대한산업공학회/한국경영과학회 2000 춘계공동학술대회 논문집. 2000년4월21-22일. 경남대학교
Session OA3.1

선후행 관계제약을 갖는 TSP 문제의 유전알고리즘 해법

Traveling Salesman Problem with Precedence Relations based on Genetic Algorithm

문치웅, 김규웅, 김종수, 허선
한양대학교 산업공학과

Abstract

The traveling salesman problem with precedence relations (TSPPR) is harder than general traveling salesman problem. In this paper we propose an efficient genetic algorithm (GA) to solve the TSPPR. The key concept of the proposed genetic algorithm is a topological sort (TS).

The results of numerical experiments show that the proposed GA approach produces an optimal solution for the TSPPR.

1. 서론

외판원 문제(traveling salesman problem : TSP)는 상당한 복잡성을 갖는 전형적인 조합 최적화 문제의 하나로 주어진 그래프 내의 각 정점(vertex)을 한번씩만 방문하는 가장 짧은 경로(Route)를 구하는 문제이다. 이러한 문제는 공정 순서결정(process sequencing), 운송 장비의 경로선정문제(vehicle routing problem) 등 많은 공학 분야에 응용되고 있다. 또한 TSP는 NP-hard 문제에 속하므로 크기가 큰 문제에 대해서 적당한 시간 안에 주어진 TSP에 대한 최적해를 구하기란 쉽지 않은 일이다.

이러한 TSP를 응용하는 산업현장의 여러 문제들 중 스케줄링(scheduling), 경로결정문제(routing) 혹은 공정계획(process Planning) 등의 문제들은 공정 또는 작업간의 선후행 관계가 있으므로 선후행 제약이 있는 TSP(TSP with precedence relations:TSPPR)로 정식화 되어야 한다. 그러나 이러한 문제는 모델 구성의 복잡성과 문제를 풀기 위한 해법 적용의 어려움으로 일반적인 TSP를 다루는 것 보다 훨씬 까다롭다. 최근 들어 TSP의 해결을 위한 효율적 방법을 찾기 위한 연구 분야로 유전 알고리듬을 이용한 해법[2] 많이 개발되고 있다. Potvin[3]은 유전 알고리듬을 이용한 일반적인 TSP 해결에 관한 기존의 연구들을 소개하였다. 하지만 기존의 여러 연구에서 선후행 조건제약의 모델화에 대한 복잡성으로 인해 TSPPR 문제에 대한 유전 알고리듬 적용에 많은 어려움이 있었다. 본 연구에서는 방향성이 있는 네트웍으로 구성된 TSPPR 문제를 효율적으로 해결할 수 있는 위상정열(topological sort:TS) 개념에 기초한 유전 알고리듬을 제안하고자 한다.

2. TSPPR 문제

TSPPR 문제를 구성하기 위해 두 개의 commodity를 갖는 네트웍 모델[1]의 예를 들어보자. c_{ij}를 정점 v_i 에서 v_j 까지의 운행 거리(경로비용 혹은 Single machine scheduling 문제에서의 Changeover time 등)라 하고 s을 최초 Graph에서 선택된 시작점이라 하자. J 개의 정점을 갖는 네트웍에서 p(Single machine scheduling문제에서 거쳐야 할 공정의 수) 와 q(수행되어진 공정의 수)의 두 가지 commodity가 존재한다고 할 때, p는 시작 정점에서 $(J-1)$ 개의 제품을 공급받아 시작정점을 제외한 다른 모든 나머지 정점에 각각 한 단위로 공급되는 것을 말하며 q는 초기 정점에서 $(J-1)$개를 처리하기 위해(모든 공정을 수행하기 위해) 각 정점으로부터 한 개의 단위로 공급받는 제품을 의미한다고 할 때. 이와 같은 흐름을 나타내는 네트웍은 다음과 같은 두 가지 특성을 갖는

다. 첫 째로 이동 가능한 어떠한 임의의 경로내의 모든 점에서 p와 q의 합은 $J-1$ 과 같아야 한다는 것과 둘 째로 한 정점에서부터 방출(혹은 공급)되는 p(혹은 q)의 양은 경로가 진행됨에 따라 점차 감소(q의 경우 증가)되는 특성이 있다. 이러한 특성들을 이용하여 TSPPR 문제에 대한 모형을 수립할 수 있다.

< 기호 정의 >

J 네트웍내의 정점의 개수.
$i,\ j$ 네트웍내의 임의의 정점(단, $i \neq j$)
c_{ij} 정점 v_i에서 정점 v_j로의 이동거리(비용).
y^p_{ij} 정점 v_i에서 정점 v_j로 p의 이동량.
y^q_{ij} 정점 v_i에서 정점 v_j로 q의 이동량.

$$y_{ij} = \begin{cases} 1, & \text{정점}j\text{가 정점}i\text{ 다음 수행될 때} \\ 0, & \text{이외의 경우} \end{cases}$$

< 모형의 수립 >

$$\text{Min} \sum_{i=1}^{J} \sum_{\substack{j=1 \\ j \neq i}}^{J} \frac{1}{J-1} c_{ij}(y^p_{ij} + y^q_{ij}), \tag{2.1}$$

Subject to

$$\sum_{j=1}^{J} y^p_{ij} - \sum_{j=1}^{J} y^p_{ij} = \begin{cases} J-1, & i=s, \\ -1, & \text{이 외의 경우} \end{cases} \tag{2.2}$$

$$\sum_{j=1}^{J} y^q_{ij} - \sum_{j=1}^{J} y^q_{ij} = \begin{cases} J-1, & i=s, \\ +1, & \text{이 외의 경우} \end{cases} \tag{2.3}$$

$$\sum_{J}^{i=1} (y^p_{ij} + y^q_{ij}) = J-1 \qquad \forall i \tag{2.4}$$

$$y^p_{ij} + y^q_{ij} = (J-1) y_{ij},\ \forall i \text{ and } j \tag{2.5}$$

$$\sum_{j=1}^{J} y^q_{ij} - \sum_{j=1}^{J} y^q_{ij} \geq 1, \quad \forall (v_u \rightarrow v_v)(v_v \neq s) \tag{2.6}$$

$$y^p_{ij} \geq 0 \qquad \forall i \text{ and } j \tag{2.7}$$

$$y^q_{ij} \geq 0 \qquad \forall i \text{ and } j \tag{2.8}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i \text{ and } j \tag{2.9}$$

모든 운송 장비의 총 운행시간(혹은 비용)을 나타내는 목적식에서 모든 운행 가능한 경로($y_{ij}=1$)에서 정점 v_i와 v_j간의 제품 p와 q의 합은 $J-1$과 같으므로 목적식은 식(2.1)과 같이 구성된다. 식(2.4)과 식(2.9)는 p의 네트웍 흐름의 가능해를 보장하며, 식(2.5)는 q에 대한 네트웍 흐름에 대한 가능해를 보장한다. 식(2.6)은 가능한 경로를 보장하며 식(2.7)은 만약 두 개의 정점 v_{ki}와 v_{kj}간 경로가 존재하면(y_{ij} = 1) 정점의 제품 p와 q의 합은 J - 1 이 됨을 설명한다. 또한 식(2.8)은 정점간의 선후행 관계를 표현한다.

3. 유전 알고리즘

3.1 위상정열

선후행 조건 제약을 갖는 TSPPR 네트웍을 G라 할 때, 각 정점(v_i)들은 특정 작업의 수행을 나타내고 정점들을 연결하는 화살표는 각 정점간의 선후행 관계를 나타낸다. TS란 네트웍 G내의 각 정점들에 대한 이동 순서를 결정하는 방법으로 만약 네트웍 내의 임의의 정점 v_i에서 v_j로 이동하는 경로(v_i, v_j)가 존재한다면 v_j는 반드시 v_i가 수행된 후에만 수행될 수 있음을 나타낸다. 단, 네트웍 내의 임의의 정점들 간에 순환경로가 존재하는 경우에 Subtour를 만들어 가는 과정에서 선행 경로가 없는 정점을 찾을 수 없기 때문에 위상정렬을 적용할 수 없게된다. 이러한 TS의 성격을 이용하여 TSPPR 문제에서 모든 정점을 단 한번만 방문하는 최단경로를 찾을 수 있다.

3.2 해의 표현과 초기화 과정

선후행 관계제약이 있는 TSP와 같은 순서를 갖는 문제에 있어서 유전 알고리듬을 이용한 해법 개발시 가장 중요한 사안은 실현 가능해를 표현하는 방법을 만들어 내는 것이다. 선후행조건을 갖는 경로를 표현하는 방법은 매우 어려운 작업이다. 최단경로를 찾기 위해서 유전자는 주어진 네트웍에서 생성 가능한 모든 위상순서(topological order)로 표현될 수 있어야 한다. 가능해의 표현을 위한 방법은 다음과 같다.

procedure: a feasible path generation
  input: directed graph.
  while (any vertex remains) **do**
    if every vertex has a predecessor,
    then the network is infeasible: **stop.**
    else pick a vertex v with the highest priority
      among vertices with no predecessors;

que<- v;
delete v and all edges leading out of v from the directed graph;
end_while.
end_procedure.

유전 알고리즘은 이렇게 표현된 가능해로 모집단을 구성하여 해를 구하게 되는데, 모집단 구성은 랜덤하게 발생시켜 구성한다.

3.3 진화와 선택 과정

해를 개선시키기 위해서 각각의 유전자는 특정한 척도에 의해서 평가되어져야 하는데, TSPPR 에서 유전자의 적합도는 경로간 이동거리(비용)을 최소화 할 수 있는 최단 경로를 갖도록 하는 유전자를 찾아내는 것과 같으므로 적합도 함수는 식(2.1)에서 주어진 목적함수식의 값과 같다.

선택과정이란 다음 세대의 유전자(Child)를 만들기 위해 현세대의 모집단 집합으로부터 두 개의 부모가 되는 유전자(Parent)를 선택하는 과정을 말한다. 일반적인 경우 Roulette wheel과 우성유전자 선택법(Elitist Selection)이 주로 채택되며 본 연구의 경우 임의의 선택법이 채택되었다.

3.4 교배연산

교배과정은 사실상 유전 알고리듬의 수행도와 효율성을 결정 짓는 가장 중요한 해의 변형과정으로 새로이 개발된 *Moon Crossover* 가 개발되었다. 새로운 교배과정에 이와 같은 이름이 붙은 이유는 새로운 유전자의 성장과정이 달의 생성과정(초승달 → 반달 → 보름달)과 유사하기 때문이다. *Moon Crossover* 방법의 전체적인 절차는 다음과 같다.

Procedure *Moon Crossover*
Begin
$osp \leftarrow$ null;
k ← 0;
Select two random chromosome p_a and p_b, where
$p_a = g_1, g_2, g_3 \ldots, g_j$
$p_b = q_1, q_2, q_3, q_4, q_5, \ldots\ldots, q_j$;
Select two genes from the p_a at random. The substrings defined by the two genes;
If the length of osp = J **then End;**
Else sub_ p_b←the remaining substring results from the deleting genes which are already selected from the p_a, $i.e.$, sub_ $p_b = p_a - osp$;
Until(length of $osp \neq J$) **Do**
If $i = 1$ **then**
$i = J+1$;
$i \leftarrow i-1$;
$k \leftarrow k+1$, $k = 1, 2, \ldots\ldots$, length of sub_ p_b;
If $g_i \neq q_k$ **then**
$osp \leftarrow < osp, g_i, q_k >$;
Else If $g_i = q_k$ **then**
$osp \leftarrow < osp, g_i >$;
Else If $j = J$ **then**
$i \leftarrow j-1$;
k ← k+1, k = 1,2,3...,length of sub_ p_b;
If $g_i \neq q_k$ **then**
$osp \leftarrow < q_k, g_i, osp >$;
Else $g_i = q_k$ **then**
$osp \leftarrow < g_i, osp >$;
Else Begin
$i \leftarrow i-1$;
$k \leftarrow k+1$,
$k = 1, 2, 3, \ldots\ldots$, length of sub_ p_b;
If $g_i \neq q_k$ **then**
$osp \leftarrow < g_i, osp, q_k >$;
Else $g_i = q_k$ **then**
$osp \leftarrow < g_i, osp >$;
End: End: End: End;

3.5 돌연변이

교환(swap)에 의한 돌연변이 과정이 수행된다. 교환에 의한 유전자 돌연변이 과정이란 한 유전자로부터 임의로 두 개의 염색체를 선정하여 두 염색체가 갖은 값을 상호 교환하는 방법을 일컫는다.

3.6 유전 알고리듬의 수행절차

$P(t)$와 $C(t)$를 현재 세대(t)의 부모와 후손으로 각각 규정한다면 제안된 유전 알고리듬의 전체적인 흐름은 다음과 같다.

Genetic algorithm Procedure :
Begin
$t \leftarrow 0$;
initialize $P(t)$;
evaluate $P(t)$;
While (not termination criteria) **Do**
apply the genetic operation into the $P(t)$ to yield $C(t)$;
evaluate $C(t)$;
select $P(t+1)$ from $P(t)$ and $C(t)$;
$t \leftarrow t+1$;
End;
End:

4. 수치실험

수치 실험을 통해 TSPPR를 풀기 위해 새로이 제안된 유전 알고리듬의 효율성을 검증한다. 다양한 크기의 문제들에 대해 각기 다른 모수(Parameter)들이 주어진 여러 가지 상황에서 여러 번의 동일한 실험이 수행되었다.
실험을 위해 6개의 정점과 6개의 선후행 관계제약을 가지는 TSPPR문제가 그림 1과 같과 같으며, 이에 대하여 표 1과 같은 정점간의 거리 데이터를 이용하겠다.

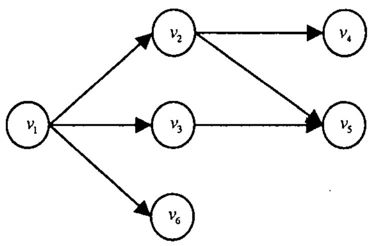

그림 1. TSPPR network

<표 1> 각 정점간 소요 거리(비용)

v_i \ v_j	v_1	v_2	v_3	v_4	v_5	v_6
v_1	-	0	0	0	0	0
v_2	0	-	14	6	10	8
v_3	0	14	-	16	16	10
v_4	0	6	16	-	10	6
v_5	0	10	16	10	-	12
v_6	0	8	10	6	12	-

위 문제를 유전 알고리듬을 이용하여 풀기 위해 필요한 모수들을 다음과 같이 정의한다. 최대 세대수(Maximum Generation) : max_gen = 30, 모집단 크기(Population size) : pop_size = 20, 교배 발생 확률(Crossover probability) : pc = 0.5; 돌연변이 발생 확률(Mutation probability) : pm = 0.2; 동일한 모델에 대한 10회 실험 결과 모든 시행에 대하여 최적해를 도출할 수 있었으며, 최적 경로는 $v_2 \rightarrow v_5 \rightarrow v_1 \rightarrow v_3 \rightarrow v_4$이며, 이에 대한 최적 값이 34임을 알 수 있다. 동일한 문제를 분지한계법(Branch-and-Bound)을 이용한 방법으로 풀었을 경우에도 같은 결과를 얻을 수 있었다.

위의 실험들에 의한 결과로부터 본 연구에서 제안된 유전 알고리듬은 TSPPR에 대해 최적해를 도출해 줄 수 있으며, 분지한계법을 이용한 방법과 비교해 볼 때 계산시간과 선후행조건에 대한 처리에 효과적임을 알 수 있다.

5. 결 론

본 연구에서는 선후행 제약 조건을 갖는 네트웍 문제에 대한 유전 알고리듬을 이용한 효율적 해법에 관한 연구가 수행되었다. 그리고 새로운 교차변이 연산자도 개발되었다. 수치 실험으로부터 TSPPR를 위한 개발된 해법이 효율성이 있음을 보였다.

참고문헌

1. Finke, G, Claus, A. and Gunn. E., A two-commodity network flow approach to the traveling salesman problem, *Congress Numerantium*, 41, 167-178, 1984.
2. Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Design,* John Wiley & Sons, New york, 1997.
3. Potvin, J. Y., Genetic algorithms for the travelling salesman problem, *Annals of Operations Research*, 63, 339-370, 1996.