

UNIVERSIDADE FEDERAL DE OURO PRETO

JOAQUIM ALVES VIANA JÚNIOR

Segundo Trabalho Prático da Disciplina BCC241 – Projeto e Análise de Algoritmos

Relatório técnico do segundo trabalho prático da
Graduação em Ciência da Computação da
Universidade Federal de Ouro Preto como
requisito parcial para conclusão da disciplina
BCC241 - Projeto e Análise de Algoritmos.

Prof.:Dr. Haroldo Gambini Santos

Ouro Preto

2018

1 Introdução

Conforme especificação fornecida em sala de aula, este trabalho apresenta uma solução, através de algoritmos aproximados e metaheurísticas, para o problema do empacotamento com conflito, também conhecido como Problema de *Bin Packing* com Conflitos (PBPC) em compartimentos de capacidade c .

O problema proposto neste trabalho considera um conjunto I de itens $\{1, \dots, n\}$, e desejamos alocar estes itens em contêineres (bin), considerando que o número de contêineres é ilimitado. Cada contêiner possui uma capacidade $c \in \mathbb{Z}^+$ e cada item possui um peso $w_i \in \mathbb{Z}^+$, e há ainda um conjunto de conflitos F_i para cada item. Com isso temos como objetivo a minimização do número de contêineres usados. Uma solução é um agrupamento de itens por contêiner, usando o menor número possível, respeitando sua capacidade e conflitos para cada contêiner usado.

2 Desenvolvimento

2.1 Formulação

O PBPC é um problema desafiador na área da otimização. Diferentes abordagens sobre o problema mostram formas mais ou menos eficientes a depender da instância considerada. Heurísticas construtivas, algoritmos evolutivos, branch & bound, e outros são alguns dos métodos que podemos usar para se alcançar uma solução ótima ou próximo dela. O problema possui uma grande relevância dado que o problema envolve restrições de capacidade e conflito, cada um pode ser vistos como Problema de Bin Packing ou Coloração de Vértices, por isso algumas metodologias desenvolvidas para um podem não funcionar bem para outro e vice-versa.

Neste trabalho foram implementados 4 métodos com heurística gulosa/pseudogulosa(*First-Fit*(2.2.1), *First-Fit Decrescente*(2.2.2), *Best-Fit*(2.2.3) e *Worst-Fit*(2.2.4)), além de uma implementação do *Simulated Annealing*(2.2.5). É importante ressaltar que além destas heurísticas, foi utilizado o algoritmo de Bron Kerbosch(2.2.6) para encontrar o número mínimo de contêineres a serem usados baseado nos itens mais conflitantes.

2.1.1 Elaboração das estruturas de dados

Dado o conjunto I_n de itens e seus respectivos pesos w_i e conflitos(simétricos, ou seja, se x possui conflito com y , então y possui conflito com x), foi elaborado um grafo $G(V,A)$ para representar o conjunto de entrada, onde os vértices representam os itens e as arestas os conflitos (se x e y são conflitantes então há uma aresta de v_x para v_y).

2.2 Algoritmos

2.2.1 First-Fit

O First-Fit é um algoritmo de aproximação bem simples, utilizando sempre a perspectiva gulosa para montar a solução. Os itens são processados pela sua ordem de entrada no programa. Para cada item, ele tenta colocar o item no primeiro pacote que pode acomodá-lo. Se nenhum

pacote é encontrado, ele abre um novo pacote e coloca o item dentro deste novo pacote. No pior caso um novo contêiner terá que ser criado para cada item logo temos $O(n^2)$.

Algoritmo 1 : First-Fit

```

1.  para  $i$  de 1 até  $N$  faça
2.      para  $j$  de 1 até ... faça
3.          se  $\text{Item}_i$  cabe em  $\text{Contêiner}_j$ 
4.              se  $\text{Item}_i$  não possui conflito com outros itens
5.                  colocar  $\text{Item}_i$  no  $\text{contêiner}_j$ 
6.              fim-se
7.          fim-se
8.      fim-para
9.      se  $\text{Item}_i$  não foi alocado em nenhum contêiner então
10.         criar um novo contêiner e colocar  $\text{Item}_i$ .
11.      fim-se
12.  fim-para

```

FONTE: ELABORADO PELO AUTOR

2.2.2 First-Fit Decrescente

O First-Fit Decrescente utiliza-se de um algoritmo de ordenação para que o peso dos itens a serem escolhidos fiquem ordenados decrescentemente. Neste trabalho foi utilizado o QuickSort(no pior caso, $O(n^2)$) logo temos $O(n^2)$.

Algoritmo 2 : First-Fit Decrescente

```

1.  Ordene os objetos decrescentemente utilizando QuickSort();
2.  Aplique o First-Fit nos itens ordenados

```

FONTE: ELABORADO PELO AUTOR

2.2.3 Best-Fit

O Best-Fit é um algoritmo por aproximação muito semelhante ao **First-Fit** que busca no momento da escolha do contêiner aquele que possui o menor espaço restante. $O(n^2)$. Pode ser feita uma melhoria no algoritmo se mantivermos os contêiner em uma fila de prioridade, heap($O(n \log n)$).

Algoritmo 3 : Best-Fit

```

1.   para  $i$  de 1 até  $N$  faça
2.   Determine  $k = \mathbf{min}$ (espaço restante dos Contêineres)
3.   se  $\text{Item}_i$  cabe dentro do contêiner $_k$  então
4.       se  $\text{Item}_i$  não possui conflito com outros itens
5.           colocar  $\text{Item}_i$  no contêiner $_j$ 
6.   senão
           Criar um novo contêiner e colocar  $\text{Item}_i$ .
7.   fim-se
8.   fim-para

```

FONTE:ELABORADO PELO AUTOR

2.2.4 Worst-Fit

O Worst-Fit é um algoritmo por aproximação muito semelhante ao **Worst-Fit** que busca no momento da escolha do contêiner aquele que possui o maior espaço restante. $O(n^2)$. Também podemos melhorar este algoritmo se mantivermos os contêiner em uma fila de prioridade ($O(n \log n)$).

Algoritmo 4 : Worst-Fit

```

1.   para  $i$  de 1 até  $N$  faça
2.   Determine  $k = \mathbf{max}$ (espaço restante dos Contêineres)
3.   se  $\text{Item}_i$  cabe dentro do contêiner $_k$  então
4.       se  $\text{Item}_i$  não possui conflito com outros itens
5.           Colocar  $\text{Item}_i$  no contêiner $_j$ 
6.   senão
           Criar um novo contêiner e colocar  $\text{Item}_i$ .
7.   fim-se
8.   fim-para

```

FONTE:ELABORADO PELO AUTOR

2.2.5 Simulated Annealing

O Simulated Annealing é uma meta-heurística que consiste na busca local. Sua ideia vem de uma analogia com a termodinâmica, onde a temperatura influencia no fator de mudanças de um determinado evento.

Essa técnica começa sua busca a partir de uma solução inicial qualquer, e a cada iteração cria uma nova solução que é ou não aceita de acordo com um fator probabilístico, este processo se repete até que o tempo limite seja alcançado ou a temperatura diminua demais. Para entender isso temos alguns parâmetros:

S_0 = Solução Inicial (Entrada);

S_i = Solução da iteração i ;

S = Configuração Final;

T_0 = Temperatura Inicial;

T_i = Temperatura na Iteração i ;

α = Factor de redução da temperatura (Entrada);

$f(S_i)$ = Valor da função objetivo correspondente à configuração S_i ;

i = Variável de controle de Loops.

Algoritmo 5 : Simulated Annealing

```

1.  S = S0
2.  T0 = TempInicial()
3.  T = T0
4.  eps = 0.0001
5.  repita
6.      enquanto (TempoLimite < TempoAtual e T > eps)
7.          i = 1
8.          repita
9.              Si = Perturba(S)
10.             ΔFi = f(Si) - f(S)
11.             se (Δfi ≤ 0) ou (exp(-Δfi/T) > Random)) então
12.                 S = Si
13.             Fim-se
14.             i = i + 1
15.         T = α.T

```

FONTE: ELABORADO PELO AUTOR

2.2.6 Clique Maximal – Bron-Kerbosch

A fim de trazer um maior refinamento nos resultados em busca da solução ótima, foi implementado o algoritmo para encontrar o clique maximal em um grafo. Um clique nada mais é que um sub-grafo completo e neste problema procura-se saber sua cardinalidade, ou seja quantos vértices pertencem a ele quais são estes vértices.

Neste trabalho as instâncias são montadas na forma de um grafo como anteriormente mencionado. A partir disso conseguimos encontrar um clique maximal onde os itens pertencentes a ele não podem compartilhar o mesmo contêiner. Com isso sabemos o número mínimo de contêiners para a instância sendo gerada.

Para encontrar o pivô foi calculado o grau modificado de cada vértice de modo a saber quais vértices tem mais chance de estar no clique.

16. Algoritmo 6 : Pivotamento pelo Maior Grau Modificado

1. **BK**(C, P, S)
 2. **se** P e S estão vazios então
 3. Adiciona a clique C no conjunto solução;
 4. **fim**
 5. Escolha o vértice de máximo grau modificado em P como pivô u;
 6. **para** cada Vértice v em $P/N(u)$ faça
 7. **BK**(C $\cup \{v\}$ P $\cap N(v)$, S $\cap N(v)$);
 8. P := P $\setminus \{v\}$;
 9. S := S $\setminus \{v\}$;
 10. **fim**
-

FONTE: <http://www.decom.ufop.br/haroldo/>

3 Experimentos e Análises

Para a rerealização dos experimentos foi utilizado um computador com as seguintes configurações: processador Intel core i5-3230M CPU @ 2.60GHz 2.60 GHz, 8GB de RAM e sistema operacional Microsoft Windows 10 64bits.

A implementação deste trabalho, feita em C++(versão 11) encontra-se no repositório público do GitHub(www.github.com/junior003/TPPAA2), bem como as instâncias que foram usadas para testar o algoritmo.

3.1 Análise estática de código com cppcheck

Conforme foi sugerido pelo professor foi utilizado uma ferramenta de verificação com uso do compilador cppCheck utilizando o seguinte comando no terminal (é necessário estar na pasta da implementação):

```
Cppcheck *.cpp
```

Como resultado obtém-se:

```
Checking BronKerbosch.cpp ...  
1/7 files checked 18% done  
Checking Container.cpp ...  
2/7 files checked 21% done  
Checking Entrada_e_Saida.cpp ...  
3/7 files checked 28% done  
Checking Item.cpp ...  
4/7 files checked 32% done  
Checking Main.cpp ...  
5/7 files checked 38% done  
Checking Metodos.cpp ...  
6/7 files checked 95% done  
Checking Sorting.cpp ...  
7/7 files checked 100% done
```

FONTE:ELABORADO PELO AUTOR

3.2 Experimentos

Os testes foram elaborados de uma forma testar diferentes formas de se chegar a uma solução. Combinando os diferentes métodos gulosos com o Simulated Annealing e em alguns casos utilizando o algoritmo do Clique Maximal.

Tempo de Execução e solução encontrada

TESTE1

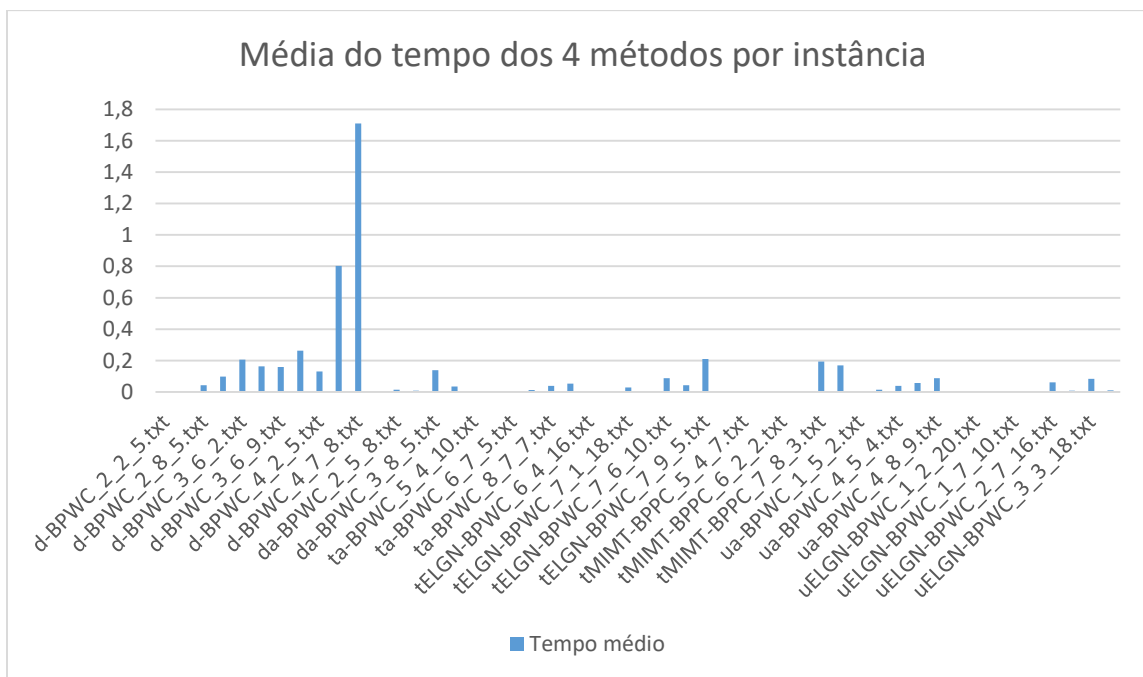
Neste primeiro teste os métodos gulosos foram executados uma única vez, os dados seguem na tabela abaixo(solução encontrada, e tempo em segundos):

Instância	FirstFit		BestFit		WorstFit		FirstFitD	
d-BPWC_2_2_5.txt	60	0.00009	60	0.001994	59	0.004066	58	0.007994
d-BPWC_2_3_1.txt	89	0.00001	90	0.00001	88	0.009838	88	0.009838
d-BPWC_2_8_5.txt	203	0.040031	204	0.034321	202	0.041819	201	0.055032
d-BPWC_3_5_3.txt	250	0.07797	251	0.06931	248	0.09958	247	0.144265
d-BPWC_3_6_2.txt	308	0.148044	310	0.178785	308	0.19113	308	0.30415
d-BPWC_3_6_6.txt	305	0.121318	307	0.12389	304	0.15567	304	0.253039
d-BPWC_3_6_9.txt	292	0.124423	294	0.12167	292	0.14959	292	0.2429
d-BPWC_3_7_7.txt	359	0.2035	362	0.2164	359	0.24005	358	0.394407
d-BPWC_4_2_5.txt	180	0.016987	195	0.0209	181	0.3547	179	0.127245
d-BPWC_4_5_6.txt	505	0.59628	512	0.57403	504	0.7198	504	1.31742
d-BPWC_4_7_8.txt	661	1.30994	664	1.34277	660	1.66145	660	2.52354
da-BPWC_1_9_9.txt	57	0.00994	53	0.00001	59	0.01099	52	0.00538
da-BPWC_2_5_8.txt	46	0.016724	46	0.002053	49	0.026297	44	0.01030
da-BPWC_3_1_1.txt	80	0.00554	80	0.00001	83	0.014	80	0.01002
da-BPWC_3_8_5.txt	136	0.11136	136	0.13618	142	0.137863	131	0.16705
da-BPWC_4_2_8.txt	156	0.019137	156	0.02223	160	0.0627	156	0.03731
ta-BPWC_5_4_10.txt	23	0.00001	22	0.00022	23	0.00001	23	0.00001
ta-BPWC_5_7_3.txt	25	0.00001	25	0.00002	26	0.00009	25	0.00001
ta-BPWC_6_7_5.txt	47	0.001275	45	0.00901	48	0.001	46	0.003006
ta-BPWC_7_9_7.txt	108	0.00099	110	0.011984	116	0.01508	106	0.01702
ta-BPWC_8_7_7.txt	188	0.011995	185	0.1014	192	0.02099	186	0.021465
tELGN-BPWC_5_9_19.txt	56	0.0002	56	0.01112	55	0.09999	55	0.09999
tELGN-BPWC_6_4_16.txt	54	0.0099	57	0.001	56	0.00999	50	0.003783
tELGN-BPWC_6_6_19.txt	75	0.002997	78	0.00265	75	0.002247	74	0.005695
tELGN-BPWC_7_1_18.txt	86	0.0009	86	0.10001	95	0.00098	90	0.008994
tELGN-BPWC_7_2_2.txt	90	0.001002	90	0.0111	94	0.00999	91	0.005995
tELGN-BPWC_7_6_10.txt	149	0.14995	153	0.1499	152	0.017991	145	0.03298
tELGN-BPWC_7_8_3.txt	205	0.03628	208	0.036674	206	0.037945	202	0.061969
tELGN-BPWC_7_9_5.txt	228	0.044973	229	0.045976	228	0.045977	225	0.69956
tMIMT-BPPC_5_3_1.txt	22	0.001997	22	0.01442	24	0.001425	23	0.00001
tMIMT-BPPC_5_4_7.txt	25	0.00252	25	0.00001	25	0.00024	24	0.00099
tMIMT-BPPC_5_5_4.txt	29	0.00169	28	0.0001	30	0.000999	27	0.001
tMIMT-BPPC_6_2_2.txt	43	0.00099	43	0.0001	47	0.00098	44	0.00019
tMIMT-BPPC_7_1_5.txt	87	0.001086	87	0.011101	93	0.001002	91	0.008898
tMIMT-BPPC_7_8_3.txt	213	0.046817	215	0.040693	213	0.039925	212	0.6466
tMIMT-BPPC_8_6_7.txt	317	0.125697	327	0.129905	321	0.14452	312	0.27767

ua-BPWC_1_5_2.txt	55	0.00006	54	0.00001	62	0.00002	52	0.00006
ua-BPWC_4_2_9.txt	426	0.011833	426	0.01081	461	0.022035	428	0.0165
ua-BPWC_4_5_4.txt	427	0.015053	426	0.054801	466	0.054807	428	0.029997
ua-BPWC_4_7_5.txt	427	0.040259	427	0.04304	465	0.090299	425	0.053763
ua-BPWC_4_8_9.txt	437	0.056605	435	0.0734	474	0.12688	442	0.089946
uELGN-BPWC_1_2_1.txt	51	0.00001	50	0.00001	56	0.00001	50	0.00001
uELGN-BPWC_1_2_20.txt	52	0.00001	53	0.00001	58	0.00001	53	0.00001
uELGN-BPWC_1_5_1.txt	69	0.00001	69	0.007858	70	0.0005	66	0.00001
uELGN-BPWC_1_7_10.txt	87	0.00001	87	0.00001	88	0.011557	84	0.006997
uELGN-BPWC_2_4_7.txt	119	0.008399	120	0.003997	128	0.006653	118	0.009156
uELGN-BPWC_2_7_16.txt	182	0.023447	182	0.15783	187	0.019744	179	0.040085
uELGN-BPWC_3_1_18.txt	208	0.001997	208	0.002	232	0.0001967	208	0.03152
uELGN-BPWC_3_3_18.txt	209	0.00001	209	0.00001	237	0.00001	212	0.33526
uELGN-BPWC_3_3_9.txt	208	0.00098	207	0.00001	227	0.009895	208	0.034115

FONTE: ELABORADO PELO AUTOR

O gráfico abaixo representa a média dos tempos de cada método de acordo com as instâncias:



Podemos observar que as instâncias iniciais são muito mais custosas que as demais, neste teste. Isso se deve ao número de itens e ao número de conflito serem simultaneamente maiores que as demais instâncias. Exemplo: d-BPWC_4_7_8 possui 500 itens e o maior conflito (clique) encontrado foi de 482 itens conflitantes.

TESTE2

Neste segundo teste o algoritmo de Bron e Kerbosch foi usado para encontrar o clique e a partir disso foi usado o First-Fit para gerar soluções, os dados seguem na tabela abaixo(solução encontrada,tempo limite 10 segundos):

Instância	Clique Maximal com FirstFit	
d-BPWC_2_2_5.txt	58	7,0009
d-BPWC_2_3_1.txt	88	7,0055
d-BPWC_2_8_5.txt	201	7,0005
d-BPWC_3_5_3.txt	247	7,0004
d-BPWC_3_6_2.txt	310	7,0001
d-BPWC_3_6_6.txt	304	7,0012
d-BPWC_3_6_9.txt	292	7,051
d-BPWC_3_7_7.txt	358	7,0004
d-BPWC_4_2_5.txt	179	7,0035
d-BPWC_4_5_6.txt	504	7,0004
d-BPWC_4_7_8.txt	660	7,0003
da-BPWC_1_9_9.txt	51	7,0074
da-BPWC_2_5_8.txt	44	7,0287
da-BPWC_3_1_1.txt	80	7,0005
da-BPWC_3_8_5.txt	137	7,0006
da-BPWC_4_2_8.txt	156	7,0004
ta-BPWC_5_4_10.txt	23	7,0018
ta-BPWC_5_7_3.txt	24	7,0000
ta-BPWC_6_7_5.txt	47	7,004
ta-BPWC_7_9_7.txt	107	7,00000
ta-BPWC_8_7_7.txt	188	7,00005
tELGN-BPWC_5_9_19.txt	55	7,00025
tELGN-BPWC_6_4_16.txt	51	7,00014
tELGN-BPWC_6_6_19.txt	74	7,00039
tELGN-BPWC_7_1_18.txt	90	7,00017
tELGN-BPWC_7_2_2.txt	91	7,00025
tELGN-BPWC_7_6_10.txt	145	7,00478
tELGN-BPWC_7_8_3.txt	202	7,00080
tELGN-BPWC_7_9_5.txt	226	7,00081
tMIMT-BPPC_5_3_1.txt	22	7,009
tMIMT-BPPC_5_4_7.txt	24	7,00084
tMIMT-BPPC_5_5_4.txt	29	7,00072
tMIMT-BPPC_6_2_2.txt	44	7,00082
tMIMT-BPPC_7_1_5.txt	91	7,005
tMIMT-BPPC_7_8_3.txt	213	7,00004
tMIMT-BPPC_8_6_7.txt	312	7,00006
ua-BPWC_1_5_2.txt	54	7,00007

ua-BPWC_4_2_9.txt	428	7,00090
ua-BPWC_4_5_4.txt	430	7,00008
ua-BPWC_4_7_5.txt	427	7,00099
ua-BPWC_4_8_9.txt	439	7,00055
uELGN-BPWC_1_2_1.txt	50	7,00005
uELGN-BPWC_1_2_20.txt	53	7,001
uELGN-BPWC_1_5_1.txt	65	7,00021
uELGN-BPWC_1_7_10.txt	85	7,00002
uELGN-BPWC_2_4_7.txt	65	7,00002
uELGN-BPWC_2_7_16.txt	85	7,00005
uELGN-BPWC_3_1_18.txt	117	7,00003
uELGN-BPWC_3_3_18.txt	180	7,00002
uELGN-BPWC_3_3_9.txt	212	7,00001

Neste teste podemos observar que o tempo gasto é sempre o mesmo, pois devido a sua grande expansão recursiva, o algoritmo acaba por sempre estourar o tempo limite estabelecido, o algoritmo está configurado para parar assim que atingir 70% do tempo total..

TESTE3

Neste terceiro teste os métodos gulosos foram executados como forma de gerar a solução inicial para o Simulated Annealing. Foi fixado o tempo limite = 600, a partir disso foram usados todos os outros métodos para gerar soluções (com temperatura = 50000 e 100000), os dados seguem na tabela abaixo (em segundos):

Temperatura = 50000:

Instância	FirstFit		BestFit		WorstFit		Solucao random	
d-BPWC_2_2_5.txt	60	3.52874	60	3.51225	59	4.34621	58	3.6251
d-BPWC_2_3_1.txt	89	6.72286	90	6.95068	88	6.9365	88	6.5148
d-BPWC_2_8_5.txt	201	31.4716	201	31.0561	202	31.0424	201	30.9851
d-BPWC_3_5_3.txt	247	48.1922	251	46.7953	248	51.27	247	57.5855
d-BPWC_3_6_2.txt	308	74.4782	310	76.7442	308	80.477	320	78.0331
d-BPWC_3_6_6.txt	305	69.3456	307	69.3038	304	73.9857	311	72.2042
d-BPWC_3_6_9.txt	292	67.5458	294	66.96668	292	77.3684	293	74.619
d-BPWC_3_7_7.txt	359	92.8234	362	93.0711	359	98.1796	358	96.5344
d-BPWC_4_2_5.txt	180	14.1888	195	18.0019	181	20.8036	255	37.6875
d-BPWC_4_5_6.txt	505	179.109	512	188.217	504	196.459	546	194.585
d-BPWC_4_7_8.txt	661	317.041	664	313.742	660	347.104	681	347.683
da-BPWC_1_9_9.txt	57	5.05391	58	4.6135	59	5.14977	55	4.83332
da-BPWC_2_5_8.txt	45	7.22978	44	6.8792	49	9.12813	79	13.6362
da-BPWC_3_1_1.txt	80	7.84308	80	7.76302	83	11.6592	130	21.6623

da-BPWC_3_8_5.txt	136	42.1542	136	41.2654	141	43.5366	180	56.6094
da-BPWC_4_2_8.txt	156	28.5099	156	29.3548	160	49.4136	162	108.501
ta-BPWC_5_4_10.txt	23	1.10224	22	1.08746	22	1.09524	23	1.2149
ta-BPWC_5_7_3.txt	24	1.32245	24	1.30502	26	1.45025	24	1.35433
ta-BPWC_6_7_5.txt	44	2.7922	44	2.79222	48	3.9175	53	4.74205
ta-BPWC_7_9_7.txt	107	13.8799	109	16.2419	114	17.0951	116	17.5923
ta-BPWC_8_7_7.txt	187	32.8908	185	32.5284	191	37.4897	205	48.8212
tELGN-BPWC_5_9_19.txt	56	3.20002	55	3.23955	55	3.3837	56	3.39539
tELGN-BPWC_6_4_16.txt	54	2.9669	57	2.85125	56	3.17897	51	3.32414
tELGN-BPWC_6_6_19.txt	75	5.85977	74	7.1198	75	6.50041	75	6.09952
tELGN-BPWC_7_1_18.txt	86	3.704	86	3.73533	95	4.91716	97	5.17728
tELGN-BPWC_7_2_2.txt	90	5.15984	90	4.50018	94	5.52989	96	6.07182
tELGN-BPWC_7_6_10.txt	149	18.9595	153	17.9501	152	19.35049	147	18.984
tELGN-BPWC_7_8_3.txt	205	29.6646	208	29.9995	206	29.6319	203	30.0216
tELGN-BPWC_7_9_5.txt	228	34.1199	229	34.8881	228	34.3097	232	34.8279
tMIMT-BPPC_5_3_1.txt	22	0.952757	22	0.949424	24	1.13776	24	1.37811
tMIMT-BPPC_5_4_7.txt	22	0.93919	22	0.92004	24	1.16000	23	1.03191
tMIMT-BPPC_5_5_4.txt	27	1.33	27	1.35	27	1.33	27	1.39
tMIMT-BPPC_6_2_2.txt	43	1.89014	43	1.86979	47	2.49133	47	2.55778
tMIMT-BPPC_7_1_5.txt	87	3.94914	87	4.30953	93	5.75144	98	4.1633
tMIMT-BPPC_7_8_3.txt	213	30.9836	215	31.6343	213	34.1951	212	31.3791
tMIMT-BPPC_8_6_7.txt	317	70.5828	327	72.584	321	76.9975	315	74.3606
ua-BPWC_1_5_2.txt	55	2.4004	54	2.36963	62	3.0995	53	2.7098
ua-BPWC_4_2_9.txt	426	44.343	426	42.1711	461	81.0346	441	58.0797
ua-BPWC_4_5_4.txt	427	63.5034	426	60.1632	466	122.679	439	88.5188
ua-BPWC_4_7_5.txt	427	64.3853	427	66.4528	465	127.598	473	103.553
ua-BPWC_4_8_9.txt	437	78.2531	435	78.8194	474	137.336	480	112.152
uELGN-BPWC_1_2_1.txt	51	2.07	50	1.96	56	2.0304	51	2.11014
uELGN-BPWC_1_2_20.txt	52	2.1222	53	2.16331	58	2.17847	53	2.1837
uELGN-BPWC_1_5_1.txt	65	3.908	66	3.7678	70	3.8008	65	4.27416
uELGN-BPWC_1_7_10.txt	83	6.48025	87	6.50769	88	7.25008	85	7.04995
uELGN-BPWC_2_4_7.txt	119	5.69783	120	5.95734	128	6.59015	115	6.35996
uELGN-BPWC_2_7_16.txt	182	20.8858	182	22.3083	187	21.7107	178	21.6018
uELGN-BPWC_3_1_18.txt	208	8.25031	208	8.017	232	10.55	213	10.3229
uELGN-BPWC_3_3_18.txt	209	11.1334	209	12.2569	237	18.5038	216	13.7702
uELGN-BPWC_3_3_9.txt	208	11.1107	207	11.1724	227	17.3076	214	13.2422

Temperatura = 100000:

Instância	FirstFit		BestFit		WorstFit		Solução Random	
d-BPWC_2_2_5.txt	58	6,8331	60	7,2564	59	8,27993	68	11,2983
d-BPWC_2_3_1.txt	89	13,3542	90	13,2992	88	13,1754	88	13,314
d-BPWC_2_8_5.txt	201	58,364	204	59,0551	202	60,2631	202	61,2849
d-BPWC_3_5_3.txt	247	95,1689	251	95,9061	248	95,2396	247	100,051
d-BPWC_3_6_2.txt	308	144,4782	310	146,7442	309	160,477	320	168,0331
d-BPWC_3_6_6.txt	305	129,3456	308	129,3038	304	143,9857	311	142,2042
d-BPWC_3_6_9.txt	292	127,5458	294	126,9668	292	147,3684	293	144,619
d-BPWC_3_7_7.txt	359	182,8234	362	183,0711	359	201,1796	358	198,5344
d-BPWC_4_2_5.txt	180	69,1888	199	80,0019	181	80,8036	255	96,6875
d-BPWC_4_5_6.txt	505	377,666	504	355,154	504	376,101	546	384,68
d-BPWC_4_7_8.txt	661	420,000	664	432,05	660	411,11	660	425,98
da-BPWC_1_9_9.txt	57	10,5842	57	11,0001	59	10,4802	61	11,1757
da-BPWC_2_5_8.txt	45	15,3483	44	14,3857	49	18,5104	72	26,3835
da-BPWC_3_1_1.txt	80	15,0453	80	15,6443	83	22,246	126	39,6209
da-BPWC_3_8_5.txt	136	80,9051	136	80,6632	141	83,6953	176	110,816
da-BPWC_4_2_8.txt	156	57,4025	156	57,4011	158	96,9376	163	217,984
ta-BPWC_5_4_10.txt	22	2,20943	22	2,18369	22	3,40367	25	3,50415
ta-BPWC_5_7_3.txt	24	2,6305	24	2,5487	24	2,9789	28	3,1564
ta-BPWC_6_7_5.txt	47	7,07003	44	5,61034	48	8,60346	51	8,4577
ta-BPWC_7_9_7.txt	107	27,7646	109	29,5978	115	35,83885	117	34,2853
ta-BPWC_8_7_7.txt	187	64,2544	185	60,9411	192	77,1413	191	100,299
tELGN-BPWC_5_9_19.txt	55	6,4797	55	7,4755	55	6,55877	55	5,4642
tELGN-BPWC_6_4_16.txt	51	5,95287	57	5,59201	56	5,9811	52	6,02154
tELGN-BPWC_6_6_19.txt	75	11,2564	74	11,3586	75	11,6581	75	11,6985
tELGN-BPWC_7_1_18.txt	86	7,2527	86	7,18	87	9,7412	86	10,3985
tELGN-BPWC_7_2_2.txt	90	10,2	90	9,6658	94	10,7584	90	11,9957
tELGN-BPWC_7_6_10.txt	149	37,9396	149	35,41257	152	37,5715	146	35,3984
tELGN-BPWC_7_8_3.txt	205	58,7507	205	57,5856	206	60,5039	208	57,9715
tELGN-BPWC_7_9_5.txt	228	66,8162	229	69,2316	226	67,7258	226	66,6829
tMIMT-BPPC_5_3_1.txt	22	1,93388	22	1,976	24	2,35283	23	2,17234
tMIMT-BPPC_5_4_7.txt	22	1,9597	22	1,92826	24	2,7811	24	2,20402
tMIMT-BPPC_5_5_4.txt	27	2,73499	27	2,63801	28	2,8901	27	2,63798
tMIMT-BPPC_6_2_2.txt	43	4,01176	43	3,77899	47	4,7112	48	5,144
tMIMT-BPPC_7_1_5.txt	87	4,778053	87	4,76852	93	4,09166	93	4,96811
tMIMT-BPPC_7_8_3.txt	213	6,60602	215	7,64084	213	7,33153	215	6,6363
tMIMT-BPPC_8_6_7.txt	317	14,4609	318	14,8012	321	16,2339	317	16,0159
ua-BPWC_1_5_2.txt	53	2,504713	54	2,46989	60	2,62197	53	2,510351
ua-BPWC_4_2_9.txt	426	88,343	426	84,1711	428	160,0346	430	120,0797
ua-BPWC_4_5_4.txt	426	123,5034	426	120,1632	436	122,679	439	169,5188
ua-BPWC_4_7_5.txt	427	122,3853	426	132,4528	465	203,598	473	204,553
ua-BPWC_4_8_9.txt	437	145,2531	435	150,8194	474	245,336	480	235,152

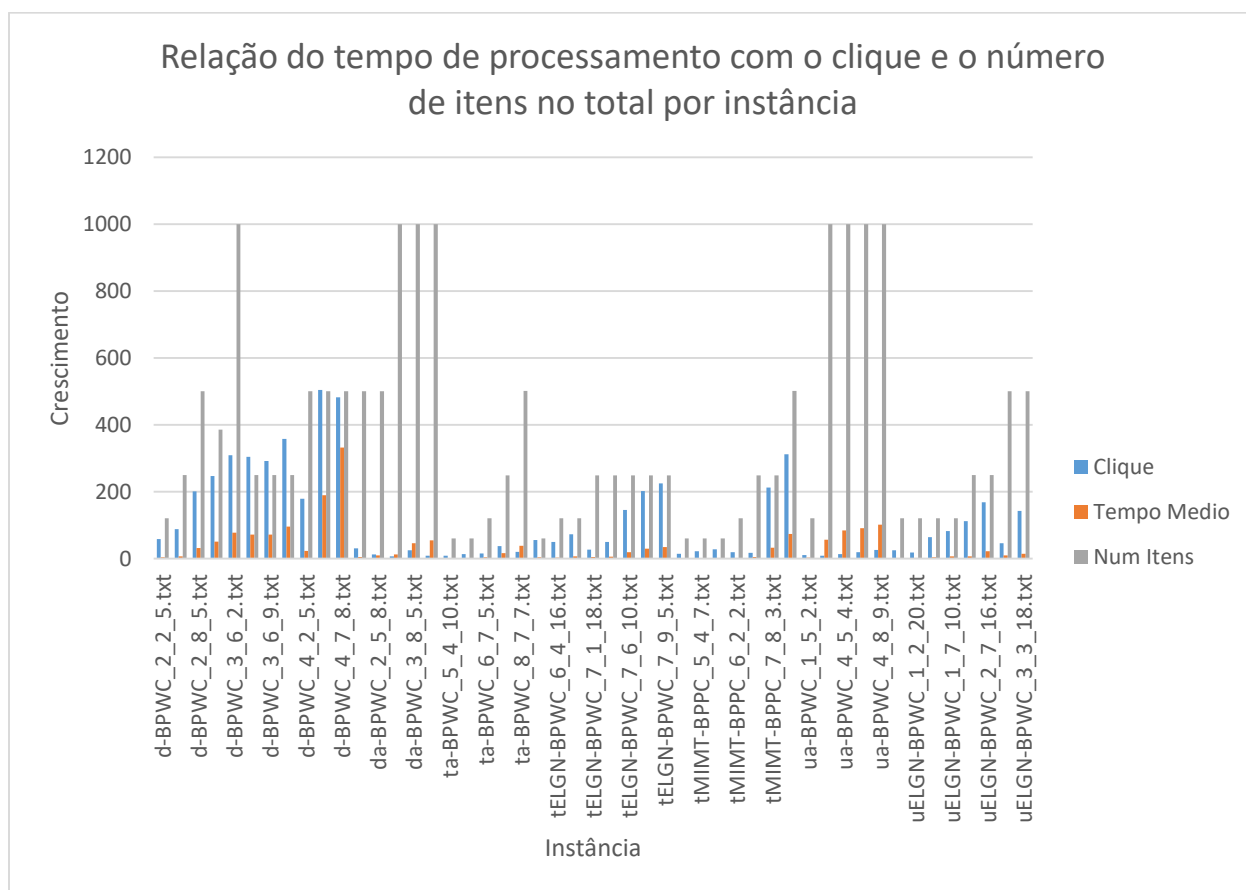
uELGN-BPWC_1_2_1.txt	50	2.07	50	1.96	56	2.0304	0,52	2.1837
uELGN-BPWC_1_2_20.txt	52	2.1222	53	2.16331	58	2.17847	53	4.27416
uELGN-BPWC_1_5_1.txt	69	3.908	66	3.7678	70	3.8008	65	7.04995
uELGN-BPWC_1_7_10.txt	83	6.48025	87	6.50769	88	7.25008	85	6.35996
uELGN-BPWC_2_4_7.txt	117	5.69783	120	5.95734	128	6.59015	117	21.6018
uELGN-BPWC_2_7_16.txt	182	20.8858	182	22.3083	187	21.7107	180	10.3229
uELGN-BPWC_3_1_18.txt	208	8.25031	208	8.017	232	10.55	212	13.7702
uELGN-BPWC_3_3_18.txt	209	11.1334	209	12.2569	237	18.5038	217	13.2422
uELGN-BPWC_3_3_9.txt	208	11.1107	207	11.1724	227	17.3076	216	2.1837

FONTE:ELABORADO PELO AUTOR

Neste experimento pudemos observar para diversas instâncias, que com a temperatura inicial mais alta alcançou-se resultados mais rapidamente principalmente as que foram geradas inicialmente por métodos randômicos.

TESTE 4

O gráfico abaixo mostra a relação tempo-número de itens – clique para cada instância:



FONTE:ELABORADO PELO AUTOR

Conclusão

Através destes teste pudemos observar que:

- 1) First- Fit - apesar de ser um algoritmo simples encontra uma boa resposta
- 2) Best-Fit e Worst- Fit – para algumas ocasiões não é bom, porém se usado com SA com uma temperatura alta consegue bons resultados quase sempre
- 3) Clique – O uso do clique maximal ajudou a entender o crescimento do tempo de execução e o comportamento de cada procedimento

O PBPC é um problema de grande relevância, seja na logistica de entrega de alguma mercadoria como no próprio empacotamento de algum certo material. Este trabalho mostrou diferentes formas de resolver e analisar diferentes formas de se alcançar uma solução próxima da ótima.