

Acesso fornecido por:
UNIVERSIDADE FEDERAL DO PIAUI
Sair

Squeaky toyMinhas configuraçõesObter ajuda

Conferências > 2009 International Conference ... 2009 International Conference ...2009 International Conference ...2009 International Conference ...2009 International Conference ...

Resolvendo o Problema do Vendedor Viajante pelo Programa de Algoritmo de Colônias de Formigas

2 Autor (es) Ju-fang Zhu ; Qing-yuan Li Visualizar todos os autores

2
Papel
Citações

226
Cheio
Exibições de
texto

Export to
Collabratec

Alerts

Manage
Content Alerts

Add to Citation
Alerts

Mais como isso

Problema de vendedor ambulante dinâmico baseado em computação evolucionária
Anais do Congresso de 2001 sobre Computação Evolutiva (IEEE Cat. No.01TH8546)
Publicado: 2001

Algoritmo evolucionário memético bacteriano discreto para o problema do vendedor ambulante
2016 Congresso IEEE de Computação Evolutiva (CEC)
Publicado: 2016

Veja mais

Veja as principais organizações de patentes em tecnologias mencionadas neste artigo

ORGANIZATION 4

ORGANIZATION 3

ORGANIZATION 2

ORGANIZATION 1

Clique para expandir

Provided by: **Innovation PLUS**
POWERED BY IEEE AND IP.COM
A PATENT SEARCH AND ANALYTICS TOOL

Abstrato

Seções do documento

EU. Introdução

II. Breve Descrição do Algoritmo de Colônia de Formigas

III Algoritmo de colônia de formigas para solução de TSP

IV. Implementação do Algoritmo Ant no VC

V. Exemplos de Simulação

Mostrar esboço completo ▾

Autores

Figuras

Referências

Citações

Palavras-chave

Métricas

More Like This

Downl
PDF

Abstract: Ant colony algorithm is a novel simulated ecosystem evolutionary algorithm, which is applied to solving complex combinatorial optimization problems. The basic principle a... **View more**

Metadata
Abstrato:
O algoritmo de colônia de formigas é um novo algoritmo evolucionário de ecossistema simulado, que é aplicado para resolver problemas complexos de otimização combinatória. O princípio básico e a realização do algoritmo de colônia de formigas são estudados neste artigo. O algoritmo é realizado sob o ambiente de compilador Visual C ++ e aplicado para resolver o problema de vendedor ambulante (TSP). O resultado está de acordo com a melhor solução de rota. Este algoritmo tem valor prático.

Publicado em: 2009 Conferência Internacional sobre Inteligência Computacional e Engenharia de Software

Data da Conferência: 11 a 13 de dezembro de 2009

Data adicionada ao IEEE Xplore : 28 de dezembro de 2009

CD-ROM ISBN: 978-1-4244-4507-3

Número de Acesso INSPEC : 11034041

DOI: 10.1109 / CISE.2009.5366235

Editora: IEEE

Localização da Conferência: Wuhan, China

Contents

SEÇÃO I.
Introdução

O algoritmo de colônia de formigas é um algoritmo de otimização

inteligente proposto pela primeira vez em 1991 pelo estudioso italiano M. Dorigo, também chamado de sistema de colônia de formigas (ACS) [1] ou otimização de colônia de formigas (ACO) [2]. Ele é aplicado principalmente para resolver tipos de problemas de otimização combinatória, uma série de abordagens heurísticas são desenvolvidas para lidar com esses problemas. As abordagens mais promissoras são: algoritmo genético, pesquisa de tabu, redes neurais e sistema de colônia de formigas. Encontrar a melhor solução é um processo evolutivo da colônia candidata.

Problema do vendedor ambulante (TSP) [3] é um típico problema de otimização combinatória, sua definição geral é a seguinte. Dado n cidades, um vendedor sai de uma cidade, visita todas as cidades, enquanto isso, toda cidade é visitada apenas uma vez, enfim, o vendedor volta para a cidade inicial, seu caminho é um passeio fechado, seu comprimento é dado pela soma dos comprimentos de todos os arcos de que é composto. O número total de caminhos possíveis cresce exponencialmente com o número de cidades, é difícil encontrar o caminho mais curto, de modo que é significativo descobrir um algoritmo aproximado eficiente. Agora, algoritmos heurísticos são reconhecidos e fazem bem neste campo.

O algoritmo de colônia de formigas [4] é um tipo de algoritmo heurístico. Devido à sua vantagem de forte robustez e a capacidade de encontrar a solução ideal, o algoritmo de colônia de formigas dá destaque ao problema de otimização combinatória. Este artigo aplica o algoritmo de colônia de formigas ao problema do TSP para encontrar uma solução ideal e implementa-a no ambiente do compilador VC++ usando a linguagem C++.

SEÇÃO II

Breve Descrição do Algoritmo de Colônia de Formigas

O algoritmo de colônia de formigas [5] é inspirado na habilidade das formigas reais em encontrar o caminho mais curto entre a fonte de alimento e o ninho. O algoritmo de colônia de formigas copia o uso de feromônio por uma colônia natural de formigas para compartilhar informações sobre as rotas encontradas [6]. Quanto feromônio um depósito de formigas em seu caminho é relativo ao comprimento de seu caminho. Quanto mais curto o caminho, maior a concentração do feromônio, ao mesmo tempo, o feromônio se volatiliza com o tempo. A feromona afeta o futuro do caminho, as formigas tendem a selecionar caminhos marcados com mais feromônio. Quanto mais feromônio, mais as formigas estão aptas a vagar nesse caminho. Podemos ver que as formigas que retornaram do ninho usando rotas mais curtas chegarão mais rapidamente e depositarão mais e mais cedo o feromônio. Isso significa que as rotas mais curtas serão fortemente marcadas com feromônio e as formigas provavelmente usarão essas rotas no futuro. Por fim, as formigas podem encontrar um caminho mais curto entre o ninho e a comida. Assim, esse mecanismo de forrageio é adequado para resolver vários problemas de otimização de caminho.

SEÇÃO III

Algoritmo de colônia de formigas para solução de TSP

TSP [7] é um problema completo polinomial não-determinístico completo, é difícil de solve. Em primeiro lugar, os algoritmos construtivos são eficientes, como vizinho mais próximo, fusão mais próxima, inserção mais próxima, inserção mais distante, adição mais próxima, inserção gananciosa e assim por diante. Então, o algoritmo

heurístico se torna o mainstream, recentemente, o algoritmo de colônia de formigas é destacado.

A. Modelo Matemático

O modelo matemático do algoritmo de colônia de formigas consiste no espaço de solução, função objetivo e solução inicial. O espaço da solução contém todos os caminhos. Função objetiva reflete o comprimento total de um caminho. Solução inicial traz aleatoriamente, é o ponto de partida da iteração do algoritmo. A solução ótima corresponde ao caminho que tem menor comprimento.

B. O mecanismo de produção e recebimento de nova solução

Passo 1

O número de formigas é m ; O número de cidades é n ; A quantidade de uma formiga feromônio liberada em um ciclo é Q ; O parâmetro de volatilização do feromônio é ρ ; O feromônio afeta a seleção do caminho das formigas em um grau, esse grau pode ser expresso por α . O grau de afeição do comprimento do caminho à seleção do caminho das formigas pode ser expresso β ; $Iter\ max$ representa os tempos máximos de iteração.

Passo 2

os m formigas são distribuídos aleatoriamente apenas para as n cidades. Toda formiga visita estas n cidades, todas as cidades devem ser visitadas uma vez e somente uma vez, quando terminar um ciclo, uma solução é apresentada. A regra de formação da solução é a seguinte:

- A regra da seleção do caminho: Com base na concentração de feromônio no solo, correspondendo à probabilidade, a formiga visita a próxima cidade que nunca passa neste ciclo. Termina um ciclo depois n vezes.

$$p_{eu}^k(j|t) = \frac{[\tau_{eu}(j|t)^\alpha] [\eta_{eu}(j|t)^\beta]}{\sum_{s \in \text{uma allowedk}} [\tau_{is}(t)^\alpha] [\eta_{is}(t)^\beta]} \quad (1)$$

[Ver fonte](#) ⓘ

Quando formiga k transferências da cidade E para a cidade j em t vezes, $p_{eu}^k(j|t)$ é a probabilidade de transferência de estado (1), $\tau_{eu}(j|t)$ significa a quantidade de feromônio que deposita no solo, $\eta_{eu}(j|t)$ é uma função heurística (2).

$$\eta_{eu}(j|t) = 1 / d_{eu\ j} \quad (2)$$

[Ver fonte](#) ⓘ

Na fórmula acima, d_{eu} é a distância da cidade E para a cidade j . Quanto menos d_{eu} , o melhor $\eta_{eu}(j|t)$ é $p_{eu}^k(j|t)$. O $allowedk$ expressa um conjunto de cidades que formiga k só não passou.

- A regra da atualização do feromônio: Quando uma formiga termina um ciclo após n vezes, de acordo com o comprimento total, todo o feromônio do caminho precisa ser atualizado, enquanto o feromônio antigo reduz com o tempo. Então, depois $t + n$ vezes, no caminho p_{eu} o feromônio é atualizado como segue:

$$\tau_{eu}(j|t + n) = (1 - \rho) \cdot \tau_{eu}(j|t) + \Delta \tau_{eu}(j|t) \quad (3)$$

[Ver fonte](#) ⓘ

$$\Delta \tau_{eu}(j|t) = \sum \Delta \tau_{eu}^k(j|t) \quad (4)$$

[Ver fonte](#) ⓘ

$$\Delta \tau_{eu}^k(j|t) = \frac{Q}{en} \quad (5)$$

[Ver fonte](#) ⓘ

Na fórmula acima, ρ é o parâmetro de volatilização do feromônio, então $1 - \rho$ é o coeficiente residual $\Delta \tau_{eu}^k(t)$ expressa incremento de feromônio no caminho p_{eu} neste ciclo, $\Delta \tau_{eu}^k(t)$ significa a quantidade de feromônio que a formiga k com folhas no caminho p_{eu} neste ciclo, seu valor é calculado referenciando o modelo de ciclo de formigas.

- Calculando o comprimento total e_{u} que cada formiga passou, e atualizando o caminho mais curto: Um grande número de formigas está apto a selecionar os caminhos que têm alta concentração de feromônio e probabilidade de transferência de estado. Finalmente, quase todos seguem o caminho curto, mas um pequeno número deles não estão de acordo com o feromônio que lidera o machismo, a fim de encontrar uma solução melhor, eles encontram patches aleatoriamente. Em termos dos passos acima, quando os tempos de iteração é igual aos tempos de iteração max $Iter\ max$, é o tempo para terminar o ciclo e obter o caminho mais curto, como uma solução ótima.

SEÇÃO IV.

Implementação do Algoritmo Ant no VC

A. Estrutura e classe definidas no algoritmo

Estrutura da cidade declara o número da cidade e a coordenação da cidade; A classe Ant declara o número da cidade atual, a probabilidade de transferência, a matriz de cidade não-passada, a matriz de cidade passada, o comprimento total de um caminho e o caminho mais curto; A classe de informações declara matriz de incremento de feromônio, matriz de feromônio e matriz de distância.

O principal programa do algoritmo Ant em C ++

```

while(iter<itermax)
{
    For(i=0;i<m;i++)//Ants' number is m
    {
        For(j=0;j<n-1;j++)
        //Each ant travelles from start city to other cities
        {
            ants[i].MoveToNextCity();
            //According to the state transfer probability, each
            ant selectes the next city from the unpassed city
            array, and updates the unpassed city array and
            passed city array
        }
    }
    For(i=0;i<m;i++)
    {
        Compare the path lengths of all the ant in this cycle;
        Record the shortest path,and its length;
    }
    //Compare the shortest path length in this cycle with the
    global shortest path length: if the fronter is shorter, then
    the global shortest path is updated by the shortest path
    length in this cycle
    If(curminlen<minlen)
    {
        minlen=curminlen;
        or(j=0;j<n;j++)
        {
            mintour[j]=curmintour[j];
        }
    }
    UpdateInfo();//Update pheromone
    For(i=0;i<m;i++)
        ants[i].Clear();
    iter++;
}

```

SEÇÃO V.

Exemplos de Simulação

As cidades do arquivo ei151 são escolhidas como objetos, o número de formigas é igual ao número da cidade; Q o valor é 100; Os valores desses coeficientes ρ , α , β são respectivamente 0,3, 1, 5; O valor do tempo máximo de iteração é 500. Executando o programa e calculando, podemos ver o caminho do resultado na Fig. 1 . Na Fig. 2 , a linha azul e a linha verde mostram, respectivamente, o comprimento do caminho mais curto e o comprimento médio de todos os caminhos em cada processo de iteração.

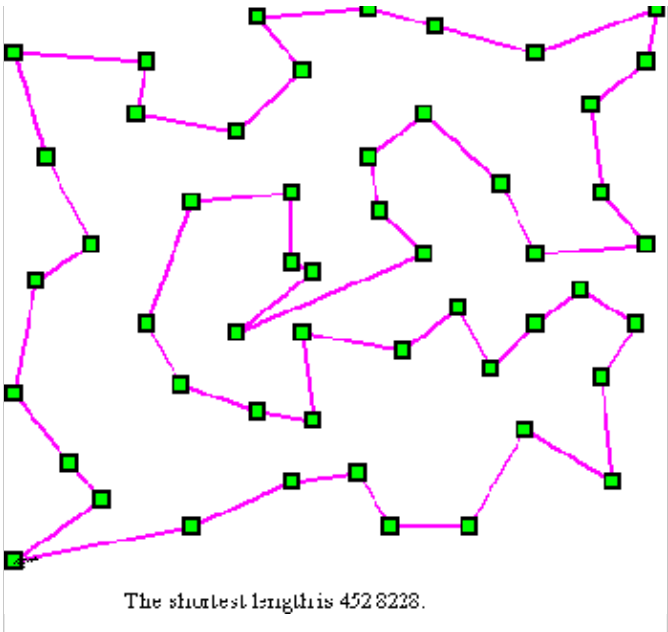


Figura 1. O caminho mais curto

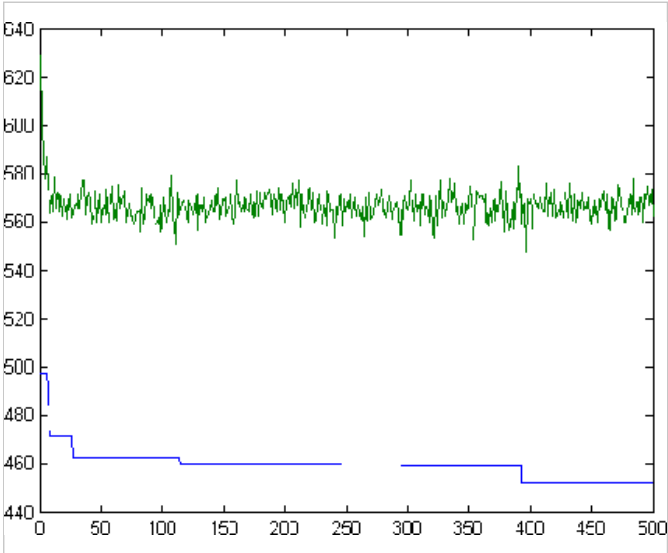


Figura 2. O show de iteração

Veja a partir da amostra, o algoritmo de formigas é eficiente para resolver o problema de vendedor ambulante, entretanto, nós achamos que a solução de otimização depende de parâmetros, em certa medida. É necessário ajustar esses parâmetros para obter a melhor solução.

TABELA I. O afeto do número de formigas na solução

CityNum	51	51	51	51	51	51	51
AntNum	10	20	50	55	100	200	300
Comprimento do percurso	481	468	466	465	460	457	457

TABELA II. O Afeto do Parâmetro de Importância da Feromona

α	0,1	0,2	0,3	0,5	0,7	0,8	0,9	1
Comprimento do percurso	481	479	467	466	463	461	461	460

Em primeiro lugar, a relação entre o número de formigas e o número de cidades é considerado na Tabela I . Quando o número de formigas é o número de cidades uma ou duas vezes, é mais provável encontrar um caminho mais curto. Se o número de formigas for pequeno, a solução não é ótima o suficiente. Note que, dado um grande número de formigas, a eficiência do tempo do algoritmo diminuirá. Em segundo lugar, a solução ótima muda com o valor do parâmetro de importância da feromônio. Na Tabela II . O melhor valor de α é 0,9. Além disso, muitos experimentos testemunham que a solução não flui muito com o

coeficiente de volatilização de feromônio ρ ou parâmetro de importância

heurística β .

SEÇÃO VI.

Conclusão

Este artigo obtém sucesso ao aplicar o algoritmo ant para resolver o problema do vendedor ambulante no ambiente do compilador VC ++. O exemplo de simulação reflete a boa capacidade de otimização do algoritmo de colônia de formigas. Analizando um grande número de dados experimentais, prometem-se métodos de valores de diferentes parâmetros, eles são significativos para o estudo futuro do algoritmo. Além disso, nós também achamos que, o comprimento do caminho rapidamente desce para ser o mais curto nos primeiros vinte passos, isso prova que o algoritmo de formigas tem a desvantagem de estagnação, como resolver este problema é digno de estudo.

Autores	▼
Figuras	▼
Referências	▼
Citações	▼
Palavras-chave	▼
Métricas	▼

IEEE Account	▼
Profile Information	▼
Purchase Details	▼
Need Help?	▼
Other	▼

A not-for-profit organization, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.
© Copyright 2019 IEEE - All rights reserved. Use of this web site signifies your agreement to the terms and conditions.

US & Canada: +1 800 678 4333
Worldwide: +1 732 981 0060

Conta IEEE	Detalhes da compra	Informação do Perfil	Preciso de ajuda?
» Alterar nome de usuário / senha	» Opções de pagamento	» Preferências de Comunicações	» EUA e Canadá: +1 800 678 4333
» Atualizar endereço	» Histórico de pedidos	» Profissão e Educação	» Em todo o mundo: +1 732 981 0060
	» Visualizar documentos comprados	» Interesses técnicos	» Contato e Suporte

Sobre o IEEE Xplore | Contate-Nos | Socorro | Acessibilidade | Termos de uso | Política de Não Discriminação | Mapa do Site | Privacidade e exclusão de cookies

Uma organização sem fins lucrativos, o IEEE é a maior organização profissional técnica do mundo dedicada ao avanço da tecnologia para o benefício da humanidade.
© Copyright 2019 IEEE - Todos os direitos reservados. O uso deste site significa sua concordância com os termos e condições.