

# Fleet Assignment Using Collective Intelligence

Nicolas E. Antoine\*, Stefan R. Bieniawski†, and Ilan M. Kroo‡  
Stanford University, Stanford, CA 94305

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

Product distribution theory is a new collective intelligence-based framework for analyzing and controlling distributed systems. Its usefulness in distributed stochastic optimization is illustrated here through an airline fleet assignment problem. This problem involves the allocation of aircraft to a set of flights legs in order to meet passenger demand, while satisfying a variety of linear and non-linear constraints. Over the course of the day, the routing of each aircraft is determined in order to minimize the number of required flights for a given fleet. The associated flow continuity and aircraft count constraints have led researchers to focus on obtaining quasi-optimal solutions, especially at larger scales. In this paper, the authors propose the application of this new stochastic optimization algorithm to a non-linear objective “cold start” fleet assignment problem. Results show that the optimizer can successfully solve such highly-constrained problems (130 variables, 184 constraints).

## Introduction

**S**CHEDULE development, a crucial aspect of profitable airline management, involves many steps, including schedule design, fleet assignment, aircraft routing, and crew pairing.<sup>1</sup> In this project, we assume that schedule design has been finalized; the focus is on *fleet assignment*, that is the assignment of available aircraft to the scheduled flights, and on *aircraft routing*, the sequence of flights to be flown by each aircraft throughout the day (Figure 1). Typical fleet assignment objectives include minimizing assignment cost or maximizing the profit from each flight. In our case, the objective is to meet the passenger demand throughout the day with the lowest total landing and takeoff (LTO) costs.

Fleet assignment problems can be classified as either “warm start”, in which case an existing assignment is used as a starting point, or “cold start”, in which only the fleet size, aircraft types, and passenger demand are known.<sup>2</sup> Fleet assignment and aircraft routing problems have been solved using various optimization methods, including integer linear programming,<sup>3,4</sup> neighborhood search,<sup>5</sup> and genetic algorithms.<sup>6</sup>

An alternate approach pursued here is to distribute the optimization among agents that represent, for example, members of the fleet or the airports in the network. Formulating the problem as a distributed stochastic 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 a Collective Intelligence (COIN) technique, Product Distribution (PD) theory, to a sample fleet assignment problem. Typically in stochastic optimization approaches probability distributions are used to help search for a point in the variable space which optimizes the objective function. In contrast, in the PD approach the search is for a probability distribution across the variable space that optimizes an associated Lagrangian. Since the probability distribution is a vector in a Euclidean space, the search can be done via gradient based methods even if the variable space is categorical. Similar techniques have been successfully applied to a variety of distributed optimization problems including network rout-

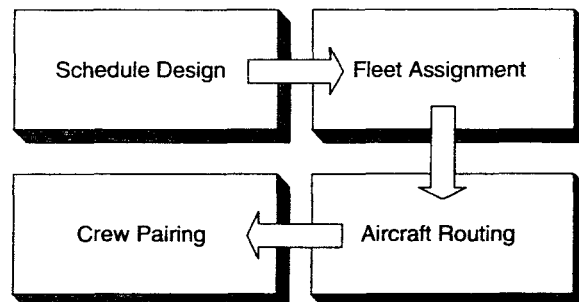


Fig. 1 Airline Schedule Development

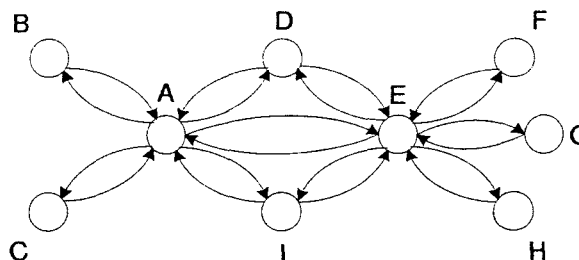


Fig. 2 The 9-airport, 20-arc problem.

ing, computing resource allocation, and data collection by autonomous rovers.<sup>7-9</sup>

The next section of the paper details the formulation of the optimization problem. This is followed by a description of the COIN and PD theory framework. Finally, results from an example fleet assignment problem are presented. These results validate the predictions of this theory, and indicate its usefulness as a general purpose technique for distributed solutions of constrained optimization problems.

## Problem Statement

The objective is to determine the aircraft routing and resident fleet size at each airport that minimizes the landing and takeoff fees levied by airports while meeting demand. The 9-airport, 20-flight directed arc sample problem (Figure 2) is used to demonstrate the performance of the approach. The passenger demand on each arc is given as a function of time (determined as part of the schedule design). The day is split into six 4-hour segments. It is assumed that each

\*Doctoral Candidate, AIAA Student Member.

†Doctoral Candidate, AIAA Member.

‡Professor, AIAA Fellow.

§Senior Research Scientist.

Pax Capacity $w$	Cost Factor $F(w)$
100	1.0
200	1.5
300	2.0

Table 1 LTO cost factor as a function of aircraft passenger capacity.

arc can be flown and the aircraft turned around in one time segment. The optimization problem is as follows:

MINIMIZE:	Total LTO Fees
VARIABLES:	Number of aircraft on each arc Resident fleet at each airport Airplane passenger capacity
CONSTRAINTS:	Passenger demand Assignment continuity Resident fleet conservation Total fleet size

The three types of variables are:  $u_{i,j}$ , the number of aircraft assigned to flight arc  $i$  at time segment  $j$ ,  $v_k$ , the number of resident aircraft at airport  $k$ , and  $w$ , the passenger capacity of the airplane. The resident fleet is the number of airplanes at each airport at the start and end of the day, which must be the same to repeat the schedule the next day. The allowable ranges for the variables are:

$$0 \leq u_{i,j} \leq 12$$

$$0 \leq v_k \leq 30$$

$$w = 100, 200, 300$$

The total daily LTO cost is a function of  $F(w)$  (see Table 1) and the number of segments flown. The non-linear objective function can be written as:

$$\min_{u_{i,j}, v_k, w} \left( G = F(w) \sum_{i,j} u_{i,j} \right)$$

There are 20 arcs and 6 time segments in this problem, which, with 9 airports and 1 aircraft type, results in a total of 130 variables. Constraints are required to ensure that passenger demand  $D_{i,j}$  is met in full by capacity  $C_{i,j}$  for each arc, at each time segment. There are 20 arcs and 6 time segments, for a total of 120 passenger demand constraints. For these non-linear constraints to be satisfied:

$$-C_{i,j} + D_{i,j} \leq 0$$

with:

$$C_{i,j} = w \cdot u_{i,j}$$

While the framework supports multiple aircraft models, in this example problem the fleet is composed of a single aircraft type, for which the passenger capacity is a variable. Assignment continuity ensures that an airplane can only be assigned to an arc if an airplane is available at the originating airport. With 9 airports and 6 time segments, 54 continuity constraints are included. Defining  $S_{k,j}$  as the state of the fleet at airport  $k$  at the beginning of time increment  $j$ , we require:

$$-S_{k,j} \leq 0$$

where:

$$S_{k,j} = S_{k,j-1} + \sum_i M_{k,i} \cdot u_{i,j} + \sum_i N_{k,i} \cdot u_{i,j-1}$$

The  $M$  matrix is used to tally outbound aircraft for each airport during a time segment. Likewise,  $N$  is used to determine the inbound aircraft to be added to an airport pool.

For example, for our 9-city, 20-arc case:

	AB	AC	...	BA	...	IE
A	-1	-1	...	0	...	0
B	0	0	...	-1	...	0
C	0	0	...	0	...	0
...	...	...	...	...	...	...
I	0	0	...	0	...	-1

$$M =$$

	AB	AC	...	BA	...	IE
A	0	0	...	1	...	0
B	1	0	...	0	...	0
C	0	0	...	0	...	0
...	...	...	...	...	...	...
I	0	0	...	0	...	0

$$N =$$

The resident fleet size  $SI_k$  at each airport  $k$  must equal the number of airplanes  $SF_k$  at the end of the day so the schedule can be restarted the following day. In equation form, we require:

$$-SI_k + SF_k \leq 0$$

with:

$$SI_k = v_k$$

$$SF_k = \sum_i N_{k,i} \cdot u_{i,j_{final}}$$

The airports in this sample problem contribute 9 resident fleet constraints. Finally, the total fleet size  $F$  is enforced using:

$$\sum_k SI_k - F \leq 0$$

This results in a total of 184 constraints.

## Collective Intelligence and Product Distribution Theory

Collective Intelligence (COIN) is a framework for designing a collective, defined as a group of agents with a specified world utility or system-level objective. In the case of the fleet assignment problem, the agents match the two types of variables: the number of airplanes assigned to each route for each time segment, and the size of the resident fleet at each airport. The world utility for this problem is the objective described above: total LTO fees.

The COIN solution process consists of the agents selecting actions (a value from the variable space) and receiving rewards based upon their private utility functions. 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. Product Distribution (PD) theory formalizes and substantially extends the COIN framework.<sup>10-12</sup> In particular PD theory handles constraints, a necessity for problems such as fleet assignment. The core insight of PD 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.

PD theory can be viewed as the information-theoretic extension of conventional full-rationality game theory to the case of bounded rational agents. 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. In any game, bounded rational or otherwise, 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. As a result 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 world utility 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 equilibrating process.

The next section reviews the game-theoretic motivation of PD theory. This is followed by the details of the resulting distributed constrained optimization algorithm used to solve the fleet assignment problem.

### 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.

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.<sup>13</sup>

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, we 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 as-

sociated  $q$  is given by the minimizer of the Lagrangian:

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

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

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 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 our prior knowledge). The same conclusion holds if our 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  $\beta_i$  in the maxent Lagrangian explicitly specifies the balance between the rational and irrational behavior of the player. In the limit,  $\beta \rightarrow \infty$ , 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,

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

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,

$$\begin{aligned} \min_{q_i} \mathcal{L}_i(q_i) \\ \text{s.t. } \sum_{x_i} q_i(x_i) = 1, \quad q_i(x_i) \geq 0, \forall x_i \end{aligned}$$

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:

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

where  $\alpha$  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.

Constraints are included by augmenting the world utility with Lagrange multipliers,  $\lambda_j$ , 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  $\eta$  is a separate step size.

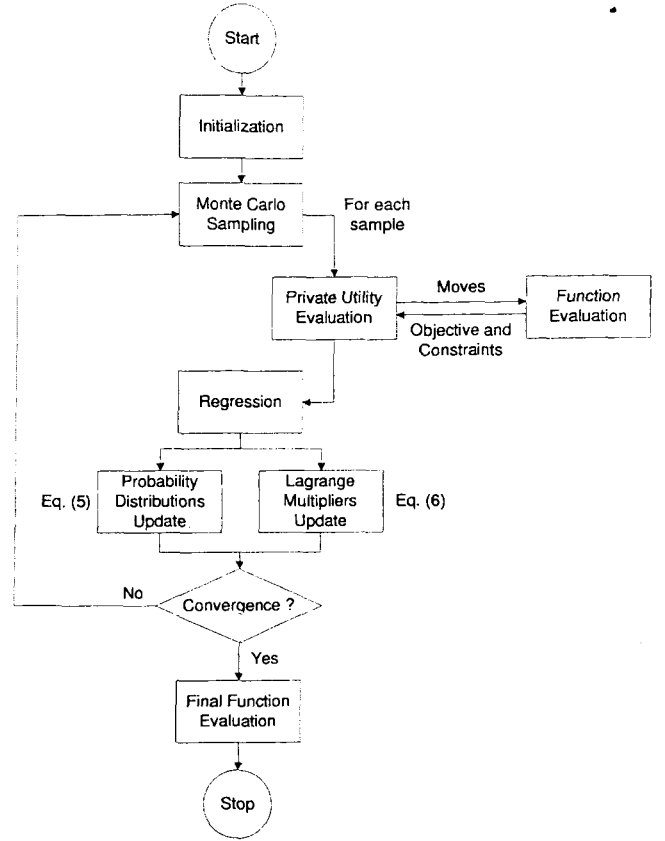


Fig. 3 Algorithm Flow Chart.

### Role of Private Utilities

Performing the update 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.<sup>14</sup>

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 were selected for use in the fleet assignment problem, Team Game (TG) and Wonderful Life Utility (WLU). These are defined as:

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

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

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  $CL_i$ . Here the clamping value fixes the agent action to its lowest probability action. Both of these utilities have zero bias. However, due to the subtracted term, WLU has much lower variance than TG.

Arc	Time Segment					
	1	2	3	4	5	6
AB	100	200	200	200	100	100
AC	200	100	100	100	200	200
AD	200	200	200	200	200	200
AE	400	600	800	800	600	400
AI	100	100	200	100	100	100
BA	100	100	200	200	200	100
CA	200	200	100	100	100	200
DA	200	200	200	200	200	200
DE	200	200	200	200	200	200
EA	400	600	800	800	600	400
ED	200	200	200	200	200	200
EF	100	0	100	100	100	100
EG	100	100	100	100	100	100
EH	200	200	200	200	200	200
EI	100	100	100	100	100	100
FE	100	0	100	100	100	100
GE	100	100	100	100	100	100
HE	200	200	200	200	200	200
IA	100	100	100	100	100	100
IE	100	100	100	100	100	100

Table 2 Passenger demand for each arc as a function of time.

### Detailed Algorithm

The algorithm used to solve the fleet assignment and aircraft routing problem presented here is illustrated in Figure 3. The initialization step involves setting each agent's probabilities to uniform over its possible moves. The Lagrange multipliers for all the constraints are initialized to zero. A loop is then repeated until convergence. Within the loop the Monte-Carlo sampling is performed, after which the private utilities for each agent are computed. The number of function evaluations required depends on the private utility. 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.<sup>7,9</sup> In order to demonstrate the performance of the algorithm without any such preprocessing, the present work makes no effort to exploit the structure of the fleet assignment formulation described above. Due to the discrete nature of the agent moves, the regression step involves simply averaging the private utility received for each agent's move over the Monte Carlo samples. Data aging is used within the regression to preserve information from previous iterations. The previous private utility estimates are weighted by a factor  $\gamma$  compared with the new samples during the regression. The probabilities and Lagrange multipliers are then updated according to Eqs. (5) and (6), respectively. Eq. (5) automatically enforces the constraint on total probability but does not prevent negative probabilities. To prevent negative probabilities the probability update is modified by setting all components that would be negative to a small positive value, typically  $1 \times 10^{-6}$ , and then re-normalizing. Finally, the convergence criterion is checked, in this case a combination of the norms of the probability and Lagrange multiplier updates:

$$U_p^{0.5} + U_\lambda^{0.5} \leq 1 \times 10^{-4}$$

with:

$$U_p = \sum_i \sum_{x_i} (q_i(x_i) - q_i(x_i)_{prev})^2$$

$$U_\lambda = \sum_j (\lambda_j - \lambda_{j,prev})^2$$

If the criterion is not met, the sampling and update process is repeated.

Arc	Time Segment					
	1	2	3	4	5	6
AB	1	2	2	2	1	1
AC	2	1	1	1	2	2
AD	2	2	2	2	2	2
AE	4	6	8	8	6	4
AI	1	1	2	1	1	1
BA	1	1	2	2	2	1
CA	2	2	1	1	1	2
DA	2	2	2	2	2	2
DE	2	2	2	2	2	2
EA	4	6	8	8	6	4
ED	2	2	2	2	2	2
EF	1	0	1	1	1	1
EG	1	1	1	1	1	1
EH	2	2	2	2	2	2
EI	1	1	1	1	1	1
FE	1	0	1	1	1	1
GE	1	1	1	1	1	1
HE	2	2	2	2	2	2
IA	1	1	1	1	1	2
IE	1	1	1	1	1	1

Table 3 Number of flights assigned to each arc at each time segment.

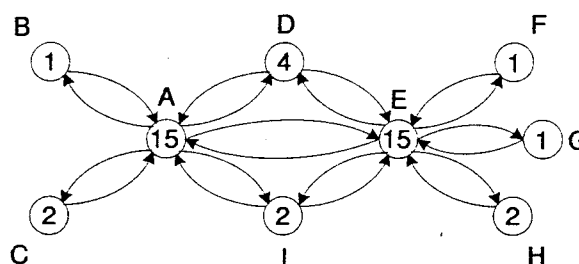


Fig. 4 The resident fleet at each airport required to optimally solve the problem.

## Results

### Linearized Constraints and Objective Function

For assessment purposes, both the PD framework and AMPL-CPLEX<sup>15,16</sup> were applied to a linearized version of the 9-city, 20-arc fleet assignment problem (CPLEX does not support non-linear constraints).

In this case, the passenger capacity  $w$  was fixed to 100. The linear objective becomes:

$$\min_{u_{i,j}, v_k} \left( G = 100 \sum_{i,j} u_{i,j} \right)$$

with the total fleet passenger capacity:

$$C_{i,j} = 100 \cdot u_{i,j}$$

The problem features a time-dependant, asymmetric demand structure as shown in Table 2.

To enhance the convergence speed, the objective was squared, effectively dramatizing the topology of the problem:

$$\min_{u_{i,j}, v_k} \left( G = (100 \sum_{i,j} u_{i,j})^2 \right)$$

Both optimization tools reached global minimum with a fleet size of 43 aircraft: 228 flights are required, yielding an objective of 51,984. The number of flights assigned to each route is shown in Table 3, and the resident fleet size at each airport is illustrated in Figure 4. In order to capture the stochastic nature of the approach, the optimization was repeated 20 times. The figures show averages and ranges for the minimum objective in each block of Monte-Carlo samples. Each iteration is an update to the probability distributions using a single block of Monte-Carlo samples.

The importance of selecting the appropriate private utility is shown in Figure 5. For each utility, the best temperature

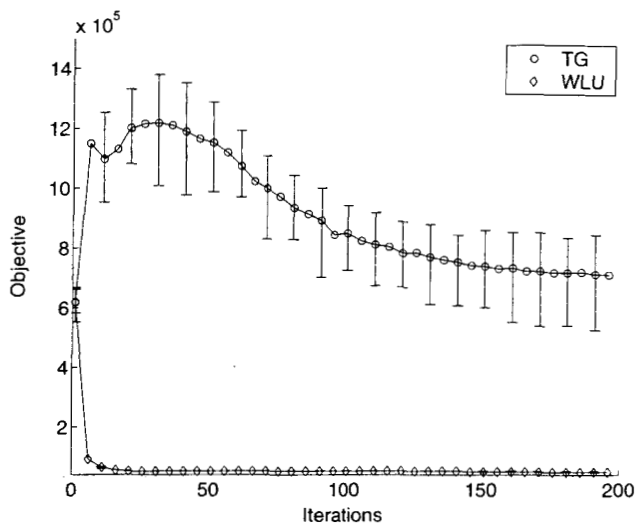


Fig. 5 Comparison of convergence with two private utilities (200 Monte Carlo samples).

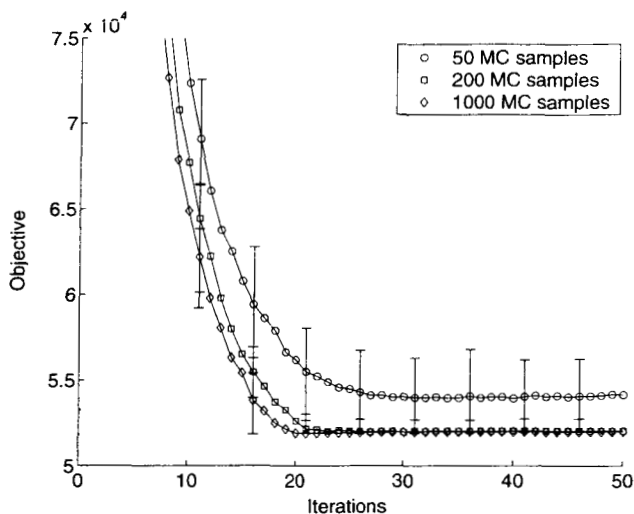


Fig. 6 Increasing sampling speeds convergence (Temperature = 10, WLU, 20 runs).

was selected, 10 for WLU and 1000 for TG. The results show that WLU performs considerably better than Team Game. This is consistent with previous applications of COIN.<sup>7,9</sup>

As illustrated in Figure 6, the number of Monte Carlo samples between updates affects the rate of convergence. In almost all cases, 50 samples were not sufficient to find the minimum objective. With 200 samples, the minimum was found in 18 of 20 cases. Increasing the number of samples to 1000 resulted in all cases converging to the minimum.

Similarly, selecting the correct temperature influences the optimization process (Figure 7). A low temperature ( $T=1$ ) did not allow enough exploration, while a high temperature ( $T=100$ ) slowed convergence. For this example, a moderate temperature ( $T=10$ ) offered the best trade-off between exploration and exploitation. In particular, the case with the lowest temperature rapidly converged to an infeasible minimum. The objective then grew as the Lagrange multipliers increased. The optimizer, at this low temperature, is unable to explore other regions of the design space.

#### Non-Linear Constraints and Objective Function

Following the above linearized study, the methodology was applied to the fully non-linear problem: the PD framework's ambivalence towards constraint and objective types makes it ideal in this case. The PD optimizer obtains a solution that is very close to optimum: it selects the correct aircraft, with 200 passenger capacity (resulting in an

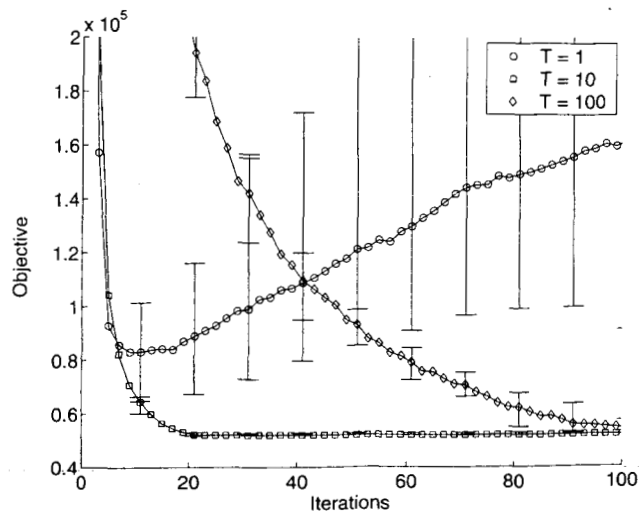


Fig. 7 Effects of temperature on convergence (200 Monte Carlo samples, WLU, 20 runs).

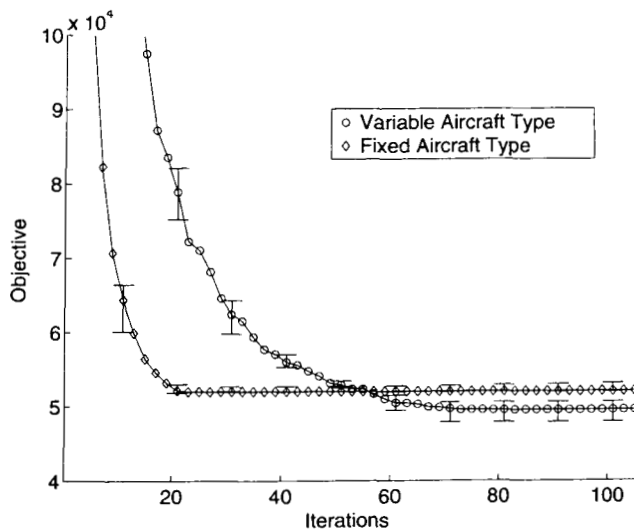


Fig. 8 Comparison of optimum for the linearized (fixed airplane capacity) and non-linear (variable airplane capacity) problems.

LTO cost factor of 1.5) and requires a total of 146 flights to meet demand. The objective is therefore 47,961. Figure 8 compares the convergence rate of the variable aircraft type problem - while the convergence takes slightly more iterations, no changes in the parameter settings were made. For comparison purposes, running AMPL-CPLEX with a fixed passenger capacity of 200 resulted in a solution with 142 flights required, for an objective of 45,369. When the PD framework is run with fixed aircraft capacity, this same result is obtained.

#### Conclusion

A collective-intelligence framework was successfully applied to a sample fleet assignment problem and yielded globally optimum solutions. With the basic framework proven to handle highly-constrained design spaces with non-linear constraints and objectives, a fleet assignment problem of more realistic size can be approached. The function evaluation was carefully formulated to allow for scalability and automation, and features such as transfer passengers, environmental considerations, and maintenance visit requirements can be implemented. Exploring other types of agents (perhaps airports or arcs) and developing problem-specific local utilities may also yield faster convergence rates and

require fewer Monte Carlo samples.

## References

- <sup>1</sup>Etschmaier, M., and Mathaisel, D., "Airline Scheduling: An Overview," *Transportation Science* 19, 1985, pp. 127-138.
- <sup>2</sup>Rushmeier, R., and Kontogiorgis, A., "Advances in the Optimization of Airline Fleet Assignment," *Transportation Science* 31, 1997, pp. 159-169.
- <sup>3</sup>Abara, J., "Applying integer linear programming to the fleet assignment problem," *Interfaces* 19, 1989.
- <sup>4</sup>Hane, C., et al., "The fleet assignment problem: Solving a large-scale integer program," *Mathematical Programming* 70, 1995, pp. 211-232.
- <sup>5</sup>Ahuja, R., et al., "A very large scale neighborhood search algorithm for the quadratic assignment problem," Submitted to *INFORMS Journal on Computing*, 2002.
- <sup>6</sup>Chung, T., and Chung J., "Airline Fleet Assignment Using Genetic Algorithms," *GECCO2002, 2002 Genetic and Evolutionary Computation Conference*, New York, New York, July 11-13, 2002, p. 255.
- <sup>7</sup>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.
- <sup>8</sup>Wolpert, D. H., Tumer, K., "Collective Intelligence, Data Routing, and Braess' Paradox," *Journal of Artificial Intelligence Research*, 2002.
- <sup>9</sup>Wolpert, D.H., Wheeler, K., Tumer, K., "Collective Intelligence for Control of Distributed Dynamical Systems," *Europhysics Letters*, vol. 49 issue 6, 708-714, 2000.
- <sup>10</sup>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.
- <sup>11</sup>Bieniawski, S., Wolpert, S., and Kroo I., "Discrete, Continuous, and Constrained Optimization Using Collectives," 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York 30 Aug - 1 Sep 2004.
- <sup>12</sup>Macready, W., Bieniawski, S., Wolpert, D., "Adaptive Multi-Agent Systems for Constrained Optimization," Submitted to *AAAI* 2004.
- <sup>13</sup>Mackay, D., "Information theory, inference, and learning algorithms," Cambridge University Press, 2003.
- <sup>14</sup>Duda, R., Hart, P., and Stork, D., *Pattern Recognition*, Wiley, 2001.
- <sup>15</sup>Fourer, R., Gay, D., and Kernighan, B., *AMPL: a Modeling Language for Mathematical Programming*, Boyd and Fraser, 1993.
- <sup>16</sup>Anon., *ILOG AMPL CPLEX System User's Guide*, ILOG, 2002.