

求解0-1背包问题的共同进化遗传算法^{*}

A Coevolutionary Genetic Algorithm for 0-1 Knapsack-Problem

刘娜 钟求喜

(国防科技大学计算机学院 长沙410073)

Abstract The efficacy and efficiency of conventional single-population-based genetic algorithm(SGA) decrease with the size of 0-1 Knapsack-Problem. A coevolutionary genetic algorithm(CGA) for 0-1 Knapsack-Problem is proposed in this paper, which includes the representation of solution, construction of feasible solution, converting illegal solution into feasible one, and fitness computing, etc. The simulation results show that the optimal/sub-optimal and convergence of CGA are better than that of SGA, and the algorithm is of practical use in engineering.

Keywords 0-1 Knapsack-problem, Genetic algorithms, Coevolutionary computation

0-1背包问题是一类组合优化问题,迄今已有40多年的研究历史^[1],可广泛应用于碎片收集、作业调度、资金预算和货物装箱等领域。0-1背包问题是一类 NP 问题,所以传统方法如持续松弛法、分枝-界限法、动态规划法和一些近似算法等等^[2~4],一般仅能获得问题的近似最优解。近年来,不少学者将稳健的遗传算法^[4,5]应用于0-1背包问题的求解^[6],在问题求解质量方面收到了较好的效果。但是,由于传统的单种群遗传算法中一个染色体编码结构代表了问题的一个完整可行解,因此可能导致对解的较好部分的利用可能被其它较差的部分所掩盖,且问题求解效率随着问题规模的增大而下降。针对上述不足,本文基于合作式共同进化计算模型^[7~9],将共同进化计算用于求解,提出一种求解0-1背包问题的共同进化遗传算法,以进一步提高问题的求解质量和算法效率。

1 问题描述

0-1背包问题可简单地描述如下: n 个物品组成的物品集 $U = \{u_1, u_2, \dots, u_n\}$ ($n > 1$), 其中 u_i 为第 i 个物品 ($1 \leq i \leq n$)。定义 $w(u_i)$ 为物品 u_i 的重量, $p(u_i)$ 为物品 u_i 的利润 ($1 \leq i \leq n$)。设背包的总容量为 W_{max} , 则0-1背包问题的优化目标为: 在不超过背包总容量的条件下, 使得背包装容的物品的利润之和最大, 即寻找 U 的一个子集 U' , 使

$$\sum_{u \in U'} p(u) = \max_{U' \subseteq U} \left\{ \sum_{u \in U'} p(u) \right\}$$

$$s. t. \quad \sum_{u \in U'} w(u) \leq W_{max}$$

2 共同进化遗传算法

共同进化在生物学上是指紧密相关物种的互补进化, 而共同进化计算则借鉴了生物界中共同进化机制, 是在现有进化计算基础上形成的一种解空间分离编码的动态搜索和优化算法。合作式共同进化计算模型中, 存在多个相关种群组成的种群集合(不妨设有 p 个种群), 每个种群中的个体代表问题解的一部分, 个体适应值与其对整个问题的贡献相关。通过多个种群间的相互合作来完成问题的求解。当前种群中个体与其它 $(p-1)$ 个种群的合作关系, 通常又是通过它与其它种群中代表的合作来体现的, 如, 将它与其它 $(p-1)$ 个种群中的最优个体合并成问题的一个完整解(“强强合作”), 来检验其对整个问题求解的贡献。

2.1 编码方式

根据0-1背包问题本身的特点, 从直观自然和便于理解的角度出发, 可直接采用二进制串作染色体的编码结构, 即种群中个体可表示为长度为 n 的二进制串, 记为 $X = (x_1, x_2, \dots, x_n)$, 其中 $x_i \in \{0, 1\}$ ($1 \leq i \leq n$)。对编码结构 X 解码时, 串 X 中各位的意义解释如下:

$$x_i = \begin{cases} 1, & \text{物品 } u_i \text{ 被选中} \\ 0, & \text{否则} \end{cases}$$

显然, 二进制串的编码方式具有完备性和非冗余性, 但不具备健全性, 即, 问题解空间的一点可以用唯一编码串表示, 但编码空间的点在问题解空间中不一

^{*} 本文受到国家自然科学基金资助(编号: 69903010, 60003002)、刘娜 硕士生, 研究方向为进化计算, 钟求喜 博士, 主要研究方向为网络计算与进化计算等。

定有其对应点,编码方式的不健全性导致了非法解产生的可能。非法解是指一个编码结构经解码后不是问题的一个可行解,即不满足问题的约束条件。编码串 $X = (x_1, x_2, \dots, x_n)$ 是非法解当且仅当下式成立:

$$\sum_{i=1}^n x_i \cdot w(u_i) > W_{\max}$$

当采用 p 个种群的共进化遗传算法时,则可将编码串 X 等长地划分成 p 段,每段对应一个种群。因此,第 i 个种群的个体编码长度 L_i 可计算如下:

$$L_i = \begin{cases} \lfloor n/p \rfloor, & 1 \leq i < n \\ n - (n-1) \cdot \lfloor n/p \rfloor, & i = n \end{cases}$$

其中, $\lfloor x \rfloor$ 返回一个不大于 x 的最大整数 ($x > 0$)。

2.2 非法解处理方式

非法解的产生主要有两种情况:第一,初始种群构造时产生非法解的染色体编码结构;第二,遗传操作(如交叉和变异算子)导致非法解的产生。第一种情况下,构造初始种群时可限定只产生可行解;第二种情况下,可考虑以下三种对非法解的处理方式:

方式1:设计专门的交叉算子和变异算子,以保证在可行解基础上经过遗传操作后产生的解仍然是可行解。

方式2:采用标准的单点交叉算子和位变异算子,当出现非法解时,则将该非法解丢失,再随机生成一可行解;

方式3:仍采用标准的单点交叉算子和位变异算子,当出现非法解时,则将该非法解转换成可行解。

在可行解基础上,按如下方式设计单点交叉算子和位变异算子可保证新解的可行性。设有可行解父本 $X_1 = (x_{1,1}, x_{1,2}, \dots, x_{1,n})$ 和 $X_2 = (x_{2,1}, x_{2,2}, \dots, x_{2,n})$, 并设交叉点为 $q (1 \leq q \leq n)$, 交叉操作后生成的新个体为 $X = (x_1, x_2, \dots, x_n)$ 和 $X' = (x'_1, x'_2, \dots, x'_n)$, 则它们满足以下条件:

$$x_i = \begin{cases} x_{1,i}, & i < q \\ x_{2,i}, & i \geq q \text{ 且 } (\sum_{j=1}^{i-1} x_j \cdot w(u_j) + x_{2,i} \cdot w(u_i)) \leq W_{\max} \\ 0, & \text{否则} \end{cases}$$

$$x'_i = \begin{cases} x_{2,i}, & i < q \\ x_{1,i}, & i \geq q \text{ 且 } (\sum_{j=1}^{i-1} x'_j \cdot w(u_j) + x_{1,i} \cdot w(u_i)) \leq W_{\max} \\ 0, & \text{否则} \end{cases}$$

同样地,设父本 $X = (x_1, x_2, \dots, x_q, \dots, x_n)$ 在 $q (1 \leq q \leq n)$ 位位变异操作得到新个体 $X' = (x_1, x_2, \dots, x'_q, \dots, x_n)$, 则 x'_q 的取值满足以下条件:

$$x'_q = \begin{cases} 1 - x_q, & \sum_{j=1, j \neq q}^n x_j \cdot w(u_j) + (1 - x_q) \cdot w(u_q) \leq W_{\max} \\ 0, & \text{否则} \end{cases}$$

从物品集 U 中随机构造可行解 $X = (x_1, x_2, \dots, x_n)$ 的算法过程如下:

步骤1:初始化 X : for ($i=1; i \leq n; i++$) $x_i=0$;

步骤2:从物品集 U 中随机选择一个物品 u_i ;

步骤3:如果条件 $(\sum_{j=1, j \neq i}^n x_j \cdot w(u_j) + w(u_i)) \leq W_{\max}$ 成立,则令 $x_i=1, U=U-\{u_i\}$, 并转步骤2; 否则算法终止。

多次应用随机构造可行解算法,就可构造出一定群体规模的初始种群。若采用 p 个种群的共进化遗传算法时,则可将可行解等长地划分成 p 段,每段指派给对应的种群。

非法解处理方式3是将非法解转换成合法解。设有非法解 $X = (x_1, x_2, \dots, x_n)$, 将其转换成一合法解 $X' = (x'_1, x'_2, \dots, x'_n)$, 则 X' 中各位满足以下条件:

$$x'_i = \begin{cases} x_i, & \sum_{j=1}^{i-1} x'_j \cdot w(u_j) + x_i \cdot w(u_i) \leq W_{\max} \\ 0, & \text{否则} \end{cases}$$

2.3 适应值计算

合作式共同进化计算模型中,为了评价种群中个体对整个问题的贡献,一般是将其与其它种群选择出的代表合并成一个完整解,然后根据该完整解的优劣计算个体的适应值。代表的选择方式有多种^[9], 本文选择种群中最优的个体作为合作代表。

p 个种群集的第 i 个种群的第 j 个个体设为 $X_{i,j} = (x_{i,j,1}, x_{i,j,2}, \dots, x_{i,j,L_i})$, 其它种群的代表分别为 $X'_k = (x_{k,1}, x_{k,2}, \dots, x_{k,L_k}) (1 \leq k \leq p, k \neq i)$ 。将个体与所有的代表合并成一个完整解,记为 $X = (X'_1, X'_2, \dots, X_{i,j}, \dots, X'_p) = (x_{1,1}, x_{1,2}, \dots, x_{1,L_1}, x_{2,1}, x_{2,2}, \dots, x_{2,L_2}, \dots, x_{i,j,1}, x_{i,j,2}, \dots, x_{i,j,L_i}, \dots, x_{p,1}, x_{p,2}, \dots, x_{p,L_p}) = (x_1, x_2, \dots, x_n)$ 。于是可将完整解 X 的利润作为个体 $X_{i,j}$ 的适应值,即个体 $X_{i,j}$ 的适应值 $f(X_{i,j})$ 为:

$$f(X_{i,j}) = \sum_{k=1}^n (x_k \cdot p(u_k))$$

2.4 共同进化遗传算法

求解0-1背包问题的共同进化遗传算法的基本框架如下:

步骤1:构造初始种群集 $Pop = (Pop_i)$, 并随机取一可行解作为初始的合作代表;

($1 \leq i \leq p, p$ 为种群个数, Pop_i 为第 i 个子种群)

步骤2:各种群 Pop_i 依次执行以下步骤: ($1 \leq i \leq p$)

步骤2.1:将种群 Pop_i 的个体 $X_{i,j}$ 与合作代表合并成一个完整解,如果该完整解是非法解,则将个体 $X_{i,j}$ 转换成合法; ($1 \leq j \leq p, p_i$ 为子种群 Pop_i 的群

体规模)

步骤2.2: 计算个体 $X_{i,j}$ 的适应值 $f(X_{i,j})$, 采用精英策略保存最好解, 并将最好解作为该种群的合作代表; ($1 \leq j \leq p_i$)

步骤2.3: 执行“轮盘赌”的按比例选择操作;

步骤2.4: 执行单点交叉操作;

步骤2.5: 执行位变异操作;

步骤3: 如果算法终止条件不满足, 转步骤2;

步骤4: 输出算法求得的最好解, 算法终止。

上述算法框架中, 各种群采用了同步执行的方式, 即各种群依次进化一代(也称整个算法进化一代)。初始种群中的个体采用可行解的随机构造算法生成, 采用方式3来处理当前种群中单点交叉操作和位变异操作后可能造成的非法解。

3 仿真实验

文[10]的理论分析表明, 基于问题分解的共同进化遗传算法的求解效率高于传统单种群遗传算法, 且问题分解粒度越小(相应种群数越多), 算法求解效率愈高。同时, 仿真实验又是进化计算中检验算法性能的主要手段之一。为检验共同进化遗传算法求解0-1背包问题的有效性, 分别对不同规模的问题进行仿真实验。

算法主要控制参数这样设置: 交叉概率为0.9; 变异概率为0.05; 最多迭代代数300代; 种群规模为20。

仿真数据中的物品重量和利润都是随机生成的, 且取值范围为1~50。表1是三种非法解处理方式对不同规模问题求解的对比实验结果。图1到图3则是算法对不同规模问题求解的静态性能收敛曲线(算法在不同代时所找到的最好解)。图中五条曲线从上到下依次对应的种群数为5个、4个、3个、2个和1个(种群数为1个时, 共同进化遗传算法则退化为单种群遗传算法)。所有图表中的结果均是算法10次独立运行的平均值。

表1的数据说明对非法解的三种处理方式中, 将非法解转换成合法解是最为理想的方式。图1到图3的算法静态性能曲线表明: 算法的收敛速度和求解质量随着种群数的增加而加快和提高。值得注意的是, 当种群数增多时, 由于每个种群都要进行选择、遗传操作、适应值计算和对非法解的处理, 因此算法进化一代所需的计算时间也增加了。当计算时间固定时, 表2给出了不同种群数的共同进化遗传算法在15秒时所进化的代数数和找到的最优解(仿真PC平台: P120/24M/Microsoft Windows 95)。表2的结果表明, 即使在相同计算时间的条件下, 共同进化遗传算法所求得的最好解也比传统单种群遗传算法的更好。

表1 非法解处理方式: 仿真实验对比结果

(n, W_{max})	(50, 800)	(100, 1500)	(200, 3500)	(500, 7500)	(1000, 12000)
方式1	1213.7	1955.4	3946.2	8149.4	13770.4
方式2	1217.1	1894.8	4021.1	8457.3	14026.7
方式3	1228.8	1934.2	4026.8	8500.2	14371.1

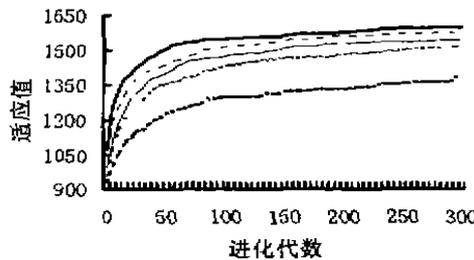


图1 仿真结果(n=100, $W_{max}=800$)

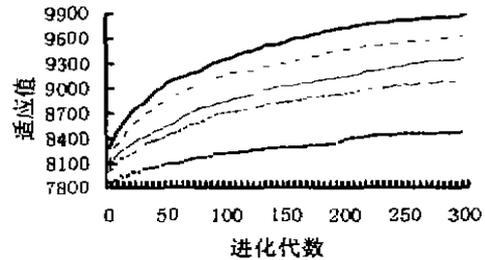


图2 仿真结果(n=500, $W_{max}=7500$)

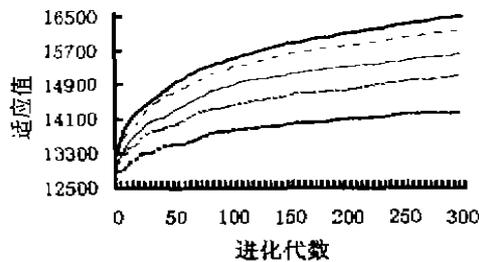


图3 仿真结果(n=1000, $W_{max}=12000$)

表2 仿真结果(n=500, $W_{max}=7500$)

种群数	计算结果	进化代数
1	8848	326
2	8940	271
3	9232	244
4	9537	218
5	9601	198

结束语 本文基于合作式共同进化计算模型,提出一种求解0-1背包问题的共同进化遗传算法。通过仿真实验比较了对非法解三种处理方式的优劣,以及不同种群数下共同进化遗传算法对不同规模问题的求解性能。研究表明:合作式共同进化遗传算法的收敛速度和问题求解质量均优于传统单种群遗传算法。本文中种群的划分采用的是手工方式,进一步工作是研究问题的自动分解方法和针对具体问题的有效合作方式,以及对0-1背包问题的扩充问题的求解,以期应用于实际工程中。

致谢 感谢博士后流动站的谢涛副教授在课题研究过程中给予的中肯建议和无私的帮助,也感谢博士生荔建琦同学的有益讨论。

参考文献

- 1 Martello S, Toth P. Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons Ltd., Chichester, England, 1990
- 2 Martello S, Toth P. An upper bound for the zero-one knapsack problem and a branch and bound algorithm. Eu-

ropean Journal of Operational Research, 1977(1): 169~175

- 3 Andonov, Rajopadhye. A Sparse Knapsack Algorithm and its Synthesis. In: Int Conf. on Application-Specific Array Processors (ASAP-94). San Francisco, CA, IEEE Press, 1994. 302~313
- 4 Holland J H. Adaptation in Natural and Artificial Systems. Ann Arbor: The University of Michigan Press, 1975
- 5 Goldberg D E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc. 1989
- 6 Dolan A. A general GA toolkit implemented in Java. Available at: Http://www.aridolan.com/ga/gaa/gaa.html
- 7 Potter M A. The Design and Analysis of a Computational Model of Cooperative Coevolution. [PhD. Dissertation]. George Mason University, 1997
- 8 Paredis J. Coevolutionary Algorithms. The Handbook of Evolutionary Computation. In: B. T. Fogel, D. Michalewicz, eds. Oxford University Press, 1998
- 9 钟求喜. 网络计算中任务分配中调度的遗传算法研究. [博士学位论文]. 国防科技大学研究生院
- 10 谢涛, 陈火旺. 基于函数分解的可伸缩进化算法. 自然科学进展, 2001

(上接第111页)

类图像。

根据假定(c): $p(x|\omega_i)$ 采用混合高斯密度函数, 其中的参数由期望最大(EM)算法得到。

统计分类是一种理论上理想的分类方法, 由于利用了最大似然决策作为图像距离的度量方法, 用 Bayes 判决方法构造的分类器在性能上优于采用欧氏距离和聚类等几何划分的方法。然而利用最大似然估计对于一些小概率事件不可避免地造成判断错误, 影响分类结果的准确性。在实际应用中, 条件概率的精确估计是难以解决的问题, 图像在特征空间中分布非常复杂, 对于不同用户的要求, 图像的分类要求也不一致, 造成条件概率估计不准确。对于特定的图像库, 经过多次训练可能找到较好的条件概率分布, 但不同类型的图像库间不存在通用的条件分布, 因此很难实现通用的检索。

结论 相关反馈技术是在图像检索领域中经常采用的重要方法之一。本文具体介绍了近几年来一些主要的相关反馈技术模型和算法, 可以作为借鉴, 在它们的基础上作更进一步研究。

人的图像理解过程往往是基于图像语义的, 因而图像的各种特征和语义之间的联系以及人们如何评价图像之间的相似度是图像检索中的难点。每幅图像都有丰富的内容, 任何特征均不能准确地描述人对图像的认识。相关反馈技术的目标就是从系统与人的交互

过程中学习, 让系统获得人对图像理解的信息, 使系统模拟人类观察和理解图像的方式, 提高图像检索的性能。我们认为图像检索领域发展的趋势之一是利用相关反馈过程提取用户查询目标图像的语义, 找到图像特征与语义之间的对应关系, 并对特征进行自动选取, 滤除与检索目标语义无关的特征。并且根据用户对样本图像相似性的评价找到最佳特征空间和距离度量方法, 在此空间中, 特征点间的距离能很好地反映人对不同特征点对应的图像间差异的感受。

参考文献

- 1 Rui Y, Huang T S. A Novel Relevance Feedback Technique in Image Retrieval. In: ACM MM, Orlando, Florida, October 30—November 5, 1999
- 2 Rui Y, et al. Relevance feedback: a power tool for interactive content based image retrieval. IEEE Trans. Circuits and systems for video technology, 1988, 8(5): 644~655
- 3 Lee C, Ma W Y, Zhang H J. Information Embedding Based on User's Relevance Feedback for Image Retrieval. [Technical Report]. HP Labs, 1998
- 4 Vasconcelos N, Lippman A. Learning From User Feedback in Image Retrieval System. NIPS' 99, Denver, Colorado, 1999
- 5 Vasconcelos N, Lippman A. Bayesian Representations and Learning Mechanisms for Content-Based Image Retrieval. SPIE Storage and Retrieval for Media Databases 2000, San Jose, California, 2000. A short version appears in Wkshp on CATVL, CVPR' 00, South Carolina, 2000