Graph Coloring Problem 해결을 위한 Ant Colony System의 생성함수 성능비교에 관한 연구

안 상 혁*, 이 승 관, 정 태 충
경희대학교 전자계산공학과 인공지능 시스템 연구실
{fisherking, lee}@iislab.kyunghee.ac.kr, tcchung@khu.ac.kr

Comparison of Constructive Methods In Ant Colony System For Solving Graph Coloring Problem

SangHuck Ahn*, SeungGwan Lee, TaeChoong Chung
Dept. of Computer Engineering, KyungHee University

요 약

그래프 착색 문제(Graph Coloring Problem)는 인접한 노드 (v_i, v_j)가 같은 색을 갖지 않도록 그래프 G의 노드 V에 색을 배정하는 문제로, NP-hard 문제로 잘 알려져 있다. 또한 최근까지 그래프 착색 문제의 최적 해를 구하기 위하여 다양한 접근 방식들과 해법들이 제안되고 있다. 본 논문에서는 기존의 그래프 착색 문제의 해법으로 잘 알려진 Greedy algorithms, Simulated Annealing, Tabu search 등이 아닌 실세계에서 개미들이 자신의 분비물을 이용하여 경로를 찾는 Ant System을 개선하여 새롭게 제안한 Ant Colony System(ACS) 알고리즘으로 해를 구하는 ANTCOL을 소개하고, ANTCOL에서 DSATUR, Recursive Largest First(RLF) 등의 방식을 사용한 기존 생성 함수들과 RLF를 개선하여 제안한 eXtend RLF방식을 사용한 생성 함수를 비교, 평가하고자 한다.

1. 서론

그래프 착색 문제의 많은 해결 방안들은 제시되었지만 그래프 착색 문제는 잘 알려진 NP-hard 문제이기 때문에, 최적 해를 찾는데 지수 시간을 필요로 하고, 문제의 크기가 증가하면 더욱 최적 해를 찾는 것이 어렵게 된다[1]. 또한 최근까지 그래프 착색 문제에서 일정 시간에 좋은 최적 해를 구하기 위하여 다양한 접근 방식의 여러 가지 조합 최적화 문제들의 해법들이 제안되고 있다. 이런 다양한 접근 방식의 조합 최적화 문제들의 해법 중 한가지로 조합 최적화 문제를 Assignment Type Problem (ATP)의 형식으로 표현하여 문제를 해결하는 방안도 제안되었다[2]. ATP는 주어진 제약 조건을 만족하며 n개의 아이템 $i(1 \le i \le n)$에 m개의 자원 $j(1 \le j \le m)$를 최적으로 배정하는 문제로 그래프 착색 문제를 ATP로 적용해보면 n개의 노드 v는 아이템 i, k개의 색 c는 자원 j, 제약 조건은 인접한 노드가 같은 색을 가지지 않게 하는 것으로 생각할 수 있다.

본 논문에서는 이렇게 ATP로 적용 된 그래프 착색 문제를, 실세계에서 개미들이 자신의 분비물을 이용하여 경로를 찾는 Ant System을 개선한 Ant Colony System(ACS) 알고리즘으로 해를 구하는 ANTCOL을 소개하고 ANTCOL에서 DSATUR, RLF 등의 방식을 사용한 기존 생성 함수들과 RLF를 개선하여 제안한 XRLF 방식을 사용한 생성 함수를 비교, 평가하고자 한다.

2. 문제 정의

2. 1 Graph Coloring Problem

노드들의 집합 $V=\{v_1,\ldots,v_N\}$, 간선들의 집합 $E=\{e_{ij} | \exists an\ edge\ between\ v_i\ and\ v_j,\ v_i, v_j \in V\}$인 그래프 $G=(V, E)$에서 그래프 착색 문제는 간선 e_{ij}가 존재하는 인접한 노드 v_i, v_j가 같은 색을 가지지 않게 착색하는 것이다. 노드 v_i에 배정된 색을 $c(v_i)$이라 하면 다음과 같이 표시된다[3].

$$\forall e_{ij} \in E,\ |\ c(v_i) - c(v_j)\ |\ \neq 0$$

2. 2 Ant Colony System[4]

Ant Colony System는 기존의 Ant System을 개선한 Heuristic Search Algorithm이다. Ant System은 개미들이 목적지를 향해 지나가는 경로에 개미들의 분비물을 경로에 쌓고, 이후에 지나가는 개미들은 그 경로에 쌓여있는 분비물을 경로의 정보로 이용하여 다음 경로를 선택하는 원리를 Heuristic Search에 적용시킨 Algorithm이다. 그러나 Ant System은 개미들이 짧은 경로가 존재한다면, 그것만을 선택하고자 하는 성질로 인하여 Local Optima에 빠질 확률이 높기 때문에, 이런 성질을 개선한 Ant Colony System이 새롭게 제안되었다.

Ant Colony System은 개미들이 지나간 경로에 분비물을 기록하는 과정에, 경로가 Global Best Tour에 속해있지 않으면 현재 기록되어 있는 분비물의 양이 증발되는 과정이 추가되었다.

또한 Local Update Rule과 Global Update Rule을 통해 분비물이 update될 때도 인자 값에 의한 계산을 통해 무조건적인 추가가 되지 않도록 하고 있다.

알고리즘을 살펴보면 먼저 각 노드에 초기 개미들이 무작위로 배치되고, 각 간선들의 초기 분비물이 설정된다. 그 후, 각 개미들은 State Transition Rule을 이용해 분비물의 양과 간선의 길이나, Local Optima에 빠지는 것을 방지하기 위해 무작위로 경로를 선택하고, 그때마다 Local Updating Rule을 적용시키므로, 선택한 각 간선의 분비물 양을 update시킨다. 이 과정이 도시의 수만큼 반복되면, 각 개미들의 완성된 Tour들에 대해 Global Best를 찾고, Best Tour에 속해 있는 edge들에 대해 Global Updating Rule을 적용시켜서 그들의 분비물을 update시킨다.

ACS State Transition Rule

$$P_k=\begin{cases}\dfrac{[\tau(r,s)\cdot[\eta(r,s)]^{\beta}}{\sum_{u\in J_k(r)}[\tau(r,u)]\cdot[\eta(r,u)]^{\beta}} & \text{, if } s\in J_k(r)\\ 0 & \text{, otherwise}\end{cases}$$

$$s=\begin{cases}\arg\max_{u\in J_k(r)}\{[\tau(r,u)]\cdot[\eta(r,u)]^{\beta}\} & \text{, if } q\le q_0\\ S & \text{, otherwise}\end{cases}$$

$\tau(r,s)$ = value of Pheromone between r and s

$\eta(r,s)$ = 1 / Length (between r and s)

ACS local updating rule

$$\tau(r,s)=(1-\rho)\cdot\tau(r,s)+\rho\cdot\Delta\tau(r,s)$$

ACS Algorithm for ATP

ATP의 각 단계에서는 두 가지 결정이 이루어져야 한다. 첫 번째는 다음에 선택할 아이템을 정하는 것이고, 두 번째는 선택된 아이템에 자원을 배정하는 것이다. 최적의 해를 위해 각 단계에서 아이템과 자원을 최적으로 하는 확률을 구해야 한다.

ACS를 이용하여 각각의 생성단계 k번째에서 아이템과 자원을 선택한다고 하면 k번째 차례에 아직 배정되지 않은 아이템 i를 선택할 확률 $p_{it}(k, i)$와 k번째 차례에 아직 배정되지 않은 아이템 i에 적당한 자원 j를 선택할 확률 $p_{re}(k, i, j)$는 다음과 같이 표시 할 수 있다.[2]

$$pit(k,i)=\frac{\tau_1(s[k-1],i)^{\alpha_1}\cdot\eta_1(s[k-1],i)^{\beta_1}}{\sum_{o\in\{o(1),\ldots,o(k-1)\}}\tau_1(s[k-1],o)^{\alpha_1}\cdot\eta_1(s[k-1],o)^{\beta_1}}$$

$$pre(k,i,j)=\frac{\tau_2(s[k-1],i,j)^{\alpha_2}\cdot\eta_2(s[k-1],i,j)^{\beta_2}}{\sum_{r\in J_i}\tau_2(s[k-1],i,r)^{\alpha_2}\cdot\eta_2(s[k-1],i,r)^{\beta_2}}$$

$\tau_1(s[k-1], i)$, $\tau_2(s[k-1], i, j)$는 아이템 i와 아이템 i에 배정할 자원 j의 분비물 값이고 $\eta_1(s[k-1], i)$, $\eta_2(s[k-1], i, j)$는 아이템 i와 아이템 i에 배정할 자원 j의 확률 요소 값이다.

2. 3 ANTCOL

ATP로 적용 된 그래프 착색 문제를, 실세계에서 개미들이 자신의 분비물을 이용하여 경로를 찾는 Ant System을 개선한 Ant Colony System(ACS) 알고리즘으로 해를 구하는 ANTCOL을 살펴보면 다음과 같다.

각각 개미들의 집합들은 여러 가지 생성 함수에 의해 그래프를 착색하고, 해 집합 생성에서 이전 단계의 분비물 값을 $n\times n$ 행렬 M에 저장한다면 행렬 M의 값은 순차적으로 갱신된다.

연결되지 않아서 같은 색으로 배정할 수 있는 노드 v_r 와 v_s의 분비물 값을 Mrs에 저장한다. 초기 행렬 M의 값은 모두 1이지만 시간이 지남에 따라 분비물의 증발 변수 $(1-\rho)$ 에 따라 증발한다.

$nants$의 개미들이 한 사이클에서 노드를 착색하여 해 집합 $\{S_a,\ldots,S_{nants}\}$을 구하고, 같은 색으로 칠해지는 노드 v_r, v_s 의 해 집합을 $Srs\subseteq\{S_a,\ldots,S_{nants}\}$라 하고, q_a는 $S_a(1\le a\le nants)$ 에서 사용된 색의 수 일 때, M_{rs}는 다음같이 갱신된다.

$$Mrs=\rho\cdot Mrs+\sum_{Sa\in Srs}\frac{1}{q_a}$$

부분 해집합 $s[k-1]=\{v_1,\ldots,v_n\}$이 주어지고, 색이 배정되지 않은 노드 v_i 배정될 색이 c라고 하면 분비물 값의 계산은 다음과 같다.

$$\tau_2(s[k-1],v,c)=\begin{cases}1 & \text{if } Vc \text{ is empty}\\ \dfrac{\sum_{x\in Vc}Mxv}{|Vc|} & \text{other}\end{cases}$$

ANTCOL의 전체 알고리즘을 살펴보면 표 1과 같다.

표 1. ANTCOL

$Mrs=1\ \forall[v_r, v_s]\notin E$; (연결되지 않은 노드들의 trail 초기화)

$f^*=\infty$ (최적의 노드 색 배정에서 사용된 색의 수)

For ncycles=1 to $ncycles_{max}$ do { // 전체 반복 횟수

$\Delta Mrs=0;\ \forall[v_r, v_s]\notin E;$

For ant=1 to nants do { // nants : 사용되는 총 개미 수

그래프 착색 (생성 함수 사용)

} // q: 생성 함수로 구해진 색의 수

If $q<f^*$ then $s^*=(V_1, \ldots, V_q)$; $f^*=q$;

$\Delta Mrs=\Delta Mrs+1/q\ \forall[v_r, v_s]\notin E;\ \{v_r, v_s\}\subseteq V_j\ j=1,\ldots,q$

$Mrs=\rho\cdot Mrs+\Delta Mrs\ \forall[v_r, v_s]\notin E;$

ANTCOL에서 그래프 착색에 사용되는 생성 함수는 Random, Largest First(LF), Smallest Last(SL), DSATUR [5], Recursive Largest First(RLF)[6] 등이 있다.

Costa and Hertz(1997)는 이런 여러 생성 함수를 비교하고, 대다수의 경우 RLF가 효과적인 해법을 제시함을 보였다[2].

그러나 RLF를 사용한 생성 함수가 그래프의 크기가 증가하면 실행 시간이 다른 생성 함수를 이용할 경우보다 많이 걸리는 것을 보였다. 그래프 착색을 위해 검색하는 노드 수가 많고, 각 단계에서 분비물 값 $\tau_2(s[k-1],v, q)$를 갱신하는데 많은 시간을 필요로 하기 때문에 생성 함수의 복잡 도가 증가한다. 반면에 Random을 사용한 생성 함수는 최적 해의 효율이 나쁜 대신, 실행시간이 아주 적게 걸렸다. 이런 이유로 생성 함수의 선택은 해를 구하기 위한 시간 배정 또한 고려해서 선택해야 한다.

따라서 본 논문에서 기존의 RLF에 Randomization을 적용하여 검색시간을 단축이 기대되는 eXtend RLF를 ANTCOL의 생성 함수에 적용하여 이전의 생성 함수들과 비교하고자 한다.

3. eXtend Recursive Largest First(XRLF) In ANTCOL

XRLF의 알고리즘을 살펴보면 각각의 색 집합 $V_i=\{V_1,\ldots,V_k\}$에 몇 개의 후보 리스트(CL)를 생성하고, 그 중 하나를 선택한다.

각 후보 리스트들은 XRLF를 통해 생성되는데, 각 생성단계에서, 색이 배정되지 않은 노드들 중 현재 칠 할 수 있는 노드의 집합인 W에서 일정 수(CSize)만큼 임의로 색이 칠해지지 않은 노드들을 선택하고, 그 중 최대 차수의 노드를 다음에 색이 칠해질 노드로 결정한다[7][8].

이와 같이 RLF에 Randomization을 적용한 eXtend RLF는 전체 후보 집합(착색을 위해 검색하는 전체 노드 수)에서 일정한 수만큼의 노드를 임의로 선택하여 후보 집합을 크기를 줄이는 것이다.

이처럼 XRLF를 ANTCOL에 적용하여 알고리즘을 표현하면 표 2와 같다. RLF에 [illegible]으로 XRLF를 표현할 수 있다.

표 2. eXtend Recursive Largest First In ANTCOL

```
q=0;  // 사용되는 색의 수
W=V; // 현재 색이 배정되지 않은 착색 가능 노드
k=0;  // 착색된 노드 수
while k<|V| do
k=k+1; q=q+1;
B=∅; // 현재 색이 배정되지 않은 착색 불가능 노드
select first vertex v randomly in W
V_q={v};
// N_w(v) : 노드 v와 연결된 모든 노드 w들의 집합 W
while W\(N_w(v)∪{v})≠∅ do
 k=k+1; B=B∪N_w(v); W=W\(N_w(v)∪{v})
 CL=∅; // (현재 색이 배정되지 않은 착색 가능 노드
          중 후보 노드들)
 if |W|≥CSize {
  for i=1 to CSize do
    select randomly v_0 ∈W
    CL=CL∪{v_0} }
 else CL=CL∪W
 Choose v∈CL Pit(k, v)
  with τ1(s[k-1],v)=τ2(s[k-1],v, q) and η1=deg_B(v);
// deg_W(v) : N_w(v)의 개수.
  V_q=V_q∪{v};
W=B∪N_w(v);
```

4. 생성 함수 성능 비교

Random Graph $G_{n,p}$는 n개의 노드, 노드들 사이에 간선이 독립적으로 확률 p로 존재하는 그래프라고 정의, $p \in \{0.4, 0.5, 0.6\}$인 20개의 $G_{100,p}$, 10개의$G_{300,p}$, 5개의$G_{500,p}$로 실험.

결과는 다음 표 3과 같다.

표 3. 생성 함수들의 성능 비교 ANTCOL

$\alpha=2$, $\beta=4$, $p=0.5$, $ncycles=50$ ( )안의 수는 실행시간				
n	p	RLF	DSATUR	XRLF
		nants=100, 300	*100, 300*	*100, 300*
100	0.4	12.6(0.5) 12.7	12.8(0.5) 13.9	12.5(0.5) 12.6
100	0.5	15.4 15.2(0.5)	15.6(0.5) 16.1	15.3 15.2(0.5)
100	0.6	18.6(0.5) 18.7	18.8 18.7(0.5)	18.6 18.6(0.5)
300	0.4	29.2 28.9(10.1)	36.0 30.4(2.0)	30.1 29.8(4.5)
300	0.5	36.3 35.7(10.2)	44.9 38.1(2.5)	35.6(5.5) 36.0
300	0.6	44.4 43.7(10.3)	55.0 47.3(2.5)	45.1 44.9(6.0)
500	0.4	46.4 45.8(45.7)	54.0 54.0(5.4)	46.8(21.2) 47.0
500	0.5	56.0 55.6(50.1)	68.2 67.8(5.5)	55.8(23.4) 56.2
500	0.6	68.8 68.2(55.5)	84.2 83.8(5.7)	69.0 68.6(24.6)

표 3을 살펴보면, ACS의 인자 값들 α, β, p, $cycle$은 고정하고, 그래프 크기와 연관된 노드 수 n, 노드간 간선 존재 확률 p, 개미 수 $nants$ 값을 조정하면서 최적 해 $\chi(G)$와 실행 시간을 구했다.

그래프 크기의 증가에 따른 실행 시간이 XRLF를 사용한 생성 함수가 RLF를 사용한 경우보다 감소했지만 비슷한 효율의 최적 해를 구할 수 있었다. 이는 XRLF를 사용한 생성 함수에서 노드 착색을 위해 검색할 노드수가 감소되었기 때문이다. 그래프 크기가 증가하면 그만큼 전체 노드를 검색해야 하는 RLF보다 시간이 단축 될 수 있는 것이다. 그러나 XRLF에서는 후보노드를 임의로 선택하므로 최적 해의 효율이 좋지 않을 가능성이 있다. 따라서 실행 시간만을 단축시키는 것보다 단축된 시간동안 착색 수를 줄이는 방법을 연구한다면 더 좋은 최적 해를 구할 수 있을 것이다.

5. 결론 및 향후 연구계획

본 논문에서는 그래프 착색 문제를 정의하고, ACS algorithm을 이용해 그래프 착색 문제의 해를 구하는 ANTCOL을 소개했다.

그리고 DSATUR, RLF를 사용한 기존 ANTCOL의 생성 함수와 XRLF를 사용한 생성 함수의 성능을 비교하고 XRLF를 사용한 생성 함수가 RLF를 사용한 경우와 비슷한 해를 보다 적은 실행시간 안에 구하는 것을 보았다.

XRLF를 이용하여 단축된 시간을 효율적으로 사용하기 위하여 착색 효율도 증가시키는 생성함수를 생각해 봐야 할 것이다. 단축된 시간을 이용하여 착색을 완료한 노드들을 재 검색한다면, 동일한 색으로 착색할 수 있는 노드들을 검색하여 착색 수를 줄일 수 있을 것이다. 이에 관해 자세하고 다양한 연구가 필요할 것이다.

6. 참고문헌

[1] M.R Grey and D.S Johnson "Computer and Intractability: A Guide to the Theory of NP-Completeness", W.H Freeman and Company: New York (1979)

[2] D. Costa and Hertz A. "Ant can colour graphs", J. Op. Res. Society, pp. 295-305, (1997)

[3] R. Dorne and Jin-Kao H. "Tabu Search for Graph Coloring, T-Colorings and Set T-Colorings" Parc Scientifique Georges Besse, Meta-Heuristics 98: Theory & Applications, France. pp.33-48, (1998)

[4] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: optimization by a colony of cooperation agents", IEEE Transactions of Systems, Man and Cybernetics-Part B, vol. 26, no. 2, pp. 29-41, (1996)

[5] D. Brelaz "New method to color vertices of a graph", Com. of ACM, vol. 22: pp. 251-256, (1979)

[6] F. T. Leighton "A graph coloring algorithm for large scheduling problems", J Res. Nat. Bur. Standard, vol. 84: no.6 pp. 489-506, (1979)

[7] D.S Johnson, C.A Aragon, L.A Mcgeoch and C. Schevon "Optimization by Simulated Annealing : An Experimental Evaluation-Part2 (Graph Coloring and Number partitioning)", Operations Research, vol. 31, pp.378-406

[8] M. Laguna and R. Marti "A GRASP for Coloring Sparse Graphs", Technical Report, USA. pp. 1-15, (1997)