

# Uma Abordagem Híbrida Aplicada ao Problema da Alocação Dinâmica de Espaços

Wagner Silva Costa  
Instituto Federal de Educação, Ciência  
e Tecnologia do Piauí - IFPI  
wagnersc@ifpi.edu.br

Lettiery D´Lamare Portela Procópio,  
Gilberto Farias de Sousa Filho,  
Lucídio dos Anjos Formiga Cabral  
Departamento de Computação Científica  
Universidade Federal da Paraíba  
lettiery.dlamare@dce.ufpb.br,  
{gilberto, lucidio}@ci.ufpb.br

Geiza Cristina da Silva  
Departamento de Estatística  
Universidade Federal de Pernambuco  
geiza.silva@gmail.com

**Resumo** — O Problema da Alocação Dinâmica de Espaços (PADE) é recente na literatura e foi inspirado na necessidade de minimizar as distâncias percorridas entre recursos solicitados para a execução de atividades em centrais de energia nuclear. Existem semelhanças do problema com projetos em que a movimentação de recursos gera custos para o planejamento final, ou em casos onde o congestionamento destes recursos não é aconselhável, como em construções de grandes obras ou na mineração. Faz-se necessário, o uso de métodos aproximativos por ser este considerado NP-Difícil. Para isso, uma nova heurística construtiva é proposta utilizando um modelo matemático baseado em fluxos de atividades que incorpora a informação do grafo de dependências entre as atividades e, então, uma aplicação híbrida da meta-heurística GRASP é discutida. Os resultados computacionais demonstram que esta abordagem é capaz de obter soluções de alta qualidade.

**Palavras-chave** — modelagem matemática; grafos de dependências; problema de alocação de recursos; GRASP

## I. INTRODUÇÃO

Abordamos, neste trabalho, o Problema da Alocação Dinâmica de Espaços (PADE) proposto por McKendall em [5], que foi inspirado na necessidade de otimização da distância percorrida por recursos necessários para a realização de atividades em projetos. 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 ser associados a espaços de trabalho e os recursos ociosos no período devem ser guardados em depósitos. O objetivo do problema é minimizar a distância total percorrida pelos recursos entre os espaços (de trabalho ou depósitos) [1].

Este problema foi, inicialmente, proposto tendo como aplicação a manutenção de redes de energia nuclear. Mais tarde, identificou-se que o mesmo tem aplicações em outros projetos em que a movimentação de um recurso é uma tarefa difícil ou cara, cujo congestionamento desses recursos não é aconselhável, como em construção de pontes e atividades que envolvem mineração.

O PADE apresenta relações muito próximas com outros problemas importantes na área de Otimização Combinatória: em [2] verificamos que o Problema Quadrático de Alocação (PQA) é o problema da designação (com custo mínimo) de objetos a locais, em que cada objeto deve ser atribuído a um único local e reciprocamente, sendo conhecidas as distâncias entre pares de locais e os fluxos de algum tipo de demanda entre pares de objetos. No Problema Dinâmico de Layout de Facilidades (PDLF) [3] a atribuição de objetos a locais deve ser programada para um número de períodos consecutivos, podendo os fluxos entre objetos variar de um período para outro, levando-se ainda em conta o custo de realocação dos objetos. O PDLF pode ser visto então como uma sucessão de PQA's. Também como uma generalização do PQA, o Problema Quadrático de Alocação Generalizado (PQAG) [4] considera alocar, com custo mínimo, múltiplos objetos a um mesmo local, de acordo com sua capacidade.

O PADE pode ser delimitado em dois subproblemas: o primeiro consiste na associação das atividades aos espaços de trabalho (*Workspace Allocation Problem - WAP*) e o segundo consiste na alocação de recursos ociosos aos depósitos (*Storage Space Allocation Problem - SSAP*). O comportamento do WAP pode ser visto como um PDLF. No PADE o custo final de um projeto é dado pela soma de todos os custos gerados pelos movimentos dos recursos, do mesmo modo que no PDLF, cujo custo total é dado pela junção do custo dos fluxos de materiais com o custo da realocação das facilidades. No segundo problema, observam-se relações com o PQAG, onde é considerada a alocação de objetos (recursos ociosos) em mesmo local (depósito), respeitando uma capacidade pré-determinada, da mesma forma que é abordado no SSAP.

Uma breve revisão da literatura mostra que há poucos trabalhos abordando o PADE.

Em [5] o problema é definido de acordo com a necessidade de minimizar a distância percorrida por recursos como equipamentos, peças, ferramentas, entre outros, na operação de planejamento do desligamento em centrais de energia nuclear. É definido um modelo matemático, executado no CPLEX, o qual consegue resolver apenas as instâncias menores.

Por fim, é proposta uma aplicação da meta-heurística *simulated annealing*, que é capaz de encontrar boas soluções.

Em [6] são propostos cinco novos algoritmos de construções e uma busca tabu. Neste trabalho foi acrescentada ao problema uma nova restrição que obriga a associar ao mesmo depósito recursos que permaneçam ociosos por períodos consecutivos de tempo. Esta nova restrição traz, como principal razão, a diminuição de movimentos de recursos ociosos, já que movimentos desnecessários geram custos no projeto final.

Em [7] é proposto um novo modelo matemático mais geral que considera a minimização da realocação dos recursos. Como custos de realocação são considerados custos de transporte e de preparação dos recursos. Neste trabalho, a imposição de não haver movimentos de recursos ociosos que ocorrem em períodos consecutivos, feita em [6], não é mantida. Além disso, o trabalho apresenta três diferentes heurísticas baseadas em busca tabu. Os resultados reportados apontam a heurística que implementa intensificação e diversificação como o melhor método.

Em [8] um método de construção e um algoritmo híbrido baseado em GRASP e Busca Tabu (HGT) são propostos. São feitas comparações entre os resultados da literatura e o HGT, mostrando que o HGT é capaz de obter soluções de melhor qualidade.

O presente trabalho enfoca uma nova proposta baseada na construção de um grafo de dependências entre as atividades, que será utilizado na fase de construção do GRASP, e explora estruturas de vizinhanças já conhecidas da literatura para a fase de busca local, principalmente as utilizadas em [8].

Este trabalho está organizado da seguinte forma: a seção 2 descreve o problema da alocação dinâmica de espaços; a seção 3 traz uma descrição da formulação matemática; a seção 4 propõe uma heurística para solução do PADE; a seção 5 mostra os resultados computacionais obtidos; por fim, na seção 6 são apresentadas algumas considerações finais.

## II. O PROBLEMA DA ALOCAÇÃO DINÂMICA DE ESPAÇOS

O PADE traz, em sua ideia principal, a organização conjunta das atividades em seus locais de trabalho e dos recursos ociosos nos depósitos, onde a capacidade permitida deve ser preservada. Com isso, percebe-se que, na estrutura principal do PADE, temos a alocação como uma importante função a ser realizada de maneira organizada para possibilitar a execução de atividades em seu determinado período de tempo.

De uma forma geral, podemos conceituar o PADE como uma agenda de atividades de um projeto dividida em períodos. A cada período um conjunto de atividades deve ser realizado, sendo que, cada atividade solicita um determinado conjunto de recursos para a sua realização. São necessários como dados de entrada do PADE:

- O *layout* (Espaços de Trabalho e Depósitos) do projeto juntamente com as distâncias e a capacidade de cada espaço, ou seja, a quantidade máxima de recursos suportada por cada local deste *layout*.

- O fluxo do projeto com a quantidade e o conjunto das atividades a serem processadas por períodos, com os recursos a serem utilizados nas atividades. Podemos considerar que apenas uma atividade pode ser realizada no espaço de trabalho por um determinado período de tempo. Também é considerado que a capacidade do espaço de trabalho é suficiente para o armazenamento dos recursos da atividade a ela associada. Deste modo, cada atividade requer apenas um local de trabalho e, pelo menos, um recurso.

Para facilitar a compreensão, é mostrado, através da Tabela 1, um exemplo de dados de entrada para uma instância do problema.

TABELA 1. AGENDA DO PROJETO

Períodos	Atividades (Recursos)	Recursos Ociosos
1	A <sub>1</sub> (6,7) A <sub>2</sub> (1,5)	2,3,4,8,9
2	A <sub>2</sub> (1,5) A <sub>3</sub> (3,4)	2,6,7,8,9
3	A <sub>4</sub> (2,8)	1,3,4,5,6,7,9
4	A <sub>4</sub> (2,8) A <sub>5</sub> (5,7) A <sub>6</sub> (6,9)	1,3,4

A Tabela 1 apresenta a agenda de um projeto com 6 atividades e com 4 períodos de tempo. Cada atividade traz, entre parênteses, os recursos que vão necessitar para suas funções, e na última coluna podemos ver os recursos ociosos a cada período. Por exemplo, no período 2, temos duas atividades em execução (A<sub>2</sub> e A<sub>3</sub>) e os recursos ociosos destes períodos são 2, 6, 7, 8 e 9, sendo requeridos apenas os recursos 1 e 5 para a atividade A<sub>2</sub> e recursos 3 e 4 para a atividade A<sub>3</sub>.

TABELA 2. MATRIZ DE DISTÂNCIAS ENTRE LOCAIS DO PADE

	1	2	3	4	5	6
1	0	1	2	1	2	3
2	1	0	1	2	1	2
3	1	2	0	3	2	1
4	1	2	3	0	1	2
5	2	1	2	1	0	1
6	3	2	1	2	1	0

TABELA 3. LAYOUT COM 6 ESPAÇOS

ES <sub>1</sub>	ES <sub>2</sub>	ES <sub>3</sub>
D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>

A Tabela 3 mostra um exemplo de *layout* com 6 localidades do PADE, sendo na primeira fileira 3 espaços de trabalho ES<sub>1</sub>, ES<sub>2</sub> e ES<sub>3</sub>, que representam na matriz distância, descrita pela Tabela 2 os índices 1, 2 e 3, respectivamente, e para os depósitos temos D<sub>1</sub>, D<sub>2</sub> e D<sub>3</sub> com índices 4, 5 e 6. Levando-se em consideração a agenda do projeto, representada pela Tabela 1, a matriz de distâncias pela Tabela 2, o *layout* de 6 espaços visto na Tabela 3 e ainda uma capacidade máxima igual a 3 recursos por depósito, podemos construir uma solução para o PADE formada por duas estruturas.

A Tabela 4 nos mostra parte da solução para o projeto correspondente à alocação das atividades em seus determinados espaços de trabalho.

TABELA 4. ALOCAÇÃO DE ATIVIDADES

Períodos	Espaços de Trabalho		
	$ES_1$	$ES_2$	$ES_3$
1	$A_2 (1,5)$		$A_1 (6,7)$
2	$A_2 (1,5)$	$A_3 (3,4)$	
3	$A_4 (2,8)$		
4	$A_4 (2,8)$	$A_5 (5,7)$	$A_6 (6,9)$

Na Tabela 5 é possível construir a solução para os recursos ociosos. Para avaliarmos a qualidade da solução descrita nas Tabelas 4 e 5 devemos somar as distâncias, definidas pela Tabela 2, percorridas pela movimentação de cada recurso entre os locais do *layout* durante as transições dos períodos. Por exemplo, o recurso 5 não é movimentado entre os períodos 1 e 2, no entanto, o mesmo entre os períodos 2 e 3 é movimentado do espaço de trabalho  $ES_1$  para o depósito  $D_1$  incorrendo em custo unitário. Em seguida o recurso 5 é movimentado para o espaço de trabalho  $ES_2$  para ser usado pela atividade  $A_5$ , adicionando um custo de duas unidades ao custo total do projeto.

TABELA 5. ALOCAÇÃO DE RECURSOS OCIOSOS

Períodos	Depósitos		
	$D_1$	$D_2$	$D_3$
1	2,8	3,4	9
2	2,8		6,7,9
3	1,5	3,4	6,7,9
4	1	3,4	

A função de custo total do projeto é obtida a partir dos custos de todas as movimentações de recursos, que podem ser classificadas em três tipos:

- *Atividade para Atividade:* neste movimento um determinado recurso é requerido por uma atividade e após seu uso (no término da atividade) é solicitado para outra atividade, criando assim um trajeto entre dois espaços de trabalho;
- *Atividade para Depósito:* neste movimento acontece a saída de um recurso de um determinado local de trabalho, no qual estava sendo utilizado por uma atividade e, ao término dessa execução, o recurso não é mais requerido por nenhuma atividade no período, necessitando de alocação em um depósito, traçando um caminho entre um espaço de trabalho e um depósito;
- *Depósito para Atividade:* com este movimento, o recurso faz o caminho inverso do Movimento de Atividade para Depósito, uma vez que, em um determinado período, o recurso que está em um depósito é solicitado para uma atividade, realizando o trajeto do depósito para um espaço de trabalho.

### III. FORMULAÇÃO MATEMÁTICA

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

#### Índices:

- $J$  = conjunto de atividades ( $j = 1, 2, \dots, |J|$ );
- $R$  = conjunto de recursos ( $r = 1, 2, \dots, |R|$ );
- $P$  = conjunto de períodos ( $p = 1, 2, \dots, |P|$ );
- $N$  = número total de locais do *layout* (espaços de trabalho e depósitos);

- $L$  = conjunto de locais do *layout*,  $|L| = N$  ( $k, l = 1, 2, \dots, N$ );
- $W$  = 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 os locais  $k$  e  $l$ ;
- $C_s$  = capacidade do depósito  $s$ .

#### Variáveis de Decisão:

- $x_{rkp}$  = se seu valor for 1 afirma para o período  $p$ , que o recurso  $r$  foi associado ao depósito  $k$ , caso contrário seu valor se torna 0.
- $y_{jw}$  = se torna 1 se a atividade  $j$  é associada ao espaço de trabalho  $w$  e 0 de outra forma.

O modelo a seguir foi definido por [4], como já mencionado anteriormente.

#### Minimizar:

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

#### Sujeito a:

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

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

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

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

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

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

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

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 cada um dos recursos ociosos, em cada período, seja associado a somente um 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 alocada 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.

#### IV. HEURÍSTICA PROPOSTA PARA O PADE

Nesta seção é proposta a construção para o PADE através de um modelo matemático baseado em um grafo de dependências entre atividades, combinada com a meta-heurística GRASP [9] para a construção de uma solução de qualidade.

##### A. Grafo de dependências

O grafo de dependências é uma representação das conexões entre as atividades de uma instância do PADE, sendo cada atividade representada por um vértice, a existência de um arco entre um par de vértices indica a dependência de um ou mais recursos necessários que estão em uma atividade e serão requeridos por outra. Assim, este grafo dirigido  $G(V, A)$  é formado por dois conjuntos, sendo um de vértices  $V$  e outro de arcos  $A$ . Cada arco descreve a ordem de precedência das atividades e seu peso  $\omega_{ij}$ , é a quantidade de recursos que partem da atividade  $i$  e são necessários para a execução da atividade  $j$ .

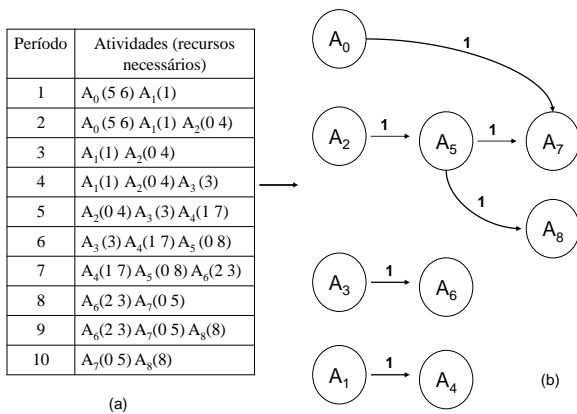


Figura 1. (a) Tabela de agendamento de atividades por período. (b) Grafo de dependência de recursos da instância.

Como exemplo, seja a agenda de um projeto com oito atividades, apresentada na Figura 1a. No grafo (Figura 1b) as atividades correspondem aos vértices. Os arcos denotam a relação de dependência entre as atividades, por exemplo, a atividade  $A_7$  depende de dois recursos, o recurso 0, utilizado anteriormente pela atividade  $A_5$  e o recurso 5 (vindo da atividade  $A_0$ ). Porém, a atividade  $A_5$  também possui uma dependência da atividade  $A_2$  que coexiste com a atividade  $A_0$  (ambas são realizadas em um mesmo período).

Observando o grafo de dependências, podemos interpretar para cada atividade  $v$  (vértice) o grau de entrada como a quantidade de recursos necessários para a execução da atividade  $v$ , que estavam anteriormente alocados a uma ou mais atividades e o grau de saída de uma atividade  $v$  como o conjunto de recursos que deverão ser liberados para uso futuro em outras atividades. Estes são denotadas, respectivamente, por  $ir_v$  e  $r_v$ . Por exemplo, para a atividade  $A_5$  temos  $ir_v = 1$  e  $r_v = 2$ .

##### B. Grafo de dependências estendido

Para delimitar as atividades que coexistem entre si, se fez necessária a extensão do grafo de dependências  $G(V, A)$  para transformá-lo em  $G'(V', A')$  e adicionar as seguintes modificações:

- Inclusão de dois vértices virtuais  $s$  e  $t$  para representarem os nós fonte e sumidouro da rede de fluxo.
- Inclusão do conjunto  $A^T$  de arcos  $(i, j)$  temporais, necessários para demarcar que a atividade  $i$  não tem conflito de período com  $j$ . Assim pode-se concluir que essas atividades  $i$  e  $j$  podem coexistir no mesmo espaço de trabalho por serem realizadas durante períodos diferentes, ou seja, o período final da primeira atividade é menor do que o período de início da segunda atividade.
- Inclusão do conjunto  $A^V$  de arcos  $(i, j)$  virtuais que unem o vértice origem  $s$  a todos os vértices (atividades)  $v \in V$  e estes ao vértice sumidouro  $t$ .

Devem ser consideradas ainda as seguintes definições:

$$V' = V \cup \{s, t\};$$

$$A' = A \cup A^T \cup A^V;$$

$$w_{ij} = 0 \quad \forall (i, j) \in A^V;$$

$$w_{ij} = -(ir_i + r_j) \quad \forall (i, j) \in A^T.$$

É possível observar na Figura 2 a ilustração do novo grafo de dependências estendido  $G'$ .

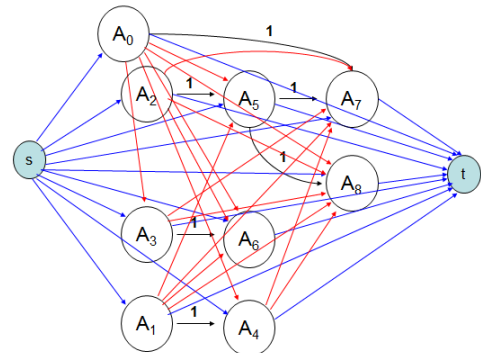


Figura 2. Grafo de dependência estendido

O método de alocação das atividades aos espaços de trabalho corresponde à escolha dos arcos com maior soma dos valores do grafo de dependências estendido. Assim, é justificada a suposição que coloca o valor negativo dos arcos do conjunto  $A^T$  para que tenham pouca influência sobre os arcos reais do conjunto  $A$ , também evitando a escolha dos arcos mais leves, ou seja, alocar atividades que possuam poucos recursos dependentes ao mesmo espaço de trabalho.

##### C. Modelo de Fluxos baseado no Grafo de Dependências

Para a conquista de uma melhor construção, é necessária uma boa escolha dos fluxos de atividades que serão associados aos espaços de trabalho, a qual é obtida através da resolução de um modelo matemático proposto para eleger os melhores fluxos com o auxílio do grafo de dependência

estendido. Para a compreensão deste modelo considera-se  $x_{ij}$  a variável de decisão que se torna 1 para indicar que a atividade  $i$  está no mesmo local de trabalho da atividade  $j$  e 0, caso contrário.

$$\text{Maximizar } \sum_{(i,j) \in A'} w_{ij} x_{ij} \quad (9)$$

Sujeito a:

$$\sum_{j \in N - \{s\}} x_{ij} = 1, \quad \forall i \in V - \{s, t\}, \quad (10)$$

$$\sum_{i \in N - \{t\}} x_{ij} = 1, \quad \forall j \in V - \{s, t\}, \quad (11)$$

$$x_{ij} \in \{0,1\} \quad \forall (i, j) \in A'. \quad (12)$$

Para este modelo, é interessante que se maximize a soma dos pesos dos arcos escolhidos, uma vez que este peso representa o número de recursos compartilhado entre as atividades em um mesmo local de trabalho, minimizando a movimentação de recursos, que é a ideia base do PADE.

Assim, a função objetivo (9) maximiza a soma dos pesos dos arcos formando fluxos de atividades com maior quantidade de recursos compartilhados, preservando as maiores dependências entre as atividades.

As restrições (10) garantem que cada vértice  $v$  possua somente um arco de saída e as restrições (11) garantem que, para cada vértice, haja somente um arco de entrada, de modo que uma atividade esteja em apenas um fluxo. As restrições (12) determinam que as variáveis de decisão são binárias.

#### D. Meta-heurística GRASP - FL

A Figura 3 ilustra uma solução, resultante do modelo proposto na subseção C, contendo três fluxos de atividades:  $\{A_0; A_3; A_6\}$ ,  $\{A_2; A_5; A_7\}$  e  $\{A_1; A_4; A_8\}$ . As atividades de cada fluxo podem coexistir no mesmo espaço de trabalho (ES), entretanto, a definição de qual ES será atribuído para cada fluxo de atividade é de responsabilidade da fase de construção da meta-heurística GRASP - FL (GFL).

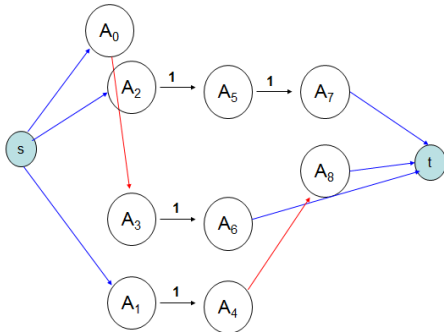


Figura 3. Solução do Modelo de Fluxos sobre o Grafo de dependências estendido

Nesta fase, iniciamos com uma solução  $S$  vazia, ou seja, não temos a presença de atividades nos ES livres. A cada iteração é formada uma lista de candidatos (LC) com a combinação de todos os pares formados por um fluxo de atividade ( $fa$ ) não alocado e um ES livre. E aleatoriamente, um

par  $\{fa, ES\}$  é escolhido a partir da lista restrita de candidatos (LCR). A lista LCR corresponde a uma parte dos melhores elementos de LC. Esta lista é obtida com o auxílio de uma função gulosa, denotada por  $g(\{fa, ES\})$ , definida como:

$$g(\{fa, ES\}) = gc(ES) + gl(\{fa, ES\}) - gd(fa), \quad (13)$$

onde:

- $gc(ES)$  : média aritmética da distância de  $ES$  a todos os outros espaços de trabalho;
  - $gl(\{fa, ES\})$  : soma das distâncias, partindo de  $ES$ , de todas as dependências contidas em  $fa$  à atividades já alocadas anteriormente;
  - $gd(fa)$  : quantidade de dependências contidas em  $fa$ , multiplicada pela distância mínima entre dois espaços de trabalho quaisquer (nas instâncias testadas o valor é igual a 1).
- Os melhores candidatos formados pelos pares candidatos serão aqueles que satisfizerem a condição:

$$g(\{fa, ES\}) \leq g_{min} + \alpha(g_{max} - g_{min}), \quad (14)$$

onde  $g_{min} = \min\{g(\{fa, ES\}) \mid (\{fa, ES\} \in LC)\}$ ,  $g_{max} = \max\{g(\{fa, ES\}) \mid (\{fa, ES\} \in LC)\}$  e  $\alpha \in (0,1)$ .

Por último, após a obtenção de LRC e a escolha aleatória de um par  $\{fa, ES\} \in LRC$ , realiza-se o processo de adaptação, ou seja, adicionamos as atividades pertencentes ao fluxo  $fa$  nos períodos específicos do ES na solução  $S$ .

A condição de parada da fase de construção foi definida quando forem atribuídos à solução  $S$ , todos os fluxos de atividades  $fa$  de entrada, que é condição necessária para obter uma solução viável do problema. Na fase seguinte do GFL o procedimento de Busca Local implementa a busca tabu de [8].

## V. RESULTADOS COMPUTACIONAIS

O modelo matemático descrito neste trabalho foi executado no software ILOG CPLEX disponibilizado pela IBM através da iniciativa acadêmica. O código da meta-heurística GFL foi implementado na linguagem de programação C++. O GFL e o algoritmo HGT da literatura descrito em [8] tiveram seus testes executados em um computador Intel Dual Core, com 4 GB de memória RAM, e sistema operacional Linux Ubuntu. Os resultados do CPLEX aqui reportados foram executados a partir de um computador Core 2 Quad, com 4 GB de memória RAM, e sistema operacional Linux.

Para cada execução do algoritmo híbrido GFL (Modelo de Fluxo em Redes + GRASP) foram realizadas 100 iterações do GRASP com o valor de alfa (parâmetro de seleção do tamanho da lista restrita de candidatos) igual a 10%.

Na Tabela 6, as colunas #Ins, HGT, GFL e CPLEX denotam respectivamente, o número da instância e os resultados das três estratégias para resolução do PADE. A coluna HGT é dividida em Z, o melhor valor obtido pelo algoritmo HGT [8] em 10 execuções, e Tempo, tempo médio em segundos de sua execução. A coluna GFL também é dividida em Z, o melhor valor obtido pelo algoritmo GFL em 10 execuções, e Tempo, tempo médio de execução em segundos de GFL. A coluna CPLEX é dividida em Z\*, valor da solução ótima para o PADE alcançado pelo solver e finalmente o seu tempo de execução em segundos.

É possível perceber que o algoritmo proposto neste trabalho (GFL) alcança uma significativa redução no tempo de processamento das instâncias-teste sem comprometer a qualidade das soluções obtidas. Na tabela 6 podemos observar que nas instâncias 31 e 45 as soluções obtidas pelo GFL têm custo maior, enquanto na instância 39 apresenta menor custo.

TABELA 6. COMPARATIVO ENTRE AS ESTRATÉGIAS HGT, GFL E CPLEX PARA INSTÂNCIAS COM 12 LOCAIS

#Inst	HGT		GFL		CPLEX	
	Z	Tempo	Z	Tempo	Z*	Tempo
25	31	2,1	31	1,6	31	59.551
26	43	2,8	43	1,6	43	20.663,1
27	43	2,4	43	0,2	43	1.008,4
28	55	1,7	55	0,1	55	582,3
29	29	2	29	1,2	29	94.398
30	49	2,3	49	1,3	49	27.609,2
31	<b>42</b>	3,4	<b>43</b>	0,3	42	1.950,3
32	69	2,5	69	0,1	69	3.716,4
33	52	9,1	52	4	*	*
34	72	7,2	72	2,8	*	*
35	73	7,4	73	0,7	73	790.671
36	95	4,9	95	0,4	95	57.012
37	48	6,2	48	3	*	*
38	83	6	83	3,2	*	*
39	<b>69</b>	6,1	<b>68</b>	1,2	68	177.400,5
40	108	3,9	108	0,4	108	211.216,3
41	78	10,6	78	7	*	*
42	102	11,3	102	5,8	*	*
43	110	10,9	110	1,1	*	*
44	140	7	140	0,8	*	*
45	<b>66</b>	13,1	<b>67</b>	6,2	*	*
46	116	10,8	116	7	*	*
47	115	13,2	115	1,5	*	*
48	171	8,3	171	0,7	*	*

Para as instâncias com 6 locais, numeradas de 1 a 24 (conforme [8]), o GFL foi capaz de encontrar um novo melhor valor para a instância 21, com custo igual a 47, que corresponde a solução ótima encontrada pelo CPLEX.

TABELA 7. INSTÂNCIAS EM QUE O HGT E GFL OBTIVERAM SOLUÇÕES DISTINTAS

#Inst	HGT		GFL	
	Z	Tempo	Z	Tempo
21	48	0,9	<b>47</b>	1,1
52	<b>98</b>	12,4	99	1,6
56	<b>100</b>	15,5	101	2,6
62	125	59,8	<b>124</b>	34,5
63	<b>124</b>	73,4	125	12,3
64	193	62,2	<b>192</b>	7,5
85	<b>126</b>	669,8	127	200,3
89	170	996,1	<b>168</b>	362
93	188	1.532,7	<b>185</b>	437,6
95	326	1.164,3	<b>321</b>	218,4
96	473	724,2	<b>470</b>	82,2

A Tabela 7 apresenta as instâncias cujas soluções tiveram valores diferentes alcançados pelas estratégias HGT e GFL, exceto pelas 3 instâncias já apresentadas na Tabela 6. Das 11 instâncias apresentadas o GFL obtém melhores valores em 7 instâncias, perdendo nas outras 4. Podemos destacar que para as maiores instâncias foram obtidas melhorias mais significativas no valor do custo do projeto pelo GFL, além de enorme redução do esforço computacional. Para as instâncias 93, 95 e 96, foram obtidas, respectivamente, reduções de 3, 5 e

3 unidades no valor do custo final do projeto. Em relação aos tempos computacionais (em segundos) estes variaram de 1.532,70; 1.164,3 e 724,2 em HGT para 437,6; 218,4 e 82,2 em GFL, nas supracitadas instâncias, respectivamente. Com base nesses resultados, podemos afirmar que a construção desenvolvida neste trabalho diminuiu bastante o tempo computacional, devido a redução das iterações na busca local.

## VI. CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvida uma meta-heurística híbrida baseada na conjunção da meta-heurística GRASP com uma nova formulação matemática usada na fase de construção para a resolução do Problema da Alocação Dinâmica de Espaços, recém-criado na literatura por McKendall Jr.

A maior contribuição deste trabalho reside na construção de uma nova heurística que traz a utilização de um grafo que representa as dependências de recursos entre as atividades. Isso possibilitou a geração de uma nova formulação matemática baseada em um modelo de fluxo em redes. Este modelo permitiu que a fase de construção da meta-heurística GRASP gerasse boas soluções iniciais para a fase seguinte de busca local. Este algoritmo híbrido provou ser eficaz, obtendo novos melhores valores para instâncias não resolvidas pelo CPLEX dentro de limite aceitável de tempo, e mostrou na média, uma significativa redução no tempo de processamento das instâncias sem comprometer a qualidade das soluções.

## AGRADECIMENTOS

A autora Geiza Cristina da Silva agradece a FACEPE (APQ-0446-3.08/12) pelo suporte financeiro.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Silva G. C., Algoritmos heurísticos e híbridos para o Problema da Alocação Dinâmica de Espaços. Rio de Janeiro, 2010. Tese de Doutorado em Engenharia de Produção, COPPE, Universidade Federal do Rio de Janeiro.
- [2] Loiola E. M., Abreu N. M., Netto P. O. Uma revisão comentada das abordagens do problema quadrático de alocação. SciELO Brasil, Rio de Janeiro 2004.
- [3] Rosenblatt M. J. The dynamics of plant layout. Management Science, v.32, p. 76-86, 1986.
- [4] Lee, C-G., Ma, Z., The generalized quadratic assignment problem, Working paper, U. of Toronto, Dept. of Mechanical and Industrial Engineering, (2005).
- [5] McKendall Jr A. R., Noble J. S., Klein C. M. Simulated annealing heuristics for managing resources during planned outages at electric power plants, Computers & Operations Research, v. 32, pp. 107-125, 2005.
- [6] McKendall Jr A. R., Jaramillo J. A tabu search heuristic for the dynamic space allocation problem, Computers & Operations Research, v. 33, pp. 768-789, 2006.
- [7] McKendall Jr. A. R. Improved tabu search heuristic for the dynamic space allocation problem, Computers & Operations Research, v.35, pp. 3347-59, 2008.
- [8] Silva G. C., Bahiense L., Ochi L. S., Netto P. O. B. The dynamic space allocation problem: Applying hybrid GRASP and Tabu search metaheuristics, Computers & Operations Research, 2012.
- [9] Resende, M.G.C. "Greedy randomized adaptive search procedures (GRASP)", Encyclopedia of Optimization, (C. Floudas e P.M. Pardalos, editores), vol. 2, pp. 373-382, Kluwer Academic Press, 2001.

