

COMPUTAÇÃO BIOINSPIRADA
PROF. RODRIGO COELHO BARROS



Ant Colony Optimization

OTIMIZAÇÃO VIA COLÔNIA DE FORMIGAS

FABRICIO, JEAN, RODRIGO, VANESSA, WALTER

AGENDA

- INTRODUÇÃO
 - Swarm Intelligence
 - Estigmergia
 - Histórico
- ACO (Ant Colony Optimization)
 - Pseudocódigo
 - Fases
 - Algoritmos
- APLICAÇÕES
 - Caixeiro Viajante (TSP)
 - Roteamento de pacotes em NOCs
 - Mapeamento de tarefas em MPSoCs
 - Otimização de Predição de Estrutura de Proteínas
- PROBLEMA PROPOSTO

INTRODUÇÃO

Swarm Intelligence

- “The emergent collective intelligence of groups of simple agents.”
(Bonabeau et al, 1999)



Capacidades individuais limitadas

Comparados



Comportamento coletivo
complexo e inteligente

INTRODUÇÃO

Swarm Intelligence



INTRODUÇÃO

- o Estigmergia é um mecanismo de coordenação indireta entre agentes ou ações, que modifica o ambiente.
- o Gera uma ação sistemática.

Insetos sociais: Feromônios

Ant Colony Optimization usa estigmergia artificial



INTRODUÇÃO



- 1959, Pierre-Paul Grassé - teoria da estigmergia para explicar o **comportamento de construção de ninhos em cupins**;
- 1983, Deneubourg e seus colegas estudaram o **comportamento coletivo** de formigas;
- 1988 e Moyson Manderick tem um artigo sobre **auto-organização** entre formigas;
- 1989, o trabalho de Goss, Aron, Deneubourg e Pasteels sobre o **comportamento coletivo** das formigas argentinas, o que dará a idéia de algoritmos de otimização colônia de formigas;
- 1991, M. Dorigo propôs o **ant system** em sua tese de doutorado

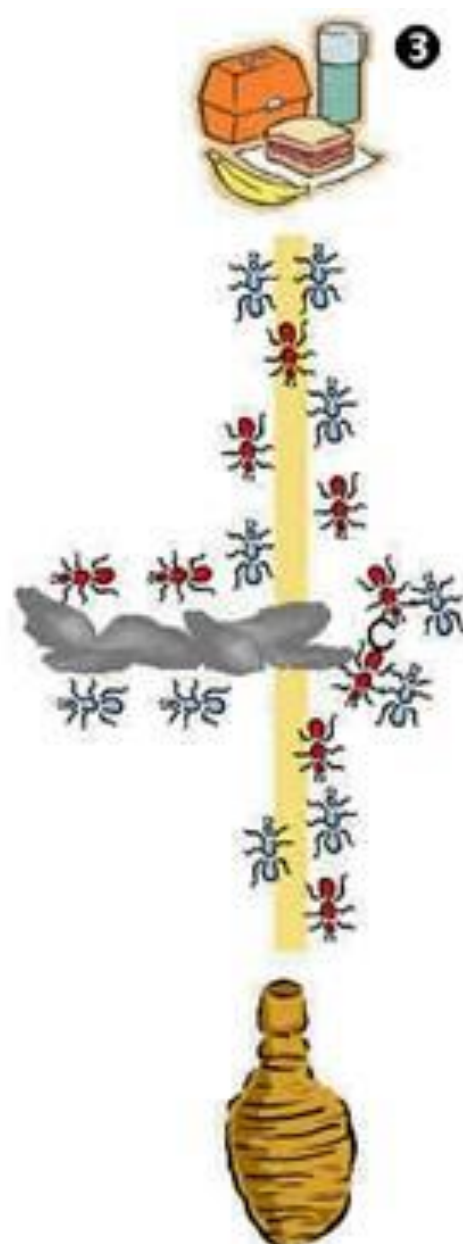
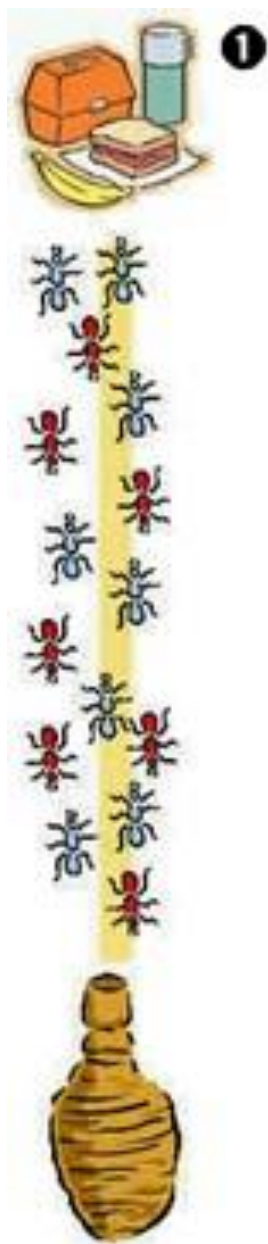
ACO (Ant Colony Optimization)

Otimização via Colônia de Formigas

- Como animais cegos conseguem estabelecer caminhos mais curtos(rota) para buscar alimentos a partir de sua colônia???

Comunicação entre os Indivíduos - trilhas de feromônio





ACO (Ant Colony Optimization)

Otimização via Colônia de Formigas

Inspirada no comportamento das colônias de formigas reais na procura de alimento

- Resolução do problema do Caixeiro Viajante
- Aplicado a ACO (Ant Colony Optimization) NP-com



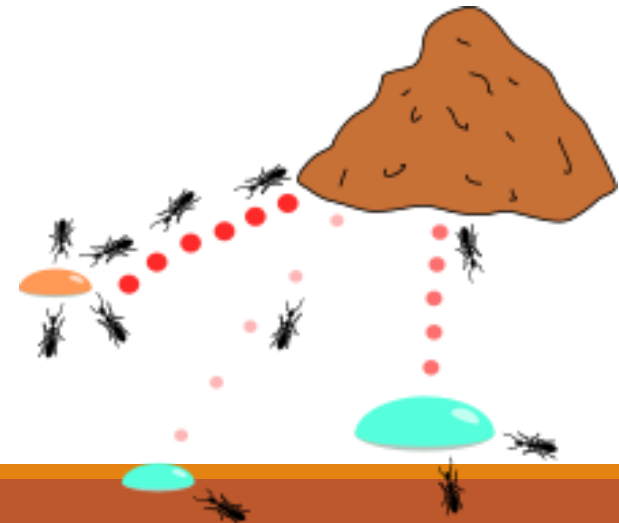
Semelhanças com o sistema real:

- Uma colônia de indivíduos (agentes) que **cooperam entre si**
- Trilhas de feromônio artificiais (informações numéricas) para **comunicação local**
- **Preferência probabilística** por caminhos com maior quantidade de feromônio
- **Caminhos curtos** tendem a ter alta taxa de feromônio

ACO (Ant Colony Optimization)

Otimização via Colônia de Formigas

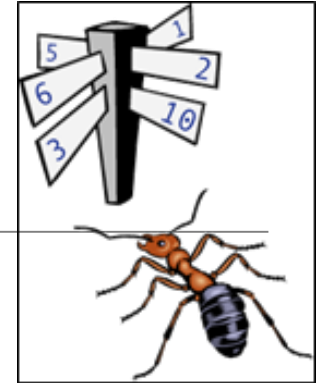
- Saem **aleatoriamente** num primeiro momento
- Se o **alimento** for encontrado, retorna ao ninho - estabelece um rastro de feromônio
- Se **feromônio** for encontrado, aumenta a probabilidade de seguir o trilha de feromônio
- No entanto, os **feromônios evaporam** através do tempo, de tal modo que a menos que sejam reforçada por mais formigas, o feromônios desaparecerão.





ACO (Ant Colony Optimization)

Otimização via Colônia de Formigas



Possui as seguintes Características:

- Versatilidade: Pode ser aplicado a **versões modificadas** de um mesmo problema (e.g. TSP E ATSP);
- Robustez: Pode ser aplicado com **mínimas modificações** de forma a tratar de outros problemas de otimização combinatorial;
- Baseado em População

ACO (Ant Colony Optimization)

Natureza	Ciência da Computação
Habitat Natural	Grafos (nós e arestas)
Ninho e Comida	Nós no gráfico: início e destino
Formigas (Ants)	Agentes ou Formigas artificiais (artificial ants)
Visibilidade	análogo a distância, η
Feromônios	Feromônios artificiais, τ
Comportamento coletor	Passeio aleatório pelo grafo (guiado por feromônios)

ACO (Ant Colony Optimization)

Pseudocódigo

Iniciar parâmetros, iniciar rastros de feromônio

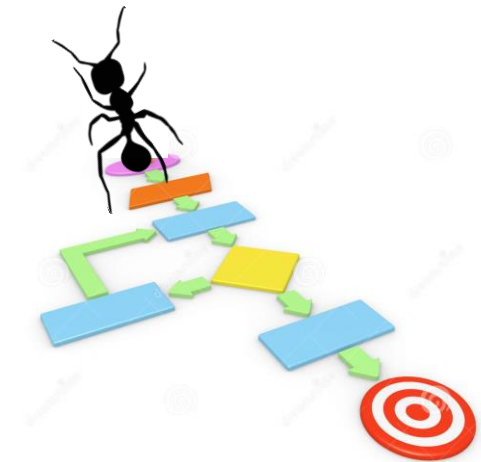
Agendar Atividades

Construção de Soluções

Ações Globais (opcional)

Atualização dos feromônios

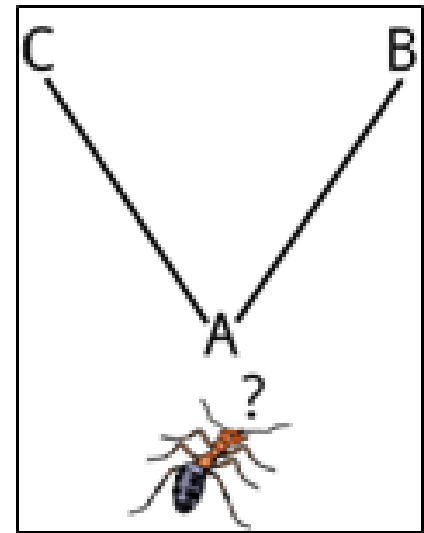
Fim do Agendamento



ACO (Ant Colony Optimization)

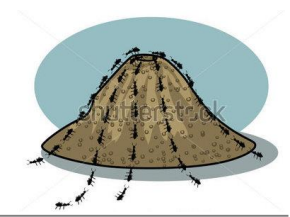
Fase - Construção de Soluções

- Cada formiga inicia em um nó aleatório do grafo;
- A cada iteração, cada formiga decide para qual nó irá em seguida, baseada em informações *heurísticas* e nos feromônios das arestas;
- Repete-se o passo anterior até que todas as formigas tenham construído uma solução completa. Por exemplo, se estiver procurando o menor caminho entre 2 pontos, a solução será sair do ponto inicial e chegar no ponto final;



ACO (Ant Colony Optimization)

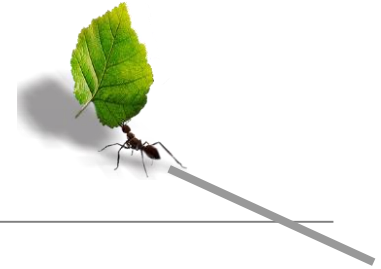
Fase - Ações Globais



- Nesta fase opcional, um agente com conhecimento global do sistema executa ações que não poderiam ser executadas por uma formiga individualmente;
- A ação mais comum nesta fase é aplicar uma Busca Local em cada uma das soluções construídas pelas formigas na fase de Construção de Soluções;
- Embora opcional, essa etapa tem sido usada para se obter os melhores resultados com o ACO. Algoritmo ACO com Busca Local pode ser considerado um algoritmo híbrido.

ACO (Ant Colony Optimization)

Fase - Atualização dos feromônios



- Nesta fase, os rastros de feromônio de cada aresta pertencentes às soluções de cada formiga são atualizados. Esta fase envolve duas etapas: a evaporação e o incremento do feromônio depositado de cada aresta;
- Na evaporação, o feromônio é diminuído por um fator constante e produz um efeito de diversificação no algoritmo;
- O incremento é baseado no uso da aresta no caminho das formigas. Quanto mais a aresta for usada pelas formigas, maior será o depósito de feromônio. O incremento produz um efeito de intensificação no algoritmo.

ACO (Ant Colony Optimization)

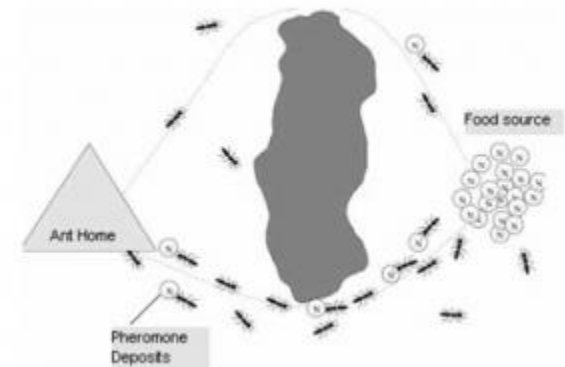
Algoritmos de ACO

Principais :

- Ant System - AS
- Ant Colony System - ACS
- MAX-MIN Ant System - MMAS

Outras variações :

- Elitist Ant System
- Rank-based Ant System
- Best-Worst Ant System



ACO (Ant Colony Optimization)

Notação

- m - parâmetro que indica o número de formigas na colônia
- α (alpha) - parâmetro que indica a importância do feromônio
- β (beta) - parâmetro que indica a importância da informação heurística, por exemplo, no Caixeiro Viajante, o custo entre uma cidade e outra.
- ρ (rho) - parâmetro que regula a taxa de evaporação do feromônio.
- τ (tau) - τ_{ij} indica a quantidade de feromônio na aresta ij , e τ_0 é a quantidade de feromônio inicial nas arestas.

ACO (Ant Colony Optimization)

Notação (cont.)

η (eta) - $\eta_{ij} = \frac{1}{d_{ij}}$, onde d_{ij} é a distância entre as arestas i e j

N_i^k - Conjuntos das arestas que uma formiga k ainda não visitou.

$\Delta\tau_{ij}^k$ - é definida por

$$\begin{cases} \frac{1}{L^k}, & \text{se aresta } ij \text{ faz parte do caminho da formiga } k \\ 0, & \text{caso contrário} \end{cases}$$

, onde L^k é o custo da aresta, percorrido pela formiga k .

ACO (Ant Colony Optimization)

Ant System - AS (versão simplificada)

- Desenvolvido por Marco Dorigo, Vitorio Maniezzo e Alberto Coloni em 1996;
- Fase de Construção: as formigas que atravessam o grafo em construção fazem uma decisão probabilística em cada nodo. Essa decisão é definida pela fórmula:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}$$

- Atualização do Feromônio: o feromônio é atualizado por todas as formigas ao completar o caminho:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k$$

ACO (Ant Colony Optimization)

Ant Colony System - ACS (versão extendida)

- Desenvolvido por Marco Dorigo e Luca Gambardella em 1997;
- Fase de Construção: Com probabilidade q_0 , escolhe a aresta que maximiza a expressão $\tau_{ij} \eta_{ij}^\beta$, caso contrário utiliza a mesma regra de decisão do AS
- Atualização Local de Feromônio: a cada iteração da construção, cada formiga atualiza o feromônio da última aresta pela qual passou: $\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0$
- Atualização Global de Feromônio: feito somente pela melhor formiga, ou seja, somente as arestas visitadas pela melhor formiga são atualizadas: $\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta \tau_{ij}^{melhor}$

ACO (Ant Colony Optimization)

Max-Min Ant System - MMAS

- Desenvolvido por Thomas Stützle e Holger Hoos em 1999;
- Atualização de Feromônio: feito somente pela melhor formiga, ou seja, somente as arestas visitadas pela melhor são atualizadas:
$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{melhor}$$
- Melhor formiga: pode ser tanto a melhor até o momento quanto a melhor de uma iteração;
- Limites para a taxa de feromônio: para evitar a estagnação precoce, os rastros são limitados a um intervalo $[\tau_{min}, \tau_{max}]$.

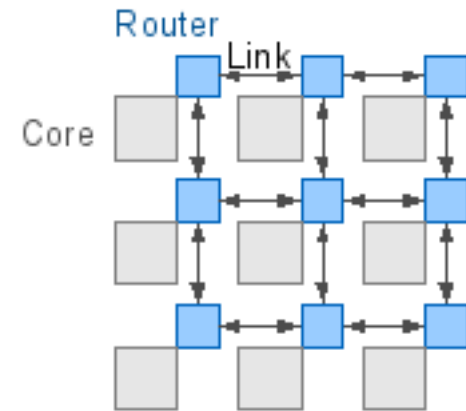
ACO (Ant Colony Optimization)

Comparativo entre atualizações de Feromônio

Alg.	Expressão	Realizado
AS	$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k$	por todas as formigas ao completar o caminho
ACS	$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0$	por todas as formigas em cada passo da construção
ACS	$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta \tau_{ij}^{melhor}$	apenas pela melhor formiga até agora ao completar o caminho
MMAS	$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \Delta \tau_{ij}^{melhor}$	apenas pela melhor formiga da iteração (ou melhor até o momento) ao completar o caminho

ACO: Aplicações

- Aplicação 1: ACO – Problema do Caixeiro Viajante
- Aplicação 2: Roteamento Adaptativo em NOCs baseado em ACO
- Aplicação 3: Mapeamento de Tarefas em MPSoCs baseado em ACO
- Aplicação 4: Otimização da Predição de Estrutura de Proteínas (Hoos, 2005)



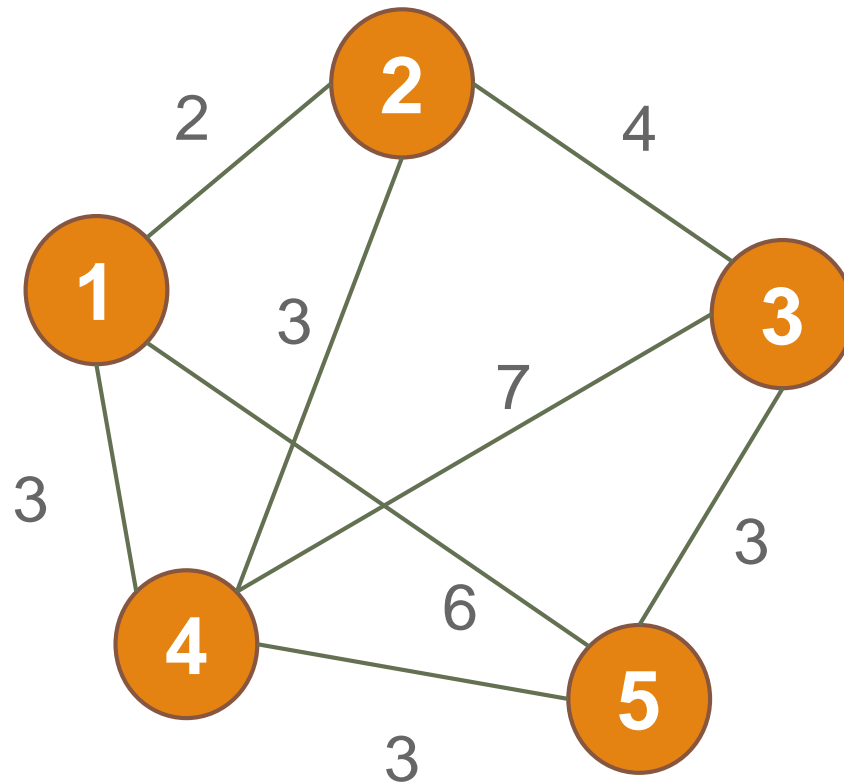
ACO - Caixeiro Viajante

Traveling Salesman Problem (TSP)

- Dado um conjunto de N cidades, o problema do caixeiro viajante pode ser definido como uma **busca pelo menor caminho** que considera **todas as cidades**;
- Escolhem a próxima cidade com uma função de probabilidade que leva em conta a distância e a quantidade de pegadas presentes numa aresta;
- Transições para cidades já visitadas não são permitidas até que uma rota completa seja encontrada; Quando uma rota completa é encontrada, uma substância (feromônio) é colocada em cada aresta visitada.

ACO - Caixeiro Viajante

Traveling Salesman Problem (TSP)



- Gera uma solução (caminho)
- Avalia essa solução
- Deposita sobre cada aresta desse caminho, uma quantidade x de feromônio.
- Sendo x uma função onde se o caminho é menor, x possui maior valor : Ex: $x = 1 / f(\text{caminho})$

Caminho 12345 = 16

Caminho 12354 = 12

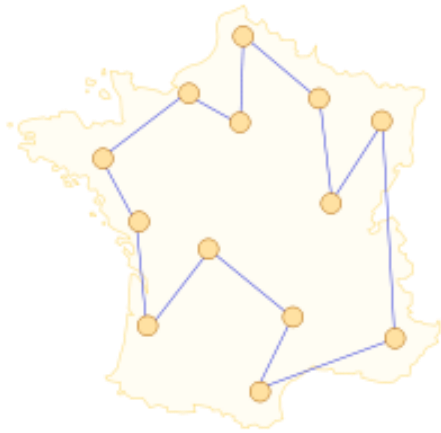
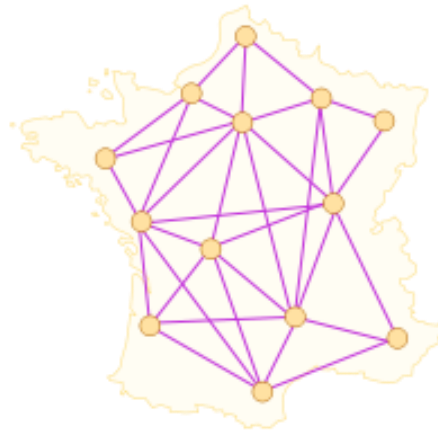
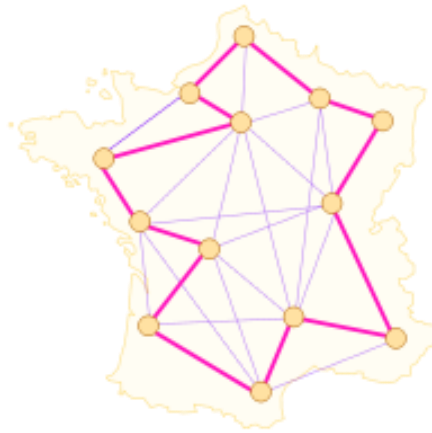
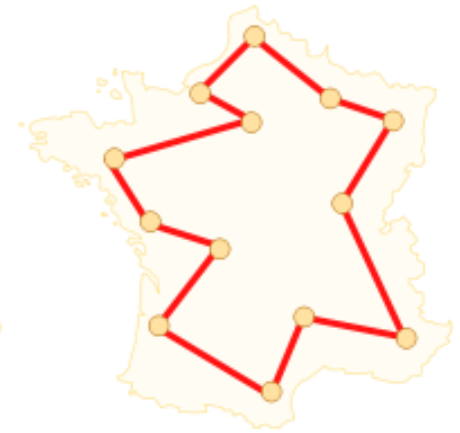
Caminho 12435 = 15

Caminho 12453 = 11

Caminho 14235 = 13....

Caminho menor - Maior feromônio!!!!

ACO - Caixeiro Viajante

**1****2****3****4**

Aplicação 1: Roteamento Adaptativo em NOCs baseado em ACO (Chang, 2010)

Motivação

Objetivo

Método

Resultados

- **Redes intra-chip (NOCs)** são utilizadas para interligar elementos de processamento (EPs) dentro de um chip.
- Problema:
 - Com **diversas aplicações** rodando nesse EPs, o **tráfego** tende a ficar **congestionado**, bloqueando pacotes e causando **atraso na rede**.
 - O aumento da latência (atraso) leva à **degradação do desempenho de aplicações de tempo real** (muito comuns nesses ambientes).
- Solução:
 - O **roteamento adaptativo** é uma abordagem essencial para **lidar** com o **congestionamento da rede**.

Aplicação 2: Roteamento Adaptativo em NOCs baseado em ACO (Chang, 2010)

Motivação

Objetivo

Método

Resultados

- O objetivo do roteamento adaptativo é **distribuir com mais eficiência o tráfego da rede** através da escolha de **caminhos menos congestionados**.
- **ACO** é utilizado como **método de escolha do melhor caminho** em cada ponto da rede de acordo com o **feromônio dos canais candidatos**.

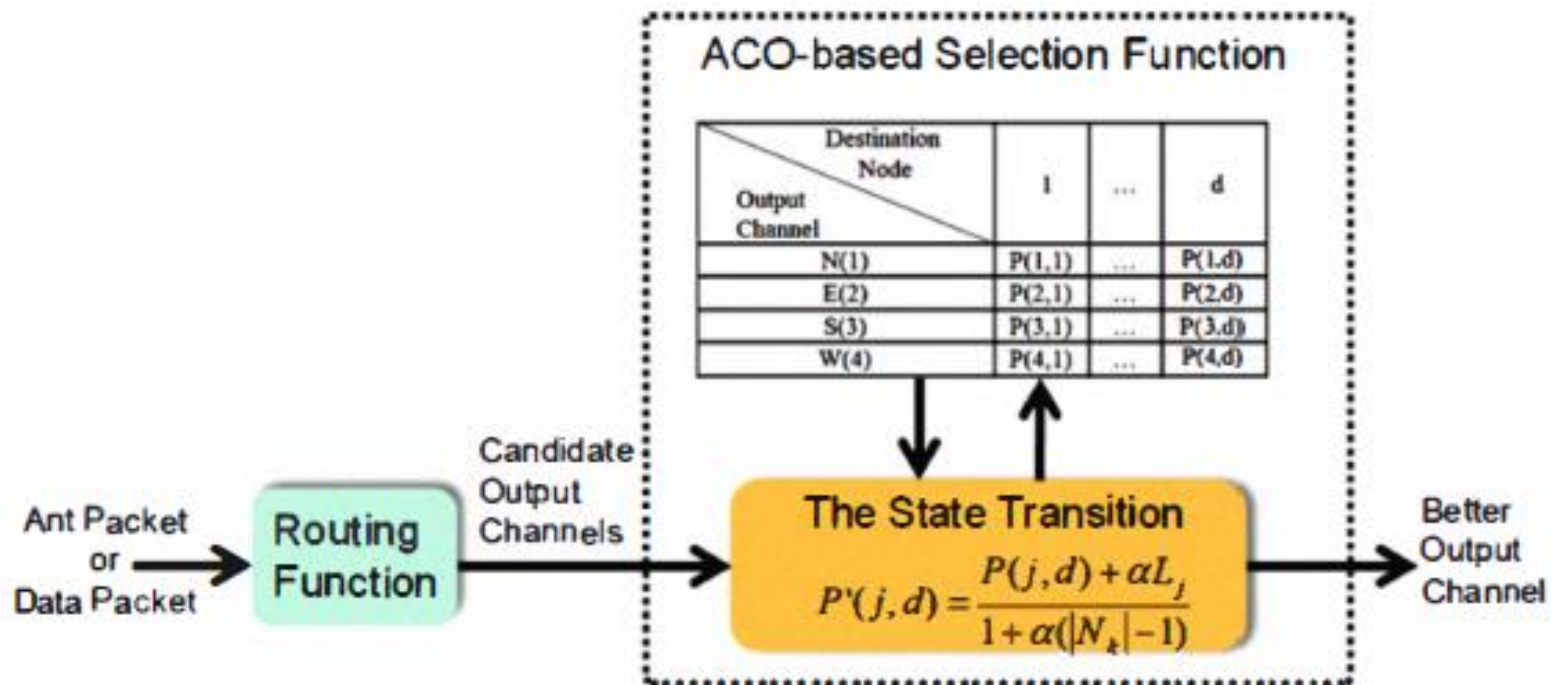
Aplicação 2: Roteamento Adaptativo em NOCs baseado em ACO (Chang, 2010)

Motivação

Objetivo

Método

Resultados



Aplicação 2: Roteamento Adaptativo em NOCs baseado em ACO (Chang, 2010)

Motivação

Objetivo

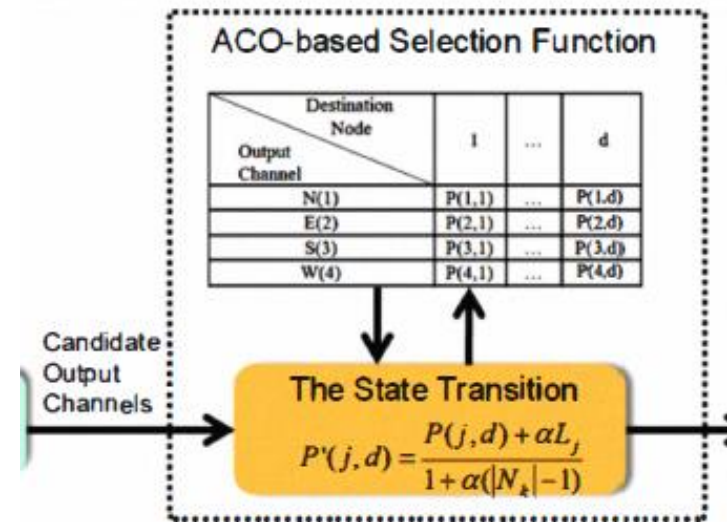
Método

Resultados

- **Feromônio normalizado $P'(j,d)$** (probabilidade de escolha do canal “j” para o destino “d”):

$$P'(j,d) = \frac{P(j,d) + \alpha L_j}{1 + \alpha(|N_k| - 1)}$$

- $P(j,d)$ = valor anterior do feromônio (histórico), armazenado em uma tabela que contém os valores de todos “j,d”;
- L_j = espaço livre no buffer do canal “j” (normalizado);
- α = peso do coeficiente “ L_j ”
- N_k = número de canais do roteador “k”.



Aplicação 2: Roteamento Adaptativo em NOCs baseado em ACO (Chang, 2010)

Motivação

Objetivo

Método

Resultados

○ Fase de treinamento:

- Para cada pacote “ant” que chegar, a coluna “d” (destino do pacote recebido) da tabela que armazena o histórico de feromônios é atualizada por:

$$P(j, d) \leftarrow P(j, d) + r(1 - P(j, d)), \quad j \in \text{seleted channel}$$

onde, $P(j, d) \leftarrow P(j, d) - rP(j, d), \quad j \notin \text{seleted channel}$

- r = fator de reforço (0 a 1), representando o congestionamento do canal “j” para o destino “d”.

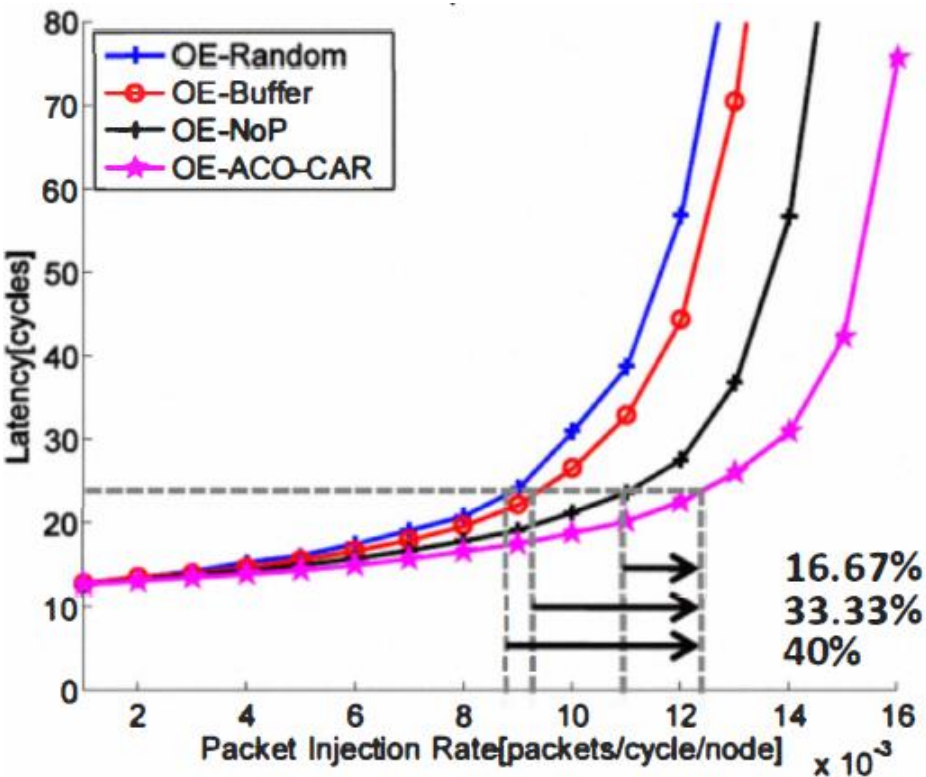
Destination Node Output Channel	1	2	...	d
N(1)	P(1,1)	P(1,2)	...	P(1,d)
E(2)	P(2,1)	P(2,2)	...	P(2,d)
S(3)	P(3,1)	P(3,2)	...	P(3,d)
W(4)	P(4,1)	P(4,2)	...	P(4,d)

Aplicação 2: Roteamento Adaptativo em NOCs baseado em ACO (Chang, 2010)

Motivação

Objetivo

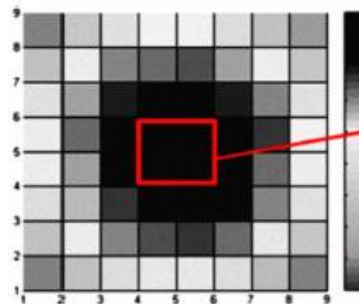
Método

Resultados

Métodos de escolha do canal de saída:

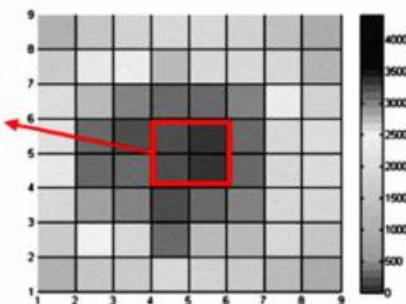
- OE-Random: Escolha aleatória
- OE-Buffer: Buffer local mais vazio
- OE-NoP: Buffer local e vizinho mais vazio
- OE-ACO-CAR: baseado em ACO

OE-NoP



σ of loads = 603

OE-ACO-CAR



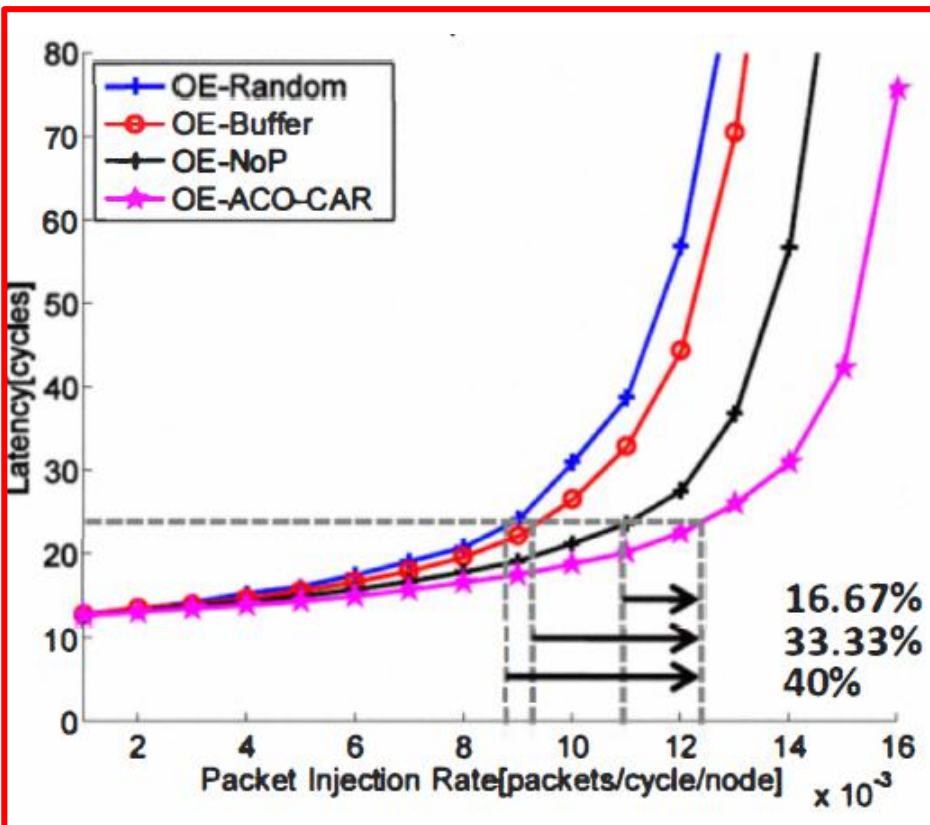
σ of loads = 432

Aplicação 2: Roteamento Adaptativo em NOCs baseado em ACO (Chang, 2010)

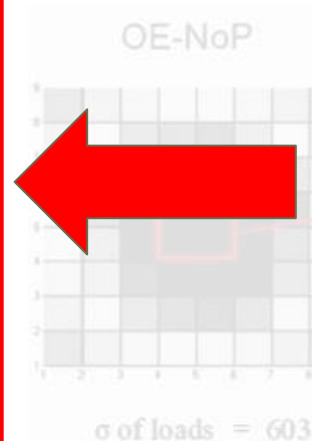
Motivação

Objetivo

Método

ResultadosMétodos de escolha do canal de saída:

- OE-Random: Escolha aleatória
- OE-Buffer: Buffer local mais vazio
- OE-NoP: Buffer local e vizinho mais vazio
- OE-ACO-CAR: baseado em ACO



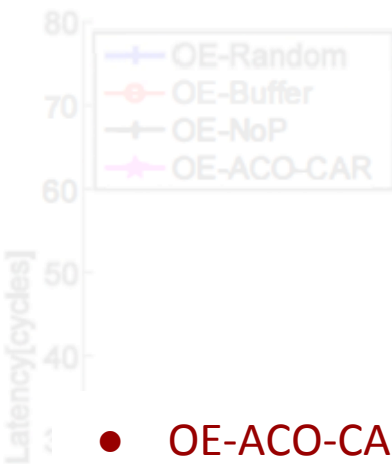
- OE-ACO-CAR apresentou um aumento da latência mais brando (até 40%) em relação aos outros métodos.
- O ponto de saturação do throughput é mais atrasado

Aplicação 2: Roteamento Adaptativo em NOCs baseado em ACO (Chang, 2010)

Motivação

Objetivo

Método

Resultados

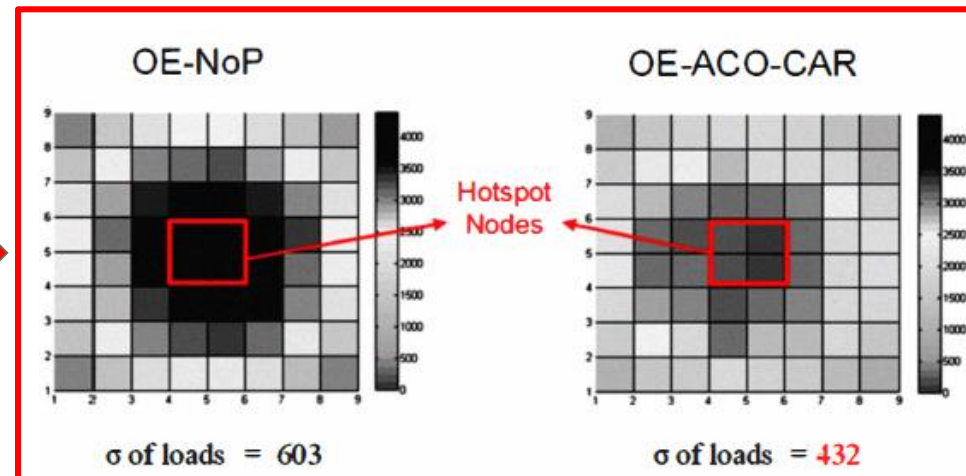
- OE-ACO-CAR apresentou bem menos hotspots.
- O desvio padrão também foi menor, representando um melhor distribuição da carga entre os roteadores.



33.3%
40%
14 16
ode] x 10⁻³

Métodos de escolha do canal de saída:

- OE-Random: Escolha aleatória
- OE-Buffer: Buffer local mais vazio
- OE-NoP: Buffer local e vizinho mais vazio
- OE-ACO-CAR: baseado em ACO



Aplicação 3: Mapeamento de Tarefas em MPSoCs baseado em ACO (Wang, 2011)

Motivação

Objetivo

Método

Resultados

- MPSoCs são utilizados efetuar o **processamento de aplicações** formadas por um conjunto de tarefas comunicantes.
- Problema:
 - A **taxa de dados transferida** em NOCs tem um importância significativa no **consumo de potência** de um MPSoC.
 - Um mapeamento de tarefas mal feito, deixa as **tarefas muito longe uma das outras**, requerendo alta taxa de comunicação entre elas.
- Solução:
 - É proposto um **mapeamento de tarefas** que se preocupe com **requisitos de largura de banda**.

Aplicação 3: Mapeamento de Tarefas em MPSoCs baseado em ACO (Wang, 2011)

Motivação

Objetivo

Método

Resultados

- O objetivo da técnica de mapeamento de tarefas proposta é **reduzir a taxa de dados transferida** entre os EPs do MPSoC arranjando as tarefas em determinados EPs de forma que a taxa de dados transferida seja a menor possível.
- Os **conceitos de ACO** são aplicados no **algoritmo que efetua o mapeamento (estático) de tarefas** das aplicações no MPSoC.

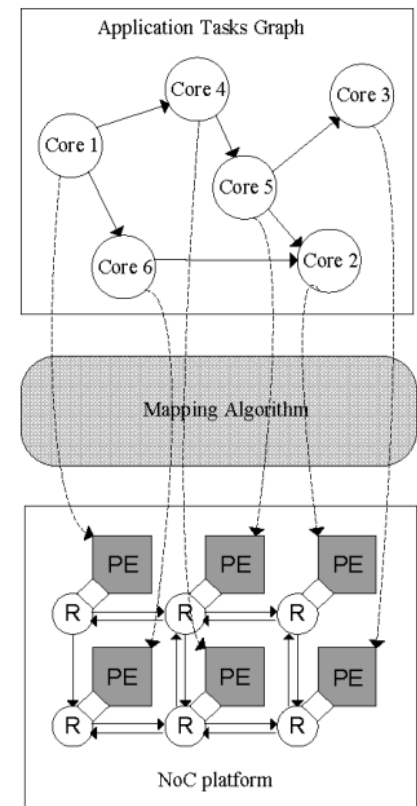


Fig.1 NoC Mapping Instance

Aplicação 3: Mapeamento de Tarefas em MPSoCs baseado em ACO (Wang, 2011)

Motivação

Objetivo

While ($NC > 0$)	
for (ants from 1 to k)	
generate the tabu-search table $Tabu$	
for (each task j)	
generate the probability	
$p_{ij}^k = \begin{cases} \frac{\tau_{ij}}{\sum_{i \notin Tabu} \tau_{ij}} & \text{if } i \notin Tabu \\ 0 & \text{if } i \in Tabu \end{cases}$	
assign task j to processor i according to the probability p_{ij}^k	
update the tabu-search table $Tabu$	
end	
calculate the total bandwidth requirement B^k	
end	

Método

Resultados

search the best ant which minimizes the value of B^k ($B^{best} = \min(B^k)$)
update the pheromone for the best ant
$\tau_{ij} = \rho \tau_{ij} + \Delta \tau_{ij}^{best}$
$\Delta \tau_{ij}^{best} = \begin{cases} \frac{1}{B^{best}} & \text{if the best ant assign task } j \text{ to processor } i \\ 0 & \text{otherwise} \end{cases}$
end
search the best ant and output the mapping results

onde,

- NC = número de ciclos $B = \sum_{\forall l} b_l = \sum_{\forall (i,j)} (c_{i,j} \times dist(i,j))$
- k = número de formigas
- $dist(i,j)$ = nº de saltos entre a localização da tarefa _{i} e da tarefa _{j}
- $c_{i,j}$ = taxa de dados transferidos entre a tarefa _{i} e a tarefa _{j}

Aplicação 3: Mapeamento de Tarefas em MPSoCs baseado em ACO (Wang, 2011)

Motivação

Objetivo

Método

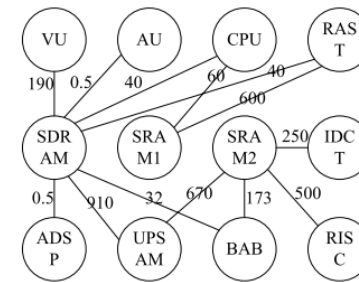
Resultados

Table 1 Comparison of Randomly and ACO Based Mapping

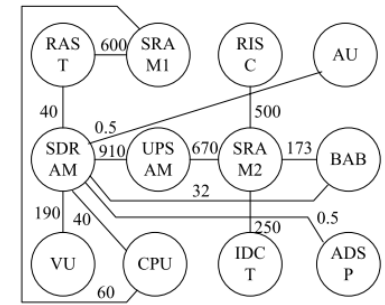
Applications	Random 1(MB/s)	Random 2(MB/s)	Our (MB/s)
tgff 0	96	106	74
tgff 1	135	117	98
tgff 2	182	144	122
tgff 3	88	102	76
tgff 4	141	124	107
tgff 5	118	139	96
tgff 6	92	134	86
tgff 7	161	144	117
tgff 8	108	132	92
tgff 9	114	97	85

Table 2 Comparison of Randomly Mapping and Our Method

Applications	Random (MB/s)	Our (MB/s)	Imp(%)
MPEG 4 decoder	14141	7266	48.6%
VOPD	4355	3834	12%

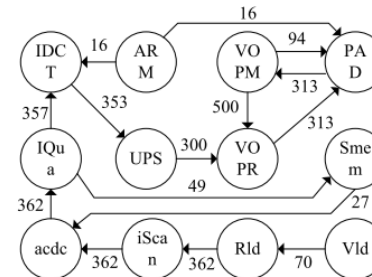


(a) The Mapped MPEG 4 Decoder in [17]

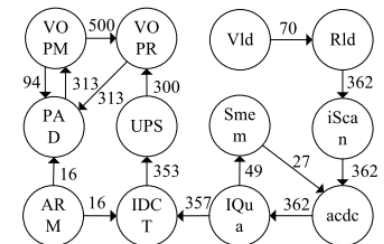


(b) The Mapped MPEG 4 Decoder of Ours

Fig. 2 the Mapped MPEG 4 Decoder on NoC



(a) The Mapped VOPD in [18]



(b) The Mapped VOPD of ours

Fig. 3 the Mapped VOPD on NoC

Aplicação 3: Mapeamento de Tarefas em MPSoCs baseado em ACO (Wang, 2011)

Motivação

Objetivo

Método

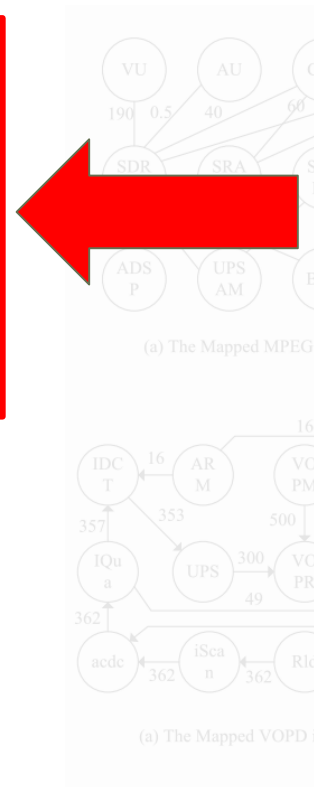
Resultados

Table 1 Comparison of Randomly and ACO Based Mapping

Applications	Random 1(MB/s)	Random 2(MB/s)	Our (MB/s)
tgff 0	96	106	74
tgff 1	135	117	98
tgff 2	182	144	122
tgff 3	88	102	76
tgff 4	141	124	107
tgff 5	118	139	96
tgff 6	92	134	86
tgff 7	161	144	117
tgff 8	108	132	92
tgff 9	114	97	85

Table 2 Comparison of Randomly Mapping and Our Method

Applications	Random (MB/s)	Our (MB/s)	Imp(%)
MPEG 4 decoder	14141	7266	48.6%
VOPD	4355	3834	12%



- Comparação da **taxa de transferência de dados** entre as aplicações em dois **métodos aleatórios de mapeamento** das tarefas e o **método com ACO**
- O método com **ACO** (última coluna) **reduziu a taxa de transferência de dados** para **todas** as aplicações sintéticas testadas

Aplicação 3: Mapeamento de Tarefas em MPSoCs baseado em ACO (Wang, 2011)

Motivação

Objetivo

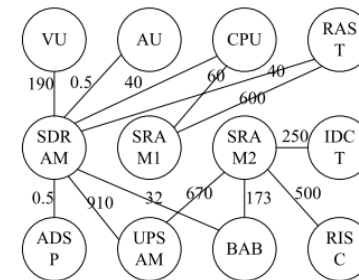
Método

Resultados

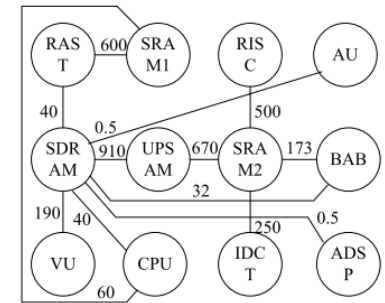
- Em aplicações reais, o método com ACO reduziu em até 48,6% a taxa de transferência de dados entre as tarefas.

Table 2 Comparison of Randomly Mapping and Our Method

Applications	Random (MB/s)	Our (MB/s)	Imp(%)
MPEG 4 decoder	14141	7266	48.6%
VOPD	4355	3834	12%

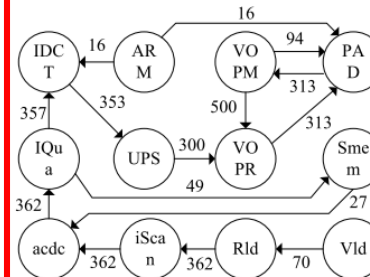


(a) The Mapped MPEG 4 Decoder in [17]

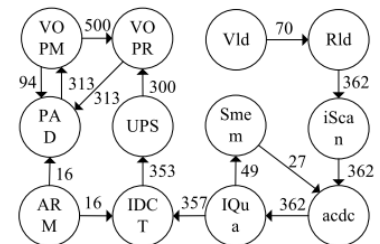


(b) The Mapped MPEG 4 Decoder of Ours

Fig. 2 the Mapped MPEG 4 Decoder on NoC



(a) The Mapped VOPD in [18]



(b) The Mapped VOPD of ours

Fig. 3 the Mapped VOPD on NoC

Aplicação 4: Otimização de Predição de Estrutura de Proteínas (Hoos, 2005)

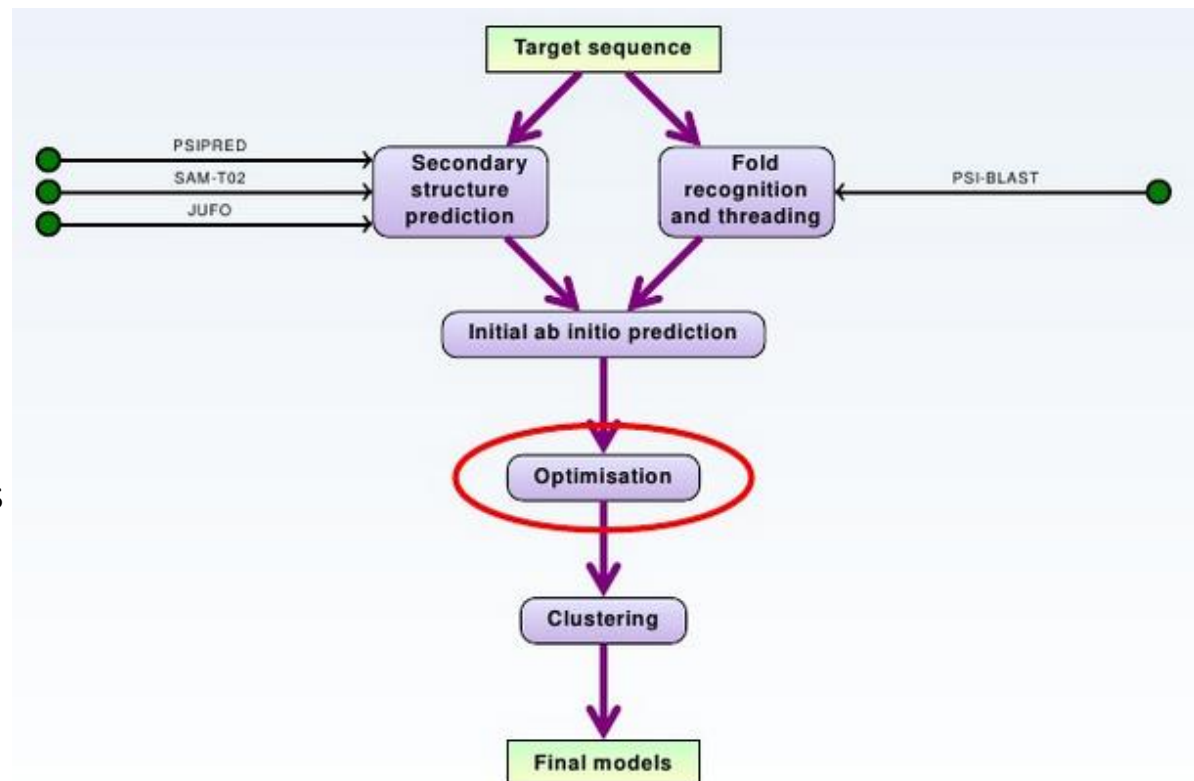
Motivação

Objetivo

Método

Resultados

- O problema da predição da estrutura 3D de proteínas é um dos problemas fundamentais da Bioinformática.
- Existem vários métodos de otimização sendo aplicados na resolução do problema pela metodologia *ab initio* que são baseados em modelos reduzidos de estrutura da proteína, incluindo Métodos de Monte Carlo, Algoritmos Evolutivos e abordagens híbridas.



Aplicação 4: Otimização de Predição de Estrutura de Proteínas (Hoos, 2005)

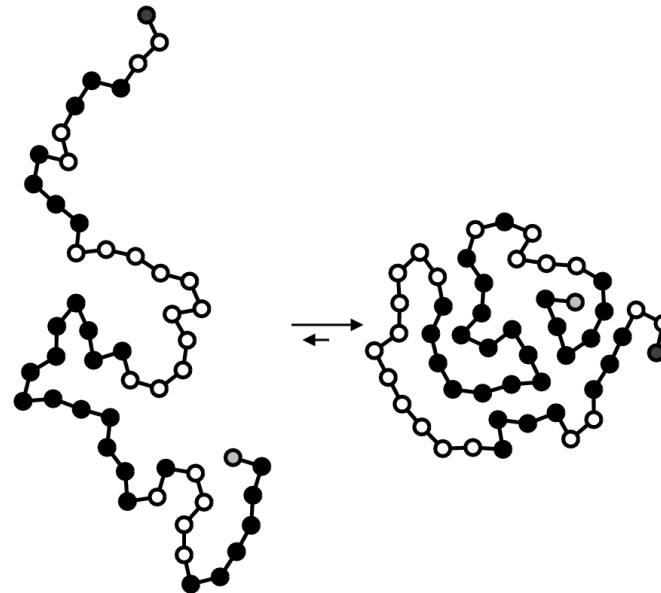
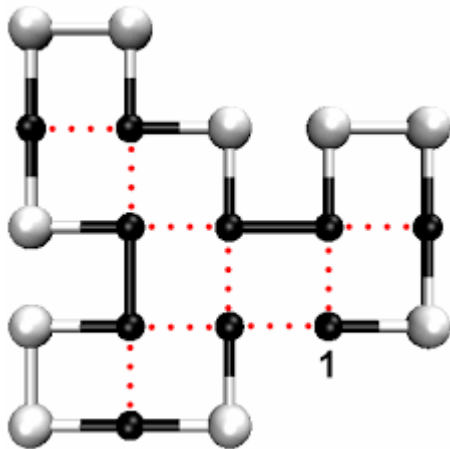
Motivação

Objetivo

Método

Resultados

- Utilizando a abstração do modelo HP para representação da estrutura, o ACO é utilizado para determinar, de todos os modelos gerados, quais são os que melhor representam as conformações nativas da proteína.



Aplicação 4: Otimização da Predição de Estrutura de Proteínas (Hoos, 2005)

Motivação

Objetivo

Método

Resultados

```
procedure ACO
  initialise pheromone trails;
  while (energy threshold not reached) do
    construct candidate conformations;
    perform local search;
    update pheromone values;
  end
end
```

Aplicação 4: Otimização de Predição de Estrutura de Proteínas (Hoos, 2005)

Motivação

Objetivo

Método

Resultados

- construct candidate conformations
 - Gera conformações aleatórias, iniciando de um ponto qualquer e dobrando para uma das direções aleatoriamente, resíduo por resíduo;
 - A probabilidade é calculada através da fórmula:

$$p_{i,d} := \frac{[\tau_{i,d}]^{\alpha} [\eta_{i,d}]^{\beta}}{\sum_{e \in \{S,L,R,U,D\}} [\tau_{i,e}]^{\alpha} [\eta_{i,e}]^{\beta}}$$

- onde $\tau_{i,d}$ indica o quão desejável é utilizar uma determinada direção de acordo com os motifs locais;
- $\tau_{i,d}$ inicial igual para todos os motifs e vai se alterando de acordo com os feromônios depositados a cada iteração, determinadas pela busca local;

Aplicação 4: Otimização de Predição de Estrutura de Proteínas (Hoos, 2005)

Motivação

Objetivo

Método

Resultados

- perform local search
 - aplicação de pequena mutação em uma conformação visando melhoria;

```
procedure IterativeImprovementLS(c)
  input: candidate conformation c
  output: candidate conformation c
  while (termination condition not satisfied) do
    i := random({1, . . . , n});
    c' := longRangeMove(c, i);
    if E(c') ≤ E(c) then
      c := c';
    end
  end
  return(c)
end
```

Aplicação 4: Otimização de Predição de Estrutura de Proteínas (Hoos, 2005)

Motivação

Objetivo

Método

Resultados

- update pheromone values;
 - a cada construção e busca local, o $\tau_{i,d'}$ é multiplicado pelo fator de persistência do feromônio (ρ)

$$\tau_{i,d'} = \rho \cdot \tau_{i,d'}$$

- e algumas formigas, que geraram conformações de menor energia, tem seu $\tau_{i,d}$ somado de $D_{i,d,c'}$, que representa a qualidade da solução naquele ponto (1 se existe um motif para aquela posição, 0 caso contrário)

ACO: Proposta de Trabalho



Referencias:

- P.-P. Grassé, *La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes natalensis et Cubitermes sp. La théorie de la Stigmergie : Essai d'interprétation du comportement des termites constructeurs*, Insectes Sociaux, numéro 6, p. 41-80, 1959.
- J.L. Deneubourg, J.M. Pasteels et J.C. Verhaeghe, *Probabilistic Behaviour in Ants : a Strategy of Errors?*, Journal of Theoretical Biology, numéro 105, 1983.
- F. Moyson, B. Manderick, *The collective behaviour of Ants : an Example of Self-Organization in Massive Parallelism*, Actes de AAAI Spring Symposium on Parallel Models of Intelligence, Stanford, Californie, 1988.
- S. Goss, S. Aron, J.-L. Deneubourg et J.-M. Pasteels, *Self-organized shortcuts in the Argentine ant*, Naturwissenschaften, volume 76, pages 579-581, 1989
- Dorigo M., V. Maniezzo et A. Coloni, *Positive feedback as a search strategy*, rapport technique numéro 91-016, Dip. Elettronica, Politecnico di Milano, Italy, 1991
- EDWIN W., PHILLIP S., ROSALYN K., PATRICK X, Ant Colony Optimization . PIC 10C SPRING 2011 <http://www.math.ucla.edu/~wittman/10c.1.11s/Lectures/Raids/ACO.pdf>
- Chang et. al. "ACO-based Cascaded Adaptive Routing for traffic balancing in NoC systems," in *Green Circuits and Systems (ICGCS), 2010 International Conference on* , vol., no., pp.317-322, 21-23 June 2010.
- Wang et. a. "Bandwidth-Aware Application Mapping for NoC-Based MPSoCs", in Journal of Computational Information Systems, 7:1, pp. 152-159, 2011.

Perguntas???

