**Algoritmo de otimização de colônia de formigas com mecanismo de mutação e suas aplicações**

O algoritmo de otimização de colônia de formigas mutadas (MACO) é proposto introduzindo o mecanismo de mutação no algoritmo ACO e é aplicado ao problema do caixeiro viajante (TSP) e detecção multiusuário neste artigo. Os algoritmos de otimização de colônia de formigas (ACO) já foram usados ​​com sucesso na otimização combinatória, no entanto, conforme o feromônio se acumula, podemos não obter um ótimo global porque ele pode ficar preso em um mínimo local, resultando em um estado estável ruim. O algoritmo MACO apresentado pode ampliar o alcance de busca e evitar mínimos locais, alterando aleatoriamente um ou mais elementos da melhor solução local, que é a operação de mutação no algoritmo genético. Como a operação de mutação é simples de implementar, o desempenho do MACO é superior com quase a mesma complexidade computacional. O MACO é aplicado à detecção de TSP e multiusuário e, por meio de simulações de computador, é mostrado que o MACO tem um desempenho muito melhor na solução desses dois problemas do que os algoritmos ACO

**Introdução**

A inteligência de enxame é uma abordagem relativamente nova para a solução de problemas que se inspira no comportamento social de insetos e de outros animais. Em particular, as formigas inspiraram uma série de métodos e técnicas, entre os quais o mais estudado e o de maior sucesso é a técnica de otimização de uso geral conhecida como otimização de colônia de formigas (ACO). Algoritmos ACO se inspiram no comportamento de forrageamento de algumas espécies de formigas (Goss, Aron, Deneubourg, & Pasteels, 1989). Essas formigas depositam feromônio no solo para marcar algum caminho favorável que deve ser seguido por outros membros da colônia. ACO explora um mecanismo semelhante para resolver problemas de otimização (Dorigo, Birattari, & Stützle, 2006; Sim & Sun, 2003).

O primeiro algoritmo ACO, o sistema de formigas (AS), é proposto como meio de resolver o problema do caixeiro viajante (TSP) (Dorigo, Maniezzo, & Colorni, 1996). AS obteve um grande sucesso na resolução de problemas de otimização combinatória, no entanto, seu desempenho ainda é pior do que alguns outros algoritmos metaheurísticos (Laguna & Glover, 1993; Zhen-Ping & Bavarian, 1992). Assim, muitos outros algoritmos ACO são propostos inspirados no AS, cujo desempenho é aprimorado notavelmente (Dorigo et al., 2006). Os principais algoritmos ACO apresentados na literatura são: ant-Q (Dorigo & Gambardella, 1996), sistema de colônia de formigas (ACS) (Dorigo & Gambardella, 1997), sistema de formigas MAX – MIN (MMAS) (Stützle & Hoos, 2000) , sistema de formigas baseado em classificação (Bullnheimer, Hartl, & Strauss, 1999), ANTS (Maniezzo, 1999), sistema de formigas hipercubo (Blum, Roli, & Dorigo, 2001), KCCAnts (Naimi & Taherinejad, 2009), e PDACO (Wu, Zhao, Ren e Quan, 2009). O desenvolvimento de metodologias bioinspiradas com base em sistemas de algoritmos inspirados em colônias de formigas é uma área de pesquisa emergente com aplicações em áreas como robótica (Lerman, Galstyan, Matinolli, & Ijspeert, 2002), problemas de atribuição quadrática (Colorni, Dorigo, & Maniezzo, 1991), TSP (Li & Gong, 2003) e seleção de subconjunto de recursos (Sivagaminathan & Ramakrishnan, 2007).

Os algoritmos ACO mais bem sucedidos são MMAS e ACS. Embora o MMAS e o ACS possam superar as desvantagens do AS e obter um desempenho muito melhor, ele pode ficar preso em um mínimo local, resultando em um estado estável ruim conforme o feromônio se acumula. Algoritmo Genético (GA) é uma ferramenta poderosa para resolver problemas de otimização combinatória, e muitos novos algoritmos ACO foram propostos pela introdução do GA aos algoritmos ACO tradicionais e atingir um desempenho muito melhor (Kaveh & Shahrouzi, 2008; Lee, Su, Chuang, & Liu, 2008). No entanto, os algoritmos ACO baseados em GA são extremamente complicados e demorados. Portanto, neste artigo, apenas a operação de mutação simples do GA é introduzida no algoritmo ACO, e o algoritmo de otimização de colônia de formigas mutadas (MACO) é proposto, o que pode melhorar a capacidade de busca global com quase a mesma complexidade computacional do algoritmo ACO. O algoritmo MACO apresentado pode ampliar o alcance de busca e evitar mínimos locais, alterando aleatoriamente um ou mais elementos da melhor solução local, que é a operação de mutação no algoritmo genético. Portanto, o desempenho do MACO é muito melhor do que os algoritmos ACO correspondentes com quase a mesma complexidade computacional. Pois o MACO não mudará a função de trabalho do ACO, ele pode ser combinado com muitos algoritmos do ACO para melhorar o desempenho deles. Assim, MMAS mutado (M-MMAS) e ACS mutado (M-ACS) são propostos combinando MACO com MMAS e ACS respectivamente, e então aplicados ao TSP. Por meio de simulações mostra-se que M-MMAS e M-ACS apresentam melhor desempenho na resolução de TSP do que seus correspondentes algoritmos ACO.

O Acesso Múltiplo por Divisão de Código (CDMA) tem sido objeto de extensa pesquisa no campo das comunicações de rádio móvel. Esta técnica permite que um grande número de usuários se comuniquem simultaneamente na mesma banda de frequência; no entanto, também cria interferência de acesso múltiplo (MAI). O MAI fabrica o detector convencional (CD), que pode demodular apenas um sinal de espalhamento espectral sem considerar outros sinais, não confiável e insensível ao efeito próximo-distante em um ambiente multiusuário. Por essa razão, a detecção de multiusuário, que pode superar esse problema, é um tópico quente agora para sistemas CDMA (Moshavi, 1996). O detector multiusuário ótimo (OMD) (Verdu, 1986) proposto por Verdu, se mostra resistente e tem desempenho ótimo, porém, a complexidade exponencial no número de usuários o torna inviável para uso nos sistemas CDMA atuais. Portanto, os esforços de pesquisa têm se concentrado no desenvolvimento de detectores subótimos, que exibem boas propriedades resistentes ao efeito de quase-longe, têm baixa complexidade computacional e alcançam desempenho relativamente alto, como o detector MMSE (Xie, Short, & Rushforth, 1990), Hopfield detector de rede neural (Kechriotis & Manolakos, 1996) e detector de rede neural celular estocástica (Wu, Zhao, Zhao, & Ren, 2007).

O ACO também pode ser usado na detecção multiusuário como uma espécie de detectores subótimos, nos quais a duração do passeio no TSP está relacionada à função objetivo do OMD (Hijazi & Natarajan, 2004). Neste artigo, o detector multiusuário MACO é proposto aplicando o algoritmo MACO à detecção multiusuário. Por meio de simulações, é mostrado que o detector multiusuário MACO tem um desempenho muito melhor na redução do efeito de perto-longe do que o detector multiusuário ACO e detector multiusuário PDACO, que foi proposto em nosso artigo anterior (Wu et al., 2009), como bem como um desempenho superior na taxa de erro de bits (BER). O restante deste artigo está organizado em três seções. Na Seção 2, algumas preliminares sobre ACO são revisadas e MACO é apresentado com suas características principais. Na Seção 3, MACO é combinado com ACS e MMAS e aplicado ao TSP. Os resultados da simulação também são apresentados. Na Seção 4, o MACO é aplicado à detecção de multiusuário. O desempenho do detector multiusuário MACO é comparado com o detector multiusuário ACO, detector multiusuário PDACO e alguns outros detectores.

**2. Algoritmos de otimização de colônia de formigas**

Muitos algoritmos ACO foram propostos. Aqui apresentamos o AS original, suas duas variantes de maior sucesso: MMAS e ACS, e o algoritmo MACO proposto. Para ilustrar as diferenças entre esses algoritmos, usamos o TSP como um exemplo concreto.

**2.1. Sistema formiga**

AS é o primeiro algoritmo ACO proposto. Sua principal característica é que, após cada iteração, os valores dos feromônios são atualizados por todas as M formigas que construíram soluções. O feromônio sij, associado à borda que une as cidades i e j, é atualizado da seguinte forma:

Na construção de uma solução, as formigas selecionam a seguinte cidade para ser visitada por meio de um mecanismo estocástico. Quando a formiga m está na cidade i e até agora construiu a solução parcial s p, a probabilidade de ir para a cidade j é dada por:

**4 equações**

AS obteve um grande sucesso na resolução de TSP, no entanto, como a escala de TSP aumenta, o desempenho de AS diminui seriamente em comparação com outros algoritmos metaheurísticos. Portanto, a maior parte da pesquisa sobre ACO tem se concentrado nos métodos para melhorar a AS. Os mais bem-sucedidos são ACS e MMAS.

**2.2. Sistema de formiga MAX – MIN**

MMAS é um aprimoramento do AS original. Seus elementos caracterizadores são que apenas a melhor formiga atualiza as trilhas de feromônio e que o valor do feromônio é limitado. A atualização do feromônio é implementada da seguinte forma:

**3 equações**

onde Tmax e Tmin são, respectivamente, os limites superior e inferior impostos ao feromônio; o operador X (ab) é definido como:

onde Lbest é a distancia do passeio da melhor formiga. Este pode ser o melhor passeio encontrado na iteração atual (iteration-best, L(ib)) ou a melhor solução encontrada desde o início do algoritmo (best-so-far, L(bs)) ou uma combinação de ambos.

**2.3. Sistema de colônia de formigas**

As contribuições mais interessantes do ACS são a introdução de uma atualização de feromônio local, além da atualização de feromônio realizada no final do processo de construção (chamada atualização de feromônio offline).

A atualização local do feromônio é realizada por todas as formigas após cada etapa de construção. Cada formiga se aplica apenas à última aresta atravessada:

**1 equação**

onde SIGMA é o coeficiente de decaimento do feromônio, e s0 é o valor inicial do feromônio

O principal objetivo da atualização local é diversificar a busca realizada pelas formigas subsequentes durante uma iteração: ao diminuir a concentração de feromônios nas bordas atravessadas, as formigas encorajam as formigas subsequentes a escolher outras bordas e, portanto, a produzir soluções diferentes. Isso torna menos provável que várias formigas produzam soluções idênticas durante uma iteração.

A atualização do feromônio offline, de forma semelhante ao MMAS, é aplicada no final de cada iteração por apenas uma formiga, que pode ser a melhor iteração ou a melhor até agora. No entanto, a fórmula de atualização é um pouco diferente:

**1 equação**

Outra diferença importante entre ACS e AS está na regra de decisão usada pelas formigas durante o processo de construção. No ACS, a chamada regra proporcional pseudoaleatória é usada: a probabilidade de uma formiga se mover da cidade i para a cidade j depende de uma variável aleatória q uniformemente distribuída em [0,1] e de um parâmetro q0; se q 6 q0, então j ¼ arg maxcil2NðspÞfsilgb ilg, caso contrário, Eq. (3) é usado.

**2.4. Otimização de colônia de formigas mutantes**

O Algoritmo Genético (GA) é uma ferramenta poderosa para resolver problemas de otimização combinatória e foi proposto pela primeira vez pelo professor John Holland em 1975 (Holland, 1992). Ele resolve o problema de otimização formulado usando a ideia da evolução darwiniana. As operações básicas de evolução, incluindo crossover, mutação e seleção, tornam o GA apto para realizar pesquisas globais de maneira muito eficaz. Para resolver os problemas que não são muito complexos e cuja exigência em tempo real é estrita, como a detecção multiusuário, apenas o mecanismo de mutação é introduzido no algoritmo ACO, e um algoritmo MACO é proposto. O algoritmo MACO apresentado pode ampliar o alcance de busca e evitar mínimos locais, alterando aleatoriamente um ou mais elementos da melhor solução local após cada iteração.

Na iteração do algoritmo MACO, assumindo que a melhor solução local após a enésima iteração é snbest = (s1nbest, ..., sKnbest) T. Escolha aleatoriamente um ou mais elementos no snbest, altere-os de uma certa maneira e mantenha os outros elementos não foram alterados. Por meio dessa operação de mutação, a solução mutada s0 nbest de snbest pode ser obtida. Se s0 nbest for melhor do que snbest, substitua snbest por s0 nbest. Caso contrário, a melhor solução local permaneceu inalterada. O número de elementos mutados é decidido pelo número K de todos os elementos na solução do problema, e quanto maior for K, mais elementos podem ser mutados. Se K for extremamente grande, o número de elementos mutados pode ser configurado para diminuir à medida que as iterações continuam.

Através da introdução do mecanismo de mutação no algoritmo ACO, o desempenho da busca local é aprimorado, a diversidade de soluções é expandida e a convergência inicial pode ser evitada.

**3. Algoritmo MACO para TSP**

Como o MACO pode ser combinado com muitos algoritmos ACO para melhorar o desempenho deles, o M-MMAS e o M-ACS são propostos combinando o MACO com o MMAS e o ACS respectivamente, e aplicados ao TSP.

**3.1. M-MMAS e sua aplicação ao TSP**

M-MMAS é proposto combinando MACO com MMAS, e segue os princípios de MMAS descritos na Seção 2.2, exceto a operação de mutação após cada iteração.

**Figura 1**

Para comparar M-MMAS com MMAS, eles são aplicados a dois problemas de TSP, respectivamente (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/). O primeiro é Eil51, e há 51 cidades nele com a melhor solução 425. Tanto o MMAS quanto o M-MMAS funcionam por 1000 iterações usando 10 formigas e, especialmente no M-MMAS, vários casos com número diferente de elementos mutados (cidades), que é aumentado para 1–5, são considerados. As cidades escolhidas aleatoriamente são transformadas pela troca de posições com as próximas cidades. Por meio de simulações, a duração média do passeio em Eil51 versus o número de cidades mutantes em cada caso das curvas M-MMAS (o número de cidades mutadas de MMAS é zero) são representados na Fig. 1.

Na Fig. 1, é mostrado que os comprimentos médios de Eil51 resolvidos por M-MMAS em todos os casos com diferentes números de cidades mutadas são todos mais curtos do que aqueles resolvidos por MMAS.

O segundo problema TSP é St70, e existem 70 cidades nele com a melhor solução 675. Ambos MMAS e M-MMAS funcionam por 1500 iterações usando 10 formigas, e especialmente em M-MMAS vários casos com número diferente de elementos mutados (cidades ), que é aumentado para 1–7, são considerados. As cidades escolhidas aleatoriamente são transformadas pela troca de posições com as próximas cidades. Por meio de simulações, a duração média do passeio em St70 versus o número de cidades mutantes em cada caso das curvas M-MMAS (o número de cidades mutadas de MMAS é zero) são representados na Fig. 2.

Na Fig. 2, é mostrado que os comprimentos médios de St70 resolvido por M-MMAS em todos os casos com diferentes números de cidades mutadas são todos mais curtos do que aqueles resolvidos por MMAS.

Resolvendo esses dois problemas típicos de TSP, Eil51 e St70, usando MMAS e M-MMAS respectivamente, podemos ver que o desempenho do M-MMAS na resolução de problemas de otimização combinatória é superior ao do MMAS.

**3.2. M-ACS e sua aplicação ao TSP**

M-ACS é proposto combinando MACO com ACS, e segue os princípios de ACS descritos na Seção 2.3, exceto a operação de mutação após cada iteração.

Para comparar esses dois algoritmos, ACS e vários casos de MACS com número diferente de cidades mutadas são aplicados aos problemas TSP Eil51 e St70, respectivamente. Os parâmetros são iguais aos descritos em 3.1. Por meio de simulações, os comprimentos médios da viagem em Eil51 e St70 versus o número de cidades mutantes em cada caso das curvas M-ACS (o número de cidades mutadas de ACS é zero) são representados nas Figs. 3 e 4.

Nas Figs. 3 e 4, é mostrado que os comprimentos médios de Eil51 ou St70 resolvidos por M-ACS em todos os casos com o número diferente de cidades mutadas são todos mais curtos do que aqueles resolvidos por ACS. Assim, podemos ver que o desempenho do M-ACS na solução de problemas de otimização combinatória é superior ao do ACS.

Para MACO apenas introduz o mecanismo de mutação no final de cada iteração e não mudará a função de trabalho do ACO, ele pode ser combinado com muitos algoritmos ACO para melhorar o desempenho deles. Nas Seções 3.1 e 3.2, M-MMAS e M-ACS são propostos combinando MACO com MMAS e ACS respectivamente, e então eles são usados ​​na resolução de TSP. Os resultados da simulação mostram que, ao alterar aleatoriamente um ou mais elementos da melhor solução local após cada iteração, o algoritmo MACO apresentado pode ampliar o intervalo de busca e evitar mínimos locais, e o desempenho dos algoritmos ACO aprimorados por MACO na resolução de problemas de otimização combinatória é superior a aquele do algoritmo correspondente com a mesma complexidade computacional. Podemos ver também que o número de cidades mutadas não deve ser muito grande, e se for assim, a melhor solução da melhor solução local não pode ser obtida devido à grande mudança dela. O número adequado de elementos mutados deve ser decidido pela escala do problema a ser resolvido.

**4. Detector multiusuário MACO**

**4.1. Detecção de multiusuário**

O CDMA tem sido objeto de extensas pesquisas no campo das comunicações de rádio móvel. Esta técnica permite que um grande número de usuários se comuniquem simultaneamente na mesma banda de frequência. No entanto, isso cria MAI, o que torna o CD de demodulação de um sinal de espectro espalhado em um ambiente multiusuário não confiável e insensível ao efeito próximo-distante. Por esse motivo, a detecção de multiusuário, que pode superar esse problema, é um tópico importante agora para sistemas CDMA.

**4.1.1. Detector convencional**

Supondo que haja K usuários de um sistema CDMA em um canal de caminho único síncrono, o sinal recebido pode ser expresso como:

**1 Equação**

onde Ak (t), gk (t) e dk (t) são a amplitude, a forma de onda do código de assinatura e as informações do k-ésimo usuário, respectivamente. n (t) é o ruído gaussiano branco aditivo (AWGN), com uma densidade espectral de potência de dois lados de N0 / 2 W / Hz. O CD é composto por um banco de K filtros combinados e pode ser mostrado na Fig. 5.

Na Fig. 5, a existência de MAI tem um impacto significativo na capacidade e desempenho do sistema de CD porque o CD segue uma estratégia de detector de usuário único. Conforme o número de usuários interferentes aumenta, a quantidade de MAI aumenta.

**4.1.2. Detector multiusuário ótimo**

A Verdu mostrou que o OMD pode ser alcançado produzindo uma estimativa para o vetor de informação transmitido com base na maximização do logaritmo da função de verossimilhança. A função objetivo do OMD é dada como:

**1 equação**

onde b 2 {+1,1}, YT = (y1, ..., yK) é o vetor linha que consiste nas saídas amostradas dos filtros combinados, A é a matriz diagonal que consiste nas amplitudes recebidas correspondentes, e H = AT RA, em que R é uma matriz de correlação uniforme K K. Apesar dos enormes ganhos de desempenho e capacidade em relação ao CD, o OMD não é prático. A complexidade exponencial no número de usuários torna o custo desse detector muito alto. Consequentemente, os esforços de pesquisa têm se concentrado no desenvolvimento de detectores multiusuário subótimos que exibem boa resistência de perto-longe, complexidade de implementação razoável e desempenho BER comparável ao do OMD.

**4.2. Detector multiusuário ACO**

Como especialidade da detecção multiusuário no sistema CDMA, os algoritmos ACO devem ser ajustados se quisermos aplicá-los à detecção multiusuário. Os ajustes são os seguintes.

(1) Para os K usuários nos sistemas serem independentes, sem perder a generalidade, podemos deixar cada formiga viajar na ordem fixa do primeiro usuário ao Kº usuário. Nesse caso, as formigas não devem decidir se o usuário viajou.

(2) Como não há nenhuma informação heurística, podemos descartar

parâmetros gij, a e b. Porque a informação transmitida

por cada usuário pode ser apenas +1 ou 1, a probabilidade de que

valor o k-ésimo usuário transmitido decidido pela formiga m no

tempo t é dado por:

**1 equação**

(3) A decisão de qual é a melhor solução é baseada nos valores da função objetivo na Eq. (14), e a solução que possui o maior valor é a melhor.

(4) Como o dado na detecção multiusuário deve ser processado em tempo real, apenas a melhor formiga na iteração atual deposita o feromônio após cada iteração, mas o feromônio em todos os caminhos ainda evapora.

Através das regras definidas acima, a detecção de multiusuário pode ser descrita como um problema de escolha de caminho que pode ser resolvido por algoritmos ACO.

**4.3. Detector multiusuário MACO**

Os algoritmos ACO foram aplicados com sucesso à detecção de multiusuário; entretanto, o desempenho do detector multiusuário ACO ainda pode ser melhorado. O MACO provou ter melhor desempenho do que o ACO nas Seções 3.1 e 3.2, então o MACO pode ser usado neste campo e o detector multiusuário MACO é proposto.

O mecanismo de mutação usado no detector multiusuário MACO pode ser descrito a seguir. Supondo que a melhor solução local após a enésima iteração seja bnbest = (b1nbest, ...,, bKnbest) T. Escolha aleatoriamente um ou mais bits no bnbest, trate-os com o operador lógico not e mantenha os outros bits inalterados. Por meio dessa operação de mutação, a solução mutada b0 nbest de bnbest pode ser obtida. Se b0 nbest for melhor do que bnbest, substitua bnbest por b0 nbest. Caso contrário, a melhor solução local permaneceu inalterada. O número de bits mutados é decidido pelo número de usuários no sistema K, e quanto maior for K, mais bits são mutados. Se K for extremamente grande, o número de bits mutados pode ser definido para diminuir à medida que as iterações continuam.

O MACO é aplicado à detecção de multiusuário seguindo as regras descritas na Seção 4.2, e o detector de multiusuário MACO pode ser obtido. É representado pelas seguintes etapas:

Etapa 1 Inicialização dos parâmetros, incluindo o número de iterações Nc, a população da colônia de formigas M, a taxa de evaporação qe o valor inicial do feromônio s (0).

Etapa 2 Defina as saídas dos filtros combinados na Fig. 5 como a melhor solução global inicial.

Etapa 3 As formigas viajam do primeiro usuário para o Kth usuário seguindo a Eq. (12), e então podemos obter soluções M na enésima iteração.

Etapa 4 Compare as soluções M com base na Eq. (11), e defina a solução que tem o maior valor (igual a Cn) da Eq. (11) como a melhor solução local bnbest nesta iteração.

Etapa 5 Calcule a solução mutada b0 nbest de bnbest seguindo o mecanismo de mutação descrito acima e obtenha o valor da Eq. (11) C0 n usando s0 nbest. Compare Cn e C0 n. Se C0 n for maior que Cn, substitua bnbest por b0 nbest e substitua Cn por C0 n.

Etapa 6 Atualize o feromônio da seguinte maneira:

**2 equações**

onde Dskj é a quantidade de feromônio depositada na borda (k, j) do k-ésimo usuário com o valor j, Q é uma constante positiva para garantir (Cn + Q) P 0 e r é uma constante para ajustar o valor de Dskj .

Etapa 7 Compare a melhor solução local com a melhor solução global. Se a melhor solução local é melhor do que a melhor solução global, defina a melhor solução local como a melhor solução global.

Etapa 8 Produza a melhor solução global se o critério de parada for satisfeito ou volte para a Etapa 3.

**Tabela**

A complexidade computacional de um algoritmo pode ser medida pelo número de multiplicações e adições. O CD de complexidade computacional, detector multiusuário ACO, detector multiusuário MACO e OMD são comparados na Tabela 1 quando há K usuários no sistema e cada um detecta apenas um dado de bit.

Na Tabela 1, N é o comprimento das sequências PN, M é o número de formigas na colônia, Nc é o número de iterações. Neste artigo, eles são definidos como K = 10, M = 10 e Nc = 20, então a complexidade computacional do detector multiusuário ACO e detector multiusuário MACO é a mesma e 80,5% menor em relação ao OMD, e será muito menor do que o de OMD quando K é ainda maior.

**4,4. Resultados experimentais**

A fim de avaliar o desempenho do detector multiusuário MACO, um sistema DS-CDMA que o utiliza é projetado conforme a Fig. 6.

Na Fig. 6, o número de usuários K = 10 no sistema CDMA e o comprimento das sequências PN usadas é 15. O número de formigas M = K = 10, o número de iterações Nc = 20, a taxa de evaporação q = 0,3, os parâmetros Q e r na Eq. (14) são iguais a 200 e 15, e há apenas um bit de informação transmitido mutado em cada iteração. O sinal recebido r (t) é tratado no banco de filtros combinados, cujas saídas são alimentadas no detector multiusuário MACO, e então podemos obter a estimativa da informação de banda base transmitida de cada usuário. Uma variedade de experimentos de simulação são apresentados comparando o desempenho do CD, o detector multiusuário ACO, o detector multiusuário PDACO (Wu et al., 2009), o detector multiusuário MACO e o OMD no sistema representado na Fig. 6.

Em primeiro lugar, o desempenho resistente ao efeito de perto-longe desses detectores é comparado. A fim de ilustrar explicitamente, apenas a energia transmitida do primeiro usuário E1 muda, e a energia de outros usuários Ek (k = 2,3, ..., K) é toda inalterada com suas relações sinal-ruído (SNR) = 6 dB. A partir dos resultados da simulação na Fig. 7, podemos ver que o desempenho resistente ao efeito de perto do detector multiusuário MACO é muito melhor do que o CD, o detector multiusuário ACO e o detector multiusuário PDACO, e está próximo ao OMD, especialmente quando o efeito de perto é sério.

**3 Figuras**

Em segundo lugar, o desempenho desses detectores sem efeito de perto é comparado. Neste caso, a energia transmitida de todos os usuários é garantida como igual (E1 = Ek, k = 2,3, ...,, K) e inalterada. É mostrado na Fig. 8 que se o efeito de perto não for considerado, o BER do detector multiusuário MACO é muito menor do que o do CD, o detector multiusuário ACO e o detector multiusuário PDACO, e está próximo do detector BER de OMD.

Portanto, os resultados da simulação mostram que o desempenho geral do detector multiusuário MACO é muito melhor do que o detector multiusuário ACO e o detector multiusuário PDACO e é próximo ou igual ao OMD ao introduzir o mecanismo de mutação no algoritmo ACO, é mais adequado como um esquema de detecção de multiusuário subótimo em sistemas CDMA.

**5. Conclusão**

Neste artigo, propusemos um algoritmo MACO, que pode ampliar o alcance de busca e evitar mínimos locais, introduzindo o mecanismo de mutação nos algoritmos ACO. Portanto, o desempenho do MACO é superior ao do ACO com quase a mesma complexidade computacional. As aplicações do MACO são discutidas e usadas na detecção de TSP e multiusuário. Uma vez que o MACO pode ser combinado com muitos algoritmos ACO para melhorar o desempenho deles, o M-MMAS e o M-ACS são propostos combinando o MACO com o MMAS e o ACS, respectivamente, e aplicados ao TSP. Os resultados da simulação mostram que M-MMAS e MACS têm um desempenho muito melhor do que os algoritmos ACO correspondentes com a mesma complexidade computacional. O detector multiusuário MACO é proposto e simulações de computador mostram que o desempenho do detector multiusuário ACO é superior e também o de resistência próxima à distância.