MANAGING COMBINATORIAL OPTIMIZATION PROBLEMS BY MEANS OF EVOLUTIONARY COMPUTATION AND MULTI-AGENT SYSTEM

Mauricio Paletta
Centro Inv. Inf. y Tec. Comp. (CITEC), Universidad de Guayana (UNEG), Av. Atlántico, Ciudad Guayana 8050, Venezuela

Pilar Herrero
Facultad de Informática, Universidad Politécnica de Madrid (UPM), Campus de Montegancedo S/N; 28660, Madrid, Spain

Keywords: Evolutionary Program, Multi-agent System, Inter-agent Communication Protocol, JADE.

Abstract: The necessity for solving a combinatorial optimization problem is very common. Evolutionary/genetic program could be used to deal with such situations. Unfortunately, depending on the complexity of the problem, high computational capabilities are required, primarily in those cases in which measuring the quality of a potential solution is very demanding. However, advances in Distributed Artificial Intelligence (DAI), Multi-Agent Systems (MAS) to be more specific, could help users to deal with this situation by parallelizing the evolutionary program aiming to distribute the computational capabilities required. This paper presents an inter-agent MAS protocol for parallelizing an evolutionary program aiming to reduce the communications requirements necessary as well as allowing a response within a reasonable period of time.

1 INTRODUCTION

Evolutionary/Genetic Programs (EPs) (Eiben et al, 2003), (Jain et al, 2007) are powerful searching techniques used to solve Combinatorial Optimization Problems (COPs) in many disciplines. Depending of the complexity of the problem, EPs could require high computational capabilities such as CPU time, memory, etc. This problem increases considerably if the calculation of the fitness function for evaluating any potential solution also requires high computational capabilities.

Parallelizing EPs has been proposed to overcome the above problem (Andre et al, 1996), (Arenas et al, 2002), (Lee, 2007). The basic idea is to divide a big population into multiple smaller sub-populations that are distributed to separate processors and can then evaluated simultaneously. Combination of EP and MAS technologies (Ferber, 1995) is essential for accomplishing this objective**.**

In order to reduce the communication necessities required for parallelizing EPs, this paper focuses on defining a multi-agent architecture, establishing an interaction protocol for obtaining a suitable solution, based on EP specifications for any specific COP.

The rest of the paper is organized as follows: section 2 presents some work related to the scope of this paper; section 3 describes the proposed inter-agents protocol; section 4 shows the evaluation of this proposal. Finally, section 5 reaches some conclusions and future work.

2 RELATED WORK

Some examples of making evolutionary algorithms in distributed environments can be consulted on (Arenas et al, 2002), (Meng et al, 2007). DREAM (Arenas et al, 2002) and G2DGA (Berntsson, 2005) are examples of frameworks concerning development of Peer-to-Peer distributed computing systems. Using JADE (Bellifemine et al, 1999), (Chmiel, 2005) as a middleware to propose a multi-agent synchronous evolutionary system can be reviewed on (Eiben et al, 2007), (Lee, 2007) and (Meng et al, 2007). The newscast protocol (Jelasity et al, 2002), which is a lazy fully distributed information propagation protocol, is an example of a work with the same goal but different scope.

Paletta M. and Herrero P. (2010).
MANAGING COMBINATORIAL OPTIMIZATION PROBLEMS BY MEANS OF EVOLUTIONARY COMPUTATION AND MULTI-AGENT SYSTEM.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Agents*, pages 253-256

3 OVERALL INTER-AGENT PROTOCOL

The proposed inter-agents communication protocol occurs in a general that basically consists of two different types of agents:

1) EPEnvAgent which: 1) informs the rest of the agent about the EP parameters and problem specifications; 2) takes control of the population growth; and, 3) allows the evolution among multiple system nodes.

2) EPAgent which represents a potential solution to the problem as well as carry out the main steps of evolutionary computation: selection and variation (crossover and mutation) - to generate descendants (new potential solutions / agents).

Other elements involved in the architecture are the EP parameters (population size, probability of applying genetic operators, among others), and the problem specifications which is necessary to define a specific COP.

Each computational node may have multiple EPAgent (a population) but only one EPEnvAgent is needed. The number of EPAgent in the system (population size) may vary, as the EP evolves. An EPAgent can be created as well as eliminated from the system. The creation of new EPAgent is the result of applying genetic operators, while the disposal is due to self-destruction or suicide. Some parameters are used to control these functions.

On the other hand, and following the FIPA ACL specifications (FIPA, 2002), agents communicate with each other according to the messages shown in Figure 1. Messages are the following:

- INFORM, from EPAgent to EPEnvAgent. For EPEnvAgent to keep some statistics of the evolutionary process and keep track of the best solution it has at any given time, any EPAgent uses this message to report or inform its measure of quality (fitness), as well as the structure that represents the solution to the COP (chromosome).
- PROPAGATE, from EPEnvAgent to EPEnvAgent. When an EPEnvAgent realizes that has obtained a better individual (solution) in the population of EPAgents, it uses this message to transmit /propagate this information to the rest of EPEnvAgent.
- REQUEST, from EPAgent to EPEnvAgent. In this proposal, EP parameters may change dynamically. An EPAgent should, from time to time, requests by using this message the EPEnvAgent to send back the parameters.
- INFORM, from EPEnvAgent to EPAgent. In response to the previous message, the EPAgent is receiving the EP parameters from the EPEnvAgent. It is important to mention that these parameters can change the way in which EPAgents may vary to produce descendants (new EPAgents) or self-destruction.
- INFORM_REF, from EPEnvAgent to EPEnvAgent. By using this message, the main EPEnvAgent informs all the auxiliary EPEnvAgent changes in the EP parameters.
- PROPOSE, from EPAgent to EPEnvAgent. In this proposal each EPAgent is responsible for searching (selecting) a suitable partner to cross. Through this message, the EPAgent makes a request to find a suitable partner by sending its fitness and Id as parameters of the message. These parameters are needed when some other EPAgent accepts the request and responds appropriately to the sender.
- PROPOSE, from EPEnvAgent to EPAgent. By this message, the EPEnvAgent replicates the proposal sent by an EPAgent for the rest of EPAgents. They become aware of it, so that, they can respond directly to EPAgent who made the original request.
- ACCEPT_PROPOSAL, from EPAgent to EPAgent. An EPAgent uses this message to respond positively to a request from another EPAgent for crossing with it. This happens only if it decides to accept the proposal (based on the fitness received from the sender agent and by using some probabilities). The EPAgent sends its chromosome so that the sender can use it for applying the corresponding genetic operation.
- REQUEST_WHENEVER, from EPAgent to EPEnvAgent. EPAgents are responsible for the selection mechanism and for applying genetic operators to generate new EPAgents. These new agents must be created by the EPEnvAgent. Through this message, the EPEnvAgent knows that a new EPAgent must be created with the chromosome given as a parameter of the message.
- CONFIRM, from EPEnvAgent to EPAgent. EPEnvAgent has the control to start and stop the evolution. When the process has to be finished, the EPEnvAgent uses this message to give to all EPAgents the order to self-destruct.
- CONFIRM, from EPAgent to EPEnvAgent. Once an EP-Agent is self-destructed, either because it received an order from the EPEnvAgent or for effect of this evolutionary algorithm (by using some parameters and based on the current fitness), its uses this message to confirm his suicide.
- CONFIRM, from EPEnvAgent to EPEnvAgent. If the receiver is the main EPEnvAgent, then some auxiliary EPEnvAgent has

decided to stop its participation in the evolution process. On the other hand, if the receiver is an auxiliary EPEnvAgent, then the entire evolution process must be stopped.

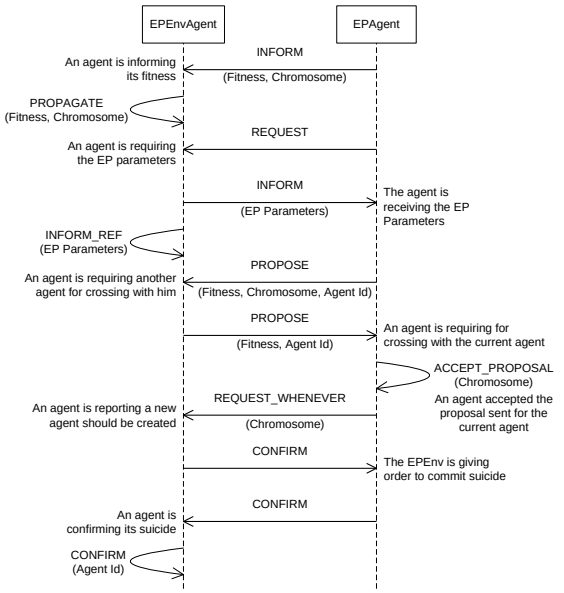

Figure 1: Inter-agents communication protocol.

A method for estimating the total number of messages *TM* requires to complete an evolution, which has duration of *T* time units, is shown in (1).

$$
\begin{aligned}
&TM = \sum_{i=1}^{12} Msg_i; \quad NA = \sum_{i=1}^{n} Na_i \\
&Msg_1 = p_1 * p_2 * NA * T; \quad Msg_2 = n * Msg_1; \\
&Msg_3 = Msg_4 = n * T / t_1; \quad Msg_5 = n * Msg_3; \\
&Msg_6 = Msg_7 = p_3 * NA * T; \\
&Msg_8 = Msg_9 = p_2 * Msg_6; \\
&Msg_{10} = Msg_{11} = p_1 * NA * T; \quad Msg_{12} = n * Msg_{10}
\end{aligned}
\tag{1}
$$

Where:

p_1: probability that an agent commits self-suicide.
p_2: probability that an agent accepts to cross with another ag.
p_3: probability to produce at least one descendant.
t_1: time in which parameters are sought.
NA: the total number of agents.
Na_i: the total number of agents for node *i*.
n: the total number of nodes.
Msg_i: the total number of messages of type *i*.

Next section provides some results from experiments conducted with the proposal.

4 EVALUATION

In order to implement the inter-agents communication protocol proposed in this paper as well as the MAS-based evolutionary algorithm, a JADE-based framework called EP-MAS.Lib (Paletta et al, 2009) was developed. For evaluating the proposal we have implemented a solution for two different problems, varying the computational capabilities demanded to calculate the fitness of an individual or potential solution for the problem.

The first issue to be considered is related to the Traveling Salesman Problem (TSP) which is a well-known NP-hard COP (Lawler et al, 1985). In this case, computational capabilities demanded are very low. The experimentation was conducted using the data relative to the 51-cities Christofides and Eilon (Christofides et al, 1972).

The second problem has to do with finding the proper setting of parameters required to configure an Artificial Neural Network (ANN) for a particular investigation that is currently being developed (Paletta et al, 2008). Fitness in this problem consists in training an ANN and obtains the corresponding total average error aiming to reduce it. Unlike the previous case, the calculation of this fitness is very demanding on computational capabilities.

Both problems were implemented using a conventional simple genetic program, as well as using the distributed model presented in this paper, aiming to compare the efficiency of each case to deal with the same problem.

Based on the results obtained we observed:

1) A solution for TSP problem was obtained more efficiently using the simple genetic program than the MAS-based distributed proposal. The difference is about 10 to 1 in execution time.

2) Related to the ANN configuration, the simple genetic program needed 42.3 hours for obtaining a satisfactory solution. However, by using the MAS-based distributed proposal (with 3 nodes), a solution was obtained in 6 hours approximately.

The results show that the proposed protocol allows distributed computational capabilities among more than one node in such a way to expedite the process of convergence of the searching algorithm.

Table 1 shows the results obtained by using the expression (1) for estimating the total number of messages *TM* required for these experiments. As expected the complexity of the problem is directly proportional to the communication needs required by the EP. Parameters are: p_1=0.55; p_2=0.8; p_3=0.5; t_1=5 min; *NA*=300; Na_i=100, $\forall i$ ($1 \le i \le 3$); *n*=3.

Table 1: Detail of the estimated total number of messages.

Messages	TSP (T = 45 min)	ANN (T = 360 min)
Msg_1	5,940	47,520
Msg_2	17,820	142,560
$Msg_3 = Msg_4$	27	216
Msg_5	81	648
$Msg_6 = Msg_7$	6,750	54,000
$Msg_8 = Msg_9$	5,400	43,200
$Msg_{10} = Msg_{11}$	7,425	59,400
Msg_{12}	22,275	178,200
TM	**85,320**	**682,560**

5 CONCLUSIONS AND FUTURE WORK

In this paper we present a communication protocol between agents to be used in a MAS scenario aiming to resolve a specific COP. As we know, this proposal differs from other similar works in the following aspects:

- It focuses on the communication process between the agents in the systems and therefore to the cooperation needed for solving the evolutionary algorithm, instead of the properly used elements of the evolutionary algorithm (selection mechanism and genetic operations).
- The evolutionary algorithm (selection and variation) is not controlled by a central entity; instead it's controlled by all individuals (agents) of the population actively involved in this process.
- The needed information (EP parameters and problem specifications) is not located in a central repository, but it is replicated for all who need it.
- The definition of a particular COP.

The obtained result point out that agent in a MAS-based environment can interact with each other to solve any COP by using the communication protocol proposed.

We are working on reducing the flow of messages and data that is required to avoid possible bottleneck.

REFERENCES

Andre, D., Koza, J., 1996. A parallel implementation of genetic programming that achieves super-linear performance. In *Proc. Intern. Conf. on Parallel and Dist. Processing Techniques and App.*, pp. 1163-1174.

Arenas, M.I., Collet, P., Eiben, A.E., Jelasity, M., Merelo, J.J., Paechter, B., Preuss, M., Schoenauer M., 2002. A Framework for Distributed Evolutionary Algorithms. In *Proc. 7th Int. Conf. on Parallel Problem Solving from Nature (PPSN VII)*. LNCS 2439, pp. 665-675.

Bellifemine, F., Poggi, A. Rimassa, G., 1999. JADE – A FIPA-compliant agent framework. *Telecom Italia internal technical report*. In *Proceedings Int. Conf. on Practical Applications of Agents and Multi-Agent Systems (PAAM'99)*, pp. 97-108.

Berntsson, J., 2005. G2DGA: an adaptive framework for internet-based distributed genetic algorithms. In *Proc. of the 2005 workshops on Genetic and Evolutionary Computation (GECCO)*, pp. 346-349.

Chmiel, K., Gawinecki, M., Kaczmarek, P., Szymczak, M., Paprzycki, M., 2005. Testing the Efficiency of JADE Agent Platform. In *Proc. 3rd Int. Symposium on Parallel and Distributed Computing (ISPDC)*, IEEE Computer Society Press, 13(2) pp. 49-57.

Christofides, N., Eilon, S., 1972. Algorithms for Large-Scale Travelling Salesman Problems. *Operations Research Quarterly*, 23(4), pp. 511-518.

Eiben, A.E., Smith, J.E., 2003. *Introduction to Evolutionary Computing*. Springer Verlag.

Eiben, A.E., Schoenauer, M., Jiménez, J.L., Castillo, P.A., Mora, A.M., Merelo, J.J., 2007. Exploring Selection Mechanisms for an Agent-Based Distributed Evolutionary Algorithm. In *Proceedings Genetic and Evolutionary Comp. Conf. (GECCO)*, pp. 2801-2808.

Ferber, J., 1995. Les systems multi-agents, *Vers une intelligence collective*. Ed. InterEditions, pp. 1-66.

FIPA, 2002. Foundation for Intelligent Physical Agents. FIPA ACL Message Structure Specification, SC00061, Geneva, Switzerland.

Jain, L.C., Palade, V., Srinivasan, D., (Eds.) 2007. Advances in Evolutionary Computing for System Design. *Studies in Computational Intelligence*, vol. 66. Springer Verlag. ISBN: 978-3-540-72376-9.

Jelasity, M., van Steen, M., 2002. Large-scale newscast computing on the Internet. *Technical Report IR-503*, Vrije Universiteit Amsterdam, Department of Computer Science, October.

Lawler, E.L., Lenstra, J.K., Rinnooy, A.H., Shmoys, D.B. (Eds.), 1985. *The Travelling Salesman Problem: A guided tour of combinatorial optimization*. New York: Wiley and Sons.

Lee, W., 2007. Parallelizing evolutionary computation: A mobile agent-based approach. *Expert Systems with Applications*, 32(2), pp. 318-328.

Meng, A., Ye, L., Roy, D., Padilla, P., 2007. Genetic algorithm based multi-agent system applied to test generation. *Computers & Education 49*, pp. 1205-1223.

Paletta, M., Herrero, P., 2008. Learning Cooperation in Collaborative Grid Environments to Improve Cover Load Balancing Delivery. In *Proc. IEEE/WIC/ACM Joint Conf. on Web Intelligence and Intelligent Agent Tech.* IEEE Computer Society, pp. 399-402.

Paletta, M., Herrero, P., 2009. EP-MAS.Lib: A MAS-Based Evolutionary Program Approach. In *Proc. 4th Int. Conf. on Hybrid Artificial Intelligence Systems (HAIS 2009)*, LNAI 5572, Springer-Verlag, pp. 9-17.