arXiv:2304.10995v1 [cs.DS] 21 Apr 2023

通过分支定价算法求解列表着色问题⋆

MauroLucci[a,b], Graciela Nasini[a,b], Daniel Sever´ın[a,b,1]

[a]*Depto. de Matemática (FCEIA), Universidad Nacional de Rosario, Argentina*
[b]*CONICET, Argentina*

摘要

本文介绍了一种基于稳定集顶点覆盖公式的分支定价算法，用于求解列表着色问题的加权版本。该问题不仅具有实际应用价值，还概括了许多其他问题，包括著名的图着色问题。通过引入不可区分颜色的概念，本文给出了一些理论结果，并将其应用于后续的算法中。我们提出了两种分支策略，均借鉴自图着色问题的相关方法：第一种改编自 Mehrotra and Trick 在其开创性的分支定价算法中采用的方法；另一种则受到 M´endez-D´ıaz 和 Zabala 在其分支切割算法中所用方法的启发。该问题丰富的结构使这两种分支策略都具有稳健性。对多种不同实例进行的大规模计算实验表明，该方法有效，并揭示了算法会因不同类型实例的结构而呈现出不同的行为。

关键词：颜色列表，分支定价，加权问题。2010 年 MSC：90C57, 05C15

1. 引言

图着色问题（GCP）可用于建模多种规划问题，例如时间表编排、调度、电子带宽分配、排序和寄存器分配。因此，人们投入了大量精力开发高性能工具，以解决这一 $\mathcal{NP}$难问题。

给定无向简单图 $G = (V, E)$ 和颜色集合 $\mathcal{C}$，$\mathcal{C}$-着色（或简称着色）是指 G 上的一个函数 $f\colon V \to \mathcal{C}$，满足对于 G 的每条边 (u, v)，$f(u) \neq f(v)$。给定 G 的一个着色 f 和 $j \in \mathcal{C}$，$f^{-1}(j) \doteq \{v \in V\colon$

⋆部分受PICT-2020-03032（ANPCyT）和PIP-1900（阿根廷国家科学技术研究委员会）项目资助。*E-mail addresses*: mlucci@fceia.unr.edu.ar （M. Lucci），nasini@fceia.unr.edu.ar （G.Nasini），daniel@fceia.unr.edu.ar （D.Sever´ín）。[1]通讯作者地址：阿根廷罗萨里奥市Pellegrini 250号，UNR，数学´系（FCEIA）。

$f(v) = j\}$ 是 j 的颜色类，即由着色为 j 的顶点构成的子集。f 中的活跃颜色记为 $\mathscr{A}(f)$，是指分配给某个顶点的颜色集合，即 $\mathscr{A}(f) \doteq \{j \in \mathcal{C} : f^{-1}(j) \neq \emptyset\}$。给定 $k \in \mathbb{Z}_+$，若着色 f 是 k-着色，则称其为 $|\mathscr{A}(f)| = k$。图着色问题（GCP）旨在求出满足 G 可进行 k-着色的最小 k。这个最小值记为 $\chi(G)$，称为 G 的色数。

在精确算法的背景下，目前已知的最佳算法基于整数线性规划（ILP）模型，并通过不同的分支定界方案求解，例如分支割平面（B&C） [23] 和分支定价（B&P） [22]。在这些经典方法中加入的新特性持续提升了可求解着色问题实例的规模，例如 [10, 11]。

在某些应用中，有必要考虑一些额外的限制，从而产生了着色变体 [24, 18]。

例如， μ-着色 [4] 出现在这样的应用中：资源按照质量或大小排序，而用户对质量或大小有特定的最低要求。一个例子是将教室（具有不同容量）分配给课程（学生人数不同）的问题。

形式化地，给定一个图 $G = (V, E)$、一个颜色集合 $\mathcal{C} = \{1, \ldots, n\}$，其中$n \in \mathbb{N}$，以及一个向量 $\mu \in \mathcal{C}^V$， G 的 μ-着色是 G 的一种着色 f，满足对于所有 $v \in V$， $f(v) \leq \mu(v)$。μ-着色问题旨在寻找 G 的一个μ-着色 f，使 $\mathscr{A}(f)$的基数最小。

在调度和超大规模集成领域，有些问题可以建模为扩展图中给定的部分着色，即为图寻找一种着色，其中某些顶点具有预指定颜色 [2]。形式上，在预着色扩展问题中，输入包括一个图 $G = (V, E)$、一个颜色集合 $\mathcal{C}$、一个子集 $W \subset V$，以及图 G 由W诱导的子图上的一种着色 f'。目标是寻找图 G 的一种着色 f，使得对于所有 $v \in W$， $f(v) = f'(v)$，并且 $\mathscr{A}(f)$的基数最小。预着色扩展问题也可用于建模拟群补全问题 [15]等理论问题。此外，在求解图着色问题（GCP）时，一种广为人知且非常有效的预处理阶段是为图中的一个极大团着色，从而将图着色问题实例转化为预着色扩展问题的一个实例。

本文研究一种着色问题，它是对前述所有问题的推广：列表着色问题（LCP）。在 LCP 中，图 G 的每个顶点 v都有一个非空的预指定颜色列表 $L(v) \subset \mathcal{C}$。实例记作 (G, L)。(G, L)的列表着色是图 G 的一种着色 f，并附加条件：对于所有 $v \in V$， $f(v) \in L(v)$。不失一般性，我们可以考虑 $\mathcal{C} = \bigcup_{v \in V} L(v)$。与通常情况一样，LCP 的目标是寻找 (G, L)的一种列表着色，使得 $\mathscr{A}(f)$的基数最小。

显然，在对于所有 $v \in V$, $L(v) = \mathcal{C}$和 $L(v) = \{1, \ldots, \mu(v)\}$分别成立的实例上，GCP 和 μ-着色问题与 LCP 等价。此外，在满足 $L(v) = \{f'(v)\}$对所有 $v \in W$以及 $L(v) = \mathcal{C}$对所有 $v \in V \setminus W$成立的实例上，预着色扩展问题与 LCP 等价。

在实际应用中，我们有时需要考虑颜色的权重（或成本），由此产生了加权列表着色问题（WLCP）。给定一个

权重向量 $w \in \mathbb{Z}_+^{\mathcal{C}}$ 和 G的一种着色 f，其权重 $w(f)$ 定义为$w(f) \doteq \sum_{j \in \mathscr{A}(f)} w_j$。WLCP 的目标是找到权重最小的列表着色 f。形式化地说：

> Weighted List Coloring Problem（WLCP）实例：一个图 $G = (V, E)$、一组有限颜色列表 $L = \{L(v) : v \in V\}$，以及 $w \in \mathbb{Z}_+^{\mathcal{C}}$。目标：求得 (G, L) 的一个列表着色 f，使得 $w(f)$ 最小；或者回答不存在这样的着色。

WLCP 的实例将表示为 (G, L, w)。显然，对于所有 $c \in \mathcal{C}$，LCP 的实例都可以视为满足 $w_c = 1$ 的 WLCP 的特殊实例。值得指出的是，GCP 的加权版本与 GCP 等价。事实上，可以通过对G 进行 $\chi(G)$-着色来获得一个最优解，其中活跃颜色是权重最低的前 $\chi(G)$ 个颜色。很容易看出，对于一般的 WLCP 实例，情况并非如此。

关于计算复杂性，由于 GCP 在一般图中是 $\mathcal{NP}$-难的，因此 LCP（以及 WLCP）也是如此。然而，一些已知结果似乎表明，LCP 比 GCP 更难。

尤其是，可行性并不容易处理。图 1 所示的著名示例对应于一个不可行的列表着色问题实例，其中所有顶点的颜色列表中都有 $\chi(G)$个元素。

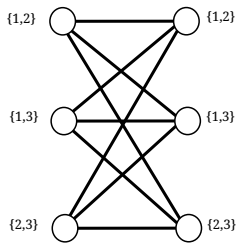


图1：一个不可行的列表着色问题实例。颜色列表显示在花括号内。

判定 (G, L) 是否存在列表着色是一个 $\mathcal{NP}$-完全问题，即使将范围限制为颜色列表大小至多为 3，且 G 是余图或完全二部 [14]的实例也是如此。相比之下，余图 [6] 和二部图上的图着色问题可以在线性时间内求解。此外，对于 G 树宽有界的图 t，存在一种动态规划算法，能够在多项式时间内求出其色数；而判定 G 是否存在列表着色则是 W[1]-困难问题[7]。尽管如此，人们仍在努力设计优秀的算法，以便在列表着色存在时找到它。据我们所知，目前最好的算法是 $O(2^n)n^{O(1)}$，适用于具有 n 个顶点且颜色列表大小任意的图 [3] 。

WLCP 的一个实际应用是设计公共交通公司司机的工作日 [19]。此外，除了着色

问题之外，WLCP 还推广了其他优化问题，例如著名的最小加权集合覆盖问题。在下一节中，我们将分析这两个问题之间的关系。

总之，开发一种用于求解一般加权列表着色问题实例的工具，将有助于应对广泛的组合优化问题及其应用。

在本研究中，我们提出了一种基于整数线性规划公式的加权列表着色问题精确算法。本研究拓展了作者在会议论文 Lucci 等人 [20] 中提出的一些思路。

第 2 节介绍了必要的符号和定义，以及我们算法所依据的一些理论结果。第 3 节基于将列表着色刻画为通过稳定集覆盖顶点，给出了具有指数级数量变量的整数线性规划公式。

在第 4、5 和 6 节中，我们介绍了 B&P 算法的主要组成部分：定价规则、分支策略和变量选择规则。

LCP 缺乏基准实例，且定义这些实例的相关参数种类繁多，因此我们设计了 5 个实例集，以涵盖第 7 节所介绍的各种结构。第 8 节将在这些实例集上评估扩展的计算实验。

最后，在第 9 节中，我们给出一些结论并提出待解决的问题。

2. 预备知识

给定图 $G = (V, E)$，$v \in V$ 和 $N_G(v)$ 表示与 v 相邻的顶点集，而 $N_G[v] = N_G(v) \cup \{v\}$。给定 $U \subset V$，$G[U]$ 表示由 U 和 $G - U = G[V \setminus U]$ 诱导的 G 的子图。当 $U = \{u\}$ 时，我们简单记作 $G - u$。如果顶点子集 S 中任意一对顶点都不相邻，则称其为 G 的稳定集（分别地，若任意一对顶点都相邻，则称其为团）。

给定一个 WLCP 实例 (G, L, w)，对于每个 $j \in \mathcal{C} \doteq \bigcup_{v \in V} L(v)$，我们定义 $V_j \doteq \{v \in V : j \in L(v)\}$ 和 $G_j \doteq G[V_j]$，分别表示（与 j 相关的）颜色顶点集和颜色图。此外，$\mathscr{S}_j$ 是 G_j 的非空稳定集集合。

注意，WLCP 的实例也可以通过给出颜色-顶点集而非颜色列表来描述。事实上，三元组 $(G, \mathcal{V}, w)$ 与 $G = (V, E)$，$\mathcal{V} = \{V_j \subset V : j \in \mathcal{C}\}$，以及 $w \in \mathbb{Z}_+^{\mathcal{C}}$ 相对应的实例为 (G, L, w)，其中 $L(v) = \{j \in \mathcal{C} : v \in V_j\}$。由于这两种表示等价，我们将根据上下文交替使用。

下面的结果表明，LCP 的实例 (G, L) 的结构取决于颜色图，而非 G，因为可以在 G 中添加（或删除）端点位于不同颜色-顶点集中的边，而不改变列表着色的集合。

引理 2.1。设 (G, L) 与 $G = (V, E)$ 构成列表着色问题的一个实例，并且 u、$v \in V$ 满足 $(u, v) \notin E$ 且 $L(u) \cap L(v) = \emptyset$。设 G' 为通过添加

边 (u,v) 到 G 所得到的图。那么，当且仅当 f 是 (G,L) 的一个列表着色时，f 才是 (G,L) 的一个列表着色。

证明。设 f 是 (G,L) 的一个列表着色。由于 $f(u) \in L(u)$、$f(v) \in L(v)$，并且根据假设，$L(u) \cap L(v) = \emptyset$，因此 $f(u) \neq f(v)$。所以，f 是 (G',L) 的一个列表着色。

反之则显而易见。 □

基于颜色图的结构，我们提出了两类在计算复杂性方面相反的实例。我们证明，当所有颜色图都是完全图时，WLCP 可在多项式时间内求解；而当颜色图是无边图时，WLCP 仍然是 $\mathcal{NP}$-困难的。

前者可以归约为二部图上的最小权完美匹配问题（MWPMP），该问题可通过匈牙利算法在多项式时间内求解（例如参见 [9]）。事实上，我们有：

命题 2.2. 对于满足以下条件的实例 (G,L,w)，WLCP 可以在多项式时间内求解：对于所有 $j \in \mathcal{C}$, G_j，其均为完全图。

证明。设 $G=(V,E)$ 和 (G,L,w) 是 WLCP 的一个实例，使得对于所有 $j \in \mathcal{C}$，G_j 是一个完全图。由于当 $|V| > |\mathcal{C}|$ 时，(G,L,w) 不可行，因此我们假设 $|V| \leq |\mathcal{C}|$。

设 Z 是一组虚拟元素，使得 $|Z| = |\mathcal{C}| - |V|$。我们构造二分图 H，其顶点划分由集合 $V \cup Z$ 和 $\mathcal{C}$ 定义。此外，每个 $v \in V$ 都与每个 $j \in L(v)$ 相邻，而每个 $z \in Z$ 都与每个 $j \in \mathcal{C}$ 相邻。另外，当 $v \in V$ 时，边 (v,j) 的权重为 w_j，而当 $z \in Z$ 时，边 (z,j) 的权重为零。

给定 H 的一个完美匹配 M，令 $f_M: V \to \mathcal{C}$，使得如果 $(v,j) \in M$，则 $f_M(v) = j$。根据构造，f_M 是 (G,L) 的一个列表着色，并且 $w(f_M) = w(M)$，其中 $w(M)$ 是 M 的权重。

反之，给定 f 的 (G,L) 列表着色，我们构造 $M_f = \{(v,f(v)) : v \in V\}$。显然，$M_f$ 是 H 的一个匹配，使得 $w(M_f) = w(f)$。令 g 为 Z 与 $\mathcal{C} \backslash f(V)$ 之间的一一对应关系。为获得 H 的完美匹配M，我们对所有 $z \in Z$ 添加弧 $(z,g(z))$ 到 M_f 中。显然，$w(M) = w(f)$。

因此，(G,L,w) 是 $WLCP$ 的一个不可行实例，当且仅当 H 不存在完美匹配。此外，如果 H 的一个最小权完美匹配为 M，那么 f_M 就是 (G,L,w) 的一个最优列表着色。

□

对于无边着色图的情况，我们得到了一个从最小权集合覆盖问题（MWSCP）出发的多项式时间归约。

给定一个集合 X 和一个子集族 $\mathcal{S}$，其中每个子集都是 X 的子集，满足 $\bigcup_{S \in \mathcal{S}} S = X$ 的 X 的一个集合覆盖是 X 的一个子集 $\mathcal{S}'$，且满足 $\mathcal{S}$ 和$\bigcup_{S \in \mathcal{S}'} S = X$。在 MWSCP 中，$\mathcal{S}$ 的元素具有由向量 $w \in \mathbb{Z}_+^{\mathcal{S}}$ 给出的权重，目标是找到 X 的一个最小权重集合覆盖 $\mathcal{S}'$，使其权重 $\sum_{S \in \mathcal{S}'} w_S$ 最小。即使在无权重的情况下（$w = \mathbf{1}$），当集合大小为 3 时，MWSCP 仍然是 $\mathcal{NP}$-困难的（参见问题 [SP5] of [12]）。

我们可以证明：

命题 2.3. 对于实例 (G, L)，列表着色问题是 $\mathcal{NP}$-困难的，其中，对于所有$j \in \mathcal{C}$, G_j 都是无边图。

证明。设 $(X, \mathcal{S}, 1)$ 是一个 MWSCP 实例。我们构造列表着色问题实例 (G, L)，其中 $G = (X, \emptyset)$ 且 $L(v) = \{S \in \mathcal{S} : \text{对于所有 } v \in X, \ v \in S\}$。

给定一个 (G, L) 的列表着色 f，显然，$\mathcal{A}(f)$ 是 X 的一个集合覆盖。反过来，给定 X 的一个集合覆盖 $\mathcal{S}'$，存在一个函数 $f : X \to \mathcal{S}'$，使得对于所有 $v \in X$, $v \in f(v)$。显然，f 是 (G, L) 的一个列表着色。此外，如果 f 是 (G, L) 的一个最优列表着色，则有 $|\mathcal{S}'| \geq |\mathcal{A}(f)| \geq |\mathcal{A}(f)|$。因此，$\mathcal{A}(f)$ 是 X 的一个最优集合覆盖。

注意，我们还证明了，$(X, \mathcal{S}, 1)$ 不可行，当且仅当 (G, L) 也不可行。

□

众所周知，给定图 G 的一种着色 f 和一种颜色 $j \in \mathcal{A}(f)$，$f^{-1}(j)$ 是 G 的稳定集。该条件使我们能够将着色定义为将顶点集划分为稳定集。在列表着色的情况下，$f^{-1}(j)$ 必须是 G_j 的稳定集，即 $f^{-1}(j) \in \mathcal{S}_j$。

然而，要从将顶点划分为稳定集的划分中还原列表着色，就必须确定划分中每个稳定集所对应的颜色。例如，考虑 G，它是一条顶点路径 $V = \{v_i : 1 \leq i \leq 5\}$ 和边 $E = \{(v_i, v_{i+1}) : 1 \leq i \leq 4\}$、$\mathcal{C} = \{1, 2, 3\}$、$V_1 = \{v_1, v_5\}$、$V_2 = \{v_2, v_3, v_4\}$，以及$V_3 = \{v_1, v_2, v_3, v_4\}$。由稳定集 $S = \{v_1, v_5\}$、$S' = \{v_2, v_4\}$ 和 $S'' = \{v_3\}$ 给出的 V 划分定义了 G 的两种可能列表着色，因为 S' 和 S'' 的顶点可以互换颜色 2 和 3。

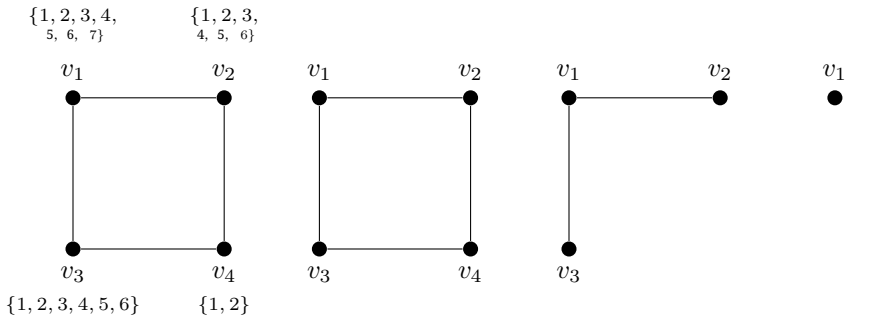
因此，将 X 表示为满足 $j \in \mathcal{C}$ 和 $S \in \mathcal{S}_j$ 的有序对 (S, j) 的集合，列表着色可以刻画为满足以下条件的集合 $X \subset X$：对于某个 $j \in \mathcal{C}\}$（其为顶点集的一个划分），对所有 $S \in \mathcal{S}$、$|\{j$，均有 $(S, j) \in \hat{} X\}| = 1$；对所有 $j \in \mathcal{C}$、$|\{S$，均有 $(S, j) \in \hat{} X\}| \leq 1$；并且 $\mathcal{S} = \{S : (S, j) \in \hat{} X$。

在某些情况下，这种列表着色的刻画会变得过于庞大。极端情况出现在源自图着色问题（GCP）的实例中。在这些实例中，对于所有 $j \in \mathcal{C}$ 均有 $V_j = V$，因此无需确定划分中的每个稳定集分别对应哪种颜色。换言之，在图着色问题的实例中，所有颜色彼此 indistinguishable。

不可区分颜色也可能出现在一般的加权列表着色问题实例中，我们可以利用这一特性。首先让我们将这一概念形式化。

定义 2.4. 给定 WLCP 的一个实例 $(G, \mathcal{V}, w)$，如果 $V_j = V_k$ 和$w_j = w_k$，则称两个颜色 $j, k \in \mathcal{C}$ 不可区分，记作 $j \cong k$。

观察可知，$\cong$ 是 $\mathcal{C}$上的一个等价关系。从现在起，$\{\mathcal{C}_j : j \in K\}$表示由 $\cong$ 定义的 $\mathcal{C}$ 的等价类划分，其中 K是 $\mathcal{C}$ 的一个子集，且每个等价类恰有一个代表颜色。因此，对于每个 $j \in K$，都有 $K \cap \mathcal{C}_j = \{j\}$。我们称每个代表颜色 $j \in K$ 具有重数 $m(j) \doteq |\mathcal{C}_j|$。此外，对于给定的顶点 v，我们记



（a）实例（G, L）。 （b） $G_1 = G_2$。 （c） $G_3 = G_4 = G_5 = G_6$。 （d） G_7。

图 2：列表着色问题实例示例。

$k(v) \doteq |K \cap L(v)|$，即列表着色中可以分配给 v 的代表颜色数量。

那么，WLCP 的一个实例也可以表示为 $(G, K, \mathcal{V}, m, w)$，其中 $\mathcal{V} = \{V_j \subset V: j \in K\}$ and $m, w \in \mathbb{Z}_+^K$。

此外，将列表着色刻画为顶点划分成稳定集的方式，可以改写如下：

注 2.5。给定 WLCP 的一个实例 $(G, K, \mathcal{V}, m, w)$，若 X 是由满足 $j \in K$ and $S \in \mathscr{S}_j$ 的配对 (S, j) 构成的集合，则每个列表着色 f 都可以表示为一个集合 $\hat{} X \subset X$，使得 $\mathcal{S} = \{S : (S, j) \in \hat{}X$；对于某个 $j \in K\}$，其为顶点集的一个划分；对于所有 $S \in \mathcal{S}$, $|\{j : (S, j) \in \hat{}X\}| = 1$；并且，对于所有 $j \in \mathcal{C}$，$\hat{}|\{S: (S, j) \in X\}| \leq m(j)$。此外，$w(f) = \sum_{(S,j) \in \hat{}X} w_j$。

请注意，列表着色的这一最新刻画，在应用于图着色问题的实例时，与经典刻画一致。事实上，由于在这些情况下所有颜色都无法区分，我们有 $|K| = 1$。不失一般性，我们可以假设$K = \{1\}$, $m(1) = |\mathcal{C}|$，且 $\mathscr{S}_1$ 与图的稳定集集合重合。

下面通过一个示例介绍上述概念。图 2a 展示了列表着色问题的一个实例，其中颜色列表以大括号表示。容易验证，我们可以考虑 $K = \{1, 3, 7\}$ 和由 $\mathcal{C}_1 = \{1, 2\}$, $\mathcal{C}_3 = \{3, 4, 5, 6\}$, $\mathcal{C}_7 = \{7\}$ 给出的等价类，其重数分别为 $m(1) = 2$, $m(3) = 4$，以及 $m(7) = 1$。它们的颜色图 $G_1 = G$, G_3 和 G_7 分别见图 2b、2c 和 2d。

由 $f(v_1) = 3$, $f(v_2) = f(v_3) = 4$ 和 $f(v_4) = 1$ 给出的列表着色 f，以及由 $g(v_1) = 3$, $g(v_2) = g(v_3) = 5$ 和 $g(v_4) = 1$给出的列表着色 g，二者具有相同的权重，均表示为 $X = \{(\{v_1\}, 3), (\{v_2, v_3\}, 3), (\{v_4\}, 1)\}$。

接下来，我们将介绍一些备注和结果，以便在预处理阶段缩减实例。

与预着色扩展问题的情况类似，WLCP 中可能存在某些顶点的颜色列表中仅包含一个代表颜色的实例。WLCP 的特定实例中，对于某些 $j \in K$、$k(v) = 1$，对于所有

$v \in V_j$（等价地，V_j 不与任何其他代表颜色顶点集相交），可以根据以下备注进行约简。

注记 2.6. 设 (G, L, w) 为 WLCP 的一个实例，并设 $j \in K$ 使得 $k(v) = 1$ 对每个 $v \in V_j$ 都成立。显然，如果 $m(j) < \chi(G_j)$，则该实例不可行。否则，在 (G, L, w) 上求解 WLCP 等价于在 G_j 上求解 GCP，并在 $(G - V_j, L|_{V'}, w|_{\mathcal{C}'})$ 上求解 WLCP，其中 $V' = V \setminus V_j$ 且 $\mathcal{C}' = \mathcal{C} \setminus \mathcal{C}_j$。$(G, L, w)$ 的最优列表着色可以通过合并两种最优着色得到。因此，如果 f^* 是 $(G - V_j, L|_{V'}, w|_{\mathcal{C}'})$ 的一个最优列表着色，则 (G, L, w) 的最优目标值为 $w_j.\chi(G_j) + w(f^*)$。

还可以对仅有部分顶点的颜色列表中包含单一代表颜色的实例进行化简。设 $j \in K$ 和 $v \in V_j$ 满足 $k(v) = 1$。注意，每个列表着色都会为 v 分配一个来自 $\mathcal{C}_j$ 的颜色，但具体分配哪一个并不重要，因为它们之间无法区分。因此，我们可以利用这一性质，将 v 预着色为 j，因为 $\mathcal{C}_k \setminus \{j\}$ 中的颜色会产生权重相同的着色。一个将 v 预着色为 j 的实例，可以被视为一个新的 WLCP 实例，其中不包含 v，并且不允许将 v 的邻居着色为 j。此外，G_j 中的任何孤立顶点 u 也可以预着色为 j，因为对于所有 $w \in N(u)$，均有 $w \notin V_j$。这些思想可以推广如下：

引理 2.7. 设 (G, L, w) 是 WLCP 的一个实例，其中 $G = (V, E)$、$j \in K$ 和 Q 是由 $\{v \in V_j$ 在 G_j 的 k-值 $= 1\}$ 所诱导的子图的一个极大团。那么，如果 $m(j) < |Q|$，则 (G, L, w) 不可行。否则，在 (G, L, w) 上求解 WLCP 可以归约为在 G 的某个真诱导子图上求解一个 WLCP 实例。

证明。设 $I_j = \{v \in V_j \setminus Q : N_{G_j}(v) \subset Q, N_{G_j}(v) \neq Q\}$ 且 $V' = V \setminus (Q \cup I_j)$。注意，$I_j$ 中的每个顶点都是 $G_j - Q$ 中的孤立顶点。

由于需要使用 $|Q|$ 种不同的颜色来为 $\mathcal{C}_j$ 着色，因此如果 $m(j) < |Q|$，则该实例不可行。否则，令 g 为 $G[Q]$ 的一种着色，其中使用的颜色属于 $\mathcal{C}_j$。

我们考虑 WLCP 的实例 (G', L', w')，其中对于所有 $v \in V'$，$G' = G[V']$，$L'(v) = L(v) \setminus g(Q \cap N(v))$；对于所有 $k \in g(Q)$，$w'_k = 0$；否则为 $w'_k = w_k$。

令 f' 为 (G', L', w') 的一个最优列表着色。定义 $f(v) = g(v)$ 对所有 $v \in Q$，$f(v) \in g(Q \setminus N_{G_j}(v))$ 对所有 $v \in I_j$，以及 $f(v) = f'(v)$ 对所有 $v \in V'$，则可得到 (G, L, w) 的一个列表着色，其中 $w(f) = w_j.|Q| + w'(f')$。

为了证明 f 是最优的，令 $\tilde{f}$ 是 (G, L, w) 的一个列表着色。无妨，我们可以假设，对于所有 $v \in Q$，$f(v) = g(v)$。那么，$\tilde{f}|_{V'}$ 是 (G', L', w') 的一个列表着色，并且

$$w(\tilde{f}) = w(\tilde{f}|_{V'}) + w_j.|Q| \geq w(f') + w_j.|Q| = w(f).$$

注意，我们还证明了：如果实例 (G', L', w') 不可行，那么 (G, L, w) 也不可行。 □

对于来自图着色问题的加权列表着色问题实例，众所周知的对极大团进行着色的预处理，可以视为应用了这一引理。

给定一个可行的加权列表着色问题（WLCP）实例，根据前一个引理，我们可以在多项式时间内将其归约为一个实例，使得每个顶点的颜色列表中至少有两个代表颜色。

然而，需要注意的是，每次应用引理时，$\mathcal{C}$ 的不可区分颜色划分都可能发生变化。特别地，如果 $m(j) \geq |Q|+1$，那么 $\mathcal{C}_j \setminus g(Q)$ 中的所有颜色仍然不可区分，而每个 $k \in g(Q)$ 都将属于其他等价类。具体而言，新实例具有：

1. for $k \in g(Q)$, $V'_k = V_k \setminus (N_{G_k}[g^{-1}(k)] \cup \tilde{I}_j)$ and $w'_k = 0$, and
2. for $k \in \mathcal{C} \setminus g(Q)$, $V'_k = V_k \setminus (Q \cup \tilde{I}_j)$ and $w'_k = w_k$.

在图 2 的例子中，v_4 只能用 $1 \cong 2$着色；不失一般性，假设选择前者。应用上述引理并取 $Q = \{v_4\}$，得到 $V'_1 = \{v_1\}$ 和 $V'_2 = \{v_1, v_2, v_3\}$，这意味着颜色 1 和 2 不再不可区分。事实上，颜色 1 现在与颜色 7 不可区分，而颜色 2 现在与颜色 3、4、5 和 6 不可区分。

3. 加权列表着色问题的集合覆盖公式

Mehrotra and Trick [22] 基于将着色刻画为顶点划分为稳定集的思想，提出了图着色问题的整数线性规划公式。我们根据注 2.5 对这些思想进行调整，以适用于加权列表着色问题。

给定加权列表着色问题的一个实例 $(G, K, \mathcal{V}, m, w)$，其中 $G = (V, E)$ 且 $X \doteq \{(S,k) : k \in K,\ S \in \mathscr{S}_k\}$，列表着色可用二元变量$x^k_S$ 表示，每个 $(S,k) \in X$对应一个变量，使得 $x^k_S = 1$ 当且仅当 S 中的所有顶点都使用 $\mathcal{C}_k$ 中的同一种颜色着色（该颜色不一定是 k 本身）。于是，加权列表着色问题可表示如下：

$$
\begin{aligned}
\text{(WLCP-SC)} \qquad & \min \sum_{(S,k)\in X} w_k x^k_S \\
& s.t. \\
& \sum_{\substack{(S,k)\in X:\\ v\in S}} x^k_S \geq 1 && v \in V && (1)\\
& \sum_{S\in\mathscr{S}_k} x^k_S \leq m(k) && k \in K && (2)\\
& x^k_S \in \mathbb{Z}_+ && (S,k) \in X && (3)
\end{aligned}
$$

约束 (1) 断言，对于每个顶点 v，至少选择一个包含 v 的颜色图稳定集。我们将其称为顶点约束。

约束 (2) 允许对于每个代表颜色 $k \in K$，至多取 $m(k)$ 个 G_k 的稳定集。我们将其称为颜色约束。关于变量的整数性约束，由于我们考虑的是非负权重，因此始终存在一个最优解，使得对于所有 $(S,k) \in X$， $x_S^k \in \{0,1\}$。对于可行实例，(3) 中的整数性约束和非负权重保证存在一个最优解，使得对于所有 $(S, k) \in X$， $x_S^k \in \{0,1\}$。

注意，如果对于某个 $k \in K$， $m(k) \geq |V_k|$，则相应的颜色约束变得冗余，可以将其省略。特别地，对于来自 GCP 的实例，存在一个代表颜色，其相应的颜色约束是冗余的。因此，对于此类实例，WLCP-SC 与 [22] 中给出的 GCP 集合覆盖公式相一致。

与 GCP 的集合覆盖公式一样，WLCP-SC 的变量数可能随着 G 的顶点数呈指数增长；对于中型或大型实例，穷举所有变量可能变得难以处理。分支定价方案是解决此类模型的通常方法。

$B\&P$ 算法的两个主要组成部分是用于求解线性松弛的列生成程序和分支策略。后者应当具有稳健性，即子问题也应当是 WLCP 的实例，以便重复使用列生成程序。关于该技术在着色问题中的背景，读者可参阅 [22, 16, 21, 11, 1]。

在接下来的章节中，我们将介绍针对这两个组成部分提出的方法。

4. 求解线性松弛

给定 $(G, K, \mathcal{V}, m, w)$ 为 WLCP 的一个实例，其中 $G = (V, E)$ 且 $X \doteq \{(S,k): k \in K, S \in \mathscr{S}_k\}$，令 LR 为 WLCP-SC 的线性松弛。对于每个 $(S, j) \in X$，将颜色约束两边乘以 -1 后，系数矩阵中对应于变量 x_S^j 的列可写为 $(x^S, -e_j)$，其中 $x^S \in \{0,1\}^V$ 是 S 的特征向量， $e_j \in \{0,1\}^K$ 是 j^{th} 的标准向量。

如前所述， LR 可能包含指数级数量的变量，因此需要采用列生成技术加以处理。

4.1. 定价过程

对于任意 $\hat{X} \subset X$，令 $LR(\hat{X})$ 表示由 LR 得到的线性规划，方法是将 $X \setminus \hat{X}$ 中的所有变量设为零。 $LR(\hat{X})$ 的可行解将记为 $(\hat{x}, 0)$，其中 $\hat{x} \in \mathbb{R}^{\hat{X}}$ 设 $(\hat{x}, 0)$ 为 LR 的一个最优解，而 (π^*, γ^*) 为 $LR(\hat{X})$ 对偶问题的最优解，其中 $\pi \in \mathbb{R}^V$ 和 $\gamma \in \mathbb{R}^K$ 分别是对应顶点约束和颜色约束的对偶变量。

那么，对于任意 $(S, j) \in X$，变量 x_S^j 在 LR 中的约化成本为

$$c_S^j \doteq w_j - (\sum_{v \in S} \pi_v^* - \gamma_j^*).$$

为了确定 (ˆx^*, 0) 是否是 LR 的最优解，我们必须判断是否存在 $(S, j) \in X$ 使得 $c_S^j < 0$。然后，定价过程必须回答以下问题：

$$Is\ there\ (S, j) \in X \setminus \hat{X}\ such\ that\ \textstyle\sum_{v \in S} \pi_v^* > \gamma_j^* + w_j?$$

定价规则可归结为：针对每个 $j \in K$，求解 G_j上的最大权稳定集问题（MWSSP），其中每个 $v \in V_j$的权重为 π_v^*。对于 $|K| = 1$的情况，这与 [22] 中针对图着色问题（GCP）提出的方法一致。

尽管 MWSSP 是 $\mathcal{NP}$-困难的，但 [16] 中给出的枚举程序在分支定价算法的背景下表现出相当快的速度（参见例如 [16, 11]）。

枚举程序会返回权重大于给定阈值的第一个稳定集；如果无法做到这一点，则返回权重最大的稳定集。我们的实现采用阈值 $\beta(\gamma_j^* + w_j)$，其中$\beta = 1.1$（例如，在 [22, 11] 中也采用相同的值）。同时，如果 G_j 中的顶点数量不足以找到满足 $c_S^j < 0$ 的稳定 S，我们便不会执行枚举程序，例如，如果 $\sum_{v \in V_j} x_v^* \leq w_j + \gamma_j^*$。

根据初步实验，在重新优化之前一次加入多列可以减少该过程的迭代次数，从而缩短总体时间。在我们的实现中，对于每个 $j \in K$，我们至多搜索一个进入变量（S, j）。此外，在将新变量加入 X 之前，会在其对应的颜色图中将稳定集以贪心方式扩展为极大稳定集。

4.2. 初始化

ˆ 列生成过程需要以一个满足LR（ˆX）可行的集合 $X \subset X$ 启动。与 GPC 不同，这样的集合 ˆX 可能不存在，即 LR 可能不可行。

这一特殊性迫使我们设计一种初始化程序，该程序要么输出一个集合 $X \subset X$，使得 $LR(X)$ 是可行的，要么指出 LR 不可行。

初始化过程使用一个扩展表述，该表述对应于通过为每个 $v \in V$ 添加一种虚拟颜色 k_v 而得到的扩展实例，其颜色顶点集为 $V_{k_v} = \{v\}$（即 v 是其颜色列表中包含 k_v 的唯一顶点）。我们将此扩展表述的线性松弛记为 LR_{ext}。由于 G_{k_v} 只有一个非空稳定集，LR_{ext} 恰好比 LR 多出 $|V|$ 个变量，具体而言，是每个 $v \in V$ 对应的 $x_{\{v\}}^{k_v}$，称为虚拟变量。

通过将虚拟变量的值设为 1、其余变量设为 0，总能为 LR_{ext} 得到一个可行解。因此，ˆ$X = \{(\{v\}, k_v) : v \in V\}$ 是一个有效的初始集合，可用于启动列生成过程以求解 LR_{ext}。

LR_{ext} 的最优解与 LR 的可行性之间的关系取决于为 LR_{ext} 定义的适当目标函数，具体如下所述。

传统方法将虚拟颜色的权重设为 1，其余颜色的权重设为 0。在这种情况下，LR 可行当且仅当 LR_{ext} 的最优值为零。此外，列生成过程

ˆLR 可以从将 X 设为 LR_{ext} 的基本最优解的基本变量集开始（其中所有虚拟变量均为非基本变量）。然而， LR_{ext} 的最优目标值可能与 LR 的最优目标值相差甚远，因为优化 LR_{ext} 时没有利用原始权重。

另一种已知方法是为虚拟颜色的权重设置一个很大的M 值，即对所有 $v \in V$设置为 $w_{k_v} = M$ ，而其他颜色的原始权重为 $w_k \quad k \in K$。我们将 LR_{ext}^M 称为该辅助问题的线性松弛。该方法所涉及的基本问题是：是否存在某个 M 值，使得 LR 可行，当且仅当 LR_{ext} 具有一个所有虚拟变量均为零的最优解。事实上，如果存在这样的 M ，则 LR_{ext} 的该最优解（删除虚拟变量后）就是 LR的一个最优解。下面的结果使我们能够证明这样的 M的存在性。

Hadamard不等式（[13]）：
设 B 为一个 $n \times n$ $\{-1, 0, 1\}$-矩阵，则 $|det（B）| \leq n^{n/2}$。

于是，我们有如下结果：

引理4.1. 存在 $M > 0$ ，使得 LR 可行，当且仅当 LR_{ext}^M具有一个所有虚拟变量均为零的最优解。此外，如果（x^*，**0**）是 LR_{ext}^M 的一个基本最优解（其中**0**是一个分量取值为虚拟变量值的向量），则 x^* 是 LR的一个基本最优解。

证明。假设 LR 的基的大小为 $n \times n$ ，并令 $H = n^{n/2}$。根据Hadamard不等式，对于所有 $M > 0$，若 B 是 LR_{ext}^M的一个基，则 $|det(B)| \leq H$。此外，若 LR 可行，则其目标函数满足上界 $W := \sum_{k \in K} w_k m_k$。

令 $M = H(W + 1)$、 B 为 LR_{ext}^M 的最优基，(x^*, d^*) 为相应的最优基本解（其中 d^* 是其分量为虚拟变量取值的向量），且 $z_{ext}^* = \sum_{(S,j) \in X} w_j x_j^{*S} + M \sum_{v \in V} d_v^*$ 为 LR_{ext}^M 的最优值。

显然，如果 $d^* = \mathbf{0}$，那么 x^* 是 LR 的可行解，因此， LR 是可行的。为证明其逆命题，假设对于某个 $v \in V$， $d_v^* > 0$。根据克拉默法则，我们有

$$d_v^* = |d_v^*| \geq \frac{1}{|det(B)|} \geq \frac{1}{H}$$

然后，

$$z_{ext}^* \geq M d_v^* \geq \frac{M}{H} = W + 1.$$

假设 LR 可行，即存在一个 LR 的可行解 $\tilde{x}$ ，其目标值为 $\tilde{z}$。那么，$(\tilde{x}$,**0**) 是 LR_{ext}^M 的一个可行解，其目标值为 $\tilde{z}$。于是， $\tilde{z} \geq z_{ext}^* \geq W + 1$。回想一下， W 是 LR目标函数的上界，这就导致了矛盾。因此， LR 不可行。

□

将 M 设为前一个引理中提出的值， LR_{ext} 的一个基本最优解可以告诉我们 LR 是否可行；若可行，则可直接得到 LR 的一个最优解。

然而， M 的这个值可能过大，从而导致数值不稳定问题。尽管如此，对于任意的 $M > 0$，并考虑在前一个引理的证明中定义的 W，我们有如下结论：

注 4.2。设 (x^*, d^*) 和 z^*_{ext} 分别为 LR^M_{ext} 的一个最优解和最优值。则：

1. 如果 $d^* = 0$，那么 x^* 是 LR 的最优解。2. 如果 $z^*_{ext} > W$，那么 LR 是不可行的。

我们的实现将 $M = 1000$ 用于初始化过程。对于计算实验中所考虑的实例而言，这样的取值效果相当不错。事实上，在通过 B&P 求解的数百至数千个线性松弛问题中，备注 4.2 中提到的剩余情况从未发生过，即 LR_{ext} 的最优解包含某个非零的虚拟变量且 $z^*_{ext} \leq W$。

5. 划分解集

在精确求解着色问题的文献中，有两种划分解集的主要策略：按边分支 [22, 21, 11, 1] 和按颜色分支 [5, 23]。本节将介绍如何在 WLCP 的背景下处理这些策略。

按边分支。该策略最初在 [25] 中针对图着色问题提出，随后用于 [22]。给定一个图 G，其思路是选取两个不相邻的顶点 u 和 v，并将解空间划分为两部分：一部分是 u 和 v 具有相同颜色的那些着色，另一部分是它们颜色不同的那些着色。后一种情况下，寻找最优着色等价于在图 G 上添加边 (u, v) 所得到的图上求解图着色问题，该图记作 $G + (u, v)$。前一种情况等价于在 G 上求解图着色问题，其中将顶点 u 和 v 合并为一个顶点，记作 uv。也就是说，移除顶点 u 和 v，并添加一个新顶点 uv，将其连接到 $N_G(u) \cup N_G(v)$ 中的每个顶点。所得图将记作 G_{uv}。

在将这一策略应用于 WLCP 时，顶点 u 和 v 必须满足一个附加条件，即它们的颜色列表中至少共享一种颜色，也就是说，$L(u) \cap L(v) \neq \emptyset$。否则，关于 G_{uv} 的子问题将不可行，因为不可能为两个顶点分配相同的颜色；并且，根据引理 2.1，关于 $G + (u, v)$ 的子问题将与原问题等价。

为说明该分支策略对加权列表着色问题是稳健的，设 (G, L, w) 是一个实例，并设 u 和 v 是 G 中两个不相邻的顶点，使得 $L(u) \cap L(v) \neq \emptyset$，或者等价地，$u$ 和 v 是某个 $j \in \mathcal{C}$ 中的两个不相邻顶点。显然，一方面，求出一个满足 $f(u) \neq f(v)$ 的最优列表着色 f，等价于求解实例 $(G + (u, v)\ L, w)$ 上的加权列表着色问题。另一方面，求出一个满足 $f(u) = f(v)$ 的最优列表着色 f，等价于求解实例 (G_{uv}, L', w) 上的加权列表着色问题，其中对于所有 $z \in V(G_{uv}) \setminus \{uv\}$ 和 $L'(uv) = L(u) \cap L(v)$，均有 $L'(z) = L(z)$。

由于经过有限次应用后，会到达一个对所有 $j \in \mathcal{C}$, G_j 均为完全图（可能是平凡图）的实例，因此该分支策略必定收敛。在这种情况下，可根据命题 2.2 在多项式时间内求解加权列表着色问题。在我们的实现中，我们没有使用匈牙利算法，而是通过列生成过程求解线性松弛，并得到整数解（因为 LR_{ext} 的系数矩阵是全幺模的）。此外，枚举树采用深度优先搜索进行探索，从顶点合并的子问题开始。

按颜色分支。这种用于图着色问题的策略出现在直接枚举算法中，例如 DSATUR [5], 以及后来的 [23]。给定一个图 G，其思路是选择一个顶点 v 和一种颜色 j，然后将解空间划分为两部分：一部分是 v 使用 j 着色的那些着色方案，另一部分是不使用该颜色的方案。此外，通过在每个子问题中为 v 指定不同的颜色，可以很容易地将这种分支扩展为非二分分支。

将这种分支策略纳入分支定价算法并不直接，因为它对于图着色问题并不稳健，即分支后需要求解的子问题并不对应于图着色问题实例。然而，可以将它们视为列表着色问题实例。事实上，寻找满足 v 使用j 着色的最优着色方案，属于预着色扩展问题（如第 1 节所述，这是列表着色问题的一个特例）。类似地，寻找满足 v 不使用 j 着色的最优着色方案，也可以建模为一个列表着色问题，其中从 v的颜色列表中移除 j 。

在将这一策略用于加权列表着色问题时，可以利用不可区分颜色。解空间可以划分为两部分：一部分是将 v 染成 $\mathcal{C}_j$ 中的某种颜色的列表着色，另一部分则不是。只有当 $k(v) \geq 2$时，这种做法才有意义；否则，将 v 染成 $\mathcal{C}_j$ 中的某种颜色的子问题将与原问题等价，而另一个子问题则不可行。因此，在分支树的每个节点中，始终按照引理 2.7 对实例进行预处理，直到每个顶点的颜色列表中都多于两个代表颜色。

为说明这种分支对加权列表着色问题同样稳健，设 (G, L, w) 是一个实例，$j \in K$，且 $v \in V_j$ 满足 $k(v) \geq 2$。容易证明，一方面，求出一个满足以下条件的最优列表着色——将 v 染成 $\mathcal{C}_j$中的某种颜色——等价于求解一个加权列表着色问题，其实例为 (G, L_1, w)，其中 $L_1(v) = \mathcal{C}_j$，并且对所有 $u \in V(G) \setminus \{v\}$，都有 $L_1(u) = L(u)$。另一方面，求出一个满足以下条件的最优列表着色——v 不染成 $\mathcal{C}_j$ 中的任何颜色——等价于求解一个加权列表着色问题，其实例为 (G, L_2, w)，其中 $L_2(v) = L(v) \setminus \mathcal{C}_j$，并且对所有 $u \in V(G) \setminus \{v\}$，都有 $L_2(u) = L(u)$。

请注意，在实例（G, L_2, w）中，$k(v)$减少了 1；而在实例（G, L_1, w）中，$k(v) = 1$。因此，该策略最终会到达一个顶点 v ，其$k(v) = 1$，随后将对其进行预处理，从而得到一个顶点更少的图。这些事实证明了该分支策略的收敛性，因为经过有限次应用后，最终会得到一个平凡图。

同样，我们的实现采用深度优先搜索遍历枚举树，这次从为一个顶点分配代表颜色的子问题开始。

6. 用于分支的变量选择

本节提出了不同的策略，用于选择上一节介绍的各类分支策略所涉及的变量。在本节余下部分， x^* 表示线性松弛的一个最优分数解。在我们的实现中，除非另有说明，否则我们选择最接近 0.5 的变量。

按边分支。回顾一下，这种分支策略要求选择两个在某个颜色图（记为 G_k）中不相邻的顶点 u 和 v。我们提出选择属于某个稳定集 $S \in \mathcal{S}_k$ 的两个顶点，使得 x^{*k}_S是分数值。下面的引理证明了存在适用于这种分支的候选变量。

引理 6.1. 设 x^* 是 LR的一个基本分数最优解。那么，存在 $(S,k) \in X$，满足 $|S| \geq 2$，使得 x^{*k}_S是分数的。

证明。令 $X' = \{(S,k) \in X\colon |S| \geq 2\}$ 并假设对于所有 $(S,k) \in X'$，都有 $x^{*k}_S \in \{0,1\}$。令 $LR^\dagger$ 为将所有 $(S,k) \in X'$ 中的 x^k_S 替换为值 x^{*k}_S 后得到的线性规划。显然， x^* 也是$LR^\dagger$的一个最优解。此外， $LR^\dagger$ 的所有变量 x^k_S 都满足 $|S| = 1$。于是，其系数矩阵的列中只有两个非零元素：一个是对应于 S 中唯一顶点的行中的 1，另一个是对应于颜色 k的行中的 -1。因此， $LR^\dagger$ 的系数矩阵是全幺模的，并且由于约束的右端项均为整数， $LR^\dagger$ 的基本最优解也都是整数，特别是 x^*，这就产生了矛盾。 □

显然，根据前一个引理，从某个稳定集 S 中任意选取顶点 u 和 v 后， S 将不再属于 $G + (u,v)$ 的可行列表着色。不过，对于 u 和 v 接受相同颜色的子问题，这一点可能并不成立。例如，如果对所有满足 $x^{*\ell}_{\hat{S}} > 0$ 的 $(\hat{S},\ell) \in X$，都有 $|\hat{S} \cap \{u,v\}| \in \{0,2\}$，那么在包含 u 和 v 的那些稳定集中，用 uv 替换 u 和 v，即可为 G_{uv} 得到一个等价的可行解。因此，我们提出了另外两种更精细的策略。

► Edg-Std。该策略扩展了 [22] 针对图着色问题（GCP）提出的标准策略（事实上，对于来自 GCP 的实例，其行为完全相同）：

i. 找到满足 $|S| \geq 2$ 和 $0 < x^{*k}_S < 1$ 的 $(S,k) \in X$。 ii. 选择 $u \in S$。 iii. 找到满足 $u \in \hat{S}$ 和 $x^{*\ell}_{\hat{S}} > 0$ 的 $(\hat{S},\ell) \in X \setminus \{(S,k)\}$。 iv. 如果 $\hat{S} \neq S$，选择 $v \in S \triangle \hat{S}$。否则，选择 $v \in S \setminus \{u\}$。

由于 $0 < x^{*k}_S < 1$ 且顶点约束必须对 u 成立，因此保证了此类 $(\hat{S},\ell)$ 的存在。

当 $k = \ell$ （正如来自图着色问题的实例的情况一样）时，可以始终从 $S \triangle S$ 中选出顶点 $v \hat{}$，因为 $\mathcal{S}_k$ 中的稳定集各不相同。

更进一步地，如果这些稳定集是极大的，那么 v 必须与另一个稳定集中的某个ˆ顶点相邻。不失一般性，假设 $v \in S \setminus S$ 和 v 与 $w \in S$相邻。那么，在 G_{uv}中，uv 和 w 彼此相邻，而 S 不再可能成为 G_{uv}的可行列表着色的一部分。

然而，对于 WLCP 的一般实例，$k \neq \ell$ 可能会出现这种情况。此时，如果从 $S \triangle S$ 中选取 ˆv ，且稳定集是极大的，那么也可以证明 x^* 会被剪除（注意到，如果 v 同时属于 V_k 和 V_ℓ，那么 v 必定与另一个稳定集中的某个顶点相邻；否则，V_k 或 V_ℓ 不包含顶点 uv ）。

遗憾的是，从 $S \setminus \{u\}$ 中选取 v时，x^* 可能不会被剪除。虽然此时可以考虑另一个稳定集 S ，但 $S = S$ 的情况极少发生，而且分支后得到相同的分数解 x^* 只可能有限次（因为 G_{uv} 少了一个顶点）。

请注意，当 u和 v 属于许多顶点-颜色集时，按边分支的结构影响更大。回想一下，在 $G+$（u，v）中，顶点 u和 v 在每个同时包含这两个顶点的颜色图中都会变为相邻；而在 G_{uv}中，顶点 u 和 v 会从每个只包含其中一个顶点的颜色图中被移除。

提出了一种替代策略，优先选择在分支时影响最大的顶点。

► Edg-alt。找出使 $k(u) + k(v)$ 最大化的顶点 u和 v，满足存在 $(S, k) \in X$，且 $\{u, v\} \subset S$和 $0 < x^{*k}_S < 1$。

按颜色分支。在此分支策略中，必须选择一个顶点 v 和一个代表颜色 k，使得 $v \in V_k$。回想一下，根据命题 2.7，$k(v) \geq 2$ 对每个顶点 v均成立。

提出了三种策略来选择 v 和 k。第一种策略沿用了 [22]的部分思路：

► Clr-Std。

i. 找出满足 $0 < x^{*k}_S < 1$ 的 $(S, k) \in X$。
ii. 选择 $v \in S$。

当v 在 G_k中有多个邻点时，按颜色分支对结构的影响更大。回想一下，命题 2.7 可应用于 v 必须接收 $\mathcal{C}_k$中的某种颜色这一子问题，因此 N_{G_k}（v）中的顶点会从 G_k中移除。此外，k（v）值较低的顶点通常是最难着色的，因为如果先给它们的邻点着色，少数几个候选颜色很快就会变得不可用。同时，它们能为 v 无法接收 $\mathcal{C}_k$中的某种颜色这一子问题提供更多信息。基于这一考虑，提出了两种替代策略：

► Clr-Alt$_1$。找到使 $|N_{G_k}(v)|$ 最大的顶点 v和 $k \in K$，满足存在 $(S, k) \in X$，且 $v \in S$和 $0 < x^{*k}_S < 1$。若出现并列，选择 $k(v)$ 最小的顶点 v，然后选择 $m(k)$ 最小的颜色 k。

► CLR-ALT$_2$:

i. 找到使 $k(v)$ 最小的顶点 v，且满足存在 $(S, k) \in X$，其中 $v \in S$ 和 $0 < x_S^{*k} < 1$。ii. 找到使 $|N_{G_k}(v)|$ 最大的代表颜色 k，且对于某个满足 $v \in S$ 的 $S \in \mathscr{S}_k$，有 $0 < x_S^{*k} < 1$。若出现并列，选择 $m(k)$ 最小的颜色 k。

注意，只要 v 属于某个具有分数 x_S^{*k}的稳定集 S，分数解 x^* 就会从其中v 无法获得 $\mathcal{C}_k$颜色的子问题可行域中被切除。但这可能不足以将 x^* 从另一个子问题的可行域中切除。

7. 实例集

本节介绍用于计算实验的实例集。

广泛用于测试图着色算法的实例有两种类型：随机实例和基准实例。对于图着色问题，生成随机实例涉及两个参数： n （顶点数）和 p（边概率，即两个不同顶点相邻的概率），而已知最佳的基准实例来自 DIMACS 挑战赛 [8]。在我们的情况下，存在两个困难：参数 n 和 p 不足以描述问题，并且没有适用于加权列表着色问题的基准实例。我们提出如下方案：

WLCP 的随机实例

用于生成 WCLP 随机实例的参数如下：

- $n \in \mathbb{N}$：顶点数，
- $p \in (0, 1)$：边概率，
- $|K| \in \mathbb{N}$：可区分颜色的数量，
- $m \in \mathbb{N}^K$：重数向量，
- $w \in \mathbb{N}^K$：权重向量，
- $q \in (0,1)$：顶点颜色集概率，即对于每个 $k \in K$， G 中的顶点属于 V_k 的概率。

生成过程如下。首先，使用参数 n和 p生成一个随机图。对于每个 $k \in \{1, \ldots, |K|\}$，考虑一个颜色 k，其重数为 $m(k) = m_k$，权重为 w_k。每个初始为空的集合 V_k 以概率 q填入顶点 $v \in V$。

请注意， n和 q 间接决定了顶点—颜色集的大小及其重叠情况。事实上，对于每个 $k \in K$ 和 k， $|V_k| \simeq qn$ (v)；对于每个 $v \in V$， $\simeq q|K|$。此外，由于生成过程具有随机性，预计颜色图的平均密度与 G 相同。

▶ 集合 1（135 个实例）由每种 p、 $q \in \{0.25, 0.5, 0.75\}$组合对应的 5 个实例组成，这些实例由随机种子生成。其他参数值为 $n = |K| = 60$、 $w = m = \mathbf{1}$。此外，通过将 $m = k\mathbf{1}$ 更改为 $k \in \{3, 5\}$ ，每个实例都制作了 2 个副本（以保留实例的结构）。

权重向量始终为 1，因为颜色的权重会扭曲实例的结构，掩盖其余参数的影响。不同权重的研究留待未来工作。

μ-着色问题的随机实例

为分析所提出的算法在不同结构实例上的性能，考虑了一组用于 μ-着色问题的随机实例。生成过程中涉及以下参数：

- $n \in \mathbb{N}$：顶点数，
- $p \in (0, 1)$：边概率，
- $|K| \in \mathbb{N}$：可区分颜色数，
- $m \in \mathbb{N}^K$：重数向量，
- $t \in (0, 1)$：从一个颜色图到下一个颜色图之间移除顶点的概率。

生成过程与前一种方法类似，但每个顶点-颜色集都是基于前一个集合生成的。这意味着， $V_1 = V$ 并且，对于每个 $k \in \{2, \ldots, |K|\}$， V_k 中的所有顶点都以概率$1 - t$ 出现在 V_{k-1} 中（空颜色-顶点集对应的颜色将被舍弃）。

▶ 集合 2（54 个实例）对于 $n \in \{70, 80\}$、 $p \in \{0.25, 0.5, 0.75\}$ 和 $t \in \{0.05, 0.1, 0.2\}$的每种组合各包含 3 个实例。其他参数值为 $|K| = n/2$、 $m(1) = \omega(G)$（ G中最大团的大小）；对于每个 $k \in \{2, \ldots, |K|\}$， $m(k)$是 $[1, \ldots, 5]$中的一个随机整数。为第一个颜色选择重数，旨在降低生成不可行实例的可能性。

作为加权列表着色问题的实例，它们具有 $w = \mathbf{1}$、 $V_{|K|} \subseteq \ldots \subseteq V_1$，因此 $k(v)$ 是可变的：对于每个 $k \in \{1, \ldots, |K|-1\}$，一个顶点 $v \in V_k \setminus V_{k+1}$ 具有 $k(v) = k$，而一个顶点 $v \in V_{|K|}$ 具有 $k(v) = |K|$。事实上，平均 $|V_k|$ 和平均 $k(v)$会随着 t 的增大而减小。

图着色问题的随机实例

还考虑了一组图着色问题的随机实例，以分析所提出算法的性能。

▶ 集合 3（50 个实例）由以下各组合的 5 个实例构成： $n \in \{70, 80\}$, $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$。作为 WLCP 的实例，它们具有 $w = \bar{\mathbf{1}}$, $K = \{1\}$, $m(1) = \Delta(G) + 1$, 和$|V_1| = V$。

GCP 的基准实例

除了用于图着色问题的随机实例外，还考虑了一组来自文献的实例。

▶ 集合4（34 个实例）包含图着色问题（GCP） [8]的 DIMACS 挑战赛中的部分实例。

具体而言，我们考虑 Malaguti等人 [21]，提出的最具挑战性的实例，即图的团数和整数规划模型的线性松弛所给出的下界与启发式算法 MMT 给出的上界不一致的实例。此外，我们排除了无法在时间限制内求解其线性松弛的实例。

WLCP 的特定实例

最后，考虑了具有不同结构的加权列表着色问题实例。

▶ 集合 5（16 个实例）包含 3 个来自 Gomes 和 Selman 的预着色扩展实例 [15], 10 个来自? 的 MWSCP 实例], 以及 3 个来自 Lucci 等人的 WLCP 实例 [19]。这些最后的实例是唯一考虑非单位权重的实例，其权重对应现实世界成本。表 1 列出了该集合中考虑的实例。各列分别报告实例名称、问题类型、顶点数 n 和边数 m、密度 p、代表颜色数 $|K|$，以及每个实例顶点数的平均值 k（v）。

Name	问题	n	m	p	$\|K\|$	$\bar{k}$
order18holes120	预着色扩展。	324	5508	0.11	18	7
order30holes316	预着色扩展	900	26100	0.06	30	11
order30holes320	预着色扩展。	900	26100	0.06	30	11
scp41	MWSCP	200	0	0	1000	20
scpcyc06	MWSCP	240	0	0	192	4
scpa1	MWSCP	300	0	0	3000	60
scpb1	MWSCP	300	0	0	3000	150
scpc1	MWSCP	400	0	0	4000	80
scpclr1	MWSCP	511	0	0	210	26
scpcyc07	MWSCP	672	0	0	448	4
rail507	MWSCP	507	0	0	62182	797
rail516	MWSCP	516	0	0	47311	610
rail582	MWSCP	582	0	0	54341	676
trips215drivers60	WLCP	215	9442	0.41	3389	142
trips587drivers136	WLCP	587	65106	0.38	19959	434
trips818drivers191	WLCP	818	119130	0.36	36761	702

表 1：集合 5 的实例。

8. 计算实验

从现在起，我们将分支定价算法称为 BP-LCol。本节旨在评估 BP-LCol 在集合1至5上的性能，并比较所提出的分支规则和变量选择策略。

实验环境包括一台配备英特尔 i7 3.6GHz 处理器的计算机（使用单线程）、操作系统 Ubuntu 18.04、GCC 7.5.0，以及求解器 IBM ILOG CPLEX 12.7 [17]。为了进行进一步比较，基准程序 dfmax 在实例 r500.5.b 上运行 10 次，报告的平均时间为 2.83 秒（用户时间）（该实例可从 [8] 免费获取）。

如果求解器在 2 小时（7200 秒）的 CPU 时间内找到最优解并证明其为最优，则认为该实例已求解。

8.1. WLCP 的随机实例

表2展示了 WLCP 随机实例（集合 1）的结果。前 3 列分别报告参数 p、q 和 m的取值。每一行总结具有这些参数的 5 个实例所得的结果，最后一行除外，该行考虑所有实例。“k”列报告图中顶点的平均 k（v）。“求解率”列表示每种配置下求解成功的实例百分比。“时间（秒）”列表示每种配置下已求解实例的平均执行时间（秒）；若没有实例得到求解，则显示“–”。为便于阅读，采用边分支的列以颜色标示。每一行中，表现最佳的配置以粗体突出显示（若一种配置求解的实例更多，则认为其优于另一种配置；若数量相同，则平均耗时更短者更优）。

关于按边分支，所有实例均已求解， ALT 的平均执行时间比 STD高 33%，分别为 83 秒和 111 秒。结果表明，对于给定的 q 值，随着图变得更稠密，平均执行时间会缩短。此外，对于给定的 p和 q 值，随着 k 增大，平均执行时间通常保持不变或缩短，但对于具有 $p = 0.5$ 和 $q = 0.25$ 的实例中的 STD 除外。此外， $m = 3$ 和 $m = 5$的性能几乎相同。

关于按颜色分支， STD 求解的实例数量最多，占总数的 86%，但其平均执行时间最长。相比之下， ALT_1 和 ALT_2 求解的实例较少，分别为 76% 和 75%，但其平均执行时间明显更短，分别为 355 秒和 459 秒，而前者为 1022 秒。结果表明，对于给定的 p 值，随着 k 增大，求解的实例数量和平均执行时间均会恶化。此外，在 p 和 q 取相同值的情况下，重数的影响似乎比按边分支时更加显著，但也更难以预测，因为随着重数增加，算法性能可能显著改善，也可能显著恶化。

比较这两种策略时，在低密度图上的表现远优于 EDG；在中密度图上，CLR-ALT_1 和 CLR-ALT_2 的执行时间略优于 EDG-STD ，并且

p	q	m	$\bar{k}$	求解率					时间（秒）				
				Edg		Clr			Edg		Clr		
				Std	Alt	Std	Alt_1	Alt_2	Std	Alt	Std	Alt_1	Alt_2
0.25	0.25	1	15	100	100	100	100	100			146.6	98.1	**25.8**
		3	15	100	100	100	100	100			145.6	104.7	**22.4**
		5	15	100	100	100	100	100			145.2	104.7	**22.4**
	0.5	1	30	100	100	100	100	100			53.1	294.2	**115.0**
		3	30	**100**	100	100	100	100			7.1	285.0	148.8
		5	30	**100**	100	100	100	100	91.7	1	390.1	284.7	148.7
	0.75	1	45	100		80	60	60		71.0	3819.3	1668.6	1875.9
		3	45	**100**	100	80	60	60	**41.4**	71.3	3461.0	509.2	1213.4
		5	45	**100**	100	80	60	60	**41.4**	71.3	3900.3	508.2	1213.0
0.5	0.25	1	15	100	100	100	**100**	100	20.9	44.9	36.1	**15.3**	19.7
		3	15	100	100	100	**100**	100	35.7	18.7	25.8	**12.1**	19.7
		5	15	100	100	100	**100**	100	35.8	18.8	25.7	**12.2**	19.8
	0.5	1	30	100		100	100	100		84.5	1322.1	875.7	939.9
		3	30	100	1	100	100	100		44.8	357.9	664.9	1176.8
		5	30	**100**	100	100	100	100	**39.9**	44.7	358.8	665.0	1176.0
	0.75	1	45	100		0	40	0	96.4	**47.7**	–	580.7	–
		3	45	**100**	100	40	20	0	**44.6**	52.5	2468.1	406.2	–
		5	45	**100**	100	40	20	0	**44.5**	52.3	2417.8	410.7	–
0.75	0.25	1	15	**100**	100	100	100	100	**1.4**	3.0	5.7	6.2	2.5
		3	15	100	**100**	100	100	100	1.7	**1.0**	3.9	2.1	1.5
		5	15	100	**100**	100	100	100	1.7	**1.0**	4.0	2.1	1.5
	0.5	1	30	**100**	100	100	80	80	**4.1**	5.3	535.6	1470.7	1753.4
		3	30	**100**	100	100	60	100	**2.5**	3.1	1107.6	532.2	590.0
		5	30	**100**	100	100	60	100	**2.5**	3.1	1144.4	533.9	870.8
	0.75	1	45	100		40	40	20	5.7	**5.6**	39.3	10.4	0.6
		3	45	100		40	20	20	4.8	**4.1**	328.1	0.4	1.1
		5	45	100		60	20	20	5.9	**4.1**	140.2	0.4	1.1
All				**100**	100	84	76	75	83.7	1	022.8	355.6	459.7

表2：对集合 1 中实例的分支策略和变量选择规则进行比较。

n	p	t	$\bar{k}$	求解率		时间（秒）	
				EDG-STD	CLR-ALT$_2$	EDG-STD	CLR-ALT$_2$
70	0.25	0.05	13	**100**	0	2030.4	–
		0.1	8	100	**100**	84.1	**22.6**
		0.2	4	100	**100**	307.0	**11.6**
	0.5	0.05	13	**100**	33	1077.8	2734.3
		0.1	8	**100**	100	**429.3**	570.5
		0.2	5	100	**100**	366.5	**129.6**
	0.75	0.05	15	**100**	0	15.7	–
		0.1	9	**100**	100	**24.3**	46.7
		0.2	5	100	**100**	22.8	**18.7**
80	0.25	0.05	15	0	0	–	–
		0.1	8	0	0	–	–
		0.2	4	0	**100**	–	780.7
	0.5	0.05	14	0	0	–	–
		0.1	10	33	**33**	4239.0	**1610.2**
		0.2	4	67	**100**	3075.3	335.0
	0.75	0.05	14	**100**	–	177.1	–
		0.1	7	**100**	100	**30.2**	678.1
		0.2	4	**100**	100	**43.5**	47.7
All				**72**	59	621	383

表3：对集合 2 中实例的分支策略和变量选择规则进行比较。

所有策略在高密度图上的执行时间都非常短。对于 k 的中等和较高取值（即 $k \in \{30, 45\}$），表现最佳的策略显然是 EDG。

这些现象可以根据每种分支类型所生成的完整枚举树来解释，而枚举树的形态取决于实例的结构。在按边分支中，给定顶点数和颜色-顶点集，当颜色图的密度较低时，枚举树往往具有更大的高度，因为添加一条边需要更多步骤才能到达叶节点（此时每个颜色图都是完全图）。在按颜色分支中，给定一个图和$^-$个颜色，枚举树的高度会随着 k的增大而增加，因为禁止某个顶点使用一种代表颜色需要更多步骤才能到达叶节点（此时每个顶点只属于一个颜色图）。

基于这些结果，针对 WLCP 的通用分支定价算法在低密度图且低 k 的实例上应$^-$采用 CLR-ALT$_2$，而在其他情况下采用 EDG-STD 。在接下来的计算实验中，我们将分析在其他实例集中评估这两种策略时，是否也会观察到类似的行为。

8.2. μ-着色的随机实例

表3展示了随机实例在μ-着色问题（集合 2）中的相应结果。

结果再次验证了前一项实验中的分析，即 CLR 在低 k （对应较高 t）的实例上表现更好，而 EDG在高密度图的实例上表现更好。

总体结果显示， EDG-STD 解决的实例数量多于 CLR-ALT$_2$，分别为72%和59%。同样，按实例的结构对结果进行细分后可以看出，CLR-ALT$_2$ 在某些情况下更为适用。

对于 k 的较低取值（ $t = 0.2$），在低密度图和中密度图上， CLR-ALT$_2$ 比EDG-STD 有效得多，而在高密度图上的结果则相当接近。在高密度图上，当 k 取中值和高值时（ $t \in \{0.05, 0.1\}$）， EDG-STD 的表现明显优于 CLR-ALT$_2$ 。此外，在 $n = 70$的低密度图和中密度图上，当 k 取最大值时（ $t = 0.05$）， EDG-STD 优于 CLR-ALT$_2$ 。

8.3. 图着色问题的随机实例 图着色问题随机实例（集合 3）的结果见表 4。

n	p	求解率		时间（秒）	
		EDG-STD	CLR-ALT$_2$	EDG-STD	CLR-ALT$_2$
70	0.1	100	**100**	4.4	**0.7**
	0.3	100	**100**	1792.7	**81.0**
	0.5	100	**100**	459.8	**10.3**
	0.7	100	**100**	21.3	**1.8**
	0.9	100	**100**	0.05	**0.04**
80	0.1	100	**100**	92.6	**11.1**
	0.3	0	**100**	–	687.3
	0.5	80	**100**	3813.1	93.5
	0.7	100	**100**	219.2	**13.8**
	0.9	**100**	100	**0.07**	0.08
All		88	**100**	641	90

表 4: 集合 3 中实例的分支策略和变量选择规则比较。

回想一下，图着色问题的实例最初具有 $k(v) = 1$ ，对于所有顶点 v皆如此，因此，预计 CLR-ALT$_2$ 在低密度和中密度图上的表现会远胜于 EDG-STD。结果再次支持这一观点。在密度为 $p \in \{0.1, 0.3, 0.5, 0.7\}$ 的图上， CLR-ALT$_2$ 的表现远胜于 EDG-STD 。对于极高密度图（ $p = 0.9$），两种策略的表现同样出色。

此外， EDG-STD 的性能会随着 p的增加而提升，但将 $p = 0.1$ 与 $p = 0.3$ 进行比较时除外。尽管对于密度极低的图（在本例中为 $p = 0.1$），完整的枚举树更大，但可以观察到，BP-LCol能够快速找到高质量的可行解并据此进行剪枝，从而使已处理节点数非常少。

8.4. 图着色问题的基准实例

我们将BP-LCol获得的结果与文献中针对图着色问题的最佳分支定价算法进行了比较，即Malaguti等人的MMT-BP算法[21], ，该算法用于图着色问题的基准实例（集合4）。MMT-BP将分支定价方案与一种称为MMT的有效元启发式相结合，以获得初始可行解并初始化列生成过程。此外，还采用禁忌搜索算法来加速定价过程；当无法找到进入列时，则直接使用CPLEX 10.1求解最大权稳定集问题的整数规划模型。考虑了两种分支策略：变量分支（VAR）和边分支（EDG）。第一种分支策略是对整数规划模型中的变量采用的标准分支策略，但其鲁棒性不足；第二种则是Mehrotra and Trick最初在 [22]中采用的分支策略。始终选择小数部分最大的变量进行分支。

由于 MMT 能够生成高质量的初始着色，为了进行公平比较，我们为 BC-LCol 提供了相同的初始解。遗憾的是，将 MMT 提供的解适配到我们的初始化过程中并非易事，因此这一工作留待未来研究。

表5的前3列分别列出每个实例的名称、顶点数 n 和色数 χ，其余各列则列出两种算法的结果。对于每种分支策略，我们报告间隙（在时间限制内达到的最佳下界与上界之间的差距）以及以秒为单位的执行时间。由于 MMT-BP 在配备 1 GB 内存的奔腾4 2.4 GHz 处理器上运行，第5列和第7列报告的执行时间经过了调整，以便与我们的机器进行近似运行时间比较；我们估计我们的机器快 2.5 倍（他们报告称，求解基准实例 r500.5.b 用时 7 秒）。MMT-BP 的时间限制为 36000 秒，其中包括 MMT 寻找初始着色所用的时间；这相当于 BC-LCol 中 14400 秒的时间限制。为便于阅读， VAR 和 EDG-STD 对应的列以颜色标示。每一行中，算法的最佳结果以粗体突出显示。“tl”表示达到时间限制；“-”表示由于初始线性松弛的求解超出时间限制，无法计算间隙；MMT-BP 对应列中的“> 0”表示间隙为正，但 [21] 中未报告确切数值。

MMT-BP 使用 VAR 在 6 个实例中证明了最优性，使用EDG在 5 个实例中证明了最优性；除 DSJC125.9 和 myciel4 外， VAR 比 EDG 更方便。BC-LCol 使用 EDG-STD 在 8 个实例中证明了最优性，使用 CLR-ALT$_2$在 10 个实例中证明了最优性；除 DSJC125.9 外， CLR-ALT$_2$ 比 EDG-STD 更方便（不过这里的差异很小：57.5 秒与 55.8 秒）。

具体而言，BP-LCol 在 3 个实例上证明了最优性，而 MMT-BP 在时间限制内无法求解这些实例（1-FullIns 4、3-Insertions 3 和 myciel5）。对于两个算法都求解出的实例，我们的算法在 4 个实例上报告了更短的执行时间（DSJC125.5、DSJC125.9、myciel4 和 queen9 9），在 2 个实例上执行时间更长（queen10 10 和 queen11 11）。在

Name	n	χ	MMT-BP [21]				BP-LCol			
			VAR		EDG		EDG-STD		CLR-ALT$_2$	
			gap (%)	time (s)	gap (%)	time (s)	gap (%)	time (s)	gap (%)	time (s)
1-FullIns_4	93	5	20	*tl*	>0	*tl*	0	29.7	**0**	**12**
1-FullIns_5	282	6	33.3	*tl*	>0	*tl*	33.3	*tl*	33.3	*tl*
2-FullIns_4	212	6	16.7	*tl*	>0	*tl*	16.7	*tl*	16.7	*tl*
2-FullIns_5	852	7	28.6	*tl*	>0	*tl*	28.6	*tl*	28.6	*tl*
3-FullIns_4	405	7	14.3	*tl*	>0	*tl*	14.3	*tl*	14.3	*tl*
3-FullIns_5	2030	8	25	*tl*	>0	*tl*	25	*tl*	25	*tl*
4-FullIns_4	690	8	12.5	*tl*	>0	*tl*	12.5	*tl*	12.5	*tl*
4-FullIns_5	4146	9	**22.2**	*tl*	>0	*tl*	-	*tl*	-	*tl*
5-FullIns_4	1085	9	11.1	*tl*	>0	*tl*	11.1	*tl*	11.1	*tl*
1-Insertions_4	67	5	40	*tl*	>0	*tl*	40	*tl*	40	*tl*
1-Insertions_5	202	6	50	*tl*	>0	*tl*	50	*tl*	50	*tl*
2-Insertions_4	149	5	40	*tl*	>0	*tl*	40	*tl*	40	*tl*
3-Insertions_3	56	4	25	*tl*	>0	*tl*	25	*tl*	**0**	551
3-Insertions_4	281	5	40	*tl*	>0	*tl*	40	*tl*	40	*tl*
4-Insertions_3	79	4	25	*tl*	>0	*tl*	25	*tl*	25	*tl*
abb313GPIA	662	9	**11.1**	*tl*	>0	*tl*	-	*tl*	-	*tl*
DSJC125.5	125	17	0	7220	>0	*tl*	**0**	**735**	0	748
DSJC125.9	125	44	>0	*tl*	0	1558	0	57.5	**0**	**55.8**
DSJC250.9	250	72	1.4	*tl*	>0	*tl*	1.4	*tl*	1.4	*tl*
DSJC500.9	500	?	3.1	*tl*	>0	*tl*	3.1	*tl*	3.1	*tl*
latin_square_10	900	?	11.8	*tl*	>0	*tl*	11.8	*tl*	11.8	*tl*
myciel3	11	4	0	1.5	0	1.5	0	1.5	0	1.5
myciel4	23	5	0	48.8	0	3.2	0	1.9	**0**	**1.7**
myciel5	47	6	>0	*tl*	>0	tl	0	5746	**0**	**199**
myciel6	95	7	42.9	*tl*	>0	*tl*	42.9	*tl*	42.9	*tl*
myciel7	191	8	37.5	*tl*	>0	*tl*	37.5	*tl*	37.5	*tl*
queen9_9	81	10	0	13.4	0	23.8	0	16.8	**0**	**6.9**
queen10_10	100	11	**0**	**273**	0	378	0	1349	0	668
queen11_11	121	11	**0**	**746**	>0	*tl*	8.3	*tl*	0	5115
queen12_12	144	12	7.7	*tl*	>0	*tl*	7.7	*tl*	7.7	*tl*
queen13_13	169	13	7.1	*tl*	>0	*tl*	7.1	*tl*	7.1	*tl*
queen14_14	196	14	6.7	*tl*	>0	*tl*	6.7	*tl*	6.7	*tl*
queen15_15	225	15	6.3	*tl*	>0	*tl*	6.3	*tl*	6.3	*tl*
queen16_16	256	16	**5.9**	*tl*	>0	*tl*	-	*tl*	-	*tl*

表5：与文献中针对集合4实例的图着色问题所提出的最佳分支定价算法进行比较。

为解释这一现象，请回想一下，对于来自图着色问题的实例，这些算法仅在预处理和求解线性松弛（初始化和定价过程）方面有所不同。随后，我们再次运行了未对实例进行预处理的 BC-LCol，但得到的结果相似，这表明差异必定源于线性松弛的求解。支持这一结论的另一点是求解初始线性松弛所需的时间：除 4-FullIns 5、abb313GPIA 和 queenN N（其中 N $\in \{10, 11, 12, 13, 14, 15, 16\}$）外，BC-LCol 几乎总是快好几倍。

值得一提的是，BC-LCol 在求解 4-FullIns 5、abb313GPIA 和 queen16 16 的初始线性松弛时超出了时间限制。具体而言，在特定颜色图中寻找稳定集时，MWSSP 的求解超过了时间限制（4 小时）。在这些情况下，可以修改定价过程，使其在经过一定时间后中止当前颜色，并继续处理下一个颜色。如果找不到任何进入列，则必须完整扫描那些被中止的颜色（如果有）。

对于其余实例，两种算法报告的间隙相近。

这些结果表明，在求解图着色问题（GCP）方面，BC-LCol 与文献中的其他方法相比具有竞争力。此外，与 [21], 采用的方法相比， [16] 开发的枚举程序在求解最大权稳定集问题（MWSSP）时似乎更加高效，但上述实例除外。这表明，本文提出的方法，即将 GCP 实例视为列表着色问题（LCP）实例，并根据按颜色分支的方式划分解空间，可能优于采用带有 Var 的 MMT-BP。为了确切验证这一点，应重新实现 MMT-BP，以便使用 [16]的枚举程序求解 MWSSP。

8.5. WLCP 的特定实例

最后，讨论了 WLCP 的特定实例（集合 5）的结果。这些实例包含 200 至 900 个顶点，颜色图的平均密度较低至中等，并且参数 k 的值通常较高（预着色扩展问题的实例以及 MWSCP 的少数几个实例除外）。尽管预计 BP-LCol 在此类实例上无法高效运行，但研究这些实例仍然很有意义，因为这有助于验证分支策略在这些特定结构上的表现。

预着色扩展问题的 3 个实例具有低密度的颜色图，且颜色较少，因此按颜色分支预计比按边分支表现更好。此外，色数等于颜色数，这意味着可行解很少，且全部都是最优解。两种策略都能在不到 1 秒内求解 order18holes120，而只有 CLR-ALT_2 能在 408 秒内求解 order30holes320。在时间限制内未能为 order30holes316 找到解。

MWSCP 的 10 个实例具有无边着色图，这被认为是按边分支最具挑战性的结构；并且参数 k 的取值范围很广，其中较高的取值预计对按颜色分支最具挑战性。 EDG-STD 求解了

2 个实例（scp41 用时不到 1 秒，rail516 用时 123 秒），而 CLR-ALT$_2$ 以相近的执行时间求解了这两个实例，并在 611 秒内求解了 scpc1。 EDG-STD在 3 个实例中报告了正间隙：scpcyc06（36%）、scpa1（8%）和 scpc1（3%）；而 CLR-ALT$_2$ 在 4 个实例中报告了正间隙：scpcyc06（41%）、scpa1（11%）、scpclr1（34%）和 scpcyc07（52%）。这意味着 CLR-ALT$_2^-$使用 $k \leq 80$为所有实例找到了可行解。在其他情况下，BP-LCol 在达到时间限制之前耗尽了内存，因为没有找到可行解来剪枝搜索树。如前所述，这些结果并非旨在与专门针对 MWSCP 的算法所取得的结果竞争。

来自运输应用的 3 个实例具有中等密度的颜色图和较大的颜色列表，这显然是对两种分支策略而言最复杂的结构。实例 trips215drivers60 使用 EDG-ALT 在 111 秒内求解，而使用 CLR-ALT$_2$ 则耗时 759 秒。同样，对于其余实例，分支定价算法在找到可行解之前就耗尽了内存。

9. 结论与未来工作

本文提出了一种分支定价算法（BP-LCol），用于求解列表着色问题的加权版本。该问题有许多应用，并且是多个问题的推广。据我们所知，这是首个基于整数线性规划求解该问题的精确算法。

在这项工作的开篇，重点关注限制于某些特定实例族的WLCP的计算复杂性，以此揭示输入结构的重要性；该结构不仅取决于图，还取决于颜色图，即由其颜色列表中具有某种给定颜色的顶点诱导出的子图。这些考量预示着，为各种可能的实例设计一种高效的通用算法并非易事。

不可区分颜色这一概念的引入从两个方面推动了这项研究。一方面，它使我们能够将 GCP 中广为人知的“先为一个极大团着色”的做法扩展到 WLCP。另一方面，所提出的整数线性规划公式中的变量和约束数量可以减少；当应用于 GCP 实例时，最终得到的公式与 Mehrotra and Trick [22] 提出的公式相同。

引入虚拟变量有助于找到一组列，以通过列生成初始化线性松弛问题的求解。本文证明，这些新变量存在一个理论权重 M，使得扩展线性松弛问题的最优解可以直接导出原线性松弛问题的最优解，或表明原问题不可行，从而避免经典的两阶段初始化。本文提出了两种分支策略：按边分支改编自 Mehrotra and Trick [22] 最初用于图着色问题的分支方法；按颜色分支则受到 DSATUR 的划分空间以及 M´endez-D´ıaz 和 Zabala [23]为图着色问题开发的分支切割算法的启发。

使用大量计算实验来评估 BP-LCol 的性能，涵盖了结构多样的实例。

WLCP 随机实例的结果证实，该算法的表现会因这些结构而异。随着颜色图变得更加稠密，按边分支的效果会更好；而随着颜色列表中的代表颜色减少，按颜色分支的效果会更好。其他实例集的结果也通过实验再次证实了这一表现。特别是，GCP 随机实例和基准实例的结果表明，将它们视为 LCP 实例可能会带来益处。事实上，可以利用针对这一更一般问题所取得的进展，例如按颜色分支和预处理。

这项研究仍有一些问题有待未来工作进一步探讨。例如，颜色权重对算法效率的影响，以及实例中是否存在其他会影响算法性能的结构。

进一步研究列生成过程是另一个可能的研究方向。目前，在重新优化线性松弛之前，每种颜色至多搜索一个进入列。对此方法的一些替代方案包括：允许每种颜色的进入列数量达到其重数；在找到给定数量的进入列后重新优化线性松弛；在这种情况下，按照某些标准对颜色排序；为求解最大权稳定集问题设置时间限制；利用具有相同颜色-顶点集但权重不同的颜色（以避免重复求解最大权稳定集问题）；维护一个进入列池，等等。

在计算实验中，初始化期间为虚拟变量设置建议权重 $M = 1000$ 的效果良好。为应对其可能失效的假设情况，应在 BP-LCol 中加入两阶段方法的实现。

最后，我们认为，为 WLCP 设计一种（元）启发式算法将大有裨益。初始可行解，即一个上界，可以从一开始就进行节点剪枝，并有助于避免在大型实例的计算实验中报告的内存问题。此外，该启发式算法还可用于生成适当的初始列集，以初始化列生成过程。即使该启发式算法只能找到部分列表着色，初始化仍可利用此解，为未着色顶点添加虚拟变量。

参考文献

参考文献

[1] Archetti, C., Bianchessi, N., 和 Hertz, A.（2014）。用于求解鲁棒图着色问题的分支定价算法。离散应用数学，165：49–59。第十届科隆/特温特图与组合优化研讨会（CTW 2011）。[2] Bir´o, M., Hujter, M., 和 Tuza, Z.（1992）。预着色扩展 I：区间图。离散数学，100(1)：267–279。[3] Bj¨orklund, A., Husfeldt, T., 和 Koivisto, M.（2009）。通过容斥原理进行集合划分。SIAM计算杂志，39(2)：546–563。

[4] Bonomo, F. 和 Cecowski Palacio, M.（2011）。介于着色与颜色列表着色之间： μ-着色。Ars Combinatoria，99：383–398。[5] Br´elaz, D.（1979）。图顶点着色的新方法。《ACM通讯》，22：251–256。[6]Chv´atal, V., Ho` ang, C. T., Mahadev, N. V. R., 和 De Werra, D.（1987）。四类完美可排序图。《图论杂志》，11(4)：481–495。[7] Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., 和 Saurabh, S.（2015）。参数化算法。施普林格出版公司。[8] DIMACS 挑战赛（2002年）。图着色实例。取自 https://mat.gsia.cmu.edu/COLOR04/。访问日期：2023年2月23日。[9] Edmonds, J. 和 Karp, R. M.（1972年）。网络流问题算法效率的理论改进。《ACM杂志》（JACM），19(2)：248–264。[10] Furini, F., Gabrel, V., 和 Ternier, I.-C.（2017年）。一种基于 DSATUR 的改进分支定界算法，用于顶点着色问题。《网络》，69(1)：124–141。[11] Furini, F. 和 Malaguti, E.（2012年）。通过分支定价求解精确加权顶点着色。《离散优化》，9(2)：130–136。[12] Garey, M. 和 Johnson, D. S.（1979年）。《计算机与难解性：NP 完全性理论指南》。W. H. Freeman。[13] Garling, D. J. H.（2007年）。《不等式：线性分析之旅》。剑桥大学出版社。[14] Golovach, P. A., Johnson, M., Paulusma, D., 和 Song, J.（2017年）。禁子图条件下图着色计算复杂性的综述。《图论杂志》，84(4)：331–363。[15] Gomes, C. P. 和 Selman, B.（1997年）。扰动条件下的问题结构。载于第十四届全国人工智能会议暨第九届人工智能创新应用会议论文集（AAAI’97/IAAI’97），第221–226页，美国罗得岛州普罗维登斯。AAAI 出版社。[16] Held, S., Cook, W., 和 Sewell, E. C.（2011年）。图着色的安全下界。载于第十五届整数规划与组合优化国际会议论文集（IPCO 2011），第261–273页，美国纽约州纽约市。施普林格。

[17] IBM（2016年）。ILOG CPLEX 优化工作室 12.7。检索自 https://www.ibm.com/products/ilog-cplex-optimization-studiol。访问日期：2023年2月23日。[18] Kubale, M.（2004年）。图着色，第352卷。美国数学学会。[19] Lucci, M.、Nasini, G. 和 Sever´ín, D.（2018年）。基于图列表着色模型规划公交车司机的工作日。SADIO 电子期刊（EJS），17(1):77–91。[20] Lucci, M.、Nasini, G. 和 Sever´ın, D.（2019年）。列表着色问题的分支定价算法。理论计算机科学电子札记，346:613–624。拉丁美洲算法、图与优化研讨会（LAGOS 2019）第十届会议论文集。[21] Malaguti, E.、Monaci, M. 和 Toth, P.（2011年）。顶点着色问题的精确解法。《离散优化》，8(2):174–190。[22] Mehrotra, A. 和 Trick, M. A.（1996年）。图着色的列生成方法。INFORMS 计算学报，8(4):344–354。[23]M´endez-D´ıaz, I. 和 Zabala, P.（2006年）。图着色的分支切割算法。离散应用数学，154(5):826–847。第四届 ALIO/EURO 应用组合优化研讨会。[24] Tuza, Z.（1997年）。具有局部约束的图着色——综述。Discus- siones Mathematicae Graph Theory，17(2):161–228。[25] Zykov, A. A.（1949年）。线性复形的一些性质。Matematich- eskii sbornik，66(2):163–188。

a)

