

一个约束可满足性问题的演化算法求解^{*}

李景治 康立山 方 宁

(武汉大学软件工程国家重点实验室 武汉430072)

摘 要 约束可满足性问题是—大类常出现于现实应用中的复杂问题^[1],因其繁多的约束条件而出名。本文针对一个经典的约束可满足性问题——斑马属谁问题,基于演化算法的框架进行求解。我们采用矩阵的表示方式,并设计了相应的杂交和变异算子。实验表明,演化算法能高效地解决该问题。

关键词 约束可满足性问题,演化算法,斑马属谁问题

An Approach to a Constraint Satisfaction Problem Using Genetic Algorithm

LI Jing-Zhi KANG Li-Shan FANG Ning

(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)

Abstract Constraint satisfaction problems are a large category of complex problems in real world applications, noted for various constraints. Based on the framework of evolutionary algorithm (EA), we solve the zebra-belong-to-who problem, a classical constraint satisfaction problem in this paper. We devise the matrix encoding and its corresponding crossover and mutation operators. Experiments show that EA can solve the problem effectively.

Keywords Constraint satisfaction problem, Evolutionary algorithm, Zebra-belong-to-who problem

1. 引言

现实世界经常存在约束可满足性问题(constraints satisfaction problems, CSP),所谓约束可满足性问题就是问题的解为众多复杂的约束条件所限制。换句话说,求解约束可满足性问题时,最重要的是必须满足问题的所有约束而不是进行优化。“斑马属谁”问题就是一个著名的可约束性满足问题。

2. 问题的描述

所谓斑马属谁问题^[1],就是在五个颜色各异房子中,分别居住着五个不同国籍的人,他们饲养五种不同的宠物、喜欢喝五种不同的饮料以及拥有的汽车也各不相同,已知下面的信息:① 英国人住在红房子里。② 西班牙人养狗。③ 居住在绿房子里的人喝可乐。④ 乌克兰人喝蛋酒。⑤ 绿房子是象牙色房子的右邻。⑥ 拥有老爷车的人养蜗牛。⑦ 拥有福特汽车的人住在黄房子里。⑧ 住在中间房子里的人喝牛奶。⑨ 挪威人住在最左边的房子里。⑩ 拥有雪佛莱汽车的人与养狐狸的人是邻居。⑪ 拥有福特汽车的人与养马的人是邻居。⑫ 拥有奔驰汽车的人爱喝桔汁。⑬ 日本人开大众汽车。⑭ 挪威人的邻居居住在蓝房子里。

问题是,谁养斑马?谁爱喝矿泉水?

这是一个经典的约束可满足性问题,问题的难度体现在两点:1)问题的搜索空间巨大——所有解的组合总共有 5^5 个之多,要从这么多的解中找出一个可行解,若用穷举的方法,人脑是不可能的,就算用最快的计算机,对于这么一个天文数字,运算时间也将难以接受;2)问题的约束条件多——这么多约束条件,初听起来,很容易使人如坠云雾之中,在求可行解

的过程中,往往顾此失彼。

3. 问题的数学描述

这个问题共有25个变量:

a_i — i 号房子的颜色; b_i —住 i 号房子的人喝的饮料; c_i —住 i 号房子的人的国籍; d_i —住 i 号房子的人拥有的汽车; e_i —住 i 号房子的人饲养的宠物。

每个变量都分别对应于一个值域:

$dom(a_i) = \{ \text{黄色}(Y), \text{蓝色}(B), \text{红色}(R), \text{象牙色}(I), \text{绿色}(G) \}$,

$dom(b_i) = \{ \text{矿泉水}(W), \text{蛋酒}(E), \text{牛奶}(M), \text{桔汁}(O), \text{可乐}(C) \}$,

$dom(c_i) = \{ \text{挪威人}(N), \text{乌克兰人}(U), \text{英国人}(E), \text{西班牙人}(S), \text{日本人}(J) \}$,

$dom(d_i) = \{ \text{福特}(F), \text{雪佛莱}(C), \text{老爷车}(O), \text{奔驰}(M), \text{大众}(V) \}$,

$dom(e_i) = \{ \text{狐狸}(F), \text{马}(H), \text{蜗牛}(S), \text{狗}(D), \text{斑马}(Z) \}$ 。

我们将已经了解的内容表示为一张表(表1),在表中列出每个变量的可能取值。

表1

房子	1	2	3	4	5
颜色	{Y}	{B}	{R,I}	{I,G}	{R,G}
饮料	{W}	{O,E}	{M}	{C,O,E}	{C,O,E}
国籍	{N}	{U,J}	{E,S,J}	{S,U,J}	{E,S,U,J}
汽车	{F}	{M,V,C}	{O,V,C}	{M,O,V,C}	{M,O,V,C}
宠物	{F,Z}	{H}	{D,S,F,Z}	{D,S,F,Z}	{D,S,F,Z}

^{*} 本课题得到国家自然科学基金(编号:60073043,70071042,60133010)资助。李景治 硕士研究生,主要研究方向为计算机科学理论、智能计算、演化计算。康立山 教授,博士生导师,主要研究方向为计算机科学理论、演化计算、并行计算。方 宁 硕士研究生,主要研究方向为演化计算、神经网络、图像处理。

4. 求解斑马属谁问题的演化算法

演化算法^[2~4]是通过模拟大自然中生物的进化过程,应用于计算机程序中的随机搜索过程,下面将基于演化算法的一般框架进行设计以求解斑马属谁问题。

4.1 解的编码方式

首先,对所有变量的可能取值进行编码映射,见表2。

表2

类别 编码	人	房子	动物	饮料	车
1	挪威人(N)	黄色(Y)	狐狸(F)	矿泉水(W)	福特(F)
2	乌克兰人(U)	蓝色(B)	马(H)	蛋酒(E)	雪佛莱(C)
3	英国人(E)	红色(R)	蜗牛(S)	牛奶(M)	老爷车(O)
4	西班牙人(S)	象牙色(I)	狗(D)	桔汁(O)	奔驰(M)
5	日本人(J)	绿色(G)	斑马(Z)	可乐(C)	大众(V)

对于此类表格形式,采用矩阵表示的编码方式是自然而清晰的,例如:

$$\begin{array}{l} \text{人} \\ \text{房子} \\ \text{动物} \\ \text{饮料} \\ \text{车} \end{array} \begin{bmatrix} 3 & 4 & 2 & 1 & 5 \\ 4 & 2 & 3 & 5 & 1 \\ 2 & 1 & 3 & 5 & 4 \\ 5 & 2 & 1 & 4 & 3 \\ 4 & 1 & 2 & 5 & 3 \end{bmatrix}$$

编码的含义为:

1) 矩阵的每一列表示一个人的情况。如上述矩阵的第一列表示:英国人(E)住象牙色(I)房子,养马(H),喝矿泉水(W),开大众(V)车。

2) 矩阵的每一行表示某一变量的一种排列顺序。如上述矩阵的第二行表示:房子从左到右的颜色依次是,象牙色(I),蓝色(B),红色(R),绿色(G),黄色(Y)。

4.2 解的适应值

设A表示问题的一个可行解,则它满足:

- 1) $A(1, x_1) = 3 \wedge A(2, x_1) = 3$
- 2) $A(1, x_2) = 4 \wedge A(3, x_2) = 4$
- 3) $A(2, x_3) = 5 \wedge A(4, x_3) = 5$
- 4) $A(1, x_4) = 2 \wedge A(4, x_4) = 2$
- 5) $A(2, x_5) = 5 \wedge A(2, y_1) = 4 \wedge x_5 = y_1 + 1$
- 6) $A(5, x_6) = 3 \wedge A(3, x_6) = 3$
- 7) $A(5, x_7) = 1 \wedge A(2, x_7) = 1$
- 8) $A(4, x_8) = 2 \wedge x_8 = 3$
- 9) $A(1, x_9) = 2 \wedge x_9 = 1$
- 10) $A(5, x_{10}) = 2 \wedge A(3, y_2) = 1 \wedge ((x_{10} - y_2) = 1 \vee (y_2 - x_{10}) = 1)$
- 11) $A(5, x_{11}) = 1 \wedge A(3, y_3) = 2 \wedge ((x_{11} - y_3) = 1 \vee (y_3 - x_{11}) = 1)$
- 12) $A(5, x_{12}) = 4 \wedge A(4, x_{12}) = 4$
- 13) $A(1, x_{13}) = 5 \wedge A(5, x_{13}) = 5$
- 14) $A(1, x_{14}) = 1 \wedge A(2, y_4) = 2 \wedge ((x_{14} - y_4) = 1 \vee (y_4 - x_{14}) = 1)$

其中, $x_1 \sim x_{14}, y_1 \sim y_4$ 均是变量,取值为从1到5的整数。

这里,我们将某个解所满足的约束条件的个数记为此解的适应值,显然,可行解的适应值为14,算法的任务是快速而准确地找到一个可行解。必须指出,这样定义的适应值只能近似地定义某个解可能成为可行解的程度,但并不是绝对的,例

如,解B与可行解几乎一样,只需稍作调整即可得到可行解,然而其适应值可能不高;反之,解C与可行解差异甚大,但其适应值却接近14。我们认为这一方面是由于我们对启发信息没有进行更深的探讨,另一方面,在没有得到问题的确切解之前,导入一定程度“朦胧”的启发信息是合理的,而且是足够找到问题的解的。

4.3 遗传算子的选择

针对矩阵表示的编码方式并结合本问题,建议使用以下杂交算子和变异算子。

1) 杂交算子:随机交换两个矩阵的同一行。例如,对以下两矩阵的第4行进行杂交,得到如下后代,杂交过程见图1。

$$\begin{bmatrix} 3 & 4 & 5 & 1 & 2 \\ 2 & 5 & 3 & 4 & 1 \\ 5 & 2 & 1 & 3 & 4 \\ 3 & 4 & 1 & 5 & 2 \\ 2 & 1 & 3 & 5 & 4 \end{bmatrix} \leftrightarrow \begin{bmatrix} 4 & 1 & 2 & 3 & 5 \\ 2 & 3 & 5 & 1 & 4 \\ 5 & 4 & 1 & 2 & 3 \\ 3 & 2 & 4 & 5 & 1 \\ 3 & 4 & 5 & 2 & 1 \end{bmatrix}$$

↓

$$\begin{bmatrix} 3 & 4 & 5 & 1 & 2 \\ 2 & 5 & 3 & 4 & 1 \\ 5 & 2 & 1 & 3 & 4 \\ 3 & 2 & 4 & 5 & 1 \\ 2 & 1 & 3 & 5 & 4 \end{bmatrix} \leftrightarrow \begin{bmatrix} 4 & 1 & 2 & 3 & 5 \\ 2 & 3 & 5 & 1 & 4 \\ 5 & 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 5 & 2 \\ 3 & 4 & 5 & 2 & 1 \end{bmatrix}$$

图1

2) 变异算子:随机选取矩阵的两列,记为第 j_1, j_2 列,其中 $1 \leq j_1 < j_2 \leq 5$,另随机选取矩阵的 m 行(m 为1到5间的随机整数),记为第 $i_1, i_2, \dots, i_m$ 行,满足 $1 \leq i_1 < i_2 < \dots < i_m \leq 5$,变异作用将第 j_1 列的第 $i_1, i_2, \dots, i_m$ 个元素和第 j_2 列的第 $i_1, i_2, \dots, i_m$ 个元素分别交换。例如,假设随机选择矩阵的第2,4列,另随机选择矩阵的第2,3,5行,进行变异操作,变异过程见图2。

$$\begin{bmatrix} 4 & 1 & 2 & 3 & 5 \\ 2 & 3 & 5 & 1 & 4 \\ 5 & 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 5 & 2 \\ 3 & 4 & 5 & 2 & 1 \end{bmatrix}$$

↓

$$\begin{bmatrix} 4 & 1 & 2 & 3 & 5 \\ 2 & 1 & 5 & 3 & 4 \\ 5 & 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 5 & 2 \\ 3 & 2 & 5 & 4 & 1 \end{bmatrix}$$

图2

4.4 选择策略

采用 $(\mu + \lambda)$ 选择策略,从规模为 PopSize 的种群 P_t 出发,经过杂交算子和变异算子,产生相同规模的群体 Q_t ,从 $P_t \cup Q_t$ 中找出适应值最好的前 PopSize 个个体放入下一代中。

4.5 停机条件

程序找到满足所有约束条件的解,或者演化过程超出预设的最大演化代数 MaxGen,则停机。

4.6 问题的演化算法主框架

Algorithm Zebra:

$\tau = 0$;
initialize the population $P_t = \{p_1, p_2, \dots, p_N\}$;
evaluate each individual of P_t and find the one p_{best} with the best

```

fitness;
while  $p_{best} < 14$  and  $t < \text{MaxGen}$  do
begin
    perform crossover and mutation operator among  $P_t$  to generate
    offspring group  $Q_t$ ;
    select the top  $N$  best individuals from  $P_t \cup Q_t$  to replace  $P_t$ ;
     $t = t + 1$ ;
end.

```

其中, MaxGen 为设定的最大演化代数。

5. 实验与分析

我们用 C++ 语言实现了上述算法, 并进行了 10 次独立实验, 实验参数设置如下: 种群规模 PopSize=100; 最大演化代数 MaxGen=10000; 杂交概率=0.2; 变异概率=0.8。

需要说明的是, 之所以将变异概率设置较大, 是因为在实验过程中, 我们发现变异算子在搜索过程中起到了更关键性的作用, 这一方面是因为变异算子在演化过程中起到了探索未知的有希望的区域的作用, 另一方面变异算子克服了种群演化过程中多样性的缺失, 避免了过早收敛。

10 次实验中, 除一次运行到最大代数未收敛外, 其他均高效地找到了问题的解。算法所找到的问题的一个可行解为:

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

(上接第 136 页)

图6 (A,-,-)作为输入时网络的输出

5. 当(A,b,3)再次作为输入时, 只有(A,b,3)被回忆出来(图7)。

图7 (A,b,3)作为输入时网络的输出

结论 通过本文的研究和仿真实验, 可以得出以下结论:

(1) 本文提出的新型混沌神经网络模型(MSLCNN)能够区分已知模式和未知模式, 并且记忆未知模式(包括与已知模式存在共同项的模式), 具有与人脑相似的连续学习能力。(2) 混沌神经网络的连续学习能力与网络结构(即考虑时空总和的范围)、网络参数、权值学习算法等因素有密切的关系。在利用混沌神经网络进行连续学习时应综合考虑这些因素的影响。

即日本人养斑马, 挪威人喝矿泉水。上述解的形式并非巧合, 而是与编码有关, 打乱编码顺序, 解的形式会有所不同, 但是答案是唯一的。

值得注意的是, 算法有一次是不收敛的, 即找不到问题的可行解就结束了, 这主要是因为解空间中有很多局部最优解, 例如, 某些解虽然能满足 12 或 13 个约束条件, 然而它们与问题的可行解在结构上却相去甚远。一旦搜索种群陷于局部最优而无法有效跳出, 得不到可行解也就不足为奇了。这一点更说明前面提及的算法的启发信息挖掘的重要性。启发信息不全面容易造成算法的误导。

结论 约束可满足性问题是实际应用中常见的一类问题, 本文初步探索了用演化算法求解约束可满足性问题, 实验表明所提方法是可行、有效的。类似的更多更具挑战性的约束可满足性问题将成为我们下一步的研究对象。

参考文献

- 1 Michalewicz Z, Fogel D B. How to Solve It: Modern Heuristics [B]. Springer, Berlin, 2000. 157~160
- 2 Holland J H. Adaptation in Nature and Artificial Systems [B]. MIT Press, 1992
- 3 Goldberg D E. Genetic Algorithms in Search, Optimization, Machine Learning [B]. Reading MA: Addison Wesley, 1989
- 4 潘正君, 康立山, 陈毓屏. 演化计算[B]. 清华大学出版社, 1998. 1~8, 16~17

参考文献

- 1 Aihara K, Takabe T, Toyoda M. Chaotic neural networks [J]. Physics A, 1990, 144(6-7): 333~340
- 2 Yao Y, Freeman W J. Model of biological pattern recognition with spatially chaotic dynamics [J]. Neural Networks, 1990, 3: 153~170
- 3 Kaneko K. Clustering, coding, switch, Hierarchical ordering and control in a network of chaotic elements. Physics D, 1990, 41: 137~172
- 4 Ishii S, Fukumizu K, Watanabe S. A network of chaotic elements for information processing. Neural networks, 1996, 9(1): 25~40
- 5 Inoue M, Nagayoshi A. A chaos neurocomputer. Physics A, 1991, 158: 373~376
- 6 Osana Y, Hattori M, Hagiwara M. Chaotic bi-directional associative memory. IEEE International Conference on Neural Networks, Houston, 1997, 2: 816~821
- 7 Osana Y, Hagiwara M. Separation of superimposed pattern and many-to-many associations by chaotic neural networks. International Joint Conference on Neural Networks, Anchorage, 1998, 1: 514~519
- 8 Osana Y, Hagiwara M. Successive learning in chaotic neural network. International Joint Conference on Neural Networks, Anchorage, 1998, 2: 1510~1515
- 9 段书凯, 刘光远. 混沌神经网络在分离叠加模式和信息恢复中的应用[J]. 西南师范大学学报(自然科学版), 2003, 28(2): 218~221
- 10 刘光远, 段书凯. 混沌神经网络在分离叠加模式和多对多联想中的应用[J]. 计算机科学, 2003, 30(3): 83~85
- 11 Shukai D, Guangyuan L, Lidian W. A Novel Chaotic Neural Network for Many-to-Many Associations and Successive Learning. In: Intl. Conf. on Neural Networks and Signal Processing, Nanjing, 2003 (acceptance)