Journal of The Korea Society of Computer and Information
Vol. 19, No. 2, February 2014
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2014.19.2.143

트리에서의 배달문제에 대한 최적해 알고리즘

이 광 의*

Optimal Solution Algorithms for Delivery Problem on Trees

KwangEui Lee*

요 약

본 논문에서는 트리에서의 배달문제를 제안하고, 이를 해결하는 두 개의 알고리즘을 제안한다. 트리상에서의 배달문제는 n개의 서로 다른 이동속도를 갖는 로봇을 이용하여 배달물을 트리의 한 노드에서 다른 노드로 배달하는 시간을 최소화 하는 문제이다. 첫 번째 알고리즘은 핸드오버가 발생하는 장소에 대한 제한을 두어 트리에서의 배달문제를 위한 최적해를 생성한다. 이 알고리즘에서 핸드오버는 주어진 트리의 정점에서만 발생하는 것으로 가정하며, 시작점에서 도착점까지 핸드오버가 발생하는 위치와 핸드오버에 참여하는 로봇의 색인을 순서대로 찾는 방법으로 문제를 해결한다. 두 번째 알고리즘은 첫 번째 알고리즘을 확장하여 핸드오버 장소에 대한 제약을 제거하며, 여전히 최적해를 생성한다. n을 로봇의 수라 하고, m을 트리의 노드 수라 할 때 두 개의 알고리즘은 모두 $O((n+m)^2)$ 시간복잡도를 갖는다.

▸ Keywords : 배달문제, 경로 계획, 로봇에이전트, 최적해 알고리즘

Abstract

In this paper, we propose the delivery problem on trees and two algorithms for the problem. The delivery problem on trees is that of minimizing the object delivery time from one node to another node using n various speed robots. Our first algorithm generates an optimal solution with some restrictions in handover places. In this algorithm, we assume that the handover can be made at a vertex of given tree. We try to find the handover places and the robots participate in handover from the start node to the destination node. The second algorithm extends the first one to remove the restriction about the handover places. The second algorithm still generates an optimal solution. The time complexities of both algorithms are $O((n+m)^2)$ where n is the number of robots and m is the number of nodes.

•제1저자 : 이광의 •교신저자 : 이광의
•투고일 : 2014. 1. 28, 심사일 : 2014. 2. 4, 게재확정일 : 2014. 2. 12.
* 동의대학교 멀티미디어공학과(Dept. of Multimedia Engineering, Dongeui University)
※ 이 논문은 2012학년도 동의대학교 교내연구비에 의해 연구되었음(과제번호: 2012AA202)

▸ Keywords : Delivery Problem, Path Planning, Robot Agent, Optimal Solution algorithm

I. 서 론

배달문제는 m차원 유클리드 공간에서 임의의 시작점으로부터 도착점까지 배달물을 배달하는 시간을 최소화 하는 문제이다[1]. 이 문제를 해결하기 위하여 서로 다른 이동속도와 시작위치를 갖는 n개의 로봇에이전트가 활용된다. 초기에 배달물은 시작점에 놓여있으며, 로봇에이전트들은 서로 협동하여 배달물을 최단시간 안에 도착점까지 배달하여야 한다. 배달 시간을 최소화하기 위하여 배달과정에서는 많은 수의 핸드오버가 발생할 수 있으며, 핸드오버 과정은 시간이 소요되지 않는 것으로 한다.

주어진 문제의 어려움을 설명하기 위하여 아주 간단한 형태의 문제를 고려하면 다음과 같다. 먼저 두 개의 로봇에이전트만 존재하는 것으로 한다. 이때, 각각의 에이전트 이름을 $r[0]$와 $r[1]$으로 하고, $r[1]$이 $r[0]$보다 b배 빠르게 이동한다고 가정한다. 또한 초기에 $r[0]$는 시작점(배달물이 있는 위치)에 있다고 하자. 만약 도착점이 시작점으로부터 충분이 떨어져 있는 경우라면, 배달시간을 최소화하기 위하여 $r[0]$는 배달물을 $r[1]$에게 넘겨주어야 하며, 이때 넘겨주는 위치는 그림 1에서와 같은 $r[0]$와 $r[1]$, 그리고 속도의 비 b에 의하여 생성되는 아폴로니안 원상의 한 점에서 발생함을 알 수 있다[2].

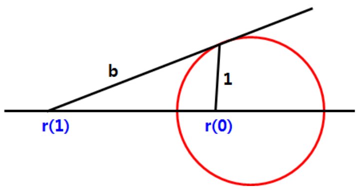

그림 1. 두 점과 비율 b로 정의되는 아폴로니안 원
Fig. 1. Apollonian circle defined by two points and ratio b

$r[i].s$를 $r[i]$의 이동속도라 하고, $r[i].p$를 $r[i]$의 초기위치, sp를 시작점, dp를 도착점이라 할 때 핸드오버 지점은 $ED(r[1].p,x)+ED(x,dp)$를 최소화 하는, $r[1].s$와 $r[0].s$ 그리고 두 로봇의 속도의 비인 $r[1].s/r[0].s$의 비로 구성되는 아폴로니안 원상의 점이 된다. 여기서, $ED(x,y)$는 점 x와 점 y사이의 유클리드 거리(Euclidean distance)이다. 하지만, 이러한 단순한 경우에도 정확한 위치를 찾는 것은 매우 어려운 문제이다[1].

배달문제는 경로계획 문제의 한 유형으로 볼 수 있다[3]. 기존의 연구를 보면, 하나의 로봇에 대한 경로계획 문제가 주로 연구되었으며[4][5], 이후 다중로봇에 대한 결과도 활발하게 연구되고 있다[6][7][8][9]. 하지만, 지금까지의 대부분의 연구는 로봇 자원과 평균대기시간의 최적화에 초점이 맞추어져 있었다. 본 논문에서 다루는 배달문제와 관련된 연구로는 2009년 문제정의와 함께 m차원 유클리드 공간에서의 배달문제를 해결하기 위한 유전자 알고리즘이 제시되었고[1], 2011년 핸드오버가 가능한 장소를 격자점으로 제한한 경우에 대한 최적해 알고리즘이 제시되었다[10]. 하지만, 실제 도로상황을 좀 더 잘 반영한다고 할 수 있는 그래프에서의 배달문제에 대한 결과는 아직 없다.

본 논문에서는 그래프에서의 배달문제에 대한 기초연구로 트리에서의 배달문제를 제안하고 이에 대한 두 개의 최적해 알고리즘을 제안한다. 첫 번째 알고리즘은 핸드오버가 트리의 노드에서만 발생함을 가정하며, 두 번째 알고리즘은 이를 확장하여, 트리의 간선을 실제 노드상의 거리로 생각하고 간선상의 임의의 위치에서 핸드오버가 발생할 수 있는 경우를 고려한 알고리즘이다. 논문의 구성은 2장에서 배달문제를 형식화하며, 3장에서 두 개의 알고리즘을 제시하고, 바르게 동작함을 증명한다. 그리고 마지막으로 4장에서 결론 및 향후 연구방향에 대하여 기술한다.

II. 배달문제

m차원 배달문제는 시작점, 도착점, 그리고 n개의 로봇에 대한 이동속도와 시작위치로 구성된다. 배달시간을 최적화하기 위하여 배달 중간에 여러 번의 핸드오버가 발생하게 된다. 배달문제는 배달에 참여하는 로봇의 순서와 로봇간의 핸드오버가 발생하는 정확한 위치를 찾는 것이다.

문제를 단순하게하기 위하여 로봇이 배달물을 집고, 전달(핸드오버)하고, 놓기 위한 시간은 걸리지 않는다고 가정한다. 배달문제는 다음과 같이 형식화 될 수 있다. 기본적으로 이러한 형식화는 논문[1]에서와 동일하나 트리에서의 문제로 변형하기 위하여 약간의 수정을 가하였다. 각각의 트리 정점과 로봇은 유일한 번호를 식별자로 갖는 것을 가정한다.

$v[x], 0 \le x < m$: 대상 그래프의 정점의 집합,

$d(v[x], v[y]), 0 \le x, y < m$: 두 개의 정점 $v[x]$와 $v[y]$를 연결하는 유일한 경로의 길이,

$r[i], 0 \le i < n$: i번째 로봇 에이전트,

$r[i].p, 0 \le i < n$: $r[i]$의 초기위치 (트리상의 정점번호),

$r[i].s, 0 \le i < n$: $r[i]$의 이동속도. $r[i]$는 $r[i].s$의 속도로 이동한다. 이때, 모든 로봇에이전트의 이동속도는 서로 다르다고 가정한다.

v_s : 시작점. 배달물의 초기위치,

v_d : 도착점. 배달물의 최종위치,

$t_i, 0 \le i \le k$: 배달물이 배달되는 경로상의 정점 번호 순서열로 $v[t_0] = v_s$, $v[t_k] = v_d$가 된다,

$s_i, 0 \le i < k, k \le n$: 배달에 참여하는 로봇 번호 순서열, 로봇 $r[s_i]$는 배달물을 $v[t_i]$에서 $v[t_{i+1}]$로 가져가 $r[s_{i+1}]$에게 전달한다. 기술과정에서 $s_i = s_{i+1}$인 경우 가 발생할 수 있으나, 이때 실제 핸드오버는 발생하지 않는다.

정의된 표기법을 활용하여 배달순서는 다음과 같이 기술할 수 있다.

$$t_0, s_0, t_1, s_1, t_2, \ldots, t_{k-1}, s_{k-1}, t_k$$

이때, v_s(시작점)과 v_d(도착점)도 역시 하나의 정점이 되며, 트리의 임의의 두 개의 정점을 연결하는 경로는 유일하므로 v_s와 v_d를 연결하는 경로도 유일하게 되는데 이 경로를 sd-path라 하자.

다음절에서는 배달순서열의 배달시간을 최소화하기 위하여 배달순서의 다양한 성질을 확인하고, 이에 따른 알고리즘을 제안한다.

III. 트리에서의 최적해 알고리즘

이 장에서는 트리에서 동작하는 최적해 알고리즘을 제안한다. 트리에서 동작한다 함은 모든 로봇이 트리의 간선을 따라 이동함을 의미한다. 일반성을 잃지 않고 각각의 로봇은 서로 다른 정점에 위치함을 가정할 수 있다. 또한 논의의 전개를 단순화하기 위하여 v_s에 로봇이 존재함을 가정한다. 정점 v_s에 로봇이 존재하지 않는 경우 이동속도 0인 로봇이 존재함을 가정하면 전체 문제에 대한 해가 변경되지 않으므로 이러한 가정은 언제든지 가능하다.

1. 트리에서의 배달문제 특성

트리에서의 배달문제를 다루기 위하여 트리와 배달문제의 특성에 대한 고찰이 필요하다. 좀 더 깊이 있게 다루기 전에 다음의 표기법들을 정의한다.

$v[x].n, 0 \le x < m$: $v[x]$로부터 가장 가까운 sd-path상의 정점.

트리에서의 배달문제를 고려할 때 다음이 성립한다.

보조정리 1. 정점에서만 핸드오버가 발생하는 배달문제에 대한 임의의 배달순서는 배달시간의 증가 없이 배달물이 sd-path에서 v_d방향으로만 이동되는 배달순서로 변경가능하다.

증명: 먼저 배달물이 $v[t_p]$에서 sd-path를 떠나고, $v[t_q]$에서 sd-path로 돌아오는 어떤 배달순서가 존재한다하자. 이때, $t_p = t_q$가 된다. 이 경우 배달순서는 다음과 같이 표기할 수 있다.

$$t_0, s_0, t_1, \ldots, s_{p-1}, t_p, \ldots, s_{q-1}, t_q, s_q, t_{q+1}, \ldots, s_{k-1}, t_k$$
$$where\, t_p = t_q.$$

따라서 이 배달순서는 배달시간의 증가 없이 다음과 같은 sd-path를 떠나지 않는 배달순서로 변경가능하다.

$$t_0, s_0, t_1, \ldots, s_{p-1}, t_p s_q, t_{q+1}, \ldots, t_{k-1}, s_{k-1}, t_k$$

또한, 배달물이 sd-path에서 v_s방향으로 이동해가는 경

우도 비슷하게 증명가능하다. 따라서 보조정리가 성립한다 □

따름정리 1. 정점에서만 핸드오버가 발생하는 배달문제에 대하여 sd-path를 따라 배달물을 v_s에서 v_d로만 이동하는 최소시간 배달순서가 존재한다 □

이러한 정리들로부터 정점에서만 핸드오버가 발생하는 배달문제에 대한 배달순서가 알려진 경우 배달에 참여하는 모든 로봇 $r[s_p]$는 $r[s_p].p$에서 $r[s_p].n$으로, 다시 $v[t_p]$로 이동한 후 대기상태에 들어가고, $v[t_p]$에 배달물이 도착하면, $v[t_{p+1}]$로 이동하는 방법으로 최소시간에 배달물을 배달할 수 있음을 알 수 있다. 이때, 만약 $v[t_p]$가 sd-path상에서 $r[s_p].n$보다 v_s에 가까운 쪽에 있다면, 우회(detour)가 발생할 수 있다. 이러한 관찰은 간선에서 핸드오버를 허용하는 알고리즘의 구현에 중요한 역할을 한다.

보조정리 2. 정점에서만 핸드오버가 발생하는 배달문제에 대한 모든 배달순서는, 배달시간의 증가 없이 모든 p에 대하여 $r[s_p].s \le r[s_{p+1}].s$가 성립하는 배달순서로 변경가능하다.

증명: 임의의 배달순서에서 $r[s_p].s \le r[s_{p+1}].s$를 만족하지 않는 최초의 p가 존재한다고 하자. 즉, $r[s_p].s > r[s_{p+1}].s$라 하자. 이 경우, $r[s_p]$가 $v[t_{p+1}]$에서 배달물을 $r[s_{p+1}]$에게 넘기지 않고 $v[t_{p+2}]$까지 직접 배달함으로써 배달시간을 늘리지 않고 배달할 수 있다. 이 경우 $r[s_{p+1}]$는 배달에 참여하지 않게 되며, 새로운 배달순서는 다음과 같게 된다.

$$t_0, s_0, t_1, \ldots, t_p, s_p, t_{p+1}, s_p, t_{p+2}, \ldots, s_{k-1}, t_k$$

새로운 배달순서는 기존의 배달순서보다 작거나 같은 시간에 배달을 완료하므로 보조정리가 성립한다 □

따름정리 2. 정점에서만 핸드오버가 발생하는 배달문제에서 모든 p에 대하여, $r[s_p].s \le r[s_{p+1}].s$가 성립하는 최소시간 배달순서가 존재한다 □

보조정리 3. 배달물을 v_s에서 시작하여 sd-path상의 노드 $v[t_q]$까지 배달하는데 필요한 최소 시간을 c_q라 하자. ($v[t_q]$는 sd-path에서 v_s로부터 v_d방향으로 q번째 정점이다.) 이러한 가정 하에 sd-path상의 노드 $v[t_{q+1}]$까지 배달하는데 필요한 최소 시간 c_{q+1}는 다음과 같이 구할 수 있다.

$$c_0 = 0 \qquad \text{(식1)}$$
$$c_{q+1} = \min\left\{ \begin{array}{l} \max\{c_q, d(v[r[i].p], v[t_q])/r[i].s\} \\ + d(v[t_q], v[t_{q+1}])/r[i].s \mid 0 \le i < n \end{array} \right\}$$

상기의 (식1)을 만족하는 i를 y라 하면, 개념적으로 $r[y]$의 의미는 $v[t_q]$를 거쳐 $v[t_{q+1}]$에 가장 먼저 도착할 수 있는 로봇이 된다. 단, 로봇은 배달물이 아직 도착하지 않은 경우 $v[t_q]$에서 배달물이 도착하기를 기다려야 한다.

증명: 배달물은 어떤 로봇에 의하여 $v[t_q]$에서 $v[t_{q+1}]$로 배달되어야 한다. 배달순서 $t_0, s_0, t_1, \ldots, s_{q-1}, t_q$는 보조정리 1과 보조정리 2의 특징을 만족하는 순서라 하자. 모든 로봇이 배달에 참여할 수 있으며, 임의의 로봇 $r[i]$가 배달물을 $v[t_q]$에서 $v[t_{q+1}]$로 배달하는데 소요되는 시간은 다음의 두 가지 경우로 나누어 생각할 수 있다.

(i) 임의의 로봇 $r[i]$가 $v[t_0]$에서 $v[t_q]$까지의 배달에 참여하지 않은 경우: 이 경우 배달에 필요한 시간은

$$\begin{array}{l} \max\{c_q, d(v[r[i].p], v[t_q])/r[i].s\} \\ + d(v[t_q], v[t_{q+1}])/r[i].s \mid 0 \le i < n \end{array} \qquad \text{(식2)}$$

가 된다. 따라서 이중 가장 빨리 배달을 완수할 수 있는 로봇이 보조정리 3에서 선택된 $r[y]$이므로 정리는 성립한다.

(ii) 임의의 로봇 $r[i]$가 $v[t_0]$에서 $v[t_q]$까지의 배달에 참여한 경우: 이 경우는 다시 $r[s_{q-1}] = r[y]$인 경우와 $r[s_{q-1}] \ne r[y]$인 2개의 부분경우로 구분가능하다.

(ii-1) $r[s_{q-1}] = r[y]$인 경우: $r[y]$에 의하여 배달물이 $v[t_q]$에 배달되었으므로 시간 c_q에 $r[y]$는 v_q에 있게 된다. 모든 로봇이 c_q이전에는 $v[t_q]$를 떠날 수 없으므로 여전히 $r[y]$가 배달물을 $v[t_{q+1}]$에 가장 빨리 배달할 수 있는 로봇이 되며 정리는 성립한다.

(ii-2) $r[s_{q-1}] \ne r[y]$인 경우: 일반성을 잃지 않고, $p < q$에 대하여 $r[y]$가 $r[y].s < r[s_p].s$인 $v[t_p]$까지 배달하였다고 하자. 그리고 다른 로봇 $r[s_p]$에게 전달하였다고 하자. 이때, $r[y]$와 $r[s_p]$를 고려하면, 두 로봇은 동시에 $v[t_p]$

를 거쳐서 이동하며, $v[t_{p+1}]$에 $r[s_p]$가 먼저 도착하게 되고 $v[t_q]$에도 $r[s_p]$가 먼저 도착하게 된다. 또한 $r[y].s < r[s_p].s$이므로 (ii-2)의 경우는 발생할 수 없다.

따라서 모든 경우에 배달물을 $v[t_{q+1}]$까지 배달하는 가장 빠른 방법은 $r[y]$에 의하여 배달되는 것이므로 보조정리는 성립한다 □

2. 정점에서만 핸드오버를 허용하는 알고리즘

이 절에서는 주어진 트리의 정점에서만 핸드오버가 가능함을 가정한다. 보조정리 1에 따라 배달물은 sd-path를 따라 이동하게 된다. 여기서 sd-path의 경로길이를 k라 하고, sd-path상의 정점 이름을 순서대로 대로 $t[0] \sim t[k]$라 하자. 알고리즘은 다음과 같다.

```
algorithm HandoverOnVertex {
  compute d(v[i],v[j]) for all i, j;
  r = {r[i] | 0<=i<n};
  k = length of the sd-path;
  for (j=0; j<k; j++) {
    for all r[i] in r {
      r[i].time = /* (식2)에 따라 계산된 값 */
    }
    s[j] = i such that r[i].time is the minimum;
    for all r[i] in r {
      if (r[i].s<r[s[j]].s) r=r-{r[i]};
    }
  }
  output the delivery path;
}
```

그림 2. 정점에서 핸드오버를 허용하는 알고리즘
Fig. 2. Algorithm that allows handover on vertices

정리 1. 알고리즘 HandoverOnVertex는 정점에서만 핸드오버가 발생하는 배달문제를 위한 최적의 배달순서를 출력한다.

증명. 위의 알고리즘에 의하여 계산된 배달순서는 보조정리 3에서 계산된 최소시간에 배달을 수행하므로 최소시간 배달순서가 된다. □

알고리즘의 시간복잡도를 계산하기 위하여 로봇의 수를 n이라 하고, 정점의 수를 m이라 하자. 또한, 하나의 정점에 여러 개의 로봇이 있는 경우 그중 가장 빠른 로봇만 배달에 참여하는 것이 의미가 있으므로, 일반성을 잃지 않고, 하나의 정점에 최대 1개의 로봇만 있음을 가정할 수 있다.

먼저 주어진 트리에 대하여 모든 정점의 쌍에 대한 최단경로길이를 구하는 것은 n번의 깊이우선탐색(depth first search)을 호출함으로써 $O(m^2)$의 시간에 해결가능하다 [11]. 그 이외의 문장은 모두 $O(n^2)$에 계산가능함을 쉽게 확인할 수 있다. 따라서 $n \le m$이므로 전체 시간복잡도는 $O(m^2)$이 된다.

3. 간선상에서 핸드오버를 허용하는 알고리즘

핸드오버가 트리의 정점이 아닌 간선의 임의의 위치에서 발생할 수 있는 경우도 그 결과는 다음과 같이 정점에서만 핸드오버를 허용하는 경우와 동일한 형태의 배달순서로 기술할 수 있다.

$$t_0, s_0, t_1, s_1, t_2, \ldots, t_{k-1}, s_{k-1}, t_k$$

여기서 핸드오버가 발생하는 위치에 대한 정보를 표현하는 t_i는 정점이 아니라 sd-path를 기준으로 v_s로부터의 거리로 생각할 수 있다. 지금부터는 t_i를 제외한 나머지 용어들의 정의를 유지하고 간선에서 핸드오버를 허용하는 배달문제에 대한 알고리즘을 구성하기 위해 앞 절에서 주어진 보조정리들을 변형하여 기술한다.

보조정리 4. 간선에서의 핸드오버를 허용하는 배달문제에서 모모든 최소시간 배달순서는 배달물이 sd-path에서 v_d방향으로만 이동되는 최소시간 배달순서로 변경가능하다.

증명: 배달물은 배달순서의 한 부분에서 sd-path를 떠나고, 다시 그 뒤의 어떤 부분에서 sd-path로 돌아온다. 이때, 보조정리 1의 증명에서와 마찬가지로 배달순서상에서 이 두 점을 표시하는 위치를 잘라서 가운데 부분을 생략하고 이어붙임으로써 시간의 증가 없이 배달물이 sd-path에서 v_d방향으로만 이동되는 배달순서로 변경가능하다. 또한, 배달물이 sd-path에서 v_s방향으로 이동해가는 경우도 비슷하게 증명가능하다 □

상기 증명과정에서 명확한 증명을 위하여 표기법을 확장할 필요가 있으나, 그 내용이 어렵지 않으므로 여기서는 생략한다.

따름정리 3. 간선에서의 핸드오버를 허용하는 배달문제에 대하여 sd-path를 따라 배달물을 v_s에서 v_d로만 이동하는 최소시간 배달순서가 존재한다 □

보조정리 5. 간선에서의 핸드오버를 허용하는 배달문제에 대한 모든 배달순서는, 배달시간의 증가 없이 모든 p에 대하여 $r[s_p].s \le r[s_{p+1}].s$가 성립하는 배달순서로 변경가능하다.

증명: 정점에서만 핸드오버가 발생하는 배달문제에서의 보조정리 2와 완전하게 동일한 방법으로 증명가능하다 □

따름정리 4. 간선에서의 핸드오버를 허용하는 배달문제에서 모든 p에 대하여, $r[s_p].s \le r[s_{p+1}].s$가 성립하는 최소시간 배달순서가 존재한다 □

위의 보조정리와 따름정리를 활용하여 간선에서의 핸드오버를 허용하는 경우의 알고리즘을 제안하고 이 알고리즘이 바르게 동작함을 증명한다. 이를 위하여 먼저 각각의 로봇들의 이동규칙(MRDP)을 제안한다.

MRDP1. 배달물을 소유하지 않은 모든 로봇은 현재의 배달물의 위치를 향한 유일한 경로를 통하여 배달물을 향해 이동한다. MRDP2. 배달물을 소유한 로봇이 자신보다 이동속도가 빠른 로봇을 만나면 배달물을 넘겨준다. MRDP3. 배달물을 소유한 로봇은 도착점 v_d를 향한 유일한 경로를 통하여 도착점을 향해 이동한다.

그림 3. 배달문제를 위한 로봇의 이동규칙
Fig. 3. Robot moving rules for the delivery problem

그림 3의 이동규칙에 따라 결정되는 핸드오버 지점 t_i와 인접한 두개의 핸드오버 지점 사이에 배달을 시행하는 로봇 $r[s_i]$에 대하여, t_i와 s_i로 구성된 배달순서가 최소시간 배달순서임을 보이고, 마지막으로 상기의 이동규칙에 따라 도출되는 t_i와 s_i를 계산하는 알고리즘을 제안한다.

보조정리 6. MRDP에 따라 도출되는 t_i와 s_i로 구성된 배달순서는 최소시간 배달순서이다.

증명. 여기서는 도출된 배달순서에 의하여 각각의 t_i까지 배달 소요시간이 최소임을 보임으로써 증명한다. 먼저, t_0까지의 배달시간은 항상 0이므로 최소시간 배달순서임을 알 수 있다. 다음이 v_s에서 t_p까지 최소시간 배달순서임을 가정하자.

$$t_0, s_0, t_1, \ldots, t_{p-1}, s_{p-1}, t_p$$

여기서 t_p가 결정되는 순간은 $t_p = v_d$이거나 $r[s_{p-1}]$이 자신보다 더 빨리 이동하는 $r[s_p]$를 만나는 경우이다. 다음에서 이동 규칙에 따라 $r[s_p]$가 $r[s_{p+1}]$을 만나는 지점인 t_{p+1}에 대하여, 이동규칙에 따라 만들어진 s_p와 t_{p+1}을 기존의 배달순서에 추가한 것이 t_{p+1}까지 최소시간 배달순서임을 보인다.

t_{p-1}에서 $r[s_{p-1}]$이 배달물을 $r[s_p]$에게 넘기는 상황을 고려하면, $r[s_p]$보다 더 빠른 로봇들은 아직 t_{p-1}에 도착하지 못한 상황이므로, 배달규칙에 따라 t_{p-1}에서 배달물을 소유할 수 있는 로봇은 $r[s_{p-1}]$과 $r[s_p]$뿐이다. 또한 $r[s_{p-1}].s < r[s_p].s$이므로 $r[s_{p-1}]$가 배달물을 $r[s_p]$에게 넘김으로써 최소시간 배달순서를 구성할 수 있게 된다. 이때, $r[s_p]$보다 더 빠른 로봇들은 최소시간 배달순서를 구성하기 위하여 먼저 $r[s_p]$를 만나야 하므로 MRDP1의 규칙에 따라 이동하게 된다. 같은 논의로 $r[s_p]$는 t_{p+1}에서 배달물을 $r[s_{p+1}]$에게 전달하게 된다. 이때, $r[s_p]$가 t_{p+1}까지 이동하는 동안 $r[s_p]$보다 빠른 로봇을 만나지 않으므로 다음의 배달순서

$$t_0, s_0, t_1, \ldots, t_{p-1}, s_{p-1}, t_p, s_p, t_{p+1}$$

는 t_{p+1}까지의 최적의 배달순서가 된다. 따라서 보조정리는 성립한다 □

상기의 최소시간 배달순서에서 c_q를 $r[s_q]$가 t_q에 있을 때의 시간이라 하자. 이때, $r[s_q]$보다 느리게 이동하는 로봇은 보조정리 5에 의하여 고려할 필요가 없으므로 고려되는 모든 $r[i]$는 $r[s_q]$보다 빠르게 이동하고 있다. $r[s_q]$와 $r[i]$의 관계를 계산하기 위하여 $r[i]$가 $r[i].n$에 도착하는 시간을 보면 $d(v[r[i].p], v[r[i].n])/r[i].s$가 된다. 표기의 편의를 위하여 이를 $r[i].nt$라 하자.

이러한 관찰에 따라 c_q시간에 $r[s_q]$와 $r[i]$의 관계는 다음의 다섯 가지 경우로 나눌 수 있다.

(i) $r[i].n$이 t_i보다 v_s에 가깝고, $r[i]$가 아직 $r[i].n$에 도착하지 않은 경우.

(ii) $r[i].n$이 t_i보다 v_s에 가깝고, $r[i]$가 $r[i].n$을 지나친 경우.

(iii) $r[i].n$이 t_i보다 v_d에 가깝고, $r[i]$가 $r[i].n$을 지나

친 경우.

(iv) $r[i].n$이 t_i보다 v_d에 가깝고, $r[i]$가 아직 $r[i].n$에 도착하지 않았으며, $r[i]$가 $r[s_q]$보다 $r[i].n$에 먼저 도착하는 경우.

(v) $r[i].n$이 t_i보다 v_d에 가깝고, $r[i]$가 아직 $r[i].n$에 도착하지 않았으며, $r[s_q]$가 $r[i]$보다 $r[i].n$에 먼저 도착하는 경우.

알고리즘 작성을 위하여 위의 다섯 가지 경우를 조건문으로 표기하고, 해당하는 조건에 대하여 $r[s_q]$와 $r[i]$가 만나게 되는 sd-path상의 위치(이를 $r[i].pt$라 하자)를 식으로 표현하면 다음과 같다. 이때, 조건식과 계산식 짧게 하기 위하여 먼저 다음을 정의한다.

$$A = (d(v_s, r[i].n) < t_i)$$
$$B = (r[i].nt > c_q)$$
$$C = (r[i].nt \le c_q + (d(v_s, r[i].n) - t_i)/r[s_q].s))$$
$$D = (r[i].s - r[s_q].s)$$
$$E = (r[i].s + r[s_q].s)$$
$$F = r[s_q].s$$
$$F = (c_q - r[i].nt)$$
$$G = d(v_s, r[i].n)$$
$$H = d(r[i], r[i].n)$$

이러한 정의에 따라 앞에서 주어진 다섯 가지 경우의 조건문과 계산식을 표현하면 다음과 같다.

(i) if$(A \wedge B)$

$$t_q + [(t_q - G) + (H - r[i].s * c_q)]/D * r[s_q].s \quad (식3)$$

(ii) if $(A \wedge \sim B)$

$$t_q + [(t_q - G) - (F * r[i].s)]/D * r[s_q].s \quad (식4)$$

(iii) if $(\sim A \wedge B)$

$$t_q + [(G - t_q) - (F * r[i].s)]/E * r[s_q].s \quad (식5)$$

(iv) if $(\sim A \wedge \sim B \wedge C)$

$$t_q + [(G - t_q) + (H - r[i].s * c_q)]/E * r[s_q].s \quad (식6)$$

(v) if $(\sim A \wedge \sim B \wedge \sim C)$

$$G + [(t_q + (-F) * r[s_q].s) - G]/D * r[s_q].s \quad (식7)$$

주어진 식의 구조가 복잡하나 그 개념은 간단하므로 위의 수식은 증명 없이 받아들이는 것으로 한다. 이렇게 계산된 $r[i].pt$ 중 최소의 값이 t_{q+1}이 되며, 이때의 i가 s_{q+1}이 된다. 이에 따라 간선에서의 핸드오버를 허용하는 배달문제 대한 알고리즘을 구성할 수 있다.

```
algorithm HandoverOnEdge {
  compute d(v(i),v(j)) for all i, j;
  compute r(i).nt for all i;
  r = {r(i) | 0<=i<n};
  j = 0;
  while(r is not empty) {
    for all r(i) in r {
      r(i).pt = /* (식3)~(식7)에 따라 계산된 값 */
    }
    s(j) = i such that r(i).pt is the minimum;
    t(j) = r(s(j)).pt;
    for all r(i) in r {
      if (r(i).s<r(s(j)).s) r=r-{r(i)};
    }
  }
  output the delivery path;
}
```

그림 4. 간선에서 핸드오버를 허용하는 알고리즘
Fig. 4. Algorithm that allows handover on edges

정리 2. 알고리즘 HandoverOnEdge는 간선에서의 핸드오버를 허용하는 배달문제를 위한 최적의 배달순서를 출력한다.

증명. 위의 알고리즘에 의하여 계산된 배달순서는 보조정리 6에서 계산되는 최소시간에 배달할 수 있는 배달순서를 출력하므로 최소시간 배달순서가 된다. □

알고리즘의 시간복잡도를 계산하면, 먼저 주어진 트리에 대하여 모든 정점의 쌍에 대한 최단경로길이를 구하는 것은 n번의 깊이우선탐색을 호출함으로써 $O(m^2)$의 시간에 해결가능하다. 그 이외의 문장은 모두 $O(n^2)$에 계산가능함을 쉽게 확인할 수 있다. 역시 $n \le m$이므로 HandoverOnVertex와 마찬가지로 전체 시간복잡도는 $O(m^2)$이 된다.

IV. 결 론

본 논문에서는 그래프상에서의 배달문제에 대한 선행연구로 트리상에서의 배달문제를 제안하고 제안된 문제에 대한 최적해 알고리즘을 제시하였다. 먼저 정점에서만 핸드오버를 허용하는 경우 $O(m^2)$ 시간복잡도의 최적해 알고리즘을 제안하였으며, 이를 확장하여 간선상의 임의의 위치에서 핸드오버를 허용하는 경우에도 여전히 $O(m^2)$ 시간복잡도로 최적해를 계산하는 알고리즘을 제안하였다.

앞으로의 연구에서는 본 연구의 확장된 형태로 임의의 그래프에서의 배달문제와 제한이 없는 임의의 m차원 유클리드 공간에서의 배달문제에 대한 연구가 진행되어야 할 것이다.

참고문헌

[1] KwangEui Lee and JiHong Kim, "Genetic Algorithm for Delivery Problem," IJCSNS, V9, N2, pp. 248-251, February 2009.
[2] Apollonian Circle, *http://en.wikipedia.org/wiki/* Circles_of_*Apollonius*.
[3] Motion planning, *http://en.wikipedia.org/wiki/Motion_planning*.
[4] S. M. Lavalle, "*Planning Algorithms*," Cambridge University Press, 2006.
[5] Hyungil Kim, "Path Planning for Cleaning Robots Using Virtual Map," Journal of The Korea Society of Computer and Information, V14, N11, pp. 31-40, November 2009.
[6] D.K. Liu, D. Wang, G. Dissanayake, "A force field method based multi-robot collaboration," Proc. IEEE Int. Conf. on Robotics, Automation and Mechatronics, Bangkok, Thailand, pp. 662-667, June 2006.
[7] Y. Guo, L.E. Parker, "A distributed and optimal motion planning approach for multiple mobile robots," IEEE Int. Conf. on Robotics Automation, pp. 2612-2619, May 2002.
[8] K. Azarm and G. Schmidt, "Conflict-Free Motion of Multiple Mobile Robots Based on Decentralized Motion Planning and Negotiation," IEEE Int. Conf. on Robotics and Automation, pp 3526-3533, vol. 4, April 1997.
[9] Kyeonah Yu and Su-Jin Cho, "Path-Planning for Group Movement in Dynamic Environments," Journal of The Korea Society of Computer and Information, V18, N2, pp. 117-126, February 2013.
[10] KwangEui Lee, "Near oiptimal Algorithm for Delivery Problem," IJCSNS, V11, N12, pp. 25-28, December 2011.
[11] R. Neapolitan and K. Naimipour, "*Foundations of Algorithms Using C++ Pseudocode, 3rd Ed.*," Addison Wesley, 2003.

저 자 소 개

이 광 의
1990: 서강대학교
전자계산학과 공학사.
1992: 서강대학교
자계산학과 공학석사.
1997: 서강대학교
자계산학과 공학박사
1997~2001: 한국전자통신연구원
선임연구원
2001~현 재: 동의대학교
멀티미디어공학과 교수
관심분야: 계산이론, 상황인지
Email : kelee@deu.ac.kr