Um Método Híbrido Multiobjetivo Aplicado ao Problema do Caixeiro Viajante com Lucro

Carla C. D. Fernandes[1], Kelly C. Poldi[1], Luiz A. N. Lorena[2], Antonio A. Chaves[1]
[1]Universidade Federal de São Paulo (UNIFESP) – Rua Talim, 330, São José dos Campos/SP
[2]Instituto Nacional de Pesquisa Espacial (INPE) – Av. dos Astronautas, 1758, São José dos Campos/SP

***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 *Multiobjetive 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.**

***Palavras-Chave* — Multiobjetivo, Caixeiro Viajante, Meta-heurísticas.**

I. INTRODUÇÃO

O Problema do Caixeiro Viajante referido na literatura como *Traveling Salesman Problem* (TSP) [1] 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 consequentemente, 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 [2].

Os TSPP podem ser vistos como problemas do caixeiro viajante biobjetivo onde um objetivo pressiona o caixeiro a viajar (ou seja, maximizar os prêmios coletados) e outro estimula o caixeiro a minimizar os custos de viagem (permitindo a ele não visitar alguns clientes). 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 *Multiobjetive Vending Problem*) [3].

c.fernandes11@unifesp.br,kelly.poldi@unifesp.br;
antonio.chaves@unifesp.br; lorena@lac.inpe.br
Este trabalho foi financiado pelo CNPq (Projeto no. 151387/2011-7)

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. Na seção 2 apresenta-se uma formulação matemática para o TSPP e faz-se uma revisão bibliográfica. Na seção 3, define-se métodos multiobjetivos. Na seção 4 descreve-se a meta-heurística CS e sua adequação ao PCS. Na seção 5 discute-se a aplicação do PCS ao MVP. Na seção 6 apresentam-se os resultados computacionais obtidos. Na seção 7 são apresentadas algumas conclusões deste trabalho.

II. PROBLEMA DO CAIXEIRO VIAJANTE COM LUCRO

O problema do Caixeiro Viajante com Lucro (TSPP, do inglês *Traveling Salesman Problem with Profits*) [2] de uma forma geral, pode ser representado em um grafo completo G = (V,A), onde V = {*0, 1*,...,*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 *Multiobjetive Vending Problem* (MVP) [3];

2. Ambos os objetivos são combinados numa função objetivo, visando encontrar uma rota que minimize os custos de deslocamento menos os prêmios coletados. Em [5] 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) [6].

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) [7] onde também é inserido o conceito de penalidades, ou *Quota TSP* (QTSP) [8].

É possível definir um conjunto de restrições básicas para as formulações matemáticas dos TSPPs. 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.

Todas formulações matemáticas TSPP compartilham as restrições a seguir :

$$\sum_{j \in V \setminus i} x_{ij} = y_i \qquad \forall i \in V \tag{1}$$

$$\sum_{i \in V \setminus j} x_{ij} = y_j \qquad \forall j \in V \tag{2}$$

$$y_0 = 1 \tag{3}$$

$$x_{ij} \in \{0,1\} \qquad \forall i, j \in V \tag{4}$$

$$y_i \in \{0,1\} \qquad \forall i \in V \tag{5}$$

As restrições (1) e (2) são chamadas restrições de atribuição e garantem que cada vértice seja visitado no máximo uma vez. A restrição (3) assegura que o depósito seja visitado, as restrições (4) e (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 trabalho trata-se do MVP que possui duas funções objetivos, descritas a seguir:

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{6}$$

$$\max \sum_{i \in V} p_i y_i \tag{7}$$

sujeito a (1-5) e restrições que eliminam as sub-rotas que não envolvem o depósito (vértice v_0).

A função objetivo (6) busca diminuir o custo de deslocamento do caixeiro, enquanto a (7) visa aumentar o lucro.

A abordagem utilizada para solucionar o MVP no trabalho [3] consiste em resolver sequencialmente versões do problema com um único objetivo e tal abordagem não procura encontrar um conjunto de soluções não-dominadas.

III. ALGORITMOS MULTIOBJETIVOS

Um algoritmo multiobjetivo consiste em minimizar 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 e uma melhora em um dos objetivos pode ser conseguida unicamente em detrimento de pelo menos um dos outros objetivos. Portanto tais soluções, denominadas soluções Pareto-ótimas, são não dominadas por outras soluções [9].

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 Fig. 1 ilustra uma possível relação de dominância entre elas.

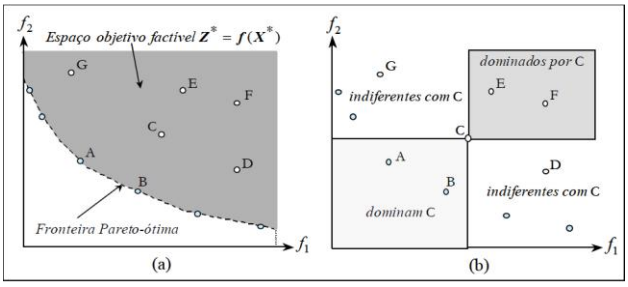

Fig. 1. Dominância de Pareto [9].

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.

IV. BUSCA POR AGRUPAMENTOS

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 método 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 λ, 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 uma forma de medir a distância $d(i, j)$ entre as soluções i e j. Neste trabalho a medida de distância utilizada foi a distância de Hamming [10].

O CS é um método iterativo que possui três componentes principais: uma meta-heurística, um processo de agrupamento e um método de busca local. O CS inicia seu processo criando *clusters* iniciais aleatórios, a cada iteração 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 λ 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 obtêm 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 Fig. 2.

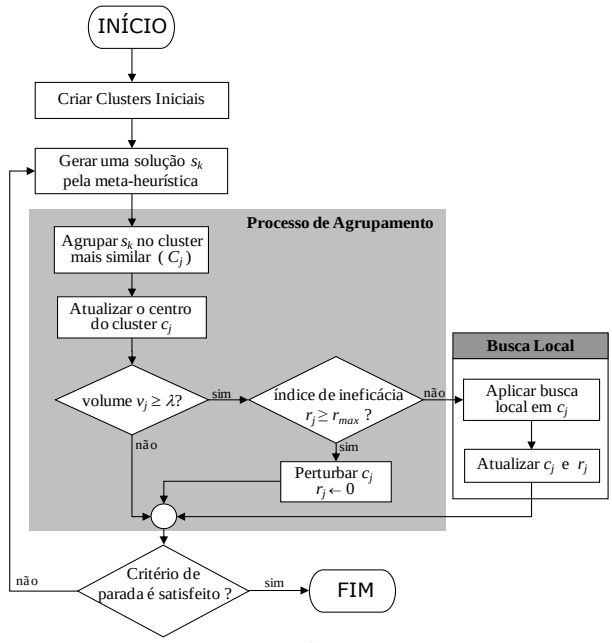

Fig. 2. Fluxograma do método CS [4].

O método *Pareto Clustering Search* (PCS), proposto neste trabalho, é uma variação do CS para a solução heurística do problema sob a visão multiobjetivo. O PCS inicia sua busca por soluções através da meta-heurística multiobjetivo *Pareto Simulated Annealing* (PSA) [11] 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:

$$P(x, y, T, \Delta) = \min \{1, \exp(-\sum_{j} \lambda_j (f_j(x) - f_j(y)/T))\} \quad (8)$$

Onde :

T é a temperatura;
x é a solução corrente;
y é a nova solução gerada a partir de x; e
$\Delta = (\lambda_{1,...,} \lambda_n)$ é o vetor de pesos.

Primeiramente T assume um valor elevado T_0 e após certo número de iterações a temperatura decai gradativamente por uma razão de resfriamento α, 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' obtido através da distância euclidiana. 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.

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, análogo ao CS.

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.

V. APLICAÇÃO DO PCS AO MVP

O *Pareto Clustering Search* (PCS) foi aplicado ao *Multiobjetive Vending Problem* (MVP) criando-se uma estrutura para representar as soluções geradas, tal estrutura contém o valor da função denominada fo_1 cujo objetivo é minimizar o custo de deslocamento e da função fo_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 as cidades 1,2,3,4 e 5, cada uma com seu premio especificado. A Fig. 3 ilustra a representação de uma possível solução para este problema.

Cidades visitadas :	1	3	4

Cidades não visitadas :	2	5

$fo_1 = 1348$

$fo_2 = 507$

Fig. 3. Representação da solução

Neste exemplo, o caixeiro visitou as cidades 1,3 e 4, a soma dos prêmios coletados, dado pela fo_2, foi de 507, e o custo da viagem, dado pela fo_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.

As estruturas de vizinhanças do PCS aplicado ao MVP 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 são 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*) [12]. O VND é composto por três diferentes heurísticas de refinamento, que combinam dois movimentos: *Add-step* e *Drop-step*. O movimento *Add-step* consiste em adicionar o vértice que possuir o melhor valor de economia de inserção. O movimento *Drop-step* consiste em retirar o vértice que possuir o melhor valor de economia de remoção. Em ambos movimentos, se o valor da economia for positivo então ambas funções objetivo irão melhorar após o movimento. O principal aspecto a ser observado é que todos os movimentos são executados preservando a viabilidade da solução.

Para atualização do centro do *cluster* é realizada a assimilação por caminhos, utilizando o método Reconexão por Caminhos (PR, do inglês *Path-Relinking*) [13]. O PR realiza movimentos exploratórios na trajetória que interconecta duas soluções. Assim sendo, o processo de assimilações 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 (solução inicial, s_i). A segunda é a solução gerada pela meta-heurística (solução guia, s_g). O método inicia-se calculando a diferença simétrica entre as duas soluções, que é o conjunto de movimentos necessários para alcançar s_g a partir de s_i. Posteriormente é construído um caminho pelo espaço de soluções (vizinhança) conectando s_i e s_g, gerando assim novas soluções (intermediárias) a cada movimento executado. Este movimento é realizado inserindo características ou atributos da solução guia na solução intermediária. O método finda-se quando a solução s_g for alcançada ou quando 30% do caminho for analisado. A melhor solução neste caminho é o novo centro do *cluster*.

VI. 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.

O vetor de pesos associado a cada solução foi composto por números racionais entre 0 e 1 gerados aleatoriamente. Os valores atribuídos para os parâmetros temperatura inicial (T_0), razão de resfriamento (α), limite para o volume do *cluster* (λ) e índice máximo de ineficácia (r_{max}) foram os seguintes :

- $T_0 = 100000$,
- $\alpha = 0.95$,
- $\lambda = 15$,
- $r_{max} = 6$.

Para testar o modelo proposto foram utilizadas as instâncias-teste burma14_100_100, att48_100_100 e berlin52_100_100 que podem ser encontradas em [14].

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. O gráficos apresentados nas Fig. 4, 5 e 6 mostram estas soluções. O eixo das abscissas representa o custo de deslocamento. O eixo das coordenadas representa a quantidade de prêmios coletados.

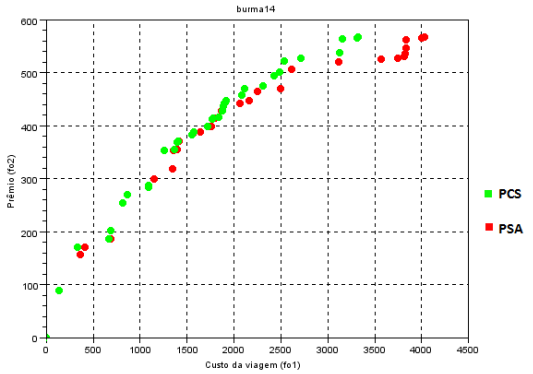

Fig. 4. Gráfico comparação PSC e PSA - burma14

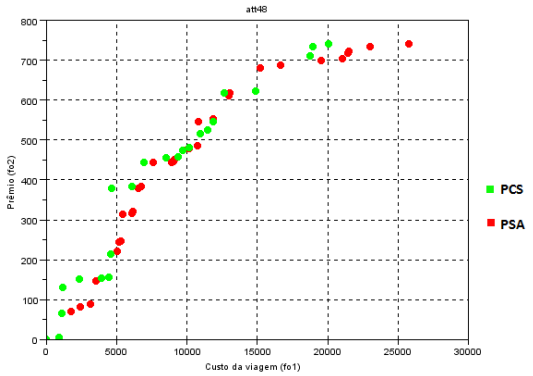

Fig. 5. Gráfico comparação PSC e PSA - att48

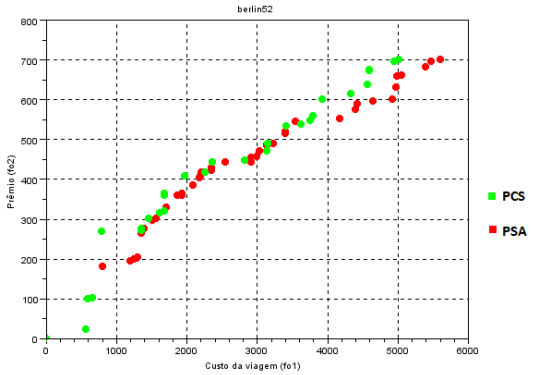

Fig. 6. Gráfico comparação PSC e PSA - berlin52

A Tabela I apresenta os melhores e piores tempos computacionais obtidos em 10 execuções do PSA e PCS. Os tempos foram medidos em segundos.

TABELA I TEMPO COMPUTACIONAL

	PSA		**PCS**	
Instâncias teste	*Tempo mínimo*	*Tempo máximo*	*Tempo mínimo*	*Tempo máximo*
burma14	310,40	323,47	305,21	323,12
att48	308,02	318,53	302,42	319,401
berlin52	315,35	330,53	303,96	327,52

Nota-se que as soluções geradas pelo PCS possuem maiores valores para o lucro (fo_2) e menores valores para o custo da viagem (fo_1) do que as soluções geradas pelo PSA e portanto o PCS gera soluções significativamente melhores que o somente PSA. Enquanto o tempo médio gasto pelos métodos PCS e PSA foram parecidos.

VII. 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

[1] G. Reinelt, "*Traveling Salesman : Computational Solutions for TSP Applications*. Springer: Lecture Note in Computer Science", 1994.

[2] D. Feillet, P. Dejax, M. Gendreau, "Traveling salesman problems with profits. *Transportation Science*", v. 2, n. 39, p. 188-205, 2005.

[3] C. P. Keller, M. Goodchild, "The multiobjective vending problem: A generalization of the traveling salesman problem". *Environment and Planning B: Planning and Design*, v. 15, p. 447–460, 1988.

[4] A. A. Chaves, L. A. N. Lorena, "Clustering search algorithm for the capacitated centred clustering problem", *Computers & Operations Research*, v. 37, 552-558, 2009.

[5] M. Dell'Amico, F. Maffioli, P. Varbrand, "On prize collecting tours and the asymetric traveling salesman problem". *International Transactions in Operational Research*, v. 2, n. 3, p. 297–308, 1995.

[6] T. Tsiligirides, " Heuristic methods applied to orienteering". *Journal of Operational Research Society*, v. 35, n. 9, p. 797–809, 1984.

[7] E. Balas, "The prize collecting traveling salesman problem". *Networks*, v. 19, p. 621–636, 1989.

[8] B. Awerbuch, Y. Azar, A. Blum, S. Vempala, " New approximation guarantees for minimum weight k-trees and prize-collecting salesmen. *SIAM Journal on Computing"*, v. 28, n. 1, p. 254–262, 1998.

[9] J.E.C. Arroyo, "Heurísticas e meta-heurísticas para otimização combinatória multiobjetivo", p. 7-21, 2002.

[10] R.W. Hamming Technical Journal, "Error detecting and error correcting codes". Bell System, v. 26, n. 2, p. 147-160, 1950.

[11] P. Czyzac, A. Jaszkiewicz, "Pareto Simulated Annealing – a metaheuristic technique for multiple objective combinatorial optimization". *Journal of Multi-Criteria Decision Analysis*, v. 7, 34-47, 1998.

[12] N. Mladenovic, P. Hansen , "Variable neighborhood search". *Computers and Operations Research*, v. 24, p. 1097–1100, 1997.

[13] F. Glover, "Tabu search and adaptive memory programing: advances, applications and challenges". In: BARR, R. S.; R. V. Helgason, J. L. Kennington. "*Interfaces in Computer Science and Operations Research"*. Norwell: Kluwer, p. 1-75, 1996.

[14] A. A. Chaves, "TSPP Problem Instances". Disponível em: <http://www.sjc.unifesp.br/docente/chaves/problem-instances>. Acesso em 24/08/2012.