

基于重要解成分的信息素更新策略

闭应洲^{1,2} 钟 智¹ 丁立新² 元昌安¹

(广西师范学院计算机与信息工程学院 南宁 530004)¹ (武汉大学软件工程国家重点实验室 武汉 430072)²

摘 要 蚁群优化算法通过信息素记录搜索过程中获取的知识,并基于信息素搜索新的解,因此好的信息素更新策略对蚁群优化算法至关重要。针对不同解成分的贡献不同的特点,提出了新的信息素更新策略:首先识别候选解的重要成分,然后在更新信息素时只允许重要的解成分得到加强。基于新的更新策略更新的信息素更好地反映了优质解的特点,从而加快了信息的正反馈过程。以 4 阶欺骗问题为例,验证了新算法的有效性。

关键词 蚁群优化算法,信息素更新策略,欺骗问题

中图分类号 301.6 文献标识码 A

Policy of Pheromone Update with Important Solution Components

BI Ying-zhou^{1,2} ZHONG Zhi¹ DING Li-xin² YUAN Chang-an¹

(College of Computer and Information Engineering, Guangxi Teachers Education University, Nanning 530004, China)¹

(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)²

Abstract The pheromone trails in ACO are used to reflect the ants' search experience, and the ants exploit them to probabilistically construct solutions to the problem, so the quality of the pheromone is crucial to the success of ACO. The main factors affecting the quality of the pheromone include the policy of updating the pheromone and the quality of the constructed solutions. In order to improve the constructed solutions, this paper presented a method to analyze the invalid components of the constructed solution, and then repaired the invalid components with immunity operator. When the pheromone density on the components is updated according to the improved solution, they will more exactly reflect the character of high quality solution, so it will speed the positive feedback procedure. The results show that the use of immunity repairing helps to find competitive solutions in a relatively short time.

Keywords Ant colony optimization, Policy of pheromone update, Deceptive problem

1 引言

蚁群优化算法通过信息素记录搜索过程中获取的知识,并基于信息素搜索新的解,因此信息素的质量十分重要,不同的信息素更新策略对蚂蚁优化算法的性能影响很大。最初的蚂蚁系统(ant system)AS 算法有 3 个不同的版本,分别为蚂蚁密度(ant-density)、蚂蚁数量(ant-quantity)和蚂蚁周期(ant-cycle)。在蚂蚁密度和蚂蚁数量这两个版本中,蚂蚁从一个城市转移到另一个城市后就直接释放信息素,但在蚂蚁周期中,只有所有蚂蚁都构建出一条路径后才执行信息素更新。实践证明,蚂蚁周期算法性能更高^[1]。

为了进一步提高蚁群优化算法的性能,更好的信息素更新策略已被应用到 ACO 算法当中,其中包括:1)每一代中只有构建最好路径的蚂蚁,或者是构建至今最优路径的蚂蚁,才允许释放信息素;2)对路径上的信息素浓度取值范围进行限制^[2,3];3)信息素动态更新策略^[4];4)将信息素分成有限个级别,通过级别的更新实现对信息素的更新^[5]等。通过改进信息素更新策略,这些新的算法都比 AS 算法获得更好的性能。为了解决蚂蚁算法初期信息素匮乏、求解速度慢的问题,文献

[6,7]提出将遗传算法与蚂蚁算法融合,采用遗传算法产生初始的信息素分布。

对于欺骗性问题,蚁群优化算法与遗传算法一样很难找到全局最优解^[8],为了更好地解决欺骗性问题,本文提出了基于重要解成分的信息素更新策略:(1)产生初始信息素阶段:首先应用遗传算法获取局部最优解,然后识别这些局部最优解的重要成分,再根据重要的解成分更新信息素;(2)蚁群优化过程阶段:只允许每代构建最优解的蚂蚁更新信息素,在更新信息素时只允许重要的解成分得到加强。试验表明,新的更新策略使得信息素能更准确地反映优质解的特点,从而加快信息的正反馈过程,达到提高蚁群优化算法性能的目的。

2 4 阶欺骗问题

所谓的 4 阶骗子问题定义如下:

$$f(x_1, x_2, x_3, x_4) = \begin{cases} \sum_{i=1}^4 x_i & x_i \in \{0, 1\}, x_i \text{ 不全是 } 0 \\ 5 & x_i \text{ 全是 } 0 \end{cases} \quad (1)$$

从函数的定义可知, $f(0001), f(0011), f(0111), f(1111)$ 的函数值分别为 1, 2, 3, 4, 似乎 x_i 中包含 1 的个数越

到稿日期:2009-11-03 返修日期:2010-01-03 本文受国家自然科学基金项目(60763012),广西自然科学基金项目(0991104)资助。

闭应洲(1967—),男,博士,副教授,CCF 会员,主要研究方向为智能计算、智能信息处理;钟 智(1963—),副教授,主要研究方向为数据挖掘;丁立新(1967—),教授,博士生导师,主要研究方向为演化计算、智能信息处理;元昌安(1963—),博士,教授,主要研究方向为智能计算、数据挖掘。

多,函数值越大,但当 x_i 全为 0 时, $f(0000)$ 的函数值等于 5,属于全局最优;当 x_i 全为 1 时,函数值等于 4,属于局部最优,也被称为欺骗吸引子。更大的 4 阶欺骗问题,比如一个 32 位的 4 阶欺骗问题由 8 个 4 阶欺骗子问题组成,如图 1 所示,上方是 4 阶欺骗子问题 x_1, x_2, x_3, x_4 的取值,下方是其对应的函数值,32 位的 4 阶欺骗问题的函数值为 26。

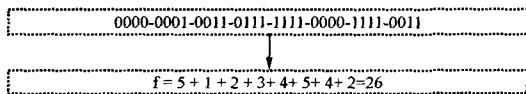


图 1 更大的 4 阶欺骗问题

在遗传算法中,模式定理和建筑块假设 (building block hypothesis) 被认为是其理论基础。根据建筑块假设,遗传算法的工作过程就是,从选择相互竞争的低阶、短定义距的模式开始,逐步将它们组合形成高阶的模式,然后不断重复这个过程,直到一个长度为 $l-1$,阶为 l 的模式被创建和发现,如果一切顺利的话它就是全局最优解^[9]。令 $f(0\#\#\#)$ 表示模式 $0\#\#\#$ 的适应值, $f(1\#\#\#)$ 表示模式 $1\#\#\#$ 的适应值, $f(00\#\#)$ 表示模式 $00\#\#$ 的适应值, $\#$ 表示相应位可取任意值,其余类推。在演化搜索过程中很容易出现 $f(0\#\#\#) < f(1\#\#\#)$, $f(00\#\#) < f(11\#\#)$, 直至 $f(00\#0) < f(11\#1)$ 的情况,也就是说低阶模式会误导高阶模式,因此求解欺骗问题对于遗传算法来说是困难的。

3 求解 4 阶欺骗问题的 ACO 算法

为了提高算法的性能,采用了遗传算法来产生初始的信息素分布。ACO 算法有两个主要的步骤,即蚂蚁构建问题的解和信息素的更新。图 2 是求解 4 阶欺骗问题的 ACO 算法框架。

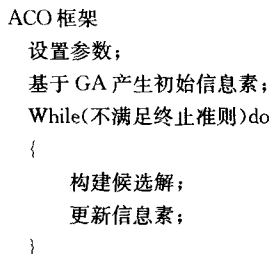


图 2 求解 4 阶欺骗问题的 ACO 算法框架

蚂蚁 $k(k=1, 2, \dots, m)$, 蚁群规模为 m 在构建候选解的过程中,根据信息素浓度决定各变量的取值。 $p_{ij}^k(t)$ 表示在 t 时刻蚂蚁 k 中变量 i 取值为 1 的概率:

$$p_{ij}^k(t) = \frac{\tau_{ij}}{\tau_{i0} + \tau_{i1}} \quad (2)$$

式中, $\tau_{i0}(t)$ 表示 t 时刻变量 i 取值为 0 的信息素浓度, $\tau_{i1}(t)$ 表示 t 时刻变量 i 取值为 1 的信息素浓度。经过 n 个时刻 (n 为变量个数),所有蚂蚁都构建好候选解后,接着是信息素更新过程。首先,所有边上的信息素都会蒸发:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \quad i=1, 2, \dots, n, j \in \{0, 1\} \quad (3)$$

ρ 表示信息素蒸发系数。其次,在每代最优蚂蚁经过的边上都要增加信息素:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij} \quad i=1, 2, \dots, n, j \in \{0, 1\} \quad (4)$$

$$\Delta\tau_{ij} = \begin{cases} \delta, & \text{if } (i, j) \in \text{每代最优} \\ 0, & \text{其它} \end{cases} \quad (5)$$

式中, $\Delta\tau_{ij}$ 表示如果每代最优蚂蚁第 i 个解成分的取值为 1 时, $\tau_{ij}(t)$ 的信息素增加 δ , 否则 $\tau_{ij}(t)$ 的信息素增加 0。

图 3 所示的算法是求解 4 阶欺骗问题的遗传算法,选择父体的概率规则为:

$$p(x_i) = \frac{f^a(x_i)}{\sum_{i=1}^N f^a(x_i)} \quad (6)$$

式中, N 为种群规模, pr 为复制概率, pc 为交叉概率, pm 为变异概率, α 控制选择压力。尽管采用遗传算法产生了初始的信息素分布,ACO 算法的性能有所提高,但对于 4 阶欺骗性问题,还是很难找到全局最优解。

遗传算法

随机初始化种群 $P(0) = \{x_1, x_2, \dots, x_N\}$, $t := 0$;

计算 $P(0)$ 中个体的适应值;

While(不满足终止准则)do

{ //由 $P(t)$ 通过遗传操作形成新的种群 $P(t+1)$;

for ($k=0; k < N, k=k+2$)

{

在 $P(t)$ 中选择两个父体;

$r = \text{random}[0, 1]$;

if ($r \leq pr$) 执行复制操作,即将两个父体直接插入到 $P(t+1)$;

else if ($r \leq pr + pc$) 执行两点交叉操作,将两个后代插入到 $P(t+1)$;

对两个父体分别执行变异操作(对个体中的每一位按概率 pm 翻位),将两个后代插入到 $P(t+1)$;

}

计算 $P(t+1)$ 中个体的适应值

$t = t + 1$;

}

图 3 求解 4 阶欺骗问题的遗传算法

4 基于重要成分的信息素更新

通过比较遗传算法和蚁群优化算法的不同运行次数所得到的局部最优解,比如某次蚁群优化算法找到的最优解为: 0000-0000-0000-0000-0000-1111-1111-0000-0000-1111-0000-0000-0000-0000-0000, 其对应适应值为 72, 不难发现局部最优解中包含有好的部分解 0000, 也包括欺骗性的部分解 1111。当好的部分解的某个解成分由 0 变为 1 时,比如由 0000 变为 0001 时,对应候选解的适应值也就下降了 4, 而当欺骗性的部分解的某个解成分由 1 变为 0 时,比如由 1111 变为 1110 时,对应候选解的适应值也就下降了 1。由此可见,在局部最优解中不同的解成分对适应值的贡献是不同的。

在蚁群优化算法中,传统的信息素更新策略没有考虑不同解成分的贡献不同,只要一只蚂蚁被允许更新信息素,相关的所有解成分都得到同样的加强。为了更准确反映优质解的特点,本文提出了基于重要解成分的信息素更新策略。

在应用某个优质候选解更新信息素之前,首先识别它的重要解成分,并将其作标记;每次更改 1 个解成分的值,其余不变,并记录适应值的变化。比如第 i 位变量由 0 变为 1 后,适应值的变化为:

$$\delta_i = f(*, *, \dots, 1, \dots, *) - f(*, *, \dots, 0, \dots, *) \quad (7)$$

δ_i 的值越小,说明第 i 个解成分被更改后,适应值下降越多,即说明第 i 个解成分对适应值的贡献越大,因此第 i 个解成分越重要。通过对 δ_i 按升序排序,一般将前面 50%~80% 的解成分标记为重要解成分。

在重要解成分的指引下,在更新信息素时更有导向性:对于每代最优蚂蚁经过的边,只有重要的边(对应重要的解成分)才增加信息素,式(5)变为式(8):

$$\Delta\tau_{ij} = \begin{cases} \delta, & \text{if}(i,j) \in \text{重要解成分} \\ 0, & \text{其它} \end{cases} \quad (8)$$

5 实验

为了验证新的信息素更新策略的效果,本文设置了3组实验,它们都基于图2所示的ACO框架求解一个60位的4阶欺骗问题,但分别应用不同的信息素更新策略。计算机实验平台是:CPU 2.93GHz, Ram 1G,操作系统 Windows XP,开发工具 VC6.0。

第1组实验采用传统的信息素更新策略,但为了进一步分析基于GA产生初始信息素的作用,进行了两个实验对比:在实验1中基于GA产生初始信息素的过程只运行一次,并将该方法称为1GA+ACO,而在实验2中基于GA产生初始信息素的过程重复运行3次,将该方法称为3GA+ACO。由于GA为带随机性的优化算法,每一次搜索找到的局部最优解很可能不一样,本文设想通过多次运行基于GA产生初始信息素的过程,使信息素能反映多个局部最优解的特点,从而使得基于信息素产生的候选解可综合多个局部最优解的优点。

GA的参数设置为 $N=100$, $pr=0.4$, $pc=0.6$, $pm=0.03$, $\alpha=15$, 迭代100代。在应用GA初始化信息素前,令 $\tau_{11}(t)=\tau_{0}(t)=c$, $i=1,2,\dots,n$, c 为常数,本文中设 $c=0.003$, 然后基于GA获取的解根据式(3)~式(5)初始化信息素。ACO算法中参数设置如下:蚁群规模 $m=100$, 迭代200代, $\rho=0.01$, $\delta=0.01$, 按照上述参数设置独立运行20次,1GA+ACO和3GA+ACO都没有找到最优解,其适应值在66~73之间。1GA+ACO的运行时间平均为0.519秒,而3GA+ACO的运行时间平均为1.318秒。部分运算结果如表1所列。

表1 基于传统信息素更新策略求解4阶欺骗问题的部分运行结果

代数	1GA+ACO			3GA+ACO		
	最小值	平均值	最大值	最小值	平均值	最大值
初始解	56	57	60	54	56	58
50代	66	68	69	66	68	70
100代	66	68	70	66	69	72
200代	66	68	72	66	69	73

第2组实验采用基于重要解成分的信息素更新策略,同样进行两个实验对比:在实验1中基于GA产生初始信息素的过程只运行一次,并将该方法称为1GA+ACO,而在实验2中基于GA产生初始信息素的过程重复运行3次,将该方法称为3GA+ACO。GA算法以及ACO算法中参数设置和第一组实验完全相同。在应用某个优质候选解更新信息素之前,首先基于式(7)计算它的某个解成分对适应值的贡献,然后选取贡献较大的解成分作为重要解成分更新信息素,即根据式(3),式(4),式(8)更新信息素。本实验中重要解成分占解成分总数的65%。按照上述参数设置独立运行20次,在1GA+ACO算法中,11次找到全局最优,而在3GA+ACO算法中,每次都找到全局最优。1GA+ACO的运行时间平均为0.536秒,而3GA+ACO的运行时间平均为1.331秒。部分运算结果如表2所列。

表2 基于重要解成分的信息素更新策略部分运行结果

代数	1GA+ACO			3GA+ACO		
	最小值	平均值	最大值	最小值	平均值	最大值
初始解	53	56	58	60	65	72
50代	71	72	75	74	75	75
100代	71	73	75	75	75	75
200代	71	74	75	75	75	75

在第3组实验中,ACO算法采用基于重要解成分的信息素更新策略,但没有基于GA产生初始信息素的过程。为了比较GA与ACO求解4阶欺骗问题的性能,以及进一步分析基于GA产生初始信息素的作用,还研究了GA算法求解4阶欺骗问题的实验结果。GA算法以及ACO算法中参数设置和前面两组的实验基本相同,唯一的区别是GA算法迭代200代。独立运行20次后,GA算法和ACO算法都找不到全局最优。GA的运行时间平均为0.813秒,而ACO的运行时间平均为0.122秒。部分运算结果如表3所列。

表3 基于GA以及单一ACO的部分运行结果

代数	GA			ACO		
	最小值	平均值	最大值	最小值	平均值	最大值
初始解	45	46	48	44	47	49
50代	64	66	70	64	67	70
100代	66	68	70	64	68	70
200代	69	70	72	67	68	70

从第1组的实验结果可以看出,如果采用传统的信息素更新策略,那么无论是1GA+ACO还是3GA+ACO都很快早熟收敛于局部最优解;而在第2组实验中,由于采用基于重要解成分的信息素更新策略,1GA+ACO尽管可能早熟收敛,但能以较大的概率找到全局最优,而且即使早熟收敛,找到的局部最优解也比采用传统的信息素更新策略时更好。对于3GA+ACO,采用基于重要解成分的信息素更新策略后,每次都能找到最优解。

从表2的实验数据来看,尽管设定了迭代200代才停机,但实际上100代以内3GA+ACO就可以找到最优解。采用基于重要解成分的信息素更新策略,3GA+ACO迭代100代的平均时间为1.233秒,而采用传统的信息素更新策略时,3GA+ACO迭代200代的平均时间为1.318秒,在本实验中运行20次都找不到全局最优。

根据表3的实验数据,面对4阶欺骗问题,GA算法与传统的ACO算法的性能相当。对比表2和表3的实验数据可以看到,虽然都采用基于重要解成分的信息素更新策略,但3GA+ACO算法能够很快找到最优解,而ACO算法却找不到最优解。通过实验分析可以发现,在3GA+ACO算法中,初始信息素是基于GA产生初始信息素的过程重复运行3次得到,GA迭代100代后找到的局部最优解的适应值在66~70之间,而在单一的ACO算法中,由于没有基于GA产生初始信息素,初始解的适应值在44~49之间(如表3所列),显然前者的重要解成分比后者的重要解成分更能反映优质解的特点,而且多次运行基于GA产生初始信息素的过程,会使信息素反映多个局部最优解的特点,因此,在高质量的信息素指引下,3GA+ACO算法能够很快找到最优解。

结束语 蚁群优化算法工作机理就是选取好的候选解进行学习,获取启发性知识(信息素模型)来指导下一步的搜索。面对欺骗性问题,蚁群优化算法跟遗传算法一样,它的搜索过

(下转第236页)

持循环与条件选择,所以可以很简单地实现这些描述的方法。

4 实验与分析

将本文提出的算法应用于大规模医学、工业等数据,首先给出绘制图像,然后以表格形式列出绘制速度以及数据的规模,验证本文方法的效能。

实验条件:使用一台普通的 PC 作为平台,具体如下: Intel(R) Core(TM)2 CPU 6600@ 2.40GHZ 2.39GHZ 配备 GeForce 8800 GTX GPU,768 MB 显存。支持 DirectX4.0,整个系统在 VC++ 环境下运行,使用 HLSL 语言编写。

图 2 给出了绘制的结果,其中 A 是战斗机喷气发动机尾部喷气的三维数据,使用本文方法对其进行了体绘制;B 为一条冷冻鱼的 CT 扫描三维数据集的绘制结果;C 为 Teddy 熊的三维 CT 扫描数据集所绘制的结果;D 为一块云彩的三位扫描数据的体绘制图像。各个数据集的规模在表 1 中列出。

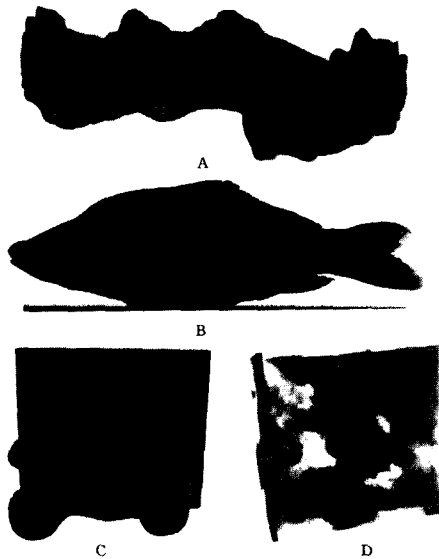


图 2 数据体绘制结果

从图 2 可以看出本文的方法可以很好地应用于多种类型的体数据,绘制结果较为理想。

从表 1 的结果可以看出,当数据集大小为 $512 \times 512 \times 512$ 时,本文的算法仍然可以在普通 PC 上以 16.7fps 的速度

进行绘制,这个速度足以满足实时交互绘制的需求。绘制结果图像和绘制速度都验证了本文所提方法的有效性。

表 1 绘制速度(帧/s)

数据集	A	B	C	D
数据规模	$256 \times 256 \times 256$	$512 \times 512 \times 512$	$128 \times 128 \times 62$	$128 \times 128 \times 128$
视窗尺寸	1264×958	1024×968	1264×958	1264×958
绘制速度(fps)	18.34	16.7	65.78	109.4

结束语 本文研究并实现了一个基于普通 PC 平台的大规模数据快速可视化算法。算法是基于三维纹理硬件的直接体绘制算法,并在第 3 节中详细描述了算法具体实现的步骤。实验结果证明了算法的有效性。算法可以应用于多种类型的体数据。在普通 PC 平台上就可以实现大规模数据快速可视化。

参考文献

- [1] 吴恩华. 图形处理器用于通用计算的技术、现状及其挑战[J]. 软件学报, 15(10):1493-1504
- [2] Lorensen W E, Cline H E. Marching Cube: A High Resolution 3D Surface Construction Algorithm[J]. Computer Graphics, 1987, 21(4)
- [3] Westover L. Footprint Evaluation for Volume Rendering[J]. Computer Graphics, 1990, 24(4):367-376
- [4] Mueller K, Shareef N, Huang J, et al. High quality splatting on rectilinear grids with efficient culling of occluded voxels[J]. IEEE Trans Vis and Comp Graph, 1999, 5(2):116-134
- [5] Tuy H K, et al. Direct 2-D Display of 3-D objects[J]. IEEE Computer Graphics and Applications, 1984, 4(10):29-33
- [6] Levoy M. Display of Surfaces from Volume Data[J]. IEEE Computer Graphics & Applications, 1988, 8(5):29-37
- [7] Cullip T, Neumann U. Accelerating volume reconstruction with 3D texture hardware[R]. UNC Technical Report. 1993
- [8] Cabral B, Cam N, Foran J. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware[C]// Proceedings of the 1994 Symposium on Volume Visualization. 1994:91-98
- [9] Kruger J, Westermann R. Acceleration techniques for GPU-based volume rendering[C]// Visualization, VIS 2003. IEEE. 2003:287-292

(上接第 205 页)

程往往也是被误导的。通过对优质候选解的重要成分进行识别,然后基于重要解成分更新信息素,使得信息素更能反映优质解的特点,消除欺骗吸引子的影响,最终找到全局最优解。

参考文献

- [1] Dorigo M, Socha K. An Introduction to Ant Colony Optimization [R]. IRIDIA/2006-10. Université Libre de Bruxelles, Belgium, 2006
- [2] Dorigo M, Stützle T. Ant Colony Optimization [M]. London: MIT Press, 2004
- [3] Stützle T, Hoos H. Max-min ant system[J]. Future Generation Computer Systems, 2000, 16(9):889-914
- [4] 朱庆保, 杨志军. 基于变异和动态信息素更新的蚁群优化算法

[J]. 软件学报, 2004, 15(2):185-192

- [5] 柯良军, 冯祖仁, 冯远静. 有限级信息素蚁群算法[J]. 自动化学报, 2006, 32(2):296-303
- [6] Ding J, Tang W, Wang L. Parallel Combination of Genetic Algorithm and Ant Algorithm Based on Dynamic K-Means Cluster [C]//Lecture Notes in Artificial Intelligence. Vol. 4114, Berlin: Springer, 2006:825-830
- [7] 丁建立, 陈增强, 袁著社. 遗传算法与蚂蚁算法的融合[J]. 计算机研究与发展, 2003, 40(9):1351-1356
- [8] Blum C, Dorigo M. Deception in Ant Colony Optimization[C]// Lecture Notes in Artificial Intelligence. Vol. 3172, Berlin: Springer, 2004:118-129
- [9] Eiben A E, Smith J E. Introduction to evolutionary computing [M]. Berlin Heidelberg: Springer-Verlag, 2003:191-193