

Investigação da Hibridização de Path Relinking com Busca Local e Busca Local Iterativa para o Problema de Alocação Quadrática

Giovanne da Silva Santos
giovannedssantos@gmail.com

13 de janeiro de 2021

Abstract

The local search, iterative local search and path relinking are simple and powerful methods, and this article applies to the Quadratic Assignment Problem solving methods. In addition, it uses them according to the final study and research methods and methods. of results. captured relative to the runtime.

Keywords: Local search, Iterative local search, Quadratic Assignment Problem, Path-Relinking

Resumo

A busca local, busca local iterativa e o path-relinking são métodos simples e poderosos, e neste artigo aplica-se aos métodos de resolução de problemas do Problema Quadrático de Alocação. Além disso, utiliza-se de acordo com os métodos finais de estudo e investigações de resultados. capturado em relação ao tempo de execução.

Palavras-chaves: Busca Local, Busca Local Iterativa, Problema Quadrático de Alocação, Path-Relinking

1 Introdução

A humanidade, em seu progresso e evolução, sempre procurou o melhor desempenho e rendimento na resolução de seus problemas em detrimento da limitação de recursos. Um dos exemplos disso seria a organização de um layout de um hospital em que é considerado os setores hospitalares como, por exemplo, UTI's, leitos, farmácias e enfermarias, e o objetivo é que a distância percorrida pelo fluxo de materiais e pessoas entre esses setores seja minimizada. Esta situação, assim como outras aplicações importantes, é modelada pelo Problema Quadrático de Alocação (PQA). A introdução deste modelo para hospitais foi aplicado no trabalho de [Elshafei (1977)]. Além disso, há outras aplicações como de modelagem de construções em universidades no trabalho de Dickey and Hopkins (1972), de design de painéis de controles em Burkard and Offermann (1977) e entre outros problemas.

O PQA consiste em alocar, de forma ótima, determinado conjunto de itens (com o custo definido pelo fluxo do transporte desses itens) em determinadas localidades (com o custo definido pela distância desses locais). O objetivo é encontrar a melhor alocação dos itens nas localidades de tal forma que a soma do produto do custo dos fluxos e o custo das distâncias seja mínima.

O Problema Quadrático de Alocação é um problema de otimização *NP*-árduo [Sahni and Gonzalez (1976)], sendo um dos mais difíceis de ser resolvido, todavia, atualmente, há grandes instâncias (tamanho 30) resolvidas por algoritmos exatos [Anstreicher et al. (2002)]. As instâncias utilizadas estão disponíveis na biblioteca QAPLIB.

Neste artigo, primeiramente na seção 2 realiza-se citações de trabalhos referenciados para o desenvolvimento das técnicas implementadas. Ademais, mostro os princípios das técnicas e aplicação delas com modificações realizadas através de pseudo-códigos e explicações feitos na seção 3. Em seguida, em seção 4 refiro o ambiente de testes. Na seção 5 exponha-se os resultados obtidos com as instâncias e análises feitas. Por último, na seção 6 termina-se o artigo com análises finais dos experimentos e técnicas aplicadas.

2 Estado da arte

A busca local foi estudada e aplicada a partir dos dados apresentados por Stützle (1999), na qual dá um panorama geral de outras técnicas, já em relação a busca local explica o seu funcionamento assim como o algoritmo.

Para a busca local iterativa temos, além do Stützle (1999) que explica os fundamentos desta técnica: Gutiérrez and Brizuela (2006) que utiliza a busca local iterativa para achar e analisar o "Big Valley-local onde concentra-se os melhores resultados- a fim aplicar boas perturbações dependendo do tamanho,

tendo em vista que para instâncias menores(até tamanho 30) é melhor usar perturbações pequenas, já para instância maiores(maiores que 30) é mais apropriado usar maiores perturbações; Stützle (2006) usa a forma básica da busca local iterativa para mostrar a estagnação dessa técnica, principalmente, por conta do critério de comparação entre a solução atual e o vizinho à ela ser simplório e fraco em determinadas instâncias, porém ele mostra e testa outras formas mais refinadas para determinar qual a melhor solução, logo torna a busca local iterativa mais eficiente e poderoso; Hussin and Stützle (2009) hibridiza a busca local iterativa com outra busca local iterativa, assim criando uma hierarquia entre as buscas locais iterativas e incentivando a hibridização, também, entre técnicas diferentes. Goldbarg and Luna (2005)

O path-relinking foi estudo e baseado pelo artigos de Ribeiro (2012) e Glover et al. (2004). Em Ribeiro (2012) é apresentado o uso do path-relinking para melhorar, com hibridização, as técnicas de heurísticas - como busca tabu, algoritmo genético e GRASP - a fim de resolver problemas de otimização combinatórias. Já em Glover et al. (2004), já são introduzidas novas ideias e aplicações para o path-relinking com a finalidade de mostrar o impacto e performance desta técnica em aplicações como rede neural, rotina clássica de veículos e problemas multi-objetivos.

3 Algoritmos

Nesta seção, temos os algoritmos desenvolvidos pelo autor, a explicação de cada procedimento e o funcionamento deles. Na seção 3.1 temos a explicação do algoritmo desenvolvido para a busca local. Já na seção 3.2 é descrito o algoritmo da busca local iterativa.

3.1 Busca Local

A busca local é um método simples, porém poderoso. Ele consiste em encontrar uma solução a partir de uma vizinhança de uma solução qualquer(um candidato) a fim de achar uma solução melhor - o que passa a ser o novo candidato. Aplica-se esse princípio até não ter como melhorar a solução localmente.

Segue o pseudo-algoritmo desenvolvido para a busca local:

Algorithm 1 Busca Local

```

1: procedure BUSCA LOCAL
2:    $s_0 \leftarrow \text{GENERATEINITIALSOLUTION}()$  ▷ Criação da primeira solução
3:    $best \leftarrow s_0$ 
4:    $candidate \leftarrow \text{BESTNEIGHBOR}(s_0)$  ▷ Gerando o melhor vizinho da primeira solução
5:   repeat
6:     if  $candidate < best$  then
7:        $best \leftarrow candidate$ 
8:        $s_0 \leftarrow candidate$ 
9:     else if  $candidate < s_0$  then
10:       $s_0 \leftarrow candidate$ 
11:    end if
12:     $candidate \leftarrow \text{BESTNEIGHBOR}(candidate)$ 
13:  until Não há mais vizinhos melhores do que  $best$ 
14:  return  $best$ 
15: end procedure

```

O algoritmo inicializa, na linha 2, com GENERATEINITIALSOLUTION na qual cria uma solução randomizando todos os valores. Na linha 4 temos o BESTNEIGHBOR que gera as vizinhanças a partir da técnica 2swap na qual consiste em trocar a posição de dois valores seguintes. Ao entrar no laço, na linha 5, temos: o percorrimento pela vizinhança investigando se o candidato (*candidate*) tem menor custo do que a melhor solução (*best*) na linha 6; na linha 7, caso seja a melhor possível, então *candidate* torna-se a solução *best*; na linha 9, caso não, porém melhor do que s_0 , então a solução atual torna-se *candidate*, mas caso *candidate* seja a pior, então escolhe-se o melhor vizinho dele, linha 12. Na linha 13 temos o final do laço no qual ocorre até que não haja mais vizinhos melhores do que *best*.

3.2 Busca Local Iterativa

Na busca local iterativa, há aplicação iterativamente da busca local com as mudanças de alguns parâmetros. Diante disso, deve-se regulamentar a geração procedural do ponto de partida, perturbações que modificam novas soluções iniciais da busca local, critérios que dão continuidade à iteração e a própria implementação da busca local. O Algorithm 1 é um pseudo-algoritmo representativo da busca local iterativa desenvolvida:

Algorithm 2 Busca Local Iterativa

```
1: procedure BUSCA LOCAL ITERATIVA
2:    $s_0 \leftarrow \text{GENERATEINITIALSOLUTION}()$  ▷ Criação da primeira solução
3:    $best \leftarrow \text{BUSCALOCAL}(s_0)$  ▷ A melhor solução atual a partir da Busca Local
4:    $candidate \leftarrow \text{BUSCALOCAL}(best)$  ▷ Candidato a partir da Busca Local da melhor
5:    $k \leftarrow 0$ 
6:   repeat
7:     if  $candidate < best$  then
8:        $best \leftarrow candidate$ 
9:     else
10:       $k \leftarrow k + 1$ 
11:    end if
12:     $\text{PERTUBATION}(candidate, k)$ 
13:     $candidate \leftarrow \text{BUSCALOCAL}(candidate)$ 
14:  until Não poder mais realizar a perturbação
15:  return  $best$ 
16: end procedure
```

Na linha 2, há a geração de solução inicial com `GENERATEINITIALSOLUTION`, na qual tem a mesma funcionalidade que o Algorithm 1, para a partir dela: na linha 3, aplicarmos uma busca local supondo que seja a melhor solução; criamos um candidato a partir da busca local em $best$, na linha 4; na linha 6, entrarmos no laço no qual se o $candidate$ é melhor, então $candidate$ passa a ser a melhor solução, linha 7 e 8 -, senão, então aumenta-se o tamanho de k (tamanho do bloco dentro da solução no qual é aplicado as trocas dos elementos, e que foi inicializado na linha 5); aplica-se a perturbação em $candidate$ na linha 12, assim criando uma nova solução candidata e repetindo o processo, colocado na linha 13, até k (o bloco) torna-se do mesmo tamanho que a $candidate$ como representado na linha 14. Não aplica-se a estratégia de *multi-start*, por isso a permutação não cobre toda a solução.

3.3 Path-relinking

O path-relinking é uma técnica que gera soluções baseadas em duas soluções, uma sendo inicial e outra final. De modo geral, a partir da inicial ao mover um valor, que na solução final está em outra posição, para a posição na qual o mesmo valor está na solução final, assim criando uma nova solução e analisando qual é a melhor. Novas soluções serão geradas até que não tenha mais como mover os valores, ou seja, a inicial seja igual a final, ou até que certa porcentagem arbitrária da inicial seja modificada.

Considere a criação de caminhos que unem duas soluções selecionadas x' e x'' produzindo uma solução $x' = x(1), x(2), x(3), \dots, x(k) = x''$. Com a finalidade de diminuir a quantidade de opções, a solução $x(i+1)$ deve ser criada a partir de $x(i)$ e a cada passo escolhe-se um movimento que reduza a quantidade de passos para chegar a x'' Glover et al. (2000).

Vale ressaltar que, para o path-relinking, é necessário uma diferença simétrica de 4 componentes entre as soluções ou como denominada de distância *Hamming* para que haja um mínimo local entre elas. Isso é descrito e demonstrado em Ribeiro (2012).

O Algorithm 3 é um pseudo-algoritmo que representa o path-relinking (do tipo *forward relinking*):

Algorithm 3 Path Relinking

```
1: procedure PATH RELINKING(SoluçãoInicial, SoluçãoFinal)
2:    $\Delta \leftarrow \text{DistnciaSimetrica}(SolucaoInicial, SolucaoFinal)$ 
3:    $best \leftarrow \text{MelhorSolucao}(SolucaoInicial, SolucaoFinal)$ 
4:   while  $\Delta \neq 0$  do
5:      $\text{Movimento}(SolucaoInicial)$ 
6:      $best \leftarrow \text{MelhorSolucao}(best, SolucaoInicial)$ 
7:   end while
8: end procedure
```

Nesse pseudo-algoritmo, temos na linha 2 o cálculo da diferença simétrica entre as duas soluções passadas como parâmetro para saber quantas soluções serão geradas e testadas e depois escolhe a melhor solução entre as duas para guardar em $best$. Em seguida, dentro do laço, é usado a função *Movimento* que modifica a solução inicial baseada na solução final - movendo um valor para a mesma posição que este valor se encontra na solução final (guia). Na linha 6, é atualizado a melhor solução. No caso, esse pseudo-código representa um simples path-relinking que não se preocupa em começar com a melhor ou pior solução como a *forward* ou *backward*. Entretanto será experimentado o path-relinking *forward*, *backward*, *back and forward* e o *mixed*. Além disso, será usado o *truncated relinking* no *forward* e *backward*.

4 Metodologia

O ambiente de teste foi um notebook Lenovo com processador Intel® Core™ i7-7500U CPU @ 2.70GHz 4, memória RAM de 3,6 GB e HD de 500 GB e sendo o sistema operacional o Ubuntu 18.0.

Foram realizados 30 execuções por instâncias para cara algoritmo. Além disso, o cálculo utilizado para o desvio percentual(distanciamento) da solução encontrada em relação a melhor solução conhecida é:

$$\text{distanciamento} = (((\text{custo da solução encontrada} - \text{custo da melhor solução}) * 100) / \text{custo da melhor solução})$$

Ademais, os algoritmos comparados são a busca local e a busca local iterativa. As instâncias utilizadas foram as conhecidas da literatura retiradas da QAPLIB. Sendo essas correspondentes às quatro classes de instâncias com a finalidade da abranger todos os tipos. Vale salientar que no nome das instâncias encontra-se o tamanho delas, por exemplo, na instância tai20a o tamanho dela é 20.

5 Experimentos

Nesta seção, analisaremos os experimentos ,sendo, a primeira tabela (Tabela 1) o teste realizado com busca local e a segunda tabela (Tabela 2)com busca local iterativa. Além disso, foram realizados 30 testes por instâncias e técnicas.A respeito das colunas temos: AET o tempo médio decorrido para resolução da instância;POSF o percentual das soluções ótimas encontradas em relação o número de testes; DMin e DMax representam o menor e o maior distanciamento ,respectivamente, entre a solução e a solução ótima ; DAFC média do distanciamento da solução em relação a solução ótima; SDC o desvio padrão dessas distâncias ; Median a mediana do custo das soluções encontradas entre os 30 testes e BSK o custo da melhor solução conhecida.

5.1 Classe I

INST	AET(s)	POSF(%)	DMin(%)	DMax(%)	DAFC(%)	SDC	Median	BSK
tai20a	0.00391874	0.00	0.32	18.97	5.93	2.92	5.60	703482.00
tai25a	0.00581294	0.00	0.42	6.88	5.38	0.98	5.37	1167256.00
tai30a	0.0120661	0.00	0.62	7.27	5.09	1.00	4.96	1818146.00
tai60a	0.285744	0.00	0.32	12.18	11.34	0.47	11.33	8524308.00
tai80a	1.39979	0.00	0.43	10.78	10.12	0.44	10.18	15637278.00
rou20	0.00237755	0.00	0.53	13.39	4.65	2.20	4.23	725522.00

Tabela 1: Experimentos realizados pela busca local da classe I

INST	AET(s)	POSF(%)	DMin(%)	DMax(%)	DAFC(%)	SDC	Median	BSK
tai20a	0.0207888	0.00	1.93	4.74	3.14	0.69	3.13	703482.00
tai25a	0.0662049	0.00	1.02	4.63	3.40	0.61	3.40	1167256.00
tai30a	0.205561	0.00	2.48	4.13	3.11	0.52	3.05	1818146.00
tai60a	12.415	0.00	10.64	13.04	12.53	0.19	12.51	8524308.00
tai80a	77.121	0.00	8.86	11.36	10.98	0.15	10.99	15637278.00
rou20	0.0214118	0.00	1.27	4.09	2.40	0.79	2.29	725522.00

Tabela 2: Experimentos realizados pela busca local iterativa da classe I

INST	AET(s)	POSF(%)	DMin(%)	DMax(%)	DAFC(%)	SDC	Median	BSK
tai20a	0.0388733	0.00	1.73	4.55	2.91	0.83	2.96	703482.00
tai25a	0.140411	0.00	0.54	3.99	2.93	0.50	2.95	1167256.00
tai30a	0.4111	0.00	1.84	3.83	3.05	0.41	3.08	1818146.00
tai60a	24.927	0.00	7.32	13.03	12.65	0.18	12.65	8524308.00
tai80a	171.211	0.00	1.34	3.77	2.30	0.67	2.18	818415043.00
rou20	0.0407124	0.00	0.45	3.33	1.61	0.89	1.49	725522.00

Tabela 3: Experimentos realizados pelo path-relinking mixed + busca local iterativa da classe I

Nesta classe,que são instâncias com estruturas randomicamente geradas, temos uma porcentagem nula em relação ao número de vezes em que foram encontradas a melhor solução.Em relação ao tempo de resolução, houve uma maior demanda de tempo na busca local iterativa, o que é totalmente natural tendo em vista da busca local iterativa chamar várias vezes a busca local,porém na busca local iterativa

foi achado custos menores ao observar os distanciamentos mínimos e máximos, além da mediana dos custos. Já com a hibridização do path-relinking e da busca local iterativa, tivemos médias de distanciamentos um pouco melhores (DAFC) e a mediana de custos mais aproximados do menor custo possível (BSK) em detrimento ao aumento do tempo de execução, o que acaba sendo uma troca de vantagens e desvantagens. O fato de ser uma classe com instância randomizada, demonstra de forma mais abrangente e clara a melhoria da busca local iterativa em relação a uma busca local simples na procura de melhores soluções. Ademais, o desvio padrão da busca local iterativa é bem menor do que a busca local, mostrando, assim, que a iterativa achou custos mais baixos com mais consistência.

5.2 Classe II

INST	AET(s)	POSF(%)	DMin(%)	DMax(%)	DAFC(%)	SDC	Median	BSK
nug30	0.0188824	0.00	1.67	12.83	4.95	148.55	6418	6124
tho30	0.134379	0.00	19.63	29.38	26.10	4394.48	214826	158018
sko42	0.0737271	0.00	2.34	12.46	4.16	270.35	16432	15812
sko49	0.138933	0.00	1.71	7.60	3.75	301.46	24210	23386
sko56	0.301476	0.00	1.09	7.73	3.57	505.40	35546	34458
sko64	0.630744	0.00	1.45	8.37	3.44	708.51	50024	48498

Tabela 4: Experimentos realizados pela busca local da classe II

INST	AET(s)	POSF(%)	DMin(%)	DMax(%)	DAFC(%)	SDC	Median	BSK
nug30	0.238431	0.00	0.69	2.58	1.71	34.31	6224	6124
tho30	0.248935	0.00	27.42	29.76	28.87	1441.15	214826	152374
sko42	2.15746	0.00	0.95	2.66	1.73	67.72	16082	15812
sko49	4.78041	0.00	0.94	2.60	1.52	87.39	23740	23386
sko56	10.8607	0.00	0.98	2.21	1.48	112.58	34970	34458
sko64	25.9897	0.00	0.82	1.88	1.31	139.57	49154	48498

Tabela 5: Experimentos realizados pela busca local iterativa da classe II

INST	AET(s)	POSF(%)	DMin(%)	DMax(%)	DAFC(%)	SDC	Median	BSK
nug30	0.513272	0.00	0.33	2.45	1.39	32.79	6204	6124
tho30	0.510496	0.00	28.60	29.85	29.20	754.67	152064	214826
sko42	4.27187	0.00	0.56	2.10	1.32	68.55	16030	15812
sko49	10.1265	0.00	0.64	2.17	1.32	84.56	23688	23386
sko56	22.9892	0.00	0.94	2.00	1.38	80.41	34930	34458
sko64	51.2232	0.00	0.70	1.65	1.28	141.18	49168	48498

Tabela 6: Experimentos realizados pelo path-relinking mixed + busca local iterativa da classe II

Nesta classe, que apresenta instâncias com distâncias em forma de retângulos, temos novamente uma aproximação da solução ótima nula. Como na classe I, temos um maior tempo de execução com a busca local iterativa, porém a diferença é bem maior embora o tamanho das instâncias serem menores do que as instâncias da classe anterior. Além disso, com a hibridização do path-relinking, a aproximação com a solução ótima, como mostrado nas colunas DMin, DMax e DAFC, foi maior e a mediana do custo das soluções demonstra a melhora das soluções encontradas principalmente com o path-relinking mixed e backward. Vale ressaltar que o mixed, além de trazer melhores soluções, foi executado mais rápido em relação ao back-and-forward.

5.3 Classe III

INST	AET(s)	POSF(%)	DMinFC(%)	DMaxFC(%)	DAFC(%)	SDC	Median	BSK
bur26a	0.00686645	3.23	0.00	0.62	0.33	0.14	0.34	5426670.00
bur26b	0.00561871	0.00	0.21	1.30	0.43	0.23	0.36	3817852.00
bur26c	0.00724416	0.00	0.23	2.74	0.44	0.55	0.26	5426795.00
bur26d	0.00704848	0.00	0.23	1.34	0.47	0.38	0.34	3821225.00
els19	0.00149694	3.23	0.00	40.18	22.48	12.34	21.29	17212548.00
kra30a	0.0102534	0.00	25.43	31.34	28.05	1.54	27.94	134770.00
kra30b	0.00974374	0.00	26.43	29.65	27.32	1.60	27.52	134180.00
ste36b	0.0314973	0.00	23.12	62.65	26.70	15.97	19.75	15852.00

Tabela 7: Experimentos realizados pela busca local da classe III

INST	AET(s)	POSF(%)	DMinFC(%)	DMaxFC(%)	DAFC(%)	SDC	Median	BSK
bur26a	0.112311	3.23	0.00	0.20	0.11	0.05	0.11	5426670.00
bur26b	0.108958	0.00	0.03	0.24	0.16	0.07	0.18	3817852.00
bur26c	0.113171	3.23	0.00	0.25	0.03	0.06	0.01	5426795.00
bur26d	0.113468	0.00	0.02	0.37	0.04	0.08	0.01	3821225.00
els19	0.0177531	6.45	0.00	17.74	5.59	5.67	4.21	17212548.00
kra30a	0.217131	0.00	30.49	33.00	31.46	0.82	31.38	134770.00
kra30b	0.218566	0.00	29.34	31.67	30.26	0.73	30.25	134180.00
ste36b	0.833402	0.00	5.23	13.44	7.69	3.33	8.23	15852.00

Tabela 8: Experimentos realizados pela busca local iterativa da classe III

INST	AET(s)	POSF(%)	DMinFC(%)	DMaxFC(%)	DAFC(%)	SDC	Median	BSK
bur26a	0.225068	9.68	0.00	0.16	0.08	0.05	0.09	5426670.00
bur26b	0.221412	6.45	0.00	0.21	0.09	0.08	0.04	3817852.00
bur26c	0.225503	9.68	0.00	0.10	0.01	0.02	0.01	5426795.00
bur26d	0.221594	12.90	0.00	0.02	0.01	0.01	0.01	3821225.00
els19	0.0344143	51.61	0.00	6.13	1.18	1.86	0.00	17212548.00
kra30a	0.427539	0.00	28.34	32.71	31.75	0.61	31.86	134770.00
kra30b	0.432758	0.00	28.52	31.50	30.40	0.46	30.29	134180.00
ste36b	1.6341	0.00	3.25	11.05	5.73	2.94	5.79	15852.00

Tabela 9: Experimentos realizados pelo path-relinking mixed + busca local iterativa

Na classe das instâncias de situações reais, temos finalmente alguma soluções ótimas encontradas nas instâncias bur26a e bur26d com a busca local iterativa e na instância els19 e bur26d em ambas as técnicas. Contudo, com o path-relinking é notório a melhora com a hibridização, e ainda mais no mixed e back-and-forward que teve acertos de 32% na instância els19. Além disso, a média das aproximações foram bem altas e a mediana dos custos bem próxima ao custo ótimo. O mixed além de ser um método mais rápido, traz aproximações maiores em comparação aos outros tipos do path-relinking's analisados.

5.4 Classe IV

INST	AET(s)	POSF(%)	DMin(%)	DMax(%)	DAFC(%)	SDC	Median	BSK
tai20b	0.00263887	0.00	15.32	41.15	18.07	13.08	12.79	122455319.00
tai25b	0.00608187	0.00	14.21	36.59	17.80	10.10	17.53	344355646.00
tai30b	0.0131817	0.00	3.12	32.59	13.43	7.79	13.13	637117113.00
tai60b	0.542431	0.00	4.21	13.82	7.54	3.63	7.26	608215054.00
tai80b	2.32356	0.00	5.23	11.18	6.79	1.91	6.49	818415043.00

Tabela 10: Experimentos realizados pela busca local da classe IV

INST	AET(s)	POSF(%)	DMin(%)	DMax(%)	DAFC(%)	SDC	Median	BSK
tai20b	0.0246432	3.23	4.24	29.92	5.31	7.04	1.76	122455319.00
tai25b	0.0971687	0.00	2.34	8.63	3.62	2.71	2.65	344355646.00
tai30b	0.284169	0.00	1.56	14.58	3.84	3.36	2.76	637117113.00
tai60b	20.1164	0.00	0.64	3.40	1.94	0.87	1.62	608215054.00
tai80b	122.816	0.00	2.04	4.51	3.09	0.83	3.00	818415043.00

Tabela 11: Experimentos realizados pela busca local iterativa da classe IV

INST	AET(s)	POSF(%)	DMin(%)	DMax(%)	DAFC(%)	SDC	Median	BSK
tai20b	0.0485007	16.13	0.00	7.87	1.16	1.52	0.69	122455319.00
tai25b	0.185015	0.00	0.00	7.39	2.12	1.76	1.32	344355646.00
tai30b	0.550327	0.00	0.12	5.60	1.94	1.30	2.01	637117113.00
tai60b	41.3775	0.00	0.10	3.53	1.53	0.94	1.06	608215054.00
tai80b	240.137	0.00	1.03	3.71	2.47	0.78	2.59	818415043.00

Tabela 12: Experimentos realizados pelo path-relinking mixed + busca local iterativa da classe IV

Na classe IV que são instâncias reais porém mais randomizadas, temos um aproveitamento nulo também com exceção da instância tai20 por parte da busca local iterativa. Porém, com o path-relinking, já se tem um aproveitamento na instância tai25 também, além disso, temos médias de aproximação de 99% com o path-relinking. Embora não tenha-se achado muito as soluções ótimas, com a hibridização, tivemos o encontro de soluções muito próximas sem precisar utilizar outras técnicas que buscam por toda a vizinhança.

6 Conclusão

Devido aos resultados dos experimentos, é evidenciada a grande melhora que a busca local iterativa traz ao executar a mesma busca várias vezes. E com o path-relinking é mais evidente a melhora que traz a hibridização das técnicas, pois, foi encontrado soluções muito próximas da solução ótima sem ter que checar todas as soluções possíveis. Além disso, o mixed demonstrou ser uns dos tipos mais poderosos dos path-relinking pois tem busca por mais soluções, ao criar caminhos tanto pelo forward quanto pelo backward, sem ter muito custo computacional ao parar a busca quando as duas soluções iteradoras são iguais.

Diante disso, mostra-se o potencial da busca local, do path-relinking e da hibridização entre algoritmos. Ademais, é necessário desenvolver novas formas de utilizar elas com parâmetros diferentes para melhorar a acurácia da procura de soluções melhores, diminuição do tempo de execução e a melhora de consistência dos resultados.

Referências

- Anstreicher, K., Brixius, N., Goux, J.-P., and Linderoth, J. (2002). Solving large quadratic assignment problems on computational grids. *Mathematical Programming*, 91(3):563–588.
- Burkard, R. E. and Offermann, J. (1977). Entwurf von schreibmaschinentastaturen mittels quadratischer zuordnungsprobleme. *Zeitschrift für Operations Research*, 21(4):B121–B132.
- Dickey, J. and Hopkins, J. (1972). Campus building arrangement using topaz. *Transportation Research*, 6(1):59–68.
- Elshafei, A. N. (1977). Hospital layout as a quadratic assignment problem. *Journal of the Operational Research Society*, 28(1):167–179.
- Glover, F., Laguna, M., and Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and cybernetics*, 29(3):653–684.
- Glover, F., Laguna, M., and Martí, R. (2004). New ideas and applications of scatter search and path relinking. In *New optimization techniques in engineering*, pages 367–383. Springer.
- Goldbarg, M. C. and Luna, H. P. L. (2005). *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier.
- Gutiérrez, E. and Brizuela, C. A. (2006). Ils-perturbation based on local optima structure for the qap problem. In *Mexican International Conference on Artificial Intelligence*, pages 404–414. Springer.

- Hussin, M. S. and Stützle, T. (2009). Hierarchical iterated local search for the quadratic assignment problem. In *International Workshop on Hybrid Metaheuristics*, pages 115–129. Springer.
- Ribeiro, Celso C e Resende, M. G. (2012). Path-relinking intensification methods for stochastic local search algorithms. *Journal of heuristics*, 18(2):193–214.
- Sahni, S. and Gonzalez, T. (1976). P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565.
- Stützle, T. (1999). Local search algorithms for combinatorial problems: analysis, improvements, and new applications.
- Stützle, T. (2006). Iterated local search for the quadratic assignment problem. *European Journal of Operational Research*, 174(3):1519–1539.