

titulo aqui

Sérgio Luis O. B. Correia, Vinícius Moura Romão

¹Mestrado Acadêmico em Ciência da Computação (MACC)
Universidade Estadual do Ceará (UECE).

{sergio, vinicius.romao}@larces.uece.br

Resumo.

1. A otimização combinatória e as colônias de formigas

1.1. Introdução

No início da década de 1990, a otimização combinatória com colônias de formigas (ACO, *Ant Colony Optimization*) [4, 7, 6] surgiu como uma nova técnica bioinspirada voltada para a solução de problemas difíceis de otimização combinatória. ACO é uma metaheurística [10], ou seja, um algoritmo aproximado usado para obter soluções suficientemente boas para problemas difíceis de otimização combinatória em tempo computacional razoável. A fonte de inspiração para a metaheurística ACO foi o comportamento forrageiro das formigas reais. Quando estão procurando por comida, as formigas inicialmente exploram a área próxima ao seu formigueiro de forma aleatória. Assim que uma formiga descobre alguma fonte de comida, ela avalia a fonte e transporta um pouco de volta para o formigueiro. Nesse caminho de volta, a formiga deposita uma trilha de feromônios – substâncias químicas capazes de despertar reações fisiológicas ou comportamentais em outros membros de uma mesma espécie que estejam num determinado raio de espaço físico ocupado pelo excretor [12]. A quantidade de feromônio depositada, que pode depender da quantidade e qualidade da comida, vai guiar outras formigas até a fonte de comida. A comunicação indireta entre as formigas por meio das trilhas de feromônios as torna capazes de encontrar o caminho mais curto entre o formigueiro e a fonte de comida [3]. Essa característica das colônias de formigas reais é explorada em colônias de formigas artificiais, sendo a base da ACO. A Figura 1 mostra essa característica. Em (i), vê-se uma formiga retornando da fonte de comida para o formigueiro, deixando um rastro de feromônio pelo caminho percorrido. Em (ii) são retratadas várias formigas indo do formigueiro à fonte de comida por caminhos diferentes, e marcando esses caminhos com feromônio. As formigas que escolhem o menor caminho retornam primeiro, e o nível de feromônio nesta trilha vai se tornando mais acentuado com o tempo, aumentando a probabilidade de futuras formigas escolherem esse caminho. (iii) mostra essa convergência, onde, após algum tempo, a maior parte das formigas passa a utilizar o caminho mais curto.

1.2. Otimização combinatória

A otimização combinatória é um ramo da matemática aplicada e da ciência da computação que lida com a solução de problemas que podem ser caracterizados como possuindo uma estrutura discreta ou combinatória. Estes problemas são modelados com uma função objetivo e envolvem achar valores para variáveis discretas tal que seja encontrada uma solução ótima para essa função objetivo. Muitos dos problemas de otimização de importância teórica e prática são de natureza combinatória, como por exemplo encontrar o melhor esquema de roteamento de pacotes em uma rede, a melhor maneira de se alocar professores e disciplinas ou a melhor maneira de se visitar um determinado conjunto de lugares diferentes.

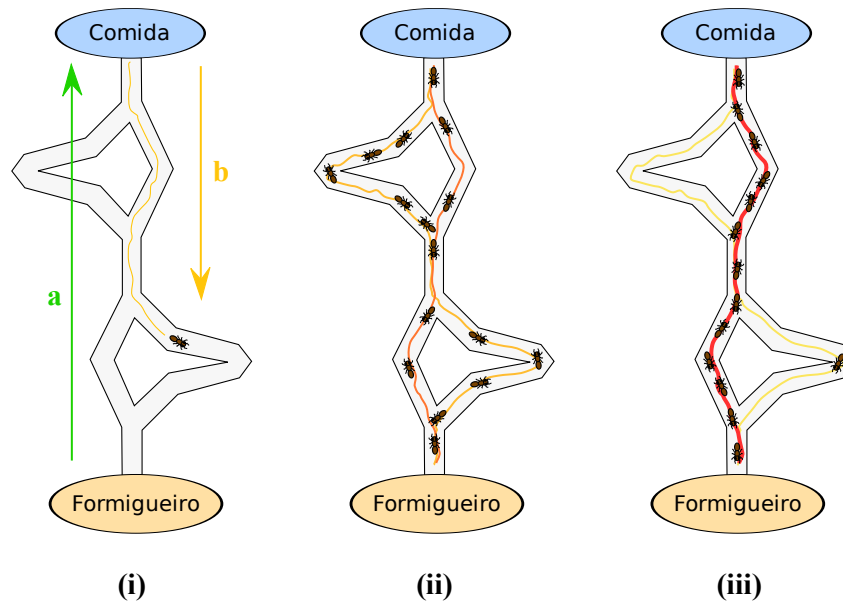


Figura 1. Colônia de formigas

Um problema de otimização combinatória ou é de maximização ou de minimização, e tem associado um conjunto de instâncias do problema. Aqui, *problema* se refere à pergunta a ser respondida, geralmente contendo vários parâmetros ou variáveis com valores não especificados. Uma *instância* se refere a um problema com os valores especificados para cada parâmetro.

Formalmente, um problema de otimização combinatória Π é uma tripla (\mathcal{S}, f, Ω) , onde \mathcal{S} é o conjunto de soluções, f é a função objetivo que atribui um valor $f(s)$ a cada solução $s \in \mathcal{S}$, e Ω é um conjunto de restrições. As soluções pertencentes ao conjunto $\tilde{\mathcal{S}} \subseteq \mathcal{S}$ que satisfazem às restrições Ω são chamadas soluções viáveis. A idéia é encontrar uma solução s^* , viável, que seja um ótimo global, ou seja, ótima dentre todas as soluções viáveis. Num problema de maximização, s^* seria a solução viável com o maior valor da função objetivo f , ao passo que, num problema de minimização, s^* seria a solução viável com o menor valor de f .

1.3. O que é uma metaheurística?

Uma metaheurística é um conjunto de conceitos algorítmicos que podem ser usados para definir métodos heurísticos aplicáveis a uma diversidade de problemas [8]. Uma metaheurística seria então um *framework* algorítmico que pode ser aplicado a diferentes problemas de otimização com relativamente poucas modificações para torná-lo adaptado ao problema específico.

Como exemplos de metaheurísticas podemos citar algoritmos evolucionários, como o *Non-dominated Sorting Genetic Algorithm* (NSGA) [15], o NSGA-II [2], o *Strength Pareto Evolutionary Algorithm* (SPEA) [18] e o SPEA-II [17]; algoritmos probabilísticos, como o *Bayesian Optimization Algorithm* (BOA) [1], o *Expectation-maximization algorithm* (EM) [13] e o *Compact Genetic Algorithm* [11]; algoritmos estocásticos, como o *Tabu Search* [9]; e também algoritmos baseados em inteligência coletiva, como algoritmos baseados em abelhas [16], cupins [14], e finalmente, em formigas, *Ant Colony Optimization* (ACO) [4],

que será detalhado na próxima seção.

1.4. Otimização combinatória com colônias de formigas - a metaheurística

Algoritmos ACO são procedimentos de busca estocástica centrados em um modelo de feromônio, que é usado para tomar amostras – probabilisticamente – dentro de um espaço de busca.

Como exemplo, vamos formular o problema do caixeiro viajante assimétrico (ATSP, *Assymmetric Traveling Salesman Problem*) a seguir. O problema em si consiste em um grafo dirigido, completamente conectado $G(V, A)$, com um peso positivo d_{ij} associado a cada arco $a_{ij} \in A$. Os nós do grafo representam cidades, e os pesos dos arcos, as distâncias entre elas. O objetivo é encontrar, dentre todos os ciclos Hamiltonianos¹ de G , o de custo mínimo, o custo sendo a soma dos pesos de todos os seus arcos. Este problema de otimização combinatória pode ser modelado da seguinte forma: cada cidade $i \in V$ é modelada por uma variável de decisão X_i cujo domínio consiste de um valor v_i^j para cada arco a_{ij} . A instância de uma variável $X_i = v_i^j$ significa que o arco a_{ij} é parte da solução considerada. O conjunto de restrições é definido tal que somente soluções candidatas que correspondam aos ciclos Hamiltonianos de G são soluções válidas. O conjunto de componentes de soluções \mathcal{C} consiste de um componente de solução c_i^j para cada combinação de variável X_i e valor de domínio v_i^j , e o modelo de feromônio \mathcal{T} consiste de uma trilha de feromônio τ_i^j , com um valor τ_i^j associado a cada componente de solução c_i^j .

1.4.1. Um framework ACO simples

O algoritmo 1 mostra o *framework* de um algoritmo ACO básico, que funciona como segue: A cada iteração, n_a formigas probabilisticamente constróem soluções para o problema de otimização combinatória considerado, explorando um dado modelo de feromônio, e, opcionalmente, um procedimento de busca local é aplicado às soluções construídas. Finalmente, antes da próxima iteração, algumas das soluções são utilizadas na execução da atualização de feromônio. Este *framework*, que foi descrito em [5], será explicado com mais detalhes a seguir.

Inicializa os Feromônios(\mathcal{T}). No início do algoritmo, os valores de feromônio são todos inicializados com uma constante $c > 0$.

Constrói Solução(\mathcal{T}). O componente fundamental de qualquer algoritmo ACO é uma heurística construtiva para a construção probabilística das soluções. Uma heurística construtiva monta soluções como sequências de elementos do conjunto finito de componentes de solução \mathcal{C} . A construção de uma solução começa com um conjunto vazio de soluções parciais $s_{\text{parcial}} = \langle \rangle$, e a cada passo da construção, o conjunto de soluções parciais é estendido adicionando-se um componente de solução viável do conjunto $\mathcal{R}(s_{\text{parcial}}) \subseteq \mathcal{C} \setminus \{s_{\text{parcial}}\}$. Este conjunto é determinado a cada passo do mecanismo de construção da solução de modo que as restrições do problema são satisfeitas. O processo de construção de soluções pode ser visualizado como o caminhamento no chamado grafo de construção $G_c = (\mathcal{C}, \mathcal{L})$, que é um grafo completamente conectado, cujos vértices são os componentes de solução em \mathcal{C} e cujas

¹Um ciclo ou caminho hamiltoniano é um caminho que permite passar por todos os vértices de um grafo G não repetindo nenhum, ou seja, passar por cada nó uma e apenas uma vez.

Entrada: Uma instância \mathcal{P} de um problema de otimização combinatória
 Inicializa os Feromônios(\mathcal{T});

$s_{melhor} \leftarrow NULL$;

enquanto condições de saída não forem satisfeitas **faça**

$\mathcal{C}_{iter} \leftarrow \emptyset$;

para $i = 1, \dots, n_a$ **faça**

$s \leftarrow \text{Constrói Solução}(\mathcal{T})$;

se s é uma solução válida **então**

$s \leftarrow \text{Busca Local}(s)$ /* opcional */;

se $(f(s) < f(s_{melhor}))$ ou $(s_{melhor} = NULL)$ **então**

$s_{melhor} \leftarrow s$;

fim

$\mathcal{C}_{iter} \leftarrow \mathcal{C}_{iter} \cup \{s\}$

fim

fim

 Atualiza Feromônio($\mathcal{T}, \mathcal{C}_{iter}, s_{melhor}$);

fim

Saída: A melhor solução até o momento, s_{melhor}

Algoritmo 1: Framework de um algoritmo ACO básico

arestas são os elementos de \mathcal{L} . Os caminhamentos permitidos em G_c são definidos implicitamente pelo mecanismo de construção de solução que define o conjunto $\mathcal{R}(s_{parcial})$ com respeito à solução parcial $s_{parcial}$. A escolha de um componente de solução $c_i^j \in \mathcal{R}(s_{parcial})$ é, em cada passo de construção, feito probabilisticamente com respeito ao modelo de feromônio adotado. A probabilidade para a escolha c_i^j é proporcional a $[\tau_i^j]^\alpha \cdot [n(c_i^j)]^\beta$, onde n é uma função que atribui a cada componente de solução válida um valor heurístico que é também conhecido como *informação heurística*. Os valores dos parâmetros α e β , $\alpha > 0$ e $\beta > 0$, determinam a importância relativa dos valores de feromônio e da informação heurística. A informação heurística é opcional, mas geralmente necessária quando se pretende alcançar uma alta performance do algoritmo. Na maioria dos algoritmos ACO, as probabilidades de se escolher o próximo componente de solução – também chamada de probabilidade de transição – são definidas como segue:

$$P(c_i^j \mid s_{parcial}) = \frac{[\tau_i^j]^\alpha \cdot [n(c_i^j)]^\beta}{\sum_{c_k^l \in \mathcal{R}(s_{parcial})} [\tau_i^l]^\alpha \cdot [n(c_k^l)]^\beta}, \forall c_i^j \in \mathcal{R}(s_{parcial}) \quad (1)$$

Na prática, há diversas formas de se escolher as probabilidades de transição, mas a ilustrada em (1) é tida como tradicional por ter sido utilizada nos primeiros algoritmos ACO propostos, [7] e [6].

Voltando ao ATSP, definido em 1.4, vamos denotar por \mathcal{I} o conjunto de índices da variável de decisão atual e das variáveis de decisão que já têm um valor atribuído. A construção da solução começa com um conjunto de soluções parciais vazio, $s_{parcial} = \langle \rangle$, com $i_c \in \{1, \dots, |V|\}$ escolhido aleatoriamente com $\mathcal{I} = \{i_c\}$. O índice da primeira variável de decisão é armazenado na variável $i_{primeira}$. Então, em cada dos $|V| - 1$ passos de construção, um componente de solução $c_i^j \in \mathcal{R}(s_{parcial})$ é adicionado à solução parcial, onde $\mathcal{R}(s_{parcial}) = \{c_{i_c}^k \mid k \in \{1, \dots, |V|\} \setminus \mathcal{I}\}$. Isto significa que em cada passo de construção,

um valor de domínio é escolhido para a variável de decisão com índice i_c . Uma vez que o componente de solução $c_{i_c}^j$ seja adicionado a $s_{parcial}$, i_c recebe o valor j . As probabilidades de transição usadas em cada um dos primeiros $|V| - 1$ passos de construção são as definidas pela equação (1). No ATSP, a informação heurística pode ser definida como $n(c_i^j) = d_{ij}^{-1}$ (esta escolha dá preferência aos menores caminhos). O último passo de construção consiste em adicionar o componente de solução $c_{i_c}^{i_{primeira}}$ à solução parcial $s_{parcial}$, que corresponde ao fechamento do ciclo Hamiltoniano.

Busca Local(s). Um procedimento de busca local pode ser aplicado para melhorar as soluções construídas pelas formigas. O uso de tal procedimento é opcional, mas experimentos mostraram que, quando disponível, seu uso melhora o desempenho geral do algoritmo [5].

Atualiza Feromônio($\mathcal{T}, \mathcal{C}_{iter}, s_{melhor}$). A idéia por trás da regra de atualização de feromônio é aumentar o nível de feromônio das melhores soluções. Segundo [5], a maioria dos algoritmos ACO utiliza uma variação da regra de atualização (2).

$$\tau_i^j \leftarrow (1 - \rho) \cdot \tau_i^j + \frac{\rho}{\mathcal{C}_{atualizacao}} \cdot \sum_{\{s \in \mathcal{C}_{atualizacao} | c_i^j \in s\}} F(s), \quad (2)$$

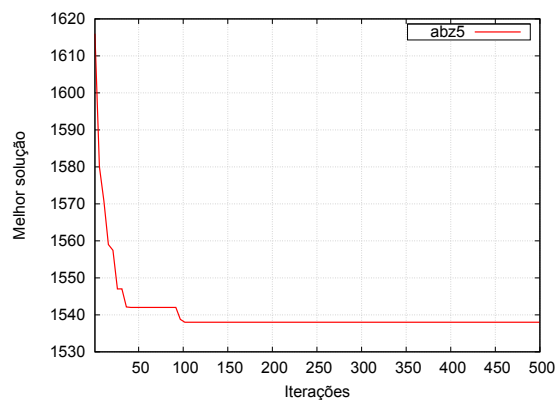
para $i = 1, \dots, n$, e $j = 1, \dots, |D_i|$. Instâncias desta regra de atualização são obtidas com diferentes especificações de $\mathcal{C}_{atualizacao}$, que é um subconjunto de $\mathcal{C}_{iter} \cup \{s_{melhor}\}$, onde \mathcal{C}_{iter} é o conjunto de soluções que foram construídas na iteração atual, e s_{melhor} , a melhor solução até o momento. O parâmetro $\rho \in (0, 1]$ é conhecido como *taxa de evaporação*, e tem a função de decrementar todos os valores de feromônio de modo uniforme. De um ponto de vista prático, a evaporação é necessária para evitar uma convergência muito rápida do algoritmo em direção a uma região subótima, ou seja, é uma maneira de se escapar de um ótimo local, favorecendo a exploração de novas áreas no espaço de busca. $F : \mathcal{C} \mapsto \mathcal{R}^+$ é uma função tal que $f(s) < f(s') \Rightarrow +\infty > F(s) \geq F(s'), \forall s \neq s' \in \mathcal{C}$, onde \mathcal{C} é o conjunto de todas as sequências de componentes de solução que podem ser construídas pelo algoritmo ACO e que correspondem a soluções viáveis. $F(\cdot)$ é conhecida como *função qualidade*. Note que o fator $\mathcal{C}_{atualizacao}^{-1}$ não é usado, normalmente, mas foi introduzido meramente para se estudar os valores esperados de atualização dos níveis de feromônio. Na maioria dos algoritmos, este fator é constante, portanto não muda o comportamento qualitativo do algoritmo em questão [5].

2. Resultados

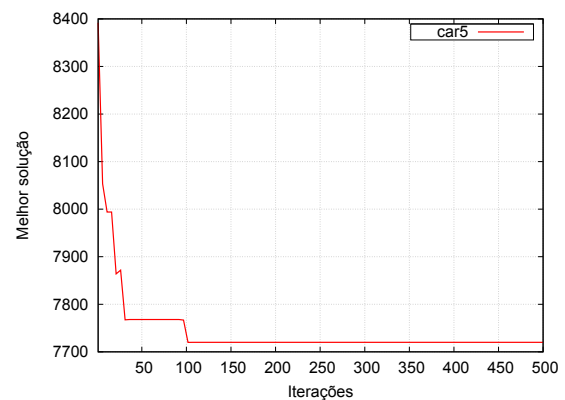
As instâncias do problema do FlowShop utilizadas com a metaheurística ACO foram as seguintes

Referências

- [1] Erick Cantu-Paz, Martin Pelikan, Martin Pelikan, David E. Goldberg, and David E. Goldberg. Linkage problem, distribution estimation, and bayesian networks. Technical report, Evolutionary Computation, 1998.
- [2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.



(a) abz5



(b) car5

Figura 2. Convergência do algoritmo Ant-System para as instâncias abz5 e car5

Instâncias Comparativas				Solução	Solução Obtida			Aproximação
Nº	Referência	m	n	“Ótima”	Inicial	tempo (s)	Melhor	Gap
1	abz5	10	10	1544	1592	19.7	1538	0.99611399
2	car5	10	6	7720	8279	16.8	7720	1.0
Média das aproximações com resultados conhecidos \rightarrow								0.998056995
Nº	Referência	m	n	Inicial	Máxima	Média	Melhor	tempo (s)
3	hel11	100	10	569	555	552	549	5636.7
4	rec07	20	10	1802	1670	1635	1582	51.76
5	rec13	20	15	2268	2096	2052	1971	57.23
6	rec37	75	20	5842	5786	5753	5713	2634.3
7	rec41	75	20	6037	5892	5835	5784	2656.9
Instâncias Opcionais								
Nº	Referência	m	n	Inicial	Máxima	Média	Melhor	tempo (s)
1	car1	11	5	7835	7038	7038	7038	20.8
2	car2	13	4	8148	7376	7292	7166	30.4
3	car3	12	5	7731	7543	7425	7312	26.1
4	car4	14	4	8841	8423	8085	8003	37.7
5	car6	8	9	9588	8754	8608	8505	11.3
6	car7	7	7	7181	6643	6621	6590	7.8
7	car8	8	8	9102	8564	8447	8366	10.8
8	rec01	20	5	1472	1360	1338	1322	95.3
9	rec03	20	5	1281	1188	1162	1133	95.4
10	rec05	20	5	1366	1284	1276	1269	95.1

- [3] J.L. Deneubourg, S. Aron, S. Goss, and JM Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3(2):159–168, 1990.
- [4] M. Dorigo. *Optimization, learning and natural algorithms (Italian)*. PhD thesis, Politecnico di Milano, Milan, Italy, 1992.
- [5] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3):243–278, 2005.
- [6] M. Dorigo, V. Maniezzo, and A. Coloni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1):29–41, 1996.
- [7] M. Dorigo, V. Maniezzo, A. Coloni, M. Dorigo, V. Maniezzo, and A. Coloni. Positive feedback as a search strategy. *Dipartimento di Elettronica e Informatica, Politecnico di*, 1991.
- [8] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. Bradford Book, 2004.
- [9] F. Glover. Tabu search: A tutorial. *Interfaces*, 20(4):74–94, 1990.
- [10] F. Glover and G.A. Kochenberger. *Handbook of metaheuristics*. Springer, 2003.
- [11] GR Harik, FG Lobo, DE Goldberg, S.G.C. Syst, and M. View. The compact genetic algorithm. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 523–528, 1998.
- [12] P. Karlson. Pheromones’: a new term for a class of biologically active substances, 1959.
- [13] TK Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.
- [14] M. Roth and S. Wicker. Termite: Ad Hoc Networking with Stigmergy. *GLOBECOM-NEW YORK-*, 5:2937–2941, 2003.
- [15] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2:221–248, 1994.
- [16] D. Teodorović and M. Dell’Orco. Bee colony optimization—a cooperative learning approach to complex transportation problems. In *Advanced OR and AI Methods in Transportation. Proceedings of the 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland*, pages 51–60. Citeseer, 2005.
- [17] E. Zitzler, M. Laumanns, L. Thiele, et al. SPEA2: Improving the strength Pareto evolutionary algorithm. In *EUROGEN*, pages 95–100. Citeseer, 2001.
- [18] E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. *Swiss Federal Institute of Technology, TIK-Report*, 43, 1998.