如何撰寫論文?

俞征武整理於 2007/3/4

中華大學資工系

大綱:

前言

1. 論文基本格式編排
2. 論文基本架構
3. 正確的寫作方式
4. 常犯的英文文法錯誤
5. 論文之引用
6. 論文之收藏
7. 寫作碩士論文之次序及合適時程
8. 作研究或寫論文應有的態度

前言

科技性質的學術論文有其一定的專業要求. 初寫這種論文的學生因較不熟悉,故常有很大的困擾.所以寫了一些提示, 希望提醒自己或對我的學生有所幫助.

第一章 論文基本格式

寫作論文一般每個 Journal 有自己的規定. 但整篇論文必須有統一的格式(format)卻是一致遵守的習慣.注意兩三個格式的錯誤就可能造成退稿.以下列出常見的規矩:

1.1 字體與字型：

- Text 中 的 font 爲 times new roman, 字體大小爲 12 點字.
- 每個 section 中的 title 其字體大小爲 14 (小 section)或 16 點字(大 section).
- Paper 的標題(title) 的字體大小爲 18 點字, 而且需置中.

1.2 段落:

- 文章中每段落左右需切齊.
- 每一個 section 前需空一行.
- 每一個 paragraph 前需縮排.

1.3 空格:

- 逗號後面接一個空格, 如: … in the paper, hence ….
- 在兩個 word 中只用一個空白分隔, 不可使用兩個空白以上.

1.4 斜體:

- 論文中所有變數都要斜體, 如: When $x>y$, the algorithm ….
- 論文中自行定義的名詞在第一次出現時需斜體, 之後就不再用斜體.
- 論文中所有constant不要斜體, 如: When $x_1>y_2$, the algorithm …. 其中 1, 2 雖然在下標一樣不需斜體.

1.5 其他:

- 縮寫要用小括號, 如 Random Access Memory (RAM).
- 使用三個小點(…)代表省略, 如 x_1, x_2, …, x_n;
- 參考論文之引用需用中括號, 如: In [16, 23], we have presented….
- 利用大括號代表集合(set), 如: Let S={1, 2, 3, 4, 5}, ….
- 整篇文章不使用粗體(bold).
- Figure 的說明在正下方且置中, Tables 的說明在正上方且置中.
- 定理證明結束的最後一行的最右方寫入“ ■”.
- 行間的距離請用 double space, 雖然不同的刊物有不同的要求.

第二章 一個論文的範本

以下利用一個例子來說明論文的基本架構.

論文標題(title)：最精簡地指出論文的特色及貢獻

【範例】

Irregular Redistribution Scheduling by Partitioning Messages

Chun-I Chen, Chang Wu Yu, Ching-Hsien Hsu, Kun-Ming Yu, and C.-K. Liang
Department of Computer Science and Information Engineering
Chung Hua University, Hsinchu, Taiwan 300, R.O.C.
{cwyu, chh, yu, ckliang}@chu.edu.tw

摘要(Abstract)

撰寫摘要可用一句話點明研究問題的領域及重要性。用兩三句話定義研究問題及動機。再用兩三句話說明研究具體的成果，最後兩句話提及主要貢獻。

【範例】

Abstract

Dynamic data redistribution enhances data locality and improves algorithm performance for numerous scientific problems on distributed memory multi-computers systems. Regular data distribution typically employs BLOCK, CYCLIC, or BLOCK-CYCLIC(*c*) to specify array decomposition. Conversely, an irregular distribution specifies an uneven array distribution based on user-defined functions. Performing data redistribution consists of four costs: index computational cost, schedule computational cost, message packing/unpacking cost, and data transfer cost. Previous results focus on reducing the former three costs. However, in irregular redistribution, messages with varying sizes are transmitted in the same communication step. Therefore, the largest sized messages in the same communication step dominate the data transfer time required for this communication step. This work presents an efficient algorithm to partition large messages into multiple small ones and schedules them by using the minimum number of steps without communication contention and, in doing so, reducing the overall redistribution time. When the number of processors or the maximum degree of the redistribution graph increases or the selected size of messages is medium, the proposed algorithm can significantly reduce the overall redistribution time to 52%. Moreover, the proposed algorithm can be applied to arbitrary data redistribution while slightly increasing the communication scheduling time.

Keywords: data redistribution, scheduling, edge coloring, bipartite graphs, multi-graphs

1. 簡介(Introduction)

1.1 首先界定您的研究問題的研究領域並說明此領域在廣度上的重要性。

【範例】

Parallel computing systems have been extensively adopted to resolve complex scientific problems efficiently. When processing various phases of applications, parallel systems normally exploit data distribution schemes to balance the system load and yield a better performance. Generally, data distributions are either regular or irregular. Regular data distribution typically employs BLOCK, CYCLIC, or BLOCK-CYCLIC(c) to specify array decomposition [14, 15]. Conversely, an irregular distribution specifies an unevenly array distribution based on user-defined functions. For instance, High Performance Fortran version 2 (HPF2) provides a generalized block distribution (GEN_BLOCK) [19, 20] format, allowing unequally sized messages (or data segments) of an array to be mapped onto processors. GEN_BLOCK paves the way for processors with varying computational abilities to handle appropriately sized data.

Array redistribution is crucial for system performance because a specific array distribution may be appropriate for the current phase, but incompatible for the subsequent one. Many parallel programming languages thus support run-time primitives for rearranging a program's array distribution. Therefore developing efficient algorithms for array redistribution is essential for designing distributed memory compilers for those languages. While array redistribution is performed at run time, a trade-off occurs between the efficiency of the new data rearrangement for the coming phase and the cost of array redistributing among processors.

1.2 指出問題的研究動機:也就是在此研究領域上,有何新研究問題需要被解決?前人的成果中有那些缺失處或未考慮的因素?或有那一些傳統難題未被徹底解決的?(本節十分重要, 有時需考慮以一個獨立章節討論之)

【範例】

Performing data redistribution consists of four costs: index computational cost T_i, schedule computational cost T_s, message packing/unpacking cost T_p and data transfer cost. The index and schedule computations are executed in complier time, with the remaining in run time. The data transfer cost for each communication step consists of start-up cost T_u and transmission cost T_t. Let the unit transmission time τ denote the cost of transferring a message of unit length. The total number of communication steps is denoted by C. Total redistribution time equals $T_i+T_s+\sum_{i=1}^{i=c}(T_p+T_u+m_i\tau)$, where m_i =Max$\{d_1, d_2, d_3, .., d_k\}$ and d_j represents the size of message scheduled in i^{th} communication step for j=1 to k.

Previous results focus on reducing the former three costs (i.e., T_i, T_s, and T_u). In irregular redistribution, messages of varying sizes are scheduled in the same communication step. Therefore, the largest size of message in the same communication step dominates the data transfer time required for this communication step.

1.3 前述的問題期望如何被解決：如降低演算法的時間複雜度或減少電源的耗費或降低網路部署的成本等。儘可能提出明確的方向，如可量化的數量或質性的改善。說明你要解決或是想更進一步瞭解的問題?

【範例】

Based on the fact, this work presents an efficient algorithm to partition large messages into multiple small ones and schedules them by using the minimum number of steps without communication contention and, in doing so, reducing the overall redistribution time.

1.4. 強調研究動機之重要性及創意：說明，如果此問題沒被解決或是充分瞭解，會有多大的負面的問題?

1.5. 明確地直接地說明本研究的目標。

1.6. 您的研究將利用何種技巧或直覺概念，來達成預期的成果。

【範例】

Specifically, the minimum value of T_s, and C are derived, along with the value of m_i reduced by shortening the required communication time for each communication step.

1.7.說明主要的貢獻及相關證明或實驗成果(佐證)。

最好加上明確的成果會量化的數據，以避免留於空談。

1.8 說明此研究對整個領域的長期影響及貢獻?(廣度)

【範例】

When the number of processors or the maximum degree of the redistribution graph increases or the selected size of messages is medium, the proposed algorithm can significantly reduces the overall redistribution time to 52%. Moreover, the proposed algorithm can be applied to arbitrary data redistribution while slightly increasing the communication scheduling time.

緊接著是整篇論文的大綱:

【範例】

The rest of the paper is organized as follows. Section 2 presents necessary definitions and notations. Next, Section 3 describes the basic graph model along with related work. The main contribution of the paper is shown in Section 4. We also conduct simulations in Section 5 to demonstrate the merits of our algorithm. Finally, Section 6 concludes the paper.

2. 前人成果介紹(survey)

最近五年內的重要期刊及會議都應搜尋並列於此地(可利用 google，但小心網路上的資料很多不嚴謹，需小心過濾查證後才引用)。

前人成果介紹是任何一篇論文不可缺少的一部份。此部份需要將和本研究的直接相關的前人成果，全部列入。只要有相關重要前人成果未被提及，有可能成爲論文被拒絕的理由，評審者會質疑此研究之原創性。

每段落以 5~6 行提及每一篇前人的成果。需用中括號[]點出參考文獻。將前人成果的優缺點加以說明。並以此引出本論文研究的動機，以突顯本論文之重要性。但說前人缺點時要小心，因爲該位作者很可能是審稿者。

【範例一】

AODV (Ad Hoc On-Demand Distance-Vector) [16, 18] Routing is an improvement to the table-driven and distance-vector based DSDV algorithm. With DSDV (Destination-Sequenced Distance-Vector) Routing [17], every mobile node maintains a routing table recording all the possible destinations and number of hops to each destination. In order to maintain routing table consistency, it requires nodes to periodically broadcast routing updates throughout the network.

【範例二】

Techniques for regular array redistribution can be classified into two groups: the communication sets identification and communication optimizations. The former includes the *PITFALLS* [17] and the *ScaLAPACK* [16] methods for index sets generation. Park *et al.* [14] devised algorithms for BLOCK-CYCLIC data redistribution between processor sets. Dongarra *et al.* [15] proposed algorithmic redistribution methods for BLOCK-CYCLIC decompositions. Zapata *et al.* [1] designed parallel sparse redistribution code for BLOCK-CYCLIC data redistribution based on *CRS* structure. Also, the *Generalized Basic-Cycle Calculation* method was presented in [3]. Techniques for communication optimizations provide different approaches to reduce the communication overheads in a redistribution operation.

3．定義或背景(notations, definitions, and background)

將本文中所須的定義或背景知識，在此段中說明。慎選符號(notation)最好所有的論文有一致性的符號(可利用共同檔案集中管理及使用)。符號需置於適當的地方，以方便閱讀。

【範例】

Any data redistribution can be represented by a bipartite graph $G=(S, T, E)$, called a *redistribution graph*. Where S denotes source processor set, T denotes destination processor set, and each edge denotes a message required to be sent. For example, a Block-Cyclic(x) to Block-Cyclic(y) data redistribution from P processors to Q processors (denoted by $BC(x, y, P, Q)$) can be modeled by a bipartite graph $G_{BC(x, y, P, Q)}=(S, T, E)$ where $S=\{s_0, s_1, \ldots, s_{|s|-1}\}$ ($T=\{t_0, t_1, \ldots, t_{|t|-1}\}$) denotes the source processor set $\{p_0, p_1, \ldots, p_{|s|-1}\}$ (destination processor set$\{p_0, p_1, \ldots, p_{|t|-1}\}$) and we have $(s_i, t_j)\in E$ with weight w if source processor p_i has to send the amount of w data elements to destination processor p_j. For simplicity, we use $BC(x, y, P)$ to denote $BC(x, y, P, P)$.

4. 主要成果(main results)

在此節中訴說主要成果。

4.1 直覺式的介紹主要的創意及技巧(Describe intuitive ideas)

先提主要的創意及技巧的直覺式概念以top-down的易懂方式介紹。

【範例】

Given a redistribution graph G with its edge coloring, the edges colored the same is a matching of G; thus represents a set of conflict-free data communication. Accordingly, for a given data redistribution problem, a conflict-free scheduling with the minimum number of communication steps can be obtained by coloring the edges of the corresponding redistribution graph G. When G is bipartite, it is well known that $\chi'(G)=\Delta(G)$ [22]. As a result, the minimum number of required communication steps equals the maximum degree Δ of the given distribution graph G.

Previous work is equivalent to finding out an edge colorings $\{E_1, E_2, E_3, \ldots, E_\Delta\}$ of G so that $\sum_{i=1}^{\Delta} \max\{w_k \mid e_k \in E_i \text{ where } w_k \text{ is the weight of } e_k\}$ (i.e., the data transfer time) can be decreased. To the best of our knowledge, it is still open to devise an efficient algorithm to minimize both of the overall redistribution time and communication steps.

Unlike existing algorithms, the main idea behind our work is to partition large

data segments into multiple small data segments and properly schedule them in different communication steps without increasing the number of total communication steps.

4.2 舉例說明(An example)

利用一個簡單例子來說明論文的精神常有畫龍點睛之效。

【範例】

For example, Figure 6 depicts a redistribution graph with the maximum degree $\Delta=4$.

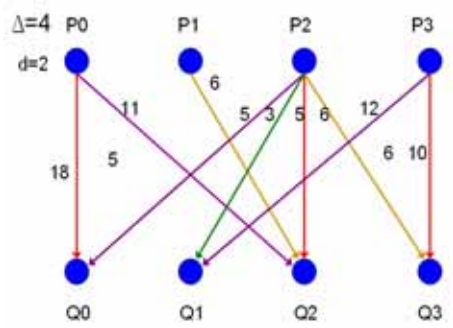

Figure 6. A redistribution graph with $\Delta=4$.

We need four communication steps for this data redistribution since $\chi'(G)=\Delta(G)=4$. In addition, the overall cost of the corresponding scheduling is 38 (See Table 1).

Table 1. The scheduling corresponds to the edge coloring in Figure 4.

Step	1(red)	2(yellow)	3(green)	4(purple)	Total
Cost	18	6	3	11	38

Note that the time cost of Step 1 (colored in red) is dominated by the data segment (with 18 data elements) from P_0 to Q_0. Suppose that we evenly partition the segment into two data segments (with 9 and 9 data elements respectively) and transmit them in different steps; then the time required for Step 1 is reduced to 10 (dominated by the data segment from P_3 to Q_3). Note that the data partition adds an edge (P_0, Q_0) in the original redistribution graph. Similarly, we can partition any large data segment into multiple small data segment if the maximum degree of the resulting redistribution graph remains unchanging. After several data partitions, the overall communication cost can be reduced to 29 and the number of required communication step is still minimized (see Figure 7 and Table 2).

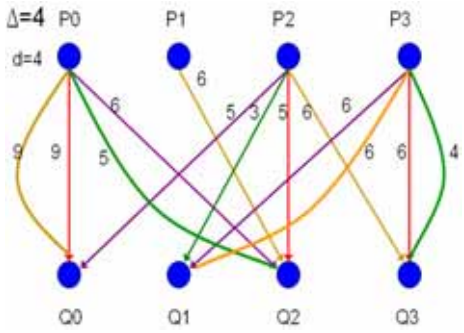

Figure 7. The resulting redistribution graph after partitioning long data segments.

Table 2. The scheduling after partition long data communications.

Step	1(red)	2(yellow)	3(green)	4(purple)	Total
Cost	9	9	5	6	29

4.3 較正式的成果說明。

利用演算法、定理或其它正式的方式來精準地描述成果以突顯成果的嚴謹度。

【範例】

The algorithm of the selection step is shown as follows.

Algorithm Selection()

Input: A redistribution graph $G=(S, T, E)$ with maximum degree Δ.

Output: A redistribution graph $G=(S, T, E \cup D)$ with maximum degree Δ, where D represents those dummy edges added in the algorithm.

Step 1. Select the edge $e_k=(s_i, t_j)$ from E such that the value $w_k/(1+v_k)$ is the largest and $d_G(s_i)<\Delta$ and $d_G(t_j)<\Delta$, where v_k denotes the number of added dummy edge with the same end points of e_k. If no such edge exists, terminate this algorithm.

Step 2. Add a dummy edge $e_{k'}=(s_i, t_j)$ to D and set $v_k=v_k+1$.

Step 3. Go to Step 1.

The time complexity of Selection is $O(m\log m)$, where m is the size of edge set of the input redistribution graph.

5. 理論上的證明(Proof)

針對所解的題目提出理論上的證明或數學上的模式有助於提昇研究的深度。論文少了此部份不易登上好的 conference 及 Journal.。

【範例】

Theorem 3: Algorithm AMEC is a 2-approximation algorithm for the max-edge-coloring problem.

Proof: Since an edge coloring of an graph G corresponds to a vertex coloring of its line graph $L(G)$, the *max-edge-coloring* problem of G can be transformed to the *max-coloring* problem of $L(G)$. Since $L(G)$ is an interval graph (by Theorem 2), we obtain a 2-approximation algorithm for max-edge-coloring problem of G by applying Pemmaraju *et al* 's 2-approximation algorithm for the max-coloring problem on interval graphs [28]. Finally, we conclude that Algorithm AMEC is a 2-approximation algorithm for the max-edge-coloring problem. ■

6. 實驗或模擬結果(experiments and simulation results)

所有實驗或模擬結果，都是設計來驗證本論文的正確性及強化本論文的貢獻度。故需小心設計，以客觀的數據說明現象。實驗或模擬結果，務必需公正無作假。圖表需夠大及清楚，否則會造成退稿。注意實驗或模擬結果，務必保留並有 LAB 專人承接並有修改的能力，否則投稿後需修改實驗時，會有大問題。

6.1 實驗環境之介紹

實驗環境中會影響實驗結果的所有參數皆需詳細介紹:

1. 實驗系統軟硬體平台: ns-2, 或是用 c++設計的。
2. 每個數據是重覆多少次之後的平均。
3. 各種實驗的假設需明確引用他人的論文及說明。
4. 需和前人的重要方法作比較, 否則易受攻擊。

【範例】

Our simulations were conducted in C for `GEN_BLOCK` distribution. Given an irregular array redistribution on $A[1{:}N]$ over P processors, the average size of data blocks is N/P. Let $T_b(T_a)$ denote the total redistribution cost without (with) applying our algorithm. The *reduction ratio* R equals $(T_b\text{-}T_a)/T_b$. Moreover, let $\{E_1, E_2, E_3, \ldots, E_\Delta\}$ of G denote the output of Scheduling step. We also define $C_i = \max\{w_k | e_k = (u,v) \in E_i \text{ and either } d(u) = \Delta \text{ or } d(v) = \Delta, \text{where } w_k \text{ is the weight of } e_k\}$.

As a result, the overall redistribution time is bounded by $B = \sum_{i=1}^{i=\Delta} C_i$ since the proposed algorithm does not select maximum-degree edges for further partition. Otherwise, the required communication step will be increased.

To thoroughly evaluate how our algorithm affects the data transfer cost, our simulations consider different scenarios. Each data point in the following figures represents an average of at least 10000 runs in each different scenario.

The first scenario assumes that the size of data array is fixed, i.e., N=100; the

number of processors range from 4, 8, 16, 32, 64, to 128; the size of data blocks is randomly selected between 1 and 50. In Figure 12, the value of T_b drastically raises as the number of processors increases. However, after applying our algorithm, the overall distribution time T_a smoothly raises as the number of processors increases. Note that the B value drops as the number of processor increase due to the decrease of the average values of data elements in a single communication. In short, when the number of processors increases, the reduction ratio R raises if applying our partition algorithm.

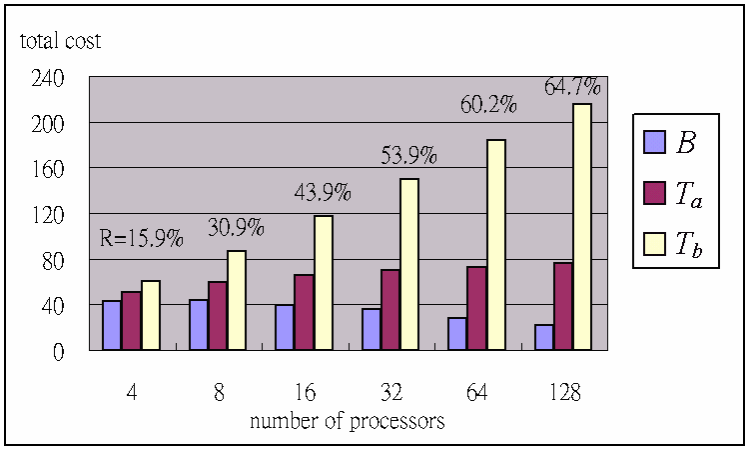

Figure 12. Simulation results of Scenario I.

6.2 實驗結果之解讀及批判

每一個實驗結果需被解讀及批判，並和前面的成果相呼應。

【範例】

The second scenario assumes that the number of processors is fixed, i.e., P=32; the size of data array N equals 1600, 3200, 6400, 9600, or 12800; and the size of data blocks is randomly selected between 1 and 2×(N/P). As shown in Figure 13, the values of T_a, T_b, and B raises as the size of data array N increases due to the increase of the average number of data elements in a single communication. However, the reduction ratio stays about 52% by applying our partition algorithm, even with the large size of data array.

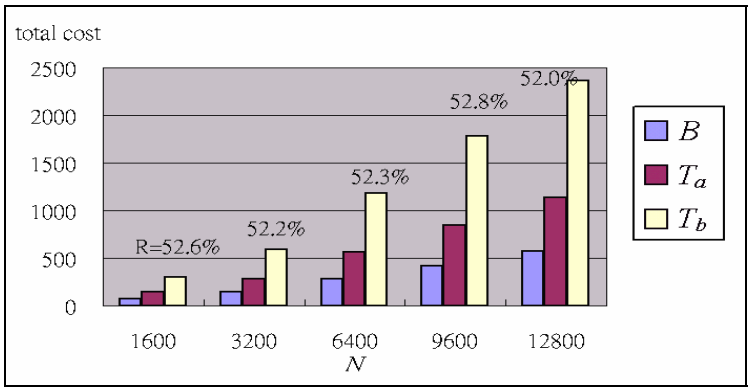

Figure 13. Simulation results of Scenario II.

7. 結論(conclusion)

結論是整篇論文的收官。若能餘音饒樑則必成佳作。

- 簡單摘要整篇論文的貢獻及使用的技巧?
- 此研究對該領域有何啓發?
- 貢獻有何處可擴展?

1. Were your results expected? If not, why not?
2. What generalizations or claims are you making about your results?
3. Do your results contradict or support other experimental results?
4. Do they suggest other observations or experiments which could be done to confirm, refute, or extend your results?
5. Do your results support or contradict existing theory?
6. Do your results suggest that modifications or extensions need to be made to existing theory? What are they?
7. Could your results lead to any practical applications?

- Stress how the results in this study confirm your engineering/Scientific motivations (specific and general) and, ultimately, your reader's interests (i.e. Engineering/Scientific need).

【範例】

We have presented an efficient algorithm to reduce the overall redistribution time by applying data partition. Simulation results indicates that when the number of processors or the maximum degree of the redistribution graph increases or the selected size of data blocks is appropriate, our algorithm effectively reduce the overall redistribution time. In future, we try to estimate the reduction ratio precisely. We also believe that the techniques developed in the study can be applied to resolve other scheduling problems in distribution systems.

參考文獻(References)

參考文獻需照作者姓之順序排列。兩位作者中需用 and 相連。三位作者中需用，及 and 相連。引用論文年次(year)、期數(vol), 頁次(page) 皆需明確記錄。注意 Journal 及 conference 的順序不同。另外此順序需一致，不可擅自改此規則。

【範例】

References:

[1] G. Bandera and E.L. Zapata, "Sparse Matrix Block-Cyclic Redistribution," *Proceeding of IEEE Int'l. Parallel Processing Symposium* (IPPS'99), San Juan, Puerto Rico, April 1999.

[2] C.-H Hsu, Dong-Lin Yang, Yeh-Ching Chung, and Chyi-Ren Dow, "A Generalized Processor Mapping Technique for Array Redistribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, vol. 7, pp. 743-757, July 2001.

[3] Minyi Guo, "Communication Generation for Irregular Codes," *The Journal of Supercomputing*, vol. 25, no. 3, pp. 199-214, 2003.

[4] Minyi Guo, Yi Pan, and Zhen Liu, "Symbolic Communication Set Generation for Irregular Parallel Applications," *The Journal of Supercomputing*, vol. 25, pp. 199-214, 2003.

第三章 正確的寫作方式

論文寫作著實不易。如何將您的研究成果, 用讀者**易讀(無負擔)**的方式，**準確地**，**邏輯地**，**結構化**地表示出論文的**創新貢獻**。不過論文於寫作前，需先確定您的研究確實達到「完整」及「美」的境地。再開始修改的程序，以免徒勞無功。每一篇論文需符合下列的每一個條件。

3.1 絕對不可抄襲

所有論文的最重要的價值在於其原創，故絕對不可抄襲他人成果並佔爲己有。此行爲會造成學術上嚴重的道德倫理問題。若有引用他人的成果,一定得引用其論文及註明出處並改寫所引用的文句。

3.2 突顯您的貢獻及創意

以下是一些可突顯您的貢獻及創意的方法。

1. 背景(background)說明須完整，尤其是前人的成果及弱點。
2. 突顯本研究的動機(motivation)常可用來突顯論文貢獻的重要。
3. 表達文字需準確簡潔。
4. 一篇好文章一般需要以下特色：
   甲、解決的問題是有意義的。
   乙、有新的創意。
   丙、嚴謹的數學分析或證明以支持貢獻。
   丁、仔細地實驗比較以支持貢獻。
   戊、易讀且前後文呼應。

3.3 可讀性，高度結構化(well structured)

1. 寫一句文章後，必須確認是否讀者可以輕鬆地就可了解。
2. 勿用太多符號干擾讀者。
3. 多利用圖解及實例。

3.4 表達的邏輯需精準嚴密

要增加文章的可讀性，則文章段落文句間的邏輯需精準嚴密。上一個段落和下一個段之間，邏輯上必須因果相互連貫的。寫一句文章後，必須確認是否已經充分將您的概念表達出來了。寫完文章後，必須不斷重新改寫，使之更簡潔有力。無關的章節需捨去。

3.5 格式需絕對正確

文中出現兩個以上格式出錯，許多 reviewer 會給與至少 major revision 的處罰。

3.6 前人文獻需完整

否則易使 reviewer 懷疑您的原創性。

3.7 英文文句需無一處錯誤

論文中有兩三個英文文句有錯，易得到 reject 的下場。

以下爲缩短文摘方法(部份引用自 EI)：

1. 取消不必要的字句及一些不必要的修辭詞：如 ”It is reported …” “Extensive investigations show that…” “The author discusses …” “This paper concerned with …” ” In this paper,” “in detail”、“briefly”、 “here”、 “new”、 “mainly”。
2. 限制文摘只表示新情況，新內容，过去的研究细节可以取消；
3. 不說無用的話。
4. 作者在文献中谈及的未来計劃不纳入文摘。
5. 儘量簡化一些措辞和重覆的單元。
6. 文摘第一句应避免与题目（Title）重复。
7. 不可有任何拼錯字的狀況發生.

3.8 閱讀自己的文章

先說明你的創見的直覺概念。再用例子解釋 ideas，最後用 algorithm 或較嚴謹的方式來準確地描述方法。於適當時機提出證明或實驗。

3.9 投稿需知:

1. 投稿後兩個月需 follow up。因爲 editor 有可能未收到您的論文。投完 conference 後，當 acceptance notification 回來後，就投稿於 Journal。需一股作氣完成否則易產生延宕的狀況。

2. Revision 需知:

revision report 必需每一條都回應，並需不卑不亢及使用委婉語氣。需完全展現誠意(例如回答比 comment 文句長)。並需註明我們在論文的何處，作如何修的改。儘可能於一週內回覆(打鐵乘熱，當 reviewer 印像仍新時回應會更快)。總之儘量愉悅 reviewer。

3. 常投之會議及期刊

期刊及會議論文

一些前人成果將可從下列 conference 或 journals 中找到(以 wireless networks 爲例):Conferences:(參考 http://lion.cs.uiuc.edu/links.html)

- Globecom 2006, 2005, 2004
- ICC 2006, 2005, 2004, 2003, 2002
- ICDCS 2006, 2005 2004
- ICNP 2006, 2004, 2003
- INFOCOM 2007, 2006, 2005 2004, 2003, 2002
- IPSN 2006, 2005, 2004, 2003
- MOBICOM 2006, 2005, 2004, 2003, 2002
- MOBIHOC 2007 2006, 2005, 2004, 2003, 2002
- SIGCOMM 2006, 2005, 2004
- WCNC 2007, 2006, 2005 2004

Journals

- IEEE/ACM Transactions on Networking
- IEEE Journal on Selected Areas in Communications
- IEEE Transactions on Communications
- IEEE Transactions on Information Theory
- IEEE Transactions on Mobile Computing
- IEEE Transactions on Wireless Communications
- IEEE Communications Letters
- ACM/Baltzer Wireless Networks
- ACM Transactions on Sensor Networks
- ACM/Baltzer Mobile Networks and Applications
- Baltzer Telecommunications Systems
- MONET

3.10. 擴大影響力

1. 若能寫一系列的論文可擴大貢獻。
2. 寫書也是一種對自己一段研究歷程後的檢視。

第四章 常犯的英文文法錯誤

您可能常犯的英文文法錯誤羅列如下：

1. even, therefore, then, 是 adverb, 不是 conjunction。
2. 一般而言，每一個句子只需一個動詞。

第五章 論文之引用

一但引用不好的論文有可能導致您的論文的價值被貶低。

以下我們介紹 IEEE 對引用論文時的規定

1. 排列順序要按照 Last name 的順序來排.
2. Journal, Conference, 書的名稱需斜體

引用專書時之寫法如下:

1. T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

引用 Journal 之寫法如下:

1. Michael Luby, "A simple parallel algorithm for the maximal independent set Problem," *SIAM Journal on Computing*, vol.15, pp. 1036-1053, 1986.

引用 Conference 之寫法如下:

1. L. Hu, "Distributed code assignments for CDMA packet radio networks, " in *Proc. INFOCOM*, 1991, pp. 1500-1509.

第六章 論文之收藏

學生於論文口試結束後，需根據口試教授擬一修正之清單，並完成修改後，需給我(精裝本)及所有口試教授一本論文完稿，並請將下列電子檔分不同目錄並燒成一份光碟給我.

1. 碩士論文(請用中文撰寫)。
2. 投稿會議或期刊論文(請用英文撰寫並根據會議或期刊規定之格式撰寫) 。
3. 論文投影片(請用英文撰寫)。
4. 實驗程式原始碼需加註解(需可重新安裝)。
5. 實驗數據及圖檔 (需分類清楚並用 mathlab 繪實驗數據的圖)。
6. 前人論文及相關文獻。
7. 個人英文簡介及照片(論文投稿使用)。

第七章 寫作碩士論文之合適時程

碩士第一年

10/1~6/31　承接學長之實驗並建立 Background 及尋找論文方向。

碩士第二年

7/1~10/31　每週討論並繳交週報(weekly report)以確認論文題目

11/1~12/31　論文演算法細節確認

1/1~2/31　開始設計論文實驗，論文實驗完成

3/1~31　中文論文投稿 mobile computing

4/1~30　論文投稿於國外會議

5/31　論文草稿之完稿

6/1~31　論文口試

7/1~31　論文之完稿及收藏。需將下列文件分不同 folder 存放；

- 碩士論文(請用中文撰寫)
- 投稿會議或期刊論文(請用英文撰寫並根據會議或期刊規定之格式撰寫)
- 論文投影片(請用英文撰寫)
- 實驗程式原始碼需加註解(需可重新安裝)
- 實驗數據及圖檔 (需分類清楚並用 mathlab 繪實驗數據的圖)
- 前人論文相關文獻(PDF file).
- 個人英文簡介及照片(論文投稿使用)

8/1　論文投稿於國外期刊

第八章 作研究或寫論文應有的態度

1. 寫論文需慎重其事，需盡全力去作。
2. 寫論文需有邏輯。
3. 未收集所有文獻時，不要倉促作研究，請多查 google 及與人討論。
4. 寫論文需要有動機，而喜歡寫論文的人才會將論文寫好。
5. 當問題重新定義正確時，答案就會浮現。Ask a good question or rephrase it in a right way then the answer follows.。
6. 寫論文如同編一個有創意的故事。此種論文易被引用。
7. 問題比解答重要，創意比方法重要。
8. single author 的論文發表代表個人的獨力研究能力。
9. 未臻完美的論文不要投，一篇論文修改 20~30 次是正常的。
10. 如何定義問題相當重要。
11. 將不相干的概念連在一起可激發創意。
12. 每個現象都應有一個正確的名字 (the broadcast storm problem) 。
13. 好的研究會有系統地討論一些主題，寫一系列的 paper.是 citation 多的原因之一。
14. 好的論文需要同時 good and original。
15. 寫論文避免 round robin 式，收尾時需不受任何干擾, conference 發表到 Journal 之發表需 pipelined, 勿拖延太久。
16. 整理並 output 有意義的圖表。
17. 主動和國外連繫，mail 自己的論文給相關領域的教授請求指正。
18. conference paper 一定最後成爲 Journal paper。
19. 不畏失敗，記取教訓，履敗履戰，鍥而不捨。
20. reviewer report 可幫助成長。
21. 面對問題的本質並求活路。
22. 想得比別人深一些。
23. 擴大戰果。
24. 看到未來的發展。Pioneering paper 容易發表。
25. 參加 high visible 的 conference 有助於 Team up 教授的力量。
26. 常常自問自己關心的研究是否爲 fundamental problems:到處會發生的問題。
27. 只作會在 resume 上增加一行事蹟的事。
28. 開創新名詞有時會成爲 pioneer。

參考書目:

1. 柯泰德線上英文論文編修網站http://owl.cmgt.nctu.edu.tw/
2. 林一平教授演講稿
3. 曾煜棋教授演講稿