

遗传模拟退火算法在约束求解中的应用

刘生礼 唐敏 董金祥

(浙江大学人工智能研究所 CAD/CG 国家重点实验室, 杭州 310027)

摘要 将遗传模拟退火算法应用于约束求解中, 提高了约束系统求解的鲁棒性和效率. 与 Newton-Raphson 数值方法相比, 由于遗传模拟退火算法是一种单纯的数值迭代方法, 不涉及到矩阵求逆, 因此克服了 Newton-Raphson 法对初始值敏感的缺点, 具有很强的鲁棒性; 与其他利用 BFGS 的优化算法相比, 由于遗传模拟退火算法是在一个初始的解空间中搜索所有可能的解, 因此克服了 BFGS 优化算法对良约束多解情况只能求出一个解的缺点; 由于遗传模拟退火算法是将约束问题转化为优化问题后才进一步求解, 因此其可以处理过约束一致和欠约束的问题.

关键词 计算机图形学(520·6030) 遗传模拟退火算法 约束求解 参数化设计

中图分类号: TP391.72 **文献标识码**: A **文章编号**: 1006-8961(2003)08-0938-08

Geometric Constraint Satisfaction Using Genetic Simulated Annealing Algorithm

LIU Sheng-li, TANG Min, DONG Jin-xiang

(State Key Laboratory of CAD&CG, AI Institute, Zhejiang University, HangZhou 310027)

Abstract This paper applies genetic simulated annealing algorithm (SAGA) to solving the geometric constraint problems. Genetic simulated annealing algorithm itself has many merits, such as implicit parallelism, stability of numerical computation and the global searching ability together with the local fast converging ability, etc. This paper takes the special characters of constraint solving into consideration and integrates the SAGA well with it. After the geometry constraint problem is transformed into the optimal one, it is solved by SAGA that making full use of the advantages of SAGA. This approach can deal with the over-/under-constraint problems because of this conversion. It also has advantages due to its not being sensitive to the initial values over the Newton-Raphson method, and its yielding of multiple solutions is an advantage over BFGS(Broyder-Fletcher-Goldfard-Shanno) for multi-solution constraint system. Our experiments have proved the robustn.

Keywords Computer graphics, Genetic simulated annealing algorithm, Geometric constraint solving, Variational design

0 引言

几何约束问题是指给定一些几何元素, 如点、线、面等, 并在这些几何元素之间指定约束关系, 如距离、角度、重合等等, 以此来确定几何元素的具体位置. 几何约束求解就是通过计算, 来确定这些几何元素的实际坐标, 并且同时满足这些约束关系.

如果一个几何约束问题有有限的解, 那么这个问题就称为良约束(well-constrained); 如果有无限

多的解, 就称为约束不足(under-constrained); 如果删除其中一个约束, 而问题还是有有限多解, 那就称为过约束(over-constrained). 在过约束(over-constrained)的情况下, 如果多余的约束与其他的约束是一致的, 那么问题还是有解的, 这种情况的约束问题称之为过约束一致, 但是在很多情况下, 多余的约束与其他约束不一致, 那么它是没有解的, 这种情况的约束称之为过约束不一致.

约束求解广泛应用于各个领域, 如实体建模、分子建模、机械设计等等^[1], 几何约束求解有很多方

法,可以分为以下种:

(1) 基于图的构造方法^[2~9].

(2) 逻辑推理和项重写技术.该方法将通用的几何推理技术应用到约束求解中.这种方法首先由 Aldefeld^[10]和 Bruderlin^[11]采用.具体请参见文献 [12]~[14].

(3) 符号代数法.该方法也将约束形式化为一组代数方程.但是,它不用数值的技术来确定解,而是采用通用的符号计算来寻找方程组的解.通常使用 Gröbner 基^[15]和 WuRitt 技术^[16]来找到解的符号表达式.如 Kondo 就通过符号计算来向一个约束系统中增加或删除约束^[17],具体请参见文献 [18]~[21].

(4) 数字代数法.该方法首先将约束翻译为一组代数方程,然后,用 Newton-Raphson 之类的方法进行迭代.很明显,数值方法是一种实例求解器.这种方法的优越之处是在收敛的情况下,它可以求解其他技术不能求解的过约束一致的问题.缺点是对初始值敏感.主要的实例有 Sketchpad^[22]、ThingLab^[23].

由于这4种方法都各有利弊,因此 Durand 提出一种系统框架(systematic framework)的方法^[24],将这4种方法有效地结合在一起,并且引入 homotopy 方法对核心方程组(core system)进行路径搜索,最后可以求出约束系统的解.但该方法速度较慢^[25],而且,它对核心系统求解时,若模板在坐标中放置的方式不一样,会大大影响 homotopy 的求解效率^[26],而且通常还要求约束方程组中的变量数与方程数相等.

David 等提出的求解二维约束的新方法^[27],将几何约束和几何元素用基本约束、基本几何元素代替,通过对距离矩阵和角度矩阵的操作,可以求解二维约束问题,包括欠约束和过约束问题.但是它不能处理三维约束问题.

Dennis 等提出的全局收敛的优化计算方法^[28],是通过定义一个能量函数来进行计算,然而该方法有时候会收敛于局部最优解,导致求解失败^[24].

Ge 提出的一种数值优化方法^[29],将 BFGS 优化算法应用到约束求解中,并且在实验中获得了良好的效果,可以处理欠约束和过约束一致问题. BFGS 是一种全局收敛的数值优化算法,对初始值不敏感,但是对于良约束多解,它只能计算出一个解.

本文将遗传模拟退火算法与代数方程化简相结合,即首先将几何约束问题转化为一组方程,然后对方程组进行简单的符号化简.化简后的约束方程组中变量大大减少,使得遗传算法中个体的基因型长度变短.约束系统的解进一步应用遗传模拟退火算法求出.

遗传模拟退火算法是在初始群体中进行选择、交叉、变异操作,对种群进行“优胜劣汰”筛选,最终使得进化后的种群逼近最优解.由于在所有操作中,没有涉及到在 Newton-Raphson 中遇到的矩阵求逆操作,因此,遗传算法具有很强的鲁棒性.将约束问题转化为数值优化的过程中,对方程的个数和变量的个数没有什么特别的要求,因此可以处理欠约束问题.对于欠约束问题,实际上有无穷多解,许多算法只是判断了有没有欠约束情况,并没有在欠约束情况下进一步给出一个适合约束条件的解,而本文中的算法可以在欠约束的情况给出一个解,使得草图中的图形“尽最大可能”地满足设计者意图.对于良约束多解的情况,可以同时求出多个解.

1 遗传模拟退火算法

1.1 遗传算法

遗传算法已经成为人工智能领域的热点,在机器学习、工程优化、任务调度、图象处理等领域得到了成功的应用.

遗传算法本身包含了在潜在解空间中的迭代演化过程,从每代中挑选出适当的个体,并用它们结合遗传操作产生下一代的新个体.在选择过程中,为了模仿生物界的“优胜劣汰”,对选择算子有如下要求:适应度较高的个体被遗传到下一代群体中的概率较大;适应度较低的个体被遗传到下一代的概率较小.

遗传算法具有良好的全局搜索能力,可以快速地将解空间中的全体解搜索出,而不会陷入局部最优解的快速下降陷阱.利用它的内在并行性,可以方便地进行分布式计算,加快求解速度.但是遗传算法的局部搜索能力较差,这就导致了单纯的遗传算法比较费时.

1.2 模拟退火算法

模拟退火算法是一种适用于大型组合优化问题的技术,算法的核心在于模拟热力学中液体的冻结与结晶或金属溶液的冷却与退火过程.它能够以随机搜索技术从概率的意义上找出目标函数的全局最

小点. 模拟退火算法可以用马尔柯夫链的遍历理论来给它以数学上的描述. 在搜索最优解的过程中, 模拟退火算法除了可以接受优化解外, 还用了一个随机接受准则(Metropolis 准则)有限度地接受恶化解(ill-conditioned), 并且接受恶化解的概率慢慢趋向于零, 这就使得算法可能从局部最优中跳出, 尽可能找到全局最优解, 并保证了算法的收敛.

但是由于模拟退火算法对整个搜索空间的状况了解不多, 不便于使搜索过程进入最有希望的搜索区域, 从而使得模拟退火算法的运算效率不高.

1.3 遗传模拟退火算法

遗传模拟退火算法是将遗传算法与模拟退火算法相结合而构成的一种优化算法. 与基本遗传算法的总体运行过程相类似, 遗传模拟退火算法也是从一组随机产生的初始解(初始群体)开始全局最优解的搜索过程, 它先通过选择、交叉、变异等遗传操作来产生一组新的个体, 然后再独立地对所产生出的各个个体进行模拟退火过程, 以其结果作为下一代群体中的个体. 这个运行过程反复迭代地进行, 直到满足某个终止条件为止.

遗传模拟退火算法将遗传算法和模拟退火算法的优点充分结合起来, 大大提高了算法的效率. 遗传模拟退火算法的程序计算基本框架为^[30]:

```

{
(1) 确定出某个问题解的表现型和潜在的解空间;
(2) 确定目标函数的类型及其数学描述形式和量化方法;
(3) 确定表示可行解的染色体编码和解码方法;
(4) 确定个体适应度的判别方法;
(5) 确定选择、交叉、变异算子;
t ← 0;
初始化种群 P(t) = {X1, X2, ..., XN};
计算 P(t) 的适应度;
Do
{
  个体交叉 P'(t) ← Crossover[P(t)];
  个体变异 P''(t) ← Mutation[P'(t)];
  个体模拟退火算法;
  P'''(t) = SimulatedAnnealing[P''(t)];
  计算 P'''(t) 的适应度;
  个体选择操作: P(t) ← Re production[P(t) ∪ P'''(t)];
  t ← t + 1;
} while (不符合中止条件)
}

```

下面就遗传模拟退火算法来对约束求解进行细化.

2 基于遗传模拟退火的约束求解

2.1 决策变量、约束条件和优化模型

约束问题可以形式化为 (E, C) , 其中 $E = (e_1, e_2, \dots, e_n)$, e_i 表示几何元素, 如点、线、圆等; $C = (c_1, c_2, \dots, c_m)$, c_i 表示加在这些几何元素之间的约束. 一般一个约束对应一个代数方程, 因此约束可以表示为

$$\begin{cases} f_1(x_0, x_1, x_2, \dots, x_n) = 0 \\ f_2(x_0, x_1, x_2, \dots, x_n) = 0 \\ f_3(x_0, x_1, x_2, \dots, x_n) = 0 \\ \dots \\ f_{m-1}(x_0, x_1, x_2, \dots, x_n) = 0 \\ f_m(x_0, x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (1)$$

$$X = (x_0, x_1, \dots, x_n)$$

式中, x_i 为表示几何元素 e_i 的一些参数值, 如二维点可以表示为 (x_1, x_2) . 约束求解就是求出满足方程(1)的 X .

令

$$F(X_j) = \sum_{i=1}^m |f_i| \quad (2)$$

显然, 若有 X_j 满足 $F(X_j) = 0$, 则 X_j 满足方程(1). 因此约束求解可以转化为优化问题, 只需求 $\min(F(X_j)) < \epsilon$ 就可以了, ϵ 是某个阈值. 为了提高遗传算法的速度, 用 f_i 的绝对值的和而不是平方和来表示约束方程组. 由方程(2)以及用遗传算法求解 $\min(F(X_j)) < \epsilon$ 知道, 并不要求 $m = n$, 因此该方法显然可以求解欠约束问题和过约束问题.

决策变量为 $x_i, i \in [0, n]$, 约束条件为 $x_i \in [L_i, L_u]$, $[L_i, L_u]$ 为决策变量 x_i 的变化范围, 方程(2)为优化模型.

2.2 编码、解码方法

在约束求解中, 个体表示为 $X_j = (x_0, x_1, \dots, x_n)$, $x_i \in [L_i, L_u], i \in [0, n]$, x_i 为决策变量, 为了提高计算速度和精度, 采用浮点数编码与格雷编码两级编码相结合的方式. X_j 的基因型为

X_j	x_0	x_1	...	x_{n-1}
-------	-------	-------	-----	-----------

基因型编码的长度为决策变量的个数 n . 实际个体的基因型和表现型是一致的, 不需要进行一些中间转化. 表现型为 $X_j = [x_0, x_1, \dots, x_{n-1}, x_n]$.

因为决策变量的变化范围是一个预估值, 变化

范围比实际范围要大得多,搜索空间扩大很多,并且计算精度也很高,所以采用浮点数编码。

为了有效地进行交叉和变异遗传操作,对交叉点或变异点所在的决策变量又进一步进行格雷编码,达到局部细化的目的。

用 20 位格雷码对某个决策变量 x_i 进行编码。20 位的二进制编码串可以表示从 0 到 1 048 575 之间 1 048 576 个数,因此将决策变量 x_i 的定义域 $[L_i, L_u]$ 离散化为 1 048 575 个相等的区域。交叉或变异后,要将该决策变量 x_i 的二进制基因编码换算成实数形式,即首先将 x_i 转换为对应的十进制代码 y_i ,然后依据个体编码方法和对定义域的离散化方法,将 y_i 转化为实际的决策变量值

$$x_i = (L_u - L_l) \times \frac{y_i}{1\,048\,575} + L_l$$

由此可以看出,采用浮点数编码与格雷编码两级编码的好处,一方面是利用浮点数编码,减少了基因型编码的长度,提高了计算速度和精度;另一方面是采用局部二进制编码可以使得变异、交叉操作变得十分简单。

2.3 个体适应度评价方法

目标函数为 $F(X_j) = \sum_{i=1}^m |f_i|$, 值域是 $(0, \max)$, 但是优化的目标是求函数的最小值,因此适应度与函数值成反比,可以计算个体 X_j 的适应度 $Fit(X_j)$ 如下

$$Fit(X_j) = 1/e^{\beta F(X_j)}, 0 \leq F(X_j) \leq \max, 1/e^{\beta \max} \leq Fit(X_j) \leq 1$$

理论上, $F(X_j) \rightarrow \infty, 0 < Fit(X_j) \leq 1$

其中, β 为适应度调节参数,在选择的不同时期,可以调整 β 值,改变种群的选择压力,防止遗传算法发生早熟现象。如在初期, β 取一较大值,而在后期取一较小值。在我们的实验系统中,在前 25 代中取 $\beta = 1.25$, 25 代以后,取 $\beta = 0.85$ 。

2.4 设计遗传算子

(1) 选择算子

采用简单的比例选择方法 (Proportional Model)。该方法的基本思想是:各个个体被选择的概率与其适应度大小成正比。尽管该方法带有很大的随机性,但是由于在遗传算法中溶入了模拟退火这一快速局部收敛的算法,因此可以避免由随机性而带来的收敛速度慢的缺陷。

(2) 交叉算子

交叉算子采用单点交叉 (One-point

Crossover)。但是由于采用了浮点数编码与格雷编码两级编码方式,因此单点交叉运算与普通的交叉运算又有所不同。

对每一对交叉的个体,随机设置某一基因之后的位置为交叉点,注意随机产生的这个交叉点位置共有 $n \times 20$ 种可能性,而不是 n 种。假如交叉点为 k , 则

$$\text{int } p = k/20;$$

$$\text{int } q = k \bmod 20;$$

$(p+1)$ 为要交叉的个体中某个决策变量 x_i 的位置, q 为相对于该决策变量 x_{p+1} 的起始交叉点位置 (如图 1 所示)。

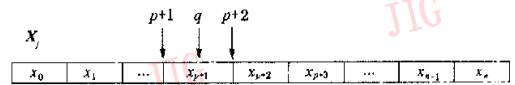


图 1 两级编码交叉示意图

交叉过程为:将两个交叉个体交叉点后的决策变量 $(x_{p+2} \dots x_n)$ 与 $(x'_{p+2} \dots x'_n)$ 直接互换,对交叉点所在的决策变量 x_{p+1} 重新进行格雷编码,然后将二进制编码中位置 q 后的二进制码进行互换,最后根据 2.2 节中所述方法重新计算此决策变量的实际值重新填回到染色体个体中。

(3) 变异算子

采用基本变异 (Simple Mutation), 但对其作一些改动,即当确定某个变异基因位置后,不是以变异概率 p_m 随机指定某一位或几位作变异操作,而是以变异概率 p_m 在 $[L_i, L_u]$ 范围内随机产生一个数来代替这个基因值。这样尽管会带有很大的随机性,但是由于结合了模拟退火算法,因此可以快速局部搜索。

2.5 确定遗传算法的运行参数

其他基本参数为:

(1) 群体 P 大小: $M = 200$, 在 $[L_i, L_u]$ 内随机产生决策变量组成个体。把用户由图面上直接输入的决策变量值组成的个体也加入初始种群 P 中。注意,对于不同的决策变量,变化范围不一样。

(2) 终止代数: $T = 100$;

(3) 交叉概率: $p_c = 0.5$;

(4) 变异概率: $p_m = 0.0005$ 。

2.6 计算终止条件

条件 1 良约束并且相隔两代变化最小 ($\delta \leq \epsilon$)。定义相隔两代 ($j, j+1$ 两代) 的变化为

$$\delta = \left| \sum_{k=1}^M F(X_{j,k}) - \sum_{k=1}^M F(X_{j+1,k}) \right|$$

其中, M 为种群空间的大小, $X_{j,k}$ 表示第 j 个个体在第 k 次的计算结果。

条件 2 欠约束并且 $|Fit(X_j) - 1| \leq \epsilon$. 因为在欠约束情况下, 有无穷多解, 所以只要求出任意一个满足约束的解就可以了。

终止条件 = (终止代数 $< T$) \cup 条件 1 \cup 条件 2.

2.7 输出结果

在输出的种群中, 取适应度最大的个体 X_i (即 $Fit(X_i) = \max[Fit(X_1), \dots, Fit(X_M)]$) 为约束问题的解。

3 部分优化计算

3.1 紧缩搜索空间

由 2.2 节中知道个体 $X_j = (x_0, x_1, \dots, x_n)$, 因此搜索空间与个体决策变量的个数 n 有线性正比关系. 对于约束问题, 特别是空间约束问题, 产生的约束方程组中包含的自变量 x_i 很多, 但这些方程之间又是有一定的关系, 因此可以对方程组进行一些符号化简操作, 以减少自变量的个数和方程的个数。

但并不是符号化简得越多, 约束系统的求解就越简单^[31], 有时当符号消减到一定程度时, 会使方程组的求解稳定性下降. 因此要在符号化简和求解稳定性方面进行一个权衡. 本文只是进行一些简单的高斯化简和常量消减, 尽量避免方程组求解稳定性的下降. 例如典型的“四点一线”问题^[25], 可以列出约束方程如下:

$$x^2 + y^2 + z^2 - r_1^2 = 0 \quad (3)$$

$$(a_2 - x)^2 + y^2 + z^2 - (a_2 u)^2 - r_2^2 = 0 \quad (4)$$

$$(a_3 - x)^2 + (b_3 - y)^2 + z^2 - (a_3 u + b_3 v)^2 - r_3^2 = 0 \quad (5)$$

$$(a_4 - x)^2 + (b_4 - y)^2 + (c_4 - z)^2 - (a_4 u + b_4 v + c_4 w)^2 - r_4^2 = 0 \quad (6)$$

$$xu + yv + zw = 0 \quad (7)$$

$$u^2 + v^2 + w^2 - 1 = 0 \quad (8)$$

将式(4)减式(3), 式(5)减式(3), 式(6)减式(3)得 3 个式子, 联立这 3 个式子可得

$$\begin{cases} x = f_1(u) \\ y = f_2(u, v) \\ z = f_3(u, v, w) \end{cases} \quad (9)$$

将此式(9)分别带入式(3)、式(7)、式(8)并联立, 则原系统由 6 个变量 x, y, z, u, v, w 简化成 3 个变量 u, v, w .

3.2 定义域 $[L_1, L_n]$ 的确定

几何元素一般为点、线、圆、面. 在二、三维空间

中的一般表示如下:

点: 2 维 (x, y) , 3 维 (x, y, z) ;

线: 2 维 (n_x, n_y, d) , (n_x, n_y) 为线的单位法向, d 为线距原点的最近距离;

3 维 (p, t) , $p_i = (x, y, z)$ 为直线上距离原点最近的点, $t_i = (t_x, t_y, t_z)$ 为直线的单位矢量方向。

圆: 2 维圆心 (x, y) , 半径 r ;

3 维圆心 (x, y, z) , 半径 r ;

面: $P(n_x, n_y, n_z, d)$, (n_x, n_y, n_z) 为面的单位法向, d 为面距离原点的最近距离。

可以将变量分为两种类型: 类型 1 为坐标、距离、半径; 类型 2 为方向矢量. 对类型 2, $n_x^2 + n_y^2 + n_z^2 = 1$, 因此变化 $[L_i, L_n] = [-1, 1]$. 对于类型 1, 可以通过以下方法进行简单的放大求解。

设输入的距离约束或半径为 $d_i, i = 0, 1, \dots, n$ (n 为距离约束的个数); 角度约束为 $a_i, i = 0, 1, \dots, m$ (m 为角度约束的个数)

$$L = \sum_{i=0}^n d_i$$

对每一个角度 a_i ($\pi/2, 3\pi/2$ 除外), $L' = L/\cos a_i$, $L = L + |L'|$, 如图 2 所示. 因此可以得到类型 1 数值变化的大概范围 $[L_1, L_n] = [-L, L]$.

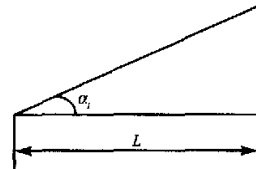


图 2 角度的等效长度计算图

4 实例分析

与普通的 Newton-Raphson 算法进行约束求解相比, 遗传模拟退火算法具有以下明显的优点:

(1) 由遗传模拟退火算法本身的特点决定了该算法很稳定. 根据我们的实验系统验证, 该算法能够鲁棒地进行约束求解. 因为遗传模拟退火算法是对初始的种群空间进行全局搜索, 并且没有矩阵求逆运算, 因此不存在像 Newton-Raphson 方法对初始值敏感的缺点。

(2) 将约束方程组转化成最优化问题时 (见式(2)), 并没有要求方程组(1)中的方程数 m 和变

量数 n 相等,因此其显然可以处理过约束和欠约束的问题.

下面举几个实例来说明遗传模拟退火算法.我们开发了 Tsketch 实验系统进行约束求解,在该系统中应用了遗传退火算法.在 Tsketch 中,支持的几何元素有点、线、圆、圆弧.支持的约束有:点点、点线之间的距离;线线角度;相切、平行、垂直、重合、对称、长度相等、角度相等;固定点.

4.1 欠约束实例

工程设计草图中经常会遇到标注尺寸不足的情况,甚至一些已经设计完成的草图也会有标注尺寸不足的地方,因为有些尺寸不重要而可以不标.但是在进行约束求解时,往往会导致求解失败.很多约束求解器只是给出求解失败原因,并没有尽最大可能去进行求解,但是遗传模拟退火算法却可以尽最大可能地得到欠约束情况的解.

Tsketch 中的一个简单图如图 3(a)所示,三角形的 3 条边长度确定,固定顶点 A ,将边 AB 约束为水平.显然三角形 ABC 处于良约束状态.进一步定义绘制一个圆,对圆施加约束:分别与 AC 、 AB 相切.此时,系统处于欠约束状态,因为平面中的圆有 3 个自由度,需要 3 个约束才能够确定一个圆的位置.图 3(b)是用 Tsketch 进行求解(重建)后的结果.由图 3(b)可以看出,尽管圆处于欠约束状态,但是它的位置仍然被重新计算过,使得圆与两条边 AC 、 AB 相切,并且三角形 ABC 仍然处于原来的位置.这样就使得处于欠约束状态的图形在求解后尽最大可能地满足设计者的意图.

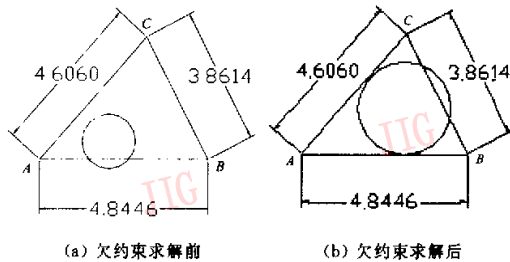


图 3

4.2 过约束实例

在工程设计中,也会遇到过约束的情形.一种情况就是设计者无意添加的,典型的情况就是标注尺寸形成首尾相接的闭合尺寸链的情形(如图 4 所示).过约束一致的情形仍然满足设计者的意图,因

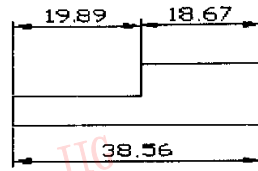
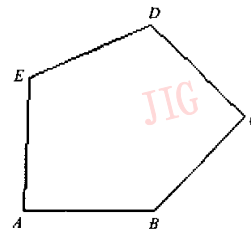


图 4 闭合尺寸链图

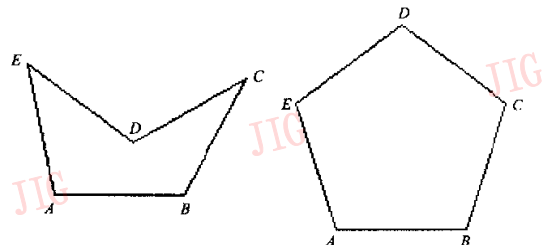
此约束求解器要能够处理这种情况.

另外一种情形是,在良约束的情况下,有时往往有多个解可以满足所添加的约束.如何从众多解中获得满足设计者意图的解,这也成为约束求解器需要解决的问题.很多约束求解器中利用启发信息进行解的筛选,如通过和初始用户输入的图形进行最近相似比较来获得解^[27],这种方法在用户输入的草图比较精确时非常成功,但是在不精确时,往往会失败.另外一种方法就是利用添加多余约束进行解的筛选,因此过约束求解变得非常重要.

如图 5(a)所示,绘制一个五边形草图.依次固定点 A 、 B ,添加约束:依次 5 条边两两相等;对角线 BE 、 AC 、 CE 两两相等.此时系统处于良约束状态,但是求解后结果却是图 5(b),显然不符合设计者的意图.实际上 Tsketch 求解出 8 个解,因为遗传模拟退火算法是在解空间中尽可能地搜索出所有的全局最优解,然后根据与初始图最大相似性匹配得到解(图 5(b)).但是若继续添加约束 $AD=BE$,则系统变为过约束状态.求解后的结果如图 5(c).



(a) 五边形草图



(b) 未加多余约束求解后的结果 (c) 添加多余约束求解后的结果

图 5

4.3 良约束实例

遗传算法与退火算法相结合,使得遗传算法的全局指导性和退火算法的局部快速收敛性很好地结合起来。由于约束求解问题本身具有一些特殊性,因

此针对这一特殊性作了一些优化,进一步提高了求解性能。如紧缩搜索空间、精确不同类型变量的变化范围等。图 6 是一个连杆轮廓图,进化过程和运行结果如图 7 所示,运行结果参数如表 1 所示。

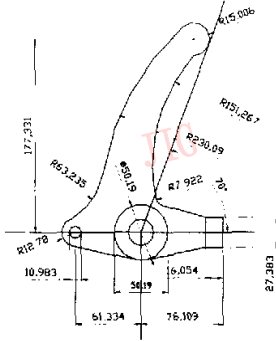


图 6 连杆轮廓图

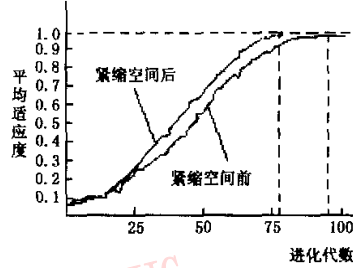


图 7 进化过程和运行结果

表 1 运动结果参数

方程数	变量数	运行时间 (s)	中止代数	最后一代的平均适应度 ($\frac{1}{M} \sum_{i=1}^M Fit(X_i)$)
紧缩空间前	41	1.32	95	0.99977
紧缩空间后	29	0.83	82	0.99997

由图 7 可以看到,由于在前 25 代调大了适应度参数 β ,因此,两条曲线的变化趋势都比较缓慢;在以后进化的代数中,调小了适应度参数 β ,因此变化趋势逐渐变陡。紧缩空间后,种群的搜索空间变小,精度变高,速度加快,由图中也可以明显地看到紧缩空间后的曲线比紧缩空间前的曲线要陡。

面体问题、四点一线问题,其余的问题一般可以通过模板问题组合来得到。Durand 根据具体的模板问题得到最佳的核心方程(core system),其余的问题可以通过求解核心方程反求得到解。Hoffman 对具体的模板问题进行几何构造,得到基本空间结构(如四面体、八面体)初始的最佳摆放位置,然后得到相关方程进行求解。这两种方法的弱点是对具体模板问题进行讨论,扩展性不高,如对“四点一线”这个问题,还没有发现通过几何构造来得到它的解。遗传模拟退火算法并没有针对具体的模板问题,它面向的只是约束方程组,因此求解面没有限制,由它本身特性决定了它可以求解出所有满足条件的解,但是它也有缺点,就是精确的求解空间比较难确定,显然如果求解空间越精确,计算速度越快。

5 结论

遗传模拟退火算法本身具有很多优点,如隐含的计算并行性、数值计算的稳定性、全局搜索与局部快速收敛搜索。将遗传模拟退火算法用于约束求解在国内外还没有发现相关文献,本文根据约束求解这一特殊问题,将遗传模拟退火算法用于约束求解,并在实验中获得了良好的效果,可以求解欠约束、过约束一致的情形。对于良约束多解的情形,可以同时求出满足条件的所有解。

多解问题在空间约束求解中更为常见,Durand 利用系统框架和 homotopy 方法对空间约束问题进行了详细讨论^[24],Hoffman 等利用几何推理对空间约束的基本问题也作了详细的讨论^[5]。它们共同的特点就是针对某个具体模板问题进行求解,如八

参考文献

- Durand C. Symbolic and numerical techniques for constraint solving [D]. Department of Computer Sciences, Purdue University, West Lafayette, USA, 1998.
- Fudos I. Editable representations for 2D geometric design[D]. Department of Computer Sciences, Purdue University, West Lafayette, USA, 1993.
- Fudos I, Hoffmann C. Correctness of a geometric constraint solver [EB/OL]. <http://www.cs.purdue.edu/homes/cmh/>

- distribution/PubsCons.html,1996.
- 4 Bounia W, Fados I, Hoffmann C M *et al*. A geometric constraint solver[J]. *Computer Aided Design*, 1995,27(6):487~501.
 - 5 Hoffmann C M, Vermeer P. Geometric constraint solving in R^2 and R^3 [A]. In: *Computing in Euclidean Geometry[M]*, second edition. Singapore, World Scientific Publishing, 1995:266~298.
 - 6 Hoffmann C M, Vermeer P. A spatial constraint problem[EB/OL]. <http://www.cs.purdue.edu/homes/cmh/distribution/PubsCons.html>,1995.
 - 7 Fudos I, Hoffmann C M. A graph constructive approach to solving systems of geometric constraints[J]. *ACM Transactions on Graphics*, 1997,16(2):179~216.
 - 8 Hoffmann C M, Joan-arinyo R. Symbolic constraints in constructive geometry[J]. *Journal of Symbolic computation*, 1997,23(2/3):287~300.
 - 9 Song Peng, Tang Min, Dong Jin-xiang. A constraint solving approach based on graph decomposition [A]. In: *Chinagraph'2000[C]*, Hang Zhou, China, 2000.
 - 10 Aldefeld B. Variation of geometries based on a geometric reasoning method[J]. *Computer Aided Design*, 1988,20(3):117~126.
 - 11 Beat Bruderlin. Symbolic computer geometry for computer aided geometric design[A]. In: *Proceedings of the NSF Design and Manufacturing Systems Conference [C]*, Tempe, Arizona, USA,1990:177~181.
 - 12 Verroust A, Schonek F, Roller D. Rule-oriented method for parametrized computer-aided design [J]. *Computer Aided Design*, 1992,24(10):531~540.
 - 13 Joan Arinyo R, Soto A. A correct rule-based geometric constraint solver[J]. *Computer and Graphics*, 1997,21(5):599~609.
 - 14 Joan Arinyo R, Soto A. A ruler-and-compass geometric constraint solver[A]. In: *Proceedings of the Fifth International Federation for Information Processing TC5/WG5. 2 International Workshop on Geometric Modeling in Computer Aided Design on Product Modeling for Computer Integrated Design and Manufacture [C]*, Airlie, Virginia, United States, 1997:384~393.
 - 15 Bose N K. editor. *Multidimensional systems theory [M]*. Dordrecht, Holland, Reidel Publishing Company, 1985,184~232.
 - 16 Wu Wen-Tsun. Principles of mechanical theorem proving[J]. *Journal of Automated Reasoning*, 1986,2(3):221~252.
 - 17 Kondo K. Algebraic method for manipulation of dimensional relationships in geometric models[J]. *Computer Aided Design*, 1992,24(3):141~147.
 - 18 Wang D. Decomposing polynomial systems into simple systems [J]. *Journal of Symbolic Computation*, 1998,25(3):295~314.
 - 19 Lazard D. On theories of triangular sets[J]. *Journal of Symbolic Computation*, 1999,28(1-2):105~124.
 - 20 Sturmfels B. Introduction to resultants[A]. In: *Proceedings of Symposia in Applied Mathematics [C]*, San Diego, California, USA,1998,53:25~39.
 - 21 Emiris I Z, Verschelde J. How to count efficiently all affine roots of a polynomial system[EB/OL]. <http://www.inria.fr/rrrt/rr-3212.html>,1997.
 - 22 Ivan E Sutherland. Sketchpad: A man machine graphical communication system [A]. In: *Proceedings-Spring Joint Computer Conference [C]*, Detroit, Michigan, USA, 1963:329~346.
 - 23 Borning A H. The programming language aspects of ThingLab, a constraint oriented simulation laboratory [J]. *ACM Transactions on Programming*, 1981,3(4):353~387.
 - 24 Durand C, Hoffmann C M. A systematic framework for solving geometric constraint analytically [J]. *Journal of Symbolic Computing*, 2000,30(5):483~520.
 - 25 Hoffmann C M, Bo Yuan. On spatial constraint solving approaches [EB/OL]. <http://www.cs.purdue.edu/homes/cmh/distribution/PubsCons.html>,2000.
 - 26 Durand C, Hoffmann C M. Variational constraints in 3D[A]. In: *Processings International Conference on shape Modeling and Application [C]*, Aizu, Japan, 1999:90~97.
 - 27 Podgorelec D, Zalik B. A geometric constraint solver with decomposable constraint set [EB/OL]. <http://wscg.zcu.cn/wscg2000/wscg-2001-full.htm>,2001.
 - 28 Dennis J E, Schnabel R B. *Numerical methods for unconstrained optimization and nonlinear equations [M]*. New Jersey:Prentice-Hall,1983.
 - 29 Ge J X, Gao X S, Chou S C. Geometric constraint satisfaction using optimization methods[J]. *Computer Aided Design*, 2000,31(14):867~879.
 - 30 周明,孙树栋. *遗传算法原理及应用[M]*. 北京:国防工业出版社,1999.
 - 31 Morgan A. Polynomial continuation and its relationship to the symbolic reduction of polynomial systems[A]. In: B. Donald, D. Kapur, and J. Mundy, editors. *Symbolic and Numerical Computation for Artificial Intelligence [M]*. New York:Academic Press,1992.



刘生礼 1975年生,2000年于浙江大学计算机系攻读博士学位。主要研究领域为参数化设计、约束求解等。



唐敏 1974年生,副教授,1999年获浙江大学计算机系博士学位。主要从事几何造型、曲面求交等领域的研究工作。



董金祥 1945年生,教授,博士生导师,1969年获浙江大学数学力学系学士学位。主要从事CIMS、用户界面、图形学等领域的研究工作。