

# Sumário

<b>1</b>	<b>BFA</b>	<b>7</b>
1.1	Inspiração Biológica . . . . .	7
1.1.1	Princípios Básicos . . . . .	8
1.1.2	Busca de Alimentos por Bactérias: E. Coli . . . . .	11
1.2	Visão Geral do Algoritmo . . . . .	15
1.2.1	Quimiotaxia, população, reprodução, eventos eliminativos e dispersivos . . . . .	16
1.2.2	O algoritmo passo-a-passo . . . . .	19
1.2.3	Seleção dos parâmetros . . . . .	22
1.2.4	Comparação com outros métodos de otimização . . . . .	25
1.3	Aplicações . . . . .	25
1.3.1	Onde Usar . . . . .	25
1.3.2	Estudo de Caso . . . . .	26
1.3.3	Exercícios . . . . .	27



# Lista de Figuras

1.1	Estratégias de Busca de Alimentos (figura adaptada de [22]). . .	9
1.2	Movimentação, saltos e quimiotaxia das bactérias E. Coli (figura adaptada de [22]). . . . .	13
1.3	Gráfico de concentração de nutrientes. . . . .	18
1.4	Modelo de atração química entre células. . . . .	19
1.5	Execução do algoritmo durante as gerações $N_{re} = 4$ . . . . .	20
1.6	Fluxograma de execução do algoritmo BFA. . . . .	23



# Lista de Tabelas

1.1	Otimização por coleta de alimentos de bactérias. . . . .	7
1.2	Parâmetros de inicialização do algoritmo BFA padrão. . . . .	24
1.3	Comparação entre técnicas de otimização global. . . . .	26



# Capítulo 1

## BFA

BFA (Bacterial Foraging Algorithm) é uma das técnicas de otimização por enxames de partículas mais recentes. Embora os primeiros modelos tenham sido propostos em 2002 [22, 24], já existem algumas variantes do algoritmo [12, 27] e aplicações em diversos problemas práticos [14, 25, 26, 28]. A otimização é inspirada nas estratégias usadas pelas bactérias para alimentação, incluindo os aspectos relacionados a movimentação das bactérias, reação à elementos atratores e repulsores no ambiente, eventos eliminativos-dispersivos e a reprodução das células.

Este capítulo apresentará a inspiração biológica, o modelo computacional adotado para resolução de problemas de otimização global e mostrará exemplos de aplicação desta abordagem na solução de problemas reais. A tabela 1.1 sumariza a técnica.

Tabela 1.1: Otimização por coleta de alimentos de bactérias.

Representação	Posição de bactérias em um gradiente nutritivo
Operadores diversidade	Movimentos quimiotáticos e dispersão/eliminação
Seleção de pais	Reprodução baseada na maior aptidão
Seleção de sobreviventes	Substituição dos pais

### 1.1 Inspiração Biológica

A seleção natural tende a eliminar animais com estratégias pobres para localização, tratamento e ingestão de alimentos, pelo fato dos animais com esses genes terem uma capacidade menor de se reproduzirem (obter comida suficiente para possibilitar a reprodução). Após muitas gerações, estratégias pobres para busca de alimentos são eliminadas ou redesenhadas. Estes princípios evolucionários levaram os cientistas da área a criar a hipótese de que é interessante modelar o processo de busca de alimentos como um processo de otimização: um animal

executa ações para maximizar a energia obtida por unidade de tempo gasta no processo de busca de comida, levando em consideração as restrições impostas pela sua fisiologia (capacidades sensoriais e cognitivas) e o ambiente (densidade de presas, risco de predadores, características físicas da área de busca). Os modelos de otimização baseado nestes conceitos também são válidos para o processo social de busca de alimentos, onde grupos de animais executam a tarefa de forma cooperativa.

Os modelos computacionais usados para otimização (incluindo o algoritmo BFA original) normalmente são baseados nas estratégias de alimentação da bactéria *E. Coli*. Mas os elementos básicos do processo de alimentação são observados por qualquer organismo vivo.

### 1.1.1 Princípios Básicos

Conforme descrito em [22], a teoria que explica a busca de alimentos pelos animais é baseada na hipótese de que os animais procuram e obtêm nutrientes de uma forma que maximizem sua energia intacta  $E$  por unidade de tempo  $T$  gasto no processo. Ou seja, tenta-se maximizar uma função como a da Eq. 1.1.

$$\frac{E}{T} \quad (1.1)$$

A maximização dessa função provê nutrientes suficientes para sobrevivência e tempo adicional para outras atividades (como lutar, reproduzir, dormir). Evidentemente o processo de alimentação é diferente em cada espécie. Herbívoros encontram comida em abundância, mas precisam ingerir uma grande quantidade de alimentos. Carnívoros encontram comida com mais dificuldade mas precisam ingerir uma quantidade menor de alimentos devido ao teor energético da sua dieta.

Para muitos animais, os nutrientes estão distribuídos em "fontes" de comida (um lago, um prado, um arbusto com frutas, um grupo de árvores com frutas). O processo envolve a descoberta de tais fontes, decidindo se deseja adotar essa fonte e procurar comida, ou se deve parar de procurar comida na fonte atual e buscar outra fonte que espera-se ter alimentos de maior qualidade e quantidade do que a fonte atual. Fontes de alimentos geralmente são encontradas seqüencialmente, e muitas vezes são necessários grandes esforços e risco para mudar de uma fonte para outra. Um animal pode encontrar um local pobre em nutrientes, mas baseado na sua experiência pode entender que ainda assim é uma boa fonte (considerando os riscos e esforços para mudar de localização).

### Estratégias de Busca de Alimentos

O estudo de estratégias de busca de alimentos evidencia aspectos comuns a muitos animais. Inicialmente, predadores precisam localizar a presa. Depois, precisam perseguir e atacar a presa. Finalmente, eles "tratam" e ingerem a presa. A importância dos vários componentes observados no processo dependem da relação entre predador e presa. Se a presa é maior do que o predador,



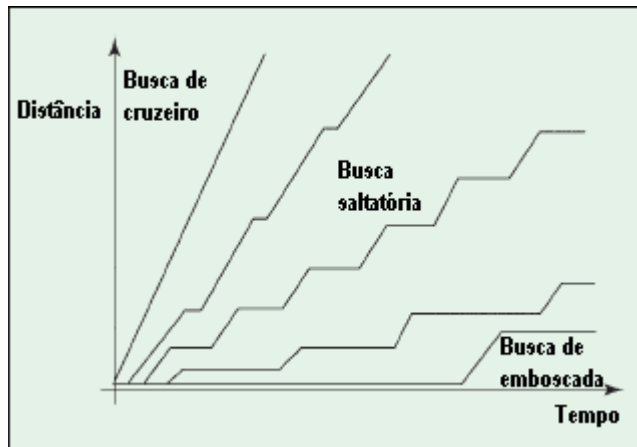


Figura 1.1: Estratégias de Busca de Alimentos (figura adaptada de [22]).

então a perseguição, ataque e movimentação são cruciais. Ou seja, pode ser fácil encontrar a presa, mas o tamanho relativo confere uma vantagem à presa. Se a presa é menor do que o predador, então o componente de busca normalmente é mais importante. O pequeno tamanho pode ser uma vantagem para a presa. Nessa situação, as presas devem ser consumidas com frequência e em grande número. Os casos em que o componente de busca é o fator determinante aplicam-se a muitas aves, peixes, lagartos e insetos.

Abordagens típicas para obtenção de alimentos são: movimentar-se e encurralar a presa; ou permanecer estacionário e armar emboscadas. No primeiro caso o predador se move continuamente, buscando por uma presa ao redor do ambiente (como o faz o peixe atum e os falcões). No caso de emboscada, o predador (como uma cascavel), aguarda estático que uma presa cruze o espaço delimitado pelo alcance de seu bote. Para muitas espécies de animais, a estratégia de busca situa-se entre as duas abordagens. Em particular, a estratégia "saltatória" define que o animal deve movimentar-se de tempos em tempos e aguardar a presa. Para ilustrar as diversas estratégias, considere a Fig. 1.1.

O gráfico da Fig. 1.1 relaciona distância percorrida na busca e o tempo decorrido. Na busca de cruzeiro a distância aumenta a uma taxa constante determinada pela velocidade na qual o animal se movimenta no ambiente. Na outra ponta, o emboscador aguarda por um longo tempo e então realiza um movimento brusco para tentar obter a presa. Entre estes dois extremos, a estratégia de saltos é baseada na alternância de movimentos de busca e de emboscadas.

### Aspectos Sociais e Busca Inteligente de Alimentos

Os conceitos abordados acima referem-se ao comportamento individual dos predadores e presas. Obviamente, os animais são beneficiados quando real-

izam o processo de alimentação em grupos. Para alcançar a interação social, deve existir algum mecanismo de comunicação. No caso dos humanos usamos a linguagem, mas para os animais, é freqüente observar a comunicação através de movimentos específicos, ruídos ou mudanças de postura. Podemos elencar as seguintes vantagens no processo de busca de alimentos em grupo:

- Como mais animais estão à procura de alimentos, a probabilidade de encontrá-los é maior. Quando um animal encontra os alimentos, os demais podem ser avisados. Emerge o conceito de centro de informações.
- Aumenta a capacidade de lidar com presas maiores.
- Proteção de membros do grupo por parte de predadores. Em algumas espécies os membros do meio do grupo são protegido pelos membros que estão nas bordas.

Às vezes é útil pensar de um grupo (enxame) dos animais como uma única criatura viva, onde através de agrupamento e de comunicação uma "inteligência coletiva" emerge, resultando em mais sucesso no processo de alimentação para cada indivíduo (e os ganhos podem compensar os efeitos da competição por alimento dentro dos grupos, trabalhando em conjunto, pode haver mais alimentos do que se não houver cooperação).

Nesse contexto do processo de alimentação em grupo pode-se pensar em uma alcatéia de lobos caçando ou um bando de pássaros, enxame de abelhas, colônia de formigas, ou cardume de peixes. As ligações entre o comportamento de colônias de formigas, otimização e aplicações de engenharia, foram amplamente estudadas. Diversos textos já foram produzidos [2, 8, 13] para explicar como colônias de formigas (ACO) podem resolver problemas de menor caminho, ou resolver o problema do caixeiro viajante (problemas de otimização combinatória em geral). Estas formigas usam uma forma de comunicação indireta conhecida como "estigmergia", onde uma formiga pode modificar seu ambiente e, posteriormente, outra formiga pode mudar o seu comportamento devido a essa modificação. Por exemplo, se uma formiga sai a procura de comida, pode pesquisar em locais distantes e selecionando caminhos inicialmente aleatórios, no entanto, uma vez que encontra uma fonte de alimento, esta formiga retorna ao formigueiro, criando uma trilha de "feromônio" (que pode ficar no lugar até alguns meses). Então, quando outras formigas iniciam seu processo de busca de comida, eles tendem a seguir o rastro de feromônio e encontrar alimentos mais facilmente. Você pode, então, pensar na primeira formiga como tendo "recrutado" exploradores adicionais e as trilhas como um tipo de memória para a colônia de formigas como um todo (isto é, utilizando as comunicações e trabalho em conjunto, ganha-se a importante capacidade fisiológica de aprendizagem).

Formas de vida inferiores podem alcançar aspectos de inteligência superiores no processo de alimentação cooperando em grupos (por exemplo, as formigas podem "aprender" onde o alimento está localizado). Indivíduos mais complexos são naturalmente dotados de mais inteligência cognitiva e portanto, maior capacidade de alimentação usando estratégias individuais. Naturalmente, no caso

dos seres humanos o processo é significativamente diferente de muitas formas de vida, pois, ao invés de simplesmente tentar lidar com um ambiente dado, tentamos modificar o nosso meio ambiente através da agricultura para melhorar o nosso processo de alimentação.

Na seção seguinte, considera-se o processo de alimentação para indivíduos e grupos de bactérias, organismos que são muito mais simples do formigas ou seres humanos e ainda assim podem trabalhar em conjunto para o benefício do grupo. Pode-se pensar nestes processos biológicos como a execução de um processo de otimização simples, mas eficaz durante a alimentação das bactérias, que podem ser úteis na solução de problemas de engenharia. Depois de explicar como as bactérias se alimentam, o processo será modelado através de um simulação em computador e, em seguida, será discutido a sua relevância como algoritmo de otimização tomando como estudo de caso o controle autônomo de veículos.

### 1.1.2 Busca de Alimentos por Bactérias: E. Coli

A bactéria *Escherichia Coli* tem uma membrana plasmática, parede celular, e a cápsula que contém o citoplasma e nucleóide. O pili (singular, pilus) são usados para um tipo de transferência de genes da E. coli para as outras bactérias, e flagelos são utilizados para a locomoção. A célula possui cerca de 1  $\mu\text{m}$  de diâmetro e 2  $\mu\text{m}$  de comprimento. A célula da E. coli pesa em torno de 1 picograma e possui cerca de 70% de água. *Salmonella typhimurium* é um tipo similar de bactéria.

A bactéria E. coli é provavelmente o microorganismo mais bem compreendido. Seu genoma foi seqüenciado e contém 4.639.221 de adenosina, citosina, guanina e timina, organizados em um total de 4.288 genes. As mutações na E. coli ocorrem a uma taxa de cerca de  $10^{-7}$  por gene, por geração, e podem afetar os seus aspectos fisiológicos (por exemplo, a eficiência reprodutiva em diferentes temperaturas). A bactéria E. coli, ocasionalmente realizam um tipo de reprodução "sexuada" chamada "conjugação" onde pequenas seqüências de genes são unidirecionalmente transferidas de uma bactéria para outra através do pilus.

Quando a E. coli cresce, torna-se mais longa, e então se divide no meio em duas "filhas". Com alimentos suficientes e considerando a temperatura do intestino humano de  $37^{\circ}\text{C}$ , a E. coli pode sintetizar e reproduzir, precisando apenas fazer uma cópia de si mesma em cerca de 20 min. Portanto, o crescimento de uma população de bactérias é exponencial e em um curto espaço de tempo obtém-se o dobro de bactérias.

A bactéria E. coli tem um sistema de controle (sistema de orientação) que permite a busca de comida e ao mesmo tempo que tenta evitar substâncias nocivas. Por exemplo, as bactérias nadam evitando ambientes alcalinos e ácidos e procuram ambientes mais neutros. Para explicar o comportamento da bactéria E. coli, é necessário entender seu atuador (flagelo), "a tomada de decisões", sensores, e os aspectos relacionados à motilidade. Percebe-se facilmente que a E. coli executa um tipo de estratégia baseada em saltos.

### Movimentação Através de Flagelos

A locomoção é feita através de um conjunto relativamente rígido de flagelos, que permitem que as bactérias possam nadar através de cada uma delas girando no mesmo sentido e em cerca de 100-200 rotações por segundo (em termos de sistemas de controle, entendemos os flagelos como atuadores). Cada flagelo é uma hélice configurada de modo que a base do flagelo (isto é, onde é ligado à célula) gira no sentido anti-horário, como visto a partir da extremidade livre do flagelo olhando para a célula, ele produz uma força contra a bactéria para que ele possa empurrar a célula. Neste tipo de movimento pode-se pensar em cada flagelo como um tipo de propulsor. Se um flagelo gira no sentido horário, ele vai puxar o celular. Sob uma perspectiva de engenharia, o eixo de rotação na base do flagelo é bastante interessante, é um mecanismo que parece usar o que os biólogos chamam de uma "junção universal" (assim a estrutura rígida do flagelo pode "apontar" em direções diferentes em relação à célula). Esse mecanismo que cria as forças de rotação para girar o flagelo em qualquer direção é descrito pelos biólogos como um motor biológico (algo relativamente raro na biologia, apesar de vários tipos de bactérias usarem esse mecanismo) [22]. O motor é bastante eficiente no que faz: um giro completo usando apenas 1.000 prótons e, ainda assim, a *E. coli* gasta menos de 1% do sua energia total com motilidade.

Uma bactéria *E. coli* pode mover-se de duas maneiras diferentes: ela pode "correr" em linha reta (nadar por um período de tempo) ou ela pode executar um pequeno movimento aleatório de giro que muda a direção do movimento de corrida. Estes movimentos são alternados durante toda a vida da bactéria (ou seja, é raro um flagelo parar de girar). Após explicar cada tipo de movimento, será mostrado como as bactérias determinam quanto tempo devem nadar em linha reta antes de alterar a direção do movimento.

Se os flagelos giram no sentido horário, cada flagelo puxa a célula, e o efeito real é que cada flagelo opera relativamente independente dos demais, e assim a bactéria se movimenta de modo aleatório e com pouco deslocamento. Observe a Fig. 1.2 (a). Para executar esse movimento depois de uma corrida em linha reta, a célula diminui a velocidade ou pára, uma vez que as bactérias são tão pequenas, elas não experimentam quase nenhuma inércia, e o efeito predominante é o da viscosidade. Por isso, quando uma bactéria pára de correr, ele pára dentro do diâmetro de um próton. Sob certas condições experimentais (ambiente isotrópico, homogêneo, sem nutrientes nem substâncias nocivas), para uma célula padrão, a duração deste movimento de giro é de cerca de  $0,14 \pm 0,19$  s (média  $\pm$  desvio padrão). Depois de um giro, a célula geralmente apontará para uma direção aleatória, mas existe uma ligeira tendência para ser colocado na direção que ela estava viajando antes do giro.

Se o movimento dos flagelos é anti-horário, seus efeitos se acumulam e há formação de um feixe (acredita-se que o feixe é formado devido à viscosidade do meio) e, portanto, observamos uma hélice composta que empurra a bactéria para que seja deslocada mais ou menos em linha reta (ver Fig. 1.2 (a)). Na média, as bactérias nadam a uma taxa de  $10\text{-}20 \mu\text{m/s}$ , ou cerca de dez vezes o comprimentos de seu corpo por segundo, mas em um meio rico elas podem nadar

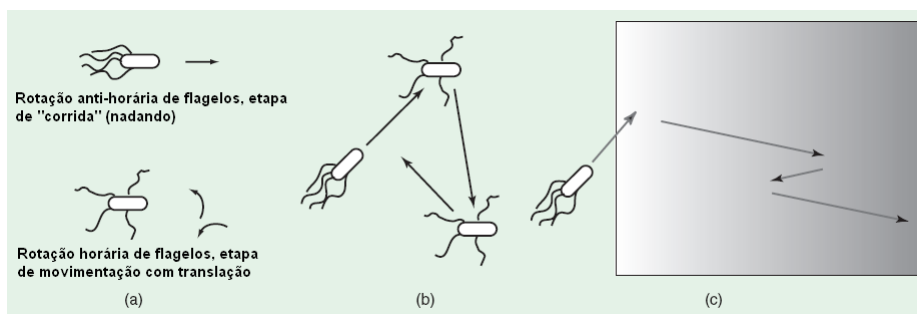


Figura 1.2: Movimentação, saltos e quimiotaxia das bactérias *E. Coli* (figura adaptada de [22]).

ainda mais rápido. Sob certas condições experimentais (ambiente isotrópico, homogêneo, igual ao mencionado acima), para uma célula média, o intervalo em que a bactéria se move em linha reta é de cerca de  $0,86 \pm 1,18$  s. Além disso, nestas condições, a velocidade média é de  $14,2 \pm 3,4 \mu\text{m/s}$ . A direção das corridas não é perfeitamente uma linha reta visto que a célula está sujeita ao movimento browniano, que faz com que ele saia fora do curso em um ângulo de aproximadamente  $30^\circ$ . Na média, após 10 s a direção é deslocada em cerca de  $90^\circ$  e esquece-se, portanto, a direção inicial do movimento. Finalmente, note que para muitas bactérias e meios de locomoção os flagelos podem induzir outros movimentos (por exemplo, a rotação das bactérias em torno de um eixo).

### Motilidade e Gradientes Nutritivos

Os padrões de movimento que as bactérias podem gerar, na presença de produtos químicos atratores e repulsores são chamados de "quimiotaxia". Para a *E. coli*, o encontro com serina ou aspartato corresponde a respostas atrativas. As respostas repulsivas surgem do encontro com íons metálicos Ni e Co, mudanças no pH, aminoácidos como leucina, e ácidos orgânicos, como o acetato. Extrapolando do comportamento individual para uma população de bactérias, em geral, os grupos vão tentar encontrar comida e evitar fenômenos prejudiciais, e quando visto sob um microscópio, percebe-se um tipo de comportamento inteligente, uma vez que as bactérias parecem se mover intencionalmente como um grupo.

Observe que, se uma *E. coli* está na presença de alguma substância neutra (no sentido de que não há alimentos ou substâncias nocivas), e se ela permanece neste meio por um tempo (por exemplo, mais de 1 min), em seguida, os flagelos, simultaneamente, alternam entre rotações no sentido horário e anti-horário, ou seja, a bactéria vai alternar entre os movimentos de "corrida" e "giro". Essa alternância entre os dois modos moverá a bactéria, mas em direções aleatórias, e isso lhe permite "buscar" por nutrientes (vide Fig. 1.2 (b)). Se as bactérias são colocadas em uma concentração homogênea de serina (isto é, com um nutriente, mas sem gradientes), em seguida, uma série de alterações ocorre nas

características dos seu comportamento de motilidade. Por exemplo, a média da distância e do tempo na fase de corrida aumenta e o tempo médio dos movimentos de giro diminui. Elas ainda produzem, no entanto, um tipo básico de comportamento de busca, mesmo que a bactéria tenha algum alimento, ela persistentemente busca mais. Como exemplo de alternância entre translações e corridas no meio homogêneo, isotrópico descrito acima, em experimentos de motilidade com duração de 29,5 s haviam 26 corridas, o tempo máximo de corrida foi de 3,6 s, e a velocidade média foi de cerca de  $21 \mu\text{m/s}$ .

Agora suponha que a bactéria encontre um gradiente de nutrientes (por exemplo, serina), como mostrado na Fig. 1.2 (c). A alteração na concentração de nutrientes provoca uma reação de tal forma que a bactéria vai gastar mais tempo correndo e menos tempo com movimentos de giro. Quando a bactéria se desloca em um gradiente de concentração positivo (movimentos em direção ao aumento das concentrações de nutrientes) tenderá a prolongar o tempo que passa correndo até um ponto. As direções dos movimentos são guiadas para o aumento do gradiente de nutrientes e a célula não muda sua direção durante uma corrida devido às mudanças no gradiente - os movimentos de giro é que basicamente determinam a direção, além das influências do movimento browniano mencionadas acima.

Por outro lado, se a bactéria corre na direção de um gradiente positivo de substâncias nocivas, ela retornará ao seu comportamento padrão para que, essencialmente, tente procurar um caminho de volta até o gradiente. Por exemplo, em certas condições, para uma célula padrão nadar até gradientes de serina, a corrida é de  $2,19 \pm 3,43$  s, mas se ela nada se afastando de um gradiente de serina o tempo médio de corrida é de  $1,40 \pm 1,88$  s. Assim, quando ela se move para cima do gradiente, aumenta a distância percorrida.

Finalmente, suponha que a bactéria atinge uma região com concentração de nutrientes constante depois de ter permanecido em um gradiente positivo por algum tempo. Neste caso, após um período de tempo (não imediatamente), a bactéria vai voltar à mesma proporção de corrida e giro observada para a substância neutra, de modo que ele retorne ao seu comportamento padrão de busca. Ela nunca está satisfeita com o quantidade de alimento ao redor, ou seja, sempre procura concentrações mais elevadas. Na verdade, sob certas condições experimentais, a célula irá comparar a concentração observada no último segundo com a concentração observada ao longo dos últimos 3 segundos antes de responder à diferença. Por isso, usa os últimos 4 s de dados de concentração de nutrientes para decidir por quanto tempo irá correr. Considerando os desvios de direção, devido ao movimento browniano discutido acima, a bactéria basicamente usa o tempo, tanto quanto possível na tomada de decisões sobre a navegação nos gradientes.

Basicamente, a bactéria está tentando nadar em locais com baixas concentrações de nutrientes para locais com altas concentrações. Um tipo de comportamento oposto é usado quando se depara com substâncias nocivas. Se a bactéria se depara com vários tipos de concentrações com o tempo, então a bactéria tende a "perseguir" os ambientes mais favoráveis e se afastar dos mais nocivos.

### Mecanismos para Tomada de Decisão

O sistema sensorial e de tomada de decisões da bactéria *E. Coli* é provavelmente o mais bem compreendido pela Biologia. Nesta seção será ignorada a química subjacente por questão de simplicidade.

Note que sistema de tomada de decisão da bactéria *E. coli* deve ter alguma capacidade de entender o conceito de derivadas e, portanto, deve possuir algum mecanismo básico de memória. À primeira vista, pode parecer que a bactéria usa o valor das concentrações em ambas as extremidades da célula e encontra uma diferença simples para reconhecer o gradiente (uma derivada espacial), entretanto, não é o caso. Experimentos têm mostrado que a bactéria usa um mecanismo de amostragem, e grosseiramente falando, ela se recorda de concentrações passadas, comparando-o com a situação atual, e toma decisões com base nesta diferença (ou seja, ela calcula algo como uma aproximação de Euler para uma derivada no tempo).

Em resumo, percebe-se que usando o mecanismo básico de memória, a habilidade de fazer comparações, algumas regras simples internas, sua composição química e a capacidade de locomoção, a bactéria é capaz de alcançar um comportamento complexo de busca e de rejeição. A evolução desenvolveu este sistema de controle. Tal sistema é robusto e muito bem sucedido em cumprir suas metas de sobrevivência quando visto de uma perspectiva da população.

### Eventos eliminativos-dispersivos

Eventos externos podem ocorrer no ambiente de forma que todas as bactérias em uma região podem ser mortas ou um grupo pode ser levado para uma outra região. Por exemplo, aumentos significativos na temperatura no local podem matar uma população de bactérias que estão numa região com alta concentração de nutrientes. Ou pode ser que um fluxo de água ou algum animal leve as populações de bactérias de um lugar para outro.

Qual é o efeito dos eventos eliminativos-dispersivos na quimiotaxia? Eles têm o efeito de eventualmente destruir o progresso quimiotático, mas também têm o efeito de ajudar na quimiotaxia, uma vez que a dispersão das bactérias pode colocar perto de fontes de boa comida. De uma perspectiva ampla, os eventos eliminativos-dispersivos são parte do comportamento de motilidade no nível de população e de longas distâncias.

## 1.2 Visão Geral do Algoritmo

Suponha que queremos encontrar o mínimo de  $J(\theta)$ ,  $\theta \in \mathbb{R}$ , quando não temos medições ou uma descrição analítica do gradiente  $\nabla J(\theta)$ . Para resolver o problema de otimização, pode-se usar as idéias do processo de alimentação de bactérias. Primeiro, suponha que  $\theta$  é a posição de uma bactéria e  $J(\theta)$  representa os efeitos combinados de atratores e repulsores do meio ambiente, como, por exemplo,  $J(\theta) < 0$ ,  $J(\theta) = 0$  e  $J(\theta) > 0$  representando que a bactéria em localizada

em um ambiente rico em nutrientes, neutro ou em um ambiente nocivo, respectivamente. Basicamente, quimiotaxia é um comportamento de alimentação que implementa um modelo de otimização, onde as bactérias tentam elevar a concentração de nutrientes (encontrar valores mais baixos de  $J(\theta)$ ), evitando substâncias nocivas, e caminhando para ambientes neutros (evitar estar em posições onde  $J(\theta) > 0$ ). O algoritmo BFA é, portanto, um tipo de busca aleatória guiada.

Nesta seção será descrito como o processo de busca de alimentos por bactérias pode ser modelado para obtenção de um algoritmo de otimização global. Inicialmente será feita uma relação entre os conceitos biológicos e um modelo passível de ser implementado, depois será apresentado uma proposta de algoritmo com pseudo-código. Depois, será feita uma discussão sobre a seleção dos parâmetros do algoritmo. E finalmente, será apresentada uma comparação entre esta técnica e outras abordagens para busca global.

### 1.2.1 Quimiotaxia, população, reprodução, eventos eliminativos e dispersivos

Define-se um passo quimiotático como um giro seguida de outro giro ou um giro seguido de uma corrida. Seja  $j$  o índice para a etapa quimiotática. Seja  $k$  o índice para a etapa de reprodução. Seja  $l$  o índice do evento eliminativo-dispersivo. A Eq. 1.2 representa a posição da colônia de bactérias  $S$  no  $j$ -ésimo passo quimiotático,  $k$ -ésimo passo de reprodução, e  $l$ -ésimo evento eliminativo-dispersivo. Aqui,  $J(i, j, k, l)$  denota o custo da  $i$ -ésima bactéria no local  $\theta^i(j, k, l) \in \mathbb{R}$  (relaxando a notação e removendo os índices pode-se usar  $\theta^i$  para se referir à posição  $i$ -ésima bactéria). Note que refere-se indistintamente a  $j$  como sendo um "custo" (usando a terminologia da teoria de otimização) ou como sendo uma superfície de nutrientes (em referência às ligações biológicas). Para populações de bactérias reais,  $S$  pode ser muito grande (por exemplo,  $S = 10^9$ ), mas  $p = 3$ . Nas simulações computacionais, usam-se tamanhos de populações muito menores e mantêm-se o tamanho da população fixo durante a execução. Também permite-se que  $p > 3$ , para que seja possível aplicar o método de otimização para problemas com muitas dimensões.

$$P(j, k, l) = \{\theta^i(j, k, l) | i = 1, 2, \dots, S\} \quad (1.2)$$

Seja  $N_c$  a duração do tempo de vida da bactéria, medida pelo número de etapas quimiotáticas ocorridas durante sua vida. Seja  $C(i) > 0, i = 1, 2, \dots, S$ , o tamanho de passo quimiotático básico que é usado para definir os comprimentos dos passos durante os movimentos de "corrida". Para representar as translações, é gerada uma unidade de comprimento de direção aleatória, digamos  $\phi(j)$ , que será utilizada para definir a direção do movimento após um giro. Em particular,  $C(i)$  será o tamanho do passo dado na direção aleatória especificada pelo giro. Se a partir de  $\theta^i(j + 1, k, l)$  o custo  $J(i, j + 1, k, l)$  é melhor (ou menor) do que em  $\theta^i(j, k, l)$ , então outro passo de tamanho  $C(i)$  neste mesmo sentido será tomado, e, novamente, se essa etapa resulta em uma posição com um valor mais barato



do que na etapa anterior, outro passo é obtido. Este movimento é continuado enquanto o custo for reduzido, mas só até um número máximo de passos,  $N_s$ . Isso representa que a célula tende a se manter em movimento e na direção de ambientes cada vez mais favoráveis.

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\phi(j) \quad (1.3)$$

A discussão acima foi para o caso simples em que não há sinalizações entre as células para que elas caminhem em conjunto no enxame. Mas há também a possibilidade de usar atratores de célula para célula com o objetivo de sinalização, que será dado por  $J_{cc}^i(\theta, \theta(j, k, l))$ ,  $i = 1, 2, \dots, S$ , para a  $i$ -ésima bactéria. Seja

$$d_{atrator} = 0, 1$$

a quantificação da porção de atrator que é liberado pela célula e seja

$$w_{atrator} = 0, 2$$

uma medida da largura do sinal do atrator (a quantificação da taxa de difusão do produto químico). A célula também pode repelir uma outra célula vizinha. Para modelar este aspecto, considere

$$h_{repulsor} = d_{atrator}$$

como sendo a altura do efeito repulsor (magnitude do efeito) e

$$w_{repulsor} = 10$$

como sendo uma medida da largura do repulsor. Os valores para esses parâmetros são escolhidos apenas para representar o comportamento geral das bactérias e não o esquema de sinalização química de alguma bactéria específica. Os valores particulares dos parâmetros foram escolhidos com o perfil nutricional da Fig. 1.3 em mente. Por exemplo, a profundidade e a largura do atrator é pequena em relação à concentração de nutrientes representada na Fig. 1.3. Considere que

$$\begin{aligned} J_{cc}(\theta, P(j, k, l)) &= \sum_{i=1}^S J_{cc}^i(\theta, \theta(j, k, l)) \\ &= \sum_{i=1}^S \left[ -d_{atrator} \exp \left( -w_{atrator} \sum_{m=1}^p (\theta_m - \theta_m^i)^2 \right) \right] \\ &+ \sum_{i=1}^S \left[ -h_{repulsor} \exp \left( -w_{repulsor} \sum_{m=1}^p (\theta_m - \theta_m^i)^2 \right) \right] \end{aligned}$$

denota os efeitos combinados de atração e repulsão entre as células, onde  $\theta = [\theta_1, \dots, \theta_p]^T$  é um ponto no domínio de otimização e  $\theta_m^i$  é o  $m$ -ésimo componente da  $i$ -ésima posição da bactéria  $\theta^i$  (por conveniência, pode-se omitir alguns

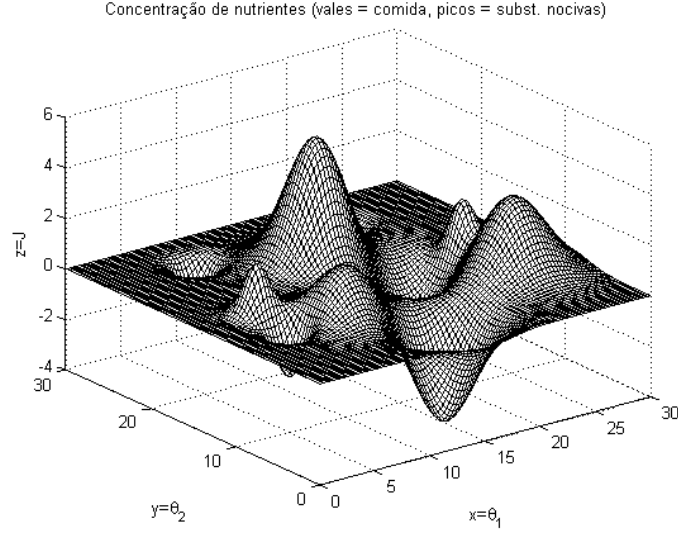


Figura 1.3: Gráfico de concentração de nutrientes.

dos índices). Um exemplo para o caso de  $S = 2$  e os valores dos parâmetros acima é mostrado na Fig. 1.4. Note que os dois picos afiados representam as localizações de células, e à medida que se move radialmente para fora da célula, o valor da função diminui e, em seguida, aumenta (modelando o fato de que as células que estão distantes não tendem a ser atraídas, enquanto que as células que estão perto tendem a tentar descer o gradiente de nutrientes entre as células, um para o outro e, portanto, formar o enxame). Para estudar o efeito do enxame, a  $i$ -ésima bactéria,  $i = 1, 2, \dots, S$ , irá executar uma subida de encosta usando

$$J(i, j, k, l) + J_{cc}(\theta, P)$$

(em vez do  $J(i, j, k, l)$  definido acima) para que as células tentem encontrar nutrientes, evitar substâncias nocivas, e ao mesmo tempo, tentar se mover em direção a outras células, mas não tão próximas umas das outras. A função  $J_{cc}(\theta, P)$  dinamicamente altera o espaço de busca a medida que as células se movem para formando o enxame (ou seja, o mecanismo de enxames é modelado como um processo de minimização).

Após  $N_c$  passos quimiotáticos, um passo reprodutivo é executado. Seja  $N_{re}$  o número de passos de reprodução serem executados. Por conveniência supomos que  $S$  é um número inteiro positivo e par. Seja

$$S_r = \frac{S}{2} \tag{1.4}$$

o número de membros da população que possuem nutrientes suficientes para

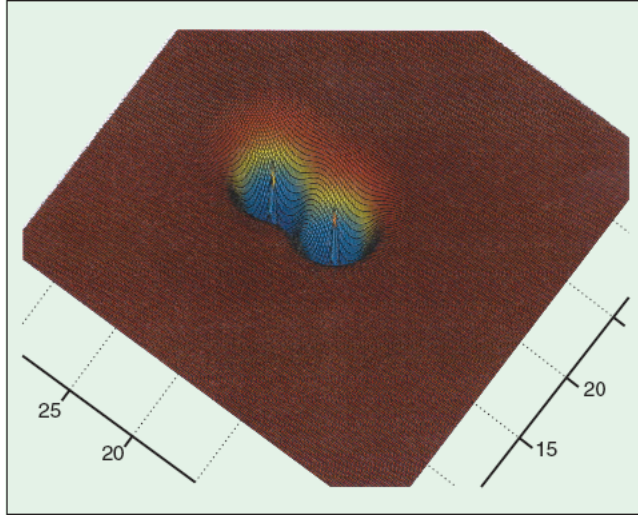


Figura 1.4: Modelo de atração química entre células.

reprodução (divisão em dois) sem mutação. Para a reprodução, a população é classificada em ordem crescente de custo acumulado (maior custo acumulado representa que uma bactéria não obteve muitos nutrientes durante seu ciclo de vida e, conseqüentemente, não é tão "saudável" e, portanto, pouco provável de se reproduzir). As  $S_r$  bactérias menos saudáveis morrem e as  $S_r$  bactérias mais saudáveis são divididas em duas bactérias e são colocados no mesmo ponto. Outras funções podem ser usadas no lugar da Eq. 1.4, este método recompensa bactérias que encontraram uma grande quantidade de nutrientes e permite manter uma população constante, o que é conveniente na implementação do algoritmo.

Seja  $N_{ed}$  o número de eventos eliminativos-dispersivos, e para cada evento eliminativo-dispersivo uma bactéria da população está sujeita à eliminação e dispersão com uma probabilidade  $p_{ed}$ . Em concordância com o modelo biológico, assume-se que a frequência de passos quimiotáticos é maior do que a frequência de passos de reprodução, que por sua vez é maior do que a frequência de eventos eliminativos-dispersivos.

A Fig. 1.5 ilustra a posição das bactérias durante uma execução em cada geração, usando  $S = 50$  e  $N_{re} = 4$ .

### 1.2.2 O algoritmo passo-a-passo

Para a inicialização, deve-se escolher  $P$ ,  $S$ ,  $N_c$ ,  $N_S$ ,  $N_{RE}$ ,  $N_{ed}$ ,  $p_{ed}$ , e  $C(i)$ ,  $i = 1, 2, \dots, S$ . Vide seção 1.2.3 para uma descrição de cada parâmetro e sugestões para os valores. Se é usado o conceito de enxame, deve-se escolher os parâmetros das funções de atração célula a célula. Além disso, deve-se escolher os valores iniciais de  $\theta^i$ ,  $i = 1, 2, \dots, S$ . Pode ser desejável distribuí-los aleatoriamente em todo

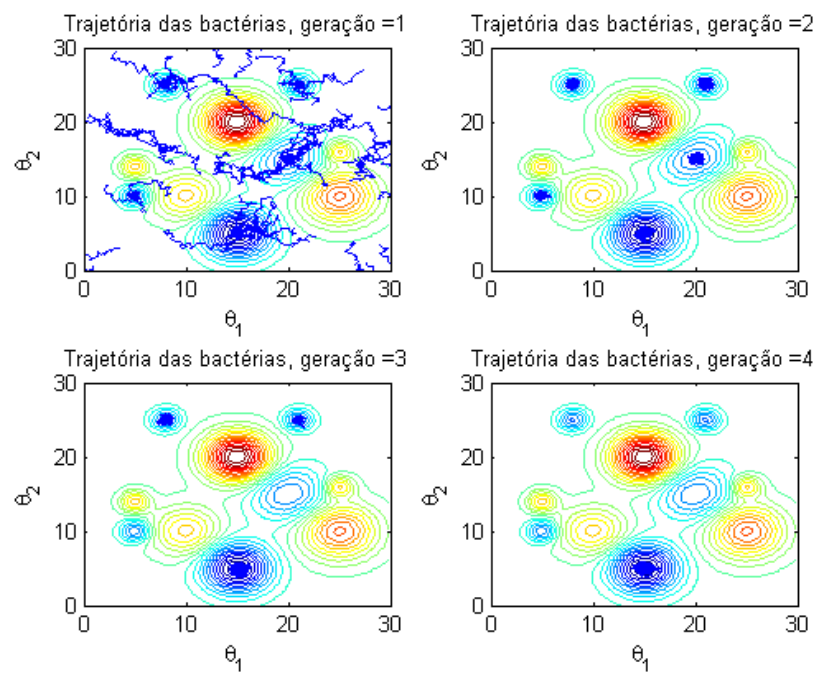


Figura 1.5: Execução do algoritmo durante as gerações  $N_{re} = 4$ .

o espaço de busca do domínio do problema. Abaixo é apresentado o pseudo-código do algoritmo que modela a quimiotaxia, movimentação, reprodução, e os eventos eliminativos-dispersivos de bactérias (inicialmente,  $j = k = l = 0$ ). Para este algoritmo, note que atualizações em  $\theta^i$  automaticamente resultam em atualizações para  $P$ . E finalmente, é possível adicionar um critério de parada mais sofisticado do que simplesmente especificar um número máximo de iterações. Na Fig. 1.6 é possível visualizar um fluxograma representativo deste pseudo-código.

Pseudo-código do algoritmo BFA:

1. Laço de eventos eliminativo-dispersivos:  $l = l + 1$
2. Laço de reprodução:  $k = k + 1$
3. Laço quimiotático:  $j = j + 1$ 
  - (a) Para  $i = 1, 2, \dots, S$ , execute um passo quimiotático para cada bactéria  $i$  como segue.
  - (b) Calcule  $J(i, j, k, l)$ . Faça:

$$J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), P(j, k, l)).$$

Isto é, adicione o efeito de atração célula a célula na concentração de nutrientes.

- (c) Faça  $J_{ultimo} = J(i, j, k, l)$  para armazenar este valor visto que podemos um melhor custo durante a corrida.
- (d) Giro: gerar um vetor randômico  $\Delta(i) \in R^p$  com cada elemento  $\Delta_m(i)$ ,  $i = 1, 2, \dots, p$  ( $p$  é um número randômico entre  $[-1, 1]$ ).
- (e) Mover: Seja

$$\theta^i(j + 1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}}.$$

Isto resultará em um passo de tamanho  $C(i)$  na direção especificada pelo passo anterior para a bactéria  $i$ .

- (f) Calcule  $J(i, j + 1, k, l)$  e então faça:

$$J(i, j + 1, k, l) = J(i, j + 1, k, l) + J_{cc}(\theta^i(j + 1, k, l), P(j + 1, k, l))$$

- (g) Nadar (é feita uma simplificação a medida que uma bactéria  $1, 2, \dots, i$  é movida mas a bactéria  $i + 1, i + 2, \dots, S$  não é, simulando que as decisões de "correr" ou "transladar" são executadas simultaneamente por todas as bactérias).
  - i. Faça  $m = 0$  (um contador para o comprimento do passo).
  - ii. Enquanto  $m < N_S$ .
    - A. Faça  $m = m + 1$ .

- B. Se  $J(i, j+1, k, l) < J_{last}$  (se melhorou), faça  $J_{last} = J(i, j+1, k, l)$  e faça:

$$\theta^i(j+1, k, l) = \theta^i(j+1, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}}.$$

e use este  $\theta^i(j+1, k, l)$  para calcular o novo valor de  $J(i, j+1, k, l)$ .

- C. Senão, faça  $m = N_S$  (fim do bloco "enquanto").

- (h) Ir para a próxima bactéria ( $i+1$ ) se  $i \neq S$  (isto é, ir para b) para tratar a próxima bactéria.
4. Se  $j < M_c$ , ir para o passo 3. Neste caso continue a quimiotaxia, visto que a bactéria ainda vive.
5. Reprodução:

- (a) Para um dado  $k$  e  $l$ , e para cada  $i = 1, 2, \dots, S$ :

$$J_{health}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l)$$

será a saúde da bactéria  $i$  (uma medida de quantos nutrientes foram acumulados durante sua vida e quanto a bactéria evitou substâncias nocivas). Classifique as bactérias e parâmetros quimiotáticos  $C(i)$  em ordem ascendente do custo  $J_{health}$  (maior custo significará baixa saúde).

- (b) As bactérias  $S_r$  com os maiores valores de  $J_{health}$  morrem e as outras bactérias  $S_r$  com melhores valores dividem-se (e as cópias que são feitas são colocados no mesmo local dos pais).
6. Se  $k < Nre$ , ir para o passo 2. Neste caso, ainda não foi alcançado o número de passos de reprodução especificados, então inicia-se a próxima geração no laço quimitático.
7. Eliminação-dispersão: para  $i = 1, 2, \dots, S$ , com probabilidade  $p_{ed}$ , elimine ou disperse cada bactéria (deve manter o número de bactérias na população constante). Pode-se simplesmente dispersar as bactérias para uma localização aleatória do espaço de busca.
8. If  $l < N_{ed}$ , ir para o passo 1; senão finalizar a execução.

### 1.2.3 Seleção dos parâmetros

O algoritmo BFA requer a especificação de uma quantidade considerável de parâmetros. Vide a tabela 1.2 para visualizar a lista completa de parâmetros

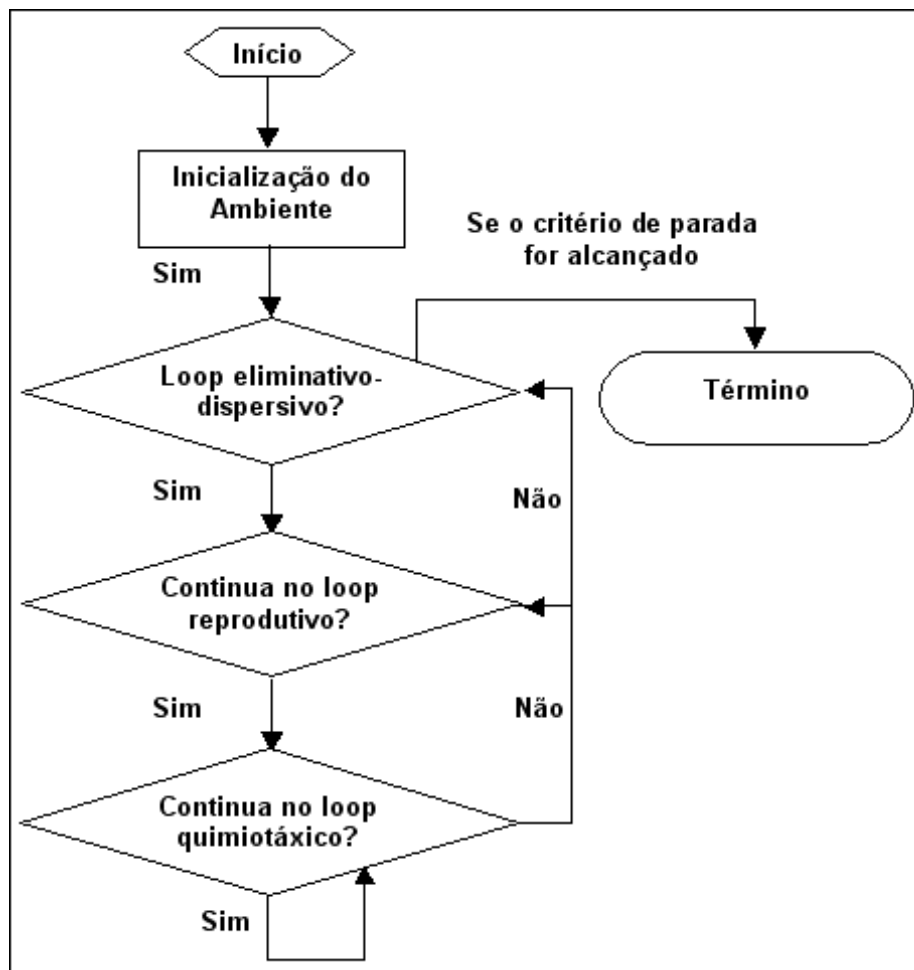


Figura 1.6: Fluxograma de execução do algoritmo BFA.

Tabela 1.2: Parâmetros de inicialização do algoritmo BFA padrão.

$P$	Posição de bactérias no gradiente nutritivo
$S$	Número de bactérias da colônia
$N_c$	Número de passos quimiotáticos
$N_S$	Número de passos de "corrida"
$N_{RE}$	Número de passos reprodutivos
$N_{ed}$	Número de passos de eliminação/dispersão
$p_{ed}$	Probabilidade de eliminação/dispersão
$C(i), i = 1, 2, \dots, S$	O tamanho de um passo de "corrida" e "giro"

do BFA original. Primeiro, deve-se escolher o tamanho da população,  $S$ . Claramente, aumentando o tamanho de  $S$  pode aumentar significativamente a complexidade computacional do algoritmo. No entanto, para maiores valores de  $S$ , se a população inicial é distribuída aleatoriamente, é mais provável que pelo menos algumas bactérias estarão perto de um ponto ótimo, e ao longo do tempo, é mais provável que muitas bactérias estarão também nessa região, seja devido à quimiotaxia ou à reprodução.

Para os valores de  $C(i), i = 1, 2, \dots, S$  pode-se escolher um valor motivado biologicamente, no entanto, tais valores não podem ser o melhor para uma aplicação de engenharia. Se os valores de  $C(i)$  são muito grandes e se o ótimo global está em um vale com bordas íngremes, a busca tende a saltar para fora do vale, ou pode simplesmente não perceber mínimos locais (passará direto dos pontos sem parar). Por outro lado, se os valores de  $C(i)$  forem muito pequenos, a convergência pode ser lenta e o algoritmo poderá ficar preso em mínimos locais. Deve-se pensar em  $C(i)$  como um ajuste da granularidade dos passos para o algoritmo de otimização.

Os valores dos parâmetros que definem as funções de atração-repulsão célula a célula  $J_{cc}^i$  irão definir as características do enxame. Se a largura e altura do fator atrator forem muito grande, as células têm uma forte tendência para permanecerem juntas (podem até evitar ir atrás de nutrientes em novas áreas para favorecer o efeito de enxame). Por outro lado, se a largura do fator de atração é pequena e de baixa profundidade, haverá grande tendência para dispersão do enxame e cada célula irá procurar por conta própria. A busca de alimentos de forma social ou independente é, então, ditada pelo equilíbrio entre as forças dos sinais de atração entre as células e a concentração de nutrientes.

Grandes valores para  $N_c$  resultam em muitos passos quimiotáticos, na esperança de um progresso maior para otimização, mas aumentando obviamente a complexidade computacional. Se o tamanho de  $N_c$  é muito pequeno, o algoritmo geralmente depende mais do acaso e da reprodução, e, em alguns casos, ficará facilmente preso em mínimos locais (convergência prematura). Deve-se pensar de  $N_s$  como um peso para a busca aleatória (o que não ocorreria se  $N_s = 0$ ).

Se  $N_c$  é grande o suficiente, o valor de  $N_{re}$  afeta a forma como o algoritmo ignora regiões ruins e concentra-se em boas, uma vez que as bactérias em regiões relativamente pobres em nutrientes morrem (nestes modelos, com



uma população fixa). Se  $N_{re}$  é muito pequeno, o algoritmo pode convergir prematuramente, no entanto, valores maiores de  $N_{re}$  claramente aumentam a complexidade computacional.

Um valor baixo para  $N_{ed}$  informa que o algoritmo não contará com eventos aleatórios de eliminação e dispersão para tentar encontrar regiões favoráveis. Um valor alto aumenta a complexidade computacional, mas permite que as bactérias busquem em mais regiões para encontrar uma boa concentração de nutrientes. Claramente, se  $p_{ed}$  é grande, o algoritmo pode passar a se comportar como uma busca exaustiva aleatória. Se, no entanto, ele é escolhido de forma adequada, pode ajudar o algoritmo para fugir de ótimos locais para alcançar um ótimo global.

### 1.2.4 Comparação com outros métodos de otimização

O leitor já deve estar familiarizado com algoritmos genéticos (ou computação evolucionária em geral) e pode fazer uma analogia entre alguns conceitos. Como entre a função de aptidão e a função de concentração de nutrientes, a seleção e a reprodução bacteriana (bactérias em ambientes favoráveis ganham vantagem seletiva para a reprodução), crossover e divisão bacteriana (os descendentes estarão com a mesma concentração de nutrientes, e no caso de crossover geralmente os filhos acabam em uma região em torno de seus pais), e entre a mutação e os eventos eliminativos-dispersivos. No entanto, os algoritmos não são totalmente equivalentes, e nem um é um caso especial do outro. Cada técnica tem suas próprias características. A função de fitness e funções de concentração de nutrientes não exatamente iguais (a primeira representa a probabilidade de sobrevivência de determinado características fenotípicas, enquanto a segunda representa as concentrações de nutriente / substâncias nocivas, ou talvez outras influências ambientais, como calor ou luz). Crossover representa as diferenças resultantes de cruzamentos e da descendência, algo que pode ser ignorado no BFA (obtem-se cópias perfeitas das bactérias para representar sua divisão). Além disso, a mutação representa a mutação genética e as alterações fenotípicas resultantes, não dispersão física em uma área geográfica.

Uma variante do algoritmo descrito acima, denominada BSA, foi proposta em [27], apresentando um estudo comparativo da otimização por enxames de bactérias com outros métodos de otimização global, usando funções de *benchmark* conhecidas. O resultado desse estudo comparativo está sumarizado na Tabela 1.3. Outros estudos comparativos podem ser encontrados em 1.3.3.

## 1.3 Aplicações

### 1.3.1 Onde Usar

Desde a primeira versão do BFA básico, vários algoritmos têm sido desenvolvidos. O Bacteria Chemotaxis (BC) usa também o comportamento quimiotático das bactérias e foi desenvolvido por Muller et al. [24] para problemas de otimização

Tabela 1.3: Comparação entre técnicas de otimização global.

Função	FEP		PSO		BSA	
	Solução	Desvio	Solução	Desvio	Solução	Desvio
$f_1$	32,532	18,700	30,139	25,818	2,466	3,059
$f_2$	0	0	0,1	0,316	0,15	0,671
$f_3$	$7,4 \times 10^4$	$2,319 \times 10^4$	$2,5 \times 10^3$	$5,28 \times 10^4$	$8,720 \times 10^4$	$7,528 \times 10^3$
$f_4$	-11397	364	-7467	980,7	-12567,5	3,25
$f_5$	13,92	3,958	56,537	19,615	1,505	6,672
$f_6$	0,4482	0,3869	$2,32 \times 10^{-9}$	$1,33 \times 10^{-9}$	$6,87 \times 10^{-2}$	$7,78 \times 10^{-2}$
$f_7$	1,197	0,419	1,395	0,694	0,998	0

global. Biswas et al. [4] desenvolveram um algoritmo híbrido que acopla características dos algoritmos BFO e PSO. Este algoritmo foi denominado Bacterial Swarm Optimization (BSO) e apresentou resultados promissores. Tang et al. [27] propuseram um outro algoritmo, incorporando modelagens de estudos recentes sobre padrões de mobilidade de bactérias, denominado Bacterial Swarming Algorithm (BSA). Korani [17] desenvolveu também um algoritmo híbrido usando BFO e PSO no mecanismo de eliminação e dispersão para a descoberta de novas soluções. O algoritmo foi denominado *Bacterial Foraging Optimization oriented by Particle Swarm Optimization (BF-PSO)* e aplicado inicialmente ao problema de sintonia de controladores PID.

O algoritmo BFO tem sido aplicado com sucesso na sintonia de controladores adaptativos e na otimização de funções [22], no ajuste dos ganhos de controlador PID [3, 16, 21, 29], no controle de manipulador robótico [6], na identificação de sistemas dinâmicos não-lineares [19], estimação de sinal [1], na redução da perda de transmissão em sistemas de potência [20], em problemas dinâmicos de otimização de custos em sistemas de potência [28], na estimação de parâmetros [7], no controle de antenas [10, 11], no despacho econômico de carga [23], no processamento de imagens [18].

Gazi e Passino [9] estudaram a estabilidade de colônias de bactérias. Kim et al. [15] propuseram uma abordagem híbrida envolvendo algoritmos genéticos (AG) e BFO para otimizações de funções. O algoritmo híbrido superou o AG e o BFO em algumas referências comparativas numéricas e em um problema prático de projeto de sintonia de controladores PID. Chen et al. [5] apresentaram uma variação do algoritmo BFO original através da introdução de abordagens cooperativas, obtendo uma melhora significativa em suas velocidade de convergência, precisão e robustez.

### 1.3.2 Estudo de Caso

Nesta seção será ilustrado a aplicação do BFA na resolução de um problema prático, com um estudo paramétrico sobre a interferência dos valores adotados para o problema em questão.

### 1.3.3 Exercícios

1. Faça isso.
2. Faça aquilo.
3. Faça aquilo outro.



# Referências Bibliográficas

- [1] Panda G. Mishra S. Lakshmi Y.V.S. Acharya, D.P. Bacteria foraging based independent component analysis. *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, pages 527–531, 2008.
- [2] J.E. Smith Agoston E. Eiben. *Introduction to Evolutionary Computation*. Springer, 2003.
- [3] Majhi S. Ali, A. Design of optimum pid controller by bacterial foraging strategy. *Proceedings of the IEEE International Conference on Industrial Technology (ICIT2006)*, pages 601–605, 2006.
- [4] Swagatam Das Ajith Abraham Arijit Biswas, Sambarta Dasgupta. Synergy of pso and bacterial foraging optimization. a comparative study on numerical benchmarks. *Innovations in Hybrid Intelligent Systems*, pages 255–263, 2007.
- [5] Zhu Y.; Hu K. He X. Niu B. Chen, H. Cooperative approaches to bacterial foraging optimization. *Lecture Notes in Artificial Intelligence*, 5227:541–548, 2008.
- [6] Silveira C.C. Coelho, L.S. Improved bacterial foraging strategy for controller optimization applied to robotic manipulator system. *Proceedings of the 2006 IEEE 6th International Symposium on Intelligent Control*, pages 1276–1281, 2006.
- [7] A.; O'Neill M.; Edelman D. (2008). Dang, J.; Brabazon. Option model calibration using a bacterial foraging optimization algorithm. *Lecture Notes in Computer Science*, 4974:113–122, 2008.
- [8] Andries P. Engelbrecht. *Computation Intelligence*. John Wiley & Sons, 2007.
- [9] Passino K.M. Gazi, V. Stability analysis of social foraging swarms. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 34:539–557, 2004.
- [10] Basbug S. Guney, K. Interference suppression of linear antenna arrays by amplitude-only control using a bacterial foraging algorithm. *Progress in Electromagnetics Research*, 79:475–497, 2008.

- [11] Basbug S. Guney, K. Phase-only pattern nulling of linear antenna arrays with the use of a bacterial foraging algorithm. *Neural Network World*, 18:257–273, 2008.
- [12] Kunyuan Hu Hanning Chen, Yunlong Zhu. Cooperative bacterial foraging optimization. *Discrete Dynamics in Nature and Society*, page 17, 2009.
- [13] Yuhui Shi James Kennedy, Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.
- [14] B. S. Sohi S. Devi M.R. Lohakare K. M. Bakwad, S.S. Pattnaik. Parallel bacterial foraging optimization for video compression. *International Journal of Recent Trends in Engineering*, 1:118–122, 2009.
- [15] Abraham A. Cho J.H. Kim, D.H. A hybrid genetic algorithm and bacterial foraging approach for global optimization. *Information Sciences*, 177:3918–3937, 2007.
- [16] Cho J.H. Kim, D.H. Adaptive tuning of pid controller for multivariable system using bacterial foraging based optimization. *Advances in Web Intelligence. In: Lecture Notes in Computer Science*, 3528:231–235, 2005.
- [17] W. Korani. Bacterial foraging oriented by particle swarm optimization strategy for pid tuning. *Proceedings of GECCO 2008: Genetic and Evolutionary Computation Conference, Atlanta, Georgia*, pages 1823–1826, 2008.
- [18] Chatterjee A. Maitra, M. A novel technique for multilevel optimal magnetic resonance brain image thresholding using bacterial foraging. *Journal of the International Measurement Confederation*, 41:1124–1134, 2008.
- [19] Panda G. Majhi, B. Bacterial foraging based identification of nonlinear dynamic system. *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pages 1636–1641, 2007.
- [20] S. Mishra. Bacteria foraging based solution to optimize both real power loss and voltage stability limit. *Proceedings of the 2007 IEEE Power Engineering Society General Meeting, PES*, 2007.
- [21] Zhu Y. He X. Zeng X. Niu, B. Optimum design of pid controllers using only a germ of intelligence. *Proceedings of the 6th World Congress on Intelligent Control and Automation*, pages 3584–3588, 2006.
- [22] Kevin M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, pages 52–67, 2002.
- [23] Venayagamoorthy G.K. Saber, A.Y. Economic load dispatch using bacterial foraging technique with particle swarm optimization biased evolution. *2008 IEEE Swarm Intelligence Symposium*, 2008.

- [24] Stefano Airaghi Petros Koumoutsakos Sibylle D. Müller, Jarno Marchetto. Optimization based on bacterial chemotaxis. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 6:16–29, 2002.
- [25] I. S. Misra T. Datta. Improved adaptive bacteria foraging algorithm in optimization of antenna array for faster convergence. *Progress In Electromagnetics Research*, 1:143–157, 2008.
- [26] J. R. Saunders W. J. Tang, Q. H. Wu. Bacterial foraging algorithm for dynamic environments. *2006 IEEE Congress on Evolutionary Computation*, pages 1324–1330, 2006.
- [27] J. R. Saunders W. J. Tang, Q. H. Wu. A bacterial swarming algorithm for global optimization. *IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 1207–1212, 2007.
- [28] J. R. Saunders W. J. Tang, Q. H. Wu. Bacterial foraging algorithm for optimal power flow in dynamic environments. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS*, 55:2433–2442, 2008.
- [29] Yanjun F. Yijian, L. Optimization design of pid controller parameters based on improved e.coli foraging optimization algorithm. *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL 2008)*, pages 227–231, 2008.