基于混合进化算法的图染色问题求解器

A Solver for Graph Coloring Problem based on Hybrid Evolutionary Algorithm

苟桂霖

华中科技大学 计算机学院

ACM1501

U201514613

fluorinedog@gmail.com

Abstract

A very promising approach for combinatorial optimization is to embed Tabu Search (TS) into framework of Evolutionary Algorithm(EA), calling Hybrid Evolutionary Algorithm(HEA). In the article, we try to implement a fast solver for Graph Coloring Problem(GCP) with famous GPX crossover operator and tabu search algorithm. To help this algorithm set foot into industry, our C++ implementation emphasizes good coding style and highly abstract OOP, without loss of efficiency. Experiments carried on DSJC series of graph prove this implementation very competitive with those written in pure C.

Keywords: Graph Coloring Problem, Hybrid Evolutionary Algorithm, Tabu Search, combinatorial optimization

1 前言

图染色问题（Graph Coloring Problem, GCP）作为一个 NP-hard 问题，在诸如任务调度、寄存器分配器、模式识别等领域有着重要的应用。GCP 不存在多项式时间的精确算法。目前已知的图染色精确算法，如分支限界等，随着图规模的增大，计算量会指数发生指数爆炸现象，即便是对于如 DSJC500.5.col 这种 500 个点的中规模图来说，也是不可接受的。因此，学界纷纷选择研究启发式近似算法。

在图染色问题的近似算法中，以禁忌搜索（Tabu Search, TS）为子算法的混合进化算法（Hybrid Evolutionary Algorithm,HEA）可以在较短时间内搜索产生质量较高的解。基于这种算法，本文根据现有的论文的伪代码描述，提出了该算法的一种面向对象的工程实现。

使用 C++17 特性，在不损失性能的前提下，我们通过面向对象的方法解耦计算过程，隐藏了编码细节。最终，在底层框架的支持下，顶层的真实程序代码与高度抽象的算法伪代码基本一致，极大增强了算法实现的可读性与可扩展性，简化了未来的研究工作和工程代码复用。

2 问题描述

对于图染色问题，我们做出如下约定：

定义辅助函数$\delta(a,b)=\begin{cases}1, & a=b\\ 0, & a\neq b\end{cases}$

给定一张无向图$G=(V,E)$, 其中V为顶点集合，E为边集合。定义$adj(v)=\{u|(u,v)\in E\}$,也就是v相邻的点。

定义染色函数 $\lambda: V\rightarrow\{1,2,3,\dots,k\}\in\Lambda_k$. 其中，$\Lambda_k$是染色数目小于等于$k$的染色方案的全集。

我们定义$conflicts(G,\lambda)=\sum_{(u,v)\in E}\delta(\lambda(u),\lambda(v))$. 对于一个$k$, 如果存在一个染色函数$\lambda_0$, 使得$conflicts(G,\lambda_0)=0$成立，那么我们说$G=(V,E)$是 k 可染色的 (k-colorable), 定义其中最小的 k 为$\chi(G)$ 也就是G 的最小染色数。

其中，染色函数$c: V\rightarrow\{1,2,3,\dots,k\}$又被称为图$G$的配置(configuration). 我们所需要做的，就是通过进化算法，在一个特定的k所决定的配置空间Λ_k中, 寻找到一个配置$\lambda_{best}\in\Lambda_k$, 使得$cost=conflicts(G,\lambda_{best})$最小化。当$cost$为 0 时，我们就解决了$G=(V,E)$的 k 染色问题，得到一个合法的图染色解λ_{best}.

3 算法描述

本文提出并实现的混合进化算法是以禁忌搜索算法和 GPX 算子为基础的。禁忌搜索使得种群中的个体向着优良的方向变异，而 GPX 算子则杂交两个优良的个体获得子代，替代当前最差的个体，使得优良的基因得以扩散。当有向变异和杂交产生理论最佳个体后，算法终止。

3.1 禁忌搜索

3.1.1 概述

禁忌搜索(Tabu Search, TS)算法是一种运用广泛的算法思想。相对于普通的领域搜索 (Local Search) 算法，它加入了 “禁忌表” 的特性。当搜索陷入局域最小值时，通过将最近进行过的操作 “禁忌“，迫使搜索过程选取暂时较差、但是可能走出局域最小值的移动。

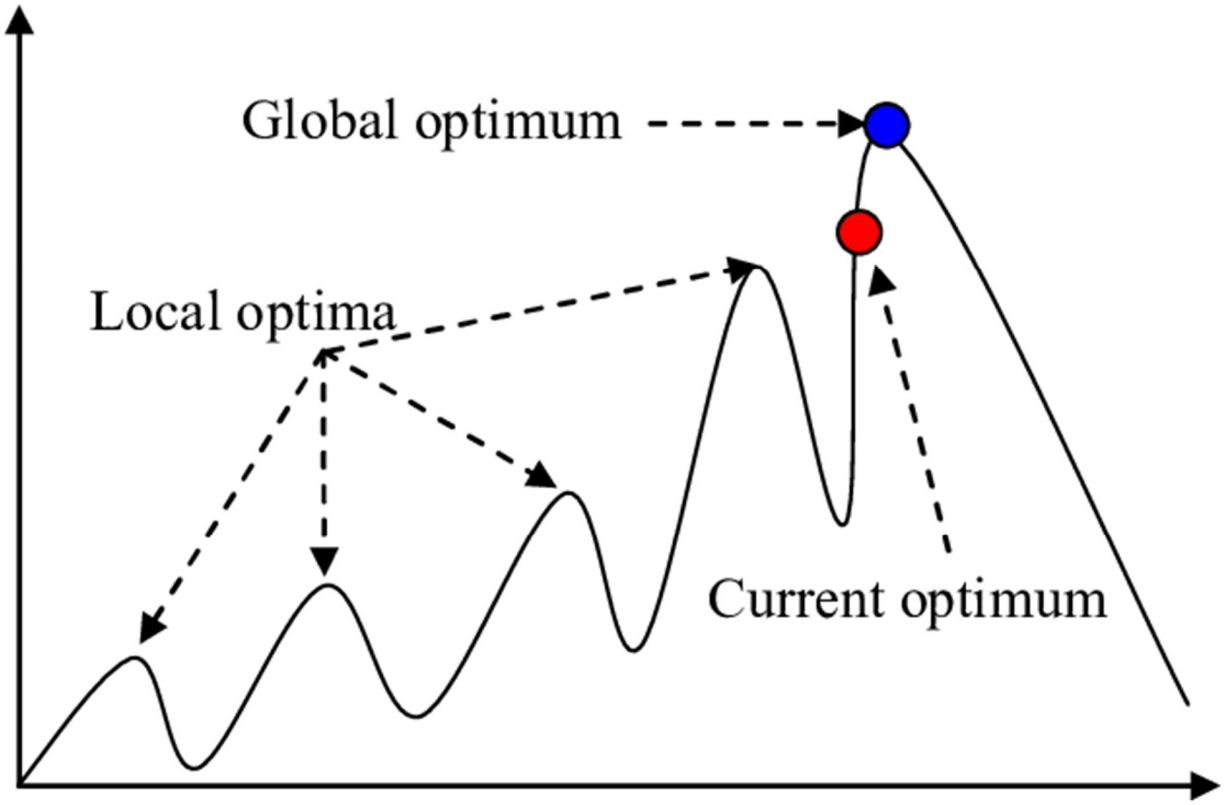

Figure 1 如图所示，当搜索陷入局部最优值时，需要暂时“劣化”，也就是选择较差值来逃脱局部最优值。

邻接动作与禁忌表是禁忌搜索的核心概念。对于某个特定的领域问题，我们需要选择合适的邻接动作和禁忌元素。对于图染色问题，我们选择“修改一个节点的颜色”作为邻接动作，并把这个动作的重复使用作为禁忌表元素。

3.1.2 邻接动作

为简化说明，定义动作$m: \Lambda_k \to \Lambda_k \in M_k$，其中$M_k$是$k$染色问题中，所有变更染色配置的函数的集合。定义邻接动作$move[v, color] \in M_k$，其中$v \in V$是$G = (V, E)$中的节点，$color \in \{1, 2, \dots, k\}$代表颜色，且满足以下条件

$$m = move[v, color] \in M_k$$

$$\lambda_y = m(\lambda_x) \in \Lambda_k$$

$$\lambda_y(u) = \begin{cases} \lambda_x(u), u \neq v \\ color, u = v \end{cases}$$

或写作

$$move[v, color](\lambda_x)(u) = \begin{cases} \lambda_x(u), u \neq v \\ color, u = v \end{cases}$$

换言之，邻接动作$move[v, color]$是将某染色配置λ_x 中其中一点v的颜色更改为$color$的操作，故此得名。

3.1.3 邻域搜索

领域搜索算法是图染色问题最简单的算法之一。在不需要禁忌表的邻域搜索算法中，对于第 n 次迭代得到的染色配置λ_n, 我们需要找到一个最佳的邻域动作$m_n = move[v_n, color_n]$, 并计算出$\lambda_{n+1} = move[v_n, color_n](\lambda_n)$, 使得$cost_n = conflicts(G, \lambda_{n+1})$达到最小值。其伪代码如下

```
Data: G = (V, E) and λ_init
Result: configuration λ
1. initialize λ_0 ← λ_init
2. n ← 0 as iteration steps
3. while{there exist improving moves}
4.    choose m_n = move[v_n, color_n] to minimize cost_n = conflicts(G, m_n(λ_n))
5.    let λ_{n+1} = m_n(λ_n)
6.    n ← n + 1
7. return λ_n
```

但是，邻域算法很容易走入局域最小值。因此，我们需要在这个算法的基础上进行修改，

3.1.4 禁忌表

作为图算法的禁忌搜索，我们有两种禁忌的方案。

1. 以染色配置 $\lambda \in \Lambda_k$ 为基本禁忌单位，一定时间内不允许全局同形;
2. 以邻接动作 $m \in M_k$为基本禁忌单位，一定时间内不允许采用同一种邻接动作。

如果我们选择方案 1，由于染色配置的数量是指数级别，为了算法的可靠性，我们需要保存大量的禁忌表。同时，即使是比较两种染色配置λ_1与λ_2是否一致， 我们也需要$O(|V|)$的时间，实践上基本不可行。

而如果选择方案 2，只需要使用一个数组，记录$k * |V|$种有限的邻接动作的截止时间即可，检测某邻接动作$m_i = move[v_i, c_i]$是否被禁忌时，只需要一次常数时间的查询即可完成任务。因此，我们选择方案 2 作为我们的是的禁忌方案。

3.1.5 禁忌搜索算法

相对于邻域搜索算法，我们做出的主要改进是加入了禁忌表的结构。每当我们做出了一次邻接动作$m_i = move[v_i, c_i]$，我们会将该动作和一个计算好的解锁时间放入禁忌表中。在查找新的最好邻接动作时，我们需要在没有被禁忌（或者已经超过了解锁时间）的邻接动作中，找到最佳的值。

一般情况下，我们不会选择被禁忌的邻接动作。但是，如果某一个被禁忌的邻接动作可以改进历史最优值，且它也是当前最好的改进方法，我们就可以选择它，以获取新的更好的历史最优解，甚至直接得到答案。

值得注意的是，在禁忌搜索中，我们所选择的邻接动作$m = move[v, c]$, 必须满足

$$\left(\sum_{u \in adj(v)} \delta\big(\lambda(v), \lambda(u) \big) \right) \geq 1$$

换言之，我们的邻接动作必须作用于有冲突的节点上，而不是在已经没有冲突的节点上浪费时间。对于邻域搜索，我们无需操心这一点，因为在没有冲突的节点上不可能做出改进。但是，对于禁忌搜索，我们为了走出局部最小值，甚至会进行 “劣化” 操作。但是我们的“劣化” 操作必须建立给 未来新优化 提供帮助的基础上，而无冲突节点上的劣化显然不会有这个性质。

同时，对于代价相同的待选邻接动作，我们应该随机从中选择，不能对其中某些元素有特殊的偏好$bias,$ 以保证算法可以走出局部最小值。

最后，我们使用高度抽象的伪代码展示这一个算法:

```
Data: G = (V, E) and λ_0
Result: configuration λ
1.  initialize λ ← λ_0
2.  initialize tabu tenure T
3.  n ← 0 // as iteration steps
4.  history ← ∞
5.  while{history > 0 and n < MAX_STEP}
6.    in M_legal ⊂ M_k where moves are not in T or have exceeded DDL
7.      choose m_n^legal = move[v_n, c_n] ∈ M_legal where v_n is in conflicts
8.      to minimize cost^legal = conflicts(G, m_n^legal(λ))
9.
10.   in M_tabu ⊂ M_k where moves are in T and still active
11.     choose m_n^tabu = move[v'_n, c'_n] ∈ M_tabu where v'_n is in conflicts
12.     to minimize cost^tabu = conflicts(G, m_n^tabu(λ))
13.
14.   if cost^tabu < history and cost^tabu < cost^legal
15.     choose cost^tabu, m_n^tabu
16.   else
17.     choose cost^legal, m_n^legal
18.
19.   if cost^chosen < history
20.     history ← cost^chosen
21.   T[m_n^chosen] ← new DDL
22.   λ ← m_n^chose (λ)
23.   n ← n + 1
24. return λ
```

以上算法即为禁忌搜索算法的数学描述。

3.1.6 实现细节-禁忌表

为了高效实现禁忌表，加速 7、10 的选择过程，我们选择了二维数组保存所有的DDL的方法。

每当更新$m = move[v, c]$时，实际上我们只需要令$EnemyTable[v][c] = new\ DDL$即可。

需要判断某个邻接规则$m' = move[v', c']$是否有效时，我们也只需要提取出$DDL = EnemyTable[v][c]$, 与当前的迭代次数 n 进行比较即可。

对于$new\ DDL$的选择，经过实验，我们发现$new\ DDL = n + rand(0, 10) + conflicts(G, \lambda_n)$的效果比较好，其中$rand(0, 10)$是 0 到 10（不含 10）之间的整数。这一个函数在保证一定程度的随机性的同时，通过 conflicts 项，在搜索过程中逐步 “退火“，拥有较好的性能。

禁忌表的具体实现可以参加 TabuTenure.h 文件。

3.1.7 实现细节-仇人表

为了高效地支持 8、9 与 12、13 行的选择操作，我们实现了 “仇人表“ 结构。每一种染色配置λ都对应着一张仇人表，其定义如下

$$Enemy[u][c] = \sum_{v \in adj(u)} \delta(c, \lambda(v)), u \in V\ and\ c \in \{1,2, \dots, k\}$$

换言之，仇人表记录了节点邻居的各个颜色的数量。

具体实现可以参见 EnemyTable.h 文件，我们只对其数学形式做出详细分析。

这个结构有以下优点：

3.1.7.1 高效查询

在我们需要知道某个节点 v 是否属于冲突节点时，只需要验证

$$enemy_v^{self} = Enemy[v][\lambda(v)] \neq 0$$

是否成立即可；

接下来，如果v恰好为冲突节点, 我们可以通过遍历$c \in \{1,2, \dots, k\}$依次计算

$$c_v^{min} = \arg\left(\min_{c \in \{1,2,\dots,k\}\ and\ cond(v,c)} (Enemy[v][c])\right)$$

$$enemy^{min} = Enemy[v][c_v^{min}]$$

$$cost_v^{min} = conflicts(G, \lambda) - enemy_v^{self} + enemy_v^{min}$$

得到这个冲突节点相关的最佳邻接动作，所对应的颜色c_v^{min}和相应的新代价$cost_v^{min}$，其中，$cond(u, v)$是禁忌与非禁忌元素的选择器，利用 3.1.6 禁忌表实现。

对于所有的冲突节点，我们都如此处理，最后得到所有邻接动作中，满足禁忌/非禁忌条件的最好元素，也就是c^{min}与$enemy^{min}$. 这样，我们就利用完成了 7-13 行所代表的查询步骤。

时间复杂度上，第一步，我们需要遍历每一个节点，找出其中的冲突节点，复杂度为$O(|V|)$. 第二步，对于所有冲突节点，我们需要遍历其对应的所有颜色. 由于最多有$\min(|V|, conflicts * 2)$ 个冲突节点，每个冲突节点耗时为$O(k)$, 因此，其整体复杂度为$O(|V| + \min(conflicts, |V|) * k)$

搜索初期，由于我们随机染色，导致冲突数极高，每一个节点附近都会有大量的冲突，因此每次迭代的时间复杂度为$O(|V|*k)$, 一次迭代的时间较长。但是由于每一次迭代，我们都很有可能找到让冲突数下降的动作，使得冲突数很快下降，因此初期实际持续时间不长，一般数千次迭代内就会进入中后期。

进入搜索中后期时，搜索的冲突数在低位徘徊，基本处于两位数甚至个位数，相对于节点数小很多，这使得整体复杂度变为$O(|V|+conflicts*k)$，单次迭代时间变得很短，算法整体性能得以保证。

查询功能的实现参见 EnemyTable 的 pick_best 函数。

3.1.7.2 高效更新

当我们通过查询，确定了一个领接动作$m_n = move[v_n, c_n]$ 后，需要更新染色配置$\lambda \leftarrow m_n(\lambda)$，也要相应地更新仇人表。更新算法如下:

$$newEnemy[u][c] = \begin{cases} Enemy[u][c]+1, & u \in adj(v_n)\ and\ c = c_n \\ Enemy[u][c]-1, & u \in adj(v_n)\ and\ c = \lambda(v_n) \\ Enemy[u][c], & otherwise \end{cases}$$

可以看出，如果我们在$Enemy$表上进行原地更新，只需要更改前两种情形。设 G 中 v_n的度为 $d(v_n) = |adj(v_n)|$, 我们有时间复杂度$O\big(d(v_n)\big)$, 在稀疏图上非常高效，对于稠密图的性能也不错。

更新功能的实现可参见 EnemyTable 的 shift 函数。

3.2 混合进化

对于图 DSJC500.5.col, 禁忌搜索可以轻松处理染色数$k = 50$的情形。但是，如果在搜索 49 种颜色时，单纯的禁忌算法已经比较吃力，对于 48 种颜色成功率更是极低。

而混合进化算法进一步提升了性能，可以在 100s 左右搜索得到 48 种颜色。它的主要特色是将禁忌搜索看作是有向变异，将可以维持亲代优良性质的GPX 算子作为杂交手段，使得优良的“基因”可以得以延续与互相交流，从而更容易收敛到全局最优值。

3.2.1 混合进化算法框架

混合进化的伪代码如下:

```
Data: G = (V, E)
Result: best configuration λ
1.  P = {λ_1, ..., λ_s} ← random
2.  while(not Stop-Condition())
3.    for each λ_i in P
4.      λ_i ← TabuSearch(λ_i)
5.    select best λ_p1, λ_p2 from P as parent
6.    λ_c ← GPX(λ_p1, λ_p2)
7.    substitute worst λ in P with λ_c
```

这里，我们模拟一个“自然环境“，环境中的 s 个独立个体受到有向变异的影响，进行有向进化。一段时间后，”捕食者“ 淘汰掉了最不适应环境的个体，替代以最优秀的两个个体的杂交子代，然后进行下一轮进化。 接下来，我们需要讨论的有如何选择模拟有向变异。

其中，s 对应的种群数量 POPULATION 是一个可调参数，在 log 文件中有体现。

3.2.1.1 有向变异选择

对于图染色问题，由于搜索空间过大，只进行随机变异的效果很差。因此，我们选择了禁忌搜索(TabuSearch)作为变异时使用的算子。具体实现可以参见前文。

值得注意的是，这里的禁忌搜索的停止条件除了得到最优值以外，还有最大迭代次数 STRIDE 的限制。

3.2.1.2 有向变异选择

普通的随机杂交算子由于不能保持亲代的优良性质，导致性能很差。

为了说明这一点，假设有两个优良的配置$\lambda_1 = \{0,0,0,1,1\}$与$\lambda_2 = \{1,1,1,0,0\}$，可以发现它们实质是等价的，只是颜色的编号不同。但是，如果我们对其进行随机杂交，那么它们产生的后代与随机生成的后代质量无异，这一点充分说明随机杂交算子的性能极差。

因此，我们选择 GPX 算子作为产生子代的方法。GPX 算法，先把亲代的染色情况分组，颜色相同的放在一个组中。交替选择两个亲代的染色数最多的颜色，将这个颜色的所有节点加入子代作为一个新的颜色，并从亲代双方剔除被加入的节点。重复多次直到 k 种颜色都被选择后，再给剩下的节点随机赋值。

伪代码如下

```
input: parent configurations λ_x, λ_y
result: child configuration λ_c
1.  split λ_x grouped by color into S_x = {V_1, V_2, ..., V_k} for v ∈ V_i ⊂ V and λ_x(v) = i
2.  split λ_y grouped by color into S_y = {V'_1, V'_2, ..., V'_k} for v ∈ V'_i ⊂ V and λ_y(v) = i
3.  n ← 0
4.  S_c ← 0
5.  λ_c ← random
6.  while{n < k}
7.    if n is even
8.      (primary, secondary) = (x, y)
9.    else
10.     (primary, secondary) = (y, x)
11.   select largest |V_i| in S_primary
12.   for each v ∈ V_i:
13.     set λ_c(v) with color n
14.     delete v from S_x and S_y
15. return λ_c
```

由于在 GPX 算子后，还需要重新构建仇人表，代价至少为$O(|E|)$，而且相对于 LocalSearch 的迭代过程，使用频率相对较低，所以一般不会成为瓶颈。

具体实现可参加 TabuSearch.h 的 GPX 函数，我的实现的复杂度为$O(|V|+k^2)$。由于此处并非瓶颈，工程实现细节和复杂度证明略。

4 计算结果与分析

图模型文件	目前最优值	计算颜色数	算法	平均时间	平均迭代次数
DSJC125.1.col	5	5	HEA(POPULATION=5|STRIDE=10000|SCALE=1)	0.0069	7194
DSJC250.1.col	8	8	HEA(POPULATION=5|STRIDE=10000|SCALE=1)	0.1751	107807
DSJC250.5.col	28	28	HEA(POPULATION=5|STRIDE=10000|SCALE=1)	5.4277	1886081
DSJC250.9.col	72	72	HEA(POPULATION=5|STRIDE=10000|SCALE=1)	1.5536	441132
DSJC500.1.col	12	12	HEA(POPULATION=10|STRIDE=10000|SCALE=1)	31.5441	14565217
DSJC500.5.col	47	48	HEA(POPULATION=5|STRIDE=10000|SCALE=1)	96.3309	15125981

以上为 DSJC 系列数据集的测试时间。每一组数据都至少运行了 4 次，最后的时间取平均值。原始数据请参见附录。

500.1 及以前的数据，我们可以得到当前最优值。对于 500.5，本算法并不能得到当前最优解，但是比纯粹的禁忌搜索（搜索 49 需要 1000s）的效率高很多。

程序效率上，越大的图单次迭代次数越长，单位时间迭代次数越低。但是对于 DSJC500.5，我们依然能够保证每秒 15 万次迭代的速度（测试环境为 i7-7500U + 2133MHz DDR4），说明 C++的抽象并未导致程序效率的损失。

算法 HEA (Hybrid Evolutionary Algorithm) 有三个会影响计算过程的编译可调参数 (之所以将这些放入编译期是为了提高程序效率)。其中，POPULATION 在 3.2.1 中有提及。STRIDE 是每一次 TabuSearch 搜索子过程的迭代次数，过小的参数会导致解的严重颠簸，过大的参数会让算法退化到禁忌搜索。而 SCALE 是一个失败的算法优化，它试图通过成倍增加子代第一次搜索的迭代次数，解决子代产生后就被剔除的颠簸现象，但是实验中其效果比较差，已被舍弃。

5 结论

本文实现了以禁忌搜索为子算法的混合进化算法，成功地在较短时间内搜索得到了比较好的结果。

为了加速禁忌搜索的迭代部分，我们实现了仇人表和禁忌表。仇人表加速了查找最优移动的过程，让中后期的漫长查询能以$O(conflicts*k+|V|)$ 的时间复杂度完成一次迭代。仇人表的更新也比较高效，可以在$O(d(u))$时间内完成。禁忌表被实现为访问时间复杂度$O(1)$的数据结构，满足了频繁调用的需求。

在进化算法部分，主要难点在于选择合适的杂交算子。我们选择了 GPX 算子，以保持亲代的优良性质。

基于以上的算法描述，使用 C++17 特性，在不损失性能的前提下，我们通过面向对象的方法解耦计算过程，对上层隐藏了底层的编码细节，只暴露了接口。最终，在底层框架的支持下，顶层的真实程序代码与高度抽象的算法伪代码基本一致，极大增强了算法实现的可读性与可扩展性，简化了未来的研究工作和工程代码复用。

6 感想与建议

禁忌搜索确实是一个非常神奇的算法。对使用了 类梯度下降的 LocalSearch 做出简单的改进，就能走出局部最小值，在图染色问题上轻松得到接近最优值的解，其效率远高于 DSTAR 等贪心算法。

而混合进化框架为禁忌搜索提供了进一步优化。其思想在于保持优良的“基因”，对于图染色而言就是经过了检验的染色子结构。禁忌搜索提供的有向变异又提供了新的”基因“，这样不断混合，优良的”基因“就得到了高质量解。

实现初期，鉴于专业背景，我曾一度希望实现算法的 GPU 加速。但是，我阅读论文发现，算法的主要瓶颈并不是在计算力上，而是在启发函数的质量。同样的图，一个算法运行了数个小时依然无法得到高质量的解，而另外一个算法却在 10 秒 内结束了计算。更糟的是，经过串行优化后的算法，后期的每次迭代的计算量非常小，过细并行粒度导致加速比过低。而迭代间的数据依赖又给并行带来了极大的挑战性。虽然如此，对于 1000.9 以上的大图，如果能减小通信开销，GPU 并行依然有利可图。

最后，希望未来的 PPT 中，给出 “师徒进化算法“ 的论文链接或者详细描述。没有能够实现最新最好的图染色的算法，是我的一大遗憾。

7 参考文献

[1]. Philippe Galinier and Jin-Kao Hao, Hybrid Evolutionary Algorithms for Graph Coloring, Journal of Combinatorial Optimization, 1998

[2]. 李淑芝, 何署芳, 基于图染色问题的混合优化算法, 计算机应用研究, 2016

[3]. Zhipeng Lü and Jin-Kao Hao. A Memetic Algorithm for Graph Coloring, European Journal of Operational Research, 203 (1), 241–250, 2010.*

8 附录

代码请参见 https://github.com/FluorineDog/Hybrid-Evolutionary-for-Graph-Coloring

以下为原始测试数据。

```
datetime, graph, algorithm, random_seed, duration, iterations, conflicts, res
ult
2018-05-22 12:49:15, data/DSJC125.1.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 67, 0.00773755s, 8726, 0, (5)4 3 2 1 0 1 2 2 2 0 0 2 2 0 4 4 2 1 1 3 1 3
4 0 4 3 4 4 3 3 1 4 3 0 3 2 0 3 2 1 0 0 1 3 3 1 4 2 4 2 1 1 2 0 3 4 4 0 2 3 3
 1 4 1 4 3 2 0 0 3 2 0 4 0 2 2 4 2 1 3 2 4 4 3 0 1 1 1 2 0 4 2 0 1 0 1 2 3 3
1 1 4 1 1 0 2 2 2 4 1 0 1 3 2 3 1 0 2 4 2 4 4 3 2 2
2018-05-22 12:49:15, data/DSJC125.1.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 10, 0.00165344s, 884, 0, (5)4 3 2 4 0 1 2 2 2 0 0 2 2 4 4 4 2 4 3 1 4 1 4
 2 3 2 4 4 1 3 3 2 1 4 1 2 0 3 3 2 0 0 3 3 0 1 4 0 4 3 3 3 2 4 0 4 1 1 2 3 0
3 4 3 3 1 2 0 0 1 3 1 4 0 2 0 4 2 1 1 2 4 1 3 0 3 2 1 2 0 4 1 3 4 2 3 2 0 1 3
 0 4 1 2 0 2 2 2 4 1 4 3 0 2 0 1 0 2 4 0 4 4 0 3 2
2018-05-22 12:49:15, data/DSJC125.1.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 20, 0.00666627s, 5404, 0, (5)1 0 2 1 3 1 2 2 2 0 3 0 2 1 0 3 2 1 0 3 4 1
1 3 1 0 3 1 0 2 1 2 2 3 1 0 0 0 2 1 4 4 3 0 0 3 1 2 1 0 0 3 3 3 2 3 4 0 2 0 4
 1 4 3 1 4 2 2 3 0 0 3 1 0 2 4 1 2 4 0 2 4 1 0 0 0 2 1 2 2 4 2 3 1 3 0 4 4 0
4 3 1 4 2 4 2 2 2 4 3 0 1 4 2 0 1 2 1 4 2 0 1 0 0 2
2018-05-22 12:49:15, data/DSJC125.1.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 30, 0.0116442s, 13760, 0, (5)4 1 2 4 3 0 2 2 2 1 3 2 2 4 4 4 2 0 0 0 0 0
4 2 0 2 4 4 3 1 0 4 3 3 1 2 3 3 1 2 3 3 0 0 3 3 4 3 4 1 0 0 2 1 3 4 3 1 2 1 3
 0 4 0 4 1 2 3 1 1 1 0 4 3 2 3 4 2 0 1 3 4 4 1 3 2 2 0 2 3 4 1 0 0 2 0 2 3 0
0 0 4 0 2 3 2 2 2 4 0 4 0 3 2 3 0 3 2 4 3 1 4 2 1 2
2018-05-22 12:49:19, data/DSJC250.1.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 67, 0.170809s, 100357, 0, (8)6 6 3 3 4 6 0 0 3 6 7 0 2 7 3 2 3 1 1 6 5 2
7 3 0 2 2 5 5 6 7 6 4 0 6 0 3 3 4 1 7 6 0 5 4 6 0 7 1 3 1 4 7 4 7 4 1 6 2 0 1
 1 3 6 1 2 7 1 2 3 5 1 1 1 1 0 5 0 7 2 2 0 4 3 5 6 4 4 1 2 0 3 3 1 0 5 5 4 5
3 4 6 3 6 2 6 5 5 4 0 0 7 7 4 6 0 5 5 7 4 1 0 3 7 3 6 7 7 0 5 7 5 0 2 7 4 4 0
 2 6 5 3 0 3 7 0 6 3 6 3 5 0 2 1 0 1 1 0 0 0 1 0 2 3 1 2 2 2 2 4 5 5 3 0 1 7
1 3 6 7 3 5 1 7 6 7 3 4 6 5 2 2 0 3 5 3 4 5 0 0 6 2 3 7 4 4 5 3 0 1 0 4 0 1 6
 3 6 3 2 7 7 1 0 1 6 6 2 5 3 1 3 7 6 3 4 7 2 5 4 4 5 2 2 7 7 1 2 6 7 1
2018-05-22 12:49:19, data/DSJC250.1.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 10, 0.316977s, 215345, 0, (8)7 4 2 7 1 3 5 1 4 0 7 1 7 5 5 4 5 6 0 5 1 0
7 3 7 1 4 1 5 5 3 6 2 5 5 2 7 4 2 4 3 3 1 5 4 4 5 0 4 5 7 3 0 1 7 2 0 4 1 6 3
 4 5 0 4 5 6 3 4 4 2 4 0 6 4 6 0 3 1 1 5 4 6 3 5 2 0 1 3 5 3 2 4 6 7 7 6 1 3
5 2 6 3 6 6 1 6 5 1 3 7 6 5 0 1 0 5 0 2 6 4 5 3 3 5 2 1 7 2 6 3 3 6 4 0 0 2 3
 6 5 6 2 7 5 0 5 6 0 7 6 7 1 0 6 7 1 4 3 3 5 3 7 7 6 4 2 5 0 0 2 7 6 2 6 2 4
3 1 0 2 1 7 4 7 6 0 3 0 7 1 0 4 5 7 3 2 2 3 7 0 5 4 5 7 3 1 0 2 3 6 6 1 3 3 4
 5 2 2 4 4 1 3 7 0 2 7 0 7 4 1 4 0 5 1 2 1 6 5 2 0 2 6 6 6 1 4 5 7 0 1
2018-05-22 12:49:20, data/DSJC250.1.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 20, 0.0994522s, 53344, 0, (8)6 3 6 0 0 3 4 7 6 1 1 3 2 7 5 0 5 0 6 3 2 1
5 2 6 6 2 6 1 2 2 7 0 4 4 0 5 1 4 4 2 7 2 4 3 3 6 7 3 5 0 4 7 3 7 1 1 3 1 0 0
 1 5 1 3 4 3 3 2 5 6 6 1 0 3 0 1 0 7 7 4 0 0 7 3 2 4 4 0 1 7 5 7 2 4 3 1 0 3
0 6 1 3 5 1 6 2 6 4 5 0 5 5 6 0 2 4 4 6 4 3 4 0 7 3 5 7 7 0 3 5 4 4 3 1 3 4 3
 2 3 2 0 3 6 7 4 6 4 7 5 3 0 0 3 0 3 1 0 3 4 2 2 5 5 6 2 2 5 5 2 6 5 5 4 1 7
```

```
5 7 6 1 1 2 3 3 1 6 5 6 4 6 1 7 4 1 1 2 4 3 5 6 2 1 5 6 4 0 6 5 7 7 2 7 3 7 2
 4 3 1 5 2 5 7 0 1 4 2 6 1 0 7 0 0 0 5 7 1 2 4 7 7 1 0 5 6 0 6 7 5 1 6
2018-05-22 12:49:20, data/DSJC250.1.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 30, 0.113249s, 62183, 0, (8)4 7 2 1 0 0 3 1 2 7 0 1 0 7 5 1 1 1 4 7 6 6 0
 7 1 3 2 7 3 0 0 4 3 1 3 1 6 5 4 5 2 4 1 3 5 4 7 5 0 7 3 4 6 0 6 1 6 7 1 4 2
5 7 7 5 5 7 7 5 2 3 5 6 6 5 6 2 3 0 5 2 1 4 5 7 7 3 3 5 7 2 0 7 5 3 4 7 3 0 7
 4 5 6 0 3 6 4 0 3 5 1 5 5 3 1 4 7 6 2 3 2 3 6 4 1 3 3 0 2 7 0 0 3 2 4 2 3 4
6 7 3 3 1 7 6 3 3 6 2 3 0 1 1 2 1 0 2 6 0 7 0 3 3 5 7 4 5 2 2 4 5 5 7 6 6 2 3
 5 7 4 6 3 2 0 7 2 4 4 4 0 6 1 1 4 0 5 0 0 4 1 4 5 7 3 3 6 6 2 4 5 4 6 6 5 2
1 7 4 5 5 1 0 1 5 1 4 6 0 2 2 6 6 3 7 4 6 3 7 2 5 7 1 2 4 1 6 2 5 6 1
2018-05-22 12:52:40, data/DSJC250.5.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 67, 9.43546s, 3326869, 0, (28)15 2 21 12 15 21 20 16 12 1 2 14 10 14 25 1
5 27 2 23 25 22 6 5 19 6 19 13 4 15 17 9 19 12 11 25 23 27 21 27 8 12 5 18 9
25 1 9 21 20 20 0 20 26 11 8 0 17 1 19 17 11 25 23 3 21 23 24 11 7 24 4 2 13
25 15 24 3 11 20 26 21 23 17 9 6 23 24 25 16 19 10 24 7 3 17 27 26 21 11 26 2
7 5 22 1 26 12 15 6 16 25 18 18 26 18 1 27 6 20 16 6 8 8 25 21 16 7 9 26 0 14
 18 2 19 13 22 5 22 4 11 12 23 26 24 23 2 14 10 16 27 14 20 6 1 17 25 8 18 16
 0 3 12 23 6 6 1 26 10 7 11 22 12 0 18 19 13 3 25 5 24 19 24 18 14 5 10 13 11
 10 27 9 2 18 17 8 12 4 16 20 17 22 3 15 3 7 9 5 0 4 23 26 27 22 14 13 0 17 8
 24 16 19 1 2 4 10 7 7 0 14 15 8 12 27 26 27 21 7 24 15 0 2 9 3 5 3 13 26 17
7 20 19
2018-05-22 12:52:43, data/DSJC250.5.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 10, 3.50683s, 1226129, 0, (28)14 8 2 24 14 11 20 7 1 0 8 1 2 1 25 14 23 1
6 2 25 7 26 19 3 20 3 3 26 14 15 24 9 15 6 11 24 23 9 23 5 22 7 19 6 25 16 2
24 20 26 21 26 27 5 11 21 5 26 3 26 11 25 11 20 22 19 17 5 1 17 15 10 21 25 1
4 15 20 6 11 27 1 12 15 24 26 1 5 25 10 20 24 17 17 16 15 23 27 24 4 27 23 21
 0 9 27 11 14 13 26 25 13 19 27 19 12 23 0 12 10 24 5 7 25 2 12 23 18 27 11 1
3 19 8 3 20 19 16 26 2 16 15 6 27 17 6 8 13 10 7 23 5 10 2 5 15 25 7 5 12 21
20 24 12 7 24 4 27 17 4 24 4 16 18 19 3 0 9 25 9 17 22 17 7 13 1 4 18 12 13 9
 21 22 19 8 22 15 0 12 22 15 0 20 14 13 18 19 16 11 18 2 27 23 9 13 26 6 12 1
0 18 10 3 26 9 18 4 6 5 8 13 14 8 1 23 27 23 17 4 7 14 16 11 21 20 0 22 18 27
 18 22 16 18
2018-05-22 12:52:51, data/DSJC250.5.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 20, 7.41225s, 2547795, 0, (28)15 12 18 19 15 3 21 17 22 10 12 2 23 2 26 1
5 25 12 24 26 3 10 6 19 4 19 2 22 15 4 19 8 7 5 26 24 25 8 25 11 18 6 1 0 26
3 23 13 21 21 10 2 27 5 11 0 7 19 8 22 5 26 24 16 18 24 14 5 20 14 7 12 0 26
15 14 16 5 21 27 18 24 8 0 4 24 14 26 17 16 19 14 20 16 7 25 27 22 5 27 25 6
2 16 27 16 15 4 22 26 1 1 27 1 19 25 4 21 17 4 24 11 26 5 17 20 9 27 19 2 1 1
2 8 21 4 6 2 7 5 7 24 27 14 24 12 7 18 17 25 20 21 17 20 3 26 11 1 17 10 16 0
 24 4 4 10 27 13 20 5 2 19 9 1 8 7 16 26 6 14 8 14 1 7 6 0 9 5 23 25 10 12 1
18 11 16 2 17 21 7 2 13 15 22 9 13 6 18 9 0 27 25 4 19 13 0 3 11 9 17 0 3 12
9 23 20 20 18 23 15 11 3 25 27 25 13 20 14 15 4 12 10 13 6 16 9 27 9 20 21 9
2018-05-22 12:52:52, data/DSJC250.5.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 30, 1.35625s, 443532, 0, (28)7 6 17 17 2 12 9 3 13 4 12 7 15 16 25 1 26 8
 22 25 19 17 0 1 18 4 1 5 11 21 6 9 3 15 25 22 26 14 10 10 14 19 8 15 25 8 15
 13 3 5 11 5 27 7 15 11 10 5 24 21 10 25 22 3 0 22 13 10 16 20 4 2 11 25 2 6
23 23 5 27 13 22 12 15 18 22 24 25 9 1 15 17 8 12 21 26 27 6 18 27 26 4 4 8 2
7 16 19 18 20 25 7 21 27 16 1 26 17 5 0 24 6 7 25 14 9 0 19 27 19 10 2 2 1 18
 12 14 14 0 14 15 22 27 10 22 4 21 5 20 26 10 5 17 7 21 25 20 7 9 11 23 12 22
 2 6 8 27 13 0 24 4 23 17 19 6 0 9 25 9 14 20 6 11 10 19 15 24 23 24 26 11 20
```

```
 18 21 20 1 20 9 20 21 4 3 16 23 3 13 19 5 12 22 27 26 4 10 13 18 23 21 3 9 1
 24 17 3 2 7 0 16 7 2 6 19 26 27 26 13 18 14 3 8 19 11 13 4 8 16 27 21 0 14 1

2018-05-22 12:51:13, data/DSJC250.9.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 67, 1.38549s, 368153, 0, (72)40 45 53 71 58 40 23 24 29 51 41 36 18 9 13
0 39 32 2 8 48 25 63 34 52 35 14 55 49 8 27 24 70 21 22 44 30 0 40 45 11 32 3
9 27 4 13 25 57 62 39 71 30 28 29 66 21 35 20 9 12 42 64 16 37 49 33 17 18 70
 58 3 61 68 60 45 60 64 14 4 18 23 50 31 54 59 13 29 36 52 44 14 4 68 60 23 3
5 13 53 43 50 17 46 41 32 43 22 11 55 23 61 16 42 22 15 54 12 65 43 61 1 71 3
3 42 63 17 70 28 10 7 8 5 5 52 56 9 37 6 26 1 49 56 6 46 54 37 2 11 45 56 3 3
8 55 7 54 7 67 39 38 48 34 19 60 36 30 25 2 69 63 20 49 3 24 20 14 32 48 15 3
4 36 57 16 38 31 10 68 38 31 46 71 66 47 61 62 67 33 26 59 17 0 57 55 41 64 5
1 47 67 63 51 70 67 69 59 12 15 35 58 57 6 21 65 16 5 44 62 27 31 19 41 15 66
 40 47 29 22 69 21 27 69 10 28 26 1 46 65 44 43 71 53 19 47
2018-05-22 12:51:14, data/DSJC250.9.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 10, 1.5797s, 464065, 0, (72)53 40 41 54 45 53 65 18 46 28 63 25 50 44 28
4 60 6 50 61 32 29 64 3 22 2 37 51 67 68 35 18 34 43 62 39 66 13 5 40 42 41 3
0 62 60 28 29 15 37 60 54 66 24 17 68 43 2 45 52 23 24 20 70 1 67 55 59 10 12
 5 38 27 48 33 40 50 34 41 20 0 11 23 34 71 14 31 38 25 22 39 49 12 48 52 7 2
 1 44 47 49 59 21 63 8 47 36 42 51 7 27 70 1 62 0 71 11 69 47 27 37 54 26 57
64 59 12 61 57 33 28 44 8 69 11 14 26 9 18 68 67 17 9 21 71 4 6 30 40 56 31 1
6 51 33 71 10 58 52 45 32 36 31 50 25 66 29 57 46 64 10 67 58 6 15 49 26 32 1
3 52 25 8 70 4 55 36 48 16 20 21 54 30 19 27 38 16 34 11 14 59 13 38 51 63 69
 3 19 58 64 3 12 58 46 22 23 0 2 45 17 9 43 4 70 1 39 55 35 36 5 63 15 56 53
19 60 62 46 43 35 42 56 24 61 65 21 7 39 47 54 65 5 19
2018-05-22 12:51:15, data/DSJC250.9.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 20, 0.818029s, 205799, 0, (72)64 34 44 71 2 64 0 29 22 69 41 21 53 58 69
4 40 59 27 44 22 40 42 51 61 25 28 36 44 44 43 35 50 60 30 62 55 15 64 34 16
59 23 43 13 69 40 39 29 6 71 10 56 27 1 60 25 20 38 22 56 31 65 46 48 58 48 3
3 47 2 6 45 23 19 34 53 31 70 12 33 17 23 67 35 33 9 11 21 61 62 26 47 53 38
32 25 46 18 3 26 11 66 41 59 3 30 13 36 0 45 65 46 30 7 24 17 63 3 45 14 71 7
0 57 42 5 47 55 57 50 69 35 70 28 17 6 63 16 10 14 48 27 16 66 8 4 19 12 34 2
0 9 8 36 5 8 61 49 38 50 18 68 37 63 21 19 12 57 54 42 32 48 13 55 32 22 59 1
8 7 38 21 39 65 4 67 58 68 68 67 66 71 1 52 45 5 49 26 17 24 9 15 39 36 41 31
 51 52 49 42 51 47 49 54 24 28 7 25 2 39 50 60 4 65 46 62 29 43 67 37 41 7 1
64 52 11 30 54 60 43 54 20 56 10 14 66 11 62 3 71 0 37 52
2018-05-22 12:51:18, data/DSJC250.9.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 30, 2.43117s, 726511, 0, (72)67 16 51 70 5 67 58 22 1 39 13 43 12 19 39 6
4 28 32 12 51 17 28 35 45 65 53 21 62 51 51 54 22 17 63 4 46 66 64 67 16 24 1
4 69 4 38 39 42 49 21 23 70 66 36 1 24 63 53 33 29 68 36 44 6 2 10 47 10 50 5
5 5 23 71 3 32 16 12 61 40 7 50 33 34 31 27 50 18 1 31 65 46 30 55 3 29 25 53
 2 19 15 30 18 60 13 14 15 9 38 62 69 71 6 2 4 11 27 0 20 15 71 21 70 40 57 3
5 56 55 44 48 52 39 19 40 34 33 23 28 8 0 68 10 0 8 60 27 69 68 7 16 48 61 41
 62 56 27 52 37 29 41 43 9 31 12 24 66 7 57 59 35 25 10 38 22 25 30 15 14 11
29 44 49 6 41 47 9 3 41 57 60 70 54 26 71 56 37 31 43 33 18 64 49 62 13 8 45
26 37 35 45 55 37 59 65 34 11 53 5 49 52 63 20 6 2 46 47 20 9 42 13 11 43 67
26 1 4 59 63 17 59 48 36 32 58 60 20 46 61 70 58 42 26
2018-05-22 13:24:59, data/DSJC500.1.col, HEA(POPULATION=10|STRIDE=10000|SCALE
=1), 67, 36.2824s, 16506833, 0, (12)9 4 2 8 5 1 4 9 4 7 6 10 11 10 9 0 6 3 0
0 6 0 0 7 1 0 7 3 2 3 11 9 8 8 2 9 4 6 8 4 8 2 2 11 10 1 4 9 4 9 4 5 5 5 1 7
```

8 4 8 8 11 1 11 5 3 6 10 4 9 2 11 9 11 2 1 5 11 11 2 5 10 6 0 0 9 5 3 7 11 3
9 7 5 6 5 10 6 1 3 10 9 0 3 6 11 11 4 1 3 2 10 8 6 10 6 11 11 7 1 0 2 9 0 6 7
5 2 11 3 7 7 7 2 4 10 4 6 6 0 8 1 3 0 4 2 2 10 7 11 3 5 4 5 3 11 3 11 4 0 11
9 2 4 10 6 7 11 9 9 0 8 1 5 4 8 3 6 6 8 6 4 4 10 8 3 5 10 6 10 8 2 8 4 8 10
8 1 7 11 8 1 10 8 4 10 2 6 8 5 5 9 5 7 8 1 5 3 5 8 6 11 5 3 2 11 6 8 11 10 6
7 2 2 11 3 8 11 9 9 2 11 3 8 0 1 6 7 9 3 10 1 1 9 7 9 3 7 10 2 3 4 8 0 10 9 1
6 0 11 2 3 0 1 10 10 2 3 9 11 0 5 5 5 2 0 3 6 11 9 4 1 4 8 7 9 9 2 6 2 10 6
7 5 7 11 0 8 8 5 0 3 5 2 1 9 10 3 1 1 8 3 0 7 1 7 3 6 11 5 0 2 10 7 4 9 3 9 3
1 5 11 2 3 2 4 3 6 8 7 5 0 7 8 7 11 4 4 1 8 8 7 4 8 4 10 8 2 9 8 8 3 3 8 6 4
5 6 4 8 0 1 0 0 8 8 8 8 5 5 9 3 5 1 7 4 0 9 4 0 5 10 2 4 4 5 1 0 0 11 6 9 6
1 2 10 7 11 0 9 11 10 7 7 6 10 7 0 9 10 10 8 10 0 0 10 11 9 0 1 8 5 6 4 7 4 7
1 8 9 5 5 4 7 6 9 9 7 6 4 0 6 0 6 10 11 7 8 9 6 6 7 0 4 6 9 0 9 2 1 2 10 8 4
8 10 10 7 1 0 4 10 1 5 7 10 4 2 6 4 1
2018-05-22 13:25:29, data/DSJC500.1.col, HEA(POPULATION=10|STRIDE=10000|SCALE
=1), 10, 30.5608s, 14317724, 0, (12)7 3 0 7 2 2 0 1 8 0 5 9 8 8 3 0 10 10 9 7
3 9 6 2 5 3 4 5 8 4 2 5 9 11 1 8 2 10 11 4 10 2 0 10 4 3 10 9 1 6 4 10 4 7 4
9 6 6 11 11 3 10 1 4 3 11 8 8 10 5 1 7 0 10 9 7 2 2 3 2 7 0 0 6 7 8 0 8 10 0
3 1 5 0 4 9 9 5 5 11 8 9 4 5 6 2 4 3 9 0 9 11 11 8 7 3 7 7 5 11 10 3 1 11 3
5 0 1 7 5 4 7 0 3 0 8 11 3 6 6 8 5 11 1 1 11 3 1 4 0 9 3 5 0 8 5 2 8 6 9 5 11
5 4 7 9 10 6 2 9 8 3 5 0 11 0 9 1 7 3 8 10 1 4 2 3 9 0 9 1 0 3 4 9 1 5 2 3 2
8 10 1 11 6 2 10 10 11 11 4 10 10 8 11 2 8 4 5 5 3 10 5 1 6 4 2 5 3 4 4 3 2
8 6 0 1 2 11 11 2 10 0 11 5 10 6 2 6 0 1 3 1 5 3 4 6 3 8 0 4 7 1 8 1 11 7 0 6
9 2 10 2 3 10 6 7 8 10 5 3 9 3 5 0 9 6 4 2 6 1 4 6 9 9 9 7 11 7 4 4 7 8 3 10
2 2 10 11 0 8 10 10 1 1 5 10 3 10 10 5 9 6 4 11 1 2 9 6 5 6 11 3 11 8 9 4 9
7 7 6 2 0 2 4 6 5 4 9 1 3 8 1 2 0 2 0 4 8 11 2 3 11 9 0 1 6 2 4 11 5 4 1 9 2
8 8 7 7 11 6 6 8 0 11 11 11 4 5 3 7 0 5 6 6 4 3 7 9 6 7 6 11 10 0 8 6 0 8 2 1
0 2 7 7 9 10 6 0 2 0 2 7 6 9 3 1 5 6 4 0 4 11 1 2 1 3 1 0 7 8 11 3 11 3 2 11
9 0 11 9 5 10 8 4 1 9 1 7 3 0 8 4 7 6 2 5 5 11 3 7 10 10 9 8 5 11 8 7 11 3 11
2 11 5 11 6 8 6 2 5 6 10 3 5 11 7 3 11 9 6 5
2018-05-22 13:25:44, data/DSJC500.1.col, HEA(POPULATION=10|STRIDE=10000|SCALE
=1), 20, 14.7663s, 6181337, 0, (12)10 5 0 3 11 11 2 7 6 3 11 9 4 7 0 3 11 3 5
1 11 8 11 9 8 2 10 5 6 5 9 0 9 1 9 7 7 3 7 6 3 11 0 7 0 2 7 9 6 5 10 2 8 1 1
0 6 0 3 8 11 5 7 6 8 4 9 6 10 0 8 11 0 2 7 11 1 10 8 8 1 11 2 3 1 2 2 10 2 3
0 0 6 2 0 11 3 6 2 6 1 10 6 5 5 9 6 1 1 4 6 5 1 1 8 8 5 4 8 6 2 0 4 0 1 11 7
3 4 0 6 7 6 0 4 7 7 1 3 10 1 1 3 1 8 8 1 2 6 7 0 0 9 10 0 7 3 9 5 3 6 4 8 8 1
0 1 9 0 9 0 1 2 1 3 10 7 0 1 6 4 9 4 7 11 4 0 9 9 3 11 11 8 8 7 11 3 6 4 3 9
8 7 2 8 4 8 2 9 0 2 7 7 7 2 0 7 6 9 11 3 4 7 8 0 3 10 11 11 9 0 11 9 5 2 4 5
7 11 9 11 3 3 11 4 2 7 4 4 9 0 5 3 9 11 11 10 0 10 6 5 2 2 3 1 7 2 8 2 6 9 11
0 5 9 8 7 10 5 3 0 4 10 8 4 8 11 10 10 4 7 2 2 11 11 0 0 2 1 10 9 6 10 11 3
3 9 10 7 1 6 4 5 3 10 4 5 6 10 1 3 11 2 3 6 1 5 10 11 9 1 3 11 6 11 10 11 5 1
1 0 5 9 5 11 6 10 10 4 1 3 2 3 2 9 1 4 4 0 8 1 10 8 1 4 4 11 11 4 10 1 11 5
7 11 4 5 6 4 9 9 8 3 7 3 8 1 11 11 6 8 7 8 4 7 1 2 0 10 11 10 2 10 8 2 4 2 8
5 6 7 0 9 10 1 4 8 5 0 5 1 8 7 8 6 0 7 6 0 1 3 5 8 6 9 8 10 7 4 2 9 8 5 1 9 4
1 8 4 1 4 5 1 4 6 9 10 2 10 9 4 6 5 0 9 3 8 2 4 4 3 3 7 0 7 4 10 4 10 1 4 6
3 1 4 1 5 10 3 7 5 10 7 10 11 5 2 2 2 8 0 7
2018-05-22 13:26:28, data/DSJC500.1.col, HEA(POPULATION=10|STRIDE=10000|SCALE
=1), 30, 44.567s, 21254974, 0, (12)6 7 11 6 11 0 3 1 8 1 4 6 6 9 2 11 9 8 6 2
6 7 2 9 3 2 4 11 1 3 9 4 0 9 8 4 2 4 5 4 5 2 8 2 11 7 7 1 10 2 3 10 5 3 0 8
5 8 11 8 5 10 3 10 2 6 4 8 3 9 0 1 5 4 11 7 2 5 9 10 4 5 11 8 2 9 0 6 8 6 9 3
0 2 7 3 8 0 3 8 5 11 8 6 2 8 5 6 1 7 6 10 4 2 11 6 1 5 5 6 2 9 2 8 0 10 10 1

1 5 8 1 7 7 10 11 7 10 1 9 7 9 3 3 11 9 2 4 5 4 11 7 8 4 1 3 6 8 9 9 5 4 9 5
4 5 0 9 0 8 10 5 8 7 4 5 3 3 0 7 5 5 9 0 4 3 3 9 11 0 8 6 5 4 0 11 4 11 6 5 5
4 7 5 9 11 0 3 7 1 4 4 3 0 2 2 7 8 4 1 3 0 4 3 1 5 4 6 9 0 10 1 2 9 3 1 7 2
6 6 2 7 3 10 2 3 1 11 9 11 7 11 9 6 1 1 2 2 10 11 5 11 8 1 11 10 3 1 7 8 2 5
11 1 8 10 2 10 0 4 5 10 10 10 5 3 9 10 1 10 7 4 9 8 8 2 9 8 1 7 5 6 5 1 10 11
11 7 8 6 1 11 10 9 10 2 0 7 9 5 10 11 2 0 5 7 3 1 9 10 3 9 0 5 9 10 6 0 0 2
6 6 6 9 0 5 11 11 8 11 9 7 10 8 1 1 7 4 7 7 5 10 8 7 9 9 9 7 3 5 8 9 3 10 11
7 1 11 7 0 6 9 9 1 5 11 0 5 6 8 3 5 1 6 2 4 3 4 7 6 11 4 10 1 11 10 9 6 8 4 1
6 1 6 1 8 5 2 11 8 10 6 6 6 3 7 1 11 0 11 4 5 3 11 11 10 10 1 6 9 0 10 10 2
9 4 0 7 8 0 10 10 7 0 6 8 11 4 11 1 5 1 5 6 3 5 10 8 2 2 1 4 0 7 1 9 11 5 0 9
8 3 8 6 5 7 9 10 3 3 2 3 4 0 7 5 10 10 10 2 10
2018-05-22 13:32:34, data/DSJC500.5.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 67, 99.7223s, 17038888, 0, (48)43 31 35 16 3 18 45 20 41 14 40 39 3 19 34
40 7 27 7 43 3 19 25 13 0 5 23 33 36 42 4 38 14 13 28 26 29 1 15 46 39 34 18
41 7 15 15 34 3 28 29 18 0 42 47 32 43 0 40 20 4 26 43 40 3 36 1 18 28 35 10
44 17 2 24 23 4 35 43 27 9 24 27 30 27 6 8 27 22 42 33 18 16 22 0 5 44 26 41
46 46 6 34 17 26 21 43 21 7 39 13 5 6 0 6 40 36 36 26 32 21 2 14 30 37 19 28
18 10 47 43 11 12 29 47 12 47 23 11 0 44 39 25 1 28 11 20 14 2 41 35 4 16 37
0 13 19 16 41 10 47 40 34 25 18 26 45 31 28 42 28 2 9 26 45 42 38 27 24 26 4
3 8 43 33 46 27 24 9 39 47 19 10 41 13 16 21 40 44 38 39 11 38 31 1 7 21 9 4
6 3 6 41 45 22 5 12 43 32 46 45 27 12 28 13 0 10 35 20 8 33 25 9 25 41 31 33
42 15 40 34 30 25 2 47 8 22 29 32 17 36 38 38 36 5 37 37 7 45 30 33 23 30 3 2
5 4 24 36 37 42 42 10 17 32 29 0 41 42 1 18 29 5 8 31 39 2 37 22 45 33 1 27 2
1 2 31 17 19 33 39 41 44 34 36 17 44 18 31 45 46 17 25 35 11 14 38 16 38 38 2
9 41 38 13 29 7 39 17 31 44 44 29 6 31 45 32 8 44 36 26 17 18 29 28 4 36 23 3
9 10 6 3 37 34 42 37 30 8 44 22 22 33 35 19 47 28 11 3 16 9 39 47 46 35 26 40
4 33 15 12 31 32 5 23 21 37 1 16 1 20 42 7 45 23 22 6 45 27 13 28 22 12 23 3
4 44 43 30 11 35 11 38 5 32 22 46 14 15 19 46 15 14 31 43 37 10 40 44 10 37 4
6 46 12 33 19 47 18 8 34 26 14 4 40 46 17 30 30 10 23 21 9 20 4 2 44 24 32 24
13 40 47 35 38 47 43 41 42 34 15 32 12 23 32 21 37 24 47 30 5 24 45 13 25 35
15 6 9 12 46 45 10 27 22 5 9 2 30 7 7 35 34 26 39 2 9
2018-05-22 13:34:12, data/DSJC500.5.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 10, 98.0295s, 15489405, 0, (48)28 42 43 5 16 10 43 44 46 14 24 45 16 40 8
14 38 36 18 17 13 3 9 6 2 44 27 26 24 36 33 11 47 28 3 21 34 1 8 12 26 34 38
17 15 8 27 23 13 27 34 17 1 23 21 36 37 29 35 39 5 19 19 11 16 30 1 5 12 39
9 44 47 9 36 22 29 27 25 33 38 30 7 41 8 36 43 44 33 11 37 39 4 40 37 9 45 11
21 38 34 29 31 11 1 31 12 45 43 9 20 27 36 15 15 35 13 30 18 36 46 46 7 41 5
19 45 38 19 32 22 18 30 4 12 24 22 0 26 7 31 6 20 34 37 9 11 47 40 43 21 42
9 36 7 6 40 28 43 23 26 24 47 40 38 28 20 27 40 29 1 42 23 10 25 18 14 28 3 4
2 41 10 37 3 0 14 6 45 22 20 43 3 2 20 45 44 35 47 16 14 46 25 35 16 7 23 37
9 30 26 20 11 13 34 33 42 24 6 7 32 25 8 34 13 43 38 44 27 22 5 28 23 16 25
19 31 12 14 21 41 31 34 44 3 35 23 42 35 0 25 24 16 5 33 33 20 30 41 45 1 41
16 22 13 3 1 46 23 24 37 0 2 34 27 46 23 4 10 23 14 18 3 37 2 44 33 12 28 44
3 31 45 19 22 21 20 21 7 30 35 17 4 44 38 35 25 38 42 9 19 47 25 4 11 11 17 1
7 43 40 25 39 7 1 24 25 31 22 42 26 27 34 6 39 32 33 21 47 46 10 8 31 29 21 4
0 27 36 15 13 8 36 0 41 28 30 10 40 38 20 40 12 18 42 21 37 8 42 46 12 29 28
11 47 45 8 20 12 46 26 31 43 46 1 47 1 26 40 32 45 42 33 4 32 32 38 45 14 46
10 30 32 24 16 39 43 39 13 35 26 29 34 47 33 42 0 44 7 5 27 29 6 37 32 31 44
0 28 17 30 40 22 10 39 37 32 47 26 31 34 29 41 41 37 10 15 19 9 13 18 0 25 8

30 14 35 35 24 5 22 39 43 11 47 28 19 6 29 17 12 33 22 25 41 18 18 45 20 13 3
9 43 39 5 39 14 2 5 12 15 24 45 26 41 35 36 0 42 40 32 36 4
2018-05-22 13:36:05, data/DSJC500.5.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 20, 112.949s, 16622319, 0, (48)45 15 9 22 17 36 27 13 40 14 8 26 33 0 11
22 21 17 34 45 8 43 32 14 39 20 30 6 22 4 41 3 28 4 3 34 25 10 8 46 21 44 41
40 33 10 10 4 32 17 6 28 35 18 47 39 45 38 8 5 20 34 45 16 33 5 10 28 23 18 3
4 42 5 13 32 28 6 38 45 5 29 1 17 13 32 16 27 5 2 41 25 31 29 4 7 19 42 43 40
 46 46 2 30 11 38 44 45 20 39 24 41 41 39 30 2 20 35 33 36 17 43 44 31 10 37
14 0 28 14 47 45 41 18 32 47 22 47 24 23 8 42 31 15 17 25 34 41 11 40 40 16 4
4 32 43 7 24 30 1 40 14 47 44 14 10 28 35 23 15 4 41 38 22 18 35 6 30 27 4 20
 38 45 9 45 25 46 32 12 36 26 47 36 43 9 39 19 12 16 42 39 6 1 19 15 43 31 44
 11 12 43 37 6 40 3 10 32 41 45 15 46 0 0 3 10 36 35 35 26 11 37 22 37 33 0 4
0 5 36 44 16 12 22 31 44 18 47 18 27 25 30 5 14 15 36 33 19 16 26 13 34 23 21
 35 6 29 28 29 43 35 30 38 4 37 25 9 44 28 34 3 23 31 32 19 29 18 37 27 31 10
 21 33 38 20 43 4 24 21 22 41 31 40 42 33 25 27 42 7 39 21 46 13 26 26 39 28
2 13 7 3 27 40 44 19 19 7 10 17 15 42 42 27 23 15 39 7 0 42 38 17 36 44 20 1
44 14 30 31 5 6 12 13 33 19 8 31 29 42 21 37 41 13 43 47 36 1 20 26 39 26 47
46 38 26 3 29 25 11 19 23 15 32 30 21 18 41 29 35 21 24 5 2 14 10 23 0 36 31
38 12 13 30 34 42 45 21 37 18 19 20 43 23 12 46 37 10 37 46 11 7 12 45 26 17
8 42 26 7 46 46 9 25 27 47 28 39 4 23 3 29 44 46 30 6 8 38 43 12 2 21 36 34 4
2 1 34 1 9 33 47 43 20 47 45 27 3 37 8 9 9 17 25 12 16 1 47 24 19 29 0 13 6 1
6 18 2 25 35 46 4 28 32 40 27 12 37 15 33 34 16 14 38 16 11 39
2018-05-22 13:37:19, data/DSJC500.5.col, HEA(POPULATION=5|STRIDE=10000|SCALE=
1), 30, 74.6228s, 11353311, 0, (48)9 2 27 43 45 17 34 7 1 43 4 22 31 30 42 25
 14 15 1 38 9 8 35 22 13 7 13 10 0 37 35 36 14 5 21 37 16 12 8 46 22 42 21 40
 0 12 37 42 13 31 0 23 17 18 47 23 31 29 2 41 40 24 43 35 16 33 17 33 13 41 2
 23 12 20 0 4 40 44 31 44 8 36 12 45 45 43 31 28 29 18 3 41 38 30 10 19 37 30
 29 46 46 44 42 26 17 16 31 16 39 24 31 19 14 2 29 36 33 11 37 12 33 1 18 0 3
2 14 6 23 6 47 21 19 11 33 47 43 47 40 5 29 45 39 31 17 3 38 7 23 15 27 6 24
20 32 38 5 35 25 36 28 47 43 42 30 43 11 34 31 30 25 38 10 18 17 34 5 36 44 3
9 7 14 25 3 35 46 31 1 20 35 47 15 12 5 24 19 37 10 9 39 0 3 36 13 21 37 40 2
9 33 5 16 0 27 34 36 21 19 28 43 46 20 1 21 38 33 45 28 22 9 1 9 30 29 2 27 1
6 24 5 26 0 42 28 30 26 47 40 44 18 8 26 24 29 39 9 19 32 32 26 34 33 1 17 38
 38 45 13 10 17 32 33 18 1 15 15 16 29 5 18 22 11 39 19 39 2 22 20 32 25 34 1
1 7 6 16 26 2 14 15 21 38 20 33 42 14 26 34 23 13 34 46 26 20 22 23 4 38 20 8
 39 39 37 30 25 41 14 17 27 25 25 15 13 2 44 34 23 41 45 11 45 26 38 44 31 40
 25 8 22 28 3 4 32 42 4 32 45 35 39 44 30 12 37 30 47 19 4 36 12 24 22 47 46
17 39 36 40 39 21 21 28 15 23 1 16 32 12 37 44 26 10 38 34 8 15 45 20 20 22 0
 19 25 40 42 43 43 28 41 6 41 36 19 9 25 46 10 6 7 46 35 10 3 14 32 27 24 4 4
 32 46 46 11 30 24 47 37 41 42 13 18 37 9 46 26 27 9 11 33 16 10 35 24 21 27
31 3 33 45 18 47 6 36 47 29 35 2 42 21 19 23 12 40 8 32 43 47 35 45 3 34 3 0
41 27 41 3 41 46 34 23 44 27 22 3 2 24 35 6 6 42 30 20 27 4