

工業工程專題

即時能源排程最佳化問題：從理論角度探討

指導老師：廖崇碩博士 組員：張綺芬、王珮瑩、林娉婷

前言

現在市面上行動裝置產品越趨多樣化，像是智慧型手機、平板電腦等，這類型產品雖然體積小、攜帶方便，但由於使用電池供電，僅能提供有限的電量，如何減少能源消耗以延長產品的使用時間就變成極為重要的議題。減少能源消耗有下列兩種機制：

(一) 動態速度規劃機制：由於處理器運作速度越快，能量消耗越高，因此藉由規劃處理器速度使能量消耗最小化。

(二) 休眠機制：當處理器閒置時仍會消耗能量，因此閒置一段時間後，將之轉為不消耗能量的休眠模式。

將行動裝置執行的每個動作視為一項工作，例如開啟網頁、播放音樂等，透過以上兩種機制將工作進行排程，使工作如期完成，並達到能源消耗最小化之目標。

由於採用雙核心與多核心處理器的產品越來越多，我們欲將目前單核心處理器的觀念延伸至雙核心處理器，提出一個合適的演算法。

問題描述

處理器可用不同速度運作，但一次只能執行一項工作。同一個時間點可有多個工作到達。每項工作包含以下資訊：工作量、到達時間、截止時間，工作可以延後開始或是提前結束，但必須在此時間區間內完成（見圖一）。

工作的輸入分為以下兩種：

- 靜態輸入（offline）：排程前已知所有工作資訊。
- 動態輸入（online）：工作到達當下才知道其工作資訊。

本研究是探討動態輸入的工作排程問題（即時排程問題），當有新工作到達時，目前的排程策略無法重新因應更動，因此動態的工作排程不能根據預先知道輸入工作的性質而發展有效率的演算法，所以在演算法的領域中，動態即時的最佳策略設計是非常困難的問題。

➤ 動態速度規劃機制

處理器的運作速度為 s ，表示每秒做多少工作量，運作速度會受工作量之多寡與實際執行時間影響。 $P(s)$ 表示處理器每秒所消耗的能量，為速度的函式，例如： $P(s)=s^3+16$ 。

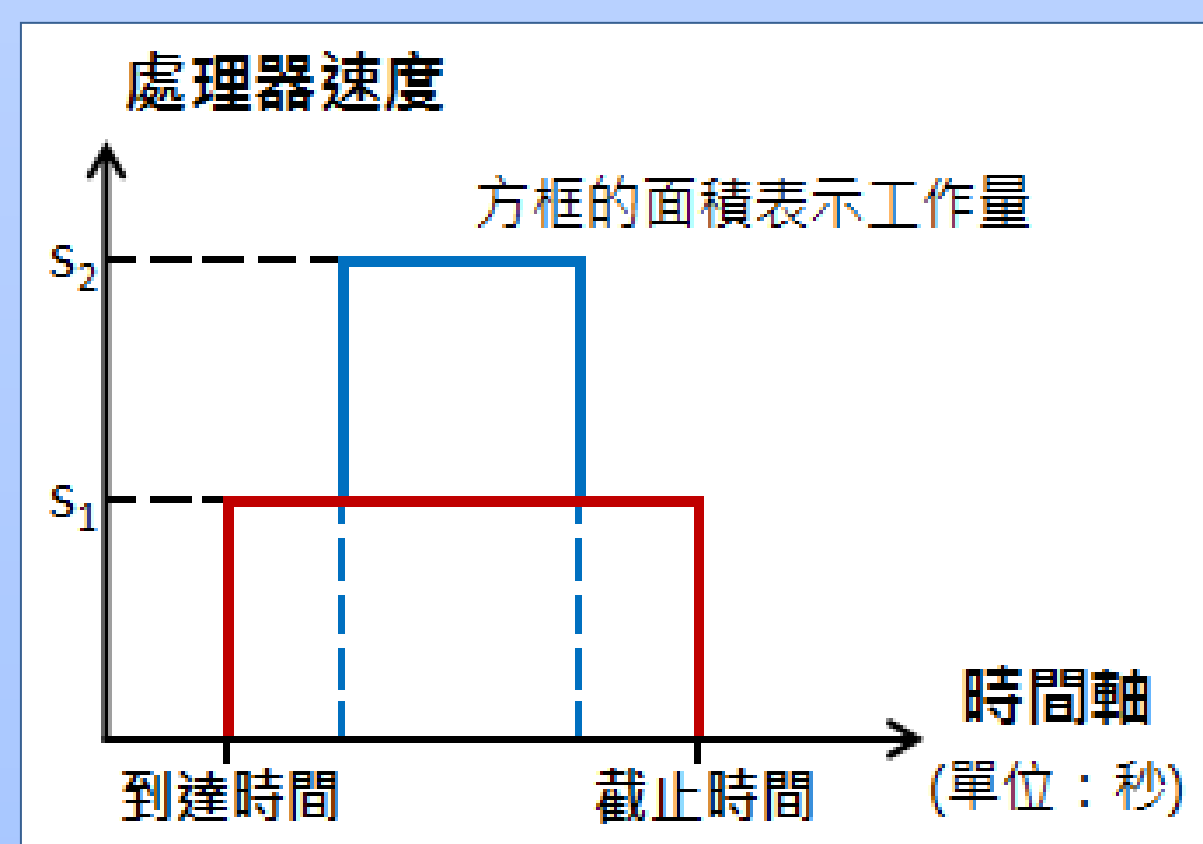
假設處理器總工作量为 R ，以速度 s 運作的執行長度為 R/s ，則能量為 $P(s) \times (R/s)$ ，將此式子微分可得出使能量為最小的速度，稱為critical speed，以 s_{crit} 表示之。（見圖二）

在現有排程下，當有新工作到達，若處理器可用 s_{crit} 在截止時間前完成工作，則以 s_{crit} 執行此工作，否則以大於 s_{crit} 的速度執行。

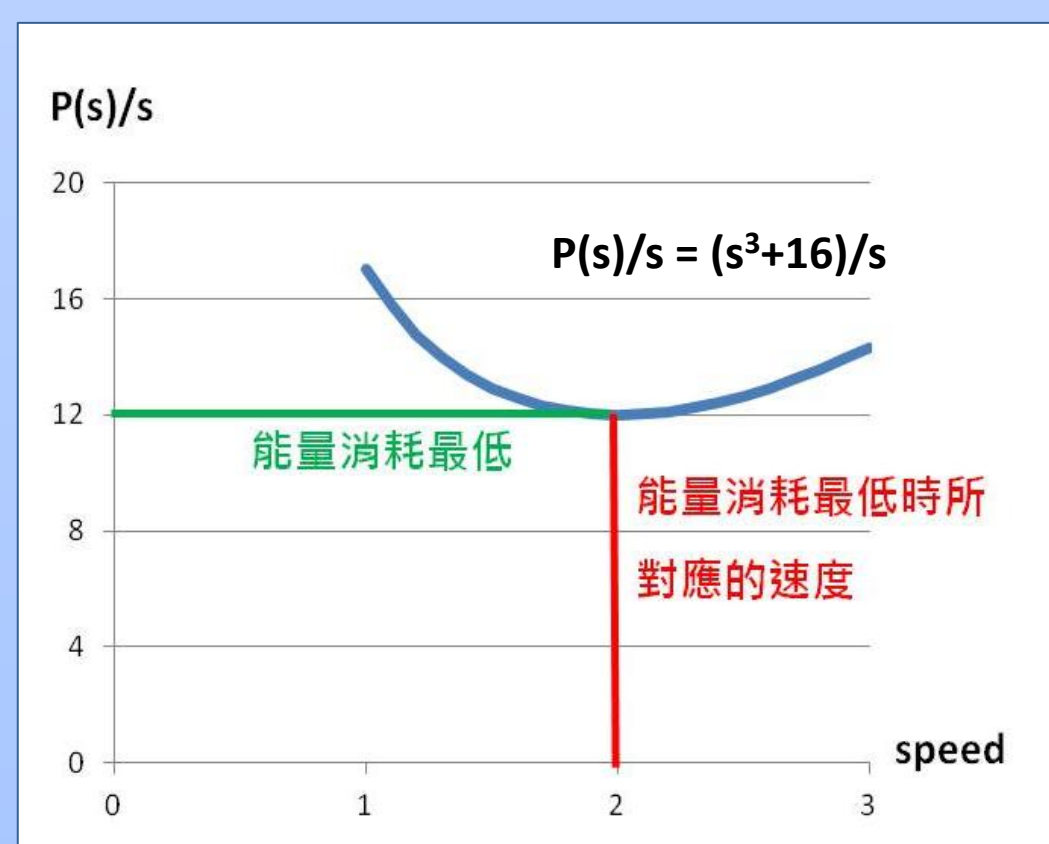
➤ 休眠機制——四種模式（見圖三）

依據critical speed的概念與休眠機制將處理器分成下列四種模式：

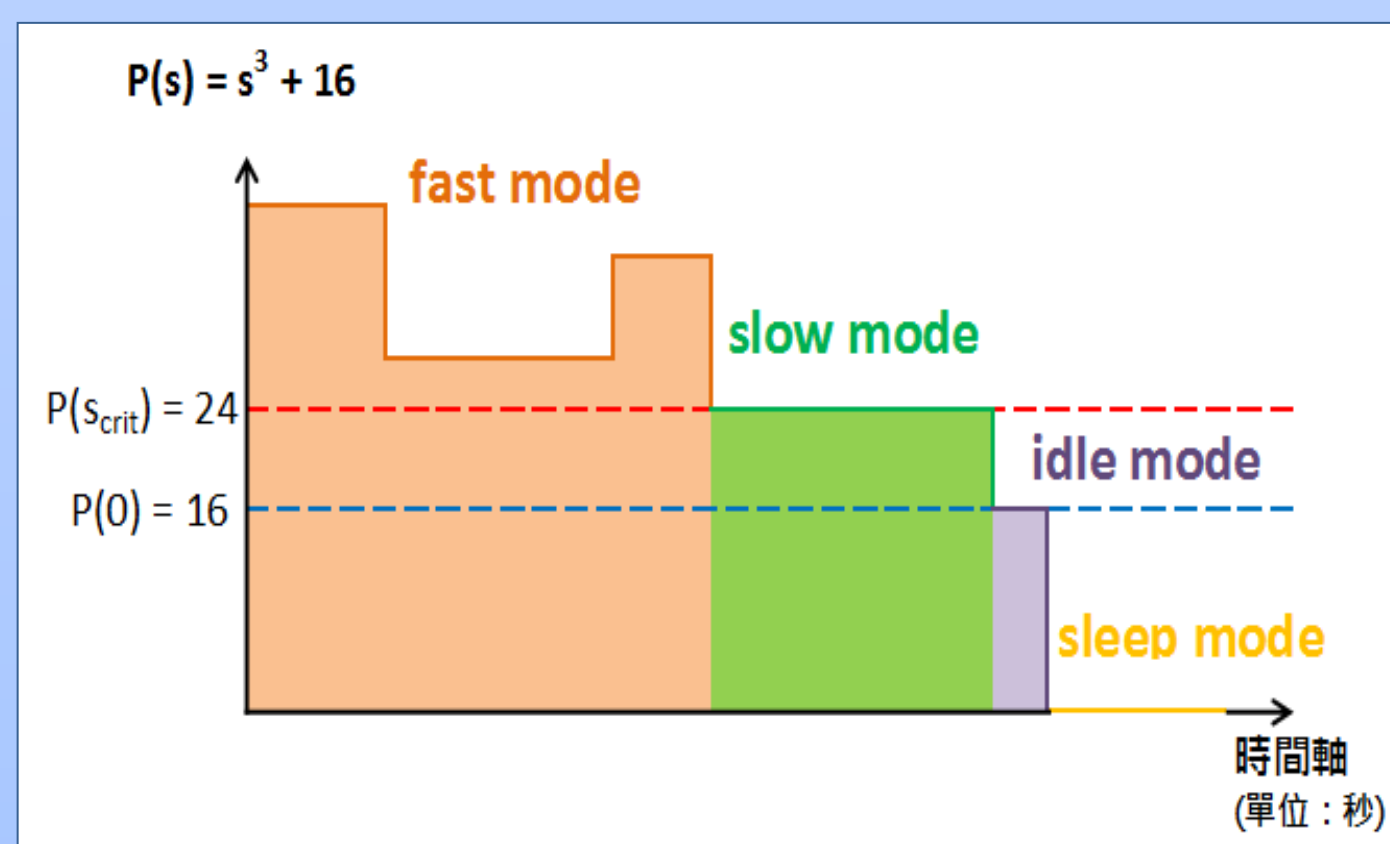
- Fast mode：處理器速度大於 s_{crit} 。
- Slow mode：處理器速度等於 s_{crit} 。
- Idle mode：處理器沒有在執行工作，速度為0，但仍需消耗固定能量以維持處理器開啟。
- Sleep mode：處理器閒置一段時間後進入不消耗能量的休眠模式，但重新開啟會消耗固定能量。



圖一、工作資訊



圖二、critical speed



圖三、四種模式之能量消耗

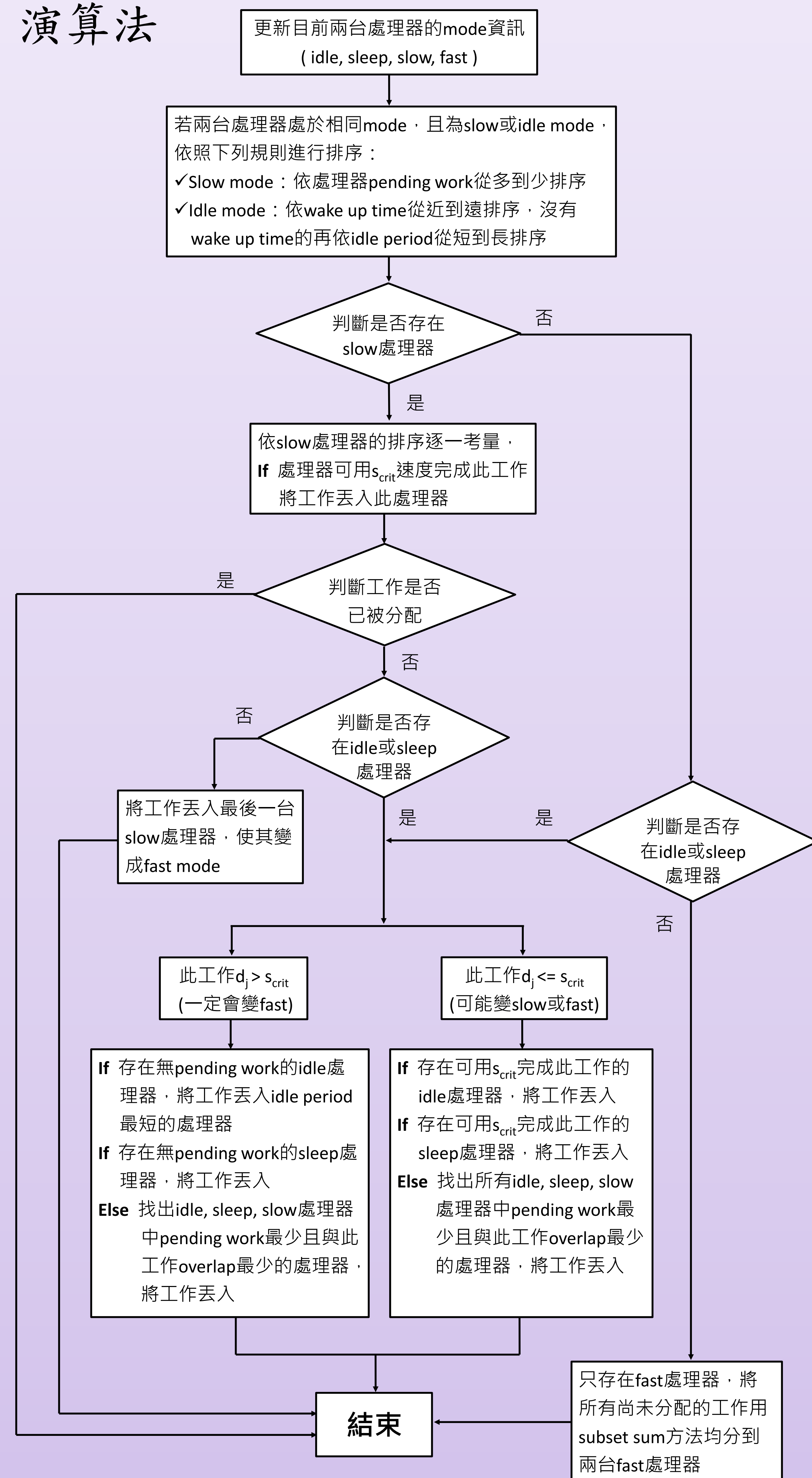
結論與展望

本研究針對以電池為供電的行動裝置，藉由結合「動態速度規劃機制」與「休眠機制」，設計出一適用於雙核心處理器的演算法，在工作如期完成的前提下，將動態輸入的工作依特定原則分配至處理器上進行排程，達到能源消耗最小化之目標，並證明此演算法能量消耗的competitive ratio為 $\max\{c_2c_1+c_2+4, 8\}$ 。

多核心處理器的排程問題更為複雜，推論上仍有幾點需要挑戰之處，像是工作分配問題、處理器的休眠與否以及competitive ratio的推導等，未來期望以雙核心處理器演算法的概念延伸至多核心處理器，以利更廣泛的應用。

研究成果

✚ 演算法



✚ Competitive ratio

Ratio = (online能量消耗 / offline OPT能量消耗)

	處理器核心數	Competitive Ratio
[1]A Scheduling Model for Reduced CPU Energy	單核心	When $P(s) = s^p$, ratio = $p^{p-2}p^p$
[2]Algorithms for Power Savings	單核心	ratio = $\max\{c_2c_1+c_2+2, 4\}$ c1: ratio of online algorithm with no sleep c2 : let $f(x) = P(x) - P(0)$, $f(x+y) \leq 2(f(x)+f(y))$
Our Results	雙核心	ratio = $\max\{c_2c_1+c_2+2, 8\}$ c1: ratio of online algorithm with no sleep c2 : let $f(x) = P(x) - P(0)$, $f(x+y) \leq 2(f(x)+f(y))$

Reference :

- [1] YAO, F., DEMERS, A., AND SHENKER, S. 1995. A scheduling model for reduced CPU energy. In FOCS. IEEE Foundations of Computer Science.
- [2] IRANI, S., SHUKLA, S., AND GUPTA, R. 2007. Algorithms for power savings. ACM Transactions on Algorithms.

