

文章编号: 1007-4619(2008)05-0724-10

多目标粒子群算法与选址中的形状优化

黎海波, 黎 夏, 刘小平, 艾 彬, 刘 涛

(中山大学 地理科学与规划学院, 广东 广州 510275)

摘 要: 选址问题是 GIS 最基本的任务之一。一般性的选址是基于点的位置优化, 可利用有关 GIS 功能完成。实际的选址问题是很复杂的。在给定设施的数量和面积前提下, 需要在空间上确定设施的最佳位置, 并对形状进行优化, 以获取最大的效用。采用一般的方法无法求解这种最优化问题。而且, 当选址问题涉及多个目标和不同的约束性条件时, 就会变得异常复杂。提出了利用多目标粒子群优化算法和区域形状变异算法相结合来解决复杂的空间选址问题。具有智能的搜索方法, 大大提高了空间搜索能力, 并保持了搜索区域的连通性, 取得了较好的效果。

关键词: 粒子群算法; GIS 多目标选址; 形态; 优化

中图分类号: P208 **文献标识码:** A

1 引 言

一般性选址问题的目的是搜索研究区域中最优的点, 各种各样比较完善的算法已被应用于这种基于点的优化研究, 例如 simplex Gauss-Newton 和 Levenberg-Marquart 等^[1]; 此后多种人工智能和进化算法也被应用于其中, 如退火算法、遗传算法、蚁群算法、粒子群算法等, 例如黎夏等将遗传算法和 GIS 相结合, 应用于空间多点的选址问题, 取得较好的效果^[2]。

但是在许多工程规划和空间决策等实际应用例子中, 常常遇到这样的问题: 在规划面积已定的前提下, 探索建设设施的最优位置和最佳形状。即区域选址, 其目的是搜索一片连续的区域, 并使区域的位置和形状同时达到最优。区域选址比一般性选址问题更具体、更实用、应用更广泛, 但同时比一般性选址问题更复杂, 更难以解决。国内外也在此方面进行了许多尝试性研究, 例如早在 1981 年 Baenwald 就对城市住宅区选址问题进行探索^[3], 此后 Gilbert^[4]、Lane^[5]、Minor^[6]等也在不同的领域遇到相似的区域选址问题, 并进行了相关的分析研究。

当一般性选址问题演变到区域选址问题时, 保持搜索区域的连通性是整个搜索过程的关键, 这就涉及形状的优化, 整个搜索过程会变得更复杂, 形状和位置同时变化将会产生无穷多的可能组合, 常规的穷尽搜索 (brute-force search) 方法无法找到最优的解决方案。针对区域选址问题的复杂性, 提取出了利用粒子群算法 (PSO) 结合形状变异算法 (shape mutate) 来解决区域的优化选址问题。研究表明该算法可以使区域解智能地找到最优位置并同时搜索到最佳的形状, 能有效地解决区域搜索问题。

2 多目标粒子群算法与空间优化选址

区域选址不同于普通的平面选址, 除了包括位置优化外, 还涉及形状优化; 不是简单考虑平面上的一点, 而是考虑一片连续的区域, 常常需要考虑多个相互矛盾的准则^[7], 例如区域面积、适宜度、花费 (cost) 及形状等^[8], 此外还涉及连通性 (contiguity) 问题, 使简单的优化问题变成了具有形状约束的复杂问题, 目的是尽可能满足多个特定目标的要求。例如住宅区选址, 既要使总地价最低, 又要令其尽可能地接近某一公共设施 (商业

收稿日期: 2007-03-09; 修订日期: 2007-06-03

基金项目: 国家杰出青年基金资助项目 (编号: 40525002); 国家自然科学基金资助项目 (编号: 40471105); 和国家 863 计划, 编号: 2006AA12Z20094-2021 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

作者简介: 黎海波 (1982—), 男, 硕士。中山大学地理科学与规划学院。主要从事地理信息系统与地理信息模型研究。E-mail: redondo@gmail.com。通讯作者: 黎 夏, E-mail: lixia@mail.sysu.edu.cn

区等)^[9,10];国外学者已对这类问题进行了大量的研究^[3-6],基本方法是将区域的连续性作为约束条件,运用线性最优化的算法寻找最优解。但这种方法常常会忽略部分最优解,并且在涉及非线性、多目标问题的时候,此类方法会变得束手无策。基于此,本文引入粒子群算法解决区域选址过程中的瓶颈,粒子群算法是一种基于生物进化技术的智能算法,能非常有效地应用于优化问题的搜索中。

2.1 Pareto最优解

区域选址的复杂性决定了选址模型很难用数学公式表达,其实质是多目标优化,但实际问题中不可能使所有的准则同时达到最优化,从而不可能得到一个单独的最优解,通常得到的是协调各个准则的一组优化解(解空间)即折衷解,或称“Pareto解”。所谓 Pareto最优解^[11],就是不存在比其中至少一个目标好而其他目标也不劣的更好解。Pareto最优解集内的元素就所有目标而言是彼此不可比较的。

多目标优化问题的求解,传统的方法是将多个目标根据某种效用函数合成单一目标,利用单目标函数的优化方法求解,如目标加权法、功效系数法、层次优化法和目标规划法等。但在大多数情况下,优化之前这种效用函数是难以确定的,并且每次优化过程中只能找到一个最优解,如果没有足够的先验知识,决策者难以判断所求解的质量。因此对于多目标问题,往往需要给决策者提供符合条件的多种方案以使其做出合理的最终选择。

图 1 表示了一个双目标无约束的多目标优化问题的解空间。显然点“R”不是最优解,因为至少“E”点的两个目标 f_1, f_2 均比它的小。同理,位于可

行域内但不在其边界“ABECD”上的解不是最优解。相反,位于曲线“ABECD”上的所有解均为最优解,整条曲线构成 Pareto解集(或称 Pareto前沿)。这对于设计人员相当重要,因为它提供了一个相对广泛的选择空间。在优化之后,设计人员只需按各个目标的具体要求在 Pareto解集中进行选择即可,而不必再去考虑是否存在更优的方案。

假定某一决策过程中,需要同时考虑 k 个目标,且要求所有目标函数在满足约束的条件下越小越好,这样的优化问题可以表述如下

$$\min_{X \in R} F(X)$$
$$F(X) = (f_1(x), f_2(x), \dots, f_k(x))^T$$
$$R = \{X | g(X) \leq 0, g(X) = (g_1(X), g_2(X), \dots, g_n(X))^T$$
$$X = (x_1, x_2, \dots, x_n)^T, X \in R \subset E^n \quad (1)$$

式中, $F(X)$ 为优化目标向量, $g(X)$ 为约束向量, X 为决策变量。

定义 1: 占优和优于 (dominate)

如果以下两条同时成立,则称解 x_1 优于 x_2 , 或 x_2 被 x_1 占优:

- (1) 对于所有目标, x_1 不比 x_2 差, 或对所有的 $k=1, 2, \dots, M$ 均存在 $f_k(x_1) \leq f_k(x_2)$;
- (2) 对于至少一个目标, x_1 严格地好于 x_2 , 或对于至少一个 $k \in \{1, 2, \dots, M\}$ 使得 $f_k(x_1) < f_k(x_2)$ 。

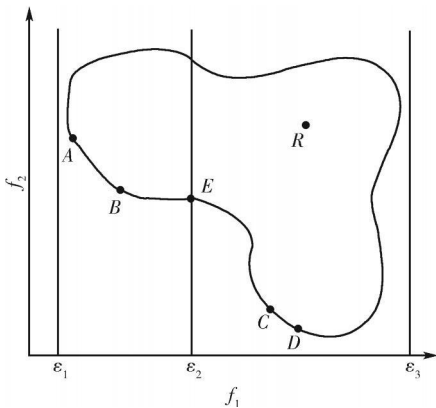
定义 2: 非劣解集和 Pareto边界(前沿) (nondominated solution and Pareto solution)

对于解集 P , 非劣解集 P' 是 P 中不被其他任一解占优的解集合。 P' 有时也被称为非劣解边界;若 P 是整个可行解空间, 则非劣解集 P' 被称为 Pareto最优解集, 简称 Pareto解集或 Pareto前沿。

因此, 多目标优化问题的任务有两个: (1) 尽可能收敛到真正的 Pareto前沿; (2) 找到的解要具有多样性和均匀性, 尽可能的覆盖整个 Pareto前沿。

2.2 PSO 算法的基本原理

粒子群优化算法 (PSO) 是一种进化计算技术 (evolutionary computation), 源于对鸟群捕食的行为研究^[12]。一群鸟在随机搜索食物, 所有的鸟都不知道食物在哪里。但是他们知道当前的位置离食物还有多远, 找到食物的最优策略就是搜寻目前离食物最近的周围区域。PSO 模型源于这种启示并用于解决优化问题。PSO 中, 每个优化问题的解类似于搜索空间中的一只鸟。我们称之为“粒子”。所有的粒子有一个由被优化的函数决定的适应值



(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

图 1 Pareto边界(前沿)示意图

Fig.1 Pareto optimal set

(fitness value), 还具有决定飞翔方向和距离的速度。然后粒子群就追随当前的最优粒子在解空间中搜索。

PSO 初始化为一群随机粒子 (随机解)。然后通过迭代找到最优解。在每一次迭代中, 粒子通过跟踪两个“极值”更新自己。第一个就是粒子本身所找到的最优解, 称为个体最优位置 P_{BEST} 。另一个极值是整个种群目前找到的最优解, 称为全局最优位置 G_{BEST} 。另外也可以不用整个种群而只是用其中一部分作为粒子的邻居, 当中的极值称为局部极值。

在找到这两个最优值时, 粒子根据以下公式来更新自己的速度和新的位置:

$$\begin{aligned} V_{ij}(t+1) &= wV_{ij}(t) + c_1 r_1(t) [p_{ij}(t) - x_{ij}(t)] + \\ &\quad c_2 r_2(t) [p_{gj}(t) - x_{ij}(t)] \quad (2) \\ x_{ij}(t+1) &= x_{ij}(t) + V_{ij}(t+1) \quad (3) \end{aligned}$$

式中, j 表示粒子的第 j 维; i 表示粒子 i ; t 表示第 t 代; $V_{ij} \in [-V_{max}, V_{max}]$ 表示粒子的速度, V_{max} 为用户设定的常数, V_{max} 太大, 粒子有可能会飞过好解, V_{max} 太小, 粒子可能会缺乏足够的探索能力, 容易陷入局部最优值; w 为惯性权重, 使粒子保持运动惯性, 使其有扩展搜索空间的趋势; c_1 、 c_2 为学习因子, 通常在 $[0, 2]$ 间取值, 代表了将每个粒子推向 P_{BEST} (个体最优位置) 和 G_{BEST} (全局最优位置) 的统计加速项的权重; r_1 、 r_2 是介于 $[0, 1]$ 间的随机数; p_i 为第 i 个粒子搜索到的最优位置; p_g 为整个粒子群搜索到的最优位置。

2.3 多目标粒子群算法 (Multi-Objectives PSO)

粒子群算法在许多单目标优化问题的应用中取得了成功, 说明了其可行性和有效性, 但是不能直接应用于多目标优化问题, 因为两者有着本质的区别: 前者一般是单个解或一组连续解, 而后者则是一组或几组连续解的集合。将粒子群算法运用于优化问题的关键是如何确定每个个体的局部和全局最优位置。单目标优化问题的解决方法很直接, 每个粒子的局部最优位置由该粒子的历史最优位置决定, 全局最优位置由当时的群体全局最优位置决定。多目标优化问题由于并无单个的最优解, 所以不能直接确定这两个量。

为了解决这个问题, Coello 和 Lechuga^[13] 在非劣最优概念的基础上应用了一个“容器”记录已找到的非劣向量, 并用这些解指导其他粒子的飞行。

Parsopoulos 和 Vrahatis^[14] 应用了权重聚合的方法,

Hu 和 Eberhart^[15] 应用了动态邻近的 PSO 算法求解多目标优化问题。另外, 一些多目标粒子群算法采用了常用的归档机制: 存在一个外部记忆体, 该记忆体始终保持在整个迭代过程中的非劣性, 群体中的每个个体的全局最优位置由该记忆体产生并存储。实验证明, 归档机制能够大幅度提高算法的收敛速度和解的质量^[16]。采用该机制比较典型的为 Coello 和 Maxinino Salazar Lechuga 的多目标粒子群算法 (本文简称 CMMOPSO)^[17]; 针对每个个体, 采用轮盘赌算法从外部记忆体中选出; 每个个体的局部最优位置为该粒子的历史最优位置, 而当需要更新粒子的历史最优位置时, 按 Pareto 支配关系从该粒子的当前位置和历史最优位置中选择较优者作为升级后的历史最优位置, 如无支配关系, 则从两者中随机选择一个。Carlo 等人对此算法进行改进, 提出了将拥挤距离 (crowding distance) 应用于粒子群算法解多目标问题的方法, 结合多样化操作、变异操作以及约束限制, 得到了较好的效果^[18]。

3 多目标粒子群算法 (MOPSO) 与多目标区域搜索

3.1 整数空间的粒子群算法

在栅格中随即产生粒子属于整数规划问题, 由于所有粒子的位置必须位于栅格单元上, 粒子的位置取值必须在整数空间中, 用栅格单元的行和列表示。而原来的粒子群算法取值范围是整个实数空间, 因此必须对原来的算法进行改进, 保证所有粒子的位置处于正整数域。

整数规划问题可描述为

$$\min f(x), x \in S \subseteq Z^n$$

式中, Z^n 为 n 维整数空间, S 为一整数集, 可以为无界集。

当采用粒子群算法求解整数规划问题时, 由于 w , c_1 , r_1 , c_2 , r_2 的存在, 即使粒子的当前位置和速度均为整数, 下一位置则可能为实数, 搜索仍在实数空间中进行。针对整数取值的问题, 对粒子群算法的速度公式式 (2) 和位置公式式 (3) 进行了以下改进:

$$V_{ij}(t+1) = \text{int}(wV_{ij}(t)) + f_1 + f_2 \quad (4)$$

$$\begin{aligned} x_{ij}(t+1) &= x_{ij}(t) + V_{ij}(t+1) \quad (5) \\ \phi_1 \in &\begin{cases} [0, c_1(p_{ij}(t) - x_{ij}(t))], & p_{ij}(t) > x_{ij}(t) \\ [c_1(p_{ij}(t) - x_{ij}(t)), 0], & p_{ij}(t) \leq x_{ij}(t) \end{cases} \quad (6) \end{aligned}$$

$$p\{\phi_1\} = 1 / \text{int}(c_1(p_{ij}(t) - x_{ij}(t))) \quad (7)$$

$$\phi_2 \in \begin{cases} [0, c(p_{kj}(t) - x_{ij}(t))] & p_{kj}(t) > x_{ij}(t) \\ [c(p_{kj}(t) - x_{ij}(t)), 0] & p_{kj}(t) \leq x_{ij}(t) \end{cases} \quad (8)$$

$$r \phi_2 = 1 / \ln(c(p_{kj}(t) - x_{ij}(t))) \quad (9)$$

ϕ_1 为区间 $[0, c(p_{kj}(t) - x_{ij}(t))]$ 中等概率分布的整数,

ϕ_2 为区间 $[0, c(p_{kj}(t) - x_{ij}(t))]$ 中等概率分布的整数。

上述改进遵照了粒子群的基本原理,并有效地保证了 v_{ij} 和 x_{ij} 都处于整数空间中。

3.2 区域搜索的适应值定义

为了更好地解释多目标区域搜索原理,定义一个多目标区域搜索的问题:寻找地价总和最低并且离公共设施点最近的区域。假定地价分布在计算之前已经知道,公共设施的位置固定。

- 定义
- i : 区域单元格的编号;
 - N : 区域包含单元格的数目;
 - C_i : 单元格 i 的地价;
 - x_i : 单元格 i 的 x 坐标 (行);
 - y_i : 单元格 i 的 y 坐标 (列);
 - x' : 公共设施的 x 坐标 (行);
 - y' : 公共设施的 y 坐标 (列);
- 则多目标优化可定义为:

$$\min \min \text{ize } c = \sum_{i=1}^N c_i \quad (10)$$

$$\min \min \text{ize } d = \frac{1}{N} \sum_{i=1}^N [(x_i - x')^2 + (y_i - y')^2]^{\frac{1}{2}} \quad (11)$$

式 (10) 表示区域所有单元格地价总和的最小值;

式 (11) 表示区域所有单元格距离公共设施平均距离的最小值。

3.3 区域搜索适应值的计算

区域选址适应值的计算不同于普通的选址问题,普通选址的目的是选取最优的点域 (一个栅格单元),可以简单地将位置参数代入适应值函数进行计算。而区域选址目的是选取一片连续的区域 (多个连通的栅格单元),这就使适应值的计算变得复杂化。计算过程涉及两个重要的因素,首先要保证区域 (多个栅格单元) 的连通性,使其形成一片完整的区域;并且要得到最优的形状,即在相同或相近位置得到的适应值最优的栅格块的形状。

3.4 区域形状变异操作

Xiao 和 David A Bennett 等于 2001 年提出保持区域连通性的算法,此算法在保持栅格连通性约束条件下,通过对栅格单元随机选取、移动、状态更新等一系列操作,得到新的区域形状^[19]。虽然算法中有相应的操作避免了区域中“洞”的形成,但此操作还不完善,在一定的情况下还会形成“洞”。本文对该算法进行了改进,排除了形成“洞”的可能性,严格保持了区域的连通性。

3.4.1 初始化区域形状 (Init Position (P))

由粒子群算法随机产生的粒子 P 的位置作为区域左上角栅格单元的位置,即位于栅格单元 0 上 (图 2),整个区域形状最先初始化为规则的矩形块状,由左至右排列,例如: 3×3 的矩形。如果区域由 10 个栅格单元组成,则剩下的一个单元被加到矩形的下部。

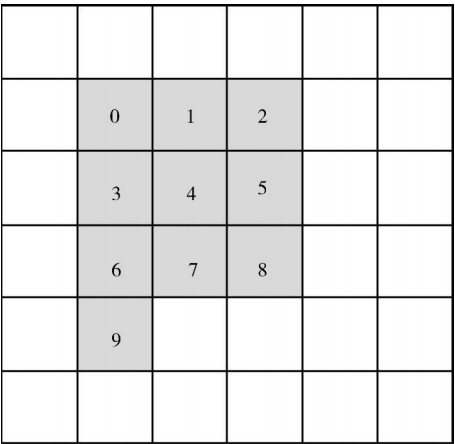


图 2 初始区域形状

Fig.2 The initial site shape

3.4.2 形状变异操作 (Shape Mutate ())

初始化区域的形状后,执行区域形状变异操作以随机改变区域的形状 (图 6)。为了保存搜索区域的状态信息,首先要对搜索区域栅格单元的状态进行编码,用 sta 记录单元的状态值,如图 3 (c)。

分别考虑栅格单元上、下、左、右 4 个方向邻域的状态,栅格邻域的状态有 3 种:

- (1) 与其他区域单元相邻。记录其相应方向的状态值 sta 为相邻的区域单元的编号。
- (2) 与边界相邻。记录其相应方向的状态值 sta 为 -2。
- (3) 不与任何区域单元或边界相邻,即为空邻域。记录其相应方向的状态值 sta 为 -1。

形状变异操作基本原理:随机选取一个可移动 (moveable) 的区域单元,将其移动到符合迁移条件

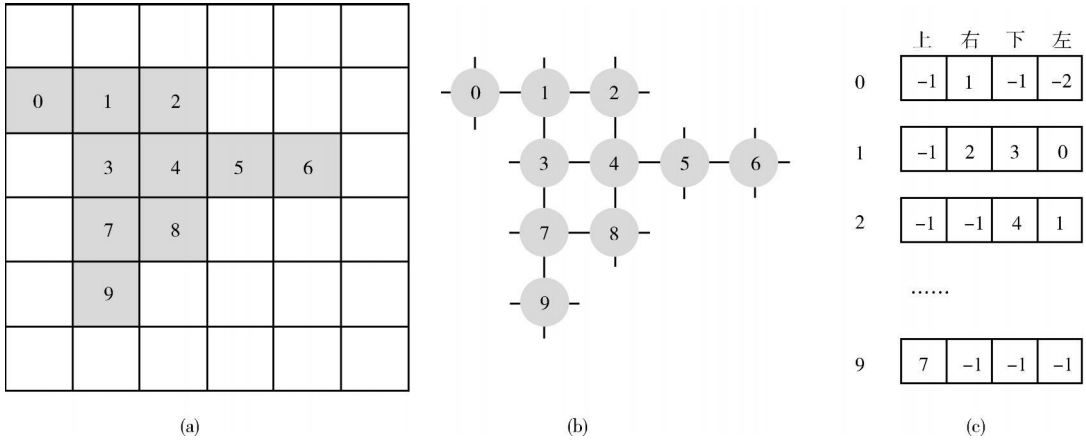


图 3 搜索区域表示方式图

(a)可行解区域原形; (b)可行解图示表达; (c)区域状态的数据结构

Fig.3 A graph representation for a search-site

(a)A feasible solution; (b)The graph representation of the solution in (a); (c)The data structure to implement the graph representation

的新位置。可移动区域单元是指将此单元从区域上移除后,区域还能保持连通。判断一个区域单元P是否可移动,首先要计算其空邻域 (sta=-1)的个数:

(1)空邻域个数为 0 时, P 不可移动。例如图 4 中区域单元 4 为不可移动单元。

(2)空邻域个数为 3 时, P 为可移动单元。例如图 4 中区域单元 0, 6, 9 为可移动单元。

(3)空邻域个数为 2 时, 如果其非空邻域单元之间存在着不经过 P 的连通路径, 则 P 为可移动单元, 反之则否。例如图 4 中区域单元 2, 8 为可移动单元 (在单元 2 的非空邻域 1 和 4 之间, 存在不经过 2 本身的连通路径 1-3-4; 在 8 的非空邻域 4, 7 之间存在不经过 8 的连通路径 4-3-7); 而 5 则是不可移动单元 (4 和 6 之间不存在不经过 5 的连通路径)。

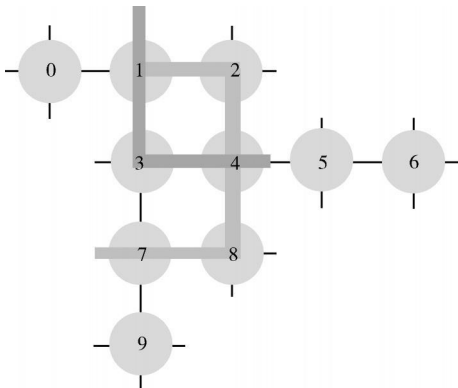


图 4 可移动单元分析

Fig.4 Moveable cell analysis

(4)空邻域个数为 1 时, 情况同 (3)。例如图 4 中区域单元 3 为可移动单元 (在单元 3 的非空邻域

1, 4 和 7 之间, 有不经 3 本身的连通路径 1-2-4-8-7); 而 1, 7 则是不可移动单元 (0, 3 和 2 之间不存在不经 1 的连通路径; 3, 8 和 9 之间不存在不经 7 的连通路径)。

符合迁移条件的新位置是指直接与区域相邻的空邻域单元, 并且迁移到此位置后, 不会在区域中产生“洞”, 即区域内部出现空单元的情况, 如图 5。一般情况下, 为了保证区域连通性, 区域选址问题不希望在搜索区域中间出现留空的情况, 因此在改变区域形状的时候, 要避免区域单元迁移时产生“洞”。以下两种条件下会导致“洞”的形成:

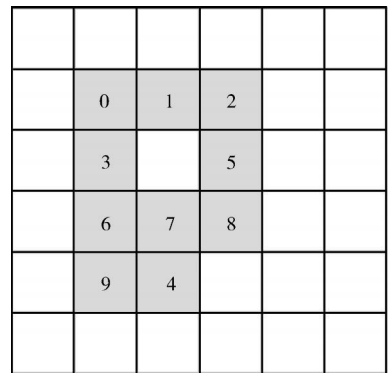


图 5 “洞”示意图

Fig.5 A graph representation for the “hole”

(1)区域单元移动到“桥”位置。如图 6 标注着的空单元格为“桥接”单元, 此空单元仅仅连接两个已存在的区域单元。当区域单元移动到此位置, 将会形成“洞”。

(2)区域单元移动到“造洞”位置。如图 7 标注着的空单元格为“造洞”单元, 此单元至少存在一个空

邻域,这个空邻域的其余 3 个邻域是已存在的区域单元。当区域单元移动到这个位置,也必定会形成“洞”。

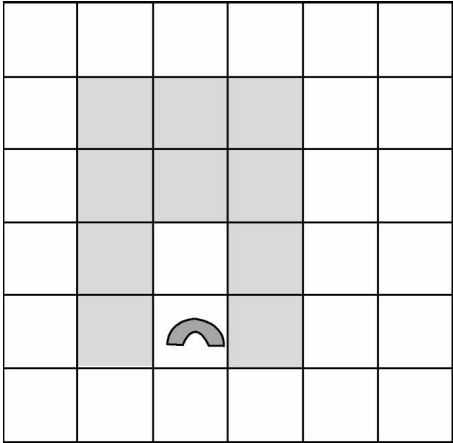


图 6 “桥接”单元示意图

Fig.6 A graph representation for the bridge cell

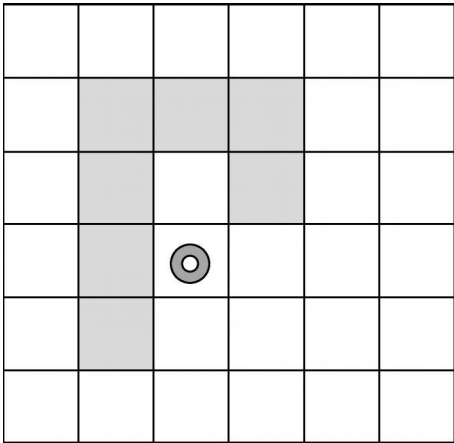


图 7 “造洞”单元示意图

Fig.7 A graph representation for the FormHole cell

4 模型应用及结果分析

4.1 测试数据

本测试使用面积为 128×128个网格的栅格数据作为研究区,以住宅小区的最佳选址为例。假设搜索区域(住宅小区)的面积为 10个单元面积(式(9)和

式(10)中 N=10),公共设施位于研究区域的正中央。地价分布分别使用 3 个不同函数作测试:随机分布函数(uniform random)、锥形分布函数(conical)和岛状分布函数(deformed sambreno)。这 3 个函数产生的值代表了不同类型的空间分布,由随机分布到结构化分布。通过分析数值空间结构对选址结果的影响,可以看出算法对各种分布函数具有适应性(表 1)。

表 1 3 种不同的测试函数

Table 1 Summary of the three cost surfaces

函数	方程	参数
Random	$C_{ij} = \text{random}(0.0, 1.0)$	服从随机分布
Conical	$C_{ij} = 1 - [f(i, j) / (2 \times 64^2)]^{\frac{1}{2}}$	$f(i, j) = (i - 64)^2 + (j - 64)^2$ $i, j \in (1, 2, \dots, 128)$
Deformed Sambreno	$C_{ij} = 0.0002f(i, j) + 4\exp[-0.01f(i, j)] + 0.00015ij - 3\exp[-0.01f(i - 50, j - 50)] - \exp[-0.01f(i + 50, j + 50)]$	$f(i, j) = (i - 64)^2 + (j - 64)^2$ $i, j \in (1, 2, \dots, 128)$

4.2 测试结果

本次试验选取了 100 个粒子进行运算,利奔腾 IV 主频为 3GHz 的计算机对 3 个测试函数做迭代 100 次的测试,所需要的时间均不超过 5s。从图 8—图 10 中可以看出,算法很快地使解空间收敛到 Pareto 最优解集中,并且每次迭代产生优于上一代的非劣解,迭代次数越多,产生的非劣解越多,也越接近于 Pareto 前沿。

4.2.1 Conical 函数测试

图 8 反映了 Conical 函数对选址的影响。可以明显看出,图 8(b)比图 8(a)更密集、更均匀,说明迭

代次数越多,解的分布越均匀,越接近 Pareto 前沿。图 8(d)中显示的选址结果分散于整个研究区域。由于地价呈锥形分布(图 8(c)),越接近研究区域中心的数值越大,即地价和到中心设施的距离成反比关系,因此图 8(a)和图 8(b)中 Pareto 前沿呈现为一直线,所有的解落在了直线上。换言之,不考虑搜索区域的形状,任何一个解是非劣最优解,因为对于每一个解,找不到比它更接近公共设施同时地价更低的解(要想更接近公共设施,必须更靠近研究区域中心,这时地价必然升高,两个目标同时优化的情况不存在)。因此选址的结果比较均匀地分布在整個研究区域,这种情况下,形状变异操作就起了筛选作用,在

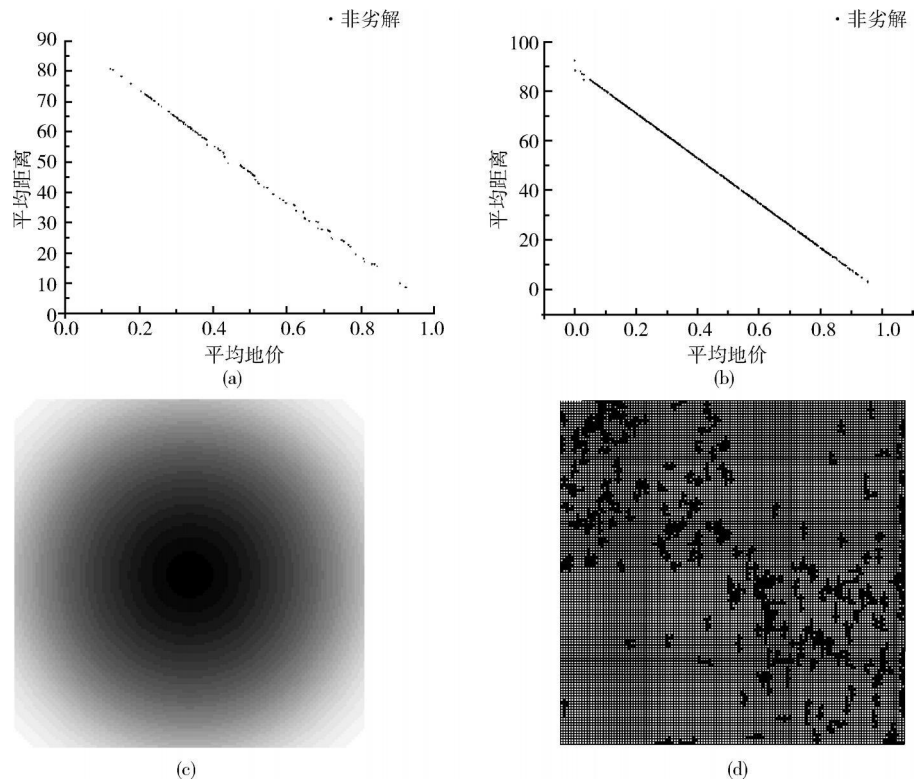


图 8 Conical分布函数的区域选址结果

(a) 迭代 10 次的非劣解分布情况; (b) 迭代 100 次的非劣解分布情况; (c) Conical 分布; (d) 区域选址结果

Fig.8 The result ofMOPSO-Site for conical surface

(a) The nondominated solutions in 10 generations (b) The nondominated solutions in 100 generations (c) Conical cost surface (d) The results

相同或者相近的位置能搜索出选址区域的最佳形状，图 8(d)中得到的结果是形状优化后的解。

4.2.2 Sambrero函数测试

图 9(a)、(b)更明显的表现了非劣解在迭代运算中的动态改善过程。如图 9, 非劣解的数量和均匀度不断地完善, 第 100 代非劣解数量已达到 400 多个, 并在整个 Pareto 前沿非常均匀地分布着。图 9(d)是选址结果分布图, 从图中可以明显看出, 在地价 sambrero 分布 (图 9(c)) 的情况下, 非劣解集集中在两个区域: 接近公共设施的位置以及地价较低的位置。非劣解形成的 Pareto 前沿 (图 9(b)) 不是一条连续的曲线, 在中间部分断开, 是由于此部分虽然平均地价相差不大, 但到公共设施的距离不优于 A 点, 故只有 A 点是最优解, 其他部分都不是非劣最优解。研究区域的右下角也出现了地价较低的区域 (图 9(c)), 但此处并没有出现非劣解 (图 9(d))。这是因为出现地价谷值的两个区域中心对称, 到设施的距离相等, 但左上角的谷峰值明显比右下角的谷峰值小, 也就是说左上角的地价要比右下角的低。因此比较两者优劣时, 在一个目标 (到设施的距离) 相同的情况下, 比较另外一个目标 (地

价) 就可以知道谁优谁劣。显而易见, 左上角的解都要比右下角的解占优, 研究区域右下角自然不会有非劣解出现的情况。

4.2.3 Random 函数测试

图 10 反映了地价随机分布时对非劣解的影响。图 10(a)清楚地显示了随着迭代次数的增加, 非劣解集的变化情况, 由第一次迭代后随机分布的状态迅速向研究区域中心靠拢, 并且解的数量增多, 分布也越来越均匀, 最终出现了图 10(c)中的选址结果。这反映了在地价随机分布的条件下, 公共设施的位置对选址结果起决定作用。因为地价随机分布时不产生空间分布趋势, 与空间地理位置无关, 非劣解自然地与空间位置相关的另一个目标——公共设施靠拢, 认为越接近公共设施位置的解越优, 因此解都集中在研究区域的中心。图 10(c)清楚显示了搜索结果的最优形状。

从上述试验可以看出, 算法对各种函数具有很强的搜索能力, 在搜索到最佳位置的同时可以得到区域的最佳形状, 并且能迅速地收敛到搜索空间的 Pareto 前沿, 产生均匀的非劣最优解集。

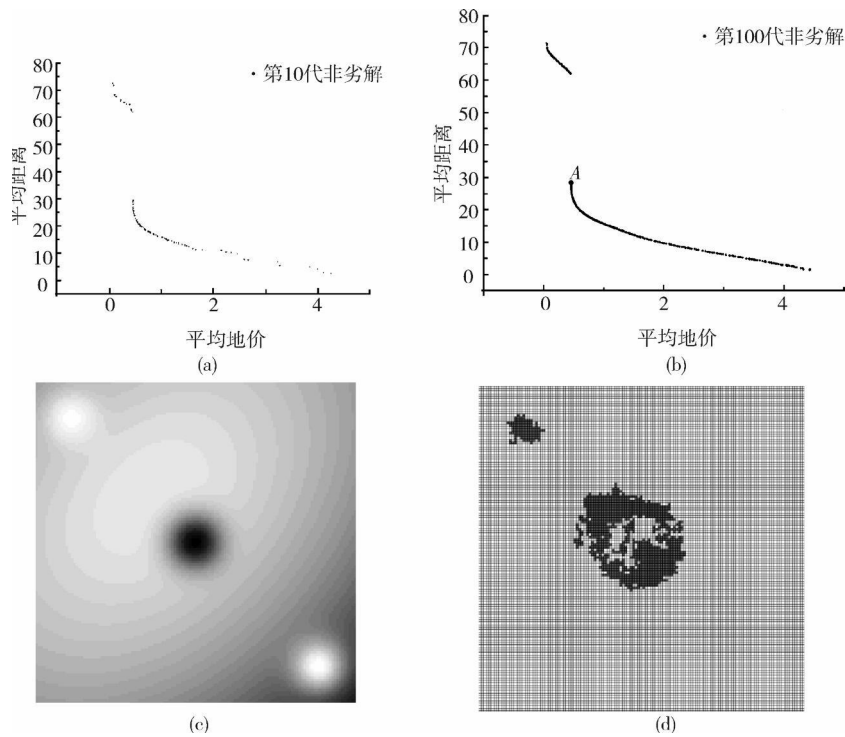


图 9 sambreno分布函数的区域选址结果

(a)迭代 10 次的非劣解分布情况; (b)迭代 100 次的非劣解分布情况; (c)Deformed Sambreno分布; (d)区域选址结果

Fig.9 The result ofMOPSO-Site for sambreno surface

(a) The nondaminated solutions in 10 generations (b) The nondaminated solutions in 100 generations
(c) Deformed Sambreno cost surfaces (d) The results

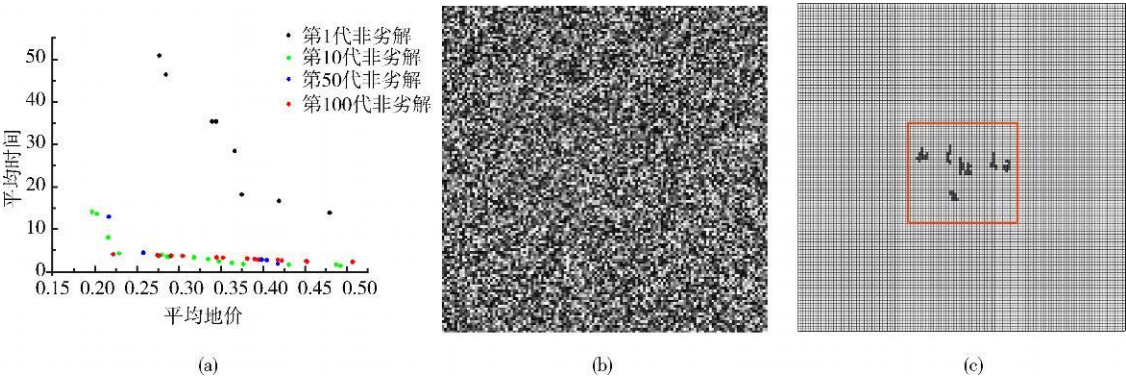


图 10 Uniform Random分布函数的区域选址结果

(a)非劣解分布情况; (b)Uniform Random分布; (c)区域选址结果

Fig.10 The result ofMOPSO-Site for uniform random surface

(a) The nondaminated solutions (b) Uniform Random cost surfaces (c)The results

5 实际应用例子及结果分析

5.1 研究区域及空间数据

进一步以简单的商业区选址为例,研究区域为广州市城区,包括海珠区,越秀区,荔湾区的全部和天河区及白云区的人口密集区。将整个研究区域分割为 100×100 个栅格单元,栅格的分辨率为 200m×200m,一个栅格单元的实际面积大概为 40000.m²。

GIS提供了空间优化模型所需的基本空间数据。实验同时考虑两个目标的优化情况,分别是商业区到大型住宅小区的平均距离以及周边交通情况(到交通站点的平均距离)。以上两个目标均认为越小越优。假设搜索区域(商业区的占地面积)为 5 个单元面积,约 20 万 km²。

5.2 实验结果及分析

实验结果如图 11。选址结果分布于大型住宅小

区集中、交通非常便利地带。主要分布在天河区繁华地段、白云区人口稠密地区,部分结果分布在海珠区西北角、荔湾区北部以及东山区和天河区交界处。

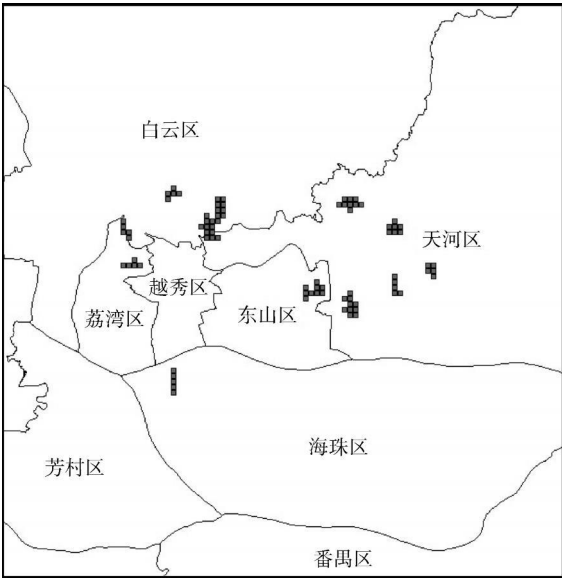


图 11 商业区选址结果分布

Fig.11 The result ofMOPSO-site for shopping center

天河区西面密集分布着豪景花园、嘉逸花园、跑马地花园、怡景花园、东城花园和华江花园等大型住宅小区,并且附近交通十分便利,还靠近火车东站、沙东客运站和天河客运站等大型车站,因此商业区的选址结果分布在此处是十分合理的。白云区处的选址结果临近祥港花园、和润花园、瑶台花园和云泉花园;荔湾区北部的选址结果临近德心花园、田心花园、雅怡花园、侨德花园和富力半岛花园等,这两处属于老城区比较发达的地区,毗邻广园客运站、省汽车站、市汽车客运站和省火车站等大型车站,此处也可以作为商业区的最佳选址地点之一。另外东山区和海珠区西北角的选址结果也是大型住宅小区分布比较密集的地方,附近分别有雅景花园、锦城花园、金山花园、新世界花园和凤安花园、宝利丰花园、金沙花园、光大花园等,并且交通也十分方便。

可见,实验产生的选址结果十分合理,在搜索到合适地点的同时,区域的最优形状也搜索出来。如果试图使用穷尽搜索法来解决商业区的区域选址问题,位置和形状的组合将会产生无穷无尽的可能性,根本是不可能解决的。现在,开发商只要根据实际情况,在搜索结果的基础上选取适合的地点进行开发即可,可以忽略搜索结果以外的所有区域,节省了大量的时间和人力物力。

6 结 论

本文提出了将多目标粒子群算法与形状变异算法结合解决多目标区域选址问题的方法。该方法可以智能地寻找最优位置并同时搜索到区域最佳形状,有效地解决区域搜索问题。在整个搜索过程中,搜索区域始终保持着空间连通性。在此前提下,结合经过改进的多目标粒子群算法,在各种不同类型分布函数的测试中,算法能迅速并有效地捕捉到非劣最优解,到达 Pareto 前沿。算法最终产生的选址结果,能直接为决策者提供选择,决策者只需要在选址结果上按具体要求进行筛选,不必再去考虑是否存在更优的方案。

参考文献 (References)

[1] Zhang H G, Lee Z P, Shi P, et al. Retrieval of Water Optical Prosperities for Optically Deep Waters Using Genetic Algorithms [J]. IEEE Transactions on Geoscience and Remote Sensing, 2003, 41(5): 1123—1128.

[2] Li X, Anthony GarOn YEH. Optimal Spatial Search Using Genetic Algorithms and GIS [J]. acta Geographica Sinica, 2004, 59(5): 745—753.

[3] Baerwald T J. The Site Selection Process of Suburban Residential Builders[J]. Urban Geography, 1981, 22, 339—357.

[4] Gilbert K C, Holmes D D, Rosenthal R E. A Multiobjective Discrete Optimization Model for Land Allocation [J]. Management Science, 1985, 31, 1509—1522.

[5] Lane W, McDonald R. Land Suitability Analysis Landfill Siting[J]. Journal of Urban Planning and Development, 1983, 109, 50—61.

[6] Minor S D, Jacobs T L. Optimal Land Allocation for Solid and Hazardous Waste Landfill Siting[J]. Journal of Environmental Engineering, 1994, 120, 1095—1108.

[7] Current J M in H, Schilling D. Multiobjective Analysis of Facility Location Decisions[J]. European Journal of Operational Research, 1990, 49, 295—307.

[8] Cova T J Church R L. Contiguity Constraints for Single-Region Site Search Problems[J]. Geographical Analysis, 2000, 32, 306—329.

[9] Arentze T A, Borgers A W J, Timmemans H J P. An Efficient Search Strategy for Site Selection Decisions in an Expert System [J]. Geographical Analysis, 1996, 28, 126—146.

[10] Malczewski J. Site Selection Problem and a Quasi-Satisfying Decision Rule [J]. Geographical Analysis, 1992, 24, 299—316.

[11] Pareto V, Cours D. Economic Politique volume I and II [M]. F Rouge Lausamme 1896.

[12] Kennedy J Eberhart R C. Particle Swarm Optimization [A]. Neural Networks 1995. Proceedings of IEEE International Conference on Publication Date, Nov/Dec 1995.

[13] Coello C A, Lechuga M S. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization [A]. IEEE Congress on

- Evolutionary Computation (CEC 2002) [C]. Honolulu Hawaii USA, 2002.
- [14] Parsopoulos K E, Vrahatis M N. Particle Swarm Optimization method in Multiobjective Problems [A]. Proc of the ACM Symp on Applied Computing 2002 (SAC 2002) [C]. New York: ACM Press, 2002.
- [15] Hu X, Eberhart R C. Multiobjective Optimization Using Dynamic Neighborhood particle Swarm Optimization [A]. IEEE Congress on Evolutionary Computation (CEC 2002) [C]. Honolulu Hawaii USA, 2002.
- [16] Purshouse R C, Fleming P J. Elitism, Sharing and Ranking Choices in Evolutionary Multi-Criterion Optimization [R]. Sheffield UK: Department of Automatic Control and Systems Engineering, University of Sheffield, 2002.
- [17] Coello C A, Lechuga M S. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization [A]. Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002) [C]. Piscataway, New Jersey: IEEE Service Center, 2002.
- [18] Raquel C R, Naval P C Jr. An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization [C]. Genetic And Evolutionary Computation Conference Proceedings of the 2005 Conference on Genetic and Evolutionary Computation Washington DC, USA, 2005.
- [19] Xiao N C, David A Bennett, Marc P Armstrong. Using Evolutionary Algorithms to Generate Alternatives for Multiobjective Site Search Problems [J]. Environment and Planning, 2002, 34: 639—656.

Particle-Swarm Optimization for Site Selection with Contiguity Constraints

LI Hai-bo, LI Xia, LIU Xiao-ping, AI Bin, LIU Tao

(School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China)

Abstract Site selection, which is one basic task of GIS functionalities, is to search for the best sites for a facility or a number of facilities. The objective is to maximize some utility functions subject to some goals. Traditional site selection methods using GIS only focus on the identifying the best locations (coordinates) of facilities. In many applications, contiguity constraints must be considered in site selection. Site selection should consider not only locations, but also patch configuration for solving many optimization problems. The objective is to maximize utility functions subject to contiguity constraints and various planning goals. The combination of locations and contiguity for site selection is a difficult problem for site selection because of involving huge solution space. The problem becomes more complex when multi-objectives are incorporated in the optimization.

Many alternative generating techniques (such as the weighting method and the non-inferior set estimation method) have been developed to help decision makers search solution spaces. Although these methods are effective under some circumstances, the approaches have several weaknesses: (1) it can only be applied to problems that are mathematically formulated; (2) it is inefficient when applied to large problems; and (3) it may fail to find important solutions. As a consequence, builders of decision-support tools require methods that overcome these limitations and efficaciously generate alternative solutions to multi-objectives decision problems. Particle-swarm optimization (PSO) can be used to achieve such goals.

This paper presents a new method to solve such problem by using particle-swarm optimization (PSO) method and shape-mutate algorithms, which is a strict mutation operator to prevent the formation of “holes” in searching for optimal contiguous sites. Particle-swarm optimization method is used to make the solutions flying to the best locations. Shape contiguity constraint and patch configuration optimization are operated by shape mutate algorithms. Here, a site is represented by using an undirected graph and a set of operations is designed to change the shape and location of sites during the search for possible solutions. These operations evolve randomly generated initial solutions into a set of optimal solutions to this type of problem; at the same time, the contiguity of a site is maintained and the “holes” of the site are prevented to formation.

This approach is applied to three different types of cost surfaces: uniform random, a conical and a deformed sombrero-like surface. The analyses are focused on a 128×128 grid of cells, where a facility is located at the center of the area. The number of cells for a site is fixed and set at 10. The results demonstrate the robustness and effectiveness of this PSO-based approach to geographical analysis and multi-objective site selection problems. This approach has also been tested in the city of Guangzhou to search for the best locations for CBD. The results are also reasonable.

The experiments have indicated that this approach is effective in solving this problem. It can successfully capture all the best solutions. The results can be used directly as the location selection by the decision makers, because these have been the best solutions.

Key words particle-swarm optimization; GIS; site selection; patch configuration; optimization