ROBÔS DE PARAQUEDAS: ALGORITMO RANDOMIZADO PARA RENDEZVOUS SIMÉTRICO UNIDIMENSIONAL COM MARCADORES

RESUMO

Dois agentes, partindo de pontos distintos e desconhecendo a localização um do outro, precisam se encontrar. Este cenário geral é conhecido na literatura como o *Problema do Rendezvous de Robôs*. Neste artigo estudamos a variante simétrica do problema, em que os dois agentes devem executar exatamente o mesmo algoritmo. O espaço considerado é unidimensional, e as posições iniciais dos agentes recebem marcadores que podem ajudá-los a se orientar durante a busca. Após revisitarmos a melhor solução conhecida para o problema, propomos uma estratégia de busca randomizada, que apresenta desempenho superior. Analisamos também a variante em que os agentes precisam retornar a seus pontos de origem, e para a qual o ganho da estratégia randomizada é ainda mais acentuado.

PALAVRAS CHAVE. Rendezvous de robôs. Algoritmos randomizados.

ABSTRACT

Two agents, starting from distinct points and unaware of the position of one another, must meet. This general setting is known in the literature as the *Robot Rendezvous Problem*. In this paper, we study the symmetric variant of the problem, whereby both agents must execute exactly the same algorithm. The space we consider is one-dimensional, and the starting points of the agents receive markers that may help them during their search. After revisiting the best known solution for the problem, we propose a randomized algorithm that presents a better performance. Furthermore, we analyse the variant in which the agents are required to return to their starting points, where the gain obtained by the randomized strategy is even more pronounced.

KEYWORDS. Robot rendezvous. Randomized algorithms.

1. Introdução

Dois robôs idênticos caem de paraquedas em pontos aleatórios de uma linha infinita e precisam se encontrar. Eles devem abandonar seus paraquedas nos locais onde pousaram e então se reunirem, mas desconhecem sua localização ou distância relativa. A cada ciclo de um relógio universal, cada robô pode andar um passo para a direita, andar um passo para a esquerda ou ficar parado. Os robôs podem ser programados em qualquer linguagem de programação, mas ambos devem executar exatamente o mesmo algoritmo. Qual seria o algoritmo mais eficiente para resolver o problema? Este cenário é um *puzzle* conhecido, e uma elegante solução determinística é tradicionalmente apresentada como sendo sua melhor resposta (Peter [2016]).

O problema acima se insere no contexto mais amplo dos problemas de *rendezvous*, em que dois ou mais agentes partindo de diferentes pontos, em algum ambiente, buscam se encontrar. Problemas dessa classe despertam interesse pois abrangem numerosas aplicações envolvendo situações reais, como duas pessoas ou robôs autônomos procurando um pelo outro em alguma região, e também situações virtuais, como agentes de software explorando uma rede, ou personagens de jogos explorando um cenário (Pelc [2012]).

Existem duas variantes principais para os problemas de *rendezvous*. A versão assimétrica é aquela em que os agentes podem executar estratégias independentes para se encontrar, como por exemplo aquela em que um deles permanece parado enquanto o outro o busca pelo ambiente. Na variante simétrica, os agentes devem executar a mesma estratégia de busca, como é o caso dos robôs paraquedistas (Beveridge et al. [2011]).

São também diversos os tipos de ambiente considerados na literatura dos problemas de *rendezvous*: regiões geométricas como linhas, planos ou esferas, ou mesmo redes de comunicações modeladas como grafos já foram amplamente estudadas, tanto em versões simétricas como assimétricas (Farrugia et al. [2015]). Para o problema de *rendezvous* sobre uma linha, são também conhecidos resultados tanto na versão simétrica (Alpern [1995], Anderson e Essegaier [1995]) quanto na assimétrica (Baston e Gal [1998], Gal [1999]). Esse tipo de cenário pode ser útil em aplicações robóticas, onde a linha pode ser usada para modelar ferrovias, cabos ou longos corredores (Beveridge et al. [2011]).

Neste artigo, consideramos uma versão da variante simétrica do *Problema do Rendezvous Unidimensional* em que os agentes deixam marcas na posição inicial (de forma lúdica, os paraquedas dos robôs). Neste artigo, analisamos a solução determinística tradicional para o problema e, a seguir, propomos uma solução randomizada cujo valor esperado para o tempo de encontro dos agentes é menor do que o da solução determinística na maioria dos casos. Mais formalmente, mostramos que, se a distância inicial d entre os robôs (medida em passos de robô) é escolhida aleatoriamente do conjunto $\{1, 2, \ldots, D\}$, então a probabilidade de o algoritmo randomizado ser mais rápido que o determinístico é maior que 0.5 para todo $D \in \mathbb{N}$. Por fim, consideramos uma variação do problema em que os agentes devem retornar a seu ponto de partida (por exemplo, para recolher os paraquedas), e comprovamos que a estratégia randomizada possui um melhor desempenho para qualquer distância inicial $d \in \mathbb{N}$.

2. O Problema do *Rendezvous* Unidimensional Simétrico com Marcadores

O Problema do *Rendezvous* Unidimensional Simétrico com marcadores na posição inicial consiste em dois agentes partindo de pontos distintos sobre uma linha infinita, deixando cada um uma marca que identifique suas posições iniciais. Eles devem se encontrar, ambos utilizando a mesma estratégia. Como mencionado, o problema dos robôs paraquedistas é um *puzzle* conhecido que ilustra essa modelagem. Quando um agente encontra o marcador do outro agente, ele passa imediatamente a conhecer de que lado, com relação a si próprio, o outro agente está. Essa informação fundamental é usada para quebrar a simetria dos movimentos, permitindo a elaboração de algoritmos determinísticos que não implicam os agentes se movimentarem de forma eternamente solidária. Um desses algoritmos, bastante simples, consiste em se estabelecer um movimento em ziguezague

em torno da posição inicial, com amplitude crescendo geometricamente até que o marcador do outro agente seja localizado, e a partir de então seguindo sempre na direção em que sabidamente o outro agente estará. Tal estratégia é eficiente para o problema em que um agente móvel busca um ponto fixo na reta, problema que é muitas vezes referido como o *Problema da Vaca Perdida* (Baeza-Yates et al. [1993]). Essa não é, contudo, sequer a melhor solução determinística para o problema em que estamos interessados.

Nas próximas seções revisitaremos a melhor solução conhecida até então para o problema, e mostraremos uma solução randomizada que obtém desempenho ainda melhor. Para efeito de cálculo dos tempo de execução, consideraremos que os agentes têm velocidades idênticas de um "passo por unidade de tempo" e os tempos de encontro serão calculados como uma função do número inicial d de passos que separam os agentes.

3. Solução determinística

A solução tradicional para o problema (veja, por exemplo, Peter [2016]) consiste em um algoritmo determinístico que é executado em duas etapas:

1. "Busca pelo paraquedas do outro": Enquanto não houver encontrado o marcador do outro agente, ande para a direita uma unidade de tempo e permaneça parado durante uma unidade de tempo.

2. "Perseguição": Após haver encontrado o marcador do outro agente, dê sempre um passo para a direita a cada unidade de tempo.

Os dois agentes executam o mesmo algoritmo. Durante a primeira etapa, eles farão os mesmos movimentos, mantendo inalterada a distância d que os separa. Note, porém, que apenas um deles, digamos o Agente A, encontrará o marcador do outro agente, digamos o Agente B, passando para a segunda etapa do algoritmo. A partir daí, o Agente A persegue o Agente B com velocidade de aproximação igual a um passo a cada duas unidades de tempo, eventualmente encontrando-o (veja Figura 1).

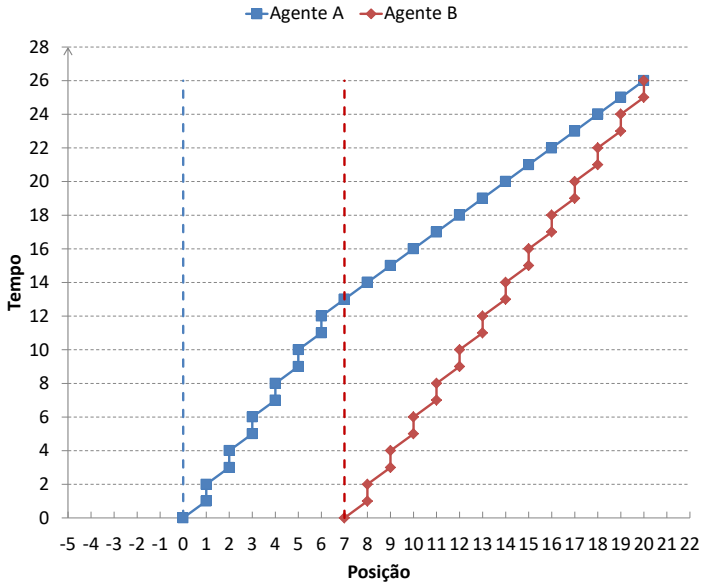


Figura 1: Deslocamento dos agentes ao longo do tempo utilizando o algoritmo determinístico

3.1. Análise da solução determinística

Na primeira etapa do algoritmo, os agentes andam um passo para a direita a cada duas unidades de tempo. Então, depois de $2d-2$ ciclos de relógio, eles terão andado uma distância igual a $d-1$ passos. No ciclo seguinte, os agentes terão percorrido distância d, e o Agente A encontra o marcador do Agente B. O tempo total transcorrido até aqui foi portanto de $2d-1$ unidades de tempo. Inicia-se a etapa da perseguição para o Agente A e, já no ciclo de relógio seguinte, a distância entre os agentes cairá para $d-1$ (uma vez que o Agente A dará um passo enquanto o Agente B, ainda na primeira etapa do algoritmo, permanecerá parado). Deste momento em diante, a cada duas unidades de tempo a distância entre eles se reduzirá em uma unidade. Serão necessárias, portanto, $2(d-1)$ unidades de tempo para que a distância entre eles caia a zero, totalizando $1 + 2(d-1) = 2d-1$ unidades o tempo total de perseguição. Executando, portanto, o algoritmo determinístico descrito acima quando partem de uma distância inicial d um do outro, os agentes levam um tempo total

$$t_{det}(d) = (2d-1) + (2d-1) = 4d-2$$

até se encontrarem.

4. Solução randomizada

Proporemos agora o seguinte algoritmo randomizado:

1. Escolha aleatória e uniformemente o sentido inicial do movimento (direita ou esquerda).
2. Enquanto não houver encontrado o marcador do outro agente, faça ziguezagues (movimentos de ida e volta com alternância de sentido) em torno do próprio ponto inicial, ininterruptamente, com amplitudes crescendo geometricamente, sendo a amplitude do i-ésimo ziguezague igual a 2^{i-1}, para $i = 1, 2, \ldots$ (o tempo para executar a i-ésima oscilação completa é portanto igual a 2^i).
3. Após haver encontrado o marcador do outro agente, continue andando um passo a cada ciclo de relógio no mesmo sentido de seu último passo.

Devido à escolha inicial dos agentes, existem quatro possíveis cenários em relação ao caminho dos agentes até se encontrarem: em dois deles, os agentes escolhem iniciar a busca seguindo no mesmo sentido; nos outros dois, escolhem sentidos opostos. Quando iniciam no mesmo sentido, um dos agentes, digamos o Agente A, encontrará o marcador inicial do Agente B e permanecerá indo em direção a ele. Quando o Agente B, em seu vai-e-vem, tiver completado seu movimento de ida e estiver voltando a seu ponto inicial, os agentes eventualmente se encontrarão. No caso em que escolhem iniciar para lados opostos, os agentes farão percursos perfeitamente simétricos, isto é, em oposição de fase, e se encontrarão exatamente no ponto médio do segmento que une suas posições iniciais antes mesmo de terem encontrado o marcador um do outro. É precisamente a possibilidade de ocorrer essa situação de oposição de fase que torna atraente a estratégia randomizada, pois, nesse caso, o encontro se dará de forma bastante rápida, mais do que compensando, em média, o fato de que a estratégia de ziguezague não os permite se encontrar tão rapidamente no caso complementar (movimentos iniciais no mesmo sentido) quanto a estratégia determinística da perseguição os permitiria. É o que provaremos a seguir.

4.1. Análise da solução randomizada

Nesta seção, desenvolveremos os cálculos do tempo esperado até o encontro dos agentes que iniciam a busca a uma distância $d \in \mathbb{N}$ um do outro. Para isso, analisaremos o tempo de encontro, como uma função de d, em cada um dos quatro possíveis cenários advindos da escolha aleatória inicial de cada agente.

Sejam os tempos de encontro $t_{EE}(d)$ e $t_{DD}(d)$ aqueles referentes aos cenários em que os agentes iniciam ambos para a esquerda (como ilustrado na Figura 2) ou ambos para a direita,

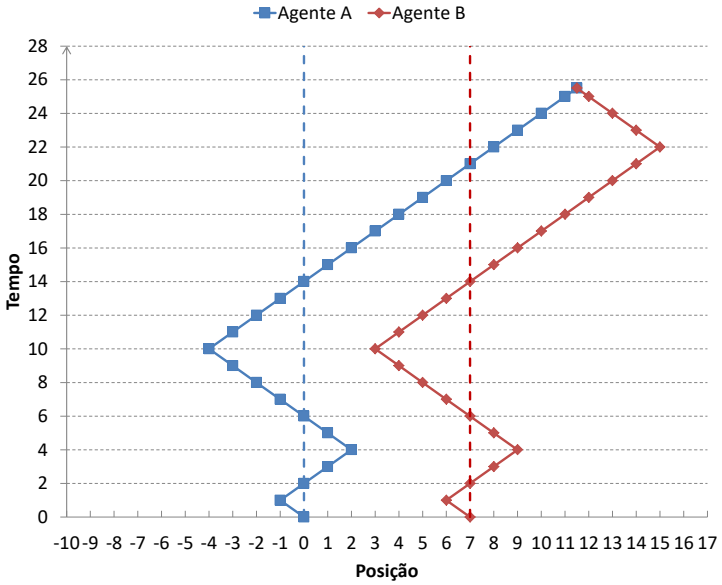


Figura 2: Deslocamento dos agentes ao longo do tempo utilizando o algoritmo randomizado quando ambos escolhem iniciar no mesmo sentido

respectivamente. Por simetria, $t_{EE} = t_{DD}$. Seja k o índice da primeira oscilação (movimento de vai-e-vem) cuja amplitude é maior ou igual à distância inicial d. Temos, então,

$$k = \lceil \log_2 d \rceil + 1,$$

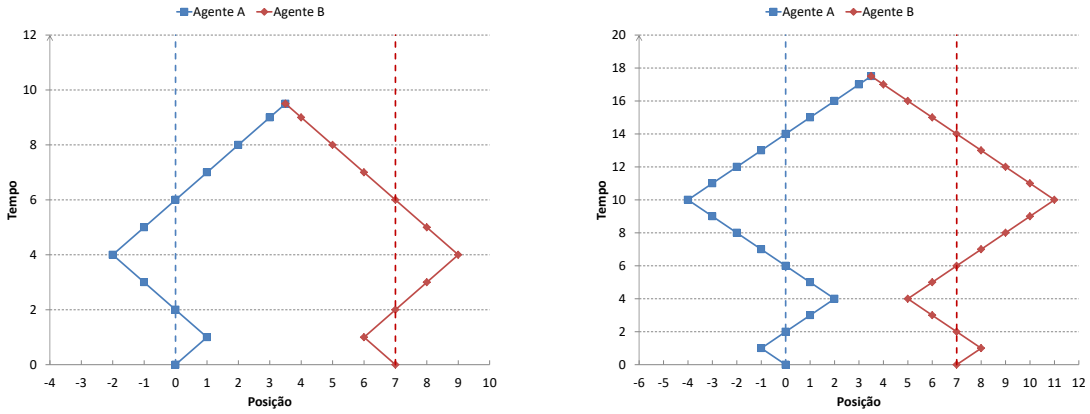
e a oscilação k será exatamente aquela em que um dos agentes, digamos o Agente A, encontrará o marcador do outro, digamos o Agente B. A oscilação $k-1$ será portanto a última que os agentes executarão completamente, retornando aos pontos de origem. Ao fim do movimento de ida da k-ésima oscilação (quando a distância entre eles será ainda igual a d), os agentes passarão a andar em sentidos opostos, pois o Agente A manterá o sentido de seu movimento (indo de encontro ao Agente B), ao passo que o Agente B terá entrado normalmente em seu movimento de retorno à posição de origem (indo de encontro ao Agente A). A partir daquele momento, portanto, os agentes levarão tempo $d/2$ até se encontrarem.

Os tempos t_{EE} e t_{DD} são, assim, compostos pelas seguintes parcelas:

- tempo até completar a $(k-1)$-ésima oscilação: $\sum_{i=1}^{k-1} 2^i$;
- duração do movimento de ida da k-ésima oscilação: $2^{\lceil \log_2 d \rceil}$;
- tempo andando em sentidos opostos após o movimento de ida da k-ésima oscilação: $d/2$.

Simplificando a soma das parcelas acima, obtemos

$$\begin{aligned} t_{EE}(d) = t_{DD}(d) &= \left(\sum_{i=1}^{\lceil \log_2 d \rceil} 2^i \right) + 2^{\lceil \log_2 d \rceil} + d/2 \\ &= \left(2^{\lceil \log_2 d \rceil + 1} - 2 \right) + 2^{\lceil \log_2 d \rceil} + d/2 \\ &= 3 \cdot 2^{\lceil \log_2 d \rceil} + d/2 - 2. \end{aligned}$$



(a) Iniciam se aproximando

(b) Iniciam se afastando

Figura 3: Deslocamento dos agentes ao longo do tempo utilizando o algoritmo randomizado e escolhem iniciar em direções opostas

Seja agora o tempo de encontro t_{DE}, referente ao cenário em que os agentes iniciam a primeira oscilação em sentidos opostos, isto é, indo na direção um do outro (o agente mais à esquerda anda para a direita, e o agente mais à direita anda para a esquerda, como ilustrado na Figura 3a). Nesse cenário, eles sempre seguirão em direções opostas e, portanto, se encontrarão no ponto médio do segmento de reta que une os pontos de partida. As oscilações de índice ímpar (a primeira oscilação é aquela de índice 1) são aquelas em que os agentes se aproximam, enquanto as de índice par são aquelas em que eles se afastam um do outro. Portanto, o índice da oscilação em que o encontro ocorre é o primeiro índice ímpar k' tal que $2^{k'-1} \geq d/2$. Isto é, $k' = 1 + \lceil \log_2(d/2) \rceil = \lceil \log_2 d \rceil$, se esse valor é ímpar, ou, caso contrário, esse valor mais uma unidade. Podemos escrever, de forma mais sucinta,

$$k' = \lceil \log_2 d \rceil + (\lceil \log_2 d \rceil + 1) \,\%2,$$

onde a notação "$\%2$" se refere à operação "módulo 2" (o resto da divisão por 2).

O tempo total $t_{DE}(d)$ é dado, portanto, pelo somatório das oscilações anteriores à k'-ésima, mais as $d/2$ unidades de tempo que correspondem ao movimento parcial de ida da própria k'-ésima oscilação (durante o qual os agentes vão de encontro um ao outro a partir de seus pontos iniciais):

$$\begin{aligned} t_{DE} &= \left(\sum_{i=1}^{k'-1} 2^i \right) + d/2 \\ &= \left(2^{\lceil \log_2 d \rceil + (\lceil \log_2 d \rceil + 1)\%2} - 2 \right) + d/2 \\ &= 2^{(\lceil \log_2 d \rceil + 1)\%2} \cdot 2^{\lceil \log_2 d \rceil} + d/2 - 2. \end{aligned}$$

Finalmente, seja t_{ED} o tempo de encontro quando os agente iniciam a primeira oscilação se afastando um do outro (Figura 3b). Assim como no cenário anterior, eles sempre se encontrarão no ponto central entre os marcadores. A diferença, nesse caso, é que eles se aproximam nas oscilações de índice par. Por raciocínio análogo ao do caso anterior, temos que o índice k'' da oscilação em que eles se encontram será dado por

$$k'' = \lceil \log_2 d \rceil + \lceil \log_2 d \rceil \,\%2,$$

e o tempo total desse cenário é calculado como se segue:

$$\begin{aligned} t_{ED} &= \left(\sum_{i=1}^{k''-1} 2^i\right) + d/2 \\ &= \left(2^{\lceil \log_2 d \rceil + \lceil \log_2 d \rceil \% 2} - 2\right) + d/2 \\ &= 2^{\lceil \log_2 d \rceil \% 2} \cdot 2^{\lceil \log_2 d \rceil} + d/2 - 2. \end{aligned}$$

Essa expressão é válida para $d \geq 2$, já que para $d = 1$ resultaria em um índice negativo para o limite do somatório, porém é fácil verificar que para $d = 1$, $t_{ED} = 5/2$.

De posse dos tempos para cada uma das possíveis escolhas iniciais dos agentes, temos que o tempo esperado de execução do algoritmo randomizado para uma distância inicial $d \geq 2$ é dado por:

$$\begin{aligned} E[t_{rand}(d)] &= \frac{t_{DD} + t_{EE} + t_{DE} + t_{ED}}{4} \\ &= \frac{t_{DD}}{2} + \frac{t_{DE} + t_{ED}}{4} \\ &= \frac{3 \cdot 2^{\lceil \log_2 d \rceil} + d/2 - 2}{2} + \frac{3 \cdot 2^{\lceil \log_2 d \rceil} + d - 4}{4} \\ &= 3 \cdot 2^{\lceil \log_2 d \rceil - 1} + d/4 - 1 + 3 \cdot 2^{\lceil \log_2 d \rceil - 2} + d/4 - 1 \\ &= 9 \cdot 2^{\lceil \log_2 d \rceil - 2} + d/2 - 2. \end{aligned}$$

5. Comparação dos algoritmos determinístico e randomizado

Sejam $T_{rand}(D)$ e $T_{det}(D)$ variáveis aleatórias que correspondem aos tempos de execução dos algoritmos randomizado e determinístico, respectivamente, quando a distância inicial d é uma variável aleatória escolhida uniformemente do intervalo $[1, D]$. Queremos mostrar que o desempenho esperado do algoritmo randomizado é sempre melhor do que o do algoritmo determinístico, qualquer que seja o tamanho D do intervalo considerado. Mais formalmente, queremos mostrar que

$$E[T_{rand}(D)] \leq E[T_{det}(D)],$$

ou ainda, que

$$\sum_{i=1}^{D} t_{rand}(i) \cdot P[d = i] < \sum_{i=1}^{D} t_{det}(i) \cdot P[d = i]$$

para todo $D \in \mathbb{N}$. O operador $E[\cdot]$ aparece aqui indicando a esperança de uma variável aleatória, e $P[\cdot]$ indica a probabilidade de ocorrência do evento indicado.

Como a distribuição de d é uniforme, as probabilidades para todos os possíveis valores de d são idênticas. Portanto, é suficiente compararmos os somatórios dos tempos entre 1 e D. Em outras palavras, queremos mostrar que

$$\sum_{i=1}^{D} t_{rand}(i) < \sum_{i=1}^{D} t_{det}(i). \tag{1}$$

Sejam $S_{rand}(D)$ e $S_{det}(D)$ os somatórios dos tempos de execução para d de 1 até D dos algoritmos randomizado e determinístico, respectivamente, correspondendo aos dois termos na desigualdade (1). Temos, então, que

$$S_{det}(D) = \sum_{i=1}^{D} t_{det}(i) = \sum_{i=1}^{D} 4i - 2 = \frac{D(2 + 4D - 2)}{2} = 2D^2. \tag{2}$$

Para o cálculo de $S_{rand}(D)$, devemos considerar a primeira parcela do somatório separadamente, já que a expressão encontrada para t_{rand} é válida apenas para $d \geq 2$.

$$\begin{aligned} S_{rand}(D) &= \sum_{i=1}^{D} t_{rand}(i) = t_{rand}(1) + \sum_{i=2}^{D} 9 \cdot 2^{\lceil \log_2 d \rceil - 2} + d/2 - 2 \\ &= t_{rand}(1) + 9 \cdot \sum_{i=2}^{D} 2^{\lceil \log_2 d \rceil - 2} + 1/2 \cdot \sum_{i=2}^{D} d - \sum_{i=2}^{D} 2. \end{aligned} \tag{3}$$

Mas

$$\begin{aligned} \sum_{i=2}^{D} 2^{\lceil \log_2 d \rceil - 2} &= \sum_{p=1}^{\lceil \log_2 D \rceil} (2^{p-1} \cdot 2^{p-2}) - (2^{\lceil \log_2 D \rceil} - D) 2^{\lceil \log_2 D \rceil - 2} \\ &= \frac{(4^{\lceil \log_2 D \rceil} - 1)}{6} - (2^{2\lceil \log_2 D \rceil - 2} - D \cdot 2^{\lceil \log_2 D \rceil - 2}) \\ &= \frac{(2^{2\lceil \log_2 D \rceil} - 1) - 6 \cdot 2^{2\lceil \log_2 D \rceil - 2} - 6 \cdot D \cdot 2^{\lceil \log_2 D \rceil - 2}}{6} \\ &= \frac{2^{\lceil \log_2 D \rceil - 1}(2^{\lceil \log_2 D \rceil + 1} - 3 \cdot D \cdot 2^{\lceil \log_2 D \rceil} + 3D) - 1}{6} \\ &= \frac{2^{\lceil \log_2 D \rceil - 1}(3D - 2^{\lceil \log_2 D \rceil}) - 1}{6}, \end{aligned}$$

o que nos permite substituir em (3) para obter

$$\begin{aligned} S_{rand}(D) &= \frac{3}{2} + 9 \left(\frac{2^{\lceil \log_2 D \rceil - 1}(3D - 2^{\lceil \log_2 D \rceil}) - 1}{6} \right) + \frac{1}{2} \frac{(D+2)(D-1)}{2} - (2D - 2) \\ &= \frac{3}{2} + 9D \cdot 2^{\lceil \log_2 D \rceil - 2} - 3 \cdot 2^{2\lceil \log_2 D \rceil - 2} - \frac{3}{2} + \frac{D^2 + D - 2}{4} - 2D + 2 \\ &= 3 \cdot 2^{\lceil \log_2 D \rceil - 2}(3D - 2^{\lceil \log_2 D \rceil}) + \frac{D^2 - 7D + 6}{4} \\ &= \frac{1}{4} \cdot \left(3 \cdot 2^{\lceil \log_2 D \rceil}(3D - 2^{\lceil \log_2 D \rceil}) + D^2 - 7D + 6 \right). \end{aligned}$$

Obtidas as expressões para $S_{det}(D)$ e $S_{rand}(D)$, queremos mostrar que a desigualdade $S_{rand}(D) < S_{det}(D)$ é verdadeira para todo $D \in \mathbb{N}$:

$$\begin{aligned} \frac{1}{4} \cdot \left(3 \cdot 2^{\lceil \log_2 D \rceil}(3D - 2^{\lceil \log_2 D \rceil}) + D^2 - 7D + 6 \right) &< 2D^2 \\ 3 \cdot 2^{\lceil \log_2 D \rceil}(3D - 2^{\lceil \log_2 D \rceil}) + D^2 - 7D + 6 &< 8D^2 \\ 3 \cdot 2^{\lceil \log_2 D \rceil}(3D - 2^{\lceil \log_2 D \rceil}) - 7D^2 - 7D + 6 &< 0 \end{aligned}$$

Fazendo $\omega = 2^{\lceil \log_2 D \rceil}$ e considerando o lado esquerdo da desigualdade acima como uma função $f(\omega)$, ocorre que $f(\omega)$ tem máximo em $\omega = 3D/2$. Portanto, $f(\omega) \leq f(3D/2)$ para todo ω no domínio da função. Resta mostrar que

$$\begin{aligned} f(3D/2) &< 0 \\ 3 \cdot \frac{3D}{2}(3D - \frac{3D}{2}) - 7D^2 - 7D + 6 &< 0 \\ -\frac{D^2}{4} - 7D + 6 &< 0 \end{aligned}$$

A parte positiva da solução da inequação acima é $D > \alpha$, onde $\alpha \approx 0.8324$, compreendendo portanto todos os inteiros positivos. Dessa forma, $S_{rand} < S_{det}$ é verdadeiro para todo $D \in \mathbb{N}$, e portanto, a estratégia randomizada permite que os agentes se encontrem, em média, mais rapidamente do que a estratégia determinística, como queríamos demonstrar.

Além de apresentar um desempenho esperado melhor, quando a distância inicial é escolhida aleatória e uniformemente do conjunto dos D primeiros naturais, é possível mostrar que o desempenho do algoritmo randomizado supera o do algoritmo determinístico *em mais da metade dos valores* de d no intervalo $[1, D]$, para todo D natural. Em outras palavras, se quisermos *maximizar a probabilidade de escolher a melhor estratégia* a ser executada para um valor qualquer de d escolhido uniformemente do conjunto dos primeiros D naturais, a melhor escolha é a estratégia randomizada. Para garantirmos isso, não basta sabermos, como já sabemos, que, *em média* (considerando-se todas as possíveis distâncias iniciais), a estratégia randomizada leva menos tempo; é preciso mostrar que, para qualquer $D \in \mathbb{N}$, mais da metade dos valores de d no intervalo $[1, D]$ satisfaz $E[t_{rand}(d)] < t_{det}(d)$, desigualdade essa que se traduz em

$$\begin{aligned} \frac{9}{4} \cdot 2^{\lceil \log_2 d \rceil} + \frac{d}{2} - 2 &< 4d - 2 \\ 2^{\lceil \log_2 d \rceil} &< \frac{7d}{2} \cdot \frac{4}{9} \\ \frac{2^{\lceil \log_2 d \rceil}}{d} &< \frac{14}{9} \end{aligned}$$

Seja $\alpha(d) = 2^{\lceil \log_2 d \rceil}/d$. Da última desigualdade, pode-se concluir que, para uma uma distância inicial d, o algoritmo randomizado terá um melhor desempenho esperado que o algoritmo determinístico quando $\alpha(d) < 14/9$. Por simplicidade, o conjunto de todas as distâncias iniciais em que o algoritmo randomizado tem um melhor desempenho será chamado de D_{rand}, e o das distância iniciais em o algoritmo determinístico possui um melhor desempenho será chamado de D_{det}.

Seja p uma potência de 2. Definimos como F_p o intervalo $[p/2 + 1, p]$. Note que, para qualquer $i \in F_p$, $2^{\lceil \log_2 i \rceil} = p$, e portanto, o valor de α será decrescente dentro do intervalo. Dito isto, importa-nos responder a seguinte questão: qual o menor valor de p, tal que o menor elemento de F_p pertence a D_{det}? É fácil ver que

$$\frac{p}{p/2 + 1} > 14/9 \Rightarrow 9p > 7p + 14 \Rightarrow p > 7.$$

Como p é uma potência de 2, temos que F_8 é o primeiro intervalo cujo primeiro elemento pertence a D_{det}. A partir deste, todos os próximos intervalos têm um d_{det} como primeiro elemento e um d_{rand} como último (já que o α referente a uma potência de 2 sempre será igual a 1). Por isso, é possível estabelecer um "ponto de corte" dentro de cada intervalo F_p, onde os valores menores que esse ponto pertencerão a D_{det}, e os demais pertencerão a D_{rand}. Esse ponto de corte é precisamente o valor $9p/14$, pois $\alpha(9p/14) = 14/9$. Esse valor não pode ser um inteiro, portanto $\lfloor 9p/14 \rfloor$ é o maior elemento do intervalo que pertence a D_{det}, e $\lceil 9p/14 \rceil$ é o menor elemento do intervalo que já pertence a D_{rand}.

A i-ésima posição do intervalo F_p contém o valor $p/2 + i$. Logo, a posição do valor $\lfloor 9p/14 \rfloor$ no intervalo é dada por

$$\lfloor 9p/14 \rfloor - p/2 = \lfloor 9p/14 - p/2 \rfloor = \lfloor p/7 \rfloor$$

Com isso, sabemos que, dos $p/2$ valores do intervalo F_p, exatamente $\lfloor p/7 \rfloor$ pertencem a D_{det}, e os demais $\lceil 5p/14 \rceil$ pertencem a D_{rand}. Analogamente, o intervalo seguinte F_{2p} possui $\lfloor 2p/7 \rfloor$ valores em D_{det} e $\lceil 5p/14 \rceil$ valores em D_{rand}; e assim por diante. Conclui-se, então, por indução, que a quantidade de valores em D_{det} de um intervalo é sempre menor que a quantidade

de valores em D_{rand} do intervalo imediatamente anterior. A base da indução é o intervalo F_8, que contém o menor valor natural em D_{det}, como visto anteriormente.

Por fim, segue da afirmação anterior que, para qualquer $D \in \mathbb{N}$, o intervalo $[1, D]$ possui mais valores em D_{rand} que em D_{det}, o que significa que para uma distância inicial escolhida aleatória e uniformemente desse intervalo, a chance de que o tempo esperado do algoritmo randomizado seja inferior ao tempo do algoritmo determinístico é maior do que 50%, como queríamos demonstrar.

6. Versão com retorno ao ponto inicial

Suponha agora que os robôs paraquedistas, após se encontrarem, precisem retornar ao ponto de partida. Numa tal versão, é necessário acrescentar o tempo de retorno entre o ponto de encontro e a posição inicial mais distante. Novamente, calcularemos e compararemos os desempenhos das estratégias determinística e randomizada.

6.1. Solução determinística

Sejam 0 e d as posições iniciais dos Agentes A e B, respectivamente. O Agente A encontra o marcador na posição d e em seguida permanece se movendo uma unidade a cada ciclo de relógio durante $2d - 1$ unidades de tempo, conforme o tempo de perseguição calculado na Seção 3.1. Portanto, o encontro acontece na posição $3d - 1$.

Seja $t'_{det}(d)$ o tempo de execução do algoritmo determinístico para a versão com retorno. Temos, então, que $t'_{det}(d)$ é igual a $t_{det}(d)$ mais o tempo de retorno do Agente A, isto é,

$$t'_{det}(d) = (4d - 2) + (3d - 1) = 7d - 3.$$

6.2. Solução randomizada

Para o cálculo do tempo esperado para o algoritmo randomizado é necessário considerar o tempo de retorno dos agentes ao ponto inicial em cada um dos 4 casos descritos na Seção 4 referentes às escolhas dos agentes para os sentidos iniciais de seus movimentos.

Os casos mais simples são aqueles em que os agentes escolhem direções opostas, pois o ponto de encontro será exatamente o ponto médio do segmento de reta que une os pontos iniciais, sendo necessário um tempo extra $d/2$ para que os agentes retornem. Sejam $t'_{DE}(d)$ e $t'_{ED}(d)$ os tempos de execução para estes casos, para $d \geq 2$. Segue, então, que

$$\begin{aligned} t'_{DE}(d) &= t_{DE}(d) + d/2 = 2^{(\lceil \log_2 d \rceil + 1)\%2} \cdot 2^{\lceil \log_2 d \rceil} + d - 2; \\ t'_{ED}(d) &= t_{ED}(d) + d/2 = 2^{(\lceil \log_2 d \rceil)\%2} \cdot 2^{\lceil \log_2 d \rceil} + d - 2. \end{aligned}$$

Sejam $t'_{EE}(d)$ e $t'_{DD}(d)$ os tempos de execução para os casos em que os agentes escolhem iniciar para o mesmo lado. Nesses casos, o agente que mais se afastará do seu ponto inicial será aquele que encontrar o marcador, digamos o Agente A. O algoritmo, então, se encerrará quando este retornar ao seu ponto de origem. Conforme visto na Seção 4, o marcador do Agente B será encontrado na oscilação de amplitude $2^{\lceil \log_2 d \rceil}$. Ao final desta, o Agente A se afastará ainda mais de seu ponto inicial uma distância $d/2$, quando ocorrerá o encontro. Portanto,

$$\begin{aligned} t'_{EE}(d) = t'_{DD}(d) &= t_{EE}(d) + d/2 + 2^{\lceil \log_2 d \rceil} \\ &= (3 \cdot 2^{\lceil \log_2 d \rceil} + d/2 - 2) + d/2 + +2^{\lceil \log_2 d \rceil} \\ &= 4 \cdot 2^{\lceil \log_2 d \rceil} + d - 2 \end{aligned}$$

Logo, o tempo esperado do algoritmo randomizado para esta versão, quando a distância inicial entre os agentes é igual a $d \geq 2$, é

$$\begin{aligned} E[t'_{rand}(d)] &= \frac{1}{4}(t'_{DE}(d) + t'_{ED}(d) + t'_{EE}(d) + t'_{DD}(d)) = \frac{1}{4}(t'_{DE}(d) + t'_{ED}(d)) + \frac{1}{2}(t'_{EE}(d)) \\ &= \frac{1}{4}(3 \cdot 2^{\lceil \log_2 d \rceil} + 2d - 4) + \frac{1}{2}(4 \cdot 2^{\lceil \log_2 d \rceil} + d - 2) \\ &= \frac{1}{4}(11 \cdot 2^{\lceil \log_2 d \rceil} + 4d - 8) = \frac{11}{4} \cdot 2^{\lceil \log_2 d \rceil} + d - 2. \end{aligned}$$

6.3. Comparação dos algoritmos determinístico e randomizado

Vimos que o ponto de encontro fornecido pelo algoritmo determinístico estará sempre a uma distância $3d - 1$ de um dos marcadores iniciais. Enquanto isso, o algoritmo randomizado estabelece que o encontro se dará na posição que acrescenta o menor tempo possível à fase de retorno com probabilidade $1/2$. Além disso, mesmo nos piores casos da estratégia randomizada, a distância ao ponto de origem mais distante é sempre menor que aquela causada pela estratégia determinística, já que $2^{\lceil \log_2 d \rceil} + d/2 < 3d - 1$ para qualquer $d \in \mathbb{N}$. Os cálculos a seguir mostram a comparação entre os tempos de execução dos dois algoritmos para a versão em que os agentes devem retornar ao ponto inicial, e como essa distância entre o ponto de encontro e os marcadores impactam em seus desempenhos.

Queremos, portanto, calcular para quais distâncias iniciais o tempo esperado de execução do algoritmo randomizado é menor que o do algoritmo determinístico. Ou seja,

$$\begin{aligned} E[t'_{rand}(d)] &< t'_{det}(d) \\ \frac{11}{4} \cdot 2^{\lceil \log_2 d \rceil} + d - 2 &< 7d - 3 \\ 2^{\lceil \log_2 d \rceil} &< \frac{24d - 4}{11} \end{aligned}$$

Para todo $d \geq 2$, é verdade que

$$2^{\lceil \log_2 d \rceil} < 2d \leq \frac{24d - 4}{11},$$

e para o único caso entre os naturais não coberto pelas expressões acima, $d = 1$, segue que $E[t'_{rand}(1)] = 5/2 < t'_{det}(1) = 4$.

Portanto, para a versão com retorno às posições iniciais, a estratégia randomizada possui um desempenho superior à determinística para *todas* as distâncias iniciais $d \in \mathbb{N}$.

7. Conclusão

Neste artigo, estudamos o Problema de Rendezvous Unidimensional Simétrico onde os agentes deixam marcas nos pontos de partida. Propomos uma estratégia randomizada e analisamos seu desempenho em função da distância inicial. Mostramos que, para uma distância inicial d escolhida aleatoriamente de um intervalo $[1, D]$, em média a estratégia randomizada leva menos tempo que a estratégia determinística mais rápida conhecida, para todo $D \in \mathbb{N}$. Mostramos também que, para todo $D \in \mathbb{N}$, é maior do que 50% a probabilidade de que o tempo esperado de execução do algoritmo randomizado para distância d uniformemente escolhida em $[1, D]$ seja menor que o tempo do algoritmo determinístico para a mesma entrada.

Uma vez que não apenas o tempo de execução, mas também a distância do ponto de encontro aos pontos iniciais é reduzida pela estratégia proposta, consideramos a versão em que os agentes devem retornar aos pontos de origem após se encontrarem. Para esta versão, o algoritmo randomizado apresenta tempo esperado de execução menor que o determinístico para *toda* distância inicial $d \in \mathbb{N}$.

Algoritmos randomizados são geralmente usados para tornar mais simples e/ou mais eficiente a solução de determinado problema. Sua análise, no entanto, é por vezes trabalhosa, podendo o emprego de aleatoriedade se dar em vários momentos durante a execução do algoritmo. Este artigo ilustra uma aplicação de randomização que envolve uma única escolha aleatória, e que traz ganhos sensíveis de performance, e que é de simples análise, tendo portanto considerável potencial didático.

Referências

Alpern, S. (1995). The rendezvous search problem. *SIAM Journal on Control and Optimization*, 33(3):673–683.

Anderson, E. J. e Essegaier, S. (1995). Rendezvous search on the line with indistinguishable players. *SIAM Journal on Control and Optimization*, 33(6):1637–1642.

Baeza-Yates, R. A., Culberson, J. C., e Rawlins, G. J. E. (1993). Searching in the plane. *Information and Computation*, 106:234–252.

Baston, V. e Gal, S. (1998). Rendezvous on the line when the players' initial distance is given by an unknown probability distribution. *SIAM Journal on Control and Optimization*, 36(6):1880–1889.

Beveridge, A., Ozsoyeller, D., e Isler, V. (2011). Symmetric rendezvous on the line with an unknown initial distance.

Farrugia, A., Gasieniec, L., Kuszner, L., e Pacheco, E. (2015). Deterministic rendezvous in restricted graphs. In *International Conference on Current Trends in Theory and Practice of Informatics*, p. 189–200. Springer.

Gal, S. (1999). Rendezvous search on the line. *Operations Research*, 47(6):974–976.

Pelc, A. (2012). Deterministic rendezvous in networks: A comprehensive survey. *Networks*, 59(3): 331–347.

Peter, D. (2016). Web page. `http://david-peter.de/parachuting-robots/`. Acessado: 2016-12-10.