ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Information Processing Letters 87 (2003) 251–255

Information Processing Letters

www.elsevier.com/locate/ipl

Linear time self-stabilizing colorings

Stephen T. Hedetniemi, David P. Jacobs, Pradip K. Srimani [*]

Department of Computer Science, Clemson University, Clemson, SC 29634-0974, USA

Received 7 August 2002; received in revised form 21 March 2003

Communicated by A.A. Bertossi

Abstract

We propose two new self-stabilizing distributed algorithms for proper $\Delta+1$ (Δ is the maximum degree of a node in the graph) colorings of arbitrary system graphs. Both algorithms are capable of working with multiple type of daemons (schedulers) as is the most recent algorithm by Gradinariu and Tixeuil [OPODIS'2000, 2000, pp. 55–70]. The first algorithm converges in $O(m)$ moves while the second converges in at most n moves (n is the number of nodes and m is the number of edges in the graph) as opposed to the $O(\Delta \times n)$ moves required by the algorithm by Gradinariu and Tixeuil [OPODIS'2000, 2000, pp. 55–70]. The second improvement is that neither of the proposed algorithms requires each node to have knowledge of Δ, as is required by Gradinariu and Tixeuil [OPODIS'2000, 2000, pp. 55–70]. Further, the coloring produced by our first algorithm provides an interesting type of coloring, called a Grundy Coloring [Jensen and Toft, Graph Coloring Problems, 1995].

Keywords: Algorithms; Distributed systems; Self-stabilizing coloring; Grundy coloring

1. Introduction

In a distributed system the computing elements or nodes exchange information only by message passing. Every node has a set of local variables whose contents specify the local state of the node. The state of the entire system, called its *global state*, is the union of the local states of all the nodes. Each node is allowed to have only a partial view of the global state, and this depends on the connectivity of the system and the propagation delay of different messages. Yet, the objective in a distributed system is to arrive at a desirable global final state, or legitimate state. One of the goals of a distributed system is to function correctly, i.e., the global state of the system should remain legitimate in presence of faults (transient). Often, malfunctions or perturbations bring the system to some illegitimate state, and it is desirable that the system be automatically brought back to a desired legitimate state. *Self-stabilization*, introduced by Dijkstra [4], is the most inclusive approach to fault tolerance in distributed systems that brings the system back to a legitimate state starting from any illegitimate state (caused by any transient fault) without any intervention by an external agent. In a self-stabilizing algorithm, each node maintains its local variables, and can make decisions based on the knowledge of its neighbors' states.

In a self-stabilizing algorithm, a node may change its local state by making a *move* (specification of

---

[*] Corresponding author.

E-mail addresses: hedet@cs.clemson.edu (S.T. Hedetniemi), dpj@cs.clemson.edu (D.P. Jacobs), srimani@cs.clemson.edu (P.K. Srimani).

0020-0190/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0020-0190(03)00299-0