4
Resultados experimentais

Neste capítulo são apresentados os resultados de experimentos realizados com os algoritmos propostos neste trabalho. São dois os objetivos principais destes experimentos. O primeiro objetivo é realizar o ajuste dos parâmetros utilizados pelos algoritmos desenvolvidos. O segundo objetivo é avaliar, na prática, o desempenho das heurísticas propostas em relação aos melhores algoritmos da literatura. As condições nas quais foram realizados os experimentos são descritas na Seção 4.1. Na Seção 4.2 são apresentadas as instâncias utilizadas nos testes. Duas medidas de qualidade de uma solução são discutidas na Seção 4.3 e utilizadas para avaliar a qualidade dos algoritmos. Na Seção 4.4 são discutidos os testes realizados com o intuito de ajustar os parâmetros das heurísticas desenvolvidas. Finalmente, na Seção 4.5 os resultados dos melhores algoritmos da literatura são comparados com os resultados das heurísticas propostas neste trabalho.

4.1
Condições dos experimentos

Os resultados apresentados neste capítulo para as novas heurísticas foram gerados a partir de experimentos realizados em um computador equipado com um processador Pentium de 2.4 GHz, 512 MB de memória principal e sistema operacional Mandrake Linux 10.0 com kernel 2.6.3. As heurísticas foram implementadas na linguagem ANSI C++ e compiladas com o compilador GNU versão 3.3.2.

4.2
Instâncias

Foram geradas diversas instâncias para realizar os testes com os algoritmos. As instâncias são divididas em três grupos: A, B e R. Os grupos

A e B foram gerados a partir de seqüências reais de DNA e o grupo R foi gerado a partir de seqüências aleatórias de DNA.

Instâncias reais

Os grupos de instâncias A e B foram propostos em [6] e são compostos por instâncias geradas a partir de seqüências de DNA codificadoras de proteínas do organismo humano. Estas seqüências foram obtidas do GenBank [2], um repositório de seqüências de DNA muito utilizado por pesquisadores do mundo todo. Os códigos de acesso destas seqüências no GenBank são listados no Apêndice A.

As instâncias do grupo A foram geradas a partir de 40 seqüências do GenBank. Cada instância do grupo A foi gerada a partir de um prefixo de uma destas seqüências. Para cada seqüência, foram geradas cinco instâncias a partir dos seus prefixos de tamanho 109, 209, 309, 409 e 509. Para cada prefixo, uma instância foi gerada simulando-se o experimento de hibridação deste prefixo com um arranjo $C(10)$. Esta simulação gerou o espectro ideal para este prefixo (todas as subseqüências de tamanho 10 do prefixo). Portanto, para cada seqüência foram gerados cinco espectros com 100, 200, 300, 400 e 500 provas. Em seguida, foram gerados alguns erros no espectro. Falsos negativos foram gerados retirando-se, aleatoriamente, 20% das provas do espectro ideal. Falsos positivos foram gerados inserindo-se 20% de provas, geradas aleatoriamente. Dos espectros de tamanho 100, por exemplo, foram retiradas 20 provas e inseridas 20 novas provas. Os erros foram gerados de tal forma que a prova inicial não foi retirada do espectro e as provas inseridas são diferentes de todas aquelas que compunham o espectro ideal. As seqüências originais não possuem repetições de subseqüências de tamanho 10 e, portanto, os espectros resultantes não contêm falsos negativos devidos a repetições de provas. Como são 40 seqüências e cada uma gerou cinco instâncias, o grupo A é composto por 200 instâncias (40 de cada tamanho).

O grupo de instâncias B foi gerado de forma a obter espectros com falsos negativos devidos a repetições de provas nas seqüências alvo. Estas instâncias foram geradas a partir de outras 78 seqüências obtidas do GenBank e seus códigos de acesso também estão listados no Apêndice A. Foram geradas 78 instâncias, uma para cada seqüência, simulando-se experimentos com um arranjo $C(7)$ de tal forma que os espectros resultantes tenham 500 provas cada. Como estas seqüências possuem repetições de provas, o comprimento do prefixo utilizado para gerar cada instância varia,

assim como a quantidade de falsos negativos (que varia entre 10 e 20). Não foram inseridos falsos positivos nestas instâncias.

Instâncias aleatórias

O grupo de instâncias R foi gerado de maneira similar ao grupo A. Entretanto, o grupo R foi gerado a partir de 100 seqüências aleatórias. A partir de cada seqüência, foram geradas 10 instâncias simulando-se o experimento de hibridação dos seus prefixos de tamanho 100, 200, ..., 1000 com o arranho $C(9)$. Desta forma, obteve-se espectros ideais com, no máximo, 92, 192, ..., 992 provas, respectivamente. Os espectros resultantes podem conter algumas provas a menos devido a repetições de provas na seqüência original. Neste grupo, também foram gerados 20% de falsos negativos e 20% de falsos positivos. O procedimento de geração de erros é idêntico àquele utilizado nas instâncias do grupo A.

4.3 Qualidade das soluções

Para realizar a comparação dos resultados de algoritmos diferentes é necessário definir a qualidade de uma solução. Desta forma, é possível determinar quando uma solução é melhor do que outra. O objetivo definido no PSBH é encontrar uma solução que utilize o maior número possível de provas do espectro. Portanto, pode-se definir uma medida de qualidade como sendo a quantidade de provas do espectro presentes na solução. Entretanto, alguns algoritmos podem utilizar uma formulação diferente para o problema de montagem e ter um objetivo diferente daquele utilizado neste trabalho. O algoritmo proposto em [13] é um exemplo disto. Portanto, utilizar a quantidade de provas para comparar soluções geradas por diferentes algoritmos pode ser injusto.

Similaridade

Uma idéia proposta em [6] para criar uma medida de qualidade para uma solução é verificar a semelhança entre a seqüência de DNA correspondente à solução e a seqüência de DNA alvo. Obviamente, esta abordagem só pode ser utilizada quando se conhece a seqüência alvo. Esta abordagem é baseada na técnica denominada *alinhamento de seqüências* que é vastamente utilizada em trabalhos na área de biologia molecular para

A T A G G -
A T - C G A

Figura 4.1: Alinhamento entre as seqüências ATAGG e ATCGA.

comparar pares de seqüência. Esta técnica é baseada na idéia de *distância de edição*. A distância de edição entre duas seqüências de símbolos é definida como a quantidade mínima de operações necessárias para transformar uma seqüência na outra, onde as operações permitidas são: inserção de um símbolo, remoção de um símbolo e substituição de um símbolo por outro. A distância de edição entre as seqüências ATAGG e ATCGA é igual a três, pois a primeira seqüência pode ser transformada na segunda através das seguintes operações: remoção do terceiro símbolo (A), substituição do quarto símbolo (G) por C, inserção do símbolo A no final da seqüência. O alinhamento entre duas seqüências é derivado do conceito de distância de edição. Um exemplo do alinhamento entre as seqüências do exemplo anterior é ilustrado na Figura 4.1. Comparando-se o alinhamento com a transformação da primeira seqüência na segunda tem-se que um "-" na segunda seqüência significa a remoção de um símbolo, um "-" na primeira seqüência significa a inserção de um símbolo e uma coluna com símbolos diferentes significa a substituição de um símbolo por outro. No alinhamento, uma coluna com símbolos iguais é denominada *igualdade*, uma coluna com símbolos diferentes é denominada *diferença* e uma coluna com "-" é denominada *espaço*. Pode-se associar custos a cada coluna do alinhamento e definir o *valor de alinhamento* entre duas seqüências como sendo a soma dos custos das colunas do alinhamento entre elas. O valor de alinhamento entre as seqüências π_1 e π_2 é denominado $\text{align}(\pi_1, \pi_2)$. Definindo-se o custo de uma igualdade como 1 e o custo de uma diferença ou de um espaço como -1, tem-se que o valor de alinhamento entre as seqüências do exemplo da Figura 4.1 é igual a zero. Utilizando-se estes custos, o valor de alinhamento varia no intervalo $[-n_{max}, n_{max}]$, onde n_{max} é o comprimento da maior seqüência. Um valor de alinhamento igual a n_{max} significa que as duas seqüências são idênticas. Um valor de alinhamento igual a $-n_{max}$ significa que as duas seqüências não têm símbolo algum em comum. O algoritmo para calcular o alinhamento entre duas seqüências é baseado em programação dinâmica e discutido detalhadamente em [31] (capítulo 9).

Utilizar o valor de alinhamento entre a seqüência correspondente à solução e a seqüência do DNA alvo como medida da qualidade de uma solução permite a comparação de soluções geradas por algoritmos utilizando diferentes formulações. Esta medida é mais intuitiva e muito popular na

comunidade que trabalha na área de biologia molecular. Entretanto, em alguns casos é interessante possibilitar a comparação de resultados entre soluções de instâncias diferentes. Para isto, define-se um valor normalizado a partir do valor de alinhamento. Este valor é denominado *similaridade* e, para uma dada solução a, é definido como:

$$\mathrm{sim}(a) = 100 \cdot \frac{\mathrm{align}(\pi(a), \pi^*) + n_{max}}{2 \cdot n_{max}},$$

onde $\pi(a)$ é a seqüência de DNA correspondente à solução a, π^* é a seqüência de DNA alvo e n_{max} é o comprimento da maior seqüência. Intuitivamente, o valor $\mathrm{sim}(a)$ indica a porcentagem de colunas do alinhamento entre a seqüência correspondente a a e a seqüência alvo que são igualdades. A similaridade entre as seqüências do exemplo da Figura 4.1, por exemplo, é igual a 50%, pois metade das colunas do alinhamento são igualdades.

Porcentagem de provas

O conceito de similaridade é muito importante na comparação de soluções geradas por um método qualquer. Entretanto, ao se avaliar as heurísticas desenvolvidas neste trabalho e compará-las entre si, é interessante observar os resultados em relação ao objetivo determinado pela formulação utilizada, ou seja, a quantidade de provas de uma solução. Isto é importante pois permite uma avaliação mais precisa da qualidade do algoritmo em alcançar o objetivo determinado. A similaridade pode causar problemas em relação a isto. Um exemplo disto ocorre quando uma instância possui outras soluções com a mesma quantidade de provas da solução que reconstrói a seqüência alvo. Este caso pode ocorrer devido ao problema de reconstrução múltipla discutido na Seção 2.5. Neste caso, o fato de um algoritmo encontrar uma solução com similaridade de 100% e outro encontrar outra solução com a mesma quantidade de provas mas com uma similaridade menor é puro acaso, ou seja, não significa que o primeiro algoritmo é melhor do que o segundo. O segundo algoritmo pode até ter encontrado a solução correta também, mas não há maneira de distinguir qual delas é a melhor porque, de acordo com a formulação utilizada, as duas têm a mesma qualidade. Por isto, a quantidade de provas de uma solução também será utilizada como medida de qualidade, principalmente na Seção 4.4 onde são realizadas comparações apenas entre as heurísticas desenvolvidas neste trabalho. Entretanto, também é interessante definir um valor normalizado para esta medida, permitindo a comparação de resultados para instâncias

diferentes. O valor normalizado definido aqui é denominado *porcentagem de provas* e, para uma dada solução a, é definido como:

$$\text{porc}(a) = \frac{100 \cdot |a|}{|a^*|},$$

onde a^* é a solução que reconstrói a seqüência alvo.

4.4 Ajuste de parâmetros

As heurísticas propostas neste trabalho dependem de parâmetros que influenciam diretamente o seu desempenho. Por isto, torna-se necessário a realização de alguns experimentos para verificar quais os melhores valores para estes parâmetros. Discute-se a seguir o ajuste dos parâmetros utilizados pelo algoritmo construtivo aleatorizado (α), pela estrutura de memória adaptativa (λ, q e d) e pelo método de construção de vocabulário (q', d' e s_{min}).

Algoritmo construtivo aleatorizado

Um aspecto muito importante do algoritmo construtivo proposto neste trabalho é a LRC. Esta lista é utilizada para intensificar as construções do algoritmo em uma região mais promissora do espaço de soluções, ao mesmo tempo que permite a construção de soluções diferentes a cada execução. O conteúdo da lista é influenciado diretamente pelo parâmetro α. Se $\alpha = 0$, a lista conterá apenas as provas com sobreposição máxima com a última prova inserida na solução, dentre aquelas ainda não utilizadas. Por outro lado, se $\alpha = 1$, a lista conterá todas as provas ainda não inseridas na solução e que possuam uma sobreposição não nula com a última prova inserida.

As heurísticas propostas neste trabalho utilizam o algoritmo construtivo aleatorizado para gerar soluções que serão posteriormente melhoradas por outro procedimento. Nestes casos, é desejável que o algoritmo construtivo gere soluções diversificadas, mas mantenha um certo nível de qualidade. Uma forma de avaliar a diversidade e a qualidade das soluções geradas por um algoritmo aleatorizado é criar um histograma. Este histograma é gerado a partir de várias execuções independentes do algoritmo, utilizando diferentes sementes. As soluções produzidas são agrupadas por qualidade. A distribuição das soluções de acordo com sua qualidade permite avaliar sua diversidade. Neste trabalho, esta avaliação foi realizada em relação às

instâncias do grupo R. Para cada instância deste grupo, o algoritmo construtivo aleatorizado foi executado 100 vezes com diferentes sementes. Um histograma foi construído de acordo com as similaridades das soluções correspondentes às instâncias de um mesmo tamanho. Nas Figuras 4.2 e 4.3 são apresentados os histogramas dos testes com as instâncias de tamanho 300 e 600, respectivamente. Em cada figura são apresentados seis histogramas para diferentes valores de α (0.0, 0.2, 0.4, 0.6, 0.8 e 1.0). De acordo com os histogramas, foram escolhidos os valores $\alpha = 0.2$ para as instâncias de tamanho 300 e $\alpha = 0.0$ para as instâncias de tamanho 600. Estes valores foram escolhidos porque seus histogramas apresentam um padrão sugerindo que foram construídas soluções bem diversificadas, mas ainda foram construídas muitas soluções boas. Na Tabela 4.1 são apresentados os valores escolhidos para diversos tamanhos de instância. Estes valores foram escolhidos da mesma forma descrita acima, ou seja, observando-se os histogramas dos testes para cada conjunto de instâncias de um determinado tamanho do grupo R. Os tamanhos nesta tabela dizem respeito ao tamanho das seqüências alvo. Para as instâncias do grupo A, que possuem tamanhos 109, 209, 309, 409 e 509, são utilizados os valores relativos aos tamanhos 100, 200, 300, 400 e 500, respectivamente. Para as instâncias do grupo B é utilizado o valor relativo ao tamanho 500.

Tamanho	α
100	0.5
200	0.3
300	0.2
400	0.1
500	0.1

Tamanho	α
600	0.0
700	0.0
800	0.0
900	0.0
1000	0.0

Tabela 4.1: Valores escolhidos para o parâmetro α de acordo com o tamanho da instância.

Memória adaptativa

A utilização da memória adaptativa envolve três parâmetros: λ, q e d. O parâmetro λ define o balanceamento entre as duas informações disponíveis durante as escolhas do algoritmo construtivo: a sobreposição entre as provas e a freqüência das arestas no conjunto elite. O valor deste parâmetro precisa ser ajustado durante a execução da heurística pois, nas primeiras iterações, as soluções na memória normalmente não contêm informações relevantes, ou seja, não são soluções muito boas. Por isto, o valor de λ deve ser grande

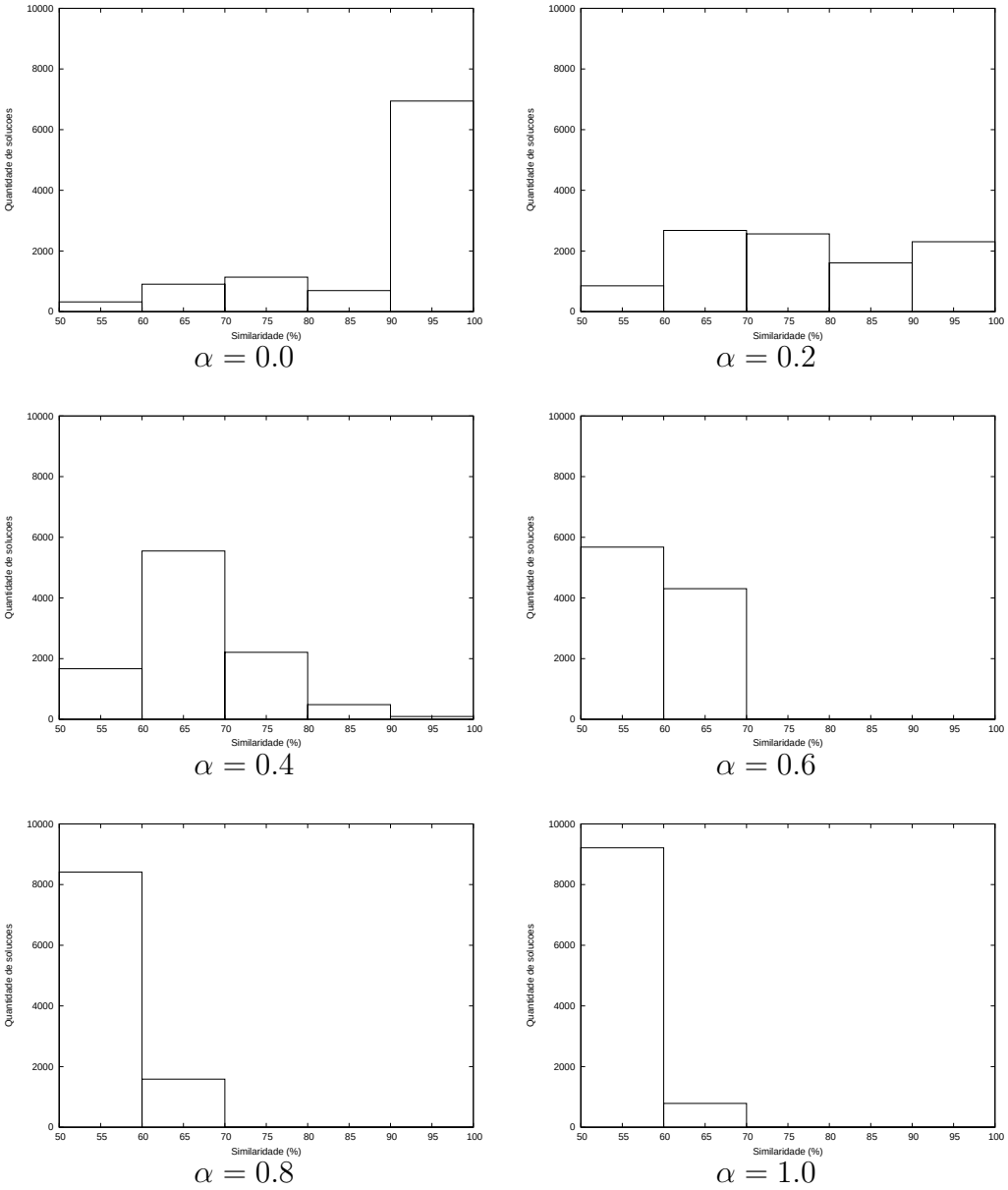

Figura 4.2: Histogramas da qualidade das soluções geradas pelo algoritmo construtivo aleatorizado utilizando diferentes valores de α. Os valores do eixo horizontal são relativos à similaridade média dos resultados de 10000 execuções do algoritmo (100 execuções para cada uma das 100 instâncias de tamanho 300 do grupo R).

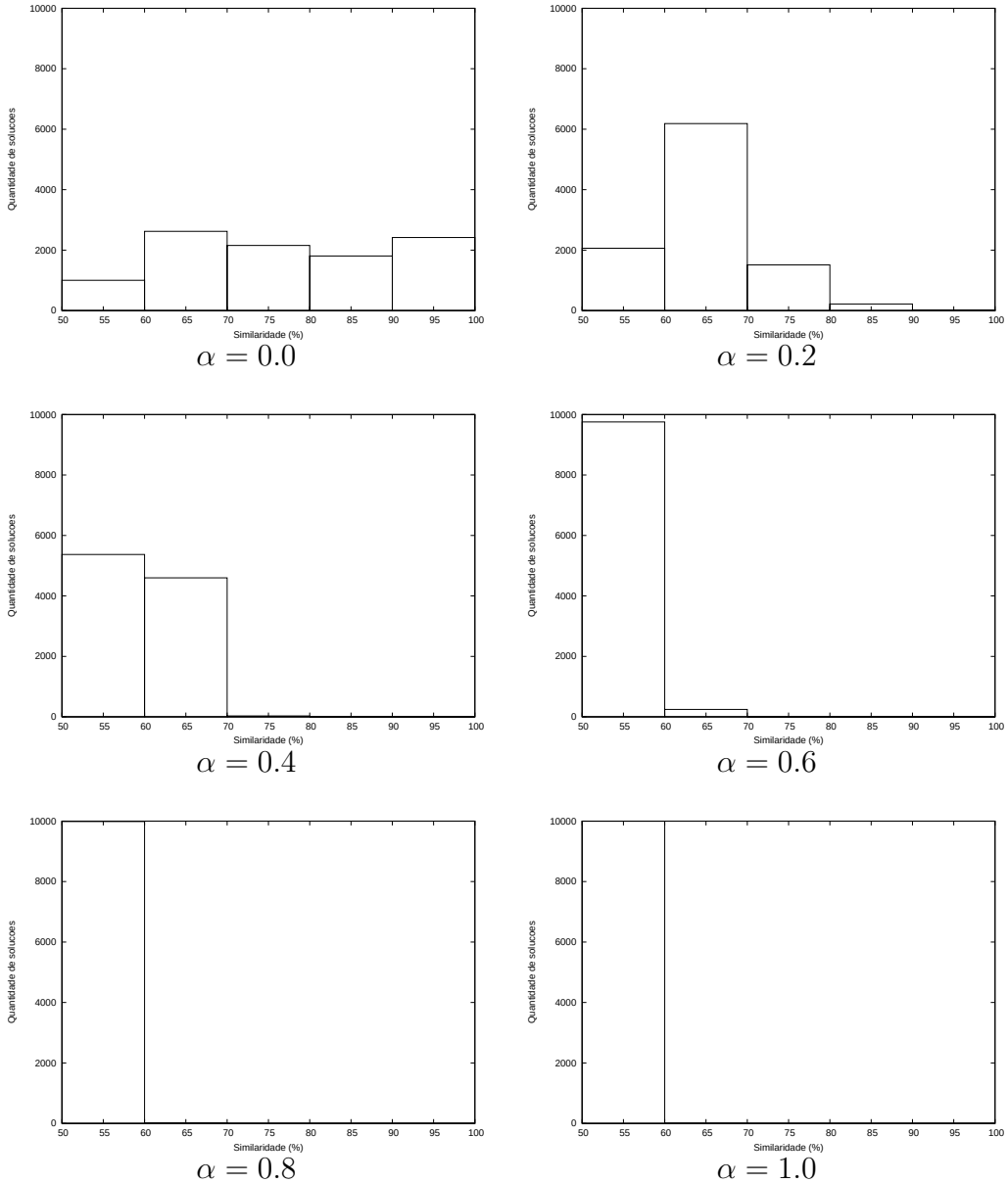

Figura 4.3: Histogramas da qualidade das soluções geradas pelo algoritmo construtivo aleatorizado utilizando diferentes valores de α. Os valores do eixo horizontal são relativos à similaridade média dos resultados de 10000 execuções do algoritmo (100 execuções para cada uma das 100 instâncias de tamanho 600 do grupo R).

no início do algoritmo, e diminuir a medida em que mais soluções são construídas.

Durante testes preliminares, realizados na tentativa de entender a influência do parâmetro λ no algoritmo, observou-se que é imprescindível o sincronismo entre este parâmetro e o parâmetro α do algoritmo construtivo. É importante iniciar a heurística `MP+Mem` com um valor relativamente baixo para α, restringindo a LRC às melhores opções possíveis. Desta forma, o algoritmo construtivo é capaz de gerar soluções boas já nas primeiras iterações, enriquecendo rapidamente a memória. A medida que o valor de λ é diminuído e as decisões do algoritmo construtivo passam a ser baseadas principalmente na memória, a LRC deve conter mais provas do que nas iterações anteriores para permitir que o algoritmo construtivo gere soluções diferentes daquelas construídas anteriormente. Isto faz com que o algoritmo construtivo selecione com mais freqüência as arestas que aparecem mais vezes na memória mas, em algumas situações, escolha uma aresta nunca escolhida anteriormente e, possivelmente, construa uma solução melhor. Este comportamento pode ser implementado aumentando-se o valor de α a medida que o valor de λ é diminuído.

Antes de discutir-se o ajuste do parâmetro λ é importante analisar a relação entre os valores de $v(u,s)$ e $i(u,s)$ na Equação 3-4, lembrando-se que u é a última prova inserida na solução sendo construída e s é uma prova da LRC. O valor de $v(u,s)$ foi normalizado e varia no intervalo $(0,1]$. Este valor não pode ser zero porque somente elementos com alguma sobreposição com a prova u podem entrar na LRC. Para a prova s da LRC que possui a maior sobreposição com a prova u, o valor de $v(u,s)$ é igual a um. O valor de $i(u,s)$ também foi normalizado, mas varia no intervalo $[0,q]$ pois é igual à soma do custo de todas as soluções de elite que utilizam a aresta (u,s) dividida pelo custo da melhor solução de elite. Portanto, se todas as soluções de elite possuem o mesmo custo e utilizam a aresta (u,s), o valor $i(u,s)$ será igual a q. No outro extremo, se nenhuma solução de elite utiliza a aresta (u,s), o valor $i(u,s)$ será igual a zero. Portanto, é razoável considerar que $\lambda = q$ implica em um equilíbrio entre os valores $v(u,s)$ e $i(u,s)$.

A estratégia proposta neste trabalho para ajustar os parâmetros λ e α durante a execução das heurísticas que utilizam a memória adaptativa (`MP+Mem` e `MP+Mem+CV`) é descrita a seguir. Todas as heurísticas propostas realizam $10n$ construções, ou seja, $n_{iter} = 10n$. Estas iterações são divididas em 20 *passos* (cada passo consiste de $n/2$ iterações). No primeiro passo, o valor de λ é mantido como $100q$, levando o algoritmo construtivo a considerar somente as informações sobre a sobreposição entre cada par de

	d				
q	0	2	$\|S\|/32$	$\|S\|/16$	$\|S\|/8$
$n/100$	98.94	98.95	98.88	98.80	98.46
$n/80$	98.97	**98.99**	98.93	98.80	98.42
$n/40$	98.91	98.93	98.87	98.67	98.28
$n/20$	98.79	98.78	98.77	98.58	98.07
$n/10$	98.70	98.73	98.68	98.46	97.92

Tabela 4.2: Resultados para várias combinações dos parâmetros q e d do conjunto de soluções de elite utilizado pela memória adaptativa na heurística MP+Mem. Os valores na tabela são as médias da porcentagem de provas nas soluções construídas pelo algoritmo para as instâncias de tamanho 1000 do grupo R.

provas. No segundo passo, faz-se $\lambda = 10q$. A cada passo seguinte, o valor de λ é diminuído de q até alcançar zero e fica assim até o final do algoritmo. O valor de α é iniciado de acordo com os valores descritos na Tabela 4.1 e a cada cinco passos seu valor é aumentado em 0.1, até alcançar 1.0 e permanecer assim até o final do algoritmo.

Para escolher os valores para os parâmetros q e d foram realizados testes com a heurística `MP+Mem` utilizando diversas combinações de valores para estes parâmetros. Foram utilizados cinco valores para o parâmetro q: $n/100$, $n/80$, $n/40$, $n/20$ e $n/10$. Para o parâmetro d foram experimentados outros cinco valores: 0, 2, $|S|/32$, $|S|/16$ e $|S|/8$. O parâmetro d é expresso em função de $|S|$, pois está diretamente ligado a este valor. O parâmetro d indica a diferença mínima permitida entre as soluções de elite. Devido ao modo como esta diferença é calculada, seu valor está sempre no intervalo $[0, |S|]$. Os resultados dos testes são apresentados na Tabela 4.2. Cada valor nesta tabela é uma média da porcentagem de provas das soluções retornadas pela heurística `MP+Mem`, utilizando os parâmetros correspondentes, para as 100 instâncias de tamanho 1000 do grupo R. O valor em destaque na tabela diz respeito aos valores que obtiveram o melhor resultado. Os valores escolhidos foram $q = n/80$ e $d = 2$.

Construção de vocabulário

A implementação do método de construção de vocabulário proposta neste trabalho envolve três parâmetros: q', d' e s_{min}. Os parâmetros q' e d' indicam, respectivamente, o tamanho do conjunto de soluções de elite e a distância mínima permitida entre suas soluções. O parâmetro s_{min} é referente ao tamanho mínimo, exigido pelo algoritmo `BuscaPalavras`, para classificar um vetor de adjacência como uma palavra. Para avaliar

	s_{min}			
d'	$\|S\|/8$	$\|S\|/4$	$\|S\|/2$	$\|S\|3/4$
0	97.78	97.78	97.79	97.81
2	97.77	97.78	97.80	97.81
$\|S\|/32$	97.78	97.78	97.78	97.81
$\|S\|/16$	97.79	97.79	97.78	97.79
$\|S\|/8$	97.77	97.81	**97.97**	97.80

Tabela 4.3: Resultados para várias combinações dos parâmetros d' e s_{min} utilizados pelo método de construção de vocabulário na heurística `Mem+CV`, utilizando $q' = n/8$. Os valores na tabela são as médias da porcentagem de provas nas soluções construídas pelo algoritmo para as instâncias de tamanho 1000 do grupo R.

os melhores valores para estes parâmetros, foram realizados testes com a heurística `MP+CV` utilizando várias combinações de valores. Foram testados os seguintes valores para o parâmetro q': $n/8$, $n/4$, $n/2$, n e $2n$. Para o parâmetro d' foram experimentados os valores 0, 2, $|S|/32$, $|S|/16$ e $|S|/8$. O parâmetro s_{min} foi testado com os valores $|S|/8$, $|S|/4$ e $|S|3/4$. Os melhores resultados foram obtidos utilizando $q' = n/8$. Os resultados dos testes utilizando este valor para o parâmetro q' e todas as combinações possíveis dos valores citados acima para os parâmetros d' e s_{min} são apresentados na Tabela 4.3. Os valores dizem respeito às médias de porcentagem de provas das soluções encontradas pela heurística `MP+CV` para as instâncias de tamanho 1000 do grupo R. Estes resultados indicam os valores $d' = |S|/8$ e $s_{min} = |S|/2$ como os melhores para estes parâmetros.

4.5 Comparação de resultados

Com o intuito de avaliar o desempenho das heurísticas propostas neste trabalho, foram realizados testes comparando os resultados destas heurísticas entre si e com os resultados dos três melhores algoritmos disponíveis na literatura. Comparações entre os resultados das quatro heurísticas propostas aqui (`MP`, `MP+Mem`, `MP+CV` e `MP+Mem+CV`) visam analisar o impacto dos dois métodos propostos, a memória adaptativa e a construção de vocabulário, e identificar qual das heurísticas é a mais adequada para o PSBH. As comparações com os três melhores algoritmos propostos anteriormente tem o objetivo de avaliar o desempenho da heurística escolhida.

Avaliação das heurísticas desenvolvidas

A melhor forma de comparar as quatro heurísticas propostas aqui é analisar os resultados destas heurísticas para as instâncias do grupo R. Utilizar as instâncias aleatórias, ao invés daquelas dos grupos A e B, implica em uma análise mais ampla e conclusiva. Instâncias geradas a partir de seqüências de DNA de uma única espécie, como é o caso dos grupos A e B, possivelmente possuem uma estrutura singular. Este problema é menos provável de ocorrer em instâncias geradas a partir de seqüências aleatórias. A partir destes resultados é possível avaliar o impacto da memória adaptativa e do procedimento de construção de vocabulário no desempenho da heurística `MP`. Um gráfico com um resumo dos resultados dos testes utilizando as heurísticas propostas para solucionar as instâncias do grupo R é apresentado na Figura 4.4. Cada ponto neste gráfico indica a porcentagem média de provas das soluções encontradas por uma das heurísticas para as instâncias de um determinado tamanho. Os resultados das heurísticas `MP` e `MP+Mem` indicam que a memória adaptativa melhora consideravelmente o desempenho da heurística, em relação à qualidade das soluções encontradas. Da mesma forma, os resultados das heurísticas `MP+CV` e `MP+Mem+CV` indicam que a memória adaptativa é responsável por uma grande melhoria na heurística `MP+CV`. Estes resultados destacam a eficiência da memória adaptativa como mecanismo de auxílio ao algoritmo construtivo em uma heurística multi-partida.

Em relação à construção de vocabulário, nota-se, comparando-se os resultados das heurísticas `MP` e `MP+CV`, que esta estratégia também é eficiente. Ela foi capaz de gerar novas soluções a partir de um dado conjunto de soluções e, ainda, gerar soluções melhores do que aquelas fornecidas. A utilização da memória adaptativa juntamente com a construção de vocabulário também apresenta-se atraente, apesar da diferença de qualidade entre as soluções encontradas pela heurística `MP+Mem+CV` e aquelas encontradas pela heurística `MP+Mem` ser pequena. Este fato é devido principalmente à alta qualidade das soluções encontradas pela heurística `MP+Mem`. Esta heurística encontrou soluções com a quantidade ótima de provas para mais de 90% das instâncias do grupo R com tamanho menor ou igual a 600 bp. Ou seja, a heurística `MP+Mem+CV` só poderia encontrar soluções melhores do que a heurística `MP+Mem` para menos de 10% destas instâncias. Portanto, as soluções encontradas pela heurística `MP+Mem+CV` não poderiam ser, na média, muito melhores do que aquelas encontradas pela heurística `MP+Mem`. Pode-se observar que a diferença de qualidade das soluções encon-

tradas pelas duas heurísticas cresce a partir das instâncias maiores do que 600 bp.

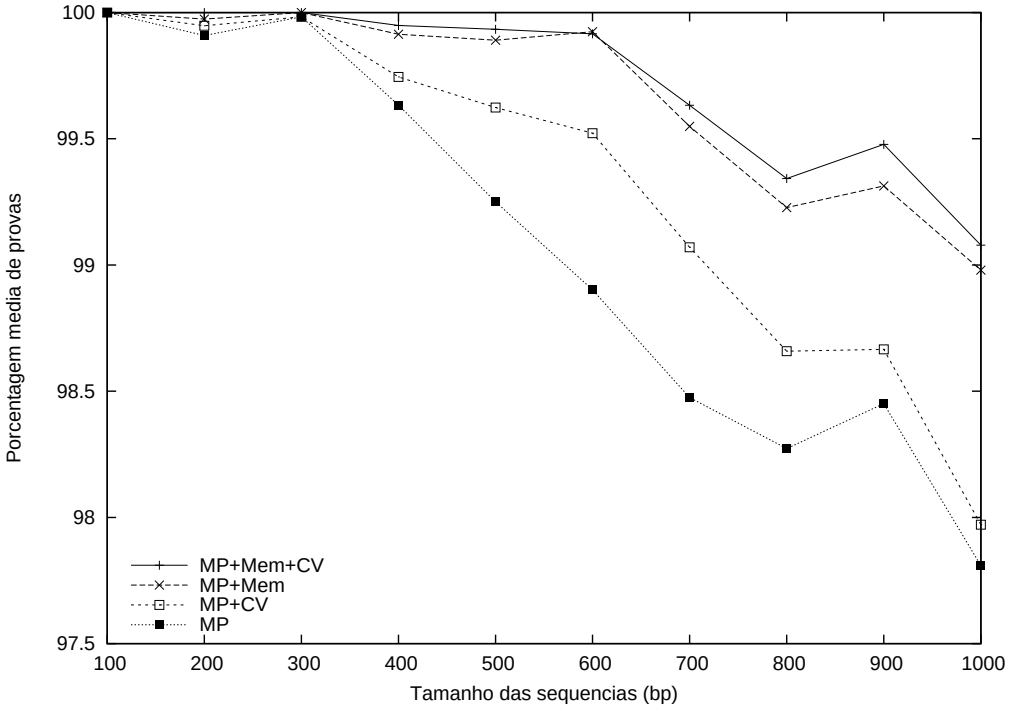

Figura 4.4: Porcentagem média de provas das soluções encontradas para as instâncias do grupo R utilizando as heurísticas propostas.

Na Figura 4.5 é apresentado um gráfico com as similaridades médias das mesmas soluções utilizadas no gráfico da Figura 4.4. Comparando-se estes dois gráficos, é possível avaliar a aderência entre as duas medidas de qualidade aqui utilizadas: porcentagem de provas e similaridade. Apesar de haver uma diferença clara entre os valores dos dois gráficos, a relação entre eles é consistente. Ou seja, soluções com mais provas geralmente representam seqüências mais parecidas com as seqüências alvo. A aderência não é tão boa devido aos problemas discutidos na Seção 4.3. Por exemplo, a heurística `MP+Mem+CV` encontrou soluções com a quantidade ótima de provas para todas as 100 instâncias do grupo R com tamanho de 600 bp. Entretanto, apenas 69 destas soluções reconstroem exatamente as seqüências alvo. Ou seja, pelo menos 31 instâncias deste conjunto possuem mais de uma solução com a quantidade ótima de provas. Estes resultados ilustram a principal dificuldade da técnica de SBH, que é o baixo poder de seqüenciamento dos arranjos padrão, representado pela baixa probabilidade de reconstrução única. Por isto, as propostas de diferentes projetos de arranjos de DNA discutidas na Seção 4.3, são importantes. Estes arranjos possuem um maior poder de seqüenciamento, ou seja, aumentam a probabilidade de reconstrução única.

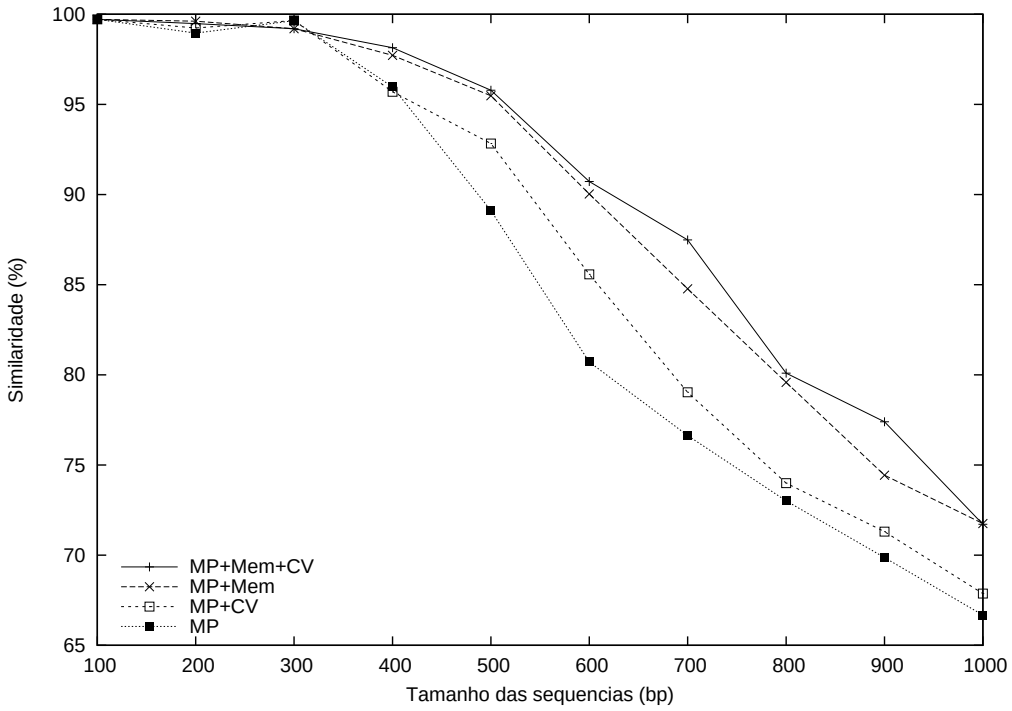

Figura 4.5: Similaridade média das soluções encontradas para as instâncias do grupo R utilizando as heurísticas propostas.

As médias dos tempos de computação das heurísticas nos testes ilustrados na Figura 4.4 são apresentados na Figura 4.6. A utilização da memória adaptativa implica em um aumento no tempo médio de computação devido à necessidade de atualizar o conjunto de soluções de elite e os valores relativos à freqüência de utilização das arestas. O método de construção de vocabulário também implica em um custo adicional no tempo de computação das heurísticas. Entretanto, os tempos da heurística mais lenta (`MP+Mem+CV`) são muito baixos. O tempo médio de computação desta heurística para as maiores instâncias testadas (instâncias de tamanho 1000 do grupo R) é aproximadamente meio minuto. Levando em consideração a aplicação, estes tempos são plenamente satisfatórios.

Para avaliar o impacto que a quantidade de erros no espectro têm no desempenho da heurística `MP+Mem+CV`, foram realizados testes com instâncias aleatórias contendo diferentes taxas de erros. Estas instâncias foram geradas com o mesmo procedimento utilizado para gerar as instâncias do grupo R. Foram geradas instâncias de um arranjo $C(9)$ com as seguintes taxas de falsos negativos e falsos positivos: 0%, 10%, 20% e 30%. Os resultados dos testes da heurística `MP+Mem+CV` com estas instâncias são apresentados nas Figuras 4.7 e 4.8. Os valores da primeira figura dizem respeito a similaridade média das soluções encontradas, enquanto a segunda figura apresenta a porcentagem média de provas das soluções. Também fo-

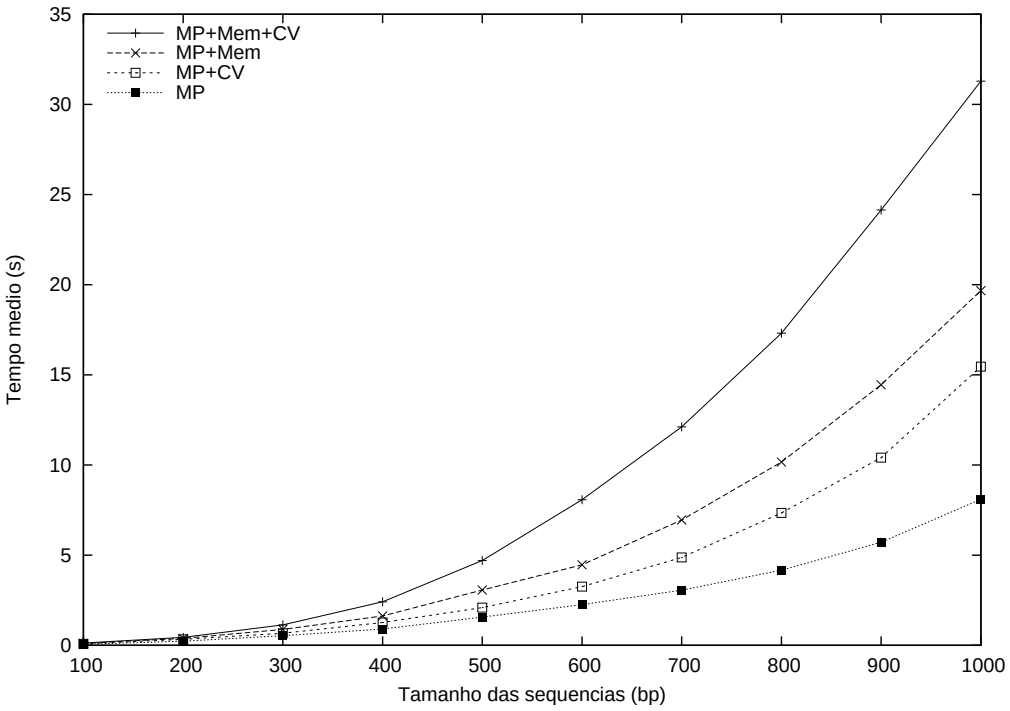

Figura 4.6: Tempo médio de computação necessário para resolver as instâncias do grupo R utilizando as heurísticas propostas.

ram realizados testes com instâncias correspondentes a arranjos com diferentes tamanhos de provas. Foram geradas instâncias com 20% de erros através da simulação do experimento de hibridação com arranjos $C(l)$, para $l \in \{7, 8, 9, 10, 11\}$. Os resultados dos testes com estas instâncias são apresentados nas Figuras 4.9 e 4.10. Estas figuras apresentam a similaridade média e a porcentagem média das soluções, respectivamente. Os tempos médios de computação da heurística `MP+Mem+CV` para estes dois grupos de instâncias são praticamente idênticos àqueles apresentados na Figura 4.6, pois esta heurística não é sensível às características variadas nestas instâncias.

Comparação com os melhores algoritmos da literatura

Apresentam-se a seguir algumas comparações entre os resultados obtidos pela heurística `MP+Mem+CV` e os três melhores algoritmos encontrados na literatura, denominados aqui *algoritmos de base*. Os algoritmos utilizados como base de comparação foram apresentados na Seção 3.1 e são denominados aqui como:

- BT – heurística baseada na metaheurística Busca Tabu [5];
- JS – heurística baseada na idéia de Janelas de Sobreposição [4];

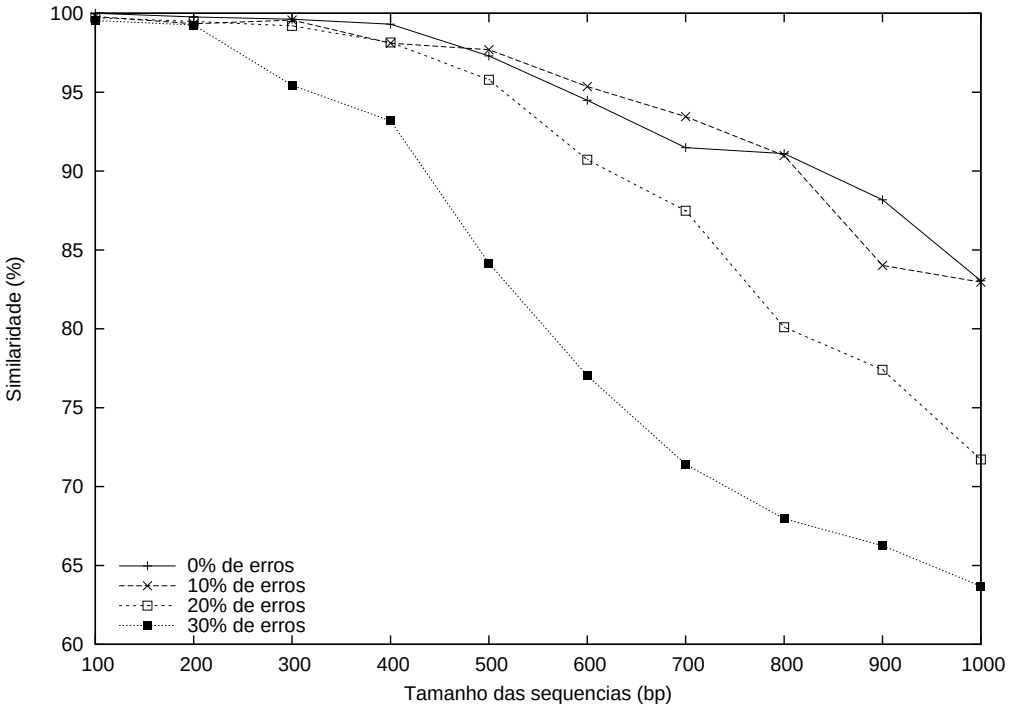

Figura 4.7: Similaridade média das soluções encontradas pela heurística `MP+Mem+CV` para as instâncias do grupo R com diferentes taxas de erro.

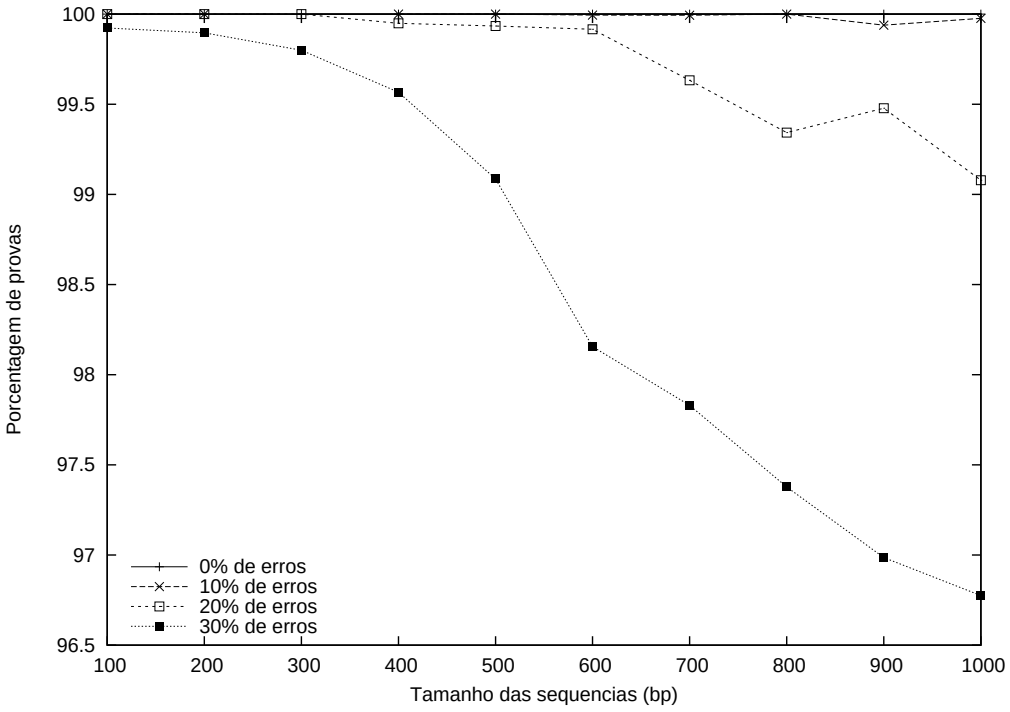

Figura 4.8: Porcentagem média de provas das soluções encontradas pela heurística `MP+Mem+CV` para as instâncias do grupo R com diferentes taxas de erro.

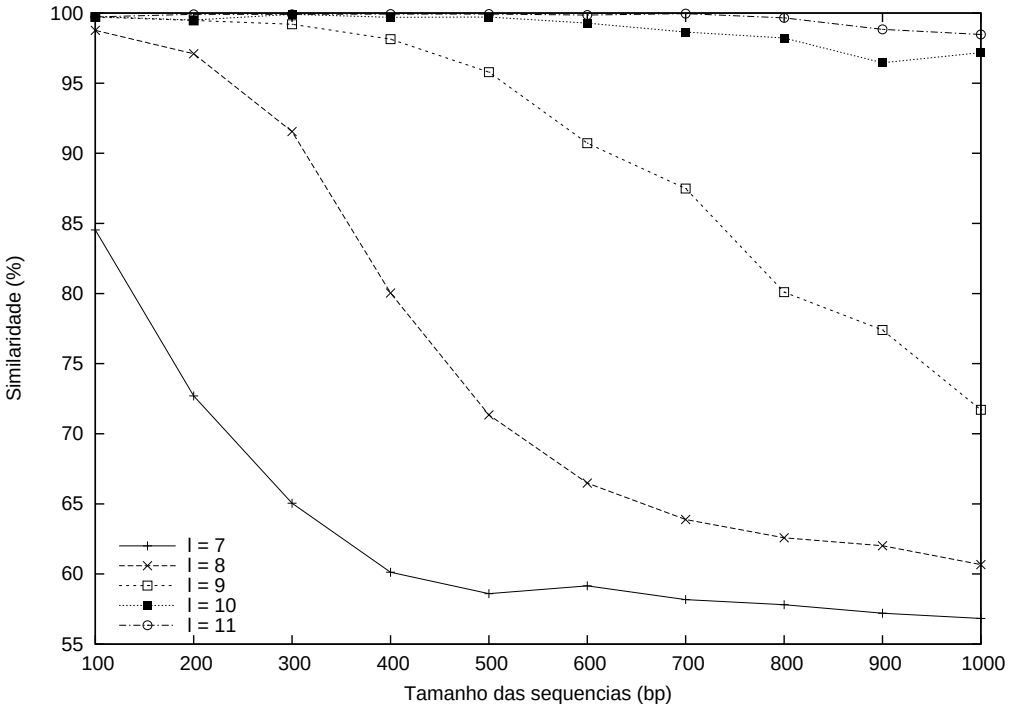

Figura 4.9: Similaridade média das soluções encontradas pela heurística `MP+Mem+CV` para as instâncias do grupo R com diferentes tamanhos de provas.

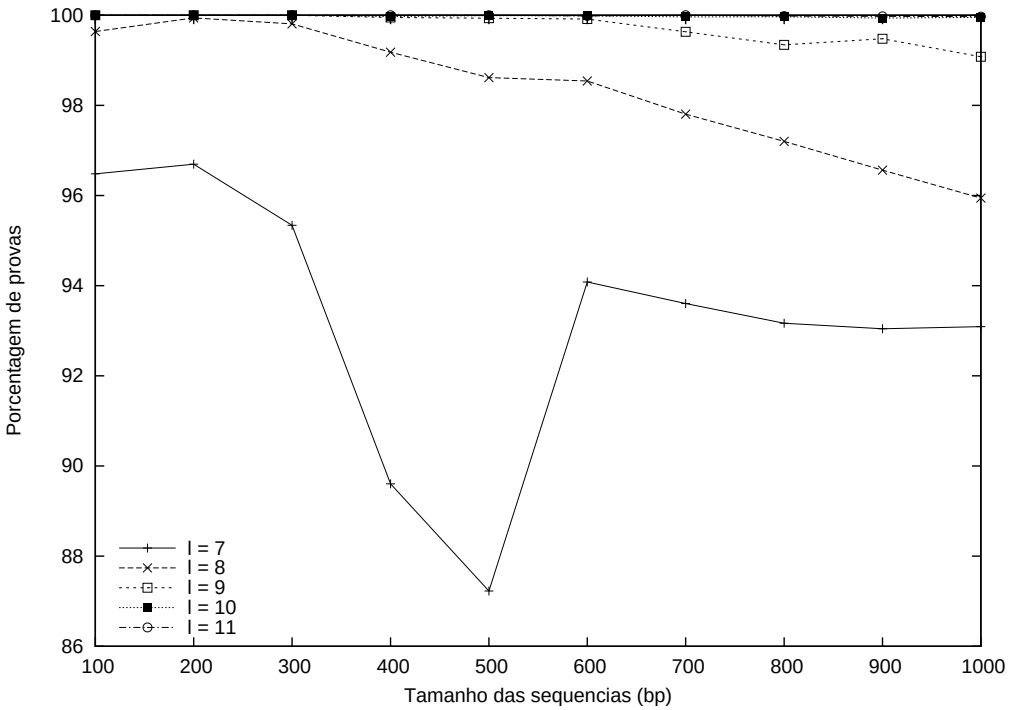

Figura 4.10: Porcentagem média de provas das soluções encontradas pela heurística `MP+Mem+CV` para as instâncias do grupo R com diferentes tamanhos de provas.

	Tamanho das seqüências				
Algoritmo	109	209	309	409	509
BT	98.6	94.1	89.6	88.5	80.7
JS	99.4	95.2	95.7	92.1	90.1
SOPAS	98.3	97.9	99.1	98.1	93.5
`MP+Mem+CV`	100.0	100.0	99.2	99.4	99.5

Tabela 4.4: Similaridade média das soluções encontradas pela heurística `MP+Mem+CV` e pelos algoritmos de base para as instâncias do grupo A.

	Tamanho das seqüências				
Algoritmo	109	209	309	409	509
BT	28	23	17	10	10
JS	28	20	21	13	14
SOPAS	37	30	37	30	28
`MP+Mem+CV`	40	40	39	39	39

Tabela 4.5: Quantidade de seqüências alvo encontradas pela heurística `MP+Mem+CV` e pelos algoritmos de base para as instâncias do grupo A.

– SOPAS – heurística baseada em algoritmos genéticos, denominada pelo autor "SOlid Probe ASsembler" [13].

Um resumo dos resultados da heurística `MP+Mem+CV` e dos algoritmos de base para as 200 instâncias do grupo A é apresentado na Tabela 4.4. Cada valor nesta tabela diz respeito à similaridade média das soluções encontradas por um dos algoritmos para um dos conjuntos com 40 instâncias de um mesmo tamanho. Na Tabela 4.5 são apresentadas as quantidades de seqüências alvo encontradas por cada algoritmo nestes testes. Os resultados da heurística `MP+Mem+CV` para estas instâncias são claramente superiores aos resultados dos demais algoritmos. Esta heurística foi capaz de encontrar as seqüências de DNA alvo para 197 instâncias do grupo A. Os tempos médios de computação são apresentados na Tabela 4.6. Apesar dos tempos da heurística JS serem os melhores, nota-se que estes valores são relativos a testes realizados em um supercomputador CRAY T3E-900, enquanto os testes com a heurística `MP+Mem+CV` foram realizados em um computador pessoal com poder computacional muito menor.

Um resumo dos resultados da heurística `MP+Mem+CV` e dos algoritmos de base para as 78 instâncias do grupo B são apresentados na Tabela 4.7. Cada linha desta tabela corresponde aos resultados de um algoritmo. As colunas dizem respeito, respectivamente, às seguintes informações: similaridade média das soluções geradas, quantidade de seqüências alvo encontradas e tempo médio de computação. Nestes testes a heurística `MP+Mem+CV` superou o algoritmo SOPAS. Entretanto, o algoritmo BT e, principalmente, o

	Tamanho das seqüências				
Algoritmo	109	209	309	409	509
BT	< 1	5	14	28	51
JS	< 1	< 1	< 1	< 1	< 1
SOPAS	< 1	< 1	< 1	1	2
`MP+Mem+CV`	< 1	< 1	1	3	6

Tabela 4.6: Tempo médio de computação da heurística `MP+Mem+CV` e dos algoritmos de base para solucionar as instâncias do grupo A

Algoritmo	Similaridade	Ótimos	Tempo
BT	61.2	0	27
JS	75.4	0	< 1
SOPAS	43.2	0	800
`MP+Mem+CV`	60.2	0	7

Tabela 4.7: Resumo dos resultados da heurística `MP+Mem+CV` e dos algoritmos de base para as 78 instâncias do grupo B. Para cada algoritmo, as colunas dizem respeito a: similaridade média, quantidade de seqüências alvo reconstruídas e tempo médio de computação.

algoritmo JS encontraram melhores soluções do que a heurística proposta. Estas instâncias são particularmente difíceis devido às repetições de provas nas seqüências alvo. Tipicamente, erros desta natureza aparecem juntos na seqüência alvo. Isto dificulta muito o problema de montagem, diminuindo, inclusive, a probabilidade de reconstrução única.

Em relação às instâncias do grupo R, foram realizadas comparações apenas entre a heurística `MP+Mem+CV` e o algoritmo SOPAS, pois estes resultados não estão disponíveis para os outros algoritmos de base. Os resultados apresentados aqui para o algoritmo SOPAS são relativos a testes realizados no mesmo computador utilizado para executar as heurísticas propostas. Isto foi possível porque o código fonte do algoritmo SOPAS foi disponibilizado pelo autor [12]. Os resultados relativos às instâncias do grupo R estão resumidos na Figura 4.12. Cada ponto no gráfico diz respeito à similaridade média das soluções encontradas por um dos algoritmo para um dos conjuntos com 100 instâncias de um mesmo tamanho. Estes resultados mostram, mais uma vez, a superioridade da heurística `MP+Mem+CV` em encontrar melhores soluções para o PSBH. As quantidades de seqüências alvo encontradas pelos algoritmos nestes testes são apresentadas na Tabela 4.8. Os tempos médios de computação são ilustrados na Figura 4.12. Observando-se esta figura nota-se que a heurística proposta aqui é mais eficiente também em relação ao tempo de computação. O pico observado na figura no ponto relativo ao tempo médio do algoritmo SOPAS para as

	Heurística	
Tamanho	`MP+Mem+CV`	SOPAS
100	79	70
200	74	61
300	83	55
400	73	37
500	61	23
600	52	11
700	34	9
800	10	3
900	13	1
1000	2	2

Tabela 4.8: Quantidade de ótimos encontrados pela heurística `MP+Mem+CV` e pelo algoritmo SOPAS para as instâncias do grupo R.

instâncias de tamanho 700 é devido a três instâncias. O algoritmo teve muita dificuldade em resolver estas instâncias, levando 39, 84 e 178 minutos para cada uma.

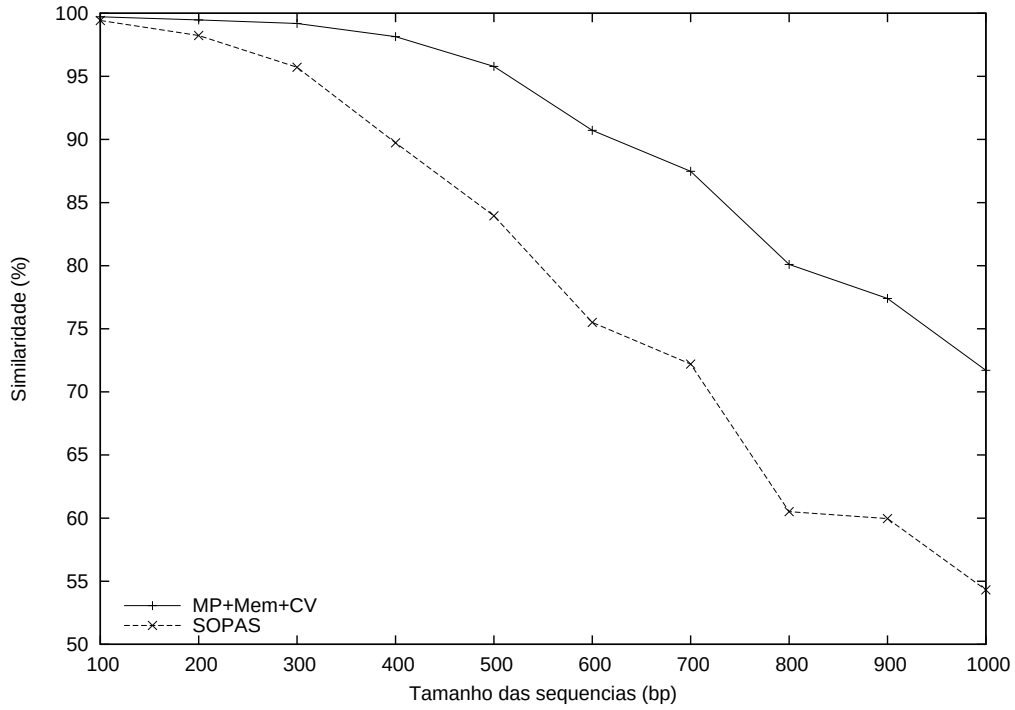

Figura 4.11: Porcentagem média de provas das soluções encontradas para as instâncias do grupo R utilizando a heurística `MP+Mem+CV` e o algoritmo SOPAS.

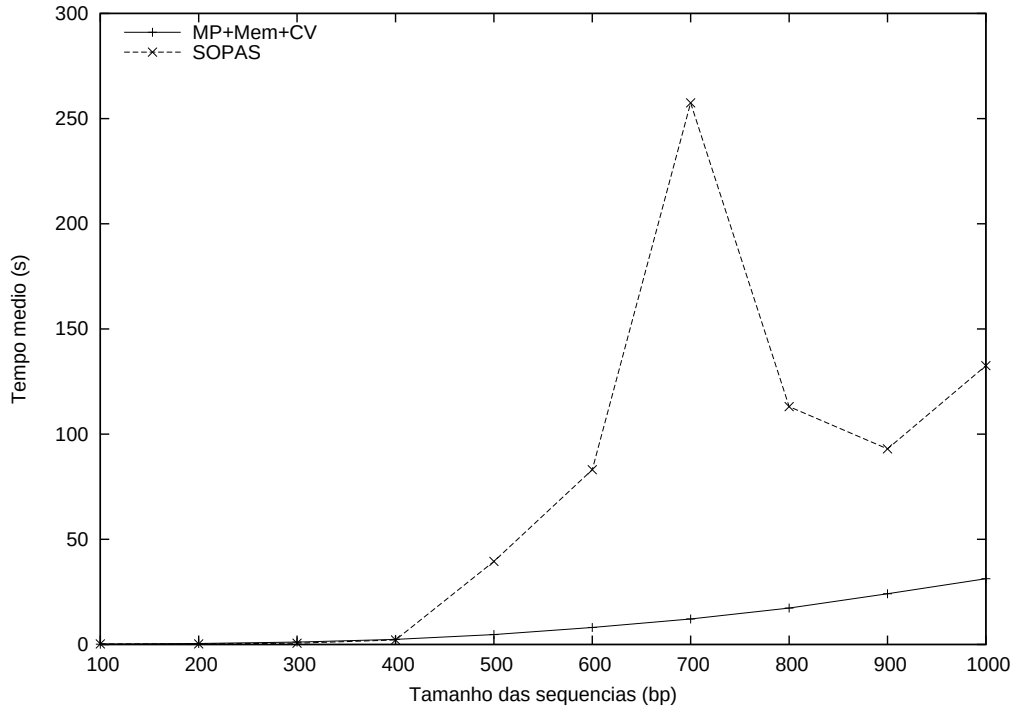

Figura 4.12: Tempo médio de computação necessário para resolver as instâncias do grupo R utilizando as heurísticas `MP+Mem+CV` e SOPAS.