# Optimum Placement of Satellite Components with Genetic Algorithms

Ayman H. Kassem
King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia +966-3-860-1246
akassem@kfupm.edu.sa

ABSTRACT: In this paper, the formulation and the solution procedure for optimal placement of satellite components is given. Satellite internal space is discretized into three- dimensional grid (cubes) such as each cube represents a unit volume. Components or subsystems are approximated by a box with integer dimensions based on its largest three dimensions. The placement problem is formulated as a discreet optimization problem and solved with genetic algorithms. The optimization variables represent the location of the center of mass of each component. Different constraints and requirements including the non- intersection of components are also applied. The solution procedure was tested with simple two-dimensional test cases.

## INTRODUCTION

Space technology is growing every day due to, human ambition of invading the space. Satellites are increasingly used in telecommunications, scientific research, surveillance, and meteorology. These satellites differ widely in their size and onboard components. To put these satellites in their orbits around the earth we are limited to the size of the storage compartment inside the launching vehicle. So depending on the launcher, we need to make sure that the satellite size doesn't exceed certain dimensions. It is also preferable to reduce the product of inertia to simplify the modeling and control of satellite motion so there would be small or no coupling of forces and moments between different axes.[1,2]

In this work we are dealing with the problem of optimum allocation of satellite component. Components placement in a satellite or any other type of vehicle is not a trivial task. It is a popular problem in many industries, such as car or aircraft industries. It is usually required to place the components or subsystems, such as to satisfy certain requirements like mass balance, location of center of mass, requirements on moments of inertia, etc. There are also different constraints such as some component should not placed close to each other, such as magnetic torquers which should not be placed close to magnetometers. Some other components should be placed as far as possible in the satellite such as GPS antennas when used for attitude control. Such problems can be formulated as a discrete optimization problem and genetic algorithms will be used to solve it.

Generally, the problem is three dimensional with optimization variables representing the location of the center of mass of each component. But for trays type satellites, the problem can be reduced to few two dimensional problems (depending on the number of trays). Additional constraints have to be introduced to guaranty non-intersection of components.

The solution procedure is based on genetic algorithms. The use of GA's has been instrumental in achieving good solutions to discrete optimization problems. The discrete nature of the components placement problem has been recognized and the GA approach has been successfully applied to solve such problems[3,4].

The paper will introduce the general formulation and solution procedure of the problem including some cost functions and constraints thought by the authors to be the most frequently used in components allocation. Also some two dimensional problems will be solved at the end of the paper.

## FORMULATION OF THE PROBLEM

Generally, the satellite volume is discretized into three- dimensional grid of cubic elements such as each cube represents a unit volume (based on unit length cm, mm, inch, etc.). Each component is approximated by rectangular with integer dimensions based on its largest three dimensions. For example, if the largest dimensions of a certain component are 3.7, 7.6 and 4.5 then the used dimensions are 4, 8, and 5.

For satellites with volume divided into trays, the problem reduces to two-dimensional with tray area divided into a grid of square elements. Each square in that grid represents a unit area (i.e. unit length in x direction multiplied by unit width in y direction). Each component or subsystem is converted to a box in the three-dimensional case or rectangle in the two-dimensional case using its maximum dimensions.

This integer approximation of the element shape keeps the same location of the center of mass and moments of inertia of the original element. Figure 1 shows an example of a two-dimensional tray with two elements a circular element approximated as a square and an elliptical element approximated as a rectangle.

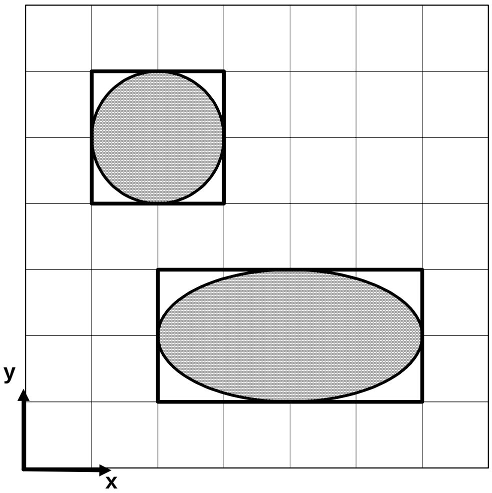


**Figure 1 A satellite layer with two components**

Three dimensional array is used to represent satellite internal volume (two dimensional array for the tray case). Each array element represents a unit volume (area). The array is initialized with zeros for all internal elements and ones to represent the outer contours. Elements are represented with arrays the same way but initialized to ones. To add an element inside the satellite or a tray, element array of ones is inserted in the global array (satellite or tray arrays) in a location equivalent to its center of mass. For example, a two dimensional array representing a square tray with four units at each dimension and a rectangle element with area three units by one unit are shown in figure 2. The element center of mass is inserted at location (3,3) in the array.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & \boxed{1 \;\; 1 \;\; 1} & & & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

**Figure 2. Array representation of a square tray with one rectangle element**

Optimization parameters are the location of the center of mass of each component. So by moving the center of any rectangle to a certain location on the layer we are moving the entire component to that location (i.e. inserting the element array in this location in the global array). Knowing the center of mass location will allow the calculation of the full satellite (tray) center of mass, and the equivalent moment of inertia about this center as follows.

***Calculating the Center of mass***

The center of mass of all components can be calculated through the following equation:

$$\overline{X} = \frac{\sum_{components} mx}{\sum_{components} m} \qquad \overline{Y} = \frac{\sum_{components} my}{\sum_{components} m} \qquad \overline{Z} = \frac{\sum_{components} mz}{\sum_{components} m} \tag{1}$$

Where

x,y,z: position of the center of mass of each component measured from the lower left corner of the satellite or tray (as shown in figure 1).

$\overline{X}$ : component of center of mass of all-components in the x-direction.

$\overline{Y}$ : component of center of mass of all-components in the y-direction.

$\overline{Z}$ : component of center of mass of all-components in the z-direction.

***Calculating the Product of Inertia***

The product of inertia of the components about the center of the layer can be calculated by:

a) Calculating the product of inertia about the origin,

$$I_{xy} = \sum_{Components} I_{xyComp.} - \sum_{Components} m(xy) \qquad I_{yz} = \sum_{Components} I_{yzComp.} - \sum_{Components} m(yz) \qquad I_{zx} = \sum_{Components} I_{zxComp.} - \sum_{Components} m(zx) \tag{2}$$

b) Transferring the product of inertia to the center of the satellite

$$I_{xyc} = I_{xy} + (\sum_{Components} m)\overline{X}\,\overline{Y}$$

$$I_{yzc} = I_{yz} + (\sum_{Components} m)\overline{Y}\,\overline{Z} \quad (3)$$

$$I_{zxc} = I_{zx} + (\sum_{Components} m)\overline{Z}\,\overline{X}$$

Where

$I_{xy}$, $I_{yz}$, $I_{zx}$: products of inertia at the origin.

$I_{xyComp}$, $I_{yzComp}$, $I_{zxComp}$: products of inertia of each component at its center of mass..

$I_{xyc}$, $I_{yzc}$, $I_{zxc}$: products of inertia at the center satellite or tray.

### *Calculating the Distance of the Center of Mass from the Center of the Layer*

The distance $D_1$ between the center of mass of the components and the geometric of the satellite (or a layer), or any other predefined point is used to balance the weight inside satellite and it can be calculated as

$$D_1 = \sqrt{(\overline{X}_{sat} - \overline{X})^2 + (\overline{Y}_{sat} - \overline{Y})^2 + (\overline{Z}_{sat} - \overline{Z})^2} \quad (4)$$

Where

$\overline{X}_{sat}$ : component of center of mass of satellite in the x-direction.

$\overline{Y}_{sat}$ : component of center of mass of satellite in the y-direction.

$\overline{Z}_{sat}$ : component of center of mass of satellite in the z-direction.

### *Sizing the Satellite*

Minimizing equation (4) does not guaranty the compactness of the satellite (small size). Additional equation witch sum the absolute distance of each component to the center of mass has to be used as follows

$$D_2 = \sum_{element} \sqrt{(\overline{X}_{sat} - x)^2 + (\overline{Y}_{sat} - y)^2 + (\overline{Z}_{sat} - z)^2} \quad (5)$$

By minimizing $D_2$ with the avoidance of intersection (as shown in the next section), the size of the satellite can be optimized.

### *Intersection Detection*

Component intersection detection and avoidance is very vital in component placement procedure. Using the zero-one matrix formulation, simplify this task. If any component crossed the satellite volume boundary or intersect with another boundary, the global matrix will have elements with value greater than one. The more of such elements the more intersection of components and the value of a certain number, greater than one, represent the number of intersections. For example, if the global matrix has an element with value 3 then it means that there are three intersecting components at the equivalent location of this value.

## THE GENETIC ALGORITHMS

In 1975, Holland[4] introduced genetic algorithms (GA). Genetic Algorithms are stochastic global search techniques based on the mechanics of genetics. Roughly, a genetic algorithm works as in figure 3. Further description of genetic algorithms can be found in Goldberg[5].

```
BEGIN GA
Make initial population at random.
WHILE NOT (stopping condition) DO
          BEGIN
             Select parents from the population.
             Produce offspring from the selected
             parents   (crossover).
             Mutate the individuals.
             Extend the population adding the
             offspring to it.
             Reduce the extended population.
           END
        Output the best individual found.
END GA
```

**Figure 3. The Pseudo-Code of The Genetic Algorithms (GA)**

The main advantages of Genetic Algorithms are its global optimization performance and the ease of distributing its calculations among several processors or computers as it operate on population of solutions that can be evaluated concurrently. It is a very simple method, generally applicable, and needs no special mathematical treatment of the problem under consideration. It is also well suited for discreet optimization.

The GA used here is binary coding of the x, y and z coordinates of the center of mass of each component concatenated to form the chromosome. Figure 4 shows the coding for "m" components.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $z_1$ | $y_1$ | $z_1$ | $z_2$ | $y_2$ | $z_2$ | ……. | $z_m$ | $y_m$ | $y_m$ |

**Figure 4. A Simple Representation of Elements Position on A Chromosome**

An initial population of members is randomly produced and evaluated. Successive populations are produced by the GA operations of selection, crossover, and mutation. The evaluation process is done through constructing a so-called fitness function (a function to be maximized). The fitness function can include product of inertia (equation (3)), distance from the center of mass (equation (4)), or satellite sizing (equation (5)) or any other requirements.

The crossover is done using Roulette Wheel method with probability (Pc=0.5) and the mutation probability (Pm=0.1). It is also 1-elitist so the top-performing individual of each generation is assured to be included in the next population.

The intersection banality can be introduced into the optimization by rejecting the chromosome which produce intersection of components or by penalizing the fitness function by a certain factor.

The optimization algorithm is shown in table 1.

**Table 1 Optimization procedure**

| |
|---|
| 1-Start with n chromosome concatenating components positions.<br>2-Build the global matrix.<br>3-Run the intersection check procedure.<br>4-calculate fitness function and penalize it in case of intersection.<br>5- Run the GA procedure (figure 3).<br>6- Go to step (2) until stop criterion is reached or for m generations.<br>7- Stop. |

## TWO DIMENSIONAL TEST CASES

The three dimensional general case can be reduced to two-dimensional case especially for small satellites where trays are used. Each tray can be optimized alone as 2D and then the whole satellite can be studied as a multi-two-dimensional problem. Different fitness functions will be studied to study the effectiveness of the solution procedure. A general fitness function F is chosen as follow:

$$F = \frac{1}{\alpha_1 D_1 + \alpha_2 D_2 + \alpha_3 \left| I_{xyc} \right| + \alpha_4} \qquad (6)$$

Where $\alpha_1$, $\alpha_2$, $\alpha_3$, and $\alpha_4$ are constant weighting factors.

The selected problems are simple and easy to be checked and verified for correct answer by inspection. All satellite elements or subsystems are assumed equal mass and square shape of one unit by one unit. The intersection of components is penalized in by setting the cost function F to zero.

For example if the product of inertia was chosen alone to be minimized (i.e. $\alpha_1 = \alpha_2 = 0$) this will require the symmetry of component distribution inside satellite tray. If the distance from the center of mass minimization is the target (i.e. $\alpha_2 = \alpha_3 = 0$) this will require no specific distribution as long as the center of mass coincide (i.e. $D_1 = 0$).

A first trivial test case is to place a single component in a square three units by three units tray (3x3 tray) such that to minimize F with unity weight factors. The solution is as expected (shown in figure 5. the element is centered in the middle of the tray. It is a simple problem and it can be done annually by testing 3*3 = 9 positions.

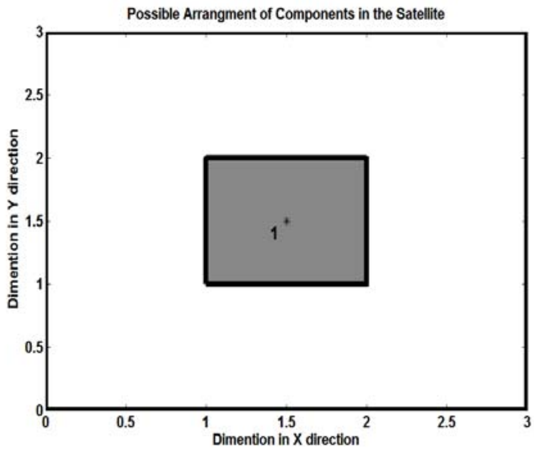


**Figure 5. The Placement of one Component on A (3x3) Layer**

Another simple problem is of placing 3 components all of length unity in same layer. The expected solution is that the 3 components be placed next to each other in a straight line (horizontally o vertically). The output of the algorithm was still identical to this solution as shown in Figure 6 (the vertical case).

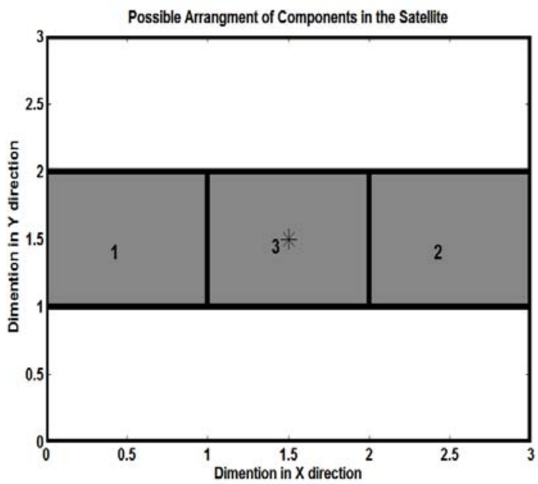


**Figure 6. The Placement of Three Components on A (3x3) Layer**

The combinatorial characteristic of the problem shows a sharp increase of the size of the search space to be 7*8*9 = 504 positions.

A more practical case is five components on a (5x5) layer, the solution was just as expected and the output for this case is shown in Figure 7. Here the search space, for direct enumeration of all possible combinations, increases dramatically to be 25*24*23*22*21 which is approximately 6.3 million possible combinations. For small number of components relative to the tray size, the solution space has approximately $(n \times m)^l$ where *n*, *m* are the dimensions of the tray and *l* is the number of components. The GA procedure suggested here could reach the solution using a generation of 500 and population of 100.

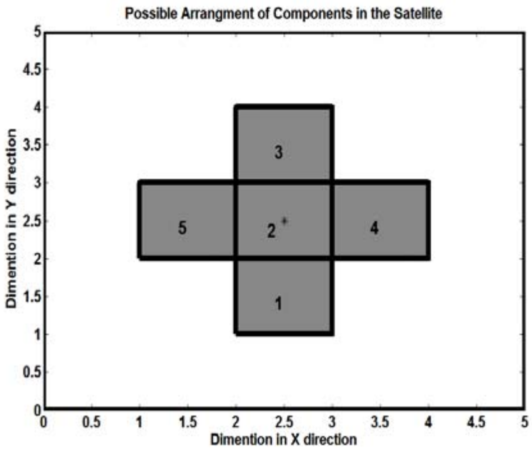


**Figure 7. The Placement of Five Components on A (5x5) Layer**

## CONCLUSIONS

The formulation and solution procedure for satellite-subsystems-placement problem has been introduced in this work. The problem is formulated as discreet optimization problem and Genetic algorithms were used to solve it. GA formulation shows great advantage over direct enumeration or trial and error procedures and it also saves a lot of execution time. Some two-dimensional problems were tested and the result agreed with the optimal solution.

## REFERENCES


1. James R. Wertz, Spacecraft Attitude Determination and Control, Kluwer Academic Publishers, 1978.
2. Marcel J. Sidi, Spacecraft Dynamics and Control: A Practical Engineering Approach, Cambridge University Press, 1997.
3. Gang Wang, T. W. Dexter, W. Punch, "Optimization of a GA and within a GA for a 2-Dimensional Layout Problem", Proceedings on the First International Conference on Evolutionary Computation and Its Applications, Russian Academy of Sciences. 1996.
4. J. L. Rogers, "Optimum Actuator Selection with a Genetic Algorithm for Aircraft Control", Conference on Artificial Neural Network in Engineering, St. Louis, Missouri, November 7-10, 1999.
5. J. Holland, Adaptation in natural and artificial systems. The University of Michigan Press, 1975.
6. D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.