基于密度约束和间隙约束的对比模式挖掘

魏芹双 武优西 刘靖宇 朱怀忠

(河北工业大学计算机科学与软件学院 天津300401) (河北省大数据重点实验室 天津300401)

摘 要 对比模式挖掘是序列模式挖掘的一个重要分支,带有密度约束的对比模式有助于生物学家发现生物序列中的特殊因子的分布情况。为此,文中提出了 MPDG(Mining distinguishing sequence Patterns based on Density and Gap constraint)算法,该算法应用网树结构挖掘满足密度约束和间隙约束的对比模式,在仅需扫描一遍序列库的情况下,该算法可计算当前模式的所有超模式的支持度,从而提高挖掘效率。最后,在真实蛋白质数据集上进行实验,实验结果验证了 MPDG 算法的有效性。

关键词 模式挖掘,对比模式,密度约束,网树

中图法分类号 TP311 文献标识码 A **DOI** 10.11896/j.issn.1002-137X.2018.04.042

Distinguishing Sequence Patterns Mining Based on Density and Gap Constraints

WEI Qin-shuang WU You-xi LIU Jing-yu ZHU Huai-zhong
(School of Computer Science and Engineering, Hebei University of Technology, Tianjin 300401, China)
(Hebei Province Key Laboratory of Big Data Calculation, Tianjin 300401, China)

Abstract Distinguishing patterns mining is an important branch of sequence patterns mining, and distinguishing patterns with density constraint can help biologists to find the distribution of special factors on biological sequences. This paper proposed an algorithm, named MPDG (Mining distinguishing sequence Patterns based on Density and Gap constraint), which employs Nettree data structure to mine the distinguishing patterns satisfying the density and gap constraints. The algorithm is efficient since it calculates all super-patterns' supports of current pattern with one-way scanning the sequence database. Experimental results on real protein datasets verify the effectiveness of MPDG.

Keywords Pattern mining, Distinguishing pattern, Density constraint, Nettree

1 引言

自 Agrawal 和 Srikant^[1]提出序列模式挖掘以来,其作为数据挖掘的热门分支,吸引了大批研究者的关注^[2-6]。与传统挖掘方法不同,对比模式挖掘^[6]能够描述两类或者多类样本中的对比信息,识别不同类别样本集合的特征,在疾病预测、生物信息发现等领域有着广泛的应用^[7-11]。Ji 等人^[8]提出了对比模式并提出了解决满足间隙约束的最小对比模式挖掘算法——ConSGapMiner 算法;在此基础之上,Wang 等人^[9]提出了满足密度约束的模式挖掘算法——gd-DSPMiner 算法;杨皓等人^[10]为解决由于用户设置不恰当的支持度阈值而造成模式丢失的问题,提出了带间隔约束的 top-k 对比模式挖掘算法——kDSP-Miner 算法;王慧锋等人^[11]提出了一个有效的方法来解决对比序列挖掘中间隔设定的问题。尽管文献 [9]中提出的 gd-DSPMiner 算法较 ConSGapMiner 算法^[8]有

较好的提高,但其在计算模式在序列中的支持数时采用了匹配三元组的方法,该方法中不管模式在序列的某位置是否会出现,都会为该位置建立匹配三元组,进而降低算法效率和浪费空间。为此,本文提出了 MPDG 算法。该算法应用网树结构[12-14]在扫描一次序列的情况下,就可以计算当前模式的所有超模式在序列中的支持数,从而提高了算法的挖掘效率。

本文第 2 节给出了问题的定义;第 3 节提出了 MPDG 算法,通过一个实例说明了该算法的工作原理并对该算法的复杂度进行分析;第 4 节对实验结果进行分析;最后总结全文。

2 问题定义

定义 1 二分类序列库是由多个序列构成的集合,表示为 $D=\{S_1,S_2,\cdots,S_M,S_{M+1},\cdots,S_{M+N}\}$,其中 S_i 是由若干元素组成的序列,用 len(D)表示序列库中所有元素的个数。前 M个序列属于正例序列库,记为 D+;后 N 个序列属于负例

到稿日期:2017-02-06 返修日期:2017-05-11 本文受国家自然科学基金(61673159),河北省自然科学基金(F2016202145),黑龙江省自然科学基金(F2017019),河北省科技计划项目(15210325),河北省教育厅青年基金(QN2014192)资助。

魏芹双(1991—),女,硕士生,主要研究方向为数据挖掘,E-mail: shuang0118@qq. com;**武优西**(1974—),男,博士,教授,CCF 高级会员,主要研究方向为数据挖掘、智能计算,E-mail: wuc567@163. com(通信作者);刘靖宇(1976—),男,博士,副教授,主要研究方向为数据挖掘、绿色存储;朱怀忠(1978—),男,硕士,讲师,主要研究方向为数据挖掘。

序列库,记为 D-。

定义 2 给定模式 $P = p_1 p_2 \cdots p_{l-1} p_l$,间隙约束为[min, max],模式 P 在序列 S 中的一个位置索引序列 $I = \langle i_1, \cdots, i_j, \cdots, i_l \rangle$ 满足: $S_{ij} = P_j$, $i_{j-1} < i_j$,以及 min $\leqslant i_j - i_{j-1} - 1 \leqslant$ max。其中, $1 \leqslant j \leqslant l$ 且 $1 \leqslant i_j \leqslant l - 1$ 。称位置索引序列 I 是模式 P 在序列 S 中的一个出现。将所有出现的个数称作支持数,用 count(P,S)表示。

定义 3 模式 $P = p_1 p_2 \cdots p_{l-1} p_l$ 和模式 $Q = q_1 q_2 \cdots q_l q_{l+1}$ 是两个长度分别为 l 和 l+1 的模式,对于任意的 $1 \le i \le l$,有 $p_i = q_i$,则称 $P \neq Q$ 的子模式, $Q \neq P$ 的超模式。

定义 4 给定序列 S 的长度为 n,模式 P 的长度为 l,如果索引序列 $L=\langle d_1,d_2,\cdots,d_l\rangle$ 满足: $1\leqslant d_j\leqslant n,d_{j-1}\leqslant d_j$,以及 $\min \leqslant d_j-d_{j-1}-1\leqslant \max$,那么 L 就是模式 P 在序列 S 中的一个偏移出现,所有偏移出现的个数用 ofs(P,S)表示。

定义 5 当间隙约束为 $[\min,\max]$ 时,将模式 P 在序列 S 的支持数与偏移出现数的比值定义为模式在序列中的密度,用 density(P,S) 表示,即 density(P,S) = count(P,S)/ofs (P,S)。

定义 6 给定二分类序列库 D、密度阈值 mindensity、间隙约束[min, max],用 sup(D, P, mindensity)来表示模式 P在序列库 D 中的支持度。其计算公式如下:

 $sup(D, P, mindensity) = \sum |\{S \in D \mid density(S, P) > mindensity\}|/len(D)$

若模式 P 满足以下 3 个条件,则称模式 P 是一个满足密度约束和间隙约束的对比模式(gd-MPDG):

1)在正例序列库 D+中是频繁的,即 $sup(D+,P,min-density) \geqslant minsup_D+$;

2)在负例序列库 D-中是不频繁的,即 $sup(D-,P,mindensity) \leq minsup_D-$;

3)最小性条件:模式 P 的子模式都不同时满足上述两个 条件。

3 挖掘算法

本文提出的挖掘算法为 MPDG,挖掘出来的满足密度约束和间隙约束的对比模式被称为 gd-MPDG。该算法基于一种特殊的数据结构——网树^[12],其定义如下。

定义 7 网树(Nettree)是一种树结构的拓展,除具有树结构中的根结点、叶子结点、子孙、祖先、层等概念外,还具有如下几个特殊特点:一棵网树可以有多个根结点;网树除了根结点之外的其他结点都可以有多个双亲结点;一个结点可以有多条路径到达网树的同一个根结点;同一个结点可以在网树的不同层中出现,用 n_i 来表示在第j层上出现的结点i。

定义8 从结点 n_i 到该网树所有的根结点的路径数被称为该结点的树根路径数(Number of Root Path, NRP),用 $R(n_i^i)$ 来表示。其计算方法为:根结点的NRP为 1;第 j(j>1)层结点 n_i^i 的树根路径数 $R(n_i^i)$ 为其所有父结点的NRP之和

计算长度为l的模式P在序列S的支持度可以视为一个模式匹配问题[12-14],可将其转化为一棵深度为l的网树,模式P的支持度值为l层叶子结点的树根路径数之和。由于在计算模式P的超模式Q的支持度过程中,无需前l-1层结点

的信息,因此仅仅保留模式 P 第 l 层叶子结点的信息即可,而这样的网树被称为不完整网树 $^{[15]}$ 。利用网树结构提出的MPDG 算法分为以下 3 个步骤。

1)计算支持数。因为模式 P 在某个序列中的匹配可以用不完整网树表示,所以在序列库 D 上需要建立多棵不完整网树,其中第 id(1 < id < M+N) 棵不完整网树代表模式在序列 S_{id} 上的匹配。首先扫描序列 S_{id} 建立的所有长度为 1 的模式的不完整网树,得到长度为 1 的模式在序列 S_{id} 中的支持数;然后扫描序列 S_{id} ,建立长度为 l(l > 1) 的模式的不完整网树,从而得到所有以当前模式为最大前缀的模式在序列中的支持数。

2)生成待挖掘模式。在 MPDG 算法中利用广度优先的 方法生成候选模式树。在深度优先挖掘的方法中,很多最小 的对比模式不能被及时挖掘出来,从而造成时间和空间的浪 费。我们摒弃传统的将字符集中所有的元素直接与候选模式 连接生成待挖掘模式的方法,采用一种新的生成待挖掘模式 的方法。为了更好地说明该方法,给出下面两个定义。

定义 9(最大前缀和最大后缀) 给定一个模式 $P = p_1 p_2 \cdots p_{l-1} p_l (l \ge 2)$,模式 P 的最大前缀 $pre^l(P) = p_1 p_2 \cdots p_{l-1}$,模式 P 的最大后缀 $suf^l(P) = p_2 \cdots p_{l-1} p_l$ 。

定义 10 P 和 Q 是长度为 $l(l \ge 1)$ 的候选模式,如果 $pre^{l}(P) = suf^{l}(Q)$,就可以生成一个长度为 l+1 的待挖掘模式,即 $P \oplus Q = q_1 p_1 p_2 \cdots p_{l-1} p_l$ 或 $P \oplus Q = q_1 q_2 \cdots q_{l-1} q_l p_l$ 。

根据上述定义,可以通过长度为l的模式生成所有长度为l+1的待挖掘模式。

3)挖掘 gd-MPDG。挖掘 gd-MPDG 的详细过程为:首先将字符集 Σ 中的每个字符看作长度为 1 的模式 P,为每个长度为 1 的模式建立不完整网树森林并且计算它们在序列 S_{id} 中的支持数、密度,从而计算它们在序列库 D+和 D-中的支持度。如果模式是 gd-MPDG,就将该模式加入到对比模式集 F中;否则将该模式及其不完整网树森林加入到队列的尾部。然后从队列中取出所有长度为 l(l)1)的候选模式,生成长度为 l+1 的待挖掘模式,扫描一次序列计算当前模式的所有超模式在序列 S_{id} 中的支持数、密度以及在序列库 D+和 D-中的支持度。如果该模式是 gd-MPDG,那么将它加入到对比模式集 F中;否则将该模式及其不完整网树森林加入到队列的尾部,迭代上述过程直到队列为空。MPDG 算法的过程描述如算法 1 所示。

算法 1

输入:D,id,min,max,minsup_D+,minsup_D-,mindensity,M,N 输出:所有对比模式集合 F

- 1. 为所有模式长度为 1 的模式建立 Fnode;//Fnode 为不完整网树森林 2. for(k=0;k<Fnode. size();k++)
- 3. if $id \le M$ then $sup(D+,P,mindensity) + = |S_{id}|/len(D+)$
- 4. else sup(D-,P,mindensity) $+= |S_{id}|/len(D-)$
- 5. end if

6. end for

- 7. if sup(D+,P,mindensity)>=minsup_D+&& sup(D-,P,mindensity)<=minsup_D- then 将该模式加入到对比模式集F中
- 8. else 将该模式及其不完整网树森林加入到队列 meta 中

9. end if

10. while(!meta. empty())

- 11. oldFnode = = meta. dequeue();
- 12. 对每一个模式生成待挖掘模式,并对它们建立 newFnode
- 13. for(k=0; $k \le \text{newFnode. size}()$; k++)
- 14. if id $\leq = M$ then sup(D+,P,mindensity)+= $|S_{id}|/len(D+)$
- 15. else sup(D-,P,mindensity) += $|S_{id}|/len(D-)$
- 16. end if
- 17. end for
- 18. if sup(D+,P,mindensity)>=minsup_D+&&sup(D-,P,mindensity)<=minsup_D- then 将该模式加入到对比模式集 F 中
- 19. else 将该模式及其不完整网树森林加入到队列 meta 中
- 20. end if
- 21. end while
- 22. return F

例 1 给定序列库 D,如表 1 所列,字符集 $|\Sigma| = \{A, C, G, T\}$, mindensity = 0.38, $minsup_D + = 0.5$, $minsup_D - = 0$,间隙约束为[0,5],挖掘此序列库中的 gd-MPDG。

表1 一个序列库

Table 1 A sequence dataset

序列号	序列	集合	
1	AATTTC	D+	
2	AAAATTTC	D+	
3	AGTACT	D+	
4	CACCGA	D-	
5	TACGGC	D-	
6	CCGG	D-	

首先将字符集中的每个字符当作长度为 1 的模式,在序列库中建立 6 棵不完整网树。以序列 S_1 为例,扫描一次序列建立所有长度为 1 的模式在序列 S_1 中的不完整网树,如图 1 所示,得到模式 $P_{11}=A$, $P_{12}=C$, $P_{13}=G$, $P_{14}=T$ 在 S_1 中的支持数分别为 1+1=2,10,1+1+1=3。由于长度为 1 的模式在 S_1 中的偏移出现数为 6,因此这些模式的密度分别为 1/3,1/6,0,1/2。同理可以建立其他不完整网树,得到各模式在序列中的密度,从而得到上述模式在 D+中的支持度分别为 1/30.4,1/30,1/30。因此,长度为 1 的模式均不是 1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30,1/30 ,1/30

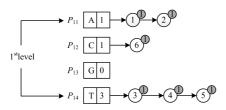


图 1 长度为 1 的模式在序列 S_1 中的不完整网树

Fig. 1 Incomplete Nettree of the pattern with length 1 in S_1

根据队列中所有长度为 1 的模式生成长度为 2 的待挖掘模式: $P_{21} = AA$, $P_{22} = AC$, $P_{23} = AG$, $P_{24} = AT$, $P_{25} = CA$, $P_{26} = CC$ 等。计算长度为 2 的模式在各序列中的支持数。扫描一次 S_1 得到以模式 P_{11} 为最大前缀的超模式的不完整网树,如图 2 所示。由图 2 可以得到,模式 P_{21} , P_{22} , P_{23} , P_{24} 在 S_1 中的出现数分别为 1,2,0,2+2+2=6,长度为 2 的模式在 S_1 中的偏移出现数为 15,从而可以得到各模式在 S_1 中的密度。同理可以得到这些模式在 $S_2 - S_6$ 中的密度,可得 S_1 S_2 S_3

 P_{24} ,0.38)=($|S_1|+|S_2|$)/20=0.7>0.5;同理可以得到 sup ($D-,P_{24}$,0.38)=0。因此模式 P_{24} =AT 是 gd-MDSP,将其 加入到对比模式集 F中。其他长度为 2 的模式在 D+中的支持度为 0,将它们人队等待生成长度为 3 的待挖掘模式。迭代上述过程直到队列为空为止。综上,根据定义可得,在这个例子中挖掘的所有 gd-MPDG 为 P_{24} =AT。

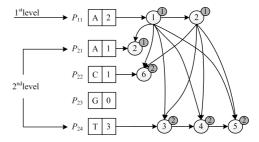


图 2 模式 P11 超模式的不完整网树

Fig. 2 Incomplete Nettree of pattern P_{11}

下面对算法的复杂度进行分析。由于不完整网树中的每个结点存储名字和 NRP,但其空间消耗为 1,因此结点的空间复杂度为 O(1)。若序列库中最长的序列长度为 n,每棵不完整网树平均包含 $n/|\Sigma|$ 个结点($|\Sigma|$ 是字符集中的元素个数),则每棵不完整网树的空间复杂度为 $O(n/|\Sigma|)$ 。在序列库中有 M+N 个序列,每求得一个模式的支持度需建立 M+N 棵不完整网树。 MPDG 算法采用队列结构实现,空间消耗与队列的最大长度有关,当挖掘的所有的 gd-MPDG 的长度均为 MaxLen 时,队列的最大长度为 $|\Sigma|^{MaxLen}$ 。 因此 MPDG 算法的空间复杂度为 $O(|\Sigma|^{MaxLen-1}*n*(M+N))$ 。

每个模式在单序列的不完整网树中的平均结点数为 $n/|\Sigma|$,并且每个结点需要使用 $W=\max-\min+1$ 次,在单个序列中根据子模式计算超模式的支持数的时间复杂度为 $O(n/|\Sigma|*W)$,因此在序列库中的时间复杂度为 $O(n/|\Sigma|*W*(M+N))$ 。当挖掘到所有 gd-MPDG 的长度均为 MaxLen时,MPDG 算法的时间复杂度为 $O(|\Sigma|^{MaxLen+1}*n*W*(M+N))$ 。

4 实验结果与分析

本文在蛋白质序列上进行实验来验证 MPDG 算法挖掘 gd-MPDG 的有效性和高效性。采用的蛋白质序列集是从 PFam:Protein Family Database¹⁾ 中获取的,其具体特征如表 2 所列。MPDG 算法和 gd-DSPMiner 算法均用 C++实现,所 有实验都在配置为 Intel Core i5-4200U 1.6 GHz CPU,4.0 GB 内存、Windows 8 操作系统的 PC 上完成。本文从 *minsup_D*+对挖掘模式的数量的影响、不同间隙约束对挖掘模式的数量的影响和不同间隙约束的挖掘时间这 3 个方面进行分析。

表 2 序列库特征

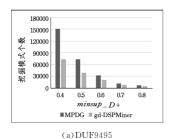
Table 2 Summary of dataset characteristics

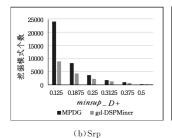
数据集名称	序列类型	D+	D-	备注
DUF9495	蛋白质序列	16	5	各序列不等长
Srp	蛋白质序列	5	4	各序列不等长
Cbi	蛋白质序列	80	76	各序列不等长

图 3 给出了改变 minsup_D+对 MPDG 算法和 gd-

¹⁾ http://pfam. sanger. ac. uk/

DSPMiner 算法挖掘出的对比模式个数的影响。我们知道,随 着 minsup_D+的增大,挖掘出来的对比模式的个数会逐渐减 少。图 3 明显地呈现了这一现象,从而验证了算法的有效性。 此外,从本实验亦可看出,在参数相同的情况下,MPDG 算法 挖掘模式的个数远远大于 gd-DSPMiner 算法挖掘模式的个 数。例如,在图 3(b)中, minsup_D+为 0.125 时, MPDG 算 法可以挖掘出 24041 个对比模式,而算法 gd-DSPMiner 只能 挖掘出8900个对比模式。这种现象在其他数据集中也存在, 此处不再赘述。





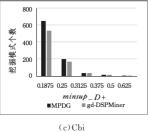
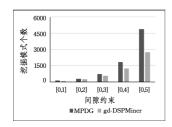


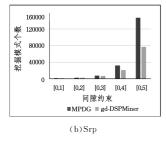
图 3 不同 minsup_D+对挖掘个数的影响

Fig. 3 Influence of different minsup_D+ on mining results

图 4 给出了改变间隙约束对 MPDG 算法和 gd-DSPMiner 算法挖掘对比模式个数的影响。



(a)DUF9495



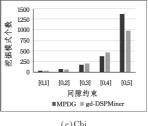


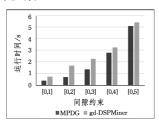
图 4 不同间隙约束对挖掘个数的影响

Fig. 4 Influence of different gap constraints on mining results

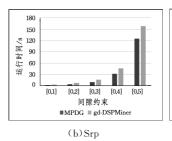
从图 4 中可以得到,随着间隙的增大,两种算法挖掘到的 对比模式的个数都在增大,并且本文提出的 MPDG 算法挖掘 出的对比模式的个数远远大于传统的 gd-DSPMiner 算法,因 此验证了本文算法的性能。

图 5 给出改变间隙约束对 MPDG 算法和 gd-DSPMiner

算法挖掘速度的影响。



(a)DUF9495



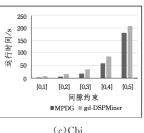


图 5 不同间隙对挖掘速度的影响

Fig. 5 Influence of different gap constraints on mining speed

从图 5 可以看出,随着间隙约束的增大,挖掘时间会越来 越长,算法 MPDG 的时间复杂度为 $O(|\Sigma|^{MaxLen+1} * n * W *$ (M+N)),算法的运行时间会随着间隙的增大而变长,这验 证了算法的有效性。但是不管间隙如何变化,本文提出的 MPDG 算法的消耗时间均比 gd-DSPMiner 算法少,并且间隙 越小时 MPDG 算法的优势越显著。此外,从图 4、图 5 可以看 出,MPDG 算法挖掘对比模式的数量多于 gd-DSPMiner 算 法,而挖掘时间又少于 gd-DSPMiner 算法,这进一步说明了 MPDG 算法在时间效率方面优于 gd-DSPMiner 算法。

综上,在影响挖掘模式个数的各参数相同的条件下,本文 提出的 MPDG 算法较 gd-DSPMiner 算法可以挖掘出更多的 对比模式,并且 MPDG 算法挖掘消耗的时间更短,挖掘效率 更高。

结束语 本文对已有的对比模式挖掘算法进行了分析, 介绍了满足密度约束和间隙约束的对比模式挖掘问题。在给 定定义的基础上,采用队列结构实现了广度优先生成候选模 式树的方法,提出了应用网树结构对对比模式进行挖掘的 MPDG 算法。本文在蛋白质数据集上进行了大量的实验,实 验结果验证了 MPDG 算法的有效性,并且在相同的挖掘条件 下,MPDG算法的挖掘效率更高。

参考文献

- [1] AGRAWAL R, SRIKANT R. Mining sequential patterns [C]// 11th International Conference on Data Engineering. 1995:3-14.
- [2] ZHANG L, LUO P, TANG L, et al. Occupancy-based frequent pattern mining [J]. ACM Transactions on Knowledge Discovery from Data(ACM TKDD), 2015, 10(2): 1-33.
- [3] MIN F, WU Y, WU X. The Apriori property of sequence pattern mining with wildcard gaps [J]. International Journal Functional Informatics and Personalised Medicine, 2010, 4(1):138-143.
- [4] DING B, LO D, HAN J, et al. Efficient mining of closed repetitive gapped subsequences from a sequence database[C] // International Conference on Data Engineering, IEEE Computer Socie-

- ty,2009:1024-1035.
- [5] FANG W W, XIE W, HUANG H B, et al. Sequential pattern mining based on privacy preserving [J]. Computer Science, 2016,43(12):195-199. (in Chinese) 方炜炜,谢伟,黄宏博,等. 基于隐私保护的序列模式挖掘[J]. 计算机科学,2016,43(12):195-199.
- [6] DONG G, LI J. Efficient mining of emerging patterns; Discovering trends and differences [C] // Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1999; 43-52.
- [7] GHOSH S,FENG M,NGUYEN H,et al. Risk prediction for acute hypotensive patients by using gap constrained sequential contrast patterns [C] // AMIA Annual Symposium Proceedings American Medical Informatics Association. 2014;1748-1754.
- [8] JI X, JAMES B, DONG G. Mining minimal distinguishing subsequence patterns with gap constraints [J]. Knowledge Information Systems, 2007, 11(3):259-286.
- [9] WANG X,DUAN L,DONG G, et al. Efficient mining of density-aware distinguishing sequential patterns with gap constraints
 [C] // 19th International Conference of Database Systems for Advanced Applications. 2014;372-387.
- [10] YANG H, DUAN L, HU B, et al. Mining top-k distinguishing sequential patterns with gap constraint [J]. Journal of Soft-

- ware, 2015, 26(11): 2994-3009. (in Chinese)
- 杨皓,段磊,胡斌,等.带间隔约束的 Top-k 对序列模式挖掘[J]. 软件学报,2015,26(11);2994-3009.
- [11] WANG H F, DUAN L, ZUO J, et al. Efficient mining of distinguishing sequential patterns without a predefined gap constraint [J]. Journal of Computer, 2016, 39 (10): 1979-1991. (in Chinese)
 - 王慧锋,段磊,左劼,等. 免预设间隔约束的对比序列模式高效挖掘[J]. 计算机学报,2016,39(10):1979-1991.
- [12] WU Y X, WU X D, JIANG H, et al. A heuristic algorithm for MPMGOOC [J]. Journal of Computers, 2011, 34(8): 1452-1462. (in Chinese) 武优西,吴信东,江贺,等. 一种求解 MPMGOOC 问题的启发式算法[J]. 计算机学报, 2011, 34(8): 1452-1462.
- [13] WU Y, TANG Z, JIANG H, et al. Approximate Pattern Matching with Gap Constraints[J]. Journal of Information Science, 2016, 42(5):639-658.
- [14] WU Y, FU S, JIANG H, et al. Strict approximate pattern matching with general gaps [J]. Applied Intelligence, 2015, 42(3): 566-580
- [15] WU Y, WANG L, REN J, et al. Mining sequential patterns with periodic wildcard gaps[J]. Applied Intelligence, 2014, 41(1):99-116.

(上接第 239 页)

- [5] ZHANG G, GAO L, SHI Y, et al. An Effective Genetic Algorithm for the Flexible Job-Shop Scheduling Problem [J]. Expert Systems with Application, 2011, 38(4): 3563-3573.
- [6] AL-HINAI N, ELMEKKAWY T Y. Robust and Stable Flexible Job Shop Scheduling with Random Machine Breakdowns Using a Hybrid Genetic Algorithm [J]. International Journal of Production Economics, 2011, 132(2):279-291.
- [7] WANG J, CHU K. An Application of Genetic Algorithms for the Flexible Job-Shop Scheduling Problem [J]. International Journal of Advancements in Computing Technology, 2012, 4(3):271-278.
- [8] YAZDANI M, AMIRI M, ZANDIEH M. Flexible Job-Shop Scheduling with Parallel Variable Neighborhood Search Algorithm [J]. Expert Systems with Application, 2010, 37(1):678-687.
- [9] DEFERSHA F M, MINGYUAN C. A Parallel Genetic Algorithm for a Flexible Job-Shop Scheduling Problem with Sequence Dependent Setups [J]. International Journal of Advanced Manufacturing Technology, 2010, 49(1/4):263-279.
- [10] KENNEDY J, EBERHART R C. A Discrete Binary Version of the Particle Swarm Algorithm [C] // IEEE International Conference on Systems. 2002:4104-4108.
- [11] CEHN J, PAN Q K. A Discrete Particle Swarm Optimization Algorithm for Independent Task Scheduling Problem [J]. Computer Engineering, 2008, 34(6):214-215, 218. (in Chinese)

- 陈晶,潘全科. 求解独立任务调度的离散粒子群优化算法[J]. 计算机工程,2008,34(6):214-215,218.
- [12] COELLO C A C, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256-279.
- [13] MONTGOMERY D C. Design and Analysis of Experiments [M]. New York: John Wiley & Sons, 2008.
- [14] TANG X X, HE W P, HE Y L, et al. Research and Implementation of Job Shop Scheduling System [J]. Aeronautical Manufacturing Technology, 2011(5):69-73. (in Chinese) 唐欣欣,何卫平,和延立,等. 面向作业车间的调度系统研究与实现[J]. 航空制造技术, 2011(5):69-73.
- [15] ZIAEE M. A Heuristic Algorithm for Solving Flexible Job Shop Scheduling Problem [J]. The International Journal of Advanced Manufacturing Technology, 2014, 71(1-4):519-528.
- [16] WOJCIECH B, MARIUSZ U, MIECZYSLAW W. Parallel Hybrid Metaheuristics for the Flexible Job Shop Problem [J]. Computers & Industrial Engineering, 2010, 59(2); 323-333.
- [17] ZHANG T N, HAN B, YU B, et al. Flexible Job Shop Scheduling Optimization with Production Capacity Constraints [J].

 System Engineering-Theory & Practice, 2011, 31(3): 505-511.

 (in Chinese)
 - 张铁男,韩兵,于渤,等.生产能力约束条件下的柔性作业车间调度优化[J].系统工程理论与实践,2011,31(3):505-511.