

规则低密度校验码的比特翻转解码算法研究^{*})

高敬伯 周曼丽

(华中科技大学电子与信息工程系 武汉 430074)

摘要 本文提出一种规则低密度校验码的比特翻转迭代解码算法。在解码算法的每一次迭代运算过程中,解码运算可以从总体上分为两个阶段:首先,满足可靠性要求的校验节点从与其相邻接的信息节点中选择一个信息比特作为翻转候选比特;然后,解码算法采用投票的方法对于这些候选翻转比特进行进一步的筛选。本算法由于对于最终翻转比特的选择结果是通过两次筛选而得到的,从而极大地降低了误翻的概率,加快了迭代解码算法的收敛速度,提高了系统的性能。另外,在第一阶段的比特选择过程中,我们综合校验节点所提供的校验检测信息和信道输出所提供的可靠性信息,提出了新的翻转比特选择标准。仿真结果表明,本文所提出的解码算法有着较好的性能,在解码运算复杂度和纠错性能之间提供了另外一个均衡。

关键词 纠错码,低密度校验码,比特翻转解码算法

A Two-stage Bit-flipping Decoding Algorithm for Regular LDPC Codes

GAO Jing-Bo ZHOU Man-Li

(Department of Electronics and Information Engineering, Huazhong University of Science & Technology, Wuhan 430074)

Abstract A two-stage bit-flipping decoding algorithm for regular low-density parity-check(LDPC) codes is proposed in this paper. In the first stage, the decoding process is carried on check nodes. Every reliable check node selects one bit-flipping candidate from its neighbor variable nodes. And on the second stage, the decoder refines the selections from the first stage using maximum votes queue algorithm. Since the ultimate flipping bits are got through two selections, the probability of doing wrong flipping is greatly reduced. Therefore, the convergence of the decoding algorithm becomes faster and the performance of the system is improved. Moreover, a new bit-selection criterion which incorporates the syndrome and the absolute value of the log-likelihood ratio of the bit vector is proposed during the first stage of decoding. Simulation results show that our proposed decoding algorithm provides another kind of good tradeoff between error-correcting performance and decoding complexity.

Keywords Error-correcting code, Low-density parity-check code, Bit-flipping decoding algorithm

1 引言

低密度校验码(Low-Density Parity-Check, LDPC)首先是由 Gallager 在 20 世纪 60 年代在他的博士论文中首先提出的^[1,2]。但是在接下来的近 30 年时间里,除了 Tanner 对于 LDPC 码进行过研究以外^[3],LDPC 码由于条件的限制,几乎被编码理论界遗忘了。这种状况一致持续到上个世纪 90 年代末,LDPC 码才由 Neal 与 Mackey 等人重新发现^[4,5]。近几年来,LDPC 码由于其优异的性能和相对较低的解码运算复杂度(能够在相对于码长的线性时间内完成解码操作),使得 LDPC 码成为了理论界和工业界研究的热点之一。在文[6]中,Neal 等人证明了非规则 LDPC 码在采用消息传递(message passing)的软判决解码算法的条件下,可以在信噪比(Signal-to-Noise Ratio)仅仅距 Shannon 限 0.0045dB 的条件下可靠地传输。从总体上讲,LDPC 码的迭代解码算法可以有 3 类^[7]:软判决算法、硬判决算法和混合判决算法。这 3 种解码算法有着不同的纠错性能和解码运算复杂度。由 Gallager 提出的比特翻转(Bit-Flipping, BF)算法是一种硬判决解码算法,这种算法的运算最为简单,其运算复杂度也最低。但是与其它两类解码算法相比,其纠错性能也是最差的。与

之相反,基于置信传播(Belief Propagation, BP)的迭代解码算法是一种软判决解码算法,这种算法的纠错性能是最好的,但是与其优异的性能相对应,BP 算法的运算复杂度在 LDPC 码的各种迭代解码算法中也是最高的。与前两类迭代解码算法不同,加权比特翻转(Weighted Bit-Flipping)算法作为一种典型的混合判决算法为 LDPC 码提供了纠错性能与解码运算复杂度之间的均衡。

本文针对规则 LDPC 码提出了一种新的迭代解码算法。在解码算法的每一次迭代运算中,迭代运算可以分为两个步骤:首先,校验检测失败的校验节点从与其邻接的信息节点中选择一个最有可能错误的信息比特作为翻转候选比特;然后,解码算法再从这些由第一阶段所选出的候选信息比特中进行选择,从而最终确定本次迭代所需要翻转的信息比特。本文所提出的比特翻转算法中,我们对于信息比特进行了两次筛选来确定最终被翻转的比特,从而充分地利用了关于信息比特可靠性的各种信息,大大降低了误翻的可能性,提高了系统的性能。仿真结果表明,与加权 BF 算法相比,本文所提出的算法具有更好的性能。因而,本文所提出的解码算法在解码运算复杂度和纠错性能之间提供了另外一个均衡。

本文第 1 部分为文章的引言,简要介绍有关背景知识和

^{*}国家自然科学基金资助项目(60472015)。高敬伯 博士生,研究方向为信道编码理论、调制技术;周曼丽 教授、博士生导师。

全文概述。在第2部分,首先描述系统模型,然后详细描述本文所提出的比特翻转解码算法,并且对于解码算法的运算复杂度进行简单的分析。第3部分为仿真运算结果。最后总结全文。

2 比特翻转解码算法的设计

2.1 系统模型和基本定义

LDPC码是一种线性分组码,因此LDPC由校验矩阵 H 所唯一确定。校验矩阵 H 为稀疏矩阵,定义校验矩阵 H 中每一列中“1”的个数为列重。同样,定义 H 中每一行中“1”的个数为行重。如果 H 矩阵中的每一列均有相同的列重,而且每一行也均有相同的行重,那么我们称由校验矩阵 H 所确定的LDPC码为规则码。假设校验矩阵 H 含有 M 行、 N 列,一个长度为 N 的比特向量是LDPC码的码字的充分必要条件是这一比特向量满足由 $M \times N$ 维校验矩阵 H 所给出的 $M \geq (N-K)$ 个校验方程。在校验矩阵 H 中,矩阵的第 (m, n) 元素为“1”当且仅当码字中的第 n 个比特参与了第 m 个校验方程的校验运算。同时,LDPC码也可以通过二分图(Tanner图)来表示。此时LDPC码的校验矩阵 H 称为Tanner图的邻接矩阵。在Tanner图中,含有两类不同的节点:对应于码字比特的 N 个信息节点(又称为左节点)和对应于校验方程的 M 个校验节点(又称为右节点)。信息节点 n 与校验节点 m 通过边相连(又称为信息节点 n 与校验节点 m 相邻接)当且仅当Tanner图所对应的校验矩阵的第 (m, n) 元素的值为“1”。

按照文[5]中的表示方法,我们可以定义集合 $M(n) = \{m | H_{nm} = 1\}$ 为与信息节点 n 相邻的所有校验节点的集合。同样,定义集合 $N(m) = \{n | H_{nm} = 1\}$ 为参加第 m 个校验方程的所有信息节点的集合。

假设我们采用 (N, K) 的二进制LDPC码 C 作为纠错码,信道采用均值为0、功率谱密度为 $N_0/2$ 的加性高斯白噪声(Additive White Gaussian Noise, AWGN)信道,调制方式为二进制移相键控(Binary Phase Shift Keying, BPSK),则码字 $v = (v_1, v_2, \dots, v_N)$ 按照方程 $x_n = 2v_n - 1, 1 \leq n \leq N$,映射为取值分别为 ± 1 的序列 $x = (x_1, x_2, \dots, x_N)$ 。假设信道匹配滤波器的输出采样序列为 $r = (r_1, r_2, \dots, r_N)$,则匹配滤波器输出 $y_n, 0 \leq n \leq N$ 的概率密度分布函数为

$$p(r_n | x_n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n - x_n)^2}{2\sigma^2}\right) \quad (1)$$

其中, $\sigma^2 = \frac{1}{2R \cdot (E_b/N_0)}$ 为噪声的方差, R 为LDPC码的码率。假设AWGN信道为对称信道,即 $\Pr(v_n = 1) = \Pr(v_n = 0) = 1/2$,则每一个信道输出 $r_n, n = 1, 2, \dots, N$ 的对数似然比(Log-Likelihood Ratio)为

$$y_n = \log \frac{\Pr(v_n = 0 | y_n)}{\Pr(v_n = 1 | y_n)} = \frac{2}{\sigma^2} r_n \quad (2)$$

那么从信道输出向量 y ,我们可以得到码字比特向量的初始硬判决向量 $z = (z_1, z_2, \dots, z_N)$,其中

$$z_i = \begin{cases} 1 & \text{for } y_i > 0 \\ 0 & \text{for } y_i \leq 0 \end{cases} \quad (3)$$

从而可以得到初始判决向量 z 的纠错伴随式(syndrome) s 为:

$$s = (s_1, s_2, \dots, s_M) = z \cdot H \quad (4)$$

在这里,乘法运算在 $GF(2)$ 域上进行。由LDPC码的性质可

知,判决向量 z 是一个码字的充分必要条件是 $s = 0$ 。如果 $s \neq 0$,那么在判决向量中一定存在着错误的比特。

为了更加简洁地描述算法,我们对于每一个校验节点定义“平均校验可靠度” $l_m, 1 \leq m \leq M$ 为:

$$l_m \triangleq \text{mean}_{z \in N(m)} |y_n| \quad (5)$$

在本文所提出的算法中,我们将综合考虑平均校验可靠度和由伴随式所导出的可靠度来定义出我们翻转比特的选择标准。

2.2 比特翻转解码算法的描述

比特翻转解码算法的基本原理可以简述如下:首先我们可以从信道的输出向量 y 中得到初始的比特硬判决向量 $z^{(0)}$,然后根据某个准则选取最有可能出错的比特进行翻转,从而得到新的比特硬判决向量 $z^{(1)}$ 。对于这一新得到的比特向量 $z^{(1)}$,我们重新计算伴随式 s 用于检测 $z^{(1)}$ 。如果伴随式 $s \neq 0$,从而说明在比特向量 $z^{(1)}$ 中仍然存在着错误的比特。也就是说,比特向量 $z^{(1)}$ 并不是LDPC码 C 的一个有效码字。我们将以比特向量 $b^{(1)}$ 为基础,重新进行比特选择、比特翻转和比特向量校验这一过程。与此相反,如果伴随式 $s = 0$,则说明LDPC码的所有的校验和均为0,从而所有的校验方程都得到了满足。此时,我们就可以认为解码成功,而比特向量 $z^{(1)}$ 就是我们通过解码算法得到的解码码字。在解码运算过程中,我们不断重复这一选择—翻转—验证过程,直到我们最终得到有效的码字(解码成功),或者迭代运算的次数达到了我们预先设定的上限(解码失败)。

在详细地描述本文所提出的比特翻转算法之前,我们必须首先陈述如下事实:如果校验检测 m 失败,那么在所有参加校验检测 m 的信息比特 $N(m)$ 中,一定至少存在一个信息比特 $n \in N(m)$ 为错误比特,需要解码算法翻转。当然,这里需要特别指出的是,上述事实陈述并没有说参加成功的校验检测的信息比特都是正确的。因此,本文所提出的比特翻转算法可以看作是一个不断地寻找满足上述条件的信息比特的过程。

基于如上事实,我们在本文中提出了两阶段的比特翻转算法。在第一阶段,解码算法从校验节点的角度来选择候选翻转比特。也就是说,校验检测失败的校验节点从与其相邻接的信息节点中选择最有可能错误的那个信息比特作为候选翻转比特。这一个阶段可以看作是翻转比特的初选。在接下来的第二阶段,解码算法从信息节点的角度来最终确定翻转比特。解码算法从在第一阶段所选择出的信息比特中再次进行筛选,并将最终选择的信息比特进行翻转,从而完成一次迭代解码运算。从上述的算法描述不难看出,LDPC码的比特翻转解码算法的关键在于尽可能正确地选择错误比特,避免误翻。

因为在解码迭代运算的第一阶段,翻转比特的初选是在校验检测失败的校验节点进行的。此时,我们有两类资源可以利用,帮助我们尽可能正确地选择翻转比特。这两类资源是:信息比特向量的纠错伴随式 s 和每一个信息比特信道输出的对数似然比的绝对值 $|y_n|$ 。在更进一步详细地描述本文所提出的解码算法之前,我们注意到:(1)一个信息比特所参加的失败的校验检测越多,则该信息比特就越有可能是错误比特,从而被翻转;(2)信息比特的对数似然比的绝对值 $|y_n|$ 可以看作是这一信息比特的初始硬判决可靠性的另外一个度量: $|y_n|$ 值越小,则说明该信息比特的初始硬判决就越不可靠。从以上两个事实不难得出,我们应该选择那些参加的失

败的校验检测最多,而且 $|y_n|$ 值最小的信息比特进行翻转。但是,往往所参加失败的校验检测最多的信息比特并不是 $|y_n|$ 值最小的信息比特,因此我们必须综合考虑和权衡这两个因素,从而形成一个新的翻转比特选择标准。在本文中,对于每一个信息节点 $n=1,2,\dots,N$,我们提出了度量值 $\phi_{n,m}^{(k)}$ 为:

$$\phi_{n,m}^{(k)} \triangleq |y_n| - \alpha \cdot s_m^{(k-1)} \cdot L_m \quad (6)$$

其中, m 为与信息节点 n 邻接的校验节点, k 为迭代循环次数。在这里, α 作为权衡参数对于上述两个不同的比特选择标准提供不同的加权。值 α 的选择应该使得解码器有着尽可能好的性能而且尽可能减少解码运算的迭代次数。实际上, $\phi_{n,m}^{(k)}$ 可以看作是校验节点 m 向其它信息节点所传递的关于信息节点 n 可靠性的信息。在信息节点 n ,我们将所有这些由邻接的校验节点发送来的信息相加,从而得到

$$\phi_n^{(k)} \triangleq \sum_{m \in M(n)} \phi_{n,m}^{(k)}, 1 \leq n \leq N \quad (7)$$

作为我们在第一阶段的翻转比特选择标准。

在第一阶段的比特选择运算中,我们可以看出,所选出的候选翻转比特为错误比特的可靠性并不相同。也就是说,有一些校验节点所做出的选择的可靠性要高于另外一些校验节点的选择结果。而在比特翻转算法中,误翻不但会增加迭代运算的次数,而且甚至会导致解码失败,从而降低了系统的性能。因此,这些可靠性低的校验节点所选择的候选翻转比特就不应该在第二阶段的解码算法加以考虑。在这里,我们定义 $f_n, n=1,2,\dots,N$ 为信息节点 n 所参加的失败的校验检测的数目,定义 $f_{\max} = \max\{f_1, f_2, \dots, f_N\}$ 。在这里,我们认为凡是这样的校验节点,与其相邻接的信息节点所参加的失败的校验检测数均小于 f_{\max} ,所做出的候选翻转比特的选择均为不可靠。从而,只有与参加失败的校验检测数为 f_{\max} 的信息节点相邻接的校验节点 $\{m | \forall n \in N(m), f_n < f_{\max}\}$ 才被认为满足可靠性要求,第一阶段的比特选择运算也只是在这些校验节点上进行。因此,只有非常小的一部分校验节点进行第一阶段的比特选择运算,从而使得在第一阶段中的运算复杂度得到了有效的控制。

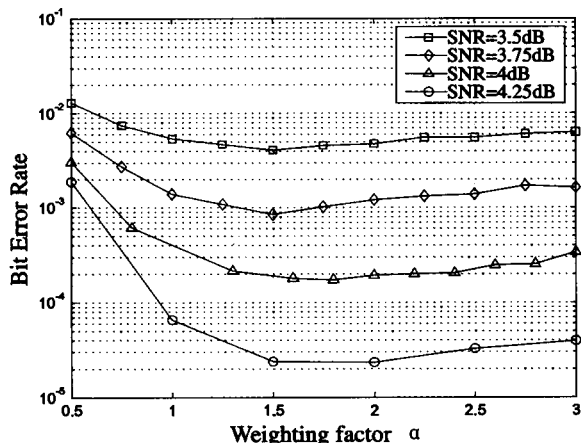


图1 加权参数 α 对于Type-I 2-D (1023, 781) EG-LDPC码误码率的影响曲线

在第一阶段的比特选择算法完成后,每一个被认为可靠的校验节点都将选出一个候选翻转比特。为了更进一步降低误翻的可能性,我们对于从第一阶段所得到的候选翻转比特做进一步的筛选。在这里,我们采用投票的方法。如果一个满足可靠性的校验节点选择一个信息比特作为翻转候选比特,即可以认为是该校验节点给这一信息比特投了一票。最

终,当考虑了所有校验节点的选择后,我们将得票数超过某一个阈值 δ 的所有信息比特认为是错误比特,并将它们放入到集合 S 当中。接下来,我们翻转在集合 S 中的所有信息比特:

$$z_n^{(k)} = z_n^{(k-1)} \oplus 1, \text{ for } z_n^{(k-1)} \in S \quad (8)$$

其中,加法运算是在GF(2)域上进行。在本文中,我们设定阈值 δ 为所有信息比特中得到的最大票数。最后,我们重新计算所得到信息比特向量 $z^{(k)}$ 的伴随式 s ,来确定 $z^{(k)}$ 是否为LDPC码 C 的一个码字。

2.3 运算复杂度分析

在解码算法开始运行时,我们需要在每一个校验节点计算平均校验可靠度。然而,在整个迭代解码运算过程中,这一运算过程仅仅需要计算一次。并且,在运算过程中求数学期望的计算可以进一步简化为寻找信息比特对数似然比的中位数,而这样简化并不会引起性能的明显下降。由于本文所提出的比特翻转算法是一种迭代解码算法,完成解码所需要的迭代次数是一个随机变量,因此很难计算出每次解码运算所需要确切迭代次数。然而,在本文所提出的比特翻转解码算法中,解码算法在每一次迭代过程中将翻转所有被选择的信息比特,而不是例如加权BF算法那样,在每一次迭代中仅仅翻转一个最有可能错误的信息比特。因此,按照本文所提出的算法进行解码运算,其平均迭代次数将会小于按照加权BF算法所需的平均迭代次数。很明显,迭代次数的减少说明解码运算复杂度的降低。在本文的第3部分,我们将通过仿真运算来比较本文所提出的算法与加权BF所需要的平均迭代次数。

3 仿真运算结果

为了证明本文所提出的解码算法的性能,我们选择码率为0.763的Type-I 2-D(1023, 781)的欧式几何(Euclidean geometry, EG) LDPC码,仿真其在不同信噪比条件下的性能。除了本文所提出的解码算法外,我们还将仿真其它3种目前较为常用的解码算法作为比较。这其它的3种解码算法分别是:(1) BP算法^[5]、(2) Gallager的BF算法^[2]、(3) 加权BF算法^[7]。在以上4种算法中,最大迭代次数均设为100。另外,为了确保仿真运算结果的准确性,在每一次仿真运算中均至少检测到100码字错误。

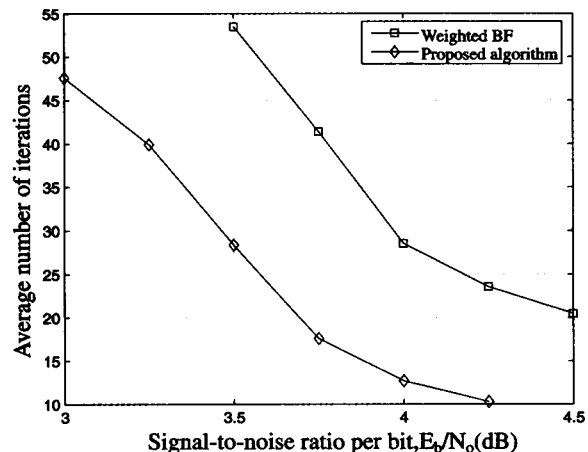


图2 本文所提出的解码算法与加权BF算法的平均迭代次数

对于一个给定的LDPC码,加权参数 α 可以定义为在给定的信噪比条件下,使得解码器取得最优误码率性能的值。对

于不同的 LDPC 码和在不同的信噪比的条件下, α 的最优值往往是不相同的, 而且最优值也只能通过仿真运算得到。图 1 给出了加权参数对于 Type-I 2-D(1023, 781) EG-LDPC 码性能的影响。从仿真结果可以看出, 本文所提出的解码算法的性能随着加权参数 α 的变化而变化, 并且在 α 值的仿真范围内, 存在 α 的最优值。另外, 我们还注意到, 在 α 的某一特定范围内, 解码算法的性能对于 α 的变化并不是十分的敏感。也就是说, 当 α 的取值为这一范围内任意值时, 解码算法仍然可以取得接近最优值的性能。

在图 2 中, 我们给出了本文所提出的解码算法在加权参数 α 取最优值条件下解码 Type-I 2-D(1023, 781) EG-LDPC 所需要的平均迭代次数。另外, 图中还给出了加权 BF 算法所需的平均迭代次数, 作为比较。从图中可以看出, 与加权 BF 算法相比, 本文所提出的算法在不同的信噪比条件下, 均需要较少的迭代运算次数, 从而说明本文所提出的解码算法有着更快的收敛速度。

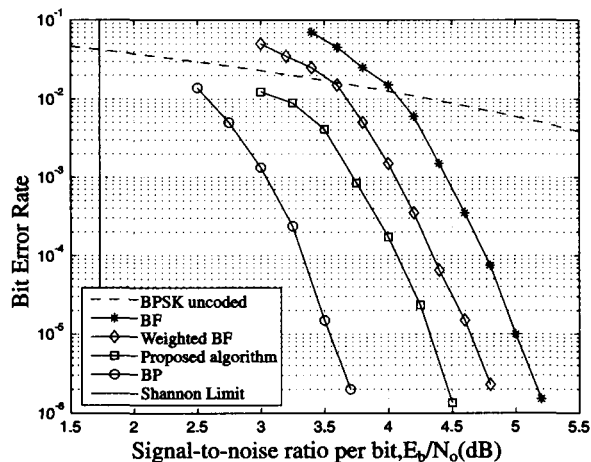


图 3 Type-I 2-D (1023, 781) EG-LDPC 码在不同解码算法下的性能

最后, 我们采用 Monte Carlo 法仿真了当采用这 4 种不同的解码算法时 Type-I 2-D(1023, 781) EG-LDPC 码在 AWGN 信道下的性能, 如图 3 所示。同时, 在图中我们还标出了没有采用纠错码而仅采用 BPSK 调制时系统的性能和 AWGN 信道的 Shannon 限作为研究系统性能的参考。从仿真结果可以看出, 当误码率 BER $< 10^{-5}$ 时, 本文所提出的算

法与加权 BF 算法相比, 额外提供了 0.35 dB 的编码增益。如果与 Gallager 提出的 BF 算法相比, 则额外提供了 1.4 dB 的编码增益。而且, 本文所提出的算法与标准的置信传播算法相比, 仅仅有 0.7 dB 的距离。从而说明, 本文所提出的解码算法作为一种比特翻转算法, 具有良好的性能, 在解码复杂度与纠错性能两者之间提供了一种很好的均衡。

结论 本文对于规则 LDPC 码提出了一种两阶段的比特翻转解码算法。在解码算法的每一次迭代过程中, 解码算法的翻转比特选择过程可以分为两个阶段。首先, 解码算法从校验节点的角度选择候选翻转比特。所有满足可靠性要求的校验节点从与它们邻接的信息节点中各自选择一个最有可能错误的信息比特作为候选翻转比特。在第一阶段的选择过程中, 我们综合考虑信息比特向量的纠错伴随和对数似然比, 提出了新的翻转比特选择标准。另外, 由于只有小部分校验节点进行比特选择运算, 因此校验节点选择运算的运算复杂度不会太高。然后, 解码算法对于从第一阶段所选择的候选翻转比特进一步细化, 从而进一步降低了误翻的概率。具体而言, 解码算法对于候选翻转比特采用投票的方法, 从中选择出最有可能是错误的信息比特最为本次迭代运算最终的选择结果。仿真结果表明, 本文所提出的解码算法在解码运算复杂度和纠错性能之间提供了一种很好的均衡, 为工程人员在设计通信系统时提供了另外一个选择。

参考文献

- Gallager R G. Low density parity check codes. IRE Trans Info Theory, 1962, IT-8: 21~28
- . Low Density Parity Check Codes. Research monograph series. Cambridge, Mass.: MIT Press, 1963
- Tanner R M. A recursive approach to low complexity codes. IEEE Transactions on Information Theory, 1981, 27(5): 533~547
- MacKay D J C, Neal R M. Near Shannon limit performance of low density parity check codes. Electronics Letters, 1996, 32(18): 1645~1646
- MacKay D J C. Good error correcting codes based on very sparse matrices. IEEE Transactions on Information Theory, 1999, 45(2): 399~431
- Chung S Y, Forney G D, Richardson T, et al. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. IEEE Commun Lett Feb. 2001, 5: 58~60
- Kou Y, Lin S, Fossorier M P C. Low density parity check codes based on finite geometries: A rediscovery and new results. IEEE Transactions on Information Theory, 2001. 2711~2736

(上接第 63 页)

以期进一步提高预测精度。

参考文献

- Foster I, Kesselman C. The Grid; Blueprint for a New Computing Infrastructure. Morgan-Kaufmann, 1999
- Yang L, Foster I, Schopf J. Homeostatic and Tendency-based CPU Load Predictions. In: Proc. of IPDPS 2003, April 2003
- Wolski R, Spring N, Hayes J. The Network Weather Service: A Distributed Resource Performance Forecasting Service for Meta-computing. Journal of Future Generation Computing Systems, 1998
- Leland W, Taqqu M, Willinger W, Wilson D. On the self-similar nature of Ethernet traffic. IEEE/ACM Transactions on Networking, Feb. 1994
- Croyelia M, Bestavros A. Self-similarity in world wide web traffic: Evidence and possible causes. IEEE/ACM
- Park K, Kim G, Crovella M. On the effect of traffic self-similarity on network performance. In: Proc. of the SPIE International

Conference on Performance and Control of Network Systems, Nov. 1997

- Bernan M, Serman R, Taqqu M. Long-range dependence in variable-bit-rate video traffic. IEEE Transactions on Communications, March 1995
- Gribble S, Mandu S, Roselli D. Brewer Self-similarity in file systems. available from <http://www.cs.berkeley.edu/~gribble>
- Dinda P, O'Halloran D. The statistical properties of host load. In to appear in the Fourth Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers(LCR98) and CMU Tech, report CMU-CS-98-143, 1998, available from <http://reports-archive.adm.cs.cmu.edu/anon/1998/CMU-CS-98-143>, ps
- Barnsley M. Fractals Everywhere, Academic Press Inc, 1998
- Wolski R, Spring N, Hayes J. Predicting the CPU availability of Time-shared Unix Systems. In: Proc. of 8th IEEE High Performance Distributed Computing Conference (HPDC8), 1999
- 陈凌, 等编著. 分形几何学. 北京: 地震出版社, 1998
- 唐立春, 李光熹, 熊曼丽. 基于分形的电力系统负荷预测. 电力系统及其自动化学报, 1999, 11(4)
- 梁平, 樊福梅, 吕玉坤. 电力系统负荷分形预测及 R/S 分析. 华北电力大学学报, 2004, 31(4)