[文章编号] 1003－4684(2024)02-0017-06

求解最小支配集问题的禁忌遗传混合算法

吴歆韵，彭　瑞，熊才权

(湖北工业大学计算机学院，湖北 武汉 430068)

[摘　要] 将最小支配集问题转换为一系列判定问题——k 支配集问题，并提出一种禁忌遗传混合算法对 k-DS 问题进行求解。此算法将禁忌搜索算法和遗传算法两种启发式算法结合起来，互补不足。高效的邻域结构保证了算法的运行效率，禁忌策略防止算法过早陷入局部最优陷阱，遗传算法框架进一步增强了算法的疏散性。经过与现有求解最小支配集算法的结果进行分析比较，禁忌遗传混合算法的结果较其它算法更优。

[关键词] 最小支配集；NP 难问题；禁忌遗传混合算法；k 支配集

[中图分类号] TP393　　[文献标识码] A

作为经典组合优化问题之一，最小支配集问题已被证实为 NP 难问题，精确算法无法在规定时间内实现大规模顶点集的求解。较为有效的方案是采用启发式算法求解该类问题。常见的启发式算法[1-2]多基于遗传算法[3]，禁忌搜索算法[4]，模拟退火算法[5]，蚁群优化算法[6]等。文献[7]提出的 ScBppw 算法是一种基于分数检测的禁忌策略与概率随机游走相结合的局部搜索算法，在求解最小支配集时使搜索方式更具多样化。Chalupa[8]提出一种基于排序的随机局部搜索的 RLSo 算法，它可在大规模图中求解最小支配集，且求得的结果优于贪心近似算法[9]、基于局部搜索的蚁群优化算法(ACO-LS)[10]以及其带预处理的扩展算法(ACO-PP-LS)[11]。Cai[12]等提出的一种基于局部搜索框架的 FastDS 算法。该算法主要特点是在已知一个可行解集合元素个数为 k 后，继续判断是否存在解集合元素个数为 k-1 或者 k-2 的解。这种算法加快了收敛速度，扩大了搜索区域。最近 Alipour 等提出一种适用于网络不断变化的动态模型的分布式算法来求解大规模图的最小支配集。

传统的启发式算法多基于局部搜索算法或遗传算法。作为一种典型的局部搜索算法，禁忌搜索算法优势在于可以接受劣解，并采取禁忌策略使解集合不会局限于某一搜索空间无法跳出，从而获得全局最优。但是禁忌搜索算法的迭代搜索过程是一种串行结构，故该算法一定程度受限于初始解。两个不同的初始解的求解结果可能会相差甚远。而遗传算法作为一种并行结构，具有群体搜索的特点，不依赖于初始解。但是，其局部搜索能力差，易陷入局部最优，种群的多样性可能会消失，出现“早熟”现象。本文提出了一种禁忌遗传混合(TSG)算法，它在禁忌搜索算法的基础上嵌入遗传算法，很好地融合了双方的优点，并弥补相互的不足。最后通过对比实验，证明 TSG 算法的优越性。

1　算法主体框架

本节介绍算法的整体框架，其主要思想为将最小支配集问题转化为一系列 k-DS 问题进行求解。k-DS 问题定义为：给定一个无向图 $G=(V,E)$，寻找图 G 中顶点数为 k 的支配集。算法主框架如算法 1 描述，给定格局 X，求解此格局下的 k-DS 问题，若当前 k-DS 问题能够求解，则将 k 值减小 1 并重复 k-DS 问题求解直至无法找到合法解，最后一个记录的合法支配集为算法所给出的最小支配集。本文格局表示为顶点数为 k 的待求解支配集与其它集合的顶点分布情况。图 GG 的顶点集合 VV 显然是一个支配集，于是 k 初始化为图 G 的总顶点数。随着不断迭代求解 k-DS 问题，k 值不断减小，其求解难度也逐渐增大。迭代终止条件是达到程序设定的最大迭代次数。本文主要描述如何求解 k-DS 问题。

算法 1　算法整体框架

输入：$G=(V,E)G=(V,E)$

输出：k

[收稿日期] 2022-08-21

[基金项目] 国家自然科学基金(6192116)

[第一作者] 吴歆韵(1987－)，男，湖北宜昌人，湖北工业大学副教授，研究方向为优化算法。

```
procedure 最小支配集问题
  k ← |V|
  iter ← 0
  while iter < Max_iter do
求解格局 X 下的 k -DS 问题
  if 找到合法的 k -DS then
    k ← k − 1
  end if
  iter ← iter + iterTS
  end while
  return k
  end procedure
```

2 k -DS 问题的求解

本节介绍用于求解 k -DS 问题的 TSG 算法，TSG 算法是禁忌搜索与遗传算法相结合的混合算法，其结构见图 1。

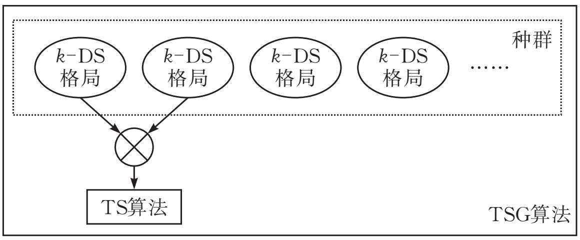

图 1 TSG 算法结构

TSG 算法将禁忌(TS)算法嵌入遗传算法，其中遗传算法负责维护种群，种群中个体的杂交后代由 TS 算法进行优化后再融入种群。种群的个体为 k -DS 问题的一个解格局。注意，TS 算法也是一个完整的 k -DS 问题求解算法。将 TS 算法与遗传相结合能够提升算法的稳定性。

后续小节首先介绍 TS 算法，包括 TS 算法的搜索空间、评估函数、临域结构以及算法逻辑过程；随后介绍嵌入 TS 算法的遗传算法即 TSG 算法，包括种群构成、个体杂交方式以及种群更新方法。

2.1 TS 算法

禁忌搜索(TS)算法由 Glover 等提出，是一种基于局部搜索框架的启发式算法，通过禁忌策略，即禁止程序搜索之前搜索的区域，帮助其跳出当前局部最优，转向新的区域进行搜索解集合以期获得更好的局部最优解。本节将会对 TS 算法中的搜索空间，目标评估函数和邻域结构作出定义，接着详细描述禁忌搜索的过程以及伪代码。

2.1.1 搜索空间 给定一个无向图 $G=(V,E)$，根据支配集的定义，将图 G 的顶点进行如下分类：

S 集合：待求解支配集。

S^+ 集合：该集合表示被集合 S 支配的集合，其每个顶点都与 SS 集合中某一顶点相邻，$S^+=\{v:v \in V\backslash S, u \in S, (u,v) \in E\}$ 。

S^- 集合：该集合表示不与 S 集合中任一顶点相邻的顶点集合，$S^-=\{V\backslash(S \cup S^+)\}$ 。

$S^\pm$ 集合：该集合表示 $S^\pm$ 集合与 S^- 集合的并集，$S^\pm=\{S^+ \cup S^-\}$ 。

本文算法对 k -DS 问题的搜索空间为 $F=\{S \mid S \subset V, \mid S \mid = k\}$，通过对候选解集合 $S \in F$ 不断修改，使得 S^- 集合为空，从而得到一个顶点数为 k 的合法支配集。

2.1.2 目标评估函数 找到给定无向图 $G=(V,E)$ 的支配集 S，需要保证所有顶点都属于 S 集合或 S^+ 集合，即当 S^- 集合为空集时 S 集合为图 G 的支配集。定义目标评估函数 $f(S)=\mid S^- \mid$ $f(S)=\mid S^- \mid$ 作为求解 k -DS 问题的评判标准，该函数表示待求解集合为 S 集合时 S^- 集合的元素个数。当 $f(S)=0$ 时表示 S 集合是一个合法支配集。

2.1.3 顶点交换邻域结构 本文算法的邻域结构是一种在特定规则下的顶点间交换策略。规则定义为：若 $\exists u \in S, \exists v \in S^\pm$，记 $sw<u,v>sw\langle u,v \rangle$ 为一次邻域动作，表示将顶点 u 从 S 集合移除，并将其移入 $S^\pm$ 集合；将顶点 v 从 $S^\pm$ 集合中移除，并移入到 S 集合中。

图 2 展示了一个顶点交换邻域动作示例。在图 2a 中，$S=\{A,G\}$(深灰色顶点)，$S^+=\{B,C,D,F\}$(浅灰色顶点)，$S^-=\{E\}$(白色顶点)，$S^\pm=\{B,C,D,E,F\}$。当前 $f(S) \neq 0$，通过邻域动作 $SW<G,F>$，可得到图 2b 所示格局，$S=\{A,F\}$，$S^+=S^\pm=\{B,C,D,E,G\}$，$S^-=\varnothing$，此时 $f(S)=0$，表明当前 S 集合是一个 $k=2$ 的合法支配集。

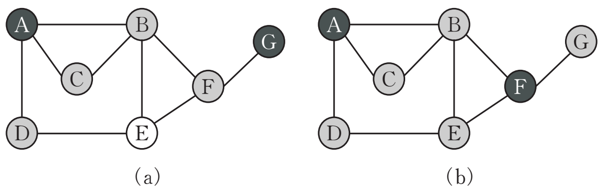

图 2 邻域动作 $SW<G,F>$

2.1.4 禁忌搜索过程 禁忌搜索过程是整个算法的核心。本小节将详细描述搜索过程。在算法 2 中，将对输入的待求解格局 X 进行禁忌搜索，并初始化 f_{best}。f_{best} 记录算法搜索到的历史最优目标函数值，其初始值为输入格局的目标函数数值。

算法通过一系列邻域动作对格局进行调整，直到 S^- 集合为空集。算法在每次迭代中对 S 集合和 $S^\pm$ 集合进行搜索，期望找到最优邻域动作作为邻域交换来找到合法支配集。对于一次迭代，首先对 S 集合和 $S^\pm$ 集合依次遍历顶点的方式进行搜索，在该

过程中要忽略 S 集合中被禁忌的顶点，防止搜索时陷入局部最优；其次，对两个集合中遍历到的每一对顶点进行函数评估操作，计算该邻域动作后的评估函数值 f 并与当前 f_{best} 比较，如果小于 f_{best} 则使用当前值覆盖 f_{best} 并更新最优邻域动作 $Move_{best}$。

当一轮搜索结束后，如果存在最优邻域动作 $Move_{best}$ 则按该邻域动作进行顶点交换（S 集合中的顶点 a 和 $S^{\pm}$ 集合中顶点 b），并更新邻域结构以及禁忌表（算法 2 第 15 和 16 行）。注意，移入 S 集合的顶点 b 要进行禁忌操作，算法设置一个在区间 $[t_{Min}, t_{Max}]$ 上的随机值来赋值给禁忌长度 t。在程序搜索过程中顶点 b 在被禁忌时期时无法移出 S 集合，避免程序陷入局部最优。

若更新之后的 f_{best} 不为 0，那么继续对 S 集合和 $S^{\pm}$ 集合进行新一轮的搜索迭代。反之，返回所得的合法解格局。如果长时间无法找到合法的支配集，则返回函数评估值为 f_{best} 的解格局。

算法 2　禁忌搜索过程

输入：待求解格局 X

输出：X_{best}

1)　procedure TS_ k -DS(X)
2)　$f_{best} \leftarrow f(S)$
3)　while $f_{best} \neq 0$ do
4)　　for $\forall a \in S$ and $\forall b \in S^{\pm}$ do
5)　　　if 顶点 a 被禁忌 then
6)　　　　continue
7)　　　else
8)　　　　$f \leftarrow$ Evaluate_Move $\langle a, b \rangle$
9)　　　　if $f < f_{best}$
10)　　　　　$f_{best} \leftarrow f$
11)　　　　　$Move_{best} \leftarrow \langle a, b \rangle$
12)　　　　end if
13)　　　end if
14)　　end for
15)　　$S_{best} \leftarrow$ Move_Operate($Move_{best}$)
16)　　Move_Tabu(b,t)
17)　　if S_{best} 长时间未变化 then
18)　　　break
19)　end while
20)　return X_{best}
21)　end procedure

2.2　TSG 算法求解 k -DS 问题

TS 算法是一种串行结构，虽然该算法在求解 NP 难问题上能够接受劣解，跳出局部最优陷阱，但是它一定程度上依赖初始解，影响了最终求解结果。然而遗传算法作为一种并行结构的算法，不太依赖于初始解，可以弥补 TS 算法的缺点。本文提出一种禁忌遗传混合（TSG）算法，将 TS 算法嵌入到遗传算法中去。本小节详细阐述了求解 k -DS 问题的 TSG 算法，算法 3 给出了该求解过程的伪代码。

算法 3 的输入是由 N 个 k -DS 问题随机格局所组成的集合构成的一个初始种群。算法首先对种群中个体依次进行禁忌搜索操作，如果找到合法的支配集个体，直接返回该个体，否则记录种群中最差个体，即 $f(S)$ 值最大的个体。随后在种群中随机选择一对双亲个体，将其进行杂交并产生一个子代个体（第 2.3 节），对子代个体进行禁忌搜索操作（算法 2）。如果该子代个体评估函数 $f(S)=0$，那么证明已经找到一个合法的支配集，结束当前算法过程并返回该支配集个体。若 $f(S) \neq 0$，则将该子代个体与种群中最次个体比较。若优于最次个体，则将该个体替换掉最次个体，并进行下一轮种群繁衍。通过种群内个体不断迭代优化来寻找合法的最优支配集个体，直至达到算法 1 所述的终止条件。

算法 3　TSG 算法求解 k -DS 问题

输入：初始种群

输出：支配集个体 S_{best}

procedure k -DS 问题
for parent = 1…N do
　TS_ k -DS(parent)
　if f(parent) = 0 then
　　return parent
end if
　Worst $\leftarrow$ worstInPopulation()
end for
while 未在种群中找到 k -DS do
　$S_1 \leftarrow$ RAND(N)
　$S_2 \leftarrow$ RAND(N)
　$S \leftarrow$ PopulationGenetic(S_1, S_2, k)
　$S_{best} \leftarrow$ TS_ k -DS(S)
　if $f(S_{best}) = 0$ then
　　return S_{best}
　end if
　if $f(S_{best}) < f$(Worst) then
　更新种群
　end if
end while
return S_{best}
end procedure

由算法 3 可知，TS 算法嵌入到遗传算法后的 TSG 算法，求解 k -DS 问题的初始解就不再是单个集合个体，而变成了集合种群。这样增加了算法的

随机性、多样性，以便获得更高质量的集合个体。而 TS 算法相当于模拟个体的“变异”过程。种群个体间的杂交所产生的子代并通过“变异”的进化迭代方式而获得的优秀个体，比不进行禁忌搜索的集合个体更易获得合法解。

2.3 种群杂交方式

本小节介绍种群间个体的杂交方式以及维持种群个体数量恒定的精英保留策略。

为维持种群个体数量恒定，可采用精英保留策略，即采用优胜劣汰的种群更新方式。双亲杂交产生子代个体，替换掉劣质个体来优化种群。具体实现如算法 4 所示，该算法目标是从给定的双亲个体 S_1, S_2 以及参数 k 中通过杂交算法产生子代个体并输出。k 表示要生成的子代集合元素的个数。

如算法 4 所描述，子代个体先获得双亲共有的顶点，剩下的顶点从双亲顶点随机获取，直到该个体顶点数达到 k ，至此子代个体生成。将生成的子代返回进行后续的种群更新。对于遗传算法来说，禁忌搜索过程相当于变异操作，且“变异”后的个体大概率优于之前个体。变异作为遗传算法中维持种群多样性的重要手段，为求解最小支配集提供了一种新思路，保证了种群的优良性以及数量的恒定。

假设双亲的顶点元素分别为 $S_1=\{A,B,D,F,G,H\}$ 和 $S_2=\{A,C,F,H,J,L\}$，子代个体 S 首先继承双亲顶点的交集，$S=S_1 \cap S_2=\{A,F,H\}$。此时该个体顶点数量并未达到 k，所以还要在双亲集合中随机继承 3 个顶点，例如 $S=\{A,B,F,H,J,L\}$ 。此时杂交过程结束，返回产生的子代个体 S 。

算法 4 种群杂交方式

输入：双亲 S_1, S_2 以及子代集合个数 k

输出：子代个体 S

```
procedure PopulationGenetic( S1,S2,k )
S←S1∩S2
U←S1∪S2
while 子代个体顶点数量未达到 k do
   S_Add(RAND(U))
end while
return S
end procedure
```

3 求解结果与对比试验

本节对 TSG 算法求解最小支配集问题的效果进行了实验验证。本实验中，采用 2 个公共算例集与文献中提到过的算法的结果进行对比分析，测试 TSG 算法的效果。

TSG 算法使用 Java 语言编写，采用 JDK8，实验平台为 Intel Core i5-7300HQ 2.5 GHz，8 GB 内存的 PC 机，操作系统使用 Windows 10。

3.1 测试算例

对比实验中所用到的 2 个公开数据集为：

Social Network：该算例集来自社交网络样本 Google＋和 Pokec 中的公开数据集，它们都是顶点数在 10000 以内的稀疏无向图。

DIMACS：该算例集基于 DIMACS 标准格式，常应用于图染色问题，它们是一些顶点数在 20000 以内的稠密无向图。

3.2 参数校准实验

对 TSG 算法的两个重要参数取值进行确定：禁忌长度 t 和种群内个体恒定的数量 N。通过对比不同的禁忌长度和种群个体数的设置组合的算法结果差异，来寻找最优参数设置。本实验基于实际情况，对禁忌长度设置了三组范围值，每组$[t_{\mathrm{Min}}, t_{\mathrm{Max}}]$分别为[0,10]，[10,20]，[20,50]，在每组禁忌长度下将种群个体数量 N 设为 1、3、5 和 10 进行测试。另外，对于程序迭代终止条件（算法 1 中的第 4 行）根据数据集的不同而进行设定，本算法的最大迭代次数在 2000 到 5000 之间，由于稠密图求解较为容易，最大迭代次数可以设置稍小，稀疏图最大迭代次数要设置得更大，以便算法有足够时间找到最优解。下面的实验结果也可以对此进行证明。

在参数校准实验中，选取了 3.1 节提到的 2 个算例集中的 5 个算例，每个算例集的顶点数都在 200 到 500 之间，包括稀疏图和稠密图。表 1 中第一列和第二列表示可选的参数，每组参数运行 20 次，每次运行时间为 30 s。第三列 rate 表示这 5 个算例在每组参数设定下运行 20 次获得最优解次数占运行总次数的比值。

表 1 参数校准实验

参数		结果
禁忌长度 t	种群个体数 N	rate
[0,10]	1	0.89
[0,10]	3	0.93
[0,10]	5	0.94
[0,10]	10	0.90
[10,20]	1	0.94
[10,20]	3	1.00
[10,20]	5	0.97
[10,20]	10	0.91
[20,50]	1	0.92
[20,50]	3	0.97
[20,50]	5	0.91
[20,50]	10	0.87

从表 1 中可以观察到，不同的参数设置之间所

得到的结果都有一定的差距,但所有组合均能算到相同的最好结果,说明了算法的稳定性。同时可以看到,当禁忌长度 t 设置成[10,20],种群个体数设置成 3 的效果最好。

3.3　算法对比实验

根据 3.2 节对两个参数禁忌长度 t 和种群个体 N 的校准实验,本节使用 3.1 节提出的 2 个算例集并在这两个参数设定下与其它求解算法进行对比实验并分析结果。

3.3.1　Social Network 算例集　对 Social Network 中的 6 个算例进行了对比实验,把 TSG 算法与贪心近似算法[9]、基于局部搜索的蚁群优化算法(ACO-LS)[10] 以及其带预处理的扩展算法(ACO-PP-LS)[11]的所求结果进行对比。

表 2 列出了 TSG 算法与 GREEDY 算法、ACO-LS 算法、ACO-PP-LS 算法求解最小支配集后得到的结果。本次实验的算例集为顶点数 200 到 10000 左右的稀疏无向图。记录每个算例在 20 次运算中获得的最好值和平均值,若两者相等只记录最优值。从表 2 中可以分析出 TSG 算法求得的最小支配集均优于另外三种算法,且随着算例顶点数增大,其差距越来越大。当顶点数到达 10000 个时,TSG 算法求出的最小支配集比 ACO-LS 算法的求解结果小 100 多个,可见 TSG 算法求解较大规模的无向图具有较大优势。

表 2　Social Network 算例集对比实验

Instance	TSG	GREEDY	ACO-LS	ACO-PP-LS
gplus200	19	19	19	19
gplus500	42	42(42.08)	42	42
pokec500	16	16	16	16
gplus2000	170	175(177.38)	170	170
pokec2000	75	75	75	75
pokec10000	413	413(413.1)	590(623.3)	591(620.3)

3.3.2　DIMACS 算例集　使用算例集 DIMACS 的算例将 TSG 算法与 TSG 算法的三个简化版本:TS 算法,局部搜索(LS)算法以及禁忌搜索扰动(TS-P)算法进行对比实验。

TS 算法:仅使用上文所介绍的禁忌搜索的策略求解最小支配集问题。

LS 算法:可认为是 TS 算法的一个框架算法,未使用其核心的禁忌搜索策略来求解最小支配集问题。

TS-P 算法:TS 算法的基础上增加了一个扰动操作而实现的算法,当邻域结构长时间未发生变化时,通过交换集合间部分顶点的方式重构邻域结构求解最小支配集问题。

表 3 记录四种算法在 20 次独立运算中获得的结果的最优值。

表 3　DIMACS 算例集对比实验

Instance	TSG	TS	TS-P	LS
fpsol2.i.1	229	229	229	229
fpsol2.i.2	90	90	90	90
fpsol2.i.3	64	64	64	64
inithx.i.1	347	347	347	349
inithx.i.2	89	89	89	89
inithx.i.3	64	64	64	64
le450_5a	30	32	32	58
le450_5b	29	33	29	52
le450_5c	20	22	20	34
le450_5d	20	23	21	35
le450_15a	25	26	27	61
le450_15b	26	27	27	82
le450_15c	11	12	11	20
le450_15d	11	12	12	21
le450_25a	37	36	39	136
le450_25b	33	32	32	196
le450_25c	13	14	13	23
miles250	25	25	25	36
miles500	9	9	9	15
miles750	6	6	6	7
miles1000	4	4	4	4
miles1500	2	2	2	2
zeroin.i.1	87	87	87	87
zeroin.i.2	56	56	56	56
zeroin.i.3	51	51	51	51
school1	15	15	16	75
games120	13	13	13	17
homer	96	96	96	370

从表 3 观察分析知:在 20 次独立运算中,TSG 算法均优于另外三种算法。其中,TS 算法优于 LS 算法,说明局部搜索过程相对禁忌搜索过程存在一定劣势,LS 算法更容易陷入局部最优陷阱且 TS 算法能够通过禁忌策略跳出部分局部最优陷阱,从而获得更好的解。TSG 算法是将 TS 算法嵌入到遗传算法进行并行计算并对种群进行优化,可得到更好的解,可见 TSG 算法效果更好。TS-P 算法在大部分算例上比 TS 算法和 LS 算法结果好,说明加入扰动机制后也会对结果进行一定优化,但是依旧不如 TSG 算法。从对比实验上看,TSG 算法的各项组成模块均对算法求解效率有着重要意义。

4　总结

本文提出一种禁忌遗传混合(TSG)算法来求解最小支配集问题。它是一种基于局部搜索框架的禁忌搜索技术嵌入到遗传算法后组成的混合算法。TSG 算法通过禁忌策略解决了搜索过程会陷入局部最优的问题;因为遗传算法的嵌入,克服了禁忌算法对初始解依赖的问题。最后,通过 2 个算例集的

对比实验,验证了TSG算法求解最小支配集问题的优越性。

[参 考 文 献]

[1] YANG X S. Nature-inspired metaheuristic algorithms [M]. Bristol: Luniver press, 2010.

[2] 胡书丽. 启发式搜索算法求解组合优化问题的研究[D]. 长春:东北师范大学, 2019.

[3] HEDAR A R, ISMAIL R. Hybrid genetic algorithm for minimum dominating set problem[C]. // International conference on computational science and its applications. Springer, Berlin, Heidelberg, 2010: 457-467.

[4] WU X, LÜ Z, GALINIER P. Restricted swap-based neighborhood search for the minimum connected dominating set problem[J]. Networks, 2017, 69(02): 222-236.

[5] HEDAR A R, ISMAIL R. Simulated annealing with stochastic local search for minimum dominating set problem[J]. International Journal of Machine Learning and Cybernetics, 2012, 3(02): 97-109.

[6] ROMANIA Q S. Ant colony optimization applied to minimum weight dominating set problem[C]. // Proceedings of the 12th WSEAS international conference on automatic control, modelling and simulation. Catania, Italy. 2010: 29-31.

[7] FAN Y, LAI Y, LI C, et al. Efficient local search for minimum dominating sets in large graphs[C]. // International Conference on Database Systems for Advanced Applications. Springer, Cham, 2019: 211-228.

[8] CHALUPA D. An order-based algorithm for minimum dominating set with application in graph mining[J]. Information Sciences, 2018, 426: 101-116.

[9] CHVATAL V. A greedy heuristic for the set-covering problem[J]. Mathematics of operations research, 1979, 4(03): 233-235.

[10] POTLURI A, SINGH A. Two hybrid meta-heuristic approaches for minimum dominating set problem[C]. // International Conference on Swarm, Evolutionary, and Memetic Computing. Springer, Berlin, Heidelberg, 2011: 97-104.

[11] POTLURI A, SINGH A. Hybrid metaheuristic algorithms for minimum weight dominating set[J]. Applied Soft Computing, 2013, 13(01): 76-88.

[12] CAI S, HOU W, WANG Y, et al. Two-goal local search and inference rules for minimum dominating set [C]. // Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. 2021: 1467-1473.

Hybrid Tabu Search and Genetic Algorithm for Solving Minimum Dominating Set Problem

WU Xinyun, PENG Rui, XIONG Caiquan

(*School of Computer Science, Hubei Univ. of Tech., Wuhan* 430068, *China*)

Abstract: The minimum dominating set problem can be transformed into a series of decision problems-the k dominating set (k DS) problem and a hybrid tabu search and genetic (TSG) algorithm for solving the k DS problem is proposed. The algorithm combines a tabu search algorithm and a genetic algorithm. The efficient neighborhood structure is used to ensure the efficiency of the algorithm, the tabu strategy is used to prevent the algorithm from falling into the local optimal trap, and the genetic algorithm framework is used to further enhance the evacuation of the algorithm. Compared with the existing algorithms for solving the minimum dominating set problem, the result of proposed algorithm outperforms the others.

Keywords: minimum dominating set; NP hard problem; hybrid tabu search and genetic algorithm; k-dominating set

[责任编校:张岩芳]