Mobile Agents Implementing Local Computations in Graphs

Bilel Derbel\textsuperscript{1,⋆}, Mohamed Mosbah\textsuperscript{2,⋆⋆}, and Stefan Gruner\textsuperscript{3,⋆⋆⋆}

\textsuperscript{1} Laboratoire d'Informatique Fondamentale de Lille (LIFL), Université des Sciences et Technologies de Lille 1, France
bilel.derbel@lifl.fr
\textsuperscript{2} Laboratoire Bordelais de Recherche en Informatique (LaBRI), Université Bordeaux 1/ENSEIRB, France
mosbah@labri.fr
\textsuperscript{3} Department of Computer Science, University of Pretoria, South Africa
sgruner@cs.up.ac.za

Abstract. Mobile agents are a well-known paradigm for the design and implementation of distributed systems. However, whilst their popularity continues to grow, a uniform theory of mobile agent systems is not yet sufficiently elaborated, in comparison with classical models of distributed computation. In this paper we show how to use mobile agents as an alternative model for implementing distributed local computation rules. In doing so, we approach a general and unified framework for local computations which is consistent with the classical theory of distributed computations based on graph relabeling systems.

Keywords: Distributed algorithms, Mobile agents, Relabeling systems.

1 Introduction

Models of local computations, described by graph relabeling systems provide a useful theoretical framework to specify and reason about various aspects of distributed computation with distributed algorithms \cite{9,10,2}. Assuming that the reader is already familiar with this theoretical background, we will only briefly recapitulate the basic characteristics and features of modeling distributed systems by local computations and graph relabeling systems. This well-known paradigm will be our starting point from where we shall proceed towards a more recent paradigm of distributed computation by mobile agents.

Our aim is to demonstrate that all basic building blocks of the graph relabeling paradigm can be implemented by the activities of mobile agents, leading to

\begin{itemize}
  \item \textsuperscript{⋆} Supported by the DOLPHIN INRIA project team.
  \item \textsuperscript{⋆⋆} This work has been supported by the french project “Agence Nationale de la Recherche ANR-06-SETI-015”.
  \item \textsuperscript{⋆⋆⋆} Supported by the National Research Foundation (NRF) of the Republic of South Africa.
\end{itemize}
the hypothesis that mobile agents are as powerful as classical distributed systems, i.e., message passing systems [4]. In practice, the use of mobile agents for the implementation of distributed algorithms can have advantages over classical implementations, because roaming agents can better cope with temporary network failures and also consume less computational resources, in comparison with the global network activities induced by classical implementations of distributed algorithms. In addition, mobile agents allow to bring a new level of abstraction in distributed computing. For instance, in the message passing model, the nodes represent both the topology of the network and the autonomous computation entities. In opposite, in the mobile agent model, the nodes define only the topology of the network, while the agents define the computation entities of the network.

The consideration (description, reconstruction) of agent systems in terms of graph transformation systems is not a new idea; take for example [8] as an early contribution to this field of study. In [8], however, graph transformation techniques are used to model internal properties and/or actions of agents, whereas the focus of our paper is on their external properties, mainly motion between network places, motivated by our intention to demonstrate the possibility of expressing (respectively implementing) classical distributed algorithms in terms of mobile agent systems. To this end, graph transformation systems can be regarded as the bridge formalism between the domain of classical distributed algorithms and the domain of mobile agent systems.

**Graph Relabeling Systems:** Processor networks, which are the substrate of distributed computation, are represented by labeled graphs $G = (V, E, L, \lambda)$ with a set of labels $L$ and a (possibly partial) labeling function $\lambda : (V \cup E) \rightarrow L$ that attaches labels to vertices (nodes) and/or edges (arcs) of the network graph. The labels, which may lexically appear arbitrarily complex, are used to model the internal states of the network components during the run of a distributed algorithm on the network. A final label configuration represents the result of a terminated algorithm. Thereby, the models must be designed in such a way that three locality conditions are always fulfilled:

- **c1:** Relabeling does not modify the underlying graph structure (from a topological point of view);
- **c2:** Each step can only relabel a limited, connected sub graph (fixed in size);
- **c3:** The applicability of a relabeling step in a “neighborhood” is constrained only by the local conditions within such a neighborhood, not by the global state of the entire network.

Distributed algorithms described in such a framework are usually composed of basic units which correspond to certain types of relabeling rules. These various rule types, which are classified and explained in [5], comprise constructs such as: single node relabeling depending on only one neighbor, two neighbor relabeling, single node relabeling depending on labels of all neighbors (star relabeling), single node relabeling in the center of a ball of radius 2, relabeling of an entire ball of radius $k > 1$. 