

# Algoritmo de Colônia de Formigas Aplicado ao Problema de Alocação de Equipamentos em Satélites

Gustavo Furtado de Oliveira Alves<sup>1</sup>, Sandra Sandri<sup>1</sup>, José Carlos Becceneri<sup>1</sup>

<sup>1</sup>Instituto Nacional de Pesquisas Espaciais – Laboratório Associado de Computação e Matemática Aplicada (LAC/INPE)  
Caixa Postal 515 – 12.227-010 – São José dos Campos – SP – Brasil

***Abstract.** Here we describe a method for solving the problem of allocating equipments inside satellites, using the Ant Colony Optimization algorithm. Each equipment is characterized by its mass and geometry. The goal is to position the equipments, in such a way that the satellite center of mass is minimized and its moment of inertia is maximized. Two case studies are presented.*

***Resumo.** Descreve-se aqui um método de solução do problema de alocação de equipamentos em satélites utilizando o algoritmo de Otimização por Colônia de Formigas. Para cada equipamento considera-se a sua massa e sua geometria. O objetivo é posicionar os equipamentos de tal maneira que o centro de massa do satélite seja minimizado e seu momento de inércia seja maximizado. Dois estudos de caso são apresentados.*

## 1. Introdução

Problemas de otimização de layout são aqueles nos quais um conjunto de itens devem ser colocados num recipiente qualquer, sem sobreposição, de acordo com um conjunto de objetivos e restrições. Estes problemas podem ser vistos como um problema de Corte e Empacotamento (C&P) e são geralmente NP-difíceis [Dyckhoff 1990].

O problema de posicionamento de equipamentos no interior do módulo de satélites artificiais que visa otimizar um conjunto de objetivos, enquanto satisfaz restrições espaciais e/ou de desempenho, é denominado Problema de Projeto de Layout de Módulo de Satélites ou SMLDP, sigla em inglês para "Satellite Module Layout Design Problem" [Wang et al. 2009].

Os problemas tradicionais de alocação de objetos, em geral, visam maximizar a quantidade de objetos no interior do container e balancerar a massa do conjunto. Outros objetivos comuns para problemas de layout buscam minimizar o tamanho do container e mantê-lo balanceado (vide [Parreira et al. 2011]).

O SMLDP é um problema particularmente difícil, devido à complexidade derivada de um grande número de variáveis a serem otimizadas e de restrições de projeto, considerando tanto o momento do lançamento do satélite quanto sua operação durante sua vida útil no espaço. É necessário, por exemplo, otimizar a distribuição de calor criada pela dissipação térmica dos equipamentos eletrônicos, minimizar o efeito de interferências eletromagnéticas, otimizar o balanço de massa, maximizar a proximidade entre equipamentos de um mesmo subsistema, etc. A localização dos equipamentos afeta diretamente a estrutura, o desempenho, a vida útil, o custo de montagem e a integração de todo o

sistema. Por estas razões, a otimização do layout dos equipamentos é fundamental para melhorar o desempenho global de um satélite.

Como discutido em [Cuco 2011], usualmente a arquitetura mecânica de um novo satélite é determinada "manualmente" baseada na experiência da equipe de projetistas e em projetos implementados anteriormente para missões semelhantes. Para auxiliar os projetistas, existe uma tendência de se utilizar sistemas automatizados para gerar um conjunto inicial de layouts otimizados, a partir dos quais os projetistas obtêm um layout final ajustando as soluções iniciais [Souza et al. 2013].

Os sistemas de controle de satélite precisam conhecer o centro de massa e o momento de inércia do conjunto para realizar manobras. A distribuição de objetos no interior do satélite influencia diretamente estas duas variáveis. Em geral, busca-se minimizar o centro de massa e maximizar o momento de inércia, dois objetivos conflitantes. Encontrar o valor máximo ou mínimo de uma ou mais funções objetivo, possivelmente sujeitas a um conjunto de restrições, é conhecido como Problema de Otimização [Becceneri and Silva Neto 2009].

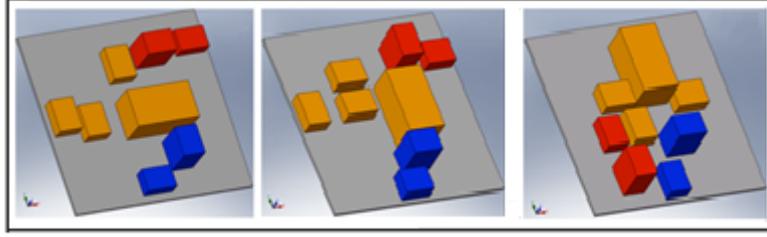
Técnicas de Inteligência Computacional têm sido muito utilizadas atualmente para encontrar soluções para problemas de otimização. Estas técnicas utilizam estratégias de balanço dinâmico entre diversificação e exploração ("exploration" e "exploitation" em inglês) para evitar o confinamento em mínimos e máximos locais [Becceneri and Silva Neto 2009]. Algumas destas técnicas buscam imitar o comportamento da natureza para encontrar soluções de problemas de otimização.

O Algoritmo de Otimização por Colônia de Formigas (ACO), proposto em [Dorigo 1992], é um algoritmo bio-inspirado que se baseia na forma como as formigas são capazes de encontrar um bom caminho, talvez o melhor, entre a colônia e a fonte de comida. Para isso, cada formiga, individualmente, trilha um caminho baseado na experiência das formigas que já fizeram o trajeto.

Neste trabalho, estamos aplicando o ACO para resolver um problema de alocação de objetos em satélites. Para cada objeto, são considerados sua massa e sua geometria. Trabalhos anteriores na literatura (vide por exemplo [Xu et al. 2010]) consideravam apenas a minimização do centro de massa. Neste trabalho, introduzimos a maximização do momento de inércia. A meta-heurística utilizada busca encontrar uma disposição para os objetos que minimize o centro de massa e maximize o momento de inércia de todo o conjunto. Casos de estudos são apresentados, mostrando que a técnica produz bons resultados.

## **2. O problema de alocação de equipamentos em satélites**

Neste trabalho, consideramos o problema de alocação de equipamentos em um satélite tendo como objetivos minimizar o centro de massa em relação ao centro do conjunto e maximizar o momento de inércia, para permitir melhor controle nas manobras. Os equipamentos são caixas retangulares que devem ser dispostos em uma superfície também retangular. As condições de existência para alocar objetos consistem em não permitir sobreposições entre eles e garantir que todos estejam inteiramente na superfície de alocação. Algumas soluções válidas para este tipo de problema podem ser vistas na Figura 1.



**Figura 1. Ilustração de soluções válidas para alocação de objetos dentro de um satélite. Fonte: [Souza et al. 2013]**

Consideremos um recipiente descrito por uma superfície retangular  $S$  e uma coleção de  $N$  objetos (equipamentos)  $O = \{O_1, \dots, O_N\}$ . Considerando que o centro do plano é o centro de  $S$  e supondo que todos os objetos têm massa uniformemente distribuída, podemos calcular o centro de massa do conjunto, levando-se em conta a posição  $(x_i, y_j)$  e a massa  $m_i$  de cada objeto  $O_i$ , conforme a expressão 1.

$$C_{massa} = \sqrt{\left(\frac{\sum_{i=1}^N m_i x_i}{\sum_{i=1}^N m_i}\right)^2 + \left(\frac{\sum_{i=1}^N m_i y_i}{\sum_{i=1}^N m_i}\right)^2} \quad (1)$$

O momento de inércia do sistema é obtido com a soma dos momentos de inércia de cada objeto em relação ao eixo central, através da expressão 2.

$$M_{inerçia} = \sum_{i=1}^N m_i r_i^2 \quad (2)$$

Minimizar o centro de massa e maximizar o momento de inércia pode levar a escolhas de soluções que podem ser conflitantes. Isso caracteriza o problema como multi-objetivo, conseqüentemente não existe uma solução única para o problema, mas um conjunto de soluções que combinam a otimização de cada objetivo. Os conceitos de dominância e de otimalidade de Pareto podem ser aplicados às soluções viáveis para ser definido um conjunto de soluções de compromisso [Cuco 2011].

Uma maneira utilizada por projetistas para resolver este problema multi-objetivo é agregar o centro de massa e o momento de inércia através da expressão 3, com parâmetros  $\lambda_1$  e  $\lambda_2 \in [0, 1]$ . Desta forma o problema multi-objetivo torna-se mono-objetivo.

$$f_{obj} = \lambda_2 \max(M_{inerçia}) - \lambda_1 \min(C_{massa}) \quad (3)$$

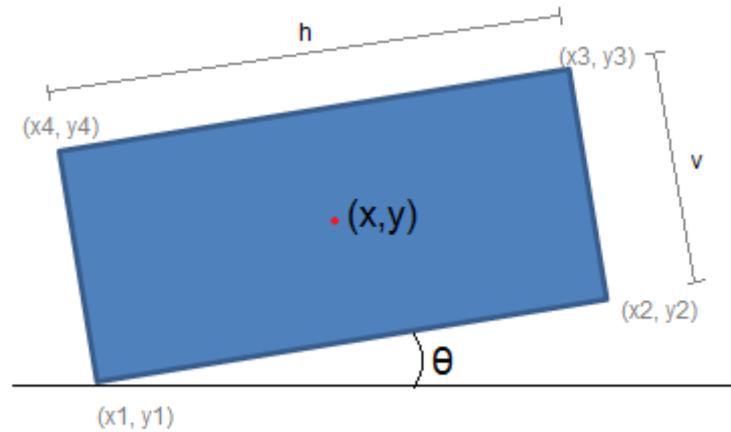
Para validar espacialmente uma solução, é necessário garantir que os objetos retangulares  $O_i$  estão dentro da superfície  $S$  e que não há sobreposição de eles. Para tanto, é necessário identificar a posição dos 4 vértices  $[(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)]$  do retângulo. Esses pontos podem ser obtidos algebricamente utilizando-se a posição  $(x_i, y_i)$  e a inclinação  $\theta_i$  de cada objeto  $O_i$ , conforme as equações 4, 5, 6 e 7. A Figura 2 apresenta as propriedades de cada objeto alocado.

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} -\frac{h}{2} \\ -\frac{v}{2} \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \frac{h}{2} \\ -\frac{v}{2} \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \frac{h}{2} \\ \frac{v}{2} \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} -\frac{h}{2} \\ \frac{v}{2} \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \quad (7)$$



**Figura 2. Variáveis identificadas em um objeto  $O_i$**

Um objeto  $O_i$  não estará contido na superfície  $S$  se o valor absoluto da coordenada  $x$  dos vértices extremos dos objetos forem maiores que  $h_s/2$ , onde  $h_s$  é o tamanho do lado horizontal da superfície de alocação  $S$ . Na Figura 2, as coordenadas horizontais dos vértices extremos são dadas por  $x_2$  e  $x_4$ . O mesmo procedimento se aplica ao eixo vertical.

Para validar a sobreposição entre objetos, Um procedimento que pode ser utilizado é descrito em duas etapas.

1. Garantir que nenhum vértice de um objeto  $O_i$  encontra-se dentro de um outro objeto  $O_j$ .
2. Garantir que nenhum lado de um objeto  $O_i$  cruza com um lado de outro objeto  $O_j$ .

### 3. Algoritmo de Otimização por Colônia de Formigas

O Algoritmo de Otimização por Colônia de Formigas é um dos mais conhecidos algoritmos bio-inspirados e foi proposto por [Dorigo 1992]. O método de busca utilizado por este algoritmo é conhecido como método construtivo ou de trajetória. Neste método, parte-se de uma solução vazia e são acrescentados elementos a este conjunto até que se obtenha uma solução viável para o problema [Becceneri and Silva Neto 2009]. Como

formigas são insetos que vivem em colônias, uma única formiga tem um comportamento muito simples, mas a cooperação de todas as formigas faz com que a colônia seja capaz de executar tarefas complexas [Camazine et al. 2003].

Uma colônia de formigas pode encontrar o melhor caminho para uma fonte de comida, e podemos descobrir como isso acontece observando cada formiga individualmente. As formigas depositam uma substância chamada feromônio no solo quando caminham. O acúmulo de feromônio em um caminho faz com que este caminho seja atraente para uma formiga que vai em busca de alimento. Desta forma, quanto mais formigas percorrem um caminho, mais este caminho se destaca dos demais. Além disso, o feromônio se evapora gradativamente ao longo do tempo, fazendo com que caminhos pouco percorridos percam sua atratividade. O que se nota na natureza é que, após um determinado tempo, a maioria das formigas trilham um caminho mais curto (possivelmente mais otimizado) entre o seu ninho e a fonte de alimento [Becceneri and Silva Neto 2009].

Baseado no comportamento das formigas, [Dorigo 1992] propôs um método de otimização onde as melhores soluções são armazenadas na memória como um feromônio virtual. Assim as melhores soluções encontradas podem influenciar na tentativa de construir uma nova solução ainda melhor.

Existe hoje uma classe de algoritmos conhecida genericamente como "Algoritmo das Formigas", baseadas no primeiro algoritmo proposto por [Dorigo 1992], com diferentes parâmetros e procedimentos. Há por exemplo, várias estratégias para o depósito de feromônio virtual. Em geral, adota-se uma de duas possibilidades: ou todas as formigas depositam feromônios nos caminhos por elas percorridos, ou apenas a melhor formiga de cada geração, algoritmos estes conhecidos como ACS (Ant Colony System) e ACO (Ant Colony Optimization), respectivamente.

A seguir, descrevemos o ACO [Dorigo et al. 1999], considerando o problema de percorrimento mínimo de um grafo direcionado acíclico. O problema consiste em encontrar o melhor caminho entre um nó inicial  $N_0$  e um nó final  $N_f$  pré-estabelecidos, considerando que os arcos entre cada dois nós  $i$  e  $j$  estão associados a um custo  $c(i, j)$ . Cada formiga  $k$  estando no nó  $i$  calcula sua probabilidade de ir para o nó  $j$  através da equação 8, onde:

- $\tau(i, j)$  é a quantidade de *feromônio* entre os nós  $i$  e  $j$ ;
- $\eta(i, j)$  é a informação heurística de atração entre os nós  $i$  e  $j$ . Este parâmetro é chamado de *desejabilidade* e no problema do percorrimento do grafo, é o inverso do custo associado ao arco entre os nós  $i$  e  $j$ ;
- $C^i$  é o conjunto de nós ainda não visitados pela formiga  $k$ .

$$p_k(i, j) = \frac{\tau(i, j)^\alpha \cdot \eta(i, j)^\beta}{\sum_{l \in C^i} \tau(i, l)^\alpha \cdot \eta(i, l)^\beta} \quad (8)$$

Após o término de cada iteração, acontece a atualização do feromônio depositado nos arcos em todas as soluções. Baseado em uma regra pré-estabelecida, o feromônio de todos os caminhos evapora de acordo com uma constante de evaporação  $\rho$ , e as melhores soluções recebem um incremento de feromônio. A equação 9 pode ser utilizada para a atualização, mas outras estratégias para a atualização da quantidade de feromônio nos caminhos, mas outras estratégias podem ser definidas [Blum 2005].

$$\tau(i, j) = (1 - \rho) \cdot \tau(i, j) + \sum_{k=1}^m \Delta\tau_k(i, j) \quad (9)$$

onde  $\Delta\tau_k(i, j)$  é o incremento de *feromônio* se o arco  $(i, j)$  é percorrido pela formiga  $k$ .

No algoritmo ACO, utiliza-se também uma variável aleatória  $q$ , distribuída uniformemente entre  $[0, 1]$ , e um parâmetro de ajuste  $q_0 \in [0, 1]$ . Cada vez que uma formiga  $k$  está para escolher o próximo nó para visitar, um número  $q$  é sorteado entre 0 e 1. Se  $q \leq q_0$ , seleciona-se o arco que obteve o maior valor de probabilidade, segundo a equação 8, privilegiando a memória armazenada na matriz de feromônio. Caso contrário, um arco é escolhido aleatoriamente, segundo a distribuição de probabilidades calculada na equação 8. Obviamente, neste caso, o arco com maior probabilidade tem mais chance de ser escolhido que os demais. No algoritmo AS original,  $q_0 = 0$  e todas as formigas depositam feromônio.

A metodologia para o uso do ACO no problema de alocação de objetos adotada neste trabalho foi proposta em [Xu et al. 2010], onde o ACO é aplicado na escolha da ordem em que os objetos são alocados. No problema de percorrimto de grafo acima, cada formiga deve encontrar uma sequência de nós para visitar. Aqui, no problema de alocação de objetos, a formiga deve encontrar uma ordem para colocar itens num recipiente.

#### 4. Plataforma de Busca de Soluções Otimizadas para o Problema de Alocação de Equipamentos em Satélites

Uma plataforma para testar meta-heurísticas para o problema de alocação de equipamentos em satélites está em desenvolvimento no INPE. Esta plataforma está sendo projetada para satélites contendo uma prateleira retangular (com duas faces), colocada no centro do satélite. Apesar de simples, uma plataforma deste tipo já é capaz de modelar sistemas realísticos, como o satélite brasileiro de pequeno porte ITASAT 1 [ITASAT 2012], atualmente em desenvolvimento por um grupo liderado pelo Instituto Tecnológico de Aeronáutica - ITA (vide Figura 3).

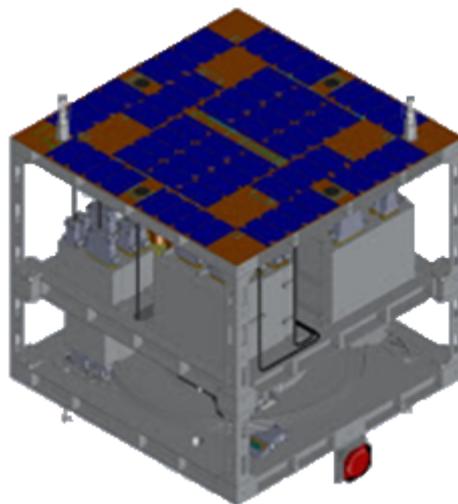


Figura 3. Satélite ITASAT com uma prateleira de duas faces.

O "Algoritmo de Otimização por Colônia de Formigas" foi o primeiro algoritmo implementado nesta plataforma, mas um dos objetivos dela é que vários algoritmos de otimização sejam implementados. O ACO foi escolhido para ser o primeiro com o objetivo de testar variações do mesmo que estão sendo pesquisadas no INPE, tais como "Algoritmo de Otimização por Colônia de Formigas com Dispersão de Feromônios" [Becceneri and Sandri 2006] e "Fuzzy Ant Colony Optimization Algorithm" [Souto et al. 2009].

Devido à complexidade do problema, o projeto da plataforma foi dividido em três fases, descritas em seguida, que incrementam gradativamente a dificuldade da implementação.

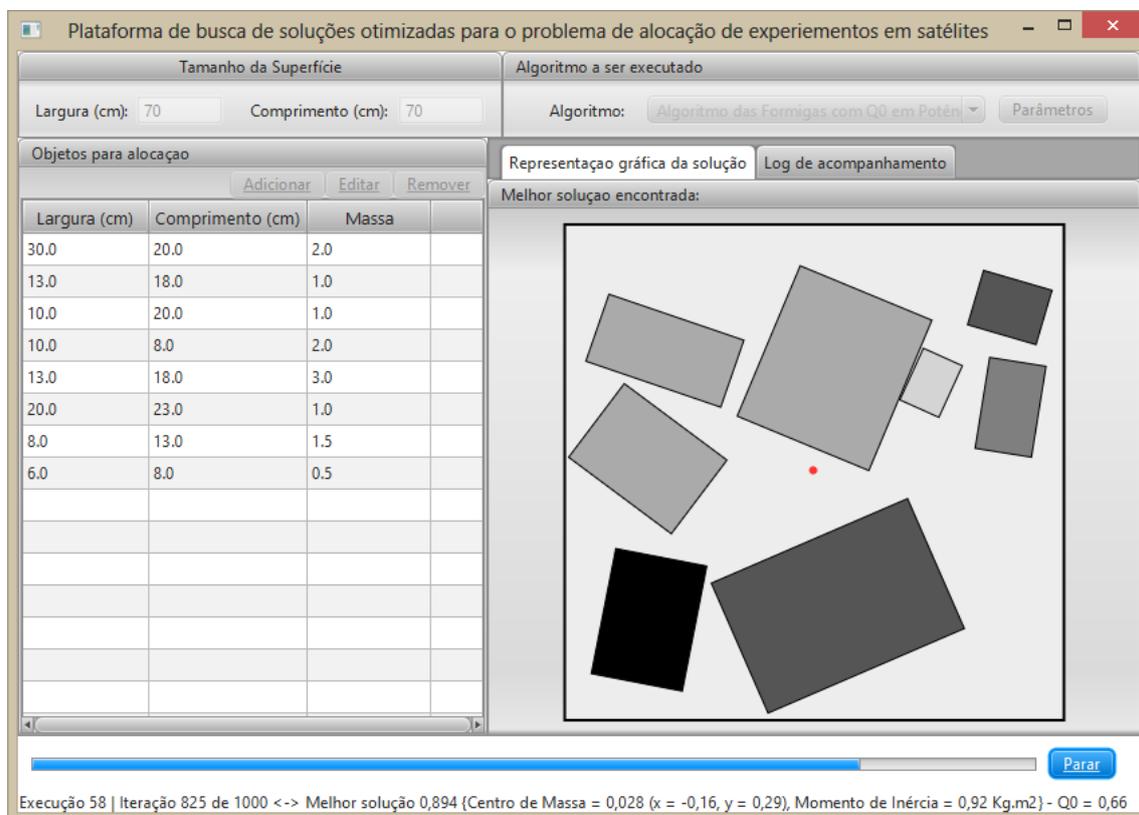
- **Fase 1:** Alocação de objetos retangulares em uma superfície retangular. O primeiro objetivo desta fase do projeto é a de garantir a validade das soluções. Nas configurações válidas, não existe sobreposição de objetos e nenhum dos objetos está fora do recipiente. O segundo objetivo é a de implementar um pequeno conjunto de algoritmos de otimização, incluindo o ACO, descrito anteriormente. O terceiro objetivo é a criação de uma interface básica para configuração dos parâmetros do algoritmos e para visualização de resultados.
- **Fase 2:** Alocação de objetos retangulares em duas faces da prateleira retangular. O objetivo da segunda fase do projeto é a tratar o mesmo problema, mas considerando-se as duas faces de uma mesma prateleira. Em relação à entrada de dados, a interface resultante não modifica aquela criada na Fase 1. Porém, a apresentação do resultado deste módulo deve exibir os objetos alocados em cada uma das faces da prateleira separadamente.
- **Fase 3:** Implementação de restrições na alocação dos objetos. O objetivo da terceira fase é permitir ao usuário (especialista) introduzir conhecimento existente a-priori sobre a alocação de alguns objetos, baseado em sua experiência, aproximando-se do problema real do usuário. Nesta fase, restrições de alocação também poderão ser criadas pelo usuário especialista, tais como definir uma distância mínima entre dois objetos para evitar aquecimento, ou uma distância máxima permitida para possibilitar a ligação de fios entre equipamentos.

Mesmo quando as duas faces da prateleira são consideradas ao mesmo tempo, a interface foi projetada para tratar os objetos 3D como sendo 2D, sem perda de generalidade, já que não há possibilidade de sobreposição em 3D entre os equipamentos.

A Fase 1 está praticamente finalizada, tendo sido atingidos plenamente os objetivos 1 e 3 (validação e criação de interface). O objetivo 2 foi parcialmente atingido com a implementação do ACO como algoritmo de otimização; outros algoritmos deverão estar implementados em breve. As Fases 2 e 3 ainda não foram iniciadas.

A plataforma criada na Fase 1 já permite testar a qualidade do uso de ACO para problemas de alocação de objetos em uma única superfície, como será visto na próxima seção. A Figura 4 traz uma imagem da interface, com uma configuração para um problema com 8 objetos de tamanhos e massas diferentes. As cores em escala de cinza representam a massa de um objeto; quanto maior a massa, mais escura é a representação visual do objeto. É importante salientar que, dadas as características de descrição dos objetos, várias configurações são equivalentes: a configuração obtida após a rotação de

objetos de uma solução  $s$  produz o mesmo valor de  $f(s)$ , se a configuração resultante for válida.



**Figura 4. Solução de problema com 8 objetos de geometria e massas distintas.**

A interface da plataforma tem duas características particularmente interessantes:

- (a) É possível fixar um objeto numa posição específica qualquer e executar o algoritmo de otimização, tomando-se a existência deste objeto nesta posição como restrição.
- (b) É possível modificar uma configuração, mudando as coordenadas de um ou mais objetos e/ou rotacionando-os.

Estas características são muito úteis para o projetista do layout de um satélite, pois isso lhe possibilita testar hipóteses. No caso a, o projetista pode ter razões para preferir um determinado posicionamento de um objeto particular. No caso b, uma vez que o sistema encontrou um conjunto de soluções, o projetista pode testar ajustes destas soluções.

Neste trabalho, o problema bi-objetivo foi considerado mono-objetivo, utilizando-se a Equação 3. A interface permite a escolha pelo usuário dos parâmetros  $\lambda$  para privilegiar um ou outro objetivo. Em particular, para se verificar a qualidade das soluções, basta compará-las a um limitante para cada objetivo. No nosso caso, o limitante para o objetivo 1 (minimização do centro de massa), é obviamente igual a 0 para qualquer conjunto de objetos. Para se verificar a qualidade real das soluções, pode-se compará-las a um limitante. Um bom limitante para o objetivo 2 (maximização do momento de inércia), um problema não trivial, pode ser obtido ao executar o algoritmo, isolando-se o momento de

inércia na equação 3, tomando o parâmetro  $\lambda_1$  (correspondente ao objetivo 1) como igual a 0.

## 5. Estudos de caso

Nesta seção detalhamos dois estudos de caso, que ilustram a adequação de ACO para o problema de alocação de objetos em um satélite. Em ambos estudos de caso, são considerados 8 objetos e alguns parâmetros em comum, conforme mostra a Tabela 1. Os parâmetros  $\lambda_1$  e  $\lambda_2$  determinam que os dois objetivos considerados neste trabalho têm o mesmo peso de importância, isso pode ser alterado pela equipe especialista do projeto. Os parâmetros  $\alpha$  e  $\beta$  controlam o peso relativo da quantidade de feromônio e da desejabilidade dos caminhos percorridos pelas formigas, neste trabalho, estamos considerando também o mesmo peso para ambos. A taxa de decaimento é a constante de evaporação ( $\rho$ ) utilizada para a atualização dos feromônios. Uma constante de evaporação bastante utilizada na literatura é 10% [Becceneri and Sandri 2006].

**Tabela 1. Parâmetros comuns utilizados nos testes.**

$\alpha$	$\beta$	Taxa de decaimento	$\lambda_1$	$\lambda_2$
1	1	0,9	1	1

Foram realizados 2 tipos de experimentos para cada caso de teste, um deles considera 10 formigas e o outro 50. Em ambos os casos foram considerados 3 valores para  $q_0$  (0.1, 0.2 e 0.3), determinados heurísticamente baseado em recentes investigações que quando  $q_0 = 0,2$ , obtêm-se melhores resultados na média (vide [Becceneri and Sandri 2006]). Foram executados 25 testes com sementes diferentes, para os Casos 1 e 2. Os testes foram realizados em um computador com a seguinte configuração: Intel Core™ i7 e 6 GB de memória.

No Caso 1, todos os objetos tem a mesma geometria, mas existem dois valores de massa possíveis. A Figura 5 apresenta a interface da plataforma configurada para o Caso 1. No Caso 2, tanto a geometria quanto a massa dos objetos é variável, como mostrado na Figura 4.

**Tabela 2. Resultados dos experimentos para o Caso 1.**

Nro de Formigas	10			50			
	$q_0$	0,1	0,2	0,3	0,1	0,2	0,3
Melhor solução		0.849637	0.899758	0.798484	0.965214	0.899903	0.950060
Melhor Mom. de inércia		0.881905	1.003902	0.854358	0.993158	0.920958	1.046256
Melhor Centro de massa		0.032267	0.104144	0.055874	0.027945	0.021055	0.096196
Pior solução		-12.5097	-11.4285	-12.3781	-10.3063	-10.7452	-11.0105
Pior Mom. de inércia		0.790610	0.819222	0.774967	0.611757	0.842860	0.850703
Pior Centro de massa		13.30029	12.24774	13.15306	10.91809	11.58806	11.86122
Média		0.744505	0.742207	0.720794	0.804318	0.788994	0.816179
Desvio-padrão		0.066588	0.055197	0.043297	0.052018	0.039320	0.051472
Tempo médio		47s	49s	49s	3min 58s	4min	3min 54s

A Tabela 2 traz os resultados dos experimentos para o Caso 1. Para este caso, as melhores configurações, considerando-se somente o momento de inércia, são obviamente



**Tabela 3. Resultados dos experimentos para o Caso 1.**

Nro de Formigas	10			50		
	$q_0$	0,1	0,2	0,3	0,1	0,2
Melhor solução	0.932837	0.902009	0.891566	0.958097	0.986899	0.940808
Melhor Mom. de inércia	0.992116	1.000798	0.943330	0.977293	1.023249	1.031385
Melhor Centro de massa	0.059279	0.098790	0.051763	0.019196	0.036351	0.090577
Pior solução	-15.2178	-14.5062	-14.5255	-0.96786	-14.8809	-16.2409
Pior Mom. de inércia	0.934205	0.871961	0.802549	0.796688	0.886599	0.991575
Pior Centro de massa	16.15197	15.37818	15.32808	1.764549	15.76753	17.23244
Média	0.749770	0.749078	0.749317	0.842457	0.855024	0.852599
Desvio-padrão	0.072386	0.056291	0.072978	0.052827	0.065642	0.051354
Tempo médio	1min 49s	1min 28s	1min 12s	9min 12s	7min 41s	5min 57s

## 6. Conclusões e Trabalhos Futuros

A utilização do ACO para o problema analisado neste trabalho mostrou ser uma meta-heurística que produz bons resultados, tanto do ponto de vista de desempenho computacional (tempo) como de qualidade de solução (centro de massa perto do valor mínimo e momento de inércia perto do valor máximo). Porém, observamos que aplicar uma meta-heurística apenas na ordem de alocação dos objetos como proposto por [Xu et al. 2010] não tem uma influência muito representativa no problema de alocação de objetos, pois o mais importante para os objetivos é, na verdade, a posição dos objetos. Por isso, é interessante considerar o desenvolvimento de uma heurística que influencie a posição dos objetos.

Como trabalhos futuros, serão implementadas as Fases 2 e 3 do projeto, descritas na Seção 4 (uso de duas faces da prateleira, introdução de mais restrições sobre disposição dos equipamentos, etc.). Também serão implementadas outras meta-heurísticas, como previsto na Fase 1.

No futuro, também será estudada a viabilidade de se variar o parâmetro  $q_0$  do ACO no decorrer das iterações. A ideia é começar com um valor de  $q_0$  baixo que será gradativamente aumentada, similar ao que ocorre na variação da temperatura na meta-heurística *Simulated Annealing* [Becceneri and Silva Neto 2009].

## 7. Agradecimentos

Os autores são gratos aos pesquisadores Fabiano Luis Souza, Wilson Yamaguti e Eduardo Fávero Pacheco da Luz por discussões elucidativas sobre vários aspectos deste trabalho.

## Referências

- Becceneri, J. C. and Sandri, S. A. (2006). Function optimization using ant colony systems with pheromone dispersion. *11th Intl. Conf. on Info. Processing and Management of Uncertainty in Knowledge-based Systems (IPMU06)*.
- Becceneri, J. C. and Silva Neto, J. A. (2009). *Técnicas de Inteligência Computacional Inspiradas na Natureza Aplicação em Problemas Inversos em Transferência Radiativa*, volume 41. SBMAC, São Carlos, SP.

- Blum, C. (2005). Ant colony optimization: Introduction and recente trends. *Physics of Life Review Letters*, pages 353–373.
- Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2003). *Self-organization in Biological Systems*. Princeton University Press.
- Cuco, A. P. C. (2011). Desenvolvimento de uma metodologia multiobjectivo para a otimização do layout de equipamentos em satélites artificiais. Master's thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP.
- Dorigo, M. (1992). *Ottimizzazione, Aprendizimento Automático, ED Algoritmi Basati Si Metafora Naturale*. PhD thesis, Politécnico di Milano, Italy.
- Dorigo, M., Caro, G., and Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Massachusetts Institute of Technology*.
- Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 41:145–159.
- ITASAT, E. (2012). Itasat satélite universitário. <http://www.itasat.ita.br/index.php/pt-br/>. [Acessado em 15/07/2013].
- Parreira, M. O., Ruwer, S. G., Sandri, S. A., Becceneri, J. C., and Neto, L. L. S. (2011). Aplicação do algoritmo infestação de ervas daninhas (iwo) ao problema multi-objetivo de alocação de objetos circulares. *VIII Encontro Nacional de Inteligência Artificial*.
- Souto, R. P., Luz, E. F. P., Becceneri, J. C., Stephany, S., Velho, H. F. C., Sandri, S. A., and Neto, A. J. S. (2009). A fuzzy ant colony optimization algorithm for the estimation of radiative properties in one-dimensional homogeneous participating media. *20th International Congress of Mechanical Engineering*.
- Souza, F. L., Galski, R. L., Rocco, E. M., Becceneri, J. C., Santos, W. A., and Sandri, S. A. (2013). A tool for multidisciplinary design conception of spacecraft equipment layout. Artigo Submetido.
- Wang, Y.-S., Teng, H.-F., and Y.-J., S. (2009). Cooperative co-evolutionary scatter search for satellite module layout design. *International Journal for Computer - Aided Engineering and Software*, 26(7):761–785.
- Xu, Y. C., Dong, F. M., Liu, Y., Xiao, R. B., and Amos, M. (2010). Ant colony algorithm for the weighted item layout optimization problem. *In Proceedings of CoRR*.