

独粒钻石棋的化学反应优化解法

曹建立¹, 王泳², 徐刚³

(1. 洛阳师范学院数学科学学院, 河南洛阳 471934; 2. 武汉光谷现代有轨电车运营有限公司, 湖北武汉 430075; 3. 洛阳师范学院学报编辑部, 河南洛阳 471934)

摘要: 本文介绍了 CRO (Chemical Reaction Optimization) 算法的基本原理, 并对该算法进行了改进, 增加了精英选择策略, 其用 Java 语言加以实现, 并将算法用于求解独立钻石棋局问题, 之后分析了 CRO 参数选择对实验结果的影响, 最后总结了 CRO 算法的优缺点和适用场景。

关键词: 化学反应优化算法; 独粒钻石; 精英选择; Java

中图分类号: TP311

文献标识码: A

文章编号: 1009-4970(2017)08-0036-05

1 CRO 算法基本思想

在过去的几十年中, 研究者通过观察和模仿生物演化活动, 发明了大量的进化算法。有些算法受到生物基因进化过程的启发, 如遗传算法^[1] (Genetic Algorithm)、文化基因算法^[2] (Memetic Algorithm)、差分进化算法^[3] (Differential Evolution); 有些则受到生物种群活动的启发, 如蚁群算法^[4] (Ant Colony Optimization)、粒子群算法^[5] (Particle Swarm Optimization); 此外, 还有研究者将模拟对象由生物领域拓展到了非生物领域, 如模拟退火算法^[6] (Simulated Annealing) 源于对物质降温结晶现象的模拟; 2010 年, 受到化学反应原理的启发, Lam 和 Li 提出了化学反应优化算法^[7]。

简单来说, 化学反应遵循两个规律: 一是能量守恒, 即能量不能被创造, 也不能被消灭, 只能从一种形态转化成另一种形态, 或在物质之间转移。分子结构趋于低能量、高稳定性的原理。二是系统的熵 (即系统的混乱程度) 趋向于增加。

分子包含两种能量: 势能和动能。分子的势能由分子的内部结构决定, 即分子本身包含的能量; 分子的动能则由分子运动速度决定。当物质分子包含多余的势能时, 即它处于不稳定状态, 其试图通过改变自己的分子结构来释放掉多余的势能, 从而达到稳定状态。从宏观上看, 反应得到的产品往往比参与反应的物质更加稳定。

CRO 算法就是在计算机中模拟出一群分子在容

器中进行化学反应的过程。算法每次按一定概率决定是发生单分子反应还是双分子反应, 然后随机从容器中选择待反应的分子。对单分子反应, 其检测该分子的动能、势能、环境能量是否满足分裂条件, 如满足则发生分裂, 否则发生碰壁; 对双分子反应, 先检测两个分子是否满足发生合成反应的条件, 如果满足则进行, 如果不满足, 则进行交换反应。

需要注意的是, 满足反应发生的条件仅仅触发模拟反应过程, 不能保证反应的成功和分子结构的更新。模拟完后需要对新生成分子的势能进行检测, 当满足反应成功条件时, 则给新分子分配动能, 然后用新分子替代原分子, 进而完成一次反应。否则, 本次模拟失败, 容器中分子保持原状, 继续进行下一次反应的模拟。

分子通过算法给出的碰壁、碰撞、合成、分解四种基本反应, 分阶段多次改变自身结构, 并向环境中释放掉蕴藏在自身中的多余势能, 从而形成更稳定结构。每次基本反应都要遵循化学反应的规律。CRO 算法允许分子在反应过程中按一定的概率, 从环境中获得能量, 来增加自身的势能。这样可以使得该分子具备逃离局部极小值的能力, 从而得以继续寻找远处的极值点, 进而避免了因收敛过快而很快陷入局部极小值的问题。CRO 算法的这个特点和 SA 模拟退火算法用小概率接受更差的解作

收稿日期: 2017-03-02

基金项目: 国家自然科学基金(61572246); 河南省科技创新人才支持计划(164100510003)

作者简介: 曹建立(1978—), 男, 河南洛阳人, 博士, 讲师。研究方向: 计算机体系结构、智能算法。

· 36 ·

用类似. 碰壁、碰撞、合成、分解这四种基本反应 的描述如表 1.

表 1 四种基本反应的描述

反应名称	描述	反应前后分子数量	反应前提条件	反应成功条件
撞墙	一个分子和容器碰撞, 结构发生轻微变化, 释放一部分能量到环境中	1/1	--	新分子势能 \leq 原分子动能 + 原分子势能
分裂	一个分子从环境中吸收能量, 分裂成新两个分子	1/2	分子碰壁次数大于 Alpha	两个新分子势能之和 \leq 原分子动能 + 原分子势能 + 从环境中吸收的能量
交换	两个分子碰撞后, 结构都发生轻微变化	2/2	--	两个新分子势能 \leq 两个原分子动能 + 两个原分子势能
合成	两个分子发生碰撞后, 结合一个具有不同结构的新分子	2/1	两个分子势能均小于 Beta	新分子势能 \leq 原分子动能 + 原分子势能

如果将不同的分子结构看做问题的不同解法, 则化学反应过程和计算机领域中的优化问题十分相似, 都是在解空间中寻找势能最低的结构(问题结果最优的解)的过程. CRO 算法正是将问题的不同解映射为分子的不同结构, 然后将这组解(分子结构)按照化学反应的规律进行处理. 当分子结构达到势能较低的状态时, 其对应的解也得了优化, 这就是我们利用 CRO 算法来求解优化问题的原理.

2 独粒钻石棋

独粒钻石棋^[8] (Solitaire) 和中国人设计的“华容道”、匈牙利人设计的“魔方”一起被人们称为智力游戏界的三大发明. 其棋盘上共有 33 个交叉点. 在不同的位置上放置棋子便构成了多种不同的棋局. 较经典的棋局有十字架、大十字、金字塔、古、台灯、大金字塔、五边形、戴维斯跳跃、传统钻石. 几种常见独粒钻石棋局如图 1 所示.

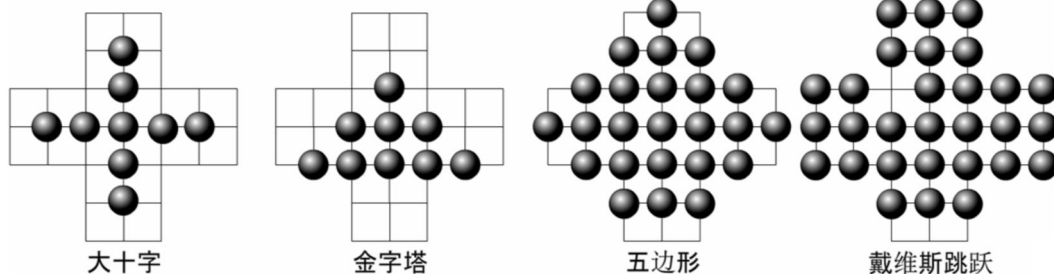


图 1 几种常见独粒钻石棋

该游戏的行棋规则是当某棋子跳过相邻的棋子到空位上, 就把被跳过的棋子拿掉. 棋子可以沿格线横、纵方向跳, 但不能斜跳. 每跳一步, 棋子数量会减少一个, 行棋的目标是剩下尽量少的棋子. 最后能剩下 3 枚棋子的级别为“聪明”, 剩下 2 枚棋子是“尖子”, 剩下 1 枚棋子是“大师”, 最后剩下 1 只, 而且位于棋盘正中央则是“天才”.

一枚棋子连续跳过和吃掉多个棋子可以当做一步, 在剩余棋子数量相同的情况下, 步数更少的解法更优秀.

3 算法设计及实现

Java 是一种面向对象、与平台无关的计算机编程语言, 拥有功能丰富的类库、自动垃圾回收机制, 是进行算法验证、原型设计的优秀工具. 本文

的算法使用 Java 5.0 来实现.

用 CRO 来求解独粒钻石问题, 需要先将求解方案用编码的方式表达出来, 并认为该组编码代表了一种分子结构, 从而对应一个分子势能(即该求解方案的优劣程度). 然后初始化反应环境和一定数量的分子, 再模拟化学反应的规律使分子势能不断下降(不断优化方案), 当某个分子的结构对应的解决方案达到需求后, 便可终止模拟, 进而输出解决方案.

3.1 将求解方案用二进制编码表示

在该问题中, 每一步移动操作包含两个部分: 选择棋子和选择移动方向. 选中的棋子可以用其在棋盘中的坐标来确定, 则右上角坐标为(6,6), 用 0、1、2、3 四个数字表示右、下、左、上方向. 编码 3,3,1 表示将棋盘正中的棋子向下跳动一步.

因为坐标数字最大为 6, 用 3 位二进制可以表示; 方向数字最大为 3, 用 2 位二进制可以表示; 则一步可能的移动用 8 位二进制表示. 例如: 编码 3, 3, 1 二进制为 01101101. 采用这种二进制的编码方案会造成一定程度的冗余, 但优点也较为明显: 在模拟化学反应的过程中只用处理二进制, 简化了操作.

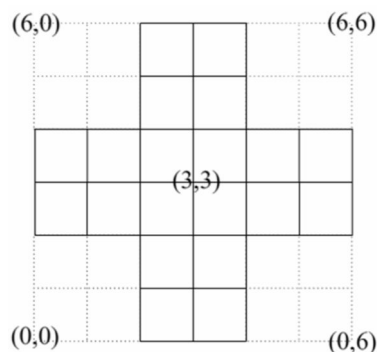


图 2 棋盘坐标定义

一个可能的解应该是一个由若干步构成的序列, 因为序列中包含有错误的走法(如棋子坐标无效、移动方向无效等), 所以序列长度可以根据开局时棋子的数量不同灵活调整, 应多于棋子的数目, 经验值取 2000.

3.2 初始化分子结构

每个分子的结构用一个二进制序列表示, 并使用随机函数来初始化. 由于化学反应模拟中需要随机的访问序列中的指定位, 因此采用 Java 中支持随机访问, 且顺序访问效率较高的 ArrayList 作为容器来保存.

3.3 计算每个分子的势能

分子的势能取决于分子结构, 是该分子所对应的解决方案的量化优劣程度. 对于独粒钻石棋局求解问题, 就是该分子对应的走棋序列能最终剩余的棋子数量. 该值越小, 说明该分子对应的解决方案越优秀.

每个分子结构表示的移动序列中的数据是随机产生的, 而且其要经过碰壁、分裂、交换或合成操作. 分子结构对应的移动序列中有些步是非法的, 比如棋子位置是(0,0), 或者试图将棋子移动到棋盘外, 或者没有空位的地方. 这些非法的移动步在计算势能时可以直接忽略掉, 也可以作为负反馈, 以降低该分子对应解法的优秀程度.

3.4 开始模拟化学反应

由于标准算法是单线程实现, 每次只能模拟一个或者一对分子的反应. 由随机函数和参数

MoleColl 决定是单分子反应还是双分子反应. 然后再次采用随机函数从容器中选择待反应的分子, 并检测该分子的动能、势能、环境能量是否满足分裂(单分子反应)或者合成(双分子反应)反应的条件, 如果满足则进行; 如果不满足, 则分别进行撞墙和交换反应.

本文对标准 CRO 算法进行了一点改进, 加入了遗传算法中的精英选择策略, 在模拟反应前先判断当前选择的分子是否是势能最低的分子(即代表最优解的分子). 如果是, 则放弃本次反应. 该策略可以保证最优算法不会被破坏掉, 从而加快收敛速度.

3.5 结束模拟

当某个分子的势能降低到结束阈值时, 即得到了满足条件的解, 则模拟过程结束. 对于独粒钻石棋局求解问题, 希望只留下一颗棋子, 因此将结束阈值设为 1. 但对于某些复杂棋局, 为了在可接受的时间内结束模拟, 可将阈值设为 2 或者更大.

4 实验结果及分析

CRO 的参数会直接影响到模拟结构的优劣和模拟时间, 目前尚未有成熟的公式可以确定最优参数配置, 因而需要研究者根据问题的特点, 通过反复实验来确定. 本组实验采用的参数如表 2 所示.

表 2 参数配置

参数名称	含义	参数值
MoleLen	分子长度	2000
InitSize	初始化分子数量	150
DEFAULT_KE	分子默认动能	40
InitBuffer	初始化缓冲区能量	0
KELossRate	撞墙分子能量保留的比例下限	0.6
MoleColl	发生多分子反应的概率	0.2
Alpha	引发分裂的单分子无效撞击次数阈值	100
Belta	引发合成的动能阈值	40
Threshold	结束模拟的势能阈值	1/2

模拟时, 该算法在 Intel i3 2120, 4G 内存, Win7 操作系统, Java5.0 的环境下运行通过, 部分实验结果如表 3 所示.

从结果中可以看到, 由于棋局复杂程度的不同, 在算法进行的多次实验中, 绝大多数棋局均可在有限次模拟化学反应后得到满意的结果, 仅留下一枚棋子; 而部分棋局在可以容忍的时间内得到的是余下两枚棋子的解法, 说明算法还需通过优化参数来提高性能, 以得到更好的结果. 由于 CRO 是非

表 3 部分实验结果

布局名称	开局棋子数量	平均反应次数 (含不成功的反应)	平均耗时/毫秒	余下棋子个数
十字架	6	1.00	322.61	1
大十字	9	648.50	1145.83	1
金字塔	9	75522.00	53813.50	1
古	15	6828.59	8197.68	1
大金字塔	16	121830.70	85714.20	1
台灯	17	575480.50	377514.50	1
五边形	24	987804.20	673961.60	1
戴维斯跳跃	31	1827246.60	1217932.70	1
传统钻石	31	488413.80	313092.70	2

确定性算法, 算法结果的优劣程度和运行时间均和概率有关, 所以有时会出现复杂的棋局求解时间短于简单棋局的现象(如表 2 中 15 枚棋子的棋局求解时间短于 9 枚棋子的求解时间)。但整体来说, 复杂的棋局需要的运行时间更长。

在实验中调整任一个参数的值, 都会影响到最终实验结果和收敛速度。其中, 增加分子长度和初始分子数量能加快算法的速度, 但要消耗较多的内存和处理器资源。

图 3 描述了在求解“戴维斯跳跃”棋局问题时, 四种基本反应发生的频率。四种基本反应发生的频率和参数 Alpha、Beta、MoleColl 的值有很大关系。Alpha 是分裂反应发生的撞墙次数阈值, 较高的 Alpha 值会降低分裂反应的发生次数; Beta 是合成操作的分子势能阈值, 较小的 Beta 值会减少合成操作的发生次数; MoleColl 则决定了单分子和双分子反应的比例。本组实验中, MoleColl 的值为 0.2, 即多分子反应概率是 20%, 单分子反应概率是 80%, 所以两种单分子反应的撞墙反应频率 78% 和分解反应频率 2% 之和等于 80%。

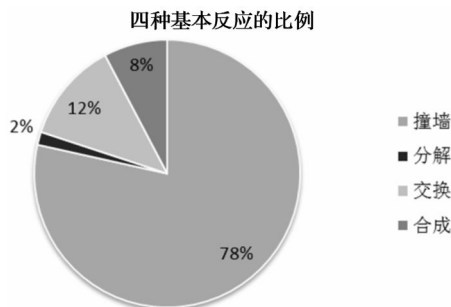


图 3 四种基本反应的比例

上述几个参数决定了不同类型反应发生的频率, 但是反应是否能成功, 即得到新的分子结构, 逼近最优解, 则取决于是否满足表 1 中描述的条件。

图 4 给出四种基本反应的成功率。撞墙反应的反应物和产品都是一个分子, 条件比较简单, 容易满足, 故成功率较高; 交换反应的反应物是两个分子, 产品也是两个分子, 是否成功涉及到 4 个分子的动能势能分配, 条件较为苛刻, 所以成功率较低; 而分解反应和合成反应的条件涉及 3 个分子, 成功率在前二者之间。

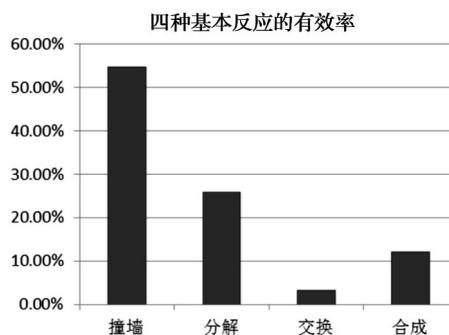


图 4 四种基本反应的有效率

5 结语

CRO 算法属于启发式算法, 其优点是无需理解待求解问题的内部机制, 只需要给出解空间和判断解的优劣程度的判定函数, 即可利用化学反应中的分子运动趋向于释放能量、达到低能量稳定状态的原理, 以获得较好的结果。根据 Wolpert 和 Macready 的“*No Free Lunch*”^[9]理论, 所有的启发式算法本质上都是相似的, 某个算法如果在某些问题上有突出性能, 那么在另外一些问题上则会有较差的表现, 这些算法在全部问题上的整体效率是相近的。CRO 算法的出现, 可以弥补其他智能算法在某些问题上的缺陷, 从而拓展了智能算法的应用范围。

但是此算法的其缺点也很明显, 主要有: (1) 无法保证获得最优解, 甚至无法保证获得次优解, 也无法保证在确定时间内获得结果; (2) 结果的优劣

严重依赖于算法参数组的选择,而参数组的取值依赖经验,目前尚未有精确的公式可循.不同的问题和不同的编码方案,适应不同的参数组,需要在实验中反复调整优化,这给 CRO 算法的应用带来一定难度;(3)和确定性算法相比,CRO 算法耗费更多的内存和处理器资源,收敛速度较慢.

但其原理简单、易于实现的特点还是使得该算法在大量问题求解中得到了广泛应用.尤其对于不存在确定性解法的 NP 完全问题,该算法和其他的智能算法,如遗传算法、蚁群算法、粒子群算法等结合,可以获得更好的解法和更高的效率.

标准 CRO 算法是串行算法,同一时刻只能模拟一个基本反应.当前 CPU 处理器大都拥有多核、超线程技术,GPU 也具有通用计算的能力,如何充分利用这些计算资源,并通过并行技术来加速 CRO 算法,是下一步的研究目标.

参考文献

- [1] Goldberg D E. Genetic Algorithms in Search, Optimization and Machine Learning[J]. 1989.
- [2] Ong Y S, Meng H L, Chen X. Research frontier: memetic computation – past, present & future[J]. IEEE Computa-

tional Intelligence Magazine, 2010, 5(2): 24 – 31.

- [3] Price K V, Storn R M, Lampinen J A. Differential Evolution – A Practical Approach to Global Optimization [J]. 2005, 141(2): 1 – 24.
- [4] Dorigo M, Stützle T. Ant Colony Optimization: Overview and Recent Advances [J]. International, 2010, 146(5): 227 – 263.
- [5] Bonabeau E, Dorigo M, Theraulaz G. Swarm intelligence: from natural to artificial systems [M]. Oxford: Oxford University Press, Inc., 1999.
- [6] Laarhoven P J M, Aarts E H L. Simulated annealing: theory and applications [M]. Dordrecht; Boston: D. Reidel Publishing Company, 1987: 108 – 111.
- [7] Lam A Y S, Li V O K. Chemical – reaction – inspired metaheuristic for optimization [J]. IEEE Transactions on Evolutionary Computation, 2010, 14(3): 381 – 399.
- [8] 曹建立,周震.基于遗传算法的独粒钻石棋求解方法 [J]. 洛阳师范学院学报, 2011, 30(2): 69 – 71.
- [9] Wolpert D H, Macready W G. No free lunch theorems for optimization [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67 – 82.

[责任编辑 王保玉]

Using CRO Algorithm to Solve Solitaire Chess Problem

CAO Jian-li¹, WANG Yong², XU Gang³

- (1. College of Mathematics and Science, Luoyang Normal University, Luoyang 471934, China;
2. Wuhan Optics Vally Modern Tram Co., Ltd., Wuhan 430075, China;
3. Academic Journal Editorial Office, Luoyang Normal University, Luoyang 471934, China)

Abstract: This paper introduces the principle of CRO (Chemical Reaction Optimization) algorithm, improves it using elite selection strategy, and implements this algorithm it with Java language. We use CRO in resoling solitaire chess problem and analyze the influence of parameters on the results. In the end, we summarize the merits and drawbacks of CRO, and point out its applicable scope.

Key words: CRO; solitaire; elite selection strategy; Java