $m < n$이면, 1段階로 간다.

아니면 종료한다.

1.2 時間複雜度函數

작업의 수가 n인 問題의 境遇에 단계의 時間複雜度는 다음과 같다.

2, 3段階 : $0((n-1)(6n^2+6n+1)+1$
$=0(6n^3-n)$

4段階 : 2, 3단계가 모두 n번의 反復演算을 하므로
$0(n(6n^3-n)$
$=0(6n^4-n^2)$

그러므로 새로운 휴리스틱 알고리즘은 $0(6n^4-n^2)$의 時間複雜度函數를 가진다. n이 충분히 클 경우 時間複雜度函數는 $0(n^4)$이 되어 最短 作業準備時間法과 同一한 次數(order)를 가진다.

1.3 最惡의 境遇 比率

問題의 크기가 n=2m이고 작업처리순서에 따른 작업준비시간이 다음과 같은 例題를 통하여 새로운 휴리스틱 알고리즘의 最惡의 境遇 比率을 구한다.

$s(i, i+1) = 1$	$i \in N\text{-}\{n\}$
$s(m, m+1) = m,$	
$s(i, j) = 1$	$m+1 \le i \le n,\ 1 \le j \le m,\ i \ne j$
$s(i, j) = 1$	$1 \le i \le m,\ 1 \le j \le m,\ i \ne j$
$s(i, j) \ge L,$	$1 \le i \le m,\ m+1 \le j \le n,$ L은 충분히 큰 陽의 演算

위 例題의 最適 作業處理順序는 $T^*=(1, 2, 3, \cdots, n, 1)$이고 총작업완료시간은 $M(T^*)=n+L-1$이다.

새로운 휴리스틱 알고리즘으로 問題를 풀면 最善의 작업처리순서와 총작업완료시간은 다음과 같다.

最善의 작업처리순서 :
$T_m=(n, n-1, \cdots, m+1, 1, 2, \cdots, m, n)$
총작업완료시간 :
$M(T_m)=mL+m=n(L+1)/2$이다.

그러므로 最惡의 境遇 比率 $R_n \ge n(L+1)/(n+L-1)$이며 L이 n에 비해 상당히 크다면 R_n은 n/2로 近似되어 最短 作業準備時間法의 最惡의 境遇 比率과 같아짐을 알 수 있다.

4. 比較分析

1.1 實驗設計

最短 作業準備時間法과 새로 考案된 휴리스틱 알고리즘을 比較分析하기 위하여 使用한 절차는 다음과 같다.

1) 檢定問題의 生成

두 가지 휴리스틱 알고리즘을 遂行시키기 위한 檢定問題는 컴퓨터에 의한 시뮬레이션으로 生成되었으며 問題의 크기는 n=40이며 모두 20개의 問題를 發生시켰다. 各 問題의 작업준비시간은

$s(i,j) := [100*R(i,j)],\ i \ne j$
$:= 0 \qquad i=j$

R(i,j)는 구간이 (0,1)인 균등분포를 따르는 난수

2) 알고리즘의 遂行

發生된 20개의 檢證問題를 各各 最短 作業準備時間法과 새로운 휴리스틱 알고리즘으로 遂行시켜서 最善의 작업처리순서와 그때의 총작업완료시간을 出力시키어 演算結果를 確認하였다.

3) 下限의 設定

최적해의 下限으로 해밀턴經路問題의 緩和(relaxation)型인 割當問題의 最適解를 사용하였다. 割當問題의 최적해는 해밀턴經路問題의 現實可能解가 아닌 경우가 일반적이지만 사용이 간편하다. 휴리스틱 알고리즘의 수행결과에 대한 成果尺度(Performance measure)로 演算結果가 下限을 초과하는 비율을 사용하였다.

4) 遂行結果

이상의 절차를 거쳐 얻어진 實驗의 최종결과는 〈표 1〉과 같다.

〈표 1〉 휴리스틱 알고리즘의 결과

檢證問題 順序	下限	最短 準備時間法		새 알고리즘	
		길이	比率	길이	比率
1	217	300	1.382	258	1.189
2	275	320	1.164	303	1.102
3	167	220	1.317	249	1.491
4	284	320	1.127	318	1.120
5	178	210	1.180	217	1.219
6	165	250	1.515	221	1.339
7	227	269	1.185	284	1.251
8	213	255	1.197	243	1.141
9	184	254	1.380	239	1.299
10	234	266	1.137	262	1.120
11	268	296	1.104	294	1.097
12	243	332	1.366	271	1.115
13	197	266	1.350	228	1.157
14	214	256	1.196	293	1.369
15	203	265	1.305	248	1.222
16	281	341	1.214	362	1.288
17	175	254	1.451	224	1.280
18	238	259	1.088	254	1.076
19	235	298	1.288	280	1.191
20	228	309	1.355	271	1.189

1.2 統計處理

휴리스틱 알고리즘의 成果를 評價하기 위하여 사용되는 방법으로는 最惡의 境遇 分析法(worst case analysis), 確率的 分析法(probabilistic analysis), 實證分析法(empirical analysis)등이 있으며 實證分析法은 앞의 두 가지 방법에 비해 자칫 주관적이고 비과학적인 方法으로 실시되기 쉬운 위험이 있으나 실시가 容易하고 現實 問題에 대한 適用可能性이 크다는 長點을 지니고 있다.

本 研究에서는 휴리스틱 알고리즘의 比較分析을 위해 실증분석법에 속하는 統計分析法(Statistical analysis)을 사용한다.

1) 윌콕슨의 符號있는 順位檢定法

본 연구에서 휴리스틱 알고리즘의 成果尺度로 사용한 下限超過比率은 그 모집단이 特定分布에 따른다고 볼 수 없으므로 非母數檢定法(nonparametric test)의 하나인 윌콕슨의 符號있는 順位檢定法(Wilcoxon signed rank test)을 使用한다. 윌콕슨의 符號있는 順位檢定法은 자료에서 주어진 數値나 계산에서 도출된 數値를 직접 사용하지 않고, 그 數値가 전체에서 차지하는 순위와 數値差異에 의한 符號를 假說檢定의 근거로 삼는 방법이다.

n개의 휴리스틱 x와 휴리스틱 y를 檢定할 때 필요한 假定은 다음과 같다.

i) i번째 자료는 $(x_i,\ y_i)$의 順序雙으로 나타나며 그 차이는 $d_i=x_i-y_i$로 나타낸다.

ii) 모든 d_i는 連續確率變數이다.

iii) 각 d_i의 分布는 대칭적이다.

iv) 각 順序雙 $(x_i,\ y_i)$는 二變量 分布(bivariate distribution)에서 抽出된 確率分布이다.

歸無假說은 $E(x)=E(y)$이고 對立假說은 $E(x)>E(y)$ 또는 $E(x)<E(y)$중의 하나를 세울 수 있다.

檢定統計量 w를 구하는 절차는 먼저 모든 자료에서 $|d_i|$를 구하고 만일 $d_i=0$인 자료가 있으면 제외시키고 크기 n을 그만큼 감소시킨다. 또한 차이의 절대값이 동일할 경우 자료가 가지게 될 順位의 평균값을 부여한다. 두 번째로 각 자료의 順位에 x_i-y_i의 符號를 붙이고 이 符號있는 順位를 R_i로 表示한다. 마지막으로 符號있는 순위의 총합인 檢定統計量 w를 구한다.

棄却域은 유의수준 α에서

對立假說이 $E(x)=E(y)$인 경우는 $W>W(1-\alpha/2)$, 또는 $W>W(\alpha/2)$

對立假說이 $E(x)>E(y)$인 경우는 $W>W(1-\alpha)$

對立假說이 $E(x)<E(y)$인 경우는 $W<W(\alpha)$이다.

그러나 자료의 크기 n이 10이상인 경우, 臨界値 $w(\alpha)$는 다음과 같이 近似될 수 있다.

$$W(\alpha) = Z(\alpha)\sqrt{\frac{n(n+1)(2n+1)}{6}}$$

2) 檢定結果

〈표 1〉에 나타난 수행결과를 윌콕슨의 부호 있는 순위검정법에 의해 계산한 결과는 〈표 2〉에 나타나 있다.

본 연구에서는 最短 作業準備時間法에 대한 새로운 알고리즘의 우위성을 입증하려는데 목적이 있으므로 對立假說을 $E(x)>E(y)$로 세운다. 分析에서 時間複雜度函數와 最惡의 境遇 比率이 $0(n^4)$와 $R_n=n/2$로 同一한 水準임을 알 수 있었다.

유의수준 $\alpha=0.05$에서 임계치 $W(1-\alpha)$는

$W(1-\alpha)=(1.645)20*21*41/6=88.127$이므로

$W>W(1-\alpha)$가 된다.

그러므로 歸無假說 $E(x)=E(y)$는 유의수준

α=0.05에서 기각되고 對立假說 E(x)>E(y)이 採擇된다. 따라서 새로운 휴리스틱 알고리즘의 最適解 近似性이 最短 作業準備時間法보다 優越하다고 주장할 수 있다.

5. 結 論

작업준비시간을 고려한 總作業準備時間 최소화 문제는 組合最適化問題의 代表的인 문제로 처리해야 할 작업들의 작업처리순서를 결정하는 문제이다. 이 문제는 NP-complete 계열에 속하므로 최적해를 효과적으로 구할 수 있는 多項時間 알고리즘이 存在하지 않는다. 따라서 문제규모가 큰 경우 최적해를 찾는다는 것은 계산시간상 현재의 컴퓨터 계산능력으로는 거의 불가능하게 된다. 그러므로 시간과 비용의 補償效果에 의해서 휴리스틱 접근법을 사용함으로써 비록 최적해는 아니지만 만족할만한 解를 구해 사용하게 된다.

〈표 2〉 윌콕슨 검정결과

最短 準備時間法	새 알고리즘		$\lvert x_i - y_i \rvert$의	符號있는
x_i	y_i	$x_i - y_i$	順位	順位 R_i
1.382	1.189	.193	18.5	18.5
1.164	1.102	.062	7	7
1.317	1.491	-.174	16	-16
1.127	1.127	.007	1.5	1.5
1.180	1.219	-.039	5	-5
1.515	1.339	.176	17	17
1.185	1.251	-.066	8	-8
1.197	1.141	.056	6	-6
1.380	1.299	.081	11	11
1.137	1.120	.017	4	4
1.104	1.097	.007	1.5	1.5
1.366	1.115	.251	20	20
1.350	1.157	.193	18.5	18.5
1.196	1.369	-.173	15	-15
1.305	1.222	.083	12	12
1.214	1.288	-.074	9	-9
1.451	1.280	.171	14	14
1.088	1.076	.012	3	3
1.268	1.191	.077	10	10
1.355	1.189	.166	13	13
				W=104

본 연구에서 비교 분석한 最短 作業準備時間法과 새로 개발한 휴리스틱 알고리즘은 작업준비시간을 고려한 總作業準備時間 최소화 문제에서 문제를 구성하는 모든 작업들을 처리하는 작업순서를 순차적으로 완성해나가는 알고리즘이다. 양 알고리즘의 기본구조는 동일하지만 最短 作業準備時間法은 부분 작업처리순서를 확장시켜나가는 과정에서 새로이 부분 작업처리순서에 포함될 작업의 選擇基準으로 最短 작업준비시간을 사용하는데 반하여 새로 개발한 휴리스틱은 選擇基準으로 最大 작업준비시간의 機會費用을 사용한다.

最短 作業準備時間法과 새로 개발한 휴리스틱 알고리즘의 구조를 분석하여 時間複雜度函數와 最惡의 경우 비율이 $O(n^4)$와 $R_n = n/2$로 동일함을 알 수 있었다.

마지막으로 컴퓨터를 使用하여 시뮬레이션으로 生成시킨 20개의 檢證問題를 各各 두 알고리즘에 의해 수행시키고 그 결과를 下限超過比率이라는 成果尺度에 의해 비교한 후 윌콕슨의 符號있는 順位檢定法을 통하여 새로운 개발된 휴리스틱 알고리즘의 解가 最短 作業準備時間法의 解보다 개선되었음을 입증하였다.

參 考 文 獻

[1] Rosenkrantz, "An analysis of serveral Heuristics for the TSP", SIAM J. comp 6, pp 563-581

[2] W. R. Stewart, JR A computationally efficient heuristic for the traveling salesman problem. Proc 13th Annual Meeting of S. E. TIMS, pp 75-85, 1977

[3] Norback, R. F. Love Heuristic for the Hamiltonian path problem in Euclidian two space. J. Oper. Res. Soc. 30, pp 363-368 1979

[4] Lin. S, "Computer solutions of the TSP", BSTJ, No. 10, pp 2245-2246, Dec. 165

[5] Aki, The minimal directed spanning subgragh for cmbinatorial optimization, Austral. Comput. J., 1980, pp. 132-136.

[6] B. L. Golden, L. D. Bodin & W. Stewart, Approximate travelling salesman algorithm, Oper, Res., 1980, pp. 694-711.

[7] Golden, Bodin, "Approximate TS algorigthms", O. R. 28, pp 694-711, 1980

[8] M. H. J. Webb, Some methods of producing approximate solutions to travelling salesman problems with hundreds or thousands of cities. Oper. Res. Quart. 22, pp 49-66, 1971

[9] L. L. Karg & G. L. Thompson, A heuristic approach to traveling salesman, Man. Sci., 1964, pp. 225-248.

[10] P. Weiner & S. Savage, Neighborhood search algorithms for guaranteeing optimal traveling salesman tours must be inefficient, J. Comp. Sys. Sck., 1976, pp. 25-35.

[11] A. M. Frieze, g. Galbiati & F. Maffioli, On the worse case performance df some algorithms for the asymmetric traveling salesman problem, Networks, voi. 12, 1982, p. 27.