最大团问题研究进展及算法测试标准*

王丽爱[1]，周旭东[2]，陈 崚[1,3]

(1. 扬州大学 信息工程学院 计算机系，江苏 扬州 225009; 2. 扬州大学 信息工程学院 计算中心，江苏 扬州 225009; 3. 南京大学 软件新技术国家重点实验室，江苏 南京 210093)

摘 要: 定义了最大团问题，分析和研究了使用启发式算法求解最大团问题的进展，介绍了当前求解最大团问题的典型启发式算法，最后给出了测试这些启发式算法性能的测试基准图。

关键词: 最大团问题; 启发式算法; 组合优化

中图分类号: TP301 **文献标志码**: A **文章编号**: 1001-3695(2007)07-0069-02

Research Development of MCP and Benchmark Graphs

WANG Li-ai[1], ZHOU Xu-dong[2], CHEN Ling[1,3]

(1. Dept. of Computer Science, Institute of Information Technology, Yangzhou University, Yangzhou Jiangsu 225009, China; 2. Computer Center, Institute of Information Technology, Yangzhou University, Yangzhou Jiangsu 225009, China; 3. National Key Laboratory of Novel Software Technology, Nanjing University, Nanjing Jiangsu 210093, China)

Abstract: The definition of MCP was described, and the development about using heuristic algorithms to solve MCP was analyzed and researched. Several typical heuristic algorithms about solving MCP were introduced. Finally the test benchmark graphs for testing the performance of these algorithms were described.

Key words: maximum clique problem(MCP); heuristic algorithm; combinatorial optimization

最大团问题是图论中的一个经典组合优化问题，也是一类NP完全问题，对最大团问题的研究具有很大的理论价值。最大团问题也被称为最大独立集问题，在实践中有非常广泛的应用，被应用于市场分析、方案选择、信号传输、计算机视觉、故障诊断等不同领域。最大团问题在国外有相对广泛的研究，在国内的研究则刚刚起步。图分为有权图和无权图，本文主要讨论针对无权图的最大团问题。

1 最大团问题定义

1.1 最大团问题的基本定义

对于给定图 $G=(V,E)$。其中，$V=\{1,\cdots,n\}$ 是图 G 的顶点集，$E\subseteq V\times V$ 是图 G 的边集。图 G 的团就是一个两两之间有边的顶点集合。如果一个团不被其他任一团所包含，即它不是其他任一团的真子集，则称该团为图 G 的极大团。顶点最多的极大团，称之为图 G 的最大团。最大团问题的目标就是要找到给定图的最大团。

最大团问题也被称为最大独立集问题，独立集指的是两两顶点之间没有边的顶点集合。顶点最多的独立集就是最大独立集。如果 C 是图 G 的最大团，其实 C 也是 $\overline{G}$ 的最大独立集。

1.2 最大团问题的数学描述

最大团问题作为一个整数规划问题有许多等价的描述。整数规划问题的描述(二次 0-1 问题)[1]:

设 $t:(0,1)^n\to 2^V$, $t(x)=\{i\in V: x_i=1\}$, $\forall x\in\{0,1\}^n$, $\forall S\in 2^V$, 则 $x=t^{-1}(S)=\{x_i: i=1,2,\cdots,n\}$

其中，$x_i=\begin{cases}1 & i\in S\\ 0 & i\notin S\end{cases}$; n 为图的顶点数。

$$\min f(x)=-\sum_{i=1}^{n}x_i \tag{1}$$
$$\text{s.t.}\quad x_i+x_j\le 1,\quad \forall(i,j)\in\overline{E},\ x\in\{0,1\}^n$$

如果 x^* 是式(1)的最优解，那么集合 $C=t(x^*)$ 是图 G 的一个最大团，且 $|C|=-f(x^*)$。

由于 $x_i,x_j\in\{0,1\}$, $x_i+x_j\le 1$, $\forall(i,j)\in\overline{E}$ 当且仅当 $x_ix_j=0$。有 $f(x)=-\sum_{i=1}^{n}x_i+2\sum_{(i,j)\in E,i>j}x_ix_j=x^{\mathrm{T}}(A_{\overline{G}}-I)x$。其中，$A_{\overline{G}}$ 为图 G 的补图 $\overline{G}$ 的邻接矩阵。

MCP 等价于下面的全局二次 0-1 问题:

$$\min f(x)=x^{\mathrm{T}}Ax \tag{2}$$
$$\text{s.t.}\quad x\in\{0,1\}^n$$

其中，$A=A_{\overline{G}}-I$。如果 x^* 是式(2)的最优解，那么集合 $C=t(x^*)$ 是图 G 的一个最大团，且 $|C|=-f(x^*)$。

2 最大团问题研究进展

1957 年，Hararv 和 Ross 首次提出求解最大团问题的确定性算法(Exact Algorithm)[1]，随后研究者们提出各种各样的确定性算法来求解最大团问题。随着研究的深入，遇到的问题复杂度越来越高(顶点增多和边密度变大)，确定性算法逐渐不

收稿日期: 2006-05-22; 修返日期: 2006-07-05 基金项目: 国家自然科学基金资助项目(60473012); 国家科技攻关资助项目(2003BA614A-14); 江苏省自然科学基金资助项目(BK2005047); 南京大学软件新技术国家重点实验室开放基金资助项目

作者简介: 王丽爱(1975-)，女，山西祁县人，助教，硕士，主要研究方向为算法设计及人工智能(wla001@163.com); 周旭东(1979-)，男，山西武乡人，助教，硕士，主要研究方向为智能算法、Agent; 陈崚(1951-)，男，江苏宝应人，教授，博导，主要研究方向为算法设计和并行计算.

能有效解决这些问题。

由于最大团问题是一类 NP 完全问题,确定性算法不能够有效解决这类问题,典型地体现在求解问题时运算时间过长,且不能够很好地克服这一问题。目前确定性算法主要用于求解顶点数相对较少、密度相对较低的图的最大团。

针对上面的情况,在 20 世纪 80 年代末开始采用启发式算法求解最大团问题,提出了各种各样的启发式算法,并且取得了令人满意的效果。在时间上,由于采用了启发式信息,启发式算法的运算时间与确定性算法的运算时间之间的比值会随着图的顶点、密度的增加而变得越来越小。唯一的缺点就是不一定能找到最优值,有时只能找到近优值。

启发式算法的出现给求解顶点多、密度高的图的最大团问题带来了希望,并且经过这十几年的发展,取得了长足进步。大部分确定性算法所不能解决的图,用启发式算法都能得到有效解决。人们对启发式算法关注程度越来越高。

当前求解最大团问题的启发式算法都基于以下几类:局部搜索启发式算法(Local Search Heuristics)、遗传算法、神经网络、模拟退火和禁忌搜索、基于连续的启发式算法等。在局部搜索启发式算法中,比较著名的算法是 K-interchange 启发式算法,该算法基于 K-neighbor 实现。遗传算法在 1993 年被 Carter 和 Park 第一次引进来求解最大团问题[2]。神经网络在 1987 年被 Ballard 等人引进来解决最大团问题[3]。1989 年,Aarts 和 Korst 把模拟退火引进来求解最大团问题[4]。禁忌搜索在 1989 年由 Friden 等人[5]被用来求解该问题。1990 年由 Pardalos 等人[6]提出一种基于连续的启发式算法来求解最大团问题。

研究表明,单独使用一种启发式算法求解最大团问题,算法性能并不是很好,因此,需要算法之间互相取长补短,形成新的混合算法来求解最大团问题。当前求解该问题最好的启发式算法有反作用禁忌搜索算法(Reactive Tabu Search)、简单启发式算法(Simple Heuristic Based Genetic Algorithm, HGA)和 DLS-MC 等。

1998 年,Marchiori[7]给出了一种基于遗传算法的简单启发式算法(HGA)。该算法由两部分组成,即简单遗传算法(Simple Genetic Algorithm, SGA)和简单的贪婪启发式局部搜索算法。Marchiori 在基准图上对算法 HGA 的性能进行测试,证明了该算法在解的质量和计算速度上均优于基于遗传算法的其他算法。

Battiti 和 Protasi[8]提出的反作用禁忌搜索算法,通过引入局部搜索法(Reactive Local Search)扩展了禁忌搜索的框架。与一般的禁忌搜索算法相比,该算法的特点是,在执行过程中,根据所得到的解的情况形成一种内部反馈来控制和调整算法的控制参数。所以该算法的控制参数是动态变化的。比如,禁止表长度参数动态变化,使禁忌表长度动态变化。在 DIMACS 基准图上进行测试,取得了非常好的效果。

Wayne Pullan 等人[9]最近提出解最大团问题的方法 DLS-MC。该算法基于迭代改善法和 Plateau Search 算法。在该方法中引入了顶点惩罚函数(Vertex Penalties),该函数在算法的求解过程中能够动态改变。在算法执行过程中,迭代改善法和 Plateau Search 算法轮流执行以提高解的质量。在基准图上对该算法进行了测试,性能非常好。

3 启发式算法测试标准——DIMACS 基准图

DIMACS 组织是由美国政府成立的离散数学及理论计算机科学中心。该中心是当前离散数学和理论计算机科学的重要研究阵地之一。

DIMACS 基准图是 DIMACS 提供的图,在 1993 年以前由于对最大团算法性能测试没有统一的标准,算法之间无法进行有效比较。因此,DIMACS 在 1993 年集中当时比较有代表性的图形成基准图,目的是提供统一的最大团算法测试标准图[10]。DIMACS 组织统计了当时各种启发式算法在 DIMACS 基准图测试所取得结果,并且由其中最好的解组成 DIMACS Best,作为比较对象(表 1)。DIMACS 基准图可通过登录 ftp://dimacs.rutgers.edu/pub/challeng/或者 http://dimacs.rutgers.edu/challenges/获得。

表 1 DIMACS 基准图及 DIMACS Best

基准图名	顶点数	边密度	DIMACS Best
C125.9	125	0.898	34*
C250.9	250	0.898	44*
C500.9	500	0.900	57
C1000.9	1 000	0.901	67
C2000.9	2 000	0.900	75
C2000.5	2 000	0.500	16
C4000.5	4 000	0.500	18
DSJC500.5	500	0.501	14*
DSJC1000.5	1 000	0.500	15*
MANN_a27	378	0.990	126*
MANN_a45	1 035	0.996	345*
MANN_a81	3 321	0.998	1 098
brock200_2	200	0.496	12*
brock200_4	200	0.657	17*
brock400_2	400	0.749	29*
brock400_4	400	0.748	33*
brock800_2	800	0.651	21
brock800_4	800	0.649	21
gen200_p0.9_44	200	0.900	44*
gen200_p0.9_55	200	0.900	55*
gen400_p0.9_55	400	0.900	55
gen400_p0.9_65	400	0.900	65
gen200_p0.9_75	400	0.900	75
hamming8-4	256	0.639	16*
hamming10-4	1 024	0.828	40*
keller4	171	0.649	11*
keller5	776	0.751	27
Keller6	3361	0.818	59
p_hat300-1	300	0.243	8*
p_hat300-2	300	0.488	25*
p_hat300-3	300	0.744	36*
p_hat700-1	700	0.249	11*
p_hat700-2	700	0.497	44*
p_hat700-3	700	0.748	62
p_hat1500-1	1 500	0.253	12*
p_hat1500-2	1 500	0.506	65
p_hat1500-3	1 500	0.753	94

4 结束语

最大团问题是现实世界中的一类真实问题。(下转第 107 页)

作, 需要一种能够自动收集进度信息、给出进度分析的自动化系统。基于上面的进度度量模型, 设计了一个进度度量工具(PMT), 该工具基于 B/S 架构, 实现灵活的使用和部署。PMT 在 Lucky ERP System(Lucky ERP)、统计局业务系统(SIS)、办公自动化系统(MHOA)几个项目中得到了应用。表 5 是这几个项目的跟踪状况。

表 4　分析指标体系表

级别	指标	计算方法
执行官级别	项目进度偏差	$P - P'$
	项目需求稳定指数	$(\Delta P \times U_t \times t_u - \Delta C_r) / (\Delta P \times U_t \times t_u)$
	需求与分析进度偏差	$P_a - P'_a$
部门经理级别	设计与构建进度偏差	$P_d - P'_d$
	测试进度偏差	$P_t - P'_t$
	项目错误工作指数	$\Delta C_f / (\Delta P \times U_t \times t_u)$
	生产率偏差	$t_u - t'_u$
	单个员工工作进度偏差	$Et_i - Et_i'$
	单个员工生产率偏差	$t_{uj} - t_u$
	单个员工错误工作指数	$\Delta C_{fj} / (\Delta Et_j \times U_j \times t_{uj})$
项目经理级别	单个用例需求与分析进度偏差	$P_{ia} - P'_{ia}$
	单个用例设计与构建进度偏差	$P_{id} - P'_{id}$
	单个用例测试进度偏差	$P_{it} - P'_{it}$
	单个用例需求稳定指数	$(U_i \times t_u - C_{ri}) / (U_i \times t_u)$
	单个用例错误工作指数	$\Delta C_{fi} / (\Delta P_i \times U_i \times t_u)$

表 5　项目进度跟踪表

项目	计划	实际	偏差	需求变动指数
Lucky	24	26	2	10%
SIS	16	22	6	25%
MHOA	15	14	- 1	8%

从表 5 中可以看出, Lucky ERP System 基本上是按照计划完成了项目; MHOA 提前了 1 个月; 只有 SIS 延期了 6 个月。其中不可忽视的一个原因是该项目的需求变动指数过高, 频繁的需求变动直接导致了项目的超期。图 1 是对 MHOA 更详细的跟踪报告(月报)。

从图 1 可以看出在前几个月里项目已经落后进度, 对于这种状况项目管理人员采取了一些强化措施。可以看出, 措施的有效执行保证了项目的如期完成, 最后, 项目甚至提前一个月结束。

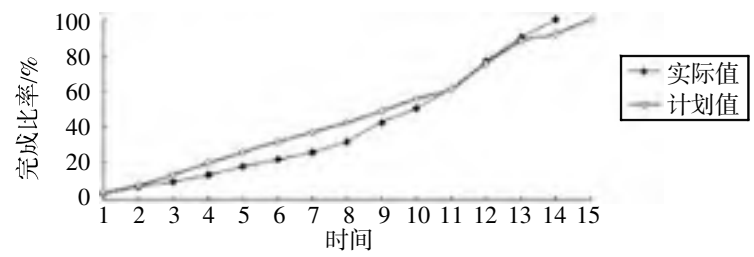

图 1　MHOA 进度对照图

5　结束语

本文提出的进度度量方法对于尽早发现项目的进度偏差具有较大作用。当进度偏差超过组织设定的基线时, 可以对进度偏差进行分析, 进度分析模型提供诸如错误工作指数、需求变动指数以及更详细的指示器可以为分析偏差原因提供有效的支持。

参考文献:

[1] ARDLEY D. Successful IT project delivery learning the lessons of project delivery learning the lessons of project failure[M]. New York: Addison Wesley, 2003.

[2] JACOBSON I, CHRISTERON M, TONSSON P, *et al*. Object-oriented software engineering: a use case driven approach[M]. New York: Addison Wesley Publishing Company, 1992.

[3] KARNER G. Metrics for objectory[D]. Sweden: University of Link ping, 1993.

[4] DAMODARAN M, WASHINGTON A N E. 用例点估算方法[J]. 徐异婕, 译. 非程序员, 2005(45): 61-65.

[5] 林锐. 软件工程与项目管理解析[M]. 北京: 电子工业出版社, 2003.

[6] BOEHM B W, ABTS C, BROWN A W, *et al*. Software cost estimation with COCOMO II[M]. USA: Pearson Education, 2005.

[7] 章恒种, 陈思功, 秦晓. 基于 UML 的软件复杂度估计及进度控制[J]. 计算机工程与设计, 2002, 23(10): 62-64.

[8] 李亚红, 郝克刚, 葛玮. 基于 GQM 模型的软件项目进度的度量过程[J]. 计算机应用, 2005, 25(6): 1448-1450.

(上接第 70 页)传统的确定性算法不能有效地进行求解, 因此使用了各种各样的启发式算法。目前国内通过启发式算法求解最大团问题的研究还处于初期阶段。

大量研究表明, 使用启发式算法求解最大团问题, 将多种算法结合使用所呈现的性能要优于单纯使用一种算法时的性能。目前, 测试求解最大团问题的启发式算法性能标准是由 DIMACS 组织所提供的 DIMACS 基准图, 其并不是最合理的算法性能比较的依据, 需要提出更为合适的测试标准。

参考文献:

[1] DLEMAN L M. Molecular computation of solutions to combinatorial problems[J]. Science, 1994, 226(11): 1021-1024.

[2] CARTER B, PARK K. How good are genetic algorithms at finding large cliques: an experimental study, Technical Report Bu-CS-93-015[R]. Boston: Computer Science Dept., Boston University, 1993.

[3] BALLARD D H, GARDNER P C, SRINIVAS M A. Graph problems and connectionist architectures, Technical Report TR 167[R]. New York: Dept Computer Science, University of Rochester, 1987.

[4] AARTS E, KORST J. Simulated annealing and boltzmann machines, a stochastic approach to combinational optimization and neural computing[M]. New York: Wiley, 1989.

[5] FRIDEN C, HERTZ A, DeWERRA D. Stabulus: a technique for finding stable sets in large graphs with tabu search[J]. Computing, 1989, 42(1): 35-44.

[6] PARDALOS P M, PHILLIPS A T. A global optimization approach for solving the maximum clique problem[J]. International Journal of Computer Mathematics, 1990, 33: 209-216.

[7] MARCHIORI E. A simple heuristic based genetic algorithm for the maximum clique problem: proc. of the ACM Symposium on Applied Computing[C]. New York: ACM Press, 1998: 366-373.

[8] BATTITI R, PROSTASI M. Reactive local search for the maximum clique problem[J]. Algorithmica, 2001, 29(4): 610-637.

[9] PULLAN W, HOOS H H. Dynamic local search for the maximum clique problem[J]. Journal of Artificial Intelligence Research, 2006, 25: 159-185.

[10] JOHNSON D, TRICK M. Cliques, coloring and satisfiability: second DIMACS implementation challenge[J]. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1996, 26: 533-558.