* Hybrid Ant Colony System and Flower Pollination Algorithms for Global Optimization

Ku Ruhana Ku-Mahamud
*Introduction
* Ant Colony Optimization
* Flower Pollination Algorithm
* Hybrid ACS and FPA
* Experiment and Results
* Conclusion

*Presentation Outline
*Metaheuristic (coined by Glover, 1986) to introduce tabu search algorithm
*Term used for approximation algorithm
*Nature - source for inspiration to solve problems
*Flower pollination algorithm is a recent metaheuristics algorithm inspired by nature.
Motivation

Exploration in ACS (a variant of ACO) is a stochastic process which has the influence of random search making the exploration inefficient.

FPA algorithm shows strong exploration using Levy distribution.

Hybridization approach of ACS and FPA is proposed
Inspired from biological ants
Marco Dorigo in 1990
Constructive algorithm - ability to construct solution part based on probabilistic mechanism
Ants are not smart but ant colonies are. A colony
- 30 ants to millions of ants
- solve problems unthinkable for individual ant
- Consists of queen, male and female workers
Allocate workers to different tasks (defense, food collection, nest brooming and building, brood care, slave-maker)

Communicate by touch and smell - sniffs with its antennae

Basic concepts
- Graph modelling (bipartite, hamiltonian)
- Edge selection
- Pheromone update (local and global)
- Constructive approach
- Exploration and exploitation

*Ant Colony Optimization*
Finding shortest path to food source
Application areas: routing (VRP, TSP, computer network), assignment & scheduling.

Variants of ACO

- Ant System (AS)
- Elitist Ant System (EAS)
- Rank-Based Ant System (AS_{rank})
- Max-Min Ant System (MMAS)
- Ant Colony System (ACS) - best variant
- Multiple Ant Colony System (MACO)
*ACS algorithm - enhancement over AS algorithm.
*Enhances the exploration mechanism via implementation of aggressive rule.
*During execution, ant that finds the best solution is allowed to deposit a pheromone trail to the arcs belonging to that solution.
*Pheromone that belongs to the arcs used by the ants will evaporate to increase the exploration of alternative arcs.

*Ant Colony System
Probability of ant[k] move from node[i] to node[j]:

\[ p_{i,j}(k) = \begin{cases} 
C \in \left\{ \left[ \begin{array}{c} \cdot \\
\cdot 
\end{array} \right] \right\}, & \text{if } \frac{C}{\sum C} \leq q_0; \\
\frac{\left[ \begin{array}{c} \cdot \\
\cdot 
\end{array} \right]}{\sum \left[ \begin{array}{c} \cdot \\
\cdot 
\end{array} \right]}, & \text{otherwise}; 
\end{cases} \]

- \( q_0 \) - a random variable uniformly distributed in [0, 1] & controls degree of exploitation and exploration
- \( 0 \leq q_0 \leq 1 \),
- \( J \) is a random variable selected according to the probability distribution with \( \alpha=1 \).
*Pheromone update will be applied only by the best-so-far ant (global update)\n\[ \tau \leftarrow (1 - \)\tau + \Delta, \quad \forall (, ) \in , \]
where \( \Delta = 1/\) , and \( \) is the evaporation parameter.

*Pheromone evaporation and pheromone update occur at the same time.

*Another local update is applied after the ant moves on the arc:
\[ \tau \leftarrow (1 - \)\tau + \tau \] where \( , 0 < < 1\), and \( \) is same as the initial value of the pheromone.

**Ant Colony System**
* A metaheuristics algorithm inspired by nature
* For single and multi objective optimization problems.
* Mimics the processes of real-world flowering plants pollination in terms of biotic and abiotic.

* **Flower Pollination Algorithm**
* Two phases

* **Global optimization:** mimics the biotic (plants, animals) and cross-pollination process. The pollinators move using Levy flights method (exploration mechanism).

* **Local optimization:** implements the abiotic (water, light, wind etc.) and self-pollination process using uniform distribution (exploitation mechanism).

* **Flower Pollination Algorithm**
*FPA controls global and local optimization via switching variable.

*Interaction between global and local pollination is by a switch probability $p \in [0, 1]$ which is slightly biased towards local pollination.

*Flower Pollination Algorithm
*Hybrid ACS + GA

*Scheduling of jobs in computational grid

*12 instances from the Expected Time to Compute model

*Best makespan results for 7 instances

*Hybrid ACS and FPA
Two levels of hybridization to overcome the weakness of stand alone algorithm

ACS(FPA)
  * low level (strongly couple) which interchange their inner procedure.
  * enhance the exploration mechanism in ACS algorithm

ACS+FPA
  * high level (loosely couple) where each algorithm preserves their identity
  * provide the refining technique to the final solution produced by ACS algorithm.

Hybrid ACS and FPA
Hybrid ACS and FPA

**Procedure ACS (FPA)**

1. Initialize the number of ants \( n \).
2. Initialize parameters and pheromone trails.
3. Initialize \( q = \text{random} \ [0, 1] \).
4. While (Termination condition not met) Do
   - For \( i = 1 \) to \( n \) Do
     - Construct new solution: \( q = \text{random} \ [0, 1] \).
     - If \( q \leq q_0 \) then
       - Exploration;
     - Else
       - Exploration;
   - End If
   - Apply local pheromone update;

   // Flower pollination algorithm starts here;
   - Initialize a population of \( fn \) flowers with random;
   // one of the solutions is passed from ACS;
   - Find the best solution \( g_s \) in the initial population;
   - Define a switch probability \( p \in [0, 1] \);
   - While (Termination condition not met) Do
     - For \( j = 1 \) to \( fn \) DO
       - If (\( \text{rand} < p \)) then
         - Global pollination;
       - Else
         - Local pollination;
       - End If
       - Evaluate the new solution;
       - If (new solution is better) then
         - Update the population;
       - End If
     - End For
     - Find the current best solution \( g_s \);
   - End While;
   - If (\( g_s \) better than ACS best so far solution) then
     - Update the ACS best;
   - End If
   // Flower pollination algorithm ends here;
   - Apply pheromone evaporation;
   - Apply Global pheromone update;
   - Update best-so-far solution \( s^* \);
   End For;
   - Output the best solution found;

**End Procedure**

**Procedure ACS + FPA**

1. Initialize the number of ants \( n \).
2. Initialize parameters and pheromone trails.
3. Initialize \( q = \text{random} \ [0, 1] \).
4. While (Termination condition not met) Do
   - For \( i = 1 \) to \( n \) Do
     - Construct new solution: \( q = \text{random} \ [0, 1] \);
     - If (\( q \leq q_0 \)) then
       - Exploration;
     - Else
       - Exploration;
   - End If
   - Apply local pheromone update;
   End For;
   - Apply pheromone evaporation;
   - Apply Global pheromone update;
   - Update best-so-far solution \( s^* \);
   End while;

   // Flower pollination algorithm starts here;
   - Initialize a population of \( fn \) flowers with random;
   // The ACS best solution passes to FPA population;
   - Find the best solution \( g_s \) in the initial population;
   - Define a switch probability \( p \in [0, 1] \);
   - While (Termination condition not met) Do
     - For \( j = 1 \) to \( fn \) DO
       - If (\( \text{rand} < p \)) then
         - Global pollination;
       - Else
         - Local pollination;
       - End If
       - Evaluate the new solution;
       - If (new solution is better) then
         - Update the population;
       - End If
     - End For
     - Find the current best solution \( g_s \);
   - End While;
   - If (\( g_s \) better than ACS best so far solution) then
     - Update the ACS best;
   - End If
   // Flower pollination algorithm ends here;
   - Output the best solution found;

**End Procedure**
Start

Initialize the algorithms parameters

Start the main algorithm iterations

End of iterations? Yes

Exploration phase selected? No

Perform exploitation

No

Yes

Start the subordinate algorithm

Display the best solution

End
Generic High Level Hybridization

1. **Start**
   - Initialize the algorithms parameters

2. **Start the main algorithm iterations**
   - End of iterations?
     - Yes
       - Display the best solution
       - End
     - No
       - Exploration phase selected?
         - Yes
           - Perform exploration
           - End of the first algorithm
           - Start the second algorithm
         - No
           - Perform exploitation

   - Exploration phase selected?
     - Yes
       - Perform exploitation
     - No
       - End of iterations?
         - Yes
           - Display the best solution
           - End
         - No
           - Start the main algorithm iterations
*Experiments on job scheduling in static grid computing environment
*Benchmarks problems known as Expected Time to Compute (ETC).
*12 datasets which represent different heterogeneity and consistency.
*A simulator which was developed using C# language
*Results compared with ACS and FPA algorithms
*Performance metrics are flow time and makespan.

*Experiment and Results
Best makaspan value

*Experiment and Results*
Average makespan value

Experiment and Results
Best flowtime value

*Experiment and Results*
Average flowtime value

*Experiment and Results*
Preliminary results show room for enhancement for FPA.

* ACS suitable for discrete problems
* FPA is designed for continuous problems. Hence FPA needs a discretization process.
* Levy distribution (inverse gamma) is complex - longer computational time
* 90 seconds for job scheduling in grid computing. Such a restriction is not practical for Levy distribution.

Conclusion
*Hybridization between other variants of ACO and other metaheuristics algorithms can be developed based on the proposed generic structure.*
Thank you

ruhana@uum.edu.my
ICOCI 2015: International Conference on Computing and Informatics

Welcome

ISSN/ ISI and Scopus Index Proceeding
Since 2009, the Proceedings of ICOCI (i.e. ICOCI 2009, ICOCI 2011, and ICOCI 2013) are recognized with ISSN publication series, ISSN 2289-3784 and have been indexed in ISI and Google Scholar system.

ICOCI2015 eProceedings ISBN No: 978-967-6910-02-4

All papers included in the proceeding will be indexed in Google Scholar. On top of that, ICOCI 2015 Proceedings will be sent to ISI (WoS) and SCOPUS, for evaluation and indexing.

Journal Publication
Selected (extended) papers will be considered for publication in the following journals (subject to reviewers’ evaluation):

- Journal of Information and Communication Technology (JICT), JICT, currently indexed by SCOPUS and Google Scholar, covers all aspects of information and communication technology.