KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 177–184

STIGMERGIC AGENT SYSTEMS FOR SOLVING NP-HARD PROBLEMS

C. CHIRA(1), D. DUMITRESCU(2), AND R. GĂCEANU(3)

Abstract. Problems classified in complexity theory as NP-hard are inherently harder than those that can be solved non-deterministically in polynomial time. Many solutions adopt nature-inspired metaheuristics to solve NP-difficult problems. A hybrid metaheuristic - called Stigmergic Agent System (SAS) - combining the strengths of Ant Colony Systems and Multi-Agent Systems concepts is proposed. The aim of the SAS model is to address NP-hard problems by exploring the solution space using cooperative proactive agents guided by direct and stigmergic communication. Numerical experiments use the Traveling Salesman Problem to evaluate the introduced algorithm. Testing results indicate the great potential of the proposed SAS metaheuristic for complex problems.

1. INTRODUCTION

Metaheuristic techniques refer to strategic frameworks for solving a wide variety of problems as opposed to individual heuristic algorithms designed to solve a specific problem. High-quality near optimal solutions for real-world complex problems can be efficiently identified. Metaheuristics inspired from nature represent a powerful and robust approach to solve NP-hard problems [12].

The aim of this paper is to combine the Ant Colony Optimization [5, 6] approach to solve NP-hard problems with elements of Multi-Agent Systems [4, 7]. A new hybrid metaheuristic called Stigmergic Agent System (SAS) able to better address NP-hard problems is described and investigated. The SAS model involves several cooperating agents able to communicate both directly and in a stigmergic manner to solve problems.

Evaluation results using Traveling Salesman Problems are presented indicating the robustness of the SAS technique.

2. ANT COLONY SYSTEMS

The ACO (Ant Colony Optimization) metaheuristic is composed of different algorithms in which several cooperative agent populations try to simulate real ants behavior [6, 9].

Initially ants wander randomly in order to find food, but they leave pheromone trails in their way. If another ant finds the trail it will likely follow it rather than continue its random path, thus reinforcing the trail.

Over time however, pheromone trails tend to evaporate and thus, in time, longer paths will tend to have a lower pheromone level.

When an ant has to choose between two or more paths, the path(s) with a larger amount of pheromone has(have) a greater probability of being chosen by ants. As a result, ants eventually converge to a short path which hopefully represents the optimum or a near-optimum solution for the target problem.

3. MULTI-AGENT SYSTEMS

The modern approach to Artificial Intelligence (AI) is becoming increasingly centered around the concept of agent.

An agent is anything that can perceive its environment through sensors and act upon that environment through actuators [3]. An agent that always tries to optimize an appropriate performance measure is called a rational agent. Such a definition of a rational agent is fairly general and can include human agents (having eyes as sensors, hands as actuators), robotic agents (having cameras as sensors, wheels as actuators), or software agents (having a graphical user interface as sensor and as actuator).

However, agents are seldom stand-alone systems. In many situations they co-exist and interact with other agents in several different ways. Such a system that consists of a group of agents that can potentially interact with each other is called a multi-agent system (MAS).

The agents of a MAS are considered to be autonomous entities (such as software programs or robots). Their interactions can be either cooperative or selfish [4, 7]. MAS can manifest self-organization and complex behaviors even when the individual strategies of agents are simple.

To share knowledge, agents in a MAS can use an Agent Communication Language such as KQML (Knowledge Query Manipulation Language) [11] or FIPA ACL [10].

4. STIGMERGIC AGENTS

A metaheuristic algorithm called Stigmergic Agent System (SAS) that uses a set of autonomous reactive agents has been proposed [1]. The search space is explored by agents based on direct communication and stigmergic behavior.

4.1. STIGMERGY.

Stigmergy occurs as a result of individuals interacting with and changing a environment [9]. Stigmergy was originally discovered and named in 1959 by Grasse, a French biologist studying ants and termites. Grasse was intrigued by the idea that these simple creatures were able to build such complex structures. The ants are not directly communicating with each other and have no plans, organization or control built into their brains or genes. Nevertheless, ants lay pheromones during pursuits for food, thus changing the environment. Even though ants are not able to directly communicate with each other, they do communicate however - indirectly - through pheromones.

Stigmergy provides a general mechanism that relates individual and colony level behaviors: individual behavior modifies the environment, which in turn modifies the behavior of other individuals.

4.2. **SAS ALGORITHM.** SAS mechanism employs several agents able to interoperate on the following two levels in order to solve problems [1]:

- Direct communication: agents are able to exchange different types of messages in order to share knowledge and support direct interoperation; the knowledge exchanged refers both local and global information.

- Indirect (stigmergic) communication: agents have the ability to produce pheromone trails that influence future decisions of other agents within the system.

The initial population of active agents has no knowledge of the environment characteristics. Each path followed by an agent is associated with a possible solution for a given problem. Each agent leaves pheromone trails along the followed path and is able to communicate to the other agents of the system the knowledge it has about the environment after a complete path is created [1].

The pseudo-code of the SAS algorithm [1] is outlined below:

Algorithm 4.2.1. *Stigmergic Agent System*

```
Begin
    Set parameters
    Initialize pheromone trails
    Initialize knowledge base
    Loop
        Activate a set of agents
        Each agent is positioned in the search space
        Loop
            Each agent applies a state transition rule to incrementally build a solution
            Next move is pro-actively determined based on stigmergic strategy or direct
            communication
            A local pheromone updating rule is applied
            Propagate learned knowledge to the other agents
        Until all agents have built a complete solution
        A global pheromone updating rule is applied
        Update knowledge base (using learned knowledge)
    Until endCondition
End.
```

One of the major properties of an agent is autonomy and this allows agents to take the initiative and choose a certain path regardless of communicated or stigmergic information. Agents can lead the way to the shortest path in a proactive way ensuring that the entire solution space is explored. Agents can demonstrate reactivity and respond to changes that occur in the environment by choosing the path to follow based on both pheromone trails and directly communicated information [1].

Using a purely stigmergic approach the solution of a problem could fall into a local optimum, but due to direct communication ability of the agents they can proactively break out of the local optima and continue to explore the search space in order to find a better solution.

5. SAS EVALUATION

SAS model is implemented and tested for solving the Traveling Salesman Problem.

5.1. PROBLEM STATEMENT.

Given a number of cities and the costs of traveling from any city to any other city, the Traveling Saleman Problem (TSP) refers to finding the cheapest round-trip route that visits each city exactly once and then returns to the starting city.

An equivalent formulation in terms of graph theory is: Given a complete weighted graph (where the vertices would represent the cities, the edges would represent the roads, and the weights would be the cost or length of that road), find a Hamiltonian cycle with the least weight [9].

5.2. THE SAS ALGORITHM FOR SOLVING THE TRAVELING SALESMAN PROBLEM.

SAS algorithm for solving TSP is given below:

Algorithm 5.2.1. *Algorithm SAS for TSP*

```
Begin
    *initialize noOfAgents, stigmergyLevel, startingCity, knowledge base
    While(true) execute
        LaunchAgents(noOfAgents, stigmergyLevel, startingCity);
            * wait until all agents finish execution
            * handle best solution found - a global update rule is applied, that is, update
            the pheromone level and store the best solution found so far
    endWhile
End.
```

Algorithm 5.2.2. *LaunchAgents(noOfAgents, stigmergyLevel, startingCity)*

```
Begin
    For i = 0,noOfAgents do
        Agent agent = createAgent(stigmergyLevel, startingCity)
```

** execute the agent's behavior*
endFor
End.

Algorithm 5.2.3. *Subalgorithm AgentBehavior*
Begin
While (a solution is not found) execute
** proactively determine if the next city should be chosen stigmergically or using direct communication*
** if the agent decides to behave stigmergically then the next city to be visited is chosen using standard ACS ; otherwise the next city to be visited is chosen using direct communication with the other agents*
** handle best solution so far - a local update rule is applied, that is, update the pheromone level*
endWhile
End.

The procedure starts by setting algorithm parameters such as the number of agents, the stigmergy level of the agents, the starting city - the knowledge base of the system in general.

The process runs until certain conditions are met. At the first step agents with the given parameters are launched. Once their task is completed, the best found solution is compared with the best already known solution (if any) and a global update is performed.

For updating the pheromone level the following local update rule (see [2]) is used:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\frac{1}{n * L^+}, \tag{1}$$

where τ_{ij} represents the stigmergy level of the edge (i, j) at moment t, ρ is the evaporation level and L^+ is the cost of the best tour.

The global update rule is similar:

$$\tau_{ij} = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t), \tag{2}$$

where $\Delta\tau_{ij}(t)$ is the inverse cost of the best tour.

Agents are autonomous entities meaning that they can choose to ignore the path communicated by the system and proactively choose another city to explore. This is crucial for the SAS success since only using a purely stigmergic approach the solution of a problem could be trapped into a local optimum.

The algorithm allows stigmergic selection of the next city based on the probability (see [2]):

$$p_{ij}^k = \frac{\tau_{iu}(t)[\eta_{iu}(t)]^\beta}{\sum_{o \epsilon J_i^k} \tau_{io}(t)[\eta_{io}(t)]^\beta}, \tag{3}$$

where J_i^k represents the unvisited neighbors of node i by agent k, $\eta_{io}(t)$ is *visibility* and denotes the inverse of the distance from node i to node o and β shows what is more important between the cost of the edge and the pheromone level.

Using direct communication agents can proactively choose another city to explore. So at a certain point in time if an agent decides that it should use direct communication it can ask the other agents if they have already visited a certain city. This way an unexplored city can be identified and the agent can autonomously choose it as its next move (regardless of pheromone trail intensities).

Cooperating proactive agents capable of both direct and stigmergic communication provide a robust way to find a solution greatly reducing the risk of being trapped into local minima.

5.3. **SAS - A CASE STUDY.** The SAS algorithm for solving TSP was tested for different data sets (see Figure 1).

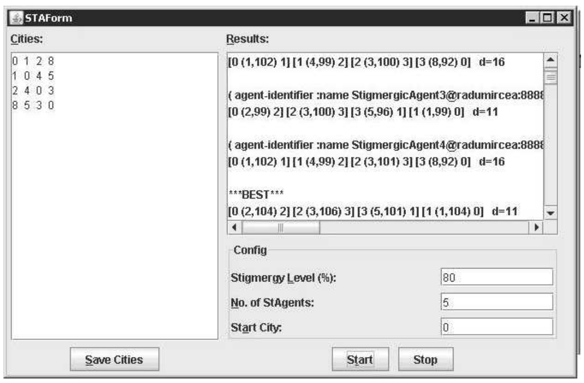

FIGURE 1. SAS application for solving TSP.

In the left pane the cities to be visited are given. The stigmergy level of agents, the number of agents and the starting city can be dynamically configured.

The agents stigmergically find paths of length 16 or 19 at the first iteration. On the given example the agents would be blocked in a local minimum given by nodes [0, 1, 2, 3] of length 16 if they would only act stigmergically. However, at certain moments in time, agents proactively decide to go on another path then the one indicated by the pheromone level, thus avoiding the local optimum.

In the above example, at the second iteration an agent proactively found a better solution (actually the best one) [0, 2, 3, 1] of length 11 and a global update rule was applied for making this new information available to all the other agents.

As the program is running one would observe that more and more agents tend to follow the better path, indicating that the best solution has been probably identified.

Running an ACS algorithm on the same testing data would cause the ants to be trapped in a local minimum of length 16 (when the best tour is of length 11).

5.4. **NUMERICAL EXPERIMENTS.** SAS algorithm for solving TSP is compared to standard Ant Colony System (ACS) model. In the ACS algorithm the values of the parameters were chosen as follows: $\beta = 5$, $\rho = 0.5$.

Table 1 presents comparative results of the proposed SAS algorithm and the ACS model for solving some instances of TSP taken from [5].

Problem	**Number of Agents**	**Number of Generations**	**Best Known Solution**	**ACS Result**	**SAS Result**
swiss42	3	30	1273	1589	**1546**
swiss42	5	30	1273	1539	**1517**
swiss42	10	30	1273	1491	**1489**
swiss42	15	30	1273	1472	**1470**
swiss42	20	30	1273	1472	**1439**
bays29	3	30	2020	2312	**2312**
bays29	5	30	2020	2288	**2225**
bays29	10	30	2020	2288	**2209**
bays29	15	30	2020	2288	**2202**
bays29	20	30	2020	2288	**2177**
gr120	3	30	6942	12271	**11999**
gr120	5	30	6942	12220	**10339**
gr120	10	30	6942	9571	**9548**
gr120	15	30	6942	9488	**9488**
gr120	20	30	6942	9488	**8668**

TABLE 1. Comparative testing results

Numerical experiments suggest a beneficial use of direct and stigmergic communication in cooperative multi-agent systems for addressing combinatorial optimization problems.

6. CONCLUSIONS AND FUTURE WORK

The proposed SAS approach is a powerful optimization technique that combines the advantages of two models: Ant Colony Systems and Multi-Agent Systems.

Interoperation between agents is based on both indirect communication - given by pheromone levels - and direct knowledge sharing, greatly reducing the risk of falling into the trap of local minima.

Ongoing research focuses on numerical experiments to demonstrate the robustness of the proposed model. The SAS method has to be further refined in terms of types of messages that agents can directly exchange. Furthermore, other metaheuristics are investigated with the aim of identifying additional potentially benefic hybrid models.

References

[1] C. Chira., C. M. Pintea, D. Dumitrescu, *Stigmergic Agent Optimization*, Romanian Journal of Information Sciences and Technology, pp. 175 - 183, Vol. 9, No. 3, 2006.

[2] C. M. Pintea, P. Pop, C. Chira, *Reinforcing Ant Colony System for the Generalized Traveling Salesman Problem*, Volume of Evolutionary Computing, International Conference Bio-Inspired Computing - Theory and Applications (BIC-TA), pp 245 - 252, Wuhan, China, September 18- 22, 2006.

[3] N.R. Jennings, K.P. Sycara, M. Wooldridge, *A Roadmap of Agent Research and Development*, Journal of Autonomous Agents and Multi-Agent Systems, Vol. 1, No. 1, 1998, pp. 7-36.

[4] M. Wooldridge, *Intelligent Agents, An Introduction to Multiagent Systems*, ed. G. Weiss, 1999.

[5] The TSPLIB Symmetric Traveling Salesman Problem Instances - http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html

[6] V. Maniezzo, L. M. Gambardella, F. de Luigi - *Ant Colony Optimisation*, New Optimization Techniques in Engineering, by Onwubolu, G. C., and B. V. Babu, Springer-Verlag Berlin Heidelberg, pp. 101-117, 2004.

[7] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2002.

[8] Fabio Bellifemine, Giovanni Caire, Tiziana Trucco, Giovanni Rimassa, *JADE Programmer's GUIDE*, http://jade.tilab.com/, 2006

[9] M. Dorigo, G.D. Caro, *The Ant Colony Optimization Meta-Heuristic*, New Ideas in Optimization, D. Corne, M. Dorigo, F. Glover, 1999.

[10] *http://www.fipa.org/*

[11] T. Finin, Y. Labrou, J. Mayfield, *Kqml as an Agent Communication Language*, Software Agents, B.M. Jeffrey, MIT Press, 1997.

[12] C. Blum and A. Roli, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*, ACM Computing Surveys 35(3) 268-308.

(1) Department of Computer Science Babes-Bolyai University Cluj-Napoca 1B M. Kogalniceanu, 400084, Romania

E-mail address: cchira@cs.ubbcluj.ro

(2) Department of Computer Science Babes-Bolyai University Cluj-Napoca 1B M. Kogalniceanu, 400084, Romania

E-mail address: ddumitr@cs.ubbcluj.ro

(3) Department of Computer Science Babes-Bolyai University Cluj-Napoca 1B M. Kogalniceanu, 400084, Romania

E-mail address: gr90900@scs.ubbcluj.ro