10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
30 August - 1 September 2004, Albany, New York

AIAA 2004-4580

Discrete, Continuous, and Constrained Optimization Using Collectives

Stefan R. Bieniawski* and Ilan M. Kroo†
Stanford University, Stanford, CA 94305

David H. Wolpert‡
NASA Ames Research Center, Moffett Field, CA 94035

Aerospace systems continue to grow in complexity while demanding optimal performance. This requires the systems to be both designed and controlled optimally. Aerospace systems are also typically comprised of many interacting components, some of which may have competing requirements. The optimization approaches used for aerospace systems usually require centralized coordination and synchronous updates. In addition, while the approaches treat the large numbers of variables, they may not take advantage of the fact that the coupling may only be between a relatively small number of the variables. Distributed optimization algorithms, such as the approach based upon collectives presented in this paper, attempt to exploit this aspect. A collective is defined as a multi-agent system where each agent is self-interested and capable of learning. Furthermore, a collective has a specified system objective which rates the performance of the joint actions of the agents. Although collectives have been used for a number of distributed optimization problems in computer science, recent developments based upon Probability Collectives (PC) theory enhance their applicability to discrete, continuous, mixed, and constrained optimization problems. This paper will present the theoretical underpinnings of the approach for these various problem domains. Several example problems are used to illustrate the technique and to provide insight into its behavior. The examples include discrete, constrained, and continuous problems. In particular a constrained discrete structural optimization and a continuous trajectory optimization illustrate the breadth of the collectives approach.

Introduction

COMPLEX systems consist of many interacting components which may have elements that compete with one another. Aerospace systems have always been among the most complex of systems and they continue to grow in complexity. Throughout this growth they have still been required to deliver optimal performance which is often rated by a specified system objective. These aspects, a large system of interacting components and a specified system objective, allow aerospace systems to be viewed as a collective. Typically for aerospace systems centralized optimization approaches have been applied, although some work has also been performed on distributed architectures. In the latter, the optimization is usually only divided into a small number of coordinated sub-level optimizations.

An alternate approach pursued in the current work is to distribute the optimization among agents that represent the variables in the system. Formulating the problem as a distributed optimization allows for the application of techniques from machine learning, statistics, multi-agent systems, and game theory. The current work leverages these fields by applying Collective Intelligence (COIN) to several illustrative problems. COIN is a framework for designing a "collective", defined as a system of adaptive computational agents with a system-level performance criteria. COIN techniques have been applied to a variety of

*Doctoral Candidate, AIAA Member.
†Professor, AIAA Fellow.
‡Senior Research Scientist.

distributed optimization problems including network routing, computing resource allocation, and data collection by autonomous rovers.[1,2,6]

The COIN solution process consists of the agents selecting actions (a value from the variable space) and receiving rewards based upon the system objective. These rewards are then used by the agents to determine their next choice of action. The process reaches equilibrium when the agents can no longer improve their rewards by changing actions. Probability Collectives (PC) theory formalizes and substantially extends the COIN framework.[7,11] In particular PC theory handles constraints more explicitly, a necessity for the problems considered in aerospace systems. The core insight of PC theory is to concentrate on how the agents update the probability distributions across their possible actions rather than specifically on the joint action generated by sampling those distributions. PC theory has been compared with results obtained with traditional COIN approaches[9] and has also been demonstrated on new problem domains.[10] The collectives approach is also in contrast to traditional aerospace optimization methods which also concentrate on a specific choice for the design variables and on how to update that choice. Even stochastic approaches such as Genetic Algorithms and Particle Swarm Optimization still operate on the design variables rather than their probability distributions. Since the collectives approach operates directly on probability distributions, it offers a direct approach for incorporating uncertainty, which is also represented through probabilities.

One way to view PC theory is as an extension of conventional game theory. In any game, the agents are independent, with each agent i choosing its move x_i at any instant by sampling its probability distribution (mixed strategy) at that instant, $q_i(x_i)$. Accordingly, the distribution of the joint-moves is a product distribution, $P(x) = \prod_i q_i(x_i)$. In this representation, all coupling between the agents occurs indirectly; it is the separate distributions of the agents $\{q_i\}$ that are coupled, while the actual moves of the agents are independent. Bounded rational agents balance their choice of best move with the need to explore other possible moves. Information theory shows that the equilibrium of a game played by bounded rational agents is the optimizer of a Lagrangian of the probability distribution of the agents' joint-moves. Since the joint probability distribution is still a product, the optimization of the Lagrangian can be done in a completely distributed manner.

When constraints are included, the bounded rational equilibrium optimizes the expected value of the system objective subject to those constraints. Updating the Lagrange parameters weighting the constraints focuses the agents more and more on the optimal joint pure strategy. This approach provides a broadly applicable way to cast any constrained optimization problem as the equilibrating process of a multi-agent system, together with an efficient method for that process.

The next section reviews the theory behind Probability Collectives. Included is a discussion of the game- and information-theoretic motivation of the theory and its application to distributed constrained optimization. This is followed by the details of the resulting optimization algorithm and its demonstration on several example problems.

Probability Collectives Theory

Bounded Rational Game Theory

In noncooperative game theory one has a set of N players. Each player i has its own set of allowed pure strategies. A mixed strategy is a distribution $q_i(x_i)$ over player i's possible pure strategies.[15]

Each player i also has a private utility function g_i that maps the pure strategies adopted by all N of the players into the real numbers. Given mixed strategies of all the players, the expected utility of player i is:

$$E(g_i) = \int dx \prod_j q_j(x_j) g_i(x)$$

In a Nash equilibrium, every player adopts the mixed strategy that maximizes its expected utility, given the mixed strategies of the other players. Nash equilibria require the assumption of full rationality, that is, every player i can calculate the strategies of the other players and its own associated optimal distribution.

In the absence of full rationality, the equilibrium is determined based on the information available to the players. Shannon realized that there is a unique real-valued quantification of the amount of syntactic information in a distribution $P(y)$. This amount of information is the negative of the Shannon entropy of that distribution:

$$S(P) = -\int dy \, P(y) \ln[P(y)]$$

Hence, the distribution with minimal information is the one that does not distinguish at all between the various y, i.e., the uniform distribution. Conversely, the most informative distribution is the one that specifies a single possible y. Given some incomplete prior knowledge about a distribution $P(y)$, this says that the estimate $P(y)$ should contain the minimal amount of extra information beyond that already contained in the prior knowledge about $P(y)$. This approach is called the maximum entropy (maxent) principle and it has proven useful in domains ranging from signal processing to supervised learning.[16]

Now consider an external observer of a game attempting to determine the equilibrium, that is the joint strategy that will be followed by real-world players of the game. Assume that the observer is provided with a set of expected utilities for the players. The best estimate of the joint distribution q that generated those expected utility values, by the maxent principle, is the distribution with maximal entropy, subject to those expectation values.

To formalize this approach, assume a finite number of players and of possible strategies for each player. Also, to agree with convention, it is necessary to flip the sign of each g_i so that the associated player i wants to minimize that function rather than maximize it.

For prior knowledge consisting of the set of expected utilities of the players $\{\epsilon_i\}$, the maxent estimate of the associated q is given by the minimizer of the Lagrangian:

$$\mathcal{L}(q) \equiv \sum_i \beta_i [E_q(g_i) - \epsilon_i] - S(q) \quad (1)$$

$$= \sum_i \beta_i \Big[\int dx \prod_j q_j(x_j) g_i(x) - \epsilon_i \Big] - S(q) \quad (2)$$

where the subscript on the expectation value indicates that it is evaluated under distribution q, and the $\{\beta_i\}$ are "inverse temperatures" $\beta_i = 1/T_i$ implicitly set by the constraints on the expected utilities.

The mixed strategies minimizing the Lagrangian are related to each other via

$$q_i(x_i) \propto e^{-E_{q_{(i)}}[G|x_i]} \quad (3)$$

where the overall proportionality constant for each i is set by normalization, and

$$G(x) \equiv \sum_i \beta_i g_i(x)$$

The subscript $q_{(i)}$ on the expectation value indicates that it is evaluated according to the distribution $\prod_{j \neq i} q_j$. The expectation is conditioned on player i making move x_i. In Eq. (3) the probability of player i choosing pure strategy x_i depends on the effect of that choice on the utilities of the other players. This reflects the fact that the prior knowledge concerns all the players equally.

Focusing on the behavior of player i, consider the case of maximal prior knowledge. Here the actual joint-strategy of the players and therefore all of their expected utilities are known. For this case, trivially, the maxent principle says the "estimate" q is that joint-strategy (it being the q with maximal entropy that is consistent with the prior knowledge). The same conclusion holds if the prior knowledge also includes the expected utility of player i.

Removing player i's strategy from this maximal prior knowledge leaves the mixed strategies of all players other than i, together with player i's expected utility. Now the prior knowledge of the other players' mixed strategies can be directly incorporated into a maxent Lagrangian for each player,

$$\begin{aligned}\mathcal{L}_i(q_i) &\equiv \beta_i[\epsilon_i - E(g_i)] - S_i(q_i) \\ &= \beta_i[\epsilon_i - \int dx \prod_j q_j(x_j) g_i(x)] - S_i(q_i)\end{aligned}$$

The solution is a set of coupled Boltzmann distributions:

$$q_i(x_i) \propto e^{-\beta_i E_{q_{(i)}}[g_i|x_i]}. \quad (4)$$

Following Nash, Brouwer's fixed point theorem can be used to establish that for any non-negative values $\{\beta\}$, there must exist at least one product distribution given by the product of these Boltzmann distributions (one term in the product for each i).

The first term in $\mathcal{L}_i$ is minimized by a perfectly rational player. The second term is minimized by a perfectly *irrational* player, i.e., by a perfectly uniform mixed strategy q_i. So β_i in the maxent Lagrangian explicitly specifies the balance between the rational and irrational behavior of the player. In the limit, $\beta \to \infty$, the set of q that simultaneously minimize the Lagrangians is the same as the set of delta functions about the Nash equilibria of the game. The same is true for Eq. (3). In fact, Eq. (3) is just a special case of Eq. (4), where all player's share the same private utility, G. Such games are known as team games. This relationship reflects the fact that for this case, the difference between the maxent Lagrangian and the one in Eq. (2) is independent of q_i. Due to this relationship, the guarantee of the existence of a solution to the set of maxent Lagrangians implies the existence of a solution of the form Eq. (3).

Optimization Approach

Given that the agents in a multi-agent system are bounded rational, if they play a team game with world utility G, their equilibrium will be the optimizer of G. Furthermore, if constraints are included, the equilibrium will be the optimizer of G subject to the constraints. The equilibrium can be found by minimizing the Lagrangian in Eq. (2) where the prior information set is empty, e.g. for all i, $\epsilon_i = \{\emptyset\}$.

Specifically for the unconstrained optimization problem,

$$\min_{\vec{x}} G(\vec{x})$$

assume each agent sets one component of $\vec{x}$ as that agent's action. The Lagrangian $\mathcal{L}_i(q_i)$ for each agent as a function of the probability distribution across its actions is,

$$\begin{aligned}\mathcal{L}_i(q_i) &= E[G(x_i, x_{(i)})] \;-\; T\,S(q_i) \\ &= \sum_{x_i} q_i(x_i) E[G(x_i, x_{(i)})|x_i] \;-\; T\,S(q_i)\end{aligned}$$

where G is the world utility (system objective) which depends upon the action of agent i, x_i, and the actions of the other agents, $x_{(i)}$. The expectation $E[G(x_i, x_{(i)})|x_i]$ is evaluated according to the distributions of the agents other than i:

$$P(x_{(i)}) = \prod_{j \neq i} q_j(x_j)$$

The entropy S is given by:

$$S(q_i) = -\sum_{x_j} q_i(x_j)\, ln\, q_i(x_j)$$

Each agent then addresses the following local optimization problem,

$$\min_{q_i} \mathcal{L}_i(q_i)$$

$$\text{s.t.} \quad \sum_{x_i} q_i(x_i) = 1, \quad q_i(x_i) \geq 0, \forall x_i$$

The Lagrangian is composed of two terms weighted by the temperature T: the expected reward across i's actions, and the entropy associated with the probability distribution across i's actions. During the minimization of the Lagrangian, the temperature provides the means to trade-off exploitation of good actions (low temperature) with exploration of other possible actions (high temperature).

The minimization of the Lagrangian is amenable to solution using gradient descent or Newton updating since both the gradient and the Hessian are obtained in closed form. Using Newton updating and enforcing the constraint on total probability, the following update rule is obtained:[8]

$$\begin{aligned}q_i(x_i) \rightarrow q_i(x_i) - \alpha q_i(x_i) \times \\ \left\{ \frac{E[G|x_i] - E[G]}{T} + S(q_i) + \ln q_i(x_i) \right\}\end{aligned} \quad (5)$$

where α plays the role of a step size. The step size is required since the expectations result from the current probability distributions of all the agents. The update rule ensures that the total probability sums to unity but does not prevent negative probabilities. To ensure this, all negative components are set to a small positive value, typically 1×10^{-6}, and then the probability distribution is re-normalized.

Extension to Constrained Problems

Constraints are included by augmenting the world utility with Lagrange multipliers, λ_j, and the constraint functions, $c_j(\vec{x})$,

$$G(\vec{x}) \rightarrow G(\vec{x}) + \sum_j \lambda_j c_j(\vec{x})$$

where the $c_j(\vec{x})$ are non-negative. The update rule for the Lagrange multipliers is found by taking the derivative of the augmented Lagrangian with respect to each Lagrange multiplier, giving:

$$\lambda_j \rightarrow \lambda_j + \eta E[c_j(\vec{x})] \quad (6)$$

where η is a separate step size.

Role of Private Utilities

Performing the update according to Eq. 5 involves a separate conditional expected utility for each agent. These are estimated either exactly if a closed form expression is available or with Monte-Carlo sampling if no simple closed form exists. In Monte Carlo sampling the agents repeatedly and jointly IID (identically and independently distributed) sample their probability distributions to generate joint moves, and the associated utility values are recorded. Since accurate estimates usually require extensive sampling, the G occurring in each agent i's update rule can be replaced with a private utility g_i chosen to ensure that the Monte Carlo estimation of $E(g_i|x_i)$ has both low bias (with respect to estimating $E(G|x_i)$ and low variance.[12]

Intuitively bias represents the alignment between the private utility and world utility. With zero bias, updates which reduce the private utility are guaranteed to also reduce the world utility. It is also desirable for an agent to distinguish its contribution from that of the other agents: variance measures this sensitivity. With low variance, the agents can perform the individual optimizations accurately without a large number of Monte-Carlo samples.

Two private utilities are typically used with the solution method, Team Game (TG) and Wonderful Life Utility (WLU). Both are presented here for completeness, but only the Team Game utility is used during the current work. These utilities are defined as:

$$g_{TG_i}(x_i, x_{(i)}) = G(x_i, x_{(i)})$$

$$g_{WLU_i}(x_i, x_{(i)}) = G(x_i, x_{(i)}) - G(CL_i, x_{(i)})$$

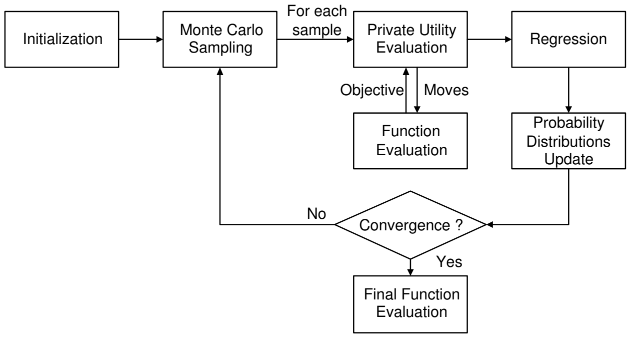

Fig. 1 Algorithm Flow Chart.

For the team game, the local utility is simply the world utility. For WLU, the local utility is the world utility minus the world utility with the agent action "clamped" by the value CL_i. The choice of clamping value can strongly affect the performance,[6] although clamping to the lowest probability action can be shown to be minimum variance.[7] Both of these utilities have zero bias. However, due to the subtracted term, WLU has much lower variance than TG.

For problems with known structure, other private utilities will be unbiased and result in low variance. The second continuous problem in this work will use such a private utility.

Solution Algorithm

The basic algorithm developed from PC theory and used to solve the example problems is illustrated in Figure 1. The algorithms proceeds as follows:

1. Initialize.
   - **(a)** Set the parameters $\{T, \alpha, \eta, \gamma\}$. Set the convergence criteria value δ.
   - **(b)** Select the number of Monte-Carlo samples.
   - **(c)** Assign the starting probabilities for each variable, typically uniform over its possible values.
2. Optimize the Lagrangian.
   - **(a)** Increment the iteration number, k.
   - **(b)** For each of the m Monte-Carlo samples,
     - Jointly IID sample the system.
     - Evaluate the objective function.
     - For each agent (variable) compute the private utility.[1]
   - **(b)** Compute the expected utility for each variable for each of its possible moves.

$$E(g_i|x_i = j) = \frac{N_{ij}^{(k)}}{D_{ij}^{(k)}} =$$

$$\frac{\sum_m g_i(x_i = j, x_{(i)}) 1(x_i = j) + \gamma N_{ij}^{(k-1)}}{\sum_m 1(x_i = j) + \gamma D_{ij}^{(k-1)}}$$

     where $1(x_i = j)$ equals 1 when $x_i = j$ and 0 otherwise. For the discrete problems this is simple averaging with data aging controlled by the parameter γ.
   - **(d)** Update the probability distributions according to Eq. (5). Ensure all the probabilities are non-negative.
   - **(e)** Update the Lagrange multipliers according to Eq. (6).
   - **(f)** Evaluate the convergence criteria,

$$\|\vec{\lambda}^k - \vec{\lambda}^{k-1}\| + \sum_i \|\vec{q_i}^k - \vec{q_i}^{k-1}\| \leq \delta$$

     If not satisfied, return to step 2(a), otherwise proceed to 3.
3. Final Evaluation.
   - **(a)** Determine the highest probability value for each variable.
   - **(b)** Evaluate the objective function with this set of values.

Extension to Continuous Variables

The solution algorithm can be extended to continuous domains by replacing the probability distribution across the discrete variables with a probability density across the domain of the variable. Many of the basic elements are easily extended to the continuous domain, such as the Monte-Carlo sampling, the use of private utilities, and the basic solution algorithm. Two aspects of the solution algorithm must, however, be modified. First, the probability density function must be represented in a way which maintains the favorable updating properties of the discrete case. In this work, the probability density is parameterized by values at fixed, equally spaced locations across the domain of the variables. Second, since the sampling occurs at only a scattering of points across the range of the variables, a regression is necessary. In the current work a simple regression based upon exponentially weighted averaging across samples is used. Since the regression is a fit to the private utility value across the domain of the variable it represents a one-dimensional response surface, an area for which there is a large body of research

[1]Team Game requires one function evaluation for each Monte-Carlo sample, while the generic version of the Wonderful Life utility requires as many function evaluations as there are variables. Often the structure of the objective function and constraints can be exploited in the evaluation of WLU to avoid unnecessary function calls [1,6]

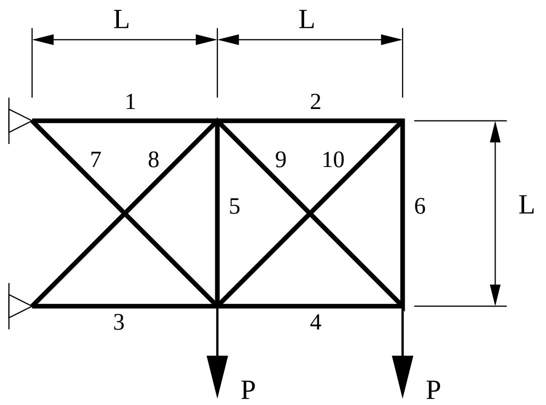

Fig. 2 10-bar Truss.

Table 1 Properties for the 10-bar truss

Material:	$E = 10^7\, psi, \rho = 0.1\, lbm/in^3$
Stress limit:	$25000\, psi$
Displacement limit:	$2\, in$
Load:	$100\, kip$
Length:	$L = 360\, in$

literature. There is significant area for improvement in the algorithm presented here if more refined techniques are applied.

Results

Discrete Constrained Optimization Problem

The performance of the solution algorithm on a discrete constrained problem is illustrated using the ten bar truss. This problem has been previously used to study a variety of optimization approaches including branch and bound[3] algorithms, genetic algorithms,[4,5] and particle swarm optimization algorithms.[17] Figure 2 shows the geometry and the element numbering while Table 1 lists the relevant properties.

Two cases were considered, the first enforcing just the stress constraints on each element,[4,5] the second enforcing both the stress and vertical displacement constraints.[3] The objective for both cases was the total weight of the structure. For the results in this paper, the weight was divided by a factor of 1000 to be of the same order of magnitude as the constraints.

Figure 3 shows the convergence history for the first case. The parameters settings used were $\alpha = 0.01$, $\gamma = 0.5$, and $\eta = 1.0$. Team Game utility was used with a starting temperature of 0.5 which was multiplied by a factor of 0.95 each iteration. Sixteen values, varying linearly from 0.1 in^2 to 10 in^2, were permitted.[4,5] The number of Monte-Carlo samples for each iteration was varied from 50 to 200 to 500. For comparison purposes both the continuous optimum and rounded-up continuous optimum value are shown. For

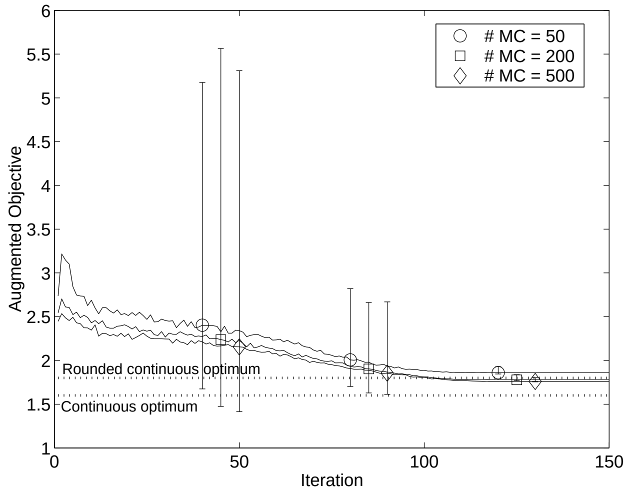

Fig. 3 Iteration history for 10-bar truss with stress constraints.

the latter the optimum areas from the continuous solution were rounded up to the next allowable value. The true integer optimum is bounded by these two values. The curves are the averages over 20 runs for the mean of the samples within each iteration. The error bars are then the averages of the minima and maxima of the samples within each iteration. Although the convergence rate is similar, the range of values explored is clearly wider for the higher Monte-Carlo samples, which leads to a lower optimum with increasing samples. The converged result is typically in the range between the continuous optimum and the rounded-up continuous optimum.

Figure 4 shows the convergence history for the second case. The same settings as for the first case were used except with a slower annealing rate. For this case the temperature was multiplied by 0.99 each iteration instead of 0.95 to allow for additional exploration. The optimizations were also only performed for 500 Monte-Carlo samples. For this case there were 81 possible values, $x_i \in \{0.1, 0.5, 1.0, ..., 40.0\}$.[3] Again the curve in Figure 4 represents the mean at each iteration averaged over 20 runs. The error bars show that the approach has converged by 500 iterations to a value a few percent above the discrete optimum found with other approaches.[3] Table 2 directly compares the median and best collectives solutions with the result presented in Reference.[3]

Continuous Optimization Problems

The solution approach was also applied to two continuous problems. The first consists of several Gaussian peaks, as shown in Figure 5. Note that variable x can not see the global minimum until variable y changes its distribution away from uniform. As a result, cooperation is required between the variables to

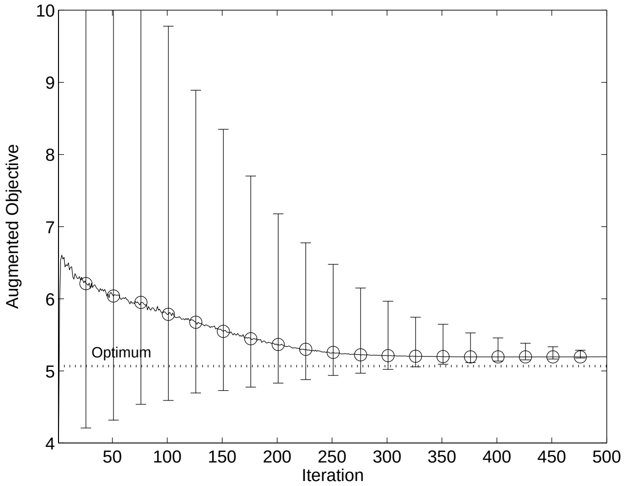

Fig. 4 Iteration history for 10-bar truss with stress and displacement constraints.

Table 2 Solution comparison for the 10-bar truss.

	Median Collectives	Best Collectives	Reference[3]
W ($lbs/1000$)	5.1926	5.1417	5.0673
Feasible	yes	yes	yes
x_1 (in^2)	30.0	31.0	30.5
x_2	0.1	0.5	1.0
x_3	23.5	24.5	24.5
x_4	15.0	14.5	14.5
x_5	0.1	0.1	0.1
x_6	0.5	0.1	0.1
x_7	7.5	8.5	8.5
x_8	21.0	21.0	21.5
x_9	22.0	20.5	20.5
x_{10}	0.1	1.0	1.5

find the optimum. The objective function is given by,

$$G(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10(x/5 - x^3 - y^5)e^{-x^2-y^2} - 1/3e^{-(x+1)^2-y^2} \quad (7)$$

Figure 6 shows the convergence history for this problem averaged over 10 runs. The results indicate that the technique quickly finds the minimum, typically within 10 iterations. Although a small test problem, the quick convergence here with only 10 Monte-Carlo samples per iteration illustrates the potential of the approach. The parameter settings used were T = 0.1, τ = 0.1, α = 0.25, and γ = 0.5. The probability densities were each represented using 200 points. The slight oscillation in the converged optimum is due to the fixed, moderate temperature value. This choice results in probability distributions which are only cen-

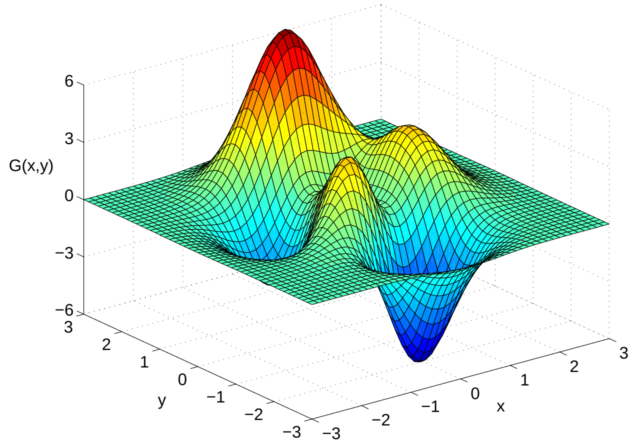

Fig. 5 Peaks function.

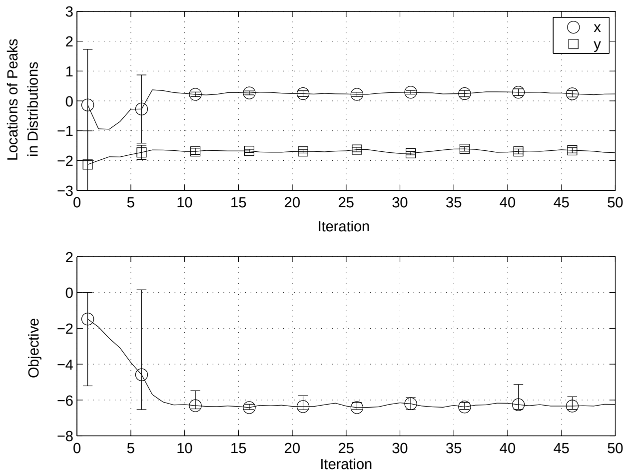

Fig. 6 Convergence history for peaks function.

tered about the optimum, but when sampled will provide values away from the optimum. This is clear in Figure 7 which shows the converged probability distributions. This figure also illustrates another advantage of the approach, the sensitivity of the objective to each of the variables. Since the converged probabilities are related to the expected utilities through Eq. 4, given the probabilities $q_i(x_i)$, the expected utility, $E(g_i|x_i)$ is obtained.

The second continuous problem considered is a classical calculus of variations problem, the Brachistochrone problem.[13] The objective is to find the minimum time trajectory between two points for an object moving only under the influence of gravity, Figure 8. Following[13] the objective function is:

$$t_{12} = \int_{(x_1,y_1)}^{(x_2,y_2)} f \, dx$$

where,

$$f = (1 + (dy/dx)^2)^{1/2} \, (2gy)^{1/2}$$

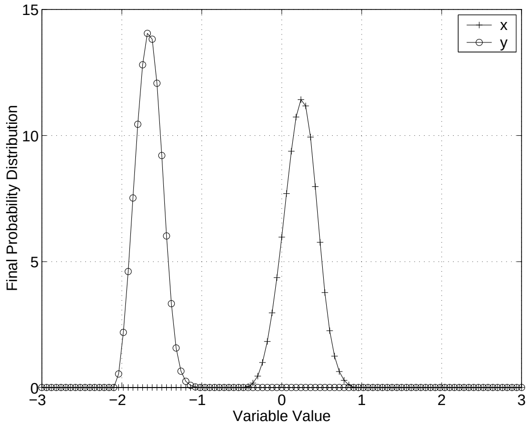

Fig. 7 Final probability distributions for peaks function.

A trapezoidal approximation is made to the integral at N points and a central finite difference is used for the derivative. This results in the following optimization problem with respect to the N vertical locations, $y_1, ..., y_N$:

$$\min_{\vec{y}} G = \frac{\Delta x}{2}\left[f_0 + 2f_1 + ... + 2f_N - 1 + 2f_N\right]$$

where, for the interior points

$$f_i = (1 + [\frac{1}{2\Delta x}(y_{i+1} - y_{i-1}]^2)^{1/2}\ (2gy_i)^{1/2} \qquad (8)$$

For the boundary points, f_0 and f_N, forward or backward approximations are used for the derivatives.

This optimization problem was solved by a commercially available gradient based optimizer[14] and by the collectives approach. Collectives are particularly applicable to objectives such as Eq. 8 due to the sparse nature of the interactions between the variables. Since contributions to the objective f_i are only functions of a single variable and its neighbors, a suitable private utility is,

$$\begin{aligned} g_i(y_{i-2}, y_{i-1}, y_i, y_{i+1}, y_{i+2}) = \frac{\Delta x}{2}\big[&2f_{i-1}(y_{i-2}, y_{i-1}, y_i) \\ &+ 2f_i(y_{i-1}, y_i, y_{i+1}) \\ &+ 2f_{i+1}(y_i, y_{i+1}, y_{i+2})\big] \end{aligned} \qquad (9)$$

Eq. 9 is used for the interior agents while similar private utilities can be obtained for the first and last agents. This private utility has no bias since it includes all the dependencies of the world utility upon agent i.

Figure 9 shows the objective function convergence history for the gradient based and collectives approaches. Relevant parameters for the collectives results are $\alpha = 0.2$, $\gamma = 0.8$, 10 Monte-Carlo samples per iteration, and $T = 0.01$. Ten optimizations were performed and the 90% confidence bars are shown. For the gradient based optimization a random starting point was used each time and all other settings used the default values. The collectives approach performs comparably, finding a minimum about 5% higher than the gradient based optimum. Since the collectives approach is a stochastic optimization technique, it searches for a distribution of good solutions. To encourage the approach to find a single solution, as with the gradient based approach, either the temperature should be lowered or it should annealed as the optimization proceeds. Additional Monte-Carlo samples also aid in finding a better optimum. Figure 10 compares the convergence rate for the baseline, increasing the number of Monte-Carlo samples to 100, and annealing the temperature by 0.99 each iteration. Note that this figure compares the performance versus number of iterations rather than number of function calls. Changing the parameters is seen to result in similar improvement in the converged result.

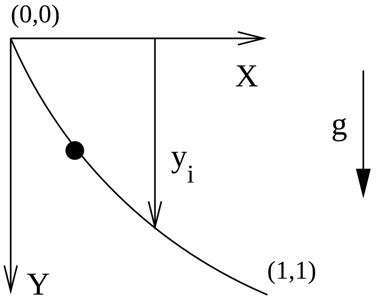

Fig. 8 The Brachistochrone problem.

Figure 11 shows the converged probability distributions for the agents. This again illustrates the non-linear sensitivity information provided by the collectives approach. Another advantage of this approach which still remains to be explored is the ability to treat stochastic boundary conditions. Since these types of boundary conditions are typically represented as probability distributions, they can be incorporated as additional agents whose probabilities are simply not updated.

Summary

This paper has illustrated the application of collectives to a range of optimization problems of interest in aerospace systems. These have included the discrete, constrained, and continuous problem domains. The theory behind the approach as well as illustrative examples are intended to promote further research into their application to aerospace systems. Given the continued growth in the complexity of aerospace systems and the pressure for optimal performance, techniques such as the one presented in this paper offer potential.

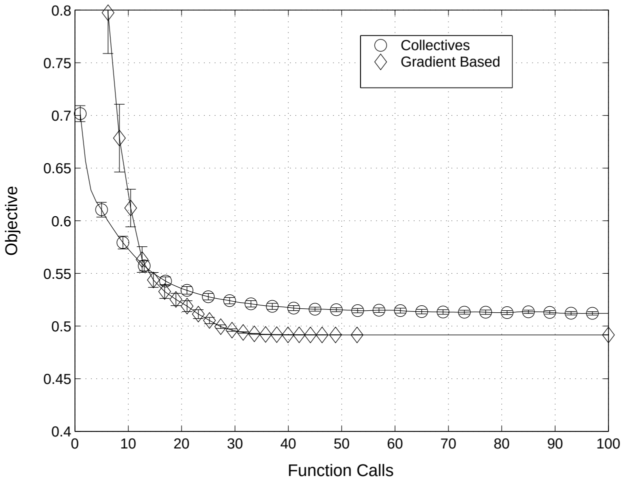

Fig. 9 Iteration history for the Brachistochrone problem.

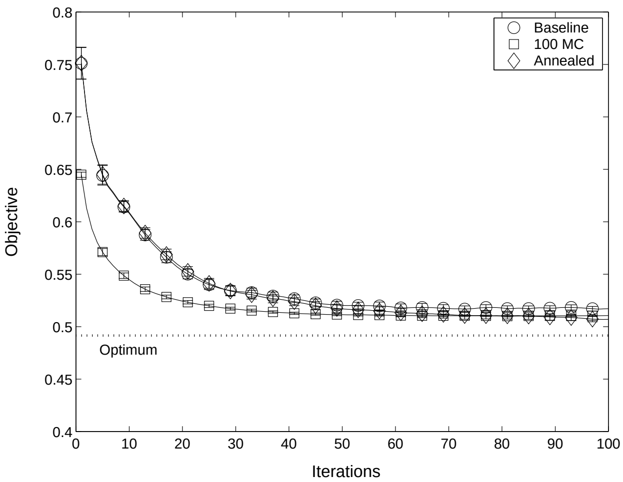

Fig. 10 Iteration history for the Brachistochrone problem with different parameter settings.

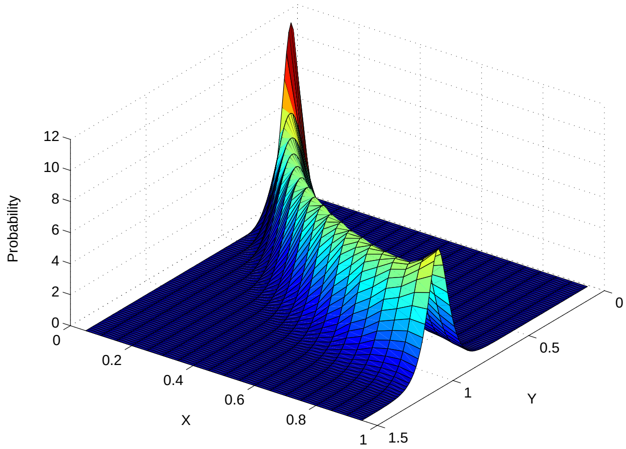

Fig. 11 Converged probability distributions for the Brachistochrone problem.

References

[1]Tumer, K., Agogino A., and Wolpert, D. H., "Learning sequences of actions in collectives of autonomous agents," In *Proceedings of the First International Joint Conference on Autonomous and Multi-Agent Systems*, Bologna, Italy, July 2002.

[2]Wolpert, D. H., Tumer, K., "Collective Intelligence, Data Routing, and Braess' Paradox," Journal of Artificial Intelligence Research, 2002.

[3]Ringertz, U., "On Methods for Discrete Structural Optimzation," Engineering Optimization, Vol. 13, pp. 47-64, 1988.

[4]Goldberg, D. E., and Samtani, M.P., "Engineering Optimization via Genetic Algorithms," *Proceedings of the Ninth Conference on Electronic Computation*, 471-482, 1986.

[5]Goldberg, D. E., **Genetic Algorithms in Search, Optimization, and Machine Learning**, Addison Welsey Longman, 1989.

[6]Wolpert, D.H., Wheeler, K., Tumer, K., "Collective Intelligence for Control of Distributed Dynamical Systems," Europhysics Letters, vol. 49 issue 6, 708-714, 2000.

[7]Wolpert, D.H., "Information theory - the bridge connecting bounded rational game theory and statistical physics", in Complex Engineering Systems, D. Braha, Ali Minai, and Y. Bar-Yam (Editors), Perseus books, in press.

[8]Wolpert, D.H., Bieniawski, S., "Distributed Control by Lagrangian Steepest Descent" submitted to 43rd IEEE Conference on Decision and Control, December 2004.

[9]Bieniawski, S., Wolpert, D.H., "Adaptive, distributed control of constrained multi-agent systems," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, 19-23 July 2004, New York, New York.

[10]Macready, W., and Wolpert, D., "Distributed Optimization," International Conference on Complex Systems, Boston, May 2004.

[11]Wolpert, D., and Lee, C.F. "Adaptive Metropolis Sampling with Product Distributions," International Conference on Complex Systems, Boston, May 2004.

[12]Duda, Richard O., Hart, Peter E. Hart, and Stork, David G., **Pattern Recognition**, Wiley, 2001.

[13]Eric W. Weisstein, "Brachistochrone Problem." From *MathWorld* – A Wolfram Web Resource. http://mathworld.wolfram.com/BrachistochroneProblem.html

[14]*MATLAB Optimization Toolbox User's Guide.* The MathWorks, 2000.

[15]Fudenberg, D., and Tirole, J., **Game Theory**, MIT Press, Cambridge, MA, 1991.

[16]Mackay, D., **Information theory, inference, and learning algorithms.** Cambridge University Press, 2003.

[17]Schutte, J., and Groenwold, A., "Optimal Sizing Design of Truss Structures Using the Particle Swarm Optimization Algorithm," AIAA 2002-5639, *9th AiAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, September 2002.