HEURÍSTICAS PARA O PROBLEMA DA CADEIA DE CARACTERES MAIS PRÓXIMA

Marcelo Henrique Ribeiro de Almeida
Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG
Av. Amazonas, 7675 – Nova Gameleira – CEP 30510-000 –
Belo Horizonte, MG, Brasil
marcelohra@yahoo.com.br

Rafael Vilela Rabelo
Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG
Rua Álvares de Azevedo 400 - Bela Vista – CEP 35503-822–
Divinópolis, MG, Brasil
rafaelvrabelo@hotmail.com

Daniel Morais dos Reis
Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG
Av. Amazonas, 7675 – Nova Gameleira – CEP 30510-000 –
Belo Horizonte, MG, Brasil
daniel.morais@gmail.com

Sérgio Ricardo de Souza
Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG
Av. Amazonas, 7675 – Nova Gameleira – CEP 30510-000 –
Belo Horizonte, MG, Brasil
sergio@dppg.cefetmg.br

RESUMO

O Problema da Cadeia de Caracteres mais Próxima (PCCP) tem como objetivo encontrar uma sequência t mais próxima possível de todas as sequências de um conjunto dado, usando como métrica a distância de *Hamming*. O PCCP possui várias aplicações, em especial, na área da bioinformática e na Teoria de Códigos. Classificado como um problema de complexidade NP-Difícil, vários algoritmos de aproximação e metaheurísticas têm sido propostas para encontrar soluções ótimas e com tempos de processamentos aceitáveis. Neste trabalho, um novo algoritmo é proposto, baseado no framework do Algoritmo Genético *Biased Random-Key Genetic Algorithm (BRKGA)* e será comparado com o algoritmo *Iterated Greedy Search (IGS)*.

PALAVARAS CHAVE. Problema da Cadeia de Caracteres mais Próxima. Hamming. BRKGA.

ABSTRACT

Closest String Problem (CSP) aims to find a string t as close as possible of all the strings of a given set, using as metrics the Hamming distance. CSP has several applications, in special, in the area of Bioinformatics and in Coding Theory. Classified as a problem of complexity NP – Hard, several approximation algorithms and metaheuristic, have been proposed to find great solutions and with time of processing acceptable. In this work, a new algorithm is proposed, based on the framework of Genetic Algorithm *BiasedRandom-Key GeneticAlgorithm (BRKGA)* and will be compared with the algorithm *IteratedGreedySearch (IGS)*.

KEYWORDS. Closest String Problem. Hamming. BRKGA.

1. INTRODUÇÃO

No Problema da Cadeia de Caracteres mais Próxima (PCCP), deseja-se encontrar uma sequência de caracteres t que se aproxime ao máximo, segundo uma métrica, de todas as sequências de um dado conjunto, no qual as cadeias de caracteres possuem a mesma dimensão. Neste trabalho, a ideia principal é encontrar uma sequência centro t que se assemelhe, ao máximo, a um conjunto de cadeias dado, ou seja, o objetivo é minimizar a distância máxima desta cadeia de caracteres t às demais cadeias do conjunto.

Desde 1986 e do surgimento do Projeto Genoma Humano, tem ocorrido um grande crescimento da investigação científica em atividades dedicadas à biologia molecular, como o entendimento das interações entre os diversos sistemas de uma célula, incluindo as interações de DNA, RNA e sínteses proteicas. De acordo com Festa (2007), esse crescente desenvolvimento na área da biologia teve o interesse de pesquisadores na área da computação, auxiliando no desenvolvimento de programas capazes de elaborar uma grande quantidade de dados armazenados.

Representando um tema de pesquisa ativa na área de otimização combinatória, o PCCP tem sido mais frequente em biologia molecular e na teoria da codificação. Geralmente, estes problemas apresentam a necessidade de se comparar cadeias de caracteres (ou sequências) de DNA (Ácido Desoxirribonucleico), RNA (Ácido Ribonucleico) ou proteínas, a fim de encontrar determinadas características presentes em cada uma delas, o que levou a problemas como rearranjo do genoma, ver Li et al. (2006), alinhamento de sequências, em Edgar e Batzoglou (2006), e problemas de sequências de consenso em Kennelly e Krebs (1991), a maioria dos quais são conhecido como NP-Difícil como mostrado em Lanctot et al. (2003).

2. DESCRIÇÃO DO PROBLEMA E CONCEITOS BÁSICOS

2.1. DISTÂNCIA DE HAMMING

Várias métricas têm sido propostas para encontrar as similaridades (ou diferenças) entre as sequências. A mais utilizada é à distância de *Hamming*, uma métrica que aparece em vários contextos (Andronescu e Rastegari (2003), Yamamoto (2004), entre outros. Uma justificativa técnica para o uso frequente da distância de *Hamming* na comparação de sequências em biologia molecular pode ser encontrada no trabalho de Li et al. (1999).

A distância de *Hamming* entre duas cadeias de caracteres corresponde ao número de posições nas quais as sequências se diferem, ou seja, compara-se os caracteres de mesma posição entre duas sequências, a cada comparação em que os caracteres são distintos, é contada uma unidade. Assim para o alfabeto $\Omega = \{A, C, G, T\}$ e um conjunto de cadeias de caracteres $S = \{s_1, s_2, s_3\}$, em que $s_1 = (GATTG)$, $s_2 = (GATCA)$, $s_3 = (CTCGA)$, se $t = (GATTC)$ então, a distância de *Hamming* entre t e s_1, s_2, s_3 é dada por: $d_H(s_1, t) = 1$, $d_H(s_2, t) = 2$, $d_H(s_3, t) = 5$

A distância a ser minimizada será a da sequência s_3, pois $d_H(s_3, t) = 5$ é a maior d_H computada e o objetivo do problema é encontrar uma sequência t, mais próxima de todas as sequências do conjunto S ao mesmo tempo. Todas essas sequências possuem a mesma dimensão e são construídas a partir de um único alfabeto, que é utilizado para definir as cadeias de caracteres. Estas cadeias são utilizadas para representar as bases nitrogenadas do DNA, na forma do alfabeto $\Omega = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$.

2.2. DESCRIÇÃO DO PROBLEMA

O PCCP consiste, como dito, em minimizar a distância máxima de *Hamming*, que é dada entre duas cadeias de caracteres, sendo uma a solução corrente, a qual deve ser comparada com as demais cadeias do conjunto. Em se tratando de otimização combinatória, o PCCP é considerado um problema NP-Difícil, ou seja, sua complexidade é não polinomial. Sendo assim,

as técnicas mais comumente empregadas para resolver problemas NP-Difíceis são algoritmos heurísticos e algoritmos exatos que possuem complexidade de tempo não polinomial.

O PCCP é definido como segue: dado um conjunto $\Sigma = \{s^1, \dots, s^n\}$ de sequências de caracteres com $s^i = \{s_{1,\dots,}s_m\} \in \Omega^m$ para $i = 1, 2, \dots, n$. O objetivo do PCCP é encontrar uma sequência centro $t \in \Omega^m$ tal que, $D(t) = max_i d_H(s^i, t)$ é a menor possível, ou seja, a distância máxima entre t e as demais sequências do conjunto deve ser minimizada. Sendo que Ω é o alfabeto utilizado, ou seja, um conjunto finito de elementos chamados caracteres. No qual $\Omega = \{1, \dots, \omega\}$ com $\omega \in \mathbb{Z}$, então definimos Ω^m como sendo o conjunto de todas as sequências de tamanho m formado pelos elementos de Ω.

2.3. FORMULAÇÃO MATEMÁTICA PARA O PCCP

A função objetivo do PCCP baseia-se nas definições da distância de *Hamming* entre duas cadeias de caracteres de mesmo comprimento. Dadas duas sequências s e t em Ω (alfabeto), tais que $|s| = |t| = m$ (dimensão da sequência), a distância de *Hamming* entre s e t, denotada por $d_H(s, t)$, é calculada pela seguinte expressão

$$d_H(s, t) = \sum_{i=1}^{|s|} \phi(s_i, t_i); \quad (1)$$

em que a função $\phi(.)$ é tal que:

$$\phi: \Omega \; x \; \Omega \rightarrow \{0,1\} \quad (2)$$

sendo:

$$\phi(a, b) = \begin{cases} 0, & se\ a = b; \\ 1, & caso\ contrário. \end{cases} \quad (3)$$

No início do processo, a distância de *Hamming* (d_H) recebe o valor zero e, a cada diferença encontrada entre elementos de uma mesma posição, mas em cadeias diferentes, é acrescida uma unidade ao valor de d_H. O processo se repete até que todas as cadeias da instância dada sejam comparadas com a cadeia que representa a solução atual, que deverá ser modificada com a finalidade de minimizar o valor máximo de d_H. A distância máxima d_H deverá ser minimizada, mantendo todas as outras abaixo dela.

Desta forma, não basta apenas calcular essa distância na totalidade das cadeias, mas obter o valor da distância de *Hamming* individualmente e fazer alterações na solução que minimize essa distância máxima encontrada. Esse valor poderá se modificar de uma linha para outra, porém deve diminuir a cada busca.

O seguinte modelo de programação inteira, é uma das três formulações apresentadas por Meneses et al. (2004), algumas formulações diferentes podem ser encontradas nos trabalhos de Kelsey e Kotthoff (2010), Soleimani-damaneh (2011), dentre outros trabalhos. Tais variações são decorrentes do método a ser utilizado, todos com base na distância de *Hamming*. Os resultados computacionais deste modelo serão apresentados na seção 5.

Sejam, então, as variáveis de decisão Z_k^i, definidas como:

$$Z_k^i = \begin{cases} 1 & \text{se } t_k \neq x_k^i; \\ 0 & \text{caso contrário.} \end{cases}$$

O PCCP, assim, é formulado como:

$$\text{minimizar} \quad d \quad (4)$$

sujeito a:

$$\sum_{k=1}^{m} z_k^i \leq d; \; i = 1, \dots, n \quad (5)$$

$$t_k - x_k^i \leq kz_k^i \; i = 1, \ldots, n \; ; k = 1, \ldots, m \quad (6)$$

$$x_k^i - t_k \leq kz_k^i \; i = 1, \ldots, n \; ; k = 1, \ldots, m \quad (7)$$

$$z_k^i \in \{0,1\} \qquad i = 1, \ldots, n \; ; k = 1, \ldots, m \quad (8)$$

$$d \in Z_+ \quad (9)$$

$$t_k \in Z; k = 1, \ldots, m \quad (10)$$

Neste modelo, tem-se que: k é o tamanho do alfabeto usado; d representa a diferença máxima entre t e a cadeia de caracteres testada, e é o valor a ser minimizado; a restrição (5), determina o limite inferior para o valor d, que deve ser maior ou igual ao número total de diferenças para cada cadeia do conjunto; as restrições (6) e (7) determinam os limites inferior e superior para a diferença entre $t_k - x_k^i$; as restrições (8) e (10), garantem valores inteiros para a solução.

3. TRABALHOS RELACIONADOS

A palavra consenso aparece no trabalho de Ben-Dor et al. (1997) para definir os caracteres de uma sequência, e o autor faz o uso de programação inteira para modelar o PCCP e utiliza um algoritmo de aproximação com base em arredondamento randomizado. Ele também retrata em seu trabalho o *Distinguishing String Selection Problem (DSSP)*.

Além dos três problemas de seleção de cadeias que Festa (2007) retrata em seu trabalho, Lanctot et al. (2003), aborda mais quatro problemas. São eles: *Closest Substring Problem*, *Farthest Substring Problem*, *Close to Most String Problem*, *Distinguishing String Selection Problem*. A fim de solucionar estes problemas, o autor propõe técnicas de relaxamento de programação linear, algoritmos de aproximação e um algoritmo heurístico, que mostrou resultados eficientes para o *Closest Substring Problem*.

No trabalho de Meneses et al. (2004) foi estudado o PCCP. Os autores desenvolveram três formulações de programação inteira e uma heurística, usadas para fornecer limite superior para o valor de uma solução ótima. Desenvolveram um algoritmo *branch-and-bound* baseado na heurística apresentada e em uma das formulações. São utilizadas, no trabalho, três classes de instâncias: a primeira tem como alfabeto o conjunto $\{0,1\}$; a segunda utiliza um alfabeto de quatro caracteres; a terceira utiliza um alfabeto de 20 caracteres (representando sequências de proteínas).

Festa (2007), em seu trabalho, faz uma descrição detalhada de problemas biológicos, formulados como problemas de otimização combinatória, e propõe uma nova heurística, com a finalidade de encontrar boas soluções para uma classe particular desses problemas, conhecida como *Far From Most String Problem*, dando uma ênfase especial à formulação matemática e a métodos eficientes de programação inteira.

Um algoritmo exato chamado *Distance First Algorithm (DFA)*, foi apresentado por Liu et al. (2011) para resolver problemas do PCCP com três cadeias e alfabeto de tamanho dois. Para o *Closest String Problem* (CSP) geral foi projetada uma heurística polinomial, que é uma combinação dessa proposta com a aproximação do algoritmo *Largest Distance Decreasing Algorithm (LDDA)*, encontrado em Liu e Shao (2004). O autor utilizou dados biológicos reais e simulados, sendo seis instâncias reais de alfabeto 20 do conjunto de dados do trabalho de McClure et al. (1994).

Mousavi e Esfahani (2012) utilizam a heurística *Greedy Randomized Adaptive Search Procedure* (GRASP), originalmente proposto por Feo e Resende (1995), para solucionar o PCCP. Os autores utilizam uma função que avalia o custo de duas soluções candidatas diferentes, porém com mesma distância de Hamming, diminuindo assim o custo computacional do algoritmo. Os

resultados computacionais do algoritmo GRASP-CSP proposto por Mousavi e Esfahani (2012), foram comparados com os resultados das heurísticas dos trabalhos de Liu et al. (2008b), Julstrom (2009), Faro e Pappalardo (2009) e Hufsky et al. (2011). Os autores, na maioria das comparações, conseguiram boas soluções com tempos de processamentos reduzidos, comprovando assim a competitividade do algoritmo proposto.

4. METODOS PROPOSTOS PARA O PCCP

Nesta seção serão apresentadas duas heurísticas para a resolução do PCCP, a heurística de Busca Gulosa Iterativa (*Iterated Greedy Search - IGS*) e o Algoritmo Genéticos Baseado em Chaves Aleatórias Tendenciosas (*Biased Random-Key Genetic Algorithm - BRKGA)*.

4.1. HEURÍSTICA BRKGA-PCCP

Um Algoritmo Genético Baseado em Chaves Aleatórias (do inglês – Biased Random Key Genetic Algorithm (BRKGA)) foi proposto em Garcia et al. (1998). O uso de Algortimo Genéticos Baseado em Chaves Aleatórias foi motivado pelo sucesso da utilização desta metaheurística para a resolução de problemas de otimização combinatória (Arulselvan et al. (2007), Framinan e Leisten (2008), Goulart et al. (2011), Hyytiä, e Virtamo (1998), Spears, e deJong (1991)).

Os cromossomos em BRKGA-PCCP são compostos por vetores de caracteres. Cada caractere recebe um número real aleatório entre 0 e 1. Posteriormente, a população é ordenada em ordem crescente, de acordo com o valor aleatório atribuído à cada caractere. A distância de *Hamming* entre as cadeias de caracteres e a sequência resultante da ordenação dos caracteres é calculada. A Distância de *Hamming* calculada por BRKGA-PCCP é atribuída como valor de adaptabilidade do cromossomo.

Em todas as gerações, a população é dividida em dois conjuntos, denominados TOP e $REST$. O conjunto TOP da população armazena os cromossomos com as melhores soluções. Estes indivíduos são copiados, sem alterações, para as próximas gerações. O restante da população da nova geração é encontrado pela aplicação de operadores de cruzamento e mutação.

O cruzamento é realizado entre um cromossomo da parcela TOP com um cromossomo da parcela $REST$. Cada filho recebe suas chaves ou da porção da população mais apta (ou TOP), com uma probabilidade de 0.25, ou a chave é herdada do cromossomo pertencente à população menos apta (ou $REST$) com probabilidade de 0.7. Assim, o tamanho total da população inicial é $TOP + REST$, e a quantidade de cromossomos criados a cada geração através de cruzamento é $REST + BOT$ cromossomos filhos. Cromossomos mutantes são colocados na parcela BOT (com probabilidade de 0.05) da nova população gerada.

Não existe mutação no algoritmo BRKGA-PCCP, pois, a cada geração, um número fixo de cromossomos mutantes é inserido na população. Para escapar de mínimos locais, uma parte dos cromossomos menos aptos é substituída por cromossomos gerados aleatoriamente, de modo semelhante à forma como é gerada a população inicial. A figura 1 exibe o pseudocódigo da heurística BRKGA-PCCP.

BRKGA-PCCP (G, R, p_{rec}, T_{max})

```
1-  Para cada elemento da cadeia de caracteres faça
2-      Gerar vetor com chaves aleatórias;
3-      Avaliar vetor de acordo com a Distância de Hamming;
4-  fim-Para
5-  Enquanto Tempo < T_max faça
6-      Ordene a população pela Distância de Hamming;
7-      Divide em TOP, REST e BOT;
8-      Copiar para a próxima geração a parcela TOP da população;
9-      Para n ← 1 até |REST| faça
10-         Selecione aleatoriamente um elemento da parcela TOP;
11-         Selecione aleatoriamente um elemento da parcela REST∪BOT;
12-         Para cada chave do novo cromossomo faça
13-             p3 ← Crossover(p1, p2, p_rec);
14-         fim-Para
15-     fim-Para
16-     Gerar aleatoriamente parcela BOT da nova população;
17-     Avaliar cada elemento da população;
18- fim-Enquanto
```

Figura 1: Pseudocódigo do Algoritmo Genético com Chaves Aleatórias para o Problema da Cadeia de Caracteres mais Próxima – BRKGA-PCCP.

Na linha 2 deste algoritmo, os vetores de chaves são inicializados com valores aleatórios. Na linha 3, é calculada a Distância de *Hamming* para cada vetor. O laço entre as linhas 5 e 15 é executado até que o tempo T_{max} ou outro critério de parada seja alcançado. Na linha 6, a população é ordenada de acordo com a Distância de *Hamming* de cada indivíduo. Na linha 8, é realizada a reprodução da porção TOP da população, que é integralmente copiada para a próxima geração. Entre as linhas 11 e 13 é realizado o cruzamento, gerando os cromossomos da porção MID da população. Na linha 16 são gerados os novos elementos da parcela BOT, responsáveis pela diversificação da população. Na linha 17, a nova população é avaliada.

4.2. HEURÍSTICA IGS-PCCP

A heurística de Busca Gulosa Iterativa (IGS - *Iterated Greedy Search*) foi utilizada com sucesso em Fanjul-Peyroa e Ruiz (2010), Framinan e Leisten (2008), Ruiz e Stützle (2007), Garcia et al. (1998) e em outros problemas de otimização combinatória.

Esta heurística parte de uma solução inicial para um problema de otimização combinatória, para, em seguida, fazer buscas locais e melhorar diversas vezes a solução inicial encontrada através de um determinado número de iterações, segundo Gagnaire e Doumith (2007).

O IGS-PCCP gera aleatoriamente uma solução inicial s_0, na qual, posteriormente, é feita uma busca local utilizando o refinamento iterativo, resultando na solução intermediária s. Em seguida, é aplicada uma perturbação gulosa em k caracteres de s, removendo uma quantidade de caracteres selecionados aleatoriamente e atribuindo novos caracteres também selecionados aleatoriamente em suas posições, gerando, assim, a solução intermediária s'. Em seguinda, IGS-PCCP calcula a Distância de *Hamming* da nova cadeia de caracteres em relação às demais cadeias, e esta solução é comparada à solução atual. A cada iteração de IGS-PCCP, a solução

com menor Distância de *Hamming* encontrada pelo algoritmo é armazenada em s^*, sempre sendo comparada com a solução s' da iteração atual. A figura 2 exibe o pseudocódigo de IGS-PCCP.

início IGS-PCCP(s', T_{max})
1. $s, s^* \leftarrow$s';
2. **Enquanto** Tempo $< T_{max}$ **faça**
3. $s' \leftarrow$ Perturbação(k, s);
4. **Se** s' é melhor que s^* **então** $s^* \leftarrow s'$;
5. **Se** s é melhor que s' **então** $s \leftarrow s'$;
6. **fim-Enquanto**;
7. **retorne** s^*;

fim

Figura 2: Pseudocódigo da heurística IGS-PCCP

Começando a partir da solução inicial s' gerada aleatoriamente, as soluções ótimas locais s e s^* são criadas na linha 1. Em seguida, o laço das linhas 2-6 é realizado até o critério de parada ser satisfeito. Novos ótimos locais são obtidos através da aplicação de uma perturbação gulosa de tamanho k na linha 3. Em seguida, se a solução resultante é melhor que a solução corrente s^*, a solução corrente é atualizada na linha 4. Na sequência, o s do ótimo local corrente é substituído por um s' se sua Distância de *Hamming* for maior. A solução s' é aceita se o seu custo for menor que o custo de s. A melhor solução encontrada por esta heurística é retornada na linha 7.

5. EXPERIMENTOS COMPUTACIONAIS

Os algoritmos foram testados para um conjunto de 19 instâncias. O alfabeto utilizado foi o mesmo para todas elas $\Omega = \{A, C, G, T\}$. A tabela 1 mostra os resultados obtidos, sendo que cada heurística foi executada 5 vezes. As colunas n e m, Mín., Méd., Máx., representam respectivamente a dimensão das instâncias testadas, a menor distância de *Hamming*, a média das distâncias e distância máxima encontrada. As instâncias testadas foram retiradas do trabalho de Meneses et al. (2004), o qual apresentamos os resultados computacionais obtidos pelo modelo de programação inteira apresentado na seção 2.3.

Tabela 1: Resultados para 5 execuções para cada instância.

Instâncias		**BRKGA-PCCP**			**IGS-PCCP**			**Programação Inteira**			
n	m	Mín.	Méd.	Máx.	Mín.	Méd.	Máx.	Mín.	Méd.	Máx.	Tempo (s)
10	300	193	196	200	208	224,8	245	174	175.50	177	0.12
10	500	329	333,8	339	357	372,8	393	288	289.50	292	0.34
10	700	468	475	481	506	524,4	556	405	407.50	410	1.84
15	300	195	202,2	206	208	226,4	244	183	184.25	187	6.02
15	500	339	344,2	348	350	375,4	402	306	306.50	307	44.98
15	700	481	487,2	491	492	523	546	426	428.50	432	21.78
20	300	202	207,4	211	213	225,2	242	190	190.25	191	1,170.98
20	500	345	351,6	355	352	374,8	398	314	316.25	319	901.44
20	700	488	494,2	499	493	523,6	558	442	442.75	444	940.21
25	300	204	211	211	204	224,4	243	195	195.75	196	2,741.15
25	500	343	356,2	362	341	374,4	396	321	323.25	325	741.62
25	600	414	425,4	430	427	448,8	487	387	388.00	389	1,805.16
25	700	487	498,8	506	495	524,2	554	450	452.25	453	907.61
25	800	558	573	583	568	599,8	632	515	516.75	520	1,254.19
30	300	202	213,6	218	205	224,2	244	198	198.25	199	3,223.68
30	400	275	286,2	290	280	299,6	322	262	263.25	264	1,852.67
30	500	342	360,4	365	356	374,6	398	326	329.25	331	2,215.67
30	600	422	431,6	435	424	450,4	476	393	394.25	395	2,700.30
30	800	560	576	584	566	601	634	522	523.50	526	2,337.20

6. CONCLUSÕES E TRABALHOS FUTUROS

A partir dos resultados apresentados na seção 5, pode-se concluir que os algoritmos propostos para o Problema da Cadeia de Caracteres Mais Próxima demonstraram serem eficientes na resolução do problema em questão.

Observamos que a heurística IGS-PCCP não gerou bons resultados, ficando sempre a atrás da heurística BRKGA-PCCP. Entretanto, é importante ressaltar que o tempo de processamento foi limitado em 5 minutos para todas as instâncias, com 5 iterações cada. Com esse tempo de processamento, conseguimos obter resultados próximos dos valores ótimos.

Podemos observar que, para valores de n pequenos e $m = 300$, os valores ótimos têm diferenças de no máximo $\cong 9{,}8\%$. Ao passo que n aumenta, essa diferença cai para $\cong 1{,}9\%$, quando n alcança valor 30. Com relação ao tempo, para alcançar essa última diferença de $\cong 1{,}9\%$, nosso algoritmo BRKGA-PCCP se mostrou 90% mais rápido em alcançar uma solução próxima da ótima.

Conclui-se então que a heurística BRKGA-PCCP se mostrou bastante robusta, obtendo valores próximos do ótimo em um tempo de processamento reduzido.

Como trabalhos futuros, há a necessidade de um melhor ajuste dos parâmetros das heurísticas, como número de iterações, tempo de processamento, entre outros que devem ser feitos através de testes, desenvolvimento de outras formulações matemáticas e também a implementação de algoritmos aproximativos que ainda não foram testados.

REFERÊNCIAS

Andronescu, M. e Rastegari, B. (2003). Motif-grasp and motif-ils: Two new stochastic local search algorithms for motif finding. Relatório técnico, *Computer Science Department, University of British Columbia*, Vancouver, Canada.

Arulselvan, A., Commander, C., and Pardalos, P. (2007). A random keys based genetic algorithm for the target visitation problem. *In Advances in Cooperative Control and Optimization*, vol. 369. Springer Berlin Heidelberg, pp. 389–397.

Ben-Dor, A., Lancia, G., Perone, J. e Ravi, R. (1997). Banishing bias from consensus sequences. in: *Proceedings of the 8th Annual Symposium on Combinatorial Pattern Matching*, v. 247–261Terceira referência.

Edgar, R. C., Batzoglou S. (2006). Multiple sequence alignment. *Current Opinion in Structural Biology*, 16(3), pp.368-373, June 2006.

Fanjul-Peyroa, L. e Ruiz, R. (2010). Iterated greedy local search methods for unrelated parallel machine scheduling. *European Journal of Operational Research* 207:55–69.

Faro, S. e Pappalardo, E. (2009). An ant colony optimization algorithm for the closest string problem. *Proceedings of the 36th Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM '10*, p. 370 – 381, Berlin, Heidelberg. Springer-Verlag. ISBN 978-3-642-11265-2. doi: http: //dx.doi.org/10.1007/978-3-642-11266-9_31. URL http://dx.doi.org/10.1007/ 978-3-642-11266-9_31.

Feo, T.A. e Resende, M.G.C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, p. 109–133.

Festa, P. (2007). On some optimization problems in molecular biology. *Mathematical Biosciences*, v. 207, p. 219–234.

Framinan, J., and Leisten, R. (2008). A multi-objective iterated greedy search for flowshop scheduling with makespan and flowtime criteria. *OR Spectrum* 30, 787–804. 10.1007/s00291-007-0098-z.

Gagnaire, M. e Doumith, E. (2007). An iterative greedy algorithm for scheduled traffic grooming in WDM optical networks. *Proceedings of the Advanced Networks and Telecommunication Systems, First International Symposium on*.

Garcia, B. L., Mahey, P., and LeBlanc, L. J. (1998). Iterative improvement methods for a multiperiod network design. *European Journal of Operational Research* 110, 150–165.

Goulart, N., Noronha, T. F., de Souza, S. R., and Dias. (2011). Heurísticas para o problema de instalação de fibras em redes óticas. *Master's thesis*, CEFET MG, Belo Horizonte, MG.

Hufsky, F., Kuchenbecker, L., Jahn, K., Stoye, J. e Böcker, S. (2011). Swiftly computing center strings. *BMC Bioinformatics*, v. 12.

Hyytiä, E., and Virtamo, J. (1998). Wavelength assignment and routing in WDM networks. *In Fourteenth Nordic Teletraffic Seminar (Copenhagen)*, pp. 31–40.

Julstrom, Bryant A. (2009). A data-based coding of candidate strings in the closest string problem. *GECCO (Companion)'09*, p. 2053–2058.

Kelsey, T. e Kotthoff, L. (2010). The exact closest string problem as a constraint satisfaction problem. *Computing Research Repository*, v. abs/1005.0089.

Kennelly, P. e Krebs, E. (2008). Consensus sequences as substrate specificity determinants for protein kinases and protein phosphatases. *Journal of biological chemistry*, 266(24), pp. 15555-15558, August 1991.

Lanctot, J. K., Li, M., Ma, B.; Wang, S. e Zhang, L.. (2003). Distinguishing string selection problems. *Information and Computation*, v. 185, p. 41–55.

Li, M., Ma, B. e Wang, L. (1999). Finding similar regions in many strings. *Proceedings of the Thirty-First Annual ACM Symposium on Theory of computing (STOC'99),* p. 473–482, (1999).

Li, Z., Wang, L. e Zhang, K. (2006). Algorithmic approaches for genome rearrangement: a review, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 36, No. 5, pp. 636-648, 2006.

Liu, X., Fu, K. e Shao, R. (2004). Largest distance decreasing algorithm for the closest string problem. *Journal of Information & Computational Science*, v. 1(2), p. 287–292.

Liu, X., Holger, M., Hao, Z. eWu, G. (2008)b. A compounded genetic and simulated annealing algorithm for the closest string problem. *The 2nd International Conference on Bioinformatics and Biomedical Engineering*, p. 702–705.

Liu, X., Liu, S., Hao, Z. e Mauch, H. (2011). Exact algorithm and heuristic for the closest string problem. *Computers & OperationsResearch*, v. 38, p. 1513–1520.

McClure, M. A., Vasi, T. K. e Fitch, W. (1994). Comparative analysis of multiple protein-sequence alignment methods. *Molecular Biology and Evolution*, v. 1994, p. 571–592.

Meneses, C. N., Lu, Z., Pardalos, C. A. O. e Panos, M. (2004). Optimal solutions for the closest-string problem via integer programming. *INFORMS Journal of Computing*, v. 16, n. 4, p. 419–429.

Mousavi, S. R. e Esfahani, N. N. (2012). A grasp algorithm for the closest string problem using a probability-based heuristic. *Computers & Operations Research*, v. 39, n. 2, p. 238 – 248. ISSN 0305-0548. doi: DOI:10.1016/j.cor.2011.02.025. URL http://www.sciencedirect.com/science/article/pii/S030505481100089X.

Soleimani-damaneh, M. (2011). An optimization modelling for string selection in molecular biology using pareto optimality. *Applied Mathematical Modelling*, v. 35,n. 8, p. 3887–3892.

Spears, W., and deJong, K. (1991). On the virtues of parameterized uniform crossover. *In Proceedings of the Fourth International Conference on Genetic Algorithms (San Mateo)*, pp. 230–236.

Ruiz, R. e Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research* 177:2033–2049.

Yamamoto, K. (2004). Arredondamento randômico e o problema da sequência mais próxima. *Dissertação de Mestrado*, Universidade Federal Fluminense, RJ, (2004).