3-维匹配问题的一种固定参数枚举算法

刘运龙[1,2] 王建新[1]
(中南大学信息科学与工程学院 长沙 410083)[1] (湖南师范大学继续教育学院 长沙 410013)[2]

摘 要 枚举问题的多个最优解是计算机科学中人们日益关注的一个研究方向。运用固定参数枚举理论和着色技术对3-维匹配问题提出了一个高效的固定参数枚举算法,即给定一个含有n个带权值的元组集合S,两个非负整数k和z,该算法能在时间$O(5.48^{3k}kn^2z)$内枚举出S中权值最大的z个k-matchings,进而表明了3-维匹配问题是固定参数线性可枚举的。

关键词 3-维匹配问题,固定参数枚举,着色技术
中图法分类号 TP301 **文献标识码** A

Efficient Fixed-parameter Enumeration Algorithm for the 3-D Matching Problem

LIU Yun-long[1,2] WANG Jian-xin[1]
(School of Information Science and Engineering, Central South University, Changsha 410083, China)[1]
(School of Continuing Education, Hunan Normal University, Changsha 410013, China)[2]

Abstract Enumerating a number of good solutions to a problem has an increasing demand in recented research in computational science. In this paper, we presented a fixed-parameter enumeration algorithm for the 3-D Matching problem. More precisely, we developed an algorithm that, on a given set S of n weighted triples and two integers k and z, produces z "best" k-matchings in time $O(5.48^{3k}kn^2z)$. Our algorithm is based on the recent improved color-coding techniques and the fixed-parameter enumeration framework. This result shows that the 3-D Matching problem is fixed parameter linearly enumerable.

Keywords 3-D Matching, Fixed-parameter enumerable, Color-coding

1 引言

枚举是计算复杂性理论中的一类基本运算,是理论计算机科学中一个重要分支。自从1979年Valiant首次提出枚举运算以来[1],人们对枚举运算进行了大量的研究[2,3],其中最典型的枚举方法有枚举问题实例的所有解或统计问题实例中解的个数。然而,枚举(或统计)问题实例的所有解在许多情况下是难解的,因此这些方法不能很好地满足实际需要[4]。随着参数计算理论的发展,文献[4]最近提出了一种新的枚举方法,即固定参数枚举。

固定参数枚举就是枚举NP优化问题实例中一定数量的最优参数解,同时根据解的个数来研究其平均时间复杂度。该方法不仅在理论上是一种新的枚举技术,而且实际有效地求解了一些用传统枚举方法难以求解的问题。例如,在一个图上统计所有长度为k的路径是#W[1]完全的,但枚举其中z条($z>1$)长度为k的最优路径却是固定参数可解的[4]。

本文运用固定参数枚举方法研究3-维匹配枚举问题。具体来说,对于一个3-维匹配问题的实例,本文将枚举它的多个最优参数解,并根据解的个数研究其平均时间复杂度。下面给出该问题的相关定义。

设X,Y,Z为3个互不相交的有限符号集,如果$x\in X$,$y\in Y$,$z\in Z$,$\rho=(x,y,z)$,则称ρ为$X\times Y\times Z$中的一个三元组。设$S\subseteq X\times Y\times Z$,$M$是$S$的一个子集,如果$M$中任何两个元组之间都没有相同符号,则称$M$是$S$中的一个匹配(matching)。$M$的大小等于$M$中所含元组的个数。一个匹配如果正好含有$k$个元组,则称之为一个$k$-匹配,本文用$k$-matching表示。

定义1(参数化) 3-维匹配[5]:给定一个组对(S,k),其中S是一个三元组集合,k是一个非负整数,目标为判定S中是否存在一个k-matching。

3-维匹配问题是6个经典的NP完全问题之一。近年来,人们运用参数计算技术对参数化3-维匹配问题进行了一系列的研究[5,6]。

在参数复杂性理论中,参数k通常远远小于实例的大小n。因此在一个3-维匹配问题的实例中通常存在多个k-matchings。对于任意一个含有n个元组的集合S,枚举S中所有的k-matchings目前还不存在有效的算法[7]。本文运用固定参数枚举的方法来枚举其中多个k-matchings。为便于

到稿日期:2009-07-27 返修日期:2009-10-05 本文受国家973前期研究专项课题"信息处理中的参数算法研究"(No. 2008CB317107),国家自然科学基金项目"参数计算理论及应用"(60773111)资助。
刘运龙 博士生,主要研究方向为计算机算法、参数计算等,E-mail: hnsdlyl@163.com;王建新 博士,教授,博士生导师,主要研究方向为计算机算法、网络优化理论和生物信息学等。

区分不同 k-matching 的优先顺序，假定 S 中每一个元组都带有一个非负权值。一个 k-matching 的权值等于该 matching 中 k 个元组的权值之和。下面给出该问题的正式定义。

定义 2(固定参数枚举) 3-维匹配：给定一个组对 (S,k,z)，其中 S 为一个含有 n 个带权值的元组集合，k 和 z 为两个非负整数，目标为枚举 S 中权值最大的 z 个 k-matchings。

2 相关定义和引理

下面首先给出一些相关术语和定义。

对于 S 中的任意一个三元组 $\rho=(x,y,z)$，本文给出如下表示方法：$Val(\rho)=\{x,y,z\}$，$\mathrm{Val}^1(\rho)=\{x\}$，$\mathrm{Val}^2(\rho)=\{y\}$，$\mathrm{Val}^3(\rho)=\{z\}$，$\mathrm{Val}(S)=\bigcup\rho\in{}_S\mathrm{Val}(\rho)$，$\mathrm{Val}^i(S)=\bigcup\rho\in{}_S\mathrm{Val}^i(\rho)$，$i=1,2,3$。其中 $\mathrm{Val}^i(S)$ 也叫做 S 的第 i 列元素，$i=1,2,3$。

定义 3[4] 设 (I,k) 为 NP 优化问题的任意一个实例，其可行解集为 $S_Q(I,k)$，并设该解集中的所有解在一个函数 f 的映射下其函数值可以按非增(或非减)的顺序排列为 $\tau_{I,k}=\{f_Q(I,k,y_1),f_Q(I,k,y_2),\cdots\}$，如果 z 个解 $\{y_1',\cdots,y_z'\}$ 的函数值 $f_Q(I,k,y_1'),\cdots,f_Q(I,k,y_z')$ 排序后正好与列表 $\tau_{I,k}$ 中前 z 个值排列顺序一致，则称 $\{y_1',\cdots,y_z'\}$ 是实例 (I,k) 解集 $S_Q(I,k)$ 中最大(或最小)的 z 个解。

定义 4[4] 对于一个参数化 NP 优化问题 Q，如果存在两个算法 A_1 和 A_2 并且满足如下条件：(1)对于给定问题 Q 的任意一个实例 (I,k)，算法 A_1 能在时间 $f(k)n^{O(1)}$ 内产生一个关于解的结构 $\pi_{I,k}$，其中 f 是独立于 $n=|I|$ 的一个递归函数；(2)对于给定的结构 $\pi_{I,k}$ 和一个整数 $z\geqslant0$，算法 A_2 能在时间 $O(|\pi_{I,k}|^{O(1)}z^{O(1)})$ 内产生问题实例 (I,k) 的 z 个最优的解，则称问题 Q 是固定参数可枚举的(Fixed-parameter enumerable，简称 FPE)，其中算法 A_1 称为结构算法，算法 A_2 称为枚举算法。特别地，如果枚举算法 A_2 的运行时间为 $O(|\pi_{I,k}|^{O(1)}z)$，则称该个 NP 优化问题是固定参数线性可枚举的①。

从上述定义可知，由于算法 A_1 的运行时间是 $f(k)n^{O(1)}$，结构 $\pi_{I,k}$ 的大小 $|\pi_{I,k}|$ 有界于 $f(k)n^{O(1)}$，因此枚举算法 A_2 的运行时间有界于 $f_1(k)n^{O(1)}z^{O(1)}$，其中 f_1 是独立于 $n=|I|$ 的一个递归函数。

着色技术是本文中结构算法用到的主要技术，下面给出着色技术的相关定义[8]。

定义 5 设 B 是一个包含多个元素的集合，B 的一种着色是一个把 B 映射到自然数集 $\{1,2,\cdots\}$ 的函数。特别地，B 的一个 k-着色是一个将 B 映射到 $\{1,2,\cdots,k\}$ 的函数。对于 B 的一种着色 f 和 B 的一个子集 W，如果 W 中的任意两个元素都没有被 f 着上相同的颜色，那么称 W 被 f 正确着色。如果 H 是 B 的多个 k-着色的集合，并且对于 B 中的任意一个含有 k 个元素的子集 W_k，H 中总有一个 k-着色把该子集 W_k 正确着色，则说 H 是 B 的一个 k-着色方案。k-着色方案 H 的大小等于 H 中 k-着色函数的个数。

着色技术是文献[9]提出来的，该文献证明了对于任意一个含有 n 个元素的集合，存在一个大小为 $O(2^{O(k)}n)$ 的 k-着色方案。最近文献[8]将这一结果大大改进，得到了下述引理。

引理 1[8] 对于任意含有 n 个元素的有限集 B 和非负整数 $k(k\leqslant n)$，存在 B 的一个大小为 $O(6.4^kn)$ 的 k-着色方案 H，并且 H 可在时间 $O(6.4^kn)$ 内被构造。

在枚举算法中，下述技术是本文用到的一项重要技术。设 L 为一个含有 n 个实数的列表，则可在时间 $O(n)$ 内找出列表 L 中第 z 大(或第 z 小)的数 r[10]，然后以 r 为“枢支点”，便可在时间 $O(n)$ 内找出列表中最大(或最小)的 z 个数。

3 固定参数枚举算法

在详细描述算法之前，下面简要说明算法的基本思想。

对于 3-维匹配问题的任意一个实例 S，首先运用文献[8]中的着色技术对 S 中元组的元素构造一个着色方案 H，然后依次运用 H 中的每一个着色函数 f 对 S 中元组的元素进行着色，同时求出 S 在当前着色函数映射下正确着色的权值最大的 z 个 k-matchings 并存放在集合 R 中。根据着色方案的定义，S 中每一个 k-matching 必定至少被一个着色函数 f 正确着色，因此容易证明 R 必定包含了 S 中权值最大的 z 个 k-matchings。在 R 中选择权值最大的 z 个 k-matchings 就得到了实例 S 的权值最大的 z 个 k-matchings。

接下来根据固定参数枚举的相关定义分别介绍其结构算法和枚举算法。

3.1 结构算法

对于 3-维匹配问题，显然可以直接运用文献[8]中的着色算法作为结构算法。进一步地，由 3-维匹配问题的定义可知其实例中每一个元组的 3 个元素都来自 3 个互不相交的符号集。为尽量减少着色方案的大小，本文只用 $2k$ 种颜色对 S 中的第 2 列和第 3 列符号进行着色，由此构造的着色方案为一个 $(2k)$-着色方案。相应地，本文后续部分中正确着色的 matching 均指其第 2 列和第 3 列符号没有被着上相同颜色的 matching。

根据引理 1，关于 S 第 2,3 列符号的 $(2k)$-着色方案 H 的大小为 $O(6.4^{2k}n)$。如果依次用 H 中 $O(6.4^{2k}n)$ 个着色函数分别对 S 的第 2,3 列符号进行着色，$O(6.4^{2k}n)$ 个着了色的 S 就可以看作结构算法生成的结构 $\pi_{I,k}$。

3.2 局部枚举算法

设 f_i 为着色方案 H 中的任意一个 $(2k)$-着色函数，下面介绍如何在 f_i 映射下的 S 中枚举正确着色且权值最大的 z 个 k-matchings。图 1 详细描述了该算法的具体过程，其中 $cl(\mathrm{Val}^i(\rho))$ 表示元组 ρ 的第 i 列符号被着的颜色，$i=2,3$，组对 (C,M) 中的 M 表示正确着颜色集 C 且权值最大的 z 个 matchings 的集合。

算法 ELMM(S,k,z,f_i)

输入：一个元组集合 S，符号集 $\mathrm{Val}^2(S)\cup\mathrm{Val}^3(S)$ 的一个 (2k)-着色函数 f_i，两个非负整数 k，z

输出：S 中被 f_i 正确着色的且权值最大的 z 个 k-matchings；

1 如果某一元组中的两个元素被着上了同种颜色，则把它从 S 中删除；

2 设剩下的元组构成集合 S'；

3 设 $\mathrm{Val1}(S')$ 的元素为 $x_1,x_2,\cdots,x_r$；

4 $Q_{old}=\{(\emptyset,\{\emptyset\})\}$；$Q_{new}=\{(\emptyset,\{\emptyset\})\}$；

5 for i=1 to r do

5.1 for Q_{old} 中的每个组对(C,M) do

① 为讨论问题的简便，本文中规定 z 指值 $z'=\min\{z,|S_Q(I,k)|\}$。

5.2 for S′ 中每个满足 $\mathrm{Val}^1(\rho)=x_i$ 的元组 ρ do

5.3 if $C\cap\{cl(\mathrm{Val}^2(\rho)),cl(\mathrm{Val}^3(\rho))\}=\emptyset$

5.4 then $\{C'=C\cup\{cl(\mathrm{Val}^2(\rho)),cl(\mathrm{Val}^3(\rho))\}$；

5.5 对 M 中每一个 matching m 执行 $m'=m\cup\{\rho\}$，并存放到新集合 M′ 中；

5.6 if C′中的颜色数不超过 2k

5.7 then if Q_{new} 中另有一个组对(C″,M″)用到的颜色 C″ 正好与 C′ 相同

5.8 then {将 M′ 中 z 个 matchings 添加到 M″中；

5.9 在 M″中保留权值最大的 z 个 matchings；}

5.10 else 将(C′,M′)直接添加到 Q_{new} 中；}

5.11 $Q_{old}=Q_{new}$；

6 如果 Q_{old} 中含有组对(C_{2k},M)，其中 C_{2k} 是 2k 种颜色的集合，则返回对应 M 中的 k-matching，否则返回不存在

图 1 在 f_i 映射下的 S 中进行局部枚举的算法

该枚举过程的基本思想是：从 S 中第 1 个元组出发，用动态规划技术依据其被着的颜色逐步进行组合扩充，直到 S 中所有元组相互之间都进行了组合。同时，在相同颜色的 j-matchings($j\leqslant k$)中及时枚举权值最大的 z 个。在动态规划过程中，如果存储空间中已存在一个着同种颜色的组对，则从这两个 matching 组中选择权值最大的 z 个并保存，否则将新的组对直接放入存储空间。

算法 ELMM(S,k,z,f_i)第 1 步的正确性是显然的，任何一个其中两个符号着了相同颜色的元组不可能属于一个正确着色的 matching。因此，只要考虑其第 2 列和第 3 列符号着上不同颜色的元组。

对于任意一个固定的整数 $q(0\leqslant q\leqslant r,r=|\mathrm{Val}^1(S')|)$，设 S_q 为 S' 中第 1 列符号属于 $\{x_1,x_2,\cdots,x_q\}$ 的元组集合，则首先可以得到如下引理。

引理 2 对于任意一个整数 h ($h\leqslant k$)，如果集合 S_q 中含有 d 个被颜色集 C' ($|C'|=2h$) 正确着色的 h-matchings，则算法 ELMM(S,k,z,f_i)第 5 步执行第 q 次循环后，超集合 Q_{old} 中必定存在一个组对(C',M)，其中 M 是 S_q 中被 C' 正确着色的且权值最大的 z ($z\leqslant d$)个 h-matchings 的集合。

证明：现用归纳法来证明。当 $q=0$ 时，显然 S_0 是一个空集，算法的第 4 步给出了该初始情况。

对于一般情况，首先作下述假设。设 A 为 S_q 中被颜色集 C' 正确着色的所有 h-matching 的集合，$|A|=d$，T_q 为 S' 中第 1 列符号都为 x_q 的元组的集合。进一步地，设 A_1 为 S_q 中被 C' 正确着色且不含有 T_q 中元组的 h-matching 的集合，$|A_1|=d_1$，A_2 为 S_q 中被 C' 正确着色且含有 T_q 中的一个元组的 h-matching 的集合，$|A_2|=d_2$，显然 A 中的 h-matching 可以分成 A_1，A_2 两个子集合，即 $A=A_1\cup A_2$，$d_1+d_2=d$。

当 $q\geqslant1$ 时，根据归纳法，算法第 5 步的循环语句在第 $q-1$ 次执行之后、第 q 次执行之前，超集合 Q_{old} 中含有一个组对(C,M) 的充分必要条件是集合 S_{q-1} 中至少含有一个被 C 正确着色的($|C|/2$)-matching。特别地，对于 A_1，超集合 Q_{old} 必定含有一个组对(C',M_1)，其中 M_1 是包含 A_1 中权值最大的 z 个 h-matchings 的集合。算法第 5 步的第 q 次执行过程中依次产生 A_2 中的 h-matching 并依据其权值大小动态调整 M_1 中的 z 个 h-matchings。为此，算法从 Q_{old} 中依次取出组对(C,M)，将 M 中的 z 个($h-1$)-matchings 与 T_q 中满足条件 $C\cap\{cl(\mathrm{Val}^2(\rho)),cl(\mathrm{Val}^3(\rho))\}=\emptyset$ 和 $C'=C\cup\{cl(\mathrm{Val}^2(\rho)),cl(\mathrm{Val}^3(\rho))\}$ 的元组 ρ 组合成 z 个 h-matchings，并由算法的第 5.8 步添加到 Q_{new} 组对(C',M_1)的集合 M_1 中。同时通过第 5.9 步使得 M_1 中始终保留当前权值最大的 z 个 h-matchings。

在算法第 5 步的第 q 次循环过程中，进一步可以得到下述事实。

设 m_1 和 m_2 为 S_{q-1} 中的两个($h-1$)-matching，它们分别被颜色集 C_1 和 C_2 正确着色，且 $|C_1|=|C_2|=2h-2$，并设 ρ_1，ρ_2 为 T_q 中的满足下述关系的两个元组：$C_1\cap\{cl(\mathrm{Val}^2(\rho_1)),cl(\mathrm{Val}^3(\rho_1))\}=\emptyset$，$C_2\cap\{cl(\mathrm{Val}^2(\rho_2)),cl(\mathrm{Val}^3(\rho_2))\}=\emptyset$，$C'=C_1\cup\{cl(\mathrm{Val}^2(\rho_1)),cl(\mathrm{Val3}(\rho_1))\}=C_2\cup\{cl(\mathrm{Val}^2(\rho_2)),cl(\mathrm{Val}^3(\rho_2))\}$。

(1) 如果 $\rho_1\neq\rho_2$，显然 $m_1\cup\{\rho_1\}$ 和 $m_2\cup\{\rho_2\}$ 含有 T_q 中的不同元组，因此 $m_1\cup\{\rho_1\}$ 和 $m_2\cup\{\rho_2\}$ 为两个不同的被 C' 正确着色的 h-matching。

(2) 如果 $\rho_1=\rho_2$，则必定有 $C_1=C_2$。由于 m_1 和 m_2 都为 S_{q-1} 中被 C_1 正确着色的两个($h-1$)-matching，$m_1\cup\{\rho_1\}$ 和 $m_2\cup\{\rho_2\}$ 都是被 C' 正确着色的 h-matching，因此 $m_1\cup\{\rho_1\}$ 不同于 $m_2\cup\{\rho_2\}$ 的充分必要条件是 m_1 不同于 m_2。

因此，算法第 5 步的循环语句在第 q 次执行之后，A_2 中每一个 h-matching 都将与 A_1 中的 d_1 个 h-matchings 进行权值比较，超集合 Q_{old} 中组对(C',M)的 M 必定是 S_q 中 $d_1+d_2=d$ 个被 C' 正确着色的权值最大的 z 个 h-matchings 的集合。

由此引理可以得到本部分的主要定理。

定理 1 算法 ELMM(S,k,z,f_i)返回在 f_i 映射下 S 中正确着色且权值最大的 z 个 k-matchings，其时间复杂度为 $O(2^{2k}knz)$。

证明：根据引理 2，当算法执行结束时，Q_{old} 中的组对(C_{2k},M)表明 S_r 中被 C_{2k} 正确着色的且权值最大的 z 个 k-matchings 已经被选择出来。由于 $S_r=S'$，且 C_{2k} 含有 f_i 的所有颜色，因此，M 正好是在($2k$)-着色函数 f_i 映射下 S 中正确着色且权值最大的 z 个 k-matchings 的集合。即算法 ELMM(S,k,z,f_i)是正确的。

接下来分析算法 ELMM(S,k,z,f_i)的时间复杂度。对于任意满足 $0\leqslant h\leqslant k$ 的 h 和每一个含有 $2h$ 种不同颜色的集合 C，超集 Q_{old} 中保存了一个对应的组对(C,M)。由于在 $2k$ 种不同颜色中共有 $\binom{2h}{2k}$ 个不同的颜色集，因此 Q_{old} 中不同组对的总数有上界 $\sum_{h=0}^{k}\binom{2k}{2h}\leqslant2^{2k}$。对于任意 $i(1\leqslant i\leqslant k)$，算法的第 5.3 步判定任意组对($C,M$)中的颜色集 C 是否合并元组 ρ 的颜色，可以在时间 $O(k)$ 内完成。第 5.5 步显然可以在时间 $O(z)$ 内完成。第 5.7 步用一个大小为 2^{2k} 的有序数组保存所有颜色集的状态，并用二分查找法查找一个给定颜色集的状态，可以在时间 $O(k)$ 内完成。算法的 5.9 步计算各 matching 的权值需要 $O(kz)$ 时间，从 $2z$ 个着相同颜色集的 matchings 集合 M 中找出第 z 大的 matching m_0 可以在时间 $O(z)$ 内完成，然后以 m_0 作为“枢组点”，从 M 中找出权值最大的 z 个 matchings。在第 5 步的第 q 次循环执行过程中，第 5.2 步的循环只验证 S' 中 n_q 个元组，其中 n_q 表示 S' 中第 1 列符号都为 x_q 的元组的个数。因此第 5 步第 q 次循环共需要时间 $O(2^{2k}n_q(kz+k+z))=O(2^{2k}n_qkz)$。由于算法的时间主要集中在第 5 步，可得整个算法时间复杂度为 $\sum_{q=1}^{r}O(2^{2k}n_qkz)=$

$O(2^{2k}kz)\sum_{q=1}^{r} n_q = O(2^{2k}knz)$。

3.3 整个枚举算法

对于3-维匹配问题的任意一个实例 S，给出枚举 S 中权值最大的 z 个 k-matchings 的算法。图2详细描述了该算法的主要步骤。

算法 EMM(S,k,z)
输入：一个含有 n 个带权的元组集合 S，两个非负整数 k，z
输出：S 中权值最大的 z 个 k-matchings
1 R=∅；
2 对 S 的第 2,3 列符号构造一个(2k)-着色方案 H；
3 for H 中的每一个(2k)-着色函数 f_i do
  3.1 T=ELMM(S,k,z,f_i)；
  3.2 R=R∪T；
4 if R≠∅
  4.1 then {用基数排序法删除 R 中重复的 k-matching；
  4.2 在 R 中输出权值最大的 z 个 k-matchings；}
  4.3 else 提示不存

图2 3-维匹配问题的固定参数枚举算法

定理2 设 S 为带权的3-维匹配问题的一个含有 n 个元组的实例，则算法 EMM(S,k,z)能在时间 $O(5.48^{3k}kn^2z)$ 内枚举 S 中权值最大的 z 个 k-matchings。

证明：首先证明算法 EMM(S,k,z)的正确性。设 ma 为 S 中的任意一个 k-matching，H 为 S 中第2,3列符号的一种($2k$)-着色方案，由着色方案的定义可知，ma 一定被 H 中的某一个 f_i 正确着色。进一步地，如果 ma 为 S 中权值最大的 z 个 k-matchings 中的任意一个，则由算法 ELMM(S,k,z,f_i)的正确性可知，ma 一定由算法 ELMM(S,k,z,f_i)输出，即一定包含在集合 R 中，因此在 R 中选择权值最大的 z 个 k-matchings 就是实例 S 的权值最大的 z 个 k-matchings。如果 S 中不存在 k-matching，算法的第4步将返回空集。

为避免集合 R 中出现 k 个元组完全相同的多个 k-matchings，算法的第4.1步用基数排序删除可能重复的 k-matching。基数排序时用 k-matching 中 k 个元组名的组合作为关键字，元组名顺序均按实例 S 中元组的自然顺序排列。

接下来分析它的时间复杂度。算法 EMM(S,k,z)第2步的运行时间也就是着色算法的运行时间，即 $O(6.4^{2k}n)$，第3步用 $O(6.4^{2k}n)$ 个着色函数依次对 S 的第2,3列符号进行着色，并求出每种着色函数映射下的权值最大的 z 个 k-matchings，运行时间为 $O(2^{2k}nkz)\times O(6.4^{2k}n)=O(12.8^{2k}n^2kz)$，产生最多 $O(6.4^{2k}nz)$ 个权值最大的 k-matchings 并构成集合 R。第4.1步基数排序需运行时间 $O(6.4^{2k}nzk)$。第4.2步对已清除重复 k-matching 的 R 可以在 $O(6.4^{2k}nz)$ 时间内找到第 z 大的 k-matchings，然后在 $O(6.4^{2k}nz)$ 时间内找出权值最大的 z 个互不相同的 k-matchings。

因此，算法 EMM(S,k,z)的时间复杂度为：$O(6.4^{2k}n)+O(12.8^{2k}n^2kz)+O(6.4^{2k}nkz)+O(6.4^{2k}nz)=O(12.8^{2k}n^2kz)=O(5.48^{3k}kn^2z)$。

定理2表明产生 z 个权值最大的 k-matchings 的"平均"时间是 $O(5.48^{3k}kn^2)$。

推论1 3-维匹配问题是固定参数线性可枚举的。

结束语 本文运用最新的着色技术和动态规划技术对带权的3-维匹配问题提出了一个时间复杂度为 $O(5.48^{3k}kn^2z)$ 的固定参数枚举算法，表明了带权的3-维匹配问题是固定参数线性可枚举的。

本文的结论具有重要的理论意义。对于3-维匹配问题的任意一个实例，枚举其中所有的 k-matchings 目前还不存在高效算法，但本文的结论表明枚举其中权值最大的 z 个 k-matchings 一定存在一个高效算法。从这个意义上讲，3-维匹配问题是又一个固定参数可枚举的典型例子。

3-维匹配问题有着广泛的应用背景，如线路布置、多级调度和工种分派等。因此本文提出的固定参数枚举算法也具有较大的应用价值。

参考文献

[1] Valiant L. The complexity of enumeration and reliability problems [J]. SIAM Journal on Computing, 1979, 8(3): 410-421

[2] Matsui Y, Matsui T. http://dmawww.epfl.ch/roso.mosaic/kf/enum/comb/combenum.html

[3] Flum J, Grohe M. The parameterized complexity of counting problems [J]. SIAM Journal on Computing, 2004, 33(4): 892-922

[4] Chen J, Kanj I, Meng J, et al. On effective enumerability of NP problems[C]// Proceedings of the 2nd International Workshop on Parameterized and Exact Computation. Berlin: Springer-Verlag, 2006: 215-226

[5] Liu Y, Lu S, Chen J, et al. Greedy localization and color-coding: improved matching and packing algorithms [C]//Proceedings of the 2nd International Workshop on Parameterized and Exact Computation. Berlin: Springer-Verlag, 2006: 84-95

[6] 宁丹，王建新. 一个基于着色和动态规划的3-维匹配问题算法[J]. 计算机科学，2007，34(7)：222-224

[7] Liu Y, Chen J, Wang J. On counting 3-D matchings of size k [J]. Algorithmica, 2009, 54: 349-359

[8] Chen J, Lu S, Sze S -H, et al. Improved algorithms for path, matching, and packing problems [C]//Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. 2007: 298-307

[9] Alon N, Yuster R, Zwick U. Color-coding [J]. Journal of the ACM, 1995, 42: 844-856

[10] Cormen T, Leiserson C, Rivest R, et al. Introduction to Algorithms 2nd Edition [M]. Boston, MA: McGraw-Hill Book Company, 2001

(上接第145页)

[2] www.aco-metaheuristic.org

[3] Colorni A, Dorigo M, Maniezzo V. Distributed Optimization by ant colonies[C]//F. J. Varela & P. Bourgin, eds. Proceedings of the First European Conference on Artificial Life. Cambridge, MA, MIT Press: 134-412

[4] Minton S, Johnston. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems[J]. Artificial Intelligence, 1992

[5] Reischle M F, Schmeck H. Multi colony ant algorithms[J]. Journal of Heuristics, 2002

[6] Dorigo M, Stutzle T. Ant Colony Optimization. 2007

[7] www.isi.edu/nsnam/ns/