



**Unioeste - Universidade Estadual do Oeste do Paraná**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**  
Colegiado de Ciência da Computação  
*Curso de Bacharelado em Ciência da Computação*

**Enxame de Partículas aplicado ao Problema do Caixeiro Viajante**

*Anderson Bottega da Silva*

**CASCADEL**  
**2017**

**ANDERSON BOTTEGA DA SILVA**

**ENXAME DE PARTÍCULAS APLICADO AO PROBLEMA DO  
CAIXEIRO VIAJANTE**

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientadora: Prof. Adriana Postal

CASCADEL  
2017

**ANDERSON BOTTEGA DA SILVA**

**ENXAME DE PARTÍCULAS APLICADO AO PROBLEMA DO  
CAIXEIRO VIAJANTE**

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em  
Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel,  
aprovada pela Comissão formada pelos professores:

---

Prof. Adriana Postal  
Colegiado de Ciência da Computação,  
UNIOESTE

---

Prof. Josué Pereira de Castro  
Colegiado de Ciência da Computação,  
UNIOESTE

---

Prof. Suzan Kelly Borges Piovesan  
Colegiado de Ciência da Computação, UTFPR -  
Santa Helena

Cascavel, 7 de fevereiro de 2018

*Tudo o que temos de decidir é o que fazer com o  
tempo que nos é dado.  
Gandalf, O Senhor dos Anéis*

# Lista de Figuras

2.1	Tomada de decisão dos pássaros . . . . .	5
2.2	Fluxograma do algoritmo do PSO . . . . .	7
2.3	Aplicação do algoritmo sobre partículas aleatórias . . . . .	8
2.4	Criação de novos arcos . . . . .	11
3.1	Fluxograma do algoritmo do PSO de Clerc . . . . .	14
3.2	Comparação entre a melhor solução e os algoritmos VMP, AG, AM e PSO . . .	18
A.1	Percurso obtido com o PSO para a entrada eil51.tsp . . . . .	22
A.2	Percurso obtido com o PSO para a entrada rat99.tsp . . . . .	23
A.3	Percurso obtido com o PSO para a entrada bier127.tsp . . . . .	23
A.4	Percurso obtido com o PSO para a entrada kroA200.tsp . . . . .	24
A.5	Percurso obtido com o PSO para a entrada linhp318.tsp . . . . .	24

# Lista de Abreviaturas e Siglas

PCV	Problema do Caixeiro Viajante
AG	Algoritmo Genético
AM	Algoritmo Memético
VMP	Vizinho Mais Próximo
PSO	<i>Particle Swarm Optimization</i>
AGM-mcd	Problema da Árvore Geradora Mínima Restrita em Grau Multicritério
QPSO	<i>Quantum Particle Swarm Optimization</i>
LKH	<i>Lin-Kernighan Helsgaun</i>
OX	<i>Order Crossover</i>
BLS	<i>Best Local Search</i>
BGS	<i>Best Global Search</i>
NC	Número de Cidades

# Lista de Símbolos

$v_i(t + 1)$	Nova velocidade
$v_i(t)$	Velocidade atual da partícula
$c1, c2$	Operadores cognitivos
$P_b$	Melhor partícula local
$G_b$	Melhor partícula global
$x_i(t)$	Posição atual da partícula
$x_i(t + 1)$	Nova posição da partícula
$X_{i+1}$	Nova posição da partícula
$X_i$	Posição atual da partícula
$V_i$	Velocidade atual da partícula
$V_{i+1}$	Nova velocidade da partícula
$r1, r2$	Valores aleatórios
$c1$	Peso do BLS
$c2$	Peso do BGS
$dBLS$	Distância atual entre a posição e o BLS
$dBGS$	Posição atual do BGS
$N$	Número de vértices
$P\_MAX$	Limite do espaço de busca
$V\_MAX$	Velocidade máxima
$X$	Vetor de cidades
$\Omega^n$	Espaço $n$ -dimensional
$\Phi^n$	Espaço do PCV
$\pi$	Permutação de $n$
$S$	Conjunto solução
$V_e$	Conjunto velocidade
$\ v\ $	Tamanho do conjunto de trocas

# Sumário

<b>Lista de Figuras</b>	<b>v</b>
<b>Lista de Abreviaturas e Siglas</b>	<b>vi</b>
<b>Lista de Símbolos</b>	<b>vii</b>
<b>Sumário</b>	<b>viii</b>
<b>Resumo</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos . . . . .	2
1.2 Trabalhos Correlatos . . . . .	2
1.3 Organização do Trabalho . . . . .	3
<b>2 Enxame de Partículas</b>	<b>4</b>
2.1 Otimização por Enxame de Partículas . . . . .	4
2.2 PSO aplicado ao PCV . . . . .	8
2.3 Método Base para Implementação . . . . .	11
<b>3 Implementação e Testes</b>	<b>13</b>
3.1 PSO Implementado . . . . .	14
3.2 Ambiente Computacional . . . . .	17
3.3 Testes . . . . .	17
3.4 Análises . . . . .	19
<b>4 Considerações Finais e Trabalhos Futuros</b>	<b>21</b>
<b>A Melhores Percursos Encontrados para o PSO dos Arquivos da Biblioteca TSPLIB</b>	<b>22</b>
<b>Referências Bibliográficas</b>	<b>25</b>

# Resumo

Neste trabalho, primeiro é apresentado um estudo da aplicação do PSO ao PCV. Em seguida é apresentado uma implementação do PSO aplicado ao PCV para a realização de testes. Por fim apresentamos uma comparação entre o PSO e os algoritmos VMP, AG e AM para a análise dos mesmos, analisando a eficiência de cada algoritmo.

**Palavras-chave:** Otimização por Enxame de Partículas, Problema do Caixeiro Viajante, PSO de Clerc.

# Capítulo 1

## Introdução

Os problemas de otimização procuram encontrar o valor máximo ou mínimo de funções dentro de intervalos de interesse e são comuns em diversas áreas como economia, projeções de vendas em fábricas, entre outros. O Problema do Caixeiro Viajante (PCV) é um problema de otimização e pertence à classe de problemas NP-Completo. Por esse motivo tornou-se muito utilizado na área da ciência da computação, pois se uma solução for encontrada e ele puder ser resolvido em tempo polinomial, uma gama de outros problemas da mesma classe de complexidade serão solucionados (LEVITIN, 2003).

O PCV é descrito como um vendedor que deve passar por um número  $n$  qualquer de cidades. O vendedor precisa encontrar o menor caminho entre todas as cidades obedecendo a seguinte regra: ele deve visitar todas as cidades uma única vez e voltar para a sua cidade de origem (MACHADO; LOPES, 2005) (HERRERA, 2007). O PCV é representado através de um grafo.

As soluções aproximadas para o problema do caixeiro viajante podem ser encontradas por diversas técnicas, tais como, algoritmos genéticos (AGs) (HOLLAND, 1975), algoritmos meméticos (AMs) (MOSCATO, 1989), algoritmo do vizinho mais próximo (VMP) (LEVITIN, 2003), entre outras técnicas. Neste trabalho foi estudada a técnica de otimização por enxame de partículas (PSO), desenvolvido por Kennedy e Eberhart (1995a) que observaram o comportamento de pássaros, formigas, abelhas entre outros sistemas biológicos e notaram que esses sistemas utilizam o trabalho coletivo para sobrevivência (KENNEDY; EBERHART, 2001).

## 1.1 Objetivos

Este trabalho tem como objetivo descrever as principais características do PSO e verificar se o mesmo pode ser aplicado ao PCV. Foi realizada a comparação com os resultados dos algoritmos genéticos e algoritmos meméticos apresentados por Fredrich (2010), de modo que possa ser analisado rapidamente a melhor técnica.

## 1.2 Trabalhos Correlatos

Souza (2006) propõe dois algoritmos aplicados ao PCV. Esses algoritmos são baseados no PSO e utiliza-se de busca local e *path-relinking*<sup>1</sup> como operadores de velocidade. Souza (2006) também aplica estes dois algoritmos ao Problema da Árvore Geradora Mínima Restrita em Grau Multicritério (AGM-mcd).

Kennedy e Eberhart (1995b) apresenta o PSO aplicado ao treinamento de redes neurais e mostra resultados tão eficientes quanto os algoritmos já consolidados mas em menos tempo.

Secret (2001) utiliza o PSO para missões de inspeção ou vigilância aplicado ao PCV.

Jerald et al. (2004) utiliza o PSO aplicado a manufatura e apresenta bons resultados.

Hoffmann (2011) propõe a movimentação de partículas baseada em um centroide com uma nova interpretação de “adição” e “velocidade”, pois a “adição” e a “velocidade” utilizada por Clerc (2004) apresentam desvantagens. Este novo modelo de PSO é aplicado ao PCV. Suas soluções foram melhores que o PSO puro, ou seja, o PSO sem a adição de métodos para melhorar sua performance.

Herrera (2007) propõe um algoritmo do PSO com as heurísticas *Quantum Particle Swarm Optimization* (QPSO) fazendo com que as partículas se movam com regras definidas na mecânica quântica. Para operar em conjunto com o QPSO é utilizado a heurística *Lin-Kernighan Helsgaun* (LKH) que restringe e dirige a busca dentro do espaço de busca. Este método foi aplicado ao PCV e apresenta resultados bons, porém o tempo de execução aumentou.

Outros quatro autores apresentam o PSO com algumas mudanças aplicado ao PCV. Esses autores são: Machado e Lopes (2005), Clerc (2004), Wang (2003) e Pang (2004). Veremos esses algoritmos com mais detalhes no capítulo 2.

---

<sup>1</sup>O *Path-Relinking* é uma técnica que opera com uma população de soluções e não apenas com uma solução. Esta técnica aplica alguns procedimentos combinando as soluções e gerando novas (FRINHANI, 2011).

## **1.3 Organização do Trabalho**

No capítulo 2 será apresentado o Enxame de Partículas, sua estrutura básica e alguns dos algoritmos adaptados para o PCV. No capítulo 3 encontraremos a descrição do funcionamento do PSO aplicado ao PCV que foi escolhido, o pseudo-código do algoritmo proposto neste trabalho, os testes e as análises para comparação. No capítulo 4 encontraremos as considerações finais e algumas sugestões para trabalhos futuros.

# Capítulo 2

## Enxame de Partículas

Existe uma grande variedade de algoritmos para otimização de funções, como os baseados na teoria da evolução das espécies e algoritmos que tentam simular redes neurais. Ainda para a otimização de funções, outros algoritmos podem ser desenvolvidos, como aqueles fundamentados em comportamentos sociais e coletivos. A partir do estudo desses comportamentos sociais, realizado por Kennedy e Eberhart (1995a), foi desenvolvida a otimização por enxame de partículas (*Particle Swarm Optimization* - PSO), que observou o comportamento de pássaros, formigas, abelhas, entre outros sistemas biológicos que utilizam o trabalho coletivo para sobrevivência. Desse estudo foi possível extrair três características principais de seus comportamentos (KENNEDY; EBERHART, 2001):

**Avaliar:** É a capacidade do indivíduo em avaliar a importância da situação e tomar suas decisões.

**Comparar:** É a capacidade de examinar um conjunto de obras com certas semelhanças e aperfeiçoar seu conhecimento.

**Imitar:** É uma maneira de fazer exatamente o que o outro indivíduo está fazendo.

### 2.1 Otimização por Enxame de Partículas

O PSO contém particularidades em comum com algoritmos genéticos e pode ser usado para resolver os mesmos problemas, com a diferença de não ser necessário implementar operadores evolucionários; além disso tem como principal vantagem a simplicidade de implementação,

já que não existem muitos parâmetros para serem configurados (KENNEDY; EBERHART, 1995a).

Para compreender melhor o PSO, suponha que há uma revoada de pássaros e esses estão procurando por comida. Nenhum deles sabe exatamente onde ela está mas, conforme deslocam-se no espaço de voo, chegam mais próximos ou se distanciam dela. Em algum momento, um ou mais pássaros podem, por exemplo, perceber a presença de comida e então sabem que estão se aproximando. Os demais pássaros precisam tomar uma importante decisão: continuar se deslocando em busca do alimento através de seu próprio conhecimento e intuição ou seguir o pássaro que pode ter encontrado a comida, como mostra a figura 2.1. Devido a esta decisão, alguns pássaros podem encontrar o alimento no espaço de busca e outros não (ALOISE; OLIVEIRA; SILVA, 2006).

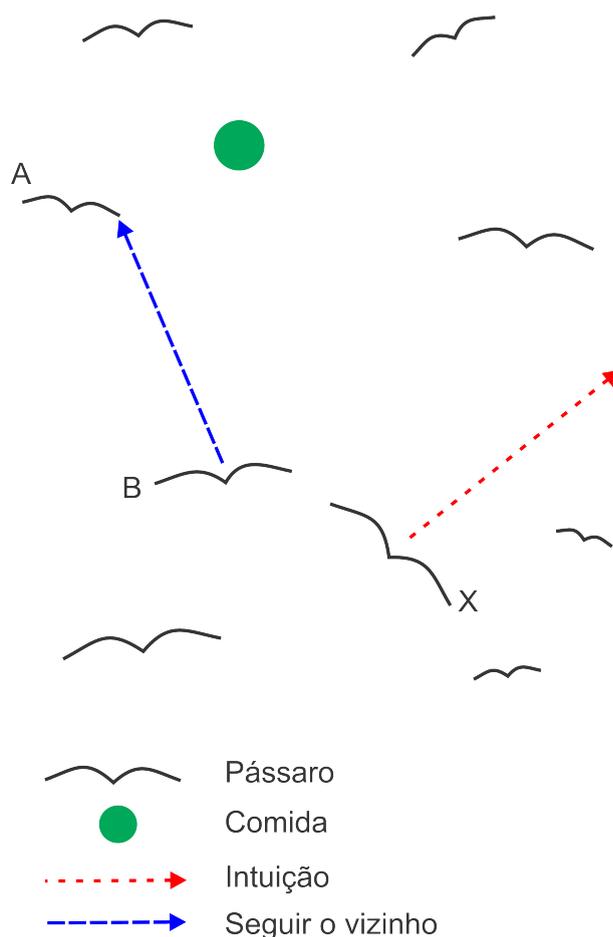


Figura 2.1: Tomada de decisão dos pássaros

Ao iniciarmos o PSO, definimos o tamanho da população que será utilizado para encontrar

uma solução e suas posições serão aleatórias dentro do intervalo da função. Com as posições das partículas já definidas, é necessário determinar sua velocidade inicial. Este valor pode ser qualquer  $\mathbb{Z}^+$ , normalmente iniciado em zero. No decorrer do processamento, a velocidade e a posição das partículas serão atualizadas pelas equações 2.1 e 2.2 respectivamente. Conforme essas atualizações acontecem, um problema pode ocorrer: o valor da posição das partículas pode ser superior ou inferior aos limites do intervalo da função; para resolver este problema, definimos uma velocidade máxima e mínima. Quando a posição de uma partícula é superior a velocidade máxima, definimos sua posição com este mesmo valor máximo, o mesmo é feito quando a partícula atinge a velocidade mínima.

$$v_i(t + 1) = v_i(t) + c1 * (P_b - x_i(t)) + c2 * (G_b - x_i(t)) \quad (2.1)$$

onde,

$v_i(t + 1)$  é a nova velocidade;

$v_i(t)$  é a velocidade atual da partícula;

$c1$  e  $c2$  são operadores cognitivos com valores aleatórios entre 0 e 1;

$P_b$  é a melhor partícula local;

$G_b$  é a melhor partícula global;

$x_i(t)$  é a posição atual da partícula.

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (2.2)$$

onde,

$x_i(t + 1)$  é a nova posição da partícula;

$x_i(t)$  é a posição atual da partícula;

$v_i(t + 1)$  é a velocidade calculada na equação 2.1.

Atualizamos agora a função *fitness* responsável por aplicar a posição de cada partícula na função. Armazenamos a melhor partícula global e então verificamos se o número máximo de iterações foi alcançado. Caso tenha sido alcançado finalizamos o laço do algoritmo, senão continuamos atualizando as partículas. A figura 2.2 mostra os passos descritos em forma de

fluxograma.

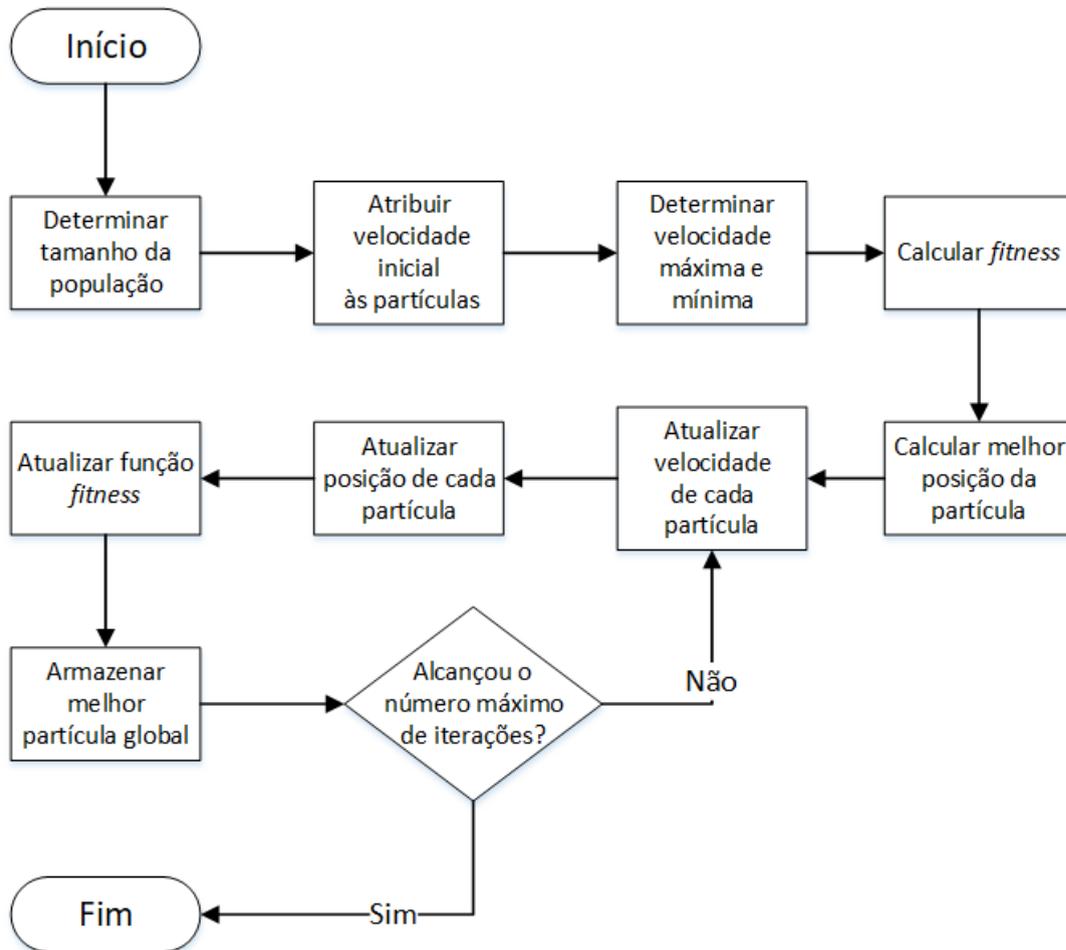
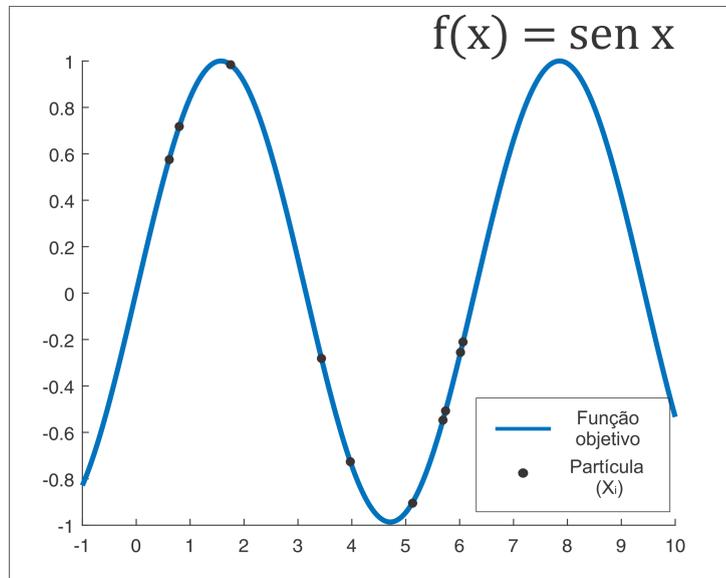
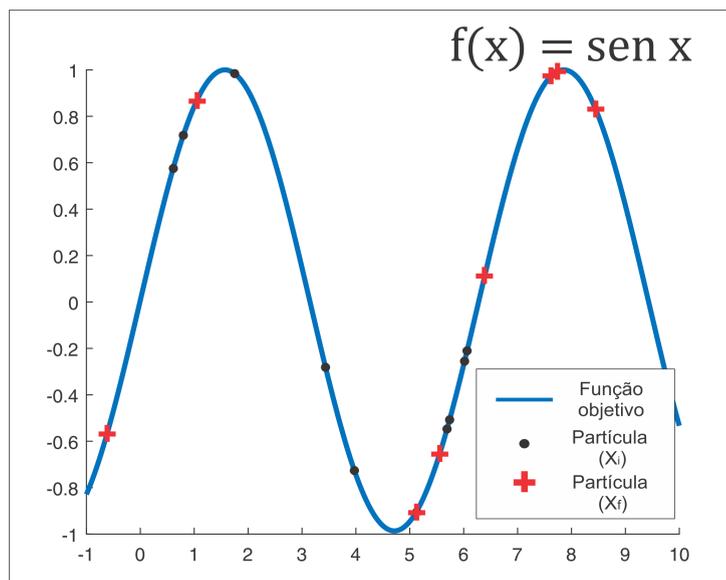


Figura 2.2: Fluxograma do algoritmo do PSO

Na figura 2.3, observamos o comportamento de dez partículas geradas aleatoriamente no espaço de busca. A figura 2.3(a) representa a posição inicial das partículas. A figura 2.3(b) representa a posição final das partículas.



(a) Partículas no início do algoritmo



(b) Partículas após aplicação do algoritmo

Figura 2.3: Aplicação do algoritmo sobre partículas aleatórias

## 2.2 PSO aplicado ao PCV

O PSO aplicado ao PCV necessita de algumas adaptações. Essas adaptações são feitas para que o PSO possa ler os grafos; as partículas já não tem mais um valor dentro da função *fitness* e sim um vetor com uma possível solução; a velocidade também deixa de ter um valor e passa a ser um vetor para troca de posições, segundo Souza (2006) essas características são importantes,

e propõe quatro soluções que podem ser adotadas e consideradas aceitáveis quando aplicadas ao PCV: o PSO de Machado e Lopes (2005), o PSO de Clerc (2004), o PSO de Wang (2003) e o PSO de Pang (2004).

O PSO proposto em Machado e Lopes (2005) é um modelo híbrido que utiliza definições de AGs. Este modelo é fortemente acoplado, pois, a característica que pode lidar com o problema é o operador OX<sup>1</sup>, utilizado em AGs. Neste modelo, as partículas são compostas por algumas informações, tais como: posição atual, velocidade atual, aptidão atual da partícula e melhor aptidão obtida até aquela iteração (*Best Local Search* - BLS). Também são definidos alguns parâmetros globais: número máximo de iterações, número de partículas, distância mínima para computação física, número de cidades do problema e a matriz de distância das cidades. Durante as iterações, é calculada a distância média entre as partículas e a melhor solução global (*Best Global Search* - BGS): se este valor for menor que 5% do número de cidades (NC), é caracterizada aglutinação. Pode haver aglutinação em torno do BLS de uma partícula; se isso acontecer, todas as partículas que estão com o BLS aglutinado, receberão um valor aleatório. Os movimentos das partículas são baseados nas equações 2.3 e 2.4 e no operador OX recombinação duas possíveis soluções.

$$X_{i+1} = X_i + V_i \quad (2.3)$$

onde,

$X_{i+1}$  é a nova posição da partícula;

$X_i$  é a posição atual da partícula;

$V_i$  é a velocidade atual da partícula.

$$V_{i+1} = c_1 * r_1.dBLS + c_2 * r_2.dBGS \quad (2.4)$$

onde,

$V_{i+1}$  é a nova velocidade da partícula;

$r_1$  e  $r_2$  são valores aleatórios no intervalo  $[0, 1]$ ;

---

<sup>1</sup>Segundo Mole (2002) o operador OX representa a forma do caminho, onde a importância está na ordem ao invés da posição das cidades.

$c_1$  e  $c_2$  são os pesos de BLS e BGS;

$d_{BLS}$  e  $d_{BGS}$  são a distância atual entre a posição e o BLS e a posição atual de BGS respectivamente.

No PSO de Clerc (2004), procuramos por ciclos hamiltonianos, sendo  $G = (V, E)$ , onde  $V$  é o conjunto de vértices e  $E$  o conjunto de arestas. Todo o elemento de  $E$  é uma tripla  $(i, j, w_{i,j})$  de forma que  $i, j \in \{1, \dots, N\}$  e  $w_{i,j} \in \mathbb{R}$ . Então avaliamos as sequências com  $N + 1$  vértices diferentes, com exceção do último elemento que é igual ao primeiro. Cada partícula armazena um conjunto solução e durante as iterações novos conjuntos são criados até que as execuções terminem e o melhor conjunto solução encontrado no decorrer do processamento é retornado como resposta.

O PSO proposto em Wang (2003) é um modelo parecido com o apresentado por Clerc (2004). A maior diferença está em que o conjunto solução não é apresentado com  $N + 1$  vértices, mas, somente  $N$ . Outra diferença são as trocas feitas pela velocidade onde é realizada a troca das posições do conjunto solução e não dos valores daquelas posições no conjunto solução.

Para o modelo de Wang (2003) é utilizado o operador *Swap* para fazer a troca de nós na solução  $S$ , sendo definido por  $S' = S + SO(i_1, i_2)$ . A sequência de *Swap*  $SS$  é constituída por um ou mais operadores. Esta sequência de trocas atua nas soluções das partículas.

O PSO proposto em Pang (2004) utiliza um número real positivo para limitar o espaço de busca que é denominado por  $P\_MAX$ , desta forma uma partícula deve estar posicionada no intervalo  $[-P\_MAX, +P\_MAX]$ . Uma velocidade máxima é denominada  $V\_MAX$  e estipulada por  $V\_MAX = a * P\_MAX$ ,  $a \in (0, 1)$ , então  $V_{id} \in [-V\_MAX, +V\_MAX]$ . Pang define  $\Omega^n$  como o espaço  $n$ -dimensional e  $\Phi^n$  é o espaço do problema do PCV para  $n$  cidades. Desta forma  $f : \Omega^n \rightarrow \Phi^n$ .

Segundo Pang (2004), o mapeamento deve satisfazer duas condições:

1. Para um vetor  $X$  qualquer,  $X \in \Omega^n$  e tem uma única permutação  $\pi$  correspondente a  $f(X) = \pi$ , onde  $\pi$  é a permutação de  $n$  inteiros.
2. O mapeamento deve utilizar esta relação entre os componentes do vetor  $X$  e indicar a

relação de ordem  $n$ .

Pang (2004) define alguns métodos para a construção do mapeamento. Uma dessas técnicas é a Transformada Rápida de Fourier.

## 2.3 Método Base para Implementação

O método apresentado por Clerc (2004) foi escolhido como base para a implementação do algoritmo do PSO e comparação com os demais métodos apresentados em Fredrich (2010). Este método foi escolhido por apresentar fortemente as características do PSO original. Segundo Clerc (2004), o PSO precisa ser estudado para que novas variações do mesmo sejam satisfatórias, portanto este método foi escolhido como base para que possamos averiguar o quão bom ele é e sugerir meios para otimizá-lo.

A função objetivo consiste em uma partícula  $x = (n_1, n_2, \dots, n_N, n_{N+1}), n_1 = n_{N+1}$ . Esta partícula  $x$  é aceita apenas se todos os arcos existirem como na figura 2.4(b) diferente da figura 2.4(a); para esses novos arcos, valores exorbitantes serão dados às suas distâncias, de forma que uma solução criada que contenha os novos arcos não será uma boa solução. Então todas as partículas tem uma possível função objetivo.

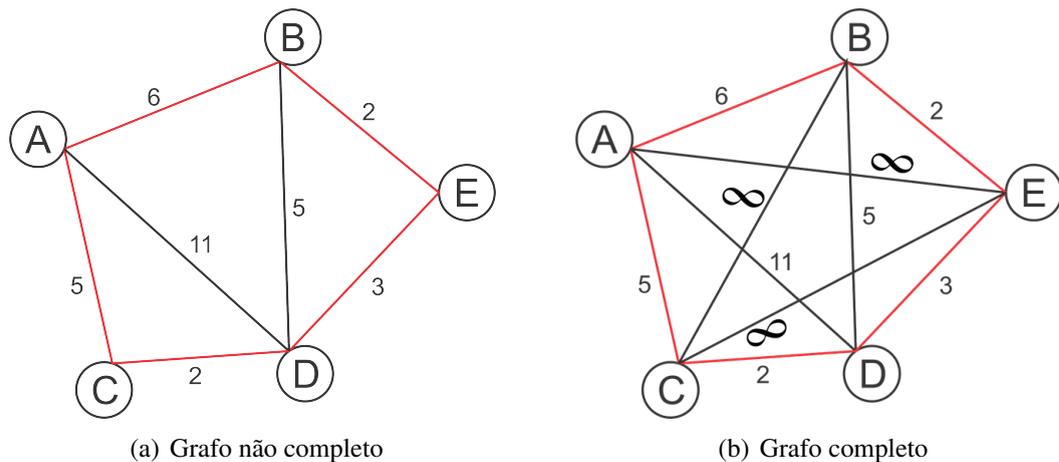


Figura 2.4: Criação de novos arcos

A equação 2.5 contém um número finito de valores e o mínimo global é a melhor solução.

$$f(x) = \sum_{i=1}^N w_{n_i, n_{i+1}} \quad (2.5)$$

A velocidade  $V_e$  é composta por uma lista de trocas que serão aplicadas às soluções obtidas. Esta velocidade é definida pela equação 2.6.

$$V_e = (i_k, j_k) \tag{2.6}$$

Onde,  $i_k, j_k \in \{1, \dots, N\}$ , tal que  $1 \leq k \leq \|v\|$ .

A movimentação da partícula  $x'$  é dada aplicando a transposição da velocidade  $V_e$  na posição  $x$ .

O exemplo a seguir, apresenta uma solução  $S$  e uma velocidade  $V_e$ .

$$S = (1, 2, 3, 4, 5, 1)$$

$$V_e = (1, 2), (2, 3)$$

Para a troca dos valores (1, 2) teremos a solução  $S = (2, 1, 3, 4, 5, 2)$ . Quando aplicamos a troca dos valores (2, 3) teremos uma nova solução  $S = (3, 1, 2, 4, 5, 3)$ .

# Capítulo 3

## Implementação e Testes

No PSO proposto por Clerc (2004), inicialmente é definido o tamanho da população (considere como exemplo a figura 2.4(b) onde temos um grafo representando cidades para o PCV). Cada partícula terá um conjunto solução, por exemplo, a partícula  $x_1$  pode ter o conjunto  $\{A, C, D, E, B, A\}$  e a partícula  $x_2$  pode ter o conjunto  $\{C, A, D, B, E, C\}$ , esses conjuntos são obtidos na primeira iteração de forma aleatória. Em seguida aplicamos a velocidade em cada partícula para que possamos calcular qual o melhor conjunto solução até o momento. Quando concluímos a primeira iteração, atualizamos a velocidade e o conjunto solução das partículas até que o número máximo de iterações seja alcançado. Em cada etapa calculamos a melhor partícula, ou seja, aquela que tem o melhor conjunto solução. A figura 3.1 apresenta um fluxograma desta descrição.

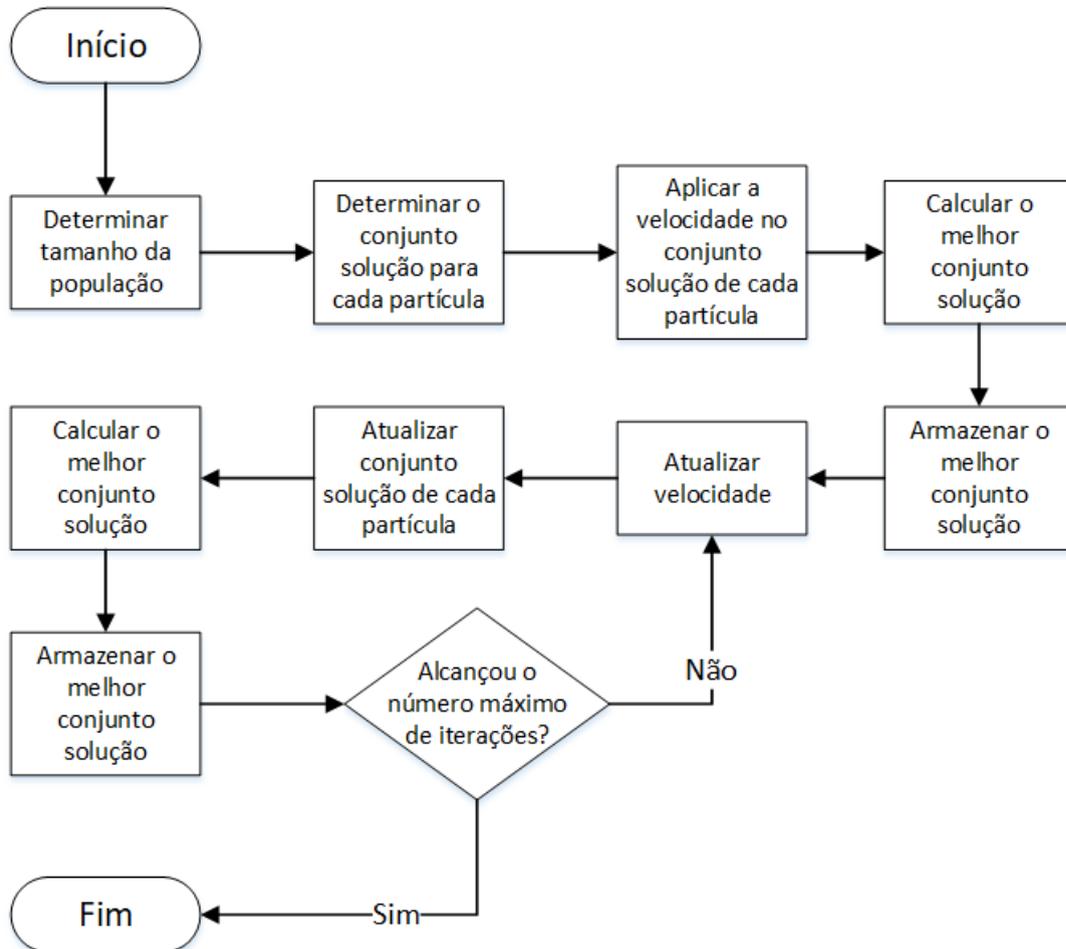


Figura 3.1: Fluxograma do algoritmo do PSO de Clerc

### 3.1 PSO Implementado

Com base no algoritmo proposto por Clerc (2004) foi desenvolvido o algoritmo 1 e o algoritmo 2 para aplicar ao problema do caixeiro viajante. A velocidade deste algoritmo pode ser descrita em três partes: a primeira parte obtemos quando temos o caminho da melhor partícula global e o caminho da partícula que está sendo atualizada, deste modo, a velocidade são as mudanças que a partícula precisa fazer para ser o caminho da melhor partícula global; a segunda parte utiliza-se o melhor caminho que a partícula já alcançou. A terceira parte é a concatenação da parte um e da parte dois da iteração anterior.

---

**Algoritmo 1** Pseudo-Código do PSO para a Primeira Iteração (Parte 1)

---

INÍCIO

```
TAM = Número de população;  
IT = Número de iterações;  
C1 = Quanto a Partícula confia em si mesma;  
C2 = Quanto a partícula confia em seus vizinhos;  
NC = Número de cidades;  
gb = Infinito; %Distancia Global  
LEIA (GRAFO);  
  
PARA i=1 até TAM  
    P(i).x = Caminho aleatório;  
    P(i).cls = Caminho da partícula;  
    PARA j=1 até NC-1  
        P(i).ls = Calcular distância do caminho da partícula;  
    FIM PARA  
    SE gb maior P(i).ls ENTÃO  
        gb = P(i).ls  
        Caminhogb = P(i).x; %Armazena melhor caminho global  
    FIM SE  
FIM PARA
```

---

---

**Algoritmo 2** Pseudo-Código do PSO para a Atualização da Velocidade (Parte 2)

---

```
PARA i=1 até IT
  PARA j=1 até TAM
    PARA a=1 até NC
      PARA b=1 até NC
        SE Caminhogb(a) igual P(j).x(b) ENTÃO
          %vg e a velocidade global
          P(j).vg(s) = a * C1;
          P(j).vg(s+1) = b * C2;
          s = s + 2;
        FIM SE
        SE P(j).cls(a) igual P(j).x(b) ENTÃO
          %vl e a velocidade local
          P(j).vl(s1) = a * C1;
          P(j).vl(s1+1) = b * C2;
          s1 = s1 + 2;
        FIM SE
      FIM PARA
    FIM PARA
  PARA q=1 variando de 2 em 2 até NC
    Fazer troca para P(j).x ser igual P(j).vg;
    Fazer troca para P(j).x ser igual P(j).vl;
    %P(j).v e a velocidade anterior
    Fazer troca para P(j).x ser igual P(j).v;
  FIM PARA
  PARA l=1 até NC-1
    distancia = Calcular distância do caminho
da partícula;
    FIM PARA
    SE distancia menor P(j).ls ENTÃO
      P(j).ls = distancia;
      P(j).cls = P(j).x;
    FIM SE
    SE gb maior P(j).ls ENTÃO
      gb = P(j).ls;
      Caminhogb = P(j).x;
    FIM SE
  FIM PARA
FIM PARA
FIM
```

---

## 3.2 Ambiente Computacional

Para a execução dos testes foi utilizado o sistema Matlab - R2015a, 64-bits, de 12 de fevereiro de 2015, em uma máquina com as seguintes configurações:

- Processador Intel i7 4510U - 2.6 GHz;
- Memória Cache:
  - Dividida em três níveis com aproximadamente 5Mb.
- Memória RAM: DDR 3 - 8 GB;
- Capacidade de Disco 1 TB Híbrido;
- Sistema Operacional - Windows 10 Home Single Language 64 bits.

## 3.3 Testes

Para a realização dos testes foram utilizados os arquivos do TSPLIB escolhidos por Fredrich (2010) com número de cidades entre 51 e 1060. Os arquivos usados são: eil51.tsp; rat99.tsp; bier127.tsp; kroA200.tsp; linhp318.tsp; pr439.tsp; rat575.tsp; p654.tsp; rat783.tsp; u1060.tsp (o número contido em cada arquivo representa a quantidade de cidades). Todos os arquivos são grafos simétricos e completos.

A aplicação do algoritmo do PSO para cada grafo, obteve os resultados apresentados na tabela 3.1, cada arquivo foi executado dez vezes, desta forma os valores da coluna do PSO são a média dessas execuções, assim como a coluna tempo, no apêndice A encontra-se o melhor caminho percorrido pelo PSO para cada arquivo executado. Já os valores contidos na coluna diferença podem ser calculados com a equação 3.1. Durante a execução dos testes para os arquivos eil51.tsp, rat99.tsp, bier127.tsp, kroA200.tsp, linhp318.tsp notou-se uma demora excessiva e as soluções encontradas não foram satisfatórias, desse modo, a execução dos demais arquivos não foi realizada. Os parâmetros utilizados para as execuções foram: População = 600; Iterações = 200; Coeficientes:  $c1 = 0,5$  e  $c2 = 8$ .

$$\frac{(PSO - MelhorSolucao) * 100}{MelhorSolucao} \quad (3.1)$$

Tabela 3.1: Resultados para o PSO aplicado ao PCV

Arquivo	Melhor Solução	PSO	Tempo (Seg.)	Diferença (%)
eil51.tsp	426,0000	1206,3300	366,0046	183,18
rat99.tsp	1211,0000	6530,4300	1176,7616	439,26
bier127.tsp	118282,0000	529210,0000	1892,2971	347,41
kroA200.tsp	29368,0000	286476,0000	4453,9178	875,47
linhp318.tsp	41345,0000	523830,0000	10645,7168	1166,97

Observando os dados na tabela 3.2 e com o auxílio da figura 3.2, nota-se que o algoritmo mencionado na seção 3.1 não foi capaz de encontrar soluções próximas ao melhor resultado e nem soluções próximas aos algoritmos VMP, AG e AM.

Tabela 3.2: Comparação entre o VMP, AG, AM e PSO aplicados ao PCV

Arquivo	Melhor Resultado	VMP	AG	AM	PSO
eil51.tsp	426,0000	495,4429	446,8852	433,5419	1206,3300
rat99.tsp	1211,0000	1383,7725	1303,8663	1282,4055	6530,4300
bier127.tsp	118282,0000	136884,6655	123748,1599	123748,1599	529210,0000
kroA200.tsp	29368,0000	37117,8222	33852,8857	32903,0496	286476,0000
linhp318.tsp	41345,0000	50357,0873	47653,0490	47653,0490	523830,0000

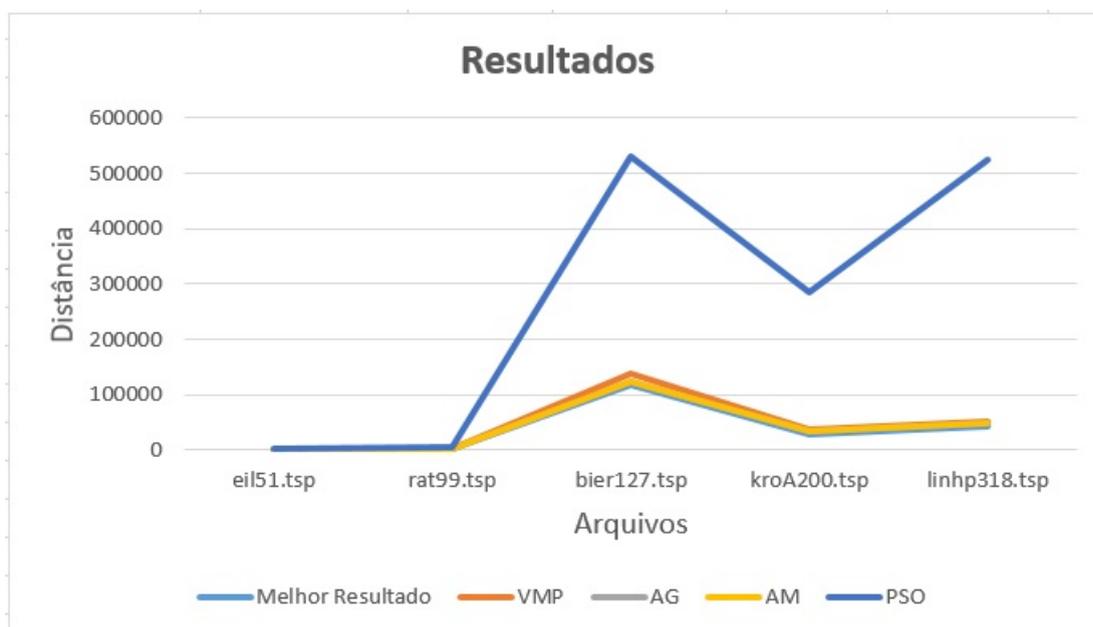


Figura 3.2: Comparação entre a melhor solução e os algoritmos VMP, AG, AM e PSO

### 3.4 Análises

Como dito anteriormente, o algoritmo proposto não alcançou bons resultados, tanto na distância dos arquivos quanto no tempo de execução que se mostrou inviável. Para melhorar o tempo de execução, algumas medidas podem ser tomadas para a otimização do mesmo, visto que este algoritmo tem complexidade  $\theta(n^4)$ . Uma sugestão para melhorar esta complexidade seria não percorrer o vetor inteiro na velocidade e diminuir o tamanho do laço a cada passo.

A análise realizada apresenta a comparação entre os algoritmos apresentados por Fredrich (2010) com o PSO separadamente. A primeira comparação é entre o PSO e o AG. Na tabela 3.3 é possível verificar a diferença em porcentagem entre os algoritmos, onde o PSO se mostrou muito inferior ao AG.

Tabela 3.3: Diferença entre o PSO e o AG

Arquivo	AG	PSO	Diferença (%)
eil51.tsp	446,8852	1206,3300	169,94
rat99.tsp	1303,8663	6530,4300	400,85
bier127.tsp	123748,1599	529210,0000	327,65
kroA200.tsp	33852,8857	286476,0000	746,24
linhp38.tsp	47653,0490	523830,0000	999,26

A tabela 3.4 mostra a diferença entre o PSO e o AM, segundo Fredrich (2010) o AM apresentou resultados melhores que o AG, embora o tempo de execução seja maior. Logo o PSO também mostrou-se inferior ao AM.

Tabela 3.4: Diferença entre o PSO e o AM

Arquivo	AM	PSO	Diferença (%)
eil51.tsp	433,5419	1206,3300	178,25
rat99.tsp	1282,4055	6530,4300	409,23
bier127.tsp	123748,1599	529210,0000	327,65
kroA200.tsp	32903,0496	286476,0000	770,67
linhp38.tsp	47653,049	523830,0000	999,26

Por fim, a tabela 3.5 apresenta a diferença entre o PSO e o VMP. Para Fredrich (2010) o VMP foi o pior algoritmo. O PSO foi inferior a ele.

Esta grande diferença entre os métodos propostos por Fredrich (2010) e o PSO, ocorre porque o algoritmo implementado baseado no PSO de Clerc (2004) não consegue convergir,

Tabela 3.5: Diferença entre o PSO e o VMP

Arquivo	VMP	PSO	Diferença (%)
eil51.tsp	495,4429	1206,3300	143,49
rat99.tsp	1383,7725	6530,4300	371,93
bier127.tsp	136884,6655	529210,0000	286,61
kroA200.tsp	37117,8222	286476,0000	671,80
linhp38.tsp	50357,0873	523830,0000	940,23

o espaço de busca do PSO original é euclidiano, quando utilizamos o PSO para o PCV, o espaço de busca deixa de ser euclidiano e passa a ter um espaço de busca próprio e não vetorial, desta forma não é possível apontar para a solução ótima, a velocidade que é responsável por direcionar as partículas para a melhor solução através de duas listas de transposição, não consegue aplicar um direcionamento eficaz, além de tornar o algoritmo lento como mencionado por Hoffmann (2011). Outro problema observado durante as execuções, foi a aglutinação, ou seja, as partículas estão convergindo para um ótimo local, o que faz com que poucas delas se aproximem do ótimo global, então o algoritmo demora para encontrar a solução ótima ou neste caso, não consegue encontrar o ótimo global, obtendo assim performances ruins.

## Capítulo 4

# Considerações Finais e Trabalhos Futuros

Neste trabalho o PSO implementado não foi capaz de alcançar soluções ótimas e próximas aos resultados obtidos em Fredrich (2010). Como contribuição temos a teoria do PSO original e do PSO para o PCV, os resultados apresentados mostram a necessidade de métodos para a otimização na distância encontrada para os grafos pelo algoritmo apresentado neste trabalho. O pseudo-código apresentado no capítulo 3 contribui para o entendimento do PSO aplicado ao PCV. Em Clerc (2004), o autor diz que o PSO tem muito a oferecer, mas até que encontremos os melhores métodos e estratégias, vários estudos serão necessários e muitas otimizações precisam ser realizadas no método do enxame de partículas para a obtenção de melhores resultados com a técnica. Souza (2006) cita algumas áreas de aplicação para o PSO, como o treinamento de redes neurais, programação de sistemas de manufatura e outros, onde seus resultados foram positivos e em alguns casos os resultados foram melhores que algoritmos já conhecidos.

Para trabalhos futuros sugere-se o aperfeiçoamento do algoritmo do PSO implementado neste trabalho, utilizando outros parâmetros, novos métodos e estratégias. Algumas sugestões para melhorar o tempo de execução podem ser encontradas em Hoffmann (2011), visto que o PSO proposto por Clerc (2004) é lento devido principalmente a comparação entre dois vetores para a troca de cidades. Para a velocidade que não possui um direcionamento, alguns métodos podem ser adotados, como a busca local e o *path-relinking* utilizados por Souza (2006). Outro método utilizado pode ser a heurística *Lin-Kernighan Helsgaun* usada por Herrera (2007). Além de melhorias aplicadas ao algoritmo apresentado neste trabalho, há outros métodos para serem estudados, sendo eles: o PSO de Machado e Lopes (2005), Wang (2003) e Pang (2004).

## Apêndice A

# Melhores Percursos Encontrados para o PSO dos Arquivos da Biblioteca TSPLIB

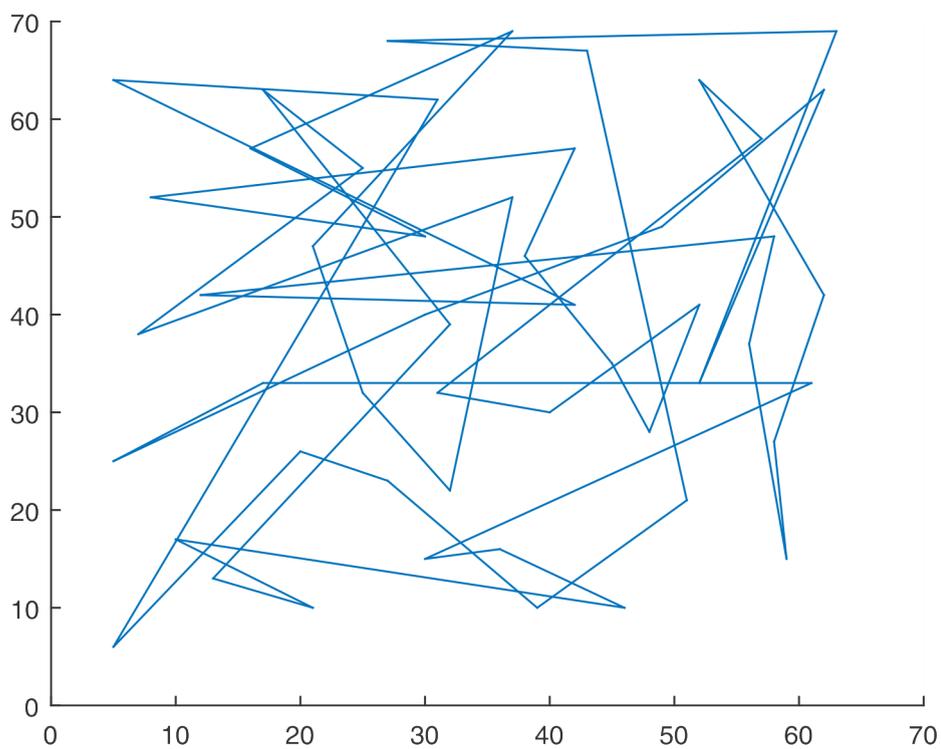


Figura A.1: Percurso obtido com o PSO para a entrada eil51.tsp

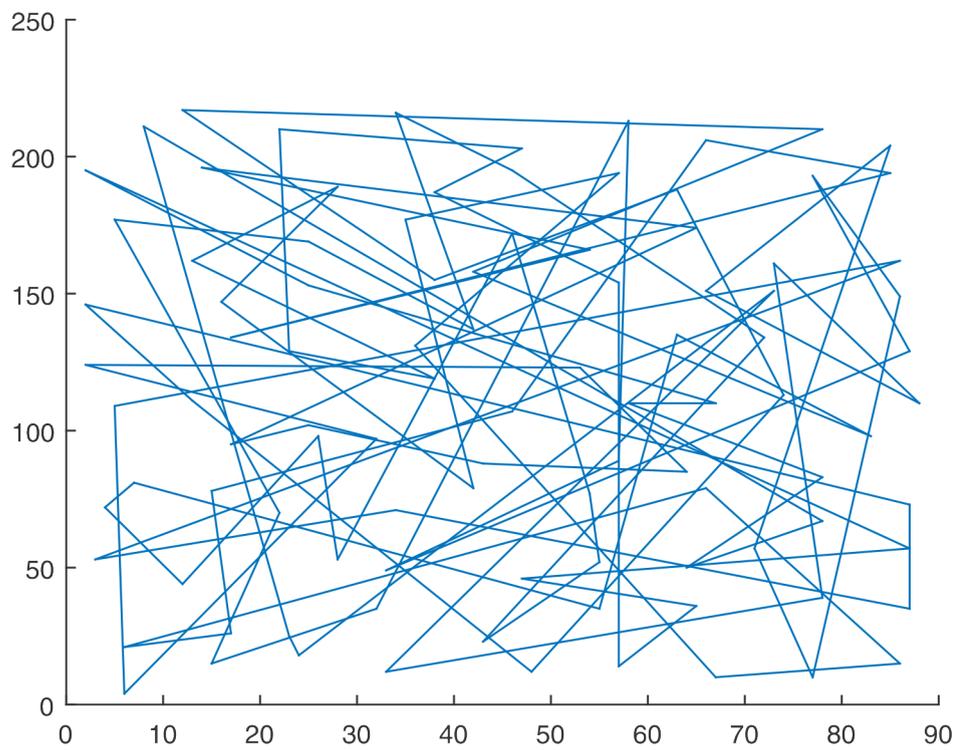


Figura A.2: Percurso obtido com o PSO para a entrada rat99.tsp

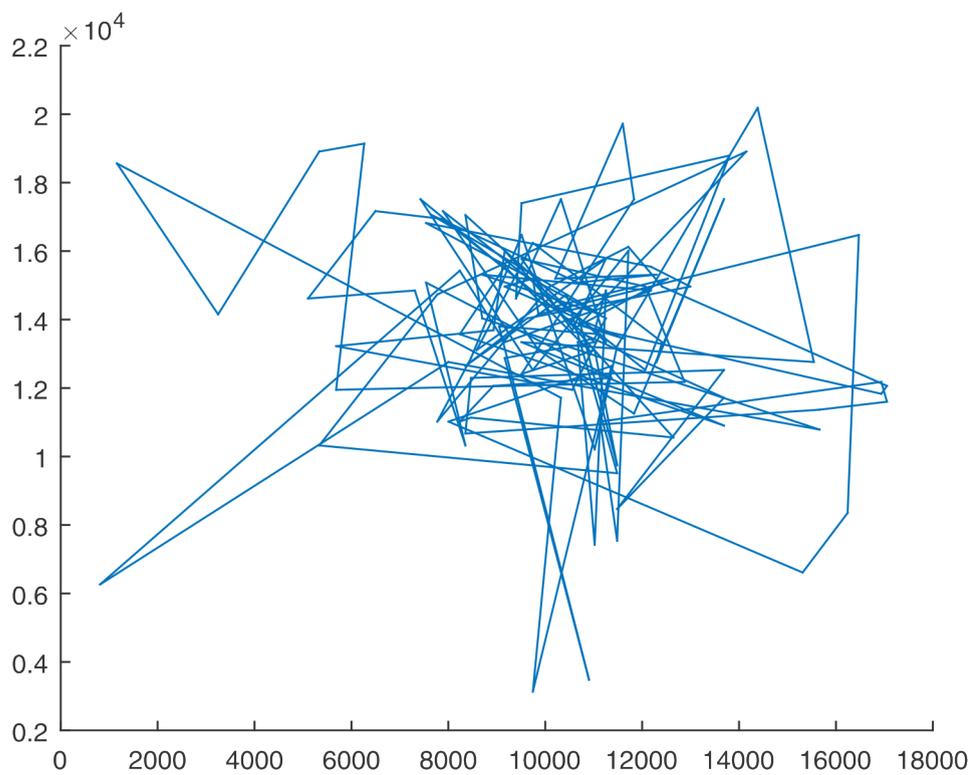


Figura A.3: Percurso obtido com o PSO para a entrada bier127.tsp

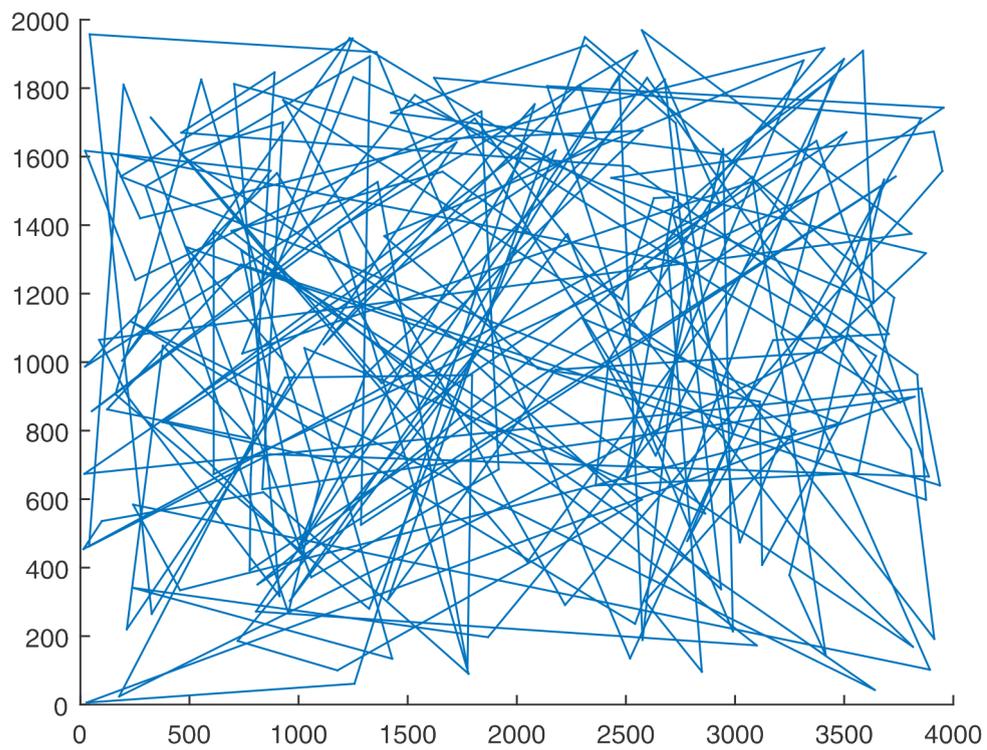


Figura A.4: Percurso obtido com o PSO para a entrada kroA200.tsp

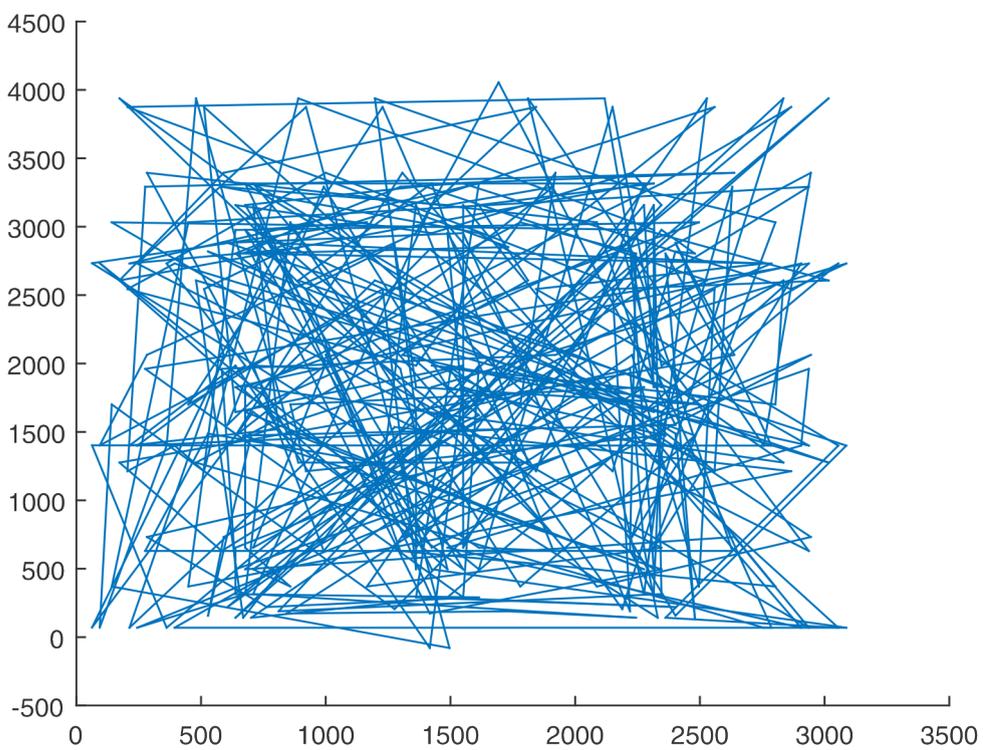


Figura A.5: Percurso obtido com o PSO para a entrada linhp318.tsp

# Referências Bibliográficas

- ALOISE, D. J.; OLIVEIRA, M. C. S.; SILVA, T. L. Otimização discreta por nuvem de partículas aplicada ao problema do caixeiro viajante. *Gepros*, São Paulo, v. 1, n. 2, p. 87–95, Abril 2006.
- CLERC, M. Particle swarm optimization, illustrated by traveling salesman problem. In: \_\_\_\_\_. Heidelberg: Springer Berlin Heidelberg, 2004. cap. 08: New Optimization Techniques in Engineering, p. 219–239.
- FREDRICH, A. P. *Um Estudo Sobre Algoritmos Meméticos e sua Eficiência em Relação aos Algoritmos Genéticos*. Dissertação (Dissertação de Graduação) — Universidade Estadual do Oeste do Paraná, Cascavel - PR, Novembro 2010.
- FRINHANI, R. de M. D. *Grasp com Path-Relinking para Agrupamento de Dados Biológicos*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte - MG, Março 2011.
- HERRERA, B. A. L. de M. *Combinação de Enxame de Partículas com Inspiração Quântica e Método Linkernighan-Helsgaun aplicada ao Problema do Caixeiro Viajante*. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Paraná, Curitiba - PR, 2007.
- HOFFMANN, M. Discrete particle swarm optimization for tsp: Theoretical results and experimental evaluations. In: *Adaptive and Intelligent Systems: Second International Conference*. Austria: [s.n.], 2011. p. 416–427.
- HOLLAND, J. *Adaptation in Natural and Artificial Systems*. 1. ed. University of Michigan: Ann Arbor, 1975.
- JERALD, J. et al. Scheduling optimization of flexible manufacturing systems using particle swarm optimization algorithm. *Advanced Manufacturing Technology*, Londres, p. 964–971, Março 2004.
- KENNEDY, J.; EBERHART, R. A new optimizer using particle swarm theory. In: *Sixth International Symposium on Micro Machine and Human Science*. Nagoya: [s.n.], 1995. p. 39–43.
- KENNEDY, J.; EBERHART, R. Particle Swarm Optimization. Neural Networks. In: *IEEE International Conference*. Perth: [s.n.], 1995. p. 1942–1948.
- KENNEDY, J.; EBERHART, R. A. *Swarm Intelligence*. 1. ed. San Francisco: Morgan Kaufmann, 2001.

- LEVITIN, A. *Introduction to The Design & Analysis of Algorithms*. 1. ed. Boston: Addison Wesley, 2003.
- MACHADO, T. R.; LOPES, H. S. A hybrid particle swarm optimization model for the traveling salesman problem. In: *Adaptative and Natural Computing Algorithms*. Vienna: [s.n.], 2005. p. 255–258.
- MOLE, V. L. D. *Algoritmos Genéticos - Uma Abordagem Paralela Baseada em Populações Cooperantes*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Santa Catarina, Florianópolis - SC, Dezembro 2002.
- MOSCATO, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program*, v. 1, n. 826, 1989.
- PANG, W. Modified particle swarm optimization based on space transformation for solving traveling salesman problem. In: *Third International Conference on Machine Learning and Cybernetics*. Shanghai: [s.n.], 2004. p. 2342–2346.
- SECRET, B. R. *Traveling Salesman Problem for Surveillance Mission Using Particle Swarm Optimization*. Dissertação (Dissertação de Mestrado) — Air Force Institute of Technology, Ohio, Março 2001.
- SOUZA, G. R. *Uma Abordagem por Nuvem de Partículas para Problemas de Otimização Combinatória*. Dissertação (Dissertação de Mestrado) — Universidade Federal do Rio Grande do Norte, Natal - RN, Fevereiro 2006.
- WANG, K. Particle swarm optimization for traveling salesman problem. In: *Second International Conference on Machine Learning and Cybernetics*. Xi'an: [s.n.], 2003. p. 1583–1585.