软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(5):915−932 [doi: 10.3724/SP.J.1001.2013.04381]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

子网树求解一般间隙和长度约束严格模式匹配*

武优西[1], 刘亚伟[1], 郭 磊[2], 吴信东[3,4]

[1](河北工业大学 计算机科学与软件学院,天津 300401)

[2](河北工业大学 电磁场与电器可靠性省部共建重点实验室,天津 300130)

[3](合肥工业大学 计算机科学与信息工程学院,安徽 合肥 230009)

[4](Department of Computer Science, University of Vermont, Burlington, USA)

通讯作者: 武优西, E-mail: wuc@scse.hebut.edu.cn, http://www.scse.hebut.edu.cn

摘 要: 具有通配符间隙约束的模式匹配问题在信息检索、计算生物学和序列模式挖掘等研究领域有重要的应用.提出了更一般性的模式匹配问题,即一般间隙和长度约束的严格模式匹配(strict pattern matching with general gaps and length constraints,简称 SPANGLO).该问题具有如下 4 个特点:它是一种严格的精确模式匹配;允许序列中任意位置的字符被多次使用;模式串中可以包含多个一般间隙;对出现的总体长度进行了约束.最坏情况下,一个 SPANGLO 实例将转换出指数个非负间隙的严格模式匹配实例.为了有效地解决该问题,提出了子网树及其相关概念和性质.在此基础上提出了求解算法 SubnettreeSpanglo(SETS),并给出算法的正确性和完备性证明,同时指出该算法的空间复杂度与时间复杂度分别为 $O(m\times MaxLen\times W)$和 $O(MaxLen\times W\times m^2\times n)$,其中,$m$,$n$,$MaxLen$ 和 W 分别是模式和序列的长度、出现的最大长度约束和模式的最大间距.实验结果既验证了 SPANGLO 问题转换方法的正确性,又验证了该算法的正确性和有效性.

关键词: 模式匹配;一般间隙;长度约束;子网树

中图法分类号: TP301 **文献标识码**: A

中文引用格式: 武优西,刘亚伟,郭磊,吴信东.子网树求解一般间隙和长度约束严格模式匹配.软件学报,2013,24(5):915−932. http://www.jos.org.cn/1000-9825/4381.htm

英文引用格式: Wu YX, Liu YW, Guo L, Wu XD. Subnettrees for strict pattern matching with general gaps and length constraints. Ruan Jian Xue Bao/Journal of Software, 2013,24(5):915−932 (in Chinese). http://www.jos.org.cn/1000-9825/4381.htm

Subnettrees for Strict Pattern Matching with General Gaps and Length Constraints

WU You-Xi[1], LIU Ya-Wei[1], GUO Lei[2], WU Xin-Dong[3,4]

[1](School of Computer Science and Software, Hebei University of Technology, Tianjin 300401, China)

[2](Province-Ministry Joint Key Laboratory of Electromagnetic Field and Electrical Apparatus Reliability, Hebei University of Technology, Tianjin 300401, China)

[3](School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China)

[4](Department of Computer Science, University of Vermont, Burlington, USA)

Corresponding author: WU You-Xi, E-mail: wuc@scse.hebut.edu.cn, http://www.scse.hebut.edu.cn

Abstract: Pattern matching with gap constraints has important applications in many fields such as information retrieval, computational biology, and sequential pattern mining etc. This paper proposes a pattern matching problem named Strict Pattern Matching with General Gaps and Length Constraints (SPANGLO) which has four characteristics: It is strict exact pattern matching; Any position in the given

* 基金项目: 国家重点基础研究发展计划(973)(2013CB329604); 国家自然科学基金(61229301); 国家高技术研究发展计划(863)(2012AA011005); 河北省自然科学基金(H2012202035, F2013202138); 河北省教育厅重点项目(ZH2012038)

收稿时间: 2012-09-13; 定稿时间: 2013-02-05

sequence can be used more than once; The pattern can have more than one general gap constraint; Each matching substring has length constraints. An instance of SPANGLO can be transformed into an exponential number of matching instances with non-negative gaps in the worst case. In order to solve the problem effectively, an algorithm named SubnettreeSpanglo (SETS) is proposed based on Subnettree and its special concepts and properties. The correctness and completeness of the algorithm are given, and the space and time complexities of the algorithm are $O(m\times MaxLen\times W)$ and $O(MaxLen\times W\times m^2\times n)$ respectively, where n, m, $MaxLen$ and W are the sequence length, the pattern length, the maximum length of the occurrence and the maximum gap of the pattern respectively. Experimental results validate the correctness of the transforming method of SPANGLO and the efficientiveness and correctness of SETS.

Key words: pattern matching; general gap; length constraint; Subnettree

模式匹配(串匹配)是计算机科学的基本问题之一,也是经典问题之一.Fischer 和 Paterson[1]最早对模式串中具有单一通配符的模式匹配问题进行了研究,但其研究中通配符所能通配的字符个数是常数.虽然 Manber 等人[2]的研究依然是单一通配符,但所通配的字符个数是一个范围值,这种通配符称为具有间隙约束的通配符,对应的模式匹配称为具有间隙约束的模式匹配.近年来,研究者们更加致力于具有多个间隙约束的模式匹配问题研究,这类问题的模式串可以描述为 $P=p_0[min_0,max_0]p_1...[min_{j-1},max_{j-1}]p_j...[min_{m-2},max_{m-2}]p_{m-1}$[3],这里,$min_{j-1}$,$max_{j-1}$ 分别指在 p_{j-1} 和 p_j 之间可通配的最小和最大间隙.具有多个间隙约束的模式匹配问题在诸多领域具有重要应用.在生物计算领域,Navarro 和 Raffinot[4]在借鉴反向搜索的基础上提出了更为优化的搜索方法并应用于蛋白质查找;Wang 等人[5]对模式串中有重复的模式子串问题进行研究并在 DNA 序列上验证了方法的有效性;在文本匹配方面,Cole 等人[6]在近似模式匹配下有效地判断模式串是否在指定的文本或字典中;Crochemore 等人[7]提出了(δ,α)-近似匹配并指出这种匹配方法可以应用于音乐信息检索等领域;Cantone 等人[8]提出了用于音乐信息检索和分析的位并行方法;在序列模式挖掘方面,Ji 等人[9]提出了 ConSGapMine 算法挖掘具有最小识别序列的模式;Ferreira 和 Azevedo[10]提出了 gIL 算法用于蛋白质序列挖掘.

本文对一般间隙和长度约束的模式匹配问题(pattern matching with general gaps and length constraints,简称 SPANGLO)进行研究.该问题具有如下 4 个特点:它是一种严格的精确模式匹配;允许序列中任意位置的字符被多次使用;模式串中可以包含多个负间隙;对出现的总体长度进行了约束.对这些特点进行举例说明如下:

例 1:给定序列串 $S=s_0s_1s_2s_3s_4s_5$=aabbcc 和模式串 $P=p_0[min_0,max_0]p_1[min_1,max_1]p_2$=a[0,1]b[0,1]c.

子模式“a[0,1]b”的含义是,在“a”和“b”之间通配符可以通配 0 或 1 个字符.按照文献[11]的计算方法,〈0,2,4〉是一个合法的出现,这是因为 $p_0=s_0$=“a”,$p_1=s_2$=“b”,$p_2=s_4$=“c”且 $min_0\leqslant 2-0-1\leqslant max_0$ 和 $min_1\leqslant 4-2-1\leqslant max_1$.尽管 $p_1=s_3$=“b”,但是〈0,3,4〉不是一个合法的出现,原因是 3−0−1=2 不能满足[0,1]间隙约束.易知按照文献[11]的计算方法,共有 4 个出现,即〈0,2,4〉,〈1,2,4〉,〈1,3,4〉和〈1,3,5〉.但是按照文献[4,6−8]的计算方法,例 1 的出现数为 2 分别为 4 和 5,这是因为 p_2 在序列串的位置 4 和 5 上均可以产生出现.显然存在两种出现的描述方式.我们称前者为严格模式匹配.它采用一组值来描述一个出现.这组值是模式串中对应字符在序列串中的位置.只要这一组值中有 1 个值发生变化即视为一个新的出现.这会导致组合爆炸现象的发生,其解空间大小为 $O(n\times w^m)$,这里,n,w 和 m 分别是序列串长度、间隙宽度和模式串长度.后者称为宽松模式匹配.它采用模式串最后一个字符在序列串中的位置值来表示这个出现,因此,其解空间大小是 $O(n)$.

如果模式串所有最小间隙均大于或等于 0,则模式串属于非负间隙模式串.显然,例 1 的模式串属于非负间隙模式串;如果有一个最小间隙或最大间隙小于 0,则这样的间隙称为负间隙.包含有负间隙的模式串称为一般间隙模式串,如模式串 $P1$=a[−1,1]b[0,1]c 可称为一般间隙模式串.

如果一个出现中 p_j 与 s_i 必须相同则称为精确模式匹配.与精确模式匹配对应的是近似模式匹配[12].近似模式匹配依据距离计算公式的不同,可以分为 Hamming 距离下近似模式匹配、编辑距离下近似模式匹配和δ近似模式匹配等.若 p_j 与 s_i 相同,则距离为 0;否则,距离为 1.这种方式是 Hamming 距离下近似模式匹配.例 1 中,〈1,2,3〉在严格精确模式匹配下不是一个合法的出现,因为 s_3=“b”≠p_2=“c”,但在 Hamming 距离为 1 的近似模式匹配下则是一个合法的出现,因为序列子串“abb”与模式串“abc”的 Hamming 距离为 1 且满足间隙约束.一个字符串经过最少的替换、插入或删除等操作转换为另外一个字符串,这种操作次数称为两个字符串的编辑距离,又称为

Levenshtein 距离.在音乐信息检索领域,更为常用的是δ近似模式匹配[7,8],δ近似是指 $s_i=\delta p_j$ 当且仅当$|s_i-p_j|\leqslant\delta$.在δ近似模式匹配中也需满足间隙约束.

在⟨0,2,4⟩和⟨1,2,4⟩两个出现中,位置 2 和位置 4 均被两次使用.若任何一个位置只能在所有出现中最多使用 1 次则称为一次性条件[13,14]约束.在一次性条件约束下,例 1 最多的出现为⟨0,2,4⟩和⟨1,3,5⟩.此外,例 1 没有对出现的总体长度进行约束[13],它是指出现中最大位置与最小位置的差需要满足一定约束.如果约定例 1 的最小和最大长度均为 4,则只有⟨1,2,4⟩和⟨1,3,4⟩是合法的出现,因为 4−1+1=4,满足长度约束;而⟨0,2,4⟩和⟨1,3,5⟩的长度为 5,不满足长度约束,均属于不合法出现.本文主要贡献在于:

- 提出了具有一般间隙和长度约束的严格模式匹配问题 SPANGLO.在该模式匹配中,不但模式串允许含有多个负间隙,而且对出现的总体长度进行了约束,是一种允许序列中任意位置的字符被多次使用的严格精确模式匹配.
- 提出了对一般间隙模式串进行分解的两种特殊运算,即并且运算和或运算,实现了由一个 SPANGLO 实例等价转换为多个具有非负间隙和长度约束的严格模式匹配实例的方法,并实验验证了这种转换方法的正确性.但这样的转换方法过于复杂,且最坏情况下,一个 SPANGLO 实例可以产生出指数个非负间隙模式匹配实例.
- 提出了子网树结构.通过子网树结构来确定出现中的最大值,进而对长度约束进行处理.利用子网树的创建规则控制了模式匹配过程中多次往复回溯的范围,降低问题控制难度.在此基础上,在子网树上构造一些新的概念与性质,以便对 SPANGLO 问题进行求解.
- 提出了 SubnEtTreeSpanglo(SETS)算法,并给出了算法的正确性和完备性证明,同时指出该算法的空间复杂度和时间复杂度分别为 $O(m\times MaxLen\times W)$和 $O(MaxLen\times W\times m^2\times n)$.这里,$n$,$m$,$MaxLen$ 和 W 分别是序列 S 和模式 P 的长度、出现的最大长度约束和模式 P 的最大间距.

本文第 1 节对相关的研究工作进行综述.第 2 节对 SPANGLO 问题进行定义,理论分析该问题与非负间隙的严格模式匹配问题之间的转换关系.第 3 节给出子网树定义、相关性质以及本文所有主要符号及其说明.第 4 节提出问题的求解算法,在分析算法复杂性的同时,理论证明 SETS 算法的正确性和完备性;之后,通过运行实例来阐明算法的工作原理.第 5 节通过真实生物数据来验证 SETS 算法的正确性和性能.第 6 节得出本文结论.

1 相关工作

模式匹配是模式挖掘技术的核心与基础,在序列模式挖掘中具有广泛应用.如,Li 等人[15]提出有效算法挖掘具有间隙约束的闭合频繁模式,并应用于分类或聚类的特征选择问题;Zhang 等人[16]提出有效算法挖掘伴随关系(FCIP)模式,用于购买模式挖掘.而具有间隙约束的序列模式挖掘也逐渐引起人们的注意.如,Zhang 等人[17]采用 MPP 算法实现了具有周期间隙约束的序列模式挖掘,并将该方法应用于 DNA 序列挖掘;Tanbeer 等人[18]在流数据中挖掘周期间隙约束的模式;Li 等人[19]对周期运动行为进行挖掘;Zhu 和 Wu[20]提出了基于 GSP 算法的模式挖掘算法,进一步有效地解决了文献[17]的问题.而 GSP 算法是一个具有间隙约束的模式匹配算法,这充分地说明了具有间隙约束的模式匹配算法在序列模式挖掘中发挥着重要作用.

通过例 1 的说明可知,目前具有间隙约束的模式匹配问题可从以下 5 个方面进行划分:宽松模式匹配还是严格模式匹配、一般间隙模式串与否、精确匹配与否、是否具有最多使用 1 次的限制、是否具有长度约束.

在音乐信息检索中,大多使用宽松模式匹配[7,8];在生物信息计算中,大多采用严格模式匹配[11,13].在严格模式匹配下,由于任何位置字符可多次使用,因此模式串在序列中的出现数与模式串的长度呈指数形式增长.在此情况下,人们更关注出现数,而较少关心具体出现是什么,这一特点在序列模式挖掘中得到充分的展示[17].

无论是否具有一次性条件约束的模式匹配,均具有重要的实际应用意义.Min 等人[11]开展的研究就是不具有一次性条件的模式匹配;在文献[17,20]的周期间隙约束的序列模式挖掘中,也允许 1 个位置在多个出现中重复使用.Chen 等人[13]开展的研究就是具有一次性条件的模式匹配,并且在序列模式挖掘中,Ferreira 和 Azevedo[10]、Huang 等人[21]和 Ding 等人[22]的研究都是基于一次性条件的序列模式挖掘.此外,长度约束不仅在

模式匹配[11,13]中有应用,而且文献[9,10]的序列模式挖掘中均使用了长度约束.

近年来,随着对具有间隙约束的模式匹配问题的深入研究,人们逐渐认识到一般间隙模式匹配问题的重要性.Myers[23]最早对一般间隙模式匹配问题进行了研究;Navarro 和 Raffinot[4]指出,更加一般化且求解难度更大的模式匹配研究是一般间隙模式匹配;Fredriksson 和 Grabowski 先后在 2006 年[24]和 2008 年[25]对一般间隙的模式匹配问题进行了研究,并在音乐信息检索和蛋白质序列匹配等问题中进行了应用.但上述一般间隙模式匹配研究均属于宽松模式匹配.表 1 给出了几种具有间隙约束的模式匹配对比.

Table 1 Comparison of pattern matching with gap constraints

表 1 几种具有间隙约束的模式匹配对比

文献	匹配类型	间隙类型	间隙个数及类型	匹配类型	一次性与否	长度约束
Manbe 和 Baeza-Yates[2]	严格匹配	非负间隙	一个可变间隙	精确匹配	否	无
Akutsu[26]	宽松匹配	非负间隙	一个可变间隙	编辑距离下近似	–**	–
Bille 等人[27]	宽松匹配	非负间隙	多个可变间隙	精确匹配	–	–
Rahman 等人[28]	宽松匹配	非负间隙	多个可变间隙	精确匹配	–	–
Fredriksson 和 Grabowski[24,25]	宽松匹配	一般间隙	多个可变间隙	δ近似	否	无
Chen 等人[13]	严格匹配	非负间隙	多个可变间隙	精确匹配	是	有
He 等人[29]	严格匹配	非负间隙	多个可变间隙	Hamming 距离下近似	是	有
Min 等人[11]	严格匹配	非负间隙	多个可变间隙	精确匹配	否	有
本文	严格匹配	一般间隙	多个可变间隙	精确匹配	否	有

** 在宽松模式匹配下通常不考虑一次性条件和长度约束问题,因此,表中以“–”表示不作考虑.

通过表 1 可以看出,本文研究工作与文献[11]的研究工作最为接近,其区别在于,文献[11]对非负间隙的严格模式匹配进行研究,而本文是间隙可为负的一般间隙模式匹配,是更一般性的研究.本文的研究不但求解难度更大,而且更具有实际意义.在非负间隙作用下,模式子串 p_{j+1} 对应在序列串中的位置只能大于 p_j 对应在序列串中的位置,因此在查找一个出现的过程中,仅仅需要从左向右(从前向后)单向扫描即可;而在负间隙的作用下,这种关系并不成立,因而在查找一个出现的过程中不能单向扫描,可能需要回溯负间隙个字符,这样,在整体的模式匹配过程中可能存在多次往复的回溯过程.此外,在非负间隙作用下,出现中最小和最大值分别是出现的第 1 个值和最后一个值,这样在处理长度约束时,易于控制并处理;而在负间隙的作用下,出现中最小和最大值的位置可能是出现中的任何一个位置,更为极端的情况是出现中最小值在最后,而最大值则在开始.综上,本文的研究工作问题难点在于匹配过程中有多次往复回溯现象且长度约束难于处理,因此该问题的求解难度更大.在应用方面,基于消费者的购买模式的相似性,可以进行购买模式的挖掘,但这种序列模式挖掘在非负间隙作用下就约定了消费者的购买次序.然而,不同消费者之间很难具有相同的购买次序,因而采用一般间隙的序列模式挖掘将有助于发现更多有价值的模式.而如前分析可知,具有间隙约束的序列模式挖掘的核心与基础是具有间隙约束的模式匹配.综上,一般间隙的模式匹配比非负间隙的模式匹配具有更大的求解难度和实际意义.

2 SPANGLO 问题

2.1 问题定义

定义 1. 序列串[24,25] $S=s_0s_1...s_i...s_{n-1}$,这里,n 表示 S 的长度,$s_i\in\Sigma$代表一种符号集.对于不同的应用,Σ可以是不同的符号集合,例如,在 DNA 序列中,Σ是由{A,T,C,G}构成的;在音乐信息检索中,Σ是由数字构成的.

定义 2. 具有一般间隙约束的模式串 P[24,25]可以表示为 $p_0[min_0,max_0]p_1...[min_{j-1},max_{j-1}]p_j...[min_{m-2},max_{m-2}]p_{m-1}$,这里,$m$ 表示 P 的长度;$p_j\in\Sigma$,min_{j-1} 和 max_{j-1} 是给定整数值,代表模式字符 p_{j-1} 和 p_j 之间通配符可以匹配的最小和最大间隙长度.这里,$min_{j-1}\leqslant max_{j-1}$ 且 min_{j-1} 和 max_{j-1} 称为局部约束且均可以为负值.

定义 3. 如果一个位置索引序列 $I=\langle i_0,...,i_j,...,i_{m-1}\rangle$服从如下约束条件:

$$s_{i_j}=p_j \tag{1}$$

$$i_{j-1}\neq i_j \tag{2}$$

$$\begin{cases}\min_{j-1} \leqslant i_j - i_{j-1} - 1 \leqslant \max_{j-1}, & \text{if } i_{j-1} < i_j \\ \min_{j-1} \leqslant i_j - i_{j-1} \leqslant \max_{j-1}, & \text{if } i_{j-1} > i_j \end{cases} \tag{3}$$

其中,$0 \leqslant j \leqslant m-1$ 且 $0 \leqslant i_j \leqslant n-1$,则称 I 是 P 在 S 中的一个出现.

定义 4. 模式 P 在序列 S 中的一个出现 I 满足长度约束[13]是指服从如下约束条件:

$$MinLen \leqslant i_{\max} - i_{\min} + 1 \leqslant MaxLen \tag{4}$$

其中,$MinLen$ 和 $MaxLen$ 分别是出现的最小长度和最大长度约束.因此,长度约束 LEN 是由 $MinLen$ 和 $MaxLen$ 两个整数值所构成的.此外,$i_{\max}=\max(i_0,...,i_j,...,i_{m-1})$,$i_{\min}=\min(i_0,...,i_j,...,i_{m-1})$.

定义 5. 令集合 $T(S,P)$表示模式串 P 在序列串 S 中的所有出现,其长度用$|T(S,P)|$来表示.令集合 $T(S,P,LEN)$表示模式串 P 在序列串 S 中满足长度约束 LEN 的所有出现,其长度用$|T(S,P,LEN)|$来表示.SPANGLO 是指计算$|T(S,P,LEN)|$的值.

例 2:给定模式串 P=a[0,1]b[−1,1]c,序列串 S=acbacbc 以及最小和最大长度分别为 $MinLen$=3,$MaxLen$=4.

当没有长度约束时,模式串 P 在序列串 S 中所有出现 $T(S,P)$是{⟨0,2,1⟩,⟨0,2,4⟩,⟨3,5,4⟩,⟨3,5,6⟩},此时,$|T(S,P)|$为 4;当给定长度约束 $MinLen$=3 和 $MaxLen$=4 时,所有出现为{⟨0,2,1⟩,⟨3,5,4⟩,⟨3,5,6⟩},则$|T(S,P,LEN)|$为 3.

通过例 2 可知,SPANGLO 问题的难度不仅是字符“c”可以在字符“b”之前,而是更难处理的长度约束.如果在非负间隙下出现的最大值和最小值分别在 i_2 和 i_0,此时易于处理长度约束.在 SPANGLO 问题中出现的最大值和最小值的位置不固定,如,出现⟨3,5,6⟩和⟨3,5,4⟩的最大值分别在 i_2 和 i_1.此外,某些位置的字符既可以与其他位置的字符构成满足长度约束的出现,也可以与其他位置的字符构成不满足长度约束的出现,如位置 2 的字符“b”与位置 0 和位置 1 的字符“a”和“c”一起构成了满足 LEN 的出现⟨0,2,1⟩;同时,位置 2 的字符“b”与位置 0 和位置 4 的字符“a”和“c”一起又构成了不满足 LEN 的出现⟨0,2,4⟩.而在求解该问题过程中又不能对每个可能的候选解进行逐一判断,这导致了该问题求解难度的增加.

2.2 理论分析

定义 6. 将一个一般间隙模式串 $P=p_0[min_0,max_0]p_1...[min_{j-1},max_{j-1}]p_j...[min_{m-2},max_{m-2}]p_{m-1}$ 等价地转换为多个非负间隙模式串 $Q_1,Q_2,...,Q_k$ 是指 $|T(S,P,LEN)|=\sum_{i=1}^{k}|T(S,Q_i,LEN)|$,可以记为 $P \Leftrightarrow Q_1|Q_2|...|Q_k$,其中,“⇔”代表等价于.

为了解决这个问题,我们引入两种运算:“并且运算(&)”和“或运算(|)”.

定义 7. 并且运算(&)可以将模式 P 分解为两个模式子串的并且,即 $P=p_0[min_0,max_0]p_1...[min_{j-1},max_{j-1}]p_j...[min_{m-2},max_{m-2}]p_{m-1} \Leftrightarrow (p_0[min_0,max_0]p_1...[min_{j-1},max_{j-1}]p_j)\&(p_j...[min_{m-2},max_{m-2}]p_{m-1})$.其中,$0<j<m-1$,$m$ 为模式 P 的长度.

定义 8. 或运算(|)可以将某个间隙 $p_{j-1}[min_{j-1},max_{j-1}]p_j$ 分解为两个间隙的或,即 $p_{j-1}[min_{j-1},max_{j-1}]p_j \Leftrightarrow (p_{j-1}[min_{j-1},a]p_j)|(p_{j-1}[a+1,max_{j-1}]p_j)$,其中,$0<j<m$,$min_{j-1}<a<max_{j-1}$,$m$ 为模式 P 的长度.

为了将模式串 P 转为多个等价非负间隙模式,我们对单个间隙 $p_{j-1}[min_{j-1},max_{j-1}]p_j$ 进行详细讨论.显然,间隙 $p_{j-1}[min_{j-1},max_{j-1}]p_j$ 存在 3 种形式,分别是:

(i) $0 \leqslant min_{j-1}$,此时无须做任何变换;

(ii) $max_{j-1}<0$,此时,等价变换为 $p_j[-1-max_{j-1},-1-min_{j-1}]p_{j-1}$;

(iii) $min_{j-1}<0<max_{j-1}$,此时,等价变换为

$$(p_{j-1}[min_{j-1},-1]p_j)|(p_{j-1}[0,max_{j-1}]p_j) \Leftrightarrow (p_j[0,-1-min_{j-1}]p_{j-1})|(p_{j-1}[0,max_{j-1}]p_j).$$

因此,$p_{j-1}[min_{j-1},max_{j-1}]p_j[min_j,max_j]p_{j+1}$ 变换为非负间隙模式将存在如表 2 所示的 9 种情况.

- 情况(1)为$(p_{j-1}[min_{j-1},max_{j-1}]p_j)\&(p_j[min_j,max_j]p_{j+1})$;
- 情况(2)为$(p_{j-1}[min_{j-1},max_{j-1}]p_j)\&(p_{j+1}[-1-max_j,-1-min_j]p_j)$;
- 情况(3)为$(p_{j-1}[min_{j-1},max_{j-1}]p_j)\&((p_{j+1}[0,-1-min_j]p_j)|(p_j[0,max_j]p_{j+1}))$;
- 情况(4)为$(p_j[-1-max_{j-1},-1-min_{j-1}]p_{j-1})\&(p_j[min_j,max_j]p_{j+1})$;

- 情况(5)为($p_j[-1-max_{j-1},-1-min_{j-1}]p_{j-1}$)&($p_{j+1}[-1-max_j,-1-min_j]p_j$);
- 情况(6)为($p_j[-1-max_{j-1},-1-min_{j-1}]p_{j-1}$)&(($p_{j+1}[0,-1-min_j]p_j$)|($p_j[0,max_j]p_{j+1}$));
- 情况(7)为(($p_j[0,-1-min_{j-1}]p_{j-1}$)|($p_{j-1}[0,max_{j-1}]p_j$))&($p_j[min_j,max_j]p_{j+1}$);
- 情况(8)为(($p_j[0,-1-min_{j-1}]p_{j-1}$)|($p_{j-1}[0,max_{j-1}]p_j$))&($p_{j+1}[-1-max_j,-1-min_j]p_j$);
- 情况(9)为(($p_j[0,-1-min_{j-1}]p_{j-1}$)|($p_{j-1}[0,max_{j-1}]p_j$))&(($p_{j+1}[0,-1-min_j]p_j$)|($p_j[0,max_j]p_{j+1}$)).

Table 2 9 cases of transformation $p_{j-1}[min_{j-1},max_{j-1}]p_j[min_j,max_j]p_{j+1}$ into non-negative gap constraints patterns

表 2 $p_{j-1}[min_{j-1},max_{j-1}]p_j[min_j,max_j]p_{j+1}$ 变换为非负间隙模式的 9 种情况

	$p_j[min_j,max_j]p_{j+1}$ 属(i)	$p_j[min_j,max_j]p_{j+1}$ 属(ii)	$p_j[min_j,max_j]p_{j+1}$ 属(iii)
$p_{j-1}[min_{j-1},max_{j-1}]p_j$ 属(i)	情况(1)	情况(2)	情况(3)
$p_{j-1}[min_{j-1},max_{j-1}]p_j$ 属(ii)	情况(4)	情况(5)	情况(6)
$p_{j-1}[min_{j-1},max_{j-1}]p_j$ 属(iii)	情况(7)	情况(8)	情况(9)

上述 9 种情况,模式串均为非负间隙模式串.这 9 种情况实际是由如下 4 种形式构成的:

- 形式 1:($p_{j-1}[a,b]p_j$)&($p_j[c,d]p_{j+1}$);
- 形式 2:($p_j[a,b]p_{j-1}$)&($p_j[c,d]p_{j+1}$);
- 形式 3:($p_{j-1}[a,b]p_j$)&($p_{j+1}[c,d]p_j$);
- 形式 4:($p_j[a,b]p_{j-1}$)&($p_{j+1}[c,d]p_j$).

其中,这 4 种形式的间隙 a,b,c 和 d 都为正数或 0,即 $0\leqslant a\leqslant b$ 且 $0\leqslant c\leqslant d$.

易知,形式 1 和形式 4 的等价模式分别为 $p_{j-1}[a,b]p_j[c,d]p_{j+1}$ 和 $p_{j+1}[c,d]p_j[a,b]p_{j-1}$.而形式 2 和形式 3 的等价模式则较为复杂,由于一个可变长度间隙 $X[a,b]Y$ 可以等价地写成 $b-a+1$ 个固定长度间隙,即($X[a,a]Y$)|($X[a+1,a+1]Y$)|...|($X[b,b]Y$),因此,形式 2 和形式 3 将可以推导出$(b-a+1)\times(d-c+1)$个模式串.显然,如果有 $m-1$ 个模式子串属于形式 2 和形式 3,则将产生出指数个模式串.尽管其中部分模式串可以合并为 1 个模式串,但是这样的结果依然过于复杂.为此,我们保留形式 2 和形式 3,而不继续推导.故此,上述 9 种情况的结果如下:

- 情况(1)的结果为 $p_{j-1}[min_{j-1},max_{j-1}]p_j[min_j,max_j]p_{j+1}$;
- 情况(2)的结果为($p_{j-1}[min_{j-1},max_{j-1}]p_j$)&($p_{j+1}[-1-max_j,-1-min_j]p_j$);
- 情况(3)的结果为($p_{j-1}[min_{j-1},max_{j-1}]p_j[0,max_j]p_{j+1}$)|(($p_{j-1}[min_{j-1},max_{j-1}]p_j$)&($p_{j+1}[0,-1-min_j]p_j$));
- 情况(4)的结果为($p_j[-1-max_{j-1},-1-min_{j-1}]p_{j-1}$)&($p_j[min_j,max_j]p_{j+1}$);
- 情况(5)的结果为 $p_{j+1}[-1-max_j,-1-min_j]p_j[-1-max_{j-1},-1-min_{j-1}]p_{j-1}$;
- 情况(6)的结果为

  ($p_{j+1}[0,-1-min_j]p_j[-1-max_{j-1},-1-min_{j-1}]p_{j-1}$)|(($p_j[-1-max_{j-1},-1-min_{j-1}]p_{j-1}$)&($p_j[0,max_j]p_{j+1}$));
- 情况(7)的结果为(($p_j[0,-1-min_{j-1}]p_{j-1}$)&($p_j[min_j,max_j]p_{j+1}$))|($p_{j-1}[0,max_{j-1}]p_j[min_j,max_j]p_{j+1}$);
- 情况(8)的结果为

  ($p_{j+1}[-1-max_j,-1-min_j]p_j[0,-1-min_{j-1}]p_{j-1}$)|(($p_{j-1}[0,max_{j-1}]p_j$)&($p_{j+1}[-1-max_j,-1-min_j]p_j$));
- 情况(9)的结果为

  ($p_{j+1}[0,-1-min_j]p_j[0,-1-min_{j-1}]p_{j-1}$)|(($p_j[0,-1-min_{j-1}]p_{j-1}$)&($p_j[0,max_j]p_{j+1}$))|(($p_{j-1}[0,max_{j-1}]p_j$)&($p_{j+1}[0,-1-min_j]p_j$))|($p_{j-1}[0,max_{j-1}]p_j[0,max_j]p_{j+1}$).

例 3:我们选取文献[14]中的模式 $P8$=a[5,6]c[4,7]g[3,8]t[2,8]a[1,7]c[0,9]g 的变化形式作为实例,介绍如何将一般间隙模式转换为非负间隙模式.之所以选择该模式是因为与其他模式相比,该模式的各个间隔变化较大,更能够体现转换的一般性.将 $P8$ 的最小间隙均改为对应的负值,并且由于转换的繁琐性,我们选取了该模式串长度为 4 的模式子串形成本实例中模式,即 P=a[−5,6]c[−4,7]g[−3,8]t,同时设定长度约束为 11 和 25.

转换模式 P 的方法如下:

P=a[−5,6]c[−4,7]g[−3,8]t⇔(c[0,4]a|a[0,6]c)&(g[0,3]c|c[0,7]g)&(g[−3,8]t)⇔((g[0,3]c[0,4]a)|((c[0,4]a)&(c[0,7]g))|((a[0,6]c)&(g[0,3]c))|(a[0,6]c[0,7]g))&(g[−3,8]t).

上式进一步展开,可以由如下 4 部分组成:

第 1 部分:

(g[0,3]c[0,4]a)&(g[−3,8]t)⇔(g[0,3]c[0,4]a)&(t[0,2]g|g[0,8]t)⇔(t[0,2]g[0,3]c[0,4]a)|((g[0,3]c[0,4]a)&g[0,8]t).

满足模式 t[0,2]g[0,3]c[0,4]a 的出现最大长度为 13,可以满足长度约束 11,25,因此可以保留;而满足模式(g[0,3]c[0,4]a)的出现最大长度为 10,满足 g[0,8]t 的出现最大长度也为 10,因此,((g[0,3]c[0,4]a)&g[0,8]t)的最大长度为 10,不能满足长度约束,因此被略去.故此,(g[0,3]c[0,4]a)&(g[−3,8]t)在满足 11,25 的长度约束下等价模式为(t[0,2]g[0,3]c[0,4]a).这里,令 $Q1$=t[0,2]g[0,3]c[0,4]a.

第 2 部分:

((c[0,4]a)&(c[0,7]g))&(g[−3,8]t)⇔((c[0,4]a)&(c[0,7]g))&(t[0,2]g|g[0,8]t)⇔((c[0,4]a)&(c[0,7]g)&(t[0,2]g))|
((c[0,4]a)&(c[0,7]g)&(g[0,8]t))⇔
((c[0,4]a)&(c[0,7]g)&(t[0,2]g))|((c[0,4]a)&(c[0,7]g[0,8]t)).

易知,((c[0,4]a)&(c[0,7]g)&(t[0,2]g))不能满足最小长度约束,而((c[0,4]a)&(c[0,7]g))&(g[0,8]t)可以满足最小长度约束,因此,((c[0,4]a)&(c[0,7]g))&(g[−3,8]t)在满足 11,25 的长度约束下等价模式为

((c[0,4]a)&(c[0,7]g[0,8]t)).

这里,令 $Q2$=(c[0,4]a)&(c[0,7]g[0,8]t).

第 3 部分:

((a[0,6]c)&(g[0,3]c))&(g[−3,8]t)⇔((a[0,6]c)&(g[0,3]c))&(t[0,2]g|g[0,8]t)⇔((a[0,6]c)&(g[0,3]c)&(t[0,2]g))|
((a[0,6]c)&(g[0,3]c)&(g[0,8]t))⇔
((a[0,6]c)&(t[0,2]g[0,3]c))|((a[0,6]c)&(g[0,3]c)&(g[0,8]t)).

易知,((a[0,6]c)&(t[0,2]g[0,3]c))不能满足最小长度约束,而((a[0,6]c)&(g[0,3]c)&(g[0,8]t))可以满足最小长度约束,因此,((a[0,6]c)&(g[0,3]c))&(g[−3,8]t)在满足 11,25 的长度约束下等价模式为

((a[0,6]c)&(g[0,3]c)&(g[0,8]t)).

这里,令 $Q3$=(a[0,6]c)&(g[0,3]c)&(g[0,8]t).

第 4 部分:.

(a[0,6]c[0,7]g)&(g[−3,8]t)⇔(a[0,6]c[0,7]g)&(t[0,2]g|g[0,8]t)⇔((a[0,6]c[0,7]g)&(t[0,2]g))|(a[0,6]c[0,7]g[0,8]t).

而((a[0,6]c[0,7]g)&(t[0,2]g))和(a[0,6]c[0,7]g[0,8]t)皆可以满足长度约束 11,25,因此均可以保留.

这里,令 $Q4$=(a[0,6]c[0,7]g)&(t[0,2]g),$Q5$=a[0,6]c[0,7]g[0,8]t.

综上,P⇔$Q1$|$Q2$|$Q3$|$Q4$|$Q5$.这里,$Q2$,$Q3$ 和 $Q4$ 均非正常模式,这里略去这些模式的推导过程.表 3 给出了它们的对应等价模式.

Table 3 Equivalent non-negative gap constraints patterns of $Q2$ to $Q4$ with length constraints 11 and 25

表 3 $Q2$~$Q4$ 在长度约束为 11 和 25 时对应的等价非负间隙模式

模式	等价的一般间隙模式	等价的非负间隙模式
$Q2$=(c[0,4]a)&(c[0,7]g[0,8]t)	a[−5,−1]c[0,7]g[0,8]t	(c[0,0]a[0,6]g[0,8]t)|(c[1,1]a[0,5]g[0,8]t)|(c[2,2]a[0,4]g[0,8]t)|(c[3,3]a[0,3]g[0,8]t)|(c[4,4]a[0,2]g[0,8]t)|(c[0,0]g[0,0]a[0,7]t)|(c[0,0]g[1,1]a[0,6]t)|(c[0,0]g[2,2]a[0,5]t)|(c[0,0]g[3,3]a[0,4]t)|(c[1,1]g[0,0]a[0,7]t)|(c[1,1]g[1,1]a[0,6]t)|(c[1,1]g[2,2]a[0,5]t)|(c[2,2]g[0,0]a[0,7]t)|(c[2,2]g[1,1]a[0,6]t)|(c[3,3]g[0,0]a[0,7]t)
$Q3$=(a[0,6]c)&(g[0,3]c)&(g[0,8]t)	a[0,6]c[−4,−1]g[0,8]t	(a[0,5]g[0,0]c[0,7]t)|(a[0,4]g[1,1]c[0,6]t)|(a[0,3]g[2,2]c[0,5]t)|(a[0,2]g[3,3]c[0,4]t)
$Q4$=(a[0,6]c[0,7]g)&(t[0,2]g)	a[0,6]c[0,7]g[−3,−1]t	(a[0,6]c[0,6]t[0,0]g)|(a[0,6]c[0,5]t[1,1]g)|(a[0,6]c[0,4]t[2,2]g)

这样,一个一般间隙模式串就可以等价地转换为多个非负间隙模式串.在具体给定序列串情况下,一个一般间隙模式匹配实例就可以等价地转换为多个非负间隙模式匹配实例.显然,在删除许多不满足间隙约束的模式串情况下,从表 3 可以看出,模式 P 在长度约束为 11 和 25 时,依然对应了 1+15+4+3+1=24 个模式串.

3 子网树定义及性质

文献[30]最早提出了网树概念.为了求解 SPANGLO 问题,在网树基础上给出子网树的定义及其下面的新概念和性质,并对这些概念和性质进行解释.

定义 9. 网树[30]数据结构是结点的集合,这个集合可以为空集,也可以由若干不同的根结点 $r_1,...,r_m$ 和 0 或多个非空子网树 $T_1,T_2,...,T_n$ 构成,这些子网树的树根至少存在 1 条边与网树的根结点 r_i 相连接.这里,$1\leqslant m$,$1\leqslant n$ 且 $1\leqslant i\leqslant n$.图 1 给出了一棵一般意义的网树.

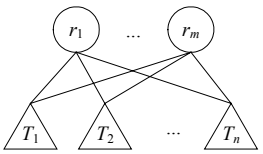


Fig.1 A generic Nettree

图 1 一棵一般意义的网树

网树具有如下 5 个性质[14,30]:

(1) 网树是树结构的拓展,它具有很多与树相似的概念,如根结点、叶子结点、层、双亲、孩子等.

(2) 一棵网树可以有 n 个根结点,其中,$n\geqslant 1$.

(3) 除了根结点之外,网树的其他结点可以有多个双亲结点.

(4) 从一个结点到达网树的一个根结点的路径不唯一.

(5) 同一结点可以在网树的不同层上多次出现.

定义 10. 由于同一结点可以在网树的不同层上多次出现,这里用 n_j^i 来表示第 j 层的结点 i[14].

为了解决长度约束,我们采用子网树的概念以确定出现中的最大值,并在其上构造一些新概念及性质以实现对 SPANGLO 问题的求解.

定义 11. 如果结点 n_b^c 在结点 n_j^i 与某一根结点的路径上,且 $c\leqslant i$,则称结点 n_b^c 是结点 n_j^i 的祖先,其中,$1\leqslant b<j$.结点 n_j^i 的祖先集是由结点 n_j^i 的所有祖先所构成的,用 $A(n_j^i)$ 来表示.

定义 12. 如果结点 n_f^e 在结点 n_j^i 与某一叶子结点的路径上,且 $e<i$,则称结点 n_f^e 是结点 n_j^i 的子孙,其中,$j<f\leqslant m$,m 是网树的最大深度.结点 n_j^i 的子孙集是由结点 n_j^i 的所有子孙所构成的,用 $D(n_j^i)$ 来表示.

定义 13. 子网树是由网树中的一部分结点所构成的,以结点 n_j^i 为基点的子网树是由 $A(n_j^i)$,n_j^i 和 $D(n_j^i)$ 这 3 部分所构成的.由祖先集和子孙集的定义可知,子网树中第 j 层的仅有唯一结点 n_j^i,且该子网树中结点名最大的是 i,其祖先结点名可以为 i 但是其子孙结点名则不能为 i.

定义 14. 第 j 层的最小兄弟是子网树的第 j 层中结点名最小的,用 b_j 来表示;反之,第 j 层的最大兄弟是结点名最大的,用 e_j 来表示.

定义 15. 设 M 是一条从结点 n_j^i ($0\leqslant i<n$ 且 $1\leqslant j$)到达结点 n_b^c ($0\leqslant c<n$ 且 $1\leqslant b$)的路径,a 是这条路径中最小的结点名,即 $a=\min(M)$,如果 M 这条路径满足长度约束,即 $MinLen\leqslant i-a+1\leqslant MaxLen$,则称此路径是一条满足长度约束的路径;否则,$M$ 是一条不满足长度约束的路径.

定义 16. 当前结点 n_j^i 到达祖先结点 n_k^l 的路径数称为祖先结点路径数 NAP(number of ancestor paths),用 $N_A(n_j^i,n_k^l)$ 来表示.当前结点 n_j^i 到达自身的结点路径数为 1,即 $N_A(n_j^i,n_j^i)=1$.

定义 17. 当前结点 n_j^i 到达祖先结点 n_k^l 的路径数中,满足长度约束 LEN 的路径数称为满足约束的祖先结点路径数 NAPLC(number of ancestor paths with length constraints),用 $N_A^C(n_j^i,n_k^l,LEN)$ 来表示,可按如下公式计算:

$$N_A^C(n_j^i,n_k^l,LEN)=\begin{cases}N_A(n_j^i,n_k^l), & MinLen\leqslant i-l+1\leqslant MaxLen\\ \sum_{q=1}^{t}N_A^C(n_j^i,n_{k+1}^{d_q},LEN), & \text{else}\end{cases} \tag{5}$$

其中, $n_{k+1}^{d_q}$ 和 t 分别代表 n_k^l 结点的第 q 个孩子结点及 n_k^l 在子网树内的孩子结点数目.

定义 18. 当前结点 n_j^i 到达树根层结点的路径数中,满足长度约束 LEN 的路径数称为满足约束的树根层结点路径数 NRPLC(number of roots paths with length constraints),用 $N_R^C(n_j^i,LEN)$ 来表示,可按如下公式进行计算:

$$N_R^C(n_j^i,LEN)=\sum_{q=1}^{t}N_A^C(n_j^i,n_1^{d_q},LEN) \tag{6}$$

其中, $n_1^{d_q}$ 和 t 分别代表 n_j^i 结点的第 q 个树根结点及 n_j^i 在子网树内的可以抵达的树根结点数目.

定义 19. 当前结点 n_j^i 到达祖先结点 n_k^l 的路径数中,不满足长度约束 LEN 的路径数称为不满足约束的祖先结点路径数 NCAPLC(number of complement of ancestor paths with length constraints),用 $N_A^{\sim}(n_j^i,n_k^l,LEN)$ 来表示. 显然, $N_A(n_j^i,n_k^l)=N_A^C(n_j^i,n_k^l,LEN)+N_A^{\sim}(n_j^i,n_k^l,LEN)$.$N_A^{\sim}(n_j^i,n_k^l,LEN)$ 可按如下公式进行计算:

$$N_A^{\sim}(n_j^i,n_k^l,LEN)=\begin{cases}0, & MinLen\leqslant i-l+1\leqslant MaxLen\\ \sum_{q=1}^{t}N_A^{\sim}(n_j^i,n_{k+1}^{d_q},LEN), & \text{else}\end{cases} \tag{7}$$

其中, $n_{k+1}^{d_q}$ 和 t 分别代表 n_k^l 结点的第 q 个孩子结点及 n_k^l 在子网树内的孩子结点数目.

定义 20. 当前结点 n_j^i 到达树根层结点的路径数中,不满足长度约束 LEN 的路径数称为不满足约束的树根层结点路径数 NCRPLC(number of complement of roots paths with length constraints),用 $N_R^{\sim}(n_j^i,LEN)$ 来表示,可按如下公式进行计算:

$$N_R^{\sim}(n_j^i,LEN)=\sum_{q=1}^{t}N_A^{\sim}(n_j^i,n_1^{d_q},LEN) \tag{8}$$

其中, $n_1^{d_q}$ 和 t 分别代表 n_j^i 结点的第 q 个树根结点及 n_j^i 在子网树内的可以抵达的树根结点数.

定义 21. 当前结点 n_j^i 到达子孙结点 n_k^l 的路径数,称为子孙结点路径数 NDP(number of descent paths),用 $N_D(n_j^i,n_k^l)$ 来表示.当前结点 n_j^i 到达自身的子孙结点路径数为 1,即 $N_D(n_j^i,n_j^i)=1$.

定义 22. 当前结点 n_j^i 到达子孙结点 n_k^l 的路径数中,满足长度约束 LEN 的路径数称为满足约束的子孙结点路径数 NDPLC(number of descent paths with length constraints),用 $N_D^C(n_j^i,n_k^l,LEN)$ 来表示,可按如下公式计算:

$$N_D^C(n_j^i,n_k^l,LEN)=\begin{cases}N_D(n_j^i,n_k^l), & MinLen\leqslant i-l+1\leqslant MaxLen\\ \sum_{q=1}^{t}N_D^C(n_j^i,n_{k-1}^{d_q},LEN), & \text{else}\end{cases} \tag{9}$$

其中, $n_{k-1}^{d_q}$ 和 t 分别代表 n_k^l 结点的第 q 个双亲结点及其在子网树内的双亲结点数.

定义 23. 当前结点 n_j^i 到叶子层(第 m 层,m 为网树的最大深度)结点的路径数中,满足长度约束 LEN 的路径数称为满足约束的叶子层结点路径数 NLPLC(number of leaves paths with length constraints),用 $N_L^C(n_j^i,LEN)$ 来表示,可按如下公式进行计算:

$$N_L^C(n_j^i,LEN)=\sum_{q=1}^{t}N_D^C(n_j^i,n_m^{d_q},LEN) \tag{10}$$

其中,m 代表子网树的深度, $n_m^{d_q}$ 和 t 分别代表 n_j^i 的第 q 个叶子结点及 n_j^i 在子网树内的可以抵达的叶子结点数.

定义 24. 当前结点 n_j^i 到达子孙结点 n_k^l 的路径数中,不满足长度约束 LEN 的路径数称为不满足约束的子孙结点路径数 NCDPLC(number of complement of descent paths with length constraints),用 $N_D^{\sim}(n_j^i,n_k^l,LEN)$ 来表示. 显然, $N_D(n_j^i,n_k^l)=N_D^C(n_j^i,n_k^l,LEN)+N_D^{\sim}(n_j^i,n_k^l,LEN)$.$N_D^{\sim}(n_j^i,n_k^l,LEN)$ 可按如下公式进行计算:

$$N_D^{\sim}(n_j^i,n_k^l,LEN)=\begin{cases}0, & Minlen\leqslant i-l+1\leqslant Maxlen\\ \sum_{q=1}^{t}N_D^{\sim}(n_j^i,n_{k-1}^{d_q},LEN), & \text{else}\end{cases} \tag{11}$$

其中, $n_{k-1}^{d_q}$ 和 t 分别代表 n_k^l 结点的第 q 个双亲结点及 n_k^l 在子网树内的双亲结点数目.

定义 25. 当前结点 n_j^i 到叶子层(第 m 层)结点的路径数中,不满足长度约束 LEN 的路径数称为不满足约束的叶子层结点路径数 NCLPLC(number of complement of leaves paths with length constraints),用 $N_L^{\sim}(n_j^i,LEN)$ 来表示,可按如下公式进行计算:

$$N_L^{\sim}(n_j^i,LEN)=\sum_{q=1}^{t}N_D^{\sim}(n_j^i,n_m^{d_q},LEN) \tag{12}$$

其中,m 代表子网树的深度, $n_m^{d_q}$ 和 t 分别代表 n_j^i 的第 q 个叶子结点及 n_j^i 在子网树内的可以抵达的叶子结点数.

定义 26. 设 M 是一条从某树根结点到某树叶结点的路径且经过结点 n_j^i,若其满足长度约束,则称 M 是一条满足长度约束的树根-叶子路径;否则,M 是一条不满足长度约束的树根-叶子路径.

定义 27. 在子网树内,经过 n_j^i 结点满足约束的树根-叶子结点路径数 NRLPLC(number of roots- leaves paths with length constraints)用 $N_T^C(n_j^i,LEN)$ 来表示,可按如下公式进行计算:

$$N_T^C(n_j^i,LEN)=N_R^C(n_j^i,LEN)\times N_L^{\sim}(n_j^i,LEN)+N_R^{\sim}(n_j^i,LEN)\times N_L^C(n_j^i,LEN)+N_R^C(n_j^i,LEN)\times N_L^C(n_j^i,LEN) \tag{13}$$

性质 1. $|T(S,P,LEN)|$可按如下公式进行计算:

$$|T(S,P,LEN)|=\sum_{i=MinLen-1}^{n}\sum_{j=1}^{m}N_T^C(n_j^i,LEN) \tag{14}$$

其中,n,m,$MinLen$ 和 $MaxLen$ 分别是序列和模式的长度以及最小和最大长度约束.

为了便于理解,表 4 给出了本文所使用符号的描述.

Table 4 Key symbols used in this paper

表 4 本文主要符号的描述

符号	描述
S	表示序列串,由 n 个字符 $s_0s_1...s_i...s_{n-1}$ 构成
P	表示模式串,有 m 个字符 $p_0p_1...p_{m-1}$ 和 $m-1$ 个间隙构成
min_{j-1}, max_{j-1}	表示通配符可以匹配的最小和最大间隙长度
I	由 m 个位置构成的一个位置索引序列$\langle i_0,...,i_j,...,i_{m-1}\rangle$
LEN	表示长度约束,由最小长度 $MinLen$ 和最大长度 $MaxLen$ 两个正整数构成
$T(S,P,LEN)$	表示模式串 P 在序列串 S 中满足长度约束 LEN 的所有出现的集合,其长度用$\lvert T(S,P,LEN)\rvert$来表示;$T(S,P)$表示无长度约束
⇔	表示前后两个模式串相等
&和|	表示将模式串和间隙分别进行分解的并且运算和或运算
n_j^i	表示第 j 层的结点 i
$A(n_j^i)$	表示结点 n_j^i 的祖先集
$D(n_j^i)$	表示结点 n_j^i 的子孙集
b_j, e_j	第 j 层的最小兄弟和最大兄弟
$N_A(n_j^i,n_k^l)$	表示结点 n_j^i 到达祖先结点 n_k^l 的祖先结点路径数 NAP
$N_A^C(n_j^i,n_k^l,LEN)$	表示结点 n_j^i 到达祖先结点 n_k^l 的路径数中满足长度约束 LEN 的路径数 NAPLC
$N_R^C(n_j^i,LEN)$	表示结点 n_j^i 到达树根层结点的路径数中满足长度约束 LEN 的路径数 NRPLC
$N_A^{\sim}(n_j^i,n_k^l,LEN)$	表示结点 n_j^i 到达祖先结点 n_k^l 的路径数中不满足长度约束 LEN 的路径数 NCAPLC
$N_R^{\sim}(n_j^i,LEN)$	表示结点 n_j^i 到达树根层结点的路径数中不满足长度约束 LEN 的路径数 NCRPLC
$N_D(n_j^i,n_k^l)$	表示结点 n_j^i 到达子孙结点 n_k^l 的子孙结点路径数 NDP
$N_D^C(n_j^i,n_k^l,LEN)$	表示结点 n_j^i 到达子孙结点 n_k^l 的路径数中满足长度约束 LEN 的路径数 NDPLC
$N_L^C(n_j^i,LEN)$	表示结点 n_j^i 到叶子层结点的路径数中满足长度约束 LEN 的路径数 NLPLC
$N_D^{\sim}(n_j^i,n_k^l,LEN)$	表示结点 n_j^i 到达子孙结点 n_k^l 的路径数中不满足长度约束 LEN 的路径数 NCDPLC
$N_L^{\sim}(n_j^i,LEN)$	表示结点 n_j^i 到叶子层结点的路径数中不满足长度约束 LEN 的路径数 NCLPLC
$N_T^C(n_j^i,LEN)$	经过 n_j^i 结点满足长度约束的树根-叶子结点路径数 NRLPLC

4 算法设计

4.1 SETS算法

当接收一个字符 $s_i(0\leqslant i<n)$时,检查其是否满足 $s_i=p_{j-1}(0\leqslant j<m)$,如果满足,则以结点 n_j^i 为基点创建满足长度约束的子网树,创建规则如下:

规则 1. 如果 $s_i=p_{j-1}$,则在第 j 层创建结点 n_j^i.

规则 2. 在向上生成 $A(n_j^i)$ 的过程中,根据子网树的第 k+1 层的最小和最大兄弟 b_{k+1} 和 e_{k+1} 的值计算第 k 层 $(1\leqslant k<j)$的 b_k 和 e_k 的值,并依次检测这个区间上所有 $s_t(b_k\leqslant t\leqslant e_k)$是否满足精确匹配和局部间隙约束,即公式(1)~公式(3),其中,b_k 和 e_k 分别按照如下公式进行计算:

$$b_k=\begin{cases}\max(0,i-\mathrm{Max}Len+1,b_{k+1}-\max_{k-1}-1), & \max_{k-1}\geqslant 0\\ \max(0,i-\mathrm{Max}Len+1,b_{k+1}-\max_{k-1}), & \max_{k-1}<0\end{cases} \tag{15}$$

$$e_k=\begin{cases}\min(i,e_{k+1}-\min_{k-1}-1), & \min_{k-1}\geqslant 0\\ \min(i,e_{k+1}-\min_{k-1}), & \min_{k-1}<0\end{cases} \tag{16}$$

如果 $s_t=p_{k-1}$ 且 s_t 与第 k+1 层结点 n_{k+1}^u 满足公式(2)和公式(3),则可以在第 k 层创建结点 n_k^t,并在这两个结点之间建立“双亲-孩子”关系.之后,依次检测第 k+1 层结点 n_{k+1}^v 与结点 n_k^t 之间是否满足局部间隙约束,如果满足,则在这两个结点之间建立“双亲-孩子”关系.

规则 3. 在向下生成 $D(n_j^i)$ 的过程中,根据子网树的第 k–1 层的最小和最大兄弟 b_{k-1} 和 e_{k-1} 的值计算第 k 层 $(j<k\leqslant m-1)$的 b_k 和 e_k 的值,并依次检测这个区间上所有 $s_t(b_k\leqslant t\leqslant e_k)$是否满足精确匹配和局部间隙约束,即公式(1)~公式(3),其中,b_k 和 e_k 分别按照如下公式进行计算:

$$b_k=\begin{cases}\max(0,i-\mathrm{Max}Len+1,b_{k-1}+min_{k-2}+1), & min_{k-2}\geqslant 0\\ \max(0,i-\mathrm{Max}Len+1,b_{k-1}+min_{k-2}), & min_{k-2}<0\end{cases} \tag{17}$$

$$e_k=\begin{cases}\min(i-1,e_{k-1}-max_{k-2}+1), & max_{k-2}\geqslant 0\\ \min(i-1,e_{k-1}-max_{k-2}), & max_{k-2}<0\end{cases} \tag{18}$$

如果 $s_t=p_{k-1}$ 且 s_t 与第 k–1 层结点 n_{k-1}^u 满足公式(2)和公式(3),则可以在第 k 层创建结点 n_k^t,并在这两个结点之间建立“双亲-孩子”关系.之后,依次检测第 k–1 层结点 n_{k-1}^v 与结点 n_k^t 之间是否满足局部间隙约束,如果满足,则在这两个结点之间建立“双亲-孩子”关系.

下面给出 SETS 算法:

SETS 算法.

输入:$P=p_0[min_0,max_0]p_1\ldots[min_{j-1},max_{j-1}]p_j\ldots[min_{m-2},max_{m-2}]p_{m-1}$, $S=s_0s_1\ldots s_{n-1}$, *MinLen* and *MaxLen*.

输出:解|$T(S,P,LEN)$|.

```
1: sum=0;
2: for i=MinLen–1 to i<n step 1
3:     for j=m–1 downto j≥0 step –1
4:         if (s[i]==p[j])
5:             for k=j–1 downto k≥0 step –1
6:                 按照规则 2 建立第 k+1 层子网树结点
7:                 建立子网树结点同时计算该结点的 NAPLC 和 NCAPLC
8:             next k
9:             依据公式(6)和公式(8)分别计算 NRPLC 和 NCRPLC
10:            for k=j+1 to m–1 step 1
11:                按照规则 3 建立第 k+1 层子网树结点
```

12: 建立子网树结点同时计算该结点的 NDPLC 和 NCDPLC
13: next k
14: 依据公式(10)和公式(12)分别计算 NLPLC 和 NCLPLC
15: 依据公式(13)计算 $N_T^C(n_j^i,LEN),sum+=N_T^C(n_j^i,LEN)$
16: end if
17: next j
18: next i
19: return sum;

4.2 算法复杂性分析

易知,SETS 算法(SETS 算法的源程序可在 http://wuc.scse.hebut.edu.cn/nettree/subnettrees/下载)的空间复杂度为 $O(m\times MaxLen\times W)$,这是因为子网树最多有 m 层,每层最多有 $MaxLen$ 个结点,而每个结点最多有 W 个双亲结点(或孩子结点),即 $W=\max(max_j-min_j+1)(0\leqslant j\leqslant m-1)$,这里,$m$,$MaxLen$ 和 W 分别是模式 P 的长度、最大长度约束和模式 P 的最大间距.

SETS 算法的时间复杂度分析如下:由于每层最多有 $MaxLen$ 个结点,每个结点最多有 W 个双亲结点(或孩子结点),因此,第 6 行、第 7 行及第 11 行、第 12 行的时间复杂度为 $O(MaxLen\times W)$;第 9 行和第 14 行的时间复杂度为 $O(MaxLen)$;第 15 行的时间复杂度为 $O(1)$;综上,第 5 行~第 15 行的时间复杂度为 $O(MaxLen\times W\times m)$.进而可知,SETS 算法的时间复杂度为 $O(MaxLen\times W\times m^2\times n)$.

Min 等人[11]对非负间隙的严格模式匹配问题进行研究并提出了 PAIG 算法,其算法的空间复杂度和时间复杂度分别为 $O(m\times W)$和 $O(W^2\times m^2\times n)$.与 PAIG 算法相比,SETS 算法的空间复杂度和时间复杂度均略大,这是因为 SETS 算法需要处理一般间隙,而 PAIG 算法无须处理一般间隙.在第 2.2 节的理论分析中指出,最坏情况下,一个 SPANGLO 实例可产生出指数个非负间隙模式匹配实例,因此在求解 SPANGLO 问题时,宜采用 SETS 算法.

4.3 算法正确性及完备性

本节给出算法的正确性及完备性证明.

定理 1(算法的正确性). 问题的解是所有子网树中满足约束的树根-叶子结点路径数之和.

证明:在以 n_j^i 为基点的子网树上,通过归纳的方法,易知 $N_A^C(n_j^i,n_k^l,LEN),N_A^{\sim}(n_j^i,n_k^l,LEN),N_D^C(n_j^i,n_k^l,LEN)$ 和 $N_D^{\sim}(n_j^i,n_k^l,LEN)$ 的计算方法的正确性,进而能够知道 $N_R^C(n_j^i,LEN),N_R^{\sim}(n_j^i,LEN),N_L^C(n_j^i,LEN)$ 和 $N_L^{\sim}(n_j^i,LEN)$ 的计算法方法的正确性.在以 n_j^i 为基点的子网树内,满足约束的树根-叶子结点路径数 $N_T^C(n_j^i,LEN)$ 是由树根层满足约束且叶子层不满足约束的路径数、树根层不满足约束且叶子层满足约束的路径数和树根层及叶子层均满足约束的路径数共 3 部分构成的,并且一条完整的树根-叶子路径是由树根层结点和叶子层结点以基结点为桥梁的两部分结点的连接,因此, $N_T^C(n_j^i,LEN)$ 采用公式(13)的计算方法是正确的,进而,问题的解$|T(S,P,LEN)|$采用公式(14)的计算方法是正确的,这样,我们就证明了算法的正确性. □

定理 2(算法的完备性). 任意一个满足约束的出现存在且只存在于某一棵特定的子网树内,且可以用该子网树内的某条树根-叶子结点路径来表示.

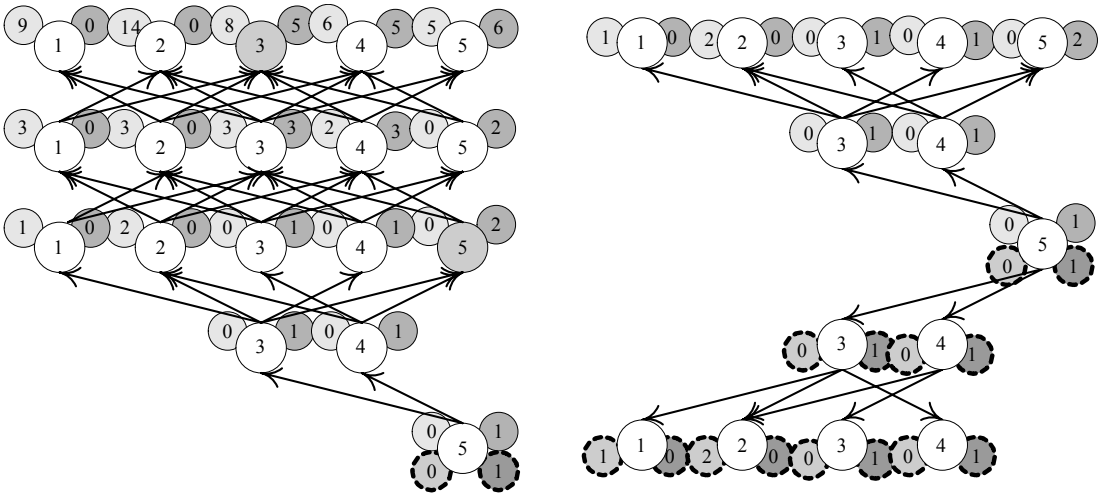
证明:令出现 $I=\langle i_0,...,i_j,...,i_{m-1}\rangle,i_k=\max(i_0,...,i_{m-1})$,如果出现 I 中有若干个位置可以取得最大值 i_k,即$\{i_{a1}=i_k,i_{a2}=i_k,...,i_{aq}=i_k\}$,则 $k=\max(a1,a2,...,aq)$.由子网树的创建规则可知,任何一个出现均以一条树根-叶子结点路径的形式存在于一棵子网树中.如果出现 I 可以在以 n_j^i 和 n_k^i 为基点的两棵不同子网树上存在$(j\neq k)$,则可知 $i_k\in I$ 且 $i_j\in I$.假设 $j<k$,这样,在以 n_j^i 为基点的子网树中,其子孙结点集中包含比自己更大的结点 n_k^i,而这与以 n_j^i 为基点的子网树定义相互矛盾.因此,不存在以 n_j^i 为基点的子网树.而由子网树的创建规则可知,任何一个满足约束的出现都可以用该子网树内的某条树根-叶子结点路径来表示.故此证明:出现 I 存在且只存在于以 n_k^i 为基点的唯一一棵子网树上,且可以用该子网树内的某条树根-叶子结点路径来表示.这样就证明了算法的完备性. □

4.4 运行实例

在如下实例中,我们令序列串 S 和模式串 P 都是全“a”的字符串,以说明 SETS 算法的工作原理,同时展示算法的正确性和完备性.

例 4:给定序列 S=aaaaaaaa,模式 P=a[−2,1]a[−2,1]a[−2,1]a[−2,1]a 以及长度约束 $MinLen$=4 和 $MaxLen$=5,求 $|T(S,P,LEN)|$.

我们仅以 s_5=“a”为例,说明 SETS 算法的工作原理.由于 s_5=p_4=“a”,所以可能存在最后一个位置是 5 的出现,因此,SETS 算法以 n_5^5 结点为基点向上创建满足长度约束的子网树,结果如图 2(a)所示.图 2 中,箭头方向为创建子网树的方向,此外,图中白色圆圈○、浅灰圆圈○、深灰圆圈●、虚框浅灰圆圈◌和虚框深灰圆圈◌内数字分别代表结点名、NAPLC、NCAPLC、NDPLC 和 NCDPLC.在创建结点的同时,计算该结点的 NAPLC 和 NCAPLC.图 2(a)中,每层都选取 1 个典型结点介绍其 NAPLC 和 NCAPLC 是如何计算的,其他结点的 NAPLC 和 NCAPLC 不做详述.由定义 17 和定义 19 可知, $N_A^C(n_5^5,n_5^5,LEN)=0$ 且 $N_A^{\sim}(n_5^5,n_5^5,LEN)=1$. 由公式(5)和公式(7)可知, $N_A^C(n_5^5,n_4^3,LEN)=0$ 且 $N_A^{\sim}(n_5^5,n_4^3,LEN)=1$. 由于 n_3^2 结点有两个孩子结点,分别是 n_4^3 和 n_4^4 且 $MinLen=4\leqslant 5-2+1=4\leqslant MaxLen=5$,因此, $N_A^C(n_5^5,n_3^2,LEN)=2$ 且 $N_A^{\sim}(n_5^5,n_3^2,LEN)=0$. 同理, $N_A^C(n_5^5,n_2^3,LEN)=3$ 且 $N_A^{\sim}(n_5^5,n_2^3,LEN)=3$. 此外, $N_A^C(n_5^5,n_1^3,LEN)=8$ 且 $N_A^{\sim}(n_5^5,n_1^3,LEN)=5$. 这样,依据公式(6)可知, $N_R^C(n_5^5,LEN)=9+14+8+6+5=42$, 因此易知 $N_T^C(n_5^5,LEN)=42$.



(a) 以 n_5^5 为基点的子网树　　(b) 以 n_3^5 为基点的子网树

Fig.2　Some Subnettrees of including 5

图 2　包含 5 的部分子网树

由于 s_5=“a”=p_3=p_2=p_1= p_0,因此不但会形成以 n_5^5 为基点的子网树,而且还会形成以 n_4^5,n_3^5,n_2^5 和 n_1^5 为基点的子网树.这里我们再介绍以 n_3^5 为基点的子网树是如何计算的,这是因为以 n_3^5 为基点的子网树计算更加典型,其他子网树的计算可以类推.以 n_3^5 结点为基点,分别向上和向下创建满足长度约束的子网树,结果如图 2(b)所示.易知各个结点的 NAPLC,NCAPLC,NDPLC 和 NCDPLC.依据公式(6)可知, $N_R^C(n_3^5,LEN)=1+2+0+0+0=3$.同理, $N_R^{\sim}(n_3^5,LEN)=0+0+1+1+2=4$, $N_L^C(n_3^5,LEN)=1+2+0+0=3$ 以及 $N_L^{\sim}(n_3^5,LEN)=0+0+1+1=2$.根据公式(13), $N_T^C(n_3^5,LEN)=3\times 2+3\times 4+3\times 3=27$. 我们可以穷举出在出现的第 3 个位置是 5 且满足长度约束的出现数是 27 个,这样就验证了算法的正确性和完备性.

同理,我们可以分别计算以 n_4^5,n_2^5 和 n_1^5 为基点的子网树中满足长度约束的出现数,这样就完成了对 s_5=“a”的

计算.序列中其他位置字符出现数的计算方法与此相同,不再赘述.

5 实验结果及分析

5.1 实验结果

我们采用例 3 中的模式 P 作为本节实验的模式 $P1$.我们还采用了文献[14]中若干最小间隔不为 0 的模式的变化模式,并将这些最小间隔变成对应的负值,以此形成了本文的模式 $P2$~$P4$.为了研究长度约束中最大长度 *MaxLen* 与问题求解时间之间的关系,我们增加了 $P5$ 和 $P6$,使得这两个模式串与 $P3$ 相比仅在 *MaxLen* 上有差异,其他方面均保持一致.为了研究模式串长度与问题求解时间之间的关系,我们又增加了模式 $P7$ 和 $P8$,使得这两个模式串与 $P6$ 相比仅在模式串长度方面有差异,其他方面均保持一致.为了研究模式最大间隔与问题求解时间之间的关系,我们又增加了模式 $P9$ 和 $P10$,使得这两个模式与 $P8$ 相比仅在最大间隔方面有差异,而其他方面均保持一致.这些具体模式见表 5.

Table 5 General gap patterns used in this paper

表 5 本文使用的一般间隙模式

模式名	模式串	最小长度	最大长度
$P1$	a[−5,6]c[−4,7]g[−3,8]t	11	25
$P2$	g[−1,5]t[0,6]a[−2,7]g[−3,9]t[−2,5]a[−4,9]g[−1,8]t[−2,9]a	24	57
$P3$	g[−1,9]t[−1,9]a[−1,9]g[−1,9]t[−1,9]a[−1,9]g[−1,9]t[−1,9]a[−1,9]g[−1,9]t	21	101
$P4$	g[−1,5]t[0,6]a[−2,7]g[−3,9]t[−2,5]a[−4,9]g[−1,8]t[−2,9]a[−1,9]g[−1,9]t	27	73
$P5$	g[−1,9]t[−1,9]a[−1,9]g[−1,9]t[−1,9]a[−1,9]g[−1,9]t[−1,9]a[−1,9]g[−1,9]t	21	71
$P6$	g[−1,9]t[−1,9]a[−1,9]g[−1,9]t[−1,9]a[−1,9]g[−1,9]t[−1,9]a[−1,9]g[−1,9]t	21	31
$P7$	g[−1,9]t[−1,9]a[−1,9]g[−1,9]t[−1,9]a[−1,9]g[−1,9]t[−1,9]a	21	31
$P8$	g[−1,9]t[−1,9]a[−1,9]g[−1,9]t[−1,9]a[−1,9]g	21	31
$P9$	g[−1,7]t[−1,7]a[−1,7]g[−1,7]t[−1,7]a[−1,7]g	21	31
$P10$	g[−1,5]t[−1,5]a[−1,5]g[−1,5]t[−1,5]a[−1,5]g	21	31

此外,为了验证一个一般间隙模式模式匹配实例转换为多个非负间隙模式匹配实例的正确性,例 3 中的 $Q1$~$Q5$ 模式也将继续使用.表 4 中,$Q3$ 在长度约束为 11 和 25 的情况下,等价的非负间隙模式串为

$Q31$=a[0,5]g[0,0]c[0,7]t,$Q32$=a[0,4]g[1,1]c[0,6]t,$Q33$=a[0,3]g[2,2]c[0,5]t,$Q34$=a[0,2]g[3,3]c[0,4]t.

文本所采用的真实生物序列是文献[14]中所使用的猪流感 H1N1 病毒序列中的一个候选序列(该病毒有很多候选序列,其病毒的 DNA 序列可从美国国家生物计算信息中心 http://www.ncbi.nlm.nih.gov/下载),该序列是 2010 年 3 月 30 日公布的一个结果(A/Managua/2093.01/2009(H1N1)),本文选用其全部 8 个片段(可从 http://www.ncbi.nlm.nih.gov/genomes/FLU/SwineFlu.html 下载)作为测试序列(见表 6).实验运行的软、硬件环境为 Intel(R) Pentinum(R) Dual T2310 处理器、主频 1.46GHZ、内存 1.0 GB、Windows XP 操作系统的计算机.

Table 6 Sequences of real biological data

表 6 真实生物数据序列

序号	片段名称	位点	片段长度
$S1$	Segment 1	CY058563	2 286
$S2$	Segment 2	CY058562	2 299
$S3$	Segment 3	CY058561	2 169
$S4$	Segment 4	CY058556	1 720
$S5$	Segment 5	CY058559	1 516
$S6$	Segment 6	CY058558	1 418
$S7$	Segment 7	CY058557	982
$S8$	Segment 8	CY058560	844

为了验证本文提出的一个一般间隙模式匹配实例转换为多个非负间隙模式匹配实例的正确性及 SETS 算法的求解性能,表 7 给出了 PAIG 算法在模式 $Q31$~$Q34$ 在 $S1$~$S8$ 上的解及运行时间(本文对每个实例采取运行 100 次,然后运行时间为总时间除以 100,以便较为准确地计算出算法在各个实例上的运行时间),表 8 和表 9 分别给出了 SETS 算法求解模式 $Q1$~$Q5$ 以及 $P1$~$P10$ 在序列 $S1$~$S8$ 上的解和运行时间.

Table 7 Numbers of occurrences and running time of patterns $Q31$ to $Q34$ in sequences $S1$ to $S8$ using PAIG

表 7 PAIG 算法求解模式 $Q31$~$Q34$ 在序列 $S1$~$S8$ 上的出现数及运行时间

	出现数(个)				运行时间(ms)			
	$Q31$	$Q32$	$Q33$	$Q34$	$Q31$	$Q32$	$Q33$	$Q34$
$S1$	147	110	98	53	14.85	14.69	14.06	13.75
$S2$	146	129	87	57	14.84	14.38	14.07	14.38
$S3$	166	128	90	57	13.9	13.44	13.28	12.97
$S4$	121	88	62	36	10.94	11.1	10.94	10.79
$S5$	122	77	85	41	9.69	9.69	9.53	9.53
$S6$	131	112	67	54	9.06	9.06	8.91	8.75
$S7$	74	52	54	27	6.09	6.25	6.25	6.09
$S8$	62	53	33	19	5.31	5.16	5.31	5.15

Table 8 Numbers of occurrences of patterns $Q1$ to $Q5$ and $P1$ to $P10$ in sequences $S1$ to $S8$ using SETS (No.)

表 8 SETS 算法求解模式 $Q1$~$Q5$ 和 $P1$~$P10$ 在序列 $S1$~$S8$ 上出现数(个)

	$Q1$	$Q2$	$Q3$	$Q4$	$Q5$	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$	$P8$	$P9$	$P10$
$S1$	54	1 372	408	518	2 967	5 319	822 443	18 642 233	6 855 066	18 246 453	1 021 204	471 464	138 490	54 046	7 234
$S2$	84	1 572	419	504	3 477	6 056	915 866	21 736 881	8 010 515	21 280 825	924 837	494 779	127 788	51 942	5 856
$S3$	75	1 230	441	580	2 936	5 262	751 855	20 207 620	6 924 970	19 751 067	945 590	382 489	127 330	52 360	6 199
$S4$	63	1 041	307	397	2 117	3 925	632 606	13 675 415	4 842 997	13 411 504	580 978	333 144	92 426	37 158	5 474
$S5$	55	1 052	325	324	2 018	3 774	516 352	13 675 526	4 396 525	11 399 942	450 192	250 445	87 036	33 538	3 716
$S6$	54	977	364	478	2 173	4 046	530 953	14 101 378	4 982 424	13 745 869	473 615	239 533	78 859	30 903	3 378
$S7$	48	709	207	257	1 348	2 569	330 532	7 437 824	2 895 318	7 333 410	320 980	150 580	60 303	29 650	4 584
$S8$	35	491	167	218	1 258	2 169	333 962	9 469 875	3 003 216	9 037 213	283 158	141 578	53 484	20 858	2 175

Table 9 Running time of patterns $Q1$ to $Q5$ and $P1$ to $P10$ in sequences $S1$ to $S8$ using SETS (ms)

表 9 SETS 算法求解模式 $Q1$~$Q5$ 和 $P1$~$P10$ 在序列 $S1$~$S8$ 上求解时间 (ms)

	$Q1$	$Q2$	$Q3$	$Q4$	$Q5$	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$	$P8$	$P9$	$P10$
$S1$	1.41	2.03	1.57	1.56	2.18	3.12	40	100.62	88.44	99.69	53.59	33.91	17.97	13.12	8.13
$S2$	1.4	2.03	1.56	1.56	2.19	3.28	43.59	103.75	90.78	103.28	54.37	35.31	17.97	12.97	7.66
$S3$	1.25	1.88	1.56	1.41	2.03	3.13	39.69	101.25	87.35	99.53	51.56	32.97	17.03	12.81	7.35
$S4$	0.94	1.56	1.25	1.1	1.56	2.5	32.35	83.28	68.6	80.94	40.78	26.72	14.07	9.54	5.94
$S5$	0.94	1.41	1.1	1.1	1.41	2.18	26.41	61.87	52.5	62.66	32.81	21.41	11.25	8.28	4.84
$S6$	0.78	1.25	0.94	0.94	1.41	2.03	25.94	66.09	55.31	66.4	33.59	20.93	10.94	8.13	4.68
$S7$	0.62	0.93	0.63	0.78	0.94	1.56	16.4	44.22	38.44	43.6	21.72	14.07	7.35	5.47	3.44
$S8$	0.47	0.78	0.63	0.62	0.94	1.09	15.78	40	33.44	39.69	19.85	12.81	6.41	4.54	2.66

5.2 实验结果分析

(1) 从表 7 和表 8 不难看出,在长度约束 11,25 下,不但模式 $Q3\Leftrightarrow Q31|Q32|Q33|Q34$ 和模式 $P1\Leftrightarrow Q1|Q2|Q3|Q4|Q5$ 成立,而且验证了 SETS 算法的正确性.在表 8 中,SETS 算法求解 $Q3$ 在序列 $S1$ 的出现数为 408,而表 7 中采用 PAIG 算法[11]求得 $Q31$~$Q34$ 在序列 $S1$ 的出现数之和也是 408,我们可以验证 $Q3$ 在其他 7 个序列上的出现数均等于 $Q31$~$Q34$ 在对应序列上的出现数之和.此外,从表 8 中可以看出,模式 $P1$ 在序列 $S1$~$S8$ 中出现数也均等于 $Q1$~$Q5$ 在对应序列上的出现数之和,如 $P1$ 在 $S1$ 上出现数为 5 319,而 $Q1$~$Q5$ 在 $S1$ 上出现数之和也是 5 319.这样既验证了本文提出的一般间隙模式匹配实例转换为多个非负间隙模式匹配实例的正确性,又验证了本文求解算法的求解正确性.

(2) SETS 算法的性能好于 PAIG 算法的性能.尽管 SETS 算法的空间复杂度和时间复杂度比 PAIG 算法都略大,但是从表 7 和表 9 的实际运行时间上看,SETS 算法要好于 PAIG 算法.表 7 中,PAIG 算法求解 $Q31$ 在 $S1$ 的运行时间为 14.85ms,而表 9 中 SETS 算法求解 $Q3$ 在 $S1$ 的运行时间仅为 1.57ms.我们已经实验验证了 $Q3\Leftrightarrow Q31|Q32|Q33|Q34$,这就是说,如果采用 PAIG 算法求解 $Q3$ 在 $S1$ 的总运行时间为 57.35ms.这充分地说明了 SETS 算法好于 PAIG 算法.造成这种现象的原因是:一方面,PAIG 算法需要创建稀疏二维表并在该表上计算,而 SETS 算法则仅对子网树结点进行计算;另一方面,PAIG 算法需要有相对复杂的表合并操作,而 SETS 算法利用子网树及其多种概念与性质进行计算,提高了算法的计算速度.

(3) 尽管本文提出的由一般间隙模式串转换为多个非负间隙模式串的方法是正确的,但是实际应用中并不

可行,其原因是一个一般间隙模式模式匹配实例对应的等价非负间隙模式模式匹配实例的数量非常大.例如,$Q2$是仅含有1个负间隙的模式,在利用长度约束滤掉诸如c[0,0]g[0,0]t[0,2]a或c[0,0]g[1,1]t[0,1]a等模式的情况下,依然对应了15个非负间隙模式.这充分地说明了实际运行中,采用由一般间隙转换为非负间隙的方法实际上并不可行.

(4) 由表8和表9可以看出,SETS算法的运行时间与$MaxLen$,W,m和n等因素正相关,这与理论分析算法SETS时间复杂度为$O(MaxLen\times W\times m^2\times n)$相一致,与问题的解的大小无关,具体分析如下:

由表6可以看出,序列$S2$是全部8个候选序列中最长序列,而序列$S8$是最短的.从表9可以看出,所有模式几乎均在序列$S2$上所花费时间最长,而在序列$S8$上运行时间全部最短.例如,$P1$在$S2$上运行时间为3.28ms,而在其他序列的运行时间均小于3.28ms;反之,$P1$在$S8$上运行时间仅为1.09ms,而在其他序列的运行时间均大于1.09ms.这反映出问题的求解时间与序列串长度是正相关的,即序列串越长,求解时间越长;反之亦然.

由表5可知,$P3$的最大长度约束大于$P5$的最大长度约束且$P5$的最大长度约束大于$P6$的最大长度约束,而其他方面均相同.从表9可以看出,模式$P3$在全部序列上求解时间均长于$P5$的求解时间,而$P5$的求解时间均长于$P6$的求解时间.这反映出问题的求解时间与最大长度约束是正相关的.

由表5可知,$P7$和$P8$在模式串长度上均小于$P6$,而其他方面均相同.从表9可以看出,模式$P8$在全部序列上求解速度最快,这是因为模式$P8$的模式串长度最短;而模式$P6$在全部序列上求解速度最慢,这是因为模式$P6$的序列长度最长.这反映出问题的求解时间与模式串长度是正相关的.

由表5可知,$P9$和$P10$在最大间隙方面均小于$P8$,而其他方面均相同.从表9可以看出,模式$P10$在全部序列上求解速度最快,这是因为模式$P10$的最大间隙最小;而模式$P8$在全部序列上求解速度最慢,这是因为模式$P8$的最大间隙最大.这反映出问题的求解时间与最大间隙是正相关的.

由表8可以看出,$P2$~$P5$模式在$S8$上均没有取得最小解,但从表9可以看出,所有模式在$S8$上求解速度均最快.这是由于序列$S8$的长度最短,因此求解速度最快,且算法的运行时间与序列串长度呈正相关.这充分地说明了算法的运行时间与问题解的大小无关.

综上所述,SETS算法的运行时间与$MaxLen$,W,m和n这4个因素均正相关,验证了算法时间复杂度分析的正确性.

6 结 论

本文提出了具有一般间隙和长度约束的严格模式匹配问题SPANGLO.它是一种允许序列中任意位置的字符被多次使用、模式串中可以包含多个负间隙并且具有长度约束的严格精确模式匹配.本文提出了一般间隙模式串转换为多个非负间隙模式串的方法,并实验验证了这种转换方法的正确性.但在实际求解过程中并不能使用该方法,因为转换的实例数目过多.为此,本文在网树概念基础上建立了子网树结构,并将SPANGLO问题转换为多个子网树结构进行处理,并在子网树上构造了祖先集、子孙集、最小兄弟、最大兄弟、NRPLC、NCRPLC、NLPLC和NCLPLC等概念并形成了SETS算法.本文理论证明了该算法的正确性与完备性,同时指出该算法的空间复杂度与时间复杂度分别为$O(m\times MaxLen\times W)$和$O(MaxLen\times W\times m^2\times n)$.这里,$m$,$n$,$MaxLen$和$W$分别是模式$P$和序列$S$的长度、出现的最大长度约束和模式$P$的最大间距.大量真实生物数据实验验证了SETS算法的正确性和性能.

在实际应用中更多的是近似模式匹配,而本文并未对近似模式匹配下的SPANGLO问题进行研究.此外,在很多序列模式挖掘中,禁止序列中的字符被多次使用,而本文也未对在一次性条件下对一般间隙约束的严格模式匹配开展研究.这些问题的研究不但具有实际应用价值,而且比本文研究的SPANGLO具有更大的求解难度,这些都是进一步开展研究的方向.

致谢 在此,我们向对本文提出宝贵建议的审稿专家表示衷心的感谢,同时对给予本文帮助和支持的大连理工大学江贺教授和河北工业大学的郭迎春副教授及唐志强同学表示衷心的感谢.

References:

[1] Fischer MJ, Paterson MS. String matching and other products. In: Karp R, ed. Proc. of the 7th SIAM AMS Complexity of Computation. Cambridge: American Mathematical Society, 1974. 113−125.

[2] Manber U, Baeza-Yates R. An algorithm for string matching with a sequence of don't cares. Information Processing Letters, 1991, 37(3):133−136. [doi: 10.1016/0020-0190(91)90032-D]

[3] Lewenstein M. Indexing with gaps. In: Grossi R, Sebastiani F, Silvestri F, eds. Proc. of the Int'l Symp. String Processing and Information Retrieval. Pisa: Springer-Verlag, 2011. 135−143. [doi: 10.1007/978-3-642-24583-1_14]

[4] Navarro G, Raffinot M. Fast and simple character classes and bounded gaps pattern matching, with applications to protein searching. Journal of Computational Biology, 2003,10(6):903−923. [doi: 10.1089/106652703322756140]

[5] Wang H, Xie F, Hu X, Li P, Wu X. Pattern matching with flexible wildcards and recurring characters. In: Hu X, Lin TY, Raghavan VV, Grzymala-Busse JW, Liu Q, Broder AZ, eds. Proc. of the 2010 IEEE Int'l Conf. on Granular Computing. Silicon Valley: IEEE Computer Society, 2010. 782−786. [doi: 10.1109/GrC.2010.156]

[6] Cole R, Gottlieb LA, Lewenstein M. Dictionary matching and indexing with errors and don't cares. In: Babai L, ed. Proc. of the 36th ACM Symp. on the Theory of Computing. Chicago: ACM Press, 2004. 91−100. [doi: 10.1145/1007352.1007374]

[7] Crochemore M, Iliopoulos C, Makris C, Rytter W, Tsakalidis A, Trichlas K. Approximate string matching with gaps. Nordic Journal of Computing, 2002,9(1):54−65.

[8] Cantone D, Cristofaro S, Faro S. New efficient bit-parallel algorithms for the (δ,α)-matching problem with applications in music information retrieval. Int'l Journal of Foundations of Computer Science, 2009,20(6):1087−1108. [doi: 10.1142/S0129054109007054]

[9] Ji X, Bailey J, Dong G. Mining minimal distinguishing subsequence patterns with gap constraints. Knowledge and Information Systems, 2007,11(3):259−286. [doi: 10.1007/s10115-006-0038-2]

[10] Ferreira PG, Azevedo PJ. Protein sequence pattern mining with constraints. In: Jorge A, Torgo L, Brazdil P, Camacho R, Gama J, eds. Proc. of the European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD). Porto: Springer-Verlag, 2005. 96−107. [doi: 10.1007/11564126_14]

[11] Min F, Wu X, Lu Z. Pattern matching with independent wildcard gaps. In: Proc. of the 8th Int'l Conf. on Pervasive Intelligence and Computing. Chengdu: IEEE, 2009. 194−199. [doi: 10.1109/DASC.2009.65]

[12] Navarro G. A guided tour to approximate string matching. ACM Computing Surveys, 2001,33(1):31−88. [doi: 10.1145/375360.375365]

[13] Chen G, Wu X, Zhu X, Arslan AN, He Y. Efficient string matching with wildcards and length constraints. Knowledge and Information Systems, 2006,10(4):399−419. [doi: 10.1007/s10115-006-0016-8]

[14] Wu YX, Wu XD, Jiang H, Min F. A heuristic algorithm for MPMGOOC. Chinese Journal of Computers, 2011,34(8):1452−1462 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011.01452]

[15] Li C, Yang Q, Wang J, Li M. Efficient mining of gap-constrained subsequences and its various applications. ACM Trans. on Knowledge Discovery from Data (TKDD), 2012,6(1):Article 2. [doi: 10.1145/2133360.2133362]

[16] Zhang S, Zhang J, Zhu X, Huang Z. Identifying follow-correlation itemset-pairs. In: Proc.of the Int'l Conf. on Data Mining (ICDM). Hong Kong: IEEE Computer Society, 2006. 765−774. [doi: 10.1109/ICDM.2006.84]

[17] Zhang M, Kao B, Cheung D, Yip K. Mining periodic patterns with gap requirement from sequences. ACM Trans. on Knowledge Discovery from Data, 2007,1(2):Article 7. [doi: 10.1145/1267066.1267068]

[18] Tanbeer SK, Ahmed CF, Jeong BS. Mining regular patterns in data streams. In: Kitagawa H, Ishikawa Y, Li Q, Watanabe C, eds. Proc. of the Database Systems for Advanced Applications. Tsukuba: Springer-Verlag, 2010. 399−413. [doi: 10.1007/978-3-642-12026-8_31]

[19] Li Z, Ding B, Han J, Kays R, Nye P. Mining periodic behaviors for moving objects. In: Rao B, Krishnapuram B, Tomkins A, Yang Q, eds. Proc. of the 16th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). Washington: ACM Press, 2010. 1099−1108. [doi: 10.1145/1835804.1835942]

[20] Zhu X, Wu X. Mining complex patterns across sequences with gap requirements. In: Veloso MM, ed. Proc. of the 20th Int'l Joint Conf. on Artificial Intelligence (IJCAI). Hyderabad: Springer-Verlag, 2007. 2934−2940.

[21] Huang Y, Wu X, Hu X, Xie F, Gao J, Wu G. Mining frequent patterns with gaps and one-off condition. In: Proc. of the IEEE Int'l Conf. on Computational Science and Engineering (CSE 2009). Vancouver: IEEE Computer Society, 2009. 180−186. [doi: 10.1109/CSE.2009.160]

[22] Ding B, Lo D, Han J, Khoo SC. Efficient mining of closed repetitive gapped subsequences from a sequence database. In: Ioannidis YE, Lee DL, Ng RT, eds. Proc. of the IEEE 25th Int'l Conf. on Data Engineering (ICDE). Shanghai: IEEE, 2009. 1024−1035. [doi: 10.1109/ICDE.2009.104]

[23] Myers E. Approximate matching of network expressions with spacers. Journal of Computational Biology, 1996,3(1):33−51. [doi: 10.1089/cmb.1996.3.33]

[24] Fredriksson K, Grabowski S. Efficient algorithms for pattern matching with general gaps and character classes. In: Crestani F, Ferragina P, Sanderson M, eds. Proc. of the Int'l Conf. on String Processing and Information Retrieval. Glasgow: Springer-Verlag, 2006. 267−278. [doi: 10.1007/11880561_22]

[25] Fredriksson K, Grabowski S. Efficient algorithms for pattern matching with general gaps, character classes, and transposition invariance. Information Retrieval, 2008,11(4):335−357. [doi: 10.1007/s10791-008-9054-z]

[26] Akutsu T. Approximate string matching with variable length don't care characters. IEICE Trans. on Information and Systems, 1996, E79-D(9):1353−1354.

[27] Bille P, Gørtz I, Vildhøj H, Wind D. String matching with variable length gaps. In: Chávez E, Lonardi S, eds. Proc. of the 17th Int'l Conf. on String Processing and Information Retrieval (SPIRE 2010). Mexico: Springer-Verlag, 2010. 385−394. [doi: 10.1007/978-3-642-16321-0_40]

[28] Rahman S, Iliopoulos C, Lee I, Mohamed M, Smyth W. Finding patterns with variable length gaps or don't cares. In: Chen DZ, Lee DT, eds. Proc. of the 12th Annual Int'l Conf. on Computing and Combinatorics. Taipei: Springer-Verlag, 2006. 146−155. [doi: 10.1007/11809678_17]

[29] He D, Wu X, Zhu X. SAIL-APPROX: An efficient on-line algorithm for approximate pattern matching with wildcards and length constraints. In: Proc. of the 2007 IEEE Int'l Conf. on Bioinformatics and Biomedicine (BIBM 2007). Silicon Valley: IEEE Computer Society, 2007. 151−158. [doi: 10.1109/BIBM.2007.48]

[30] Wu Y, Wu X, Min F, Li Y. A Nettree for pattern matching with flexible wildcard constraints. In: Proc. of the 2010 IEEE Int'l Conf. on Information Reuse and Integration (IRI 2010). Las Vegas: IEEE Systems, Man, and Cybernetics Society, 2010. 109−114. [doi: 10.1109/IRI.2010.5558954]

附中文参考文献:

[14] 武优西,吴信东,江贺,闵帆.一种求解 MPMGOOC 问题的启发式算法.计算机学报,2011,34(8):1452−1462. [doi: 10.3724/SP.J.1016.2011.01452]

武优西(1974－),男,河北故城人,博士,教授,CCF 会员,主要研究领域为智能计算,数据挖掘.
E-mail: wuc@scse.hebut.edu.cn

郭磊(1968－),男,博士,教授,主要研究领域为图像处理,数据挖掘.
E-mail: guoshengrui@163.com

刘亚伟(1987－),男,硕士生,主要研究领域为模式匹配.
E-mail: lywde2011@163.com

吴信东(1963－),男,博士,教授,博士生导师,主要研究领域为数据挖掘,基于知识的系统,万维网信息探索.
E-mail: xwu@hfut.edu.cn