文章编号:1674-7070(2011)01-0023-05

# 一种基于稀疏优化的数独求解新方法

张煜东[1]　王水花[1]　霍元恺[1]　吴乐南[1]

**摘要**

为了更好地求解数独问题,提出一种新的求解方法:采用实数编码去除整数约束,同时采用0范数作为目标函数来保证解的稀疏性.在此基础上,根据RIP(Restricted Isometry Property)与KGG(Kashin Garnaev Gluskin)条件,用1范数近似0范数.最后引入松弛矢量,使1范数转换为一个凸线性规划问题.采用主对偶内点法求解该线性规划问题.实验表明:该方法对简单、中等、困难、恶魔级别的数独,可达到100%成功率;对最小提示数目的17数独,达到86.4%的成功率.另外,该算法耗时短,且与数独的难度无关.因此,该算法在成功率与运行时间上均优于约束规划与Sinkhorn算法.

**关键词**

数独;约束规划;整数规划;线性规划;主对偶内点法

**中图分类号** O29
**文献标志码** A

**收稿日期** 2010-07-20
**资助项目** 国家自然科学基金(60872075)

**作者简介**

张煜东,男,博士后,研究方向为数据挖掘. zhangyudongnuaa@gmail.com

1 东南大学 信息科学与工程学院,南京,210044

## 0 引言

数独(Sudoku)是一种数字游戏,在9×9格的大九宫格中有9个3×3格的小九宫格,并提供一定数量的数字[1],根据这些数字,利用逻辑和推理,在其他空格上填入1~9的数字,每个数字在每个小九宫格内只能出现一次,每个数字在每行、每列也只能出现一次.数独问题在国外尤其是美国研究非常热,每年在期刊上大约有200多篇论文发表,每天的纽约时报上都会附带一个数独问题,在下一期提供答案.数独有$9!\times72^2\times2^7\times27\ 704\ 267\ 971=6\ 670\ 903\ 752\ 021\ 072\ 936\ 960$个组合,2005年学者们利用穷举法計算出,如果不考虑重复(如数字转换、反射面等),则有5 472 730 538个组合.数独的难度根据提示数目($N_C$)分为4个级别,如果$32\leqslant N_C\leqslant34$,称为"简单";$29\leqslant N_C\leqslant31$,称为"中等";$26\leqslant N_C\leqslant28$,称为"困难";$23\leqslant N_C\leqslant25$,称为"恶魔"[2].

最原始的求解方法是暴力破解(Brute-Force)法,缺陷是需要大量的时间与内存;约束规划(Constraint Programming)[3]通过设置一个目标函数与约束函数,采用分支定界法求解,由于它本质上是一个NP-C问题,所以耗时较多,且结果对初值选取敏感[4].一种改进方法是采用随机搜索或者智能优化[5],首先随机生成解,然后计算误差,通过不断迭代使得误差为零,这种方法比前两者稍快,但同样需要大量时间,且性能不稳定.

本文提出一种新的求解思路,将Sudoku问题转化为一个1范数的优化问题,然后通过引入松弛变量,转化为一个线性规划(Iinear Programming,LP)问题.由于线性规划问题是经典的优化问题,存在很多优异算法,所以可迅速求解[6].

## 1 背景介绍

图1给出了一个"简单"的数独.假设解为$\boldsymbol{x}$,则一种简化的约束规划形式为

$$\min 1$$
$$\text{s.t.}\begin{cases} row_i(\boldsymbol{x})=\{1,2,\cdots,9\}, \\ col_i(\boldsymbol{x})=\{1,2,\cdots,9\}, \\ box_i(\boldsymbol{x})=\{1,2,\cdots,9\}, \\ Clue_i(\boldsymbol{x})=Clue_i, \\ x_{ij}\in\{1,2,\cdots,9\}. \end{cases}\tag{1}$$

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

图 1　一个简单的数独例子

Fig. 1　A simple example of sudoku

式(1)目标函数始终为 1，约束函数分别表示每行、每列、每个小九宫格仅出现一次 1 ~ 9，且 $\boldsymbol{x}$ 满足预定的提示，须为 1 ~ 9 的整数. 这 5 个约束条件存在冗余，但为叙述方便，此处保留.

## 2　模型

### 2.1　实数编码

由式(1)可见，该约束规划同时也是一个整数规划，给求解带来难度. 采用一种新的编码策略，将 $\boldsymbol{x}$ 放宽到实数域，去除整数约束.

图 2 显示了实数编码的策略，将原始的整数编码中每一点 $x_{ij}$ 用 9 个 0 ~ 1 之间的实数 $x_{ijk}(k=1,2,\cdots,9)$ 表示，两者的转换关系为

$$x_{ijk} = I(x_{ij} == k), \tag{2}$$

$$x_{ij} = k^* \left( x_{ijk^*} = \max_k (x_{ijk}) \right). \tag{3}$$

式中 $I(\cdot)$ 是指示函数，等式成立输出 1，等式不成立输出 0.

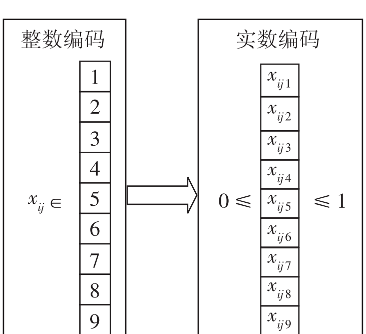



图 2　实数编码策略

Fig. 2　Schematic chart of real-number encoding

### 2.2　约束条件

采用实数编码去除了整数约束，同时约束问题转化为一个线性方程组，使搜索更快. 假设 $\boldsymbol{x}$ 采用按行拉直方法，则 $\boldsymbol{x}$ 为一个 $1 \times 729$ 的行向量，行约束条件(以第 1 行为例)可写为

$$\left[ \underbrace{I_{9\times9} I_{9\times9} \cdots I_{9\times9}}_{9} 0_{9\times648} \right] \boldsymbol{x} = 1_{9\times729}; \tag{4}$$

列约束条件(以第 1 列为例)可写为

$$\left[ \underbrace{I_{9\times9} 0_{9\times72} \cdots I_{9\times9} 0_{9\times72}}_{9} \right] \boldsymbol{x} = 1_{9\times729}; \tag{5}$$

小九宫格约束(以左上角为例)可写为

$$[ I_{9\times9} I_{9\times9} I_{9\times9} 0_{9\times54} I_{9\times9} I_{9\times9} I_{9\times9} 0_{9\times54} I_{9\times9} I_{9\times9} I_{9\times9} 0_{9\times540} ] \times \boldsymbol{x} = 1_{9\times729}; \tag{6}$$

提示约束(以第 1 格子提示为 5 为例)可写为

$$\left[ 0\ 0\ 0\ 0\ 1\ \underbrace{0 \cdots 0}_{724} \right] \boldsymbol{x} = 1; \tag{7}$$

填充约束(以第 1 格子必须填充 1 ~ 9 之间的一个数为例)可写为

$$\left[ \underbrace{1 \cdots 1}_{9} \quad \underbrace{0 \cdots 0}_{720} \right] \boldsymbol{x} = 1. \tag{8}$$

显然，所有的约束条件转换为一个线性方程组，简记为

$$\begin{aligned} &\min 1 \\ &\text{s.t.}\quad \boldsymbol{A}_{\text{eq}} \boldsymbol{x} = \boldsymbol{b}_{\text{eq}}. \end{aligned} \tag{9}$$

需要注意的是，约束条件中让 $\boldsymbol{x}$ 可以在实数范围内取值，没有规定二进约束，即“$\boldsymbol{x}$ is binary”. 这样做的优点是保证约束是线性的，缺点是方程组的元素为 $9 \times 9 \times 9 = 729$，方程组的个数为 $81 + 81 + 81 + 81 + N_{\mathrm{C}} = 324 + N_{\mathrm{C}}$，显然是一个欠定方程组，因此根据式(9)求得的解 $\boldsymbol{x}$ 不一定满足二进形式，某个元素的误差传递到其它元素，最终得到错误结果.

### 2.3　稀疏优化

利用稀疏优化的概念可将二进约束条件“$\boldsymbol{x}$ is binary”转化为形如[7]：

$$\begin{aligned} &\min \| \boldsymbol{x} \|_0 \\ &\text{s.t.}\quad \boldsymbol{A}_{\text{eq}} \boldsymbol{x} = \boldsymbol{b}_{\text{eq}}. \end{aligned} \tag{10}$$

因为 $\boldsymbol{x}$ 的 0 范数为 $\boldsymbol{x}$ 不为零的元素个数，显然当 $\| \boldsymbol{x} \|_0$ 最小时，$\boldsymbol{x}$ 一定是二进制形式. 此时的解等价于增加了约束条件“$\boldsymbol{x}$ is binary”. 表 1 给出了 $x_{ij} = 1$ 的 4 种实数编码形式，当 $x_{ij}$ 为二进编码时，其 0 范数最小.

### 2.4　$L_1$ 范数近似

式(10)中的 0 范数计算较为复杂，在某些情况下，0 范数与 1 范数等价. 根据 RIP(Restricted Isometry Property)[8] 与 KGG(Kashin Garnaev Gluskin)不等式[9]，以及大量的实际数独实例，可将式(10)近似改写为

$$\begin{aligned} &\min \| \boldsymbol{x} \|_1 \\ &\text{s.t.}\quad \boldsymbol{A}_{\text{eq}} \boldsymbol{x} = \boldsymbol{b}_{\text{eq}}. \end{aligned} \tag{11}$$

**表 1　稀疏优化示意**

Table 1　Diagram of sparse optimization

| $x_{ij}$ | $\Vert x_{ij} \Vert_0$ |
|---|---|
| [1 0 0 0 0 0 0 0 0] | 1 |
| [0.6　0.4　0 0 0 0 0 0 0] | 2 |
| [0.6　0.3　0.1　0 0 0 0 0 0] | 3 |
| [0.6　0.2　0.1　0.1　0 0 0 0 0] | 4 |

## 2.5　线性规划

$\boldsymbol{x}$ 的 1 范数为 $\boldsymbol{x}$ 的元素的绝对值的和,计算仍较复杂[10].引进 2 个大小均为 $1\times729$ 的非负松弛矢量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$,使得 $\boldsymbol{x}=\boldsymbol{u}-\boldsymbol{v}$,则式(11)改写为

$$\min(\boldsymbol{u}+\boldsymbol{v})$$
$$\text{s.t.}\begin{cases}\boldsymbol{A}_{\mathrm{eq}}\boldsymbol{x}=\boldsymbol{b}_{\mathrm{eq}},\\ \boldsymbol{x}-\boldsymbol{u}+\boldsymbol{v}=0,\\ \boldsymbol{u},\boldsymbol{v}\geqslant 0.\end{cases}\tag{12}$$

显然 $\Vert\boldsymbol{x}\Vert_1\leqslant\boldsymbol{u}+\boldsymbol{v}$,由夹逼定理,当收敛到最小时,$\Vert\boldsymbol{x}\Vert_1=\boldsymbol{u}+\boldsymbol{v}$.因此最小化 $\boldsymbol{u}+\boldsymbol{v}$ 等价于最小化 $\Vert\boldsymbol{x}\Vert_1$.再次观察式(12),此时目标函数与约束均为凸的、线性的,这是一个标准的凸线性规划问题.

# 3　主对偶内点法

目前求解线性规划问题有如下 4 种方法:内点法、单纯形法、主动集法、信赖域法.由于约束条件中同时存在等式约束与边界约束,所以信赖域法不可用.另外,内点法在众多测试中表现良好[11],所以本文采用内点法.为了进一步提高性能,选择主对偶内点法[10]求解.主对偶内点法的步骤可简述如下:

1)将主问题转化为对偶问题;

2)利用拉格朗日函数法,包含主问题与对偶问题,得到一个无约束优化问题;

3)对无约束优化问题求解,采用 KKT(Karush-Kuhn-Tucker)条件,形成牛顿-拉夫逊迭代方程;

4)求解方程,得到主问题的解.

# 4　实验

采用主频 3 GHz、内存 2 GB 的 IBM P4 机,操作系统为 Windows XP,软件采用 Matlab 2010a,并借助 Optimization Toolbox.实验从 www.websudoku.com 下载简单、中等、困难、恶魔 4 个级别的数独问题各 500 例.对比算法采用约束规划法[3]与 2009 年发表在 IEEE 信息论会刊上的 Sinkhorn 法[12],计算各种算法的成功率与平均时间,列于表 2.

**表 2　各种算法性能比较(运行 500 例)**

Table 2　Performance comparison of different algorithms (500 samples)

| 难度 | 约束规划法[3] | | Sinkhorn 法[12] | | 本文算法 | |
|---|---|---|---|---|---|---|
| | 成功次数 | 时间/s | 成功次数 | 时间/s | 成功次数 | 时间/s |
| 简单 | 489 | 15.34 | 500 | 2.32 | 500 | 0.12 |
| 中等 | 423 | 52.80 | 500 | 4.61 | 500 | 0.14 |
| 困难 | 439 | 55.79 | 500 | 6.77 | 500 | 0.13 |
| 恶魔 | 359 | 59.32 | 498 | 10.05 | 500 | 0.12 |

本文算法不仅成功率最高,且运算时间最短,这归功于其只需求解一个 LP 问题.另外,本文算法运算时间几乎与数独难度无关,而其余算法则随着难度的上升,耗时明显升高.从各个难度中选择一个典型例子示于图 3,其中黑色粗体表示提示,红色正体表示求得的解(下同).

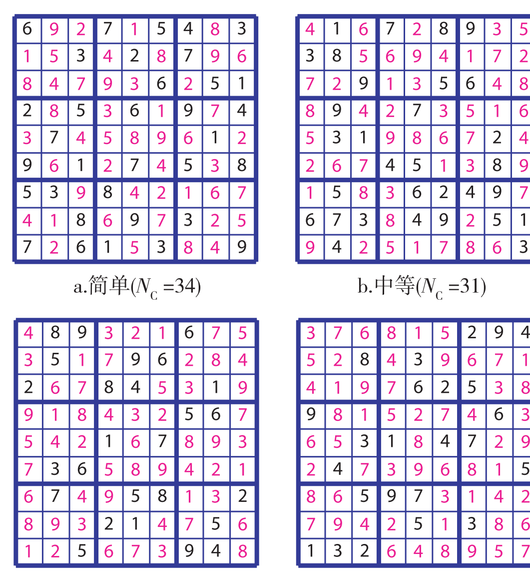


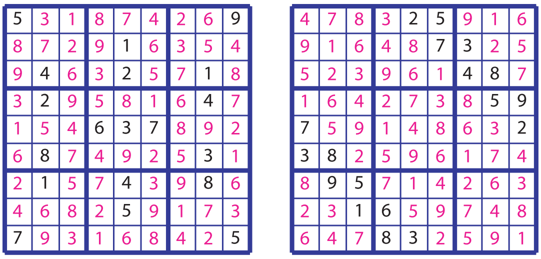

图 3　本文算法求解结果

Fig. 3　Results of the proposed algorithm

# 5　推广

提示数目小于 20 的数独不仅稀少,而且求解异常困难,采用本文算法对经典的 19-Clue 与 18-Clue 数独问题进行测试,结果如图 4 所示.

数独的最小 $N_C$ 是 17,更小就不能保证单一解.从网站 http://mapleta.maths.uwa.edu.au 下载1 000 个 17-Clue 数独,采用本文方法求解.成功率为 86.4%,平均计算时间为 0.187 8 s,计算时间的直方图示于图 5.



a. $N_C$ =19　　　　b. $N_C$ =18

图 4　对提示数目小于 20 的数独求解

Fig. 4　Sudoku solutions of number clue less than 20

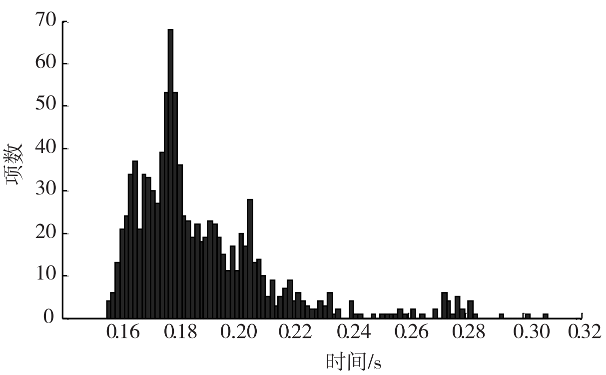
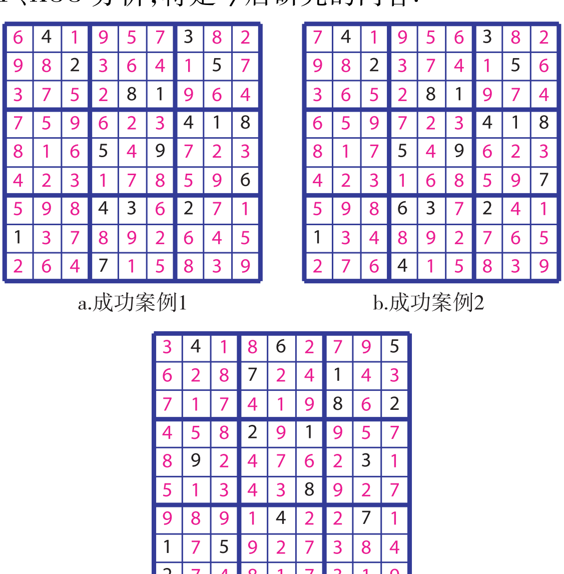



图 5　1 000 个 17-Clue 数独运行时间直方图

Fig. 5　Histogram of computation time of 1 000 17-clue sudokus

对 17-Clue 数独，挑选两个典型成功案例及一个失败案例，示于图 6. 图 6c 之所以失败，是因为不满足 RIP 与 KGG 条件. 对给定的数独，如何进行快速 RIP、KGG 分析，将是今后研究的内容.



a.成功案例1　　b.成功案例2

c.失败案例

图 6　17-Clue 成功与失败案例

Fig. 6　Success and fail cases of 17-clue Sudokus

## 6　结论

本文提出一种新的基于稀疏优化的数独求解策略，将一个约束规划问题转化为凸线性规划问题. 对简单、中等、困难、恶魔级别的数独，本文算法达到 100% 成功率. 对理论上最严格最苛刻的 17-Clue，也有 86.4% 的成功率. 更重要的是，该算法耗时极短，且耗时与数独难度无关. 未来可尝试将该算法与其他算法混合，使其既保证 100% 的成功率，又能大幅缩短时间.

不少工程问题可以用下列模型表示：约束规划、整数规划、或 0 范数优化[13-14]. 本文的数学意义在于，对这些难以求解的模型，提出一种基于稀疏优化的变换策略，将其转化为容易求解的凸线性规划问题.

## 参考文献
References

[1] Pang S C, Li E Y, Song T, et al. Rating and generating sudoku puzzles[C]// Proceedings of the 2010 Second International Workshop on Education Technology and Computer Science, 2010: 457-460

[2] Sato Y, Inoue H. Solving sudoku with genetic operations that preserve building blocks[C]// Proceedings of the 2010 IEEE Symposium on Computational Intelligence and Games (CIG), 2010

[3] Crawford B, Aranda M, Castro C, et al. Using constraint programming to solve sudoku puzzles[C]// Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology, 2008: 926-931

[4] 莫惠栋，许如根. 田间试验的一种新设计：数独方[J]. 作物学报，2008，34(9): 1489-1493
MO Huidong, XU Rugen. Sudoku square: A new design in field experiment[J]. Acta Agronomica sinica, 2008, 34(9): 1489-1493

[5] Mantere T, Koljonen J. solving, rating and generating sudoku puzzles with GA[C]// Proceedings of the IEEE Congress on Evolutionary Computation, 2007: 1382-1389

[6] Babu P, Pelckmans K, Stoica P, et al. Linear systems, sparse solutions, and sudoku[J]. IEEE Signal Processing Letters, 2010, 17(1): 40-42

[7] Saab R, Yilmaz Ö. Sparse recovery by non-convex optimization-instance optimality[J]. Applied and Computational Harmonic Analysis, 2010, 29(1): 30-48

[8] Needell D, Tropp J A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples[J]. Applied and Computational Harmonic Analysis, 2009, 26(3): 301-321

[9] Ganguli S, Sompolinsky H. Statistical mechanics of compressed sensing[J]. Physical Review Letters, 2010, 104(18): 188701

[10] Edlund K, Sokoler L E, Jrgensen J B. A primal-dual inte-

rior-point linear programming algorithm for MPC[C]// Proceedings of the 48th IEEE Conference on Decision and Control, Jointly with the 2009 28th Chinese Control Conference,2009:351-356

[11] Kwon O M, Park J H. Delay-dependent stability for uncertain cellular neural networks with discrete and distribute time-varying delays[J]. Journal of the Franklin Institute,2008,345(7):766-778

[12] Moon T K, Gunther J H, Kupin J J. Sinkhorn solves sudoku[J]. IEEE Transactions on Information Theory,2009, 55(4):1741-1746

[13] 张煜东,吴乐南,王水花,等. 一种基于神经网络的遥感图像压缩编码[J]. 南京信息工程大学学报:自然科学版,2009,1(1):82-88
ZHANG Yudong, WU Lenan, WANG Shuihua, et al. A neural network based compression coding for remote sensing images[J]. Journal of Nanjing University of Information Science & Technology: Natural Science Edition,2009,1(1):82-88

[14] 张煜东,吴乐南,王水花,等. Li-Hopfield 神经网络用于汉字字符识别[J]. 南京信息工程大学学报:自然科学版,2010,2(1):6-12
ZHANG Yudong, WU Lenan, WANG Shuihua, et al. Li-Hopfield neural network used for Chinese character recognition[J]. Journal of Nanjing University of Information Science & Technology: Natural Science Edition, 2010, 2(1):6-12

# A novel sudoku solving method based on sparse optimization

ZHANG Yudong[1] WANG Shuihua[1] HUO Yuankai[1] WU Lenan[1]

1 School of Information Science & Engineering, Southeast University, Nanjing 210096

**Abstract** In order to solve the sudoku more efficiently, a novel approach was proposed. We employed the real-number coding to get rid of the integer constraint, meanwhile used the $L_0$-norm to guarantee the sparsity of the solution. Moreover, the $L_1$-norm was used to approximate the $L_0$-norm on the basis of RIP and KGG condition. Finally, the slack vectors were introduced to transfer the $L_1$-norm into a convex linear programming problem, which was solved by the primal-dual interior point method. Experiments demonstrate that this algorithm reach 100% success rate on easy, medium, difficult, and evil levels, and reach 86.4% success rate on only 17-clue sudokus. Besides, the average computation time is quite short, and has nothing to do with the difficulty of sudoku itself. In all, this algorithm is superior to both constraint programming and Sinkhorn algorithm in terms of success rate and computation time.

**Key words** sudoku; constraint programming; integer programming; linear programming; primal-dual interior point method