Optimizing Routes Quality and Scattering in the AODV Routing Protocol

Dr. Mustapha GUEZOURI and Abdelaziz OUAMRI
Signal Image Laboratory, Department of Electronics, Faculty of Electrical Engineering, University of Science and Technology (USTO), P.O. Box 1505 – El-m'naouar, Oran, ALGERIA.
e-mail: mguezouri@yahoo.fr

ABSTRACT

An ad-hoc mobile network is a collection of mobile nodes that are dynamically and arbitrarily located in such a manner that the interconnections between nodes are capable of changing on a continual basis. Routing protocols are used to discover routes between nodes. Many mobile ad-hoc networks protocols such as AODV construct route only when desired by the source node (reactively). The advantage hereof is that no prior assumptions of the network topology are required. In highly mobile networks this is an attractive property. Other used protocols (such as OLSR) are said proactive. Such protocols maintain information about routes to all destinations all times. The consequence of this approach is that the amount of control traffic is independent of the actual traffic and mobility in the network.
In this paper we describe three major optimization schemes for the well-known AODV routing protocol in order to get some of the proactive protocols features in it. The targeted characteristics are: traffic independent control and shortest path routes.

Keywords: Manet, Ad-hoc networks, Mobile networks, Wireless networks, Dynamic routing

1. INTRODUCTION

A mobile ad-hoc network (MANET) is a collection of nodes capable of movement and connected dynamically in an arbitrary manner. Nodes of theses networks function as routers which discover and maintain routes to other nodes in the network.
The issue in MANETs is that routing protocols must be able to respond rapidly to topological changes in the network. At the same time the amount of control traffic generated by the routing protocols must be kept at a minimum due to the limited available bandwidth through radio interfaces.
Since the advent of DARPA packet radio networks in 1970's [9] several protocols dealing with the problems of routing in mobile ad-hoc networks have been developed. These protocols may generally be categorized as (a) proactive or table driven [14], [4] and (b) reactive or on demand driven.
Proactive routing protocols attempts to maintain consistent, up-to-date routing information from each node to every other node all times. Theses protocols require each node to maintain on or more tables to store routing information and respond to topological changes by propagating updates through the network.
Thus using a proactive protocol, a node is immediately able to route or drop a packet. Examples of proactive protocols are TBRPF ''Topology Broadcast based on Reverse Path Forwarding'' [18] and OLSR [19] (Optimized Link State Routing protocol''.
Reactive routing creates routes only when desired by the source node. When a node requires a route to a destination, a query is flooded on the network and replies containing possible routes to the destination are returned. Examples of reactive protocols include AODV ''Ad-hoc On Demand Distance Routing protocol'' [15] and DSR ''Dynamic Source Routing'' [2].
In this paper three optimization schemes of the AODV will be presented. Theses optimizations aim in on hand to render the amount of control traffic independent of the actual traffic and mobility in case of high utilization of the network and keep it as low as possible otherwise. In a second hand ensure that the learned routes are the shortest ones in term if hop count.
The reminder of this paper is organized as follows: in section 2, a short overview of the AODV routing protocol is given, emphasizing on the path setup stages and route maintenance. In section 3, we introduce three optimization schemes for the previously described AODV protocol. The paper is concluded in section 4.

2. AODV OVERVIEW

Ad-hoc On demand Distance Vector algorithm [15][16] is described by its authors as a pure on-demand route acquisition system, as nodes that are not on a selected path do not maintain routing information. A node doesn't discover and maintain a route to another node until a communication is needed. Local connectivity is maintained by the use of local broadcast known as *hello messages.*
The path discovery process is initiated whenever a source node needs to communicate with another node for which it has no routing information. Path initiation is done by broadcasting a route request RREQ packet to the neighbours, which then forward the request to their neighbours, and so on, until either the destination or an intermediate node with a route to the destination is located (fig 2.a). AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. The RREQ contains the following fields:
<*source_addr, source_sequence_#, broadcatid, dest_addr, dest_sequence_#, hop_cnt*>.
broadcatid and *source_sequence_#* are incremented whenever the source issues a new RREQ. The pair <*source_addr, source_sequence_#*> uniquely identifies a RREQ. The source node includes in the RREQ the most recent sequence number it has for the destination.
Each neighbour either satisfies the RREQ by sending a route reply (RREP) back to the source or broadcasts the RREQ to its neighbours after increasing the hop count (*hop_cnt*).

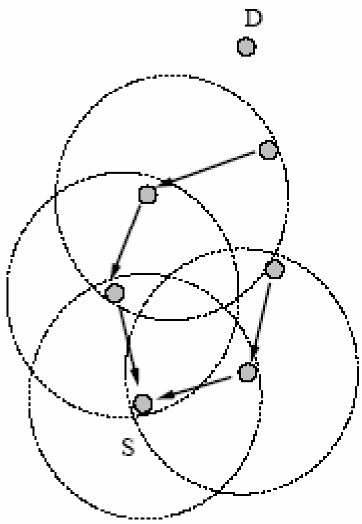

Figure 2. a

An Intermediate node can reply to the RREQ only if it has a route to the destination whose corresponding destination sequence number is greater than or equal to that contained in the RREQ. If a node cannot satisfy the RREQ, it keeps track of the following information to implement the reverse path setup as well as the forward path setup:

Destination IP
Source IP
Broadcast id
Expiration time for reverse path route entry
Source node sequence number

As the RREP is routed back along the reverse path, node along this path setup up forward route entries pointing to the node from witch the RREP came (fig 2.b).
A timer is associated with each route entry in order to delete it if it is not used within a specified lifetime. If a source moves, it is able to reinitiate a route discovery to find a new route. When either the destination or some intermediate node moves, a special RREP is sent to the affected source nodes.
To maintain local connectivity, the protocol uses periodic local broadcasts of hello messages to inform each mobile node of the others nodes in the neighbourhood. The use of hello messages is not necessary; nodes can listen for retransmission of data packets to determine if the next hop is within communication range.

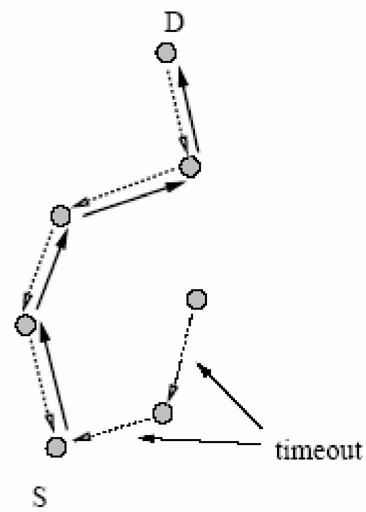

Figure 2. b

3- AODV Optimization schemes
In this section, we propose modified schemes for the AODV routing algorithm. The objectives herein are: (a) reduce the amount of control traffic during high network utilization and mobility periods and make it as possible independent of the actual traffic and (b) get the shortest path for a destination node. Theses schemes rely on modifying the rules a node obey during the reverse and forward path setup stages.

3.1. Reverse path setup
In AODV, to set up a reverse path, a node records the address of the neighbour from witch it received the first copy of the RREQ. This only guarantees a fast setup, and not the shortest path to the source (fig 3.a). In our modified scheme, a node updates the reverse path each time it receives a RREQ request from the source with hop count less than the stored one. This RREQ request is not forwarded.

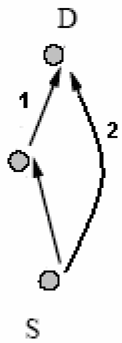

Figure 3. a

3.2. Forward path setup
In AODV, if an intermediate (possibly the destination it self) does have a current route to the destination and if it has not been processed previously then the node unicasts a route reply packet back to the source. This also doesn't guarantee that the forward route to the destination is the shortest one. To ensure the selection of the shortest path, we propose a new scheme in which an intermediate node, replies each time it receive a RREQ with hop count less than the one previously processed.

3.3. Route Scattering

The third scheme we propose in this paper concerns route scattering. Each node on the newly discovered route by the AODV algorithm knows only how to reach the end points of the path and not the other nodes on the same path (fig 3.b). Hence, if two nodes on an active path need to communicate, the whole process must be restarted.

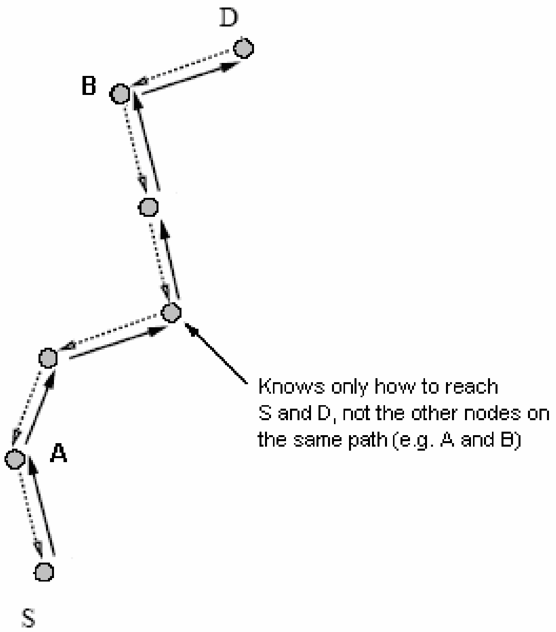

Figure 3. b

To prevent this, we propose to modify the RREQ and RREP to contain an additional field *routers_list.* Upon the receipt of a route request RREQ, each node either satisfies the RREQ by sending a route reply RREP back to the source with *routers_list* containing the IP addresses of all the nodes from the source to the destination or rebroadcasts the RREQ to its own neighbours after adding its address to the *routers_list* and increasing the *hop_cnt* field.

As the RREQ travels from a source to a destination, it automatically sets up reverse path to all the nodes back to the source by using the routers_list field of RREQ (fig3.c).

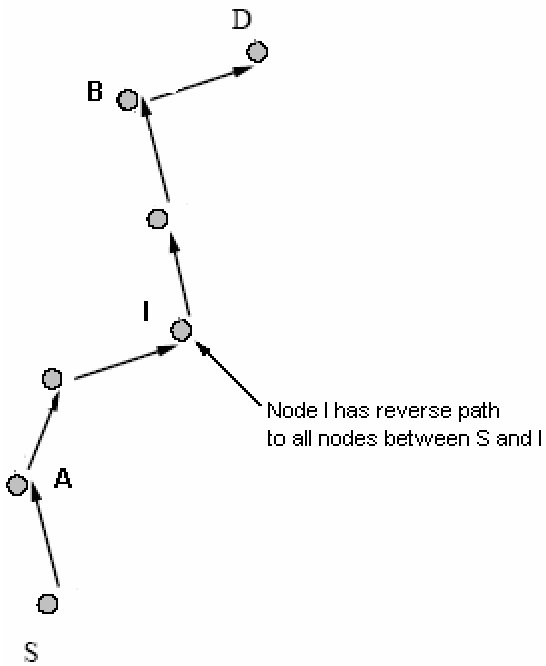

Figure 3. c

The same way, as the RREP travels back to the source, each node along the reverse path sets up a forward pointer to all nodes along the way to the destination. This is also possible because of the *routers_list* in RREP (fig3.d).

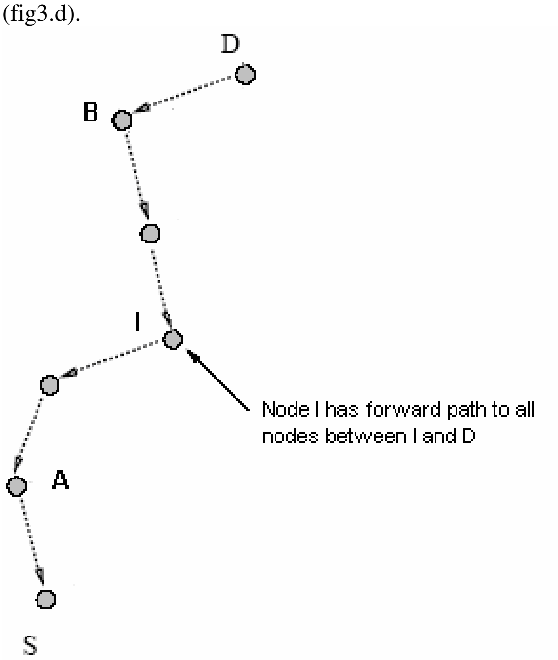

Figure 3. d

Finally, all nodes on the path from the source to the destination broadcast their forward and reverse entries to their neighbours outside the path. This, permit to those nodes in the neighbourhood of the path to use it as a backbone (fig 3.e)

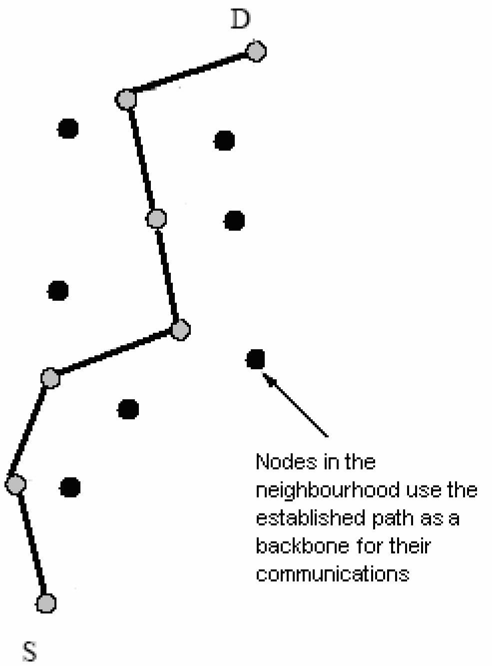

Figure 3. e

4. CONCLUSION AND FURTHER WORKS

Non-optimal routes bring a non negligible overhead that is proportional to the data load of the network. We have shown that the AODV algorithm yields non-optimal routes and then proposed three modified simple schemes with the goal of reducing route length overhead. This is done by first modifying the node's behaviour face of RREQs and RREPs and second by adding a new field to RREQ and RREP. Route scattering is another presented scheme which aims to provide a backbone to the nodes in the neighbourhood of an active path.
Currently, are about specifying the details of the proposed schemes in an Internet Draft to be submitted to the IETF manet working group. Simulation work is in progress to test theses new schemes under different traffic and mobility scenarios.

REFERENCES

[1] D. Baker, M. S. Corson, P. Sass and S. Ramanathm, "Flat vs. Hierarchical Network Control Architecture," ARO/DARPA Workshop on Mobile Ad-Hoc Networking, http://www.isr.umd.edu/Courses/Workshops/MANET/program.html, March 1997
[2] J. Broch, D. B. Johnson, D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad-Hoc Networks," IETF Internet Draft draft-ietf-manet-dsr-01.txt, December 1998 (Work in Progress).
[3] C. C. Chiang, M. Gerla and L. Zhang "Adaptive Shared Tree Multicast in Mobile Wireless Networks," Proceedings of GLOBECOM '98, pp. 1817-1822, November 1998.
[4] C. C. Chiang, H. K. Wu, W. Liu and M. Gerla, "Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel," Proceedings of IEEE SICON'97, pp. 197-211, April 1997.
[5] M. S. Corson and A. Ephremides, "A Distributed Routing Algorithm for Mobile Wirelss Networks," ACM/Baltzer Wireless Networks Journal, Vol. 1, No. 1, pp. 61-81, February 1995.
[6] R. Dube, C. D. Rais, K.Y. Wang, and S.K. Tripathi, "Signal Stability based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks," IEEE Personal Communications, pp. 36-45, February 1997.
[7] M. Gerla, C. C. Chiang, and L. Zhang, "Tree Multicast Strategies in Mobile, Multihop Wireless Networks," ACM/Baltzer Mobile Networks and Applications Journal, 1998.
[8] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks," Mobile Computing, ed. T. Imielinski and H. Korth, Kluwer Academic Publishers, pp.153-181, 1996.
[9] J. Jubin and J. Tornow, "The DARPA Packet Radio Network Protocols," Proceedings of the IEEE, Vol. 75, No. 1, pp. 21-32, 1987.
[10] Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," Proceedings of ACM/IEEE MOBICOM '98, October 1998.
[11] S. Murthy and J. J. Garcia-Luna-Aceves, "An E_cient Routing Protocol for Wireless Networks," ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks, pp. 183-197, October 1996.
[12] S. Murthy and J. J. Garcia-Luna-Aceves, "Loop-Free Internet Routing Using Hierarchical Routing Trees," Proceedings of INFOCOM '97, April 7-11, 1997.
[13] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," Proceedings of INFOCOM '97, April 1997.
[14] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Computer Communications Review, pp. 234{244, October1994.
[15] C. E. Perkins and E. M. Royer, "Ad Hoc On Demand Distance Vector (AODV) Routing," IETF Internet Draft, draft-ietf-manet-aodv-02.txt, November 1998 (Work in Progress).
[16] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999.
[17] A. S. Tanenbaum, Computer Networks, Third Edition. Prentice Hall, Englewood Cliffs, Chapter 5, pp. 357-358, 1996.
[18] Philippe Jacquet, Paul Muhlethaler, Amir Qayyum, Anis Laouiti, Laurent Viennot and Thomas Clausen. "Optimized Link-State Routing Protocol," Technical report, Project HIPERCOM, INRIA Rocquencourt, March 2001, draft-ieft-olsr-04.txt – Work in progress.
[19] Richard G. Ogier, Fred L. Templin, Bhargav Bellur, and Mark G. Lewis. "Topology broadcast based on reverse-path forwarding (tbrpf)," Internet Draft, draft-ietf-manettbrpf-03.txt, November 28 2001, Work in progress.

Received: Jun. 2006. Accepted: Aug. 2007.

Using ant colonies to solve time-table problems

Fahima NADER
Laboratoire LMCS
Institut National de formation en Informatique,
BP 68M, 16270, Oued Smar, Algérie.
Email : f_nader@ini.dz

ABSTRACT

Decision Support Systems (DSS) are a constantly growing area. More and more domains of the daily life take advantage of the available tools (medicine, trade, meteorology…). However, such tools are confronted to a particular problem: the great number of characteristics that qualify data samples. They are more or less victims of the abundance of information. On the other hand, sat domain benefits from the appearance of powerful solvers that can process huge amounts of data in short times.
This paper presents an approach for translating timetable problems (which are a particular case of DSS), into a Boolean formula which is then provided to an environment that allows executing an artificial ants algorithm in order to find solutions that satisfy a maximum number of clauses (Max-Sat problem). Finally, the best solutions are back-translated into the original problem in order to find an adequate schedule that satisfies the characteristics and constraints of the timetable problem.

Keywords: Decision Support Systems, Timetable, Satisfiability, Optimization, Ant Colony Systems.

1. INTRODUCTION

The Decision Support System (DSS) concept was born at the USA during the seventies, to help managers in the decision process.
DSSs have been developed to solve decision systems that have been few or badly structured [10], and that have at least one of the following characteristics: decider preferences are essential; the criteria for making a decision are numerous, they raise a or the problem evolves rapidly [15].
There exist a great number of DSS methods, based on fundamentally different principles. From a mathematical point of view, the main difficulty that occurs in DSS is the problem formulation [21].
From an organizational point of view, the main problem is the identification of the actors and their relation [18]. These two aspects are complementary, since on one hand, the choice of a DSS method requires a deep knowledge of the decision context, and on the other hand, the materialization of the result is conditioned by the opportunity of the chosen approach [19].
[23] tackled the search for an optimum. He explains that the existence of an optimal solution is conditioned by three constraints: exclusivity, exhaustivity and transitivity of the actions.
However, the decider preferences are often fuzzy, incompletely formulated and non-transitive. Besides, they tend to evolve during the decision process.
[22] showed the limits of this type of methods in the resolution of timetable problems, which is the problem our works deal with.
This paper presents an approach for translating timetable problems, expressed as sets of positive and negative examples, into Boolean formulas composed of CNF (Conjunctive Normal Form) clauses. The Sat problem obtained is then provided to an environment that allows executing ant colonies in order to extract solutions that satisfy a maximum number of clauses (Max-Sat problem). The best solutions are back-translated into cases that are applied to the data sets in order to extract the pertinent information solving the original learning problem.
Section 2 gives an overview about timetable problem. The third one introduces the Sat/Max-Sat problem. Section 4 references the translation method used to transform a set of examples into a CNF formula. Section 5 presents the approach used to solve the sat problem. The sixth section gives some results obtained on a real benchmark.

2. TIMETABLE PROBLEM

Timetable problem was defined by [4] as the follows: "Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives".
This problem is strongly constrained. There are two types of constraints: the hard (imperative) ones and the soft (desirable) ones [22].

Imperative constraints: They must not be violated or relaxed in any case. For example, a section (set of student following the same lecture at the same time) must not be scheduled for more than one lecture at the same time. A section cannot be in two different rooms at the same time.
Soft constraints: They are desirable but not essential. For example, the daily scheduled hours should be limited for a section. A lecture should be scheduled during the morning.

3. DESCRIPTION OF THE STUDIED PROBLEM

The problem we are dealing with in this paper is a course timetabling. It can be expressed as follows:
A set of subjects S ={s1, …, sn}; each teacher is associated to one or more subjects according to his skills.
A set of time slots called periods P = { p1, …, pm}. There two morning periods and three afternoon ones per day.
A set of rooms R = {r1, …, rl}

In order to build a timetable, on must take into account the following imperative constraints:

1. During a given period, the following elements: lecture units, sections (sets of student groups following a same lecture at the same time) and rooms must appear no more than one time.
2. A scheduled lecture unit must be done exactly one time.
3. The load of a room must be respected.
4. The same lecture cannot be scheduled in two different rooms at the same time;
5. The same room cannot be used for two different lectures at the same time;
6. A section (or a group) cannot be in two different rooms at the same time;

And the following desirable constraints:
1. Avoid to schedule a lecture twice in the same day;
2. Avoid the blanks in section schedules;
3. Avoid to overload a professor during one day;

A lecture should be scheduled during morning periods

4 SATISFIABILITY AND THE MAXIMUM SATISFIABILITY PROBLEM (SAT/MAX-SAT)

In the theory of complexity, the problem of Satisfiability (SAT) of a Boolean formula plays a very significant theoretical and historical role. It is from this problem and the concept of nondeterministic Turing machine that Cook defined the NP-Complete class [5]. Since then, a whole branch of theoretical computer science is dedicated to the study of NP-completeness [12]. This interest is due to the fact that very significant problems like partitioning, scheduling belongs to this class [16]. The SAT problem is central in the class of NP-Complete problems. Several problems having practical applications can be reduced to a SAT one in a polynomial time [12]. Moreover, domains of application of this problem are large such as: the integrity and the consistency of data bases [11] and inconsistency of knowledge bases in expert systems [20, 13].

We define in the following the problem of Satisfiability of a Boolean formula (SAT problem). Let X= {x1, x2 … xn} and C = {C1, C2 … Ck} be respectively a set of n variables and a set of k clauses. F is a Boolean formula in its conjunctive normal form (called system SAT) if

F = ? Ci (1 ? i ? k) where each Ci = ? xj (1 ? j ? n);

xj being a literal (a propositional variable or its negation). F is said to be satisfiable if and only if there exists a truth assignment I such as I(F) is true, I being a function which associates to each variable a truth value (Yes or No).

The inconsistency of a SAT system (i.e. its nonsatisfiability) leads us to ask the following question: "How many clauses of F can be satisfied simultaneously?" This problem is called the maximum satisfiability problem (Max-Sat). The Max-Sat problem is obviously a problem of optimization; it has been classified as a NP-hard problem [1].

5.ENCODING TIMETABLE PROBLEMS AS SAT FORMULAS

The characteristics and constraints of the timetable problem can be expressed as a set of examples in a truth table. Each line of this table corresponds to a positive or a negative example. An example is considered positive if it corresponds to a soft constraint. It is considered negative if it corresponds to a hard one. Although this truth table is not exhaustive (since all the combinations cannot be coded), the amount of data generated grows exponentially, making it impossible to manually deal with. Note that

On the other hand, interest in Satisfiability (SAT) is always on the rise, not only because it is a central problem in NP-completeness, but also because of the recent availability of powerful tools that are sufficiently efficient and robust to deal with the large-scale SAT problems.

This is why we proposed to translate the timetable problems to Sat ones. It will then be possible to: experiment number of general heuristics to try solving them, and overcome the limitation of the number of variables imposed by classical DSS approaches.

Translation approach

Note that the translation is not as easy as it may appear. If the timetable problem is described with a set of examples that constitute an exhaustive one (all the cases are present in the truth table), then the translation can be solved using De Morgan rules. However, if, as it is the case in a wide majority of the problems encountered, the truth table is incomplete, the approach proposed in [14] can then be used. Using the sets of translation rules described in these works, we developed a tool that automatically translates an incomplete truth table of positive and negative examples to a CNF formula that can be solved by Sat tools.

The number of variables (NV) and clauses (NC) generated in the CNF formula are calculated as follows:

$$NV = K * (2 * N + A)$$

$$NC = K * (N * (A+1) + R) + A$$

With: N: number of variables in the case description of the original problem; A: number of positive examples in the original problem; R: number of negative examples in the original problem; K: number of product terms.

6. SOLVING THE TIMETABLE PROBLEM WITH ANTS

As mentioned earlier, the training set used as benchmark in this study is extracted from a real case: the timetable of the National Computing Institute of Algiers.

Using the approach referenced in 4.1, the translation of the benchmark gave a CNF formula with 410 variables and 1081 clauses.

An execution model, named PARME (Partitioning and Max-sat Environment) [2, 3, 17] was used

Heuristic used

In PARME, different meta-heuristics have been implemented, such as: simulated annealing, taboo search, genetic algorithms, ant colony algorithms as well as several dedicated algorithms. Each one was implemented with a set of strategies. In this paper, we will be focusing on ant colony optimization method (ACO).

Research on the collective behaviors of the social insects provides to computer scientists powerful methods for the design of optimization algorithms.

In addition to their capacity, already surprising, to solve a broad spectrum of problems, these techniques offer a high degree of flexibility (the colony adapts to the abrupt changes of environment) and of *robustness*, (the colony continues to function when certain individuals fail to achieve their task). They solve in a more effective way problems of optimization, such as the problem of the quadratic assignment.

The algorithms of optimization by ant colony are inspired by the behavior of the real ants [7]. The ants are social

insects; therefore, they live and behave for the survival of the whole colony rather than for the survival of only one individual.
The ACO (Ant Colony Optimization) is a meta-heuristic to solve optimization problems introduced by Marco Dorigo [7, 8]. The artificial ants used in the ACO are procedures of construction of stochastic solutions, by adding iteratively components to the partial solution, by taking into account (i) heuristic information on the instances of the problem to be solved, if they are available, and (ii) the track of pheromone, which changes dynamically [9]. It is significant to note that the ants move jointly and independently, and that each ant is complex enough to find a solution to the problem considered. Typically, the solutions of good quality are result of the collective interaction of the ants.
The steps of the Ant Colony System (ACS) we used are given in figure 2.

Begin
1-Initialize value of pheromone on components/connections to the value $?_0$ given by the user
While not stopping criteria do
  2-Position the ants on the starting nodes
  Repeat
    For each ant do
      3- Choose the next literal to be assigned either randomly or choose the literal that satisfies the maximum clauses
      4- Put this literal in a taboo list
      5-Apply step-by-step pheromone update by decreasing the value of the pheromone on the component/connection to make it less attractive for coming ants (Diversification strategy)
    End for
  Until the solution is built
  6- Update the best solution found
  7- Improve the solution using a local search algorithm (GSAT)
  8- Offline pheromone update. */* We either decide to intensify the search by adding extra pheromone on the component/connection to make it more attractive or diversify the search by decreasing the value of the pheromone if a stagnation of the solution is detected*
End do
End.

Figure. 1. *ACS Procedure.*

7. SIMULATIONS AND RESULTS

In this section, we will present the tests and results obtained by the GA and ACS on the medical benchmark described in section 4.2.
All the tests have been run on an Intel Pentium IV, 2GHz with a 256 Mo RAM. This benchmark is a formula consisting of 68 variables and 346 clauses.
Table 1 summarizes test parameters and results for the considered benchmark.
These tests allowed us to tune the parameters of the ACS. We noticed that the simulation that gave us the best results was the first one. The maximum number of ants to be used is : 2n ants (n being the number of variables). The pheromone was put on the component. The candidate list should not exceed 30. If we increase that size, the performance of the algorithm decreases. The choice of the next literal to be assigned is based on a heuristic. We proposed two different ones: a static and a dynamic one. We noticed that the latter gave best results.
The parameters ?0, ?, Q0, ? and ? should be well chosen by the user. The best values are given in the first entry of table 1.

S	NA	NG	H	LS	?0	?	?	?	Q0	EC	ET
S1	100	5	D	100	0.3	0.1	0.5	0	0.99	**1081**	46
S1	820	2	D	20	0.1	0.1	0.1	4	0.8	*1080*	**29270**
S1	70	5	D	10	0.1	0.1	0.01	1	0.8	*1078*	457
S1	100	5	D	100	0.1	0.1	0.01	1	0.8	**1081**	386
S2	100	5	D	100	0.3	0.1	0.5	0	0.99	*1080*	18135

S: Strategy (S1: pheromone on component, S2: pheromone on connections, S3: pheromone on connections with sums); **NA**: Number of Ants; **NG**: Number of Generations; **H**: Heuristic; S: Static; D: Dynamic; N: without heuristic; **LS**: Candidate List Max Size; **?0**: initial value of pheromone; ?: heuristic; **Q0**: balance between exploitation and exploration; ?: pheromone persistance; ? : pheromone decadence; **EC**: Elite Solution Cost; **ET**: Execution Time (seconds).

Table 1: *ACS executed on the benchmark.*

Notice that ACS, when having the values of the parameters of simulation 1 (table 1), could satisfy in 100 per simulation the optimum solution (1081 clauses out of 1081).
This is due to the approach used: each ant in the ACS builds its own solution based on the pheromone it finds on its way (pheromone put by the other ants). The ants cooperate to find the best solution.
Moreover, the ACS has two interesting strategies: diversification and intensification that are used in step 8 of the algorithm (Figure. 2). If stagnation is noticed (the best solution did not change) after a given number of iterations then we will diversify the search by decreasing the value of the pheromone on components/connection of the best solution found. Otherwise, we will intensify the search by adding extra pheromone on it.
The number of ants has a big influence on the execution time, as showed in simulation 2 (table 1) where the algorithm takes more than 8 hours.

8. CONCLUSION

This paper introduced an approach that uses ant colonies to solve timetable problems. The data was first translated into a Sat system in order to benefit from a parallel environment named PARME that was designed to solve Sat/Max-Sat problems.
The advantage of translating the timetable problem into a Sat one is essentially to be able to deal with great numbers of variables and large data sets.
An other advantage of using PARME is that, unlike many other tools that exclude the expert from the learning process, it is possible here to choose a certain number of solutions that the latter finds "satisfying", according to his knowledge and skills. Besides, too dedicated tools are difficult to generalize as soon as small changes are performed on the data sets. The translation to sat problems and the use of general heuristics avoid this kind of drawback.

9. REFERENCES

1 André P., "Aspects probabilistes du problème de la satisfaction d'une formule booléenne. Etude des problèmes SAT, #SAT et Max-SAT". Thèse de doctorat de l'Université Paris VI, 1993.

2 Benatchba K., Koudil M, Drias H., Oumsalem H. et Chaouche K., "PARME UN ENVIRONNEMENT POUR LA RESOLUTION DU PROBLEME MAX-SAT", CARI'02, OCT. 2002.

3 Benatchba K., Admane L., Koudil M. and Drias H. "Application of ant colonies to data-mining expressed as Max-Sat problems", *International Conference on Mathematical Methods for Learning, MML'2004, Italy,* June 2004.

4 Burke E., Kingston J., Jackson K. and Weare R., "Automated University Timetabling: The State of the Art", The Computer Journal 40 (9) 565-571, 1997.

5 Cook S.A., "The complexity of theorem-proving procedures". Proc. 3rd ACP symp. On theory of computing Association of Computing Machinery, New York, p151-1158, 1971.

6 Dorigo M. and Gambardella L.M., "Ant Colony system : A cooperation learning approch to the traveling salesman problem. *IEEE Trans. Evol. Comp*., 1(1):53-66, 1997.

7 Dorigo M. and Dicaro G., "Ant Colony Optimisation : A new meta-heuristic", IEEE, 1999.

8 Dorigo M. and DicaroG., "Ant Colony Optimisation meta-heuristic: In D. Corne, M. Dorigo, et F. Glover, editors, *New Ideas in Optimization*, pages11-32. McGraw-Hill, 1999.

9 Dorigo M., "The Ant Colony Optimization Metaheuristic:Algorithms, Applications, and Advances". Université Libre de Bruxelles, IRIDIA.Thomas Stützle, TU Darmstadt, Computer Science, Intellectics Group, 2001.

10 Eierman M. A., Niederman F. et al., "DSS theory: a model of constructs and relationships", Decision Support Systems 14(1): 1-26, 1995.

11 Gallaire H., Minker J. and Nicolas J.M., "Logic and databases : A deductible approch". Computing Surveys, 1984.

12 Garey R.M. and Johnson, "Computers and Intractability, A guide to the Theory of NP-Completeness". W.H. Freeman and CO., San Francisco, 1979.

13 Hayes E.D.A. Waterman and LENAT D.B., "Building expert system", Addisson-Westely, 1983.

14 Kamath A.P., Karmakar N.K. Ramakrishnan K.G. and Resende M.G.C., "A continous approach to inductive inference", MATHEMATICAL PROGRAMMING, 57, 1992.

15 Klein M. and Methlie L. B., "Expert Systems: A Decision Support Approach with applications in management and finance", Addison-Wesley Publishing Compagny, 1990.

16 Koudil M., Benatchba K. and Dours D., "Using genetic algorithms for solving partitioning problem in codesign", *Lecture Notes in Computer Science, Springer-Verlag, Vol. 2687,* pp. 393–400, June 2003.

17 Koudil M., Benatchba K., Gharout S. and Hamani N., "Solving partitioning problem in codesign with ant colonies", *International Work-conference on the Interplay between Natural and Artificial Computation, Lecture Notes in Computer Science, Springer-Verlag, Vol. 3562,* June 2005.

18 Landry M., "Les rapports entre la complexité et la dimension cognitive de la formulation des problèmes", Association française des sciences et technologies de l'information et des systèmes, p. 3-32, 1987.

19 Martel J.M., Rousseau A., "Cadre de référence d'une démarche multicritère de gestion intégrée des ressources en milieu forestier", Projet de développement de la gestion intégrée des ressources, document technique 93011, Université Laval, Québec, 1993.

20 Nguyen, Perkins, Laffey and Pecora, "Checking an expert systems Knowledge base for consistency and completeness, *IJCAI 85*, Arvind Joshi (ed.), Los Altos California, pp.375-378, 1985.

21 Pictet J., « Dépasser l'évaluation environnementale, Procédure d'étude et insertion dans la décision globale", Collection Meta, Presses Polytechnique et Universitaires Romandes, 1015, Lausanne, Suisse, 1996.

22 Roy B., "The optimisation problem formulation : criticism and overstepping", The Journal of the Operational Research Society, Vol. 32, N06, pp. 427-436, 1994.

23 Vodoz L., "Enjeux et limites du recours à la négociation", La négociation son rôle, sa place dans l'aménagement du territoire et la protection de l'environnement", Edité par Ruegg J., Mettan N., Vodoz L., Presses Polytechniques et Universitaires Romandes, CH-1015 Lausanne, 1994.

Received: Jun. 2006. Accepted: Jun. 2007.