AIAA 2002–5644

# Multidisciplinary Optimization of a Transport Aircraft Wing using Particle Swarm Optimization

Gerhard Venter (gventer@vrand.com) *
*Vanderplaats Research and Development, Inc.*
*1767 S 8th Street, Suite 100, Colorado Springs, CO 80906*

Jaroslaw Sobieszczanski-Sobieski (j.sobieski@larc.nasa.gov) †
*NASA Langley Research Center*
*MS 240, Hampton, VA 23681-2199*

***The purpose of this paper is to demonstrate the application of particle swarm optimization to a realistic multidisciplinary optimization test problem. The paper's new contributions to multidisciplinary optimization is the application of a new algorithm for dealing with the unique challenges associated with multidisciplinary optimization problems, and recommendations as to the utility of the algorithm in future multidisciplinary optimization applications. The selected example is a bi-level optimization problem that demonstrates severe numerical noise and has a combination of continuous and truly discrete design variables. The use of traditional gradient-based optimization algorithms is thus not practical. The numerical results presented indicate that the particle swarm optimization algorithm is able to reliably find the optimum design for the problem presented here. The algorithm is capable of dealing with the unique challenges posed by multidisciplinary optimization as well as the numerical noise and truly discrete variables present in the current example problem.***

## Introduction

PARTICLE Swarm Optimization (PSO) is a recent addition to a growing collection of non-gradient based, probabilistic search algorithms. Some examples of well known and widely used algorithms include genetic algorithms[1] that models Darwin's principle of survival of the fittest and simulated annealing[2] that models the equilibrium of large numbers of atoms during an annealing process. Although this class of algorithms typically require many more function evaluations than comparable gradient-based algorithms, they do provide the designer with several attractive characteristics and has attracted a lot of interest in recent years. For example, these algorithms are generally easy to implement, can efficiently make use of large numbers of parallel processors, do not require continuity in response functions and are better suited for finding global or near global solutions. Although these non-gradient based algorithms provide the designer with several advantages, they should be applied with care. Due to their high computational cost these algorithms should only be used when a gradient-based algorithm is not a viable alternative, such as integer/discrete and discontinuous problems.

Many non-gradient based search algorithms are based on some natural phenomena and PSO is no exception. Particle swarm optimization is based on a simplified social model that is closely tied to swarming theory and was first introduced by Kennedy and Eberhart.[3,4] A physical analogy might be a school of fish that is adapting to its environment. In this analogy each fish makes use of its own memory as well as knowledge gained by the school as a whole to efficiently adapt to its environment. Most of the known PSO applications in the literature are mathematical

*Senior R&D Engineer, AIAA Member

†Senior Research Scientist, Analytical and Computational Methods Branch, Structures and Materials Competency, AIAA Fellow

problems of academic interest. Few engineering applications are known. Fourie and Groenwold applied the algorithm to structural shape and sizing[5] and topology optimization[6] problems.

The authors investigated enhancements to the basic PSO algorithm and applied the algorithm to the minimum weight design of a ten design variable cantilevered beam problem with integer design variables.[7] The enhancements studied include a convergence criterion, dealing with constrained and integer/discrete problems and automatically adjusting the problem parameters during the optimization. The present work will build on this previous effort and apply the PSO algorithm to the multidisciplinary optimization of a typical transport aircraft wing. The example considered here makes use of a bi-level approach to perform the system level optimization of an aircraft for maximum range, accounting for the trade-off between the aerodynamic drag and the structural weight. In this formulation the aerodynamic optimization is performed at the system level and the structural optimization is considered as a sub-problem at the discipline level. The example problem include truly discrete design variables, which means that function evaluations can be performed at specific discrete points only and no gradient information exists between discrete points. The motivation for using PSO in the present design problem is the presence of these truly discrete design variables and severe numerical noise, which makes the use of a gradient-based optimizer impractical.

## Particle Swarm Optimization Algorithm

Particle swarm optimization is based on the social behavior that a population of individuals adapts to its environment by returning to promising regions that were previously discovered.[8] This adaptation to the environment is a stochastic process that depends on both the memory of each individual as well as the knowledge gained by the population.

### Basic Algorithm

In the numerical implementation of this simplified social model, the population is referred to as a swarm and each individual as a particle. The numerical implementation repeatedly updates the position of each particle over a time period to simulate the adaptation of the swarm to the environment. The position of each particle is updated using the current position, a velocity vector and a time increment. The process can be outlined as follows:

1. Create an initial swarm, with a random distribution and random initial velocities
2. Calculate a velocity vector for each particle, using the particle's memory and the knowledge gained by the swarm
3. Update the position of each particle, using its velocity vector and previous position
4. Go to Step 2 and repeat until convergence

The new position of each particle at iteration $k+1$ is calculated from (1)

$$\boldsymbol{x}_{k+1}^{i} = \boldsymbol{x}_{k}^{i} + \boldsymbol{v}_{k+1}^{i}\Delta t \tag{1}$$

where: $\boldsymbol{x}_{k+1}^{i}$ is the position of particle $i$ at iteration $k+1$; $\boldsymbol{v}_{k+1}^{i}$ is the corresponding velocity vector; and $\Delta t$ is the time step value. Throughout the present work a unit time step is used.

The velocity vector of each particle can be obtained from one of many different formulations, depending on the particular PSO algorithm under consideration. In their previous work, the authors examined different schemes for calculating the velocity vector and identified the scheme introduced by Shi and Eberhart[9] as a good candidate. This formulation is widely used in the literature and is shown in (2).

$$\boldsymbol{v}_{k+1}^{i} = w\boldsymbol{v}_{k}^{i} + c_1 r_1 \frac{\left(\boldsymbol{p}^{i} - \boldsymbol{x}_{k}^{i}\right)}{\Delta t} + c_2 r_2 \frac{\left(\boldsymbol{p}_{k}^{g} - \boldsymbol{x}_{k}^{i}\right)}{\Delta t} \tag{2}$$

In (2) $r_1$ and $r_2$ are random numbers between 0 and 1, $\boldsymbol{p}^{i}$ is the best position found by particle $i$ so far and $\boldsymbol{p}_{k}^{g}$ is the best position in the swarm at time $k$. Again, a unit time step ($\Delta t$) is used throughout the present work. There are three problem dependent parameters, the inertia of the particle ($w$), and two "trust" parameters $c_1$ and $c_2$. The inertia controls the exploration properties of the algorithm, with larger values facilitating a more global behavior and smaller values facilitating a more local behavior. The trust parameters indicate how much confidence the particle has in itself ($c_1$) and how much confidence it has in the swarm ($c_2$).

The initial swarm is generally created with all particles randomly distributed throughout the design space, each with a random initial velocity vector. In the present work, (3) and (4) are used to obtain the random initial position and velocity vectors.

$$\boldsymbol{x}_{0}^{i} = \boldsymbol{x}_{min} + r_1\left(\boldsymbol{x}_{max} - \boldsymbol{x}_{min}\right) \tag{3}$$

$$\boldsymbol{v}_{0}^{i} = \frac{\boldsymbol{x}_{min} + r_2\left(\boldsymbol{x}_{max} - \boldsymbol{x}_{min}\right)}{\Delta t} \tag{4}$$

In (3) and (4), $r_1$ and $r_2$ are random numbers between 0 and 1, $\boldsymbol{x}_{min}$ is the vector of lower bounds and $\boldsymbol{x}_{max}$ is the vector of upper bounds for the design variables.

**Enhancements**

The authors[7] have introduced a number of enhancements to the basic PSO algorithm that will be used here. These enhancements are discussed in detail in the provided reference and only those applicable to the current work will be briefly mentioned here.

1. **Convergence Criterion**
   A simple convergence criterion is used. Changes in the objective function is monitored for a specified number of consecutive design iterations. If the maximum change in the objective function is less than a predefined allowable change, convergence is assumed.

2. **Problem Parameters**
   Constant trust parameter values of $c_1 = 1.5$ and $c_2 = 2.5$ are used. This setup puts slightly more trust in the group than in the individual particle. The inertia weight $w$ is adjusted dynamically throughout the optimization. The value is adjusted based on the coefficient of variation of the objective function values for a 20% subset of best particles. If the coefficient of variation falls below a specified threshold value (1.0 in the present work), the $w$ value is reduced using (5)
   $$w_{k+1} = w_k f_w \tag{5}$$
   where $f_w$ is a constant between 0 and 1. Smaller $f_w$ values would result in a more dramatic reduction in $w$, that would in turn result in a more local search. In the present work $f_w = 0.975$ is used, resulting in a PSO algorithm with a fairly global search characteristic. A starting value of $w = 1.4$ is used to initially accommodate a more global search and is dynamically reduced to no less than $w = 0.35$. The idea is to terminate the PSO algorithm with a more local search.

3. **Dealing with Violated Design Points**
   When dealing with constrained optimization problems, particles with violated constraints requires special attention. The authors introduced a new enhancement to the basic PSO, based on the idea of usable, feasible directions e.g., Vanderplaats.[10] That is, a direction that would reduce the objective function while pointing back to the feasible region of the design space. The enhancement modifies the velocity vector of (2) for particles with one or more violated constraints by re-setting the velocity vector of particle $i$ at iteration $k$ to zero, the velocity vector at iteration $k + 1$ is obtained from (6).
   $$\boldsymbol{v}^i_{k+1} = c_1 r_1 \frac{\left(\boldsymbol{p}^i - \boldsymbol{x}^i_k\right)}{\Delta t} + c_2 r_2 \frac{\left(\boldsymbol{p}^g_k - \boldsymbol{x}^i_k\right)}{\Delta t} \tag{6}$$
   The velocity of particle $i$ at iteration $k + 1$ is thus only influenced by the best point found so far for that particle and the current best point in the swarm. In most cases this new velocity vector will point back to a feasible region of the design space. The result is to have the violated particle move back towards the feasible region in the next design iteration.

   Although the system level optimization problem where the PSO algorithm is applied is an unconstrained problem in the present work, the above modification is still useful for design points that has a new position, $\boldsymbol{x}^i_{k+1}$, outside the design variable bounds. The position of these points are reset to the closest bound to avoid analyzing any points outside the specified design space, and the velocity vector is modified as shown in (6).

4. **Craziness Operator**
   To avoid premature convergence of the PSO algorithm additional randomness is introduced using a craziness operator. The craziness operator acts similarly to the mutation operator in genetic algorithms. The craziness operator used here, changes both the position and the velocity vector of affected particles. The position of the particles are changed randomly, while the velocity vector of each modified particle is reset to only the second component of (2) as shown in (7).
   $$\boldsymbol{v}^i_{k+1} = c_1 r_1 \frac{\left(\boldsymbol{p}^i - \boldsymbol{x}^i_k\right)}{\Delta t} \tag{7}$$
   The particles to be affected are identified using the coefficient of variation for the objective function values of all particles, at the end of each design iteration. If the coefficient of variation falls below a predefined threshold value, it is assumed that the swarm is becoming too uniform. In this case, particles that are located far from the center of the swarm are identified, using the standard deviation of the position coordinates of the particles. Particles that are located more than 2 standard deviations from the center of the swarm are subjected to the craziness operator. In the present work, a coefficient of variation threshold value of 0.1 is used.

## MDO Formulation

Multidisciplinary design optimization can be described as a methodology for the design of systems

where interaction between several disciplines is considered and there are design variables that directly affect more than one of the disciplines. Multidisciplinary design optimization problems typically have increased computational requirements and organizational complexity as compared to single discipline optimization. Many different approaches have been proposed to deal with these additionally challenges, with Sobieszczanski-Sobieski and Haftka[11] providing a survey of developments in aerospace design.

The added computational burden is associated with an increase in the number of design variables, since variables from the different disciplines are now considered at the same time. The increase in the computational burden is however larger than just the increase in the number of design variables, since adding design variables to an optimization problem causes a super linear increase in the associated computational cost. The organizational complexity is related to the coupling of different discipline level software into a single software system. The complexity of the interaction between the disciplines vary greatly from application to application, but is generally not just a straightforward transfer of data.

In the present work the multidisciplinary design of a typical transport aircraft wing is considered. Here the goal is to simultaneously optimize aerodynamic drag and structural weight, similar to the problem introduced by Garcelon et al.[12] The aerodynamic analysis is simplified to specifying a "reasonable" pressure distribution in the chord-wise and span-wise directions of the wing. This simplified aerodynamic analysis is coupled with a commercially available finite element structural code, GENESIS.[13] Although the aerodynamic analysis is simplified, the interaction between the aerodynamic and the structural disciplines is complex. First the aerodynamic analysis provides a pressure distribution that is converted to concentrated nodal forces for the structural analysis. The resulting structural deformations, in turn, influences the aerodynamic pressure distribution, resulting in a circular dependency between the two disciplines. Additionally, changes in the aerodynamic design variables, like the aspect ratio and depth-to-chord ratio, affect the structural finite element model.

A two-level approach similar to that proposed by Garcelon et al.[12] is used here. The system-level optimization manipulates the overall wing geometry and operates on the truly-discrete structural variables of the number of the ribs and spars and the type of the wing cover construction. The continuous variables within each type of construction are allocated to the structural sub-optimization. By using this bi-level approach the organizational and computation burden associated with the problem are significantly reduced. First interactions between the two disciplines is isolated. Second, the bi-level approach allows for simplified initial analysis tools, for example the aerodynamic analysis used here, that may be replaced by a more detailed analysis at a later time. Third, it is possible to take advantage of state-of-the-art software at the discipline level. For example, GENESIS makes use of advanced approximation concepts to reduce the computational effort for structural optimization problems. By using GENESIS to perform the structural optimization as a sub-problem, the overall computational cost is reduced.

**Outline of Optimization Process**

The bi-level approach used here, considers the aerodynamic optimization at the system level, where the goal is to maximize the range. This is an unconstrained problem. The structural optimization is done as a sub-problem at the discipline level. The goal is to minimize the weight of the wing, subject to stress and local buckling constraints. The approach may be summarized as follows:

1. Initialize all the design variables
2. Update the structural finite element model according to the current values of the system level design variables
3. Calculate the aerodynamic pressure distribution and convert to concentrated nodal forces applied to the wing finite element model
4. Use GENESIS to solve the structural sub-problem based on the current aerodynamic loads
5. Calculate the aerodynamic drag
6. Calculate the range
7. Use PSO to change the system level design variables and go back to Step 2 until convergence

In this abstract, the interaction between the aerodynamic pressure distribution and structural deflections are ignored. Instead, it is assumed that the wing will be build to a jig-shape to account for the structural deflections.

## MDO Example Problem

The PSO algorithm is applied to the multidisciplinary design of a typical long-range transport aircraft wing in the Boeing 767 class, similar to that

of Garcelon et al.[12] Two independent load cases are considered, a 3.75 G maneuver and a -1.5 G maneuver. The wing is optimized relative to a reference wing with properties summarized in Table 1. Note that the wing area ($S$), the take off gross weight ($TOGW$), the root chord to tip chord ratio ($c_t/c_r$) and the sweep angle ($p$) are all assumed to be constant.

**Table 1 Reference wing**

| Parameter | Value |
|---|---|
| Span ($b_{ref}$) | 120 $ft$ |
| Root Chord ($c_{rref}$) | 0.5 $ft$ |
| Drag ($D_{ref}$) | 4000 $lbs$ |
| Range ($R_{ref}$) | 5000 $n.\ mi.$ |
| Area ($S$) | 2100 $ft^2$ |
| Take off Gross Weight ($TOGW$) | 300000 $lbs$ |
| Aspect Ratio ($A_{ref}$) | 6.8571 |
| $(h/c)_{ref}$ Ratio | 0.12 |
| $c_t/c_r$ Ratio | 0.4 |
| Sweep ($p$) | 25/120 |

**System Level Optimization**

The aerodynamic optimization is performed at the system level. The goal of the system level optimization is to maximize the range of the wing by changing the aspect ratio ($A$), the depth-to-chord ratio ($h/c$), the number of internal spars, the number of internal ribs, and the type of wing cover construction. The system level optimization problem is thus an unconstrained problem with five design variables, three of which are truly discrete variables. Truly discrete variables are variables for which an analysis can be performed at only specified discrete points and the gradient information have no meaning. Optimization problems with truly discrete design variables cannot be solved using a gradient-based optimization algorithm.

The range is calculated using the simplified Breguet formula that does not account for the required fuel reserve, as shown in (8)

$$R = C_r \ \frac{L}{D} ln \left( \frac{TOGW}{W_c + W_p + W_{opt}} \right) \qquad (8)$$

where: $R$ is the range; $L$ the lift; $D$ the total drag; $W_c$ the non-structural weight; $W_p$ the payload weight; $W_{opt}$ the structural weight and $C_r$ is a constant. The take-off gross weight $TOGW$ includes all the weight components $TOGW = W_c + W_p + W_{opt} + W_f$, where $W_f$ is the fuel weight. In the present work, a structural sub-optimization of the reference wing generates $W_{opt}$ and the range of the reference wing (5000 $n.\ mi.$) is used to obtain a $C_r$ value of 155.81. The following assumptions are maintained: $W_c = 0.48\ TOGW$ and $W_p = 0.13\ TOGW$. Additionally, the lift ($L$) must equal the $TOGW$, which is considered constant in the present work. With these assumptions, the range formula is simplified as shown in (9).

$$R = C_r \frac{TOGW}{D} ln \left( \frac{TOGW}{0.61\ TOGW + W_{opt}} \right) \qquad (9)$$

Equation (9) represents the system level objective function for the aerodynamic optimization and consists of two designable components: $D$ and $W_{opt}$. $W_{opt}$ is the weight from the structural sub-optimization problem factored by 1.3 to account for structural non-optimum weight and $D$ is the total drag obtained from the aerodynamic analysis. The structural sub-optimization problem and the aerodynamic analysis is discussed in more detail in the following sections.

The finite element model of the wing-box for the reference wing with three spars and nine ribs is shown in Fig. 1. For the present design problem the front and back spars and the ribs at the root and tip of the wing are always present. However, the internal spar is optional (the number of internal spars is allowed to vary between zero and one) and the number of internal ribs is allowed to vary between zero and seven. The rib spacing ($S_{rib}$) is a constant and equal to $S_{rib} = b_{ref}/(N_{rib\_max} + 1)$, where $N_{rib\_max} = 7$ is the maximum number of allowable internal ribs. The internal ribs are always placed from the root out, observing the rib spacing $S_{rib}$, with the number of internal ribs obtained from the corresponding discrete system level design variable. Additionally, the construction of the upper and lower wing covers are selected from either a sandwich or a hat-stiffened construction.

**Structural Sub-Optimization**

For the structural sub-optimization problem, a simplified finite element model is used to model the wing-box. The wing-box model consists only of shell elements and, for simplicity, the spar and rib caps are not modeled. It is assumed that the wing-box is manufactured from aluminum with a density of 0.1 $lb/in^3$, an allowable tensile strength of 50 $ksi$ and an allowable compression strength of 25 $ksi$. The reference finite element mesh used in the analysis is shown in Fig. 1 and consists of 72 nodes and 72 shell elements. During the system level optimization, the number of finite elements used to model the wing-box varies between 50 and 72 depending on the number of internal spars and ribs for the cur-

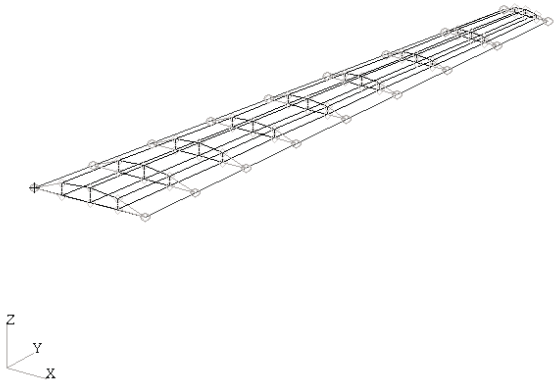


**Fig. 1 Wing-Box structural finite element model**

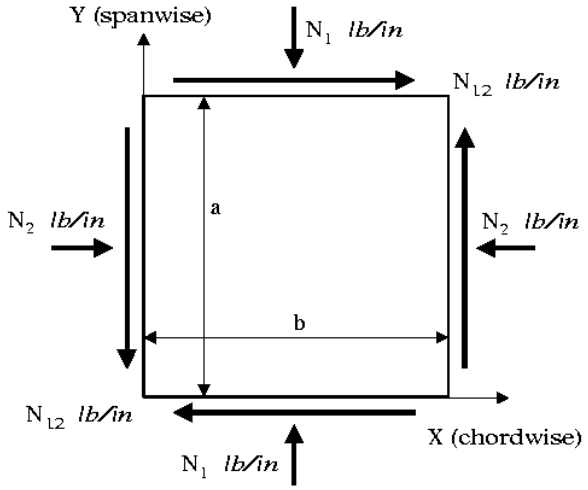


**Fig. 2 Typical wing cover panel**

rent configuration. The nodes shown on the leading and trailing edges of Fig. 1 are non-structural and are included only to transfer aerodynamic loads from the aerodynamic analysis to the structural analysis. The load transfer is done using rigid elements that do not add stiffness to the finite element model.

The goal of the structural sub-optimization problem is to minimize the weight of the wing-box by changing the thickness values of the shell elements, subject to allowable stress and local buckling constraints. The number of design variables in the structural sub-optimization problem depends on the number of internal spars and ribs and the upper and lower wing cover construction selected at the system level. The spars and ribs are modeled as aluminum panels with a single thickness design variable per panel. All three spars between the same two ribs share the same thickness design variable. Similarly, a single design variable is used to design the thickness of each rib. There is thus a maximum of eight design variables for designing the spar thickness and eight design variables for designing the rib thickness.

The top and bottom wing cover panels are designed on a per panel basis, where each panel is constrained by two spars and two ribs. There is a maximum of 16 panels for the top wing cover of the wing-box and 16 panels for the bottom wing cover of the wing-box. A typical wing cover panel with positive force directions, orientation and dimensions is shown in Fig. 2. In the model each cover panel is represented by a single quadrilateral element if the internal spar and all seven internal ribs are present. If the number of internal spars and/or ribs is reduced some cover panels are represented by more than one quadrilateral element.

The number of design variables used to design each panel depends on the selected construction. For the sandwich construction there are two design variables for each panel as shown in Fig. 3. The design variables are the thickness of the wing cover $t$ and the thickness of the core $t_c$. There is thus a maximum of 80 structural sub-optimization design variables when considering the sandwich construction, which corresponds to 16 spar and rib thickness design variables and 64 top and bottom wing cover design variables. The core density is assumed to be $1.736\ 10^{-3}\ lb/in^3$.

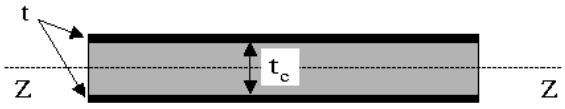


**Fig. 3 Sandwich construction**

For the hat-stiffened construction there are six design variables for each panel as illustrated in Fig. 4. The design variables are the number of hat stiffeners per panel $n$, the thickness of the wing cover $t_1$, the thickness of the stiffener $t_2$, and the dimensions of the stiffener $d_2$, $d_3$, and $d_4$. The number of stiffeners per panel is currently assumed to be a continuous variable that can be rounded to the next highest number in the final design. There is thus a maximum of 208 structural sub-optimization design variables when considering the hat-stiffened construction, which corresponds to the 16 spar and rib thickness design variables and 192 top and bottom wing cover design variables.

The structural sub-optimization problem is subject to both stress and local buckling constraints. Each element is subject to maximum Von Mises

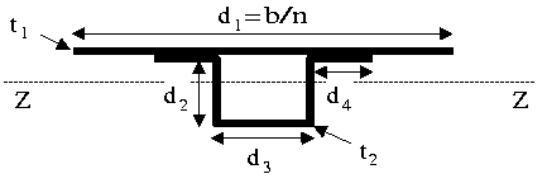


**Fig. 4 Hat-stiffened construction**

stress constraints at both the upper and lower surfaces. The local buckling constraints are applied to each upper and lower wing cover panel. For the sandwich construction (see Fig. 3) the local buckling equations are summarized in equations (10) through (12)

$$\frac{\sigma_1}{\sigma_{cr}} - 1 \leq 0 \tag{10}$$

$$\frac{\tau_{12}}{\tau_{cr}} - 1 \leq 0 \tag{11}$$

$$\frac{\sigma_1}{\sigma_{cr}} + \left(\frac{\tau_{12}}{\tau_{cr}}\right)^2 - 1 \leq 0 \tag{12}$$

where:

$$\begin{aligned} \sigma_1 &= \frac{N_1}{2t} \qquad \sigma_{cr} = k\frac{3.60\ E}{(b/t_{eq})^2} \\ \tau_{12} &= \frac{N_{12}}{2t} \qquad \tau_{cr} = k\frac{4.85\ E}{(b/t_{eq})^2} \\ t_{eq} &= \left(6t\left(t_c + t\right)^2 + 2t^3\right)^{1/3} \end{aligned} \tag{13}$$

In (13), $k$ is an additional safety factor that guards against delamination of the core from the face sheets. A safety factor of $k = 4.0$ is used throughout the present work.

For the hat-stiffened construction (see Fig. 4) the local buckling equations are summarized in equations (14) through (19)

$$\frac{N_1 d_1}{P_{cr}} - 1 \leq 0 \tag{14}$$

$$\frac{\sigma_1}{\sigma_{cr2}} - 1 \leq 0 \tag{15}$$

$$\frac{\sigma_1}{\sigma_{cr3}} - 1 \leq 0 \tag{16}$$

$$\frac{\sigma_1}{\sigma_{cr4}} - 1 \leq 0 \tag{17}$$

$$\frac{\tau_{12}}{\tau_{cr2}} - 1 \leq 0 \tag{18}$$

$$\frac{\sigma_1}{\sigma_{cr2}} + \left(\frac{\tau_{12}}{\tau_{cr2}}\right)^2 - 1 \leq 0 \tag{19}$$

where:

$$\begin{aligned} P_{cr} &= \frac{\pi^2 E I_{ZZ}}{a^2} \qquad w = d_1 - d_3 - d_4 \\ \sigma_1 &= \frac{N_1}{t_{eq}} \qquad \sigma_{cr2} = \frac{3.60\ E}{(w/t_1)^2} \\ \sigma_{cr3} &= \frac{3.60\ E}{(d_2/t_2)^2} \qquad \sigma_{cr4} = \frac{3.60\ E}{(d_3/t_2)^2} \\ \tau_{12} &= \frac{N_{12}}{t_1} \qquad \tau_{cr2} = \frac{4.85\ E}{(w/t_1)^2} \\ t_{eq} &= t_1 + nt_2\left(\frac{2d_2 + d_3 + 2d_4}{b}\right) \end{aligned} \tag{20}$$

The aerodynamic loads are applied as nodal forces to the bottom surface nodes only, including the non-structural leading and trailing edge nodes. The normalized span-wise and chord-wise distribution of the assumed forces are shown in Figs. 5 and 6.

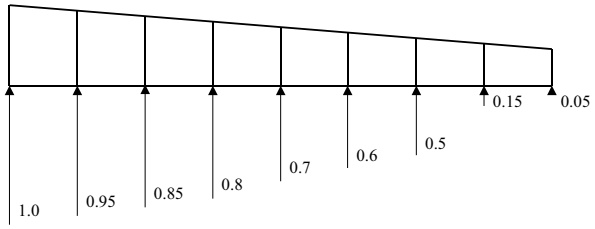


**Fig. 5 Normalized span-wise pressure distribution**

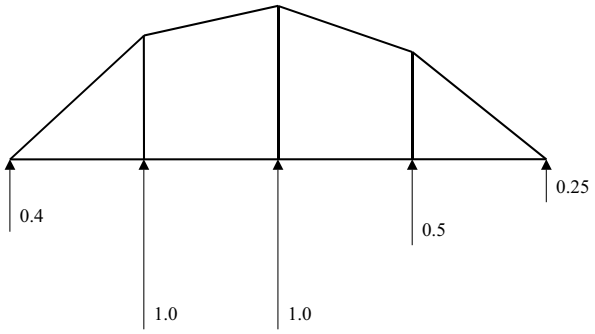


**Fig. 6 Normalized chord-wise pressure distribution**

The aerodynamic pressure distribution is converted to concentrated nodal forces, accounting for the areas associated with each node, using (21)

$$F_{i,j} = C_f\ LP_j\ CP_i\ S_{i,j} \tag{21}$$

where $i$ is the line number of the node and $j$ is the chord number of the node according to Fig. 7. Additionally, $C_f$ is the wing loading, which is 143 *lb/sf* for the current wing, $LP_j$ is the span-wise pressure

distribution value based on the line number from Fig. 6 and $CP_i$ is the chord-wise pressure distribution based on the chord number from Fig. 5. The area associate with each node is represented by $S_{i,j}$. The actual nodal forces are obtained by multiplying the forces obtained from (21) with a factor equal to the $G$ value of the maneuver. For the 3.75 $G$ maneuver this factor is 3.75 and for the -1.5 $G$ maneuver, the factor is -1.5.

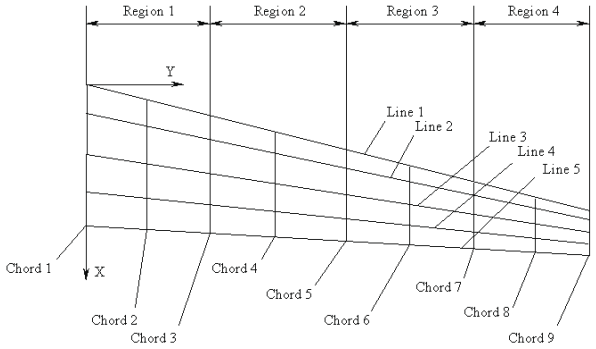


**Fig. 7 Wing structural design regions**

An important note regarding (21) should be made here. The aerodynamic nodal forces are influenced by the deformation of the wing. In the present work this interaction is not accounted for. Instead it is assumed that the wing will be build to a jig-shape that off-sets the deformation due to the aerodynamic loads.

**Aerodynamic Analysis**

The aerodynamic analysis is used to calculate the total drag required to calculate the range at the system level, using (9). A simplified drag calculation is used with the total drag ($D$) for the current wing calculated based on the total drag of the reference wing ($D_{ref}$). The calculations consists of the induced drag ($D_I$), the wave drag ($D_w$) and a constant fraction of the original drag, as shown in (22)

$$D = D_I + D_w + C_{dt} D_{ref} \tag{22}$$

where $C_{dt}$ is a constant and is assumed to be 0.4 in the present work. The total drag of the reference wing is $D_{ref} = 40000\ lb$.

The induced drag depends on the aspect ratio of the current wing ($A$) relative to the aspect ratio of the reference wing ($A_{ref}$), as follows:

$$D_I = C_{di} D_{ref} \frac{A_{ref}}{A} \tag{23}$$

$C_{di}$ is a constant and is assumed equal to 0.4 in the present work. The wave drag depends on the frontal area of the wing as projected on the streamline direction. The frontal area depends on the span and the mean height of the wing as follows

$$S_h = b \frac{(h_r + h_t)}{2} \tag{24}$$

where $S_h$ is the frontal area, $b$ is the span and $h_r$ and $h_t$ are the height at the root and the tip of the wing respectively. The wave drag is then obtained relative to the reference wing using (25)

$$D_w = C_{dw} D_{ref} \frac{S_h}{S_{href}} \tag{25}$$

where $C_{dw}$ is a constant assumed equal to 0.2 in the present work, $S_h$ is the frontal area of the current wing and $S_{href}$ is the frontal area of the reference wing.

**Geometric Issues**

The system level design variables include the aspect ratio and depth-to-chord ratio for a wing with constant area and sweep. These two design variables determine the nodal coordinates of the structural model through basic geometrical relationships. The Y-coordinate (see Fig. 1) is obtained from (26)

$$Y_{i,j} = Y_{i,j\ ref} \left( \frac{b}{b_{ref}} \right) \tag{26}$$

where $i$ and $j$ represents the line and chord numbers respectively as before (see Fig. 7), $Y_{i,j}$ represents the Y-coordinate for nodes in the current model and $Y_{i,j\ ref}$ represents the corresponding Y-coordinate in the reference model. The new X-coordinate $X$, depends on the sweep $p$ and is obtained from (27)

$$X_{i,j} = (X_{i,j\ ref} - p\ Y_{i,j\ ref}) \left( \frac{c_r}{c_{rref}} \right) + p\ Y_{i,j} \tag{27}$$

where $c_r$ is the root chord length of the current wing and $c_{rref}$ is the root chord length of the reference wing. Finally, the Z-coordinate, $Z$ is obtained from (28)

$$Z_{i,j} = Z_{i,j\ ref} \left( \frac{c_r}{c_{rref}} \right) \left( \frac{h/c}{(h/c)_{ref}} \right) \tag{28}$$

where $h/c$ is the depth-to-chord ratio of the current wing and $(h/c)_{ref}$ is the depth-to-chord ratio of the reference wing.

## Results

The PSO algorithm was used to solve the system level optimization. This is an unconstrained problem that aims to maximize the range of the wing by

changing the aspect ratio, the depth-to-chord ratio, the number of internal spars, the number of internal ribs and the wing cover construction. The PSO parameters used to solve the problem are summarized in Table 2. These parameters were selected from previous work by the authors[7] and no attempt was made to fine tune the algorithm to the current example problem. The main emphasis of the present work was not to fine tune the PSO algorithm to a specific example problem, but rather to investigate the utility of using the PSO algorithm in a multidisciplinary optimization environment.

**Table 2 PSO parameters**

| Parameter | Value |
|---|---|
| Number of particles | 100 |
| Initial inertia weight, $w$ | 1.4 |
| Trust parameter 1, $c_1$ | 1.50 |
| Trust parameter 2, $c_2$ | 2.50 |

Due to the stochastic nature of the algorithm, the optimization was repeated ten times, each using a different random seed resulting in a different initial population. For each repetition the best, worst, mean and standard deviation of both the best objective and the number of function evaluations to convergence were recorded.

The statistical results for the ten repetitions are summarized in Table 3. From Table 3 it is clear that the results are well converged between the ten independent repetitions with a standard deviation of only 19 *n. mi.* for the mean range of 5328.5 *n. mi.* The PSO algorithm is very successful at dealing with any local minima and/or numerical noise that may exist. However, this robustness comes at a fairly high average cost of 9660 analyses per optimization for the five design variable system level optimization problem. Part of this high cost is due to the characteristics of the algorithm itself, but part of it is related to the PSO parameters used. In particular it seems that the convergence criterion used is too strict. For convergence, the objective function had to change less the 0.1% in 5 consecutive iterations.

The best design point found from the ten repetitions is summarized in Table 4. The optimizer increased the range by increasing the aspect ratio of the reference wing by 34.7%, while decreasing the $h/c$ ratio by 10.8%. Additionally, the internal spar is removed and the number of internal ribs is reduced from 7 to 4, while the hat-stiffened construction is preferred. These changes result in a longer, thinner wing with a 6.9% increase in range to 5343.6 *n. mi.* Note that the structural sub-optimization problem

**Table 3 Cost and objective function (range) statistics**

| | Cost (*# Fn. Eval.*) | Range (*n. mi.*) |
|---|---|---|
| Mean | 9660 | 5328.5 |
| StdDev | 1826 | 19.0 |
| Best | 6700 | 5343.6 |
| Worst | 13300 | 5276.3 |

is also performed for the reference wing, so that the reference wing represents the optimum wing for the reference aspect ratio and depth-to-chord ratio.

**Table 4 Optimum design point**

| Parameter | Optimum | Reference |
|---|---|---|
| Range (*n. mi.*) | 5343.6 | 5000.0 |
| Aspect ratio | 9.2360 | 6.8571 |
| $h/c$ ratio | 0.1071 | 0.1200 |
| Internal spars | 0 | 1 |
| Internal ribs | 4 | 7 |
| Construction | Hat-Stiffened | – |

The spar and rib configuration for the optimum wing is shown in Fig. 8. As mentioned previously, the internal ribs are added from the root to the tip of the wing, while preserving the original rib spacing. The ribs are absent along the long outer wing segment because the optimization procedure finds that they are not needed for anti-buckling support. Of course, in a realistic wing box, such a long segment without ribs might not be acceptable because the ribs are also needed to support the wing cover panels, acting as plates loaded by the aerodynamic pressure. However, this effect was not modeled and therefore could not affect the course of the optimization.

The PSO algorithm successfully dealt with the truly discrete design variables. All ten optimization runs resulted in the same values for all the truly discrete design variables with only small changes in the aspect and depth-to-chord ratio. The PSO algorithm also successfully dealt with any spurious local minima that may exist as a result of numerical noise. To illustrate the numerical noise present in the current example problem, it was decided to change the aspect ratio from 5.0 to 10.0 with increments of 0.125 while fixing all the remaining system level design variables at their optimum values from Table 4. Each range calculation consists of a structural sub-optimization which may introduce numerical noise. The range of the aspect ratio was selected to include both the reference and optimum designs. The results

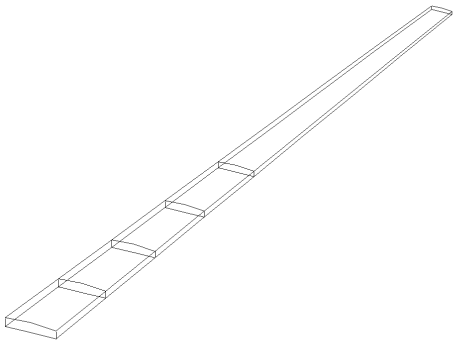

**Fig. 8 Spar and rib configuration for optimum wing-box**

of the numerical noise study is shown graphically in Fig. 9.

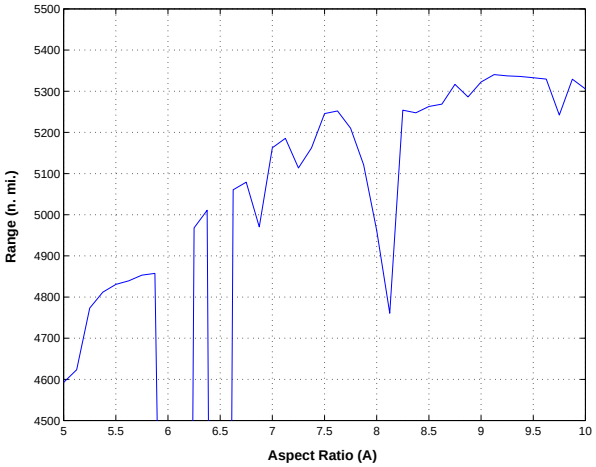


**Fig. 9 Range vs. aspect ratio**

From Fig. 9 it is clear that the aspect ratio has a significant impact on the range. Fig. 9 also illustrates the fact that the reference design represents a good design, hence the fairly small improvement found by the PSO algorithm. However, Fig. 9 also illustrates the severe numerical noise associated with this design problem. The data includes three data points ($A = 6.0$, $A = 6.125$ and $A = 6.5$) for which the structural sub-optimization could not find a feasible solution. The remaining numerical noise is a result of incomplete convergence in the sub-optimization. Although one could most probably reduce the numerical noise in the present example by forcing more strict convergence in the sub-optimization problem, a bi-level multidisciplinary optimization problem will in general be subject to numerical noise. In the present example the numerical noise is so bad that a gradient-based optimizer would have a tough time finding the true optimum design point without being trapped in a spurious local minimum. However, the PSO algorithm was able to overcome this numerical noise and reliably find the same optimum design point starting with ten different initial swarms.

## Computational Cost

Computational cost is always a concern when using a non-gradient based optimization algorithm. As shown in the present work, this class of algorithms provide the designer with many desirable features, but require many more function evaluations as compared to traditional gradient-based optimization algorithms.

A useful measure of the computational cost is to compare the cost of using the PSO algorithm against that of a gradient-based algorithm. For the present example problem, a single PSO run required an average of 9660 analyses to converge (see Table 3). To estimate the computational cost associated with a gradient-based algorithm, one needs to explicitly deal with the truly discrete design variables present in the current example problem, since no meaningful gradient information exist for these variables. There are three truly discrete design variables at the system level (the number of internal spars, the number of internal ribs and the wing cover construction type) that result in a total of 49 possible combinations. When using a gradient-based algorithm, it is possible to do a separate optimization for each of these combinations. The result is 49 independent optimization runs, each with two system level design variables that are continuous (the aspect ratio and the depth-to-chord ratio of the wing). For each sub-optimization problem, consisting of two continuous system level design variables, thirty analyses is a reasonable estimate for the computation cost required for convergence. The result is a total cost of 1470 analyses to complete a single optimization run.

If one compares the estimated cost of using a gradient-based optimization algorithm versus the average cost of using the PSO algorithm, it would seem that a gradient-based algorithm is the obvious choice for solving the present example problem. However, it is not a fair comparison to only look at the number of function evaluations as outlined above. First, a very stringent convergence criterion was used for the PSO algorithm. One can expect a dramatic reduction in computational cost for the PSO algorithm when addressing this issue. Second, and more importantly, is the presence of

severe numerical noise (see Fig. 9). The PSO algorithm was demonstrated to effectively and robustly deal with the presence of this numerical noise. In contrast, a gradient-based algorithm would not be able to deal with the numerical noise present in the current example problem and would easily be trapped in spurious local minima, resulting in false convergence. Third, the implementation of the proposed gradient-based optimization scheme consisting of 49 sub-optimization problems would place an additional burden on the designer. Finally, as the number of discrete combinations increase, either by adding discrete design variables or by have a larger set of discrete values for each design variable, it would quickly become infeasible to follow the all-combinations gradient-based approach outlined here. In contrast, the PSO algorithm only searches a small subset of all the possible discrete combinations, making the algorithm more attractive for larger numbers of discrete combinations.

Most non-gradient based algorithms parallelize well, resulting in a dramatic reduction of the overall time required to complete an optimization run. When considering the computational cost of these algorithms, it is thus important to also look at the ability of using large numbers of concurrent processors. For the present example, where a swarm size of 100 was used, the PSO algorithm requires an average of 96.6 design iterations to converge. If 100 processors are available, the average time to complete a single PSO optimization would be equivalent to the real time of 96.6 analyses. Of course, the proposed scheme for solving this problem using a gradient-based algorithm can also be parallelized by running each of the 49 optimization runs on a separate processor. In this case a single optimization, consisting of 49 sub-optimization runs, would complete in the real time of 30 analyses.

## Conclusions

The results presented clearly illustrates the usefulness of the PSO algorithm in a bi-level multidisciplinary optimization environment. The PSO algorithm is able to reliably find the same optimum design point, despite the presence of truly discrete variables and severe numerical noise. Although using the PSO algorithm is clearly more expensive than a gradient-based implementation, the advantages of directly dealing with the truly discrete design variables and the severe numerical noise present in the current example problem, makes the PSO algorithm an attractive alternative. For problems with a larger number of discrete combinations, the PSO algorithm should be a more appropriate choice than a gradient-based implementation similar to that proposed in Section . Additionally, exploiting a large number of concurrent processors could dramatically reduce the overall computational time required to complete the PSO optimization. A future investigation into the PSO parameters, specifically those related to the convergence criterion, and their influence on the efficiency of the algorithm would be appropriate.

## References


[1]Michalewicz, Z. and Dasgupta, D., editors, *Evolutionary Algorithms in Engineering Applications*, Springer Verlag, 1997.

[2]Nemhauser, G. L. and Wolsey, L. A., *Integer and Combinatorial Optimization, Chapter 3*, John Wiley & Sons, 1988.

[3]Kennedy, J. and Eberhart, R. C., "Particle Swarm Optimization," *Proceedings of the 1995 IEEE International Conference on Nerual Networks, Perth, Australia*, 1995, pp. 1942–1948.

[4]Eberhart, R. C. and Kennedy, J., "A New Optimizer Using Particles Swarm Theory," *Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan*, 1995, pp. 39–43.

[5]Fourie, P. C. and Groenwold, A. A., "Particle Swarms in Size and Shape Optimization," *Proceedings of the International Workshop on Multidisciplinary Design Optimization, Pretoria, South Africa*, August 7-10 2000, pp. 97–106.

[6]Fourie, P. C. and Groenwold, A. A., "Particle Swarms in Topology Optimization," *Extended Abstracts of the Fourth World Congress of Structural and Multidisciplinary Optimization, Dalian, China*, June 4-8 2001, pp. 52–53.

[7]Venter, G. and Sobieszczanski-Sobieski, J., "Particle Swarm Optimization," *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, CO*, Vol. AIAA-2002-1235, April 22-25 2002.

[8]Kennedy, J. and Spears, W. M., "Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator," *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, Alaska*, May 4-9 1998.

[9]Shi, Y. and Eberhart, R. C., "A Modified Particle Swarm Optimizer," *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, Alaska*, May 4-9 1998.

[10]Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design*, Vanderplaats Research and Development, Inc., 3rd ed., 1999.

[11]Sobieszczanski-Sobieski, J. and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *34th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV*, Vol. AIAA-1996-0711, January 15-18 1996.

[12]Garcelon, J. H., Balabanov, V. O., and Sobieski, J., "Multidisciplinary Optimization of a Transport Aircraft Wing using VisualDOC," *Proceedings of the AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, St. Louis, MO*, Vol. AIAA-1999-1349, April 12-15 1999, pp. 1306–1313.

[13]*GENESIS Version 7.0 Users Manual*, Vanderplaats Research and Development, Inc., 2001.