一种求解子图同构问题的改进遗传算法

项英倬[1] 魏 强[1] 游 凌[1] 石 浩[2]
(盲信号处理国家重点实验室 成都 610041)[1] (中国科学技术大学自动化系 合肥 230031)[2]

摘 要 子图同构(Subgraph Isomorphism)技术在计算机视觉、人工智能以及生物化学工程等领域具有重要的应用。文章聚焦于子图同构问题的求解算法,提出了一种基于基因遗传算法的改进算法。结合子图同构的特点,针对遗传算法中的杂交过程和进化过程,改进了传统的子代生成算法,提出了一种新的适应度函数来评估子代的适应性。新算法可以指引搜索过程更快地收敛到最优解,并能够以更高的概率求得最优解。通过仿真实验表明,提出的改进算法相较于传统的算法能够更好地处理大规模子图,并取得更好的效果。

关键词 子图同构,遗传算法,适应度函数,杂交过程
中图法分类号 TP311 **文献标识码** A

Improved Genetic Algorithm for Subgraph Isomorphism Problem

XIANG Ying-zhuo[1] WEI Qiang[1] YOU Ling[1] SHI Hao[2]
(National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu 610041, China)[1]
(Department of Automation, University of Science and Technology of China, Hefei 230031, China)[2]

Abstract Subgraph isomorphism plays an important role in computer vision, artificial intelligence and bio-chemical engineering. This paper focused on the subgraph isomorphism (SI) problem and proposed a novel method based on the genetic algorithm to solve it. The sub-generation producing method is improved during the crossover and evolution process. Moreover, a new fitness function was presented to measure the fitness of the population. The new algorithm is more fast to get convergence and can find the optimal solutions with higher probability. Experiments show that the proposed improved algorithm outperforms other traditional methods by processing large graphs.

Keywords Subgraph isomorphism, Genetic algorithm, Fitness function, Crossover

1 引言

图(Graphs)在科学工程领域里是描述数据结构的一个强大工具,图的节点(Node)通常用来表示物体,边(Edge)通常用来表示物体之间的关系。对未知物体、模式的识别过程中,首先会将该物体的结构转化为图,然后与已知的原型图进行匹配。该匹配过程就是求解图同构(Graph Isomorphism)问题。如果两个图 G 和 G' 的每一个节点和每一条边都能够一一对应,并且之间的关系可以保持不变,那么二者是同构的。如果一个图与另一个图的子图同构,那么称其为子图同构(Subgraph Isomorphism)。

在计算机视觉、集成电路检测[1]、控制优化[2]、机器人规划[3]、模式识别[4]、人工智能以及生物化学工程[5]等领域,子图同构技术有重要的应用价值。例如在计算机视觉领域,给定一个图 G 与图 H 作为输入,子图同构就是从图 G 中找到一个与图 H 同构的子图的计算任务。子图同构问题被认为是寻找最大完全子图问题[6]以及检测哈密顿环问题[7]的一个泛化,因此该问题是 NP-complete[8]。然而对于许多特殊的图,仍然可以采用一些剪枝回溯的方法在多项式时间完成子图同构的搜索。

如何快速求解子图同构问题一直是学术界的研究热点,随着智能时代的到来,该技术获得了越来越多的关注。最经典的图同构算法和子图同构算法由 Ullmann 于 1970 年提出[5]。该算法采用基于节点度的剪枝方法,大幅减少了回溯的搜索空间,从而加快了算法的速度。另一种经典的回溯算法由 Schmidt 和 Druffel 于文献[9]中提出,该算法将图表示为距离矩阵(distance matrix),并通过距离矩阵中的信息对图中的点进行初步划分。同时,利用距离矩阵中的信息对搜索树进行剪枝操作,来减少不必要的匹配操作。Cordella 等[10]提出了 VF2 算法,该算法首先在要匹配的图中找到一个满足匹配条件的点集,然后判断新加入的点是否可以满足匹配条件,如果满足匹配条件,则再找下一个点,否则进行回溯。Mckay 等[11-12]提出了将图转变成 Canonical Form 的形式并进行匹配的方法,该方法对于某些种类的图具有相当快的速度。然而,这些算法仅能够解决规模比较小的子图同构问题,而对于大规模的子图同构,一般采用启发式的搜索方法,比如 Kirkpatrick 等[13]的模拟退火算法以及基因遗传算法[14]。然而相比于模拟退火算法,遗传算法在计算性能上具有一定优势[15]。

2 子图同构

所谓的图是指一个数学结构 (V_g, E_g, ψ),其中 V_g 是非空

本文受国家自然科学基金(61174124)资助。

项英倬(1990-),男,博士生,主要研究方向为人工智能,E-mail:xiangyzh@foxmail.com(通信作者);魏 强(1987-),男,工程师,主要研究方向为信息处理;游 凌(1971-),男,高级工程师,主要研究方向为网络态势感知;石 浩(1990-),男,博士,主要研究方向为网络优化。

集，E_g 是定义在 V_g 上的二元关系集，而 ψ 是 E_g 到 $V_g \times V_g$ 的函数，若 $\psi(E_g)$ 中的元素都是有序对，则 (V_g, E_g, ψ) 称为有向图(Digraph)，记为 $D=(V_g(D), E_g(D), \psi_D)$。若 $\psi(E_g)$ 中的元素全是无序对，则 (V_g, E_g, ψ) 称为无向图(Undirected Graph)，记为 $G=(V_g(G), E_g(G), \psi_G)$。如果删去一个图 g 的一个顶点，以及该顶点所附带的边，则可以得到一个 g 的子图，记做 $g' \subset g$。图 G 与图 H 的同构指存在两个双射：则

$\theta: V_g(D) \rightarrow V_g(H)$

$\varphi: E_g(D) \rightarrow E_g(H)$

使得对任何 $a \in E_g(D)$ 都满足：

$\Psi_D(a)=(x,y) \Leftrightarrow \Psi_H(\varphi(a))=(\theta(x), \theta(y))$

而子图同构指图 G 与图 H 的一个子图存在同构关系。通常，子图同构问题的解可能存在多个，比如图 1 中至少存在着 $\{a,b,c,d\}$ 与 $\{1,2,3,4\}$ 以及 $\{6,3,2,4\}$ 两个对应关系。

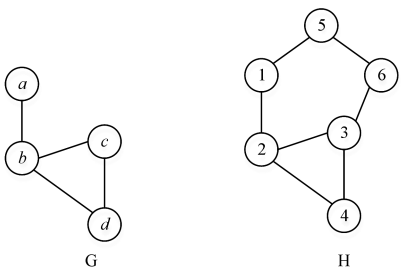

图 1　子图同构示例

3　子图同构问题建模

为了方便处理，我们把输入的每个图建模为邻接矩阵[16]，矩阵中的每个元素只能为 0 或者 1，每一行和列的元素都与图中的点一一对应，每个元素代表图中连接两个节点的边，如果对应的元素为 1，则表示该边存在，如果为 0 则相反。这样，每一个图都可以转化为与之对应的邻接矩阵，而每个邻接矩阵也可以转化为唯一的图，两者之间是等价的。容易知道，无向图的邻接矩阵是对称矩阵。对于图 $G=(V_g, E_g)$ 和图 $H=(V_h, E_h)$，对应的邻接矩阵分别为 $G=[g_{i,j}]$ 和 $H=[h_{i,j}]$，记图 G 的节点数为 N_g，图 H 的节点数为 N_h。我们定义一个转移矩阵[5] P，该矩阵的大小为 $N_g(rows) \times N_h(columns)$，其中的每一个元素均为 0 或者 1，而且矩阵的每一行只有一个 1，每一列最多包含一个 1。那么，如果图 H 是图 G 的一个子图，则邻接矩阵 G 与 H 满足：

$$P^{\mathrm{T}} * G * P = H \tag{1}$$

从式(1)可以看出，转移矩阵 P 代表了图 G 和图 H 的对应关系，转移矩阵中的每个 1 代表了图 G 和图 H 中节点的对应。假如图 G 的节点 i 与图 H 的节点 j 是对应的，那么转移矩阵 P 中 (i,j) 处的元素为 1。因此，求解图 G 与图 G 的子图同构问题等价于寻找满足式(1)的邻接矩阵。

4　子图同构问题的改进遗传算法

遗传算法的步骤一般包括编码、种群生成、交叉、变异和选择评估，本节将介绍遗传算法解决子图同构问题的一般步骤，并着重介绍本文提出的 NUPMX 算法。

4.1　编码

编码步骤是遗传算法解决各种问题中最基础也是最重要的一步，一个好的编码可以大幅减少求解的耗时并大幅提高结果的性能。根据第 3 节的建模，我们需要寻找满足式(1)的转移矩阵 P，然而直接对 P 使用遗传算法并不是很好的一个选择。为了方便后续更加高效地搜索，我们将 P 编码为一个长度为 N_h 的向量置换 V，其中的每个元素代表图 G 中的节点编号，因此其值均为不小于 N_g 的整数。例如，假设向量 V 中位置 i 的值为 j，记为 $V[i]=j$，可以转化为邻接矩阵 $P^{\mathrm{T}}[i,j]=1$。图 2 展示了一个转移矩阵与编码的置换向量之间的对应的关系。显然，每一个转移矩阵 P 都与一个向量 V 一一对应，两者可以互相转化，每一个编码向量 V 都代表了一个解。因此，提出的新的编码满足完整性、健全性，并且无冗余。

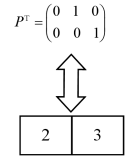

图 2　转移矩阵编码

4.2　种群生成

种群中的每一个个体都代表了一种可能的解，我们已经将每一种解编码为了一个置换向量，那么一个种群可以表示为 $Pop=[V_1, V_2, \cdots, V_n]$。我们采用随机函数从 $[1, N_g]$ 中选择 N_h 个数字构成每个置换向量，重复该过程 n 次，产生个体数量为 n 的种群。一般来说，生成的种群中个体的数量与所求解的问题规模有关，种群数量越大，计算开销越大。本文将种群数量设置为 9500，以满足一般工程应用的需求。

4.3　交叉过程

交叉是遗传算法中生成下一代个体的过程，该过程中由当代种群中的两个个体模仿基因交叉的过程，生成下一代的个体。一般来说，在交叉过程中我们希望下一代的个体能够继承上一代“基因”中优秀的部分。目前有许多优秀的交叉算法，如部分匹配交叉[17]、顺序交叉[18]、均匀部分匹配交叉[19]、循环交叉[20]、多点交叉[21]等，以及上述各算法的组合。然而这些算法并不能很好地处理子图同构问题，因为这些算法需要基因的一个全排列，而本文的置换向量只是基因的一个部分排列。为了更好地说明这一点，我们考虑 $N_h=5, N_g=8$ 的例子，如图 3 所示，图中展示了采用均匀部分匹配交叉算法(UPMX)产生的冲突。

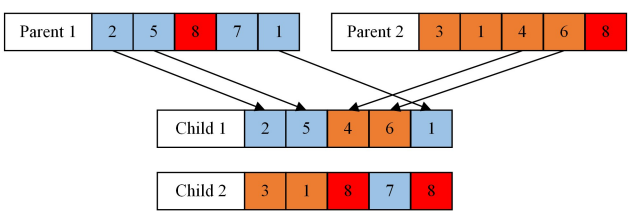

图 3　UPMX 算法的冲突

图 3 中的 Parent 1 和 Parent 2 都按照 4.2 节中的方法随机生成的，容易发现，这两个个体都是合法的。采用 UPMX 算法后，生成的两个下一代个体为 Child 1 和 Child 2。此时，Child 1 是合法的，可以转化为一个转移矩阵 P，而 Child 2 却是非法的，该向量转化为转移矩阵后，第 8 行存在两个 1，这与定义冲突。类似地可以发现，这种冲突也存在于其他一些交叉算法中。

由于这种冲突的存在，导致经典的遗传算法生成子代的效率大大降低，从而严重影响了算法的速度以及求解成功的

概率。为了解决该问题,本文提出了非均匀部分匹配交叉算法(NUPMX 交叉算法)来改进传统算法的不足之处。相关算法伪代码如算法 1 所示。

算法 1 NUPMX

```
Input parameters:
ind1:The first individual participating in the crossover
ind2:The second individual participating in the crossover
Indpb:the probability to execute the swap
1. for each indices in ind1 and ind2:
2.   if random() < indpb:# execute the swap
3.     if ind1(i) and ind2(i) both exist in ind1 and ind2:
4.       swap ind1(position(ind2(i))),ind2(position(ind1(i)))
5.       swap ind1(i) ,ind2(i)
6.     elseif ind1(i) does not exist in ind2,ind2(i) exist in ind1:
7.       ind1(position(ind2(i)))=ind1(i)
8.       swap ind1(i),ind2(i)
9.     elseif ind2(i) does not exist in ind1,ind1(i) exist in ind2:
10.      Ind2(position(ind1(i)))=ind2(i)
11.      swap ind1(i),ind2(i)
12.    elseif ind1(i) and ind2(i) both do not exist in ind1 and ind2:
13.      swap ind1(i),ind2(i)
algorithm end
Output parameters:
ind1:The first child after the crossover
ind2:The second child after the crossover
```

相比于传统算法,所提算法在每次执行交换操作时,首先对操作对象进行一次判断,根据交换的对象是否存在于对方,分为 4 种情况分别进行处理,从而避免了上述冲突的出现。

4.4 变异

变异是遗传算法中保证种群多样性的一种手段[22],从而使得算法在搜索最优解的过程中有足够高的概率跳出局部最优解。本文采用随机洗牌的方法来使个体变异[23]。

4.5 种群评估和个体选择

为了评估种群中每个个体的适应度,并引导算法收敛到最优解,我们采用式(2)所示的适应度函数。

$$fitness=Vo*abs(P^{\mathrm{T}}*G*P-H)*Vo^{\mathrm{T}} \tag{2}$$

其中,Vo 是一个 $1\times N_h$ 的一维向量,向量中每个元素均为 1。由此得到的适应度为一个自然数,代表不能匹配的节点数量。适应度函数越小,代表图 H 与图 G 的子图越相似。因此需要求解适应度函数的最小值,即:

$$\min(fitness) \tag{3}$$

如果适应度函数值为零,那么此时的个体为所要求解的一个解,由该个体生成的转移矩阵就是图 G 和图 H 的对应关系。

在遗传算法的每次迭代过程中,计算种群中每个个体的适应性来判断每个解得好坏。当所有个体经过交叉和变异步骤后,需要从新生成的 Child 以及原来的 Parent 中选择合适的个体构成下一代,学术界已有许多选择算法,如轮盘赌[24]、玻尔兹曼法[25]、顺序选择法[26]等。本文采用 K-tournaments 算法[27]选择下一代的个体。

至此,已经介绍完整个遗传算法中每个步骤,下面给出本文改进后的遗传算法流程,如图 4 所示。

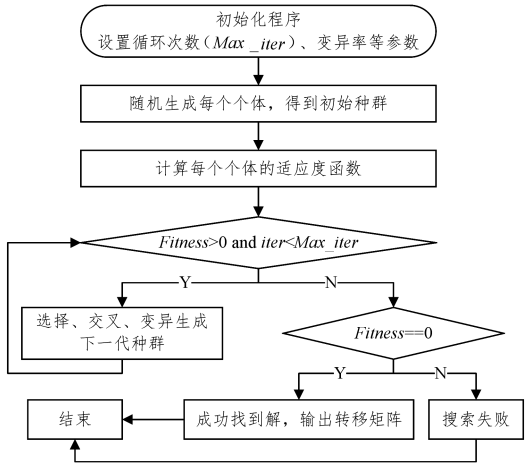

图 4 遗传算法的流程

5 仿真实验

本文首先随机生成一个图 G,然后从图中随机选择一部分节点构成一个子图 H,作为要求解的目标解,重复该过程,生成实验用的数据集合。然后分别使用 VF2 算法、遗传算法 GA * ILS[28],以及本文改进的算法 GA * NUPMX 对数据集进行求解。本文采用求解成功的比例作为算法性能的指标,即算法成功搜索到最优解的比例。实验结果如表 1 所列。

表 1 3 种算法的性能对比

	VF2	GA * ILS/%	GA * NUPMX/%
$N_h=13,N_g=18$	100	99.29	100.00
$N_h=19,N_g=24$	Inf	97.94	99.09
$N_h=25,N_g=30$	Inf	95.50	97.29

由于 VF2 算法仅能够处理比较小的图,当图的规模变大后,已经难以在短时间内求解。从求解成功率上看,本文提出的改进算法要优于 GA * ILS 算法。

下面尝试将 GA * NUPMX 应用于更大规模的子图同构问题,图 5 给出算法求解失败率与 N_g 的关系图。

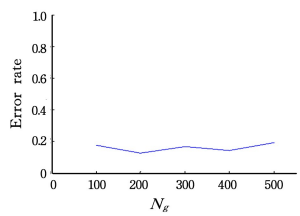

图 5 GA * NUPMX 算法性能

从图 5 可以看出,GA * NUPMX 算法在求解 500 个节点规模的子图匹配问题时具有较好的稳定性,求解失败率基本保持在 20%左右,工程应用中基本可以接受。

结束语 本文提出了改进的遗传算法 GA * NUPMX 来求解子图同构问题,使用邻接矩阵对图进行建模,并使用置换向量对邻接矩阵进行编码,将子图同构问题转化为容易计算的形式。我们设计了一个高效的适应度函数来引导种群的进化过程,以及一个特殊的基因交叉算法来高效地搜索最优解。最后通过实验验证了改进算法的有效性,并通过大量实验表明,该算法可以适用于 500 个节点的子图同构问题。

如何提高算法的运行速度仍然有待研究,如何更快地搜索到最优解也是一个很有潜力的研究方向。此外,尽管算法

目前已经能以较高的概率搜索到最优解，但是算法的性能距离最优还有一定距离。

参 考 文 献

[1] BROWN A D, THOMAS P R. Goal-oriented subgraph isomorphism technique for IC device recognition[J]. IEE Proceedings I (Solid-State and Electron Devices), 1988, 135(6): 141-150.

[2] GUHA A. Optimizing codes for concurrent fault detection in microprogrammed controllers[C]// Proc. Int. Conf. Computer Design: VLSI in Computers and Processors (ICCD'87). 1987: 486-489.

[3] LANG S Y T, WONG A K C. A sensor model registration technique for mobile robot localization[C]// Proceedings of the 1991 IEEE International Symposium on Intelligent Control. IEEE, 1991: 298-305.

[4] EPPSTEIN D. Subgraph isomorphism in planar graphs and related problems[M]// Graph Algorithms And Applications I. 2002: 283-309.

[5] ULLMANN J R. An algorithm for subgraph isomorphism[J]. Journal of the ACM (JACM), 1976, 23(1): 31-42.

[6] BOMZE I M, BUDINICH M, PARDALOS P M, et al. The maximum clique problem[M]// Handbook of Combinatorial Optimization. Springer, Boston, MA, 1999: 1-74.

[7] SAHNI S, GONZALEZ T. P-complete approximation problems [J]. Journal of the ACM (JACM), 1976, 23(3): 555-565.

[8] COOK S A. The complexity of theorem-proving procedures [C]// Proceedings of the Third Annual ACM Symposium on Theory of Computing. ACM, 1971: 151-158.

[9] SCHMIDT, DOUGLAS C, DRUFFEL L E. A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices[J]. Journal of the ACM (JACM), 1976, 23(3): 433-445.

[10] CORDELLA L P, FOGGIA P, SANSONE C, et al. A (sub) graph isomorphism algorithm for matching large graphs [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(10): 1367-1372.

[11] MCKAY B D. Practical graph isomorphism[J]. Congressus Numerantium, 1981, 30: 45-87.

[12] MCKAY B D, PIPERNO A. Practical graph isomorphism, II[J]. Journal of Symbolic Computation, 2014, 60: 94-112.

[13] KIRKPATRICK S, GELATT C D, VECCHI M P. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671-680.

[14] FOGEL D B. An introduction to simulated evolutionary optimization[J]. IEEE Transactions on Neural Networks, 1994, 5(1): 3-14.

[15] ZHONG Q, WU Z, LIN L, et al. Computing resources assignment in rtds simulators with subgraph isomorphism based on genetic algorithm[C]// 2011 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT). IEEE, 2011: 1144-1149.

[16] CVETKOVIĆ D M, DOOB M, SACHS H. Spectra of graphs: theory and application [M]. Deutsher Verlag der Wissenschaften, 1980.

[17] GOLDBERG D E, LINGLE R. Alleles, loci, and the traveling salesman problem[C]// Proceedings of an International Conference on Genetic Algorithms and Their Applications. Lawrence Erlbaum, Hillsdale, NJ, 1985, 154: 154-159.

[18] GOLDBERG D E. Genetic algorithms in search, optimization, and machine learning [M]// Genetic Algorithms in Search, Optimization, and Machine Learning. 1989.

[19] CICIRELLO V A, SMITH S F. Modeling GA performance for control parameter optimization[C]// Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation. Morgan Kaufmann Publishers Inc., 2000: 235-242.

[20] 刘荷花，崔超，陈晶. 一种改进的遗传算法求解旅行商问题[J]. 北京理工大学学报，2013，33(4)：390-393.

[21] 朱学军，陈彤，薛量，等. 多个体参与交叉的 Pareto 多目标遗传算法[J]. 电子学报，2001，29(1)：106-109.

[22] MUHLENBEIN H. How genetic algorithms really work: I. mutation and hillclimbing[C]// Proc. 2nd Int. Conf. on Parallel Problem Solving from Nature, 1992. Elsevier, 1992.

[23] WHITLEY D. A genetic algorithm tutorial [J]. Statistics and Computing, 1994, 4(2): 65-85.

[24] 梁宇宏，张欣. 对遗传算法的轮盘赌选择方式的改进[J]. 信息技术，2009(12)：127-129.

[25] GOLDBERG D E. A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing[J]. Complex Systems, 1990, 4(4): 445-460.

[26] GOLDBERG D E, DEB K. A comparative analysis of selection schemes used in genetic algorithms[M]// Foundations of Genetic Algorithms. Elsevier, 1991: 69-93.

[27] DEB K. An efficient constraint handling method for genetic algorithms[J]. Computer methods in Applied Mechanics and Engineering, 2000, 186(2-4): 311-338.

[28] FARAHANI M M, CHAHARSOUGHI S K. A genetic and iterative local search algorithm for solving subgraph isomorphism problem[C]// 2015 International Conference on Industrial Engineering and Operations Management (IEOM). IEEE, 2015: 1-6.