

電腦象棋分散式搜尋技術之研究

A Study on Distributed Search Techniques for Computer Chinese Chess

計畫編號: NSC90-2213-E-002-090

執行時間: 90年8月1日至91年7月31日

計畫主持人: 許舜欽 台大資訊系教授

摘要

電腦象棋的棋力主要受審局優劣及對局樹搜尋深度的影響。審局函數設計的良窳，受到程式設計者本身棋力高低的限制，而對局樹搜尋的深度則受到目前硬體計算速度的牽制，因此，目前的電腦象棋程式常因而錯失先機。

本報告討論電腦象棋最常使用的 α - β 切捨演算法。分析此演算法應用於分散式計算環境，所面臨的問題及解決的方法，並以此架構實作一電腦象棋程式，以加深對局樹搜尋深度的方式，讓電腦象棋具有更高的棋力。

一、系統發展環境

由於微處理器技術的進步，一般工作站的執行效率已可直逼以往的超級電腦，加上高速網路技術的突飛猛進，網路資料傳輸速率大幅提昇，再加上分散式作業環境漸趨於成熟，以及新的應用程式對分散式作業環境的需求，使得如何有效應用現有的電腦進行分散式運算，以分析求解更大的問題，成為目前十分熱門的研究課題。

1989年，Vaidy Sunderam 提出 PVM (Parallel Virtual Machine) 的概念。隔年，與 Al Geist 完成 PVM 的原型系統—PVM 1.0。1991年，Robert J. Manchek 完成了第一個公開

的版本—PVM 2.0，並於1993年，完成 PVM 3.0版。目前最新的版本為 PVM 3.4.4。它增加了許多新的特性，包括：Communication Context、Message Handlers、Persistent Messages、User Defined Tracing 及 Windows 95/NT Ports。

二、PVM 的設計原則

(1) 使用者可以在一群電腦中指定行程要在哪幾部機器上執行。此電腦群組可包括單 CPU 的機器，也可以有多 CPU 的機器，使用者可動態增加或減少此群組中的機器。

(2) 與硬體無關的存取方式。使用者可以將所處的硬體環境視為一個虛擬的行程單元，亦可選擇最合適的機器進行運算。

(3) PVM 的運算單元是 task，它是一個獨立的執行緒，交替進行運算及通訊的工作。PVM 並不強制行程與處理器之間的對應關係，因此，許多個 task 可以在同一個處理器上執行。

(4) 一個進行運算的行程以傳遞及接收訊息來進行通訊。傳送訊息的大小只受限於可用的記憶體。

(5) 支援異質性 (heterogeneity) 的計算環境。PVM 可支援不同的機器、網路及應用程式。

式。PVM 的訊息傳遞方式可允許兩個擁有不同資料表示方式的機器，在一個訊息內傳遞多個不同的資料型式。

(6) 支援多處理器的機器。在多處理器的機器上，PVM 可利用機器原生的傳遞方式。

三、CCL (Chinese Chess Library)

我們所發展的 CCL 是一組用於撰寫象棋程式的函式庫，適用於 C/C++ 的程式設計之用，它提供了基本的資料結構及運作的副程式。此外，為了應付各種不同的狀況，以及日後對審局函數或搜尋演算法改進的空間，我們在資料結構和副程式的設計上，做了許多擴充性的設計。

藉著 CCL 的使用，可以避免旁枝末節的干擾，影響了對主程式的分析思考。除此之外，CCL 還提供了簡單的使用者介面，以利程式設計時的除錯。

為了搜尋最佳著手，CCL 提供了副程式 SearchBestMove()。此副程式是以 α - β 切捨演算法進行對局樹的搜尋工作。在搜尋最佳著手時，可能因為某些因素，而必須中斷搜尋的工作，例如，所允許的搜尋時間已經用完，或利用知識庫已找到最佳著手等。為了達到這個目的，SearchBestMove() 必須能夠由主程式中止，而不需待其執行結束才停止。

CCL 所採行的解決方式有二。在 C 程式部分，CCL 利用回呼函數 (callback function) 的技術，使用者必須撰寫一回呼函數，並將此函數的函數指標傳給 SearchBestMove()；在 C++ 程式部分，則利用虛擬函數 (virtual function) 來處理指標。

四、系統實作

4.1 α - β 切捨演算法

象棋對奕過程可以用一個對局樹表示。對局樹的節點分為三種：偶數層的內部節點、奇數層的內部節點、葉節點。

假設雙方都選擇對己方最有利的著手，那麼由葉節點往上倒推至根節點時，即可找出我方應採行的最佳著手。然而，由於雙方的可行著手過多，為面臨組合爆炸 (combinational explosion) 的問題。

為了解決此問題，我們採用淺層切捨 (shallow pruning) 和深層切捨 (deep pruning) 兩種限制分支 (branch-and-bound) 的技巧。

4.2 平行化 α - β 切捨演算法

所謂平行化就是同時搜尋對局樹不同的部分，以多台機器同時計算，增加 α - β 切捨演算法的效率。

我們將所有機器組成一階層式的架構，由其中一台機器為 Master，其餘則為 Slave。Master 負責將搜尋工作分配給 Slave，當 Slave 完成搜尋工作之後，就將結果送回 Master，Master 就可根據這些資料找出最佳著手。

以平行的方式實作 α - β 切捨演算法，會遇到三個關鍵性的問題：

(1) 同步化 (synchronize)。 α - β 切捨演算法在搜尋對局樹時，是某部分的節點分析完後，再分析對局樹的其它部分。若在多台機器上執行相同的程式，則這些機器必須同步化。

(2) 分析某些節點時，應該判斷平行化是否為最佳策略。某些節點以循序方式搜尋，會比平行化的方式更有效率，或甚至只要搜尋主變化搜尋的路徑，其餘子節點根本不必理會。

(3) 為了避免對同樣的盤面做重複的搜尋，可利用雜湊表儲存已搜尋過的盤面。但若有多台機器同時搜尋，雜湊表的儲存位置、如何存取雜湊表都必須謹慎評估，以免儲存雜湊表的機器成為搜尋的瓶頸。

4.3 雜湊表之設計

雜湊表的存放方式，可分為三種：

(1) 雜湊表固定存放在某一台機器上，其它的機器有需要時，再向此機器發出讀寫的請求。

(2) 每台機器都存放雜湊表，以供自己在搜尋時使用。可採取存放完整雜湊表，以保證絕對不會有重複搜尋盤面的情形；或各自儲存自己的雜湊表，彼此之間並不會交換雜湊表的内容兩種方式。

(3) 每台機器皆提供一部份記憶體，以共用記憶體 (shared memory) 的方式組成存放雜湊表。

4.4 負載平衡

在一個分散式的系統中，為了確保系統的效能，必須設法將工作平均分配，以免因工作過度集中於少數幾台機器，造成許多機器的閒置。

就 α - β 切捨演算法的特性而言，可能某些子樹下絕大部分的節點都被截斷了，某些子樹能被截斷的節點卻很少。因此，搜尋不同子樹的時間是很難評估的。

五、系統效能

本系統的效能評估包含三個部分：

(1) 處理器速度。測量方式是執行一組整數運算的副程式，計算完成此副程式所需的時間。

(2) 主記憶體大小。在程式中初始化一個陣列，此陣列的大小和雜湊表及搜尋程式的大小相等。藉此比較主記憶體與雜湊表的大小，以得知是否使用虛擬記憶體。

(3) 網路傳輸速度。在各機器之間依次傳輸 10 至 10000 個整數資料，計算往返傳輸所耗費的時間，以平均值做為網路的傳輸速度。

我們以「順炮橫碑先手勝直車正馬」為例，列出三臺基器分別搜尋及同時搜尋所耗費的時間，如表一所示。

表一 棋局搜尋時間

| 使用機器 | NTUCSV | NTUCSX | NTUCSL | Distributed | 效率增進 |
|------|----------|-----------|-----------|-------------|--------|
| 搜尋六層 | 64 秒 | 1 分 35 秒 | 1 分 50 秒 | 33 秒 | 48.4 % |
| 搜尋七層 | 6 分 30 秒 | 13 分 20 秒 | 13 分 11 秒 | 2 分 56 秒 | 54.9 % |
| 搜尋八層 | 25 分 3 秒 | 64 分 16 秒 | 65 分 53 秒 | 11 分 15 秒 | 55.1 % |

六、結論與未來發展

參考文獻

6.1 結論

本報告探討循序式的 α - β 切捨演算法，並分析將此演算法應用於分散式計算的環境中，所面臨的問題及解決的方法，並以此時作一個電腦象棋程式，有效地提升電腦象棋思考的速度。此象棋程式所實作的環境為 PVM，藉由 PVM 所提供的各項功能，在程式發展的過程中節省了不少的時間。

本報告所討論的分散式 α - β 切捨演算法，不僅可用於電腦象棋程式，相同的架構也可用來實作任何兩人對奕、零和、定義明確的遊戲，例如西洋棋、Othello 等。

在系統實作的過程中，結合了許多相關象棋的研究成果，包括審局函數的設計、 α - β 切捨演算法的分析與修正等。未來若有新的研究成果，在更新系統的程式模組之後，將可增強電腦象棋程式的棋力。

6.2 未來發展

本系統在研發時，所使用的作業環境為 PVM。雖然因 PVM 強大的功能而減少了程式發展的困難度，但整個系統卻受限於 PVM 所支援的作業系統及硬體架構。舉例而言，若能將網路傳輸的機制改為 TCP/IP 協定，使用 Socket 的方式來傳遞訊息，將可避免受限於 PVM 所支援的平台。此外，直接以 Socket 傳遞資料，還可免除因 PVM 所需耗費的時間。

- 【1】許舜欽，“電腦對局的搜尋技巧”，台大工程學刊，第 51 期，頁 17-31，1991
- 【2】許舜欽、林益興，“電腦象棋的盲點解析”，電腦學刊，第三卷，第四期，頁 1-6，1991
- 【3】陳俊麟，“分散式計算系統 PVM 介紹”，1994 年 8 月
- 【4】R. Luling, B. Monien, “Load Balancing for Distributed Branch & Bound Algorithms”, Proc. Of the 6th International Parallel Processing Symposium, (IPPS '92), pp. 543-549, 1992.
- 【5】Pierre Boulet, Jack Dongarra, “Algorithmic Issues on Heterogeneous Computing Platforms”, Parallel Processing Letters, Volume9, Number 2, pp. 197-213, 1999
- 【6】Jack Dongarra, Markus Fischer, “Experiences with Windows 95/NT as a Cluster Computing Platform for Parallel Computing”, Journal of Parallel and Distributed Computing Practices (PDCP 1999 Vol. 2 Nr 2)