

O PROBLEMA DA ALOCAÇÃO DINÂMICA DE ESPAÇOS: APLICAÇÃO DAS METAHEURÍSTICAS GRASP E BUSCA TABU

Geiza Cristina da Silva¹

Departamento de Ciências Exatas e Aplicadas
Universidade Federal de Ouro Preto
e-mail: geiza@decea.ufop.br

Paulo Oswaldo Boaventura-Netto

Laura Bahiense

Programa de Engenharia de Produção, COPPE
Universidade Federal do Rio de Janeiro
e-mails: boaventu@pep.ufrj.br, laura@pep.ufrj.br

Luiz Satoru Ochi

Instituto de Computação
Universidade Federal Fluminense
e-mail: satoru@ic.uff.br

Neste trabalho é abordado o Problema da Alocação Dinâmica de Espaços onde um projeto é dividido por um número de períodos consecutivos e, em cada um deles, uma quantidade de atividades é realizada. Os recursos necessários para as atividades devem estar disponíveis em espaços de trabalho e os recursos ociosos no período devem ser guardados em espaços de armazenamento. Este problema contém o Problema Quadrático de Alocação (PQA) como um caso particular e, portanto, pertence a classe NP-difícil. Neste contexto a dificuldade em encontrar soluções ótimas, mesmo para instâncias de média dimensão, justifica o uso de métodos heurísticos. Este trabalho propõe algoritmos de construção e heurísticas baseados no GRASP e Busca Tabu para a obtenção de soluções aproximadas. Os resultados computacionais obtidos mostram que os métodos propostos são competitivos na resolução das instâncias do problema existentes na literatura.

Palavras-chave: Metaheurísticas, Otimização Combinatória, Problema Quadrático de Alocação.

Abstract

This work is dedicated to the Dynamic Space Allocation Problem, where a project is divided in a number of consecutive periods, each of them associated to a number of activities. The resources required by the activities have to be available in the corresponding workspaces and those staying idle during a period have to be stored in depots. This problem contains the Quadratic Assignment Problem (QAP) as a particular case, which puts it in the NP-hard class. In this context, the difficulty to identify optimal solutions, even for instances of medium size, justify the use of heuristics. This work proposes construction algorithms and heuristics based on GRASP and Tabu Search for finding approximate solutions. Computational results show the proposed methods as being competitive with relation to the literature instances and the existing techniques.

Keywords: Metaheuristics, Combinatorial Optimization, Quadratic Assignment Problem.

¹ Agradecemos ao CNPq/(CT-Info e Universal), CAPES, FAPEMIG e FAPERJ pelo apoio a este trabalho.

1 Introdução

O Problema da Alocação Dinâmica de Espaços (PADE) é relativamente novo na literatura e foi inspirado na necessidade de se otimizar os custos de realocação dos recursos necessários para a realização de atividades em projetos, em princípio relacionados com o manejo de equipamentos e de outros recursos. Como custos de realocação são considerados, além da distância percorrida pelos recursos, o custo de preparação dos recursos para serem movimentados.

Um projeto é dividido em um número de períodos de tempo consecutivos e, em cada um deles, um determinado número de atividades deve ser realizado. Um recurso é chamado de *recurso necessário* quando necessário para a realização de uma atividade em um dado período e *ocioso*, quando não for usado por uma das atividades do período. O local de realização do projeto é dividido em *espaços de trabalho* e *espaços de armazenamento*, ou *depósitos*. As atividades e seus recursos são associados a espaços de trabalho e os recursos ociosos, a espaços de armazenamento. O conjunto de períodos, com suas atividades e os respectivos recursos necessários e ociosos, constitui a *agenda* do projeto. O objetivo do problema é alocar os recursos de maneira que a distância total percorrida por estes seja minimizada durante os períodos de tempo associados ao projeto.

A Figura 1 apresenta um pequeno problema-exemplo do PADE. A agenda, consta de 4 períodos, 5 atividades e 9 recursos. Na segunda tabela da Figura 1, é apresentado o *layout* dos espaços: na primeira linha, são localizados 3 espaços de trabalho (E_1 a E_3) e, na segunda, 3 espaços de armazenamento (E_4 a E_6). A métrica de Manhattan é utilizada na determinação da distância entre os espaços. Por exemplo, a distância entre os espaços E_1 e E_2 é de 1 unidade, enquanto entre E_1 e E_6 ela é de 3 unidades. A capacidade de cada espaço é de 3 recursos.

Períodos	Atividades (rec. necessários)	Recursos ociosos
1	$A_1(6,7)$ $A_2(1,5)$	2,3,4,8,9
2	$A_2(1,5)$ $A_3(3,4)$	2,6,7,8,9
3	$A_4(2,8)$	1,3,4,5,6,7,9
4	$A_4(2,8)$ $A_5(6,9)$	1,3,4,5,7

E_1	E_2	E_3
E_4	E_5	E_6

Figura 1: Uma pequena instância do PADE: agenda e layout dos espaços

Para que uma solução seja considerada viável, as seguintes condições devem ser consideradas:

- A cada período, somente uma atividade pode ser realizada em um espaço de trabalho;
- uma atividade deve ser associada a um único espaço de trabalho, isto é, se uma atividade for realizada em períodos consecutivos, ela deve permanecer no espaço de trabalho onde se encontra;
- a capacidade do espaço de trabalho ao qual uma atividade é associada deve ser suficiente para guardar seus respectivos recursos;
- a capacidade dos depósitos deve ser respeitada.

A Figura 2 ilustra uma solução ótima para o problema-exemplo da Figura 1, onde a distância total percorrida pelos recursos é de 12 unidades.

A ₂ (1,5)		A ₁ (6,7)	A ₂ (1,5)	A ₃ (3,4)	
2,8	3,4	9	2,8		6,7,9
Período 1			Período 2		
A ₄ (2,8)			A ₄ (2,8)		A ₅ (6,9)
1,5	3,4	6,7,9	1,5	3,4	7
Período 3			Período 4		

Figura 2: Uma solução para a instância exemplo

O presente trabalho utiliza uma formulação matemática do problema para obter soluções ótimas, ou limites inferiores, para algumas instâncias da literatura. Em termos de algoritmos, são propostas heurísticas baseados nos conceitos do GRASP e da Busca Tabu. A seção 2 apresenta uma revisão bibliográfica. A seção 3 é dedicada à formulação matemática do PADE. A seção 4 descreve os algoritmos propostos. Os resultados computacionais obtidos são mostrados na seção 5, após a qual se apresentam as conclusões.

2 Literatura

O PADE, embora seja aplicável em muitas situações práticas, ainda não apresenta muitas contribuições na literatura. Ele foi originalmente proposto por McKendall Jr. *et al.* (2005) que o associaram ao problema de minimizar o custo do transporte de recursos dados – equipamentos, ferramentas e peças de reposição, entre outros – na ocorrência de desligamentos planejados em centrais de energia elétrica. Na literatura, esses custos têm sido diretamente associados às distâncias percorridas pelos recursos. Os autores também apresentam uma formulação matemática (apresentada na Seção 3) e duas heurísticas do tipo *simulated annealing*.

Em McKendall Jr. e Jaramillo (2006) foram apresentadas cinco heurísticas de construção e uma busca tabu (Tabu I). Neste trabalho, uma nova restrição é incorporada ao problema: recursos que se apresentam ociosos em períodos de tempo consecutivos devem permanecer em um mesmo depósito, o que é razoável em termos de custo e diminui o espaço de busca da solução. Tabu I mantém duas listas tabu: uma para os movimentos de atividades e outra, para os movimentos de recursos ociosos. A cada iteração, o melhor movimento é escolhido e a solução resultante torna-se corrente para a próxima iteração. Através de testes comparativos dos métodos anteriores com os propostos, os autores concluíram que a combinação das heurísticas de construção com a busca tabu proporcionou melhor performance, tanto em termos de qualidade das soluções como em tempo de computação.

Em McKendall Jr. (2008), um modelo matemático modificado e três diferentes heurísticas baseadas em busca tabu são propostos. A primeira heurística (Tabu II) é uma busca tabu simples que se diferencia de Tabu I somente na maneira como os recursos ociosos são explorados. Em Tabu II os recursos ociosos são movimentados individualmente. Na segunda heurística deste trabalho (Tabu III) as listas tabu têm seus tamanhos recalculados a cada iteração e consideram ainda estratégias de intensificação e diversificação. A terceira, Tabu IV, é semelhante a Tabu II diferindo em como o movimento a ser executado na solução corrente é escolhido: todos os possíveis movimentos são avaliados e ordenados; os melhores M movimentos são considerados e selecionados para fazer parte de uma *lista de movimentos candidatos* (LMC). A partir daí é considerado o primeiro movimento de LMC. Este é aceito com uma probabilidade p . Se o movimento for aceito, a solução resultante passa a ser corrente para a próxima iteração. Caso contrário, o próximo movimento é considerado e aceito com uma probabilidade p , sendo este passo repetido até que um movimento seja selecionado. Se nenhum dos movimento de LMC for aceito, o melhor movimento da lista é escolhido. A metodologia apresentada é comparada com os

algoritmos da literatura e os resultados reportados apontam a segunda heurística como o melhor método dentre os que figuraram na comparação.

Apesar de sua definição recente, o PADE pode ser correlacionado com outros problemas muito estudados na área de Pesquisa Operacional. A tarefa de associar atividades a espaços de trabalho corresponde ao Problema Quadrático de Alocação (PQA) (Koopmans, 1957; Loiola. *et al.*, 2007). No entanto, o PADE propõe que a alocação de atividades deva ocorrer em múltiplos períodos de tempo, recaindo exatamente na definição do Problema Dinâmico de *Layout* de Facilidades (PDFL) (Rosenblatt, 1986). Já o sub-problema do PADE, de associação dos recursos ociosos a depósitos, é associado ao Problema Quadrático de Alocação Generalizado (PQAG) (Lee e Ma, 2005). Este sub-problema é de complexidade não menor que a do PQAG uma vez que a associação dos recursos ociosos deve ocorrer em mais de um período.

3 Modelo Matemático

Na definição formal do PADE, os seguintes índices e parâmetros necessários devem ser considerados:

Índices:

- J = número total de atividades ($j = 1, 2, \dots, J$);
- R = número total de recursos ($r = 1, 2, \dots, R$);
- P = número total de períodos ($p = 1, 2, \dots, P$);
- N = número total de locações (espaços de trabalho e de armazenamento);
- L = conjunto de locações, $|L| = N$ ($l = 1, 2, \dots, N$);
- W = conjunto de espaços de trabalho, $W \subset L$ ($w \in W$);
- S = conjunto de depósitos, $S \subset L$ ($s \in S$) e $W \cup S = L$;
- R_j = conjunto de recursos necessários para realizar a atividade j ;
- I_p = conjunto de recursos ociosos no período p ;
- A_p = conjunto de atividades no período p .

Parâmetros:

- d_{kl} = distância entre as locações k e l ;
- C_s = capacidade do depósito s .

Deve-se considerar ainda a variável de decisão x_{prk} que se torna 1 se no período p , o recurso r é associado ao depósito k e 0 caso contrário; e a variável de decisão y_{jw} que assume valor 1 se a atividade j é realizada no espaço de trabalho w e 0 caso contrário. O modelo matemático proposto em McKendall *et al.* (2005) é definido como:

Minimizar:

$$(1) \quad \sum_{r=1}^R \sum_{k=1}^N \sum_{l=1}^N \sum_{p=1}^P d_{kl} x_{prk} x_{p+1rl}$$

Sujeito a:

$$(2) \quad \sum_{s \in S_j} x_{prs} = 1, \forall p \in P, \forall r \in I_p,$$

$$(3) \quad \sum_{r \in I_p} x_{prs} \leq C_s, \forall s \in S, \forall p \in P,$$

$$(4) \quad \sum_{w \in W} y_{jw} = 1, \forall j \in J,$$

$$(5) \quad \sum_{j \in A_p} y_{jw} \leq 1, \forall w \in W, \forall p \in P,$$

$$(6) \quad \sum_{r \in R_j} x_{prk} = |R_j| y_{jw}, \forall p \in P, \forall j \in A_p, \forall w \in W,$$

$$(7) \quad x_{prk} = 0 \text{ ou } 1, \forall p \in P, \forall r \in R, \forall k \in L,$$

$$(8) \quad y_{jw} = 0 \text{ ou } 1, \forall j \in J, \forall w \in W.$$

A função objetivo (1) minimiza a distância percorrida pelos recursos ao longo dos períodos em que o projeto é realizado. As restrições (2) garantem que todos os recursos ociosos em cada período sejam associados a um único depósito e as (3) asseguram que a capacidade dos depósitos seja respeitada em cada período de tempo. As restrições (4) e (5) garantem, respectivamente, que toda atividade seja associada a um único espaço de trabalho e que cada espaço de trabalho tenha no máximo uma atividade associada. As restrições (6) garantem que todos os recursos necessários à realização de uma atividade sejam associados ao mesmo espaço de trabalho ao qual a atividade tenha sido associada. Finalmente, as restrições (7) e (8) determinam que as variáveis de decisão são binárias.

Esta formulação claramente quadrática, tem sua forma linearizada substituindo-se a função objetivo (1) pelo termo (1'), onde $w_{p+1prkl}$ é uma variável de decisão binária. Além disso, as restrições 9 – 11 devem ser acrescentadas.

$$(1') \quad \sum_{r=1}^R \sum_{k=1}^N \sum_{l=1}^N \sum_{p=1}^{P-1} d_{kl} w_{p+1prkl}$$

$$(9) \quad x_{prk} + x_{p+1rl} - 1 \leq w_{p+1prkl}, \forall p \in P, \forall r \in R, \forall k, l \in L, l \neq k,$$

$$(10) \quad x_{prk} + x_{p+1rl} \geq 2w_{p+1prkl}, \forall p \in P, \forall r \in R, \forall k, l \in L, l \neq k,$$

$$(11) \quad w_{p+1prkl} = 0 \text{ ou } 1, \forall p \in P, \forall r \in R, \forall k, l \in L, l \neq k.$$

4 Algoritmos Propostos

Neste trabalho são propostos algoritmos construtivos e heurísticas usando o conceito de GRASP e Busca Tabu. A escolha destas metaheurísticas, em parte se deve aos bons resultados que as mesmas têm mostrado em diferentes problemas de otimização combinatória (Santos *et al.* 2005; Silva *et al.* 2007).

4.1 Algoritmos de Construção

São propostos dois algoritmos de construção, ambos obtendo uma solução a partir da execução de duas etapas: na primeira delas, ocorre a associação das atividades a espaços de trabalho e, em seguida, a associação dos recursos ociosos a depósitos. Os dois algoritmos propostos diferem um do outro na segunda etapa e são doravantes denotados por: Construção Proposta 1 (C1) e Construção Proposta 2 (C2).

Na primeira fase de C1 e C2, consideremos J , o número de atividades e W , o conjunto de espaços de trabalho.

Inicialmente, os espaços de trabalho são ordenados, a partir do espaço mais bem localizado, formando uma *lista inicial de candidatos* (LIC). O tamanho de LIC é, portanto, $|W|$. A LIC tem, então, os espaços de trabalho ordenados do mais bem localizado para o pior e para cada atividade os elementos de LIC que estão disponíveis são selecionados para compor a LRC.

A solução é então construída iterativamente, a cada iteração uma atividade a sendo associada a um espaço de trabalho. Para isto, é criada uma *lista restrita de candidatos* (LRC), um espaço de trabalho é escolhido aleatoriamente da LRC e associado a a . Na construção da LRC, a LIC é percorrida para que se verifique se, em todos os períodos onde a ocorre, o espaço de trabalho escolhido não é ocupado por alguma outra atividade. Se essa ocupação não ocorrer, ele será incluído na LRC. O tamanho da LRC é determinado por: $\min(n_{disp}, \alpha \times |W|)$, onde n_{disp} é o número de espaços de trabalho disponíveis e α é um parâmetro.

Na segunda etapa, C1 é a implementação da *Randomized Storage Policy* (RSP) (McKendall Jr. *et al.*, 2006) que associa cada recurso ocioso ao depósito mais próximo do espaço de trabalho onde

a atividade que o requer estiver alocada, dando maior prioridade para os recursos que são usados mais cedo e C2 é uma adaptação de RSP por considerar a associação de recursos feita no período anterior.

4.2 Estruturas de Vizinhança

Dada uma solução s , são utilizadas as estruturas de vizinhanças $NA(s)$, definida sobre a parte da solução que se refere à associação de atividades e $NR(s)$, que corresponde à parte dos recursos ociosos. Um movimento de troca em $NA(s)$ consiste em permutar os espaços de trabalho de duas atividades e um movimento de realocação, consiste em retirar uma atividade de um espaço de trabalho ao qual está associada e reassociá-la a outro espaço de trabalho até então vazio. Os possíveis movimentos de recursos ociosos em $NR(s)$ consistem em trocar os depósitos de dois (ou mais) recurso(s) em todos os períodos consecutivos nos quais este(s) permaneça(m) ocioso(s) e em remover um (ou mais) recurso(s) de um depósito em todos os períodos consecutivos no qual este(s) permaneça(m) ocioso(s) e realocá-lo(s) a um depósito até então vazio.

4.3 GRASP Proposto (GRASP)

Na fase de construção, GRASP obtém uma solução inicial por meio de C1.

Na fase de busca local, a cada iteração, são realizados todos os movimentos de atividades e de recursos ociosos na solução corrente, descritos em 4.2. O melhor movimento que represente uma melhora é executado e a solução obtida torna-se corrente para a próxima iteração. A busca termina quando a solução corrente não puder mais ser melhorada.

O algoritmo GRASP retorna a melhor solução obtida em um número dado de iterações.

4.4 Busca Tabu Proposta (Tabu)

Tabu executa os seguintes passos até um dado número de iterações sem melhora: no início, o algoritmo avalia todos os possíveis movimentos de atividades. Para cada um deles, a heurística RSP é executada para realocar os recursos ociosos de acordo com a nova matriz de atividades. O movimento não tabu que, quando executado seguido de RSP, gerar a melhor solução é feito e a solução correspondente torna-se corrente para a próxima iteração. O critério de aspiração permite que um movimento tabu possa ser realizado se a solução gerada por sua execução seguida de RSP tiver menor custo que a melhor solução. Uma lista tabu de tamanho dinâmico é utilizada para armazenar os movimentos recentes de atividades. O tempo que um movimento é considerado tabu é determinado pelo tamanho da lista tabu. Inicialmente e a cada η iterações sem melhora, o tamanho da lista é escolhido aleatoriamente dentro de um limite inferior (li) e limite superior (ls).

A maior diferença entre a Tabu aqui proposta e as heurísticas apresentadas na literatura está nos movimentos usados (somente de atividades), enquanto as buscas tabu propostas em McKendall e Jaramillo (2006) e McKendall (2008) executam o melhor movimento (de atividade ou de recursos ociosos) e no uso da heurística RSP para a geração de uma solução parcial de recursos ociosos. Deve-se notar ainda que tanto as buscas tabu da literatura, como a aqui apresentada, podem ser consideradas implementações simples. Entretanto, conforme mostrado na próxima seção, o método proposto é capaz de um desempenho na média superior para as instâncias do problema testadas. Este mesmo resultado, a partir de uma busca tabu simples, foi obtido no estudo de Santos *et al.* (2005), mostrando com isso, que para obter bons resultados a heurística não precisa necessariamente ser muito sofisticada.

5 Resultados Computacionais

Nos testes computacionais foram utilizadas as instâncias da literatura propostas em McKendall *et al.* (2005). Todos os parâmetros envolvidos nos métodos propostos foram empiricamente definidos através de testes preliminares. Os testes foram realizados em um Computador Intel

Quad Core 2 com 2394 Mhz e 3278 Mb de memória RAM usando o sistema operacional Linux Ubuntu na versão 2.6.24-19.

A Tabela 1 apresenta os resultados para as instâncias com 6 e 12 locações e a Tabela 2, com 20 e 32 locações, totalizando 96 instâncias testadas. Foram computadas para cada instância o melhor valor de função objetivo obtido de um total de 10 execuções dos algoritmos Tabu e GRASP (Colunas Tabu e GRASP), o valor médio das execuções (arredondado para inteiro inferior) (M_Tabu e M_GRASP), o tempo de processamento, em segundos, da melhor solução (Tempo(s)) e o tempo médio das execuções (M_Tempo(s)). Reportamos também, para cada método proposto, a diferença percentual entre o valor da melhor solução obtida pelo método e o valor da melhor solução conhecida (Dif.) e a diferença percentual entre a média de valores de soluções obtidas pelo método e o valor da melhor solução conhecida (M_Dif.).

Inst.	Lit.	Tabu	Dif.	Tempo(s)	M_Tabu	M_Dif.	M_Tempo(s)	GRASP	Dif.	Tempo(s)	M_GRASP	M_Dif.	M_Tempo(s)
P1	16	16	0,0	0,0	16	0,0	0,0	16	0,0	0,0	16	0,0	0,0
P2	25	26	3,8	0,0	26	3,8	0,0	26	3,8	0,0	26	3,8	0,0
P3	18	18	0,0	0,0	18	0,0	0,0	18	0,0	0,0	18	0,0	0,0
P4	25	26	3,8	0,0	26	3,8	0,0	26	3,8	0,0	26	3,8	0,0
P5	16	16	0,0	0,0	16	0,0	0,0	16	0,0	0,0	16	0,0	0,0
P6	27	27	0,0	0,0	27	0,0	0,0	27	0,0	0,0	27	0,0	0,0
P7	16	16	0,0	0,0	16	0,0	0,0	16	0,0	0,0	16	0,0	0,0
P8	31	31	0,0	0,0	31	0,0	0,0	31	0,0	0,0	31	0,0	0,0
P9	25	25	0,0	0,0	25	0,0	0,0	25	0,0	0,1	25	0,0	0,1
P10	46	46	0,0	0,0	46	0,0	0,0	46	0,0	0,1	46	0,0	0,1
P11	32	32	0,0	0,0	32	0,0	0,0	32	0,0	0,1	32	0,0	0,1
P12	41	43	4,7	0,0	43	4,7	0,0	43	4,7	0,1	43	4,7	0,1
P13	28	28	0,0	0,0	28	0,0	0,0	28	0,0	0,1	28	0,0	0,1
P14	45	45	0,0	0,0	45	0,0	0,0	45	0,0	0,1	45	0,0	0,1
P15	35	35	0,0	0,0	35	0,0	0,0	35	0,0	0,1	35	0,0	0,1
P16	49	49	0,0	0,0	49	0,0	0,0	49	0,0	0,1	49	0,0	0,1
P17	35	35	0,0	0,0	35	0,0	0,1	35	0,0	0,2	35	0,0	0,2
P18	60	62	3,2	0,1	62	3,2	0,0	62	3,2	0,2	62	3,2	0,2
P19	46	46	0,0	0,1	46	0,0	0,1	46	0,0	0,2	46	0,0	0,2
P20	60	63	4,8	0,0	63	4,8	0,0	63	4,8	0,1	63	4,8	0,1
P21	46	48	4,2	0,1	48	4,2	0,1	48	4,2	0,2	48	4,2	0,2
P22	67	67	0,0	0,0	67	0,0	0,0	67	0,0	0,2	67	0,0	0,2
P23	55	56	1,8	0,1	56	1,8	0,1	56	1,8	0,2	56	1,8	0,2
P24	74	74	0,0	0,0	74	0,0	0,0	74	0,0	0,2	74	0,0	0,2
P25	*30	31	0,0	0,1	31	0,0	0,2	31	0,0	0,5	31	0,0	0,5
P26	*42	43	0,0	0,1	43	0,0	0,2	43	0,0	0,6	43	0,0	0,6
P27	43	43	0,0	0,1	43	0,0	0,1	43	0,0	0,5	43	0,0	0,6
P28	*54	55	0,0	0,1	55	0,0	0,1	55	0,0	0,5	55	0,0	0,5
P29	29	29	0,0	0,1	29	0,0	0,1	29	0,0	0,4	29	0,0	0,4
P30	50	49	-2,0	0,1	49	-2,0	0,1	49	-2,0	0,5	49	-2,0	0,5
P31	42	42	0,0	0,1	42	0,0	0,2	42	2,3	0,5	42	0,0	0,6
P32	*66	69	0,0	0,1	69	0,0	0,1	69	0,0	0,5	69	0,0	0,4
P33	53	52	-1,9	0,7	53	0,0	0,4	52	1,9	1,7	53	0,0	1,7
P34	72	72	0,0	0,4	72	0,0	0,3	72	2,7	2,0	72	0,0	1,9
P35	68	73	6,8	0,5	73	6,8	0,4	73	6,8	1,9	73	6,8	1,9
P36	90	95	5,2	0,3	95	5,3	0,3	95	5,2	1,4	95	5,3	1,4
P37	47	48	2,1	0,4	48	2,1	0,4	48	2,1	1,9	48	2,1	1,8
P38	77	83	7,2	0,5	83	7,2	0,4	83	7,2	1,7	83	7,2	1,7
P39	67	69	2,9	0,4	69	2,9	0,4	69	2,9	1,9	69	2,9	1,9
P40	104	108	3,7	0,2	108	3,7	0,3	108	3,7	1,3	108	3,7	1,4
P41	78	78	0,0	1,3	78	0,0	0,9	78	2,5	4,0	78	0,0	4,1
P42	104	102	-2,0	0,7	103	-1,0	0,8	102	-2,0	4,2	102	-2,0	4,1
P43	110	110	0,0	1,1	110	0,0	0,9	110	0,0	4,1	110	0,0	4,3
P44	137	140	2,1	0,4	140	2,1	0,5	140	2,1	3,0	140	2,1	3,0
P45	66	66	0,0	0,7	66	0,0	0,9	66	2,9	4,5	66	0,0	4,5
P46	111	116	4,3	0,6	116	4,3	0,7	116	5,9	3,8	116	4,3	3,9
P47	111	115	3,5	1,1	115	3,5	1,0	115	3,5	3,9	115	3,5	4,1
P48	169	171	1,2	0,5	171	1,2	0,5	171	1,7	3,1	171	1,2	3,1

Tabela 1: Resultados para as instâncias de 6 e 12 locações

Inst.	Mck	Tabu	Dif.	Tempo(s)	M Tabu	M Dif.	M Tempo(s)	GRASP	Dif	Tempo(s)	M GRASP	M Dif.	Tempo(s)
P49	45	44	-2,3	1,5	44	-2,3	1,3	44	-2,3	6,1	44	-2,3	6,2
P50	63	60	-5,0	1,8	60	-5,0	1,3	60	-5,0	6,9	60	-5,0	7,0
P51	55	54	-1,9	1,1	54	-1,9	1,1	53	-3,8	6,9	54	-1,9	6,8
P52	98	89	-10,1	0,7	89	-10,1	0,9	89	-10,1	4,5	89	-10,1	4,6
P53	49	47	-4,3	2,2	47	-4,3	1,4	47	-4,3	6,4	47	-4,3	6,5
P54	67	62	-8,1	1,6	63	-6,3	1,4	62	-8,1	7,3	62	-8,1	7,3
P55	63	60	-5,0	2,2	60	-5,0	1,4	60	-5,0	7,1	60	-5,0	7,4
P56	97	89	-9,0	0,8	90	-7,8	1,1	89	-9,0	6,4	89	-9,0	6,3
P57	67	66	-1,5	5,2	67	0,0	4,0	66	-1,5	22,2	67	0,0	22,0
P58	106	101	-5,0	3,1	103	-2,9	3,2	100	-6,0	20,7	102	-3,9	20,8
P59	101	93	-8,6	4,1	95	-6,3	3,6	93	-8,6	19,8	96	-5,2	20,2
P60	159	153	-3,9	1,4	154	-3,2	1,8	153	-3,9	13,1	153	-3,9	13,2
P61	82	74	-10,8	6,0	76	-7,9	4,6	74	-10,8	26,3	75	-9,3	26,8
P62	129	112	-15,2	5,2	114	-13,2	5,0	113	-14,2	25,7	115	-12,2	26,7
P63	121	118	-2,5	6,0	120	-0,8	4,2	117	-3,4	25,4	119	-1,7	25,6
P64	190	175	-8,6	2,0	177	-7,3	3,0	176	-8,0	20,3	177	-7,3	20,6
P65	105	101	-4,0	10,7	103	-1,9	10,1	102	-2,9	49,3	104	-1,0	47,9
P66	156	150	-4,0	7,3	154	-1,3	6,1	152	-2,6	45,6	156	0,0	45,2
P67	157	145	-8,3	6,5	148	-6,1	7,3	148	-6,1	44,3	150	-4,7	44,7
P68	234	218	-7,3	3,3	222	-5,4	3,7	218	-7,3	29,3	221	-5,9	29,6
P69	112	114	1,8	12,1	118	5,1	10,7	117	4,3	60,8	118	5,1	61,4
P70	178	166	-7,2	9,1	170	-4,7	9,7	167	-6,6	62,8	172	-3,5	61,6
P71	170	162	-4,9	2,2	167	-1,8	13,2	169	-0,6	63,2	171	0,6	63,4
P72	265	251	-5,6	6,3	255	-3,9	6,5	255	-3,9	48,5	258	-2,7	47,4
P73	74	71	-4,2	8,0	71	-4,2	7,0	71	-4,2	47,9	71	-4,2	48,5
P74	97	90	-7,8	7,0	93	-4,3	8,4	91	-6,6	58,7	92	-5,4	58,7
P75	110	102	-7,8	12,7	104	-5,8	9,6	104	-5,8	66,3	104	-5,8	65,5
P76	155	144	-7,6	5,2	148	-4,7	6,7	144	-7,6	48,8	149	-4,0	48,3
P77	73	70	-4,3	9,1	70	-4,3	7,9	70	-4,3	49,2	70	-4,3	49,8
P78	101	92	-9,8	23,9	96	-5,2	10,6	93	-8,6	62,8	95	-6,3	62,8
P79	110	99	-11,1	8,4	101	-8,9	10,1	100	-10,0	82,6	101	-8,9	82,1
P80	175	163	-7,4	7,0	166	-5,4	8,3	165	-6,1	65,9	167	-4,8	64,4
P81	119	109	-9,2	44,9	113	-5,3	29,4	114	-4,4	215,8	115	-3,5	221,6
P82	176	161	-9,3	38,9	167	-5,4	24,5	168	-4,8	212,7	171	-2,9	210,8
P83	192	185	-3,8	30,1	193	0,5	31,8	194	1,0	222,9	199	3,5	228,4
P84	282	268	-5,2	21,2	278	-1,4	18,5	273	-3,3	164,9	279	-1,1	167,6
P85	125	117	-6,8	39,8	121	-3,3	34,2	124	-0,8	221,5	125	0,0	222,3
P86	192	181	-6,1	30,1	187	-2,7	28,6	185	-3,8	216,5	194	1,0	216,4
P87	193	180	-7,2	41,5	186	-3,8	32,2	191	-1,0	256,8	194	0,5	258,9
P88	302	275	-9,8	44,8	282	-7,1	30,5	288	-4,9	197,6	291	-3,8	198,3
P89	171	162	-5,6	58,9	167	-2,4	67,7	168	-1,8	586,3	175	2,3	577,4
P90	262	251	-4,4	81,9	259	-1,2	57,7	257	-1,9	507,2	260	-0,8	521,2
P91	284	271	-4,8	68,1	283	-0,4	54,2	285	0,4	551,0	288	1,4	547,1
P92	395	388	-1,8	20,5	399	1,0	25,4	391	-1,0	365,5	399	1,0	363,1
P93	189	178	-6,2	81,3	186	-1,6	73,6	186	-1,6	581,3	190	0,5	577,8
P94	281	270	-4,1	87,6	281	0,0	74,4	284	1,1	1455,9	289	2,8	1206,2
P95	318	298	-6,7	69,7	312	-1,9	76,8	318	0,0	2317,9	325	2,2	1402,2
P96	464	437	-6,2	43,1	458	-1,3	41,769	460	-0,9	552,5	465	0,2	606,3

Tabela 2: Resultados para as instâncias de 20 e 32 locações

Procurou-se resolver exatamente instâncias de pequeno e médio porte, através do modelo discutido na Seção 3. Dos 96 problemas-testes disponíveis, a literatura reporta os limites inferiores P1 até P24 e P27, num total de 25 problemas. A formulação matemática foi processada no CPLEX 11 e além dos ótimos já conhecidos, outras instâncias foram resolvidas de forma exata, a saber: P25, P26, P28, P29, P31 e P32. Os valores destacados em itálico, na Tabela 1, indicam o valor ótimo. O uso da técnica exata foi abandonado a partir do problema P32 visto que neste o tempo computacional já era demasiadamente elevado. Surpreendentemente, nas instâncias P25, P26, P28 e P32, os melhores valores reportados na literatura (McKendall, 2008) são menores que o ótimo, o que mostra equívocos destes resultados da literatura. Estes valores equivocados estão destacados com um asterisco (*). Em relação ao desempenho dos algoritmos propostos, Tabu e GRASP chegaram ao ótimo em 24 instâncias diferentes. Verificamos ainda que nas instâncias onde o ótimo não foi alcançado, os métodos estiveram no máximo a 4,8% do ótimo.

Para as 64 instâncias cujo ótimo é desconhecido (P30 e P33 - P96), as heurísticas propostas foram comparadas com os melhores resultados da literatura. Tabu obteve melhores soluções em 50 instâncias mostrando uma redução média de -2,5% e a maior redução (-15,2%) foi obtida na instância P62. O algoritmo foi capaz de manter o resultado da literatura em 5 instâncias e, nas demais, a maior diferença percentual entre Tabu e o melhor da literatura foi de 7,2%. GRASP foi

capaz de melhorar o resultado conhecido 46 vezes, empatou 5 vezes com a literatura e obteve a maior redução de custo também na instância P62, em -14,2%. Em média, encontrou soluções de custo -1,5% melhores do que os da literatura.

As Figuras 3 e 4 mostram as diferenças percentuais entre o melhor resultado da literatura e a média dos resultados obtidos pelos algoritmos propostos para as instâncias das Tabelas 1 e 2. A diferença percentual, para cada instância, foi calculada a partir da fórmula: $(\text{MelhorLiteratura} - \text{Média_Proposto}) / \text{Média_Proposto} * 100$.

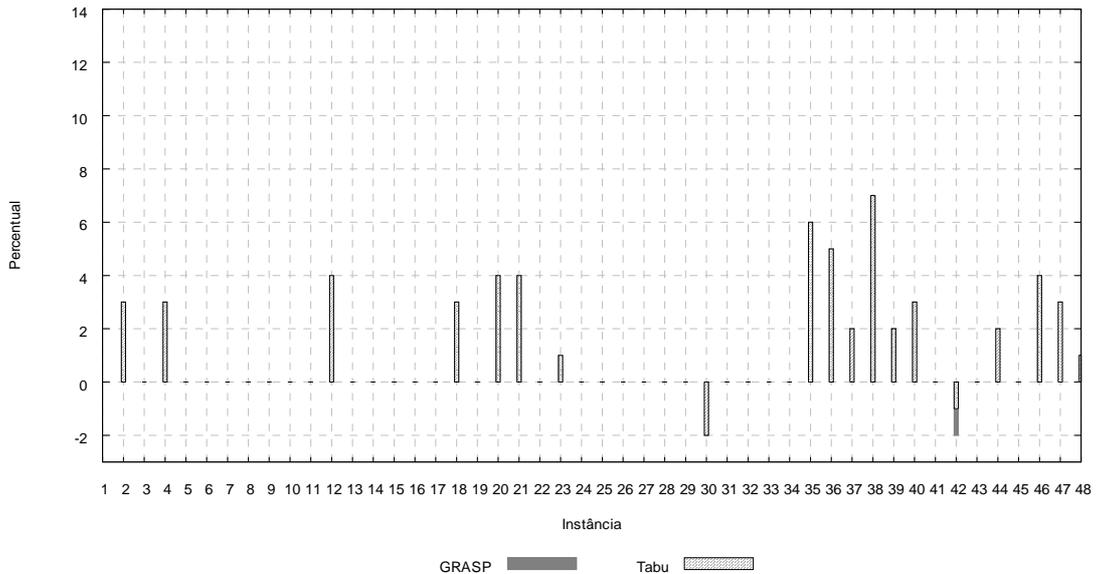


Figura 3: Gráfico de diferenças percentuais para as instâncias de 6 e 12 locações

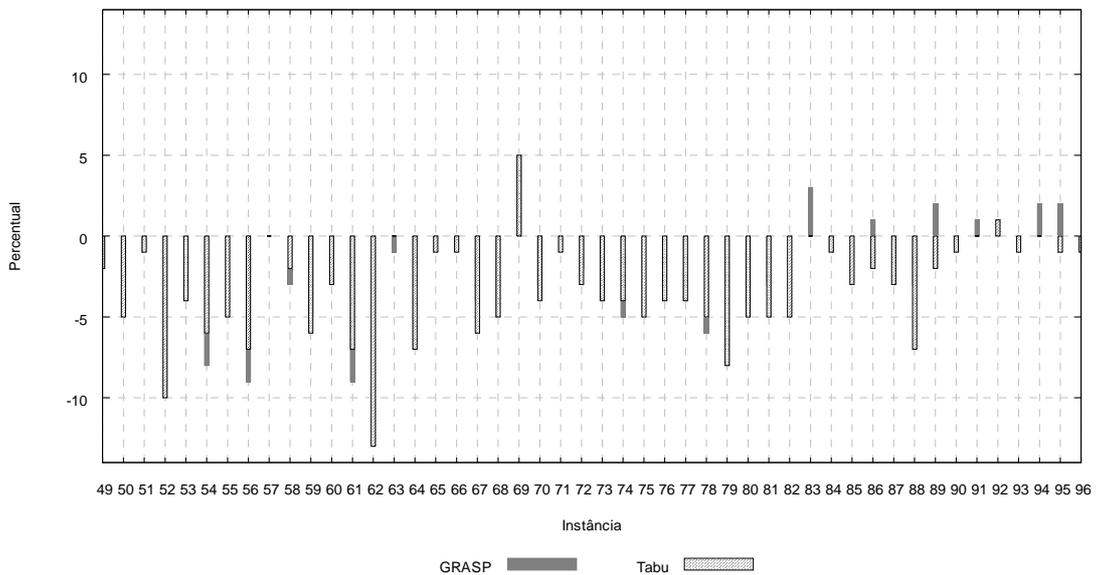


Figura 4: Gráfico de diferenças percentuais para as instâncias de 20 e 32 locações

Em tempo computacional, o algoritmo Tabu se mostrou mais eficiente que o GRASP para os parâmetros utilizados. Não foi feita a comparação de tempo entre os algoritmos da literatura e propostos, visto que os algoritmos foram executados em máquinas de poder computacional diferente.

Através dos resultados empíricos apresentados nota-se que Tabu mostra eficiência maior em instâncias com 20 e 32 locações, o que, na prática, mostra que o algoritmo proposto é menos míope que a versão da literatura. Em relação ao método GRASP, pela maneira simples que foi implementado, acredita-se que este pode ser melhorado a partir de um estudo mais amplo do fator de determinação de gulosidade/aleatoriedade do algoritmo de construção (valor de α) cuja escolha é determinante para o bom desempenho da metaheurística, como pode ser visto em (Resende, 2008). Neste mesmo sentido, acredita-se que seria interessante a aplicação dos conceitos de GRASP Reativo que tem sido frequentemente utilizado com sucesso para outros problemas de semelhante complexidade, como em Silva et al. (2007).

6 Conclusões

Neste trabalho, o Problema da Alocação Dinâmica de Espaços é estudado. Este problema é relativamente novo na literatura mas, embora de definição recente, tem sua importância justificada por poder ser aplicado a casos nos quais mover um recurso é uma tarefa difícil ou dispendiosa. Além disso, o PADE pode ser correlacionado a um dos problemas mais examinados da área, o PQA, além do PDFL e o PQAG.

Dois algoritmos baseados em GRASP e Busca Tabu são propostos e os resultados são comparados com os resultados ótimos (quando estes estão disponíveis) e com os melhores resultados da literatura. Nesta comparação nota-se que a heurística baseada em busca tabu é, na média, capaz de produzir soluções de custos melhores que os demais em questão, para o conjunto de instâncias testadas.

Uma contribuição complementar do trabalho está na correção dos limites inferiores das instâncias P25, P26, P28 e P32, apresentados incorretamente na literatura, além de resolver instâncias da literatura que estavam em aberto.

Referências

- Koopmans, T. C. e Beckmann, M.** (1957). Assignment problems and the location of economic activities. *Econometrica*, 25, 53-76.
- Lee, C-G, e Ma, Z.**, The generalized quadratic assignment problem, *Working paper*, U. of Toronto, Dept. of Mechanical and Industrial Engineering, 2005.
- Loiola, E. M., Abreu, N. M. M. A., Boaventura-Netto, P. O., Hahn, P. e Querido, T.M.** (2007), A survey for the quadratic assignment problem, *European Journal of Operational Research*, 176, 657–690.
- McKendall Jr., A.** (2008), Improved tabu search heuristic for the dynamic space allocation problem, *Computers & Operations Research*, 35, 3347–3359.
- McKendall Jr., A. e Jaramillo, J.** (2006). A tabu search heuristic for the dynamic space allocation problem, *Computers & Operations Research*, 33, 768–789.
- McKendall Jr., A., Noble, J. e Klein, C.** (2005). Simulated annealing heuristics for managing resources during planned outages at electric power plants, *Computers & Operations Research*, 32, 107–125.
- Resende, M. G. C.** (2008). Metaheuristic hybridization with Greedy Randomized Adaptive Search Procedures. *TutORials in Operations Research*, Zhi-Long Chen and S. Raghavan (Eds.), 295-319, INFORMS, 2008
- Rosenblatt, M. J.** (1986), The dynamics of plant layout, *Management Science*, 32, 76–86.

Santos, H. G., Ochi, L. S. e Souza, M. J. F. (2005). A Tabu Search Heuristic with Efficient Diversification Strategies for the Class/Teacher Timetabling Problem, *The ACM Journal of Experimental Algorithmics*, 10, 1-16.

Silva, G. C., de Andrade, M. R. Q., Ochi, L. S., Martins, S. L. e Plastino, A. (2007). New heuristics for the maximum diversity problem, *Journal of Heuristics*, 13, 315-336.