基于遗传算法求解数独难题

刘延风 刘三阳
(西安电子科技大学应用数学系 西安 710071)

摘 要 为了求解数独难题,首先将其转化成一个组合优化问题。然后,提出一个在编码、初始化、交叉、变异、局部搜索等方面具有特点的遗传算法来求解它。实验结果表明,对于所有难度等级的数独难题,算法都是有效的。

关键词 遗传算法,数独难题,局部搜索
中图法分类号 TP181 **文献标识码** A

Algorithm Based on Genetic Algorithm for Sudoku Puzzles

LIU Yan-feng LIU San-yang
(Department of Applied Mathematics, Xidian University, Xi'an 710071, China)

Abstract To solve the Sudoku puzzles, above all, they were changed into a combinatorial optimization problem. Then, a genetic algorithm with specialized encoding, initialization and local search operator was presented to optimize it. The experimental results show the algorithm is effective for all difficulty levels Sudoku puzzles.

Keywords Genetic algorithm, Sudoku puzzle, Local search

1 引言

数独名称来源于日语 Sudoku,可以理解为"独立的数字"。数独是一个很容易使人着迷的难题,在世界各地以及互联网上都十分流行。它的规则十分简单,在9×9的空格里填入合适的数字,使得每行(从左到右)、每列(从上到下)以及每个小方块都要包含从1到9的数字。通常,有些格子里的数字事先给定。图1就是一个数独和它的解。由于数独和有些调度问题非常类似,例如课表编排问题,因此求解数独不能完全等同于玩游戏。

		3	7	2		1		
		6	9		5	8		
4	9			1			5	2
	5						6	1
8		4				2		9
7	6						4	
2	3			5			1	7
		1	2		3	9		
		5		8	6	4		

5	8	3	7	2	4	1	9	6
1	2	6	9	3	5	8	7	4
4	9	7	6	1	8	3	5	2
3	5	9	8	4	2	7	6	1
8	1	4	5	6	7	2	3	9
7	6	2	3	9	1	5	4	8
2	3	8	4	5	9	6	1	7
6	4	1	2	7	3	9	8	5
9	7	5	1	8	6	4	2	3

图1 一个数独及其解

数独看起来是一个基于逻辑推理的问题,很自然的解法是采用基于回溯的蛮力搜索方法[1]。目前网上的许多求解器都是采用这种方法。但是,由于数独是一个NP难问题[2],这意味着利用上述方法求解很多数独问题是不现实的。文献[3]提出利用现代优化算法求解数独问题,并提出了一种基于模拟退火的求解方法。文献[4]提出了一种基于遗传算法的求解算法。该算法只是简单地应用遗传算法,加之没有局部搜索,因此效果并不理想,对于难度适中乃至困难的数独,100次独立运行中能够成功的次数不超过10次。

遗传算法是一种基于种群的随机优化算法,自从提出以来在很多应用方面取得了极大的成功。基于遗传算法求解数独问题的难点在于必须搜索到数独的解(即最优解),而找到次优解是没有任何意义的。这就对算法性能提出了很高的要求。为了求解数独难题,作者在编码、初始化种群等方面做了一些改进。另外,考虑到虽然遗传算法全局搜索能力比较强,但是局部搜索能力弱,为了取长补短,定义并添加了局部搜索。从实验结果来看,提出的算法对于所有难度等级的问题都是有效的。

2 问题的转化

数独可以看作一个组合优化问题:将每一个小方块中未出现的数字不重复地填入其空格里,目标函数是使得满足要求的行或列的数目之和最大化。显然,最优解对应的目标函数值为18,它就是数独的解。将每一个小方块中未出现的数字任意不重复地填入其空格内,就是一个可行解。

3 求解数独的局部搜索遗传算法

3.1 编码

采用符号编码,按照小方块的排列顺序(先行后列、从左到右),对每一个小方块未出现的数字进行编码,然后将各自的编码拼接而成染色体。例如,图1中第一个小方块中未出现的数字为1,2,5,7,8;第二个小方块中未出现的数字为3,4,6,8,等等。图1数独的某个可行解编码如下所示(阴影部分表示处于同一个小方块中未出现的数字)。

到稿日期:2009-04-24 返修日期:2009-07-14

刘延风(1970—),男,讲师,主要研究方向为智能优化算法及其应用,E-mail:teacher2003@tom.com;刘三阳(1959—),男,教授,主要研究方向为最优化理论与算法。

[redacted]

由于舍弃掉了那些已经出现的数字，和文献[4]的编码相比，编码长度变小了，提高了编码效率。

3.2 初始化染色体种群

最简单的方法是随机生成每一个小方块中未出现数字的排列，然后将其按照小方块的排列顺序拼接成一个染色体。但是，这样生成的初始种群质量很差(很多个体满足要求的行和列数目之和为0)，直接导致算法无法正常运行。

为了得到质量比较好的初始种群，采用基于规则的方式生成初始种群。具体方法如下：对于每一个小方块，首先随机生成一个未出现数字的排列。然后，对这个小方块的每一个空格，从排列中按照从前到后的顺序，依据下面3条规则依次找出所要填充的数字，并在排列中删除该数字。规则1：如果在排列中能找到与空格所在的行与列上数字都不重复的数字，则将其填入该空格。规则2：如果找不到满足规则1的数字，但是能找到与空格所在的行或列上的数字不重复的数字，则将其填入该空格。规则3：如果满足规则1和规则2的数字都找不到，则将排列中第一个数字填入该空格。最后，将每个小方块得到的结果按照小方块的排列顺序拼接成一个染色体。

例如，图1数独的一个初始染色体为：

[redacted]

3.3 选择

适应度定义为满足要求的行或列的数目之和。选择机制采用依据适应度的比例概率，适应度较高的个体有较大的机会繁殖后代。为了兼顾全局搜索性能和收敛速度，在下一代种群构成上，采用最佳个体保留方式。

3.4 交叉

首先对染色体群体做随机配对。然后，随机产生两个交叉位置，两个交叉位置之间的所有位置为交叉区域。下面是具体的实现方法。为了叙述方便，略去了染色体中与交叉操作无关的其他数字。

(1)如果交叉位置都在一个小方块对应的染色体中，则采用部分匹配交叉[5]操作。

例如，对于

P1=6 5 8 |4 7 9|

P2=6 7 5 |8 4 9|

首先，交换P1和P2中两个交叉位置之间的数字，得到P12和P22。

P12=6 5 8 |8 4 9|

P22=6 7 5 |4 7 9|

对于P12和P22中出现的重复数字，依据交叉位置内的位置映射关系，逐一进行交换。由于P12中有8到4，4到7的位置数字映射，对P12中交叉区域以外的8用7替换，则得到C1。同理可得C2。

C1=6 5 7 |8 4 9|

C2=6 8 5 |4 7 9|

(2)如果交叉位置不在同一个小方块对应的染色体中，如下所示(阴影数字分别为第一个和第二个交叉位置所在小方块对应的染色体)：

P1=[redacted] 2 7 6 1 3 5 3 9 8 7 2 1 5 6 7 5 8 4 9

[redacted]

P2=[redacted] 7 2 1 3 5 6 1 9 5 2 8 3 7 6 5 7 9 4 8

[redacted]

则在第一个交叉位置所在小方块的染色体中，以第一个交叉位置和这个小方块的最后一个位置作为新的交叉位置；在第二个位置所在的小方块对应的染色体中，以其开始位置和第二个交叉位置作为新的交叉位置，如下所示：

P12=[redacted] 2 7 6 1 3 5 3 9 8 7 2 1 5 6 7 5 8 4

9 [redacted]

P22=[redacted] 7 2 1 3 5 6 1 9 5 2 8 3 7 6 5 7 9 4

8 [redacted]

分别做部分匹配交叉操作，得到：

P13=[redacted] 2 7 6 1 3 5 3 9 8 7 2 1 5 6 7 5 8 4

9 [redacted]

P23=[redacted] 7 2 1 3 5 6 1 9 5 2 8 3 7 6 5 7 9 4

8 [redacted]

最后，对于处在第一个交叉位置所在的小方块与第二个交叉位置所在的小方块之间的小方块对应的染色体，直接做交换操作，得到的交叉结果为：

C1 =[redacted] 7 2 1 3 5 6 1 9 5 2 8 3 7 6 5 7 9 4

8 [redacted]

C2 =[redacted] 2 7 6 1 3 5 3 9 8 7 2 1 5 6 7 5 8 4

9 [redacted]

3.5 变异

按照变异概率计算出参加变异的染色体个数。变异的具体实现过程如下：

首先，随机确定哪一个小方块参加变异。然后，在这个小方块中随机确定两个变异位置，例如P=2 [redacted] 6 1 [redacted]，最后将这两个位置上对应的数字交换，得到结果M=2 3 6 1 7。

3.6 局部搜索

为了提高算法的性能，引入了局部搜索：对每一个小方块对应的染色体中的数字分别做两两交换。实现方式有两种：第一种，以遗传算法得到的解作为初始解，做局部搜索，得到局部最优解；第二种，开始时将遗传算法得到的解作为初始解，做完局部搜索后，将得到的局部最优解作为初始解继续做局部搜索，直到找不出更好的局部最优解为止。

对于事先没有给定任何数字的数独，采用第一种局部搜索方法就能够找到最优解。对于难度等级分别为easy，medium，hard，evil的数独，必须采用第二种局部搜索方法才能找到最优解。

4 实验结果

为了验证算法的有效性，采用http://www.websudoku.com上提供的难度等级分别为easy，medium，hard，evil编号为1的实例以及一个全空的数独new进行测试，每个实例测试100次。实验硬件环境的处理器主频为1.70GHz，内存504M，操作系统为Windows XP，算法用VC++6.0编程实现。

(下转第233页)

③④的标注效果最差。但就对应样本和全部样本上的综合指标来看，SVM类别标注的整体性能还是可以接受的。

3.3 对应关系基础类型的类别标注

基于支持向量机的分类方法在以上类别标注任务中已经取得了较好的效果，但还不能认为相应基础类型必然包含标注类别的对应方式。原因有两个：①句对的类别标注结果还不是100%正确；②句对的次范畴化对应关系基础类型是通过基于规则的方法得到的，必然由此带来更大程度的不确定性。所以，针对英汉动词次范畴化对应关系基础类型的类别标注问题，采用较高阈值 $\theta = 0.3$ 的最大似然假设检验的方法来保证标注类别的统计可靠性。具体算法如下：

输入：某一英汉动词次范畴化对应关系基础类型的全部支持句对集合 S；

输出：该基础类型的标注类别 X；

操作：

1) 应用SVM自动标注每一个句对的对应关系类别；

2) 从 S 中删除那些标注为“非对应”的句对，得到 S'；

3) 在 S' 上对每一个出现的标注类别 C_i 统计相对频率 f_{Ci} = [C_i 的出现次数]/$|S'|$；

4) 取 $X = \bigcup_{f_{Ci} \geqslant \theta} f_{Ci}$ 为当前基础类型的标注类别。

若某一次范畴化对应关系基础类型的支持句对为 m 个，且其中 n 个句对被SVM标注为“非对应”类别，根据统计学习理论，以上算法标注该类型为某一类别的统计可靠程度将不低于 $1-(1-84.16\%)^{0.3*(m-n)} = 1-(0.1584)^{0.3*(m-n)}$。当 $m-n \geqslant 10$ 时，类别标注的统计可靠程度将接近于1。

结束语 本文基于大规模语料对汉英动词次范畴化对应类型进行了统计分析。首先自动识别出那些可能包含跨语言次范畴化关系的句子对，然后通过启发式方法和双重过滤的假设检验方法初步估计了654种汉英次范畴化对应类型的概率分布，最后在语言学分类的基础上对每一种对应类型和背景语料进行了基于支持向量机的类别标注和统计可靠性分析。

此外，本文研究的基本定义、获取方法和实验规模等方面，还都有待于进一步调整、扩大和改进。并且，英文句式转换信息必然会提高双语次范畴化的分析性能；关于特定谓词对的跨语言次范畴化统计信息尚有待研究；应用动词的语义分类信息也可能发现更多的双语次范畴化对应类型。

参 考 文 献

[1] Korhonen A. Subcategorization Acquisition[D]. Trinity Hall University of Cambridge, 2001

[2] Han Xiwu, Zhao Tiejun, Qi Haoliang, et al. Subcategorization Acquisition and Evaluation for Chinese Verbs[A]//Proceedings of the COLING 2004[C]. 2004: 723-728

[3] 韩习武. 汉语动词次范畴化自动获取技术的研究[D]. 哈尔滨：哈尔滨工业大学，2005

[4] Collins M. Head-Driven Statistical Models for Natural Language Parsing[D]. University of Pennsylvania, 1999

[5] 曹海龙. 基于词汇化统计模型的汉语句法分析研究[D]. 哈尔滨：哈尔滨工业大学，2006

[6] Chang Chih-Chung, Lin Chih-Jen. LIBSVM: a library for support vector machines[EB/OL]. http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001

(上接第226页)

参数设置为：种群规模60，交叉概率1.0，变异概率0.1。程序的终止条件为找到最优解或迭代次数达到设定的值(对于new，迭代次数设为1000，其他设为200)。

从表1可以看到，对于全空的数独new以及难度等级为easy，medium的数独，算法在100次运算中都求出了最优解。对于难度等级为hard和evil的数独，算法虽然不能做到100%求出最优解，但是100次独立运行中求出了最优解的次数超过70次。测试结果表明了算法的有效性。

表1 算法测试结果

难度等级	找到最优解的次数	最大迭代次数/时间(秒)	最小迭代次数/时间(秒)	平均迭代次数/时间(秒)
easy	100	20/7	9/3	13.6/4.7
medium	100	39/14	17/7	27.3/10.4
hard	90	43/19	28/13	33/15.3
evil	70	68/27	30/15	40.9/18.9
new	100	452/2	177/1	255.6/1.4

为了考察局部搜索在算法中的作用，程序的终止条件应设为找到最优解或迭代次数达到20000，屏蔽局部搜索，其余设置不变，得到的测试结果如表2所列。

表2 不含局部搜索的算法测试结果

难度等级	得到最优解的次数	最大迭代次数/时间(秒)	最小迭代次数/时间(秒)	平均迭代次数/时间(秒)
easy	60	19257/31	1480/3	11619/18
medium	5	5174/8	——	5174/8
hard	0	——	——	——
evil	0	——	——	——
new	80	13663/18	2190/3	6268/9.4

从表2可见，虽然迭代次数扩大了100倍，但是在100次测试中找到最优解的次数却明显地减少了，尤其是对于难度等级为hard以及evil的数独，最优解一次也没有找到。这说明局部搜索不仅使得算法可以在迭代次数较小时找到最优解，更重要的是提高了找到最优解的几率。

结束语 为了求解数独难题，首先将其转化为一个组合优化问题。然后，提出了一种在编码、初始化、交叉、变异、局部搜索等方面具有特点的遗传算法来求解它。实验结果表明，对于各种难度等级的数独问题，算法都是有效的。

参 考 文 献

[1] 孟庆铃. 数独问题人工解法的程序实现[J]. 甘肃科技，2006(9)：150-151

[2] Yato T, Seta T. Complexity and Completeness of Finfing Solution and Its Application to Puzzles[J]. IEICE Trans. Fundamentals, 2003, E86-A(5): 1052-1060

[3] Lewis R. Metaheuristics can solve sudoku puzzles[J]. Journal of Heuristics, 2007, 13: 387-401

[4] Mantere T, Koljonen J. Solving and rating sudoku puzzles with Genetic Algorithm[C]//Proceedings of the 12th Finish Artificial Intelligence Conference. Sept. 2006

[5] Goldberg DE, Alleles LR. The Traveling Salesman Problem[C]//Proceedings of an International Conference on Genetic Algorithms and Their Applications. 1985: 154-159