中国科学
1992 年 6 月　　A 辑　　第 6 期

凸多边形置入问题求解的近似算法

李庆华　鄢勇　刘键
(华中理工大学计算机科学与工程系，武汉 430074)

摘　　要

本文在引进了挤压弹性势能另一等价定义的基础上，把求凸多边形间的嵌入深度转化为求最小碰撞距离．文中着重研究了凸多边形初始碰撞问题，得到了求嵌入深度的快速算法，据此构造出了求解凸多边形置入问题的快速近似算法．应用本文所提出的算法，不仅可解决求凸多边形间嵌入深度的实质性困难，而且可实现凸多边形几何布局问题求解自动化．

关键词：　凸多边形，嵌入深度，挤压弹性势能，初始碰撞，最小碰撞距离

平面凸多边形置入问题(亦称 Packing 问题)，在工业部门、图像处理、VLSI 的设计与制造等众多领域均有广泛的应用[1-4]，但由于它是 NP 完全问题[5]，因此关于它的近似求解算法的研究，一直受到人们的重视[6-8]．文献[6]中提出的拟物方法，存在求物体间嵌入深度的实质性困难；文献[7]对于空间物体间的嵌入深度，提出了一种可行的近似算法，但将它应用于平面凸多边形，其计算复杂度过高．本文参照文献[7]的思想，引进“弹性势能”另一等价定义，把求凸多边形间嵌入深度问题转化为求凸多边形间最小碰撞距离的问题．文中着重研究了凸多边形间的初始碰撞问题，给出了求最小碰撞距离的快速算法，从而解决了求嵌入深度的实质性困难，并据此构造了求解置入问题的快速近似算法．利用本文中所提出的算法，可实现凸多边形布局问题求解自动化．

一、置入问题求解的近似算法

按文献[6]所述想法，将区域 D 及其外部视为第零个物体，并与 N 个给定的凸多边形组成一个物体体系，如下定义体系的总的弹性势能：

$$\mathscr{U}=\sum_{i=0}^{N-1}\sum_{j=i+1}^{N}u_{ij}, \tag{1}$$

其中

$$u_{ij}=\begin{cases}0, & \text{当 } L_{ij}\geqslant 0,\\ L_{ij}^2, & \text{当 } L_{ij}<0,\end{cases} \tag{2}$$

是第 i 个凸多边形与第 j 个凸多边形间的挤压弹性势能，L_{ij} 是它们之间的距离．当 $L_{ij}<0$

本文 1989 年 5 月11日收到，1991 年 12 月 9 日收到修改稿．

时，$|L_{ij}|$ 表征了 i,j 两个凸多边形间的嵌入深度 $(i,j=0,1,2\cdots,N)$.

$\mathscr{U}$ 作为状态空间的指标函数，按文献[8]中的模式，可把求解置入问题转换成对于给定的 $\varepsilon>0$（根据实际需要给定），求使 $\mathscr{U}\leqslant\varepsilon$ 的状态．由于凸多边形用它们的顶点坐标给出，因此状态由各凸多边形的顶点坐标确定，但只有 $3N$ 个独立变元，可有多种方法求得这些状态[9].

如何求得 L_{ij}，是一实质性困难[7]．为此，参照文献[7]的思想，引进弹性势能定义的另一等价形．为叙述方便，先引进下面的定义：

定义 1. 设 $\boldsymbol{d}$ 为任一给定方向，若凸多边形 P 沿 $\boldsymbol{d}$ 以平移方式移动与凸多边形 Q 相撞，则称 $\boldsymbol{d}$ 为 P 与 Q 的可碰撞方向，并称 P 为动体，Q 为静体．

设 i,j 为两个凸多边形，$\boldsymbol{d}$ 为 i 与 j 的任一可碰撞方向，将 j 沿 $\boldsymbol{d}$ 以平移方式移动一段距离 c_{ij}（可预先设置一个充分大的常数 H），使 i,j 完全脱离（即它们交的 Lebesgue 测度为 0），然后将 j 沿 $-\boldsymbol{d}$ 以平移方式移向 i，至 j 刚刚碰到 i（即它们交的 Lebesgue 测度为 0^+），此时 j 所移过的距离记为 s_{ij}，用下式定义 u_{ij}：

$$u_{ij}=\begin{cases}0 & \text{当 } s_{ij}\geqslant H,\\ (H-s_{ij})^2 & \text{当 } s_{ij}<H.\end{cases}\tag{3}$$

将 j 刚碰到 i（即交的 Lebesgue 测度由 0 变到 0^+）的碰撞称为 i 与 j 的最初碰撞，此时 j 所移过的距离称为 i 与 j 之间的最小碰撞距离．

利用(3)式求 u_{ij} 的关键问题是怎样求得 s_{ij}，这个问题可转化为下述问题：

问题 A. 平面内两互不相交的凸多边形 P 与 Q，动体 P 沿可与 Q 碰撞的方向 $\boldsymbol{d}$ 以平移方式移向 Q，如何求得 P 与 Q 的最小碰撞距离？

因此，只要寻到求 P 与 Q 最小碰撞距离的快速算法，求 u_{ij} 的问题便告解决．

二、求最小碰撞距离的快速算法

本节研究求解问题 A 的快速算法，在下面的讨论中，假定平面凸多边形均以标准形式给出[10].

设 $P=(p_0,p_1,\cdots,p_{N-1})$ 是平面内任一凸多边形，$\boldsymbol{d}$ 是平面内任一给定方向，p,q 是平面内任意两点，以 $\overline{pq}$ 表示这两点间的线段，$|\overline{pq}|$ 表示该线段的长度，$\overrightarrow{pq}$ 表示过 p,q 两点且具有从 p 到 q 方向的直线．约定 $\varphi(\boldsymbol{d})$ 表示从 x 轴逆时针方向到 $\boldsymbol{d}$ 的夹角，$(\boldsymbol{d},\overrightarrow{pq})$ 表示从 $\boldsymbol{d}$ 逆时针方向到 $\overrightarrow{pq}$ 的夹角，显有 $(\boldsymbol{d},\overrightarrow{pq})=(\varphi(\overrightarrow{pq})-\varphi(\boldsymbol{d}))\bmod 2\pi$；对 P 进行 $\boldsymbol{d}$ 变换[11]后把 P 中具有最大 y' 值的顶点集记为 $\max(\boldsymbol{d},P)$，而把具有最小 y' 值的顶点集记为 $\min(\boldsymbol{d},P)$．用 $\{p_i\cdots p_j\}$ 表示从顶点 p_i 顺时针方向到 p_j 所经过的所有顶点的集合（包括 p_i 与 p_j 两点）．

设 $p_{\max(\boldsymbol{d})}\in\max(\boldsymbol{d},P)$，$p_{\min(\boldsymbol{d})}\in\min(\boldsymbol{d},P)$，定义集合：

$$\begin{cases}\mathrm{E}(\boldsymbol{d},P)=\{p_{\max(\boldsymbol{d})}\cdots p_{\min(\boldsymbol{d})}\},\\ \mathrm{I}(\boldsymbol{d},P)=\{p_{\min(\boldsymbol{d})}\cdots p_{\max(\boldsymbol{d})}\};\\ \mathrm{SE}(\boldsymbol{d},P)=\{\overline{p_ip_{i+1}}\mid p_i,p_{i+1}\in E(\boldsymbol{d},p)\},\\ \mathrm{SI}(\boldsymbol{d},P)=\{\overline{p_ip_{i+1}}\mid p_i,p_{i+1}\in I(\boldsymbol{d},P)\}.\end{cases}\tag{4}$$

由文献[11]中引理 2，对任意给定的 P 和 $\boldsymbol{d}$，可在 $O(1)$ 时间内判定 P 的任一顶点属于集合 $\max(\boldsymbol{d},P)$，$\min(\boldsymbol{d},P)$，$\mathrm{E}(\boldsymbol{d},P)$ 和 $\mathrm{I}(\boldsymbol{d},P)$ 中哪一个.

在下面的讨论中，均假定 P 与 Q 为任二互不相交的凸多边形，$\boldsymbol{d}$ 为 P 与 Q 的任一可碰撞方向.

P 沿 $\boldsymbol{d}$ 以平移方式移动与 Q 碰撞，有如下几种情况:

(1) P 的顶点与 Q 的顶点相撞；(2) P 的顶点与 Q 的边相撞；(3) P 的边与 Q 的顶点相撞；(4) P 的边与 Q 的边相撞.

由于边与边的碰撞可归为点与边的碰撞，P 的边与 Q 的顶点碰撞只要交换动体与静体的地位，便可变为情况(2)，因此只需讨论情况(1)与(2)两类碰撞即可.

定义 2. 设 p_i 是 P 的某个顶点，从 p_i 引与 $\boldsymbol{d}$ 平行的射线，如果该射线不穿过 P 的内部区域且不穿过 Q 的内部区域而能射到 Q 的顶点 q_j 或边 $\overline{q_jq_{j+1}}$，则称 p_i 是 P 的可碰撞顶点，q_j 或 $\overline{q_jq_{j+1}}$ 是 Q 的被 p_i 所撞的顶点或边(简称被撞点或边).

定理 1. 设 p_i 是 P 的可碰撞点，Q 的顶点 q_j 或边 $\overline{q_jq_{j+1}}$ 是 p_i 的被撞点或边，则有

$$p_i\in \mathrm{E}(\boldsymbol{d},P) \text{ 并且 } (q_j\in \mathrm{I}(\boldsymbol{d},Q) \text{ 或 } \overline{q_jq_{j+1}}\in \mathrm{SI}(\boldsymbol{d},Q)).$$

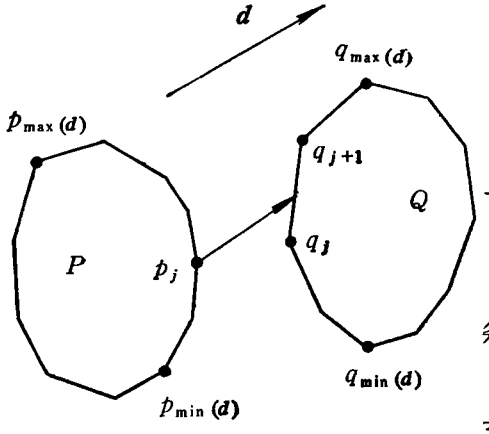

图 1 碰撞点和被撞点或边

证. 由定义 2 和(4)式，参看图 1，结论显然.

据此显然可得:

推论 1. 任 $p_i\in \mathrm{E}(\boldsymbol{d},P)$，则在 $\mathrm{I}(\boldsymbol{d},Q)$ 中最多有一个被撞点，或在 $\mathrm{SI}(\boldsymbol{d},Q)$ 中最多有一条被撞边.

引理 1. 设 $p_i\in \mathrm{E}(\boldsymbol{d},P)$，则有

(1) $q_j\in \mathrm{I}(\boldsymbol{d},Q)$，$q_j$ 是 p_i 的被撞点的充分必要条件是: $(\boldsymbol{d},\overrightarrow{p_iq_j}) \bmod 2\pi=0$;

(2) $\overline{q_jq_{j+1}}\in \mathrm{SI}(\boldsymbol{d},Q)$，边 $\overline{q_jq_{j+1}}$ 是 p_i 的被撞边的充分必要条件是: $(\overrightarrow{p_iq_j},\overrightarrow{p_iq_{j+1}})>(\overrightarrow{p_iq_j},\boldsymbol{d})$.

证. 由定义 2，引理 1 结论显然成立.

定理 1 限定了 P 与 Q 中可碰撞点和被撞点或边所在的范围，引理 1 给出了判定它们的条件，据此可构造出求 P 中的顶点 $p_i\in \mathrm{E}(\boldsymbol{d},P)$ 在 Q 中的被撞点 $q_j\in \mathrm{I}(\boldsymbol{d},Q)$ 或被撞边 $\overline{q_jq_{j+1}}\in \mathrm{SI}(\boldsymbol{d},Q)$ 的快速算法.

算法 $\mathrm{QFPS}(p_i,\{q_x\cdots q_y\},m,\boldsymbol{d})$ 求 p_i 在点集 $\{q_x\cdots q_y\}$ 中的被撞点或被撞边

输入 $p_i\in \mathrm{E}(\boldsymbol{d},P)$，$\{q_x\cdots q_y\}\subseteq \mathrm{I}(\boldsymbol{d},Q)$，$Q$ 的顶点数 m，方向 $\boldsymbol{d}$.

输出 j,s. s 是一个标志，当 $j\neq F_1$ 或 F_2 时，$s=0$ 表示 q_j 是 p_i 的被撞点；$s=1$ 表示边 $\overline{q_jq_{j+1}}$ 是 p_i 的被撞边.

Begin

if $(\boldsymbol{d},\overrightarrow{p_iq_x})<\pi$ then ($j:=F_1$; return);

if $(\boldsymbol{d},\overrightarrow{p_iq_y})>\pi$ then ($j:=F_2$; return);

/*$j=F_1$ 或 F_2，表示 p_i 在 Q 中无被撞点或边*/

```
        s:=0;
(A)     if x < y then k:=(x + y) div 2
                 else k:=((m - x + y) div 2 + x) mod m;
        /* 取 {q_x··q_y} 的中间点进行检验*/
        if φ(p_iq_k) = φ(d)then (s:=0; j:=k; return);
        /* 由引理 1, q_k 为 p_i 的被撞点 */
        if (d,p_iq_k) < π
          then if (d,p_iq_{k-1}) > π
                 then (s:=1; j:=k - 1; return)
                 /* 此时 q_{k-1}q_k 为 p_i 的被撞边 */
                 else if φ(p_iq_{k-1}) = φ(d)
                        then (s:=0; j:=k - 1; return)
                        else (y:=k - 1; goto(A))
                        /*下次搜索范围为 {q_x··q_{k-1}}*/
          else if φ (p_iq_{k+1}) = φ(d)
                 then (s:=0; j:=k + 1; return)
                 else (x:=k + 1;goto(A))
                 /* 下次搜索范围为 {q_{k+1}··q_y} */
End/*算法 QFPS*/
```

算法的正确性是显然的．由于判定每一 q_k 或 $\overline{q_kq_{k+1}}$ 是否为 p_i 的被撞点或边所花时间为 $O(1)$，而算法采用二分搜索，因此算法的时间复杂度为 $O(\log r)$，r 为集合 $\{q_x\cdots q_y\}$ 中顶点的个数．

定义 3. 设任 $p_i\in \mathrm{E}(\boldsymbol{d},P)$，如下定义 p_i 沿 $\boldsymbol{d}$ 到 Q 的碰撞距离 $\mathrm{dis}(p_i,Q,\boldsymbol{d})$：

$$\mathrm{dis}(p_i,Q,\boldsymbol{d})=\begin{cases}+\infty & p_i\text{ 在 }Q\text{ 中无碰撞点},\\ |\overline{p_iq_j}| & q_j\text{ 是 }p_i\text{ 的被撞点},\\ p_i\overset{\boldsymbol{d}}{\vdash\!\!-}\overline{q_jq_{j+1}} & \overline{q_jq_{j+1}}\text{ 是 }p_i\text{ 的被撞边},\end{cases}$$

其中 $p_i\overset{\boldsymbol{d}}{\vdash\!\!-}\overline{q_jq_{j+1}}$ 表示点 p_i 沿 $\boldsymbol{d}$ 到 $\overline{q_jq_{j+1}}$ 的距离．

由定义 3，当 p_i，q_j 以及 q_{j+1} 的坐标与方向 $\boldsymbol{d}$ 给定后，可在 $O(1)$ 时间内求得 $\mathrm{dis}(p_i,Q,\boldsymbol{d})$．

定义 4. 如下定义 P 沿 $\boldsymbol{d}$ 与 Q 碰撞的最小碰撞距离 $\mathrm{mindis}(P,Q,\boldsymbol{d})$：

$$\mathrm{mindis}(P,Q,\boldsymbol{d})=\min(\mathrm{dis}(P_i,Q,\boldsymbol{d}),\ \mathrm{dis}(q_j,P,-\boldsymbol{d})\mid p_i\in(\boldsymbol{d},P)\text{ 并且 }q_j\in \mathrm{I}(\boldsymbol{d},Q)).$$

定义 5. 若 $p_i\in \mathrm{E}(\boldsymbol{d},P)$（或 $q_j\in \mathrm{I}(\boldsymbol{d},Q)$）且有

$$\mathrm{mindis}(P,Q,\boldsymbol{d})=\mathrm{dis}(P_i,Q,\boldsymbol{d})\ (\text{或 }\mathrm{dis}(q_j,P,-\boldsymbol{d})),$$

则称 p_i 与 Q（或 q_j 与 P）的碰撞为 P 与 Q 的最初碰撞，点 p_i（或 q_j）为最初碰撞点．

定理 2. 设 $p_i \in \mathrm{E}(\boldsymbol{d},P)$, $q_j \in \mathrm{I}(\boldsymbol{d},Q)$, 且 q_j 是 p_i 的被撞点,如果条件

(i) $p_i \in \max(\boldsymbol{d},P) \wedge [q_j \in \min(\boldsymbol{d},Q) \vee (\boldsymbol{d},\overrightarrow{p_{i+1}p_i}) \leqslant (\boldsymbol{d},\overrightarrow{q_{j-1}q_j})]$, (5)

(ii) $p_i \in \min(\boldsymbol{d},P) \wedge [q_j \in \max(\boldsymbol{d},Q) \vee (\boldsymbol{d},\overrightarrow{p_ip_{i-1}}) \geqslant (\boldsymbol{d},\overrightarrow{q_jq_{j+1}})]$, (6)

(iii) $p_i \in (\mathrm{E}(\boldsymbol{d},p) - \max(\boldsymbol{d},P) - \min(\boldsymbol{d},P)) \wedge \{[q_j \in \max(\boldsymbol{d},Q) \wedge$

$(\boldsymbol{d},\overrightarrow{p_{i+1}p_i}) \leqslant (\boldsymbol{d},\overrightarrow{q_{j-1}q_j})] \vee [(\boldsymbol{d},\overrightarrow{p_{i+1}p_i}) \leqslant (\boldsymbol{d},\overrightarrow{q_{j-1}q_j}) \wedge (\boldsymbol{d},\overrightarrow{p_ip_{i-1}}) \geqslant (\boldsymbol{d},\overrightarrow{q_jq_{j+1}})]\}$ (7)

中的一个成立,则 $\mathrm{mindis}(P,Q,\boldsymbol{d}) = |\overline{p_iq_j}|$.

证. 只证 (iii),(i) 与 (ii) 的证明类似.

分三种情况证明.

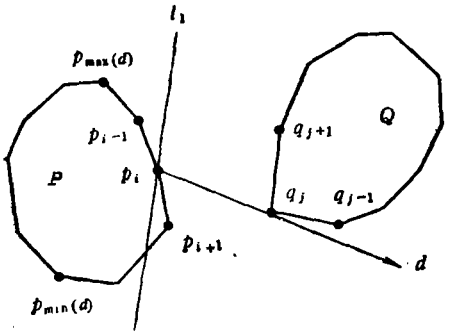

图 2 直线 l_1 平行于 $\overrightarrow{q_jq_{j+1}}$

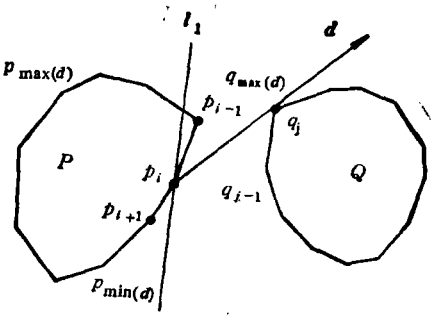

图 3 过点 p_i 作直线 l_1 平行于 $\overrightarrow{q_{j-1}q_j}$

情况 1. $q_j \in \min(\boldsymbol{d},Q)$.

过点 p_i 作直线 l_1 平行 $\overrightarrow{q_jq_{j+1}}$, 由于 $(\boldsymbol{d},\overrightarrow{p_ip_{i-1}}) \geqslant (\boldsymbol{d},\overrightarrow{q_jq_{j+1}})$, 所以点集 $\{p_{\max(d)} \cdots p_{i-1}\}$ 与 Q 分处在 l_1 两侧(图 2). 又由于 $q_j \in \min(\boldsymbol{d},Q)$, 所以 $\{p_{i+1} \cdots p_{\min(d)}\}$ 中无顶点可与 Q 碰撞,因此 $\mathrm{E}(\boldsymbol{d},P)$ 中所有可与 Q 碰撞的点中, p_i 与 q_j 的碰撞是 P 与 Q 的最初碰撞, 因此有

$$\mathrm{mindis}(P,Q,\boldsymbol{d}) = |\overline{p_iq_j}|.$$

情况 2. $q_j \in \max(\boldsymbol{d},Q)$.

证明方法与情况 1 相同 (参看图 3).

情况 3. $q_j \in (\mathrm{I}(\boldsymbol{d},Q) - \max(\boldsymbol{d},Q) - \min(\boldsymbol{d},Q))$.

过点 p_i 作直线 l_1 平行于 $\overrightarrow{q_jq_{j+1}}$ (图 4), 由于 $(\boldsymbol{d},\overrightarrow{p_ip_{i-1}}) \geqslant (\boldsymbol{d},\overrightarrow{q_jq_{j+1}})$, 所以在点集 $\{p_{\max(d)} \cdots p_i\}$ 中 p_i 与 q_j 的碰撞先于其它点与 Q 的碰撞.

又过点 p_i 作直线 l_2 平行于 $\overrightarrow{q_{j-1}q_j}$,由于 $(\boldsymbol{d},\overrightarrow{p_{i+1}p_i}) \leqslant (\boldsymbol{d},\overrightarrow{q_{j-1}q_j})$, 所以在点集 $\{p_i \cdots p_{\min(d)}\}$ 中, p_i 与 q_j 的碰撞先于其它点与 Q 的碰撞. 综上所叙,可得

$$\mathrm{mindis}(P,Q,\boldsymbol{d}) = |\overline{p_iq_j}|.$$

定理 3. 设 $p_i \in \mathrm{E}(\boldsymbol{d},P)$, $q_j \in \mathrm{I}(\boldsymbol{d},Q)$, 且 q_j 是 p_i 的被撞点,如果 $\mathrm{mindis}(P,Q,\boldsymbol{d}) = |\overline{p_iq_j}|$, 则(5)—(7)式中至少有一个成立.

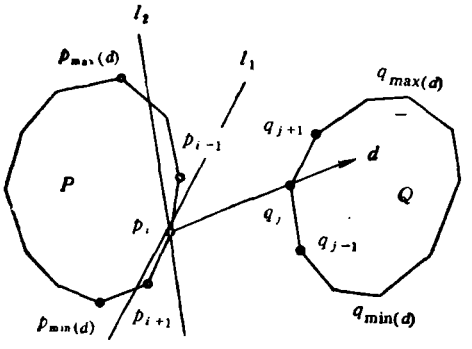

图 4　l_1 平行于 $\overrightarrow{q_jq_{j+1}}$，$l_2$ 平行于 $\overrightarrow{q_{j-1}q_j}$

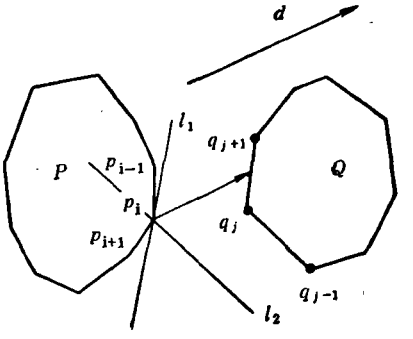

图 5　点 p_i 与边 $\overrightarrow{q_jq_{j+1}}$ 碰撞

证. 设 $p_i \in (\mathrm{E}(\boldsymbol{d},P) - \max(\boldsymbol{d},P) - \min(\boldsymbol{d},P))$. 若 $q_j \in \max(\boldsymbol{d},Q)$，则 $\{p_{\max(d)} \cdots p_{i-1}\}$ 中无点与 Q 相撞，只有 $\{p_{i+1} \cdots p_{\min(d)}\}$ 中可能有点与 Q 相撞. 假定 $(\boldsymbol{d}, \overrightarrow{p_{i+1}p_i}) > (\boldsymbol{d}, \overrightarrow{q_{j-1}q_j})$，参看图 3，此时 p_{i+1} 与 Q 处于 l_1 同侧，于是 p_i 与 q_j 的碰撞不是 P 与 Q 的最初碰撞，与题设矛盾，因此应有 $(\boldsymbol{d}, \overrightarrow{p_{i+1}p_i}) \leqslant (\boldsymbol{d}, \overrightarrow{q_{j-1}q_j})$. 若 $q_j \in (\mathrm{I}(\boldsymbol{d}, Q) - \max(\boldsymbol{d}, Q))$，假定 $(\boldsymbol{d}, \overrightarrow{p_ip_{i-1}}) < (\boldsymbol{d}, \overrightarrow{q_jq_{j+1}})$，参看图 2 和图4，点 p_{i-1} 与 Q 同处于 l_1 同侧，此时有 $|\overline{p_iq_j}| > p_{i-1} \overset{d}{\vdash} \overline{q_jq_{j+1}}$ 或 $q_{j+1} \overset{d}{\vdash} \overline{p_ip_{i-1}}$，因此 $\mathrm{mindis}(P, Q, \boldsymbol{d}) \neq |\overline{p_iq_j}|$，与题设矛盾，故应有 $(\boldsymbol{d}, \overrightarrow{p_ip_{i-1}}) \geqslant (\boldsymbol{d}, \overrightarrow{q_jq_{j+1}})$. 同理可证应有 $(\boldsymbol{d}, \overrightarrow{p_{i+1}p_i}) \leqslant (\boldsymbol{d}, \overrightarrow{q_{j-1}q_j})$，即(7)式成立.

(5)或(6)式的成立同理可证.

定理 4. 设 $p_i \in \mathrm{E}(\boldsymbol{d},P)$，$\overline{q_jq_{j+1}} \in \mathrm{SI}(\boldsymbol{d},Q)$ 是 p_i 的被撞边，$\mathrm{mindis}(P,Q,\boldsymbol{d}) = p_i \overset{d}{\vdash} \overline{q_jq_{j+1}}$ 的充分必要条件是

$$(\boldsymbol{d}, \overrightarrow{p_{i+1}p_i}) \leqslant (\boldsymbol{d}, \overrightarrow{q_{j-1}q_j}) \wedge (\boldsymbol{d}, \overrightarrow{p_ip_{i-1}}) \geqslant (\boldsymbol{d}, \overrightarrow{q_jq_{j+1}}). \tag{8}$$

证. 过点 p_i 作直线 l_1 平行 $\overrightarrow{q_jq_{j+1}}$，作直线 l_2 平行 $\overrightarrow{q_{j-1}q_j}$ (图 5)，余下的证明方法与定理 2 和 3 的证明方法相同，故从略.

由定理 2，3 与 4，在已知 p_i 和它的被撞点 q_j 或被撞边 $\overline{q_jq_{j+1}}$ 的情况下，可在 $O(1)$ 时间内判定 p_i 与 q_j 或 $\overline{q_jq_{j+1}}$ 的碰撞是否为 P 与 Q 的最初碰撞，亦即可在 $O(1)$ 时间内确定 $\mathrm{dis}(p_i,Q,\boldsymbol{d})$ 是否为 $\mathrm{mindis}(P,Q,\boldsymbol{d})$.

下面构造算法 FMDIS，用以确定 $\mathrm{E}(\boldsymbol{d},P)$ 中的初始碰撞点，若初始碰撞点存在，则输出 P 与 Q 的最小碰撞距离 $\mathrm{mindis}(P,Q,\boldsymbol{d})$，若无初始碰撞点，则输出无初始碰撞点的标记 $t=1$.

Procedure FMDIS($\{p_{\max(d)} \cdots p_{\min(d)}\}, n, \{q_{\min(d)} \cdots q_{\max(d)}\}, m, \boldsymbol{d}$)

Begin

$p_{i_1} := P_{\max(d)}$; $p_{i_1} := p_{\min(d)}$; $q_{i_2} := q_{\max(d)}$; $q_{i_2} := q_{\min(d)}$; $t := 0$;

(B)　if $j_1 \geqslant i_1$ then $i := (i_1 + j_1)$ div 2

```
            else i:=((n - i1 + j1) div 2 + i1) mod n;
    QFPS(pi,{qi1··qi2},m,d);
    if j = F1 then (j1:=i - 1; goto(B));
    if j = F2 then (i1:=i + 1;goto(B));
    if s = 0∧j = k /* qj 为 pi 的被撞点,即 (d,piqj) mod 2π = 0 */
       then {if (条件(5)∨(6)∨(7))
                then (mindis:=|piqj|; return)
                else goto(C)}
       else /*此时必有 s = 1,j = k, 边 qjqj+1 为 pi 的被撞边 */
          if 条件(8) then (mindis:=pi|--d--qjqj+1; return);
(C) if i1 = j1 then (t:=1; return); /* 输出无初始碰撞点标志 */
    if(d,pipi-1) < (d,qjqj+1)
      then[if i = i1 then (t:=1; return)
                else (j1:=i - 1; j2:=j; goto(B))]
      else[if j1 = i then (t:=1; return)
                else (i1:=i + 1; i2:=j; goto(B))]
End. /* 算法 FMDIS */
```

算法 FMDIS 每次选取 $\{p_{i_1}\cdots p_{i_1}\}$ 的中点 p_i，调用一次 QFPS，寻找 p_i 的被撞点或边。当找到它的被撞点或边，就判定 p_i 是否为最初碰撞点，若是则求出 $\mathrm{dis}(p_i,Q,\boldsymbol{d})$，否则继续折半查找。算法中片段 (C) 将查找范围折半，由定理 2，3 和 4 可知，折半后省略掉的点一定不是最初碰撞点，故算法的正确性是显然的．由于算法 QFPS 的时间复杂度为 $O(\log m)$，因此算法 FMDIS 的时间复杂度在最坏情况下为 $O(\log n \times \log m)$。

利用算法 QFPS 和 FMDIS，可构造出求 P 与 Q 碰撞的最小碰撞距离的算法 MPQD：

Procedure MPQD $(P,Q,\boldsymbol{d})$

Begin

(1) 利用文献 [11] 中算法 1 求得 $p_{\max(\boldsymbol{d})}$，$p_{\min(\boldsymbol{d})}$，$q_{\max(\boldsymbol{d})}$ 与 $q_{\min(\boldsymbol{d})}$，并判定 $\boldsymbol{d}$ 是否为 P 与 Q 的可碰撞方向，若不是，则无最小碰撞距离，停止；否则

(2) FMDIS $(\{p_{\max(\boldsymbol{d})}\cdots p_{\min(\boldsymbol{d})}\}$，$n$，$\{q_{\min(\boldsymbol{d})}\cdots q_{\max(\boldsymbol{d})}\}$，$m$，$\boldsymbol{d})$；若 $t=0$，则得 $\mathrm{mindis}(P,Q,\boldsymbol{d})$，停止；否则

(3) $\mathrm{FMDIS}(\{q_{\min(\boldsymbol{d})}\cdots q_{\max(\boldsymbol{d})}\},m,\{p_{\max(\boldsymbol{d})}\cdots p_{\min(\boldsymbol{d})}\},n,-\boldsymbol{d})$

End.

若 P 与 Q 碰撞，其最小碰撞距离一定存在，在 $\mathrm{E}(\boldsymbol{d},P)$ 或 $\mathrm{I}(\boldsymbol{d},Q)$ 中一定可找到最初碰撞点，因此算法中的(2)或(3)一定求得 $\mathrm{mindis}(P,Q,\boldsymbol{d})$. 算法的时间复杂度为 $O(\log n \times \log m)$。

至此问题A得到解决，即利用算法 MPQD 可在不超过 $O(\log n \times \log m)$ 时间内求得 s_{ij}，亦即求得 u_{ij}。这是目前求 u_{ij} 的最为快速的算法。

三、凸多边形与区域D间的挤压弹性势能

欲用(1)式给定$\mathscr{U}$，尚需确定凸多边形 i 与区域D间的挤压弹性势能 $u_{0i}(i=1,2,\cdots,N)$。平面区域D随着实际问题的需要可能是各种形状，但使用最多的是矩形区域和圆形区域，下面仅对这两种区域进行讨论。

(一) 设区域D为矩形，边长分别为 a 和 b。如图 6 所示建立坐标系 xoy，对于凸多边 i，求出 $i_0=i_{\max(x)}$，$i_1=i_{\min(y)}$，$i_2=i_{\min(x)}$ 和 $i_3=i_{\max(y)}$，并记为 $i_k(x_{i_k},y_{i_k})(0\leqslant k\leqslant 3)$。若有点 $i_l(0\leqslant l\leqslant 3)$ 满足下式

$$y_{i_l}>b\vee y_{i_l}<0\vee x_{i_l}>a\vee x_{i_l}<0,$$

则点 i_l 落在D的外部，否则落在D的内部。于是可如下求得 i 与D的嵌入深度 $s_l(0\leqslant l\leqslant 3)$：

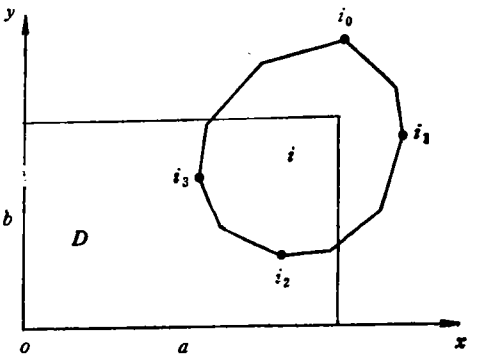

图 6　在区域D上建立坐标系 xoy

```
if 0 ≤ x_{i_l} ≤ a
  then if y_{i_l} > b then s_l := y_{i_l} - b
                  else if y_{i_l} < 0 then s_l := |y_{i_l}|
                                  else s_l := 0
  else if x_{i_l} < 0 then s_l := |x_{i_l}| else s_l := x_{i_l} - a.
```

可如下确定 u_{0i}：

$$u_{0i}=\sum_{l=0}^{3}s_l^2.$$

于是对任凸多边形 i，设其顶点数为 n_i，则可在 $O(\log n_i)$ 时间内确定$u_{0i}(i=1,2,\cdots,N)$。

(二) 设区域D为圆形，其半径为 R。以D的圆心为坐标原点建立坐标系 xoy，设凸多边形 i 距圆心最远的顶点为 $i_l(x_{i_l},y_{i_l})$，则如下确定 u_{0i}：

$$u_{0i}=\begin{cases}0, & x_{i_l}^2+y_{i_l}^2\leqslant R^2\\ (\sqrt{x_{i_l}^2+y_{i_l}^2}-R)^2, & x_{i_l}^2+y_{i_l}^2>R^2.\end{cases}$$

求得点 i_l 需 $O(n_i)$ 时间，因此可在 $O(n_i)$ 时间内求得 u_{0i}。

至此可得(1)式定义的总弹性势能$\mathscr{U}$，可采用本文第一节所述方法进行求解，至于计算方法和程序设计中的若干考虑，可参考文献[7]与[11]。

纵观$\mathscr{U}$的计算过程，可以看出主要工作量集中在求 $u_{ij}(i\neq j,\ i,j=1,2,\cdots,N)$上。事实上并不是所有的凸多边形都互相嵌入，因此没有必要对任二凸多边形都计算它们间的 s_{ij}。下面简要讨论一下计算量削减问题。

(1) 可碰撞方向 $\boldsymbol{d}$ 的确定。设 i,j 为二凸多边形，i_l 与 j_k 分别 i 与 j 的任一顶点，可以证明 $\overrightarrow{i_lj_k}$ 是 i 与 j 的一可碰撞方向，于是取 $\boldsymbol{d}$ 为 $\overrightarrow{i_lj_k}$，不必做其他的计算；

(2) 关于 s_{ij} 的计算。

a. 可应用文献[10]中的算法 IGG，判定 i 与 j 是否相交，若不交，则 $u_{ij}=0$，无需调用

MPQD 计算 s_{ij};

b. 对 i 与 j 分别求出 $i_{\max(x)}, i_{\min(y)}, i_{\min(x)}, i_{\max(y)}$ 和 $j_{\max(x)}, j_{\min(y)}, j_{\min(x)}, j_{\max(y)}$, 并分别用 $i_l(x_{i_l}, y_{i_l})$ 与 $j_k(x_{j_k}, y_{j_k})$ $(0 \leqslant l, k \leqslant 3)$ 记之. 如果条件

$$y_{i_0} \leqslant y_{j_1} \vee y_{j_0} \leqslant y_{i_2} \vee x_{i_1} \leqslant x_{j_3} \vee x_{j_1} \leqslant x_{i_3}$$

成立,则 i 与 j 互不嵌入, $u_{ij}=0$; 否则 x 轴(或 y 轴)方向便是 i 与 j 的一可碰撞方向[11], 于是可选 $\boldsymbol{d}$ 为 x (或 y)轴方向,调用算法 MPQD 时,算法中的(1)可不执行.

做 a 或 b 的工作只需 $O(\log(n_i + n_j))$ 时间,而计算 s_{ij} 需 $O(\log n_i \times \log n_j)$ 时间, 从而可使计算量大大削减.

四、结 束 语

上面主要讨论置入问题的求解,为要解决布局问题,还必须考虑诸凸多边形间的相互方位应满足的条件,可把这些条件当做约束条件与$\mathscr{U}$ 相加作为指标函数,再按所述方法求解.

以$\mathscr{U}$ 作为指标函数求可行点,尚存在两个实质性困难[6,7],一是求 u_{ij}, 二是怎样避免虚假最小点造成的计算时间过量消耗. 本文所做的工作解决了凸多边形求 u_{ij} 的实质性困难. 至于第二个困难,可望借助计算机图形学和图形数据库的成果获得解决.

参 考 文 献

[1] Rogers, C. A., *Packing and Covering*, Cambridge University Press, 1964.

[2] Ebeling, C. W., *Pallet Pattern by Computer*, Modern Packing, 1971, 66—69, 70.

[3] Preas, B. T. & Vancleempur, W. M., *Design Automation Conference Proc*, 16(1979), 474—480.

[4] Sha, L. & Dutton, R. W., *ibid*., 22(1985), 602—608.

[5] Fowler, R. J. et al., Optimal Packing and Covering in the Plane are NP-Complete, *Inform, Press, Lett.*, 12 (1981), 133—137.

[6] 黄文奇、詹叔浩,应用数学学报,2(1979),176—180.

[7] 黄文奇等,同上, 9 (1986), 4: 443—453.

[8] Dorit, S., Hochbaum & Wolfgang Mass, *J. of ACM*, 32(1985), 1: 130—136.

[9] 钱焕延,计算方法,上海交通大学出版社, 1988, 271—338.

[10] Chazlle, B. & Dobkin, D. P., *J. of ACM*, 36(1987), 1: 1—27.

[11] 李辉,中国科学 A 辑, 1987, 12: 1301—1308.