

AN APPLICATION OF ELISA TO PERFECT HASHING WITH DETERMINISTIC ORDERING

M. Bianchini †, S. Fanelli ††, and M. Gori, IEEE Member †††

Email:fanelli@mat.uniroma2.it

†Dip. di Sistemi e Informatica, Università degli Studi di Firenze, Via S. Marta, 3 — Firenze, Italy
††Dip. di Matematica, Università di Roma “Tor Vergata”, Via della Ricerca Scientifica — Roma, Italy
†††Dip. di Ingegneria dell’Informazione, Università degli Studi di Siena, Via Roma, 56 — Siena, Italy

ABSTRACT

This paper describes a practical application of a novel terminal attractor algorithm to the construction of Perfect Hash Functions (PHF) for a predefined set of keys. The proposed method, which can be used in the ordering phase of a classical Mapping, Deterministic Ordering and Searching (MDOS) approach, is based on a neural network canonical form of efficient gradient descent, for solving linear systems, named ELISA. Numerical experiments clearly show that ELISA is able to determine the optimal solution in many difficult cases, where other alternative algorithms fail or are ineffective.

KEYWORDS: Perfect Hash Functions, Neural Networks, Computational Complexity, Terminal Attractors, Linear Systems with Singular Matrices.

1. INTRODUCTION

Given a predefined set N of n keys and a string T of r consecutive recordable addresses, a storing process can be performed through a *hash table of hash values* in the r locations. The associated retrieval problem is to determine the record corresponding to a key or to report that no such record exists. A *Perfect Hash Function (PHF)*, h , is an injection from N into T , i.e. h allows retrieval of records stored in T in one access. If $r = n$, h is called a *Minimal Perfect Hash Function (MPHF)* since it requires a minimum space utilisation. By examining the classical algorithms of [5] and [4], it can be easily seen that the Mapping, Ordering and Searching (MOS) method is an important contribution in the field of perfect hashing. In the MOS approach, in fact, the construction of a MPHF h is accomplished in three phases. Firstly, the *Mapping* step transforms the key set from the original classification to the desired representation. Secondly, the *Ordering* step solves the fundamental and most difficult problem of placing the keys in a sequential order, determining the order in which the hash values are assigned to the keys. The Ordering phase may partition the assigned order into subsequences of consecutive keys, called *levels* and the hash values must be assigned to the keys of each level at the same time. In the latter case, the final *Searching* step attempts to assign the hash values to the keys of each level.

In this paper we are interested to the Ordering phase. Assuming a formalisation and an extension of Cichelli’s approach (see [12]), it can be proved that the problem can be solved by employing an ordering heuristic, based on the determination of short cycles in the graph representing the constraints among keys. This heuristic algorithm, named *minicycle*, has a time complexity $\mathcal{O}(n^4)$ and is based on three pseudo-random functions. In [11] was suggested an alternative algebraic approach for code computing. By defining a proper isomorphism between keys and boolean matrices, it is possible to translate MPHF into an algebraic context and considering hashing as an *interpolation problem*. Since the latter ap-

proach requires the resolution of a linear system, it exhibits in general a time complexity $\mathcal{O}(n^3)$ (see also [8]).

Although a probabilistic approach to MPHF is extremely effective in some cases (see [7, 9]) and may lead to the implementation of a stochastic algorithm having *expected time complexity* $\mathcal{O}(n)$, the fundamental problem of improving the efficiency of a Deterministic Ordering in the MOS approach is still an open question (MDOS).

A remarkable contribution, in particular, was proposed in [6], where it was described a heuristic algorithm, based on the construction of a suboptimal spanning tree with a time complexity $\mathcal{O}(n^2)$. However, from a theoretical point of view, since the computational complexity of the Ordering phase in the MDOS approach is, in general, $\mathcal{O}(n^3)$ or $\mathcal{O}(n^2 \log_2 7)$, it is worthwhile investigating the performances of new algorithms.

If the algebraic approach for code computing is used, as suggested in [11], and n indicates the number of distinct keys, it is important to distinguish two situations, respectively named *under- n* case and *equal- n* case. In the latter case, feasible solutions can be often efficiently found [11], while in the former one, there is no guarantee for obtaining a solution.

This work deals with the *under- n* case in order to investigate the possible solutions when hash tables cannot be mapped and arranged directly in a lexicographic order. Since, in this case, the corresponding linear system has a singular matrix, it is practically impossible to apply classical tools to solve the problem. The method presented in this paper is based on a special neural network canonical form of gradient descent, approaching the minimum of a quadratic function in constant time. The associated algorithm ELISA (*Efficient Linear System Algorithm*), described in [3], represents a suitable application of the theory of *Terminal Attractors*, introduced in [14] and implemented in the field of neural networks in [1, 2, 13].

2. PROBLEM DESCRIPTION

The following notations are utilised:

n	number of predefined keys;
s	number of distinct letters used for coding the keys;
m	maximum key length;
K_h	h th letter of a generic key;
L	s -dimensional vector representing the letters;
C	s -dimensional vector representing the code letters;
Γ	m -dimensional vector representing the letter weights;
T	vector of recordable consecutive addresses;
y_t	address of the t th key; Y denotes the address vector.

Assuming that each key can be represented by a suitable boolean matrix P_k , $k = 1, \dots, n$, defined in the following way:

$$\text{if } K_j = L_i \text{ then } p_{ij} = 1 \text{ else } p_{ij} = 0,$$

the address vector Y can be determined by the linear relationship:

$$Y = W_\Gamma C, \quad (1)$$

where

$$W_\Gamma^T = [w_1, \dots, w_n], \quad w_k = P_k \Gamma.$$

When $s < n$,

$$\text{rank}(W_\Gamma) < n,$$

and we say that we are in an *under- n* case¹. So, in general, a real solution of (1) can be obtained only in the least-square sense. This leads to the following optimisation problem:

$$\min E(Y) = \min \|Y - W_\Gamma C\|^2. \quad (2)$$

By classical results it can be easily shown that the optimal solution of (2) must satisfy the linear system:

$$(W_\Gamma^T W_\Gamma)C = W_\Gamma^T Y.$$

Hence, a generalised real solution of (2) can be obtained in the following way:

$$\min E(Y) = \min \|Y - W_\Gamma (W_\Gamma^T W_\Gamma)^{-1} W_\Gamma^T Y\|^2.$$

Setting:

$$Q_\Gamma = W_\Gamma (W_\Gamma^T W_\Gamma)^{-1} W_\Gamma^T,$$

we are, therefore, lead to find an eigensolution, called *eigenaddress*, of the system:

$$Q_\Gamma Y = Y. \quad (3)$$

Since an eigenaddress Y is a permutation of the assigned address set T , we are interested to an integer solution of (3). Nevertheless, the integer vector Y can be simply determined by a real solution of (3), without round-off errors, by a proper choice of a scaling parameter ψ [11].

Moreover, as $\|T\|_1 = \frac{n(n+1)}{2}$, we can also impose this linear constraint on the entries of the vector Y , finally solving the linear system

$$\tilde{Q}Y = B, \quad \tilde{Q} = \begin{bmatrix} \bar{Q} \\ 1 \dots 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{n(n+1)}{2} \end{bmatrix}, \quad (4)$$

where $\bar{Q} = Q_\Gamma - I$.

3. THE ALGORITHM ELISA

Most of Numerical Analysis is based upon the application of the Fixed Point Theorem [10], which assures the convergence of the iterative procedures by means of a contraction of the distance between successive terms of the sequence approximating the optimal solution. Typically, the minimisation of the error function $E(W)$ of a neural network can be carried out by following the differential scheme:

$$\begin{aligned} \frac{dW}{dt} &= -\gamma \nabla_W E = f(t, W), \\ W(0) &= W_0, \end{aligned} \quad (5)$$

whose discretisation, Euler's approximation of (5), is:

$$\begin{aligned} W_{k+1} &= W_k - \gamma \tau(k) \nabla E_k = \phi(W_k), \\ W(0) &= W_0. \end{aligned} \quad (6)$$

¹ *Under- n* cases may happen even if $s \geq n$, when the matrix W_Γ has no full row-rank.

Formula (6) represents gradient descent's standard iterative method. The convergence of the sequence W_k to a stationary point for $E(W)$ is guaranteed by classical Lipschitz's condition for the problem (5), assuring the existence of a fixed point for the operator $\phi(\cdot)$ in (6). In [14] it was pointed out by Zak that there exist singular solutions of the differential equation (5), named "*terminal attractors*", violating Lipschitz's condition. The main characteristic of these singular solutions lies upon the fact that they can be reached in finite time. So, at least from a theoretical point of view, the computation of the optimal solution could be performed with a finite number of steps. If we choose in (5):

$$\gamma \doteq \frac{\chi(E)}{\|\nabla_W E\|^2},$$

being χ a non-negative continuous function of E , the dynamics of the function $E(W)$ becomes:

$$\frac{dE}{dt} = (\nabla_W E)^T \frac{dW}{dt} = -\chi(E). \quad (7)$$

In this way, the function E is forced to continuously decrease to zero and approaching the optimal solution, which is a terminal attractor.

Therefore, a *terminal attractor algorithm* computes the minimum of $E(W)$ by determining a solution of (5) evaluated by a singular solution of (7).

The problem of solving a linear system

$$AW = b, \quad A \in \mathbb{R}^{n,n}, \quad W, b \in \mathbb{R}^n, \quad (8)$$

can be easily reformulated as an optimisation problem. As a matter of fact, if the linear system admits a solution, it can be discovered by minimising the **error** — *residual* — function

$$E(W) \doteq \frac{1}{2} \frac{\|AW - b\|^2}{\|b\|^2}, \quad \|b\| \neq 0. \quad (9)$$

Hence, we have to solve the following optimisation problem

$$\min_{W \in \Omega} E(W), \quad (10)$$

with $E(W)$ defined in (9). Note that a solution for this problem can be computed even if A is a singular matrix, whereas most numerical methods fail in this circumstance. In [3] it was described a novel terminal attractor algorithm, for solving problem (10), named ELISA, which made use of some special "*non-suspiciousness conditions*". The latter conditions, which can be seen as a generalisation of convexity, are in particular satisfied by problem (10), being the residual error a convex function, and so it can be proved (see [3] for details) that problem (10) is non-suspect. Hence, the following important result holds:

Theorem 1 *Given the quadratic function (9), $\forall \varepsilon_a \in \mathbb{R}^+$, the inequality*

$$\|\nabla E_k\| \geq \frac{1}{k(A)} \frac{\sqrt{2\varepsilon_a} \|A\|}{\|b\|},$$

where $k(A)$ is the condition number of A , is satisfied during the gradient descent, apart from those indexes k such that $E_k \leq \varepsilon_a$.

Remark 1 It can be shown that *establishing the non-suspect nature of the linear system resolution would lead to an iterative linear solver with $\mathcal{O}(n^2)$ as computational complexity*, since this is indeed the computational burden due to the gradient evaluation. Obviously, this is also the lower bound for the problem (8) — $\mathcal{O}(n^2)$ represents also the cost of the data-acquisition phase — and, therefore, the algorithm based on equation (5) would be a candidate optimal algorithm. Theorem 1 states, in fact, that if a sequence of matrices A_n satisfies the condition:

$$k(A_n) \leq x, \forall n,$$

then there exists a linear gradient descent dynamics with $\mathcal{O}(n^2)$ as computational complexity.

3.1. Perfect Hashing with ELISA

Let us now consider the linear system

$$(Q_\Gamma - I)Y = \bar{Q}Y = 0,$$

and the associated minimisation problem

$$\min \bar{E}(Y) = \min \frac{1}{2} \|\bar{Q}Y\|^2, \quad (11)$$

where the square matrix \bar{Q} is n -dimensional but has rank s at most. We can firstly observe that the matrix Q_Γ has a very particular structure, since it is symmetric, positive-semidefinite and has eigenvalues equal to 0 or 1. Therefore, the matrix \bar{Q} is also symmetric, negative-semidefinite, with eigenvalues equal to -1 or 0. By virtue of these properties the following corollary to Theorem 1 may be proved.

Corollary 1 *Given the quadratic function (11), $\forall \varepsilon_a \in \mathbb{R}^+$, the inequality*

$$\|\nabla \bar{E}_k\| \geq \sqrt{2\varepsilon_a}, \quad (12)$$

is satisfied during the gradient descent, apart from those indexes k such that $\bar{E}_k \leq \varepsilon_a$.

Proof Let us suppose that $\bar{E}_k = \frac{1}{2} \|\bar{Q}Y_k\|^2 \geq \varepsilon_a$. Being the matrix \bar{Q} symmetric and negative-semidefinite, we can decompose it as

$$\bar{Q} = U^T D U,$$

where U is a unitary matrix and D is a diagonal matrix, having only -1 or 0 entries on the diagonal. Therefore:

$$\begin{aligned} \|\bar{Q}^T \bar{Q}\| &= \|U^T D^T U U^T D U\| = \|U^T D^T D U\| \\ &= \|-U^T D U\| = \|\bar{Q}\|, \end{aligned}$$

and

$$\|\nabla \bar{E}_k\| = \left\| \left(\bar{Q}^T \bar{Q} \right) Y_k \right\| = \|\bar{Q}Y_k\| \geq \sqrt{2\varepsilon_a}.$$

Remark 2 Using the peculiarities of the matrix Q_Γ , Corollary 1 allows an estimation of the lower bound for the norm of the gradient during optimisation procedure via gradient descent, overcoming the impossibility of calculating the condition number of a singular matrix.

Remark 3 Considering the optimisation problem

$$\hat{E}(Y) = \frac{1}{2} \|\hat{Q}Y - B\|^2,$$

with matrix \hat{Q} and vector B defined as in (4), the equivalence between $\nabla \bar{E}_k$ and $\nabla \hat{E}_k$ may be easily assessed, thus assuring that the estimation (12) still remains valid in the case of the rectangular matrix \hat{Q} .

4. NUMERICAL EXPERIMENTS

In this section we present a statistical analysis of hash solvability. Two examples are reported here, in order to show the performances of ELISA in approaching the perfect hash problem. In both the examples, the error function used is a scaled error function, defined as

$$\begin{aligned} \tilde{E}_s(Y) &= \frac{1}{2} \frac{\|\hat{Q}Y - B\|^2}{R_0^2}, \\ R_0 &= \|\hat{Q}Y_0 - B\|^2. \end{aligned}$$

Therefore $\tilde{E}_s(Y_0) = 0.5$.

Example 1

In [11] was proposed an example in which the key set S is defined as

$$S \doteq \{ABD, ACD, BC, BB\}$$

and $\Gamma = (1, 1, 1)$. Therefore, $n = s = 4$ and $m = 3$. This is a particularly difficult *under- n* case. As a matter of fact, the matrix:

$$W_\Gamma = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}$$

has rank 3; this implies that the matrix $W_\Gamma^T W_\Gamma$ is not invertible. In order to evaluate matrix Q_Γ , we thus use the pseudo-inverse of $W_\Gamma^T W_\Gamma$, nevertheless preserving the peculiarities of Q_Γ . Using a random permutation of the vector T as the starting guess for the solution address vector Y , the probability of finding a MPHf was statistically evaluated (100 runs of the algorithm) as 0.34. In the following Table 1, some cases are shown in which the ELISA-MPHf approach was successfully applied.

$Y_0 = \text{Perm}(T)$	Y	C
3, 2, 1, 4	4, 2, 1, 3	1, 2, 0, 1
4, 2, 1, 3	4, 2, 1, 3	1, 2, -1, 1
1, 3, 4, 2	1, 3, 4, 2	0, 1, 3, 0
4, 1, 2, 3	4, 2, 1, 3	1, 2, 0, 1
2, 1, 3, 4	2, 1, 3, 4	0, 2, 1, 0

Table 1: Successfull MPHf with ELISA.

Example 2

Consider the key set of italian male and female proper names:

$$S \doteq \{\text{MARCO, MONICA, MARIO, MARIA, MARTA, TAMARA, MARA, CORRADO, DARIO, NORMA}\}$$

and choose $\Gamma = (1, \dots, 1) \in \mathbb{R}^7$. Here, $n = 10$, $s = 9$, $m = 7$, and $n > s$ obviously produce an *under- n* problem. In this case, we suppose to relax the minimality condition on the hash function, by requesting that the function h be perfect

only. Therefore, when a solution of the linear system provides a function which is not perfect, the set of addresses is rescaled by a coefficient representing the minimal distance among the components of the computed real solution. In the majority of cases such a post-processing procedure allows to obtain a PHF. Moreover, the computational cost of the post-processing phase is $\mathcal{O}(n^2)$. Results statistically evaluated over 100 runs of ELISA with post-processing, where $Y_0 = Perm(T)$, assess a 0.89 probability of devising a PHF, with about 30% of MPHf, and an averaged loading factor of 58%.

Remark 4 As far as the weight vector choice is concerned, the experimental results clearly show that there is no significant influence of Γ on the statistical distribution previously described. However, the choice of Γ may be crucial in order to obtain full row-rank matrices W_Γ . This becomes statistically more important when n is near s because, in the latter case, the probability of obtaining a W_Γ having maximum rank with a single Γ probe is lower.

5. CONCLUSIONS

This work represents a further contribution to the determination of PHF and MPHf by utilising a direct algebraic approach [11]. Even if, from a computational complexity point of view, alternative and more efficient heuristic algorithms can be fruitfully applied in some cases, ELISA's implementation is, in general, particularly recommended because it requires the minimum effort in terms of non-arithmetic operations and storage. In other words, ELISA for the simplicity of its iterative scheme can be actually competitive with other methods having a lower computational complexity (see [6, 7]). On the other hand, since algorithms for solving linear systems with singular matrices have an interest in themselves, the present application shows that the innovative ideas based on the theory of Terminal Attractors are interesting and deserve further investigations. Extensions of these methods to the resolution of non linear systems are the object of our present research.

References

- [1] M. Bianchini, M. Gori, and M. Maggini, "Does terminal attractor Backpropagation guarantee global optimization?" in *International Conference on Artificial Neural Networks*, vol. 1, (Sorrento), pp. 377-380, Springer Verlag, 1994.
- [2] M. Bianchini, S. Fanelli, M. Gori, and M. Maggini, "Terminal attractor algorithms: A critical analysis," *Neurocomputing*, vol. 15, pp. 3-13, 1997.
- [3] M. Bianchini, S. Fanelli, M. Gori, and M. Protasi, "Solving linear systems by a neural network canonical form of efficient gradient descent," in *Progress in Connectionist-Based Information Systems, ICONIP-ANZIIS-ANNES'97*, vol. 1, (Dunedin), pp. 531-534, Springer Verlag, 1997.
- [4] N. Cercone, M. Krause and J. Boates, "Minimal and almost minimal perfect hash function search with application to natural language lexicon design," *Comput. Math. Appl.*, vol. 9-1, pp. 215-231, 1983.
- [5] R. J. Cichelli, "Perfect hash function made simple," *Communication of ACM*, vol. 23-1, pp. 17-19, 1980.
- [6] Z. J. Czech, B. S. Majewski, "Generating a minimal perfect hashing function in $\mathcal{O}(m^2)$ time," *Archiwum Informatyki Teoretycznej i Stosowanej*, vol. 4, pp. 3-20, 1992.

- [7] Z. J. Czech, B. S. Majewski, "A linear time algorithm for finding minimal perfect hash functions," *Computer Journal*, vol. 36-6, pp. 579-587, 1993.
- [8] E. A. Fox, Q. F. Chen, L. S. Heath, and S. Datta, "A more cost-effective algorithm for finding perfect hash functions," in *Proceedings of the Seventeenth Annual ACM Computer Science Conference*, (Louisville), pp. 114-122, 1989.
- [9] E. A. Fox, Q. F. Chen, A. M. Daoud and L. S. Heath, "Order preserving minimal perfect hash functions and information retrieval," *ACM Transactions on Information Systems*, vol. 9-3, pp. 281-308, 1991.
- [10] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore and London: The Johns Hopkins University Press, 1989.
- [11] M. Gori and G. Soda, "An algebraic approach to Cichelli's perfect hashing," *BIT*, vol. 29-1, pp. 2-13, 1989.
- [12] T. J. Sager, "A polynomial time generator for minimal perfect hash functions," *Communication of ACM*, vol. 28-5, pp. 523-532, 1985.
- [13] S. Wang and C. H. Hsu, "Terminal attractor learning algorithms for Backpropagation neural networks," in *International Joint Conference on Neural Networks*, (Singapore), pp. 183-189, IEEE Press, November 1991.
- [14] M. Zak, "Terminal attractors in neural networks," *Neural Networks*, vol. 2, pp. 259-274, 1989.