<u>PLANEJAMENTO DE CAMINHOS</u>

MÉTODOS BASEADOS EM DECOMPOSIÇÃO EM CÉLULAS – DECOMPOSIÇÃO EXATA

<u>Princípio</u>:

- Decomposição de C_L em regiões não superpostas cuja união é exatamente C_L.
- Construção de um grafo de conectividade G que representa as relações de adjacência entre as células.
- Busca de um <u>Canal</u> (seqüência de células adjacentes ligando a célula contendo q_{ini} à célula contendo q_{fin}) em G
- Extração de um caminho entre q_{ini} e q_{fin} a partir do canal.

<u>Características desejáveis da Decomposição</u>:

- Geometria de célula simples, que permita computar caminhos com facilidade.

- Deve ser fácil testar a adjacência entre células.

- Deve ser fácil achar caminhos através dos limites entre células.

Observações:

- Células são regiões não críticas, mudanças dentro delas ocorrem de forma contínua.

- Limites entre células correspondem a condições críticas, mudanças bruscas ocorrem ao atravessar um limite.

- Um Canal é menos restritivo do que um caminho. Permite mais flexibilidade no planejamento.

<u>Decomposição Exata em Espaço de Configuração Poligonal</u>:

- C = de ⌡^{2}. C_B = Região Poligonal. C_L = C\CB limitado.

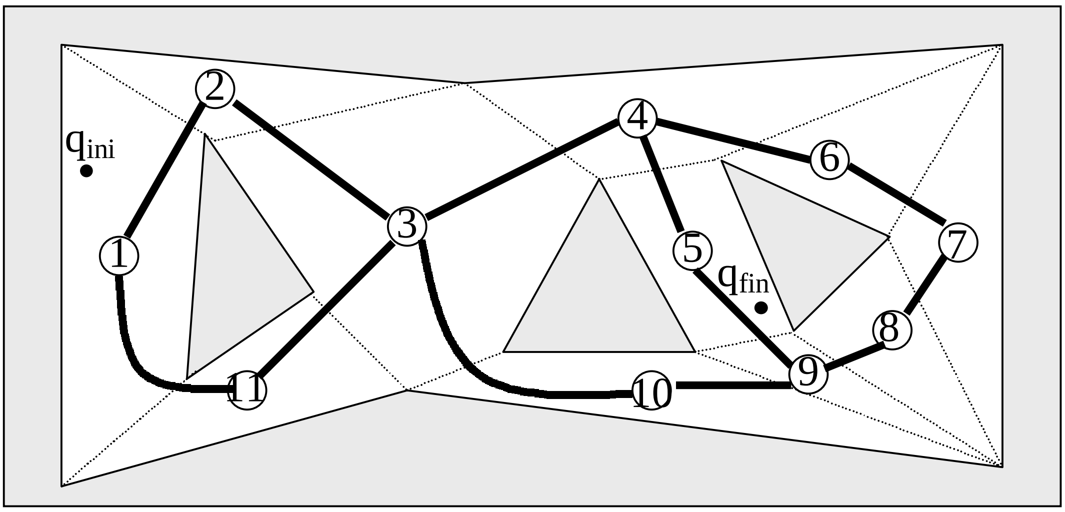
- <u>Decomposição Poligonal Convexa, K, de C_L</u> = coleção finita de polígonos convexos (<u>Células</u>), tal que:

    - Dadas duas células quaisquer, k e k' da decomposição K, $\mathrm{int}(k) \cap \mathrm{int}(k') = \emptyset$.
    - A união de todas as células é exatamente $\mathrm{cl}(C_L)$.

⇒ Duas células k e k' em K são adjacentes se e somente se $k \cap k'$ é um segmento de reta de comprimento não nulo.
⇒ Sempre existe um caminho livre entre duas configurações dentro de uma mesma célula, o qual é um segmento de reta.

<u>Grafo de Conectividade, G</u>, associado a uma decomposição K de C_L, é um grafo não direcionado, tal que:

- Seus nós são as células da decomposição K.
- Dois nós em G são conexos por um arco se e somente se as células correspondentes são adjacentes.



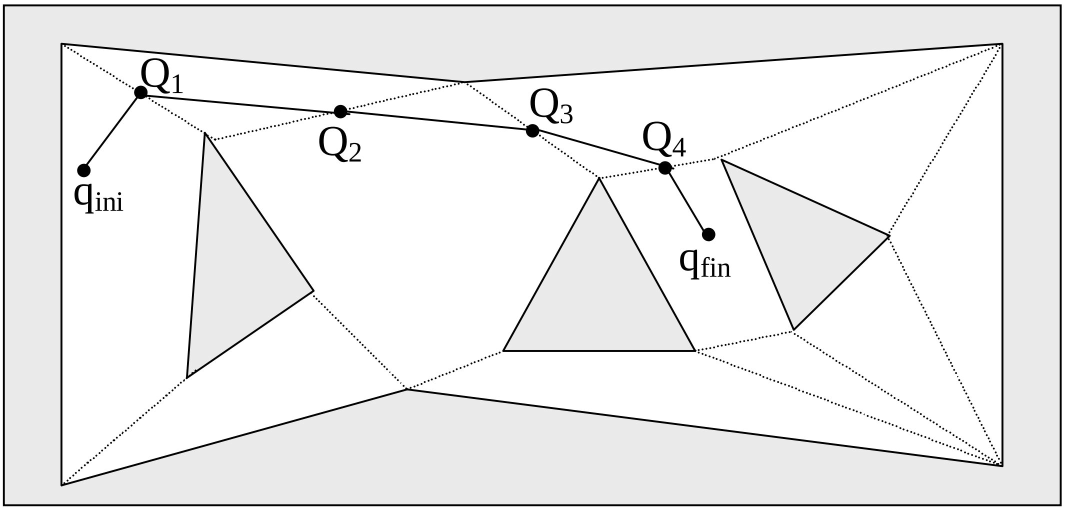
Algoritmo de planejamento:

1. Gerar uma decomposição poligonal K de C_L.
2. Construir o grafo de conectividade G associado a K.
3. Buscar, em G, uma seqüência de células adjacentes entre k_{ini} (célula que contém q_{ini}) e k_{fin} (célula que contém q_{fin}).
4. Se sucesso, retornar a seqüência gerada, se não, reportar falha.

- Canal: seqüência de células adjacentes , $k_1, \ldots, k_p$, tais que $q_{ini} \in k_1$, $q_{fin} \in k_p$ e $\forall\ i \in [1, p-1]$, k_i e k_{i+1} são adjacentes, com $int(\cup k_i) \subset C_L$.

Método simples para obter um caminho dentro de um canal:

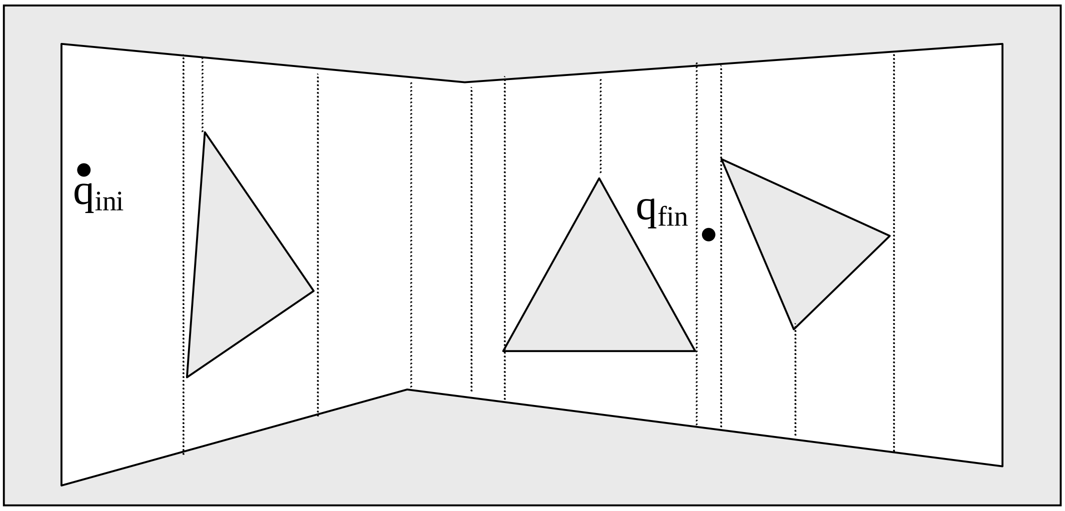
- Dados os limites $\beta_i = \partial k_i \cap \partial k_{i+1}$, entre duas células adjacentes k_i e k_{i+1}, determinar os pontos médios Q_i de β_i.
- ligar q_{ini} a q_{fin} através da linha poligonal por $Q_1, Q_2, \ldots, Q_{p-1}$.
- Caso β_i e β_{i+1} estejam contidos na mesma reta suporte, criar um ponto intermediário $Q_{i,i+1}$.



Observação: a decomposição convexa ótima de um polígono é computável em tempo polinomial no número de vértices. A presença de buracos no polígono torna o problema NP completo.

<u>Decomposição Trapezoidal (não ótima)</u>:

- Varrer C_L com uma linha reta vertical.
- Quando um vértice X de CB é encontrado, um máximo de dois segmentos de reta verticais, contidos em C_L, são criados de modo a conectar X aos eixos de CB imediatamente acima e abaixo do mesmo.
- Os limites de CB e os segmentos verticais determinam a <u>Decomposição Trapezoidal</u> de C_L. $\Rightarrow$ Cada célula é um trapezóide ou um triângulo.
- Duas células são adjacentes se e somente se seus limites partilham um dos segmentos verticais gerados na varredura.



Observações:

- Quando um segmento vertical é atravessado, a restrição correspondente imposta por CB muda descontinuamente.
- Se C_L é não limitado, a decomposição incluirá células que se estendem infinitamente, para cima e/ou para baixo.
- Durante a varredura, é possível computar de modo concorrente (com O(n.log(n)) , n = número de vértices de CB) a criação dos segmentos verticais, a geração do grafo de conectividade, bem como a identificação das células que contém q_{ini} e q_{fin}.

Busca do Caminho no Grafo de Conectividade:

- Busca baseada no Algoritmo A^* produz o menor caminho, (em métrica euclidiana), entre q_{ini} e q_{fin} dentro do grafo de conectividade.
- Para uso do algoritmo A^*, um grafo de conectividade alternativo, G’, pode ser especificado:

    - Os nós de G’ são q_{ini}, q_{fin}, e os pontos centrais Q_i’s dos limites compartilhados por células adjacentes.
    - Dois nós em G’ são conexos por um arco se e somente se eles pertencem à mesma célula (ou seja, podem ser ligados por um segmento de reta inteiramente contido em C_L).

- Cada arco de G’ é ponderado pela distância euclidiana entre os nós que ele conecta.
- A distância euclidiana a q_{ini} pode ser adotada como função heurística na busca.

Observações:

- Os caminhos gerados são significativamente melhores do que aqueles gerados através do grafo de visibilidade: evitam melhor os obstáculos, sem se afastar muito deles.

- O método pode ser estendido para $C = \mathbb{R}^3$, com obstáculos poliédricos, varrendo o espaço com um plano de modo a criar células paralelepípedas. Duas células são adjacentes se e somente se partilham, em uma face, um trapezóide de área não nula. O algoritmo A^* pode ser aplicado ao grafo G’ definido acima, com cada ponto Q_i escolhido no interior do trapezóide resultante da interseção dos limites de duas células adjacentes.

Decomposição Exata - Translação e Rotação no Plano:

- **W** = $\mathbb{R}^2$, povoado de obstáculos poligonais (cuja união é **B**).
- **A** = segmento de reta PQ de comprimento d.
- **C** = $\mathbb{R}^2 \times S^1$.
- Configuração q = (x, y, θ), com (x, y) = posição de P e θ = ângulo de orientação entre o vetor PQ e o eixo x de F_W.

Princípio: decompor o conjunto de posições de **A** em regiões não críticas bidimensionais, definir células tridimensionais "elevando" estas regiões ao longo do eixo θ e extrair delas um *Grafo de Conectividade* que represente as relações de adjacência entre as mesmas.

- Os C-obstáculos mantêm estrutura constante dentro da projeção tridimensional de uma região não crítica.
- As células contidas em tais projeções tridimensionais (prismas) tornam explícita esta estrutura.
- Os limites das regiões não críticas são as curvas críticas.

Curvas Críticas e Regiões Não Críticas:

- Uma face de CB do tipo A corresponde a configurações em que o segmento PQ contém um vértice de CB.
- Uma face de CB do tipo B corresponde a configurações em que um eixo de CB contém o ponto P ou o ponto Q.

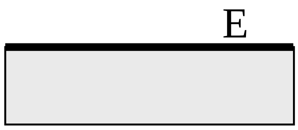
- Curvas críticas são projeções no plano xy das seguintes curvas:

  - os limites das faces de CB;
  - as curvas nas faces de CB onde o plano tangente a CB é perpendicular ao plano xy.

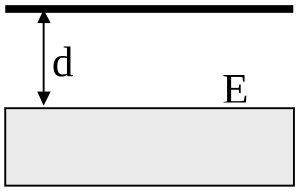
⇒ Quando uma curva crítica é atravessada, muda o conjunto de faces de CB que são intersectadas por uma linha perpendicular ao plano xy, ou muda o número de pontos de interseção.

Tipos de curvas críticas:

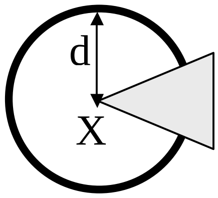
- **Tipo 0**: eixo E de um obstáculo B.

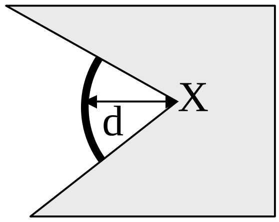


- **Tipo 1**: segmento de reta a uma distância d de um eixo E de um obstáculo B. (Possui o mesmo comprimento de E).

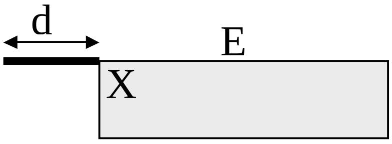


- **Tipo 2**: arco de circunferência de raio d, centrado no vértice X de um obstáculo B e limitado pelas semi-retas que contêm os eixos de B que têm X por extremidade.

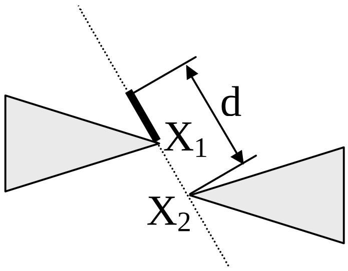




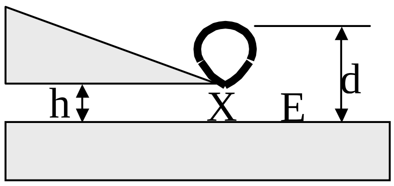
- **Tipo 3**: segmento de reta traçado por P quando **A** desliza contido na reta suporte de um eixo E limitado por um vértice convexo X, de tal modo que Q permanece contido em E.



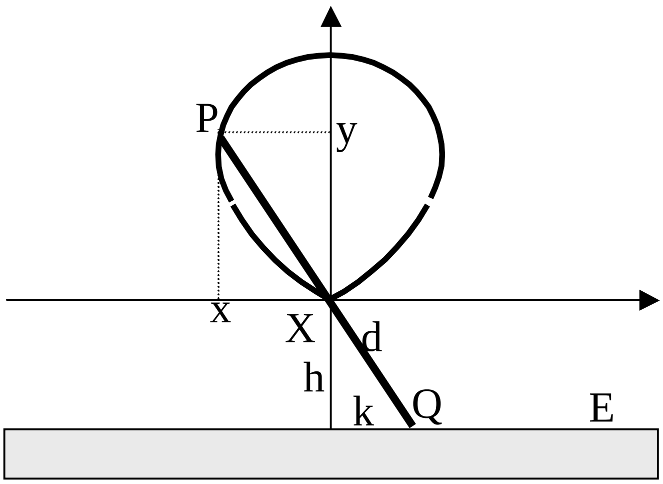
- **Tipo 4**: segmento de reta traçado por P, enquanto **A** mantém contato simultâneo com dois vértices convexos, X_1 e X_2.



- **Tipo 5**: curva traçada por P enquanto **A** mantém contato simultâneo com um eixo E e um vértice convexo X situado a uma distância h < d de E, de tal modo que **A** é tangente a B em X e Q permanece contido em E.



- Esta curva é um subconjunto de um tipo de curva algébrica de 4º grau denominada Conchóide de Nicomedes.

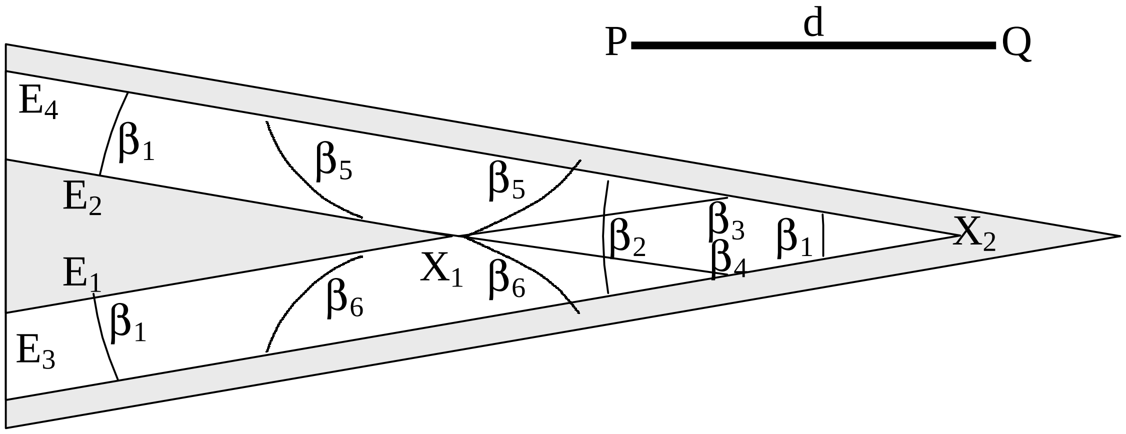


Da figura acima: $d^2 = (y+h)^2 + (x+k)^2$

$y/x = h/k$

$\Rightarrow \quad x^2 = y^2.(d^2/(y+h)^2 - 1)$

Exemplo de decomposição:

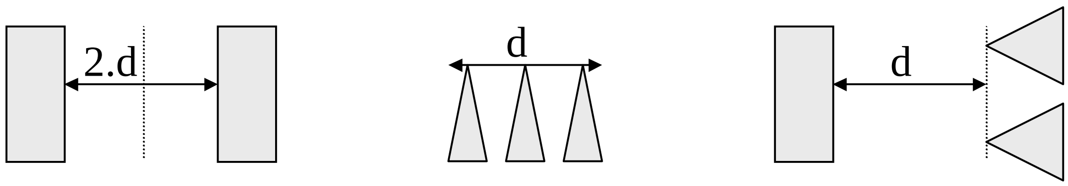


E_1, E_2, E_3 e E_4: curvas críticas do tipo 0.
β_1 e β_2: curvas críticas do tipo 2.
β_3 e β_4: curvas críticas do tipo 3.
β_5 e β_6: curvas críticas do tipo 5.

Observações:

- As curvas só levam em conta o contato com vértices e/ou eixos, independente do resto da região de obstáculos, portanto são um superconjunto do conjunto de curvas críticas. Algumas delas podem ser geradas por configurações de **A** que intersectam o interior de **B** (em outra componente convexa). Tais curvas são ditas <u>Redundantes</u> e podem ser removidas.
- O conjunto de curvas críticas é finito. Cada extremidade de uma curva crítica está localizada em outra curva crítica ou no infinito. Curvas críticas são curvas algébricas suaves, de 1º, 2º ou 4º grau, portanto possuem um número finito de interseções.
- Exceção: curvas coincidentes. Cada coincidência pode ser eliminada com deslocamentos arbitrariamente pequenos em **B**.

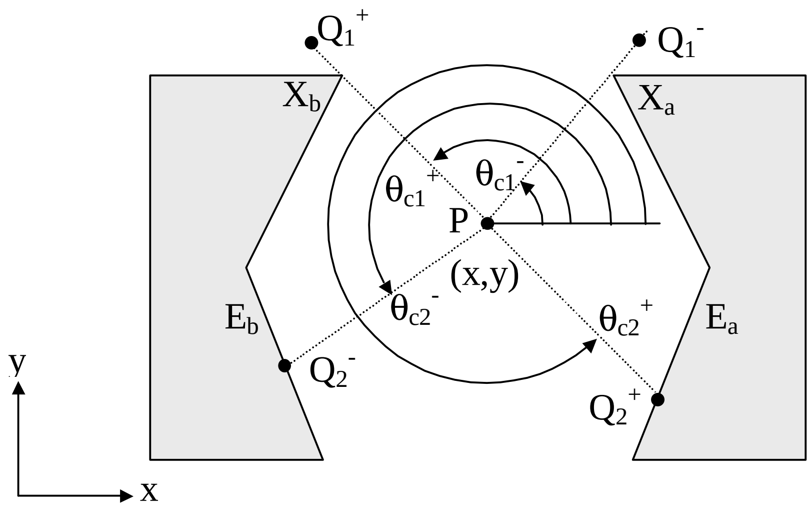
Exemplos de coincidências:



Decomposição do Espaço Livre:

- $F(x,y) = \{\theta \ / \ (x, y, \theta) \in C_L\}$ = conjunto de todas as orientações livres de **A** na posição não crítica (x, y). F(x,y) é a união de um número finito de intervalos conexos máximos abertos.
- Cada extremo de um intervalo conexo é chamado de Orientação Limite, θ_c, a qual pode ser Horária, θ_c^-, ou Anti-horária, θ_c^+. $\{\forall\theta \in (\theta_c^-, \theta_c^+) \subseteq F(x,y) \Rightarrow \theta_c^- \leq \theta \leq \theta_c^+\}$.
- O vértice ou eixo da região de obstáculos **B** que é tocado por $\mathbf{A}(x,y,\theta_c^-)$, (respectivamente $\mathbf{A}(x,y,\theta_c^+)$), é denominado Parada Horária, $S^-(x,y,\theta_c^-)$, (resp. Parada Anti-horária, $S^+(x,y,\theta_c^+)$).
- O par $S^{\pm}(x,y,\theta_c^-,\theta_c^+) = [S^-(x,y,\theta_c^-), S^+(x,y,\theta_c^+)]$ é único, visto que a posição (x, y) é não crítica; caso contrário, seria crítica.
- $\sigma(x,y)$ é o conjunto de todos os pares de paradas $S^{\pm}(x,y,\theta_c^-,\theta_c^+)$, tais que $S^-(x,y,\theta_c^-)$, (respectivamente $S^-(x,y,\theta_c^-)$), é uma parada horária, (respectivamente anti-horária), em (x,y) e o intervalo $(\theta_c^-, \theta_c^+) \subseteq F(x,y)$. Se $F(x,y) = [0, 2\pi) \Rightarrow \sigma(x,y) = \{[\Omega, \Omega]\}$, onde Ω designa uma parada não existente. Por convenção, atribui-se $\theta_c^-(x,y,\Omega) = 0$ e $\theta_c^+(x,y,\Omega) = 2\pi$.

Exemplo:



$S_1^{\pm}(x,y,\theta_{c1}^-,\theta_{c1}^+) = [S_1^-(x,y,\theta_{c1}^-), S_1^+(x,y,\theta_{c1}^+)] = [X_a, X_b]$
$S_2^{\pm}(x,y,\theta_{c2}^-,\theta_{c2}^+) = [S_2^-(x,y,\theta_{c2}^-), S_2^+(x,y,\theta_{c2}^+)] = [E_b, E_a]$
$\sigma(x,y) = \{ S_1^{\pm}, S_2^{\pm}\} = \{[X_a, X_b], [E_b, E_a]\}$

- Região Não Crítica, R, é um subconjunto conexo aberto do plano xy, delimitado por curvas críticas, tal que,

  $R = \{(x,y) \in \mathbb{R}^2 \,/\, \forall (x,y), (x',y') \in R : \sigma(x,y) = \sigma(x',y') = \sigma(R)\}$

- Denomina-se Célula ao subconjunto conexo aberto de C_L:

  $cel(R, S^{\pm}) = \{(x,y,\theta) \,/\, (x,y) \in R \text{ e } \theta \in (\theta_c^-(x,y,S^-),\ \theta_c^+(x,y,S^+))\}$

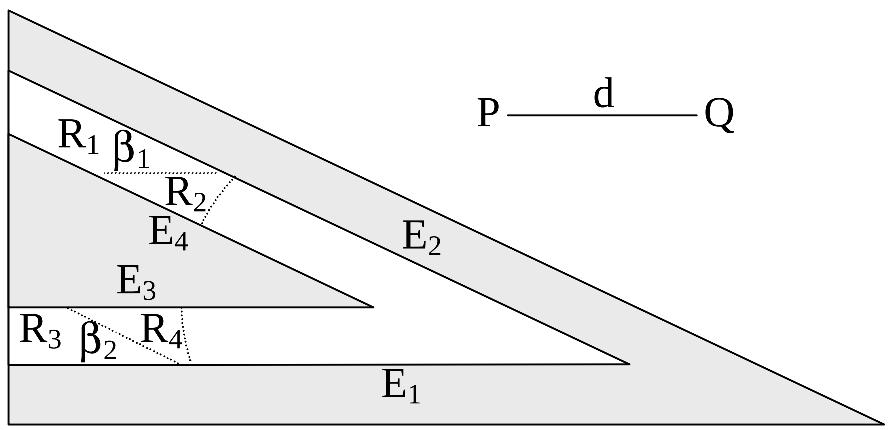
- Para toda célula $k = cel(R,S^-,S^+)$ e toda posição (x,y) em R:

  $(x,y,\theta) \in k \Leftrightarrow \theta \in \theta_c^{\pm}(x,y,S^{\pm}) = (\theta_c^-(x,y,S^-), \theta_c^+(x,y,S^+))$

  onde, $\theta_c^-(x,y,S^-)$ e $\theta_c^+(x,y,S^+)$ são funções contínuas $R \rightarrow \mathbb{R}$.

- O conjunto de todas as células deste tipo forma uma decomposição de C_L, tal que as células são disjuntas e a clausura da sua união é igual a $cl(C_L)$.
- As células pertencem a prismas que projetam-se em regiões não críticas do plano xy.
- Se duas regiões não críticas, R e R', compartilham uma seção β de curva crítica, tal que $\sigma(R) = \sigma(R')$, então β é redundante.

Exemplo:



$\Rightarrow \sigma(R_1)=\sigma(R_2)=\{[E_2,E_4],[E_4,E_2]\},\ \sigma(R_3)=\sigma(R_4)=\{[E_1,E_3],[E_3,E_1]\}$

Grafo de Conectividade:

Duas células $k = cel(R, S^{\pm})$ e $k' = cel(R', S'^{\pm})$ são adjacentes $\Leftrightarrow$

a) os limites ∂R e $\partial R'$ compartilham uma seção β de curva crítica;
b) $\forall(x,y) \in int(\beta) : \theta_c^{\pm}(x,y,S^{\pm}) \cap \theta_c^{'\pm}(x,y,S^{'\pm}) \neq \emptyset$.

- Se k e k' são adjacentes, qualquer configuração em k pode ser conectada a qualquer configuração em k' por um caminho livre cuja projeção no plano xy atravessa β transversalmente, com orientação constante na vizinhança do ponto de cruzamento.

- Se β é uma seção de curva crítica não redundante e não está contida em duas curvas críticas coincidentes, sempre que $int(\beta)$ é atravessado transversalmente:

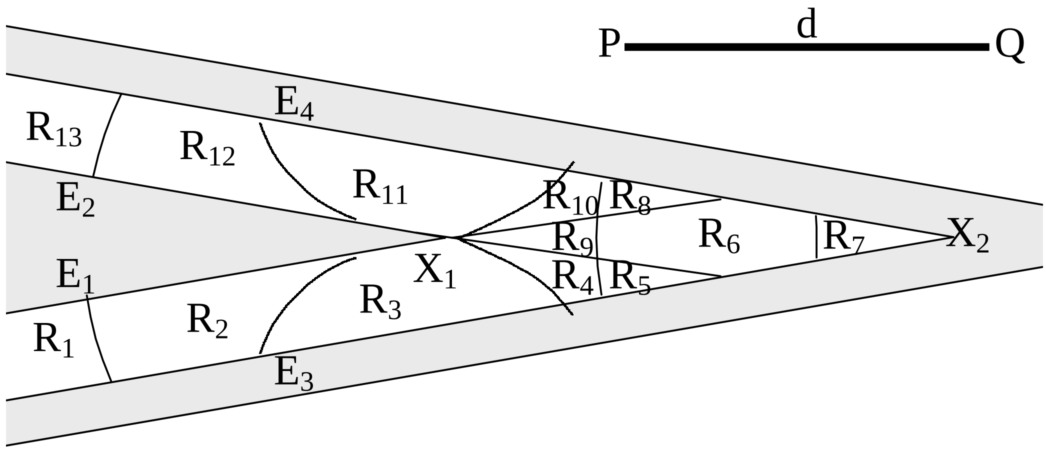
  a) um par em $\sigma(x,y)$ aparece ou desaparece.
  b) um elemento em um par de $\sigma(x,y)$ muda.

$\Rightarrow$ Regra de cruzamento: conectar $cel(R,S^{\pm})$ a $cel(R',S^{\pm})$ para cada $S^{\pm} \in \sigma(R) \cap \sigma(R')$ e conectar cada célula $cel(R, S^{\pm})$, com $S^{\pm} \in \sigma(R) \backslash \sigma(R')$, (se existir alguma), a cada célula $cel(R',S'^{\pm})$, com $S'^{\pm} \in \sigma(R') \backslash \sigma(R)$, (se existir alguma).

- Grafo de Conectividade, G, é o grafo não direcional cujos nós são todas as células $cel(R, S^{\pm})$, onde R é uma região não crítica e $S^{\pm} \in \sigma(R)$. Os nós de G são conexos por um arco se e somente se as células correspondentes forem adjacentes.

$\Rightarrow$ Dadas duas configurações em C_L, $q_{ini} = (x_{ini},y_{ini},\theta_{ini})$ e $q_{fin} = (x_{fin},y_{fin},\theta_{fin})$, tais que nem (x_{ini},y_{ini}) nem (x_{fin},y_{fin}) estão numa curva crítica, existe um caminho entre ambas se e somente se as células que as contêm são conexas por um caminho no grafo de conectividade G.

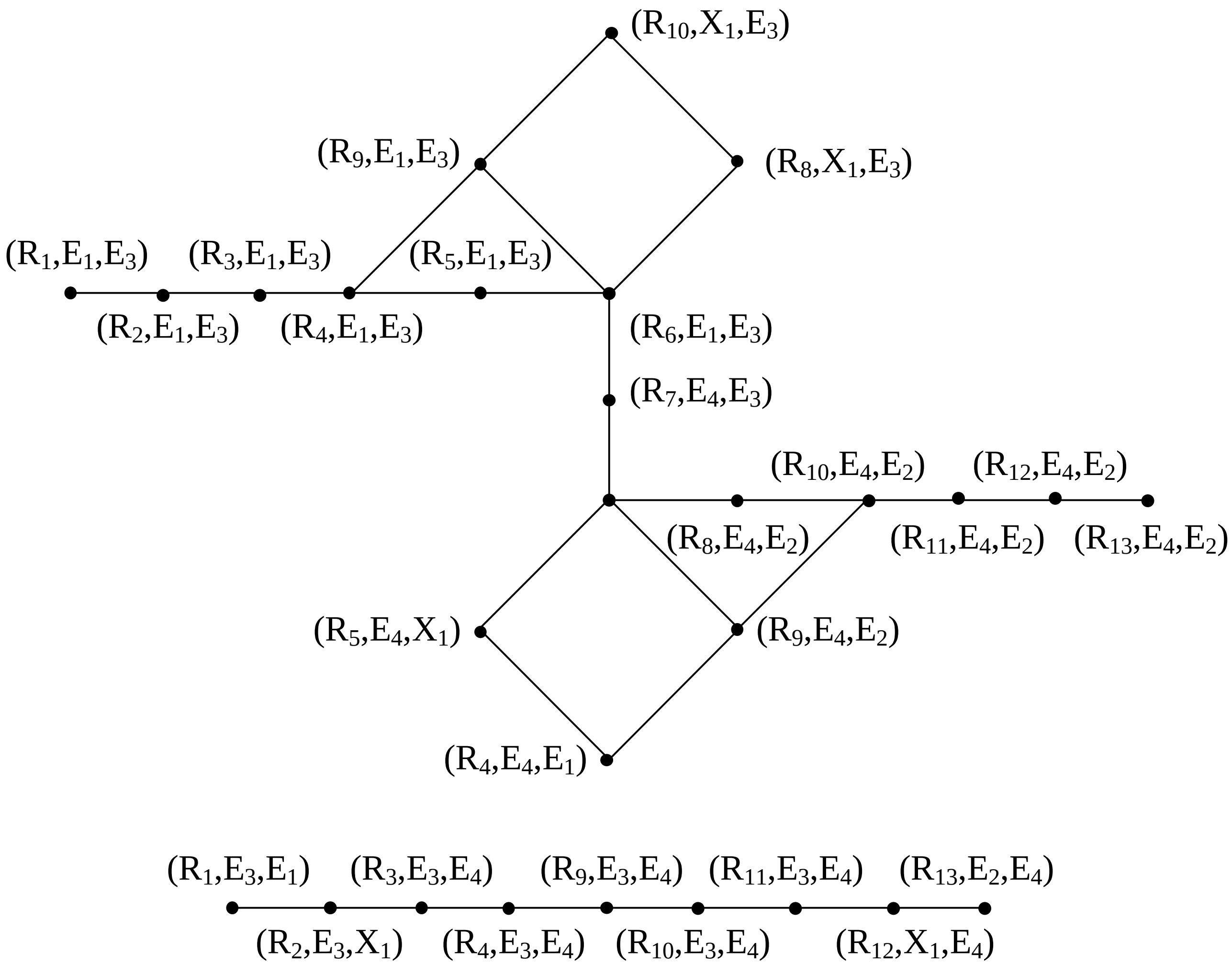
Exemplo:



$\sigma(R_1) = \{[E_1,E_3],[E_3,E_1]\}$
$\sigma(R_2) = \{[E_1,E_3],[E_3,X_1]\}$
$\sigma(R_3) = \{[E_1,E_3],[E_3,E_4]\}$
$\sigma(R_4) = \{[E_1,E_3],[E_3,E_4],[E_4,X_1]\}$
$\sigma(R_5) = \{[E_1,E_3],[E_4,X_1]\}$
$\sigma(R_6) = \{[E_1,E_3],[E_4,E_2]\}$
$\sigma(R_7) = \{[E_4,E_3]\}$
$\sigma(R_8) = \{[E_4,E_2],[X_1,E_3]\}$
$\sigma(R_9) = \{[E_4,E_2],[E_1,E_3],[E_3,E_4]\}$
$\sigma(R_{10}) = \{[E_4,E_2],[X_1,E_3],[E_3,E_4]\}$
$\sigma(R_{11}) = \{[E_4,E_2],[E_3,E_4]\}$
$\sigma(R_{12}) = \{[E_4,E_2],[X_1,E_4]\}$
$\sigma(R_{13}) = \{[E_4,E_2],[E_2,E_4]\}$

- Observação: o robô pode percorrer os dois corredores, mas não pode executar uma rotação completa dentro dos mesmos. Assim, a orientação com que o robô entra num corredor determina a orientação com a qual sairá no outro corredor.

Grafo de conectividade resultante:

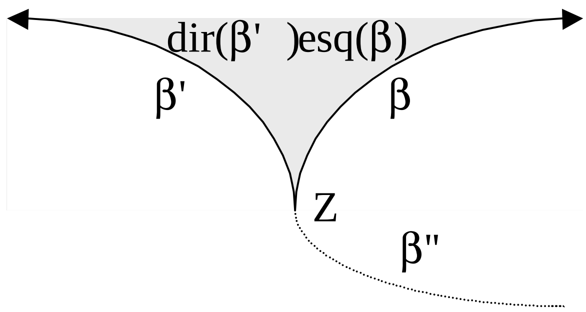


- Observação: o grafo de conectividade possui duas componentes conexas, que correspondem a dois subconjuntos conexos que compõem C_L. A primeira componente corresponde à orientação de **A** para a qual P está mais próximo de X_2, (entrando de ré e saindo de frente). A segunda componente corresponde à situação inversa, (entrando de frente e saindo de ré).

Refinamento do Algoritmo:

- Não é necessário construir uma representação explícita das regiões não críticas.
- Para cada seção de curva crítica β, definir uma orientação e criar duas regiões: $dir(\beta)$ e $esq(\beta)$, (à direita e à esquerda de β).

$\Rightarrow$ Cada célula aparece repetida no grafo de conectividade tantas vezes quantas forem as seções de curvas que constituem o limite da região não crítica na qual a célula é projetada.



- Dois nós, $cel(R,S^{\pm})$ e $cel(R',S'^{\pm})$, são conexos por um arco do grafo de conectividade G se (a) representam a mesma célula, ou (b) são adjacentes (de acordo com a regra de cruzamento):

a) Dadas as regiões $R = esq(\beta)$ e $R' = dir(\beta')$, com β e β' partindo de uma mesma extremidade Z. Se não existir nenhuma outra seção de curva crítica, β'' partindo de Z entre β e β', então R e R' são a mesma região não crítica $\Rightarrow$ $k = cel(R,S^{\pm})$ e $k' = cel(R',S'^{\pm})$ são conexas por um arco em G se e somente se $S^{\pm} = S'^{\pm}$, (ou seja, k e k' são a mesma célula).

b) Dadas as regiões $R = dir(\beta)$ e $R' = esq(\beta')$, as células $k = cel(R,S^{\pm})$ e $k' = cel(R',S'^{\pm})$ são conexas por um arco em G se e somente se são adjacentes de acordo com a regra de cruzamento.

Detalhamento do Algoritmo:

1. **Computar as seções de curva crítica e suas interseções**:
   - Para cada curva, ordenar os seus pontos de interseção de acordo com sua ordem na curva (usando uma parametrização adequada).
   - Decompor cada curva em seções limitadas por pontos de interseção consecutivos.

2. **Computar a relação de adjacência entre células**. Para cada curva crítica β que não é parte de um eixo de obstáculo:
   - Computar $\sigma(dir(\beta))$ e $\sigma(esq(\beta))$: tomar um ponto X em $int(\beta)$ e cortar β em X, perpendicularmente, por uma reta L. Calcular as interseções de L com as curvas críticas. Sobre L, escolher um ponto X' (resp. X") em $dir(\beta)$ (resp. $esq(\beta)$). Computar σ em X' e X".
   - Usando a regra de cruzamento, construir a relação de adjacência entre as células acima de $dir(\beta)$ e $esq(\beta)$.
   - Se $\sigma(dir(\beta)) = \sigma(esq(\beta))$, β é redundante. (Desconsiderar).
   - Para curvas do tipo 0, não há relação de adjacência.

3. **Formar agrupamentos de seções de curva adjacentes**:
   - Para cada ponto de interseção Z, formar uma lista circular de todas as curvas críticas incidentes em Z.
   - Ordenar esta lista em ordem anti-horária, de acordo com a direção do vetor tangente à curva em Z.
   - Quando duas destas direções forem iguais, computar derivadas de ordem superior.

4. **Identificar as células inicial e final**:

- Achar uma seção de curva β_{ini} (resp. β_{fin}) tal que $(x_{ini},y_{ini}) \in dir(\beta_{ini})$ ou $esq(\beta_{ini})$ (resp. $(x_{fin},y_{fin}) \in dir(\beta_{fin})$ ou $esq(\beta_{fin})$).
    - Traçar um segmento de reta entre (x_{ini},y_{ini}) e (x_{fin},y_{fin}) e computar suas interseções com todas as curvas críticas. (caso não exista interseção, significa que estão na mesma região não crítica).
    - Achar β_{ini} e β_{fin} cujas interseções com o segmento são as mais próximas a (x_{ini},y_{ini}) e (x_{fin},y_{fin}), respectivamente.
    - Identificar o lado de β_{ini} (resp. β_{fin}) que contém (x_{ini},y_{ini}) (resp. (x_{fin},y_{fin})).
    - Achar o par $S_{ini}^{\pm}$ (resp. $S_{fin}^{\pm}$) tal que $\theta_{ini} \in \theta_c^{\pm}(x_{ini},y_{ini},S_{ini}^{\pm})$ (resp. $\theta_{fin} \in \theta_c^{\pm}(x_{fin},y_{fin},S_{fin}^{\pm})$).
- Fazer k_{ini} (resp. k_{fin}) igual à célula que contém $(x_{ini},y_{ini},\theta_{ini})$ (resp. $(x_{fin},y_{fin},\theta_{fin})$)

5. **Buscar um canal no grafo de conectividade**:
    - Inicializar G com k_{ini}.
    - Expandir iterativamente os nós que não foram expandidos.
    - Caso for gerado o nó k_{fin}, retornar a seqüência de células conectando k_{ini} a k_{fin} e parar.
    - Caso não existam mais nós para expandir e k_{fin} não tiver sido encontrado, reportar falha.

Observações:

- A saída do algoritmo é a seqüência de células adjacentes, (canal), $(k_{ini},...,k_i,...,k_{fin})$, com $k_i = dir(\beta_i)$ ou $esq(\beta_i)$.
- Todo caminho entre q_{ini} e q_{fin} contido no interior do canal é um caminho livre.
- Um caminho semi-livre pode ser construído a partir da seqüência $(\beta_{ini},...,\beta_i,...,\beta_{fin})$, o qual pode ser transformado em caminho livre através de deformações arbitrariamente pequenas.

PLANEJAMENTO DE CAMINHOS

MÉTODOS BASEADOS EM DECOMPOSIÇÃO EM CÉLULAS – DECOMPOSIÇÃO APROXIMADA

Princípio:

- Decomposição de C_L em regiões não superpostas denominadas células, de formato padrão simples, (por exemplo, retangulóides), cuja união é uma aproximação conservadora de C_L. $\Rightarrow$ Não é possível representar C_L de maneira exata.
- Construção de um grafo de conectividade G que representa as relações de adjacência entre as células.
- Busca de um Canal (seqüência de células adjacentes ligando a célula contendo q_{ini} à célula contendo q_{fin}) em G
- Extração de um caminho entre q_{ini} e q_{fin} a partir do canal.

Vantagens:

- Decomposição de C_L por processo iterativo simples.
- Relativamente insensível a computação aproximada numericamente.
- Implementação mais fácil do que a decomposição exata.
- Permite controlar a "quantidade de espaço livre" em torno de um caminho através da especificação do tamanho mínimo das células.

Desvantagens:

- As relações de adjacência entre células são arbitrárias, não caracterizando descontinuidades nas restrições de movimento, incorporando de maneira menos explícita os aspectos da estrutura matemática do problema de planejamento.
- Por ser baseado numa aproximação conservadora de C_L, em certos casos o método pode falhar na busca de um caminho livre, mesmo existindo um. $\Rightarrow$ O método é incompleto.

Considerações sobre o tamanho da célula:

- Tamanho de célula grande $\Rightarrow$ menor esforço computacional e maior probabilidade de falhar na busca de um caminho livre, caso exista um.
- Tamanho de célula pequeno $\Rightarrow$ maior esforço computacional e menor probabilidade de falhar na busca de um caminho livre, caso exista um.
- No limite, o método pode ser tornado completo fazendo o tamanho da célula tender a zero, a expensas de um maior esforço computacional, (o qual tende a infinito no limite).
- Solução de compromisso: Decomposição Hierárquica. Começar com uma decomposição de baixa resolução e, progressivamente, refinar a decomposição localmente, em torno dos obstáculos, até encontrar um caminho livre ou atingir uma resolução máxima especificada.

Aplicabilidade do método:

- Método geral, aplicável ao problema básico de planejamento e a várias das suas extensões.
- Na prática, a complexidade temporal e espacial do método cresce rapidamente com a dimensão m do espaço de configuração.
- Aplicável com vantagem em relação a outros métodos para espaços de configuração de dimensão baixa ($m \leq 5$).

Descrição Geral:

- **Retangulóide**, D, é a região fechada de $\Re^n$:

  $D = \{(x_1,...,x_n) \,/\, x_1 \in [x_1', x_1''], ..., x_n \in [x_n', x_n'']\}$

onde, $d_i = x_i'' - x_i'$ = dimensões de D, com $d_i \neq 0$.

Se o conjunto de possíveis posições de **A** está contido em um retangulóide $D \subset \Re^n$, o espaço de configuração livre é representado por: $C_L = R \backslash CB$, onde CB é a região de C-obstáculos e a região R é definida como:

$R = int(D)$ se $C = \Re^n$.
$R = int(D) \times [0,2\pi]$ se $C = \Re^2 \times S^1$.
$R = int(D) \times [0,2\pi] \times [0,\pi] \times [0,2\pi]$ se $C = \Re^3 \times SO(3)$.

- Uma Decomposição em Retangulóides, P, do retangulóide de dimensão m, $\Omega = cl(R)$, (onde m = dimensão de C), é a coleção finita de retangulóides $\{k_i\}$, com i = 1,...r, tal que:

  - Ω é igual à união dos retangulóides: $\Omega = \cup k_i$.
  - A interseção dos interiores dos k_i's é nula:
    $\forall i, j \in [1, r]$, com $i \neq j$: $int(k_i) \cap int(k_j) = \emptyset$.

- Os retangulóides k_i's, denominados células da decomposição P, são adjacentes se e somente se sua interseção é um conjunto de tamanho não nulo em $\Re^m$, levando em consideração que:

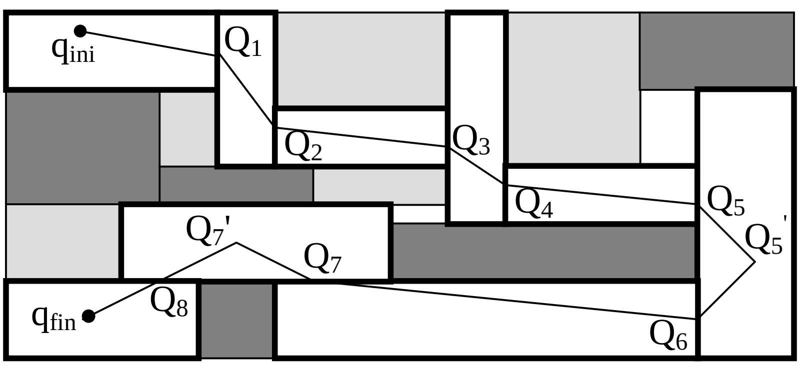
  - Se $C = \Re^2 \times S^1 \Rightarrow (x,y,2\pi) = (x,y,0)$
  - Se $C = \Re^3 \times SO(3) \Rightarrow$ $(x,y,z,2\pi,\theta,\psi) = (x,y,z,0,\theta,\psi)$
    $(x,y,z,\phi,\pi,\psi) = (x,y,z,\phi,0,2\pi-\psi)$
    $(x,y,z,\phi,\theta,2\pi) = (x,y,z,\phi,\theta,0)$

- Uma célula k_i é classificada como:
  - Vazia, (célula-V): $\Leftrightarrow k_i \cap CB = \emptyset$.
  - Cheia, (célula-C): $\Leftrightarrow k_i \subseteq CB$.
  - Mesclada, (célula-M): $\Leftrightarrow k_i \cap CB \neq \emptyset$ e $k_i \cap C_L \neq \emptyset$.

Grafo de Conectividade, G, associado à decomposição P de Ω é o grafo não direcional, tal que:

- Os nós de G são as células vazias e mescladas de P.
- Dois nós de G são conexos por um arco, se e somente se, as células correspondentes são adjacentes.

- Canal: seqüência de células-V e/ou células-M, k_i, com i=1,...,p, tal que quaisquer duas células consecutivas, k_i e k_{i+1} são adjacentes.
- Canal-V: canal que contém apenas células-V. $\Rightarrow$ Todo caminho entre k_1 e k_p em $int(\cup k_i)$ é um caminho livre.
- Canal-M: canal que contém ao menos uma célula-M. $\Rightarrow$ É possível, mas não garantida, a existência de um caminho no canal entre k_1 e k_p.
- O objetivo do método é achar um canal-V, tal que $q_{ini} \in k_1$ e $q_{fin} \in k_p$. Seja $\beta_i = \partial k_i \cap \partial k_{i+1}$, pode-se extrair um caminho livre entre q_{ini} e q_{fin} a partir do canal, ligando com um segmento de reta, para cada célula, um ponto $Q_i \in int(\beta_i)$ a um ponto $Q_{i+1} \in int(\beta_{i+1})$. Se Q_i e Q_{i+1} pertencem a subconjuntos distintos da mesma face da célula k_i, criar um ponto auxiliar intermediário Q_i' entre os mesmos.



Planejamento Hierárquico de Caminhos:

- Geração de um canal-V através da construção de decomposições sucessivas de Ω, P_i, com $i \in \rfloor$.
- P_i é obtida a partir de P_{i-1}, (com $P_0 = \Omega$), decompondo uma ou mais células-M.
- Busca de um canal entre k_{ini} (com $q_{ini} \in k_{ini}$) e k_{fin} (com $q_{fin} \in k_{fin}$) no grafo de conectividade G_i associado a P_i.

Algoritmo de Planejamento por Primeiro Corte:

1. Computar P_1 de Ω. Fazer $i = 1$.
2. Buscar, em G_i de P_i, um canal entre k_{ini} e k_{fin}.
   - Se um canal-V é encontrado, sucesso. Retornar o canal.
   - Se um canal-M, Π_i, é encontrado, ir para o passo 3.
   - Caso contrário, reportar falha.
3. Fazer $P_i \leftarrow P_{i+1}$.
   - Para cada célula-M, k, em Π_i, computar a decomposição P^k de k e fazer $P_{i+1} \leftarrow [P_{i+1} \backslash \{k\}] \cup P^k$.
   - $i \leftarrow i+1$ e voltar para o passo 2.

Observações:

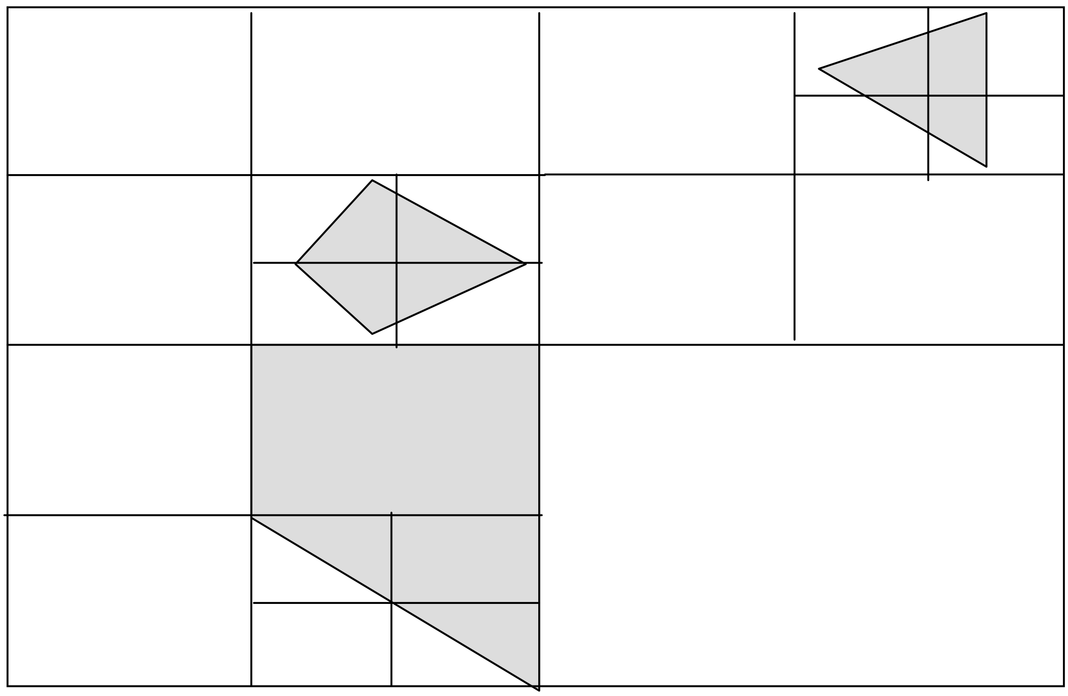
- A busca em G_i pode ser guiada por várias heurísticas, por exemplo: buscar um canal-V antes de um canal-M, buscar canais mais curtos, etc.
- Um canal-V existente em P_i continuará a existir em P_j, com $j>i$.
- O método pode ser tornado completo quando o tamanho das células-M tende a zero (e o tempo de cômputo tende a infinito). Pode-se limitar este tempo (às custas de tornar o método incompleto) colocando restrições no tamanho mínimo das células. Exemplo: impor que o volume de células-C e células-V seja maior que uma certa percentagem do volume de P^k, rotular como cheias as células-M menores do que um tamanho especificado, etc.

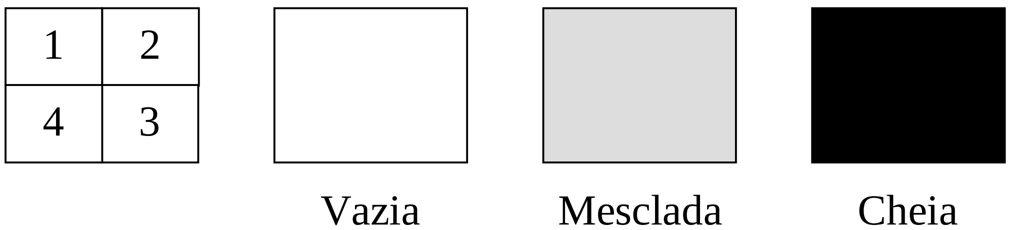
<u>Método de Decomposição por Divisão e Rotulagem</u>:

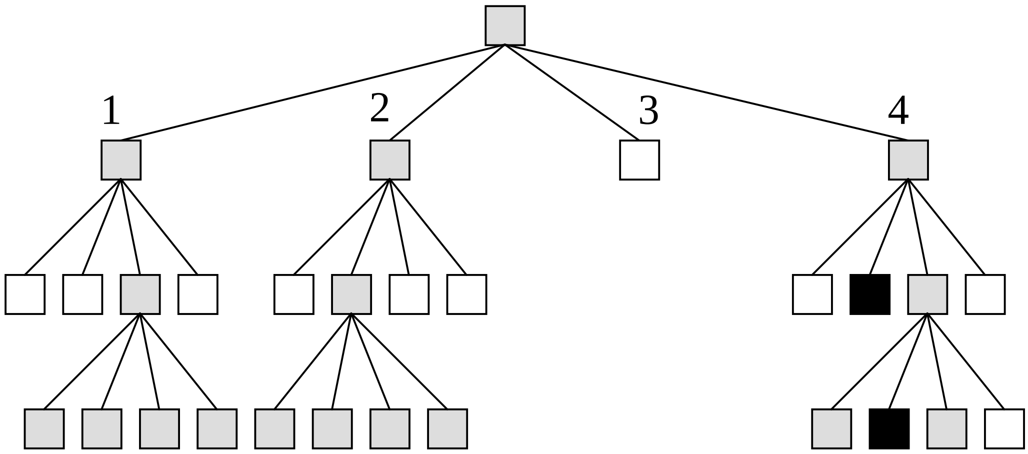
<u>Princípio</u>: Decompor uma célula mesclada dividindo-a em células menores. Rotular as células resultantes de acordo com a sua interseção com CB.

<u>Decomposição em Árvore-2^m</u>:

- Seja m = dim(W), a decomposição de Ω em árvore-2^m consiste em uma árvore na qual cada nó, (célula retangulóide vazia, cheia ou mesclada), se não for folha da árvore, é pai de exatamente 2^m nós filhos.
- A raiz da árvore é Ω.
- Somente os nós que são células mescladas podem ter 2^m filhos.
- Todos os nós filhos possuem o mesmo tamanho e são obtidos dividindo a célula mãe ao meio em todos os seus eixos.
- Se m = 2, a árvore é <u>quádrupla</u>. Se m = 3, a árvore é <u>óctupla</u>.
- A <u>altura</u> h_i de um nó, (número de arcos entre a raiz e o nó), correspondente a um nível de decomposição P_i, determina o tamanho da célula correspondente em relação a Ω.
- A altura h da árvore, (altura do nó mais alto) determina a resolução da decomposição (menor tamanho de célula).
- Na inicialização do algoritmo de decomposição, devem ser decompostas todas as células-M cuja altura for menor do que a altura h_1 da decomposição inicial P_1.
- Nos passos subseqüentes, somente as células-M localizadas em um canal-M são decompostas.
- Uma altura máxima h_{max} é especificada para limitar o processo iterativo.
- Cada célula-M cuja altura for igual a h_{max} deve ser re-rotulada como Cheia.
- O número de folhas cresce exponencialmente com m e h, (pior caso = $2^{m.h}$ folhas). Na prática, este número é significativamente menor, pois a árvore é podada em cada célula-V e em cada célula-C encontradas.




1
2
4
3
Vazia
Mesclada
Cheia


1
2
3
4

<u>Rotulagem em $\rfloor^2$</u>:

- $\Omega \subset \rfloor^2$, com robô e obstáculos poligonais, robô com orientação fixa.
- CB descrito por C-Sentença: $S_\Omega = \vee_i \wedge_j e_{ij}$.
- Cada C-restrição e_{ij} é da forma: $a_{ij}.x + b_{ij}.y + c_{ij} \leq 0$.
- Cada conjunção $\wedge_j e_{ij}$ corresponde a um C-obstáculo convexo.
- C-Sentença S_k associada a uma célula k da decomposição descreve a sua interseção com CB.
- S_k é derivada iterativamente a partir de S_Ω.
- Quando uma célula-M, k, é decomposta, S_k é utilizada para rotular as células resultantes (V, M, ou C).
- No processo de rotulagem de uma nova célula, a C-Sentença correspondente pode ser simplificada sempre que uma das C-restrições e_{ij} é satisfeita ou contradita em todos os pontos dela.
- A C-Sentença simplificada é associada à nova célula se esta for mesclada, reduzindo o esforço computacional.
- O número de C-restrições em S_k tende a diminuir consideravelmente com o tamanho da célula.

- Uma célula k está <u>Dentro</u> da C-restrição e_{ij} se todos os pontos $(x,y) \in k$ satisfazem e_{ij}. Teste $\Rightarrow$ k está dentro de e_{ij} se seus quatro vértices satisfazem: $a_{ij}.x + b_{ij}.y + c_{ij} \leq 0$.

- Uma célula k está <u>Fora</u> da C-restrição e_{ij} se todos os pontos $(x,y) \in k$ contradizem e_{ij}. Teste $\Rightarrow$ k está fora de e_{ij} se seus quatro vértices satisfazem: $a_{ij}.x + b_{ij}.y + c_{ij} \geq 0$.

- Uma célula k é <u>Cortada</u> pela C-restrição e_{ij} se não está dentro nem fora de e_{ij}.

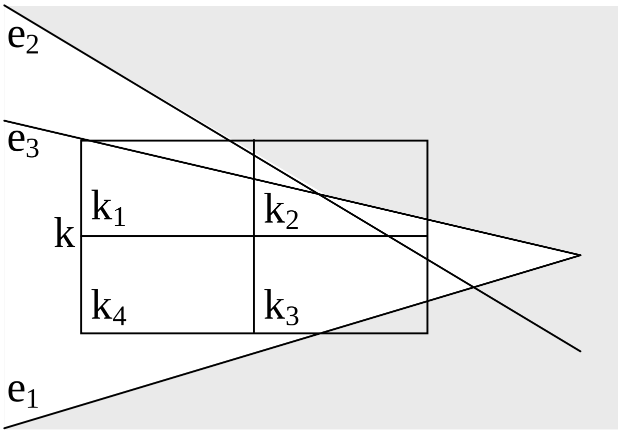
Algoritmo para rotulagem de uma nova célula k e simplificação da C-Sentença S associada a sua célula mãe:

Sejam:

> S = C-Sentença associada à célula-M, mãe da célula k.
> Λ = conjunção, $\Lambda \in S$.
> e = C-restrição, $e \in \Lambda$.

```
procedimento ROTULAGEM(k,S);
começar
    para cada conjunção Λ∈ S faça
    começar
        para cada C-restrição e ∈ Λ faça
        começar
            se k está dentro de e então Λ ← Λ\{e};
            se não, se k está fora de e então S←S\{Λ};
        fim;
        se Λ ∈ S e Λ = ∅ então
        começar
            rotular k como CHEIA;
            sair do procedimento;
        fim;
    fim;
    se S = ∅ então rotular k como VAZIA;
    se não
    começar
        rotular k como MESCLADA;
        associar S a k;
    fim;
fim;
```

Exemplo:



$S_k = e_1 \vee (e_2 \wedge e_3) \Rightarrow$ k = MESCLADA.

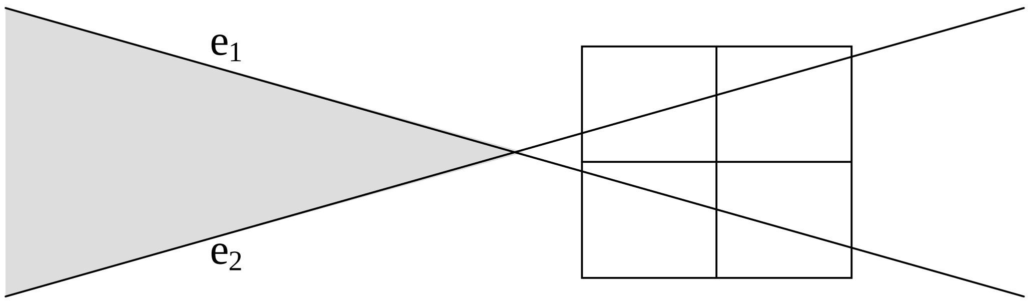
$S_{k1} = (e_2 \wedge e_3) \Rightarrow$ k_1 = MESCLADA.
$S_{k2} = (e_2 \wedge e_3) \Rightarrow$ k_2 = MESCLADA.
$S_{k3} = e_1 \Rightarrow$ k_3 = MESCLADA.
$S_{k4} = \emptyset \Rightarrow$ k_4 = VAZIA.

Observações:

- O procedimento de rotulagem pode ser utilizado para testar adjacência entre duas células, k_1 e k_2, (uma delas mesclada e a outra mesclada ou vazia), que são adjacentes se e somente se:
  - $\beta = \partial k_1 \cap \partial k_2$ é um segmento de comprimento não nulo.
  - β, considerado como célula degenerada unidimensional, não rotulada como cheia ao aplicar o procedimento de rotulagem com $S = S_{k1}$ (se k_1 é mesclada) e $S = S_{k2}$ (se k_2 é mesclada).
- O procedimento de rotulagem é aplicável a $\Omega = ⌡^3$, com robô e obstáculos poliédricos, robô com orientação fixa. Cada C-restrição e_{ij} numa C-Sentença é da forma: $a_{ij}.x+b_{ij}.y+c_{ij}.z+d_{ij} \leq 0$. Para determinar se uma célula retangulóide está dentro ou fora de e_{ij}, deve-se checar a C-restrição com os seus oito vértices.

- O procedimento é conservativo. Cada C-restrição é tratada independentemente das outras. Portanto, algumas células podem ser rotuladas como mescladas, mesmo não intersectando qualquer C-obstáculo. Apesar disso, esta rotulagem errônea é corrigida posteriormente, em decomposições mais refinadas destas células.

Exemplo:



Rotulagem em $\mathbb{R}^2 \times S^1$:

- $\Omega = R \times [0,2\pi]$
- CB descrito por coleção S_Ω de C-sentenças $\sigma = \vee\wedge e_{ij}$
- Cada C-restrição é da forma $e_{ij} = a_{ij}(\theta).x + b_{ij}(\theta).y + c_{ij}(\theta) \leq 0$.
- σ aplicável somente sobre certo intervalo fechado δ_σ das orientações θ de **A**. $\Rightarrow$ σ só descreve a região CB dentro de δ_σ.
- Os extremos de δ_σ são orientações críticas de **A**, nas quais há mudança na natureza dos possíveis contatos com os obstáculos.
- A rotulagem de uma célula $k = [x,x'] \times [y,y'] \times [\theta,\theta']$ é complicada devido a:
  a) A C-sentença pode variar ao longo de $[\theta,\theta']$.
  b) As C-sentenças são funções não lineares de θ.

Multiplicidade de C-Sentenças:

- Uma C-restrição e_{ij} pode não ser aplicável totalmente numa célula k $\Rightarrow$ não se pode afirmar que k está dentro ou fora de e_{ij}.

Solução:

- particionar $[\theta,\theta']$ em r sub-intervalos fechados máximos, $I_l=[\theta_l,\theta_l']$, com $l \in [1,r]$, tal que uma C-sentença simples S_l de S se aplica a cada I_l.
- Para cada $l \in [1,r]$, considerar a sub-célula k_l de k correspondente, tal que $k_l = [x,x'] \times [y,y'] \times I_l$.

- k_l está <u>Dentro</u> de uma C-restrição $\mathbf{e} \in S_l$ se esta é satisfeita para todos os pontos $(x,y,\theta) \in k_l$.
- k_l está <u>Fora</u> de uma C-restrição $\mathbf{e} \in S_l$ se esta é contradita para todos os pontos $(x,y,\theta) \in k_l$.
- Se k_l não está nem dentro nem fora de S_l, dizemos que ela é <u>Cortada</u> pela C-restrição **e**.

- Uma sub-célula k_l pode ser rotulada como VAZIA, CHEIA ou MESCLADA, aplicando o procedimento ROTULAGEM(k_l,S_l)

Uma célula k é rotulada como:

- VAZIA, se todas as suas sub-células k_l forem VAZIAS.
- CHEIA, se todas as suas sub-células k_l forem CHEIAS.
- MESCLADA, caso contrário.

- O procedimento ROTULAGEM(k_l,S_l) simplifica cada S_l resultando na C-sentença simplificada S_l' . Se k for rotulada como MESCLADA, a coleção de C-sentenças simplificadas correspondentes, $S_k = \{S_1', \ldots, S_r')$ é associada a k, com S_l' avaliada como VERDADEIRA (resp. FALSA) se k_l for rotulada como CHEIA (resp. VAZIA).

Não Linearidade das C-restrições:

- Considere uma sub-célula retangulóide $k_l = [x,x'] \times [y,y'] \times I_l$, onde $I_l=[\theta_l,\theta_l']$ está contido no domínio de aplicabilidade de uma C-restrição **e**:

$$\mathbf{e} = \{ a(\theta).x + b(\theta).y + c(\theta) \leq 0\}$$

- Considere a projeção no plano xy da porção da C-superfície $a(\theta).x + b(\theta).y + c(\theta) = 0$ contida em I_l:
  - O segmento ligando os pontos (x,y,θ_l) e (x,y,θ_l') está de um único lado da C-superfície se o ponto (x,y) está fora desta projeção.
  - O segmento ligando os pontos (x,y,θ_l) e (x,y,θ_l') atravessa a C-superfície se o ponto (x,y) está dentro desta projeção.

$\Rightarrow$ Pode-se rotular uma sub-célula k_l de k verificando a posição do retângulo $[x_k,x_k'] \times [y_k,y_k']$ em relação a esta projeção.

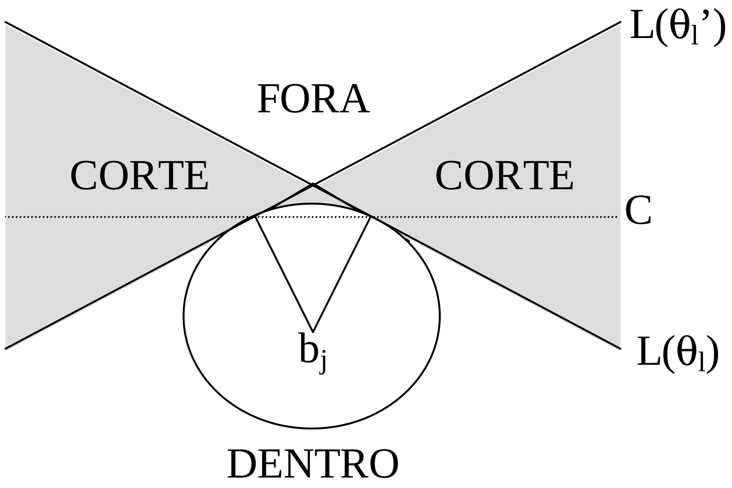
C-restrição de contato tipo A:

$\Rightarrow$ $-x.\cos(\phi_i+\theta)-y.\mathrm{sen}(\phi_i+\theta)+\|b_j\|.\cos(\phi_i+\theta-\beta_j)-\|a_i\|.\cos(\phi_i-\alpha_i) \leq 0$

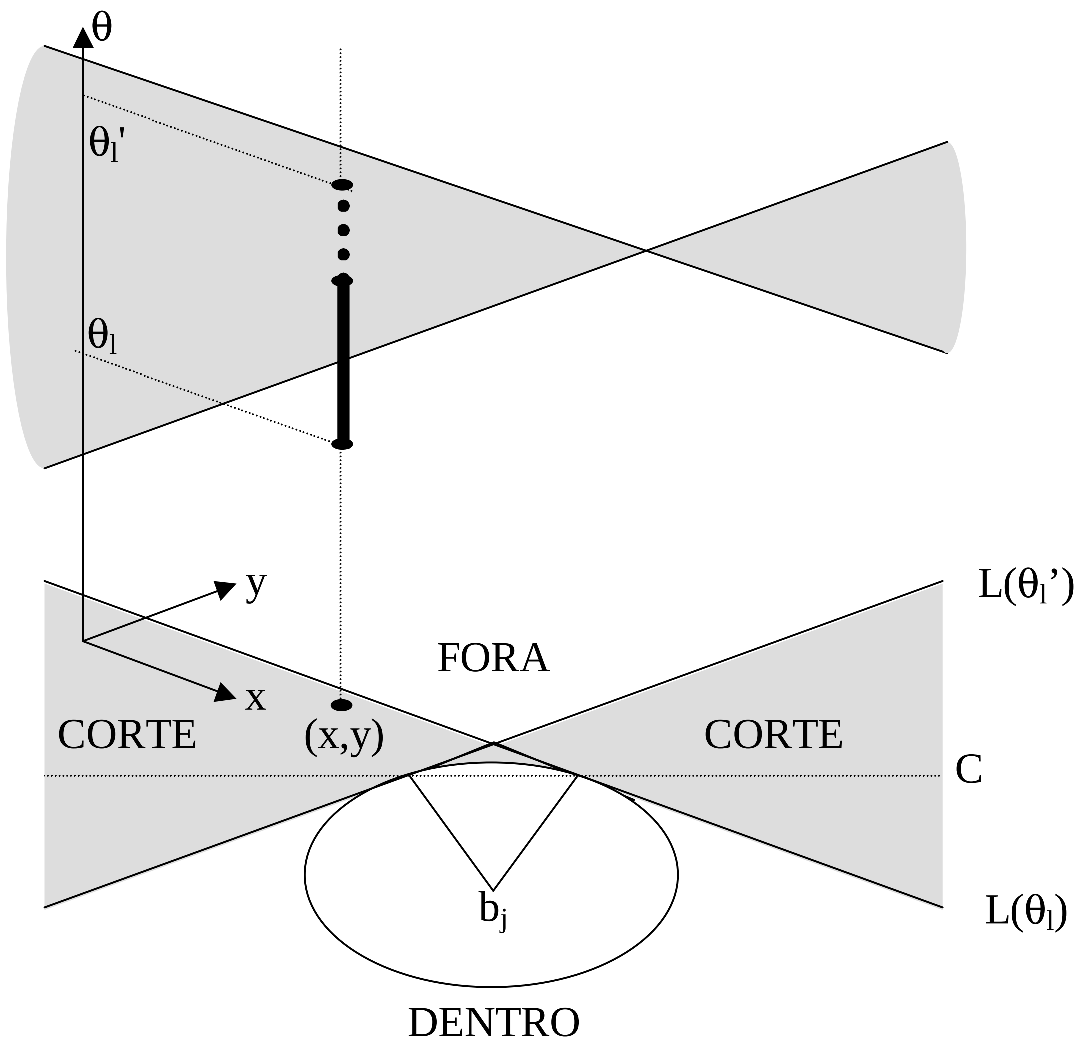
- Contato entre um eixo E_i^A entre os vértices a_i e a_{i+1} do robô e um vértice b_j de obstáculo.
- A interseção da C-superfície com o plano θ é a reta $L(\theta)$:

$$-x.\cos(\phi_i+\theta)-y.\mathrm{sen}(\phi_i+\theta)+\|b_j\|.\cos(\phi_i+\theta-\beta_j)-\|a_i\|.\cos(\phi_i-\alpha_i) = 0$$

- Para θ variando dentro de $[\theta_l,\theta_l']$, a reta $L(\theta)$ gira de modo correspondente em torno do vértice b_j.
- A projeção da fatia $[\theta_l,\theta_l']$ da C-superfície no plano xy corresponde à projeção de $L(\theta)$, com θ variando entre θ_l e θ_l'.
- As orientações limite θ_l e θ_l' definem as retas limites correspondentes $L(\theta_l)$ e $L(\theta_l')$.
- A projeção da fatia $[\theta_l,\theta_l']$ da C-superfície no plano xy é denominada região de CORTE da C-restrição, a qual separa as regiões FORA e DENTRO desta C-restrição.



$\Rightarrow$ Para todo ponto (x,y) FORA (resp. DENTRO) da região de CORTE, o segmento entre (x,y,θ_l) e (x,y,θ_l') está completamente fora (resp. dentro) de e_{ij}.


θ
θl'
θl
y
x
FORA
CORTE
(x,y)
CORTE
C
L(θl')
L(θl)
bj
DENTRO

Observações:

- O vértice b_j está na região de projeção DENTRO da C-restrição se $\cos(\phi_i-\alpha_i) > 0$.
- O vértice b_j está na região de projeção FORA da C-restrição se $\cos(\phi_i-\alpha_i) < 0 \Rightarrow$ robô não convexo.
- b_j está exatamente no ponto de interseção das retas $L(\theta_l)$ e $L(\theta_l')$ se $\cos(\phi_i-\alpha_i) = 0$.

- Cada um dos pontos do retângulo $[x_k,x_k']\times [y_k,y_k']$ estará FORA da região se estiver no lado de fora de $L(\theta_l)$ e $L(\theta_l')$.
- Cada um dos pontos do retângulo $[x_k,x_k']\times [y_k,y_k']$ estará DENTRO da região se estiver no lado de dentro de $L(\theta_l)$ e $L(\theta_l')$, mas não na pequena área entre as duas retas e o círculo.

- Pode-se checar se um ponto está nesta pequena área testando primeiro se a sua distância a b_j é maior do que o raio do círculo e, caso for, checar de que lado da corda C se encontra o ponto.

Teste de corte pela C-restrição e_{ij} para a sub-célula k_l:

- Se os quatro vértices do retângulo $[x_k,x_k']\times [y_k,y_k']$ estiverem na região FORA da C-restrição, a sub-célula k_l correspondente está FORA da C-restrição.
- Se os quatro vértices do retângulo $[x_k,x_k']\times [y_k,y_k']$ estiverem na região DENTRO da C-restrição, a sub-célula k_l correspondente está DENTRO da C-restrição.
- Caso contrário, a sub-célula k_l é CORTADA pela C-restrição.

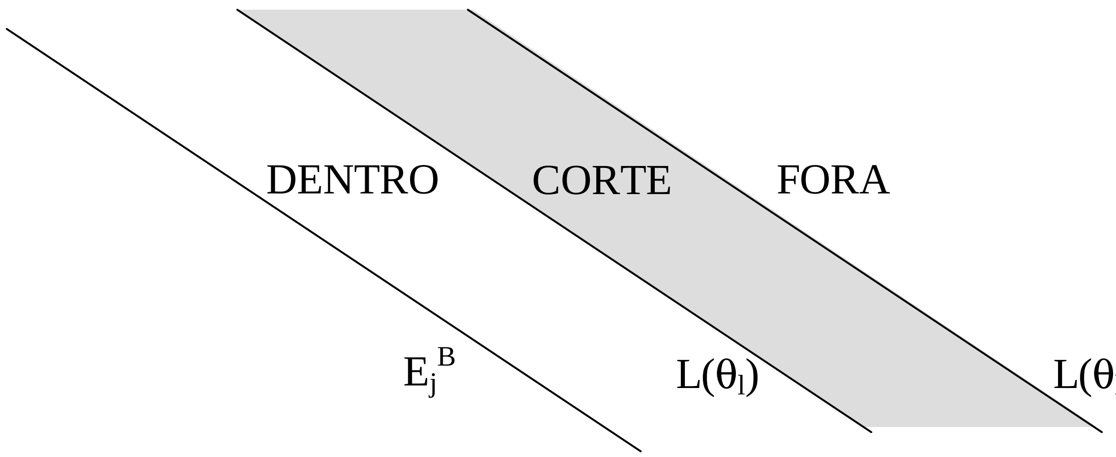
C-restrição de contato tipo B:

$\Rightarrow x.\cos(\xi_j) - y.\text{sen}(\xi_j) + \|a_i\|.\cos(\alpha_i+\theta-\xi_j) - \|b_j\|.\cos(\beta_j-\xi_j) \leq 0.$

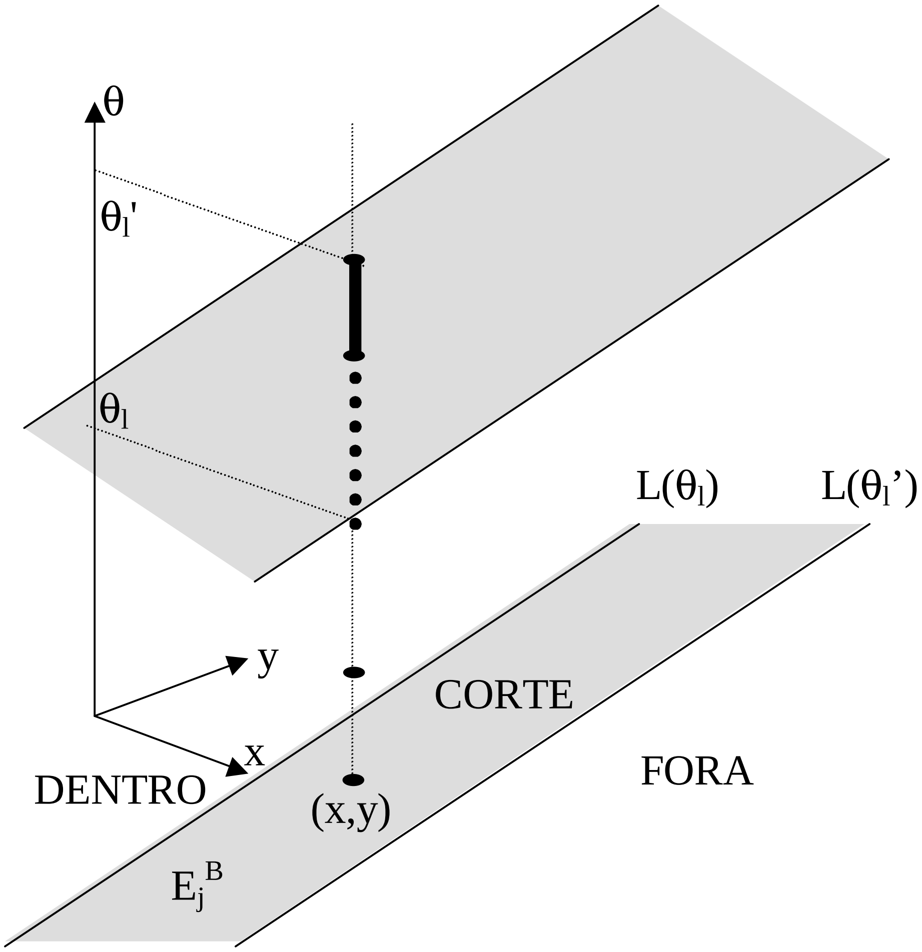
- Contato entre um eixo E_j^B entre os vértices b_j e b_{j+1} de obstáculo e um vértice a_i do robô.
- A interseção da C-superfície com o plano θ é a reta $L(\theta)$:

$$x.\cos(\xi_j) - y.\text{sen}(\xi_j) + \|a_i\|.\cos(\alpha_i+\theta-\xi_j) - \|b_j\|.\cos(\beta_j-\xi_j) = 0.$$

- Para θ variando dentro de $[\theta_l,\theta_l']$, a reta $L(\theta)$ sofre uma translação (sem rotação) paralela ao eixo de contato E_j^B.
- A projeção da fatia $[\theta_l,\theta_l']$ da C-superfície no plano xy corresponde à projeção de $L(\theta)$, com θ variando entre θ_l e θ_l'.
- As orientações limite θ_l e θ_l' definem as retas limites correspondentes $L(\theta_l)$ e $L(\theta_l')$.
- A projeção da fatia $[\theta_l,\theta_l']$ da C-superfície no plano xy é denominada região de CORTE da C-restrição, a qual separa as regiões FORA e DENTRO desta C-restrição.



$\Rightarrow$ Para todo ponto (x,y) FORA (resp. DENTRO) da região de CORTE, o segmento entre (x,y,θ_l) e (x,y,θ_l') está completamente fora (resp. dentro) de e_{ij}.


θ
θl'
θl
L(θl)
L(θl')
y
CORTE
x
FORA
DENTRO
(x,y)
Ej B

Observações:

- Cada um dos pontos do retângulo $[x_k,x_k']\times[y_k,y_k']$ estará FORA da região se estiver no lado de fora de $L(\theta_l)$ e $L(\theta_l')$.
- Cada um dos pontos do retângulo $[x_k,x_k']\times[y_k,y_k']$ estará DENTRO da região se estiver no lado de dentro de $L(\theta_l)$ e $L(\theta_l')$, mas não na pequena área entre as duas retas e o círculo.
- As equações das retas limites são:

  $$x.\cos(\xi_j) - y.\mathrm{sen}(\xi_j) + \|a_i\|.\cos(\alpha_i+\theta_a-\xi_j) - \|b_j\|.\cos(\beta_j-\xi_j) = 0.$$

  $$x.\cos(\xi_j) - y.\mathrm{sen}(\xi_j) + \|a_i\|.\cos(\alpha_i+\theta_b-\xi_j) - \|b_j\|.\cos(\beta_j-\xi_j) = 0.$$

  Onde θ_a e θ_b fornecem os valores extremos para $\cos(\alpha_i+\theta-\xi_j)$ sobre $\theta \in [\theta_l,\theta_l']$:

  - θ_a e $\theta_b \in \{\theta_l, \theta_l', (\xi_j-\alpha_i)\}$, se $(\xi_j-\alpha_i) \in [\theta_l,\theta_l']$.
  - θ_a e $\theta_b \in \{\theta_l, \theta_l'\}$, caso contrário.

Teste de corte pela C-restrição e_{ij} para a sub-célula k_l:

- Se os quatro vértices do retângulo $[x_k,x_k']\times[y_k,y_k']$ estiverem na região FORA da C-restrição, a sub-célula k_l correspondente está FORA da C-restrição.
- Se os quatro vértices do retângulo $[x_k,x_k']\times[y_k,y_k']$ estiverem na região DENTRO da C-restrição, a sub-célula k_l correspondente está DENTRO da C-restrição.
- Caso contrário, a sub-célula k_l é CORTADA pela C-restrição.

Busca Hierárquica no Grafo de Conectividade:

Princípio: Utilizar a busca feita no grafo G_{i-1} associado à decomposição P_{i-1} para ajudar a busca em G_i de P_i.

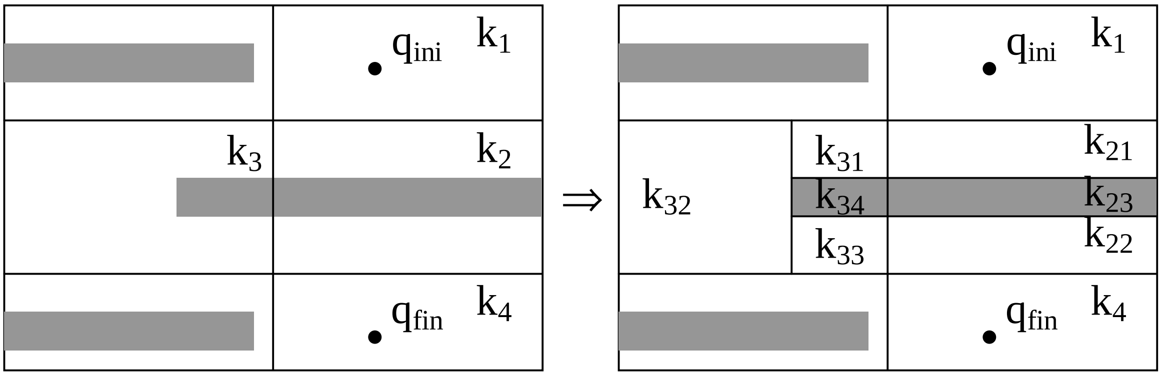
Algoritmo de busca hierárquico:

- Construir uma hierarquia de grafos de conectividade (GC's).
- O GC raiz, G^{Ω}, corresponde ao grafo G_1 da primeira decomposição, P_1, de Ω. Qualquer outro GC na hierarquia, G^k, corresponde à decomposição de uma célula-M k.
- Buscar um canal Π_1 entre k_{ini} e k_{fin}. Se Π_1 = canal-V, sucesso, retornar Π_1. Se nenhum canal é encontrado, reportar falha.
- Se Π_1 é um canal-M, cada célula-M, $k \in \Pi_1$, é decomposta recursivamente, buscando um sub-canal Π^k que conecte apropriadamente as duas partes de Π_1 divididas por k.
- Sucessivos canais completos Π_i (i = 2, 3, ...) conectando k_{ini} a k_{fin} são construídos iterativamente procurando por um canal-V.
- Π_i é construído a partir de Π_{i-1}, partindo de k_{ini} e analisando seqüencialmente cada célula $k \in \Pi_{i-1}$:
    - Se k é vazia, k é incluída no extremo do canal Π_i corrente.
    - Se k é mesclada, k é decomposta em um conjunto de células menores P^k, cujo GC associado é G^k, no qual busca-se um sub-canal Π^k que satisfaça:
        a) Se k é a primeira célula em Π_{i-1}, ($k = k_{ini}$), então, a primeira célula de Π^k deve conter também conter q_{ini}.
        b) Se k é a última célula em Π_{i-1}, ($k = k_{fin}$), então, a última célula de Π^k deve conter também conter q_{fin}.
        c) Se k não é a primeira célula em Π_{i-1}, a primeira célula de Π^k deve ser adjacente à última célula do Π_i corrente.
        d) Se k não é a última célula em Π_{i-1}, a última célula de Π^k deve ser adjacente à célula seguinte a k em Π_{i-1}.
- Se Π^k é obtido com sucesso, Π^k é anexado ao final de Π_i.
- Se todas as células de Π_{i-1} foram tratadas com sucesso, Π_i é um canal-V, (sucesso), ou um canal-M, (refinar Π_i em Π_{i+1}).

Ocorrência de Células em um Canal:

- Eventualmente, um canal-V pode ser gerado refinando um canal-M, sem precisar reconsiderar um GC global. ⇒ É importante considerar múltiplas ocorrência de uma mesma célula-M em um canal-M.
- Diferentes ocorrências podem levar a diferentes sub-canais.

Exemplo:



- Canal-M $(k_1,k_2,k_3,k_2,k_4) \Rightarrow$ Canal-V $(k_1,k_{21},k_{31},k_{32},k_{33},k_{22},k_4)$.
- O canal-V não poderia ser encontrado caso a célula k_2 não tivessem sido consideradas duas ocorrências da célula k_2.

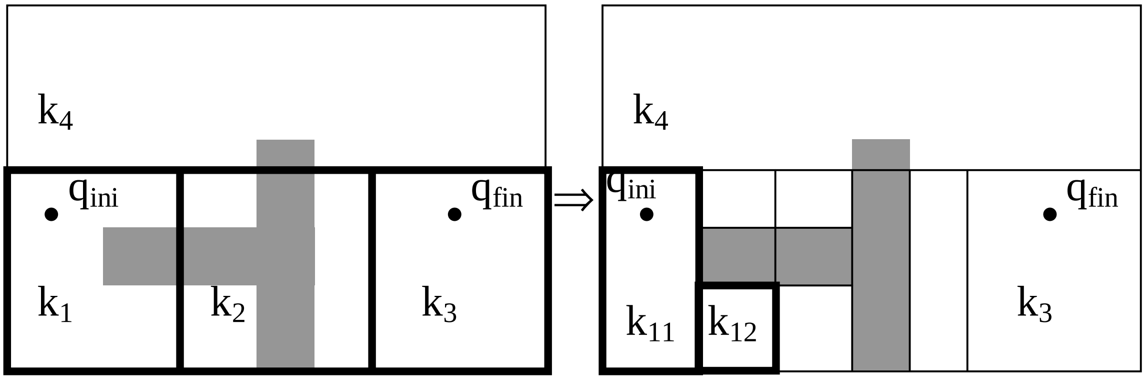
⇒ Elementos de um canal Π_i = Ocorrência de célula.

- w_i^j = j-ésima ocorrência de célula no canal Π_i.
- cel(w) = célula correspondente à ocorrência w.
- A ocorrência w é VAZIA (resp. MESCLADA) se e somente se cel(w) é VAZIA (resp. MESCLADA).
- Π^w = sub-canal resultante do refinamento da ocorrência w, construído na pesquisa de G^k associado a k = cel(w).
- Se o canal-M Π_i contém múltiplas ocorrências da célula-M k, uma única decomposição P^k é realizada para construir Π_{i+1}, mas cada diferente ocorrência de k pode levar a um sub-canal diferente.

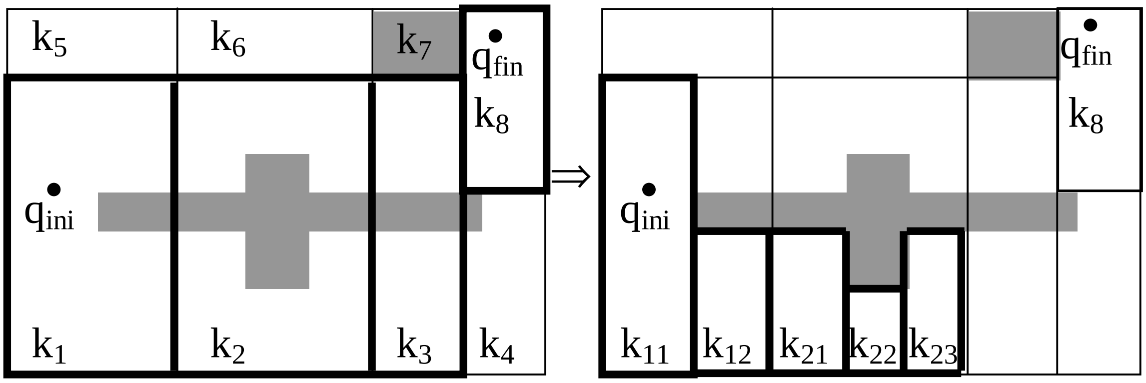
Falhas:

- planejador pode falhar na construção do canal Π_{i+1} ao tentar refinar a ocorrência-M $w_i^j \subset \Pi_i$ em um sub-canal se:
  - Π_i não contém um canal-V.
  - o canal parcial Π_{i+1} corrente gerado é um beco sem saída.

Exemplos:



- Dado o canal-M $\Pi_1 = (k_1,k_2,k_3)$, o sub-canal-V $\Pi^{k1} = (k_{11},k_{12})$ é obtido a partir da decomposição de k_1, mas não pode ser conectado a k_3 através de k_2 pois o C-obstáculo obstrui a passagem.



- Dado o canal-M $\Pi_1 = (k_1,k_2,k_3,k_8)$, o sub-canal-V $\Pi^{k1} = (k_{11},k_{12},k_{21},k_{22},k_{23})$ obtido a partir da decomposição de k_1 e k_2, mas não pode ser conectado a k_8 através de k_3 pois se chegou a um beco sem saída.

<u>Recuperação de condição de falha</u>:

- O planejador não tem meios de determinar diretamente a causa de uma falha ao tentar refinar a ocorrência-M w_i^j. Três casos mutuamente exclusivos são considerados:

- **Caso (1): j = 1**

  - $w_i^j = w_i^1$ ($1^{\underline{a}}$ ocorrência em Π_i). Não há meios de conectar q_{ini} à $2^{\underline{a}}$ ocorrência em Π_i.
  - Deve-se gerar um novo Π_i. Caso não seja possível e i>1, tenta-se recursivamente gerar um novo canal Π_{i-1}. Caso nenhum novo canal possa ser gerado e i=1, reportar falha.

- **Caso (2): j > 1 e $\mathbf{w_i^{j-1}}$ VAZIA**

  - Não há meios de conectar qualquer configuração em w_i^{j-1} a uma configuração em w_i^{j+1} através de w_i^j.
  - Proceder como no caso (1).

- **Caso (3): j > 1 e $\mathbf{w_i^{j-1}}$ MESCLADA**

  - Não há meios de conectar qualquer configuração na última ocorrência do Π_{i+1} corrente (a qual é subconjunto de w_i^{j-1}) a uma configuração em w_i^{j+1} através de w_i^j.
  - Tentar gerar outro sub-canal no GC de cel(w_i^{j-1}).
  - Caso um novo sub-canal for encontrado, substituir o sub-canal prévio no Π_{i+1} corrente e tentar refinar w_i^j em um sub-canal no GC de cel(w_i^j).
  - Caso contrário, tratar iterativamente w_i^{j-1} como foi tratada w_i^j. Fazer j = j-1 e aplicar os casos (1), (2) ou (3) de acordo com a situação.

Gravação das condições de falha:

- Como o mesmo GC pode ser pesquisado muitas vezes na busca de canais e sub-canais, se as falhas forem anotadas, o algoritmo pode ser mais eficiente, pois erros não serão repetidos.
- Uma anotação é feita sempre que o planejador se recuperar de uma falha ao buscar um sub-canal numa ocorrência-M.
- Para evitar um grande custo adicional devido a anotações de células excessivas, restringi-las aos casos (1) e (2), quando não precedidas de uma iteração de caso (3).

- **Caso (1)**:

Sejam: ξ = cel(w_i^1) e ψ = cel(w_i^2) as duas 1as células em Π_i. (ψ = λ = célula inexistente, se o número de ocorrências em Π_i, $n_i = 1$).

- A célula ξ é anotada com (ψ).
- Se um novo Π_i for gerado e se o mesmo contiver uma ocorrência w_ξ de ξ na posição inicial, então uma ocorrência w_ψ de ψ não será considerada como sucessora válida de w_ξ.
- Observação: a anotação só é válida para as ocorrências w_ξ no início do novo Π_i.

- **Caso (1)**:

- Sejam: ξ = cel(w_i^j), ($j>1$), φ = cel(w_i^{j-1}) e ψ = cel(w_i^{i+1}) no Π_i corrente. (se $j = n_i$, então $\psi = \lambda$).
- A célula ξ é anotada com (φ,ψ).
- Se um novo Π_i é gerado e este contiver w_ξ precedida pela ocorrência w_φ de φ, então uma ocorrência w_ψ de ψ não será considerada como sucessora válida de w_ξ.
- Observação: quando $\psi \neq \lambda$, a anotação (φ,ψ) é comutativa, ou seja aplicável ao tentar conectar w_φ a w_ψ através de w_ξ em qualquer sentido. (o par (φ,ψ) deve ser invertido se o sentido for invertido).

Busca no grafo de conectividade:

- Seja $\Pi_0 = (w_0^1)$ o canal-M associado a $cel(w_0^1) = \Omega$.
- Considere o refinamento de uma ocorrência-M, $w_i^j \in \Pi_i$, de uma célula $k = cel(w_i^j)$.
- Se k já foi decomposta previamente, reutilizar o seu GC, G^k.
- Caso contrário, decompor k em P^k, construir G^k e buscar nele um sub-canal a ser anexado ao Π_{i+1} corrente.
- A busca em G^k requer que se determinem as possíveis células inicial e final:
  - Se $j = 1$, w_i^j é a 1ª célula de Π_i. A única célula inicial possível é aquela que contém q_{ini}.
  - Caso contrário, as possíveis células iniciais de G^k são todas as células de P^k adjacentes à célula φ da última ocorrência w_φ no canal Π_{i+1} corrente tais que sua inserção em Π_{i+1} como sucessora de w_φ não viole qualquer anotação.

  - Se $j = n_i$, w_i^j é a última ocorrência de Π_i. A única célula final possível em G^k de P^k é aquela que contém q_{fin}.
  - Caso contrário, as células finais possíveis são todas as células de P^k que são adjacentes a $cel(w_i^{j+1})$.

- Se não houver possíveis células iniciais e finais em G^k, considerar como condição de falha no refinamento de w_i^j e aplicar os procedimentos de recuperação de falhas.
- Se ao menos uma célula inicial e uma célula final são determinadas em G^k, buscar um sub-canal em w_i^j conectando qualquer das possíveis células iniciais a qualquer das possíveis células finais, de modo tal que nenhuma anotação seja violada.

- Deve-se permitir que o sub-canal a ser gerado em G^k possa conter repetições de ocorrências de células-M em P^k.
- Para limitar o infinito Nº de potenciais sub-canais, pode-se limitar o Nº de ocorrências permitidas para uma mesma célula.