

USO DE ALGORITMO CULTURAL COM UMA NOVA ABORDAGEM MEMÉTICA POR MEIO DO SIMULATED ANNEALING PARA O PROBLEMA DO CAIXEIRO VIAJANTE

DEAM J. A. SILVA, JOAQUIM A. L. SILVA, CAROLINA M. AFFONSO, ROBERTO C. L. OLIVEIRA

Laboratório de Computação Bio-Inspirada (LCBio), Faculdade de Enga. da Computação e de Telecomunicações (FCT), Universidade Federal do Pará (UFPA)

Av. Augusto Corrêa, No. 01, Instituto de Tecnologia, UFPA

E-mails: djames2871@gmail.com, jalsjr@hotmail.com, carolina@ufpa.br, limao@ufpa.br

Abstract— In many optimization problems there are difficulties in achieving a good result or even a result close to the optimum value in a feasible time, especially when working on large scale. These problems are normally addressed by heuristics or meta-heuristics which perform searches for better solutions on previously defined search space. There are several meta-heuristics in Natural Computing, among them there are Evolutionary Computation, with the focus on Cultural Algorithms (CA) and Genetic Algorithms (GAs), which are used as a complement due to the dual mechanism of culture and genetic inheritance. The purpose of this paper is to use such mechanisms and incorporate local search heuristics and simulated annealing applied to a combinatorial optimization problem called Travel Salesman Problem (TSP). Some experiments were performed to make an assessment regarding the performance of these hybrid mechanisms with other algorithms available in the literature.

Keywords— Genetic Algorithms, Cultural Algorithm, Memetic Computation, Simulated Annealing, Local Search, Travel Salesman Problem

Resumo— Em muitos problemas de otimização há dificuldades em alcançar um resultado ótimo ou mesmo um resultado próximo ao valor ótimo em um tempo viável, principalmente quando se trabalha em grande escala. Por isso muitos desses problemas são abordados por heurísticas ou meta-heurísticas que executam buscas por melhores soluções dentro do espaço de busca definido. Na Computação Natural existe várias meta-heurísticas, dentre elas existem as Evolutivas, onde se destacam os Algoritmos Culturais (ACs) e os Algoritmos Genéticos (AGs), que são utilizados de forma complementar devido ao mecanismo dual de herança cultural/genética. A proposta do presente trabalho é utilizar tais mecanismos acrescentando heurísticas de busca local e o uso do *simulated annealing* para pesquisar a vizinhança, aplicado em um problema de otimização combinatória (Problema do Caixeiro Viajante). São executados alguns experimentos para efetuar uma avaliação em relação ao desempenho desses mecanismos híbridos com outros algoritmos disponíveis na literatura.

Palavras-chave— Algoritmos Genéticos, Algoritmos Culturais, Computação Memética, *Simulated Annealing*, Busca Local, Problema do Caixeiro Viajante.

1 Introdução

Os métodos de otimização modernos, também, por vezes, chamados métodos de otimização não tradicionais, surgiram como métodos poderosos e populares para resolver problemas complexos de engenharia nas últimas décadas (Goldberg, 1989; van Laarhoven e Aarts, 1987; Dorigo *et al.*, 1996; Kennedy e Eberhart, 2001; Osborne, 2010; Lin, 2013). Este surgimento tem relação com a estratégia de solução baseada em heurísticas. A heurística é uma estratégia que visa produzir soluções aceitáveis para um problema complexo em um tempo razoavelmente prático. A complexidade do problema de interesse torna impossível a busca a cada possível solução, o objetivo é encontrar boas soluções viáveis em uma escala de tempo aceitável mesmo que não exista garantia de que as melhores soluções podem ser encontradas (Rao, 2009).

Usualmente uma heurística é identificada com um problema específico (Weise, 2009). O grau de dificuldade em resolver um problema determinado,

com um algoritmo dedicado, está estreitamente relacionado com a sua complexidade computacional, ou seja, a quantidade de recursos como tempo e memória necessária para fazê-lo. Uma das classes de complexidade mais difíceis é a NP, o conjunto de todos os problemas de decisão que são resolvidos em tempo polinomial por máquinas de Turing não-determinísticas. Uma abordagem para a obtenção de soluções quase ótimas para os problemas NP em tempo razoável é a aplicação de meta-heurística, guiadas por funções objetivo. Sendo normalmente, utilizadas funções objetivo que são consideradas difíceis, do ponto de vista matemático, por não serem contínuas, não diferenciáveis ou que tenham múltiplos valores máximos e mínimos. Mais ainda, em muitas aplicações no mundo real de otimização meta-heurística, as características das funções objetivo não são conhecidas antecipadamente.

Dentre as dificuldades encontradas no uso de meta-heurísticas, a convergência prematura é o que mais se destaca. Um dos problemas na otimização global, é que muitas vezes não é possível determinar se a melhor solução atualmente conhecida está situada em um ótimo local ou global. Na convergência

prematura, um processo de otimização converge para um mínimo local, e não é capaz de explorar outras partes do espaço de busca, mesmo que exista outra região que contenha uma solução superior. E isto é muito comum em otimização combinatória.

Um dos problemas de otimização combinatória mais relevante é o do caixeiro viajante (Travel Salesman Problem - TSP). Este modelo de problema serve para representar várias situações reais (Davenport, 2010), e também servem como padrões de testes de novos algoritmos de otimização (TSPLIB, 2012). Uma maneira eficaz de resolver o problema do TSP em grande escala (grande número de cidades) é a aplicação de algoritmos de otimização bio-inspirados tais como: os AGs e os algoritmos baseados na Colônia de Formigas (Ant Colony Optimization - ACO) (Michalewicz, 1996; Wang e Cao, 2002; Duan *et al.*, 2006; Karaboga e Gorkemli, 2011, Zhu *et al.*, 2013). O uso do AG, normalmente, leva a um ótimo local e causa a estagnação do processo de otimização. O uso destes métodos de otimização, com informação heurística, no TSP permite uma melhora no desempenho do algoritmo (Jiao e Wang, 2000; Purisa *et al.*, 2010; Delin *et al.*, 2011; Badillo *et al.*, 2013).

No início dos estudos sobre Computação Memética (CM) a mesma era sinônimo do AG com busca local para resolver o TSP. Mais tarde, ficou evidente que a área de CM era mais ampla e envolvia, de uma maneira geral, o uso de métodos gerais de otimização (meta-heurísticas) com informações heurísticas de busca local ou de vizinhança, que grosseiramente pode se denominar de métodos híbridos (Neri e Cotta, 2012). Dentre as técnicas evolutivas híbridas, inicialmente tinha-se o uso do AG com busca local (Goldberg e Voessner, 1999), depois passou-se a utilizar o AG combinado com o *simulated annealing* para tratar a vizinhança (Wang *et al.*, 2005). Mais recentemente o AG vem sendo utilizado em combinação com o *simulated annealing*, com o acréscimo de uma heurística de busca local (Xu *et al.*, 2011). Em outra direção, o AG vem sendo utilizado junto com o Algoritmo Cultural (AC) (Xue e Guo, 2007, Yan *et al.*, 2012), que gerou uma das estratégias híbridas mais avançada desta classe de algoritmos, o AC junto com AG e busca local (Digalakis e Margaritis, 2002).

Especificamente, este trabalho tem a proposta de utilizar um Algoritmo Cultural, com população criada por um AG. Além disto, é utilizado como estratégia de busca de vizinhança o *simulated annealing*, para em seguida utilizar em sequência os algoritmos de busca local 2-opt e 3-opt. Esta estratégia tem o objetivo de superar as deficiências de aprisionamento em mínimos locais e/ou convergência prematura, de forma a garantir uma maior diversidade no conjunto de possíveis soluções, para o problema combinatório do caixeiro viajante.

Mais ainda, aqui o Algoritmo Cultural utiliza apenas duas fontes de conhecimento, o situacional e o normativo, diferente do trabalho de Kim e Cho (Kim

e Cho, 2009) que utilizam o Algoritmo Cultural com as quatro possíveis fontes de conhecimento (situacional, normativo, topográfico e temporal) e a busca local com o 2-opt ou com o 3-opt para o TSP. Na literatura existe um trabalho que utiliza o Algoritmo Genético como busca global, e o *simulated annealing* como busca local para resolver o problema do caixeiro viajante (Delin *et al.*, 2011). Também tem-se um outro trabalho que utiliza um Algoritmo Memético (AM) com busca local pelo 2-opt (Badillo *et al.*, 2013). Desta maneira é possível observar que este trabalho tem a vantagem do uso do AC+AG, com o aumento de desempenho devido ao uso do *simulated annealing* na busca da vizinhança, acrescido das melhorias oriundas da busca local, mas com o uso sequencial das conhecidas heurísticas 2-opt e 3-opt.

2 Algoritmo Cultural

A computação evolutiva utiliza conceitos metafóricos, princípios e mecanismos de extração da compreensão de como sistemas naturais são capazes de ajudar a resolver problemas computacionais considerados complexos. Atualmente, muitos trabalhos são focados no processo de seleção natural e genética como os AGs (Back *et al.*, 1997) que são uma poderosa ferramenta para mapear a evolução biológica. No entanto, os AGs não são suficientes para modelar a evolução cultural que é um traço fundamental das sociedades humanas. Embora existam pontos de coincidência, a evolução cultural tem suas próprias características, diferentes da evolução genética. Entre eles o fato de que a evolução cultural se torna mais rápida do que a evolução genética e da existência de um espaço de crença compartilhado que permite que os indivíduos incorporem diretamente o conhecimento sem ter de reaprender tudo a partir do início.

Este tipo de meta-heurística é denominado de Algoritmo Cultural (AC). Os ACs foram propostos por Robert Reynolds (Reynolds, 1994) como um complemento à metáfora evolutiva utilizada na Computação Evolutiva. Dessa forma, os ACs são algoritmos Evolutivos baseados tanto na metáfora genética quanto no processo de evolução cultural da humanidade.

O princípio do algoritmo cultural pode ser abordado através da definição de cultura. Segundo Reynolds e Zhu (2001), o termo cultura aqui pode ser entendido como um veículo de armazenamento de informações acessíveis globalmente por todos os membros da sociedade e que pode ser útil em direcionar suas atividades para solução de problemas. Nestes modelos fica claro que o conhecimento é a parte do sistema que é compartilhado pela população e compilado por membros da sociedade, com uma codificação que permita o acesso a todos. Ao mesmo tempo distingue a parte do sistema que é individual, que significa também incluir as experiências individuais, e como essas experiências podem trazer algo para o conhecimento compartilhado (Becerra, 2002).

Reynolds *et al.* (2010) desenvolveram um sistema duplo de herança nos AC's focando também o processo de herança em dois níveis (microevolutivo e macroevolutivo). No nível microevolutivo tem-se a modelagem da população em si composta por um conjunto de indivíduos, levando-se em consideração qualquer modelo evolutivo baseado em populações (como por exemplo, o Algoritmo Genético). Já no nível macroevolutivo é modelado o conhecimento adquirido pelos indivíduos ao longo das gerações e que codificado e armazenado no espaço de crenças, ajuda a guiar o comportamento dos indivíduos em suas populações.

2.1 Funcionamento Básico de um Algoritmo Cultural

Os Algoritmos Culturais (AC's) possuem um funcionamento básico proposto por Reynolds (Reynolds, 1994; Reynolds *et al.*, 2010) onde são descritos dois componentes principais: Espaço Populacional e Espaço de Crenças. A idéia básica dos Algoritmos Culturais é apresentada pela Figura 1.

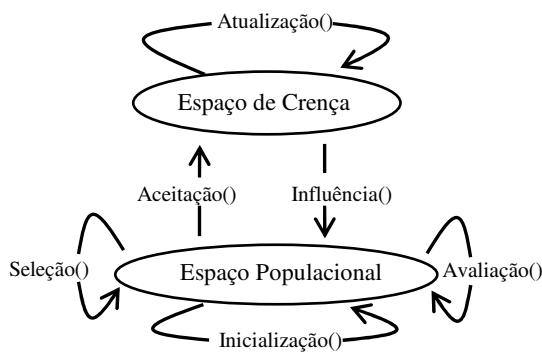


Figura 1. Funcionamento Básico de um Algoritmo Cultural (Reynolds *et al.*, 2010).

O Espaço Populacional (P) é o conjunto de soluções que pode ser modelado utilizando qualquer técnica de Inteligência Computacional que faça uso de uma população de indivíduos, já o Espaço de Crença ou Mapa do grupo (EP) é o local onde ocorre o armazenamento e representação do conhecimento (experiência ou mapas individuais) adquirido ao longo do processo evolutivo. No espaço de crença podem-se armazenar diferentes tipos de conhecimento como: conhecimento situacional, normativo, do domínio, topográfico e histórico. É importante ressaltar que não é obrigatória a implementação de todos esses tipos de conhecimento no espaço de crença, entretanto as mais utilizadas são o conhecimento situacional e normativo. É a partir desse conhecimento armazenado que os indivíduos são guiados na direção das melhores regiões do espaço de busca.

Os indivíduos são descritos por um conjunto de características e comportamentos e por um mapa que generaliza os conhecimentos e experiências adquiridas por esse indivíduo. Essas características e comportamentos são modificados por operadores genéticos que podem ser influenciados socialmente. Da

mesma maneira, os conhecimentos e experiências são unidos e modificados para formar o espaço de crenças.

A cada nova geração de indivíduos é feita uma avaliação e o conhecimento dos melhores indivíduos pode ou não fazer parte do Espaço de Crença. A nova população a ser gerada é influenciada com o conhecimento anteriormente armazenado através de operadores. Tanto o espaço de crença quanto os indivíduos de uma população podem ser influenciados pelo que se chama de Protocolos de Intercomunicação. A comunicação entre os indivíduos de uma geração e o Espaço de Crenças é dada por um protocolo dito Função de Aceitação. Já a interação do Espaço de Crenças com a população de indivíduos é chamada de Função de Influência.

3 Hibridização de Algoritmos Evolutivos com Busca Local (Algoritmos Meméticos)

A otimização é um tipo de problema comumente encontrado em vários campos do conhecimento onde se tenta encontrar a solução ideal entre um conjunto de soluções viáveis. Muitos problemas do mundo real e teórico podem ser formulados como problemas de otimização e resolvidos através da aplicação ou o desenvolvimento de várias técnicas de otimização. No mundo real existem problemas de otimização que são do tipo caixa-preta, no qual um conhecimento do problema não está disponível previamente e o uso de meta-heurísticas passa a prevalecer. Algoritmos Evolutivos, otimizações por enxame de partículas, algoritmos de colônia de formigas, e afins, são alguns dos renomados tipos de meta-heurísticas que têm sido amplamente adotados (Lin e Chen, 2011).

As técnicas que utilizam um modo híbrido entre meta-heurísticas e heurísticas de problemas específicos têm sido criadas para fornecer técnicas de otimização mais eficientes para os problemas mais complexos. Técnicas que empregam meta-heurísticas como busca global e heurísticas específicas do problema como a busca local, são comumente referidos como Algoritmos Meméticos (AMs). Com o desenvolvimento de um projeto adequado, os AMs não só podem apresentar uma boa capacidade exploratória, similar ao que faz um algoritmo com base populacional de busca global, mas também proporciona um bom desempenho de exploração durante a busca, semelhante ao que faz um algoritmo de busca local. Como resultado, algoritmos meméticos tem um desempenho melhor do que algoritmos sem hibridização (Yew-Soon *et al.*, 2010).

Com o aumento do interesse em pesquisar algoritmos meméticos, a computação memética tem evoluído a partir de hibridação de busca global com busca local (Meuth *et al.*, 2009). A capacidade da busca global da parte evolutiva de um algoritmo memético cuida da exploração, tentando identificar as regiões mais promissoras do espaço de busca, a parte de busca local examina os arredores de uma solução

inicial, explorando-o desta forma. Esta ideia não é só atraente como também é muito bem sucedida. De fato, para a maioria dos problemas de otimização combinatória esta combinação produz bons resultados. O papel da busca local é fundamental e a seleção de suas regras de busca juntamente com sua harmonização com o esquema da busca global produz o sucesso dos algoritmos meméticos (Neri *et al.*, 2011).

Na implementação de um algoritmo memético (AM) não é exigido apenas um mecanismo de busca global juntamente com os operadores de pesquisa locais, mas também se deve estabelecer uma coordenação sutil para expor o ponto de vista de ambos. Ao utilizar a heurística de busca local, é assumido que a busca evolutiva fornece uma exploração ampla do espaço de busca, enquanto a busca local pode desenvolver um tipo de exploração com base em soluções promissoras. Estes métodos permitem a pesquisa local refinar uma solução de acordo com a vizinhança que esta possui. Em outras palavras, dada uma solução particular s , um método de busca local explora a vizinhança de s , a fim de encontrar uma solução s' melhor do que uma solução s . Então, deve-se definir de que forma s' substitui s .

3.1 Implementação de Algoritmos Culturais por meio da Abordagem Memética com Busca Local

O objetivo principal desta abordagem é auxiliar os ACs a superarem o desafio de escapar de ótimos locais em problemas de otimização. A abordagem memética aqui descrita e implementada leva em consideração o problema do Caixeiro Viajante (TSP). Neste trabalho o espaço populacional é representado pela estrutura de um algoritmo genético e o espaço de crença utiliza dois tipos de conhecimento, que são os mais utilizados: (a) conhecimento situacional, que fornece o ponto exato onde o melhor indivíduo de cada geração foi encontrado, e (b) conhecimento normativo, que armazena os intervalos para as variáveis de decisão do problema, nas regiões onde foram encontrados os bons resultados. A busca local escolhida esta baseada no *simulated annealing*, acrescentado do uso dos algoritmos 2-opt e 3-opt. O uso do *simulated annealing* foi motivado pela sua simplicidade de implementação, pela capacidade de aceitar parâmetros que permitam uma configuração ideal em relação à energia inicial (T) e ao decremento dessa energia e , o mais importante, a capacidade de escapar de ótimos locais devido a aceitação de configurações pouco favoráveis em relação à aptidão da função objetivo (Weise, 2009; Rao, 2009; Tian *et al.*, 1999).

3.1.1 Estrutura de Vizinhança para Problemas Discretos: Problema do Caixeiro Viajante (TSP)

A criação de estruturas de vizinhança ou mecanismos de geração de soluções de vizinhança, é um elemento crítico na concepção do *simulated annealing* eficientes para problemas de otimização discreta (Henderson *et al.*, 2003). Tian *et al.* (1999) investi-

gam a aplicação do *simulated annealing* para problemas de otimização discreta com uma propriedade de permutação. Eles introduzem seis tipos de esquemas de perturbação para a geração de soluções aleatórias de permutação e provam que cada sistema satisfaz os requisitos de convergência. Os resultados das avaliações experimentais sobre o TSP, o problema de programação de fluxo de loja, e o problema de atribuição quadrática sugerem que os ganhos de eficiência dos esquemas de perturbação são diferentes para cada tipo de problema e espaço de solução. Eles concluem que com o esquema de perturbação adequada, o *simulated annealing* produz soluções eficientes para problemas diferentes de otimização discreta que possuem uma propriedade de permutação. O algoritmo de busca local denominado 2-opt, junto com a sua versão generalizada, denominada de 3-opt, são as técnicas de buscas locais mais utilizadas para o TSP (Johnson e McGeoch, 1997).

O AC, com população gerada pelo AG, com busca de vizinhança por meio *simulated annealing* (Quadro 1), aplicado ao TSP, proposto nesse trabalho utiliza a estrutura de busca local do tipo 2-opt (Quadro 1, linha 7) do 3-opt (Quadro 1, linha 8) e uma combinação do 3-opt com 2-opt (Quadro 1, linha 9). Observa-se que o algoritmo retorna a melhor vizinhança encontrada, através da função de avaliação (Quadro 1, linha 19, 20 e 21). Além disso, optou-se pela utilização do cruzamento por mapeamento parcial *partially mapped crossover* (PMX) e da recombinação de arestas *edge recombination* (ERX) (Linden, 2006).

Quadro 1: Algoritmo para Geração de Vizinhança.

```

Algoritmo GerarVizinhoTSP(S)
01: INICIO
02: D ← -1; MinD [ ]; MaxD [ ]; Pb [ ] = MaxD [ ]; incD = - MinD [ ];
03: Inicializa vetor perturbacao2-OPT (P2opt [ ])
04: Inicializa vetor perturbacao3-OPT (P3opt [ ])
05: Indivíduo: Inteiro [K] [ tamCromossomo ]; /*K=3 */
06: PARA i = 1 to K FAÇA
07:   Case i =1 Indivíduo [i] ← 2-OPT(S);
08:   Case i =2 Indivíduo [i] ← 3-OPT(S);
09:   Case i =3 Indivíduo [i] ← 3-OPT (2-OPT(S));
10: FIMPARA i
11: Prob ← random(0,1);
12: SE (Prob ≤ 0.5) ENTÃO
13:   INICIO
14:     N ← random(1,K)
15:     RETORNE Indivíduo (N);
16:   FIM
17: ELSE
18:   INÍCIO
19:     avaliar_individuos(Indivíduo [ ])
20:     melhorIndivíduo = selecionaMelhor(Indivíduo [ ])
21:     RETORNE melhorIndivíduo;
22: FIM
23: FIM

```

A mutação é realizada através de operadores de inversão do circuito em um determinado trecho, onde se inverte o cromossomo de acordo com valores de início e fim que podem ser escolhidos aleatoriamente. Outra operação é de troca de posições, onde duas posições são trocadas aleatoriamente. Os espaços de crenças utilizados nessa proposta são o espaço de crença situacional e o normativo baseado no trabalho de Carvalho e Barriveira (2009). O espaço situacional guarda os melhores indivíduos encontrados com uma percentagem p sobre a população (geralmente 20%). Para o problema abordado cada cromossomo representa um circuito, constituído por valores inteiros i que indicam a ordem do percurso. Para cada indivíduo i da população que será aplicada a mutação é feito o seguinte processo: primeiro busca-se uma posição no vetor do circuito (cidades), que tenha a mesma cidade do melhor cromossomo. Logo em seguida é realizado um sorteio para identificar o número de trocas que irá ocorrer (máximo de 50% de trocas). A mutação é feita, aplicando-se a cidade vizinha do melhor cromossomo.

O algoritmo híbrido proposto (Algoritmo Cultural baseado em Busca Local e Vizinhança ou Cultural Algorithms with Local Search - CA-LSN) oferece ao ACs meios para melhorar a velocidade de convergência e encontrar soluções ótimas. Neste algoritmo são realizadas quatro chamadas ao protocolo de comunicação, ou seja, duas a mais do que o AC tradicional. Lembrando que $P(t)$ é a População em uma geração t e $EP(t)$ é o Espaço de Crença em uma geração t . As quatro chamadas são: Comunicação ($P(t)$, $EP(t)$), Comunicação ($EP(t)$, $P(t)$) e duas chamadas Comunicação (Melhor_Encontrado(t), $EP(t)$). O framework do algoritmo cultural com busca local e vizinhança proposto é mostrado na Figura 2.

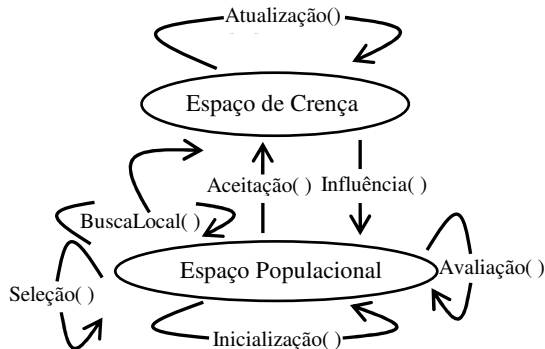


Figura 2. Framework do AC com busca local e vizinhança proposto.

Toda a operação do CA-LSN leva em consideração a busca local em dois pontos seguindo a estrutura do AC de acordo com o Quadro 2. A busca local é realizada de acordo com o valor de diferença mínima entre a geração atual e a geração marcada como a última melhoria ocorrida na função de otimização. Ou seja, se no processo evolutivo não há melhorias de otimização isso significa que a diferença entre a geração atual e a última geração onde ocorreu um

melhor resultado (valor ótimo) vai aumentando com o tempo. Dependendo do valor configurado para essa diferença em z , o algoritmo executa a busca local (Quadro2, linha 22). As diferenças entre o algoritmo cultural tradicional e o algoritmo aqui proposto podem ser visualizadas nas linhas 12, 13, 14, 15 e 16 do Quadro 2. O melhor indivíduo é selecionado da população ($P(t)$) (linha 12). Esse cromossomo é trabalhado pelo algoritmo de busca local que retornará o melhor resultado encontrado (linha 13). Em seguida é feita a comunicação com o melhor cromossomo encontrado com o espaço de crença ($EP(t)$) para comparar com os elementos presentes (linha 14). Em seguida é feita a atualização do Espaço de Crença ($EP(t)$). As linhas 22 a 27 fazem o mesmo processo de busca local relatado nas linhas anteriores. A única diferença é que ao invés de escolher o melhor indivíduo da população (linha 12) escolhe-se um indivíduo aleatório da população (linha 24).

Quadro 2: Algoritmo Cultural com Busca Local (CA-LSN).

```

Algoritmo Cultural com Busca Local e Vizinhança
01: INÍCIO
02: variáveis:  $t=0$  (geração atual);
                $tu$  (geração do último valor ótimo),
03:  $z$  (diferença mínima para chamar a BuscaLocal-SA)
04: inicializar população  $P(t)$  ;população inicial aleatória
05: Inicializar Espaço de Crença  $EP(t)$ 
06: avaliar população  $P(t)$  ;calcula  $f(i)$  para cada indivíduo
07: ENQUANTO (não condição_fim) FAÇA
08: diferença= $t - tu$ ;
09: Comunicação ( $P(t)$ ,  $EP(t)$ ); ;votação(Aceitação)
10: Atualização  $EP(t)$ ; ;uso de operadores culturais
11: SE (diferença $\geq z$ ) ENTÃO
12: Selecionar Melhor_Individuo da população  $P(t)$ ;
13: Melhor_Encontrado  $\leftarrow$  BuscaLocal-
SA(Melhor_Individuo);
14: Comunicação (Melhor_Encontrado( $t$ ),  $EP(t)$ );
15: Atualização  $EP(t)$ ; ;uso de operadores culturais
16: FIMSE
17: Comunicação ( $EP(t)$ ,  $P(t)$ ); ;promoção (função de influência)
18:  $t \leftarrow t+1$  ;próxima geração
19: selecionar  $P(t)$  de  $P(t-1)$ 
20: altera  $P(t)$  ;crossover e mutação
21: avaliar  $P(t)$  ;calcula  $f(i)$  para cada indivíduo
22: SE (diferença $\geq z$ ) ENTÃO
23: Selecionar Indivíduo_Aleatório da população  $P(t)$ ;
24: Melhor_Encontrado  $\leftarrow$  BuscaLocal-
SA(Indivíduo_Aleatório);
25: Comunicação (Melhor_Encontrado( $t$ ),  $EP(t)$ );
26: Atualização  $EP(t)$ ; ;uso de operadores culturais
27: FIMSE
28: FIMENQUANTO
29: FIM

```

4 Resultados

Foram realizados testes com o TSP para duas instâncias de cidades, uma com número de cidades iguais a 442 (PCB442) e outra com 532 cidades

(Att532). Essas instâncias são provenientes da biblioteca TSPLIB (TSPLIB, 2012). Os resultados do algoritmo proposto foram comparados com o trabalho de He *et al.* (2006), que apresenta um algoritmo de Busca Paralela Tabu (*Parallel Tabu Search* - PTS) para PCB442 e Att532. Outros trabalhos utilizados para comparação foram: o de Cheng *et al.* (2002), que utiliza um Algoritmo Genético baseado em Agrupamentos; o de Verhoeven *et al.* (1995), que utiliza um algoritmo 2-opt paralelo; o de Wang *et al.* (2011), que utiliza um Algoritmo Memético com busca local do tipo Lin–Kernighan e por último o uso de um Algoritmo Cultural com população gerada pelo Algoritmo Genético dos próprios autores (sem as melhorias de busca de vizinhança e de busca local). A Tabela 1 mostra os resultados do AC proposto em comparação com os resultados da literatura.

Tabela 1 – Resultados obtidos

	Instância do TSP	
	PCB442	Att532
Ótimo Conhecido	50778	27686
Algoritmo	GAP % (Erro)	
Busca Tabu Paralela (He <i>et al.</i> , 2006)	0,12	3,85
Algoritmo Genético (Cheng <i>et al.</i> , 2002)	12,76	18,41
Algoritmo 2-opt Paralelo (Verhoeven <i>et al.</i> , 1995)	----	6,5
Algoritmo Memético com Busca Local L-K (Wang <i>et al.</i> , 2011)	0,00	----
Algoritmo Cultural	8,2	10,75
Algoritmo Cultural com <i>simulated annealing</i> e 2-opt mais 3-opt (Algoritmo Memético) Proposto)	0,0	0,07

A medida do GAP % (Erro) é calculada de acordo com a Equação (1).

$$GAP \% (Erro) = \frac{melhor - otimo}{otimo} \times 100 \quad (1)$$

As figuras a seguir mostram o comportamento do algoritmo memético proposto, a partir dos caminhos encontrados pela população inicial e após algumas gerações, além da solução final.

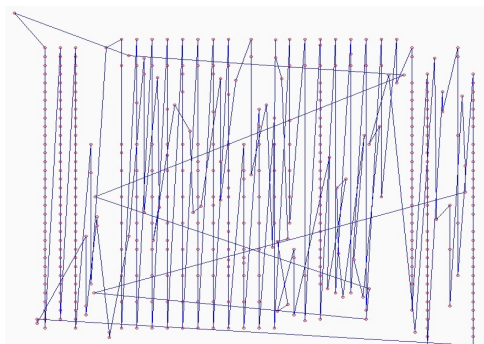


Figura 3. Caminho definido pela população inicial da instância PCB442 do TSP, para o Algoritmo Memético Proposto.

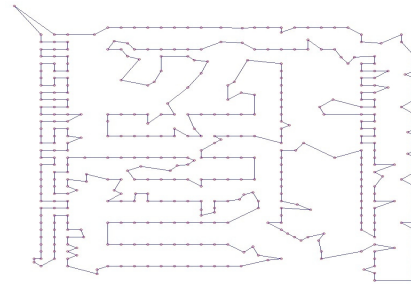


Figura 4. Caminho definido com a distância percorrida igual a 51050 da instância PCB442 do TSP, para o Algoritmo Memético Proposto.

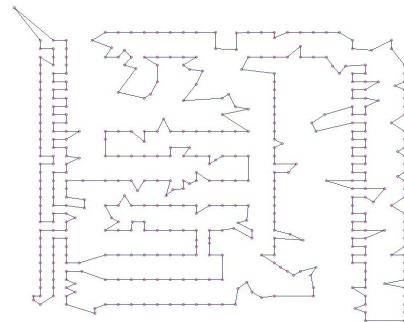


Figura 5. Caminho definido com a distância percorrida igual a 50778 da instância PCB442 do TSP, para o Algoritmo Memético Proposto.

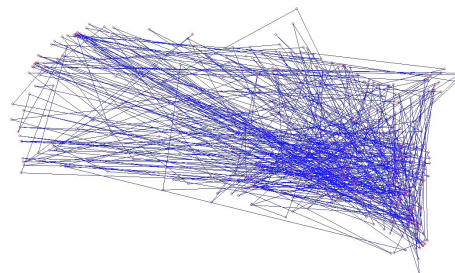


Figura 6. Caminho definido pela população inicial da instância Att532 do TSP, para o Algoritmo Memético Proposto.

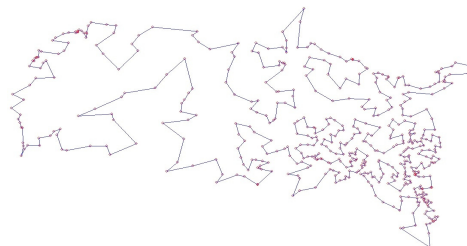


Figura 7. Caminho definido com a distância percorrida igual a 27912 da instância Att532 do TSP, para o Algoritmo Memético Proposto.

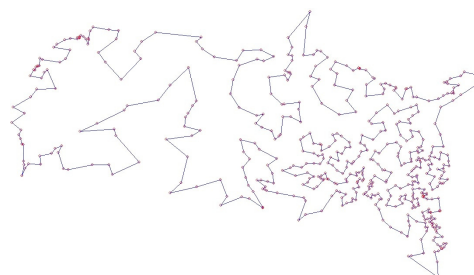


Figura 8. Caminho definido com a distância percorrida igual a 27686,0002528 da instância Att532 do TSP, para o Algoritmo Memético Proposto.

As duas figuras seguintes mostram o comportamento do algoritmo memético proposto geração a geração.

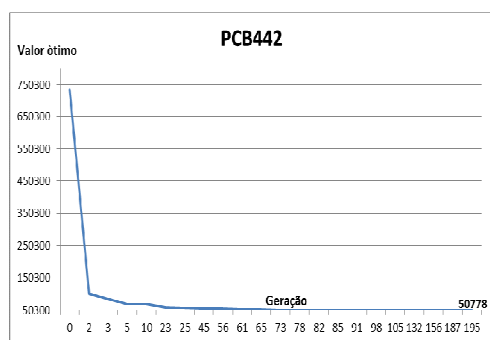


Figura 9. Evolução do melhor caminho, geração a geração, da instância PCB442 do TSP, para o Algoritmo Memético Proposto.

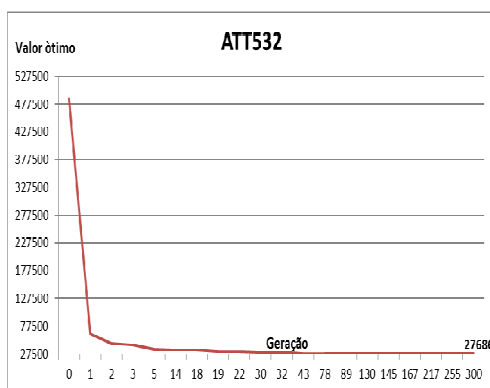


Figura 10. Evolução do melhor caminho, geração a geração, da instância Att532 do TSP, para o Algoritmo Memético Proposto.

Os resultados mostram que o algoritmo proposto encontrou o ótimo conhecido para a instância **pcb442** e chegou a um resultado muito próximo para a instância **att532**. Da Tabela 1 observa-se que o algoritmo proposto tem um melhor desempenho do que os outros três algoritmos da literatura e o AC+AG. O comportamento do algoritmo memético proposto, visto nas Figuras 9 e 10, mostram um mesmo perfil para as duas instâncias do TSP. Vale ressaltar que cada execução para essas instâncias levou em torno de 3 horas (computador Core 2 duo 1.8 GHz, 2 GB RAM), com o código em Java do framework Java Evolutionary Framework –JEF (Ventura et al., 2008), embora neste trabalho não tenha sido explorado os parâmetros de complexidade na comparação dos resultados.

4 Conclusão

No problema do TSP, os autores realizaram vários experimentos apenas com o AC, e a população gerada pelo AG, e em vários casos o AC não conseguiu escapar de ótimos locais. Mais ainda, o mesmo aconteceu com o uso do AC+AG, como no caso anterior, acrescido apenas do *simulated annealing*. O mesmo comportamento aconteceu para o caso de uso do AC+AG acrescido apenas da busca local 2-opt, ou da 3-opt. Observou-se que a inclusão das heurísticas de busca de vizinhança, baseado no *simulated annealing*,

e busca local, por meio do 2-opt + 3-opt, para a redução do custo computacional, permitiu alcançar valores ótimos tão bons e/ou melhores dos que outras meta-heurísticas utilizadas para tal problema, em um tempo polinomial como, por exemplo, Busca Tabu Paralela, AGs. Alcança resultados tão bons ao Algoritmo Memético com Busca Local L-K. Apesar de apresentar um maior custo computacional em relação ao L-K, o algoritmo proposto é bem mais simples de ser implementado.

O resultado obtido apoia a ideia de que a hibridização dos Algoritmos Culturais, com busca de vizinhança e busca local, é uma abordagem altamente desejável para direcionar o espaço de busca para encontrar soluções mais próximas do valor ótimo.

Agradecimentos

Este trabalho contou com o apoio, na forma de bolsa de doutorado, da Fundação de Amparo a Pesquisa do Estado do Pará (FAPESPA).

Referências Bibliográficas

- Agarwala, R., Agarwala, D. L., Maglott, D., Schuler, G. D., Schaffer, A. A. (2000) A Fast and Scalable Radiation Hybrid Map Construction and Integration Strategy. *Genome Research* 10:350-364.
- Ahmed, Z., Younas, I. (2011). A Dynamic Programming based GA for 0-1 Modified Knapsack Problem. *International Journal of Computer Applications*, Volume 16– No.7, pp. 1-6, February.
- Badillo, A. R., Ruiz, J. J., Cotta, C., Fernández-Leiva, A. J. (2013) On user-centric memetic algorithms. *Soft Computing*, Vol. 17, Issue 2, pp- 285-300.
- Back T, Fogel, D.B, Michalweicz, Z., (1997). *Handbook of Evolutionary Computation*, IOP Press, NY, (eds.).
- Becerra, R. L.. (2002). *Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Multiobjetivo*. Tese (Doutorado) - Instituto Politécnico Nacional do México.
- Carvalho, A. B., Barriveira, R. Algoritmo Cultural para o Problema do Caixeiro Viajante Relatório Técnico (UFPR) disponível em <http://www.inf.ufpr.br/aurora/disciplinas/topicosia2/downloads/trabalhos/CulturaisTSP.pdf>, acessado em outubro de 2009.
- Cheng, C.-H., Lee, W.-K., Wong, K.-F., (2002). A Genetic Algorithm-Based Clustering Approach for Database Partitioning. *IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews*, Vol. 32, No. 3, pp.215-230.

- Cotta, C., Troya, J. M. (1998). A Hybrid Genetic Algorithm for the 0-1 Multiple Knapsack Problem. *Artificial Neural Networks and Genetic Algorithms 3*, Springer-Verlag, Berlin, 250-254.
- Davendra, D., Eds. (2010). *Traveling Salesman Problem, Theory and Applications*. InTech, Rijeka, Croatia.
- Delin, L., Lixiao, Z., Zhihui, X., (2011). Heuristic Simulated Annealing Genetic Algorithm for Traveling Salesman Problem. 6th International Conference on Computer Science & Education (ICCSE), On page(s): 260 - 264, Volume: Issue: 3-5, August.
- Digalakis, J. G., Margaritis, K. G. (2002) A multipopulation cultural algorithm for the electrical generator scheduling problem. *Mathematics and Computers in Simulation*, 60, pp. 293–301.
- Dorigo, M., Maniezzo V., Coloni A. (1996). The ant system optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, Vol. 26, No. 1, pp. 29–41.
- Duan, H. B., Wang, D. B., Yu, X. F., (2006). MAX-MIN meeting ant colony algorithm based on cloud model theory and niche ideology. *Journal of Jilin University (Engineering and Technology Edition)*, 36(5): 803-808.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, USA.
- Goldberg, D. E., Voessner, S. (1999) .Optimizing Global-Local Search Hybrids. IlliGal Report No. 99001, Depart. General Engineering, University of Illinois at Urbana-Champaign, USA.
- He, Y., Guanyuan, L., Qiu, Y., (2006). A Parallel Tabu Search Algorithm Based on Partitioning Principle for TSPs. *Proceedings of IJCSNS International Journal of Computer Science and Network Security*, VOL.6 No.8A, August.
- Henderson, D., Jacobson, S. H., Johnson, A. W., (2003). The theory and practice of simulated annealing. Glover, F. and Kochenberger, G. (eds), *Handbook on Metaheuristics*, pp. 287–319. Kluwer Academic Publishers, Norwell, MA.
- Hill, R. R., Hiremath, C. (2005). Improving genetic algorithm convergence using problem structure and domain knowledge in multidimensional knapsack problems. *Int. J. Operational Research*, Vol. 1, Nos. 1/2, pp.145–159.
- Hong, T.-P., Lin, W.-Y., Liu, S.-M., Lin, J.-H. (2007). Dynamically Adjusting Migration Rates for Multi-Population Genetic Algorithms. *JACIII* 11(4): 410-415.
- Jagtap, S. B., Pani, S. K., Shinde, G., (2011). A Hybrid Parallel Multi-Objective Genetic Algorithm for 0/1 Knapsack Problem. *JSEA* 4(5): 316-319.
- Jiao, L. C., Wang, L. A., (2000). Novel Genetic Algorithm Based on Immunity. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, Sept., 30(5): 552-561.
- Johnson, D., McGeoch, L. (1997). The Traveling Salesman Problem: A Case Study in Local Optimization. *Local Search in Combinatorial Optimization*, E. H. L. Aarts and J. K. Lenstra (editors), John Wiley and Sons, Ltd., pp. 215-310.
- Karaboga, D., Gorkemli, B., (2011). A Combinatorial Artificial Bee Colony Algorithm for Traveling Salesman Problem. *International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011)*, 15-18, Istanbul, Turkey, June.
- Kellerer, H., Pferschy, U., Pisinger, D. (2004). *Knapsack Problems*. Springer Verlag, Berlin, Germany.
- Kennedy J., Eberhart R. C. (2001). *Swarm Intelligence*. Morgan Kaufmann, San Francisco, USA.
- Kim, Y., Cho, S.-B., (2009). A Hybrid Cultural Algorithm with Local Search for Traveling Salesman Problem. *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, page(s): 188 - 192, 15-18 Dec.
- Lin, J.-Y., Chen, Y.-P, (2011). Analysis on the Collaboration Between Global Search and Local Search in Memetic Computation. *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, october, pp. 608 - 623.
- Lin, S., Kernighan, B., (1973). An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Operations Research* 21, 498–516.
- Lin, Y.-C. (2013) A Mixed-Integer Memetic Algorithm Applied to Batch Process Optimization. *Applied Mechanics and Materials*, Vols. 300-301, pp. 645-648.
- Linden, R., (2006). *Algoritmos Genéticos*. 1. ed. Rio de Janeiro: Brasport,. v. 1, 372 p.
- Meuth, R., Lim, M.-H., Ong, Y.-S., Wunsch, D., (2009). A proposition on memes and meta-memes in computing for higher-order learning. *Memetic Comput.*, vol. 1, no. 2, pp. 85–100.
- Michalewicz, Z., (1998). *Genetic Algorithms + Data Structures=Evolution Programs*. Third, Revised and Extended Edition, Springer Verlag, Berlin, Germany.
- Neri, F., Cotta, C., Moscato, P. (Eds.), (2011). *Handbook of Memetic Algorithms*, Studies in Computational Intelligence, Springer, Vol. 379, November.
- Neri, F., Cotta, C., (2012). Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2, pp. 1-14.
- OR-Library, (1990). <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>. Acessado em 04 de abril de 2012.
- Osborne, M. (2010) Bayesian Gaussian processes for sequential prediction, optimisation and

- quadrature. DPhil thesis, University of Oxford, Oxford, UK.
- Puchinger, J., Raidl, G. (2005). Cooperating memetic and branch-and-cut algorithms for solving the multidimensional knapsack problem. *Proceedings of the 2005 Metaheuristics International Conference*, Vienna, Austria, pp. 775–780.
- Purisa, A., Belloa, R., Herrera, F. (2010) Analysis of the efficacy of a Two-Stage methodology for ant colony optimization: Case of study with TSP and QAP. *Expert Systems with Applications*, Volume 37, Issue 7, July, Pages 5443–5453.
- Rao, S. S. (2009). *Engineering Optimization – Theory and Practice*. John Wiley & Sons, New Jersey, USA.
- Reynolds, R. G. (1994). An introduction to cultural algorithm. *Proceedings of the Third Annual conference on Evolutionary Programming*, Sebalk, A. V. e Fogel, L. J., editors, pages 131–139, River Edge NJ. World Scientific Publishing.
- Reynolds, R.G., Che, X. D., Ali, M. (2010). Weaving the social fabric: The past, present and future of optimization problem solving with cultural algorithms. *Int. J. Intell. Comput. Cybern.*, 3(4): 561-592.
- Reynolds, R.; Zhu, S.. (2001). Knowledge-based function optimization using fuzzy cultural algorithms with evolutionary programming. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, v. 31, n. 1, p. 1-18.
- Tian, P., Ma, J., Zhang, D. M., (1999). Application of the simulated annealing algorithm to the combinatorial optimisation problem with permutation property: an investigation of generation mechanism. *European Journal of Operational Research*, 118, 81–94.
- TSPLIB. [http://http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html](http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html). Acessado em 04 de abril de 2012.
- van Laarhoven, P., Aarts E. (1987). *Simulated Annealing: Theory and Applications*. D. Reidel, Boston, USA.
- Ventura, S., Romero, C., Zafra, A., Delgado, J. A., Hervás, C., (2008). JCLEC: a Java framework for evolutionary computation. *Soft Comput.*, Vol. 12, Nr. 4 (2008) , p. 381-392.
- Verhoeven, M. G. A., Aarts, E. H. L., Swinkels, P. C. J., (1995). A parallel 2-opt algorithm for the Traveling Salesman Problem, *Future Generation Computer Systems*, Vol.11. No.2, pp.175-182.
- Wang, X. P., Cao, L. M., (2002). *Genetic algorithm: theory application and software implementation*. Xi'an: Press of Xi'an Jiaotong University.
- Wang, Y.-t., Li, J.-q., Gao, K.-z., Pan, Q.-k. (2011) Memetic Algorithm based on Improved Inver-over operator and Lin-Kernighan local search for the Euclidean traveling salesman problem. *Computers and Mathematics with Applications*, 62, pp. 2743–2754.
- Wang, Z. G., Wong, Y. S., Rahman, M. (2005) Development of a parallel optimization method based on genetic simulated annealing algorithm. *Parallel Computing*, Volume 31, Issues 8–9, August–September, Pages 839–857.
- Weise, T. (2009). *Global Optimization Algorithms - Theory and Application*. Online e-book under GNU Free Documentation License, available at <http://www.it-weise.de/projects/book.pdf>.
- Xu, Q., Zhang, G., Zhao, C., An, A. (2011) A robust adaptive hybrid genetic simulated annealing algorithm for the global optimization of multimodal functions. *Chinese Control and Decision Conference (CCDC)*, May, Mianyang, China, pp. 7-12.
- Xue, Z., Guo, Y. (2007). Improved Cultural Algorithm based on Genetic Algorithm. *IEEE International Conference on Integration Technology - ICIT '07*, March, Shenzhen, China, pp. 117-122.
- Yan, X., Wu, Q., Zhang, C., Chen, W., Luo, W., Li, W. (2012) An Efficient Function Optimization Algorithm based on Culture Evolution. *International Journal of Computer Science Issues*, Vol. 9, Issue 5, No 2, pp. 11-18.
- Yew-Soon, O., Meng, H. L., Xianshun, C., (2010). Memetic Computation—Past, Present & Future. *IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE*, Volume: 5, Issue: 2, MAY, pp. 24-31.
- Zhu, L., Aono, M., Kim, S.-J., Hara, M. (2013) Amoeba-based computing for traveling salesman problem: Long-term correlations between spatially separated individual cells of *Physarum polycephalum*. *BioSystems*, Vol. 112, pp. 1–10.