

# Set Packing 问题的研究进展<sup>\*</sup>)

马振宇 王建新 冯启龙 陈建二  
(中南大学信息科学与工程学院 长沙 410083)

**摘要** Set Packing 问题起源于分割问题的应用,是在强约束条件对元素进行划分。在复杂性理论中,此问题是一类重要的 NP 难问题,被广泛应用于调度、代码优化和生物信息学等领域。特别是在参数计算理论产生后,此问题再次成为研究的热点问题。依据所研究问题的差异,本文将 Set Packing 问题分成 5 类,并给出了具体的定义。在此基础上,分别介绍了求解这 5 类问题的相关算法,着重分析和比较了参数算法中所运用的各项技术,并提出了该问题算法研究的一些发展方向。

**关键词** Set Packing 问题, NP 难问题, 复杂性理论, 参数计算

## Algorithms for Set Packing: A Survey

MA Zhen-Yu WANG Jian-Xin FENG Qi-Long CHEN Jian-Er  
(School of Information Science and Engineering, Central South University, Changsha 410083)

**Abstract** Set packing problem is originated from the application of cutting problem which is to divide the elements under strong constraint condition. In complexity theory, set packing problems is an important NP-hard problem, which is used widely in the fields of scheduling and code optimization. Especially after the emerging of the parameterized computation theory, set packing problem becomes one of the most concerned research problem. According to the differences of the concerned problem, we divide the set packing problem into five classes and give the corresponding definition. This paper gives a detailed presentation of all the current algorithms for the problems of the five classes, and puts emphasis on the analysis and comparison of the technique used in those parameterized algorithms. At last, we present some development orientation in algorithm research for the set packing problem.

**Keywords** Set Packing problem, NP-hard problem, Complexity theory, Parameterized computation

## 1 引言

Set packing 问题是在理论和实践中基本的组合问题,是一类重要的 NP 难问题<sup>[1]</sup>,被广泛应用于调度、代码优化和生物信息学等领域。如在线性调度领域中,算法必须在不知道每个工作要多少时间完成的情况下调度许多的作业。而作业调度问题在解决网络中的连续媒体问题中有新的情况出现,一个客户在一个固定的时间间隔内提出资源请求,因此资源请求的控制算法必须决定是否满足该服务。这就导致了传统的作业调度问题的两种不同方式:第一种是在一个固定时间的调度代表了对用户的一个承诺,即用户对资源没有优先权;第二种是请求在固定点开始,固定点结束,因此不能提前或延迟调度。在单个机器上的间隔时间的在线调度问题,就是要使得资源利用最大化,也就是使接受的用户请求数最大化。所以,我们的问题能表示成在一个  $m$ -区间图上找最大独立集问题。单位段的  $m$ -区间图的特殊类别中的最大独立集问题就等同于  $m$ -set packing<sup>[2]</sup>。

人们长时间地对其算法进行研究,通过不断改进算法本身来降低时间复杂度,使之在实际应用中变为可能。在参数理论产生后,此问题再次成为研究的热点问题。近年来,大量的文章都在研究该问题的参数化算法,我们在此对其相关算法进行总结,希望人们能对此问题有更全面的了解,也有利于

我们在该问题上有更加深入的研究。

本文第 2 节介绍了 set packing 的定义以及由此衍生出来的一系列问题;第 3 节将分类介绍解决此类问题的算法,特别重点分析参数化 set packing 算法技术;最后介绍 set packing 问题算法研究的结论和展望。

## 2 问题的分类

Set packing 问题属于典型的 NP 难问题,其具体定义如下<sup>[3]</sup>:

**定义 2.1**(常规 set packing) 给定一个含有  $n$  个元组的集合  $S$  且符号全集  $\mu$  的大小为  $N$ ,目标是寻找一个最大子集  $S'$ ,使  $S'$  中的任何两个元组之间没有相同元素。

此类问题被称之为常规的 set packing 问题(GSP)。在 GSP 中各元组大小不一致,由此增加了求解的难度。为了便于分析问题,人们对此增加了约束条件,将各个元组的大小进行限制,我们将其称之为  $m$ -set packing 问题( $m$ -SP)<sup>[3]</sup>:

**定义 2.2**( $m$ -set packing) 给定一个含有  $n$  个元组的集合  $S$  且每个元组包含  $m$  个元素,目标是寻找一个最大子集  $S'$ ,使  $S'$  中的任何两个元组之间没有相同元素。

对于一些实际情况中,各个元组在整体中的作用可能不一样,因此将其赋予不同大小的权值,此类问题我们称之为带权  $m$ -set packing 问题( $m$ -WSP)<sup>[4]</sup>:

<sup>\*</sup>)国家自然科学基金重点项目;生物信息学中的相关组合理论和算法研究(60433020)。马振宇 硕士研究生,主要研究领域为参数计算、计算机理论;王建新 博士,教授,博士生导师,主要研究领域为计算机算法、网络优化理论、生物信息学;冯启龙 硕士研究生,主要研究领域为参数计算、计算机理论;陈建二 博士,长江学者特聘教授,博士生导师,主要研究领域为生物信息学、计算机理论、计算复杂性及优化。

**定义 2.3(带权 m-set packing)** 给定一个含有  $n$  个元组的集合  $S$ , 每个元组都带有实数权值且至多包含  $m$  个元素, 目标是寻找一个权值总和最大子集  $S'$ , 使  $S'$  中的任何两个元组之间没有相同元素。

以上问题大都通过近似算法、启发式算法来求解, 即使存在精确算法也都是以  $n$  为指数。当  $n$  较大时, 我们都认为几乎不可解。因此近年来, 人们将 set packing 问题的研究转移到对该问题的参数化求解。所谓参数化问题, 是从原问题出发, 从中找到一较小的参数  $k$ , 使得算法的时间复杂度主要与  $k$  有关。参数化 set packing 问题(PSP)<sup>[5]</sup> 的定义如下:

**定义 2.4(参数化 set packing)** 给定一个数对  $(S, k)$ ,  $S$  是含有  $n$  个元组的集合,  $k$  是一个整数, 目标是构造一个含有  $k$  个元组的 packing。如果找不到, 返回不存在。

PSP 属于  $W[1]$ -complete 问题<sup>[5]</sup>, 在参数计算中没有很好的算法来解决此类问题。为了便于参数化求解, 我们又引入参数化 m-set packing 问题(m-PSP)的概念<sup>[5]</sup>:

**定义 2.5(参数化 m-set packing)** 给定一个数对  $(S, k)$ ,  $S$  是含有  $n$  个元组的集合, 每个元组包含  $m$  个元素,  $k$  是一个整数, 目标是构造一个含有  $k$  个元组的 packing。如果找不到, 返回不存在。

m-PSP 是属于固定参数可解 FPT(fixed-parameter tractable)问题, 也就是说此类问题存在一个时间复杂度为  $f(k)n^{O(1)}$  的 FPT 算法<sup>[6]</sup>。当参数  $k$  较小时, 这种时间复杂度在现实操作中完全可以接受。当  $m=3$  的时候, 我们称为 3-PSP 问题, 这是参数计算中研究的热点问题。

### 3 相关算法

在这节中主要根据问题的分类详细介绍其相关的算法思想, 着重分析近年来在参数计算中此参数问题所用到的各类方法和技术。我们将上一节所定义的问题分为常规 set packing、m-set packing、带权 m-set packing 和参数化 m-set packing 4 大类分别进行介绍。

#### 3.1 常规 set packing

单独研究 GSP 问题的相关文献较少, 其主要原因是研究 GSP 算法就等同于研究最大独立集(maximum independent set)和最大团(maximum clique)等问题, 因为我们可以把 GSP 问题在多项式时间内转化成这些问题<sup>[1]</sup>。

GSP 问题到独立集问题的线性归约方法如下:

思想是将每个元组看作一个顶点, 相交的元组所对应的点之间两两连边。其具体方法是:

给定一个 GSP 实例, 有  $n$  个元组和  $N$  个元素。将  $n$  个元组看作  $n$  个点,  $N$  个元素也看作  $N$  个点, 先从每个元组出发, 与所包含的元素之间连边, 再从每个元素出发, 将其相连的点两两连边, 最后将  $N$  个元素的点以及相连的边去掉, 形成了图  $G$ 。其时间复杂度为  $O(nN^2)$ 。同样, 最大独立集和最大团只需要对图求补就可以转换。将 GSP 问题转化成最大团问题的思想同样是将每个元组看作一个顶点, 但只对不相交的元组所对应的点之间两两连边, 其时间复杂度仍然为  $O(nN^2)$ 。

关于最大独立集问题和最大团问题的相关算法已经有很长的一段研究历史<sup>[7~13]</sup>。对于最大独立集的精确算法有这样的过程: 首先由 Tarjan 提出一简单算法, 打破了  $O(2^n)$  的上界<sup>[14]</sup>; 1977 年, Tarjan 和 Trojanowski 又将其算法进行改进, 使时间复杂度降低到  $O(1.2599^n)$ <sup>[15]</sup>; 在 1986 年, Jian

给出在多项式空间复杂度的新算法, 将其时间复杂度降低到  $O(1.2346^n)$ <sup>[16]</sup>; 随后由 Robson 将时间复杂度降到  $O(1.2277^n)$ <sup>[17]</sup>; 最近 Fomin 通过用加权分治技术将最大独立集合问题的时间复杂度降至  $O(1.2209^n)$ <sup>[18]</sup>。这些结果都可以同样运用到 GSP 问题之中。

#### 3.2 m-set packing

对于 m-set packing 问题, 当  $m=2$  的时候, 可以在多项式时间内求解。具体做法是将元素看作点, 元组当作边, 这样就转换成在一般图中求最大匹配问题<sup>[19]</sup>。

当  $m \geq 3$  时, 是属于 NP-hard 问题, 我们可以将其转换成最大独立集或最大团来求解, 也有相关的近似算法。直接求解 m-set packing 问题有一种简单的贪婪算法<sup>[22]</sup>: 重复地任意选取一元组放在目标集合中, 且保证在目标集合中的元组相互不相交。每放入一元组, 就从全集中将该元组和与之相交的元组移去, 直到全集变为空。很明显, 该算法返回一组互不相交的子集, 很容易证明该算法的近似率为  $m$ 。Hurkens 和 Schrijver<sup>[20]</sup> 给出一种局部搜索的方法, 得到近似率为  $m/2 + \epsilon$ , 这里的  $\epsilon > 0$ , 和改变集的大小有关。这是对于不带权值的 m-set packing 问题的最佳结果。Trevisan<sup>[21]</sup> 给出此类问题的近似算法, 近似率下界为  $\frac{m}{2\Omega(\sqrt{\ln m})}$  的证明, Hazan<sup>[22]</sup> 将其改进为  $\Omega(\frac{m}{\ln m})$ 。

#### 3.3 带权 set packing

求解带权 m-set packing 问题的最普通的启发式算法是用贪婪算法<sup>[25]</sup>: 把权值最大的元组添加在结果中, 再移去该元组和与之相交的元组。如此重复, 直到所有的元组被移去。可以很容易看出, 这结果的近似率为  $m$ 。还有一种常见的方法是局部搜索法: 试图用不在结果中的权值更大的元组去替换结果中小的元组。这样的搜索可在多项式时间内完成, 但它受到一些局限, 如增加的元组数目或移去的元组数目应为常数。

Hassin<sup>[23]</sup> 和 Bafna 等人<sup>[24]</sup> 都对带权情况的局部搜索算法进行了分析, 得到了相同的近似率  $m-1+\epsilon$ , 两者之间的区别是 Bafna 等人提出  $\epsilon=1/m$ , 而 Hassin 则证明  $\epsilon$  可为任意小的固定值。

Barun Chandra<sup>[25]</sup> 针对带权的 m-set packing 问题提出一种结合上述两种方法的近似算法 BESTIMP, 首先通过贪婪算法得到最初结果, 再经过反复的局部改进, 得到近似率为  $2(m+1)/3$ 。另一种算法为  $A_{NYIMP}$ , 也是结合贪婪和局部搜索, 不同之处在于  $A_{NYIMP}$  只找导致得到比指定阈值更大的改进而不是找最好的改进, 其近似率不超过  $(4m+2)/5$ 。Berman<sup>[26]</sup> 得到近似率为  $(m+1)/2$  的近似算法, 他的方法也是基于局部改进, 参考指标是相关点权值平方之和而不是简单的权值之和。

#### 3.4 参数化 m-set packing

到目前为止, 我们通常认为, 求解一个 NP-hard 问题是不存在一个多项式时间的确定性算法。然而, 这些问题又频繁地出现在我们实际应用中。为解决此类问题, 人们通常采用近似算法来得到一个近似解。在某些具体情况下, 这些解并不能满足人们的要求。因此, 近年来, 参数理论<sup>[6]</sup> 在这一领域中发展十分迅猛。现在, 在参数计算中我们通常都是考虑 FPT 问题, 对其设计出有效的 FPT 算法。

之所以未把参数化 set packing 问题单独拿出来介绍, 是因为对于 PSP 问题而言, 用最简单直接的参数算法可得时间

复杂度为  $O(n^k N)$ , 其中  $N$  是元素的数目。这样的算法在指数上虽然是  $k$ , 但其底数过大, 其实并不实用, 即便是在  $k$  值很小的时候 (如  $k=10$ )。所以就 PSP 问题而言, 它本身不是 FPT 问题, 而属于  $W[1]$ -complete<sup>[5]</sup>, 这也意味着没有一个更好的算法来解决这样的参数问题。而对于加了约束条件的参数化  $m$ -set packing 问题而言, 是属于 FPT 问题<sup>[5]</sup>, 特别是参数化 3-set packing 问题是当前研究的热点问题。

表 1 中给出了 3-set packing 问题参数化算法的研究进展。目前, 关于 3-set packing 问题参数化随机算法的最好结果是文[32]提出的时间复杂度为  $O^*(2.52^{3k})$  的算法, 确定性算法的最好结果是文[34]提出的时间复杂度为  $O^*(4.61^{3k})$  的算法。在设计 FPT 算法中存在很多相关的技术<sup>[6]</sup>, 如: 核心化、局部贪婪、动态规划、着色法等。下面我们就逐一将这些技术在参数化 3-set packing 中的应用进行详细的分析。

表 1 3-SET PACKING 问题的算法比较

文献	随机算法	确定性算法	相关技术
[27, 28]		$O^*((5.7k)^k)$	局部贪婪
[29]	$O^*(10.88^{3k})$	$>O^*(32000^{3k})$	代数法、着色法
[30]		$>O^*(12.67^{3k}T(k))$	核心化、着色法
[31]	$O^*(2.52^{3k})$	$O^*(16^{3k})$	分治法、着色法
[32]	$O^*(2.52^{3k})$	$O^*(12.8^{3k})$	着色法、分治法
[34]		$O^*(4.61^{3k})$	着色法、动态规划

从表 1 中我们可以清楚看到, 很多算法中运用了多种技术的结合, 我们主要根据该算法在哪些关键技术上的突破来分为以下 5 部分。

### 3.3.1 局部贪婪法

文[27, 28]首先通过在时间复杂度  $O(mn)$  内构建  $k$ -maximal set packing  $M_0$ , 当  $|M_0| \geq k$  或者  $|M_0| \leq k/m$ , 可直接返回“yes”或者“no”, 剩下只需要对  $k/m \leq |M_0| \leq k$  情况进行分析。若存在一个  $k$  大小的 set packing, 则该 set packing 中的每个元组至少包含  $M_0$  中的一个元素。接下来可用穷举法找出  $k$  个元素, 进而形成  $k$  个偏集, 偏集表示在该集合中有不超过  $k-1$  个元素用“\*”表示。

为了使穷举更加有效, 并引入生成集概念:

存在 set packing  $P$  和集合  $\beta$ , 若  $P$  中每个元组中都有元素在集合  $\beta$  中, 则  $\beta$  是  $P$  的一个生成集。

生成集的数目可通过递归式来求得。假设  $|M_0|=h$ , 设  $D(h, k)$  为  $M_0$  中包含  $k$  个元素的生成集的数目, 有以下递归式:

$$\begin{aligned} \text{如果 } h > k \text{ 或者 } mh < k \text{ 或者 } k \leq 0, & \quad D(h, k) = 0; \\ \text{如果 } mh = k, & \quad D(h, k) = 1. \end{aligned}$$

$$\text{其中 } D(h, k) = \sum_{i=1}^m \binom{m}{i} D(h-1, k-1) \leq \sum_{i=1}^m \binom{m}{i} b_m^{k-i} = b_m^k$$

偏集中其他的元素通过局部贪婪法进行猜点替换。若进行了一部分后无法继续, 则需要对新加入的元素进行穷举后再用局部贪婪法。这部分的时间复杂度通过 Stirling's 公式近似后, 这一部分操作的时间复杂度为  $O(k(m-1)(k^{m-2}(m-1)^{m-1}/e^{m-2})^k + [k(m-2)+1](k^3 + kmn))$ 。经过以上 3 步, 得到其整个算法的总时间复杂度为  $O(b_m^k [k(m-1)(k^{m-2}(m-1)^{m-1}/e^{m-2})^k + [k(m-2)+1](k^3 + kmn)] + mn)$  特别是在  $m=3$  时, 也就是 3-set packing 问题, 时间复杂度为  $O^*((5.7k)^k n) (b_3 < 3.8474)$ 。

### 3.3.2 代数法

其主要思想是将 set packing 的实例转化成代数表达式来求解<sup>[29]</sup>。如: 存在  $C = \{S_1, \dots, S_n\}, U = \bigcup S_i$  且  $N = |U|, f(S_i)$  是表示元组  $S_i$  中所包含的元素, 令  $P_k = (\sum_{S_i \in C} f(S_i))^k, P_k$  实质上是一代数多项式的  $k$  次方, 而该代数多项式中的每个因子其实是 set packing 中对应元组中的元素。通过  $k$  次方后, 若其表达式中仍有因子中所有元素的幂指数都为 1, 则表明至少存在一个  $k$  大小的 set packing, 其时间复杂度的分析是针对代数式, 为  $O(knN^2 2^N)$ 。这是对 set packing 问题是否存在  $k$  个大小的 set packing 的判断。若要找出对应的  $k$  个元组, 需对  $n$  个元组逐一进行判断, 因此总时间复杂度为  $O(kn^2 N^2 2^N)$ , 可以推得 3-set packing 的确定性算法的时间复杂度至少为  $O^*(32000^{3k})$ 。

### 3.3.3 核心化法

3-PSP 和 3D-matching 问题在定义和求解方法上都十分类似, 只是约束条件上略有不同。二者都属于 FPT 问题, 因此也可对其进行核心化。

文[30]给出 3D-matching 核心化算法, 其算法思想是: 用一些特殊的模式三元组与  $S$  中的所有三元组来匹配。如果匹配的话, 就丢掉一部分三元组, 保留的三元组的阈值由两个函数  $f_1$  和  $f_2$  来决定,  $f_1$  是  $k$  的线性函数,  $f_2$  是  $k$  的二次方函数,  $f_1$  和  $f_2$  的大小证明详见文[29]。在执行完核心化算法之后, 可以分析出  $S_0$  中的任意一个符号最多只能包含在  $f_2$  个三元组中。那么, 如果  $S$  存在一个最大的匹配, 匹配大小为  $k$ , 也就是有  $3k$  个符号, 每个符号最多出现在  $f_2$  个三元组中。故经核心化后,  $S$  最多可包含  $3k \times f_2 = O(k^3)$  个三元组, 也就是问题的核为  $O(k^3)$ 。同样的算法也适用于 3-PSP 问题, 也就是说通过核心化算法可以将一输入规模为  $n$  的问题转化成一输入规模为  $k^3$  的问题, 当  $k$  值较小时, 极大地减少了问题的大小。

### 3.3.4 分治法

文[31]中提出求解图论问题中的一项新技术, 该技术将分治法和着色法相结合 (divide and color)。这里所用的着色法并非是直接着  $k$  种颜色, 而是为了将图进行划分, 仅仅是对图着黑和白两种颜色。通过这种着色法后, 将问题划分成两个子问题, 再在这两个子问题上用同样的方法处理, 直至无法划分为止。用此方法来求解 H-graph edge-packing 问题, 对图中的所有边上着两种不同的颜色, 再根据颜色来划分为两个子问题。之所以用着色法, 是可以通过着色方案使将不确定算法来设计出确定算法。整个算法的时间复杂度主要是着色方案的时间复杂度和分治法的时间复杂度的乘积, 最后得到  $O^*(16^{3k})$ 。同样的算法也可以运用到 3-set packing 中, 其时间复杂度为  $O^*(16^{3k})$ 。

在文[32]中提出了一种随机的分治法。设在全集  $S$  中寻找  $k$  个元组标记为  $S_k$ 。首先用随机法将  $S_k$  划分为相同大小地两部分, 再循环地从每部分中寻找  $k/2$  元组。对于 3-set packing 而言, 若  $S_k$  存在, 设  $A_k$  是包含  $3^k$  个元素的子集, 在  $S_k$  中, 我们可以将  $A_k$  划分为  $A_k'$  和  $A_k''$  两部分并使得正好  $A_k'$  包含  $3/2k$  个元素  $k/2$  个元组和  $A_k''$  也包含  $3/2k$  个元素  $k/2$  个元组, 这样的概率有  $\binom{k}{k/2} / 2^{3k} = O(1/(2^{2k}\sqrt{k}))$ 。对这两部分再循环细分, 最后得到其随机算法的时间复杂度为  $O(2.52^{3k} n) = O^*(2.52^{3k})$ 。

### 3.3.5 着色法

着色技术被充分利用来求解参数问题。着色技术是 Alon 提出来的<sup>[33]</sup>,主要是利用完美哈希函数来随机化,得出了  $k$  种颜色的着色下界为  $2^{O(k)E}$  (有向图)、 $2^{O(k)V}$  (无向图)。文[30]将着色法应用到 3-D matching 问题上,具体算法是:给定集合  $S$  包含  $n$  个三元组,如果有包含  $k$  个不相交的三元组存在,我们对这  $k$  个不相交的三元组着  $3k$  种颜色,使得其中  $3k$  个元素被着成不同的颜色。对于  $3k$  种颜色,所需哈希函数的大小为  $O(2^{O(3k)})$ 。着色后用动态规划的方法找出  $k$  个不相交的元组,动态规划所需计算的所有子问题的大小也就为  $O(2^{O(3k)})$ 。整个问题的算法时间复杂度就是为核心化的构造的时间和动态规划的时间复杂度两部分,核心化的构造的时间是  $O(n)$ ,动态规划的时间复杂度是  $O(2^{O(3k)})$ ,所以总的复杂度是  $O(n + O(2^{O(3k)}))$ 。这里的  $O(3k)$  给得并不具体,按推导来算的话,至少为  $O^*(12.67^{3k} T(k))$  ( $T(k)$  是指动态规划的时间,按现有的最佳算法至少为  $O^*(10.4^{3k})$ )。

文[29]使用着色法给出了 set packing 的随机算法,但其针对的问题并非传统意义上的参数 set packing,而是对  $k$ -PSP 问题又加了约束条件: $k$  个不相交的元组中包含的元素的总数为  $t$ ,文中将其称之为  $(k, t)$ -set packing。对于当  $t=3k$  时,实质上可以看作是 3-set packing 问题。随机算法描述如下:采用随机的映射,从  $(n, N)$  到  $(k, t)$ 。 $N$  代表所有不同元素的个数, $n$  代表由这  $N$  个不同元素所组成的所有的元组。希望从  $N$  个不同元素中取出  $t$  个不同元素,从  $n$  个元组中选出  $k$  个不交的元组,而且这  $t$  个不同元素必须构成  $k$  个不交的元组。使用着色法来做,正确着色(就是说能确保找出至少一个存在的  $k$ -Set-Packing 的一种着色)的概率是  $t! / t^t$  (第一次有  $t$  种可用的颜色,第二次只有  $t-1$  种可用的颜色,……而每个元素可以被  $t$  种颜色中的任意一种着色)。对  $n$  个元组,以及对  $N$  个不同的元素,用  $t$  种颜色着色,构造一个着色方案的最坏时间复杂度为  $O(tnN)$ 。考虑含  $t$  个变量的多重线性多项式  $Q_1, Q_2$ ,这种多项式最多含  $2^t$  个项。所以  $Q_1$  乘以  $Q_2$  最多含  $2^{2t}$ 。因此  $M(Q_1 Q_2)$  可以在  $O(t^{2t})$  时间内计算出来。然后采用前面的算法,我们可以在  $O(k(2^t)^2)$  时间内构造一个 Set-Packing。为了确保构造出  $k$  个不相交的 Set-Packing,我们必须运行随机算法  $e^k$  次。最后,总的复杂度为  $O(e^k(tnN + t^{2t} \log k))$ ,大致约为  $O^*(10.88^{3k})$ 。

文[32]系统地研究了  $k$  着色方案的时间复杂度,给出了  $k$  种颜色的着色方案的下界为  $\Omega(2.718^k)$  的证明,且提出一种新的 4 层哈希的方法,用到了核心化、最小冲突和反复构建技术,该方法将  $k$  着色方案的上界降低为  $O(6.4^k n)$ 。通过在着色方案上的突破,直接降低了用着色法和动态规划求解 3-set packing 问题的参数算法的时间复杂度。具体方法是:给定集合  $S$  包含  $n$  个元组,如果有包含  $k$  个不相交元组的子集  $S_k$  存在,我们对集合  $S$  中的元素着  $3k$  种颜色,使得  $S_k$  中的  $3k$  个元素被着成不同的颜色,对于  $3k$  的着色方案的时间复杂度为  $O(6.4^{3k} n)$ ;再用动态规划的方法在时间复杂度为  $O(2^{3k} n)$  找出  $k$  个不相交的元组。整个算法的时间复杂度就是其着色方案的时间复杂度和动态规划的时间复杂度的乘积,因此总的复杂度为  $O(6.4^{3k} n) * O(2^{3k} n) = O(12.8^{3k} n^2) = O^*(12.8^{3k})$ 。

文[34]给出了运用着色法最好的时间复杂度上界  $O^*(6.1^k)$ ,且证明了 3-set packing 与 3-set packing augmentation 具有相同的时间复杂度,故降低 3-set packing augmentation 的时间复杂度是我们的目标。通过运用着色法和动态规划技

术,对 3-set packing augmentation 给出了一个有效的算法,时间复杂度为  $O^*(4.61^{3k})$ 。

首先引入文[34]中一个重要的定理:已知  $S$  和  $S$  的一个 packing  $P_k$ ,如果  $S$  中存在  $P_{k-1}$ ,则有一个  $P_{k-1}$  满足下列条件: $P_k$  中的每个元组至少有两个元素在  $P_{k-1}$  中。

在解决 3-set packing augmentation 问题时,利用上述定理,在  $P_{k-1}$  中只要再寻找  $k+3$  个元素,再加上  $P_k$  中的  $2k$  个元素,就可确定  $P_{k-1}$ 。作者运用着色法来寻找这  $k+3$  个元素,得到  $O^*(6.1^k)$  的着色机制,然后用另外  $3k$  种颜色对  $P_k$  中的元素进行着色,把上面得到的着色机制  $f$ ,加上  $P_k$  中的  $3k$  种颜色,这样就得到  $4k+3$  的着色机制。再运用动态规划的方法寻找  $P_{k-1}$ ,如果  $P_{k-1}$  存在的话,算法就可返回  $k+1$  个元组,这些元组中的元素都着了不同的颜色,时间复杂度为  $O^*(2^{4k})$ 。最后,该算法总的复杂度为  $O^*(6.1^k) * O^*(2^{4k}) = O^*(4.61^{3k})$ 。

**结论与进一步研究** set packing 是一类重要的 NP-hard 问题,并且在众多领域都有其应用。从本文中我们可归纳出求解此类问题的方法大致分为以下三类:一将其转换成最大独立集或最大团来求解。无论算法如何改进,其时间复杂度中总有个  $n$  的指数。通常我们认为  $n$  是个很大的数,因此这种方法几乎无法在实际应用中执行。二是在多项式时间内求其近似解,但我们可发现其相关的近似算法的近似率并不理想。三是以参数算法求解。现 3-set packing 问题最好的 FPT 算法的时间复杂度为  $O^*(4.61^{3k})$ ,当  $k$  是个较小数时,其时间复杂度在实际操作中完全可以接受。因此随着参数计算理论的产生,更推动了这一问题的发展,使之成为近阶段的热点研究问题。对 set packing 问题进行综述,不仅仅是让人们对此问题有更全面的了解,最主要的目的是希望通过这样的工作使我们能对该问题有更深入的研究。近阶段来,我们一直致力于在这一问题上取得新的突破,具体包括以下三个方面:

(1) GSP 问题通常可以转化成独立集求解,但在实际转换过程中只考虑了相交元组的因素,而忽略了各元组中所包含元素个数不同这一事实。我们希望在算法中引入元素,使其在某种特定条件下能降低时间复杂度;

(2) 最近参数理论中又产生了固定参数可枚举这一新概念<sup>[35]</sup>。所谓固定参数可枚举问题(fixed-parameter enumerability)就是该问题首先是属于 FPT 问题,且在时间复杂度  $O(f(k)n^{O(1)}z^{O(1)})$  内可枚举出  $z$  个最优结果。对于带权的 3-set packing 问题,我们想证明是 FPT 问题,并希望设计一个有效的固定参数可枚举的算法,能枚举出  $z$  个权值之和最大的 set packing;

(3) 我们可以发现以前 3D matching 问题和 3-set packing 问题的算法几乎相同且时间复杂度一致。最近的文[33]中对于这两类问题分别给出了不同的算法。3D matching 问题由于更强的约束条件使得其着色范围更小,从而得到更好的时间复杂度  $O^*(2.77^{3k})$ 。我们希望通过这种思想,对 3-set packing 问题进行适当的预处理和约束,目的也是为了减少其着色范围,使得时间复杂度更接近 3D matching。

## 参考文献

- 1 Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W. H. Freeman and Company, 1979

(下转第 22 页)

- gan. In: Proceedings of the Fourth Conference on Creativity and Cognition. New York: ACM Press, 2002. 20~24
- 41 Taylor R M, Hudson T C, Seeger A, et al. VRPN: A Device-Independent, Network Transparent VR Peripheral System. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, New York: ACM Press, 2001. 55~61
- 42 Torres D, Boulanger P. A Perception and Selective Attention System for Synthetic Creatures. In: Proceedings of the Third International Symposium On Smart Graphics, 2003. 141~150
- 43 Cycling '74. Max/MSP. <http://www.cycling74.com/products/maxmsp>
- 44 Torres D, Boulanger P. The ANIMUS Project: A Framework for the Creation of Interactive Creatures in Immersed Environments. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, New York: ACM Press, 2003. 91~99
- 45 Torres D. The ANIMUS Project: A Framework for the Creation of Emotional and Rational Social Agents. <http://www.cs.ualberta.ca/~dtorres/projects/animus/docs/proposal.pdf>
- 46 Taylor R, Torres D, Boulanger P. Using Music to Interact with a Virtual Character. In: Proceedings of New Interfaces for Musical Expression, Vancouver, BC, Canada, 2005. 220~223
- 47 Taylor R, Boulanger P, Torres D. Visualizing Emotion in Musical Performance Using a Virtual Character. In: 5th International Symposium on Smart Graphics, Munich, Germany, 2005. 13~24
- 48 McDonnell M. Visual Music. 2003. [http://www.soundingvisual.com/visualmusic/visualmusic2003\\_2004.pdf](http://www.soundingvisual.com/visualmusic/visualmusic2003_2004.pdf)
- 49 Baumann S. Visualization for Music IR. 2005. <http://ismir2005.ismir.net/documents/Baumann-ISMIR05Tutorial.pdf>
- 50 Chew E. Foreword to special issue on music visualization. Computers in Entertainment, New York: ACM Press, 2005, 3(4): 1~3
- 51 Winzenried R. Hear the Colors, See the Music. 2006. <http://www.symphony.org/news/room/06jfl-articles/currents.pdf>

## (上接第 15 页)

- 2 Bar-Yehuda R, Halldórsson M, Naor J, et al. Scheduling split intervals. In: Proc. Thirteenth Annu. ACM-SIAM Symp. On Discrete Algorithms, 2002. 732~741
- 3 Crescenzi P, Kann A V. Compendium of NP Optimization Problems. <http://www.nada.kth.se/viggo/problemist/compendium.html>, 1998
- 4 Arkin E M, Hassin R. On local search for weighted k-set packing. In: Proc. 5th Ann. European Symp. on Algorithms, Lecture Notes in Comput. Sci. 1284, Springer-Verlag, 1997. 13~22
- 5 Cesati M. Compendium of Parameterized Problems. <http://citeseer.ist.psu.edu/cesati01compendium.html>, 2001
- 6 Chen J. Parameterized computation and complexity: a new approach dealing with NP-hardness. Journal of Computer Science and Technology, 2005, 20: 18~37
- 7 Wigderson A. Improving the performance guarantee for approximate graph coloring. Journal of the Association for Computing Machinery, 1983, 30(4): 729~735
- 8 Bar-Yehuda R, Moran S. On approximation problems related to the independent set and vertex cover problems. Discrete Applied Mathematics, 1984, 9: 1~10
- 9 Boppana R, Halldórsson M M. Approximating maximum independent sets by excluding subgraphs. Bit 32, 1992. 180~196
- 10 Hastad J. Clique is hard to approximate within  $n^{1-\epsilon}$ . Acta Math, 1999, 182(1): 105~142
- 11 Feige U, Goldwasser S, Lov'asz L, et al. Interactive proofs and the hardness of approximating cliques. J ACM, 1996, 43(2): 268~292
- 12 Arora S, Safra S. Probabilistic checking of proofs: A new characterization of NP. Journal of the ACM, 1998, 45: 70~122
- 13 Arora S, Lund C, Motwani R, et al. Proof verification and intractability of approximation problems. Journal of the ACM, 1998, 45: 501~555
- 14 Tarjan R. Finding a maximum clique: [Technical Report]. 72-123. Ithaca, NY: Computer Sci. Dept, Cornell Univ, 1972
- 15 Tarjan R, Trojanowski A. Finding a maximum independent set. SIAM Journal on Computing, 1997, 6(3): 537~546
- 16 Jian T. An  $O(2.304n)$  algorithm for solving maximum independent set problem. IEEE Transactions on Computers, 1986, 35(9): 847~851
- 17 Robson J M. Algorithms for maximum independent sets. Journal of Algorithms, 1986, 7(3): 425~440
- 18 Fomin F V, Grandoni F, Kratsch D. Measure and conquer: a simple  $O(2^{2.288n})$  independent set algorithm. In: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm, 2006. 18~25
- 19 Gabow H N. An efficient implementation of Edmonds' algorithm for maximum matching on graphs. Journal of ACM, 1976, 23: 221~234
- 20 Hurkens C A J, Schrijver A. On the size of systems of sets every  $t$  of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. SIAM Journal Discrete Math, 1989, 2: 68~72
- 21 Trevisan L. Non-approximability results for optimization problems on bounded degree instances. In: Proc. 33rd ACM Symp. on Theory of Computing, 2001
- 22 Hazan E, Safra S, Schwartz O. On the Complexity of Approximating k-Set Packing. Computational Complexity, 2006. 20~39
- 23 Arkin E, Hassin R. Approximating weighted set packing by local search. Math Oper Res, 1998. 640~648
- 24 Bafna V, Narayan B, Ravi R. Nonoverlapping local alignments (Weighted independent sets of axis-parallel rectangles). Discrete Appl Math, 1996: 41~53
- 25 Chandra B, Halldórsson M M. Greedy Local Improvement and Weighted Set Packing Approximation. Journal of Algorithms, 2001. 223~240
- 26 Berman P. A  $d/2$  approximation for maximum weight independent set in  $d$ -claw free graphs. In: Proceedings, Seventh Scandinavian Workshop on Algebraic Theory (SWAT), Lecture Notes in Computer Science, Vol 1851. New York/Berlin, Springer-Verlag, July 2000. 214~219
- 27 Jia W, Zhang C, Chen J. An efficient parameterized algorithm for  $m$ -set packing. J. Algorithms, 2004, 50: 106~117
- 28 Chen J, Friesen D K, et al. Using nondeterminism to design deterministic algorithms. Algorithmica, 2004, 40: 83~97
- 29 Koutis I. A faster parameterized algorithm for set packing. Information Processing Letters, 2005, 94: 7~9
- 30 Fellows M R, Knauer C, Nishimura N, et al. Faster fixed-parameter tractable algorithms for matching and packing problems. Lecture Notes in Computer Science 3221 (ESA 2004), 2004: 311~322
- 31 Kneis J, Molle D, Richter S, et al. Divide-and-color. In: Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2006), LNCS 4271. Springer, 2006
- 32 Chen J, Lu S, Sze S-H, et al. Improved algorithms for path, matching, and packing problems. In: Proc. 18th ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), 2007
- 33 Alon N, Yuster R, Zwick U. Color-coding. Journal of the ACM, 1995, 42: 844~856
- 34 Liu Y, Lu S, et al. Greedy localization and color-coding: improved matching and packing algorithms. In: Proc. 2nd International Workshop on Parameterized and Exact Computation, Lecture Notes in Computer Science 4169, 2006. 84~95
- 35 Chen J, Kanj I, Meng J, et al. On the effective enumerability of NP problems. In: Proc. 2nd International Workshop on Parameterized and Exact Computation (IWPEC '06), Lecture Notes in Computer Science 4169, 2006. 215~226