

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷



[12] 发明专利申请公开说明书

G06F 7/60

G06F 17/10

G06F 17/50

G06F 17/60

[21] 申请号 03817011.6

[43] 公开日 2005 年 9 月 14 日

[11] 公开号 CN 1668996A

[22] 申请日 2003.5.21 [21] 申请号 03817011.6

[30] 优先权

[32] 2002. 5. 22 [33] US [31] 10/153,070

[86] 国际申请 PCT/US2003/016039 2003. 5. 21

[87] 国际公布 WO2003/100598 英 2003. 12. 4

[85] 进入国家阶段日期 2005. 1. 17

[71] 申请人 大木系统公司

地址 美国纽约州

共同申请人 中兴电工机械股份有限公司

[72] 发明人 江晓东 李 华

[74] 专利代理机构 上海智信专利代理有限公司

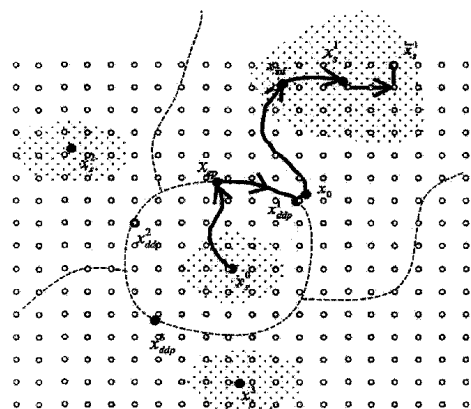
代理人 王 洁

权利要求书 17 页 说明书 55 页 附图 5 页

[54] 发明名称 解大型连续性或离散性最佳化问题的动态方法

[57] 摘要

在目标函数可为封闭模式或黑盒模式的一般最佳化问题中，找到全局最佳解的动态方法系包括有以下步骤：首先从一个起始点开始而以确定性方法来寻找一局部最佳解，然后再从此局部最佳解来寻找到另一组局部最佳解，直到从任一点开始的所有局部最佳解被找到为止，并接着从所有局部最佳解集合中求得全局最佳解。



I S S N 1 0 0 8 - 4 2 7 4

1. 一种基于连续性的动态方法，此方法可解决一般无限制条件最佳化问题，此最佳化问题有一个目标函数及它的问题解空间是有限的，求解步骤如下：

a) 构建一个非线性动态系统满足下列条件：此系统的每一个稳定平衡点是此最佳化问题的一个局部最佳解；

b) 搜寻一般无限制条件最佳化问题的所有局部最佳解的方法，步骤如下：

i) 以起始点开始；

ii) 寻找一个局部最佳解；

iii) 从步骤 (b)、(ii)找到的局部最佳解开始，运用基于出口点或动态分解点的方法，逐层地寻找所有的局部最佳解集合，也就是寻找所建构非线性动态系统的稳定平衡点集合；

c) 藉由比较步骤 (b)、(iii)所搜寻到的稳定平衡点集合，比较所对应的个别目标函数值，选择其中一个具有最小值的最佳解，输出此点为全局最佳解。

2. 根据权利要求1所述的方法，其特征在于，一般的无限制条件连续性最佳化问题形式如下：最小化 $UC(x)$ ，此处目标函数 $UC:R^n \rightarrow R$ 可为封闭模式或黑盒模式，目标函数 $UC(x)$ 是一个有下界的函数，因此它的全局最佳解是存在的且局部最佳解的数量是有限的。

3. 根据权利要求1所述的方法，其特征在于，一般的无限制条件连续性最佳化问题形式如下：最小化 $UC(x)$ ，此处目标函数 $UC:S \rightarrow R$ 可为封闭模式或黑盒模式， S 是一个离散性的解空间，目标函数 $UC(x)$ 是一个有下界的函数，因此它的全局最佳解是存在的且局部最佳解的数量是有限的。

4. 根据权利要求1所述的方法，其特征在于，在步骤(a)中构建一个非线性动态系统 $\dot{x} = F^U(x)$ ，其中 $x \in R^n$ ， n 是最佳化问题解空间的大小，构建的动态系统满足下述性质：

(a) 向量空间 $f(x)$ 满足解的唯一性及存在性；

(b) 非线性动态系统的轨迹必须是收敛的，而且收敛至其中一个平衡点；及

(c) 最佳化问题的目标函数值，沿着非线性动态系统的轨迹是递减。

5. 根据权利要求4所述的方法，其特征在于，对一个连续性最佳化问题，具有封闭模式的目标函数 $UC(x)$ ，在步骤(a)构建一个非线性动态系统 $\dot{x} = -\nabla UC(x)$ ，此处 $\nabla(UC):R^n \rightarrow R^n$ 是目标函数 $UC(x)$ 的梯度向量。

6. 根据权利要求4所述的方法,其特征在于,对一个连续性最佳化问题,具有黑盒模式的目标函数 $UC(x)$, 在步骤(a)构建一个非线性动态系统 $\dot{x}_i = \frac{UC(x+h \times e_i) - UC(x)}{h}, i=1,2,\dots,n$, 此处 x 第 i 个组件的导数, 可以用目标函数 $UC(x)$ 的单向差分近似值表示, e_i 是 Cartesian 第 i 个基底向量, h 是一个非常小的数。

7. 根据权利要求4所述的方法,其特征在于,对一个连续性最佳化问题,具有黑盒模式的目标函数 $UC(x)$, 在步骤(a)构建一个非线性动态系统 $\dot{x}_i = \frac{UC(x+h \times e_i) - UC(x-h \times e_i)}{2h}, i=1,2,\dots,n$, 此处 x 第 i 个组件的导数, 可用目标函数 $UC(x)$ 的中央差分近似值表示, e_i 是 Cartesian 第 i 个基底向量, h 是一个非常小的数。

8. 根据权利要求3所述的方法,其特征在于,对一个离散性的最佳化问题,在进行步骤(a)之前,将此离散性最佳化问题转换成一个连续性的最佳化问题。

9. 根据权利要求8所述的方法,其特征在于,转换后的系统需满足下述两条件:

a) 一个离散性最佳化问题的全局最佳解,与转换后连续性最佳化问题的全局最佳解,彼此是靠近的;

b) 一个离散性最佳化问题的局部最佳解集合,与转换后连续性最佳化问题的局部最佳解集合,彼此是靠近的。

10. 根据权利要求8所述的方法,其特征在于,该转换步骤中的转换方式如下:

$$UC_{eq}(x) = UC(x) + \mu \prod_{i=1}^n \sin^2(\pi x_i) \quad x \in R^n, \text{ 此处 } \mu > \frac{L}{2} \text{ 以及 } L \text{ 是原目标函数 } UC(x) \text{ 的}$$

Lipschitz 常数, $UC_{eq}(x)$ 是转换后的目标函数。

11. 根据权利要求8所述的方法,其特征在于,该转换步骤中的转换方式是根据分段线性转换,定义如下:

$$C_{eq}(x) = \begin{cases} C(x) & \text{when } \lfloor x \rfloor = \lceil x \rceil \\ C(\lfloor x \rfloor) + \frac{C(\lceil x \rceil) - C(\lfloor x \rfloor)}{\text{dist}(\lceil x \rceil - \lfloor x \rfloor)} \text{dist}(x - \lfloor x \rfloor) & \text{when } \lfloor x \rfloor \neq \lceil x \rceil \end{cases},$$

此处 $\text{dist}(x, y)$ 是 x 和 y 间的距离, 在大部分的情形下, 系以欧氏距离表示, $x = (x_1, x_2, \dots, x_n) \in R^n$ $\lfloor x \rfloor = (\lfloor x_1 \rfloor, \lfloor x_2 \rfloor, \dots, \lfloor x_n \rfloor)$, $\lceil x_i \rceil$ 表示大于或等于 x_i 的最小整数, $\lfloor x \rfloor = (\lfloor x_1 \rfloor, \lfloor x_2 \rfloor, \dots, \lfloor x_n \rfloor)$, $\lfloor x_i \rfloor$ 表示小于或等于 x_i 的最大整数, $C(x)$ 是原目标函数。

12. 根据权利要求2所述的方法,其特征在于,对一个连续性的最佳化问题,在权利要求1中的步骤(b)、(ii)运用一个混合式局部搜寻方法,其搜寻步骤如下:

a) 从一个起始点 x_0 开始,针对最佳化问题,运用一个有效的局部搜寻算法;

b) 如果局部搜寻算法收敛到一个局部最佳解 x_s , 停止执行; 否则到下一个步骤执行;

c) 用 x_0 当作起始点, 对非线性系统进行积分几个时间步距, 以所获得轨迹的最后一点为终端点;

d) 以终端点当作新的起始点;

e) 执行步骤(a)。

13. 根据权利要求 12 所述的方法, 其特征在于, 步骤(a)运用的局部搜寻算法是一个业已存在的局部搜寻方法。

14. 根据权利要求 3 所述的方法, 其特征在于, 在权利要求 1 的步骤(b)、(ii)中运用一个混合式局部搜寻方法, 其搜寻步骤如下:

a) 从一个起始点 x_0 开始, 针对最佳化问题, 运用一个有效的局部搜寻算法;

b) 如果局部搜寻算法收敛到一个局部最佳解 x_s , 执行步骤(e); 否则到下一个步骤执行;

c) 用 x_0 当作起始点, 对非线性动态系统进行积分几个时间步距, 以所获得的轨迹最后一点为终端点, 以终端点当作新的起始点;

d) 执行步骤(a);

e) 将计算所得的连续性局部最佳解 x_s , 产生 m (≥ 2)个最接近的可行点集合, 执行步骤 e)、(i)或 e)、(ii), 选择最小目标函数值的点为局部最佳解并停止执行; 用此可行点集合当作初值点, 继续执行步骤(f);

f) 针对每个初值点, 运用最佳邻域搜寻方法来分别求得 m 个 x_{s1}, \dots, x_{sm} 局部最佳解;

g) 从其中选择最小目标函数值的点为最佳化问题局部最佳解并停止执行。

15. 根据权利要求 14 所述的方法, 其特征在于, 步骤(a)运用的局部搜寻算法是一个业已存在的局部搜寻方法。

16. 根据权利要求 1 所述的方法, 其特征在于, 在步骤(b)、(iii)中运用一个基于出口点的方法来计算所有局部最佳解, 其搜寻步骤如下:

a) 由已知一个局部最佳解 x_s^0 开始;

b) 设 $j=0$, 初始化局部最佳解集合 $V_s = \{x_s^0\}$, 阶层 j (tier- j)的局部最佳解集合 $V_{new}^j = \{x_s^0\}$;

c) 设阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\phi\}$;

d) 对在 V_{new}^j 中的每一个局部最佳解 x_s^j , 寻找它所有阶层 1(tier-1)的局部最佳解, 并记录新产生阶层 1 (tier-1)的局部最佳解集合于 V_s 及 V_{new}^{j+1} ;

e) 检查最新计算的局部最佳解集合 V_{new}^{j+1} , 如果不是空集合设 $j = j+1$, 并执行步骤(c);

输出 V_s 为包含 x_s^0 所有的局部最佳解集合。

17. 根据权利要求 16 所述的方法, 其特征在于, 在步骤(d)中运用一个基于出口点的方法, 给予一个已知局部最佳解, 来计算所有阶层 1 的局部最佳解, 其搜寻步骤如下:

a) 由已知一个局部最佳解 x_s^0 开始, 初始化局部最佳解集合 $V_s = \{x_s^0\}$, 阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\phi\}$;

b) 针对 x_s^0 , 定义一组搜寻路径 $S_i, i = 1, 2, \dots, m$;

c) 设 $i = 1$; 及

d) 沿着预定的搜寻路径 S_i 移动, 寻找局部最佳解, 步骤如下:

i) 寻找另一个局部最佳解; 如局部最佳解被找到, 以 x_s^i 来表示并继续步(ii); 否则执行(iii);

ii) 检查 x_s^i 是否曾被找到, $x_s^i \in V_s$, 如果尚未被找到过则设 $V_s = V_s \cup \{x_s^i\}$ 及 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_s^i\}$;

iii) 设 $i = i+1$; 及

iv) 重复执行步骤(i)到步骤(iii), 直到 $i > m$;

输出 V_s 为包含 x_s^0 的局部最佳解集合。

18. 根据权利要求 17 所述的方法, 其特征在于, 在步骤(d)、(i)中运用一个基于出口点的方法, 给予一个已知局部最佳解, 来计算另一个局部最佳解, 其搜寻步骤如下:

a) 从一个已知的局部最佳解 x_s 开始并运用一条预定的搜寻路径;

b) 沿着预定的搜寻路径, 计算以 x_s 为系统的稳定平衡点而构建的非线性动态系统的出口点 x_{ex} (EP);

c) 设 $x_0 = x_s + (1 + \varepsilon)(x_{ex} - x_s)$, 其中 ε 为一极小值;

d) 从 x_0 开始, 对非线性系统进行积分以获得一条轨迹且求得接口点 x_{inf} , 在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解, 其效果是会优于用轨迹来求解局部最佳解;

e) 从此界面点 x_{inf} 开始, 运用步骤(d)所选的局部搜寻法来搜寻到一个毗连的局部最佳解。

19. 根据权利要求 18 所述的方法, 其特征在于, 步骤(d)中运用的局部搜寻算法是一个业已存在的局部搜寻方法。

20. 根据权利要求 18 所述的方法, 其特征在于, 步骤(b)针对预定的搜寻路径, 来计算出口点, 其步骤如下:

a) 从已知的局部最佳解 x_s 开始; 及

b) 沿着预定的搜寻路径, 寻找目标函数值为第一个局部最大的点, 视其为出口点。

21. 根据权利要求 18 所述的方法, 其特征在于, 步骤(b)针对预定的搜寻路径, 来计算出口点, 其步骤如下:

i) 从已知的局部最佳解 x_s 开始;

ii) 沿着预定的搜寻路径移动, 计算搜寻路径的导数和非线性动态系统向量场在每个时间点的内积值;

iii) 当内积值正负符号变更时, 即由正到负, 则表示已寻找到出口点在 $[t_1, t_2]$ 间, 然后运用在 t_1 或 t_2 时的搜寻路径的所得到的点来近似出口点。

22. 根据权利要求 18 所述的方法, 其特征在于, 步骤(d)进行轨迹搜寻来求解局部最佳解, 其中含有对非线性动态系统进行积分程序。

23. 根据权利要求 18 所述的方法, 其特征在于, 当最佳化问题是离散性, 在步骤(d)进行轨迹搜寻, 其中含有将 x_0 点调向新的点 x_0' , 即远离 x_s , 但接近 $(x_{ep} - x_s)$ 的方向, 然后由 x_0' 开始运用最佳邻域搜寻法。

24. 根据权利要求 1 所述的方法, 其特征在于, 在步骤(b)、(iii)中运用一个动态分解点为基础的方法来计算所有局部最佳解, 其搜寻步骤如下:

a) 由已知一个局部最佳解 x_s^0 开始;

b) 设 $j=0$, 初始化局部最佳解集合 $V_s = \{x_s^0\}$, 阶层 j 的局部最佳解集合 $V_{new}^j = \{x_s^0\}$, 动态分解点集合 $V_{dpp} = \{\phi\}$;

- c) 设阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\phi\}$;
- d) 对在 V_{new}^j 中的每一个局部最佳解 x_s^j , 寻找它所有阶层 1 的局部最佳解, 并记录新产生阶层 1 的局部最佳解集合于 V_s 及 V_{new}^{j+1} ;
- e) 检查最新计算的局部最佳解集合 V_{new}^{j+1} , 如果不是空集合设 $j = j+1$, 并执行步骤 (c);
- 输出集合 V_s 为包含 x_s^0 的局部最佳解集合。

25. 根据权利要求 24 所述的方法, 其特征在于, 在步骤(d)中, 当给予一个已知局部最佳解 x_s^0 , 来计算所有阶层 1 局部最佳解, 其搜寻步骤如下:

- a) 由一个已知局部最佳解 x_s^0 开始, 初始化局部最佳解集合 V_s , 阶层 j 的局部最佳解集合 V_{new}^j , 动态分解点集合 V_{ddp} 及阶层 $j+1$ 的局部最佳解集合 V_{new}^{j+1} ;
- b) 针对 x_s^0 定义一组搜寻路径 S_i , $i = 1, 2, \dots, m_j$, 设 $i = 1$;
- c) 对每一搜寻路径 S_i , 寻找它所对应的局部最佳解, 步骤如下:

i) 沿着搜寻路径 S_i 寻找动态分解点, 如动态分解点 $x_{ddp,i}$ 被找到, 继续下一步; 否则执行(vii);

ii) 检查 $x_{ddp,i}$ 是否曾被找到, $x_{ddp,i} \in V_{ddp}$, 如果曾被找到过则执行(vii); 否则设 $V_{ddp} = V_{ddp} \cup \{x_{ddp,i}\}$ 继续执行下一步骤(iii);

iii) 设 $x_{0,i} = x_s^0 + (1 + \varepsilon)(x_{ddp,i} - x_s^0)$, 其中, 如表示式存在, ε 为一极小值;

iv) 从 $x_{0,i}$ 开始, 对非线性系统进行积分以获得一条轨迹且求得接口点 $x_{inf,i}$, 在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解, 其效果是会优于用轨迹搜寻来求解局部最佳解;

v) 从此界面点 $x_{inf,i}$ 开始, 运用步骤(iv)所选的局部搜寻法来搜寻一个相对于 $x_{ddp,i}$ 毗连的局部最佳解 x_s^i ;

vi) 检查 x_s^i 是否以前被找过, 即 $x_s^i \in V_s$, 如果它尚未被找过, 设 $V_s = V_s \cup \{x_s^i\}$ 及 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_s^i\}$;

vii) 设 $i = i+1$; 及

viii) 重复执行(i)到(vii), 直到 $i > m_j$;

输出集合 V_s 为包含 x_s^0 的局部最佳解集合。

26. 根据权利要求 25 所述的方法, 其特征在于, 步骤(c)、(iv)进行轨迹搜寻来求解局部最佳解, 其中含有对构建的非线性动态系统进行积分程序。

27. 根据权利要求 25 所述的方法, 其特征在于, 最佳化问题是离散性的, 步骤(c)、(iv)进行轨迹搜寻, 其中将 x_0 点调向新的点 x_0' , 即远离 x_s , 但接近 $(x_{ep} - x_s)$ 的方向, 然后由 x_0' 开始运用最佳邻域搜寻法。

28. 根据权利要求 25 所述的方法, 其特征在于, 步骤(c)、(iv)运用的局部搜寻算法是一个业已存在的局部搜寻方法。

29. 根据权利要求 25 所述的方法, 其特征在于, 在步骤(c)、(i)中, 针对一条给定的搜寻路径, 计算以 x_s 为系统的稳定平衡点的非线性动态系统的动态出口点(DDP), 其搜寻步骤如下:

a) 从已知的局部最佳解 x_s 开始, 沿着预定的搜寻路径移动;

b) 计算出口点, 设出口点为 x_{ex} ;

c) 从 x_{ex} 开始, 应用下述步骤寻找最小距离点(MDP) x_d^0 ; 及

i) 以出口点 x_{ex} 为起始条件, 对非线性动态系统 $\dot{x} = F^U(x)$ 在几个时间步距处进行积分; 设终端点为目前出口点;

ii) 连接轨迹上的目前出口点和局部的最佳解 x_s , 也就是稳定平衡点, 成一直线, 沿着此直线从稳定平衡点寻找目标函数的第一局部最大的点, 视其为修正出口点 x_{ex} , 并以此点代替目前的点, 指定这个点为出口点 x_{ex} ;

iii) 重复执行(i)及(ii), 直到在(ii)中所获得的出口点的 *norm* 值是小于一个比较值;

iv) 此出口点为最小距离点(MDP) x_d^0 ; 及

d) 以 MDP x_d^0 为起始点, 求解非线性代数方程式 $F^U(x) = 0$, 设其解为 x_{ddp} , 则 x_{ddp} 是针对于动态系统稳定平衡点 x_s 和搜寻路径的动态分解点。

30. 一个基于离散性的动态方法, 此方法可解决一般离散性最佳化问题, 此最佳化问题有一个目标函数及它的问题解空间是有限的, 其步骤如下:

a) 构建一个离散性非线性动态系统满足下列条件: 此系统的每一个稳定固定点是此最佳化问题的一个局部最佳解;

b) 搜寻一般的离散性最佳化问题所有的局部最佳解, 步骤如下:

i) 以起始点 x_0 开始;

ii) 寻找一个局部最佳解;

iii) 从步骤 (b)、(ii)中所搜寻到的局部最佳解开始, 寻找所有局部最佳解集合, 也就是所构建离散性非线性动态系统的稳定固定点集合;

c) 藉由比较步骤(b)、(iii)稳定固定点集合中所对应的个别目标函数值, 选择其中一个具有最小值的最佳解, 输出此点为全局最佳解。

31. 根据权利要求 30 所述的方法, 其特征在于, 一般离散性最佳化问题的形式如下: 最小化 $UC(x)$, 此处目标函数 $UC:S \rightarrow R$ 可为封闭模式或黑盒模式, 其中 S 是一个离散解的空间, 目标函数 $UC(x)$ 是一个有下限的函数, 因此它的全局最佳解是存在的且局部最佳解的数量是有限的。

32. 根据权利要求 30 所述的方法, 其特征在于, 一般离散性最佳化问题的形式如下: 最小化 $C(x)$ 且受限于 $G(x) \leq 0$, 此处目标函数 $C:S \rightarrow R$ 可为封闭模式或黑盒模式, S 是一个离散性的解空间, 且限制函数向量 $G:S \rightarrow R^m$ 为封闭模式, 目标函数 $C(x)$ 是一个有下限的函数, 因此在可行区域中它的全局最佳解是存在的且局部最佳解的数量是有限的。

33. 根据权利要求 30 所述的方法, 其特征在于, 步骤(a)构建一个离散性非线性动态系统 $x_{k+1} = f(x_k)$, 其中 $x \in S$, S 是最佳化问题的解空间, 所构建的离散性非线性动态系统满足下述性质:

a) 所构建的离散性非线性动态系统的轨迹是存在且唯一的; 及

b) 所构建的离散性非线性动态系统的轨迹必须是收敛的, 而且收敛至其中一个固定点; 及, 离散性最佳化问题的目标函数值, 沿着离散性非线性动态系统的轨迹是递减的。

34. 根据权利要求 30 所述的方法, 其特征在于, 针对离散性最佳化问题, 在步骤(a)、(ii)中, 运用一个混合式局部搜寻法寻找一个局部最佳解, 其搜寻步骤如下:

a) 从给定起始点 x_0 开始, 运用一个有效的局部反复修正算法, 求解有限制条件离散性最佳化问题;

b) 如果局部搜寻算法收敛到一个局部最佳解, 那么停止; 否则到下一个步骤执行;

c) 使用最佳邻域搜寻法, 从 x_0 开始执行搜寻几个时间步距而得到终点;

d) 将此终点更新为新的起始点, 执行步骤(a)。

35. 根据权利要求 34 所述的方法, 其特征在于, 步骤(a)运用的局部搜寻算法是一个业已存在的局部搜寻方法。

36. 根据权利要求 30 所述的方法, 其特征在于, 当最佳化问题是无限制条件下, 步

骤(b)、(iii)搜寻所有局部最佳解集合，其搜寻步骤如下：

- a) 由已知一个局部最佳解 x_s^0 开始；
 - b) 设 $j=0$ ，初始化局部最佳解集合 $V_s = \{x_s^0\}$ ，阶层 j 的局部最佳解集合 $V_{new}^j = \{x_s^0\}$ ；
 - c) 设阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\phi\}$ ；
 - d) 对在 V_{new}^j 中的每一个局部最佳解 x_s^j ，寻找它所有阶层 1 的局部最佳解，并记录新产生阶层 1 的局部最佳解集合于 V_s 及 V_{new}^{j+1} ；
 - e) 检查最新计算的局部最佳解集合 V_{new}^{j+1} ，如果不是空集合设 $j=j+1$ ，并执行步骤 (c)；
- 输出 V_s 为包含 x_s^0 所有的局部最佳解集合。

37. 根据权利要求 36 所述的方法，其特征在于，步骤(d)中，当给予一个已知局部最佳解 x_s^0 ，寻找所有阶层 1 的局部最佳解，其搜寻步骤如下：

- a) 由已知一个局部最佳解 x_s^0 开始，初始化局部最佳解集合 V_s ，及阶层 $j+1$ 的局部最佳解集合 V_{new}^{j+1} ，给定 x_s^0 的一组搜寻路径 S_i ， $i=1,2,\dots,m_j$ ，设 $i=1$ ；
 - b) 对每一搜寻路径 S_i ，运用一个基于出口点(EP)或动态分解点(DDP)的搜寻方法，来寻找它所对应毗连的另一个局部最佳解；
 - c) 检查 x_s^i 是否以前被找过，即 $x_s^i \in V_s$ ；如果它尚未被找过，设 $V_s = V_s \cup \{x_s^i\}$ 及 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_s^i\}$ ；
 - d) 设 $i=i+1$ ；及
 - e) 重复执行(b)到(d)，直到 $i > m$ ；
- 输出 V_s 为包含 x_s^0 的局部最佳解集合。

38. 根据权利要求 37 所述的方法，其特征在于，在步骤(b)运用一个基于出口点的方法，给予一个已知局部最佳解，来计算另一个局部最佳解，其搜寻步骤如下：

- a) 从一个已知的局部最佳解 x_s 开始，运用一条预定的搜寻路径；
- b) 沿着预定的搜寻路径，计算出出口点 x_{ex} (EP)；
- c) 如果出口点不能被找到，则停止执行；否则进行下一步；

d) 从出口点 x_{ex} 开始, (i) 沿着预定的路径向前一步, 得到新的一点 x_0 , (ii) 设 $x_0 = x_s + (1 + \varepsilon)(x_{ex} - x_s)$, 其中 ε 为一极小值;

e) 从 x_0 开始, 使用最佳邻域搜寻法, 求得一个接口点 x_{inf} , 在此点开始运用一个局部搜寻法, 其结果是会优于最佳邻域搜寻法;

f) 从此界面点 x_{inf} 开始, 运用步骤(e)所选的局部搜寻法来搜寻到一个毗连的局部最佳解。

39. 根据权利要求 38 所述的方法, 其特征在于, 步骤(e)运用的局部搜寻算法是一个业已存在的局部搜寻方法。

40. 根据权利要求 38 所述的方法, 其特征在于, 在步骤(b)针对预定的路径, 计算出口点, 其步骤如下:

i) 从已知的局部最佳解 x_s 开始; 及

ii) 沿着预定的搜寻路径找寻最佳化问题的目标函数值为第一个局部最大的点, 视其为出口点。

41. 根据权利要求 37 所述的方法, 其特征在于, 在步骤(b)运用一个基于动态分解点的方法, 给予一个已知局部最佳解, 来计算另一个局部最佳解, 其搜寻步骤如下:

a) 从一个已知的局部最佳解 x_s 开始, 运用一条预定的搜寻路径;

b) 沿着预定的搜寻路径, 计算动态分解点 x_{ddp} (DDP);

c) 如果动态分解点不能被找到, 则停止执行; 否则进行下一步;

d) 从动态分解点 x_{ddp} 开始, (i) 沿着修正后的路径向前一步调整出口点 x_{ex} , 得到远离 x_s 的一新的点 x_0 , (ii) 设 $x_0 = x_s + (1 + \varepsilon)(x_{ddp} - x_s)$, 其中 ε 为一很小值;

e) 从 x_0 开始, 使用最佳邻域搜寻法, 求得一个接口点 x_{inf} , 在此点运用一个局部搜寻法, 其结果是会优于最佳邻域搜寻法;

f) 从此界面点 x_{inf} 开始, 运用步骤(e)所选的局部搜寻法来搜寻到一个毗连的局部最佳解。

42. 根据权利要求 41 所述的方法, 其特征在于, 步骤(e)运用的局部搜寻算法是一个业已存在的局部搜寻方法。

43. 根据权利要求 41 所述的方法, 其特征在于, 在步骤(b), 针对一条给定的路径, 计算

动态出口点(DDP)，其搜寻步骤如下：

- a) 从已知的局部最佳解 x_s 开始，沿着预定的搜寻路径移动；
- b) 计算出口点，将出口点设为 x_{ex} ；
- c) 如出口点无法找到，则停止执行；
- d) 从 x_{ex} 开始，应用最佳邻域搜寻方法，向前移动 x_{ex} 一时间步骤到新点 x_c ；
- e) 执行下列的射线调整过程并计算 DDP 点：
 - i) 构成一条射线，连接 x_s 、 x_c ；
 - ii) 如向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 可表达成表示式，则计算向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 之夹角 ε ，如果 ε 接近 0° 或 180° ，则 x_{ex} 是动态分解点并停止过程；否则执行步骤 (e)(iv)；
 - iii) 如向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 不可表达成表示式，则修正搜寻路径使得现有之 x_{ex} 落于修正后的路径上，检查 x_{ex} 、 x_c 与 x_s 是否在同一修正后的路径上，如果是的话， x_{ex} 是动态分解点并停止过程；
 - iv) 调整 x_c 到射线上，从 x_s 开始，最大目标函数值之点，设为 x'_c ；
 - v) 如果 x'_c 跟 x_{ex} 非常接近，则 x'_c 是动态分解点并停止执行；否则执行步骤(f)；
- f) 把 x'_c 当作出口点 $x'_c \rightarrow x_{ex}$ ，执行步骤(d)。

44. 根据权利要求 32 所述的方法，当限制条件为封闭模式的离散性最佳化问题，在步骤 (a) 前建构一个系统，以利找寻所有的可行单位元，此所建构的连续性非线性动态系统满足条件：

- a) 每个基于连续性非线性动态系统的轨迹必须是收敛，而且收敛到其中一个平衡流形；和
- b) 当一个集合是连续性非线性动态系统的一个稳定平衡流形，若且唯若这集合是此有限制条件离散性最佳化问题的一个可行单位元。

45. 根据权利要求 32 所述的方法，当处理限制条件为封闭模式的离散性最佳化问题，在权利要求 30 所述的步骤(b)(ii)计算一个局部最佳解，其步骤如下：

- a) 从给定起始点 x_0 开始，可能不满足部份限制条件；
- b) 经由下述步骤，找寻一个局部最佳解；

i) 设 $k=1$;

ii) 计算 $U(k)$

$$U_i(k) = \begin{cases} \lambda_i(K-1) & \text{if } g_i(x_{k-1}) \text{ is violated} \\ 0 & \text{otherwise} \end{cases}$$

此处 $\lambda_i(K)$ 是一个非负的调整因子, 用来驱动搜寻方向远离前一个不适者;

iii) 如果 x_0 在可行单位元中, 则用有限制条件较佳邻域搜寻方向, 及

$$\Delta_x L(x, \lambda) = \begin{cases} x' - x, x' \in \{Neighbor_of_x \text{ and } L(x', \lambda) \leq L(x, \lambda)\} \\ 0 & \text{if otherwise} \end{cases}$$

来计算 $\Delta_x L(x, \lambda)$; 否则用原解空间的较佳邻域搜寻方向来计算 $\Delta_x L(x, \lambda)$, 其中 $L(x, \lambda)$ 为 Lagrange 函数;

iv) 如果 $U(k)$ 及 $\Delta_x L(x, \lambda)$ 为非常小的值, 则至步骤(c); 否则依

$$x^{k+1} = x^k + \Delta_x L(x^k, \lambda^k)$$

$$\lambda^{k+1} = \lambda^k + U(k) \quad \text{修正 } x \text{ 及 } \lambda \text{ 并执行下一步;}$$

v) 设 $k = k+1$, 以预定之递回次数, 重复执行步骤(ii)至步骤(iv);

c) 如步骤(b)的搜寻是收敛的, 则现存点 x_0 为局部最佳解。

46. 根据权利要求 32 所述的方法, 在权利要求 30 所述的步骤(b)(ii)从一个不可行点开始运用一个局部优化方法计算一个局部最佳解。

47. 根据权利要求 44 所述的方法, 当处理限制条件为封闭模式的离散性最佳化问题, 在权利要求 30 所述的步骤(b)(iii)计算所有局部最佳解, 其步骤如下:

a) 以一个已知局部最佳解 x_s^0 开始;

b) 设 $j=0$, 初始化局部最佳解集合 $V_s = \{x_s^0\}$, 阶层 0(tier-0)的可行单位元集合 $V_0^j = \{x_s^0\}$;

c) 设阶层 $j+1$ 的可行单位元集合 $V_0^{j+1} = \{\phi\}$;

d) 对在 V_0^j 中的每一个可行点 x_0^j , 寻找包含 x_0^j 可行单位元内所有局部最佳解集合, 并寻找所有毗连的可行单位元:

i) 从一个已知的局部最佳解 x_0^j 开始, 来寻找最佳化问题中, 落于可行单位元内所有的局部最佳解;

- ii) 针对 x_0^j 定义一组搜寻路径 $S_i^j \quad i=1,2,\dots,k$;
- iii) 对每一搜寻路径 S_i^j , 寻找一毗连的可行单位元步骤如下:
- 1) 设 $i=1$;
 - 2) 找寻毗连可行单位元内的一点 $x_{0,i}^j$;
 - 3) 从 $x_{0,i}^j$ 开始, 运用一个局部搜寻法, 求解局部最佳化解 $x_{s,i}^j$, 检查 $x_{s,i}^j$ 是否以前被找过, 即 $x_{s,i}^j \in V_s$, 如果它尚未被找过, 设 $V_s = V_s \cup \{x_{s,i}^j\}$ 及 $V_0^{j+1} = V_0^{j+1} \cup \{x_{0,i}^j\}$;
 - 4) 设 $i=i+1$; 及
 - 5) 重复执行(d)(iii)(2)到(d)(iii)(4), 直到 $i > k$;

e) 检查最新计算的可行点集合 V_0^{j+1} , 如果不是空集合, 则设 $j=j+1$; 重复执行步骤(c)到步骤(d); 否则输出 V_s 为所有局部最佳解的集合。

48. 根据权利要求 47 所述的方法, 其特征在于, 在步骤(d)(iii)(2)寻找毗连的可行单位元内的一点, 其步骤如下:

a) 以起始点 $x_0 + \varepsilon \times S_i$ 开始, 其中 ε 为一极小值, 将连续性非线性动态系统延着逆时轨迹积分来找寻系统不稳定平衡流形中的一点, 如找到此点设为 x_d^i , 并执行下一步骤; 否则停止执行;

b) 设定 $x_p^i = x_d^i + (1 + \varepsilon)(x_d^i - x_0)$, ε 是一个小的数目;

c) 以起始点 x_p^i 开始运用下列的步骤来寻找一个分离的可行单位元并找寻出其中一个可行点 x_0^i ;

i) 用 x_0 当做起始条件, 对所建构的连续性非线性系统在几个时间步距处进行积分, 获得最后一点为终端点的轨迹;

ii) 如果轨迹收敛到一个可行点, 则停止执行;

iii) 否则运用反复改进局部搜寻法;

iv) 如果局部搜寻法收敛到一个可行点, 则停止执行;

v) 以终端点当作起始点 x_0 , 执行步骤(c)(i)。

49. 根据权利要求 47 所述的方法，其特征在于，在步骤(d)(iii)(3)运用一个混合式局部搜寻方法，找寻位于一个可行单位元内的局部最佳解，其搜寻步骤如下：

- a) 从一个起始点 x_0 开始，针对离散性最佳化问题，运用一个反复改进局部搜寻算法；
- b) 如果局部搜寻算法收敛到一个局部最佳解，则停止执行；
- c) 使用最佳邻域搜寻法或较佳邻域搜寻法，从 x_0 开始，持续搜寻数个步骤后，记录最后一点为终端点；
- d) 以终端点当作起始点；及
- e) 执行步骤(a)。

50. 根据权利要求 49 所述的方法，其特征在于，步骤(a)运用的局部搜寻算法是一个业已存在的局部搜寻方法。

51. 根据权利要求 47 所述的方法，其特征在于，在步骤(d)(i)寻找位于一个可行单位元内的局部最佳解集合，其搜寻步骤如下：

- a) 由已知一个局部最佳解 x_s^0 开始， V_s 是已找到的局部最佳解的集合；
- b) 设 $j=0$ ，初始化一个可行单位元内局部最佳解集合 $V_s^c = \{x_s^0\}$ ，阶层 j 的局部最佳解集合 $V_{new}^j = \{x_s^0\}$ ；
- c) 设阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\emptyset\}$ ；
- d) 对在 V_{new}^j 中的每一个局部最佳解 x_s^j ，寻找包含 x_s^j 可行单位元内所有阶层 1 (tier-1) 的局部最佳解。并记录新产生阶层 1 (tier-1) 的局部最佳解集合于 V_s ， V_{new}^{j+1} 及 V_s^c ；
- e) 检查最新计算的局部最佳解集合 V_{new}^{j+1} ，如果不是空集合设 $j = j+1$ ，并执行步骤(c)；

输出 V_s 为包含 x_s^0 所有的局部最佳解集合。

52. 根据权利要求 51 所述的方法，其特征在于，在步骤(d)找寻位于阶层 1 的局部最佳解集合，其搜寻步骤如下：

- a) 由一个已知的局部最佳解 x_s^0 开始， V_s 是已找到的局部最佳解的集合，初始化一个可行单位元内局部最佳解集合 V_s^c ，阶层 j 的局部最佳解集合 V_{new}^j ，对 x_s^0 定义一组搜寻路径 $S_i, i=1,2,\dots,m$ 。设 $i=1$ ；

b) 从一个已知的局部最佳解 x_s^0 开始, 沿着预定的搜寻路径 S_i 移动, 运用基于出口点方法或动态分解点方法寻找另一个局部最佳解;

c) 如局部最佳解被找到, 以 x_s^i 表示并继续下一步; 否则执行步骤(e);

d) 检查 x_s^i 是否曾被找到 ($x_s^i \in V_s$), 如果尚未曾被找到过则设 $V_s = V_s \cup \{x_s^i\}$,
 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_s^i\}$ 及 $V_s^c = V_s^c \cup \{x_s^i\}$;

e) 设 $i = i + 1$;

f) 重复执行步骤(b)到步骤(e), 直到 $i > m$;

输出 V_s 为包含 x_s^0 所有的局部最佳解集合。

53. 根据权利要求 52 所述的方法, 其特征在于, 在步骤(b)运用一个基于出口点的方法, 给予一个已知局部最佳解, 来计算另一个局部最佳解, 其搜寻步骤如下:

a) 从一个已知的局部最佳解 x_s 开始, 并运用一条预定的搜寻路径;

b) 沿着预定的搜寻路径, 计算相对应的出口点 x_{ex} (EP); 如果出口点不能被找到, 则停止执行;

c) 从出口点 x_{ex} 开始, (i) 延着预定的路径向前一步(远离 x_s), 得到新的一点 x_0 , (ii) 设 $x_0 = x_s + (1 + \varepsilon)(x_{ex} - x_s)$, 其中 ε 为一极小值;

d) 从 x_0 开始, 利用有限制条件的最佳邻域搜寻法, 求得一个接口点, 在此点运用一个有限制条件的局部搜寻法, 其结果是会优于有限制条件的最佳邻域搜寻法;

e) 从此界面点 x_{mf} 开始, 运用步骤(e)所选有限制条件的局部搜寻法来搜寻到一个毗连的局部最佳解。

54. 根据权利要求 53 所述的方法, 其特征在于, 步骤(d)运用的局部搜寻算法是一个业已存在的局部搜寻方法。

55. 根据权利要求 53 所述的方法, 其特征在于, 在步骤(b)针对预定的路径, 计算出出口点, 其步骤如下:

i) 从已知的局部最佳解 x_s 开始, 沿着预定的路径搜寻目标函数值为第一个的局部最大的点, 视其为出口点;

ii) 在过程中检查是否违反限制条件; 如违反限制条件, 则停止执行并说明出口点无法找到。

56. 根据权利要求 52 所述的方法,其特征在于,在步骤(b)运用一个基于动态分解点(DDP)的方法,给予一个已知局部最佳解,来计算另一个局部最佳解,其步骤如下:

- a) 从一个已知的局部最佳解 x_s 开始,并运用一条预定的搜寻路径;
- b) 沿着预定的路径计算所对应的动态分解点 x_{ddp} (DDP); 如果动态分解点不能被找到,则停止执行;
- c) 从动态分解点 x_{ddp} 开始, (i) 沿着修正后调整出口点 x_{ex} 的路径向前一步,得到远离 x_s 的一新点 x_0 , (ii) 设 $x_0 = x_s + (1 + \varepsilon)(x_{ddp} - x_s)$, 其中 ε 为一很小值;
- d) 从 x_0 开始,使用有限制条件的最佳邻域搜寻法,求得一个接口点 x_{inf} , 在此点运用一个局部搜寻法,其结果是会优于有限制条件的最佳邻域搜寻法;
- e) 从此界面点 x_{inf} 开始,运用步骤(d)所选的有限制条件局部搜寻法来搜寻到一个毗连的局部最佳解。

57. 根据权利要求 56 所述的方法,其特征在于,步骤(d)运用的局部搜寻算法是一个业已存在的局部搜寻方法。

58. 根据权利要求 56 所述的方法,其特征在于,在步骤(b),针对一条给定的路径,计算动态出口点 (DDP),其搜寻步骤如下:

- a) 从已知的局部最佳解 x_s 开始,沿着预定的搜寻路径移动;
- b) 侦测到出口点,将出口点设为 x_{ex} ; 在过程中检查是否违反限制条件或出口点无法找到;如是,则停止执行;
- c) 从 x_{ex} 开始,应用有限制条件的最佳邻域搜寻方法,向前移动 x_{ex} 一时间步骤到新点 x_c ;
- d) 执行下列的射线调整过程并计算 DDP 点:
 - i) 构成一条直线,连接 x_s 、 x_c ;
 - ii) 如果向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 可表达成表示式,则计算向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 之夹角 ε ; 如果 ε 接近 0° 或 180° , 则 x_{ex} 是动态分解点并停止过程; 否则执行步骤 (d)(iv);
 - iii) 如果向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 不可表达成表示式,则修正搜寻路径使得现

有的 x_{ex} 落于修正后的路径上,检查 x_{ex} 、 x_c 与 x_s 是否在同一修正后的路径上;如果是的话, x_{ex} 是动态分解点并停止过程;

iv) 调整 x_c 到直线上,从 x_s 开始第一个局部最大目标函数值的点,设为 x'_c ;

v) 如果 x'_c 跟 x_{ex} 非常接近,则 x'_c 是动态分解点并停止执行;否则执行步骤(e);

e) 把 x'_c 当作出口点, $x'_c \rightarrow x_{ex}$,执行步骤(d)。

解大型连续性或离散性最佳化问题的动态方法

发明背景

发明领域

本发明系相关于数值分析领域和非线性规划问题的最佳化技术。本发明尤其相关于用于有系统地求得连续性或离散性最佳化问题的所有的局部最佳解以及唯一全局最佳解的动态方法。

相关技术之描述

许多定量性 (*quantitative*) 的决策、设计、生产、规划和排程的问题，在科学、工程学和经济学领域中，可视为连续性或离散性最佳化问题。一般情形，系统的总表现往往可以多变量函数来做定量性表示，此函数可称此系统的目标函数。从这个观点而言，最佳化问题的最佳解时常以实数向量来表达，此实数向量所表达的可以是一个决策，或设计，或规划。在最佳化问题中的整个搜寻空间 (*search space*) 中，最佳解是满足约束条件并使得目标函数值为最小 (或最大)。如果这个向量存在的话，此向量系被称为“全局最佳解”。在实际应用中的各种最佳化问题是非常复杂的，它的目标函数通常是非线性并包含大量的变量；因此在搜寻空间中搜寻全局最佳解的任务是非常困难的。主要的挑战在于 (高维度空间中) 有许多局部最佳解；全局最佳解只是许多局部最佳解中的一个，而且全局最佳解与局部最佳解两者皆具有同样的局部特性，这使得分辨所有搜寻到的局部最佳解是否为全局最佳解的工作较为困难。

一般而言，最佳化问题的解空间拥有有限个 (通常非常大) 或无限个可行解。在全部的可行解中，只有一个且为唯一的全局最佳解。一般而言，局部最佳解的正确数目是不可知的，而且可能是非常多的。此外，局部最佳解和全局最佳解的目标函数值可能差别非常大。因此这些强烈的动机促使我们发展出多种有效方法，来搜寻全局最佳解。

本发明接下来将主要讨论离散性最佳化问题。解决离散性最佳化问题的

任务是非常具有挑战性的。它们具有 *NP-hard* 特性，这意味计算时间跟问题的规模大小成指数关系增加。除此之外，许多离散性问题属于 *NP-complete*，没有现存任何的算法可以有效的解决问题。*NP-complete* 明确的定义可以在文献中找到。大概地说，*NP-complete* 问题是很难计算；任何数值算法在最糟的情况下需要指数关系时间增加才能正确性地搜寻到全局最佳解。

一个普遍被应用到解决离散性最佳化问题的方法，是根据一个反复改进的搜寻算法[1]。其特征如下：从一个初始的可行解开始并在它的邻域搜寻到更好的解。如果较佳解存在的话，则把新解当作初始解，重复上述搜寻过程；否则停止局部搜寻过程。此种最佳解搜寻算法通常会搜寻某一个局部最佳解而停止，它通常无法找到全部解。事实上，现存绝大部分的离散性最佳解搜寻技术，通常在于提供局部最佳解，而非全局最佳解。

针对反复改进的局部搜寻算法 (*iterative improvement local search algorithms*) 的缺点，近年来几个较复杂的局部最佳解搜寻方法已经发展出来。这些算法不但可以脱离局部最佳解而且能够搜寻到更好解。某些绝大部分的搜寻方法“脱离”的机制是运用 *cost-deteriorating* 来脱离局部最佳解。上述这些局部搜寻算法 (*local search algorithms*) 包含：仿真退火法 (*simulated annealing algorithm*)、基因算法 (*genetic algorithm*)、塔布搜寻法 (*Tabu search algorithm*) 和类神经网络法 (*neural network*)。

在许多研究报告中，我们可以发现这些复杂的局部最佳解方法，在实际相关应用问题上，是需要大量的运算而且通常不能找到全局最佳解。

在搜寻全局最佳解的问题领域中，我们发展出几个有效方法来解决下列两个挑战性难题：

- (i) 该如何有效地脱离一个局部最佳解，并有效地移向另外一个局部最佳解；以及
- (ii) 该如何避免再次重复搜寻已知(已找过)的最佳解。

在过去，许多努力曾尝试解决这个问题，但只获得很小的进展。上述难题(i)是困难解决的，现存许多方法全都曾遇到这种困难。问题(ii)与搜寻效率有关，如何解决也是常遭遇到的困难，现存方法也全都曾遇到这种困难。这个问题(ii)通常会降低现存方法的求解效率；重复搜寻同样的局部最佳解；这个动作浪费计算资源。从计算的观点来说，避免重复搜寻同样的局部最佳解对于维持高效率的求解速度是重要的。

发明内容

搜寻全局最佳解(*Global optimization*)的任务,对于广大工程界及科学界是非常重要的。本发明提出唯一可靠的方法来搜寻全局最佳解,此方法采用独创的非线性技术来搜寻所有的局部最佳解,然后从它们找出全局最佳解。

本发明提出的方法也可用来解决一般性最佳化问题,首先找出多个局部最佳解(可为非全部的局部最佳解),然后从中找出最佳解(*Best solution*)。

本发明系发展出一个新的动态方法,它能够以确定性(*Deterministic*)的方法找出离散性或者是连续性的最佳化问题所有的局部最佳解。在本发明中,将详述解决下述各种最佳化问题的动态方法:

- 一个解决在无限限制条件下连续性最佳化问题 (*Unconstrained Continous Optimization Problems*) 的动态方法。最佳化问题的目标函数可为封闭模式 (*closed-form*) 或黑盒模式 (*black-box*)。(所谓黑盒模式 (*black-box*) 的目标函数是指在搜寻空间中,每一点的目标函数值是可以计算的,但却无法以一个封闭模式的表示式来表示目标函数)。
- 一个解决在无限限制条件下离散性最佳化问题 (*Unconstrained Discrete Optimization Problems*) 的动态方法。最佳化问题的目标函数可为封闭模式或黑盒模式。
- 一个解决在有限限制条件下离散性最佳化问题 (*Constrained Discrete Optimization Problems*) 的动态方法。最佳化问题的目标函数系为封闭模式。
- 一个解决在有限限制条件下离散性最佳化问题 (*Constrained Discrete Optimization Problems*) 的动态方法。最佳化问题的目标函数系为黑盒模式。

为了解决上述问题(i),本发明发展一种基于动态分解点的数值方法,其组合了动态分解点搜寻方法,来执行一有系统的搜寻过程,从一个局部最佳解搜寻到另一个局部最佳解。为了解决上述问题(ii),本发明发展一套反重复(*Anti-revisiting*)搜寻方法来避免重复搜寻已知的局部最佳解。本发明所发展的反重复搜寻方法之理论基础是依据本发明所提出的动态分解点(*Dynamical decomposition point DDP*)。

本发明中这些有效的方法可用来克服问题(i)及(ii)，并并入两个动态方法来找寻所有局部最佳解，此两个动态方法包括基于连续性(*continous-based*)的动态方法及基于离散性(*discrete-based*)的动态方法。这两个动态方法的一个显著特性是可与任何已存在的局部最佳求解方法(找寻局部最佳解)合并，以达成在求局部最佳解的计算高效率(并保有搜寻所有局部最佳解的能力)。

本发明的动态方法是可以和一个现成有效率的计算机软件(找寻局部最佳解)相结合，来彼此合作共同找出所有局部最佳解集合。本发明可设计一个接口环境方便与现成的计算机软件相连，不需要重新界定计算机软件的环境(包含使用者图形接口及数据库)，便可搜寻到完整的局部最佳解集合和全局最佳解。本发明对于使用者来说，不需要重新学习一套新的软件。这个利用现有计算机软件的智能财产特性，使得本发明非常具有吸引力。

图示简单说明

图 1 系显示出以基于动态分解点的方法来从连续空间中之一已知局部最佳解中找出一个局部最佳解的图表。

图 2 系显示出以基于出口点的方法来从连续空间中之一已知局部最佳解中找出一个局部最佳解的图表。

图 3 系显示出以基于动态分解点的方法来从离散空间中之一已知局部最佳解中找出一个局部最佳解的图表。

图 4 系显示出以基于出口点的方法来从离散空间中之一已知局部最佳解中找出一个局部最佳解的图表。

图 5 系显示出找出所有局部最佳解的说明。

本发明的详细说明

一般的最佳化问题 (*General Optimization Problems*)

本发明将一般的最佳化问题分类为非线性连续最佳化问题(*Nonlinear continuous optimization problems*)及非线性离散最佳化问题(*Nonlinear discrete optimization problems*)。他们进而分成下列四种(依有无限限制条件再细分)：

- 无限限制条件连续性最佳化问题。最佳化问题的目标函数可为封闭模式或黑盒模式。

- 有限制条件连续性最佳化问题。最佳化问题的目标函数可为封闭模式或黑盒模式。
- 无限制条件离散性最佳化问题。最佳化问题的目标函数可为封闭模式或黑盒模式。
- 有限制条件离散性最佳化问题。最佳化问题的目标函数可为封闭模式或黑盒模式。

无限制条件连续性最佳化问题

一般的无限制条件连续性最佳化问题，形式如下：

$$\text{最小化 } UC(x) \quad (4-1)$$

此处目标函数 $UC:R^n \rightarrow R$ 可为封闭模式或黑盒模式。目标函数是有下限的函数，因此它的全局最小(佳)解是存在的且局部最小(佳)解的数量是有限的。对一个无法以表示式表达的黑盒模式目标函数，其目标函数值只能被计算。此类目标函数可视为一个外部子程序：读取其解向量 x (*solution vector*) 值，然后计算出相对应的目标函数值 $UC(x)$ 。

有限制条件连续性最佳化问题

一般的有限制条件连续性最佳化问题，形式如下：

$$\begin{aligned} &\text{最小化 } C(x) \\ &\text{受限于 } G(x) \leq 0 \end{aligned} \quad (4-2)$$

此处目标函数 $C:R^n \rightarrow R$ 可为封闭模式或黑盒模式。满足不等式 $G(x)$ 状态向量的集合被称为可行区域(*feasible region*)。目标函数 $C(x)$ 是一个有下限的函数，因此它的全局最小(佳)解是存在的且局部最小(佳)解的数量是有限的。

无限制条件离散性最佳化问题

一般的无限制条件离散性最佳化问题，形式如下：

$$\text{最小化 } UC(x) \quad (4-3)$$

此处目标函数 $UC:S \rightarrow R$ 可为封闭模式或黑盒模式。 $UC(x)$ 是一个有下限的函数，因此全局最小(佳)解是存在的且局部最小(佳)解的数量为有限的。

有限制条件离散性最佳化问题

一般的有限制条件离散性最佳化问题，形式如下：

$$\text{最小化 } C(x) \quad (4-4)$$

受限于 $x \in FS$

此处目标函数 $C:S \rightarrow R$ 可为封闭模式或黑盒模式。目标函数 $C(x)$ 是一个有下限的函数，因此它的全局最小(佳)解是存在的且局部最小(佳)解的数量为有限的。限制集合 FS 包含所有的可行解(*feasible solution*)。限制集合 FS 可表示为一组限制方程式或由一个黑盒模式来表示，它也可视成一个外部子程序：只要读取其解向量 x 值，然后计算出相对应的限制满意度(*satisfiability of constraints*)。

显然的，上述的离散性最佳化问题系可以下述通式表达：

$$\begin{aligned} &\text{最小化 } C(x) \\ &\text{受到限制 } G(x) \leq 0 \end{aligned} \quad (4-5)$$

此处目标函数 $C:S \rightarrow R$ 可为封闭模式或黑盒模式。目标函数 $C(x)$ 是一个有下限的函数，因此它的全局最小解是存在的且局部最小解的数量是为有限的。限制函数向量 (*constraint function vector*) $G:S \rightarrow R^m$ 是以封闭模式表示。搜寻空间 S 为有限或无穷可数集合 (*countably infinite set*)。

下文中将针对 (4-1) (4-3) (4-5) 各类最佳化问题，分别描述说明本发明的动态方法。

5. 解决无限制条件连续性最佳化问题的动态方法

在此说明解决无限制条件连续性最佳化问题(4-1)的动态方法。

为解决无限制条件连续性最佳化问题(4-1)，本发明发展出一个(利用下列非线性动态系统相关之轨迹特性)动态方法，来搜寻所有的局部最佳解。

$$\dot{x} = f(x) \quad (5-1)$$

其中 $x \in R^n$ ，向量场(*vector field*)空间 $f(x)$ 满足解的唯一性及存在性。

本发明针对运用非线性动态系统(5-1)的轨迹来找出最佳化问题(4-1)所有的局部最佳解的方法，提出下述充分条件。

(LC1)- 非线性动态系统(5-1)的轨迹必须是收敛的, 而且收敛至其中一个平衡点; 及

(LC2)-最佳化问题(4-1)的每一个局部最佳解是系统(5-1)的一个稳定平衡点(*stable equilibrium point*), 反之亦然。

(LC3)- 最佳化问题(4-1)的目标函数值, 沿着非线性动态系统(5-1)的轨迹是递减。

我们接下来说明, 给予一个具有解析式的封闭模式或黑盒模式目标函数的无限制条件的连续性最佳化问题(4-1), 我们可建构一个满足(LC1)~(LC3)充分条件的非线性动态系统(5-1)。

我们首先发展下面两个定理, 为寻找非线性动力系统提供依据。

一个函数 $E(x)$ 如果满足如下两个条件, 我们称之为(5-1)的能量函数: (i) 沿非线性动力系统(5-1)的任意系统轨线, 函数 $E(x)$ 的导数非正, 即 $\dot{E}(x) \leq 0$; (ii) 如果 $x(t)$ 不是非线性动力系统(5-1)的一个平衡点, 则集合 $\{t \in R: \dot{E}(x) = 0\}$ 沿 $x(t)$ 在 R 中具有零测度。

定理 5-1: 如果动力系统(5-1)具有能量函数, 则条件(LC1)满足;

定理 5-2: 如果无限制条件的连续最佳化问题(4-1)的目标函数是非线性动力系统(5-1)的一个能量函数, 则条件(LC2)和(LC3)满足。

对一个最佳化问题(4-1)及具有可分析式的封闭模式目标函数 UC , 我们可以严格证明一例满足(LC1)~(LC3)充分条件的非线性动态系统是如下:

$$\dot{x} = -\nabla UC(x) \quad (5-2)$$

此处 $\nabla(UC): R^n \rightarrow R^n$ 是目标函数 UC 的梯度向量(*gradient vector*)。一般说来, 除了此非线性动态系统(5-2)外, 应该存在其它的非线性动态系统满足(LC1)~(LC3)的充分条件。我们将此类非线性动态系统以下述一般通式表达:

$$\dot{x} = F_C^U(x) \quad (5-3)$$

对一个最佳化问题(4-1), 其有黑盒模式的目标函数 UC , 如何建构一个满足(LC1)~(LC3)充分条件的非线性动态系统是困难的。我们发展了许多方法来解决它; 其中一个依据黑盒模式的目标函数 UC , 将其利用分割差分(*divided-difference*)为基础来计算:

$$\dot{x}_i = \frac{UC(x+h \times e_i) - UC(x)}{h}, i=1,2,\dots,n \quad (5-4)$$

此处对 x 第 i 个组件的导数，可以用目标函数 $UC(x)$ 的单向差分(one-sided differences)近似值表示， e_i 是 Cartesian 第 i 个基底向量， h 是一个非常小的数。另外吾人对具有黑盒模式的目标函数，也可用中央差分 (central differences) 方法来建构另一个非线性动态系统，描述如下：

$$\dot{x}_i = \frac{UC(x+h \times e_i) - UC(x-h \times e_i)}{2h}, i=1,2,\dots,n \quad (5-5)$$

此处对 x 第 i 个组件的导数，可以用目标函数 $UC(x)$ 的中央差分近似值表示， e_i 是 Cartesian 第 i 个基底向量， h 是一个非常小的数。

对一个最佳化问题(4-1)，其有黑盒模式的目标函数，除了上述(5-4)及(5-5)，应该存在其它的非线性动态系统满足(LC1)~(LC3)的充分条件。我们将此类非线性动态系统以下列一般通式表达：

$$\dot{x} = F_{UC}^U(x) \quad (5-6)$$

为了清楚及方便说明，我们将非线性动态系统(5-3)其具有可分析式封闭模式目标函数，及非线性动态系统(5-6)其具有黑盒模式的目标函数，用以下列一般非线性通式来表达：

$$\dot{x} = F^U(x) \quad (5-7)$$

我们以下说明相关于非线性动态系统(5-7)的一些基本定义及理论。对 $x \in R^n$ ，系统(5-7)的解由 $t=0$ 起称为轨迹，以 $\phi_t(x): R \rightarrow R^n$ 表示。

如果 $F^U(x)=0$ ，状态向量 $x^* \in R^n$ 称为系统 (5-7) 的一个平衡点。如果 $F(\cdot)$ 在 x^* 的 Jacobian 矩阵，其所有的特征值(eigenvalues)没有实数部分(with a zero real part)为零，我们则称平衡点 $x^* \in R^n$ 是双曲线型(hyperbolic)。如果双曲线型(hyperbolic)平衡点的 Jacobian 矩阵，有 k 个正实数之特征值，我们称此点为类型- k 的平衡点。对一个平衡点，如果它所对应的 Jacobian 矩阵的所有特征值的实数部分皆为负值，则该点是稳定平衡点(stable equilibrium point)，反之如果它所对应的 Jacobian 矩阵的所有特征值的实数部分皆为正值，则该点是不稳定平衡点(unstable equilibrium point)。一个类型- k 的双曲线型(hyperbolic)

平衡点, 它的稳定流形 $W^s(x^*)$ 和不稳定流形 $W^u(x^*)$ 定义如下:

$$W^s(x^*) = \{x \in R^n : \lim_{t \rightarrow \infty} \phi_t(x) = x^*\} \quad (\text{稳定流形})$$

$$W^u(x^*) = \{x \in R^n : \lim_{t \rightarrow -\infty} \phi_t(x) = x^*\} \quad (\text{不稳定流形})$$

稳定流形 $W^s(x^*)$ 和不稳定流形 $W^u(x^*)$ 的面积维数分别是 k 及 $n-k$ 。

如果系统 (5-7) 的每一个轨迹从此集合 K 内的某一点出发且都停留在此集合里 (for all $t \in R$), 则我们称集合 $K \in R^n$, 是此非线性系统 (5-7) 的不变量集合 (invariant set)。如果系统的每个轨迹都收敛到其中一个的平衡点, 则此动态系统被称为完全稳定的 (completely stable)。

我们称稳定平衡点 x_s 的稳定区域 (stability region) (在数学上也称吸引域 (region of attraction)) 是所有点其轨迹收敛到 x_s 的集合。数学上以 $A(x_s) := \{x \in R^n : \lim_{t \rightarrow \infty} \phi_t(x) = x_s\}$ 来表示, 此处 $\phi_t(x)$ 表示系统 (5-7) 的一个轨迹且此轨迹是在时间 $t=0$ 从 x 点出发的。稳定平衡点 x_s 的准稳定区域 (quasi-stability region) $A_p(x_s)$, 以 $A_p(x_s) = \text{int}(\overline{A(x_s)})$ 表示, 其中 $\overline{A(x_s)}$ 为稳定区域 $A(x_s)$ 的闭包 (closure), 并且 $\text{int}(\overline{A(x_s)})$ 是 $\overline{A(x_s)}$ 的内部 (interior)。从拓扑的观点出发, $A(x_s)$ 和 $A_p(x_s)$ 是路径连续的不变开集合; 并且, 它们是 R^n 的微分同胚。

如下定理 [3], 连同上述两个定理, 给出了本发明发展的动态方法的理论基础, 该方法利用满足条件 (LC1) 至 (LC3) 的非线性动力系统 (5-7) 解决无限制条件的连续性最佳化问题 (4-1)。

定理 5-3: 考虑由 (5-7) 描述并且满足条件 (LC1) 的非线性动力系统。设 $\partial A_p(x_s)$ 是 x_s 的准稳定区域, $x_d \in \partial A_p(x_s)$ 是一个动态分解点。如果 $\partial A_p(x_s)$ 上的平衡点的稳定和不稳定流形的交叉满足横断性条件, 则存在并且只存在 (5-1) 的另一个稳定平衡点 \bar{x}_s 满足:

- (1) $x_d \in \partial A_p(\bar{x}_s)$ 是对应于 \bar{x}_s 的一个动态分解点; 以及
- (2) $W^u(x_d) \cap A_p(x_s) \neq \emptyset$ 以及 $W^u(x_d) \cap A_p(\bar{x}_s) \neq \emptyset$

其中, 动态分解点指位于非线性动力系统 (5-7) 的准稳定边界上的一个类型-1 平衡点; $W^u(x_d)$ 代表动态分解点 x_d 的不稳定流形。

5.1 混合式局部搜寻方法(Hybrid Local Search Method)

我们发展一个混合式局部搜寻方法，其结合一个以轨迹为基础的方法及一个有效的局部方法，来可靠的找寻局部最佳解。我们将此混合式局部搜寻方法配入我们所发明的动态方法内。

一个混合式的局部搜寻法，从一个起始点开始(在一个连续的空间中)来找寻无限制条件连续性最佳化问题(4-1)的局部最佳解，如下表示：

给予：一个起始点 x_0 。

目的：可靠的计算无限制条件连续性最佳化问题(4-1)的一个局部最佳解。

步骤一、从一个起始点 x_0 开始，针对无限制条件连续性最佳化问题(4-1)，运用一个有效的局部搜寻法。

步骤二、如果局部搜寻算法收敛到一个局部最佳解，则停止执行；否则执行下一步骤。

步骤三、用 x_0 当做起始条件，对非线性动态系统(5-7)在几个时间步距(time-steps)处进行积分，获得最后一点为终端点。以终端点当作起始点，执行步骤一。

5.2 计算出口点(Exit Point)的方法

从一个起始点开始(一个已知无限制条件连续性最佳化问题(4-1)局部最佳解 x_s ，也就是非线性动态系统(5-7)的稳定平衡点)，运用非线性动态系统(5-7)，来找寻无限制条件连续性最佳化问题(4-1)的出口点，称之为出口点搜寻法，如下表示：

给予：一个已知局部最佳解 x_s ，预定的搜寻路径，及相关的非线性动态系统(5-7)。

目的：由预定的搜寻路径及动态系统(5-7)的稳定平衡点 x_s ，计算所对应的出口点。

从已知的局部最佳解 x_s 开始，沿着预定的搜寻路径移动，直到连续性目标函数(4-1)达到(第一)局部最大值的点上，则此点为出口点。

本发明另外提出一个计算出口点的方法：从 x_s 开始，延着搜寻路径移动，计算搜寻路径的导数和非线性动态系统(5-7)向量场在每个时间点的内积值，当内积值正负符号变更时(由正到负)，则表示已寻找到出口点在 $[t_1, t_2]$ 间。然后运用在 t_1 或 t_2 时的搜寻路径的所得到的点来近似出口点。

5.3 计算动态分解点(dynamical decomposition point)的方法

从一个起始点开始(一个已知无限制条件连续性最佳化问题(4-1)局部最佳解 x_s ，也就是非线性动态系统(5-7)的稳定平衡点)，运用非线性动态系统(5-7)，来找寻非限制连续最佳化问题(4-1)的动态分解点，称之为动态分解点(DDP)搜寻法，如下表示：

给予：一个最佳化问题(4-1)的已知局部最佳解 x_s ，一条预定的搜寻路径，及相关的非线性动态系统(5-7)。

目的：由预定的搜寻路径及动态系统(5-7)的稳定平衡点 x_s ，计算出所对应的动态分解点。

步骤一、从已知的局部最佳解 x_s 开始，沿着一条预定的搜寻路径 $\varphi_t(x_s)$ 移动，侦测到出口点，此点位于动态系统(5-7)稳定平衡点(也就是局部最佳解 x_s)的准稳定边界上(*quasi-stability boundary*)，设出口点为 x_{ex} 。

步骤二、从 x_{ex} 开始，应用下述步骤找寻最小距离点 (*minimum distance point*)，简称 MDP：

(i) 以出口点 x_{ex} 为起始条件，对非线性系统(5-7)在几个时间步距(time-steps)处进行积分；设终端点为目前出口点(*current exit point*)。

(ii) 连接轨迹上的目前出口点和局部的最佳解(也就是稳定平衡点)成一直线。沿着此直线从稳定平衡点找寻目标函数(4-1)的(第一)局部最大的点，视其为修正出口点 x_{ex} ，并以此点代替目前出口点。指定这个点为出口点 x_{ex} 。

(iii) 重复执行(i)及(ii)，直到在(ii)中所获得的出口点，其向量场平均值(*norm of vector field*)(5-7)是小于一个临界值(*threshold*)，也就是数值 $\|F^u(x_{ex})\| \leq \varepsilon$ 。那么这点被视为最小距离点(MDP)。

步骤三、以 $MDP x_d^0$ 为起始点，求解系统(5-7)其非线性代数方程式 $F^U(x)=0$ 。设其解为 x_{dap} ，则 x_{dap} 是针对于动态系统(6-11)局部最佳解 x_s 和搜寻路径 $\varphi_t(x_s)$ 的动态分解点。

5.4 搜寻另一个局部最佳解的基于出口点(EP-based)的方法

针对最佳化问题(4-1)而言，我们发明从一个已知的局部最佳解 x_s 开始来搜寻到另一个局部最佳解的方法，称之为基于出口点(EP-based)的方法。

给予：一个最佳化问题(4-1)，其已知的局部最佳解 x_s 。

目的：搜寻到最佳化问题(4-1)的另一个局部最佳解。

步骤一、从一个已知的局部最佳解 x_s 开始，计算非线性动态系统(5-7)的出口点 x_{ex} 。

步骤二、设 $x_0 = x_s + (1 + \varepsilon)(x_{ex} - x_s)$ 其中 ε 为一极小值。

步骤三、从 x_0 开始，对非线性系统(5-7)进行积分以获得一条轨迹且求得接口点(interface point) x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解，其效果是会优于用轨迹来求解局部最佳解。

步骤四、从此接口点 x_{inf} 开始，运用步骤三所选的有效混合式局部搜寻法来搜寻到一个毗连的局部最佳解。

5.5 搜寻另一个局部最佳解的基于动态分解点(DDP-based)的方法

在最佳化问题(4-1)中，我们发明一个基于动态分解点(DDP-based)的方法，从一个已知局部最佳解来搜寻到另一个局部最佳解的方法，如下表示：

给予：一个最佳化问题(4-1)其已知的局部最佳解 x_s 及一条先定的搜寻路线。

目的：搜寻到最佳化问题(4-1)的另一个局部最佳解。

步骤一、从一个已知的局部最佳解 x_s 开始，沿着搜寻路径来计算非线性系统(5-7)的动态分解点。如动态分解点 x_{dap} 。

步骤二、设 $x_0 = x_s + (1 + \varepsilon)(x_{dap} - x_s)$ ，其中如表示式存在 ε 为一极小值。

步骤三、从 x_0 开始，对非线性系统(5-7)进行积分以获得一条轨迹且求得接口点(*interface point*) x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解，其效果是会优于用轨迹来求解局部最佳解。

步骤四、从此接口点 x_{inf} 开始，运用步骤三所选的有效混合式局部搜寻法来搜寻到一个毗连的局部最佳解。

5.6 搜寻所有阶层一(*tier-one*)的局部最佳解的动态方法

在说明此方法前，我们先定义说明在最佳化问题(4-1) (相对于非线性动态系统(5-7))中，对一个已知的局部最佳解，所谓的阶层一(*tier-one*)局部最佳解及阶层 N(*tier-N*)局部最佳解。

阶层一(*tier-one*)局部最佳解：如果非线性动态系统(5-7)的局部最佳解 x_{s0} 其稳定区域的范围(*closure*)与局部最佳解 x_{s1} 其稳定区域的范围(*closure*)相交，则我们称最佳化问题(4-1)局部最佳解 x_{s1} 是局部最佳解 x_{s0} 的阶层一(*tier-one*)局部最佳解。

阶层 N(*tier-N*)局部最佳解：一个最佳化问题(4-1)的局部最佳解 x_{sn} ，如果满足下述二条件，则我们称 x_{sn} 是此已知局部最佳解 x_{s0} 的阶层 N(*tier-N* $N > 1$)局部最佳解。

(i) x_{sn} 既不是局部最佳解 x_{s0} 阶层-(N-2) 的局部最佳解也不是阶层-(N-1)的局部最佳解。

(ii) 局部最佳解 x_{s0} 的阶层-(N-1) 局部最佳解集合，其中某一个局部最佳解，是局部最佳解 x_{sn} 的阶层一(*tier-one*) 局部最佳解。

局部最佳解 x_{s0} 的阶层-0 局部最佳解是其本身 x_{s0} 。

对最佳化问题(4-2)，(4-3)及(4-4)，所谓阶层一(*tier-one*)局部最佳解及阶层-N(*tier-N*)局部最佳解的定义均相似。

对最佳化问题(4-1)，我们发明二个方法(从一个已知的局部最佳解)来搜寻到所有阶层一(*tier-one*)的局部最佳解。第一个是基于出口点(*EP-based*)的方法，第二个是基于动态分解点(*DDP-based*)的方法。

基于出口点(EP-based)的方法

给予：一个无限制条件连续性最佳化问题(4-1)，其已知的局部最佳解 x_s^0 。

目的：搜寻最佳化问题(4-1)中，相对于局部最佳解 x_s^0 的所有阶层一(tier-one)局部最佳解。

步骤一：设定所有阶层一(tier-one)局部最佳解集合 $V_s = \{\phi\}$ ，以及定义一组 x_s^0 的搜寻路径 $S_i, i=1,2,\dots,m$ 。设 $i=1$ 。

步骤二：从一个已知的局部最佳解 x_s^0 开始，沿着搜寻路径 S_i 移动，寻找局部最佳解。

(i) 运用 EP-based 方法来寻找另一个局部最佳解。如局部最佳解被找到，以 x_s^i 表示继续下一步；否则执行(iii)。

(ii) 检查 x_s^i 是否曾被找到 ($x_s^i \in V_s$)，如果尚未曾被找到过则设 $V_s = V_s \cup \{x_s^i\}$ 。

(iii) 设 $i=i+1$ 。

(iv) 重复执行步骤(i)到步骤(iii)，直到 $i > m$ 。

步骤三：输出 x_s^0 所有阶层一(tier-one)的局部最佳解 V_s 。

基于动态分解点(DDP-based)的方法

给予：一个无限制条件连续性最佳化问题(4-1)，其已知的局部最佳解 x_s^0 。

目的：搜寻最佳化问题(4-1)中，相对于局部最佳解 x_s^0 的所有的阶层一(tier-one)局部最佳解。

步骤一：设定动态分解点集合 $V_{ddp} = \{\phi\}$ ，所有阶层一(tier-one)的局部最佳解集合 $V_s = \{\phi\}$ 。

步骤二：定义一组 x_s^0 的搜寻路径 $S_i, i=1,2,\dots,m_j$ 。设 $i=1$ 。

步骤三：从一个已知的局部最佳解 x_s^0 开始，沿着预定的搜寻路径 S_i 移动，寻找局部最佳解。

(i) 运用动态分解点的方法来寻找非线性系统(5-7)的动态分解点。如动

态分解点被找到，以 $x_{ddp,i}$ 表示继续下一步；否则执行(vi)。

(ii) 检查 $x_{ddp,i}$ 是否曾被找到 ($x_{ddp,i} \in V_{ddp}$)，如果曾被找到过则执行(vi)，否则设 $V_{ddp} = V_{ddp} \cup \{x_{ddp,i}\}$ 继续执行(iii)。

(iii) 设 $x_{0,i} = x_s^0 + (1 + \varepsilon)(x_{ddp,i} - x_s^0)$ ，其中如表示式存在 ε 为一极小值。

(iv) 从 $x_{0,i}$ 开始，对非线性系统(5-7)进行积分以获得一条轨迹且求得接口点 $x_{inf,i}$ 。在此(接口)点运用一个有效的混合式局部搜寻法来找局部最佳解，其效果是会优于用轨迹来求解局部最佳解。

(v) 从此界面点 $x_{inf,i}$ 开始，运用步骤(iv)所选的有效混合式局部搜寻法来搜寻对应 $x_{ddp,i}$ 的一个毗连的局部最佳解 x_s^i ，并设 $V_s = V_s \cup \{x_s^i\}$ 。

(vi) 设 $i = i + 1$ 。

(v) 重复执行步骤(i)到步骤(vi)，直到 $i > m_j$ 。

步骤四：输出 x_s^0 所有阶层一(tier-one)的局部最佳解 V_s 。

5.7 搜寻到所有的局部最佳解的方法

对最佳化问题(4-1)，我们发明二个方法（从一个已知的局部最佳解）来搜寻到所有的局部最佳解。基于出口点(EP-based)的方法，第二个是基于动态分解点(DDP-based)的方法。

基于出口点(EP-based)的方法

给予：一个连续性最佳化问题(4-1)的起始点 x_0 （可以是任何点）。

目的：(i) 搜寻到连续性最佳化问题(4-1)的所有的局部最佳解。

(ii) 搜寻连续性最佳化问题(4-1)的全局最佳解。

步骤一：以起始点 x_0 开始应用一个有效的(混合式)搜寻方法，来寻找一个局部最佳解 x_s^0 。

步骤二：设 $j = 0$ ，初始化局部最佳解集合 $V_s = \{x_s^0\}$ ，阶层 j (tier- j) 的局部最佳解集合 $V_{new}^j = \{x_s^0\}$ 。

步骤三：设阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\phi\}$ 。

步骤四：对在 V_{new}^j 中的每一个局部最佳解 x_s^j ，寻找它所有阶层-1 的局部最佳解。

(i) 针对于 x_s^j ，定义一组搜寻路径 $S_i^j \quad i=1,2,\dots,m_j$ ，设 $i=1$ 。

(ii) 从一个已知的局部最佳解 x_s^j 开始，运用基于出口点 (*EP-based*) 的方法寻找另一个局部最佳解。如局部最佳解被找到，以 $x_{s,j}^i$ 表示并继续下一步；否则执行步骤 (iv)。

(iii) 检查 $x_{s,j}^i$ 是否以前被找过，即 $x_{s,j}^i \in V_s$ ，如果它尚未被找过，设 $V_s = V_s \cup \{x_{s,j}^i\}$ 及 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_{s,j}^i\}$ 。

(iv) 设 $i=i+1$ 。

(v) 重复执行 (ii) 到 (iv)，直到 $i > m_j$ 。

步骤五：检查最新计算的局部最佳解集合 V_{new}^{j+1} ，如果是空集合，执行步骤六；否则设 $j=j+1$ ，并执行步骤三。

步骤六：输出所有的局部最佳解集合 V_s ，藉由比较 V_s 集合中的所有局部最佳解所对应的个别目标函数值，选择其中一个具有最小值的最佳解。输出此点为全局最佳解。

基于动态分解点 (*DDP-based*) 的方法

给予：一个连续性最佳化问题 (4-1) 的起始点 x_0 (可以是任何点)。

目的：(i) 搜寻到连续性最佳化问题 (4-1) 的所有的局部最佳解。

(ii) 在连续性最佳化问题 (4-1) 的全局最佳解。

步骤一：以起始点 x_0 开始应用一个有效的 (混合式) 搜寻方法，来寻找一个局部最佳解 x_s^0 。

步骤二：设 $j=0$ ，初始化局部最佳解集合 $V_s = \{x_s^0\}$ ，阶层 j (*tier-j*) 的局部最佳解集合 $V_{new}^j = \{x_s^0\}$ 及动态分解点集合 $V_{dtp} = \{\phi\}$ 。

步骤三：设阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\phi\}$ 。

步骤四：对在 V_{new}^j 中的每一个局部最佳解 x_s^j ，寻找它所有的局部最佳解。

(i) 定义 x_s^j 的一组搜寻路径 S_i^j $i=1,2,\dots,m_j$ ，设 $i=1$ 。

(ii) 对从 x_s^j 开始的每条搜寻路径 S_i^j ，运用找寻动态分解点的方法寻找非线性系统(5-7)的动态分解点。如动态分解点被找到，以 $x_{ddp,i}^j$ 表示继续下一步；否则执行(viii)。

(iii) 检查 $x_{ddp,i}^j$ 是否曾被找到 ($x_{ddp,i}^j \in V_{ddp}$)，如果曾被找到过则执行(viii)，否则设 $V_{ddp} = V_{ddp} \cup \{x_{ddp,i}^j\}$ 。

(iv) 设 $x_{0,i}^j = x_s^j + (1+\varepsilon)(x_{ddp,i}^j - x_s^j)$ ，其中(如表示式存在) ε 为一极小值。

(v) 从 $x_{0,i}^j$ 开始，对非线性系统(5-7)进行积分以获得一条轨迹且求得接口点 $x_{inf,i}^j$ 。在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解，其效果是会优于用轨迹来求解局部最佳解。

(vi) 从此界面点 $x_{inf,i}^j$ 开始，运用步骤(v)所选的有效混合式局部搜寻法来搜寻一个毗连的局部最佳解 $x_{s,j}^i$ 。

(vii) 检查 $x_{s,j}^i$ 是否以前被找过，即 $x_{s,j}^i \in V_s$ ，如果它尚未被找过，设 $V_s = V_s \cup \{x_{s,j}^i\}$ 及 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_{s,j}^i\}$ 。

(viii) 设 $i = i + 1$ 。

(ix) 重复执行(ii)到(vii)，直到 $i > m_j$ 。

步骤五：检查最新计算的局部最佳解集合 V_{new}^{j+1} ，如果是空集合，执行步骤六；否则设 $j = j + 1$ ，并执行步骤三。

步骤六：输出所有的局部最佳解集合 V_s ，藉由比较 V_s 集合中的所有局部最佳解所对应的个别目标函数值，选择其中一个具有最小值的最佳解，输出此点为全局最佳解。

6. 解决无限制条件离散性最佳化问题的动态方法

我们系在此说明本发明如何解决无限制条件离散性最佳化问题 (4-3) 的动态方法。

搜寻问题 (4-3) 的全局最佳解是非常困难的工作, 这工作常常地只搜寻到一个解, 此解是在邻域(*neighborhood*)中最好的。本发明需要定义什么是邻域。邻域函数 $N(y)$ (*neighborhood function*) 是一个映射 ($N:S \rightarrow 2^S$), 对每一个解(*solution*) y , 所行成的集合 $N(y)$ 。此集合在某种程度上是靠近 y 的。 $N(y)$ 集合被称 y 的邻域。如果对所有的 $x \in N(x^*)$, $UC(x) \geq UC(x^*)$, 则我们称 x^* 为邻域 $N(x^*)$ 的局部最小解(局部最佳解)。

许多具有如(4-3)形式的离散性最佳化问题是 NP-hard 的。通常人们认为 NP-hard 问题是不能在多项式限定的计算时间内最佳解决的。对于如(4-3)形式的离散性最佳化问题有三种解决方法。第一种是枚举法, 它能保证找到最佳解, 但必须以巨大的计算量为代价的。第二种方法使用一种多项式时间的近似算法来找到一个接近最佳的解。最后一种方法则求助于某些启发式技术或称为局部搜寻或局部改进的反复改进技术, 它对解的质量或运行时间没有先验保证。

局部搜寻法是一个常被用来解决许多离散性最佳化问题的方法。由于所要解决问题的大小或不了解问题的结构可能会让列举法(*enumerative methods*)难以实际应用。另一方面, 多项式时间的近似算法(*polynomial-time approximation algorithms*), 除了执行效能有所限制外, 有可能会产生不好的解。而局部搜寻法系在合理的时间内可对特定之问题获得合理、或高质量的解。一个局部搜寻法(*local search algorithms*)的基本型态, 我们称之为反复改进 (*iterative improvement*), 其从某些起始点开始, 搜寻其领域中 *cost function* 值较小的解。如果解被搜寻到的话, 将取代目前解并持续搜寻; 否则目前解是局部最佳解并停止局部搜寻算法。

接下来我们提出并说明解决离散性最佳化问题 (4-3) 的局部搜寻算法, 其基础型态方法, 称为反复改进 (*iterative improvement*) 方法。

局部反复改进 (局部搜寻) 算法 (*Iterative local Improvement (Local search) Algorithm*)

步骤一：初始：选一个始起点 x^1 和设定循环计次数 $k=1$ 。

步骤二：改进：考虑局部修正集合 (*local improvement set*) $\{x \in N(x_k): C(x) < C(x_k)\}$ ， $N(x^k)$ 是环绕在 x^k 的邻域。如果 $N(x^k)$ 是空集合，停止并设 x^k 当作邻域 $N(x^k)$ 的局部最佳解；否则选择 $N(x^k)$ 邻域中的一员视为 x^{k+1} ，设 $k=k+1$ ，重复此步骤。

针对一个局部搜寻算法，有二个问题需要解决。

(i) 邻域 $N(x)$ 的定义。

(ii) 从集合 $\{x \in N(x_k): C(x) < C(x_k)\}$ 中，选择 x^{k+1} 的机制。

为获得高质量的局部最佳解，找寻有效的邻域函数是局部搜寻算法的一大挑战。目前一般方法，仍缺少如何设定一有效邻域函数的规则。对于同一问题，可能存有许许多不同的邻域函数，对不同定义的邻域，仍可能产生许多不同的结果(纵然运用相同的局部搜寻算法)。一个好的邻域函数设计常常需要依据问题本身的离散结构。显然，为了解决问题(i)，一个合适的离散邻域必须足够大以包含当前解的一些离散变量；同时又必须足够小以便能够利用可行的计算量来搜寻。为了简化符号表示，我们假定所有变量满足 0-1 限制条件。

- 单位邻域：给定一个解 x^k ， x^k 的单位邻域通过每次补足一个 x^k 的分量来形成，即：

$$N_1(x^k) = \left\{ \text{二值的 } X: \sum_j |x_j - x_j^k| = 1 \right\} \quad (6-1)$$

- K-变化邻域：K-变化邻域包括了单位邻域，它允许每次补足最多 k 个解分量。具体就是，

$$N_1(x^k) = \left\{ \text{二值的 } X: \sum_j |x_j - x_j^k| \leq k \right\} \quad (6-2)$$

- 分段交换邻域：分段交换邻域每次以一种互补的形式交换两个二值分量。

$$N_1(x^k) = \left\{ \text{二值的 } X: \sum_j |x_j - x_j^k| \leq k, \sum_j (x_j - x_j^k) = 0 \right\} \quad (6-3)$$

- K-交换邻域：K-交换邻域每次以一种互补的形式交换至多 k 个二值分

量。

$$N_1(x^k) = \left\{ \text{二值的 } X : \sum_j |x_j - x_j^k| \leq k, \sum_j (x_j - x_j^k) = 0 \right\} \quad (6-4)$$

为实现多样性，在搜寻过程中邻域函数可以被转换。

本发明中，我们采用下列三种邻域的定义：

直接(immediate)邻域：点 x' 被视为 x 的一个直接邻域，若且唯若 x' 和 x 的差距 (difference) 可用向量 $e_1 = [u_0, u_1, \dots, u_n]$ 表示，其中仅有一个 u_i 为 1，其余为 0。因此点 x 和它的直接邻域 x' 的差别有 1 的差距发生。

单向延伸(one-way extended)邻域 (OWEN)：点 x' 被视为 x 的单向延伸邻域，若且唯若 x' 和 x 的差距(也就是 $x - x'$ 或 $x' - x$)可用向量 $e_1 = [u_0, u_1, \dots, u_n]$ 表示，式中 u_i 可为 1 或 0。

完全(full)邻域：点 x' 被视为 x 的一个完全的邻域，若且唯若 x' 和 x 的差距可用向量 $e_1 = [u_0, u_1, \dots, u_n]$ 表示，式中 u_i 可为 0, 1 或 -1。

对所有的直接 (Immediate) 邻域集合，某点有最小的目标函数值，我们说此点是在离散空间 S 中直接邻域的局部最佳解。对所有的单向延伸 (one-way extended) 邻域集合，某点有最小的目标函数值，我们说此点是在离散空间 S 中单向延伸邻域的局部最佳解。对所有的完全 (full) 邻域，某点有最小的目标函数值，我们说此点是在离散空间 S 中完全邻域的局部最佳解。以下事实是上述定义的推论：

事实 1：如果点 x^* 是关于其单向延伸邻域的一个局部最佳解，则它是关于其直接邻域的一个局部最佳解。

事实 2：如果点 x^* 是关于其完全邻域的一个局部最佳解，则它是关于其直接邻域以的一个局部最佳解以及关于其单向邻域以的一个局部最佳解。

事实 3：对无限制条件离散性最佳化问题 (4-3)， $C(x)$ 的全局最佳结必定是关于其完全邻域的一个局部最佳解。

对无限制条件离散性最佳化问题 (4-3), 我们所谓邻域是指直接 (*immediate*) 邻域; 局部最佳解是针对直接 (*Immediate*) 邻域所得到的。

为解决局部搜寻算法的问题(ii), 存有二个求得 x^{k+1} 的一般性局部反复改进算法 (即在步骤二的选择规则)。

最佳邻域搜寻方法 (*Best neighborhood search method*)

最佳邻域搜寻方法, 总是选择一个邻域中目标函数有最大下降的 x_{k+1} 点。这个方法是运用确定性方式来搜寻到局部最佳解。

较佳邻域搜寻方法 (*Better neighborhood search method*)

除了前述选择邻域中最大下降者为邻域点 x_{k+1} 外, 我们的较佳邻域搜寻法是选取邻域中的第一个相遇点 x_{k+1} , 且此点目标函数值是下降的。显然的, 这个较佳邻域搜寻法是快速的; 然而依据邻域中所有点被评估的顺序, 这种局部搜寻法不能产生唯一解。显然地, 较佳邻域搜寻法本质上是随机的。

在过去的三十多年内, 大量的局部搜寻算法被发展出来并被运用于广泛的离散性优化问题。这些方法中的许多是针对于特定类型的离散性最优化问题的。同时, 它们中的大部分只能有效地找到局部最佳解, 而不是全局最佳解。在搜寻过程中, 这些方法通常会陷入一个局部最佳解而不能移动到另一个局部最佳解。为弥合这种缺陷, 并同时保持在邻域搜寻的框架内, 一些修正业已提出, 但是并未取得很大成功。对局部搜寻的一个直接的拓展就是使用不同的起始解执行多次局部搜寻法, 并以其中最好的解作为最终解。但这种多起点方法, 仍未见有较大的成功。

反复改进的局部搜寻法的缺点促成了大量更加复杂的用于寻找更佳解的局部搜寻算法的发展。这些局部搜寻法通过引入一些机制来允许搜索过程逃离局部最佳解。这当中的“逃离”机制使用特定的搜寻方案, 以允许目标值退化的邻域, 并使逃离局部最佳解成为可能。

然而, 许多研究已经发现这些复杂的局部搜寻算法, 与其它算法比较, 需要大量的计算量并且通常不能找到全局最佳解。

本发明中, 我们发展一个新的系统方法, 它是确定性方法搜寻到一般离散性最佳化问题所有的局部最佳解。针对解决无限制条件离散性最佳化问

题 (4.3), 我们发展二个动态方法:

- 一个为解决目标函数是封闭模式的无限制条件离散性最佳化问题。
- 另一个为解决目标函数是黑盒模式的无限制条件离散性最佳化问题。

依据离散性动态方式与连续性动态方式此二种方式; 本发明发展许多不同的动态方法。除此, 为搜寻全局最佳解, 本发明并发展几个有效的方法来解决下面两个重要且具挑战性的问题:

- (i) 该如何有效地脱离一个局部最佳解, 并移向另外一个局部最佳解; 和
- (ii) 该如何避免再次重复搜寻已知(已找过)的最佳解。

6.1 基于离散性的动态方式(Discrete-based Dynamical Approach)

在下文中, 我们发展一个离散性动态方式来解决无限制条件离散性最佳化问题 (4-3)。

为解决无限制条件离散性最佳化问题 (4-3), 我们考虑相关的非线性离散动态系统; 它以一组微分方程的型式来描述,

$$x_{k+1} = f(x_k) \quad (6-5)$$

其中 $x \in S$ 离散的状态空间(*state space*)。方程式(6-5)可视为一个映射(*map*)或交互循环(*iteration*)。我们视 x_{k+1} 系由 x_k 经由映射公式所对映到的。方程式(6-5)的解, 由 $x_0 \in S$ 称为一条系统轨迹(*system trajectory*)。在离散动态系统(6-5)中, 如一点 $x^* \in S$ 且 $x^* = f(x^*)$, 则此点是一个固定点(*fixed point*)。如任何系统轨迹能保持于固定点的邻域中(选择起始点 x_0 尽量接近固定点), 则称此固定点是稳定固定点(*stable fixed point*)。

我们接下对非线性离散动态系统(它的轨迹可找出所有的局部最佳解), 提出下述充分条件。

(LD1)- 非线性动态系统(6-5)的轨迹是收敛的, 而且收敛至其中一个固定点; 及

(LD2)-沿着任何离散动态系统(6-5)的轨迹, 离散性最佳化问题的目标函数值是递减的。

(LD3)- 离散性最佳化问题(4-3)的每一个局部最佳解是系统(6-5)的一个稳定固定点, 反之亦然。

离散性最佳化问题(4-3)的所有局部最佳解，可以经由离散动态系统(6-5)的某些轨迹来有系统的计算获得。从这些局部最佳解，我们可轻易求解全局最佳解。

6.1.1 混合式局部搜寻方法(Hybrid Local Search Method)

一个混合式基于离散性(*discrete-based*)的局部搜寻法(从一个起始点开始来找寻离散最佳化问题(4-3)的局部最佳解)，如下表示：

给予：一个起始点 x_0 。

目的：可靠的计算出一局部最佳解。

步骤一、从一个起始点 x_0 开始，针对离散性最佳化问题(4-3)，运用一个有效的局部反复修正算法(*iterative improvement*)。

步骤二、如果局部搜寻算法收敛到一个局部最佳解，停止执行；否则到下一个步骤执行。

步骤三、使用最佳邻域搜寻法(*Best neighborhood search method*)，从 x_0 开始，持续搜寻几个步骤后，记录最后一点为终端点。以终端点当作起始点，执行步骤一。

6.1.2 计算出口点的方法

我们接下来说明如何在离散性最佳化问题(4-3)中，从已知的局部最佳解 x_s 开始，沿着搜寻路径计算出口点。我们注意到搜寻路径 $\psi_t(x_s)$ 经过 x_s ，是一条由 t 变量参数化的曲线 $\psi_0(x_s) = x_s$ ，且搜寻路径 $\psi_t(x_s)$ 对所有的 t 而言，系位于搜寻空间(解的空间)中。

在离散性最佳化问题(4-3)中，从已知的局部最佳解开始，沿着搜寻路径计算出口点的方法称之基于离散性(*discrete-based*)的出口点搜寻法，如下表示：

给予：一个已知的局部最佳解 x_s ，预定的搜寻路径，以及相关的离散动态系统(6-5)。

目的：预定的搜寻路径及系统(6-5)的一个局部最佳解(稳定固定点) x_s ，计算出另一个局部最佳解。

从已知的局部最佳解 x_s 开始，沿着预定的搜寻路径移动，直到最佳化问题(4-3)目标函数达到第一局部最大值的点上，则此点为出口点。

6.1.3 基于离散性动态分解点的方法

在最佳化问题(4-3)中，从已知的局部最佳解 x_s 开始，我们发展计算动态分解点的方法称之为基于离散性动态分解点搜寻法，其 x_s 点是相对位于系统(6-5)内预定的搜寻路径的稳定固定点上，如下表示：

给予：一个离散性最佳化问题(4-3)，其已知的局部最佳解 x_s ，及预定的搜寻路径。

目的：由预定的搜寻路径及离散动态系统(6-5)，计算出相对应的动态分解点。

步骤一、从已知的局部最佳解 x_s 开始，沿着预定的搜寻路径移动，侦测到出口点，将出口点设为 x_{ex} 。

步骤二、从 x_{ex} 开始，应用最佳邻域搜寻方法，向前移动 x_{ex} 一时间步骤到新点 x_c 。

步骤三、执行下列的射线调整过程并计算 DDP 点：

(i) 构成一条射线，连接 x_s 、 x_c 。

(ii) 如向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 有表达式，则执行(ii) (a)；否则执行取代(ii) (a)的另一步骤。

(a) 计算向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 的夹角 ε 。如果 ε 接近 0° 或 180° ，则 x_{ex} 是动态分解点并停止过程；否则执行下一步骤。

(取代步骤(a)的步骤) 修正搜寻路径使得现有的 x_{ex} 落于修正后的路径上，检查 x_{ex} 、 x_c 与 x_s 是否在同一修正后的路径上。如果是的话， x_{ex} 是动态分解点并停止过程；否则执行下一步骤。

(iii) 调整 x_c 到射线上（从 x_s 开始）最大目标函数值的点，设为 x'_c 。如果 x'_c 跟 x_{ex} 非常接近，则 x'_c 是动态分解点并停止执行；否则执行下一步骤。

步骤四、把 x_c' 当作出口点 ($x_c' \rightarrow x_{ex}$)，执行步骤二。

6.1.4 搜寻另一个局部最佳解的基于出口点的方法

对最佳化问题(4-3)，我们发明一个从一个已知的局部最佳解 x_s 开始来搜寻到另一个局部最佳解的方法，称之为基于离散性、且以出口点为基础的方法，如下表示：

给予：一个最佳化问题(4-3)其已知的局部最佳解 x_s ，及预定的搜寻路径。

目的：搜寻到最佳化问题(4-3)的另一个局部最佳解。

步骤一、从一个已知的局部最佳解 x_s 开始，沿着预定的搜寻路径移动，运用基于离散性的方法计算非线性动态系统(6-5)的出口点。如出口点被找到，以 x_{ex} 表示继续下一步；否则停止执行。

步骤二、从找到的 x_{ex} 开始，沿着预定的搜寻路径(由 x_s 开始)向前移动一步至 x_0 (或以 $x_0 = x_s + (1 + \varepsilon)(x_{ex} - x_s)$ 表示，如表示式存在且 ε 为一极小值)。

步骤三、从 x_0 开始，运用一个最佳邻域搜寻法求得接口点 x_{mf} 。在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解，其效果是会优于用最佳邻域搜寻法来求解局部最佳解。

步骤四、从此接口点 x_{mf} 开始，运用步骤三所选的有效混合式局部最佳搜寻法来搜寻到一个毗连的局部最佳解。

6.1.5 搜寻另一个局部最佳解的基于动态分解点的方法

在最佳化问题(4-3)中，我们发明从一个已知的局部最佳解来搜寻到另一个局部最佳解的方法，称之为基于离散性、且以基于动态分解点的方法，如下表示：

给予：一个最佳化问题(4-3)其已知的局部最佳解 x_s ，及预定的搜寻路径。

目的：搜寻到最佳化问题(4-3)搜寻路径中的另一个局部最佳解。

步骤一、从一个已知的局部最佳解 x_s 开始，沿着预定的搜寻路径移动，运用基于离散性的方法计算动态分解点。如动态分解点被找到，以 x_{dpp} 表示继

续下一步；否则停止执行。

步骤二、从找到的 DDP 点开始，沿着预定的搜寻路径(由 x_s 开始)向前移动一步至 x_0 (或以 $x_0 = x_s + (1 + \varepsilon)(x_{ddp} - x_s)$ 表示，如表示式存在，且 ε 为一极小值)。

步骤三、从 x_0 开始，运用一个最佳邻域搜寻法来求得接口点 x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解，其效果是会优于用最佳邻域搜寻法来求解局部最佳解。

步骤四、从此接口点 x_{inf} 开始，运用步骤三所选的有效混合式局部最佳搜寻法来搜寻到一个毗连的局部最佳解。

6.1.6 搜寻所有阶层一(tier-one)的局部最佳解的基于离散性的方法

在最佳化问题(4-3)中，我们发明从一个已知的局部最佳解来搜寻到所有阶层一(tier-one)的局部最佳解的方法。

给予：最佳化问题(4-3)，一个已知的局部最佳解 x_s^0 。

目的：搜寻到局部最佳解 x_s^0 所有阶层一(tier-one)的局部最佳解。

步骤一：设定所有阶层一(tier-one)的局部最佳解集合 $V_s = \{x_s^0\}$ ，以及定义一组 x_s^0 的搜寻路径 $S_i, i=1,2,\dots,m$ 。设 $i=1$ 。

步骤二：从一个已知的局部最佳解 x_s^0 开始，沿着预定的搜寻路径移动，运用基于离散性且以出口点为基础的方法(或基于离散性且以动态分解点为基础的方法)寻找最佳化问题(4-3)的一个局部最佳解(此解是相对位于非线性动态系统(6-5)的一个稳定固定点)。如局部最佳解被找到，以 x_s^i 表示继续下一步；否则执行步骤四。

步骤三：检查 x_s^i 是否曾被找到 ($x_s^i \in V_s$)，如果尚未曾被找到过则设 $V_s = V_s \cup \{x_s^i\}$ 。

步骤四：设 $i = i + 1$ 。

步骤五：重复执行步骤二到步骤四，直到 $i > m$ 。

步骤六：输出 x_s^0 所有阶层一(tier-one)的局部最佳解 V_s 。

6.1.7 搜寻所有局部最佳解的基于动态离散性的方法

在离散性最佳化问题(4-3)中, 我们发明一个从一个已知的局部最佳解来搜寻到所有局部最佳解的方法。

给予: 一个最佳化问题(4-3)的起始点 x_0 (可以是任何点)。

目的: (i) 搜寻到离散性最佳化问题(4-3)中的所有的局部最佳解。

(ii) 搜寻离散性最佳化问题(4-3)中的全局最佳解。

步骤一: 以起始点 x_0 开始应用一个有效的(混合式)搜寻方法, 来寻找一个局部最佳解 x_s^0 。

步骤二: 设 $j=0$, 局部最佳解集合 $V_s = \{x_s^0\}$, 阶层 J (tier- j) 的局部最佳解集合 $V_{new}^j = \{x_s^0\}$ 。

步骤三: 设阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\phi\}$ 。

步骤四: 对在 V_{new}^j 中的每一个局部最佳解 x_s^j , 寻找它所有阶层 1 的局部最佳解。

(i) 定义 x_s^j 的一组搜寻路径 S_i^j $i=1,2,\dots,m_j$, 设 $i=1$ 。

(ii) 从一个已知的局部最佳解 x_s^j 开始, 沿着预定的搜寻路径移动, 运用基于离散性且以出口点为基础的方法(或基于离散性且以动态分解点为基础的方法)寻找最佳化问题(4-3)的一个局部最佳解(此解是相对位于非线性动态系统(6-5)的一个稳定固定点)。如局部最佳解被找到, 以 $x_{s,j}^i$ 表示继续下一步; 否则执行步骤(iv)。

(iii) 检查 $x_{s,j}^i$ 是否以前被找过, 即 $x_{s,j}^i \in V_s$, 如果它尚未被找过, 设 $V_s = V_s \cup \{x_{s,j}^i\}$ 及 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_{s,j}^i\}$ 。

(iv) 设 $i=i+1$ 。

(v) 重复执行(ii)到(iv), 直到 $i > m_j$ 。

步骤五: 检查最新计算的局部最佳解集合 V_{new}^{j+1} , 如果是空集合, 执行步骤六; 否则设 $j=j+1$, 并执行步骤三。

步骤六: 输出所有的局部最佳解集合 V_s , 藉由比较 V_s 集合中所对应的目标

函数值，选择其中一个具有最小值的最佳解，输出此点为全局最佳解。

6.2 基于连续性的动态方式

在下文中，我们发展一个基于连续性的动态方法来解决无限制条件离散性最佳化问题（4-3），其目标函数具有可分析式的封闭模式。这方法基本是先于一个等效的连续性最佳化问题空间中求解，然后如有必要再于离散性状态空间中求解。另对有黑盒模式的最佳化问题（4-3），使用以这种连续特性为基础的方法是有条件性的，在下文中我们将予以说明。

等效方法(Equivalent Approach)

我们首先将无限制条件离散性最佳化问题（4-3）的搜寻空间转换成欧氏空间（*Euclidean space*）中等效的连续性最佳化问题模式：

$$\text{最小化 } UC_{eq}(x) \quad x \in R^n \quad (6-6)$$

此处 $UC_{eq}(x)$ 是定义在欧氏空间（*Euclidean space*） R^n 中的一个连续函数， n 是状态空间 S 的自由度(*degree of freedom*)，我们认为转换工作必须满足下列两指导方针：

(G1) 最佳化问题（4-3）和等效最佳化问题（6-6）两者有共同的全局最佳解(即 $x \in S$ 是 $UC(x)$ 的全局最佳解，若且唯若 $x \in R^n$ 是 $UC_{eq}(x)$ 的全局最佳解)。

(G2) 最佳化问题（4-3）和等效的最佳化问题（6-6）两者尽可能有许多 (as many as possible) 共同的局部最佳解。

在转换后，我们能藉由解等效性(equivalent)非线性最佳化问题(6-6)的全局最佳解，来搜寻最佳化问题（4-3）的全局最佳解。因此，那些在一般非线性最佳化问题（6-6），求解局部最佳解及全局最佳解的技术，能被应用来解决离散性最佳化问题（4-3）。

在满足前述二指导方针的充分条件下，我们首先建构一个等效函数 (*equivalent functions*)，详述如下：

附加调节器的转换 (*Transformation by appending a modular*)

等效的目标函数 $UC_{eq}(x)$ 可建构成：

$$UC_{eq}(x) = UC(x) + \mu \prod_{i=1}^n \sin^2(\pi x_i) \quad x \in R^n \quad (6-7)$$

此处 $\mu > \frac{L}{2}$ 以及 L 是原目标函数 $UC(x)$ 的 Lipschitz 常数。

该变换技术有助于将离散性优化问题(4-3)变换成连续性优化问题(6-7)。如果 $UC(x)$ 是可微分的, 则该变换具有在任意状态空间上可微分的优点。其计算过程容易且直接, 同时对该变换的理论分析也是可能的。可以证明 $UC_{eq}(x)$ 是一个连续函数, 并且 $UC_{eq}(x)$ 具有与原始问题(4-3)一样的全局最佳解。然而, 该变换可能会引入新的不属于原始问题的局部最佳解。在通过计算变换后的问题(6-7)的局部最佳解来寻找全局最佳解的过程中, 该缺点导致大量的额外计算负担。并且, 该变换只能用于那类具有解析的闭合形式目标函数的离散性最优化问题(4-3)。

在本发明知此一实施例之中, 我们另外发展出一个新的转换方式, 它同样的满足指导方针(G1)和(G2)。

分段式线性转换 (Piecewise Linear Transformation)

其转换如下:

$$C_{eq}(x) = \left\{ \begin{array}{ll} C(x) & \text{when } \lfloor x \rfloor = \lceil x \rceil \\ C(\lfloor x \rfloor) + \frac{C(\lceil x \rceil) - C(\lfloor x \rfloor)}{\text{dist}(\lceil x \rceil - \lfloor x \rfloor)} \text{dist}(x - \lfloor x \rfloor) & \text{when } \lfloor x \rfloor \neq \lceil x \rceil \end{array} \right\} \quad (6-8)$$

此处 $\text{dist}(x, y)$ 是 x 和 y 间的距离。在大部份的情形下, 系以欧氏距离 (Euclidean distance) 表示, $x = (x_1, x_2, \dots, x_n) \in R^n$ $\lceil x \rceil = (\lceil x_1 \rceil, \lceil x_2 \rceil, \dots, \lceil x_n \rceil)$, $\lceil x_i \rceil$ 表示大于或等于 x_i 的最小整数, $\lfloor x \rfloor = (\lfloor x_1 \rfloor, \lfloor x_2 \rfloor, \dots, \lfloor x_n \rfloor)$ 表示小于或等于 x_i 的最大整数。该变换具有下列显著特性。

特性 1: $C_{eq}(x)$ 几乎处处连续。

引理 1: 给定一点 x_0 以及它的单向邻域点 x_1 。定义一个由单位开盒构成的连通区域, 该单位开盒由 (x_0, x_1) 和两点 x_0 及 x_1 描述。则 x_0 或 x_1 是变换后的目标函数 $C_{eq}(x)$ 在该连通区域上的最大点或最小点。

特性 2: $UC_{eq}(x)$ 的局部最佳解与原始问题(4-3)关于单向扩展邻域(OWEN)的局部最佳解相等。

特性 3: $UC_{eq}(x)$ 具有与原始优化问题(4-3)相同的全局最佳解。

上述得到的特性表明我们所发展的变换能够满足指导方针(G1)和(G2)。特性 2 说明变换后的问题具有的局部最佳解不比原始问题的多。实际上, 变换后的问题的局部最佳解是原始问题关于单向扩展邻域, 的局部最佳解, 而不是关于直接邻域的。变换后的问题通常具有比原始问题(4-3)少的局部最佳解。这是一个显著的特点, 它能够通过忽略浅度的局部最佳解来降低用以搜寻全局最佳解所需的运算量。该变换为我们运用我们的基于 DDP 的方法来有效解决离散性问题提供了途径。特性 3 说明变换后的问题和原始问题具有相同的全局最佳解。在本发明中, 转换方式系运用最佳化问题(4-3)中, 其目标函数为可分析式的封闭模式。此转换方式也可运用在目标函数是黑盒模式的最佳化问题(4-3)中, 只要于状态空间中任二点的距离是明确定义的。

延伸的方式(Extended Approach)

我们到目前为止是集中焦点在(经转换)基于连续性的动态方式。另外一个基于连续性的动态方式乃是保持目标函数不变, 离散状态空间(即搜寻空间)直接延伸成为连续状态空间。此问题被称为延伸的连续性最佳化问题。

$$\text{最小化 } UC(x) \quad (6-9)$$

此目标函数 $UC: R^n \rightarrow R$ 具有一个封闭模式的表示式, n 是状态空间 S 中的自由度, 此搜寻空间是连续性的。明显的原问题(4-3)的最佳解和延伸的问题(6-9) 是有不同的, 但彼此之间在实际问题中可能是不会相隔太远地。

经由第一阶段求解延伸性问题(6-9)后, 我们可在第二阶段利用第一阶段所求得的近似解当成起始条件来求解原最佳化问题(4-3)。从这二状态程序所求得的解将会是原最佳化问题(4-3)的局部最佳解。

我们注意到最佳化问题(4-3), 其离散性目标函数经由转换产生等效系统(6-6), 是具有与最佳化问题(4-3)相同的解空间。对于在目标函数不变状态下, 离散性最佳化问题(4-3)转换后所产生的延伸状态空间(也就是一连续的状态空间)(6-9), 如同系统(6-6)可表达为:

$$\text{最小化 } UC(x) \quad (6-10)$$

此处 $UC: R^n \rightarrow R$ 是一个封闭模式的目标函数。 n 是解的状态空间 S 中的自由度。此解的空间是连续的。

转换后的最佳化问题(6-10)是属于非限制条件连续性最佳化问题(4-1)中的一部份。针对最佳化问题(6-10)，建构满足章节 5 (LC1)-(LC3)条件的非线性动态系统。我们可以下列一般通式表达：

$$\dot{x} = F^U(x) \quad (6-11)$$

我们将会发展动态方法用来系统地计算离散性最佳化问题(4-3)所有的局部最佳解，其运用连续性的方式来求解等效的最佳化问题(6-6)或延伸的最佳化问题(6-9)，它们的一般通式表达如(6-10)。

6.2.1 混合式局部搜寻方法

一个混合式的局部搜寻法，从一个起始点开始(在一个连续的空间中)来找寻无限制条件离散性最佳化问题(4-3)的局部最佳解，如下表示：

给予：一个起始点 x_0 。

目的：可靠的搜寻无限制条件离散性最佳化问题(4-3)的一个局部最佳解。

步骤一、从一个起始点 x_0 开始，针对无限制条件离散性最佳化问题(4-3)，运用一个有效的局部搜寻法。

步骤二、如果局部搜寻算法收敛到一个局部最佳解，执行步骤四；否则设最后一点为终端点。以终端点当作起始点，执行下一步骤。

步骤三、用 x_0 当做起始条件，对非线性系统(6.11)在几个时间步距处进行积分，获得最后一点为终端点。以终端点当作起始点，执行步骤一。

步骤四、(针对延伸的方式)将计算所得的连续性局部最佳解 x_s ，产生 $m(=2)$ 个最接近的可行点集合 x_{s1}, \dots, x_{sm} 。至此有二方案可供选择；一为选择最小值的点为局部最佳解并停止执行。或另用此可行点集合当作初值点，继续至步骤五执行。

步骤五：(针对延伸的方式)针对每个初值点运用最佳最好邻域搜寻方法来分别求得 m 个 x_{s1}, \dots, x_{sm} 个局部最佳解，从其中选择最小值的点为原问题(4-3)

局部最佳解。

6.2.2 计算出口点的基于连续性的方法

一个基于连续性的方法，运用连续性动态系统(6-11)，从一个起始点开始(在一个连续的空间中)来找寻所对应的出口点，称之为出口点搜寻法，如下表示：

给予：离散性最佳化问题(4-3)的一个已知局部最佳解 x_s ，预定的搜寻路径，以及相关的非线性连续性动态系统(6-11)。

目的：由预定的搜寻路径及系统(5-7)的稳定平衡点 x_s ，计算出所对应动态系统(6-11)的出口点。

从已知的局部最佳解 x_s 开始，沿着预定的搜寻路径移动，直到连续性目标函数(6-10)达到第一局部最大值的点上，则此点为出口点。

本发明另外提出一个计算出口点的方法：从 x_s 开始，延着搜寻路径移动，计算搜寻路径的导数和非线性动态系统(6-11)向量场在每个时间点的内积值，当内积值正负符号变更时(由正到负)，则表示已寻找到出口点在 $[t_1, t_2]$ 间。然后运用在 t_1 或 t_2 时的搜寻路径的所得到的点来趋近出口点。

6.2.3 计算动态分解点的方法

在最佳化问题(4-3)中，给予已知的局部最佳解 x_s ，我们发展一个计算动态分解点的方法称之为基于连续性的动态分解点搜寻法，其 x_s 点是相对位于动态系统(6-11)的稳定平衡点，如下表示：

步骤一、从已知的局部最佳解 x_s 开始，沿着预定的搜寻路径 $\varphi_t(x_s)$ 移动，侦测到出口点，此点位于动态系统(6-11)稳定平衡点(也就是局部最佳解 x_s)的准稳定边界上(*quasi-stability boundary*)，设出口点为 x_{ex} 。

步骤二、从 x_{ex} 开始，应用下述步骤找寻最小距离点，简称 *MDP*：

(i) 以出口点 x_{ex} 为起始条件，对非线性系统(6-11)在几个时间步距(*time-steps*)处进行积分；设终端点为目前出口点(*current exit point*)。

(ii) 连接轨迹上的目前出口点和局部的最佳解(也就是稳定平衡点)成一直线, 沿着此直线从稳定平衡点找寻目标函数(6-10)的第一局部最大的点, 视其为修正出口点 x_{ex} , 并以此点代替目前的点, 指定这个点为出口点 x_{ex} 。

(iii) 重复执行(i)及(ii), 直到在(ii)中所获得的出口点, 其向量场平均值 (*norm of vector field*)(6-10)是小于一个临界值, 也就是数值 $\|F^u(x_{ex})\| \leq \varepsilon$ 。那么这点被视为最小距离点, 简称 MDP。

步骤三、以 MDP x_d^0 为起始点, 求解系统(6-11)其非线性代数方程式 $F^U(x)=0$ 。设其解为 x_{ddp} , 则 x_{ddp} 是针对于动态系统(6-11)的稳定平衡点 x_s 和搜寻路径 $\varphi_t(x_s)$ 的动态分解点。

6.2.4 搜寻另一个局部最佳解的基于出口点的方法

从最佳化问题(4-3)中, 已知的局部最佳解 x_s 开始, 我们发明一个从一个已知的局部最佳解来搜寻到另一个局部最佳解的方法, 称之基于出口点的方法。

给予: 一个最佳化问题(4-3), 其已知的局部最佳解 x_s 。

目的: 搜寻到最佳化问题(4-3)的另一个局部最佳解。

步骤一、从一个已知的局部最佳解 x_s 开始, 计算非线性动态系统(6-11)的出口点 x_{ex} 。

步骤二、设 $x_0 = x_s + (1 + \varepsilon)(x_{ex} - x_s)$ 其中 ε 为一极小值。

步骤三、从 x_0 开始, 对非线性系统(6-11)进行积分以获得一条轨迹且求得接口点 x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解, 其效果是会优于用轨迹来求解局部最佳解。

(取代步骤三的步骤)、将 x_0 点调向新的点 x_0' (远离 x_s , 但接近 $(x_{ep} - x_s)$ 的方向)。由 x_0' 开始运用最佳邻域搜寻法, 可获得一条系统轨迹且求得接口点 x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法, 其结果是会优于最佳邻域搜寻法。

步骤四、从此接口点 x_{inf} 开始，运用步骤三针对问题(4-3)所选的有效混合式局部搜寻法来搜寻到一个毗连的局部最佳解。

6.2.5 搜寻另一个局部最佳解的基于动态分解点的方法

在最佳化问题(4-3)中，我们发明一个从一个已知的局部最佳解来搜寻到另一个局部最佳解的方法，如下表示：

给予：最佳化问题(4-3)一个其已知的局部最佳解 x_s ，和一条给定的搜寻路径。

目的：搜寻到最佳化问题(4-3)另一个局部最佳解。

步骤一、从一个已知的局部最佳解 x_s 开始并沿着给定搜寻路径，运用系统(6-11)来计算所对应动态分解点 x_{dtp} 。

步骤二、设 $x_0 = x_s + (1 + \varepsilon)(x_{\text{dtp}} - x_s)$ ，其中(如表示式存在) ε 为一极小值。

步骤三、从 x_0 开始，对非线性系统(6-11)进行积分以获得一条轨迹且求得接口点 x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解，其效果是会优于用轨迹来求解局部最佳解。

(取代步骤三的步骤)、将 x_0 点调向新的点 x_0' (远离 x_s ，但接近 $(x_{\text{dtp}} - x_s)$ 的方向)。由 x_0' 开始运用最佳邻域搜寻法，可获得一条系统轨迹并求得接口点 x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法，其结果是会优于最佳邻域搜寻法。

步骤四、从此接口点 x_{inf} 开始，运用步骤三针对问题(4-3)所选的有效混合式局部搜寻法来搜寻到一个毗连的局部最佳解。

6.2.6 以动态的方法来搜寻到所有阶层一(tier-one)的局部最佳解。

在最佳化问题(4-3)中，我们发展二个基于连续性的方法，(从一个已知的局部最佳解)来搜寻到所有阶层一(tier-one)局部最佳解的方法。第一个是基于连续性且以出口点为基础的方法(或基于离散性且以动态分解点为基础的方法

基于连续性且以出口点为基础的方法

给予：离散性最佳化问题(4-3)一个其已知的局部最佳解 x_s^0 。

目的：搜寻到离散性最佳化问题(4-3)局部最佳解 x_s^0 所有阶层一(tier-one)的局部最佳解。

步骤一：设定所有阶层一(tier-one)的局部最佳解集合 $V_s = \{x_s^0\}$ ，以及定义一组 x_s^0 的搜寻路径 $S_i, i=1,2,\dots,m$ 。设 $i=1$ 。

步骤二：从一个已知的局部最佳解 x_s^0 开始，沿着给定的搜寻路径 S_i 移动，寻找局部最佳解。

(i) 运用系统(6-11)及基于出口点的方法来寻找另一个局部最佳解。如局部最佳解被找到，以 x_s^i 表示继续下一步；否则执行(iii)。

(ii) 检查 x_s^i 是否曾被找到($x_s^i \in V_s$)，如果尚未曾被找到过则设 $V_s = V_s \cup \{x_s^i\}$ 。

(iii) 设 $i=i+1$ 。

(iv) 重复执行步骤(i)到步骤(iii)，直到 $i > m$ 。

步骤三：输出 x_s^0 所有阶层一(tier-one)的局部最佳解 V_s 。

基于离散性且以动态分解点为基础的方法

给予：离散性最佳化问题(4-3)一个其已知的局部最佳解 x_s^0 。

目的：搜寻到离散性最佳化问题(4-3)局部最佳解 x_s^0 所有阶层一(tier-one)的局部最佳解。

步骤一：设定动态分解点集合 $V_{dnp} = \{\phi\}$ ，所有阶层一(tier-one)的局部最佳解集合 $V_s = \{x_s^0\}$ 。

步骤二：，定义一组 x_s^0 的搜寻路径 $S_i, i=1,2,\dots,m$ 。设 $i=1$ 。

步骤三：从一个已知的局部最佳解 x_s^0 开始，沿着预定的搜寻路径 S_i 移动，寻找局部最佳解。

(i) 运用系统(6-11)及找寻动态分解点的方法来寻找动态分解点。如动态分解点被找到，以 $x_{dnp,i}$ 表示继续下一步；否则执行(vi)。

(ii) 检查 $x_{ddp,i}$ 是否曾被找到 ($x_{ddp,i} \in V_{ddp}$), 如果曾被找到过则执行(vi), 否则设 $V_{ddp} = V_{ddp} \cup \{x_{ddp,i}\}$ 继续执行(iii)。

(iii) 设 $x_{0,i} = x_s + (1 + \varepsilon)(x_{ddp,i} - x_s)$, 其中(如表示式存在) ε 为一极小值。

(iv) 从 $x_{0,i}$ 开始, 对非线性系统(6-11)进行积分以获得一条轨迹且求得接口点 $x_{inf,i}$ 。在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解, 其效果是会优于用轨迹来求解局部最佳解。

(取代步骤(iv)的步骤) 将 $x_{0,i}$ 点调向新的点 $x_{0,i}^{\cdot}$ (远离 x_s^0 , 但接近 $(x_{ddp,i} - x_s^0)$ 的方向)。由 $x_{0,i}^{\cdot}$ 开始运用最佳邻域搜寻法, 可获得一条系统轨迹并求得接口点 x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法, 其结果是会优于最佳邻域搜寻法。

(v) 从此界面点 $x_{inf,i}$ 开始, 运用步骤(iv)对问题(4-3)所选的有效混合式局部搜寻法来搜寻对应 $x_{ddp,i}$ 的一个毗连的局部最佳解。

(vi) 设 $i = i + 1$ 。

(vii) 重复执行步骤(i)到步骤(vi), 直到 $i > m$ 。

步骤四: 输出 x_s^0 所有阶层一(tier-one)的局部最佳解 V_s 。

6.2.7 搜寻到所有的局部最佳解的方法

在离散性最佳化问题(4-3)中, 我们发明二个动态方法, (从一个已知的局部最佳解)来搜寻到所有的局部最佳解的方法。第一个是基于连续性, 根据出口点的方法, 第二个是基于连续性, 根据动态分解点的方法。

基于连续性且以出口点为基础的方法

给予: 一个最佳化问题(4-3)的起始点 x_0 (可以是任何点)。

目的: (i) 搜寻离散性最佳化问题(4-3)所有的局部最佳解。

(ii) 搜寻离散性最佳化问题(4-3)的全局最佳解。

步骤一: 以起始点 x_0 开始应用一个有效的(混合式)搜寻方法, 来寻找离散

性最佳化问题(4-3)中的一个局部最佳解 x_s^0 。

步骤二：设 $j=0$ ，初始化局部最佳解集合 $V_s = \{x_s^0\}$ ，阶层 j (tier- j) 的局部最佳解集合 $V_{new}^j = \{x_s^0\}$ 。

步骤三：设阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\phi\}$ 。

步骤四：对在 V_{new}^j 中的每一个局部最佳解 x_s^j ，寻找它所有的局部最佳解。

(i) 定义 x_s^j 的一组搜寻路径 S_i^j $i=1,2,\dots,m_j$ ，设 $i=1$ 。

(ii) 从一个已知的局部最佳解 x_s^j 开始，运用基于出口点的方法寻找另一个局部最佳解。如局部最佳解被找到，以 $x_{s,j}^i$ 表示继续下一步；否则执行步骤 (iv)。

(iii) 检查 $x_{s,j}^i$ 是否以前被找过，即 $x_{s,j}^i \in V_s$ ，如果它尚未被找过，设 $V_s = V_s \cup \{x_{s,j}^i\}$ 及 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_{s,j}^i\}$ 。

(iv) 设 $i=i+1$ 。

(v) 重复执行(ii)到(iv)，直到 $i > m_j$ 。

步骤五：检查最新计算的局部最佳解集合 V_{new}^{j+1} ，如果是空集合，执行步骤六；否则设 $j=j+1$ ，并执行步骤三。

步骤六：输出所有的局部最佳解集合 V_s 。藉由比较 V_s 集合中所对应的个别目标函数值，选择其中一个有最小目标函数值的最佳解，输出此点为全局最佳解。

以动态分解点为基础的方法

给予：一个最佳化问题(4-3)的起始点 x_0 (可以是任何点)。

目的：(i) 搜寻到离散性最佳化问题(4-3)所有的局部最佳解。

(ii) 搜寻离散性最佳化问题(4-3)中的全局最佳解。

步骤一：以起始点 x_0 开始应用一个有效的混合式搜寻方法，来寻找离散性最佳化问题(4-3)的一个局部最佳解 x_s^0 。

步骤二：设 $j=0$ ，初始化局部最佳解集合 $V_s = \{x_s^0\}$ ，阶层 J (tier- j) 的局部最佳解集合 $V_{new}^j = \{x_s^0\}$ 及动态分解点集合 $V_{dnp} = \{\emptyset\}$ 。

步骤三：设阶层 $J+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\emptyset\}$ 。

步骤四：对在 V_{new}^j 中的每一个局部最佳解 x_s^j ，寻找它所有的局部最佳解。

(i) 定义 x_s^j 的一组搜寻路径 S_i^j $i=1,2,\dots,m_j$ ，设 $i=1$ 。

(ii) 对从 x_s^j 开始的每条搜寻路径 S_i^j ，运用系统(6-11)及找寻动态分解点的有效方法寻找动态分解点。如动态分解点被找到，以 $x_{dnp,i}^j$ 表示继续下一步；否则执行(vii)。

(iii) 检查 $x_{dnp,i}^j$ 是否曾被找到 ($x_{dnp,i}^j \in V_{dnp}$)，如果曾被找到过则执行(vi)，否则设 $V_{dnp} = V_{dnp} \cup \{x_{dnp,i}^j\}$ 继续执行(iv)。

(iv) 设 $x_{0,i}^j = x_s^j + (1+\varepsilon)(x_{dnp,i}^j - x_s^j)$ ，其中(如表示式存在) ε 为一极小值。

(v) 从 $x_{0,i}^j$ 开始，对非线性系统(6-11)进行积分以获得一条轨迹且求得接口点 $x_{inf,i}^j$ 。在此接口点运用一个有效的混合式局部搜寻法来找局部最佳解，其效果是会优于用轨迹来求解局部最佳解。

(替代步骤(v)的步骤) 将 $x_{0,i}^j$ 点调向新的点 $x_{0,i}^{'j}$ (远离 x_s^j ，但接近 $(x_{dnp,i}^j - x_s^j)$ 的方向)。由 $x_{0,i}^{'j}$ 开始运用最佳邻域搜寻法可获得一条系统轨迹并求得接口点 x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法，其结果是会优于最佳邻域搜寻法。

(vi) 从此界面点 $x_{inf,i}^j$ 开始，运用步骤(v)对问题(4-3)所选的有效混合式局部搜寻法来搜寻一个毗连的局部最佳解 $x_{s,j}^i$ 。

(vii) 检查 $x_{s,j}^i$ 是否以前被找过，即 $x_{s,j}^i \in V_s$ ，如果它尚未被找过，设 $V_s = V_s \cup \{x_{s,j}^i\}$ 及 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_{s,j}^i\}$ 。

(viii) 设 $i=i+1$ 。

(ix) 重复执行(ii)到(vii)，直到 $i > m_j$ 。

步骤五：检查最新计算的局部最佳解集合 V_{new}^{j+1} ，如果是空集合，执行步骤六；否则设 $j = j+1$ ，并执行步骤三。

步骤六：输出所有的局部最佳解集合 V_s ，藉由比较 V_s 集合中所对应的个别目标函数值，选择其中一个具有最小目标函数值的最佳解，输出此点为全局最佳解。

7. 解决有限制条件离散性最佳化问题的动态方法

下文系说明本发明解决在有限制条件下离散性最佳化问题（4-4）的动态方法。

解决有限制条件离散性最佳化问题（4-4）的困难是决定于目标函数 $C(x)$ 的结构，解的空间 S 及可行解集合 (*feasible set*) FS 三者。

对于一个离散性最佳化问题（4-4），运用一个局部搜寻法 (*local search algorithms*) 试着将现行解 $x^k \in FS$ 移向围绕在 x^k 的邻域集合 $N(x^k)$ 中一个较佳解，步骤说明如下：

步骤一：初始：选一个始起点 x^1 和设定循环计次数 $k=1$ 。

步骤二：修正：考虑局部修正集合 $\{x \in FS \cap N(x_k) : C(x) < C(x_k)\}$ ， $N(x^k)$ 是围绕在 x^k 的邻域。如果 $N(x^k)$ 是空集合，即停止并把 x^k 当作邻域 $N(x^k)$ 的局部最佳解；否则选择 $N(x^k)$ 邻域的其中一员视为 x^{k+1} ，设定 $k = k+1$ ，重复此步骤。

针对一个局部搜寻算法，存有三个问题。

(i) $N(x)$ 邻域的定义。

(ii) 从集合 $\{x \in FS \cap N(x_k) : C(x) < C(x_k)\}$ 中，如何选择 x^{k+1} 的机制。

(iii) 如何找到步骤一的第一个起始可行解？

前二个问题是与无限制条件离散性最佳化问题(4-3)相似的。为处理限制条件的特性，最佳邻域搜寻方法被修正为有限制条件的最佳邻域搜寻方法，只是从可行邻域集合 (*a set of feasible neighbors* 此邻域是满足限制条件) 中选择一个目标函数有最大下降的解为 x^{k+1} 。同样地，较佳邻域搜寻方法被修正为有限制条件的较佳邻域搜寻方法，只是从可行邻域集合 (*a set of feasible*

neighbors 此邻域是满足限制条件)中选择第一个目标函数有下降的解为 x^{k+1} 。因此运用有限制条件的最佳邻域搜寻方法或有限制条件的较佳邻域搜寻方法，我们可得到 $x^{k+1} \in FS$ 。

针对第三个问题，当完全不知任何可行解时，我们可采用下列步骤来寻找一个可行解：

步骤一：初始：选一个始起点 x^1 并检查 $x^1 \in FS$ ？：如是属于则停止执行，并用吾人所找到的可行解开始局部搜寻；否则设定循环计次数 $k=1$ 。

步骤二：可行性的修正：考虑非可行性的修正集合 $\{x \in N(x_k) : d(x) < d(x_k)\}$ ，函数 $d(x)$ 是测量 x 解的非可行性(*infeasibility*)。如果 $N(x^k)$ 是空集合，停止执行并发出警告讯息说明无可行解。但如存在 $x^* \in N(x^k)$ 及 $d(x^*)=0$ 则停止执行，且 x^* 为一个可行解。否则选择 $N(x^k)$ 邻域的其中一员视为 x^{k+1} ，设定 $k=k+1$ ，重复此步骤。

限制条件下空间求解方式(Constrained solution-space approach)

为求解有限制条件离散性最佳化问题(4-4)，我们发展数个动态方法，其依据限制条件下空间求解方式，详如下述三阶段：

阶段一：在一个可行单位元集合中求解出一个局部最佳解。

阶段二：在这可行单位元集合中求解出所有的局部最佳解。

阶段三：寻找另一个可行单位元。如果无法找到则停止执行，否则执行阶段二。

我们之所以称此为限制条件下空间求解方式，系因搜寻局部最佳解的工作是仅在解空间的可行单位元中进行。

在此我们为了解决有限制条件离散性最佳化问题(4-5)，发展一个基于离散性的动态方法，其目标函数 $C:S \rightarrow R$ 可为封闭模式或黑盒模式，其限制函数向量(constraint function vector) $G:S \rightarrow R^m$ 是为封闭模式。这个基于离散性的动态方法，在阶段二我们发展出一个基于离散性的方法，来在可行单位元集合中求解出所有的局部最佳解。这动态方法，在阶段三我们发展出一个基于连续性的方式，来寻找出所有的可行单位元。

不失一般特性，我们考虑下述最佳化问题(含等式限制式)：

$$\text{最小化 } C(x) \quad (7-1)$$

受到下列限制

$$h_i(x) = 0, i \in I = \{1, 2, \dots, l\} \quad (7-2)$$

一般情形，下列的可行集合或称为可行单位元

$$M = \left\{ x \in R^n; H(x) = (h_1(x), \dots, h_m(x))^T = 0 \right\}$$

为平滑流形(smooth manifold)。大体上，这集合 M 是非常复杂，包含不连合但路径连接的可行单位元；换句话说，可行区域 M 能被分解成几个不连合但路径连接的可行单位元，如

$$M = \sum_k M_k$$

此处每个 M_k 都是路径连接的可行单位元。每个可行单位元可能包含了有限制条件最佳化问题(7-1)的几个局部最佳解。

在阶段二我们提出一基于离散性的动态方法来找寻位于每个可行单位元中的所有的局部最佳解。同样地这基于离散性的方法是可解决无限制条件离散性非线性最佳化问题 (4-3)。其所运用的离散性动态系统如下：

$$x_{k+1} = f(x_k) \quad (7-3)$$

其中系统(7-3)的解 $x_k \in S$ 空间是离散的。离散性动态系统满足第六章节中所述的(LD1)-(LD3)条件。在本发明中，我们系发展出一个基于离散性动态系统(7-3)的动态轨迹方法，用以有系统地搜寻最佳化问题(4-5) 中某一个可行单位元所有的局部最佳解。

7.1 找寻一个可行点的方法

在本发明中，我们引用一个动态轨迹为基础的方法，其被描述在发明人江晓东于2001年五月四日所申请的美国专利第09/849,213号“*DYNAMICAL METHOD FOR OBTAINING GLOBAL OPTIMAL SOLUTION OF GENERAL NONLINEAR PROGRAMMING PROBLEMS*”中，该专利申请在此合并作为本案的参考。此方法是依据某些轨迹(*trajectories*)有系统地搜寻问题(7-2)中的所有可行单位元，轨迹型式如下：

$$\dot{x} = F(x) \quad (7-4)$$

来建置如此的非线性动态系统以达成目的，此系统必须满足下列二条件：
 (g1) 每个系统(7-4)的轨迹必须收敛，而且收敛到其中一个平衡流形；和
 (g2) 一个集合是此系统(7-4)的一个稳定平衡流形，若且唯若这集合是限制条件方程式(7-2)的一个可行单位元。

本发明举例说明一个满足条件(g1)及(g2)的非线性动态系统，用以找寻所有的可行单位元，叙述如下：

$$\dot{x} = -DH(x)^T H(x) \quad (7-5)$$

此处向量 $H(x) = (h_1(x), h_2(x), \dots, h_m(x))^T$ 是定义此最佳化问题其可行区域的限制向量。

我们发明二个方法，来找寻一个可行点(feasible point)。第一个方法是以动态轨迹为基础(trajjectory-based)的方法，第二个方法则是混合式局部搜寻(hybrid local search)方法。

动态轨迹为基础(trajjectory-based)的方法

我们接着说明从一不可行点(infeasible point)找寻一个可行单位元的方法。

给予：一个不可行点 x_0 (可以是任何点)。

目的：搜寻到满足限制条件方程式(7-3)的一个可行点。

从给定起始点 x_0 开始，对动态系统(7-4)在几个时间步距处进行积分并找寻收敛的一点(它是位于稳定平衡流形)，此点是(7-2)的可行点。

混合式局部搜寻(hybrid local search)方法

我们接着提出一个混合式的方法，此方法配合已轨迹为基础(trajjectory-based)的方法(全局收敛特性)及一个局部搜寻法(快速特性)，来找寻满足限制条件方程式(7-3)的一个可行点。

给予：一个不可行点 x_0 (可以是任何点)。

目的：搜寻到满足限制条件方程式(7-3)的一个可行点。

步骤一：从给定起始点 x_0 开始，运用一个有效的局部反复修正(局部搜寻)算法。如果它收敛(即如果它找到一个可行点)，那么停止；否则执行步

骤二。

步骤二：从给定起始点 x_0 开始，对动态系统(7-4)在几个时间步距处进行积分而得到终点，如果它的轨迹收敛（即如果它找到一个可行点），那么停止；否则将此终点更新为新的起始点 x_0 ，执行步骤一。

7.2 一个局部搜寻方法

接着我们说明从一不可行点(*infeasible point*)利用 *Lagrange-based* 方法找寻离散性最佳化问题(4-5)的一个局部最佳解。

藉由 slack 变数 y ，我们可将不等限制式转换成等式限制式，如下说明：

$$g(x, y) = G(x) + y^2 = 0 \quad (7-6)$$

一个 *Lagrangian* 函数如下：

$$L(x, \lambda) = c(x) + \lambda^T g(x, y) \quad (7-7)$$

如 (x^*, λ^*) 是 Lagrange 函数(7-7)的马鞍点(*saddle point*)，则 x^* 是离散性最佳化问题(4-5)的一个局部最佳解。其中梯度运算差(*a difference gradient operator*) $\Delta_x L(x, \lambda)$ 定义如下：

$\Delta_x L(x, \lambda) = (\delta_0, \delta_1, \dots, \delta_m) \in \{-1, 0, 1\}^m$ where $\sum_{i=1}^m |\delta_i| = 1$ ，也就是 δ_i 中至多一个为非零，且满足 $L(x - \Delta_x L(x, \lambda), \lambda) \leq L(x, \lambda)$ 。如果任何异于 x 的 x' 点，至多一个变数为 1， $L(x, \lambda) \leq L(x', \lambda)$ 则 $\Delta_x L(x, \lambda) = 0$ 。

因此， $\Delta_x L(x, \lambda)$ 总是指向目标函数下降的方向， $\Delta_x L(x, \lambda)$ 移动 x 点至其邻域有较低 $L(x, \lambda)$ 值的一点 x' 。如果此 x' 点不存在则 $\Delta_x L(x, \lambda) = 0$ 。

依据此定义， $\Delta_x L(x, \lambda)$ 可表示成：

$$\Delta_x L(x, \lambda) = \begin{cases} x' - x, x' \in \{Neighbor_of_x \text{ and } L(x', \lambda) \leq L(x, \lambda)\} \\ 0 \end{cases} \quad (7-8)$$

我们注意到 $\Delta_x L(x, \lambda)$ 不是唯一性，而是指向一个有较佳邻域的方向。我们可用 *greedy* 搜寻方向(最佳邻域搜寻方向)或梯度下降方向(*gradient decent direction*)为 $\Delta_x L(x, \lambda)$ 的方向。

然后我们设立一个离散性 Lagrange-based 修正规则：

$$\begin{aligned} x^{k+1} &= x^k + \Delta_x L(x^k, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + U(k) \end{aligned} \quad (7-9)$$

其中 $U(k)$ 定义如下：

$$U_i(k+1) = \begin{cases} \lambda_i(K) & \text{if } g_i(i) \text{ is violated} \\ 0 & \text{otherwise} \end{cases} \quad (7-10)$$

此处 $\lambda_i(K)$ 是一个非负的调整因子 (*non-negative adjustment factor*)，用来驱动搜寻方向远离前一个不适者。举例说 $\lambda_i(K)$ 可设为 1。

slack 变数 y 可设定如下：

$$y_{i,k}^2 = \begin{cases} -G_i(x_k), & \text{if } G_i(x_k) \leq 0 \\ 0, & \text{otherwise} \end{cases} \quad (7-11)$$

接着我们用一个离散性 Lagrange-based 方法来搜寻有条件离散性最佳化问题(4-4)的一个局部最佳解。

给予：一个不可行点 x_0 (可以是任何点)。

目的：搜寻到满足限制条件离散性最佳化问题(4-4)的一个局部最佳解。

步骤一：从给定起始点 x_0 开始(可能不满足部份限制条件)。

步骤二：经由下述步骤找寻一个局部最佳解：

(i) 设 $k=1$ 。

(ii) 计算(7-10)式的 $U(k)$ 。如果 x_0 在可行单位元中，则用有限制条件较佳邻域搜寻方向 (*constrained better neighborhood direction*) 及式(7-8)来计算 $\Delta_x L(x, \lambda)$ ；否则用原解空间的较佳邻域搜寻方向来计算 $\Delta_x L(x, \lambda)$ 。

(iii) 如果 $U(k)$ 及 $\Delta_x L(x, \lambda)$ 为非常小的值，则现存点 x_0 为局部最佳解；否则修正 x 及 λ 并执行下一步。

(iv) 设 $k = k + 1$ 。

(v) 以预定的递回次数，重复执行步骤(ii)至步骤(iv)。

7.3 混合式局部搜寻(hybrid local search)方法

一个混合式局部搜寻方法，从给定的可行起始点开始，来找寻限制条件

离散性最佳化问题(4-5)的一个局部最佳解。

给予：一个可行点 x_0 (位于可行单位元中)。

目的：搜寻到可行单位元中的一个可行点。

步骤一：从给定起始点 x_0 开始，运用一个有效的局部反复修正 (iterative improvement local search algorithms) 算法，求解有限制条件离散性最佳化问题(4-5)。

步骤二：如果局部搜寻法收敛到一个局部最佳解，那么停止；否则将此终点更新为新的起始点 x_0 ，执行下一步骤。

步骤三：从给定起始点 x_0 开始，用有限制条件最佳邻域搜寻法或有限制条件较佳邻域搜寻法执行搜寻几个时间步距而得到终点。将此终点更新为新的起始点，执行步骤一。

7.4 计算可行出口点的方法

运用一个离散性动态系统(7-3)，从一个已知局部最佳解开始，沿着预定的搜寻路径，来找寻有限制条件离散性最佳化问题(4-3)的可行性出口点，称之为基于离散性的可行出口点(也就是出口点满足所有限制条件)搜寻法，如下表示：

给予：一个已知局部最佳解 x_s ，及一条预定的搜寻路径。

目的：计算出所对应的可行性出口点。

从已知的局部最佳解 x_s 开始，沿着预定的搜寻路径移动，直到目标函数(4-5)达到第一局部最大值的点，此点为出口点。进行过程中随时检查是否违反限制条件，如是则停止执行。

7.5 计算可行动态分解点的方法

在离散性最佳化问题(4-5)中，给予一个已知的局部最佳解 x_s 及预定的搜寻路径，我们发展一个计算可行性动态分解点的方法，如下表示：

给予：最佳化问题(4-5)中的一个已知局部最佳解 x_s ，一条预定的搜寻路径 $\varphi_t(x_s)$ ，以及所相关的离散性动态系统(7-3)。

目的：计算出所对应的可行性动态分解点。

步骤一、从已知的局部最佳解 x_s 开始，沿着预定的搜寻路径 $\varphi_t(x_s)$ 移动，侦测到离散性动态系统(7-3)的可行出口点(也就是出口点满足所有限制条件)。设出口点为 x_{ex} 。

步骤二、从 x_{ex} 开始，应用有限制条件的最佳邻域搜寻方法，向前移动 x_{ex} 一时间步骤到新点 x_c 。

步骤三、执行下列的射线调整过程并计算动态分解点。

(i) 构成一条射线，连接 x_s 、 x_c 。

(ii) 如向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 可表达成表示式，则执行(ii)、(a)；否则执行取代步骤(ii)、(a)的步骤。

(a) 计算向量 $(x_s - x_{ex})$ 及 $(x_c - x_{ex})$ 的夹角 ε 。如果 ε 接近 0° 或 180° ，则 x_{ex} 是动态分解点并停止过程；否则执行(iii)。

(取代步骤(a)的步骤) 修正搜寻路径使得现有的 x_{ex} 落于修正后的路径上，检查 x_{ex} 、 x_c 与 x_s 是否在同一修正后的路径上。如果是的话， x_{ex} 是动态分解点(DDP)并停止过程；否则执行下一步骤。

(iii) 调整 x_c 到射线上(从 x_s 开始)的目标函数最大值的点，设为 x'_c 。如果 x'_c 跟 x_{ex} 相同，则 x'_c 是动态分解点并停止执行；否则执行下一步骤。

步骤四、把 x'_c 当作出口点($x'_c \rightarrow x_{ex}$)，执行步骤二。

7.6 搜寻另一个局部最佳解的基于出口点的方法

对离散性最佳化问题(4-5)，我们发明从一个已知局部最佳解 x_s 开始来搜寻到另一个局部最佳解的方法，称之基于出口点的方法。

给予：一个有限制条件离散性最佳化问题(4-5)其已知的局部最佳解 x_s 及预定的搜寻路径。

目的：搜寻到所对应的新的一个阶层(new tier-one)的局部最佳解。

步骤一、从已知的局部最佳解 x_s 开始，运用一个基于离散性的方法(与非

线性动态系统(7-3)相关), 计算可行性出口点 x_{ex} 。如可行性出口点无法找到则停止执行; 否则执行下一步骤。

步骤二、从找到的 x_{ex} 开始, 沿着预定的搜寻路径(由 x_s 开始)向前移动一步至 x_0 (或以 $x_0 = x_s + (1 + \varepsilon)(x_{ex} - x_s)$ 表示, 其中如表示式存在, ε 为一很小值。

步骤三、从 x_0 开始, 运用一个最佳邻域搜寻法可获得一条系统轨迹并求得接口点 x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法来求解最佳化问题(4-5), 其结果会优于最佳邻域搜寻法。

步骤四、从此接口点 x_{inf} 开始, 运用步骤三所选有效的混合式局部最佳搜寻法来搜寻到一个毗连的局部最佳解。

7.7 搜寻另一个局部最佳解的基于动态分解点的方法

对离散性最佳化问题(4-5), 我们发明从一个已知局部最佳解 x_s 开始来搜寻到另一个局部最佳解的方法, 称之基于动态分解点的方法。

给予: 一个有限制条件离散性最佳化问题(4-5)其已知的局部最佳解 x_s 及预定的搜寻路径。

目的: 搜寻到所对应的新的一个阶层(*tier-one*)的局部最佳解。

步骤一、从已知的局部最佳解 x_s 开始沿着预定的搜寻路径, 运用一个基于离散性的方法(与非线性动态系统(7-3)相关), 来计算出可行动态分解点 x_{dnp} 。如可行动态分解点无法找到则停止执行; 否则执行下一步骤。

步骤二、从找到的 x_{dnp} 开始, 沿着修正后的搜寻路径(由 x_s 开始)向前移动一步至 x_0 (或以 $x_0 = x_s + (1 + \varepsilon)(x_{dnp} - x_s)$ 表示, 其中如表示式存在, ε 为一极小值。

步骤三、从 x_0 开始, 运用一个最佳邻域搜寻法可获得一条系统轨迹并求得接口点 x_{inf} 。在此接口点运用一个有效的混合式局部搜寻法来求解最佳化问题(4-5), 其结果会优于最佳邻域搜寻法。

步骤四、从此接口点 x_{inf} 开始, 运用步骤三所选的有效性混合式局部最佳搜寻法来搜寻到一个毗连的局部最佳解。

7.8 搜寻可行单位元中所有阶层一(tier-one)的局部最佳解方法。

我们发明一个基于动态离散性的方法，从一个已知的局部最佳解开始，来寻找可行单位元中所有阶层一(tier-one)的局部最佳解。

给予：一个离散性最佳化问题(4-3)，其已知的局部最佳解 x_s^0 。

目的：搜寻落于可行单位元中所有阶层一(tier-one)的局部最佳解。

步骤一：设定所有阶层一(tier-one)的局部最佳解集合 $V_s = \{x_s^0\}$ ，并定义一组 x_s^0 的搜寻路径 $S_i, i=1,2,\dots,m$ 。设 $i=1$ 。

步骤二：从一个已知的局部最佳解 x_s^0 开始，沿着预定的搜寻路径 S_i 移动，运用动态系统(7-3)及离散性基于出口点的方法(或基于动态分解点的方法)寻找另一个局部最佳解。如局部最佳解被找到，以 x_s^i 表示并继续下一步；否则执行步骤四。

步骤三：检查 x_s^i 是否曾被找到 ($x_s^i \in V_s$)，如果尚未曾被找到过则设 $V_s = V_s \cup \{x_s^i\}$ 。

步骤四：设 $i=i+1$ 。

步骤五：重复执行步骤二到步骤四，直到 $i>m$ 。

7.9 搜寻可行单位元中所有局部最佳解方法。

我们发明一个基于动态离散性的方法，从一个已知的局部最佳解开始，来搜寻离散性最佳化问题(4-5)，落于可行单位元中所有的局部最佳解。

给予：一个有限制条件离散性最佳化问题(4-3)，其已知的局部最佳解 x_s^0 。

目的：搜寻包含 x_s^0 可行单位元内所有的局部最佳解。

步骤一：以起始点 x_0 开始应用一个有效的混合式搜寻方法，来寻找一个局部最佳解 x_s^0 。

步骤二：设 $j=0$ ，初始化局部最佳解集合 $V_s = \{x_s^0\}$ ，阶层 j (tier-j) 的局部最佳解集合 $V_{new}^j = \{x_s^0\}$ 。

步骤三：设阶层 $j+1$ 的局部最佳解集合 $V_{new}^{j+1} = \{\phi\}$ 。

步骤四：对在 V_{new}^j 中的每一个局部最佳解 x_s^j ，寻找它所有阶层一(tier-one)的局部最佳解。

(i) 定义 x_s^j 的一组搜寻路径 S_i^j $i=1,2,\dots,m_j$ ，设 $i=1$ 。

(ii) 从已知的局部最佳解 x_s^j 开始，运用相关的离散性非线性动态系统(7-3)及基于出口点(或基于动态分解点)的方法来寻找另一个局部最佳解。如局部最佳解被找到，则以 $x_{s,j}^i$ 表示并继续下一步；否则执行步骤(iv)。

(iii) 检查 $x_{s,j}^i$ 是否以前被找过，即 $x_{s,j}^i \in V_s$ ，如果它尚未被找过，设 $V_s = V_s \cup \{x_{s,j}^i\}$ 及 $V_{new}^{j+1} = V_{new}^{j+1} \cup \{x_{s,j}^i\}$ 。

(iv) 设 $i=i+1$ 。

(v) 重复执行(ii)到(iv)，直到 $i > m_j$ 。

步骤五：检查最新计算的局部最佳解集合 V_{new}^{j+1} ，如果是空集合，执行步骤六；否则设 $j=j+1$ ，并执行步骤三。

步骤六：输出所有的局部最佳解集合 V_s (位于某可行单位元内)，藉由比较 V_s 集合中所对应的个别目标函数值，选择其中一个具有最小目标函数值的最佳解，输出此点为全局最佳解。

7.10 搜寻毗连可行单位元的方法

我们发明一个动态方法，从一个已知的可行单位元开始，来寻找毗连的可行单位元。

给予：一个初值的可行点 x_0 。

目的：搜寻有限制条件离散性最佳化问题(4-5)落于毗连可行单位元中的另一个可行点。

步骤一：以起始点 $x_0 + \varepsilon \times S_i$ 开始，其中 ε 为一极小值，将系统(7-4)延着逆时轨迹(reverse-time trajectory)积分，在需要时运用一个局部搜寻方法，找寻系统(7-4)不稳定平衡流形中的一点。如找到此点设为 x_d^i ，并执行下一步骤；否则停止执行。

步骤二：设定 $x_p^i = x_d^i + (1+\varepsilon)(x_d^i - x_0)$ ， ε 是一个小的数目。

步骤三：以起始点 x_p^i 开始，将系统 (7-4) 延着顺时轨迹 (forward-time trajectory) 积分，并运用一个有效的混合式局部搜寻法，来定位一个分离的 (isolated) 可行单位元并找寻出其中一个可行点。

7.11 搜寻到所有局部最佳解方法。

我们发明一个基于动态离散性的方法，从一个已知的局部最佳解开始，来寻找离散性最佳化问题 (4-5) 中所有的局部最佳解。如果所有的局部最佳解被找到，则全局最佳解也因此可获得；否则最佳解也可从已求得的局部最佳解中获得。

给予：一个初始点 x_0 。

目的：(i) 搜寻有限制条件离散性最佳化问题 (4-5) 所有的局部最佳解。

(ii) 搜寻有限制条件离散性最佳化问题 (4-5) 的全局最佳解。

步骤一：以起始点 x_0 开始应用一个有效的局部搜寻方法，来寻找一个局部最佳解 x_s^0 。

步骤二：设 $j=0$ ，初始化局部最佳解集合 $V_s = \{x_s^0\}$ ，阶层 0 (tier-0) 的可行单位元集合 $V_0^j = \{x_s^0\}$ 。

步骤三：设阶层 $j+1$ 的可行单位元集合 $V_0^{j+1} = \{\phi\}$ 。

步骤四：对在 V_0^j 中的每一个可行点 x_0^j ，寻找包含 x_0^j 可行单位元内所有局部最佳解集合，并寻找所有毗连的可行单位元。

(i) 从一个已知的局部最佳解 x_0^j 开始，来寻找最佳化问题 (4-5) 中，落于可行单位元内所有的局部最佳解。

(ii) 定义 x_0^j 的一组搜寻路径 S_i^j $i=1,2,\dots,k$ 。

(iii) 对每一搜寻路径 S_i^j ，寻找一毗连的可行单位元步骤如下：

(a) 设 $i=1$ 。

(b) 运用一个方法来找寻毗连可行单位元内的一点 $x_{0,i}^j$ 。

(c) 从 $x_{0,i}^j$ 开始, 运用一个有效混合式的局部搜寻法, 求解局部最佳化解 $x_{s,i}^j$ 。检查 $x_{s,i}^j$ 是否以前被找过, 即 $x_{s,i}^j \in V_s$, 如果它尚未被找过, 设 $V_s = V_s \cup \{x_{s,i}^j\}$ 及 $V_0^{j+1} = V_0^{j+1} \cup \{x_{0,i}^j\}$ 。

(d) 设 $i = i + 1$ 。

(e) 重复执行(b)到(d), 直到 $i > k$ 。

步骤五: 检查最新计算的可行点集合 V_0^{j+1} , 如果是空集合, 执行步骤六; 否则设 $j = j + 1$, 并执行步骤三。

步骤六: 输出所有的局部最佳解集合 V_s , 藉由比较 V_s 集合中所对应的个别目标函数值, 选择其中一个具有最小目标函数值的最佳解, 输出此点为全局最佳解。

8. 超大规模集成电路 (VLSI) 分割问题

一个大规模集成电路系统依复杂度可被分割为数个层次: (1) 系统层次 (*system-level*) 分割, 系统被分割成一群子系统, 每个子系统能在一块印刷电路板 (PCB) 板内各自分别设计及制造; (2) 基板层次 (*board-level*) 分割, 一块印刷电路板 (PCB) 板分割成许多大规模集成电路 (VLSI) 芯片组; (3) 芯片层次 (*chip-level*) 分割。在每个芯片分割成更小的子线路 (*sub-circuits*)。在每一个层次中, 分割的限制条件和目标函数的是不同的。本发明将会专注于芯片层次的分割。所得结果能被运用到其它层次分割, 且具有相同结果。

我们将分割的问题用图学 (*graph*) 理论表达。一个超图 (*hypergraph*) $G = (V, E)$ 代表一个分割问题。设 $V = \{v_1, v_2, \dots, v_n\}$ 是一组顶点 (*vertices*) 集合, $E = \{e_1, e_2, \dots, e_m\}$ 是一组超边线 (*hyperedges*) 集合。每个顶点代表一个组件。当组件间 (对应于顶点) 是相互连接时, 有一条超边线 (*hyperedge*) 连接着顶点间。因此每条超边线 (*hyperedge*) 是顶点 (*vertices*) 集合的子集, 也就是 $e_i \subseteq V, i = 1, 2, \dots, m$ 。换句话说, 每条网络是以超边线 (*hyperedge*) 表示。每个组件的面积表示为 $a(v_i), 1 \leq i \leq n$ 。分割问题是将集合 V 分割成 V_1, V_2, \dots, V_k , 则

$$V_i \cap V_j = \phi, i \neq j$$

$$\cup_{i=1}^k V_i = V$$

分割也被称为切割(*cut*)。分割的度量可称为切割尺度(*cut-size*)，是超边线(*hyperedges*)经过切割面的数目。设 C_{ij} 是 V_i 和 V_j 分割集合间的切割尺度。每一个分割集合 V_i 有一面积 $Area(V_i) = \sum_{v \in V_i} a(v)$ 及终端点数 (*terminal count*) $Count(V_i)$ 。分割集合 V_i 拥有的最大及最小面积为 A_i^{\max} 及 A_i^{\min} 。设 $P = \{p_1, p_2, \dots, p_m\}$ 为一组超路径(*hyperpaths*)。设 $H(p_i)$ 是超路径(*hyperpaths*)的数目， P_i 是一个切割(*cut*)。

在芯片层次(*chip-level*)分割问题中，限制条件和目标函数描述如下：

- 目标 1 是在分割之间交互连接(*interconnections*)最小化。减少交互连接不但减少延迟而且减少分割间的接口，使得个别独立设计和制造比较容易的。太多的交互连接将增加设计面积并使布局(*placement*)和布线(*routing*)的运算法则复杂化。使交互连接最小化目标，可陈述如下：

$$\min \sum_{i=1}^k \sum_{j=1}^k c_{ij}, (i \neq j) \quad (8-1)$$

- 目标 2 是分割后产生最小延迟。电路的分割可能在分割数次之后建造一条重要路径(*critical path*)，为强调此问题，我们设目标函数如下：

$$\text{Min} \max_{p_i \in P} (H(p_i)) \quad (8-2)$$

- 限制条件 1 是维持终端点的数目不超过预先设定的临限值 T_i 。这限制性陈述如下：

$$Count(V_i) < T_i \quad (8-3)$$

- 限制条件 2 是维持每个分割的面积于限定的界限内。芯片层次(*chip-level*)的分割问题中，这个限制是维持每个分割保持相对可比较的尺度。否则的话，在两分割问题中可能的结果，为一分割是空的而另一个分割含有问题所有的组件。这个限制在其它的分割层次(当实际大小事先设定)不是重要的。面积限制陈述如下：

$$A_i^{\min} \leq Area(V_i) \leq A_i^{\max} \quad (8-4)$$

分割问题是类属于没有封闭模式的目标函数。在分割两个特定的实

例间，没有存在连续的空间可供描述。所以只有发展离散性不连续的解法。

我们已应用第 6 节所发展的一个基于不连续性的动态分解点搜寻方法来解分割问题。其混合式的局部最佳搜寻法中所运用的局部最佳解搜寻法是工业标准 FM 方法。我们已发展完成此方法并且已试验数个指标性问题 (benchmark questions)。试验结果显示在表 4，其由不同的起始点开始，在 50 次最小割集数目，45-55%面积平均分割条件及参着实际组件大小的限制下执行。模拟中也将工业标准 FM 方法做一结果比较。

表 4. 指标性问题的分割结果

指针电路	单元数目	从 FM 方法所获得的 割集数目	动态分解点方法	
			割集数目	改善率
S1423	831	17	15	13%
S38584	22451	199	54	268%
S38417	25589	405	120	238%
S13207	9445	91	78	17%
S15850	11071	144	79	82%
S35932	19880	120	100	20%
C7552	2247	23	21	9.5%
S9234	6098	56	51	9.8%
S5378	3225	98	76	29%

藉由用传统的 FM 方法配合我们的动态分解点方法,表 4 显示我们所求的解其质量有显著的改良。改良的范围从 9.5%到 268%,愈大的电路达到愈大的改良,小的电路有小的改良。总之,动态分解点方法明显的比 FM 方法好,在某些例子中,动态分解点方法甚至能与目前最领先的多层(multi-level)分割方法相匹敌。我们近期工作显示,藉由与多层(multi-level)分割方法,并配合我们动态分解点方法的混合式搜寻法,也能够显著的改良多层(multi-level)分割方法所得到的结果。

运用动态分解点方法的另一个改良结果是解的偏差比 FM 方法明显的小。这改良特性表示动态分解点方法比 FM 方法需要更少的执行程序就可得到好的解。

文献参考:

1. "Local Search in Combinatorial optimization" Edited by E. Aarts and J.K. Lenstra, Wiley, 1997.
2. H.D. Chiang and L. Fekih-Ahmed, "quasistability Regions of Nonlinear Dynamical Systems: Theory", IEEE Trans. On Circuits and Systems: I Fundamental Theory and Applications, Vol Cas-43, August 1996, pp. 627-635.
3. Hsiao-Dong Chiang and Chia-Chi Chu, "A Systematic Search Method for Obtaining Multiple Local Optimal Solutions of Nonlinear Programming Problems", IEEE Trans. On Circuits and Systems: I Fundamental Theory and applications, Vol. 43, No.2, Feb. 1996, pp. 99-109.
4. H.D. Chiang, M Hirsch and F.F, Wu, "Stability Regions of Nonlinear Autonomous Dynamical Systems", IEEE Trans. On Automatic Control Vol. AC-33, Jan. 1988, pp. 16-27.

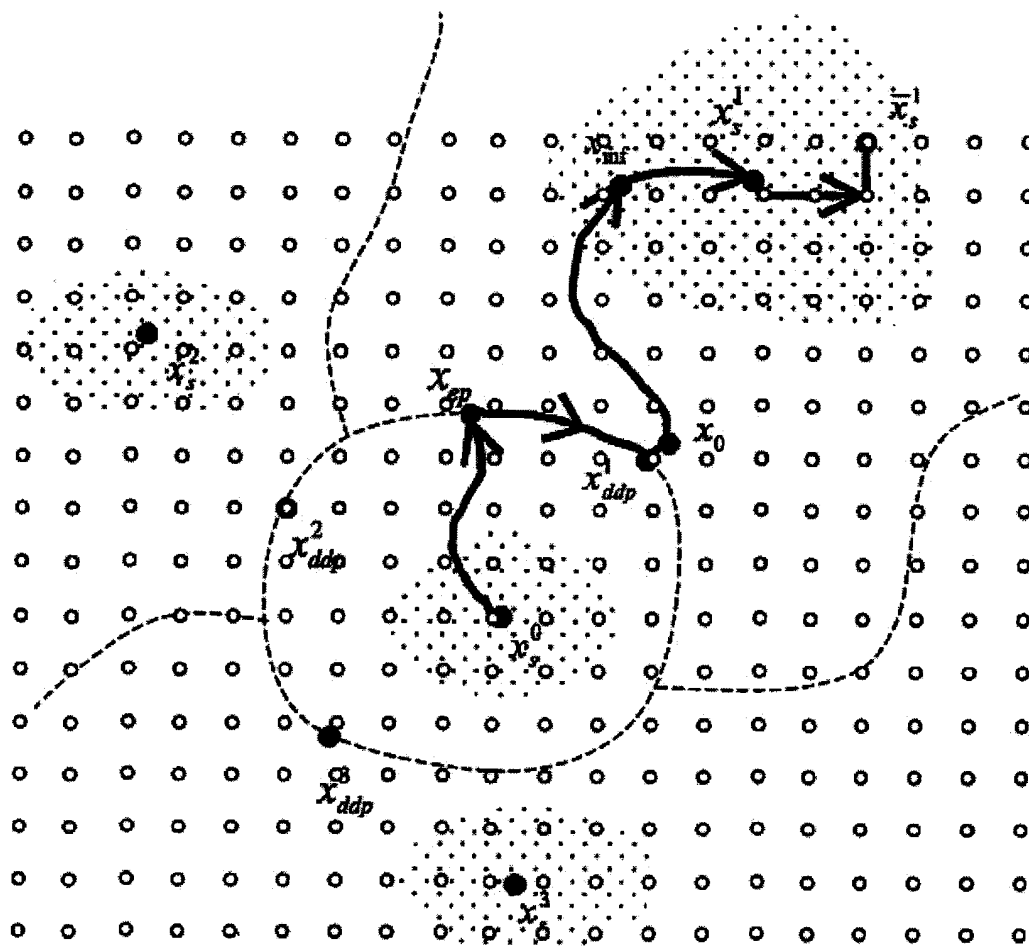


图1

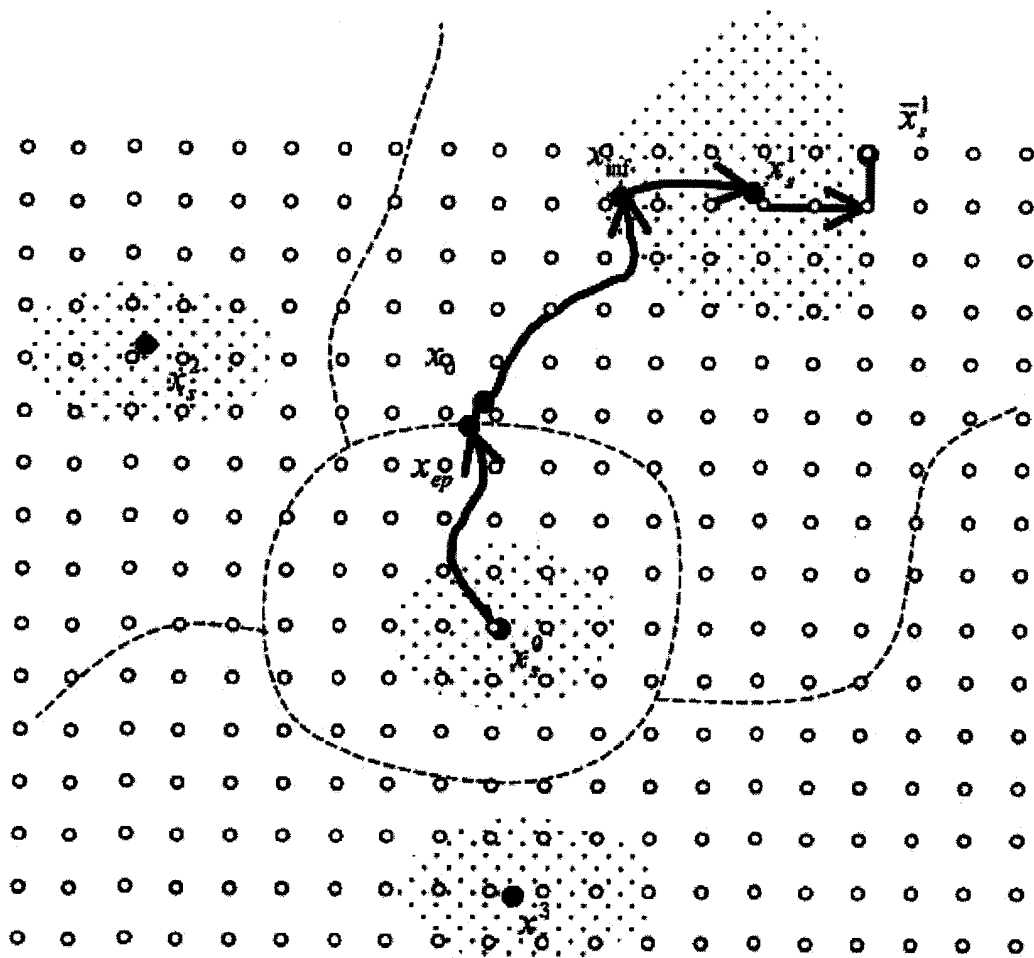


图2

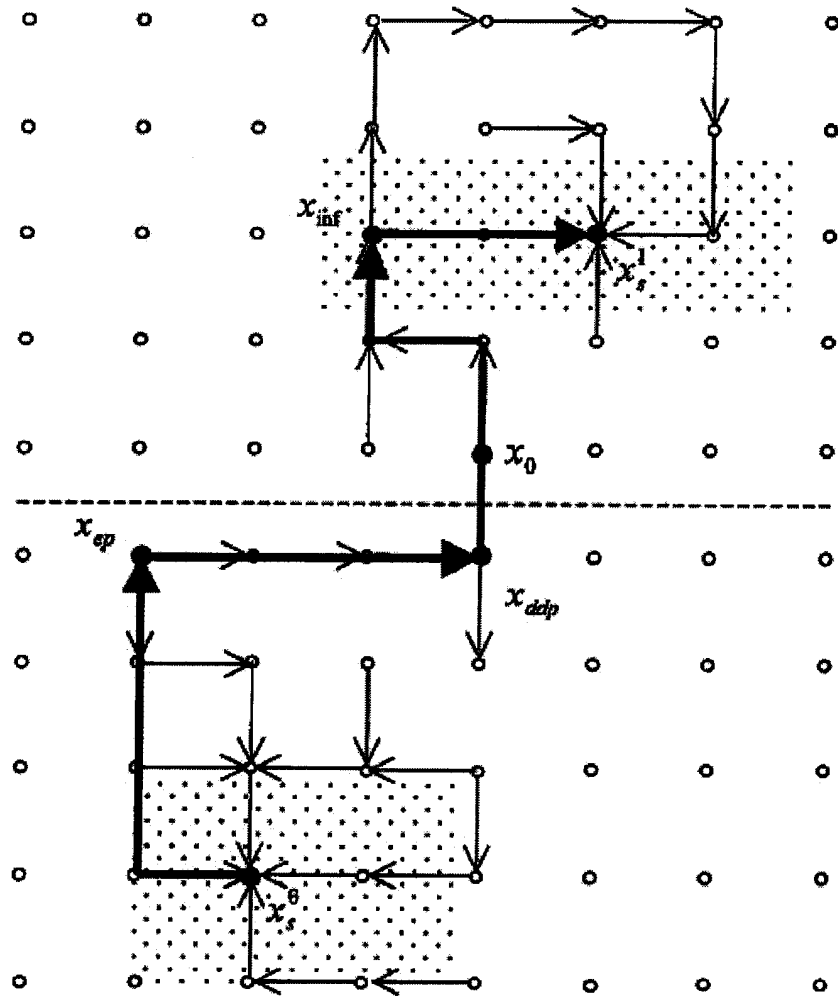


图3

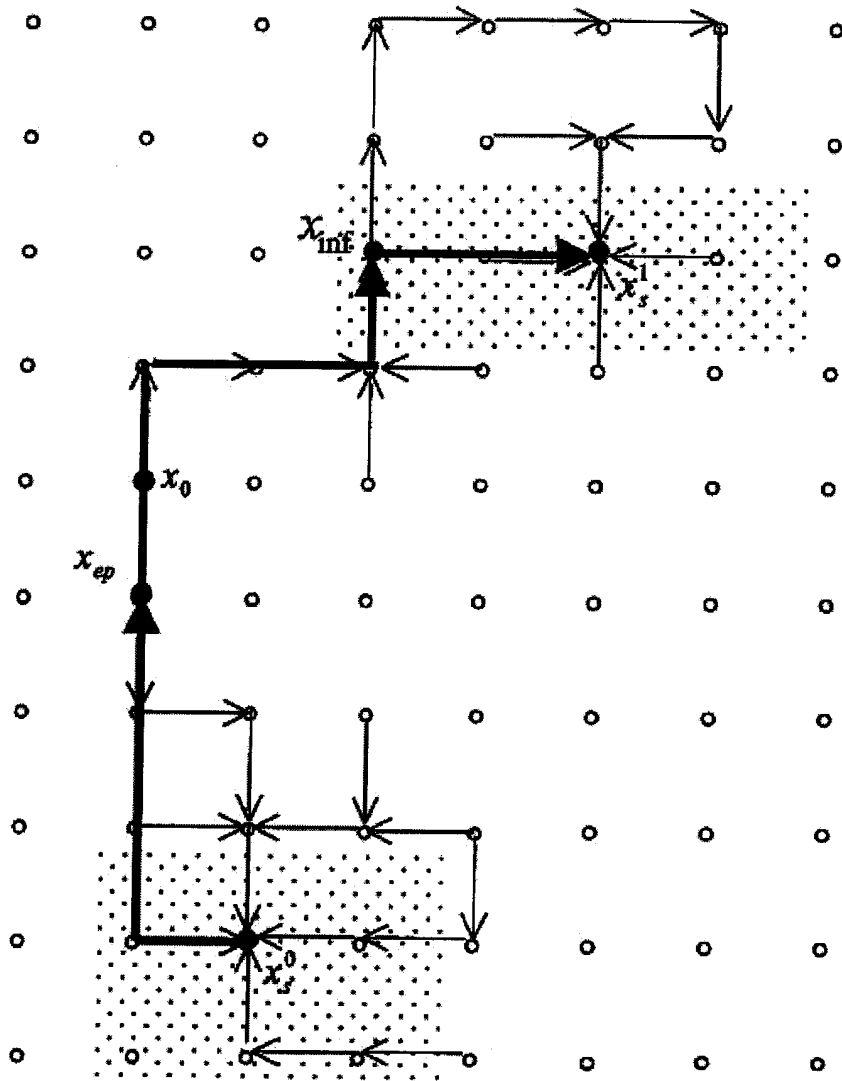


图4

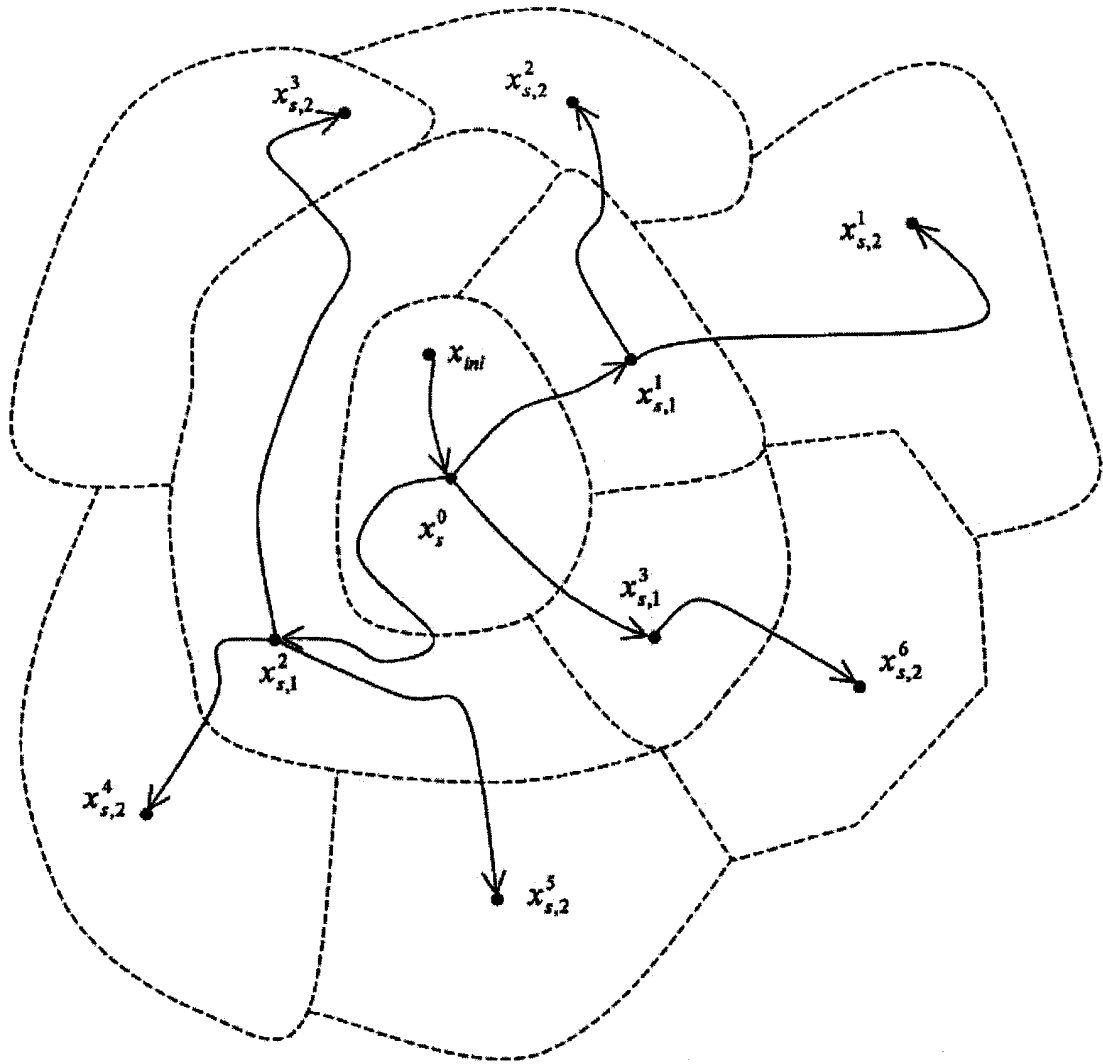


图5