

The UDC Spark

**The Community Newspaper of
The University of the District of Columbia**

**Vol. IV, Issue 4
February 24, 2006**



Personal Research Story: a Click needs Twenty Years Waiting

(Note: the following is the personal observations of Li Chen, Department of Electrical Engineering & Computer Science)

I was assigned to teach Computer Architecture this semester. We are using Stallings' Computer Organization and Architecture, a very popular text book. I just touched the material in coding theory last Monday for Memory Error Correction. "I was so surprised that the most popular code for error correction of Memory is still SEC-DED code (single error correction and double error detection)."

I was clicked immediately. If SEC-DED code is still used, then Hsiao's Codes is the best in SEC-DED codes. Thus, my work on generating the checker that tells a code is valid or not is still useful. I solved the problem when I was a college student while I was learning the Computer Memory too. The instructor, H-G Zhang, announced that: "Here is an unsolved question. Find a good algorithm to find the Check Matrix for the Optimal SEC-DED code (now called Hsiao's Codes)." What is a good algorithm? We were not very clear that time. We know today that must be a polynomial time algorithm.

I spent the winter holidays to get a solution in 1980. I was so excited. The paper received a student research award from Wuhan University. It was published five years later in 1986. This paper is still the best in my knowledge for the World well-known code because it is simple and elegant. More importantly, I knew today that it may be still useful.

When we teach students in up-division courses, to bring some state of the art unsolved problems may result an absolute research achievement. I would like to say many thanks to my professor H-G Zhang, he is still a professor in Wuhan University. I hope some days later, not need another twenty years, my student at UDC can call me and say thanks to me.

The UDC Spark

**The Community Newspaper of
The University of the District of Columbia**

**Vol. IV, Issue 4
February 24, 2006**



Dear Dr. Stallings,

I am teaching Section 5.2 of your book COA: Error Correction. I was so surprised that the most popular code for error correction of Memory is still SEC-DED code (single error correction and double error detection).

I have worked on how to generate the check matrix twenty years ago when I was a college student. I published my work five years later in 1986. I searched the Internet and found there are still some interests in generating the check matrix. There are some AI based algorithms that are proposed.

It is my pleasure to tell you and other related researcher. There is an optimum algorithm that can generate the matrix. There is no need to have an approximation algorithm.

The paper was reviewed by ZBL (German Math Abstract).

Li Chen, An optimal generating algorithm for matrix of equal-weight column and quasiequal-weight row, Journal of Nanjing Institute of Tech, 2(1986). pp 1-7.

The algorithm is an elegant algorithm. If you are interested, I would like to send you a copy.

Li Chen

Thank you. I would be interested. You can mail it to me at the address below

--

William Stallings
P. O. Box 2405
Brewster, MA 02631

> Just for your information, I have found an optimum algorithm to generate
> the check matrix of odd-column-weight Hsiao codes. The paper was in
> Chinese. If you have a Chinese student, I could mail a copy to you. The
> paper was reviewed by ZBL (German Math Abstract).
>
> Li Chen, An optimal generating algorithm for matrix of equal-weight
> column and quasiequal-weight row, Journal of Nanjing Institute of Tech,
> 2(1986). (Zbl, In Chinese)

Thank you very much for the information. Sounds like you did some nice work. I wish we were aware of it earlier. I would be interested to get a copy of the paper if you are willing to mail it. My address is below.
Thank you very much.

- Nur

Prof. Nur A. Touba
Department of Electrical & Computer Engineering
University of Texas, Austin, TX 78712-1084
(512) 232-1456
=====