International Journal of Innovative
Computing, Information and Control
Volume 4, Number 2, February 2008

 ISSN 1349-4198
pp. 335–350

# USING DOMINANCE SEARCHING TO SPEED UP TEMPORAL JOIN OPERATION

Chih-Ying Chen[1], Jui-Feng Hu[2] and Chin-Chen Chang[3]

[1]Department of Communications Engineering

[2]Department of Electronics Engineering

[3]Department of Information Engineering and Computer Science
Feng Chia University
No.100, Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
chihchen@fcu.edu.tw; ccc@cs.ccu.edu.tw

Received November 2006; revised June 2007

Abstract. *This paper is concerned with the problem of efficient processing of temporal join operation on temporal relation. By mapping time intervals to points in the plane, we first show that the problem of determining the set of all matching tuples of a temporal join is equivalent to the dominance searching problem in the plane. Then, by using an efficient data structure for solving the equivalent dominance searching problem as an index for the inner relation, we propose a nested-loops temporal join algorithm. For the case where the index for the inner relations too large to fit in the primary memory, we propose another partition-based temporal join algorithm which does not need any index for the operand relations. Finally, in order to provide more efficient processing of temporal join, we propose a cluster scheme and an index scheme to support efficient storage of tuples and direct access of matching tuples.*

**Keywords:** Temporal relation, Temporal join operation, Time interval intersection searching problem, Dominance searching problem

1. **Introduction.** Since temporal information (or time-varying information) is included in many database applications, efforts have been devoted to make the processing of temporal data efficient [Snodgrass 1987, Elmasri and Wuu 1990, Leung and Muntz 1990, Gunadhi and Segev 1991, Leung and Muntz 1992, Rana and Fotouchi 1993, Chen et al. 1994, Lu et al. 1994]. Examples include temporal data modeling [Snodgrass 1987, Elmasri and Wuu 1990] and query optimization [Leung and Muntz 1990, Gunadhi and Segev 1991, Leung and Muntz 1992, Rana and Fotouchi 1993, Chen et al. 1994, Lu et al. 1994, Chang 2007, Chang 2007].

When referring to a temporal relation, it means a set of temporal data in the relational database model. It should be pointed out that there are various ways to represent temporal data in the relational model; detailed discussions can be found in [Segev and Shoshani 1988]. The representation model we adopt in this paper is a time interval representation, which is also used in many studies concerning query optimization of temporal databases [Leung and Muntz 1990, Gunadhi and Segev 1991, Rana and Fotouchi 1993, Lu et al. 1994, Chen et al. 1994]. In this representation model, the time dimension is considered as a sequence of discrete, consecutive, equally-distanced time constants. A time interval is defined as a set of consecutive time instants $t_S$, $t_S$+1, $t_S$+2,....., $t_E$, and is denoted as $< t_S, t_E >$, where $t_S$ is the starting time and $t_E$ is the ending time. A temporal relation is a set of temporal tuples. And each temporal tuple consists of a surrogate of the tuple, some non-time varying attributes, at least one time-varying attribute (or temporal attribute),