



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS



## UM MÉTODO MULTIOBJETIVO APLICADO AO PROBLEMA DO CAIXEIRO VIAJANTE COM LUCRO

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA  
(PIBIC/CNPq/INPE)

Carla Cristina Doescher Fernandes (UNIFESP, Bolsista PIBIC/CNPq)  
E-mail: c.fernandes11@unifesp.br

Prof. Dr. Luiz Antonio Nogueira Lorena (LAC/INPE, Orientador)  
E-mail: lorena@lac.inpe.br

### COLABORADORES

Prof. Dr. Antônio Augusto Chaves (ICT/UNIFESP)  
Prof. Dra. Kelly Cristina Poldi (ICT/UNIFESP)

INPE  
São José dos Campos  
Julho de 2012

**PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE  
Gabinete do Diretor (GB)  
Serviço de Informação e Documentação (SID)  
Caixa Postal 515 - CEP 12.245-970  
São José dos Campos - SP - Brasil  
Tel.:(012) 3208-6923/6921  
Fax: (012) 3208-6919  
E-mail: pubtc@sid.inpe.br

**CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):**

Presidente:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dra. Inez Staciari Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dra. Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

**BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

**REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

**EDITORAÇÃO ELETRÔNICA:**

Vivéca Sant'Ana Lemos - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



## **UM MÉTODO MULTIOBJETIVO APLICADO AO PROBLEMA DO CAIXEIRO VIAJANTE COM LUCRO**

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA  
(PIBIC/CNPq/INPE)

Carla Cristina Doescher Fernandes (UNIFESP, Bolsista PIBIC/CNPq)  
E-mail: c.fernandes11@unifesp.br

Prof. Dr. Luiz Antonio Nogueira Lorena (LAC/INPE, Orientador)  
E-mail: lorena@lac.inpe.br

### **COLABORADORES**

Prof. Dr. Antônio Augusto Chaves (ICT/UNIFESP)  
Prof. Dra. Kelly Cristina Poldi (ICT/UNIFESP)

INPE  
São José dos Campos  
Julho de 2012

## RESUMO

A classe de problemas do Caixeiro Viajante com Lucros (TSPP, do inglês *Traveling Salesman Problem with Profits*), associa a cada cliente um valor de prêmio (lucro) a ser ganho quando este for visitado, assim o TSPP pode ser visto como um problema do caixeiro viajante com dois objetivos opostos, um que pressiona o caixeiro a viajar (ou seja, maximizar os prêmios coletados) e outro que estimula o caixeiro a minimizar os custos de viagem (permitindo a ele não visitar alguns clientes). Desta classe advém um problema conhecido como Problema do Vendedor com Multiobjetivos (MVP, do inglês *Multiobjective Vending Problem*) que trata desses dois objetivos separadamente. Neste trabalho aplica-se o método Busca por Agrupamentos (CS, do inglês *Clustering Search*) na solução deste problema biobjetivo. Para tanto, foi proposto o PCS (*Pareto Clustering Search*) uma variação do CS para a solução heurística do problema sob a visão multiobjetivo.

## LISTA DE FIGURAS

	<u>Pág.</u>
Figura 3.1 - Dominância de Pareto.....	6
Figura 4.1 - Fluxograma do método CS.....	11
Figura 5.1 - Representação da solução .....	12
Figura 6.1 - Gráfico de comparação PSA e PCS .....	15

## LISTA DE TABELAS

	<b><u>Pág.</u></b>
Tabela 6.1 - Tempo de execução PCS e PSA .....	16

## LISTA DE SIGLAS E ABREVIATURAS

INPE	Instituto Nacional de Pesquisas Espaciais
TSP	<i>Traveling Salesman Problem</i>
TSPP	<i>Traveling Salesman Problem with Profits</i>
MVP	<i>Multiobjective Vending Problem</i>
CS	<i>Clustering Search</i>
PCS	<i>Pareto Clustering Search</i>
PSA	<i>Pareto Simulated Annealing</i>
VND	<i>Variable Neighborhood Descent</i>
PR	<i>Path-Relinking</i>

## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
<b>2 PROBLEMA DO CAIXEIRO VIAJANTE COM LUCRO .....</b>	<b>3</b>
2.1. Formulação Matemática .....	4
<b>3 ALGORITMO MULTIOBJETIVOS .....</b>	<b>6</b>
3.1. <i>Pareto Simulated Annealing</i> (PSA).....	7
<b>4 BUSCA POR AGRUPAMENTOS .....</b>	<b>9</b>
<b>5 APLICAÇÃO DO PCS AO MVP.....</b>	<b>12</b>
<b>6 RESULTADOS .....</b>	<b>15</b>
<b>7 CONCLUSÃO.....</b>	<b>17</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>18</b>



## 1 INTRODUÇÃO

O Problema do Caixeiro Viajante referido na literatura como *Traveling Salesman Problem* (TSP) [16] é um dos mais tradicionais problemas de otimização combinatória. O TSP consiste em otimizar a sequência de visitas a clientes a partir de um depósito central, sendo que todos clientes precisam ser visitados, e conseqüentemente, nenhum valor é associado ao serviço de atendimento ao cliente. Porém, algumas generalizações deste problema propõem selecionar clientes dependendo de um valor de prêmio (benefício) que é ganho quando a visita acontecer. Esta característica dá origem a uma classe de problemas que foi nomeada *Traveling Salesman Problems with Profits* (TSPP) ou Problemas do Caixeiro Viajante com Lucros [7].

Os TSPP podem ser vistos como problemas do caixeiro viajante biobjetivos. Na prática, a maioria das pesquisas existentes sobre estes problemas trata-os como sendo versões de um único objetivo. Assim, ou os dois objetivos são calculados e combinados linearmente, ou um dos objetivos é transformado em restrição com um valor limite especificado. Porém, este trabalho propõe solucionar um problema oriundo da classe TSPP que trata dos dois objetivos separadamente, tal problema é denominado Problema do Vendedor com Multiobjetivos (MVP, do inglês *Multiobjective Vending Problem*) [13]. Desta forma, resolver o MVP resulta em encontrar uma fronteira de Pareto, ou seja, um conjunto de soluções viáveis tal que nenhum objetivo possa ser melhorado sem deteriorar o outro. Neste trabalho aplica-se o método Busca por Agrupamentos (CS, do inglês *Clustering Search*) [4], na solução deste problema biobjetivo. Para isto, foi proposto o PCS (*Pareto Clustering Search*) que visa combinar meta-heurísticas e heurísticas de busca local multiobjetivo, intensificando o processo de busca somente em regiões consideradas promissoras.

O restante deste trabalho está organizado da seguinte forma. No capítulo 2 apresenta-se uma formulação matemática para o TSPP e faz-se uma revisão bibliográfica. No capítulo 3, define-se métodos multiobjetivos. No capítulo 4

descreve-se a meta-heurística CS e sua adequação ao PCS. No capítulo 5 discute-se a aplicação do PCS ao MVP . No capítulo 6 apresentam-se os resultados computacionais obtidos. No capítulo 7 são apresentadas algumas conclusões deste trabalho.

## 2 O PROBLEMA DO CAIXEIRO VIAJANTE COM LUCRO

O problema do Caixeiro Viajante com Lucro (TSPP, do inglês *Traveling Salesman Problem with Profits*) [3] atribui para cada cidade um prêmio (lucro) que o caixeiro recebe caso visite a cidade ( não sendo obrigado a visitar todas cidades). Sendo assim, TSPP é um problema biobjetivo, em que um dos objetivos é minimizar os custos da viagem e o outro consiste em maximizar o lucro obtido.

O TSPP de uma forma geral, pode ser representado em um grafo completo  $G = (V,A)$ , onde  $V = \{0, 1, \dots, n\}$  é um conjunto de  $n$  vértices e  $A$  é um conjunto de arestas (grafo não direcionado). Para cada vértice  $i \in V$  existe associado um prêmio  $p_i$ , e cada aresta  $(i, j) \in A$  possui um custo de deslocamento  $c_{ij}$ . Supondo que o vértice 0, sem perda de generalidade, seja o depósito ou a cidade de origem do caixeiro, este vértice deve ter prêmio nulo ( $p_0 = 0$ ). O TSPP consiste em determinar um circuito elementar que contenha o vértice 0, levando em consideração o prêmio coletado e os custos de deslocamento.

Os diferentes problemas que formam o TSPP surgem das diferentes maneiras que existem para tratar os dois objetivos:

1. Os dois objetivos são tratados separadamente, gerando um problema multiobjetivo, onde os objetivos são, minimizar os custos de deslocamento e maximizar prêmios coletados. Este problema é conhecido como *Multiobjective Vending Problem* (MVP) [13];
2. Ambos os objetivos são combinados numa função objetivo, visando encontrar uma rota que minimize os custos de deslocamento menos os prêmios coletados. Em [6] apresenta-se esta versão como sendo o *Profitable Tour Problem* (PTP), sendo que, ao invés de coletar um prêmio por visitar uma cidade, o caixeiro paga uma penalidade caso deixe de visitar alguma cidade.
3. O objetivo do custo de deslocamento é definido como uma restrição, e o objetivo é encontrar uma rota que maximize o prêmio coletado tal que o custo

de deslocamento não exceda um valor máximo. Este problema é chamado de *Orienteering Problem* (OP) [17].

4. O objetivo do prêmio é definido como uma restrição, e o objetivo é encontrar uma rota que minimize os custos de deslocamento e que o prêmio coletado não seja menor que um valor pré-definido. Este problema é definido como *Prize-Collecting TSP* (PCTSP) [3] onde também é inserido o conceito de penalidades, ou *Quota TSP* (QTSP) [2].

## 2.1. Formulação Matemática

É possível definir um conjunto de restrições básicas para as formulações matemáticas dos TSPs. Considere  $x_{ij}$  ( $i, j \in V, i \neq j$ ) sendo uma variável binária igual a 1 se a aresta  $(i, j)$  pertencer à solução e  $x_{ij}$  igual 0 caso contrário, e uma variável binária  $y_i$  ( $i \in V$ ) que controla se o vértice  $i$  está presente na solução, assumindo valor 1 caso seja visitado e valor 0 caso contrário.

Todos TSP compartilham as restrições a seguir :

$$\sum_{j \in \mathcal{V}_i} x_{ij} = y_i \quad \forall i \in V \quad (2.1)$$

$$\sum_{i \in \mathcal{V}_j} x_{ij} = y_j \quad \forall j \in V \quad (2.2)$$

$$y_0 = 1 \quad (2.3)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in V \quad (2.4)$$

$$y_i \in \{0,1\} \quad \forall i \in V \quad (2.5)$$

As restrições (2.1) e (2.2) são chamadas restrições de atribuição e garantem que cada vértice seja visitado no máximo uma vez. A restrição (2.3) assegura

que o depósito seja visitado, as restrições (2.4) e (2.5) asseguram que as variáveis  $x_{ij}$  e  $y_i$  sejam binárias.

A função objetivo, no entanto, é diferente para cada problema da classe TSPP. Este projeto trata-se do MVP que possui duas funções objetivos, descritas a seguir:

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2.6)$$

$$\max \sum_{i \in V} p_i y_i \quad (2.7)$$

sujeito a (2.1 - 2.5).

A função objetivo (2.6) busca diminuir o custo de deslocamento do caixeiro, enquanto a (2.7) visa aumentar o lucro.

Keller e Goodchild [13] apresentam uma abordagem para solucionar o MVP que consiste em resolver sequencialmente versões do problema com um único objetivo, tal tratamento não procura encontrar um conjunto de soluções não-dominadas.

### 3 ALGORITMOS MULTIOBJETIVOS

Um algoritmo multiobjetivo consiste em minimizar e/ou maximizar simultaneamente um conjunto de critérios (objetivos) satisfazendo restrições. Nesse contexto, na otimização multiobjetivo não é encontrado apenas uma solução que otimize todos os objetivos, mas uma variedade delas onde nenhuma solução é melhor que a outra solução em todos os objetivos, ou seja, soluções que não são dominadas por outras soluções. Essa variedade de soluções é chamada de soluções Pareto-ótimas.

Suponha A, B, C, D, E, F e G soluções de um problema de otimização com dois objetivos de minimização ( $f_1$  e  $f_2$ ), a Figura 3.1 ilustra uma possível relação de dominância entre elas.

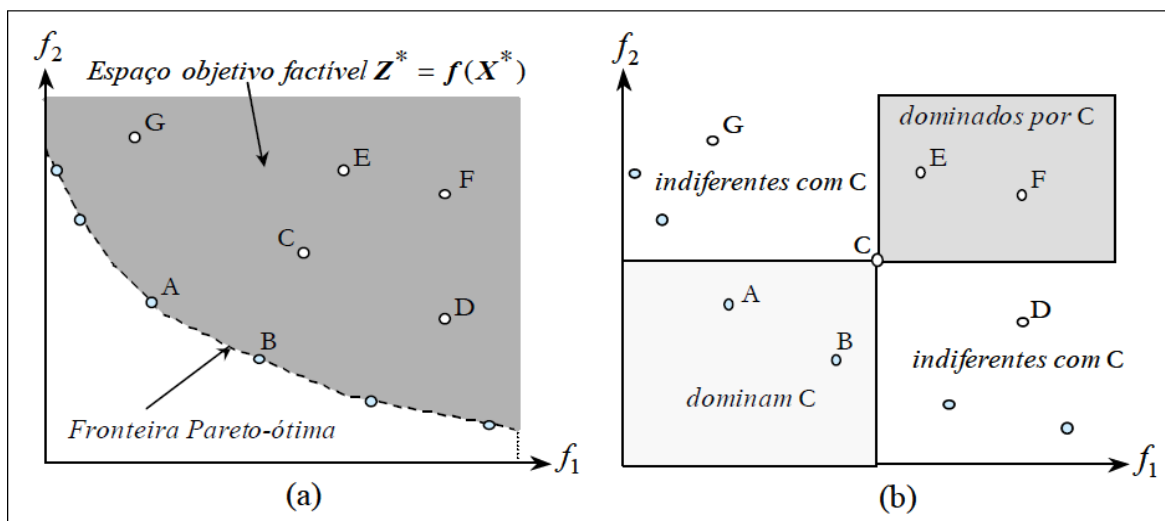


Figura 3.1 - Dominância de Pareto.

Fonte: Arroyo [1]

As soluções E e F são maiores do que C nos dois objetivos e portanto, C domina estas soluções. Por outro lado, as soluções A e B são menores do que C em ambos objetivos e sendo assim, A e B dominam C.

### 3.1. Pareto Simulated Annealing (PSA)

O *Pareto Simulated Annealing* (PSA) [15] é um método multiobjetivo que utiliza-se de uma busca local probabilística.

O PSA inicia seu processo a partir de um conjunto de soluções iniciais aleatórias. Então, é criado um *loop* que gera aleatoriamente, para cada solução corrente  $x$  um único vizinho  $y$ .

No PSA existem três condições quando a nova solução  $y$  e a solução corrente  $x$  são comparadas :

- $y$  domina ou é igual a  $x$ ,
- $y$  é dominado por  $x$ ,
- $y$  é não-dominado com relação à  $x$  ( $y$  é indiferente a  $x$ ).

Na primeira situação,  $y$  domina ou é igual a  $x$ , então a nova solução  $y$  é aceita com probabilidade igual a 1. Na segunda situação, a nova solução é considerada pior que a solução corrente e é aceita com probabilidade menor que 1. Existem várias regras de agregação para tratar a terceira situação, a regra utilizada na implementação do algoritmo é a denominada regra SL que é interpretada como uma agregação local de todos os objetivos com uma função de escala linear. Tal regra é definida pela seguinte expressão:

$$P(x, y, T, \Delta) = \min \{1, \exp (-\sum_j \lambda_j (f_j(x) - f_j(y)/T))\} \quad (3.1)$$

Onde :

- $T$  é a temperatura;
- $x$  é a solução corrente;
- $y$  é a nova solução gerada a partir de  $x$ ;
- $\Delta = (\lambda_1, \dots, \lambda_n)$  é o vetor de pesos; e
- $f_j$  é a função objetivo  $j$ .

Primeiramente  $T$  assume um valor elevado  $T_0$  e após certo número de iterações a temperatura decai gradativamente por uma razão de resfriamento  $\alpha$ , tal que  $T_k = \alpha * T_{k-1}$ , onde  $0 < \alpha < 1$ .

Os vetores de pesos são primeiramente, gerados aleatoriamente e depois são modificados iteração por iteração. Para cada solução gerada existe um vetor de pesos associada a ela. Estes vetores permitem influenciar a direção da busca no espaço objetivo para uma solução gerada em particular. O vetor de pesos associados com cada solução gerada  $x$  é modificado na tentativa de aumentar a probabilidade de mover  $x$  na direção do vizinho mais próximo  $x'$ . Para isso é feito um incremento dos pesos quando  $x$  é melhor que  $x'$  e um decrementado dos pesos quando  $x$  é pior que  $x'$ .

O PSA efetua uma exploração da fronteira de Pareto, encontrando um grande número de solução eficientes (não dominadas), essa meta-heurística multiobjetiva, além de minimizar a distância do conjunto dominante encontrado ao conjunto Pareto-Ótimo, obtém uma boa distribuição das soluções no conjunto dominante gerado.



#### 4 PARETO CLUSTERING SEARCH (PCS)

O método Busca por Agrupamentos (CS, do inglês, *Clustering Search*) [4] é um método híbrido que combina adequadamente meta-heurísticas e heurísticas de busca local. A busca é intensificada somente em regiões do espaço de busca que sejam consideradas promissoras. Este projeto propõe uma nova abordagem do CS, o PCS (*Pareto Clustering Search*) que possui a mesma proposta do CS, porém sob uma visão multiobjetivo.

O CS objetiva sofisticar o processo de escolha de soluções para aplicar busca local, ao invés de escolher aleatoriamente ou aplicar busca local em todas as soluções geradas por uma meta-heurística. Assim, espera-se uma melhoria no processo de convergência associado a uma diminuição no esforço computacional em virtude do emprego mais racional dos métodos de busca local.

O CS divide o espaço de busca e localiza regiões promissoras por meio do enquadramento dessas em *clusters*. Um *cluster* pode ser definido por três atributos,  $C_i = (c_i, v_i, r_i)$ , o centro  $c_i$ , o volume  $v_i$  e o índice de ineficácia  $r_i$ .

O centro  $c_i$  é uma solução que representa o *cluster*  $C_i$ , identificando a sua localização dentro do espaço de busca.

O volume  $v_i$  representa a quantidade de soluções agrupadas no *cluster*  $C_i$ . Um *cluster* se torna promissor quando o volume atinge certo limitante  $\lambda$ , definido *a priori*.

O índice de ineficácia  $r_i$  é uma variável de controle para indicar o número de vezes consecutivas que a busca local foi aplicada no *cluster*  $C_i$  e não melhorou a solução. Este atributo evita que a busca local fique sendo executada por mais de  $r_{max}$  vezes em regiões ruins ou regiões que já tenham sido suficientemente exploradas.

Para agrupar soluções em *clusters* define-se alguma forma de medir a distância entre duas soluções. Sendo assim, uma função de medida de distância  $d(i, j)$  é definida para calcular a distância entre duas soluções.

O CS é um método iterativo que possui três componentes principais: uma meta-heurística, um processo de agrupamento e um método de busca local. A cada iteração do CS, uma solução  $s_k$  é gerada pela meta-heurística e enviada para o processo de agrupamento, então  $s_k$  é agrupada no *cluster* mais similar  $C_j$ , que é o *cluster* mais próximo à solução  $s_k$ . E, o centro deste *cluster* ( $c_j$ ) é atualizado com informações contidas na nova solução agrupada por meio do processo de assimilação, fazendo com que o centro se desloque no espaço de busca.

Em seguida é analisado o volume  $v_j$  do *cluster*. Caso  $v_j$  tenha atingido um limitante  $\lambda$  definido a priori, esse *cluster* pode estar em uma região de busca promissora. Porém, se o método de busca local não tiver obtido sucesso nas últimas  $r_{max}$  aplicações neste *cluster* promissor (índice de ineficácia  $r_j \geq r_{max}$ ) é aplicada uma perturbação aleatória no centro  $c_j$ , objetivando escapar desta região do espaço de busca.

Porém, se  $r_j$  for menor que  $r_{max}$ , uma busca local é aplicada no centro  $c_j$  intensificando a busca na vizinhança do *cluster*. A busca obtém sucesso em um *cluster* quando encontra uma solução que seja a melhor obtida neste *cluster* até o momento .

Depois do processo de agrupamento, retorna-se para a meta-heurística que gera outra solução. O critério de parada do CS é dado pelo critério de parada da meta-heurística utilizada na geração de soluções.

A estratégia híbrida do método CS para um problema de minimização é descrita pelo fluxograma ilustrado na Figura 4.1.

O método *Pareto Clustering Search* (PCS), proposto neste trabalho, é uma variação do CS para a solução heurística do problema sob a visão multiobjetivo. Neste trabalho, o PCS inicia sua busca por soluções através da meta-heurística multiobjetivo *Pareto Simulated Annealing* (PSA) descrito no capítulo anterior.

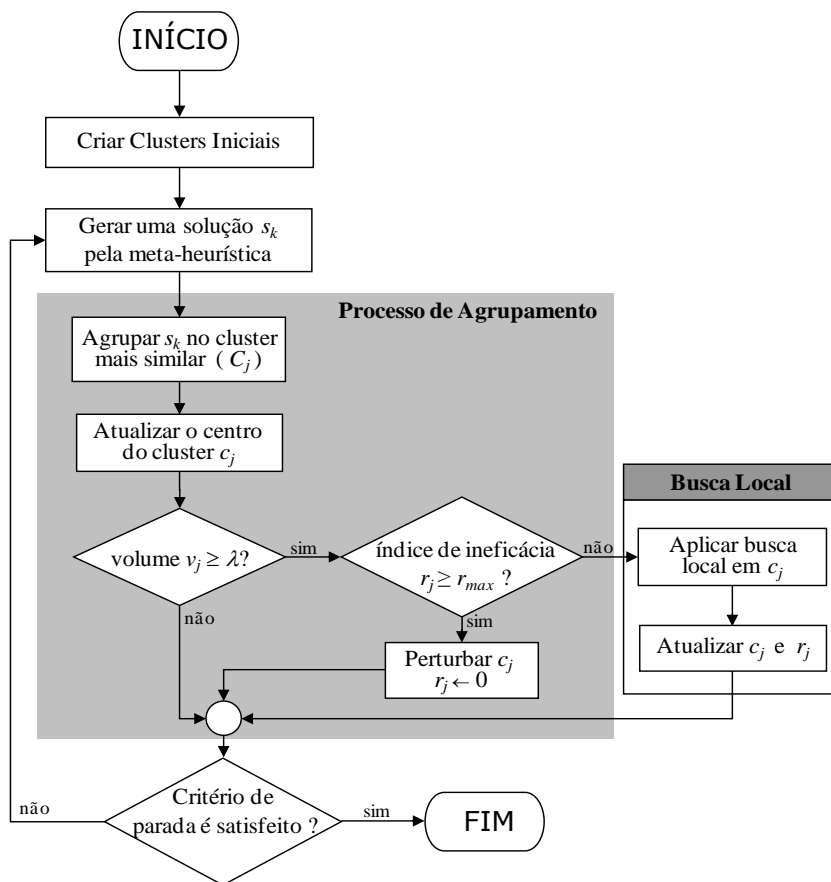


Figura 4.1 - Fluxograma do método CS.

Fonte: Chaves e Lorena [4]

O PCS possui o mesmo processo que o CS, porém utiliza a meta-heurística multiobjetivo PSA e os *clusters* representam a fronteira de Pareto. Além disso, os métodos de busca local são multiobjetivos.

O PCS cria os clusters iniciais como sendo um conjunto de Pareto, em seguida as soluções geradas pela meta-heurística PSA são agrupadas no *cluster* mais próximo, de acordo com uma medida de distância entre funções objetivo.

Assim que um *cluster* se torna promissor é aplicada uma heurística de busca local multiobjetivo, visando encontrar soluções que sejam não dominadas. O PCS é responsável por controlar a fronteira de Pareto, criando novos *clusters* com soluções não dominadas desde que não sejam similares às soluções já existentes e eliminando *clusters* com soluções que se tornem dominadas.

## 5 APLICAÇÃO DO PCS AO MVP

O *Pareto Clustering Search* (PCS) foi aplicado ao *Multiobjective Vending Problem* (MVP) criando-se uma estrutura para representar as soluções geradas, tal estrutura contém o valor da função denominada  $f_{o_1}$  cujo objetivo é minimizar o custo de deslocamento e da função  $f_{o_2}$  que visa aumentar o lucro. Além disso, essa estrutura contém um vetor contendo as cidades visitadas pelo caixeiro e outro vetor com as cidades que não foram visitadas.

Suponha uma rota que contenha as cidades 1,2,3,4 e 5, cada uma com seu prêmio específico. A Figura 5.1 ilustra a representação de uma possível solução para este problema.

Cidades visitadas :	1	3	4
Cidades não visitadas :	2	5	
$f_{o_1} = 1348$			
$f_{o_2} = 507$			

Figura 5.1 - Representação da solução.

Neste exemplo, o caixeiro visitou as cidades 1,3 e 4, a soma dos prêmios coletados, dado pela  $f_{o_2}$ , foi de 507, e o custo da viagem, dado pela  $f_{o_1}$ , foi 1348.

O PCS retorna uma série dessas soluções, cujas funções objetivos são não dominadas entre si. Esse conjunto de soluções representa a fronteira de Pareto.

O PSA foi utilizado como gerador de soluções no PCS, as estruturas de vizinhanças do PSC aplicado ao MVP foram empregadas no PSA e são definidas através dos seguintes movimentos:

- Inserir um vértice que não está sendo visitado;
- Retirar um vértice que está sendo visitado;
- Trocar 2 vértices que estão sendo visitados de posição.

O método de busca local utilizado para intensificar a busca do centro do *cluster* é chamado Descida em Vizinhança Variável (VND, do inglês *Variable Neighborhood Descent*) [14]. O VND é composto por três diferentes heurísticas de refinamento: *2-Opt*, *Add-step* e *Drop-step*.

A heurística *2-Opt* consiste na troca entre pares de aresta do grafo. Removem-se duas arestas, quebrando o circuito em dois caminhos, e os reconecta de outra maneira. O objetivo do *2-Opt* é reduzir a distância entre os vértices por meio da substituição de arestas de maior custo por outras de menor custo.

A heurística *Add-step* consiste em adicionar o vértice que possuir o melhor valor de economia de inserção, se o valor da economia for positivo então ambas funções objetivo irão melhorar após o movimento.

O movimento *Drop-step* consiste em retirar o vértice que possuir o melhor valor de economia de remoção, se o valor da economia for positivo então a função objetivo responsável por minimizar a distância irá melhorar após o movimento.

O principal aspecto a ser observado é que todos os movimentos são executados preservando a viabilidade da solução.

Para atualizar os centros dos *clusters*, utiliza-se o método Reconexão por Caminhos (PR, do inglês *Path-Relinking*). O PR realiza movimentos exploratórios na trajetória que interconecta duas soluções. Assim sendo, o processo de assimilações é responsável por intensificar e diversificar a busca dentro de um *cluster*, pois o centro será deslocado para a melhor solução avaliada nessa trajetória.

O PR inicializa a partir de duas soluções. A primeira é o centro do *cluster* mais similar ( $s_i$ ). A segunda é a solução gerada pela meta-heurística ( $s_g$ ). O método inicializa calculando a diferença simétrica entre as duas soluções, que é o conjunto de movimentos necessários para alcançar  $s_g$  a partir de  $s_i$ . Um caminho de soluções é gerado, conectando  $s_i$  e  $s_g$ . O método termina quando a

solução  $s_g$  for alcançada ou quando uma porcentagem do caminho for analisada. A melhor solução neste caminho é o novo centro do *cluster*.

## 6 RESULTADOS

O PCS para o MVP foi codificado em C++ e os experimentos foram conduzidos em um PC com processador Intel core i5, 2.5 GHz e memória de 6 GB de RAM sob plataforma *Windows 7*. Os experimentos foram realizados com objetivo de validar a abordagem proposta, mostrando que o PCS pode ser competitivo para resolução de problemas multiobjetivos. Para testar o modelo proposto foi utilizada a instâncias-teste *burma14\_100\_100*, *att48\_100\_100*, *berlin52\_100\_100* e *ulysses22\_100\_100* que podem ser encontradas em <http://www.sjc.unifesp.br/docente/chaves/problem-instances>. Visando verificar a eficiência do PCS, fez-se a comparação das soluções obtidas somente através da meta-heurística PSA e das soluções dadas pelo PCS. Os gráficos apresentado na Figura 6.1 mostra estas soluções. O eixo das abscissas representa o custo de deslocamento. O eixo das coordenadas representa a quantidade de prêmios coletados.

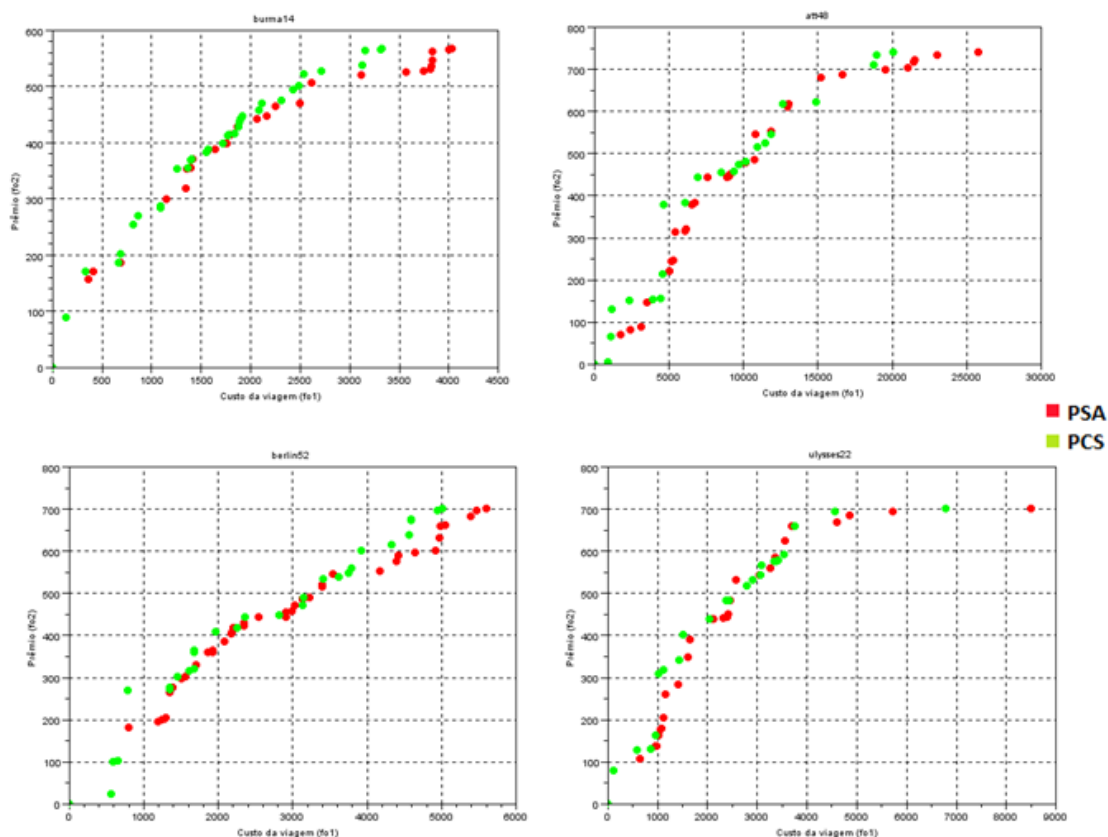


Figura 6.1 - Gráficos de comparação PSA e PCS.

A Tabela 6.1 apresenta os melhores e piores tempos computacionais obtidos em 10 execuções do PSA e PCS.

Tabela 6.1- Tempo de execução PSC e PSA

<i>Instâncias teste</i>	<b>PSA</b>		<b>PCS</b>	
	<i>Tempo mínimo</i>	<i>Tempo máximo</i>	<i>Tempo mínimo</i>	<i>Tempo máximo</i>
<b>burma14</b>	310,40 segundos	323,47 segundos	305,21 segundos	323,12 segundos
<b>att48</b>	308,02 segundos	318,53 segundos	302,42 segundos	319,401 segundos
<b>berlin52</b>	315,35 segundos	330,53 segundos	303,96 segundos	327,52 segundos
<b>ulysses22</b>	303,34 segundos	312,75 segundos	303,73 segundos	311,43 segundos

Nota-se que as soluções geradas pelo PCS possuem maiores valores para o lucro ( $fo_2$ , eixo das coordenadas) e menores valores para o custo da viagem ( $fo_1$ , eixo das abscissas) do que as soluções geradas pelo PSA e portanto o PCS gera soluções significativamente melhores que o somente PSA. Além disso, as soluções geradas pelos dois métodos possuem uma boa distribuição na fronteira de Pareto . Para a comparação entre os métodos ser justa, o tempo de execução de ambos algoritmos foi configurado de forma que executassem o algoritmo por pelo menos um determinado número de segundos.



## 7 CONCLUSÃO

Este trabalho propõe uma abordagem heurística, utilizando a meta-heurística *Pareto Clustering Search* (PCS), para a resolução do Problema do Vendedor com Multiobjetivos (MVP). Esta meta-heurística utiliza o conceito de algoritmos híbridos, combinando meta-heurísticas com um processo de agrupamento de soluções em subespaços de busca (*clusters*), visando detectar regiões promissoras. Sempre que uma região for considerada promissora é realizada uma intensificação da busca nesta região, objetivando uma aplicação mais racional do método de busca local.

Através dos testes realizados percebe-se que o algoritmo proposto é capaz de encontrar um conjunto de soluções eficientes para o problema em estudo.

Para trabalhos futuros pretende-se aplicar o PCS em problemas com maior quantidade de objetivos, e também utilizar outra meta-heurística clássica sob visão multiobjetivo para gerar soluções, tal como o algoritmo genético.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ARROYO, J.E.C. Heurísticas e Meta-heurísticas para Otimização Combinatória Multiobjetivo, p. 7-21, 2002.
- [2] AWERBUCH, B.; AZAR, Y.; Blum, A.; VEMPALA, S. (1998), New approximation guarantees for minimum weight k-trees and prize-collecting salesmen. *SIAM Journal on Computing*, v. 28, n. 1, p. 254–262.
- [3] BALAS, E. (1989), The prize collecting traveling salesman problem. *Networks*, v. 19, p. 621–636.
- [4] CHAVES, A. A., LORENA, L. A. N. Clustering search algorithm for the capacitated centred clustering problem, *Computers & Operations Research*, v. 37, 552-558, 2009.
- [5] CZYZAK P., JASZKIEWICZ A. Pareto Simulated Annealing – a metaheuristic technique for multiple objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, v. 7, 34-47, 1998.
- [6] DELL'AMICO, M.; MAFFIOLIaffioli, F.; VARBRAND, P. (1995), On prize collecting tours and the asymmetric travelingsalesman problem. *International Transactions in Operational Research*, v. 2, n. 3, p. 297–308.
- [7] FEILLET, D.; DEJAX, P.; GENDREAU, M. Traveling salesman problems with profits. *Transportation Science*, v. 2, n. 39, p. 188-205, 2005.
- [8] FONSECA C.M, FLEMING P.J. Multiobjective Genetic Algorithms made easy: Selection, Sharing and Matinig restrictions. In *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 95)*, London, UK, pp. 45-52. The Institution of Electrical Engineers, 1995.
- [9] GLOVER, F. Tabu search and adaptive memory programing: advances, applications and challenges. In: BARR, R. S.; HELGASON., R. V.; KENNINGTON, J. L. (Ed.). *Interfaces in Computer Science and Operations Research*. Norwell: Kluwer, p. 1-75, 1996.
- [10] HOLLAND, J. H. *Adaptation in natural and artificial systems*. Michigan: University of Michigan Press, 1975. 211 p.
- [11] JASKIEWICZ A. Genetic local search for multi-objective combinatorial optimization, *European Journal of Operational Research* vol.137, p. 50-71, 2002.
- [12] JONES D.F., MIRRAZAVI S.K. e TAMIZ M. Multi-objective meta-heuristics: An overview of the current state-of-art, *European Journal of Operational Research* vol.137, p. 1-19, 2002.
- [13] KELLER, C.P.; Goodchild, M., The multiobjective vending problem: A generalization of the traveling salesman problem. *Environment and Planning B: Planning and Design*, v. 15, p. 447–460, 1988.
- [14] MLADENOVIC, N.; HANSEN, P. (1997), Variable neighborhood search. *Computers and Operations Research*, v. 24, p. 1097–1100.

- [15] PEREIRA, G.W. Aplicação da Técnica de Recozimento Simulado em Problemas de Planejamento Florestal Multiobjetivo, 2004.
- [16] REINELT, G., *Traveling Salesman : Computational Solutions for TSP Applications*. Springer: Lecture Note in Computer Science, 1994.
- [17] TSILIGIRIDES, T. (1984), Heuristic methods applied to orienteering. *Journal of Operational Research Society*, v. 35, n. 9, p. 797–809.
- [18] ULUNGU E.L., TEGHEM J. e OST C. Efficiency of interactive multi-objective simulated annealing through a case study”, *Journal of the Operational Research Society*, vol.49, p. 1044-1050, 1998.
- [19] VAN VELDHUIZEN D.A., LAMONT G.B. Multiobjective evolutionary algorithms: Analysing the state-of art, *Evolutionary Computation*, vol. 8(2), p. 125-147, 2000.