그래프 착색 문제에 적용된 효과적인 Ant Colony Algorithm에 관한 연구

안 상 혁[†] · 이 승 관[††] · 정 태 충[†††]

요 약

개미 집단 시스템(Ant Colony System, ACS) 알고리즘은 조합 최적화 문제를 해결하기 위한 새로운 메타 휴리스틱 방법이다. 이것은 그리디 탐색뿐만 아니라 긍정적 피드백에 의한 탐색을 이용한 모집단에 근거한 접근법으로 조합 최적화 문제를 해결하기 위해 제안되었다. 최근까지 인접한 노드(v_i, v_j)가 같은 색을 갖지 않도록 그래프 G의 노드 V에 색을 배정하는 문제인 그래프 착색 문제의 최적 해를 구하기 위하여 다양한 접근 방식들과 해법들이 제안되고 있다. 본 논문에서는 기존의 그래프 착색 문제의 해법으로 잘 알려진 그리디 알고리즘, 시뮬레이티드 어닐링, 타부 탐색 등이 아닌 개미 집단 시스템 알고리즘으로 해법을 구하는 방법인 ANTCOL 알고리즘을 소개하고, ANTCOL을 해결하기 위해 제안된 기존의 생성 함수들(ANT_Random, ANT_LF, ANT_SL, ANT_DSATUR, ANT_RLF)과, 본 논문에서 새롭게 제안된 방법으로 RLF에 무작위 기법을 적용한 XRLF를 생성 함수로 사용한 ANT_XRLF 방법과 ANT_XRLF에 재검색을 추가한 방법(ANT_XRLF_R)의 그래프 착색 결과 및 실행 시간을 비교, 분석하여 제안된 방법이 더 빠르게 수렴할 수 있음을 실험을 통해 알 수 있었다.

A Effective Ant Colony Algorithm applied to the Graph Coloring Problem

SangHuck Ahn[†] · SeungGwan Lee[††] · TaeChoong Chung[†††]

ABSTRACT

Ant Colony System(ACS) Algorithm is new meta-heuristic for hard combinational optimization problem. It is a population-based approach that uses exploitation of positive feedback as well as greedy search. Recently, various methods and solutions are proposed to solve optimal solution of graph coloring problem that assign to color for adjacency node(v_i, v_j) that they has not same color. In this paper introducing ANTCOL Algorithm that is method to solve solution by Ant Colony System algorithm that is not method that it is known well as solution of existent graph coloring problem. After introducing ACS algorithm and Assignment Type Problem, show the way how to apply ACS to solve ATP. And compare graph coloring result and execution time when use existent generating functions(ANT_Random, ANT_LF, ANT_SL, ANT_DSATUR, ANT_RLF method) with ANT_XRLF method that use XRLF that apply Randomize to RLF to solve ANTCOL. Also compare graph coloring result and execution time when use method to add re-search to ANT_XRLF(ANT_XRLF_R) with existent generating functions.

키워드 : 개미 집단 시스템(Ant Colony System), 자원 배정형 문제(Assignment Type Problem), 그래프 착색 문제(Evolutionary Method Graph Coloring), 생성 함수(Construction Method)

1. 서 론

개미 시스템(Ant System, AS)은 잘 알려진 순회 판매원 문제(Travelling Salesman Problem, TSP)를 해결하기 위해 Colorni A, Dorigo M, Maniezzo V(1991, 1992)[1, 2]에 의해 처음 제안되었고, 이 AS를 이용해 이차 배정 문제(Quadratic Assignment Problem, QAP)[3]와 작업 배치 문제(Job Shop Problem, JSP)[4]를 해결하기 위해 많은 알고리즘들이 제안되었다. 그리고 개미 시스템을 개선해 보다 일반적인 개념으로 형식화하고 큰 범위의 조합 최적화 문제의 해를 구하기 위해 개미의 행동을 모방한 개미 집단 시스템(Ant Colony System, ACS) 알고리즘[5]을 제안하였다. 개미 집단 시스템이 해결방안으로 제안된 순회 판매원 문제, 이차 배정 문제, 작업 배치문제 그리고 잘 알려진 그래프 착색 문제(Graph Coloring Problem, GCP)는 자원 배정형 문제(Assignment Type Problem, ATP)의 한 형태로 자원 배정형 문제는 n개의 아이템 $i(1 \le i \le n)$와 m개의 자원 $j(1 \le j \le m)$가 있을 때, 주어진 제약 조건을 만족시키고, 목적함수 $f(x)$를 최적화하는 아이템에 대한 자원의 배정을 결정하는 문제이다[6].

본 논문에서는 개미 집단 시스템을 이용하여 자원 배정

† 준 회 원 : 경희대학교 대학원 컴퓨터공학과
†† 준 회 원 : 경희대학교 대학원 전자계산공학과
††† 정 회 원 : 경희대학교 컴퓨터공학과 교수
논문접수 : 2003년 7월 24일, 심사완료 : 2004년 3월 21일

형 문제의 하나인 그래프 착색 문제를 해결하고자 하며, 개미 집단 시스템을 일반적인 자원 배정형 문제에 적용하는 방법과 자원 배정형 문제로 표현된 그래프 착색 문제의 해를 구하는 방법(ANTCOL)을 소개한다.

그리고 그래프 착색 방법들 중 기존의 생성 함수들을 개미 시스템에 적용한 방법들(ANT_LF, ANT_SL, ANT_DSATUR, ANT_RLF)과 새롭게 제안한 방법을 사용한 생성함수를 이용한 착색 결과 및 실행 시간을 비교하고 결과의 개선을 위한 방안을 모색하고자 한다.

2. 관련 연구

2.1 그래프 착색 문제

일반적으로 그래프 $G=(V, E)$에서 (노드들의 집합 $V=\{v_1, \cdots, v_N\}$, 간선들의 집합 $E=\{e_{ij} \mid \exists$ *an edge between* v_i *and* $v_j, v_i, v_j \in V\}$) 그래프의 q-착색문제는 노드들의 집합 V를 같은 집합 내에 직접 연결된 노드(간선 e_{ij}가 존재하는 두 노드 v_i, v_j)가 없는 q개의 착색 집합 $V_1, \cdots, V_q$으로 새롭게 나누는 것이다.

착색 수(*chromatic number*), $\chi(G)$는 그래프 G가 q색으로 착색 가능할 경우의 최소 값 q이다[7].

2.2 개미 집단 시스템 (Ant Colony System)

개미 집단 시스템(Ant Colony System, ACS)은 개미 시스템(Ant System, AS)에서 출발한 메타 휴리스틱 탐색 알고리즘으로 실제 개미들은 먹이에서 집까지 가장 짧은 경로를 찾는 능력이 있다. 개미들은 목적지를 향해 나아가는 동안 각 경로에 페로몬을 분비하고, 이후에 지나가는 개미들은 그 경로에 쌓여 있는 페로몬 정보를 이용해 다음 경로를 선택하는 원리를 휴리스틱 탐색에 적용시킨 것이 개미 시스템(Ant System, AS)이다. 개미 집단 시스템 알고리즘은 개미 시스템의 성능을 향상시키기 위해 제안 되었다. 일반적으로 ACS는 다음과 같은 방법으로 수행을 한다. 먼저 m개의 개미들이 초기화 규칙에 따라 무작위로 n개의 도시(Node)를 선택한 다음, 각 개미들은 상태 전이 규칙에 따라 다음에 방문할 도시를 선택하고 계속해서 탐색과정을 거친다. 이러한 탐색과정을 거치는 동안 개미들은 지역 갱신 규칙에 따라 방문한 각 경로(edge)에 페로몬 양을 변경하게 된다. 그리고 일단 모든 개미들이 탐색과정을 마치게 되면 전역 갱신 규칙에 따라 다시 한번 페로몬 양을 변경하게 된다[7]. 결국, 개미 시스템처럼 각 개미들은 짧은 경로를 선택하려는 휴리스틱 정보와 많은 양의 페로몬을 가진 경로를 선택하려는 페로몬 정보에 따라 탐색경로를 완성하게 된다.

개미 집단 시스템의 전체 과정을 살펴보면 먼저 각 노드에 초기 개미들이 무작위로 배치되고, 각 간선들의 초기 분비물이 설정된다. 그 후, 각 개미들은 상태 전이 규칙을 이용해 분비물의 양과 간선의 길이나, 지역 최소에 빠지는 것을 방지하기 위해 무작위로 경로를 선택하고, 그때마다 지역 갱신 규칙을 적용시키므로, 선택한 각 간선의 분비물 양을 갱신 시킨다. 이 과정이 도시의 수만큼 반복되면, 각 개미들의 완성된 경로들에 대해 전역 최적을 찾고, 최적 경로에 속해 있는 간선들에 대해 전역 갱신 규칙을 적용시켜서 그들의 분비물을 갱신 시킨다. 계속해서 상태 전이 규칙, 지역 갱신 규칙, 전역 갱신 규칙에 대해 살펴보고 자원 배정 문제에 적용하기 위한 개미 집단 시스템 알고리즘을 소개한다.

2.2.1 상태 전이 규칙

Eq.(1)은 도시 r에 있는 개미 k가 도시 s로 이동할 확률을 나타낸 것으로 상태 전이 규칙으로 불린다. 여기서 $\tau(r, s)$는 도시 r과 s사이의 간선의 페로몬의 양, $\eta=1/\delta$은 $\delta(r, s)$ (도시 r과 s의 거리)의 역수이고 $J_k(r)$은 도시 r에 있는 개미 k가 방문할 수 있는(아직 방문하지 않은) 도시들의 집합. 그리고 β는 페로몬과 간선 길이의 상대적인 중요도를 결정하는 인자 값이다($\beta > 0$).

$$s=\begin{cases} \arg \max_{u \in J_k(r)} \{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}\} & \text{if } q \le q_0 \\ S & \text{otherwise} \end{cases} \quad (1)$$

q는 [0, 1]사이에 정규적으로 분포된 무작위 인자 값이고, q_0는 [0, 1]사이의 값을 가지는 인자, S는 Eq.(2) 식에서 주어진 확률분포에 따라 선택된 무작위 인자 값이다. 이것은 무조건 페로몬과 간선 길이의 연산만으로 다음 도시를 선택하는 기존의 개미 시스템과는 달리, 정규확률분포를 이용해서 무작위적으로 다음 도시를 선택하는 과정이 추가됨으로 인해 개미 시스템이 가지고 있는 지역 최소에 빠지기 쉬운 한계에서 어느 정도 벗어나고자 하고 있다.

$$P_k(r,s)=\begin{cases} \dfrac{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}}{\sum_{u \in J_k(r)} [\tau(r,u)] \cdot [\eta(r,u)]^{\beta}} & \text{if } S \in j_k(r) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

2.2.2 지역 갱신 규칙

개미들은 순회 판매원 문제를 해결하기 위해 각 간선들을 방문하는 동안 아래의 지역 갱신 규칙을 적용시켜 페로몬의 양을 갱신한다.

$$\tau(r,s) \leftarrow (1-\rho) \cdot \tau(r,s) + \rho \cdot \Delta\tau(r,s) \quad (3)$$

$\rho(0 < \rho < 1)$는 페로몬 증발 인자 값이고, L_{nn}이 가장 가까운 이웃에 의해 생성된 경로 길이고, n은 도시 수 일 때 $\Delta\tau(r,s)=(n*L_{nn})^{-1}$는 초기 페로몬의 양이다.

2.2.3 전역 갱신 규칙

전역 갱신은 개미들이 모든 경로를 완성 후에 수행되는데, 가장 짧은 전체 경로를 완성한 개미의 경로에 대해 다음 전역 갱신 규칙을 적용한다.

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau(r,s) \qquad (4)$$

$$where\ \Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1}, & if(r,s)\in global\_best\_tour \\ 0, & otherwise \end{cases}$$

$0<\alpha<1$는 페로몬 증발 인자 값이고, L_{gb}는 현재까지의 전역 최적 경로 길이다. $\tau(r,s)$는 도시 r과 s사이의 페로몬 양으로, 전역 최적 경로에 속해 있으면 $(1/L_{gb})$, 아니면 0으로 주어지며, 결국 증발 인자 α에 의해 페로몬은 줄어들게 된다.

2.3 자원 배정 문제를 위한 개미 집단 시스템

2.3.1 자원 배정 문제에서의 상태 전이 규칙 적용

자원 배정형 문제를 수행하기 위해서는 각 단계마다 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. 자원 배정형 문제에서 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 $j\in J_i$를 선택할 확률 $p_{re}(k,i,j)$를 개미 집단 시스템 알고리즘의 상태 전이 규칙에 적용하면 다음과 같이 나타낼 수 있다[7].

$$p_{it}(k,i) = \frac{\tau_1(s[k-1],i)\cdot\eta_1(s[k-1],i)^{\beta_1}}{\sum_{o\notin\{o(1),\cdots,o(k-1)\}}\tau_1(s[k-1],0,i)\cdot\eta_1(s[k-1],0,i)^{\beta_1}}$$

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

이때 s[k-1]은 자원이 배정된 아이템 $o\notin\{o(1),\cdots,o(k-1)\}$까지의 부분 해이고, $\tau_1(s[k-1],i)$, $\tau_2(s[k-1],i,j)$을 아이템 i와 아이템 i에 배정할 자원 j의 분비물 값이며, $\eta_1(s[k-1],i)$, $\eta_2(s[k-1],i,j)$는 아이템 i와 아이템 i에 배정 할 자원 j의 확률 요소 값이다.

2.3.2 ANTCOL

ANTCOL는 자원 배정형 문제로 적용 된 그래프 착색 문제를 개미 집단 시스템 알고리즘으로 해를 구하는 방법이다. 개미들(*nants*)이 최초의 착색 노드 및 착색할 색, 다음 착색 노드와 착색할 색을 생성 함수와 P_{it}와 P_{re}의 결과로 선택하여 그래프를 착색한다.

착색 해 집합의 생성 단계에서는 먼저 이전의 착색 해들의 분비물 값 (τ_1,τ_2)을 $n\times n$행렬 M에 저장한다. 행렬 M의 값은 모두 1로 초기화 되지만, 시간이 지나면서 갱신한다.직접 연결되지 않아 같은 색으로 배정할 수 있는 노드 v_r, v_s의 분비물 값을 M_{rs}에 저장한다. 분비물의 증발량을 ρ, 개미들이 한 사이클에서 노드를 착색하여 얻은 착색 해 집합을 $\{S_a,\cdots,S_{nants}\}$, 같은 색으로 칠해지는 노드 v_r, v_s의 착색 해 집합을 $S_{rs}\subseteq\{S_a,\cdots,S_{nants}\}$, $S_a(1\le a\le nants)$에서 사용된 색의 수를 q_a라고 하면 M_{rs}는 다음과 같이 갱신된다[7].

$$M_{rs} := \rho\cdot M_{rs} + \sum_{S_a\in S_{rs}}\frac{1}{q_a}$$

부분 해 집합 $s[k-1]=\{v_1,\cdots,v_n\}$이 주어지고 색이 배정될 노드 v, 배정될 색이 c라고 하면, 노드에 배정될 색의 확률 값 $P_{re}(k,v,c)$의 계산식을 위한 노드에 배정될 색의 분비물 값

$\tau_2(s[k-1],v,c)$은 다음과 같이 계산된다.

$$\tau_2(s[k-1],v,c) := \begin{cases} 1 & if\ V_c\ is\ empty \\ \dfrac{\sum_{x\in V_c}M_{xv}}{|V_c|} & otherwise \end{cases}$$

ANTCOL의 전체 알고리즘은 다음의 <표 1>과 같다.

〈표 1〉 ANTCOL

```
Mrs = 1∀[v_r, v_s] ∉ E ; (연결되지 않은 노드 trail 초기화)
f° = ∞ (최적의 노드 색 배정에서 사용된 색의 수)
For ncycles = 1 to ncycles_max do { // 전체 반복 횟수
  ΔMrs = 0 ; ∀[v_r, v_s] ∉ E ;
  For a = 1 to nants do{ // nants : 사용되는 총 개미 수
  그래프 착색 (생성 함수 사용)                    - (A)
 }// q : 생성 함수로 구해진 색의 수
  If q < f° then s° = (V_1,··· ,V_q) ; f° = q ;
  ΔMrs = ΔMrs + 1/q [v_r, v_s] ∉ E ; {v_r, v_s} ≤ V_j, j = 1,···,q
  Mrs = ρ · Mrs + ΔMrs ∀[v_r, v_s] ∉ E ;
}
  Research and local Search(proposal)             - (B)
```

ANTCOL 알고리즘에서 '그래프 착색(생성함수 사용)-(A)' 부분에 사용되는 생성 함수에는 Random, Largest First(LF)[8], Smallest Last(SL)[9], DSATUR[10], Recursive Largest First(RLF)[11]를 개미 집단 시스템에 적용한 ANT_Random, ANT_LF, ANT_SL, ANT_DSATUR, ANT_RLF 등이 있다. Costa and Hertz(1997)[8]는 이런 여러 생성 함수를 사용한 착색 결과와 실행 시간을 비교하였고, ANT_RLF를 사용한 착색 결과가 가장 좋음을 보였다. 결과를 분석해보면 ANT_Random을 사용한 경우 실행 시간이 가장 짧았지만 착색 결과는 가장 좋지 않았다. ANT_RLF를 사용한 경우 그래프의 크기가 증가하면 다른 생성 함수를 이용할 경우보다 실행시간이 오래 걸리는 것을 보였다. 그 이유는 그래프 착색을 위해 검색하는 노드 수가 증가하고, 각 단계에서 분비물 값 $\tau_2(s[k-1],v,q)$를 갱신하는데 많은 시간을 필요로 하기 때문에 생성 함수의 복잡도가 증가하기 때문

이다.

따라서 본 논문에서는 ANT_RLF의 문제점을 해결하고자, 무작위화를 적용한 ANT_XRLF를 제안하고, 그 성능의 우수성을 증명하고자 한다.

3. 제안 사항

3.1 XRLF In ANTCOL(ANT_XRLF)

일반적으로 무작위화는 단순한 휴리스틱 검색 시간을 단축시키기 위한 숫자 설정 방식에 자주 사용되어 왔다. 전체 아이템을 검색하지 않고 무작위로 선택한 아이템들만을 검색하기 때문에 검색하는 아이템의 수가 줄어들기 때문에 검색 시간이 단축될 수 있다.이렇게 무작위화를 사용하면 검색 시간을 단축 시킬 수 있다는 생각으로 Johnson외(1991)[12]는 일반적인 탐색에서 사용되는 RLF에 무작위화를 적용한 방법(eXtended RLF : XRLF)을 제안했다. Johnson이 제안한 XRLF는 일반적인 RLF에서 첫번째로 탐색할 아이템을 단순하게 무작위로 선택하는 것이었다. 본 논문에서 제안하는 그래프 착색의 생성 함수는 Johnson의 XRLF를 개미 집단 시스템에 적용해 그래프 착색 문제를 해결하는 것이다.

ANT_XRLF의 알고리즘의 주요부분을 살펴보면 다음과 같다.

먼저 각각의 색 집합 $V_i = \{V_1, \cdots, V_k\}$에 몇 개의 후보 리스트(Candidate List : *CL*)를 생성하고, 그 중 하나를 선택한다.

각 후보 리스트들은 XRLF를 통해 생성되는데, 각 생성 단계에서, 색이 배정되지 않은 노드들 중 현재 칠 할 수 있는 노드의 집합인 W에서 일정 수(Candidate Size : *CSize*) 만큼 무작위로 색이 칠해지지 않은 노드들을 선택하고, 그 중에서 확률 P_{it}가 최대인 노드를 착색한다. ANT_XRLF는 ANT_RLF와 달리 전체 후보 집합(착색을 위해 검색하는 전체 노드 수)에서 일정한 수만큼의 노드를 무작위로 선택하여 검색할 노드의 후보 집합을 크기를 줄였다. ANT_XRLF의 전체 알고리즘은 <표 2>와 같다.

3.2 XRLF 및 재 검색(Re-Seach)

ANT_XRLF에서는 전체 후보 집합 노드에서 후보노드를 일정 수(CSize)만큼 무작위로 선택하므로 그래프의 전체 구성 노드 수가 적을 경우 착색 결과가 다른 생성함수를 사용한 경우보다 좋지 않다. 따라서 착색 결과의 개선을 위해 일정 수만큼 반복하면서 착색된 노드를 검색하여, 착색된 노드들의 집합과 같은 색으로 가능한 다른 색으로 착색된 노드들을 찾아 합친다면 착색결과를 줄일 수 있을 것이다.

ANT_XRLF를 사용한 착색 결과의 개선을 위해 무작위화를 이용하여 단축한 검색시간으로 착색된 전체 노드를 재검색 한다.

〈표 2〉 ANT_XRLF

```
q = 0 ; // 사용되는 색의 수
W = V ; // 현재 색이 배정되지 않은 착색 가능 노드
k = 0 ; // 착색된 노드 수
while k < |V|
k = k + 1 ;  q = q + 1 ;
B = Ø ; // 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}) Ø {
  k = k + 1 ; B = B∪N_w(v) ; W = W/(N_w(v)∪{v})
    CL= Ø ;
    // (색이 배정되지 않은 착색 가능노드 중 후보 노드들)
  if |W| ≥ CSize {
   for i = 1 to CSize
     select randomly v_0 ∈ W
    CL = CL∪{v_0} }
  else CL = CL∪W
  Choose v ∈ CL Pit(k, v)
  with  τ_1(s[k-1], v) = τ_2(s[k-1],v,q) and η_1 = deg_B(v) ;
  // deg_W(v) : N_w(v)의 개수.
V_q = V_q∪{v} ;
W = B∪N_w(v) ;
```

그리고 같은 색으로 착색 가능한 노드들을 검색하여 재착색 한다. 이 같은 방법을 ANT_XRLF_R이라고 제안한다. ANT_XRLF_R의 과정을 정의해보면 다음과 같다.

먼저 착색 집합들을 착색수의 크기 순으로 정렬하고 가장 작은 착색 집합과 두 번째 작은 착색 집합(V_j, V_{j-1})를 선택하고, 첫 번째 착색 집합 (V_j)과 두 번째 착색 집합의 복사본(V'_{j-1})을 합하여 새로운 착색 집합을 만들고 그것을 s라고 한다. $E(V'_j)$를 착색 노드 집합 V'_j 내부에서 연결된 간선의 집합이라 하고 $f(s)$는 $j = 1$ to k인 경우 $E(V'_j)$의 노드들 개수의 합이라고 정의하고, $f(s)$의 값을 지역 검색을 통해 검색한다. $f(s) = 0$이면 착색 노드들의 결합이 유효한 것으로 이것들의 착색 노드 집합 V'_j을 저장한다. $f(s) > 0$이면, 착색수의 감소가 없는 것이므로 새로 크기가 작은 착색 노드들의 집합 두 개를 선택하고 같은 방법으로 반복한다. 이처럼 단축된 시간으로 착색된 노드들을 검색 하고, 같은 색으로 착색 가능한 노드들을 재착색 한다면 착색결과의 향상을 기대 할 수 있다.

ANT_XRLF_R의 전체 알고리즘은 <표 3>과 같으며, ANTCOL 전체 알고리즘 부분의 Research and local Search (proposal) - (B)에 추가된다.

4. 실험 결과

본 논문은 Costa and Hertz(1997)[8]의 착색 결과 및 실행 시간과 비교하기 위하여 환경과 변수를 동일하게 설정하고 실험하였다. 그리고 ANT_XRLF_R에서 후보집합 노

드(Candidate List)의 크기(CSize)에 따른 성능을 비교 평가하였다.

〈표 3〉 ANT_XRLF_R

```
Research (V, E, i, {V1, ···, Vi})
  do {
   Sort V as |V1| ≥ |V2| ≥ ··· ≥ |Vi-1| ≥ |Vi| ;
     k = i -1
      for j = 1 to k
        V'j = Vj
      s = {V'1, ···, V'k ∪ Vi}
  Local Search(k, s)
   if f(s) = 0 then
       for j = 1 to k ;
         Vj = V'j
       i = k
  } while f(s) = 0

  Local Search(k, s)
   NImprove = 0 // NImprove : 개선 노드수
     while (f(s) > 0 and NImprove < NImpIter) {
                  // NImpIter : 반복 수
        make all possible attempts to switch v to
        different color -(1)
     }
     if (1) success then NoImprove = 0
     else NoImprove = NoImprove +1
```

- 무작위 그래프 $G(n, p)$는 n개의 노드, 노드들 사이에 간선이 독립적인 확률 p로 존재하는 그래프며, 검색 시간은 컴퓨터의 CPU second로 측정한다.개미 집단 시스템의 P_{it}, P_{re}에서 사용하는 변수들은 각각 $\beta_{1,2}=4$, $p=0.5$, $nants \in \{100, 300\}$, CSize = 3으로 고정하고 실험.

- "Michael Trick's Graph Coloring Instances" 그래프들로 실험
  - LEI : Leighton Graphs(Leighton, 1979)
  - MYC : Graphs based on Mycielski transformation (Michael Trick)
  - REG : Graphs based on register allocation for variables in real code(Gary Lewandowski)
  - SGB : Graph from Donald Knuth's Stanford Graph Base

LEI는 착색 수가 5~25이고, 노드간 평균 차수가 11~77인 25개의 150-노드 그래프와 12개의 450-노드의 그래프로 이루어져있다. MYC와 REG는 각각 5개와 14개의 잘 알려진 최적화에 사용되는 그래프들의 집합이다.

4.1 생성 함수에 따른 성능 평가

<표 4>는 무작위 그래프의 착색 수를 보여주고 있으며, <표 5>는 최적화 그래프들의 착색 수를 보여주고 있다. 무작위 그래프에서 실험했을 때 ANT_XRLF_R를 사용한 경우 가장 적은 착색 수로 착색하고 그 다음이 ANT_XRLF를 사용한 경우였다. 최적화 그래프들의 결과도 무작위 그래프에서 실험했을 때와 마찬가지로 ANT_XRLF_R를 사용한 경우 가장 적은 착색 수로 착색하고 그 다음이 ANT_XRLF를 사용한 경우였다. <표 6>은 최적화 그래프들을 착색하는데 걸린 총 실행시간을 보여주고 있다. 실험한 모든 최적화 그래프에서 실행시간이 가장 적게 걸린 것은 ANT_XRLF를 사용한 경우였고 그 다음이 ANT_XRLF_R였다. ANT_RLF, ANT_XRLF, ANT_XRLF_R를 사용한 최적화 그래프들의 착색 결과들을 정리해보면, 재검색을 추가 제안한 ANT_XRLF_R이 단순한 ANT_XRLF보다 재 검색단계를 추가한 이유로 실행시간은 약 40% 늘었지만 착색 결과가 좋음을 보여준다. 결국 ANT_XRLF_R이 ANT_RLF보다 실행시간뿐 아니라 착색 결과도 개선된 것을 보여주는 것이다.

〈표 4〉 무작위 그래프들의 착색결과 비교(color number)

Gn, 0.5 (nants)	ANT_RLF	ANT_XRLF	ANT_XRLF_R
G100 (100)	15.4	16.6	15.3
G300 (100)	36.3	38.3	35.8
G500 (100)	56	52.6	52.5
G100 (300)	15.2	16.2	15.1
G300 (300)	35.7	38.2	35.1

〈표 5〉 최적 그래프들의 착색결과 비교(color number)

name	ANT_RLF	ANT_XRLF	ANT_XRLF_R
LEI	18.3	17.8	16.9
MYC	6.0	6.0	6.0
REG	37.9	37.8	37.4
SGB	31.1	32.0	26

〈표 6〉 최적 그래프들의 실행시간 비교(second)

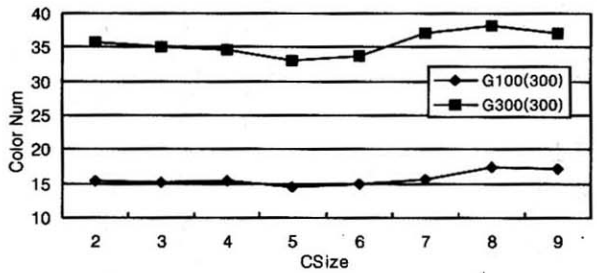
name	ANT_RLF	ANT_XRLF	ANT_XRLF_R
LEI	21.50	10.52	14.56
MYC	0.47	0.24	0.31
REG	16.56	8.93	12.85
SGB	1.05	0.45	0.62

4.2 후보 리스트 크기(CSize)의 변화에 따른 성능 평가

일반적으로 XRLF에서는 후보리스트의 크기를 3으로 고정하여 실험하였다. 본 장에서는 후보 리스트 크기(CSize)에 따른 성능 평가를 위해서 실험하였고, 무작위 그래프 G(n, p)의 변수 값들을 p = 0.5, n = {100, 300}, nants = 300으로 고정하고 후보 리스트(CL)의 크기 CSize = {2, 3, 4, 5, 6, 7, 8, 9}로 실험한 결과를 (그림 1)에서 보여주고 있다. 그림을 살펴 보면 (CSize = 3)인 이전의 실험 결과 G100(300), G300(300)의 착색 결과값(15.1, 35.1)을 기준으로 하여 비교하였으며 CSize가 5인 경우의 착색결과가 가장 좋음을 볼 수 있다.

5. Conclusion

본 논문에서는 그래프 착색 문제를 정의하고, 개미 집단 시스템과 이를 이용하여 그래프 착색 문제의 해를 구하는 ANTCOL을 소개하였다. 그리고 기존 ANTCOL의 생성 함수들 중 가장 효과적이라는 ANT_RLF와 새로 제안한 ANT_XRLF 및 ANT_XRLF_R을 사용한 경우의 착색 결과 및 실행 시간을 비교하였다. ANT_XRLF_R을 사용한 경우 기존의 ANT_RLF보다는 실행시간뿐 아니라 착색 결과도 개선됨을 보였다. 향후 연구과제로는 단순한 ANTCOL에서의 성능 개선이나 비교뿐 아니라 다른 그래프 착색 알고리즘들과의 착색 결과 및 실행 시간 비교도 필요할 것이다.또한 다른 형태의 ATP에서의 적용 가능여부에 관하여 더욱 연구하여야 할 것이다.



(그림 1) CSize에 따른 착색결과 비교

참 고 문 헌

[1] Colorni A, Dorigo M, Maniezzo V "Distributed optimization by ant colonies" Proceeding of the first European Conference on Artificial Life, Paris, MIT Press Bradford Books, Cambridge, pp.134-142, 1991.

[2] Colorni A, Dorigo M, Maniezzo V, "An investigation of some properties of an ant algorithm," Proceeding of the Second Conference on Parall Problem Solving from Nature, Brussels, North-Holland, Amsterdam, pp.509-520, 1992.

[3] Maniezzo V, Colorni A, Dorigo M, "The ant system applied to the quadratic assignment problem," Technical Report 94/28, IRIDIA, Universite Libre de Bruxelles, Belgium, 1994.

[4] Colorni A, Dorigo M, "Ant System for job shop scheduling," Belgian J Opns Res, Stat and Comp Sci, Vol.34. pp.39-53, 1994.

[5] Dorigo M, Maniezzo V, Colorni A, "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.

[6] Ferland J, A, Hertz A and Lavoie A, "An object orientedmethodology for solving assignment type problem with neighborhood search techniques," Opns Res, Vol.44, pp.347-359, 1996.

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

[8] Welsh D. J. A. and M. B. Powell, "An Upper Bound for Chromatic Number of a Graph and its Application to Timetabling Problems," Comput. J., Vol.10, pp.85-86, 1967.

[9] Matula D. W, G. Marble and J. D. Issacson "Graph Coloring Algorithms," on Graph Theory and Computing, R. C. Read, Academic Press, New York, pp.104-122, 1972.

[10] Bralaz D., "New methods to color vertices of a graph," Communications of the ACM, Vol.22, pp.251-256, 1979.

[11] Leighton F., "A graph coloring algorithm for large scheduling problems," J. National Bureau of Standards, Vol.84, pp.489-505, 1979.

[12] Johnson D. S., Aragon C. A, "Optimization by Simulated Annealing : An Experimental Evaluation-Part2(Graph Coloring and Number partitioning)," Operations Research, Vol.31, pp.378-406, 1991.

안 상 혁

e-mail : fisherking@iislab.kyunghee.ac.kr

2000년 경희대학교 전자계산공학과(공학사)

2002년 경희대학교 대학원 전자계산공학과 (공학석사)

2004년~현재 경희대학교 대학원 컴퓨터 공학과 박사과정 수료

관심분야 : 인공지능, 로봇에이전트, 최적화, 스케줄링 등

이 승 관

e-mail : lee@iislab.kyunghee.ac.kr

1997년 경희대학교 전자계산공학과(공학사)

1999년 경희대학교 대학원 전자계산공학과 (공학석사)

2004년~현재 경희대학교 대학원 전자계산공학과(공학박사)

관심분야 : 인공지능, 멀티에이전트, 최적화, 스케줄링 등

정 태 충

e-mail : tcchung@khu.ac.kr

1980년 서울대학교 전자공학과(공학사)

1982년 한국과학기술원 대학원 전자계산 공학과(공학석사)

1987년 한국과학기술원 대학원 전자계산 공학과(공학박사)

1987년~1988년 KIST 시스템 공학센터 선임연구원

1988년~현재 경희대학교 컴퓨터공학과 정교수

관심분야 : 인공지능, 자연어처리, 로봇에이전트, 최적화, 정보 보호 등