

第一章 緒論

第一節 前言

過去二十年來，電腦工程師們努力設計出高性能平行處理的電腦。在平行計算有兩個體系，一是單機體系，即一個程式在單機的多個處理器上運行，簡稱 SMP (Symmetric Multi-Processor 對稱多處理器)，二是分散式體系，即一個程式在多個計算機連成的網路上運行，稱為 Cluster (叢集式電腦)。兩者最大的區別是，單機體系的多個處理器可以共享記憶體，並且共享同一地址空間的儲存單元，而分散式體系則必須透過網路來交換信息。目前在處理電腦棋類的平行演算法中，絕大部分是架構在 SMP (Symmetric Multi-Processor) 底下。而在棋類中眾多的演算法裡，我們企圖使其平行計算，發現想得到很高的效能，實在有著相當大需要克服的困難。

在目前眾多發表的論文中，以 PVS (Principal Variation Splitting) 搜尋演算法【10】、EPVS (Enhanced Principal Variation Splitting) 搜尋演算法【6】、及 DTS (Dynamic Tree Splitting) 搜尋演算法【7】所提升的效能表現比較好，同樣使用 8 個 CPU 的電腦，其中 PVS (Principal Variation Splitting) 搜尋演算法能達到提升約 4.1 倍的效能，而 EPVS (Enhanced Principal Variation Splitting) 搜尋演算法能達到提升約 5.4 倍的效能。目前表現最好的是 DTS (Dynamic Tree Splitting) 搜尋演算法，它所能提升的效能是約 6.6 倍，已經很接近最佳理想值 8 倍效能的情形。

DTS (Dynamic Tree Splitting) 搜尋演算法的作者 Robert M. Hyatt，將此演算法應用設計在電腦西洋棋程式，他所設計的電腦西洋棋程式取名為 Crafty，Crafty 在 2004 年第 12 屆 World Computer Speed Chess Championship 比賽獲得第二名，並且也在 2005 年第 13 屆 World Computer Speed Chess Championship 比賽獲得第五名，表現成績相當卓越，並且無私的開放 Crafty 的程式碼供人使用，我們便使用 Crafty 做改良方法的研究並測試實驗。

有些研究西洋棋的學者對於 Hyatt 的 Crafty 有質疑，認為 Crafty 的程式應該是使用所謂的 Young Brothers Wait Concept (YBWC) 的方法而不是 DTS 搜尋演算法，但是為了與 Hyatt 的論文以及其它相關論文一致，本論文仍將 Crafty 的搜尋演算法稱為 DTS 搜尋演算法。

第二節 研究動機及目的

電腦西洋棋程式所研究的各種搜尋演算法往往成為我們電腦象棋程式開發者的參考對象。而電腦象棋的發展最早是在台灣開始，從 1981 年張躍騰發表了第一篇電腦象棋的論文“人造智慧在電腦象棋上的應用”【14】，到了 1986 年，黃東輝在許舜欽教授指導下發表了“中國象棋知識庫之設計與製作”【13】，對知識庫做了完整介紹，帶動了電腦象棋的研究風氣。

在國立台灣師範大學資訊工程所林順喜教授的研究群中，我們致力於研究開發電腦象棋程式：深象。深象是以象棋程式“千慮”(Contemplation)為基礎，

繼續加以研究與改良。象棋程式“千慮”是 2002 年就讀於台灣大學資訊工程研究所的吳光哲學長所開發，結合了各種電腦西洋棋先進的搜尋技術，並加上陳志昌學長的開局庫【17】，取得第九屆電腦奧林匹亞象棋比賽金牌的佳績。而深象也在 2006 年，由師大資工所黃文樟學長做了大幅度的改良【15】，以及李任軒學長加強深象的開局庫【12】，使得深象在 2006 年取得了第十一屆電腦奧林匹亞象棋比賽銅牌的佳績。我們判定深象已經擁有人類棋力七段以上的水準(大師級)，在經過平行化運算後，擬使用多核的機器來取代之前只使用單 CPU 的機器，目標是將深象的棋力提升到八段以上，使深象能夠有足夠的實力，將來有機會可以向人類的象棋棋王（九段）做挑戰。

第三節 論文架構

本論文主要根據 DTS (Dynamic Tree Splitting) 搜尋演算法，透過我們的改良方法後能達到更高的效能。第一章為緒論，簡單介紹現今平行演算法的概況，讓讀者能夠輕鬆進入本論文的探討。第二章將一些著名的平行搜尋演算法做詳細的介紹。第三章為本論文的重點，提出一些方法使其提升 DTS (Dynamic Tree Splitting) 搜尋演算法的效能。第四章為使用 Crafty20.14 版做出的實驗結果，驗證出我們提出的方法確實能提升更高的效能。第五章則對此研究做出結論，並討論未來可能的研究方向。