



METAHEURÍSTICA COLÔNIA DE FORMIGAS APLICADA AO PROBLEMA DO CAIXEIRO VIAJANTE

Fábio Ribeiro Cerqueira (orientador) e Gildásio Lecchi Cravo

Departamento de Ciência da Computação e Informática
Faculdade de Aracruz - UNIARACRUZ
{fabio, lecchi}@fsjb.edu.br

RESUMO

Este trabalho reporta os resultados da metaheurística colônia de formigas moldada para aplicação ao problema do caixeiro viajante (PCV). Em trabalho anterior, o PCV foi abordado pela metaheurística algoritmos genéticos e, neste texto, são comparados os resultados de ambas as técnicas. A motivação para a resolução do PCV partiu de um conhecido problema da área de bioinformática, denominado mapeamento físico de DNA.

Palavras-chave: Problema do caixeiro viajante. Colônia de formigas. Algoritmos genéticos. Mapeamento físico de DNA.

ABSTRACT

This paper describes results about the ant colony meta-heuristic applied to the traveling salesman problem (TSP). In a previous work we used the genetic algorithms approach to solve the TSP, and in this text the results from both techniques are compared. The motivation for TSP solving arises from a well known bioinformatics problem called Physical mapping of DNA.

Keywords: Traveling salesman problem. Ant colony. Genetic Algorithms. Physical mapping of DNA.

INTRODUÇÃO

Neste texto, apresenta-se a metaheurística colônia de formigas (CF) aplicada ao problema do caixeiro viajante (BONDY; MURTY, 1977; CORMEN, 1998). A principal motivação para a resolução dessa questão adveio do problema de mapeamento físico de DNA (PMP) que pode ser reduzido ao PCV, como será visto mais adiante.

Serão mostrados aqui a descrição do PMP, bem como sua redução ao problema do caixeiro viajante, a exposição dos conceitos e algoritmo envolvendo a técnica colônia de formigas e, finalmente, os resultados computacionais do trabalho.

MAPEAMENTO FÍSICO DE DNA COMO MOTIVAÇÃO PARA O TRABALHO

A motivação para este trabalho reside no problema do **mapeamento físico de DNA**. Para entender do que se trata, alguns conceitos se fazem necessários. Todos esses conceitos podem ser encontrados em Setúbal e Meidanis (1997), Cerqueira (2000), Cerqueira (2004) e Waterman (1995).

Em um projeto genoma, a primeira pesquisa realizada se chama **seqüenciamento**, que é voltada para a descoberta da seqüência de bases (Adenina-A, Citosina-C, Guanina-G, Timina-T) que compõe o DNA do ser vivo em análise.

É sabido que os genomas são extremamente grandes, ou seja, possuem um número alto de bases componentes. As tecnologias atuais não permitem que os laboratórios seqüenciem a molécula inteira, pois as máquinas para esse fim têm um limite de até mil bases para o DNA em questão.

Considerando o exposto, o que se faz é a quebra do DNA em pequenos pedaços de tamanho até mil para que esses possam ter suas bases decifradas pelas máquinas seqüenciadoras e sejam remontados posteriormente na ordem correta, gerando a seqüência final (**consenso**) da molécula de origem de onde provieram. Essa remontagem é o problema conhecido como **montagem de fragmentos de DNA**.

O que se faz na prática é uma primeira quebra do DNA em pedaços menores, mas ainda grandes, denominados **clones**. Cada pedaço gerado da primeira quebra é

novamente quebrado em fragmentos de mil bases, que são tratados na fase de montagem de fragmentos.

Uma vez tendo os consensos dos clones (resultado da fase de montagem de fragmentos), basta remontá-los em sua ordem original para que, juntos, dêem o consenso da molécula de DNA-alvo. Se fosse sabido de que região, na molécula original, cada clone vem, essa ordenação seria fácil. O mapeamento de cada clone, em sua respectiva região de origem no DNA-alvo, é conhecido como o problema do mapeamento físico de DNA.

Uma importante técnica no PMP é o **mapeamento físico por hibridização**. Nesse caso, identifica-se a ocorrência de pequeninos trechos de DNA na molécula-alvo. Esses trechos se chamam **sondas**. Uma vez definidas essas sondas, verifica-se a sua ocorrência também nos clones. Essa ocorrência é chamada de **hibridização**. O resultado disso é que, se houver hibridização de uma sonda y nos clones A e B, provavelmente estes se sobrepõem e, portanto, devem vir de regiões subseqüentes na molécula original.

Em todo esse processo explicitado até agora, é comum o acontecimento de uma série de problemas que dificultam o mapeamento. No que se refere ao PMP, pode-se destacar a situação de as sondas não hibridizarem onde deveriam ou hibridizarem onde não deveriam, criando problemas chamados de **falsos negativos** e **falsos positivos**, respectivamente. Há, ainda, um fenômeno que ocorre no processo de quebra do DNA, em que algumas partes que foram quebradas se juntam e formam o que se chama de **clone quimérico**, que será formado por regiões não contíguas do DNA original. Será mostrado, mais à frente, como tratar esses problemas.

MODELOS E CONSIDERAÇÕES PARA A SOLUÇÃO DO PMP

Aqui o PMP foi abstraído como o problema: encontrar “uns” consecutivos em uma matriz binária, como visto no quadro a seguir:

Clones x Sondas	S1	S2	S3	S4	S5	S6	S7	S8	S9
C1	1	1	0	1	1	0	1	0	1
C2	0	1	1	1	1	1	1	1	1
C3	0	1	0	1	1	1	1	0	1
C4	0	0	1	0	0	0	0	1	0
C5	0	0	1	0	0	1	0	0	0
C6	0	0	0	1	0	0	1	0	0
C7	0	1	0	0	0	0	1	0	0
C8	0	0	0	1	1	0	0	0	1

(a)

Clones x Sondas	S1	S2	S7	S4	S9	S5	S6	S3	S8
C1	1	1	1	1	1	1	0	0	0
C2	0	1	1	1	1	1	1	1	1
C3	0	1	1	1	1	1	1	0	0
C4	0	0	0	0	0	0	0	1	1
C5	0	0	0	0	0	0	1	1	0
C6	0	0	1	1	0	0	0	0	0
C7	0	1	1	0	0	0	0	0	0
C8	0	0	0	1	1	1	0	0	0

(b)

Quadro 1. Em (a): uma matriz M com a C1P, que indica quais sondas hibridizam quais clones; em (b): a permutação que gerou os “uns” consecutivos

A PROPRIEDADE DOS “UNS” CONSECUTIVOS (CONSECUTIVE ONES PROPERTY - C1P)

Uma matriz binária M , da forma mostrada na Quadro 1(a), tem a **propriedade dos “uns” consecutivos** (TELES, 1997), se for possível permutar as colunas de tal modo que todas as linhas tenham seus “uns” de forma consecutiva.

Se M for tal que as linhas representem clones e as colunas representem sondas, a entrada $M_{i,j}$ será 1, se o clone i foi hibridizado pela sonda j , e será 0, caso contrário. Além disso, a permutação das colunas para gerar os “uns” consecutivos é equivalente a resolver a questão das sobreposições entre os clones, ou seja, o PMP.

Na Quadro 1(b), é mostrada a matriz com os “uns” consecutivos, gerada a partir da matriz em 1(a), e a Figura 1 mostra como interpretar essa matriz resultante em termos do mapeamento físico, clarificando que tal permutação realmente é o que se procura.

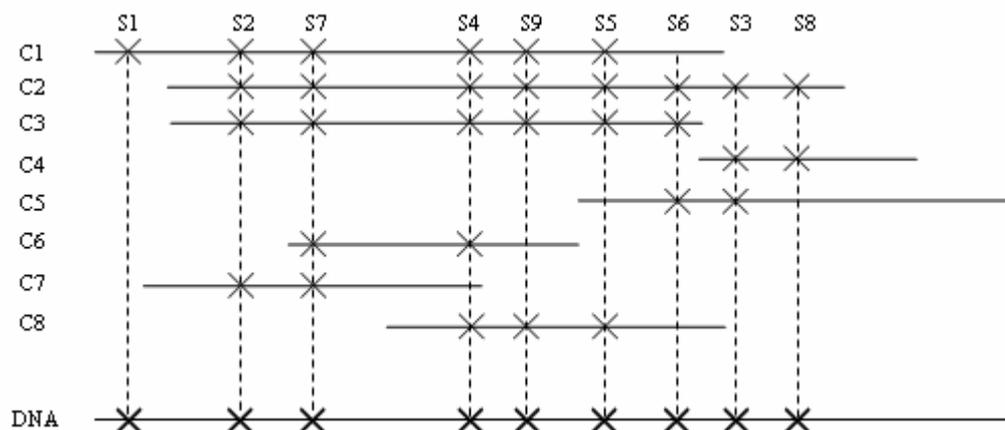


Figura 1. Alinhamento entre os clones, obtido pela matriz permutada

PMP SOBRE A PERSPECTIVA DO PROBLEMA DO CAIXEIRO VIAJANTE

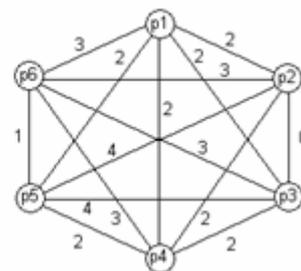
O grande problema de tudo que foi visto até o momento é que, na prática, os entraves citados, tais como: fragmentos quiméricos, falsos negativos e falsos positivos, fazem com que a matriz não tenha a C1P e, portanto, deve-se partir para uma permutação aproximada.

Suponha que M tenha a C1P e não esteja sujeita aos problemas citados. Seja M após a permutação. Em todas as linhas, seus 1s estarão consecutivos. No entanto, se uma dada linha corresponde a um clone quimérico, em que dois trechos tenham sido acidentalmente unidos no processo de quebra, então serão visualizados dois blocos de 1s separados por algum número de 0s. Esse bloco consecutivo de 0s, limitado por 1s, será chamado de **buraco**. Se, em uma outra linha, houver um falso negativo, o correspondente 0 separará dois blocos de 1s, criando outro buraco. Finalmente, um falso positivo pode quebrar um bloco de 0s em dois, possibilitando a criação de um outro buraco. Sendo assim, fica claro que há uma estreita relação entre erros e buracos na matriz. Uma interessante abordagem, então, poderia ser encontrar uma permutação que minimize o número de buracos na matriz. Tal tarefa é NP-Difícil (SETÚBAL; MEIDANIS, 1997). Dessa forma, propostas para a construção de heurísticas são obrigatórias.

Seja um grafo completo G não direcionado, com pesos, onde os vértices correspondam às colunas de M e o peso de cada aresta de G é dado pelo número de linhas onde as colunas correspondentes se diferem. No Quadro 2, é mostrado um exemplo.

Matriz M

	P1	P2	P3	P4	P5	P6
C1	1	1	1	0	0	0
C2	0	1	1	1	0	0
C3	1	0	0	1	1	0
C4	1	1	1	1	0	0



Grafo G

Quadro 2. Uma matriz Clones x Sondas e o grafo correspondente

Agora, deve-se notar que o ciclo hamiltoniano de peso mínimo em G (problema do caixeiro viajante) corresponde ao número mínimo de buracos em M (SETÚBAL; MEIDANIS, 1997). Para enxergar isso, note que, dada uma permutação das colunas, um buraco em uma linha significa que, em um certo ponto, há uma transição de 1 para 0 e mais tarde de 0 para 1. Então, para cada buraco, há duas transições e cada buraco contribui exatamente com dois para o peso do ciclo correspondente à dada permutação.

A conclusão é que basta, então, propor alguma heurística para o PCV.

COLÔNIA DE FORMIGAS E O PCV

PRINCÍPIO

Simulação do comportamento de um conjunto de agentes que cooperam entre si para resolver um problema de otimização por meio de comunicações bem simples.

Formigas são capazes de encontrar seu caminho (do ninho para uma fonte de comida e o caminho de volta ou vencendo um obstáculo) com relativa facilidade. A Entomologia tem resultados que levam a crer que, em muitos casos, essa capacidade é o resultado da interação via comunicação química entre formigas (mediante uma substância chamada feromônio) e um fenômeno causado pela presença de muitas formigas (RIBEIRO, 1996; DORIGO; GAMBARDELLA, 1997).

EXEMPLO

Para exemplificar o uso deste princípio, como método de resolução de problemas ou uma metaheurística, será utilizado o PCV.

Cada formiga age da seguinte forma:

- a) a formiga escolhe em cada iteração, usando uma regra de probabilidade, uma cidade para visitar dentre as que não foram ainda visitadas;
- b) a probabilidade de escolher a aresta (i,j) é diretamente proporcional à quantidade $t_{i,j}$ de feromônio na aresta (i,j) e inversamente proporcional ao tamanho $c_{i,j}$ da aresta;
- c) a formiga tem a capacidade de se lembrar de cada cidade já visitada;

- d) a formiga, após um ciclo ter sido completado, deixará uma trilha de feromônio Δt_{ij} (uma certa quantidade de feromônio) em cada aresta (i,j) do ciclo ($\Delta t_{ij} = 0$ para arestas que não estiverem no ciclo);
- e) a evaporação de feromônio em cada aresta (i,j) também é observada, considerando um fator b de evaporação, com $b < 1$: $t_{ij} \leftarrow b \cdot t_{ij} + \Delta t_{ij}$.

Já que os valores iniciais de t_{ij} são iguais para todas as arestas, inicialmente as formigas ficam livres para se moverem quase que randomicamente.

Se agora considerarmos M formigas se movendo simultaneamente no grafo, t_{ij} é a maneira pela qual as formigas se comunicam entre si: as formigas escolhem arestas com alta probabilidade, com grandes quantidades de feromônio. Portanto, quando as formigas deixam rastro de feromônio em um conjunto de arestas, estas ficam mais atrativas para as outras formigas (RIBEIRO, 1996).

Algoritmo Colônia de Formigas Para o PCV

O algoritmo colônia de formigas pode ser colocado como a superposição concorrente de M procedimentos de uma única formiga:

Passo 1. Sejam Q e t_0 constantes e seja $c(S^*) \leftarrow \infty$.

Passo 2. Faça $t_{ij} \leftarrow t_0$ para toda aresta (i,j) .

Passo 3. Faça $\Delta t_{ij} \leftarrow 0$

Passo 4. Para cada formiga $k = 1, 2, \dots, M$

- a) Selecione a cidade inicial da qual a formiga k partirá.
- b) Obtenha um ciclo T_k para cada formiga k de acordo com o procedimento seguido pela formiga, individualmente.
- c) Seja L_k o tamanho do ciclo T_k .
- d) Se $L_k < c(S^*)$ então faça $S^* \leftarrow T_k$
- e) Compute a quantidade de rastro de feromônio deixado pela formiga k , pela fórmula: $\Delta t_{ij}(k) = Q / L_k$, se a aresta (i,j) pertencer ao ciclo T_k . Caso não pertença: $\Delta t_{ij} = 0$.
- f) Faça $\Delta t_{ij} \leftarrow \Delta t_{ij} + \Delta t_{ij}(k)$

Passo 5. Faça $t_{ij} \leftarrow b \cdot t_{ij} + \Delta t_{ij}$.

Passo 6. Se o melhor ciclo S^* não se alterou com as últimas n_{\max} iterações, então pare, com S^* sendo a melhor solução encontrada. Caso contrário, volte ao Passo 3.

Sendo:

Q é uma constante que define a quantidade total de feromônio a ser adicionada para uma solução;

t_0 feromônio inicial;

S^* melhor solução encontrada;

$c(S^*)$ peso de S^* ;

T_k ciclo encontrado pela formiga k ;

L_k peso do ciclo T_k , ou seja, somatório dos pesos das arestas;

b fator de evaporação do feromônio;

$t_{i,j}$ feromônio de cada aresta.

Se for considerado o efeito da presença simultânea de muitas formigas, então cada uma contribui com uma parte para a distribuição de trilha de feromônio. O bom conjunto de arestas será utilizado por muitas formigas e, portanto, receberá uma quantidade grande de feromônio.

RESULTADOS COMPUTACIONAIS

Implementou-se o algoritmo anterior, utilizando-se a linguagem C++. Os testes a seguir foram feitos em um computador PENTIUM 3 550MHZ, com 128MB de RAM. Utilizou-se uma base de problemas do tipo PCV (TSPLIB), onde há várias instâncias e suas soluções ótimas, pois não foram encontradas instâncias para o problema de mapeamento físico de DNA. Serão mostrados, nesta seção, os resultados dessa implementação, comparando-os com aqueles obtidos por Cerqueira e Renarde (2004), nos quais foram utilizados algoritmos genéticos (AG) para a resolução do PCV (MICHALEWICZ, 1996; RAMOS; SALVADO, 2001; REEVES, 1993). Nos resultados a seguir, foram utilizadas as instâncias: eil51, u159, u574.

RESULTADO COLÔNIA DE FORMIGAS

A seguir, encontram-se as tabelas com os resultados da implementação do AG e do CF.

Tabela 1. Resultados obtidos com o Algoritmo Genético

Resultados do Algoritmo Genético				
Instâncias	Valor da Sol. Ótima	Valor Algoritmo Guloso	Eficiência	Tempo
eil51	426	505,800	84,223%	-
u159	42080	48586,700	86,608%	-
u574	36905	45141,500	81,754%	-
1000 Iterações				
Instâncias	Valor da Sol. Ótima	Valor encontrado	Eficiência	Tempo
eil51	426	444,900	95,752%	5s
u159	42080	43240,990	97,315%	54s
u574	36905	43324,830	85,182%	15m4s
2000 Iterações				
Instâncias	Valor da Sol. Ótima	Valor encontrado	Eficiência	Tempo
eil51	426	439,290	96,975%	11s
u159	42080	44815,820	93,895%	1m48s
u574	36905	43132,700	85,562%	29m31s
5000 Iterações				
Instâncias	Valor da Sol. Ótima	Valor encontrado	Eficiência	Tempo
eil51	426	434,910	97,951%	28s
u159	42080	42583,370	98,818%	4m28s
u574	36905	43101,550	85,623%	1h9m24s

Fonte: Cerqueira e Stelzer (2004)

Tabela 2. Resultados obtidos com a metaheurística Colônia de Formigas

Resultados da Metaheurística Colônia de Formigas					
Instâncias	Valor da Sol. Ótima	1000 Iterações			
		Valor encontrado	Eficiência	Tempo	Número de Formigas
eil51	426	437,159	97,447%	2s	15
u159	42080	42605,000	98,768%	1m6s	25
u574	36905	39274,800	93,966%	1h3m41s	400
2000 Iterações					
Instâncias	Valor da Sol. Ótima	Valor encontrado	Eficiência	Tempo	Número de Formigas
eil51	426	436,459	97,604%	4s	15
u159	42080	42562,000	98,868%	2m13s	25
u574	36905	38813,700	95,082%	2h7m8s	400
5000 Iterações					
Instâncias	Valor da Sol. Ótima	Valor encontrado	Eficiência	Tempo	Número de Formigas
eil51	426	434,088	98,137%	12s	15
u159	42080	42482,200	99,053%	5m33s	25
u574	36905	38732,800	95,281%	5h25m50s	400

A seguir, temos os gráficos dos resultados (Figuras 2 e 3) para a instância u574 do CF e AG.

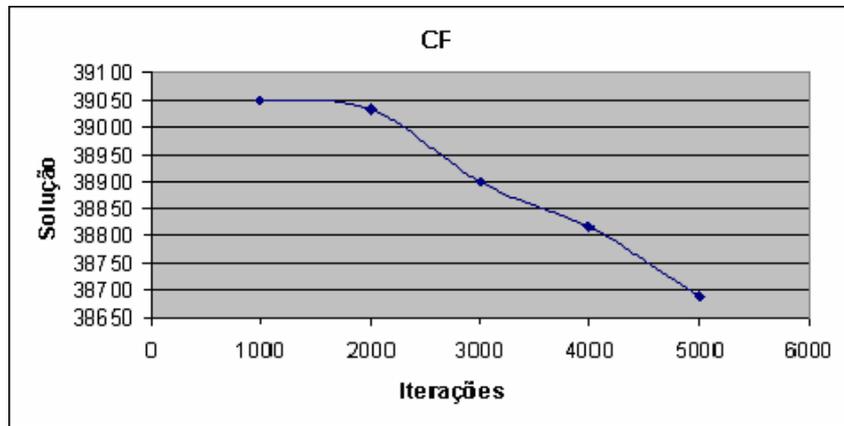


Figura 2. Curva de melhoria da solução para a instância u574 no CF

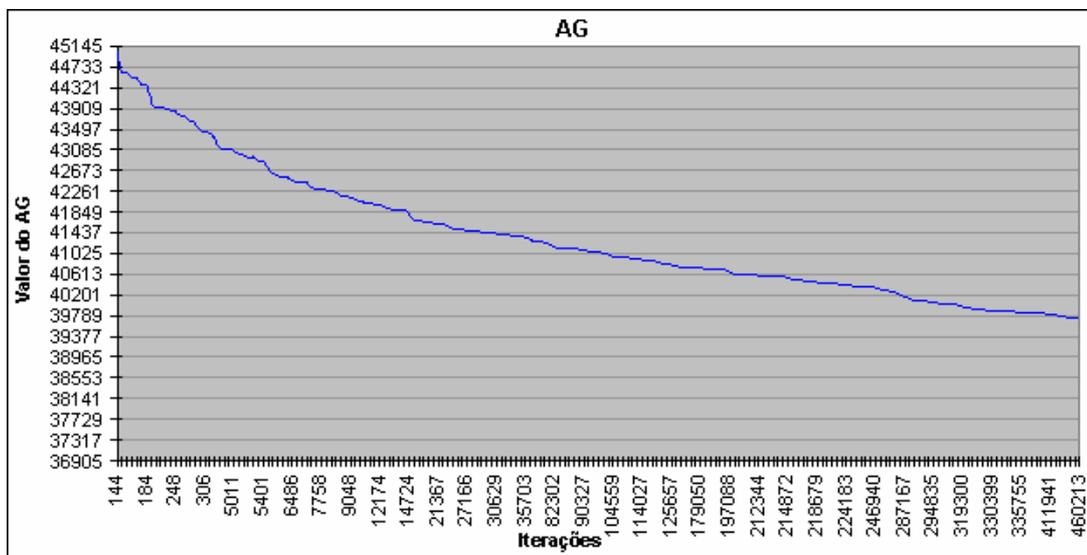


Figura 3. Curva de melhoria da solução para a instância u574 no AG

Fonte: Cerqueira e Stelzer (2004)

CONCLUSÃO

O resultado do trabalho foi satisfatório. Com um algoritmo polinomial, foram alcançados resultados bem melhores do que um simples algoritmo guloso e que vão se aproximando do valor ótimo. Como os testes foram feitos em um PC doméstico, não foi possível realizar testes com um número de iterações mais elevado, mas tudo indica que, aumentando consideravelmente o número de iterações, as soluções chegarão bem próximas ao ótimo.

Por outro lado, os resultados do AG e CF foram muito próximos, indicando que ambos são equivalentemente eficientes.

Quando foram comparadas as Figuras 2 e 3, notou-se que o CF converge mais rápido do que o AG.

Como trabalho futuro, sugere-se a implementação de outras metaheurísticas como Busca Tabu e GENIUS, além de fazer ajustes nos já implementados, no sentido de escapar de ótimos locais e, com isso, obter uma maior riqueza de comparações de resultados.

REFERÊNCIAS

- 1 BONDY, J. A.; MURTY, U. S. R. **Graph theory with applications**. Ontario: The Macmillan Press LTD, 1977.
- 2 CERQUEIRA, F. R.; STELZER, R. **Genetic algorithms applied to DNA physical mapping problem**. IV Escola Regional de Informática da Sociedade Brasileira de Computação RJ/ES, Vitória-ES, 2004.
- 3 CERQUEIRA, F. **Montagem de fragmentos de DNA**. 2000. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Campinas, Campinas - SP, 2000.
- 4 CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. **Introduction to algorithms**. New Jersey: Prantice Hall, 1998.
- 5 DORIGO M.; GAMBARELLA L. M. **Ant colonies for the traveling salesman problem**. Bruxelas – Bélgica: BioSytems, 1997.
- 6 MICHALEWICZ, Z. **Genetic algorithms + data structures = evolution programs**. 3. ed. Charlotte: Springer, 1996.
- 7 RAMOS, N.; SALVADO, S. **Jogo de xadrez por algoritmos genéticos evolutivos**. 2001. Monografia (Graduação em Ciência da Computação), Instituto Superior Técnico, Lisboa - Portugal, 2001.

- 8 REEVES, C. R. **Modern heuristic techniques for combinatorial problems.** New York: Blackwell Scientific Publications, 1993.
- 9 RIBEIRO, C. C. **Metaheuristics and applications.** Estoril – Portugal: Advanced School on Artificial Intelligence, 1996.
- 10 SETÚBAL, J.; MEIDANIS, J. **Introduction to computational molecular biology.** Pacific Grove: PWS, 1997.
- 11 TELES, G. P. **A propriedade dos “uns” consecutivos e aplicações.** Campinas-SP: Universidade Estadual de Campinas, 1997.
- 12 WATERMAN, M. S. **Introduction to computational biology.** Boca Raton: Chapman and Hall, 1995.