COMPARAÇÃO ENTRE MÉTODOS HEURÍSTICOS PARA UM PROBLEMA DE LAYOUT DE FACILIDADES

Renzo Q. Malini
Universidade Federal do Espírito Santo
Departamento de Informática
29060-900, Vitória, ES, Brasil, +55-27-3335-2420
rmalini@inf.ufes.br

André R. S. Amaral
Universidade Federal do Espírito Santo
Departamento de Informática
29060-900, Vitória, ES, Brasil, +55-27-3335-2420
amaral@inf.ufes.br

Resumo: No problema do layout de facilidades analisado aqui, desejamos atribuir *n* facilidades a *n* localidades de forma que os custos de comunicação entre as facilidades sejam minimizados. Neste trabalho, alguns algoritmos heurísticos são aplicados ao problema. Em particular, são propostas três implementações baseadas em tabu search. As implementações foram executadas usando-se problemas-teste conhecidos na literatura, e uma delas, após ajustes específicos de parâmetros, conseguiu atingir a solução ótima de todos os problemas.

Palavras-chaves: Problema quadrático da alocação; Tabu Search; Tabu Thresholding.

Abstract: In the facility layout problem analyzed here, we wish to assign *n* facilities to *n* locations so that communication costs between facilities are minimized. In this work, some heuristic algorithms are applied to the problem. In particular, three implementations of tabu search are proposed. The implementations were run using test-problems known in the literature, and one of these implementations, after specific adjustments of parameters, reached the optimal solution for all of the problems.

Keywords: Quadratic assignment problem; Tabu Search; Tabu Thresholding.

1 Introdução

Em situações reais, o problema do layout de facilidades ocorre, por exemplo, no ambiente industrial quando máquinas devem ser dispostas no chão de fábrica. O problema ocorre também quando se deseja determinar a localização de unidades de tratamento em hospitais, de salas ou departamentos em empresas, de prédios em campi universitários, dentre outros.

Uma formulação comumente usada para o problema e que será adotada neste trabalho é aquela proposta por Koopmans e Beckman (1957), conhecida como problema quadrático da alocação [1].

A busca por soluções ótimas para o problema quadrático da alocação não é elementar, o que impulsionou o desenvolvimento de um grande número de algoritmos heurísticos para se alcançar soluções tão boas quanto possíveis, em um tempo computacional razoável.

Neste trabalho, apresentamos três algoritmos heurísticos baseados no método tabu search (Glover, 1989) e testamos nossos algoritmos utilizando problemas-teste conhecidos na literatura e cujos dados estão disponíveis na biblioteca QAPLIB (veja Burkard et. al., 1991). É realizada uma comparação entre os algoritmos para ajudar um usuário na escolha do melhor algoritmo para a sua situação, supondo-se duas situações: (a) Necessidade de soluções razoavelmente boas em tempos de CPU relativamente pequenos (b) Necessidade de soluções realmente boas admitindo-se tempos de CPU consideravelmente maiores.

[1] O termo "quadrático" se deve ao fato de a função objetivo ser uma função polinomial de segundo grau das variáveis do problema.

Nos problemas considerados em nosso estudo, a distância entre duas localidades é medida a partir de seus centros; O centro de cada facilidade coincide com o centro da localidade em que ela se encontra; Não há custo de alocação de uma facilidade em determinada localidade. Sejam:

n = número de facilidades (tamanho da instância do problema)
d_{ij} = distância entre a localidade i e a localidade j
f_{pq} = fluxo entre a facilidade p e a facilidade q

Dadas a matriz simétrica $D = (d_{ij})$ de tamanho $n \times n$ e a matriz simétrica $F = (f_{pq})$ de tamanho $n \times n$, e sendo $\prod_n$ o conjunto de todas as permutações dos elementos de $\{1, 2, 3,\ldots,n\}$, o problema pode ser escrito como segue:

onde $\pi(l)$ é a facilidade que ocupa a localidade l na disposição π.

Note que na formulação acima foi assumido que o número de facilidades é igual ao número de localidades. Entretanto, para alguns problemas, o número de facilidades m pode ser menor que o número de localidades n. Tais problemas ainda podem ser formulados como o QAP, com a introdução de $n - m$ facilidades artificiais, e atribuindo-se fluxos iguais a zero entre essas facilidades artificiais e todas as outras facilidades (Kusiak e Heragu, 1987).

Na próxima seção são apresentadas três implementações baseadas em Tabu Search. Na seção 3 há uma comparação entre os resultados obtidos pelos algoritmos anteriores e pelas já bem conhecidas heurísticas *2-opt* e *3-opt*. Finalmente, as conclusões são apresentadas na última seção.

2 Implementações do Tabu Search

O tabu search (Glover987) do Tf 0.0543 - 6páa úl10.9(aee)-0.1(itar a3.2065 -1.1413 TD0.0019 Tc0.6(l)-11ac

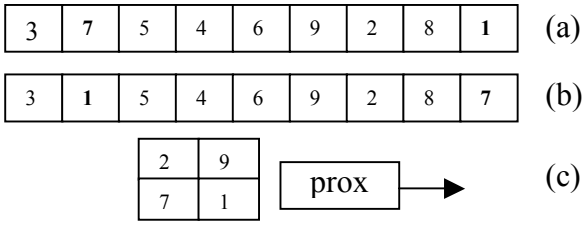

Figura 1. Estrutura da Lista Tabu

Observa-se que o algoritmo *TS1* baseia-se puramente na memória de curto prazo. Desta forma, foi implementada também uma versão do tabu search que utiliza memórias de curto e longo prazo. Esta versão foi denominada *TS2* e pode ser vista como o algoritmo *TS1* seguido por uma fase de diversificação. Essas duas fases – *memória de curto prazo* e *diversificação* – são repetidas *r* vezes durante a execução do algoritmo.

Para obter uma memória de longo prazo, define-se na fase de memória de curto prazo a matriz *LTM* de tamanho $n \times n$ tal que a cada permutação das facilidades i e j, $LTM(i, j)$ é incrementada. A partir dessa memória de longo prazo, implementamos diversificação por meio de dois procedimentos: no primeiro, utilizado nas $r / 2$ primeiras iterações, são escolhidas p facilidades randomicamente que são realocadas em posições para as quais não foram movidas freqüentemente nas soluções passadas (onde p é um parâmetro inteiro). O segundo procedimento, adotado nas $r / 2$ iterações finais, consiste em permutações das facilidades considerando apenas movimentos que levem cada facilidade para posições distantes daquelas ocupadas por ela na melhor solução encontrada na execução mais recente da fase de memória de curto prazo.

Para o algoritmo *TS2*, definimos parâmetros a e b que representam respectivamente os limites inferior e superior entre os quais será escolhido o número p de facilidades que serão realocadas, e o parâmetro r representa o número de repetições das duas fases do algoritmo.

O terceiro algoritmo implementado neste estudo baseia-se no método Tabu Thresholding (*TT*). Este consiste em duas fases que se alternam k vezes: uma fase de melhoramento e uma fase mista. Na *fase de melhoramento* são permitidos somente os movimentos que melhoram a solução atual, enquanto a *fase mista* aceita tanto os movimentos que melhoram quanto os que pioram a solução atual. A eficiência do critério de escolha de um movimento é fortemente influenciada por um parâmetro t que determina o número de iterações da fase mista. A escolha de t é feita randomicamente entre dois limites inferior e superior (denominados aqui de *LOW* e *UP* respectivamente), os quais devem ser ajustados adequadamente. O critério de parada da fase de melhoramento é o encontro de um ótimo local, enquanto o encerramento da fase mista é determinado meramente pelo parâmetro t.

A escolha de um movimento nas duas fases é governada pelo emprego da *estratégia de lista de candidatos* (Glover, 1992), para isolar subconjuntos de movimentos a serem examinados a cada iteração. Neste estudo, os subconjuntos foram definidos da seguinte forma: para n ímpar, tem-se n subconjuntos com $(n – 1) / 2$ elementos cada, e para n par, n subconjuntos com $(n – 2) / 2$ elementos e um subconjunto com $n / 2$ elementos. Consideramos movimentos tomados na ordem (1, 2), (1, 3), (1, 4), …, (n-1, n). Para compor os subconjuntos, os movimentos que definem a vizinhança de uma solução foram atribuídos seqüencialmente a cada subconjunto, ou seja, para n ímpar, por exemplo, os $(n – 1) / 2$ primeiros movimentos são atribuídos ao primeiro subconjunto, os $(n – 1) / 2$ movimentos seguintes são atribuídos ao segundo, e assim até que todos os subconjuntos sejam preenchidos.

3 Resultados Computacionais

Além dos algoritmos já discutidos, foram implementados os algoritmos de busca local *2-opt* e *3-opt* (Heragu e Alfa, 1992). Todos os algoritmos foram codificados na linguagem *C* e executados em um computador AMD Athlon 1.2 GHz.

Como já mencionado, considera-se dois cenários distintos: o *Cenário 1* envolve implementações voltadas para a obtenção de soluções razoáveis em tempos de CPU relativamente

pequenos. Abrange os algoritmos *2-opt*, *3-opt* e *TS1*. O *Cenário 2*, por sua vez, abrange implementações que têm por objetivo soluções realmente boas, o que implica tempos de CPU consideravelmente maiores do que aqueles relativos ao Cenário 1. Os algoritmos abordados neste caso são o *TT* e o *TS2,* que possuem uma fase de diversificação.

Os algoritmos estudados foram testados com 16 problemas de tamanhos entre $n = 19$ e $n = 30$ facilidades, propostos por Elshafei (1977), Krarup & Pruzan (1978) e Nugent et. al.(1968). Para cada um dos 16 problemas foram geradas 20 soluções aleatórias. Essas soluções foram usadas por todos os algoritmos como soluções iniciais para 20 execuções de cada problema, de forma que os valores médios, piores e melhores para a solução encontrada e o tempo de CPU, nas tabelas seguintes, referem-se a essas 20 execuções. A solução ótima ou melhor solução conhecida para os problemas-teste por nós utilizados está disponível na biblioteca QAPLIB.

3.1 Cenário 1 – Tempos de CPU menores

Após alguns testes foi possível fixar os valores *t_size* = 7 e *NUM_MAX* = 40, para o *TS1*. Esses valores foram mantidos para todos os testes envolvendo esse algoritmo, e são os mesmos usados anteriormente em testes com problemas de layout unidimensional (Malini & Amaral, 2002). Tal correspondência de parâmetros não foi observada nos algoritmos do Cenário 2, como será visto na próxima secção.

Na tabela 1, comparamos os métodos de busca local *2-opt* e *3-opt*. O algoritmo *2-opt* conseguiu soluções melhores e tempos de CPU menores para a maior parte dos problemas-teste, mostrando-se um método de busca local mais eficiente do que o *3-opt*.

Em nossos testes também utilizamos os algoritmos *2-opt* e *3-opt* para fornecer soluções

Tabela 1

Comparação das soluções obtidas pelos algoritmos *2-opt* e *3-opt* para o QAP
(20 execuções para cada problema)

		Melhor solução	***2-opt***			***3-opt***		
Problema	*n*	conhecida	Melhor	Média	Pior	Melhor	Média	Pior
Els19	19	17212548,00	17937024,00	21837830,20	26918794,00	17997928,00	22244588,60	27678056,00
Kra30a	30	88900,00	92770,00	95720,50	99100,00	93320,00	95944,50	98150,00
Nug12	12	578,00	590,00	613,10	640,00	590,00	604,30	626,00
Nug14	14	1014,00	1026,00	1062,40	1096,00	1030,00	1076,50	1114,00
Nug15	15	1150,00	1170,00	1209,20	1252,00	1170,00	1204,00	1262,00
Nug16a	16	1610,00	1612,00	1689,60	1762,00	1642,00	1696,10	1742,00
Nug16b	16	1240,00	1252,00	1315,90	1380,00	1252,00	1313,10	1380,00
Nug17	17	1732,00	1756,00	1813,40	1930,00	1736,00	1804,60	1868,00
Nug18	18	1930,00	1962,00	2010,80	2082,00	1970,00	2023,70	2074,00
Nug20	20	2570,00	2570,00	2679,50	2792,00	2608,00	2671,90	2754,00
Nug21	21	2438,00	2470,00	2540,10	2604,00	2480,00	2531,30	2602,00
Nug22	22	3596,00	3622,00	3718,70	3862,00	3642,00	3736,40	3952,00
Nug24	24	3488,00	3550,00	3655,40	3830,00	3526,00	3631,60	3724,00
Nug25	25	3744,00	3802,00	3894,80	4066,00	3778,00	3896,30	4066,00
Nug27	27	5234,00	5340,00	5477,30	5638,00	5344,00	5528,80	5790,00
Nug30	30	6124,00	6184,00	6374,30	6594,00	6226,00	6395,60	6578,00

iniciais ao *TS1*. Essas associações foram denominadas (*2-opt* + *TS1*) e (*3-opt* + *TS1*). A tabela A1 (Anexo 1) apresenta a comparação entre as soluções obtidas pelo *TS1* e pelas associações (*2-opt* +

TS1) e (*3-opt* + *TS1*). Os três algoritmos apresentaram resultados semelhantes, mas a associação (*2-opt* + *TS1*) obteve boas soluções no menor tempo, em média, para a maioria dos problemas-teste, confirmando-se como o melhor método dentre os testados no Cenário 1. A tabela A2 (Anexo 2) traz a comparação entre os tempos gastos pelos algoritmos testados.

3.2 Cenário 2 – Tempos de CPU maiores

Ao contrário do que ocorreu no Cenário 1, onde os parâmetros dos algoritmos testados foram os mesmos utilizados no estudo dos problemas de layout unidimensional, os testes no Cenário 2 exigiram uma nova busca por parâmetros ideais para o *TT* e o *TS2*. Observou-se que quanto maiores os parâmetros *LOW*, *UP*, *k*, *a*, *b*, e *r*, melhores as soluções encontradas e maiores os tempos de CPU requeridos.

O algoritmo *TT* não se mostrou tão eficiente para o QAP como foi para os problemas de layout unidimensional. Durante os testes, sentimos dificuldade para encontrar configurações de parâmetros que conduzissem a soluções satisfatórias, e mesmo aquelas apresentadas neste trabalho não produziram os resultados esperados. De fato, apesar de termos usado três configurações de parâmetros com tempos de execução consideravelmente maiores à medida que os parâmetros aumentavam, a melhora nas soluções providas por cada conjunto de parâmetros não foi proporcional a esse incremento no tempo de CPU despendido. Os resultados dos testes com o *TT* são apresentados na tabela 2.

Tabela 2

Resultados computacionais do algoritmo *TT* para o QAP - 3 configurações de parâmetros (valores médios em 20 execuções)

			TT					
		Melhor solução	*LOW* = 15, *UP* = 25 *k* = 20		*LOW* = 25, *UP* = 35 *k* = 30		*LOW* = 35, *UP* = 45 *k* = 40	
Problema	*n*	conhecida	Custo	Tempo	Custo	Tempo	Custo	Tempo
Els19	19	17212548,00	20135790,90	4,35	19805979,10	9,40	20228485,70	16,30
Kra30a	30	88900,00	91405,00	26,40	91477,50	56,10	91211,00	96,55
Nug12	12	578,00	584,00	0,75	583,00	1,65	582,80	2,85
Nug14	14	1014,00	1022,50	1,35	1020,00	2,95	1018,70	5,05
Nug15	15	1150,00	1158,00	1,70	1153,60	3,70	1152,80	6,55
Nug16a	16	1610,00	1625,30	2,20	1622,90	4,85	1618,30	8,45
Nug16b	16	1240,00	1247,50	2,25	1245,20	4,85	1241,10	8,40
Nug17	17	1732,00	1744,30	2,85	1741,50	6,00	1739,70	10,50
Nug18	18	1930,00	1953,40	3,55	1948,80	7,55	1945,30	13,20
Nug20	20	2570,00	2594,30	5,30	2590,20	11,45	2590,00	19,80
Nug21	21	2438,00	2459,30	6,40	2449,80	13,70	2449,60	23,80
Nug22	22	3596,00	3616,30	7,65	3610,80	16,40	3602,70	28,40
Nug24	24	3488,00	3525,90	10,90	3514,70	23,30	3514,70	40,20
Nug25	25	3744,00	3773,30	12,95	3763,10	27,40	3762,80	46,95
Nug27	27	5234,00	5276,10	17,25	5266,90	36,65	5253,10	63,30
Nug30	30	6124,00	6190,90	26,45	6193,10	55,90	6177,70	97,00

Os testes envolvendo o algoritmo *TS2* também exigiram novos parâmetros com relação àqueles aplicados aos problemas de layout unidimensional. Após diversos testes, redefinimos *NUM_MAX* igual a 100 e *t_size* igual a 30, e esses valores foram mantidos durante todo o nosso estudo. A tabela 3 apresenta os resultados computacionais do *TS2* para o QAP.

Na tabela A3 (Anexo 3) apresentamos uma comparação direta entre os métodos *TT* e *TS2*. Para o primeiro, usamos os parâmetros *LOW* = 35, *UP* = 45 e *k* = 40, e para o segundo, *a* = 2 * (*n* / 3), *b* = *n* e *r* = 10. Diferentemente do que ocorreu nos testes com problemas unidimensionais, onde *TT* e *TS2* apresentaram resultados muito próximos, para o QAP o *TS2* saiu-se melhor que o *TT*. Podemos notar que, embora os tempos de CPU dos dois algoritmos tenham sido praticamente os mesmos para todos os problemas, as soluções encontradas pelo *TS2* foram em média melhores do que aquelas obtidas pelo *TT*, para todos os problemas-teste.

Tabela 3

Resultados computacionais do algoritmo *TS2* para o QAP - 3 configurações de parâmetros (valores médios em 20 execuções)

			TS2					
		Melhor	$a = 1, b = n/3$		$a = n/3, b = 2*(n/3)$		$a = 2*(n/3), b = n$	
		solução	$r = 5$		$r = 7$		$r = 10$	
Problema	*n*	conhecida	Custo	Tempo	Custo	Tempo	Custo	Tempo
Els19	19	17212548,00	18604682,50	5,30	18479821,10	7,25	17913869,50	10,60
Kra30a	30	88900,00	90644,00	48,60	90517,00	61,75	90304,00	86,75
Nug12	12	578,00	578,40	1,05	578,00	1,45	578,00	2,05
Nug14	14	1014,00	1014,90	2,10	1014,90	3,10	1014,50	4,20
Nug15	15	1150,00	1151,00	2,50	1150,50	3,60	1150,70	5,25
Nug16a	16	1610,00	1610,90	3,80	1610,10	5,55	1610,00	7,80
Nug16b	16	1240,00	1240,00	3,10	1240,00	4,65	1240,00	6,55
Nug17	17	1732,00	1734,60	4,75	1733,10	6,50	1732,90	9,60
Nug18	18	1930,00	1935,70	6,65	1934,50	9,10	1932,10	12,85
Nug20	20	2570,00	2575,90	8,45	2573,10	11,80	2572,90	17,00
Nug21	21	2438,00	2446,50	10,90	2442,00	15,60	2441,00	22,45
Nug22	22	3596,00	3616,70	11,25	3606,40	15,80	3601,70	21,75
Nug24	24	3488,00	3497,00	18,30	3495,10	25,55	3492,30	38,25
Nug25	25	3744,00	3746,40	24,65	3744,50	33,80	3745,10	47,80
Nug27	27	5234,00	5267,40	32,05	5254,20	45,20	5242,90	64,70
Nug30	30	6124,00	6147,80	52,50	6145,00	71,10	6138,40	98,25

4 Conclusões

Neste artigo, foram comparados cinco algoritmos – *2-opt, 3-opt, TS1, TS2* e *TT*. Esta pesquisa foi estruturada de maneira a permitir uma análise separada dos algoritmos que visam soluções razoáveis e pouco tempo de CPU (Cenário 1), e daqueles que buscam soluções realmente boas e demandam maior tempo de CPU (Cenário 2).

No primeiro cenário foi possível manter, para o *TS1*, os mesmos parâmetros utilizados em testes anteriores envolvendo problemas de layout unidimensional. A associação entre os algoritmos *2-opt* e *TS1* destacou-se por apresentar a melhor relação *solução média x tempo de CPU* para os problemas-teste. No Cenário 2, onde foi necessária uma nova busca pelos melhores parâmetros para os algoritmos, o *TS2* obteve os melhores resultados em média. De fato, esse algoritmo atingiu a solução ótima de todos os problemas-teste, conhecidos na literatura.

Referências Bibliográficas

Burkard, R.E., Karisch, S.E., Rendl, F. (1991), "QAPLIB: A Quadratic Assignment Problem Library", *European Journal of Operational Research* 55, 115-119.

Elshafei, A.N. (1977), "Hospital layout as a quadratic assignment problem", *Operations Research Quarterly* 28, 167-179.

Glover, F. (1989), "Tabu Search, Part I", *European Journal of Operational Research* 1 / 3, 190-206.

Glover, F. (1992), "Simple Tabu Thresholding in Optimization", *University of Colorado, Boulder*.

Heragu, S.S., Alfa, A.S. (1992), "Experimental analysis of simulated annealing based algorithms for the layout problem", *European Journal of Operational Research* 57, 190-202.

Koopmans, T.C., Beckmann, M. (1957), "Assignment problems and the location of economic activities", *Econometrica* 25 / 1, 53-76.

Krarup, J., Pruzan, P.M. (1978), "Computer-aided layout design", *Mathematical Programming Study* 9, 75-94.

Kusiak, A., Heragu, S.S. (1987), "The facility layout problem", *European Journal of Operational Research* 29 / 3, 229-253.

Malini, R.Q., Amaral, A.R.S. (2002), "O caso unidimensional do problema do layout de facilidades: comparação entre métodos heurísticos de resolução", trabalho submetido ao Congresso Brasileiro de Automática – CBA 2002. Natal / RN, de 02 a 05 de Setembro de 2002.

Nugent, C.E., Vollmann, T.E., Ruml, J. (1968), "An experimental comparison of techniques for the assignment of facilities to locations", *Operations Research* 16, 150-173.

Tabela A1

Comparação das soluções obtidas pelos algoritmos *TS1* , *2-opt + TS1* e *3-opt + TS1* para o QAP (20 execuções para cada problema)

		Melhor	***TS1***			***2-opt + TS1***			***3-opt + TS1***		
Problema	*n*	solução conhecida	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
Els19	19	17212548,00	17374134,00	21674816,50	27901498,00	17937024,00	21632930,10	25325320,00	17997928,00	22037437,70	27614198,00
Kra30a	30	88900,00	91100,00	94381,00	96690,00	90160,00	94563,00	97480,00	90960,00	94372,00	97350,00
Nug12	12	578,00	578,00	586,70	596,00	578,00	583,90	592,00	578,00	583,40	586,00
Nug14	14	1014,00	1014,00	1032,60	1058,00	1014,00	1031,60	1056,00	1014,00	1036,80	1058,00
Nug15	15	1150,00	1150,00	1163,60	1236,00	1150,00	1161,60	1206,00	1150,00	1157,60	1186,00
Nug16a	16	1610,00	1610,00	1638,80	1684,00	1612,00	1635,80	1674,00	1610,00	1637,30	1674,00
Nug16b	16	1240,00	1240,00	1258,40	1308,00	1240,00	1257,70	1302,00	1240,00	1250,40	1282,00
Nug17	17	1732,00	1734,00	1757,90	1802,00	1732,00	1749,30	1778,00	1734,00	1758,00	1824,00
Nug18	18	1930,00	1936,00	1957,80	1986,00	1930,00	1957,10	2006,00	1938,00	1962,40	2006,00
Nug20	20	2570,00	2570,00	2602,50	2646,00	2570,00	2608,30	2654,00	2570,00	2605,50	2654,00
Nug21	21	2438,00	2438,00	2470,60	2516,00	2438,00	2489,60	2578,00	2438,00	2473,80	2532,00
Nug22	22	3596,00	3614,00	3675,80	3866,00	3596,00	3658,10	3854,00	3596,00	3676,30	3858,00
Nug24	24	3488,00	3488,00	3553,40	3678,00	3488,00	3556,90	3638,00	3488,00	3540,60	3608,00
Nug25	25	3744,00	3744,00	3783,30	3888,00	3744,00	3783,80	3874,00	3746,00	3784,20	3854,00
Nug27	27	5234,00	5234,00	5368,70	5586,00	5234,00	5352,90	5564,00	5234,00	5339,30	5494,00
Nug30	30	6124,00	6128,00	6222,30	6386,00	6148,00	6236,00	6384,00	6124,00	6234,20	6342,00

Tabela A2

Comparação dos tempos de CPU dos algoritmos *2-opt* , *3-opt* , *TS1* e combinações destes para o QAP (20 execuções para cada problema)

		2-opt		***3-opt***		***TS1***		***2-opt + TS1***		***3-opt + TS1***	
Problema	*n*	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
Els19	19	0,05	0,05	0,45	0,26	0,45	0,26	0,50	0,26	0,70	0,22
Kra30a	30	0,75	0,20	2,15	0,45	4,20	1,54	3,65	0,98	5,65	1,71
Nug12	12	0,00	0,00	0,00	0,00	0,10	0,09	0,05	0,05	0,10	0,09
Nug14	14	0,00	0,00	0,05	0,05	0,20	0,17	0,20	0,17	0,20	0,17
Nug15	15	0,05	0,05	0,05	0,05	0,25	0,20	0,25	0,20	0,30	0,22
Nug16a	16	0,00	0,00	0,10	0,09	0,40	0,25	0,30	0,22	0,40	0,25
Nug16b	16	0,00	0,00	0,10	0,09	0,30	0,22	0,35	0,24	0,35	0,24
Nug17	17	0,05	0,05	0,05	0,05	0,45	0,26	0,45	0,26	0,45	0,26
Nug18	18	0,15	0,13	0,05	0,05	0,50	0,26	0,50	0,26	0,65	0,24
Nug20	20	0,05	0,05	0,25	0,20	0,90	0,41	0,90	0,41	1,20	0,59
Nug21	21	0,15	0,13	0,25	0,20	1,15	0,34	0,95	0,05	1,15	0,24
Nug22	22	0,15	0,13	0,40	0,25	1,20	0,17	1,10	0,20	1,45	0,37
Nug24	24	0,25	0,20	0,60	0,25	1,95	0,47	1,95	0,68	2,35	0,45
Nug25	25	0,10	0,09	0,85	0,13	2,55	1,10	2,20	0,48	2,60	0,46
Nug27	27	0,40	0,25	1,15	0,13	3,55	1,31	3,30	1,17	4,20	1,01
Nug30	30	0,65	0,24	2,00	0,32	5,85	2,98	5,80	3,43	6,70	4,85

Tabela A3

Comparação de perfomance entre os algoritmos *TT* e *TS2* para o QAP - parâmetros mais altos
(20 execuções para cada problema)

		Melhor	**Custo**						**Tempo**	
		solução	Melhor		Média		Pior		**médio**	
Problema	*n*	conhecida	*TT*	*TS2*	*TT*	*TS2*	*TT*	*TS2*	*TT*	*TS2*
Els19	19	17212548,00	17997928,00	17212548,00	20228485,70	17913869,50	28702028,00	19332886,00	16,30	10,60
Kra30a	30	88900,00	88900,00	88900,00	91211,00	90304,00	92090,00	92390,00	96,55	86,75
Nug12	12	578,00	578,00	578,00	582,80	578,00	590,00	578,00	2,85	2,05
Nug14	14	1014,00	1014,00	1014,00	1018,70	1014,50	1026,00	1016,00	5,05	4,20
Nug15	15	1150,00	1150,00	1150,00	1152,80	1150,70	1160,00	1152,00	6,55	5,25
Nug16a	16	1610,00	1610,00	1610,00	1618,30	1610,00	1632,00	1610,00	8,45	7,80
Nug16b	16	1240,00	1240,00	1240,00	1241,10	1240,00	1262,00	1240,00	8,40	6,55
Nug17	17	1732,00	1732,00	1732,00	1739,70	1732,90	1748,00	1734,00	10,50	9,60
Nug18	18	1930,00	1930,00	1930,00	1945,30	1932,10	1960,00	1936,00	13,20	12,85
Nug20	20	2570,00	2570,00	2570,00	2590,00	2572,90	2618,00	2588,00	19,80	17,00
Nug21	21	2438,00	2438,00	2438,00	2449,60	2441,00	2466,00	2456,00	23,80	22,45
Nug22	22	3596,00	3596,00	3596,00	3602,70	3601,70	3618,00	3632,00	28,40	21,75
Nug24	24	3488,00	3490,00	3488,00	3514,70	3492,30	3536,00	3510,00	40,20	38,25
Nug25	25	3744,00	3744,00	3744,00	3762,80	3745,10	3794,00	3750,00	46,95	47,80
Nug27	27	5234,00	5234,00	5234,00	5253,10	5242,90	5284,00	5290,00	63,30	64,70
Nug30	30	6124,00	6146,00	6124,00	6177,70	6138,40	6222,00	6158,00	97,00	98,25