

一个基于启发式经验的立体布局进化策略¹

曹先彬 庄镇泉*

(中国科学技术大学计算机科学系 合肥 230026)

*(中国科学技术大学电子技术部 合肥 230026)

摘要 目前求解布局问题只能求得近似解, 三维布局则更加复杂。本文提出一种基于启发式经验、采用遗传算法求解布局问题的新方法, 此方法对任意维布局问题在求解策略上是一致的。实验表明, 结合领域知识和遗传算法对解决布局问题具有广泛的前景。

关键词 布局问题, 启发式经验, 遗传算法

中图分类号 O244

1 引言

布局问题广泛出现在许多领域, 遗憾的是随着应用领域的不同, 布局所涉及的诸如形状、维数、约束、目标等等都各不相同, 布局问题远比一般的 NP 完全问题更加复杂^[1]。30 多年来针对此类问题人们提出了许多算法, 但都存在许多不足。纯数学方法的计算时间过长甚至不可解; 穷举法存在着组合爆炸问题; 搜索树技术是一种启发式搜索方法, 但因引入的启发式信息依赖于个人经验, 它一般同样存在“组合爆炸”; 神经网络方法在解决布局这类最优化问题时效果较好, 但目前本身存在的收敛性差和局部极优性也使人们对它评价不高。对布局问题而言, 寻找一种能在较短时间内求得高质量解的方法是很有必要的。

遗传算法(GA)的出现为求解布局问题提供了一个崭新的思路。遗传算法作为一种建立在自然选择和群体遗传学机理基础上的迭代自适应概率性搜索算法^[2], 它具有很强的全局搜索能力, 在解决问题时具有很强的适应性, 它特别适合求出问题的近似最优解。本文就基于这一思想, 将遗传算法和布局问题特有的启发式经验结合, 试图求解布局问题。

布局主要包括一维布局、平面布局和立体布局, 后者更为复杂。本文提出的算法针对立体布局而言, 这种算法在求解其它布局问题时更加简单, 稍加改变即可。

2 问题的定义和求解策略

本文涉及的立体布局问题定义如下:

定义 1 已知一个长方体集合 $R = \{R_1, R_2, \dots, R_n\}$, R_i 表示第 i 个长方体; a_i 、 b_i 、 c_i 分别为 R_i 的三条边长; $F = L \times W \times H$ 为将要被填充的大长方体, L 、 W 、 H 分别为其长、宽、高。求 R 的一个子集 R' , 使其中的每个元素互不重叠地放入 F 中, 且使 F 中剩余体积最小。

定义 2 定义当前布局面为当前布局的底, 在布局问题中, 以装箱问题为例, 装箱是一层一层进行的; 在放第一层时, 当前布局面就是箱子的底, 是一平面; 以后在进行下一

¹ 1997-01-17 收到, 1997-09-30 定稿
国家自然科学基金资助课题

层布局(装箱)时,当前布局的底一般就不是平面了,因为已经布局的长方体的高度不尽相同,即当前布局面为一曲面。为此

定义 3 当前布局面的凹凸度,以当前布局面的最低点为基点,规定经过基点的横切面为当前布局面的底面,则底面之上的所有小长方体的顶点的坐标确定了当前布局面的凹凸度。当前布局面的凹凸度将决定下次放入的小立方体的高度。

针对立体布局问题的特点,本文所采用的启发式规则实质上是对 CLS 算法的修正^[3]。CLS 算法是针对平面布局而言,我们对其扩充,将立体布局分解为平面布局的第三维再布局。本文提出如下的立体布局问题求解策略:

STEP1 对 F 以一端点为原点建立三维坐标系。

STEP2 寻找 R 的一个子集 R' ,使 R' 中元素的长度之和接近于 F 的长度 L ,同时使它们的宽互相接近,高互相接近。

STEP3 将 R' 中的元素按其宽度递减次序重新排列。

STEP4 在当前布局面上将 R' 装入 F 中,元素总长度与 L 匹配,同时 R' 中较高(宽度较宽)的一端对应于当前 F 中较低的一端,若根据当前凹凸度判断某一小长方体在箱内可以下降,则使之下移;记下 R' 中每个小长方体的顶点的坐标(下次布局的凹凸度)。

STEP5 $R = R - R'$,若 R 已空,则算法结束;否则,进行下一步。

STEP6 若装入的元素的顶点坐标在宽度方向已明显大于 W ,则当前布局面已布局结束,进行下一步;否则,回 STEP2。

STEP7 若 R 未空且 F 仍有空,重新以当前布局面为初始底,转 STEP2;否则算法结束。

在实际运行中,为保证某一小长方体的三条边分别作为长、宽、高,我们设计下面三个函数:对 $x \in \{a_i, b_i, c_i, i = 1, 2, \dots, n\}$,

$$f_1(x) : \text{IF } x \text{ 被选作长, THEN } f_1(x) = 1, \quad \text{ELSE } f_1(x) = 0$$

$$f_2(x) : \text{IF } x \text{ 被选作宽, THEN } f_2(x) = 1, \quad \text{ELSE } f_2(x) = 0;$$

$$f_3(x) : \text{IF } x \text{ 被选作高, THEN } f_3(x) = 1, \quad \text{ELSE } f_3(x) = 0.$$

约束条件为

$$\begin{aligned} f_1(a_i)f_2(a_i) + f_1(a_i)f_3(a_i) + f_2(a_i)f_3(a_i) + f_1(b_i)f_2(b_i) + f_1(b_i)f_3(b_i) \\ + f_2(b_i)f_3(b_i) + f_1(c_i)f_2(c_i) + f_1(c_i)f_3(c_i) + f_2(c_i)f_3(c_i) = 0, \end{aligned} \quad (1)$$

$$\begin{aligned} f_1(a_i)f_2(b_i)f_3(c_i) + f_1(a_i)f_2(c_i)f_3(b_i) + f_1(b_i)f_2(a_i)f_3(c_i) \\ + f_1(b_i)f_2(c_i)f_3(a_i) + f_1(c_i)f_2(b_i)f_3(a_i) + f_1(c_i)f_2(a_i)f_3(b_i) = 1. \end{aligned} \quad (2)$$

在这一算法中,STEP2 最关键但也最困难,我们用 GA 求解。此算法即是一个基于启发式经验的立体布局进化策略。将对高度的考虑去除之后,其中的 STEP2~STEP6 就是求解平面布局问题的策略。

3 实例求解

随机产生 40 组、每组 3 个共 120 个 0 到 1 之间的数据分别作为 40 个小长方体的长、宽、高;再设 $F = L \times W \times H = 2.0 \times 2.5 \times 1.5$ 。

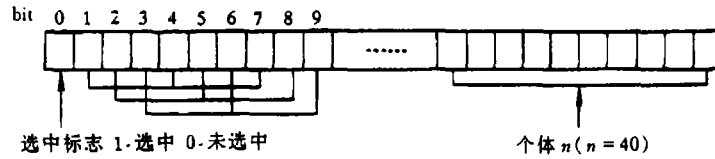


图 1 小长方体的编码

(1) 编码 编码就针对 40 个小长方体进行。布局在操作时要经过两次选择，第一次是选出一些小长方体，即求得当前待布局的 R' ；第二次决定 R' 中的元素三条边与长、宽、高的对应关系。我们的编码如图 1 所示：

每个小长方体对应 10bit，其中 bit 0=1 表示被选中，为 0 则未选中；另外 9 位规定如下：bit 1、4、7=1 分别显示 a_i 、 b_i 、 c_i 被选中作为长度；bit 2、5、8=1 分别显示 a_i 、 b_i 、 c_i 被选中作为宽度；bit 3、6、9=1 分别显示 a_i 、 b_i 、 c_i 被选中作为高度；若生成的码串不满足 (1)、(2) 式，则必须舍去。

(2) 适应度函数的设计 针对立体布局问题的特点，我们设定下面几个函数：

$$F_1 = \sum_{i=1}^n \sum_{j=i+1}^n ((a_i - a_j)^2 f_2(a_i) f_2(a_j) + (a_i - b_j)^2 f_2(a_i) f_2(b_j) + (a_i - c_j)^2 f_2(a_i) f_2(c_j) + (b_i - a_j)^2 f_2(b_i) f_2(a_j) + (b_i - b_j)^2 f_2(b_i) f_2(b_j) + (b_i - c_j)^2 f_2(b_i) f_2(c_j) + (c_i - a_j)^2 f_2(c_i) f_2(a_j) + (c_i - b_j)^2 f_2(c_i) f_2(b_j) + (c_i - c_j)^2 f_2(c_i) f_2(c_j)), \quad (3)$$

$$F_2 = \left(\sum_{i=1}^n (a_i f_1(a_i) + b_i f_1(b_i) + c_i f_1(c_i)) - L \right)^2, \quad (4)$$

$$F_3 = \sum_{i=1}^n \sum_{j=i+1}^n ((a_i - a_j)^2 f_3(a_j) f_3(a_i) + (a_i - b_j)^2 f_3(a_i) f_3(b_j) + (a_i - c_j)^2 f_3(a_i) f_3(c_j) + (b_i - a_j)^2 f_3(b_i) f_3(a_j) + (b_i - b_j)^2 f_3(b_i) f_3(b_j) + (b_i - c_j)^2 f_3(b_i) f_3(c_j) + (c_i - a_j)^2 f_3(c_i) f_3(a_j) + (c_i - b_j)^2 f_3(c_i) f_3(b_j) + (c_i - c_j)^2 f_3(c_i) f_3(c_j)). \quad (5)$$

F_1 决定 R' 中各元素之间的宽度差； F_2 决定 R' 中元素长度之和与 L 的差； F_3 决定 R' 中各元素之间的高度差。显然，进化总希望 F_1 、 F_2 、 F_3 越来越小，故得适应度函数：

$$F = 1/(A \cdot F_1 + B \cdot F_2 + C \cdot F_3), \quad (6)$$

(3) 遗传操作 选择采用依据个体适应度的随机比例选择策略；交叉采用两点交叉方式；变异概率设定为 $P_m = 0.1(1 - F(i)/F_{\max})$ ，其中 F_{\max} 为所有 $F(i)$ 的最大值。

表 1 40 个小长方体数据

i	a	b	c	i	a	b	c	i	a	b	c	i	a	b	c
1	0.29	0.50	0.70	11	0.38	0.31	0.25	21	0.60	0.57	0.35	31	0.65	0.57	0.57
2	0.69	0.55	0.58	12	0.80	0.47	0.36	22	0.45	0.28	0.45	32	0.66	0.51	0.51
3	0.84	0.81	0.72	13	0.57	0.38	0.40	23	0.39	0.41	0.82	33	0.76	0.61	0.45
4	0.17	0.75	0.42	14	0.43	0.85	0.42	24	0.42	0.72	0.45	34	0.50	0.75	0.97
5	0.46	0.46	0.67	15	0.37	0.46	0.74	25	0.41	0.80	0.60	35	0.03	0.07	0.63
6	0.45	0.40	0.29	16	0.35	0.51	0.58	26	0.94	0.41	0.81	36	0.69	0.53	0.45
7	0.36	0.72	0.36	17	0.45	0.50	0.48	27	0.75	0.60	0.61	37	0.74	0.27	0.52
8	0.51	0.45	0.59	18	0.66	0.88	0.58	28	0.76	0.77	0.40	38	0.32	0.55	0.23
9	0.75	0.63	0.32	19	0.26	0.55	0.32	29	0.50	0.78	0.38	39	0.82	0.26	0.63
10	0.67	0.77	0.85	20	0.53	0.49	0.93	30	0.26	0.41	0.75	40	0.82	0.71	0.37

设 40 个小长方体的数据如表 1 所示, 且设 $A = 20, B = 10, C = 3$, 最终的布局结果如图 2 所示 (由于第三层的高度比 H 过大, 故未给出)。

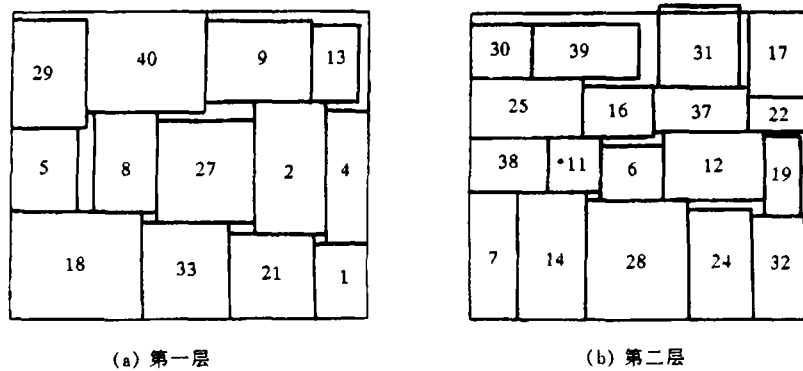


图 2 进化布局结果

表 2 给出了采用本算法的 STEP2~STEP6 和其它几种方法以表 1 中的 a 、 b 数据为对象布局时的结果^[4]。可以看出, 本文提出的布局进化策略在求解平面布局问题时效果最优; 对立体布局问题, 从图 3 可以看出, 本文提出的进化启发结合法的布局速度最快。

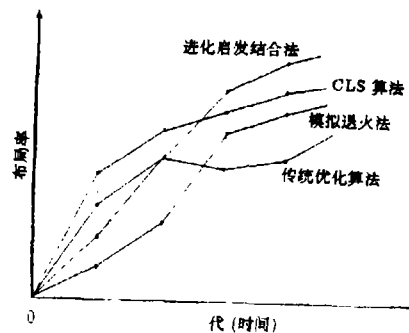


图 3 几种方法的进化速度

表 2 几种方法的比较

	长宽尺寸	面积损失率 (%)
传统优化方法	2.0 × 2.5	>24.0
模拟退火法	2.0 × 2.5	>18.3
进化启发结合法	2.0 × 2.5	< 8.1
CLS 算法	2.0 × 2.5	> 22.4

4 结束语

立体布局问题远比一般的 NP 完全问题更为复杂。本文将 GA 思想引入布局问题的解决, 提出了一个基于启发式经验的布局问题进化求解算法。该算法为解决布局问题提供了一条新的解决思路, 具有巨大的潜力。

参 考 文 献

- [1] Dowsland K A *et al.* Packing problems. *Euro. J. of Operational Research*, 1992, 56(4): 2-14.
- [2] Galletly J. An overview of genetic algorithms. *Kybernetes*, 1992, 21(6): 26-30.
- [3] Coffman E G *et al.* Average-case analysis of cutting and packing in two dimensions. *Euro. J. of Operational Research*, 1990, 44: 134-144.
- [4] Chee-kit Looi. Neural networks methods in combinatorial optimization. *Computers & Operations Res.*, 1992, 19(3/4): 191-208.

AN EVOLUTION STRATEGY TO THE THREE DIMENTIONAL PACKING PROBLEMS BASED ON HEURISTIC EXPERIENCE

Cao Xianbin Zhuang Zhenquan*

(Dept. of Computer Science, University of Science and Technology of China, Hefei 230026)

**(Dept. of Electronic Engineering, University of Science and Technology, Hefei 230026)*

Abstract It is very hard to solve the packing problem especially for the three dimensional packing problem. A new evolution strategy is presented in this paper which combines heuristic experience with genetic algorithms, its evolutionary thought is common to whatever dimensional packing problems. Finally, an experiment is given proving the strategy is better.

Key words Packing problem, Heuristic experience, Genetic algorithms

曹先彬: 男, 1969年生, 博士, 讲师, 主要研究兴趣: 智能信息处理、神经网络、进化算法等。

庄镇泉: 男, 1938年生, 教授, 博士生导师, 主要研究兴趣: 智能信息处理, 神经网络, 遗传算法等。近年来出版神经网络及其应用等专著 5 本, 论文 50 多篇。