

文章编号:1007-6735(2011)02-0163-05

多目标 0-1 规划的元胞竞争决策算法

熊小华^{1,2}, 马良¹, 宁爱兵¹

(1. 上海理工大学 管理学院, 上海 200093; 2. 上海第二工业大学 计算机与信息学院, 上海 201209)

摘要: 将元胞演化规则与竞争决策算法相结合, 提出了一种求解多目标 0-1 规划问题的元胞竞争决策算法. 大量数据测试和验证表明, 该算法能有效提高非劣解的分布性和多样性.

关键词: 竞争决策算法; 元胞自动机; 多目标; 0-1 规划

中图分类号: O 223 **文献标志码:** A

Cellular competitive decision algorithm for multi-objective 0-1 programming problem

XIONG Xiao-hua^{1,2}, MA Liang¹, NING Ai-bing¹

(1. Business School, University of Shanghai for Science and Technology, Shanghai 200093, China;

2. College of Computer and Information, Shanghai Second Polytechnic University, Shanghai 201209, China)

Abstract: Combining the concepts of cellular automata with competitive decision algorithm, a cellular competitive decision algorithm for multi-objectives 0-1 programming problem was presented. The algorithm was then used to solve many instances of multi-objectives 0-1 programming and the computational results show it can improve effectively the diversity and distribution of Pareto optimal set.

Key words: competitive decision algorithm; cellular automata; multi-objectives; 0-1 programming

多目标 0-1 规划是以一维背包问题的 0-1 整数规划为原型发展起来的多个约束、多个目标的最优化问题, 在投资决策、经济预算以及资源分配等问题中有着广泛的应用. 这种具有多个约束、多个目标的组合优化问题, 统称为多目标优化问题 (MOP)^[1-2]. 针对多个彼此冲突的目标, 如何获取这些问题的最优解, 一直都是科学界和工程界关注的焦点问题. 由于多目标意义下, 使得所有目标都达到最佳的最优

解往往并不存在, 一般所要求的都是一组相互之间无法区分优劣的非劣解 (non-dominated solution), 也称有效解或 Pareto 解. 多目标优化问题的 Pareto 解只是一个可以接受的非劣解, 并且大多数的 Pareto 解的个数很多, 甚至是无穷大^[2].

0-1 整数规划属于 NP (非确定多项式) 完备问题类, 是否存在有效算法尚不可知, 已有的精确算法仅能解决小规模问题, 而启发式算法又不能保证得

收稿日期: 2011-02-25

项目基金: 国家自然科学基金资助项目 (70871081); 上海市重点学科建设资助项目 (S30504).

作者简介: 熊小华 (1978—), 女, 博士研究生. 研究方向: 算法设计、系统工程. E-mail: xiong_xiao_hua@163.com;

马良 (联系人), 男, 教授. 研究方向: 算法设计、系统工程. E-mail: maliang@usst.edu.cn

到的解是最优的. 多目标 0-1 规划问题相比单目标 0-1 规划来说, 更增加了问题求解的难度. 对于多目标的组合优化问题, 国内外的有关研究较少, 尤其缺乏实用算法. 本文将对此作一些探索性的研究, 为这类离散型的多目标优化难题提供若干解决手段, 对这些模型的实际应用奠定方法和技术上的基础. 本文将竞争决策算法原理与元胞自动机的演化规则相结合来求解多目标 0-1 规划问题, 更好地发挥竞争决策算法的优势, 既能达到 Pareto 前沿, 又能提高解的分布性和多样性.

1 竞争决策算法简介

竞争决策算法(competitive decision algorithm, 简称 CDA)是近几年来提出的一种求解组合优化难题的新型算法^[3-8], 算法的原理详见文献[5]. 自然界中的竞争和决策都是在一定竞争规则下, 在竞争者的实力、竞争者和环境间的关系、多个竞争者实力的差距和初始竞争状态等多种因素的共同作用下, 经过多次竞争和决策后, 使不同的竞争者分别占有一定的资源而达到一种新的竞争状态, 只要新的竞争状态优于初始竞争状态, 就能达到优化的目的. CDA 算法吸收了达尔文“优胜劣汰”的进化思想以及演化博弈论中有限理性竞争者的思想, 通过构造一个或多个具有有限理性的竞争者参与到一个或多个资源的竞争过程中, 通过优胜劣汰的原则使一部分竞争者获得资源而增加实力, 一部分竞争者失去资源削弱实力甚至消亡. 当算法通过竞争不能获得更优的结果时, 通过资源交换使算法进入下一轮的竞争. 在理论方面, 现在已给出了竞争决策算法的基本概念和通用流程; 在应用方面, 已利用其通用流程给出了车辆路径问题、度约束最小生成树、旅行商问题等 NP 难题的算法并编程实现.

2 元胞自动机简介

元胞自动机(cellular automata, 简称 CA)最早由冯·诺依曼提出, 沃尔夫勒姆等^[9-10]将动力系统方法、计算理论及形式化语言方法引入元胞自动机的研究中, 促进其广泛应用. CA 是一个时间和空间都离散的动力学系统, 由元胞、状态、邻居和规则这 4 部分组成. 按一定拓扑结构分布的元胞仅取有限的离散状态, 依据确定的局部规则进行更新, 大量元胞通过简单的相互作用而构成动态系统的演化. 若

元胞的状态数为 k , 邻居数为 n , 则元胞的可能状态数为 k^{kn} . 元胞在邻域范围内具有多种变化形态, 这增加了种群的范围, 提高了解的搜索空间, 可以避免算法陷入局部最优的状况, 这也是将元胞自动机引入竞争决策算法的原因所在.

3 多目标 0-1 规划的元胞竞争决策算法

3.1 问题介绍

本文考虑如下形式的多目标 0-1 规划问题

$$\begin{aligned} \max Z = & \left\{ z_1 = \sum_{j=0}^n c_j^1 x_j, z_2 = \right. \\ & \left. \sum_{j=0}^n c_j^2 x_j, \dots, z_k = \sum_{j=0}^n c_j^k x_j \right\} \\ \text{s. t. } & \begin{cases} \sum_{j=0}^n a_{ij} x_j \leq b_i, & i = 1, 2, \dots, m \\ x_j \in \{0, 1\}, & j = 1, 2, \dots, n \end{cases} \end{aligned}$$

式中, z_i 为第 i 个目标的值; m 为约束的个数; n 为决策变量的个数; k 为目标的个数.

3.2 算法原理

具有 n 个决策变量、 m 个约束条件以及 k 个目标的多目标 0-1 规划问题, 将 n 个决策变量看成竞争者争夺的资源, 将 k 个目标看成非虚拟竞争者 A, 另一个竞争者是虚拟竞争者 N. 竞争开始时所有的资源为 N 所占有, A 不占有任何资源. 在竞争过程中, A 按照竞争力函数和决策规则占有决策变量资源, 在竞争的过程中要保证满足所有的 m 个约束. 竞争结束后 A 占有的决策变量即是值为 1 的决策变量.

3.3 基本符号及含义

为了方便描述, 现介绍本文采用的符号.

$b(i)$: 第 i 个约束条件的资源变量, $i = 1, 2, \dots, m$;

$a(i, j)$: 约束条件中的系数, 表明决策变量 x_j 在约束 i 下的系数的值, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$;

$c(o, j)$: 目标函数中的系数, 表明决策变量 x_j 在目标 o 下的目标值, $o = 1, 2, \dots, k$, $j = 1, 2, \dots, n$;

$\text{profit}(j, o)$: 决策变量 x_j 在目标 o 下的收益, $\text{profit}(j, o) = c(o, j)$;

$\text{min_profit}(j)$: 表示决策变量 x_j 在所有目标下

的最小收益;

$\max_profit(j)$:表示决策变量 x_j 在所有目标下的最大收益;

$\text{diff_profit}(j)$: $\text{diff_profit}(j) = \max_profit(j) - \min_profit(j)$;

$\text{sum_profit}(j)$: $\text{sum_profit}(j) = \sum_{o=1}^m \text{profit}(j, o)$;

$\text{density}(j, o)$:决策变量 x_j 在目标 o 下的价值密度,其数学描述为, $\text{density}(j, o) = \text{profit}(j, o) / \sum_{i=0}^m a(i, j)$;

$\min_density(j)$:表示决策变量 x_j 在所有目标下的最小价值密度;

$\max_density(j)$:表示决策变量 x_j 在所有目标下的最大价值密度;

$\text{diff_density}(j)$: $\text{diff_density}(j) = \max_density(j) - \min_density(j)$;

$\text{sum_density}(j)$: $\text{sum_density}(j) = \sum_{o=1}^k \text{density}(j, o)$;

select_items :值为1的决策变量的集合;

best_solutions :保留的 Pareto 最优解的集合;

$\text{free_v}(i)$:在确定 $\text{select_items}(j)$ 中的决策变量后,各个约束条件剩余的资源变量的值,其数学描述为, $\text{free_v}(i) = b(i) - \sum_{j \in \text{select_items}} a(i, j)$;

$\text{power}(j)$:竞争者 A 对决策变量 x_j 的竞争力.

3.4 竞争力函数、决策函数和资源交换规则

3.4.1 竞争力函数

本算法采用6个竞争力函数,现描述所采用的竞争力函数的基本思想.在满足 m 个约束条件的前提下: a. $\text{sum_profit}(j)$ 值越大的决策变量竞争力函数越大; b. $\text{min_profit}(j)$ 值越大的决策变量竞争力函数越大; c. $-\text{diff_profit}(j)$ 值越大的决策变量竞争力函数越大; d. $\text{sum_density}(j)$ 值越大的决策变量竞争力函数越大; e. $\text{min_density}(j)$ 值越大的决策变量竞争力函数越大; f. $-\text{diff_density}(j)$ 值越大的决策变量竞争力函数越大.限于篇幅,仅给出1个竞争力函数的定义,其他5个竞争力函数定义类似.

$$\text{power}(j) = \begin{cases} \text{sum_profit}(j) & j \notin \text{select_items}, \\ a(i, j) \leq \text{free_v}(i), \\ i = 1, 2, \dots, m \\ -\infty & \text{否则} \end{cases}$$

3.4.2 决策函数

本文只采用1个决策函数,即选中竞争力函数 $\text{power}(j)$ 值最大的决策变量.若两个决策变量的 $\text{power}(j)$ 值相同时,则选择编号小的决策变量.

3.4.3 资源交换规则

在元胞空间里,元胞按照演化规则有很多种变化,有助于保持种群的多样性.在多目标问题求解中,为了提高 Pareto 解集的多样性和分布性,将元胞自动机的演化规则用作竞争决策算法的资源交换规则,以使竞争进入下一个竞争均衡状态.

定义1 集合 $X = (x_1, x_2, \dots, x_n)$, $x_i \in \{0, 1\}$. X 中 x_i 的任意取值的组合构成的集合为元胞空间,则元胞空间可以定义为 $L = \{ \text{Cell}X = (x_1, x_2, \dots, x_n) \mid x_i \in \{0, 1\} \}$, $\text{Cell}X$ 表示一个元胞.

定义2 元胞邻居采用扩展 Moore 邻居类型, $N = \{ \text{Cell}Y \mid \text{diff}(\text{Cell}Y - \text{Cell}X) \leq r, \text{Cell}X, \text{Cell}Y \in L \}$, 其中, $\text{diff}(\text{Cell}Y - \text{Cell}X)$ 为两个组合的差异.若无差异,为0;有差异时,最大为2. r 表示差异的程度,本文中 r 取2.

定义3 Pareto 解集过滤器,用来存放运行时产生的 Pareto 解.对于不违反约束的解,若该解支配过滤器中某个解,则将该解加入过滤器,同时删除受支配解;若过滤器中某个解支配该解,则放弃该解.过滤器的大小可以事先设定好,过滤器的大小限制了允许保存的最多的 Pareto 解的个数.

定义4 元胞演化规则:将已经求得的解看成中心元胞 $\text{Cell}X$.按照元胞邻居的定义,计算其邻居 $\text{Cell}Y$ ($\text{Cell}Y \in N$) 的各个目标值.将 $\text{Cell}Y$ 的各目标值通过 Pareto 解集过滤器进行检测.

3.5 算法流程

现给出多目标0-1规划的元胞竞争决策算法流程.

Step 1 初始化.

$\text{p_count} = 6$; //竞争力函数的个数
 $\text{d_count} = 1$; //决策函数的个数
 $\text{la_count} = 1$; //初始格局的个数
 计算每个决策变量 x_j 的 $\text{sum_profit}(j)$, $\text{min_profit}(j)$, $\text{diff_profit}(j)$, $\text{sum_density}(j)$, $\text{min_density}(j)$ 和 $\text{diff_density}(j)$;
 $\text{select_items} = \{ \}$;
 $\text{free_v}(i) = b(i)$; $i = 1, 2, \dots, m$.

Step 2 竞争、决策及资源交换.

for $p = 1$ to p_count //竞争力函数个数循环
 for $d = 1$ to d_count //决策函数个数循环

for $la = 1$ to la_count //初始格局个数循环
根据第 p 个竞争力函数计算 $power(j)$.

Step 2.1 本轮竞争阶段 1:资源分配阶段.

repeat

按照决策函数的定义,将第 p 个竞争力函数 $power(j)$ 值最大的决策变量选中,并按如下方式修改:

$select_items = select_items + \{j\}$; $free_v(i) = free_v(i) - a(i, j)$; ($i = 1, 2, \dots, m$);

根据第 p 个竞争力函数重新计算 $power(j)$;
until(所有的 $power(j) = -\infty$)

使用 Pareto 过滤器检测阶段 1 取得的解.

Step 2.2 本轮竞争阶段 2:资源交换阶段.

以本轮求得的解为中心元胞,按照元胞邻居及演化规则的定义,在邻居范围内演化,替换当前非劣解或添加新的非劣解.

Step 3 输出竞争得到的结果.

3.6 时间复杂度分析

在 Step 1 中分别对 $sum_profit(j)$, $min_profit(j)$, $diff_profit(j)$, $sum_density(j)$, $min_density(j)$ 和 $diff_density(j)$ 采用折半插入排序进行降序排序,折半插入排序的时间复杂度为 $O(n \log^2 n)$; Step 2.1 最坏情况下时间复杂度为 $O(nm)$; Step 2.2 中寻找元胞邻居的时间复杂度为 $O(n^2)$, 对每个邻居与保存的最优解的比较次数为 $O(k * \text{Pareto 过滤器大小})$, 则 Step 2 最坏情况下的时间复杂度为 $O(\text{Pareto 过滤器大小} * k * n^2)$. 由此可知整个算法的复杂度为 $O(\text{Pareto 过滤器大小} * k * n^2)$.

4 数值算例

为验证算法的有效性,采用 Delphi 7.0 在 PC 机上实现了该算法.本算法对大量的算例进行求解,并与其他算法的结果进行比较,总体效果良好,限于篇幅,现给出其中的 4 个算例的比较结果.

算例 1 算例 1 取自文献[11-12].

$$\begin{aligned} \max z_1 &= 3x_1 + 6x_2 + 5x_3 - 2x_4 + 3x_5 \\ \max z_2 &= 6x_1 + 7x_2 + 4x_3 + 3x_4 - 8x_5 \\ \max z_3 &= 5x_1 - 3x_2 + 8x_3 - 4x_4 + 3x_5 \\ \text{s. t. } &\begin{cases} -2x_1 + 3x_2 + 8x_3 - x_4 + 5x_5 \leq 13 \\ 6x_1 + 2x_2 + 4x_3 + 4x_4 - 3x_5 \leq 15 \\ 4x_1 - 2x_2 + 6x_3 - 2x_4 + x_5 \leq 11 \\ x_j \in \{0,1\}, j = 1,2,\dots,5 \end{cases} \end{aligned}$$

本算法与其他算法运行的结果如表 1 所示,从实验结果可以看出,元胞竞争决策算法所求 Pareto 解的个数多于 One-stage 算法和 DEA 算法.

表 1 算例 1 的计算结果

Tab.1 Results of example 1

元胞竞争决策算法	One-stage 算法	DEA 算法
(8,10,13)	(8,10,13)	
(9,5,12)	(11,2,16)	(11,2,16)
(11,2,16)	(14,17,10)	(14,17,10)
(14,17,10)	(15,12,9)	
(15,12,9)		

算例 2 算例 2 取自文献[13-14].

$$\max z_1 = 4x_3 + 5x_4 + 7x_7 + 6x_8 + 5x_9 + 6x_{10}$$

$$\max z_2 = 9x_1 + 4x_2 + 6x_5 + 9x_6 +$$

$$6x_7 + 5x_8 + 6x_9 + 3x_{10}$$

$$\max z_3 = 6x_1 + 3x_2 + 2x_3 + 8x_4 + 7x_5 + 2x_8$$

$$\text{s. t. } \begin{cases} 3x_2 + 4x_3 + 8x_4 + 4x_5 + 7x_6 + 6x_7 + \\ 5x_8 + 2x_9 + 6x_{10} \leq 37 \\ 5x_1 + 3x_2 + 6x_4 + 8x_6 + 5x_7 + 6x_8 + \\ 3x_9 + 5x_{10} \leq 24 \\ 2x_1 + 2x_2 + 7x_3 + 4x_4 + 8x_5 + 9x_6 + \\ 9x_7 + 3x_8 + 7x_9 + 5x_{10} \leq 43 \\ 2x_1 + 3x_2 + 3x_3 + 6x_4 + 3x_5 + 8x_7 + \\ 4x_8 + 6x_9 \leq 31 \\ x_j \in \{0,1\}, j = 1,2,\dots,10 \end{cases}$$

本算法与其他算法运行的结果如表 2 所示,从结果可以看出元胞竞争决策算法相比于元胞蚁群算法、蚁群算法和遗传算法,可以找到更多的 Pareto 解.

表 2 算例 2 的计算结果

Tab.2 Results of example 2

元胞竞争决策算法	元胞蚁群算法	蚁群算法	遗传算法
(12,40,16)	(12,40,16)	(12,40,16)	(20,30,28)
(15,37,18)	(20,30,28)	(20,30,28)	(21,23,25)
(20,30,28)	(21,31,26)	(21,23,25)	(21,29,17)
(21,31,26)	(22,28,26)	(21,29,17)	(22,21,22)
(22,28,26)	(22,36,20)	(22,21,22)	
(22,36,20)	(23,33,20)	(28,20,19)	
(23,33,20)	(28,20,19)	(28,27,17)	
(27,30,23)	(28,35,17)	(28,30,14)	
(28,20,19)			
(28,35,17)			

算例 3 算例 3 取自文献[14-15].

$$\begin{aligned} \min \quad & z_1 = 4x_1 + 5x_2 + 3.5x_3 + 4x_4 \\ \max \quad & z_2 = 3x_1 + 4x_2 + 2.5x_3 + 3.5x_4 \\ \text{s. t.} \quad & \begin{cases} 4x_1 + 5x_2 + 3.5x_3 + 4x_4 \leq 10 \\ x_i(x_i - 1) = 0, \quad i = 1, 2, 3, 4 \end{cases} \end{aligned}$$

本算法与其他算法运行的结果如表 3 所示,相比

于文献[14-15]的算法,可以发现更多的 Pareto 解.
算例 4 $k=2, n=100, m=2$, 其约束和目标函数由于决策变量个数较多不便列出,共求得 160 个 Pareto 解,其所求解集如图 1 所示.

表 3 算例 3 的计算结果
Tab.3 Results of example 3

元胞竞争决策算法	元胞蚁群算法	分目标乘除法	功效系数法	评价函数法
(0,0);(3.5,2.5);(4.3,5);(5,4) (7.5,6);(8.6,5);(9,7.5)	(0,0); (4.3,5); (5,4); (7.5,6);(8.6,5); (9,7.5)	(9,7.5)	(8,6.5)	(9,7.5)

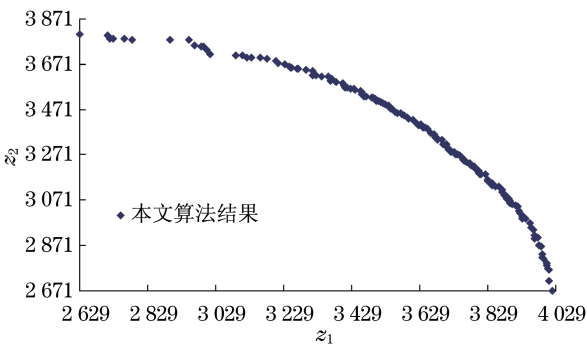


图 1 算例 4 的 Pareto 解集
Fig.1 Pareto set of example 4

通过以上测试发现,对于较小规模的问题,可以求出全部 Pareto 解;对于较大规模的问题,Pareto 解分布比较稠密,且形成一个明显的 Pareto 前沿.

5 结束语

多目标 0-1 规划问题是很多问题的原始模型,在工程实践与科学研究中都具有非常重要的意义.但是,由于其 NP 难题以及多目标问题本质上多个目标相互冲突的特性,限制了精确算法对较大规模问题的求解.竞争决策算法是一种能广泛应用于求解各类组合优化难题的新型寻优算法,其通用性和实用性都比较强.本文将元胞自动机原理与竞争决策算法相结合来求解多目标 0-1 规划问题,提高了 Pareto 解集的分布性及多样性,因此,求解的速度与效果都比较好.

参考文献:

[1] 雷德明,严新平.多目标智能优化算法[M].北京:科技技术出版社,2009.
[2] 崔逊学.多目标进化算法及其应用[M].北京:国防工业出版社,2006.

[3] 宁爱兵,马良.竞争决策算法及其在车辆路径问题中的应用[J].管理科学学报,2005,8(6):10-18.
[4] 宁爱兵,马良.度约束最小生成树(DCMST)的竞争决策算法[J].系统工程学报,2005,20(6):630-634.
[5] 宁爱兵,马良,熊小华.竞争决策算法原理及其应用[J].上海理工大学学报,2008,30(4):369-373.
[6] 熊小华,郭文夷,宁爱兵.具有偏好选择的多目标 TSP 竞争决策算法[J].上海第二工业大学学报,2005,22(1):6-12.
[7] 宁爱兵,马良.最小比率旅行商(MRTSP)问题竞争决策算法[J].计算机工程与应用,2005,41(11):30-32.
[8] 宁爱兵,马良.基于快速下界估算的瓶颈旅行商问题竞争决策算法[J].上海理工大学学报,2005,27(3):223-228.
[9] WOLFRAN S.Theory and Application of Cellular Automata [M].Singapore:The World Scientific Publishing Company Limitd,1986.
[10] 朱刚,马良.函数优化的元胞蚂蚁算法[J].系统工程学报,2007,22(3):305-308.
[11] LIU Fuh-hwa Franklin, HUANG Chueng-chiu, YEN Yu-lee. Using DEA to obtain efficient solutions for multi-objctive 0-1 linear programs[J]. European Journal of Operational Research,2000,126(1):51-68.
[12] JAHANSHALOO G R,HOSSEINZADEH F,SHOJA N, et al.A method for generating all efficient solutions of 0-1 multi-objective linear programming problem[J]. Applied Mathematics and Computation,2005,169(2):874-886.
[13] 崔雪丽,马良.多目标 0-1 规划的蚂蚁算法[J].计算机应用与软件,2007,24(7):23-24.
[14] 刘勇,马良,许燕秋.多目标 0-1 规划问题的元胞蚁群优化算法[J].系统工程,2009,27(2):119-122.
[15] 冯爱芬,杨森.一类农业生产问题的多目标规划模型及其解法[J].黄冈师范学院学报,2004,24(3):32-34.