完全支配集的规约算法

骆伟忠[1] 蔡昭权[1] 兰远东[1] 刘运龙[2]

(惠州学院信息科学技术学院 惠州 516007)[1] (湖南师范大学数学与计算机科学学院 长沙 410081)[2]

摘 要 完全支配集是一个著名的 NP 难解问题,在无线传感器网络中具有重要应用。主要研究了能降低问题规模的规约化算法设计。通过对问题结构进行深入分析并对图中顶点进行着色,得到图中顶点之间的新的组合特性,在此基础上提出一系列高效的多项式时间的局部规约规则。证明了规约规则的正确性,并通过仿真实验验证了规约规则的有效性。

关键词 完全支配集,NP-难解,规约,黑白着色

中图法分类号 TP301 **文献标识码** A

Reduction Algorithm for Total Dominating Set

LUO Wei-zhong[1] CAI Zhao-quan[1] LAN Yuan-dong[1] LIU Yun-long[2]

(School of Information Science and Technology, Huizhou University, Huizhou 516007, China)[1]

(College of Mathematics and Computer Science, Hunan Normal University, Changsha 410081, China)[2]

Abstract Total dominating set is a famous NP-hard problem, and has important applications in wireless sensor networks. In this paper, we mainly studied the design of reduction algorithm which can reduce the size of the original problem. By analyzing the problem structure and coloring the vertices in given graph, we observed many new combinatorial properties of its vertices, and presented a set of efficient and polynomial-time reduction rules exploring local structures of the graph. The rules are proved to be correct theoretically, and the effectiveness is verified by simulations.

Keywords Total dominating set, NP-hard, Reduction, Black and white coloring

1 引言

支配类问题是经典的 NP 组合优化问题,在通信网络、投票系统、生物信息学、社会学等众多领域均有应用[1-3]。完全支配集是支配问题的一种重要形式,其严格定义如下:

输入:无向图 $G=(V,E)$,非负整数 k。

任务:判断是否存在一个大小不超过 k 的点集 $T\subseteq V$,使得 V 中的每个点 v 至少与 T 中的某个顶点相邻。

完全支配集在无线网络中具有重要应用。无线传感网络由大量的传感器节点组成,其主要任务是完成目标覆盖和监控工作。但传感器节点受限于能量供应,容易受到外部攻击,这将极大影响覆盖和监控工作的完成。因此,传感器本身也需要一定程度的监控和保护,而实现这种保护的最好方式是传感器节点相互监控,构建自我保护网络。自我保护网络的构建可以抽象为图论中的完全支配集问题[4-5]。

完全支配集是 NP-难的[6],人们主要研究其组合优化和近似算法求解。文献[6]证明了完全支配集的近似率上界为 $\log n$,其中 n 为图中顶点的个数;文献[7]提出了一个近似度为 $\ln n$ 的多项式时间近似算法,并证明了其近似率下界为 $(1-\varepsilon)\ln n$;文献[8]指出完全支配集存在时间为 $O^*(6^l)$ 的精确算法,其中 l 为输入图的树宽;文献[9]证明了在顶点度至少为 3 的图上最优完全支配集大小的上界为 $n/2$;文献[10]证明了最优完全支配集大小的下界为 n/Δ,其中 Δ 为图中顶点的最大度;文献[11]分析了完全支配集问题的难解性根源,并针对平面图的上完全支配集提出了时间为 $O(4^{19.1\sqrt{k}}k^3n+n^3)$ 的精确算法,其中 k 为问题解的大小。

由上文可知问题解空间的求解与问题的规模成正比,即问题的规模决定了问题的求解效率。近年来,人们通过研究发现,问题输入实例的规模可能很大,但通过某种操作可以极大地形成小问题的规模,且能保证问题解空间的正确性,这种操作称为规约化操作[12]。规约化的基本思想是在求解问题之前,通过执行一个多项式时间的算法来对问题实例中"易处理部分"进行处理,以减少问题规模,从而留下一个相对小的"难处理部分"。

本文针对完全支配集,深入分析单个顶点及其邻接点的局部结构和一对顶点及其邻接点的局部结构,并对顶点进行黑白着色,从而得到图中顶点间的新组合特性。在此基础上,提出了多条高效的规约规则和多项式时间的规约化算法。这是针对完全支配集问题的首个规约算法,提出的规约操作可直接对完全支配集的现有近似算法或启发式算法进行优化,以在执行近似或启发式算法前进行规约,降低问题规模并提高解的精度。无线网络上的模拟测试表明,规约算法在各种

本文受国家自然科学基金项目(61370185),广东省自然科学基金博士启动项目(2015A030310445),惠州学院博士启动项目(C513.0211)资助。
骆伟忠(1978-),男,博士,讲师,主要研究方向为参数计算、网络优化,E-mail:weizhong_1013@163.com;**蔡昭权**(1970-),男,教授,硕士生导师,主要研究方向为智能算法;**兰远东**(1975-),男,博士,副教授,主要研究方向为组合优化、模式识别;**刘运龙**(1971-),男,博士后,副教授,主要研究方向为参数计算。

不同规模和不同稠密度网络上均能不同程度地降低问题规模。

2 相关术语

设 $G(V,E)$ 为一个无向、简单的连通图，其中 V 是 G 的顶点集，E 是 G 的边集，且 $|V|>1$。对于 $u,v\in V$，若 u 与 v 相邻，则采用 (u,v) 表示 u,v 之间的边。设 $\gamma(G)$ 为图 G 上的最小完全支配集的大小。设 $u,v\in V$，若 $(u,v)\in E$，则表示 v 能够支配 u。设 $V_1,V_2\subseteq V$，若对任一 $u\in V_2$，存在 $v\in V_1$ 使得 v 能支配 u，则称 V_1 能支配 V_2。

对于图 G 上的一个顶点 v，设 $N(v)$ 为 v 的所有邻接点的集合，即 $N(v)=\{u|(v,u)\in E\}$；$N[v]$ 表示集合 $N(v)\cup\{v\}$。将 $N(v)$ 划分为以下3个互不相交的子集：

$N_1(v)=\{u|u\in N(v),N(u)\backslash N[v]\neq\emptyset\}$

$N_2(v)=\{u|u\in N(v)\backslash N_1(v),N(u)\cap N_1(v)\neq\emptyset\}$

$N_3(v)=N(v)\backslash(N_1(v)\cup N_2(v))$

对于图 G 上的两个顶点 v 和 w，令 $N[v,w]=N[v]\cup N[w]$，$N(v,w)=N[v,w]\backslash\{v,w\}$。类似地，将 $N(v,w)$ 划分为3个互不相交的子集：

$N_1(v,w)=\{u|u\in N(v,w),N(u)\backslash N[v,w]\neq\emptyset\}$

$N_2(v,w)=\{u|u\in N(v,w)\backslash N_1(v,w),N(u)\cap N_1(v,w)\neq\emptyset\}$

$N_3(v,w)=N(v,w)\backslash(N_1(v,w)\cup N_2(v,w))$

3 规约化规则

给定问题实例 $G(V,E)$，G 上的所有点最初均着为黑色。规约规则在保证最优完全支配集大小不变的前提下删除 G 中的冗余顶点和边，或将黑点着色为白色来减小问题规模。对于 G 中的顶点 u，若 G 中存在最小完全支配集 D，D 不包含 u 点，但包含 $N(u)$ 中的某个点，则将点 u 从 G 中删除。对于 G 中的黑点 w，若 G 中存在最小完全支配集 D，D 不包含 w 点，则将 w 着色为白色。

为了描述规则，引入以下概念：对于黑白着色图 $G(V,E)$ 上的完全支配集 D，若 D 中顶点均为黑点，则 D 为 G 的黑完全支配集；用 $r_b(G)$ 表示图 G 的最小黑完全支配集的大小。

规则1、规则2分析 G 中单个顶点及其邻接点结构，进而进行顶点删除或着色。

规则1 对于 G 上的黑点 v，若 $N_3(v)\neq\emptyset$ 且 $N_1(v)$ 中存在黑点，则从 G 中删除 $N_2(v)\cup N_3(v)$ 上的所有点，并增加一个白点 v' 和一条边 (v,v') 到 G 中。

使用 v 支配 $N_3(v)$ 优于其他顶点，因此将 v 放入解中，白点 v' 保证 v 在解中。$N_2(v)\cup N_3(v)$ 中顶点的支配能力弱于 $\{v\}\cup N_1(v)$ 中的顶点，且 v 能支配 $N_2(v)\cup N_3(v)$，因此可以安全删除 $N_2(v)\cup N_3(v)$ 中的顶点。

引理1 设 G 为黑白着色图，G' 为对 G 执行一次规则1后的结果图，则 $\gamma_b(G)=\gamma_b(G')$。

证明：首先需证明 $\gamma_b(G)\geqslant\gamma_b(G')$。设 D 为 G 的黑完全支配集。因 $N_3(v)$ 需被支配，若 D 不包含 v，则 D 必包含 $N_2(v)\cup N_3(v)$ 中的某个点 u，使用点 v 代替该点 u 可得到另一个包含 v 的、大小与 D 相同的完全支配集。设 D 包含 v。若 D 包含 $N_2(v)\cup N_3(v)$ 中的某个点 x，则使用 $N_1(v)$ 中的任一黑点代替 x，设 D 不包含 $N_2(v)\cup N_3(v)$ 中的任何点。显然，D 是 G' 的黑完全支配集。接着证明 $r_b(G')\geqslant r_b(G)$。设 D' 是 G' 的黑完全支配集。因为 $N(v')=v$ 且 v' 为白点，所以 D' 包含 v，不包含 v'。因为 $G'\backslash v'$ 是 G 的子图，且 v 能支配 $N_2(v)\cup N_3(v)$，所以 D' 是 G 的由黑点组成的完全支配集。

规则2 对于 G 中的黑点 v，若 $N_1(v)$ 中存在黑点，则将 $N_2(v)\cup N_3(v)$ 中的所有黑点 x 着色为白色，并删除 x 与其他白点间的边。

$N_2(v)\cup N_3(v)$ 中顶点的支配能力弱于 $\{v\}\cup N_1(v)$ 中的顶点，因此将 $N_2(v)\cup N_3(v)$ 中的黑点着色为白色，表示不会将其放入解中。

引理2 设 G 为黑白着色图，G' 为对 G 执行一次规则2后的结果图，则 $\gamma_b(G)=\gamma_b(G')$。

证明：设 D 为图 G 的黑完全支配集。对于 G 中的黑点 v，设 $S=(N_2(v)\cup N_3(v))\cap D$，$w$ 为 $N_1(v)$ 中的某个黑点。若 $|S|\geqslant2$，则设 $D'=(D\backslash S)\cup\{v,w\}$；若 $|S|\leqslant1$，则设 $D'=(D\backslash S)\cup\{v\}$。显然，$D'$ 是 G' 的黑完全支配集。接着证明 $r_b(G')\geqslant r_b(G)$。设 D' 是 G' 的黑完全支配集。因为 $E(G')\subseteq E(G)$，$V(G')=V(G)$，所以 D' 是 G 的黑完全支配集。

规则3－规则7分析 G 中一对顶点及其邻接点结构，进行顶点删除或着色。

规则3 对于 G 上的相邻黑点 v 和 w，设 $B=N_3(v,w)\cup N_2(v,w)\cap N(v)\cap N(w)$。假设 $N_2(v,w)\cup N_3(v,w)$ 中的任何一个点都不能支配 $N_3(v,w)$，则有：

情况(1) v 和 w 均能支配 $N_3(v,w)$，则从 G 中删除 B 上的顶点，增加一个白点 z 及两条边 (v,z) 和 (w,z) 到 G 中；

情况(2) v 能支配 $N_3(v,w)$，但 w 不能支配 $N_3(v,w)$，则从 G 中删除 B 上的顶点，增加一个白点 v' 及一条边 (v,v') 到 G 中；

情况(3) w 能支配 $N_3(v,w)$，但 v 不能支配 $N_3(v,w)$，则从 G 中删除 B 上的顶点，增加一个白点及一条边 (w,w') 到 G 中；

情况(4) v 和 w 均不能支配 $N_3(v,w)$，则从 G 中删除 B 上了顶点，增加两个白点 v' 和 w' 及两条边 (v,v') 和 (w,w') 到 G 中。

规则3主要基于以下观察：$\{v,w\}\cup N_1(v,w)$ 中的顶点支配 $N_3(v,w)$ 的能力强于 $N_2(v,w)\cup N_3(v,w)$ 中的顶点。白点 z 保证 $\{v,w\}$ 中至少有一个顶点在解中。

引理3 设 G 为黑白着色图，为对 G 执行一次规则3后的结果图，则 $\gamma_b(G)=\gamma_b(G')$。

证明：设 D 为 G 最小的黑完全支配集。设 $M=\{v,w\}\cup N_2(v,w)\cup N_3(v,w)$，$S_1=D\cap M$，$D'=(D\backslash S_1)\cup\{v,w\}$。由于 $N_3(v,w)\cup N_2(v,w)$ 中的任何一个点均不能支配 $N_3(v,w)$，若 $\{v,w\}\cap D=\emptyset$，则 D 至少包含 $N_2(v,w)\cup N_3(v,w)$ 中的两个点，使用 v 和 w 代替这两个点可得到另一个大小不超过 D 的完全支配集。设 D 至少包含 $\{v,w\}$ 中的一个点。显然，$|S_1|\leqslant2$，否则 D' 是比 D 严格小的黑完全支配集，由此产生矛盾。如果 $|S_1|=2$，因 v 和 w 是相邻黑点，所以 D' 是 G' 的黑完全支配集。如果 $|S_1|=1$，此时 S_1 所增加的白点与 S_1 中的点在所有情况下均是相邻的，且在所有情况下删除的点来自 $N_2(v,w)\cup N_3(v,w)$，并非是 S_1 中的点。因此，D 为 G' 图的黑完全支配集。接着证明 $r_b(G')\geqslant r_b(G)$。设 D' 是 G' 最小的黑完全支配集，只需证明 D' 未包含新增点且能够支配被

删除点。4 种情况的证明过程类似，下面以情况(1)为例说明证明过程。因为 z 需要被支配，D'至少包含$\{v,w\}$中的一个点。由于 z 为白点，因此 D'不包含 z。又由于 $N_3(v,w)\cup N_2(v,w)\cap N(v)\cap N(w)\subseteq N(v)\cap N(w)$，因此 D'能支配被删除的点集 $N_3(v,w)\cup N_2(v,w)\cap N(v)\cap N(w)$。

规则 4　对于图 G 上的相邻黑点 v 和 w，若 $N_3(v,w)\neq\emptyset$，则有：对于每一个黑点，如果 x 不能支配 $N_3(v,w)$，则将 x 着色为白色并删除 x 与其他白点间的边。

规则 4 基于以下观察：$N_3(v,w)\cup N_2(v,w)$中黑点支配 $N_3(v,w)$的能力弱于$\{v,w\}$中的顶点。

引理 4　设 G 为黑白着色图，G'为对 G 执行一次规则 4 后的结果图，则 $\gamma_b(G)=\gamma_b(G')$。

证明：设 D 为 G 最小的黑完全支配集。设 $x\in N_2(v,w)\cup N_3(v,w)$不能支配 $N_3(v,w)$。如果 $x\in D$，x 不能支配 $N_3(v,w)$，则为了支配 $N_3(v,w)$，D 至少包含 $N_2(v,w)\cup N_3(v,w)\cup\{v,w\}$中的另外一个点 x'。设 $D'=D\backslash\{x,x'\}\cup\{v,w\}$。显然，$D'$中的点在 G'中均为黑点，且是 G'的完全支配集。接着证明 $r_b(G')\geqslant r_b(G)$。设 D'为图 G'的最小黑完全支配集。显然，D'中的所有点在 G 中均为黑点。因为 $E(G')\subseteq E(G)$，$V(G')=V(G)$，所以 D'是 G 的完全支配集。

规则 5　对于 G 上的相邻点 v 和 w，设 $N[v]=N[w]$且 $N(v)\cap N(w)$中存在黑点：如果$\{v,w\}$中存在白点，则将$\{v,w\}$中的某个白点从 G 中删除，否则将$\{v,w\}$中的任一个点从 G 中删除。

规则 5 中的顶点 v 和 w 是“孪生”兄弟，最优完全支配集中至多包含其中的一个点。

引理 5　设 G 为黑白着色图，G'为对 G 执行一次规则 5 后的结果图，则 $r_b(G)=r_b(G')$。

证明：设 D 为 G 最小的黑完全支配集。如果$\{v,w\}$中存在白点，因为 G'是 G 的子图，G 中没有黑点被着色为白色，且被删除的点是白点，所以 D 为图 G'的黑完全支配集。如果 v 和 w 均为黑点，不失一般性，设删除的点为 v。如果 $v\in D$ 且 w 不在 D 中，设 $D'=D\backslash\{v\}\cup\{w\}$；如果 $v\in D$ 且 w 在 D 中，设$D'=D\backslash\{v\}\cup\{x\}$，其中 $x\in N(v)\cap N(w)$为黑点。因为被删除的点只有 v，所以 D'为图 G 的黑完全支配集。接着证明 $r_b(G')\geqslant r_b(G)$。设 D'为图 G'的最小黑完全支配集。不失一般性，设删除的点为 v。因 w 需要被支配，D'一定包含 $N(w)$中的某个点 x。因为 $v\in N(x)$能被 x 支配，所以 D'为图 G 的黑完全支配集。

规则 6　对于 G 中的两个白点 u 和 v，若 $N(u)\subseteq N(v)$，则将 v 点从 G 中删除。

G 中任何完全支配集若能实现对 u 的支配，则必能实现对 v 的支配。同时由于 v 没有支配能力，因此可安全删除 v。

引理 6　设 G 为黑白着色图，G'为对 G 执行一次规则 6 后的结果图，则 $\gamma_b(G)=\gamma_b(G')$。

证明：设 D 为 G 最小的黑完全支配集。显然 D 不包含 v。因为 v 是唯一一个被删除的点，没有点被改变颜色，所以 D 是 G'的黑完全支配集。接着证明 $r_b(G')\geqslant r_b(G)$。设 D'为图 G'中最小的黑完全支配集。因为 u 需要被支配，D'至少包含一个 $N(u)$中的点，$N(u)\subseteq N(v)$，所以 D'至少包含一个图 G 中 v 的邻接点，从而 D'在 G 上能支配 v。因此 D'是 G 的一个黑完全支配集。

规则 7　对于 G 上的一对黑点 v 和白点 w，若 w 与 $N(v)$中的每个黑点均相邻，则将 w 从图 G 中删除。

因为 v 需被支配，所以 $N(v)$中至少有一个黑点 u 在解中。u 能支配 w，同时由于 w 没有支配能力，因此可安全删除 w。

引理 7　设 G 为黑白着色图，G'为对 G 执行一次规则 7 后的结果图，则 $\gamma_b(G)=\gamma_b(G')$。

证明：设 D 为图 G 的最小黑完全支配集。因为 G'是 G 的子图，G 中无黑点被着色为白色，且被删除的点是白点，所以 D 为 G'的黑完全支配集。接着证明 $r_b(G')\geqslant r_b(G)$。设 D'为图 G'的最小黑完全支配集。设 $M=\{x\in N(v)\mid x$ 为黑点$\}$。因为 v 需被支配，所以 $D\cap M\neq\emptyset$。$D\cap M$ 能够支配 w，因此 D'为 G'的黑完全支配集。

基于引理 1－引理 7，可推出规则 1－规则 7 的正确性。

定理 1　给定完全支配集实例 $G(V,E)$，设图 G 中的顶点均为黑色，G'为执行 l 次规则 1－规则 7 后的结果图，则 $\gamma(G)=\gamma(G')$。

证明：G 中的顶点最初均为黑色，因此 $\gamma(G)=\gamma_b(G)$。设 G''为在 G 上执行 $l-1$ 次规则 1－规则 7 后的结果图，G'为在 G''上执行 1 次规则 1－规则 7 后的结果图。设 $\gamma(G'')=\gamma_b(G'')$。根据引理 1－引理 7，可推出 $\gamma(G)=\gamma_b(G)=\gamma_b(G'')$，从而 $\gamma(G)=\gamma_b(G'')=\gamma(G'')$。由引理 1－引理 7 可得 $\gamma_b(G'')=\gamma_b(G')$。显然，$\gamma_b(G')\geqslant\gamma(G')$。只需证 $\gamma_b(G')\leqslant\gamma(G')$即可得 $\gamma(G)=\gamma(G')$。下面分规则证明 $\gamma_b(G'')\leqslant\gamma(G')$。

针对规则 1，设 D 为图 G'的最小完全支配集。因为 v'需要被支配，所以 D 包含 v。如果 D 包含 v'，则可用 $N_1(v)$中的某个黑点代替 v'。设 D 不包括 v'。由于删除的点包含在 $N(v)$中，因此 D 为图 G''的完全支配集，从而 $\gamma(G')\geqslant\gamma(G'')=\gamma_b(G')$。

针对规则 2 和规则 4，由于图 G'和 G''相同，而 G'中的边集合是 G''中边集合的子集，因此 $\gamma(G')\geqslant\gamma(G'')=\gamma_b(G')$。

针对规则 3，新增点是度为 1 或 2 的点，保证了新增点的至少一个邻接点在图 G'的最小完全支配集中，这些邻接点保证了被删除点的支配，且可观察到存在不包含新增点的图 G'的最小完全支配集。

针对规则 5，设被删除点为 v。对于图 G'的最小完全支配集 D，D 在 G'上能支配 w，保证了 D 在 G''能支配 v，因此 D 是 G''的完全支配集。

针对规则 6，设 D 为图 G'的最小完全支配集。因为 u 需要被支配，所以 $D\cap N(u)\neq\emptyset$。$D\cap N(u)$在 G''上支配了被删除点 v，因此 D 是 G''的完全支配集。

针对规则 7，设被删除点为 w。图 G'的最小完全支配集 D 在 D'上能支配 v，保证了 D 在 G''上能支配 w，因此 D 是 G''的完全支配集。

4　规约化算法

规约化算法是不断执行规约规则的过程，如算法 1 所示。

算法 1　Reduction_TDS 算法

输入：图 G＝(V，E)

输出：图 G′

1. G′＝G；

2. 将 G′中的所有点着色为黑色；

3. While true do

　3.1 在 G′上执行规则 1－规则 7；

　3.2 If 规则 1－规则 7 均未被成功执行 then 退出循环；

4. Return

步骤 3.1 会在图中寻找满足规则执行条件的局部结构，若存在则执行相关规则。若某规则未被成功执行，则表示当前图中不存在满足该规则执行条件的局部结构。

接下来给出算法的正确性证明。

定理 2 给定输入实例 $G(V,E)$，设 G'为算法 Reduction_TDS 的输出图，则：1）$\gamma(G)=\gamma(G')$；2）基于 G'的最优完全支配集可在 $O(n)$时间内构造出 G 的最优完全支配集，其中 $n=|V|$。

证明：规则 1—规则 7 的每次执行均会减小问题的规模，因此算法 Reduction_TDS 中规则 1—规则 7 的执行次数有限。结合定理 1 可直接得出 $\gamma(G)=\gamma(G')$。基于 G'的最小完全支配集 D'，构造 G 的最小完全支配集 D 的方法如下：令 $D=D'$。如果 D 包含非 G 中的顶点 x（执行规则后新增加的顶点），根据规则的执行过程，x 在 G'中的度为 1 或 2。如果 x 的度为 1，设 v 为其邻接点，用 v 在 G 上的某邻接点代替 x；如果 x 的度为 2，设 v 为 x 的不属于 D 的邻接点，用 v 代替 x。重复以上替换过程直到 D 中的顶点均来自 G。显然，D 是 G 的完全支配集，且以上构造过程可在线性时间内完成。

5 实验

测试规约算法在模拟无线网络环境中的性能。假设所有无线节点装配了相同的电池，其感知半径 r 相同。网络实例 $G(V,E)$按如下方法产生。设 V 为无线网络节点的集合。将所有无线节点随机洒在一个大小为 100m×100m 的区域中。对于任意两个网络节点 u 和 v，若 v 在 u 的感知范围内，则 $(u,v)\in E$。要求 $G(V,E)$上不存在孤立点，否则舍弃该拓扑。通过设置网络节点数 n 和感知半径 r 的大小来控制网络的规模和稠密度。实验平台配置：CPU 为 Intel i3-3110M，主频为 2.40GHz，内存为 4GB。

本文主要从规则处理的顶点个数和算法运行时间这两方面测试算法的性能。其中，规则处理的顶点个数为以下 3 种顶点数之和：删除的顶点个数、放入解中的顶点个数和由黑色变为白色的顶点个数。

图 1—图 3 给出了网络规模 n 为 200、感知半径 r 分别为 15，10，8 时的规约处理结果，每组实验中生成了 50 个随机实例以进行测试。图中的横坐标表示实例的编号，纵坐标表示顶点处理率。顶点处理率为规则处理的顶点数目与输入网络中顶点数目的比值。

图 1 中感知半径 r 设置为 15，网络实例中边的数目平均为 1500 左右，为较稠密网络。图 1 中的结果表明，顶点处理率平均为 40%左右。

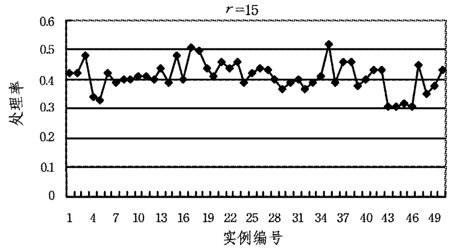

图 1 较稠密网络上的规约处理

图 2 中感知半径 r 设置为 10，网络实例中边的数目平均为 800 左右，为中等稠密网络。图 2 中的顶点处理率平均为 50%左右，处理率有所上升。

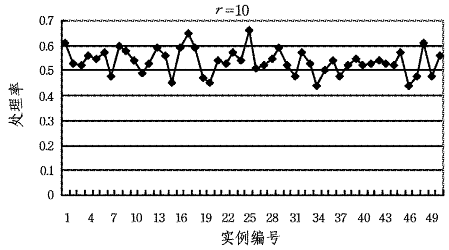

图 2 中等稠密网络上的规约处理

图 3 中感知半径 r 设置为 8，网络实例中边的数目平均为 400 左右，为较稀疏网络。图 3 中的结果表明，顶点处理率平均为 60%左右，处理率进一步上升。

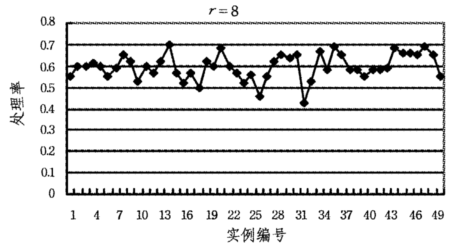

图 3 稀疏网络上的规约处理

图 1—图 3 的结果表明，规约算法的性能与网络稠密度相关。图 4 进一步给出了稠密度对算法规约效果的影响。将感知半径设为 10，顶点个数 n 从 100 逐步增加到 300（步长为 50）。当顶点数逐步增加时，网络的密度也逐步增加。对于特定的 n，生成 10 个不同的测试实例。图 4 中的横坐标为网络规模，纵坐标为 10 个不同测试实例的平均处理率。从图 4 的整体趋势来看，规约算法在稀疏网络上的效果较好。随着网络稠密度的增加，处理率逐步下降。

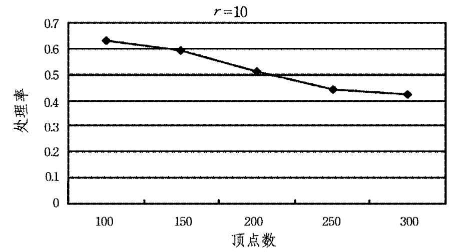

图 4 稠密度对规约效果的影响

最后测试规则的时间性能。在顶点个数为 500 的中等规模网络中，在 1s 内完成算法的运行；在顶点个数为 1500 的较大规模网络中，在 10s 内完成规约算法的运行。算法的耗时主要集中在规则 3 中，这与理论分析一致。

结束语 针对完全支配集设计了能减小问题规模的规约规则，并在理论上证明了规则的正确性。实验表明，提出的规约规则在各种不同规模和不同稠密度网络上均能较大程度地降低问题规模。设计的规约规则可直接应用到启发式或近似算法的优化中，在实际求解完全支配集前降低问题规模，从而提高启发式或近似算法的有效性。

参 考 文 献

[1] CHEN J, HE K, DU R, et al. Dominating Set and Network Coding-Based Routing in Wireless Mesh Networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2015, 26(2): 423-433.

（下转第 132 页）

ID3 算法折线的下方，这说明相较于 ID3 算法，DTAAS 算法对规模较小的数据集的分类效果较差；对于后两个数据集，DTAAS 算法的折线处于 ID3 算法折线的上方，这说明相较于 ID3 算法，DTAAS 算法对规模较大的数据集的分类效果较好；且随着数据集规模的增加，DTAAS 算法和 ID3 算法在分类精度上的差值逐渐增大，最高可达到 13 个百分点，这进一步说明在分类精度方面 DTAAS 算法对规模较大的数据集的运行效果更好。

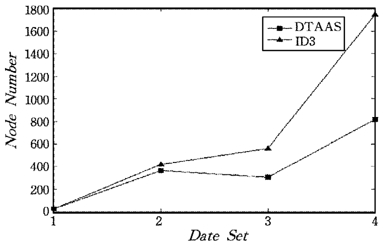

图 4　节点个数折线图

在图 4 中，随着数据集规模的增加，两条折线均呈上升趋势，这说明数据集规模越大，两种算法得到的树的节点数越多。DTAAS 算法的折线处于 ID3 算法折线的下方，这说明相较于 ID3 算法，DTAAS 算法得到的树的节点数更少，这意味着 DTAAS 算法的规则更少，其时间开销更少。且随着数据集规模的增加，DTAAS 算法和 ID3 算法在节点数上的差值逐渐增大，最高可以缩减一半左右，这进一步说明在节点个数方面，对于规模不同的数据集，DTAAS 算法对规模较大的数据集的运行效果更好。

综上，相较于 ID3 算法，DTAAS 算法在对规模较大的数据集进行分类任务时的分类精度更高，得到的树的节点数更少，算法运行得更快。DTAAS 算法适用于规模较大的数据集。

结束语　为了提高传统决策树的分类精度，本文将属性重要度与 β-正域相结合，提出了一种新的分类属性的选择标准，且通过置信度 *delta* 来控制决策树的生长，提出了 DTAAS 算法。实验证明，相较于 ID3 算法，DTAAS 算法在针对数据量较大的数据集进行决策树构造时效果更好，可以在提高分类精度的同时减小决策树的规模。但如何将粗糙集知识与决策树算法进行更好的结合，使得生成的决策树最优是未来的研究方向。

参 考 文 献

[1] 梁凤兰. 优化决策树改进挖掘算法仿真[J]. 计算机仿真，2013，30(11)：264-267.

[2] 张棪，曹健. 面向大数据分析的决策树算法[J]. 计算机科学，2016，43(6A)：374-379.

[3] QUINLAN J R. Induction of Decision Trees[J]. Machine Lerning，1986，1(1)：81-106.

[4] QUINLAN J R. Simplifying Decision Trees[J]. International Journal of Man-machine Studies，1987，27(3)：221-234.

[5] 洪家荣，丁明锋，李星原. 一种新的决策树归纳学习算法[J]. 计算机学报，1995，18(6)：470-474.

[6] 刘小虎，李生. 决策树的优化算法[J]. 软件学报，1998，9(10)：797-800.

[7] WANG S Q，WEI J M，YOU J P，et al. A VPRSM based approach for inducing decision trees[C]//RSKT2006. Chongqing，China，2006：421-429.

[8] 洪雪飞，徐维祥. 基于变精度粗糙集的决策树改进方法[J]. 计算机工程与应用，2009，45(13)：163-165.

[9] 丁春荣，李龙澍. 变精度粗糙集模型在决策树构造中的应用[J]. 计算机工程与科学，2010，32(7)：86-88.

[10] 鄂旭，任骏原，毕嘉娜，等. 基于粗糙变精度的食品安全决策树研究[J]. 计算机技术与发展，2014，24(1)：242-245.

[11] BARANAUSKAS J A. The number of classes as a source for instability of decision tree algorithms in high dimensional datasets[J]. Springer，2015，43(2)：301-310.

[12] LIANG C Q，ZHANG Y，SHI P，et al. Learning accurate very fast decision trees from uncertain data streams[J]. Taylor & Francis，2015，46(16)：3032-3050.

[13] 王婧，王兴伟，赵悦. 基于变精度粗糙集决策树垃圾邮件过滤[J]. 系统仿真学报，2016，28(3)：705-710.

[14] APNIK V. The nature of statistical learning theroy[M]. New York：Springer，1995.

[15] PAWLAK Z，SO-WINSKI R. Rough set approach to multi-attribute decision analysis[J]. European Journal of Operational Research，1994，72(3)：443-459.

[16] LIU Y，HUANG W，JIANG Y，et al. Quick attribute reduct algorithm for neighborhood rough set model[J]. Information Sciences，2014，271(7)：65-81.

[17] 娄畅，刘遵仁，郭功振. 基于块集的邻域粗糙集的快速约简算法[J]. 计算机科学，2014，41(S2)：337-339.

(上接第 118 页)

[2] 路纲，周明天，唐勇，等. 任意图支配集精确算法回顾[J]. 计算机学报，2010，33(6)：1073-1087.

[3] 马晨明，王万良，洪榛. 无线传感器网络(k，m)-容错连通支配集的分布式构建[J]. 计算机科学，2016，43(1)：128-133.

[4] WANG D，ZHANG Q，LIU J. The self-pro-tection problem in wireless sensor networks[J]. ACM Transactions on Sensor Networks (TOSN)，2007，3：20.

[5] WANG Y，LI X，ZHANG Q. Efficient algorithms for p-self-protection problem in static wireless sensor networks[J]. IEEE Transactions on Parallel and Distributed Systems，2008，19：1426-1438.

[6] PFAFF J，LASKAR R C，HEDETNIEMI S T. NP-completeness of total and connected domination and irredundance for bipartite graphs[R]. Clemson University，1983.

[7] HE J，LIANG H. Complexity of total $\{k\}$-domination and related problems [C]// Proceedings of Frontiers in Algorithmics and Algorithmic Aspects in Information and Management(FAW-AAIM). Berlin：Springer Berlin Heidelberg，2011：147-155.

[8] ALBER J，NIEDERMEIER R. Improved tree decomposition based algorithms for domination-like problems[C]// Proceedings of the 5th Latin American Symposium on Theoretical Informatics (LATIN'02). Berlin：Springer Berlin Heidelberg，2002：613-627.

[9] ARCHDEACON D，ELLIS-MONAGHAN J，FISCHER D，et al. Some remarks on domination[J]. Journal of Graph Theory，2004，46：207-210.

[10] ZHAO W，WANG H，XU G. Total k- domination number in graphs[J]. International Journal of Pure and Applied Mathematics，2007，35(2)：235-242.

[11] 骆伟忠，冯启龙，王建新，等. 完全 p-支配集的参数算法[J]. 计算机学报，2013，36(9)：1868-1879.

[12] GANIAN R，SLIVOVSKY F，SZEIDER S，et al. Meta-kernelization with structural parameters[J]. Journal of Computer and System Sciences，2016，82(2)：333-346.