2008

Implementation of Digital Pheromones in Particle Swarm Optimization for Constrained Optimization Problems

Vijay Kalivarapu
Iowa State University, vkk2@iastate.edu

Eliot H. Winer
Iowa State University, ewiner@iastate.edu

Implementation of Digital Pheromones in Particle Swarm Optimization for Constrained Optimization Problems

Abstract

This paper presents a model for digital pheromone implementation of Particle Swarm Optimization (PSO) to solve constrained optimization problems. Digital pheromones are models simulating real pheromones produced by insects for communication to indicate a source of food or a nesting location. When integrated within PSO, this principle of communication and organization between swarm members offer substantial improvement in search accuracy, efficiency and reliability. Multiple pheromones are released in the design space, and the strength of a pheromone in a region of the design space is determined through empirical proximity analysis, and. The swarm then reacts accordingly based on the probability that this region may contain an optimum. The addition of a pheromone component to the velocity vector equation demonstrated substantial success in solving unconstrained problems. The research presented in this paper explores the suitability of the developed method to solve constrained optimization problems. A sequential unconstrained minimization technique – Augmented Lagrange Multiplier (ALM) method has been implemented to address constrained optimization problems. ALM has been chosen because of its relative insensitivity to whether the initial design points for a pseudo objective function are feasible or infeasible. The development of the method and results from solving several constrained test problems are presented.

Keywords

Particle Swarm Optimization (PSO), Augmented Lagrange Multiplier (ALM), Virtual Reality Applications Center, digital pheromones

Disciplines

Computer-Aided Engineering and Design | Mechanical Engineering

Comments

This is a conference proceeding from *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (2008): AIAA 2008-1974, doi: 10.2514/6.2008-1974. Posted with permission.

49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference
 16t
7 - 10 April 2008, Schaumburg, IL

AIAA 2008-1974

Implementation of Digital Pheromones in Particle Swarm Optimization for Constrained Optimization Problems

Vijay Kalivarapu* and Eliot Winer†
Iowa State University, Ames, IA, 50011, USA

This paper presents a model for digital pheromone implementation of Particle Swarm Optimization (PSO) to solve constrained optimization problems. Digital pheromones are models simulating real pheromones produced by insects for communication to indicate a source of food or a nesting location. When integrated within PSO, this principle of communication and organization between swarm members offer substantial improvement in search accuracy, efficiency and reliability. Multiple pheromones are released in the design space, and the strength of a pheromone in a region of the design space is determined through empirical proximity analysis, and. The swarm then reacts accordingly based on the probability that this region may contain an optimum. The addition of a pheromone component to the velocity vector equation demonstrated substantial success in solving unconstrained problems. The research presented in this paper explores the suitability of the developed method to solve constrained optimization problems. A sequential unconstrained minimization technique – Augmented Lagrange Multiplier (ALM) method has been implemented to address constrained optimization problems. ALM has been chosen because of its relative insensitivity to whether the initial design points for a pseudo objective function are feasible or infeasible. The development of the method and results from solving several constrained test problems are presented.

I. Introduction

PSO [1,2] is a population based heuristic method retaining many characteristics of evolutionary search algorithms such as GA and SA. It is a recent addition to global search methods [3] and one of its key features is its simplicity in implementation due to a small number of parameters to adjust [4, 5]. In a regular PSO, an initial randomly generated population swarm (a collection of particles) propagates towards the global optimum over a series of iterations. Each particle in the swarm explores the design space based on the information provided by two members – the best position of a swarm member in its history trail (*pBest*), and the best position attained by all particles (*gBest*) until that iteration. This information is used to generate a velocity vector indicating a search direction towards a promising design point, and the location of each swarm member is updated. However, the drawback of this approach is that information from these two members alone is not sufficient for the swarm to propagate toward the global optimum efficiently. This either could cause the swarm to lock into a local minimum or take very long time to reach the global optimum.

Previous work by the authors demonstrated promising performance improvement of PSO in terms of increased solution accuracy and decreased solution times through implementing digital pheromones in PSO [6, 7] for solving unconstrained problems. A quantitative assessment has also been made through statistical hypothesis testing [8]. This paper focuses on exploring the suitability of digital pheromone implementation in PSO for solving constrained optimization problems. The development of this method, followed by results from testing it with benchmarking problems is presented.

II. Background

A. Particle Swarm Optimization

* Research Assistant, Department of Mechanical Engineering, Human Computer Interaction, Virtual Reality Applications Center, 2274 Howe Hall, Iowa State University, Ames, IA, 50011, USA, Student Member.
† Assistant Professor, Department of Mechanical Engineering, Human Computer Interaction, Virtual Reality Applications Center, 2274 Howe Hall, Iowa State University, Ames, IA, 50011, USA, Member.

PSO shares many characteristics of evolutionary search algorithms such as Genetic Algorithms (GA) and Simulated Annealing (SA) – a) Initialization with a population of random solutions, b) Design space search for optimum through updating generations and c) Update based on previous generations [9]. The success of the algorithm has brought substantial attention among the research community in the recent past [10, 11]. The working of the algorithm is based on a simplified social model similar to the swarming behavior exhibited by insects and birds. In this analogy, a swarm member uses its own memory and the behavior of the rest of the swarm to determine the suitable location of food (global optimum). The algorithm iteratively updates the direction of the swarm movement toward the global optimum. The mathematical formulation of the method is given in Equations (1) and (2).

$$V_{i+1} = w_i * V_i + c_1 * rand_p() * (pBest_i[] - X_i[]) + c_2 * rand_g() * (gBest[] - X_i[]) \tag{1}$$

$$X_{i+1} = X_i + V_{i+1} \tag{2}$$

$$w_{i+1} = w_i * \lambda_w \tag{3}$$

'*pBest*' represents the best position attained by a swarm member in its history trail, and '*gBest*' represents the best position attained by the swarm in the entire iteration history. Equation (1), represents the velocity vector update of a traditional PSO method where $rand_p()$ and $rand_g()$ are random numbers generated between 0 and 1 each for *pBest* and *gBest*. c_1 and c_2 are confidence parameters. w_i is called as the inertia weight [12, 13] and decreases in every iteration by a factor of λ_w, as represented in Equation (3). Equation (2) denotes the updated swarm location in the design space.

In addition to the originally developed PSO algorithm, significant enhancements have been proposed such as: a) mutation factors for better design space exploration [14, 15], b) methods for constraint handling [16, 17], c) parallel implementation [18, 19], d) methods for solving multi-objective optimization problems [20], e) methods for solving mixed discrete, integer and continuous variables [21].

B. PSO and Digital Pheromones

Pheromones are chemical scents produced by insects to communicate with each other to find a suitable food source, nesting location, etc. The stronger the pheromone, the more the insects are attracted to the path. A digital pheromone is analogous to an insect generated pheromone in that they are the markers to determine whether or not an area is promising for further investigation. One of the well-known applications of digital pheromones is its use in the automatic adaptive swarm management of Unmanned Aerial Vehicles (UAVs) [22, 23]. In this research, the UAVs are automatically guided towards a specific zone or target through releasing digital pheromones in a virtual environment, thereby reducing the requirement of humans physically controlling from ground stations. Other applications of digital pheromones include ant colony optimization for solving minimum cost paths in graphs [24, 25] solving network communication problems [26]. The concept of digital pheromones is considerably new [27] and has not yet been explored to its full potential for investigating n-dimensional design spaces for locating an optimum.

In a regular PSO algorithm, the swarm movement obtains design space information from only two components – *pBest* and *gBest*. When coupled with an additional pheromone component, the swarm is essentially presented with more information for design space exploration and has a potential to reach the global optimum faster.

C. Constrained Optimization

Realistic design problems are characterized by numerous inequality and equality constraints. Traditionally, these problems are solved numerically using sequential unconstrained minimization techniques such as interior and exterior penalty function methods. The exterior penalty function method is the simplest to implement, which penalizes the objective function when constraints are violated. A notable disadvantage with the method is that any optimization routine that is stopped prematurely could be unusable because the design points move from infeasible to feasible regions, and hence a selected design point potentially violates the constraints. The interior penalty function, on the other hand, penalizes the objective function as the design points approach constraints, and violations are not allowed. Although the design points in this method are always in the feasible region and improving every iteration, it comes at a cost of creating complicated minimization problem, and potential difficulties leading to discontinuities in the pseudo objective function. Extended interior penalty function methods (linear extended penalty function [28,29], quadratic extended penalty function [30], variable penalty function methods [31]) incorporate the best features of interior and exterior penalty function methods. Augmented Lagrange Multiplier (ALM) method is another class of sequential unconstrained minimization technique to solve constrained problems and have the following advantages: a) The penalty parameters do not need to be increased to infinity for convergence, b) The starting design vector need not necessarily be feasible, and c) The non-zero values of Lagrange multipliers identify active constraints automatically. This method offers considerable advantage over other constrained minimization techniques and hence used in this research to solve constrained optimization problems.

Previous research on digital pheromones for use in PSO has produced significant improvement strides in terms of solution quality and solution times. The efficacy of digital pheromones in PSO in solving constrained optimization problems has been studied through the implementation of ALM method. The methodology for development and results from solving various test problems are reported in this paper.

III. Methodology

A. Overview of digital pheromones in PSO

Figure 1 summarizes the procedure for PSO, with steps involving digital pheromones highlighted.

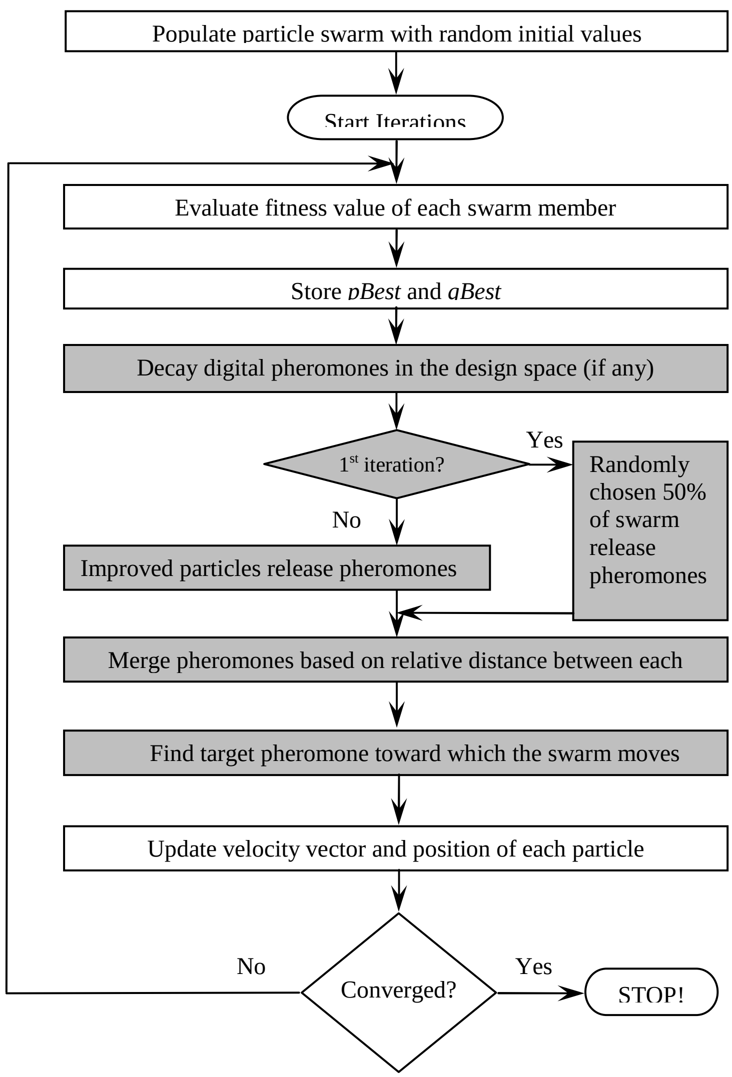

Figure 1 Overview of PSO with Digital Pheromones

The initialization of pheromone-based PSO is similar to a basic PSO except that 50% of the swarm is randomly selected to release pheromones in the first iteration. This parameter is user defined, but experimentation has shown 50% to be a good default value. For subsequent iterations, each swarm member that finds a better objective function releases a pheromone. Pheromones (from current as well as past iterations) that are close to each other in terms of design variable values are merged into a new pheromone location. This effectively creates a pheromone field across the design space while still keeping the number of pheromones manageable. Based on the pheromone level and its position relative to a particle, a probability is then used in a ranking process to select a target pheromone for each particle in the swarm. The target position for each particle will be a third component of the velocity vector update in addition to *pBest* and *gBest*. Following this, the objective value for each particle is recalculated and the entire process continues until the convergence criteria is satisfied.

Digital Pheromones and Merging

In order to populate the design space with an initial set of digital pheromones, 50% of the population is randomly selected to release pheromones, regardless of the objective function value. This is done so as to ensure a good design space exploration by the particle swarm in the initial stages of the optimization process. For subsequent iterations, the objective function value for each particle in the population is evaluated and only particles finding an improvement in the objective function value will release a pheromone. Any newly released pheromone is assigned a level *P*, with a value of 1.0. Just as natural pheromones produced by insects decay in time, a user defined decay rate, λ_P, defaulting to 0.95, is assigned to the pheromones released by the particle swarm. Digital pheromones are decayed as the iterations progress forward to allow the swarm to propagate toward a better design point instead of getting attracted to an older pheromone, which may not be a good design point.

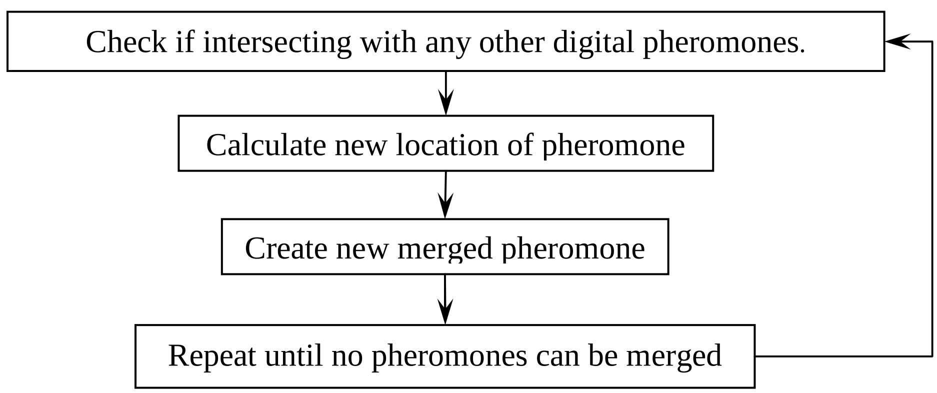

Figure 2 Illustration of pheromone merging process

Every particle that finds a solution improvement releases a pheromone potentially making the pheromone pool unmanageably large. Therefore, an additional step to reduce them to a manageable number, yet retaining the functionality, is implemented. Pheromones that are closely packed within a small region of the design space are merged together. To check for merging, each pheromone is associated with an additional property called 'Radius of Influence' (*ROI*). For each design variable of a pheromone, an *ROI* is computed and stored. The value of this *ROI* is a function of the pheromone level and the bounds of the design variables. Any two pheromones for a design variable less than the sum of the ROIs are merged into one. This is analogous to two spheres merging into one if the distance between them is less than the sum of their radii. A resultant pheromone level is then computed for the merged pheromones. Through this approach, regions of the design space with stronger resultant pheromone levels will attract more particles and therefore, pheromones that are closely packed would indicate a high chance of optimality. Also similar to the pheromone level decay, the *ROI* also has its own decay factor, λ_{ROI}, whose value is set equal to λ_P as a default. This is to ensure that both the pheromone levels and the radius of influence decay at the same rate. Figure 2 illustrates the pheromone merging process.

Attraction to a Target Digital Pheromone

With numerous digital pheromones generated within the design space, a swarm member needs to identify which pheromone it will be attracted to most. The criteria for generating this target pheromone are: a) small magnitude of distance from the particle and b) high pheromone level. To rank which digital pheromone from the pheromone pool fits this criteria, a target pheromone attraction factor *P'* is computed. The value of *P'* is a product of the normalized distance between that pheromone and the particle, and its pheromone level. Also, the attraction factor must increase when the pheromones are closer to the particles. Therefore, the attraction factor is computed as shown in Equation (5). Equation (6) computes the distance between the pheromone and each particle in the swarm. Figure 3 shows an example scenario of a particle being attracted to a target pheromone.

$$P' = (1-d)P \tag{5}$$

$$d = \sqrt{\sum_{1}^{k}\left(\frac{Xp_k - X_k}{range_k}\right)^2}, k = 1:n \quad \text{\# of design variables} \tag{6}$$

Xp – Location of pheromone

X – Location of particle

In the figure, the particle will be more attracted to a pheromone with a higher *P'* value, as opposed to pheromones that are closer but with a lower *P'* value.

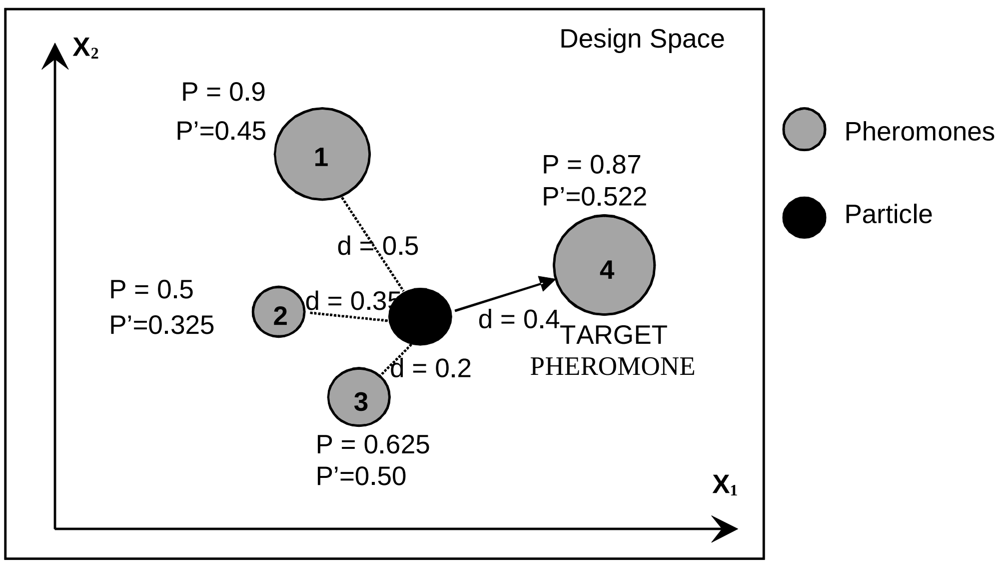

Figure 3 Illustration of Target Pheromone Selection

Velocity Vector Update

The velocity vector update implements the pheromone component as a third term in addition to the *pBest* and *gBest* components in a traditional PSO. This is shown in Equation (7).

$$V_{i+1} = w_i * V_i + c_1 * rand_p() * (pBest_i[] - X_i[]) + c_2 * rand_g() * (gBest[] - X_i[]) + c_3 * rand_T() * (Target[] - X_i[]) \tag{7}$$

c_3 is the confidence parameter for the pheromone component of the velocity vector, and is typically set to be larger than or equal to c_1 and c_2. This is done in order to increase the influence of pheromones in the velocity vector. From experimentation, it was found that a default value of 2.0 - 5.0 sufficed for most problems.

Move Limits, *ML*

The additional pheromone term in the velocity vector update, can considerably increase the computed velocity. To avoid this value from becoming unmanageably large, a move limit is imposed. The move limit is set to an initial value and reduced gradually as the iterations progress forward. This ensures a fair amount of freedom in exploration in the beginning and as the method approaches a solution, a smaller move limit exploits the current design point of a particle for a more constrained search towards an optimum. Although this is a user defined parameter, an initial set value of 10% of the design space for the move limit showed good performance characteristics. A default decay factor, λ_{ML} of value 0.95 was used.

B. Augmented Lagrange Multiplier Method

Traditionally, penalty function methods are most commonly used in numerical methods to address constrained problems. In these methods, the original constrained problem is typically substituted by a sequence of unconstrained sub-problems, called pseudo objective functions, and are solved sequentially through imposing penalties to limit constraint violations. A pseudo objective function is shown in eq. 8, where F(X) is the original objective function and P(X) is the penalty function whose form depends on the SUMT technique used (exterior penalty function methods, interior penalty function methods, extended penalty function methods, etc). 'r_p' is a scalar that determines the magnitude of the penalty imposed on constraint violations.

$$\Phi(X) = F(X) + r_p P(X) \tag{8}$$

The exterior penalty function methods typically yield feasible optimum values for extremely large r_p values but potentially yields numerically ill-conditioned formulations, and hence are generally avoided in numerical methods, especially population based heuristic methods. On the other hand, interior penalty function methods have the potential to reach discontinuous spaces, especially at constraint boundaries.

The Augmented Lagrange Multiplier (ALM) method was originally developed for addressing equality constrained problems and later extended to solve inequality constraints. The primary advantage of this method lies in that the penalty parameters do not require reaching infinity for convergence. Moreover, it is immaterial whether the design points originate in the feasible or in the infeasible region. Just as any other penalty function method, a penalty factor of 'r_p', on each constraint, is imposed. The advantage of ALM comes with the fact that there are only finite penalty factors using which the pseudo objective function A(X, λ, r_p) is optimized, and hence the original objective function F(X). With appropriate Lagrange multipliers (λ) known, one unconstrained minimization of the pseudo objective function is sufficient. Since these multipliers and penalty factors r_p are typically unknown before hand, a series of unconstrained minimizations are carried out to arrive at the appropriate Lagrange multipliers and hence the solution of the original objective function. The general augmented Lagrangian is shown in eq. (9).

$$A(X,\lambda,r_p) = F(X) + \sum_{j=1}^{m}[\lambda_j \Psi_j + r_p \Psi^2{}_j] + \sum_{k=1}^{l}\{\lambda_{k+m} h_k(X) + r_p [h_k(X)]^2\} \tag{9}$$

The first summation term after F(X) in eq. 9 correspond to inequality constraints where, ψ is given by eq. (10). The second summation term in eq. (9) correspond to the penalty function term for equality constraints. The update relations for Lagrange multipliers, λ are shown in eq. (11) for inequality constraints and eq. (12) for equality constraints.

$$\Psi_j = \max\left[g_j, \frac{-\lambda_j^p}{2r_p}\right] \tag{10}$$

$$\lambda_j^{p+1} = \lambda_j^p + 2r_p \max\left[g_j, \frac{-\lambda_j^p}{2r_p}\right], j = 1, m \tag{11}$$

$$\lambda_j^{p+1} = \lambda_j^p + 2r_p h_k(X), k = 1, l \tag{12}$$

The above characteristics make ALM a very favorable candidate for implementation in PSO with digital pheromones. Therefore, a series of unconstrained problems are required to be formulated and solved completely before arriving at the appropriate Lagrange multipliers and hence the design variable set that optimizes the original objective function. However with a reasonable approximation in solving the pseudo objective function, it is possible to arrive at the optimum of the original objective function quicker. This can be achieved through a loose convergence criterion or imposing a limited number of iterations on solving the pseudo objective function. The advantage of this approach lies in that improved Lagrange multipliers and penalty values can be obtained more frequently. The description for ALM was well explained in Gill et al [32], and has been used as the basis for the development of the method. Figure 4 shows a flow chart of the implemented methodology.

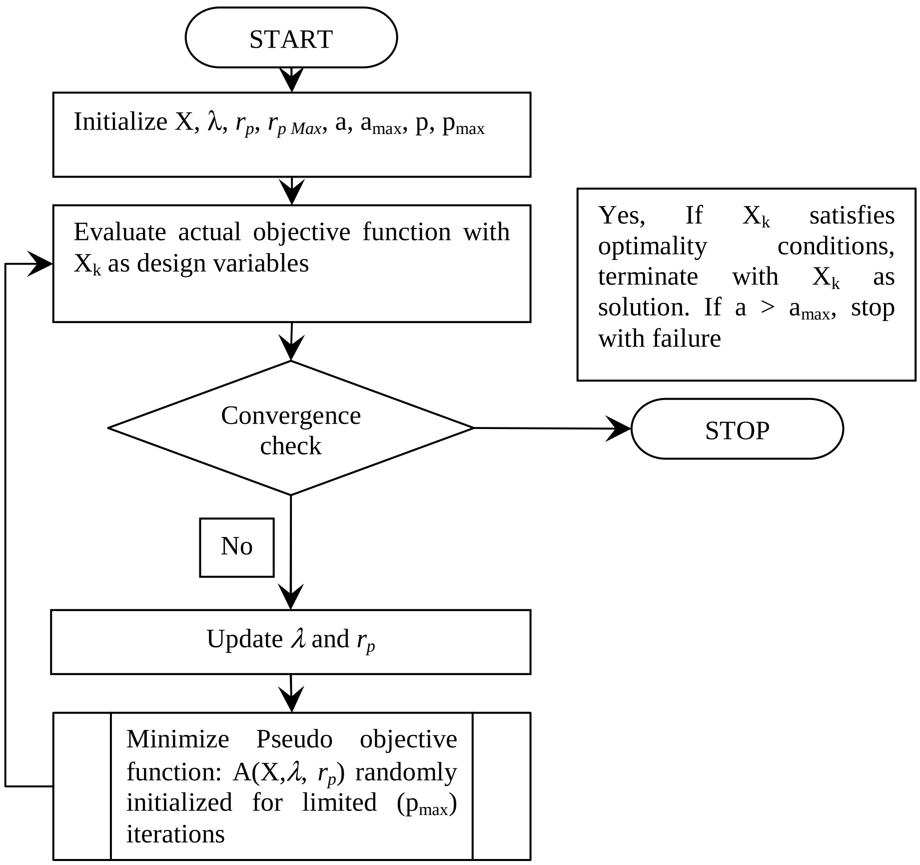

Figure 4 Flowchart for ALM implementation in PSO with digital pheromones

The figure shows the procedure overview of ALM implementation in PSO with digital pheromones. At the beginning, a population swarm with an initial selection of design variables (**X**), Lagrange multiplier $\boldsymbol{\lambda}$, and penalty parameter $\mathbf{r_p}$ for each of the design constraints, a positive integer a_{max}, serving as an upper bound of the total number of unconstrained minimizations and p_{max} to limit the number of iterations during pseudo objective function minimization are initialized. Upon evaluating the actual objective function value, a convergence check is performed to determine if it attained the optimum subjected to various design constraints. The algorithm stops with a success if a convergence is achieved or stops with a failure if the number of unconstrained minimizations exceed the maximum limit (a_{max}). The values of $\boldsymbol{\lambda}$ and $\mathbf{r_p}$ are updated if a convergence is not achieved. With updated $\boldsymbol{\lambda}$ and $\mathbf{r_p}$ values, a new pseudo objective function is therefore built and is solved using digital pheromone implementation of PSO. The pseudo objective function is minimized for a limited number of iterations with design variables for each pseudo objective function randomly initialized as outlined in the flow chart shown in figure 1. This procedure is then iterated through providing the solution ($\mathbf{X_{k+1}}$) as an input to evaluate the actual objective function and convergence check.

When the constraints are satisfied during the convergence check, the penalty values are decreased by a factor of 0.5 and increased by 2.0 when they are violated. This increase and decrease schemes are applied to both inequality and equality constraints as well. The specifications of the penalty parameter update are adopted from Sedlaczek and Eberhard [33]. Equations (13) and (14) portray these update schemes, and the lowest values for $\mathbf{r_p}$ are bounded at 1.0.

$$r_{p,j}^{a+1} = \begin{cases} 2\times r_{p,j}^{a} & if: g_j(x^a) > g_j(x^{a-1}) \quad \text{OR} \quad g_j(x^a) > \varepsilon_g \\ \frac{1}{2}\times r_{p,j}^{a} & if: g_j(x^a) \le \varepsilon_g \\ r_{p,j}^{a} & \text{Otherwise} \end{cases} \tag{13}$$

$$r_{p,k}^{a+1} = \begin{cases} 2\times r_{p,j}^{a} \quad if: \ |h_j(x^a)| > |h_j(x^{a-1})| \quad \text{OR} \quad |h_j(x^a)| > \varepsilon_h \\ \frac{1}{2}\times r_{p,k}^{a} \quad if: |h_j(x^a)| \le \varepsilon_h \\ r_{p,k}^{a} \qquad \text{Otherwise} \end{cases} \tag{14}$$

IV. Results

Three constrained test problems were run to evaluate the developed method. The algorithm was programmed using C++ and executed on a RedHat Linux workstation with an Intel dual core processor (3.2 GHz) and 2GB of system memory. The descriptions of the test cases are as below:

Test Problem 1: Two dimensional, one equality constrained problem

$$\begin{aligned} &F(X) = X_1^{\,2} + X_2^{\,2} \\ &h_1(X): X_1 + X_2 - 1 = 0 \\ &-5 \le X_i \le 5, \ i = 1, 2 \end{aligned}$$

The published solution for this problem is 0.5 and the solution set X* is: {0.5, 0.5}.

Test Problem 2: Four dimensional, eight inequality constraint weld-beam problem [34]

$$\begin{aligned} &F(X) = 1.1047 X_1^{\,2} X_2 + 0.04811 X_3 X_4 (14.0 + X_2) \\ &\textit{Subjected to}: \\ &g_1(X): \tau(X) - \tau_{\max} \le 0 \\ &g_2(X): \sigma(X) - \sigma_{\max} \le 0 \\ &g_3(X): X_1 - X_4 \le 0 \\ &g_4(X): 0.10471 X_1^{\,2} + 0.04811 X_3 X_4 (14.0 + X_2) - 5.0 \le 0 \\ &g_5(X): 0.125 - X_1 \le 0 \\ &g_6(X): \delta(X) - X_1 \le 0 \\ &g_7(X): P - P_c(X) \le 0 \\ &0.1 \le X_i \le 2.0, \quad i = 1, 4 \\ &0.1 \le X_i \le 10.0, \quad i = 2, 3 \end{aligned}$$

Where

$$\tau(X)=\sqrt{(\tau')^2+2\tau'\tau''\frac{X_2}{2R}+(\tau'')^2}$$

$$\tau'=\frac{p}{\sqrt{2}X_1X_2},\tau''=\frac{MR}{J},M=P\left(L+\frac{X_2}{2}\right)$$

$$R=\sqrt{\frac{X_2^2}{4}+\left(\frac{X_1+X_3}{2}\right)^2}$$

$$J=2\left\{\sqrt{2}X_1X_2\left[\frac{X_2{}^2}{12}+\left(\frac{X_1+X_3}{2}\right)^2\right]\right\}$$

$$\sigma(X)=\frac{6PL}{X_4X_3^2}$$

$$\delta(X)=\frac{4PL^3}{EX_3{}^3X_4}$$

$$P_C(X)=\frac{4.013\sqrt{E(X_3^2X_4^6/36)}}{L^2}\left(1-\frac{X_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P=6000lb, L=14in., E=30\times10^6\ psi, G=12\times10^6\ psi$$

$$\tau_{\max}=13{,}600\,psi,\sigma_{\max}=30{,}000\,psi,\delta_{\max}=0.25in.$$

The published solution for this problem is 2.386 and the solution set X* is: {0.2455, 6.1960, 8.2730, 0.2455}.

Test Problem 3: Five design variable, six inequality constraint Himmelblau Problem [35]

$$F(X)=5.3578547X_1{}^2+0.8356891X_1X_5+37.2932239X_1-40792.141)$$

$$Subjected\ to:$$

$$g_1(X):85.334407+0.0056858X_2X_5+0.00026X_1X_4-0.0022053X_3X_5-92.0\le0$$

$$g_2(X):-1\times(85.334407+0.0056858X_2X_5+0.00026X_1X_4-0.0022053X_3X_5)\le0$$

$$g_3(X):80.51249+0.0071317X_2X_5+0.0029955X_1X_2+0.0021813X_3^2-110.0\le0$$

$$g_4(X):-1\times(80.51249+0.0071317X_2X_5+0.0029955X_1X_2+0.0021813X_3^2)\le0$$

$$g_5(X):9.300961+0.0047026X_3X_5+0.0012547X_1X_3+0.0019085X_3X_4-25.0\le0$$

$$g_6(X):-1\times(9.300961+0.0047026X_3X_5+0.0012547X_1X_3+0.0019085X_3X_4)\le0$$

$$78\le X_1\le102,\ 33\le X_2\le45,\ 27\le X_3\le45,\ 27\le X_4\le45,\ 27\le X_4\le45$$

The published solution for this problem is -31025.56142 and the solution set X* is: {78.0, 27.0, 27.070997, 45.0, 44.96924255}.

The following PSO and digital pheromone parameters were used to run all test cases:

- c_1=c_2=2.0
- c_3=2.0 with no decay
- Pheromone decay = 0.95
- Move limit decay = 0.95
- Inertia weight initialized at 1.0 and decreased at 0.5% every iteration

- Particle swarm size of 4 times the number of design variables

The description of results from solving the test problems are shown in table 1.

Table 1 Results obtained from ALM on a swarm size of 4 times the number of design variables

Test Problem	Published Solutions	Solution Obtained	X*
1	0.50	0.508	{0.41, 0.58}
2	2.386	2.147	{0.29, 3.04, 7.47, 0.30}
3	-31025.561	-30,370.751	{87.5, 34.37, 28.01, 43.15, 40.18}

It can be seen from the table that the results were very close to the published solutions. The solutions attained did not require the penalty parameters to approach infinity thereby avoiding the problem of ill-conditioning. The method also identified active constraints for the test problems. The weld-beam problem (test problem 2) yielded a solution (2.147) better than the published solution (2.386) in one of the test runs. It was theorized that the reason could be attributed to the use of digital pheromones, which provided good information about the design space to the swarm.

The developed method however needs improvement in regards to solution reliability. While all the problems were solved, it took multiple runs in some cases to reach the published solutions. The primary reason for this behavior is attributed to the imbalance in the values returned for the actual and the pseudo objective function. When the actual objective function dominates the solution (or in other words when the actual objective function value is too low when compared to the pseudo objective function value), convergence takes place prematurely resulting in a poor quality solution. On the flipside, when the pseudo objective function dominates the solution (or in other words when the value of the pseudo objective function becomes too high), the swarm is typically exploring the design space in infeasible spaces resulting in higher penalties. This causes the swarm not to converge and the method returns with a failure. Although ALM has excellent properties over other penalty function based constrained optimization, it was theorized that the imbalance between actual and pseudo objective function values caused inconsistency in the results.

V. Conclusion

In this paper, a model for solving constrained optimization problems using PSO with digital pheromone implementation. Specifically, the Augmented Lagrange Multiplier method was implemented to extend the functionality of PSO with digital pheromones. The results from test cases indicate that the developed method can solve reasonably complex problems with decent solution accuracy. Although a satisfactory consistency was not achieved, this preliminary implementation shows promise that ALM implementation can be applied to PSO with digital pheromones and did not require penalty parameters to reach infinity. Imbalances between the actual and pseudo objective function values are theorized to cause this problem. Consistency and reliability concerns being the immediate future work, refining the performance of digital pheromones to solve a wide range of problems is an ongoing venture. Other future directions include performing statistical significance tests for unbiased evaluation of the method's performance.

References

[1] Kennedy, J., and Eberhart, R. C., "Particle Swarm Optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Vol. 4, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1995, pp. 1942-1948.

[2] Eberhart, R. C., and Kennedy, J., "A New Optimizer Using Particle Swarm Theory", *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1995, pp. 39-43.

[3] Russell C. Eberhart and Yuhui Shi, "Particle swarm optimization: Developments, applications, and resources", *In Proceedings of the 2001 Congress on Evolutionary Computation 2001*, 81–86.

[4] J.F. Schutte. Particle swarms in sizing and global optimization. Master's thesis, University of Pretoria, Department of Mechanical Engineering, 2001.

[5] A. Carlisle and G. Dozier. An off-the-shelf pso. In Proceedings of the Workshop on Particle Swarm Optimization, 2001, Indianapolis.

[6] Kalivarapu, V., Foo, J., Winer, E. , "Improving Solution Characteristics of Particle Swarm Optimization Using Digital Pheromones", Structural and Multidisciplinary Optimization Journal, DOI 10.1007/s00158-008-0240-9, January 2008.

[7] Kalivarapu, V., Foo, J., Winer, E., "A Parallel Implementation of Particle Swarm Optimization Using Digital Pheromones", 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA-2006-6908-694, Portsmouth, VA, September 2006.

[8] Kalivarapu, V., Winer, E., "A Statistical Analysis of Particle Swarm Optimization With and Without Digital Pheromones", 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 23-26, 2007, Honolulu, HI, AIAA 2007-1882.

[9] Hu X H, Eberhart R C, Shi Y H., "Engineering Optimization with Particle Swarm", *IEEE Swarm Intelligence Symposium*, 2003: 53-57.

[10] G. Venter and J. Sobieszczanski-Sobieski, "Multidisciplinary optimization of a transport aircraft wing using particle swarm Optimization", *In 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization 2002*, Atlanta, GA.

[11] P.C. Fourie and A.A. Groenwold, "The particle swarm algorithm in topology optimization", *In Proceedings of the Fourth World Congress of Structural and Multidisciplinary Optimization 2001*, Dalian, China.

[12] Shi, Y., Eberhart, R., "Parameter Selection in Particle Swarm Optimization", Proceedings of the 1998 Annual Conference on Evolutionary Computation, March 1998.

[13] Shi, Y., Eberhart, R., "A Modified Particle Swarm Optimizer", *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, pp 69-73, Piscataway, NJ, IEEE Press May 1998.

[14] Natsuki H, Hitoshi I., "Particle Swarm Optimization with Gaussian Mutation", *Proceedings of IEEE Swarm Intelligence Symposium*, Indianapolis, 2003:72-79.

[15] Hu, X., Eberhart, R., Shi, Y., "Swarm Intelligence for Permutation Optimization: A Case Study of n-Queens Problem", *IEEE Swarm Intelligence Symposium 2003, Indianapolis, IN, USA.*

[16] Venter, G., Sobieszczanski-Sobieski, J., "Particle Swarm Optimization", *AIAA Journal*, Vol.41, No.8, 2003, pp 1583-1589.

[17] Hu, X., Eberhart, R., "Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization", 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, USA.

[18] Schutte, J., Reinbolt, J., Fregly, B., Haftka, R., George, A., "Parallel Global Optimization with the Particle Swarm Algorithm", *Int. J. Numer. Meth. Engng*, 2003.

[19] Koh, B, George A. D., Haftka, R. T., Fregly, B., "Parallel Asynchronous Particle Swarm Optimization", International Journal For Numerical Methods in Engineering", *International Journal of Numerical Methods in Engineering*, 67:578-595, 2006, Published online 31 January 2006 in Wiley InterScience, DOI: 10.1002/nme.1646

[20] Hu, X., Eberhart, R., Shi, Y., "Particle Swarm with Extended Memory for Multiobjective Optimization", *Proceedings of 2003 IEEE Swarm Intelligence Symposium*, pp 193-197, Indianapolis, IN, USA, April 2003, IEEE Service Center.

[21] Tayal, M., Wang, B., "Particle Swarm Optimization for Mixed Discrete, Integer and Continuous Variables", 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, Aug 30-1, 2004.

[22] Walter, B., Sannier, A., Reiners, D., Oliver, J., "UAV Swarm Control: Calculating Digital Pheromone Fields with the GPU", *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*,Volume 2005 (Conference Theme: One Team. One Fight. One Training Future).

[23] Gaudiano, P, Shargel, B., Bonabeau, E., Clough, B., "Swarm Intelligence: a New C2 Paradigm with an Application to Control of Swarms of UAVs", In Proceedings of the 8th International Command and Control Research and Technology Symposium, 2003.

[24] Colorni, A., Dorigo, M., Maniezzo, V., "Distributed Optimization by Ant Colonies", *In Proc. Europ. Conf. Artificial Life,* Editors: F. Varela and P. Bourgine, Elsevier, Amsterdam, 1991.

[25] Dorigo, M., Maniezzo, Colorni, A., "Ant System: Optimization by a Colony of Cooperating Agents", *In IEEE Trans. Systems, Man and Cybernetics*, Part B, Vol. 26, Issue 1, pp 29-41, 1996.

[26] White, T., Pagurek, B., "Towards Multi-Swarm Problem Solving in Networks", *icmas*, p. 333, Third International Conference on Multi Agent Systems (ICMAS'98), 1998

[27] Parunak, H., Purcell M., O'Conell, R., "Digital Pheromones for Autonomous Coordination of Swarming UAV's". *In Proceedings of First AIAA Unmanned Aerospace Vehicles, Systems, Technologies, and Operations Conference*, Norfolk, VA, AIAA, 2002

[28] Kavlie, D., Moe, J., "Automated Design of Frame Structures", ASCE Journal of Struct. Div., vol. 97, no. ST1, pp 33-62, January 1971

[29] Cassis, J. H., Schmit, L. A., "On Implementation of the Extended Interior Penalty Function", International Journal for Numerical Methods in Engineering, vol. 10, no. 1, pp. 3-23, 1976.

[30] Haftka, R. T., Starnes, Jr., J. H., "Applications of a Quadratic Extended Interior Penalty Function for Structural Optimization", AIAA Journal, vol. 14, no. 6, pp. 718-724, June 1976.

[31] Prasad, B., "A Class of Generalized Variable Penalty Methods for Nonlinear Programming", Journal of Optimization Theory and Applications, vol. 35, no. 2, pp. 159-182, October 1981.

[32] Gill, P., Murray W., Wright, M., "Practical Optimization", Academic Press, ISBN: 0-12-283950-1, 1981

[33] Sedlaczek, K., Eberhard, P., "Using Augmented Lagrangian Particle Swarm Optimization for Constrained Problems in Engineering", Structural Multidisciplinary Optimization Journal, Vol. 32, pp. 277-286, 2006

[34] Rao, S.S., “Engineering Optimization – Theory and Practice”, Wiley-Interscience, 3rd Edition, ISBN: 0-471-550-345, 1996.

[35] Hu, X., Eberhart, R., Shi, Y., “Engineering Optimization with Particle Swarm”, IEEE Swarm Intelligence Symposium, pp. 80-87, 2003