

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

Airline fleet assignment involves the allocation of aircraft to a set of flights legs in order to meet passenger demand, while satisfying a variety of 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 an agent-based integer optimization algorithm to a “cold start” fleet assignment problem. Results show that the optimizer can successfully solve such highly-constrained problems (129 variables, 184 constraints).

Introduction

SCHEDULE development, a crucial aspect of profitable airline management, involves many steps, including schedule design, fleet assignment, aircraft routing, and crew pairing.¹ 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, due to the absence of an airline revenue and cost model, the objective is to meet the passenger demand throughout the day with a minimum number of flights for a given fleet.

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.² Fleet assignment and aircraft routing problems have been solved using various optimization methods, including integer linear programming,^{3,4} neighborhood search,⁵ and genetic algorithms.⁶

An alternate approach pursued here is to distribute

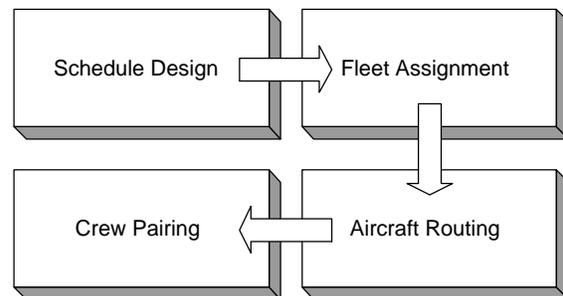


Fig. 1 Airline Schedule Development

the optimization among agents that represent, for example, members of the fleet or the airports in the network. 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 the fleet assignment problem. COIN techniques have been successfully applied to a variety of distributed optimization problems including network routing, computing resource allocation, and data collection by autonomous rovers.⁷⁻⁹

The next section of the paper details the formulation of the optimization problem. This is followed by a description of the COIN framework. Finally, results from an example fleet assignment problem are presented.

Problem Statement

The objective is to determine the aircraft routing and resident fleet size at each airport that minimizes

*Doctoral Candidate, AIAA Student Member.

†Doctoral Candidate, AIAA Member.

‡Professor, AIAA Fellow.

§Senior Research Scientist.

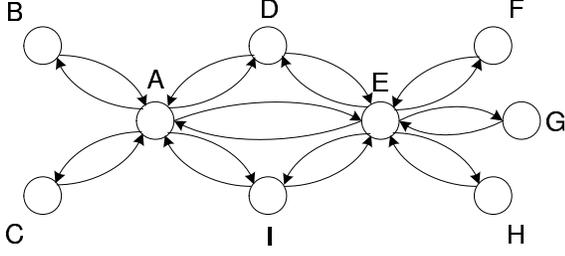


Fig. 2 The 9-airport, 20-arc problem.

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

MINIMIZE: Number of flights
 VARIABLES: Number of aircraft on each arc
 Resident fleet at each airport
 CONSTRAINTS: Passenger demand
 Assignment continuity
 Resident fleet conservation
 Total fleet size

The two types of variables are: $u_{i,j}$, the number of aircraft assigned to flight arc i at time segment j and v_k , the number of resident aircraft at airport k . 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 :

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

$$0 \leq v_k \leq 30$$

The objective function can be written as:

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

There are 20 arcs and 6 time segments in this problem, which, with 9 airports, results in a total of 129 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 constraints to be satisfied:

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

with:

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

While the framework supports multiple aircraft models, in this example problem the fleet is composed of a single aircraft type with a capacity of 100 passengers. 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

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

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 residence 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

Collective Intelligence (COIN) is a framework for designing a collective, defined as a group of agents with a specified system-level objective. Selecting the right types of agents can significantly accelerate convergence. In the case of the fleet assignment problem, two types of agents were chosen. These match the two types of variables: the first is the number of airplanes assigned to each route for each time segment, and the second is the size of the resident fleet at each airport.

The COIN solution process consists of the agents selecting actions (a value from the decision space) and receiving rewards based upon their private utility functions in order 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.¹⁰⁻¹² In particular PD theory handles constraints,¹³ 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.

Basic Formulation

Consider the unconstrained optimization problem,

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

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

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

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 $\mathbb{E}[G(x_i, x_{(i)}) | x_i]$ 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,

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

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

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 including the constraint on total probability, the following update rule is obtained:¹³

$$\begin{aligned} q_i(x_i) &\leftarrow q_i(x_i) \\ &- q_i(x_i) \left[\frac{1}{T} (\mathbb{E}[G|x_i] - \mathbb{E}[G]) + S(q_i) + \ln q_i(x_i) \right] \end{aligned}$$

Role of Private Utilities

Performing the update requires estimating the expected utility of the agents. This is accomplished through Monte-Carlo sampling, with the agents generating actions according to their current probability distributions. Since accurate estimates usually require extensive sampling, the private utility is altered from G to ensure that the sampling results in estimates which have low bias and variance.¹⁴

Intuitively bias represents the alignment between the private utility and world utility. With zero bias, updates which reduce the private utility are guaranteed to reduce the system objective. 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 too much Monte-Carlo sampling.

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.¹⁰ Both of these utilities have zero bias. However, due to the subtracted term, WLU should have much lower variance than TG.¹⁰

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 1 Passenger demand for each arc as a function of time.

Incorporating Constraints

The approach outlined above can be extended to constrained problems by augmenting the world utility with Lagrange multipliers, β_j , and the constraint functions, $c_j(\vec{x})$,

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

where the $c_j(\vec{x})$ are non-negative. In the constrained implementation, the individual agents search for distributions which minimize the augmented objective function. Private utilities have the same form as before with the substitution of the augmented world utility. The update rule for the Lagrange multipliers is found by taking the derivative of the augmented Lagrangian with respect to the Lagrange multiplier, giving:

$$\beta_j \leftarrow \beta_j + E[c_j(\vec{x})]$$

Results

The 9-city, 20-arc fleet assignment problem was solved using the PD theory framework. The problem features a time-dependant, asymmetric demand structure as shown in Table 1. The demand requires 228 flights, yielding an objective of 51,984. The minimum fleet size was found to be 45 aircraft.

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

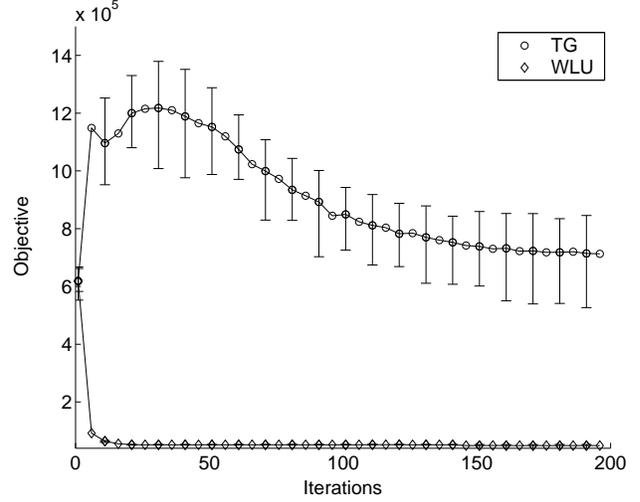


Fig. 3 Comparison of convergence with two private utilities (200 Monte Carlo samples).

problem:

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

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 3. For each utility, the best temperature 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.⁹

As illustrated in Figure 4, 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 5). 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.

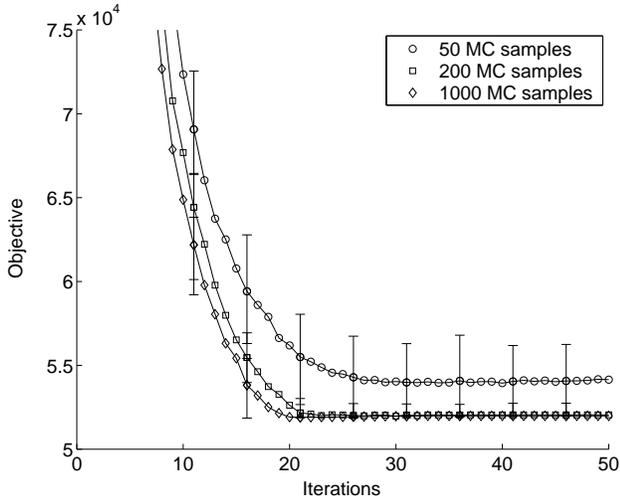


Fig. 4 Increasing sampling speeds convergence (Temperature = 10, WLU, 20 runs).

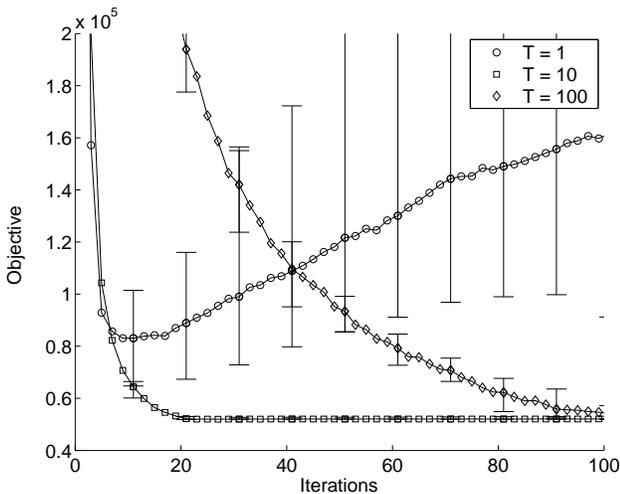


Fig. 5 Effects of temperature on convergence (200 Monte Carlo samples, WLU, 20 runs).

Conclusion

A collective-intelligence framework was successfully applied to a sample fleet assignment problem and yielded optimum solutions. With the basic framework proven to handle highly-constrained design spaces, 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 flight arcs) and developing problem-specific local utilities may also yield faster convergence rates and require fewer Monte Carlo samples.

References

- ¹Etschmaier, M., and Mathaisel, D., "Airline Scheduling: An Overview," *Transportation Science* 19, 1985, pp. 127-138.
- ²Rushmeier, R., and Kontogiorgis, A., "Advances in the Optimization of Airline Fleet Assignment," *Transportation Science* 31, 1997, pp. 159-169.
- ³Abara, J., "Applying integer linear programming to the fleet assignment problem," *Interfaces* 19, 1989.
- ⁴Hane, C., et al., "The fleet assignment problem: Solving a large-scale integer program," *Mathematical Programming* 70, 1995, pp. 211-232.
- ⁵Ahuja, R., et al., "A very large scale neighborhood search algorithm for the quadratic assignment problem," Submitted to *INFORMS Journal on Computing*, 2002.
- ⁶Chung, 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.
- ⁷Tumer, 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.
- ⁸Wolpert, D. H., Tumer, K., "Collective Intelligence, Data Routing, and Braess' Paradox," *Journal of Artificial Intelligence Research*, 2002.
- ⁹Wolpert, D.H., Wheeler, K., Tumer, K., "Collective Intelligence for Control of Distributed Dynamical Systems," *Europhysics Letters*, vol. 49 issue 6, 708-714, 2000.
- ¹⁰Wolpert, D., "Product Distribution Theory," In D. Wolpert and K. Tumer, editors, *Collectives and the Design of Complex Systems II*. World Scientific Press, 2004.
- ¹¹Wolpert, D., and Lee, C.F. "Adaptive Metropolis Hastings," In D. Wolpert and K. Tumer, editors, *Collectives and the Design of Complex Systems II*. World Scientific Press, 2004.
- ¹²Wolpert, D., and Airiau, S. "Coordinate Transformations in Product Distribution Theory," In D. Wolpert and K. Tumer, editors, *Collectives and the Design of Complex Systems II*. World Scientific Press, 2004.
- ¹³Bieniawski, S. and Wolpert, D., "Discrete and Continuous Optimization Using Collectives," In D. Wolpert and K. Tumer, editors, *Collectives and the Design of Complex Systems II*. World Scientific Press, 2004.
- ¹⁴Duda, Richard O., Hart, Peter E. Hart, and Stork, David G., *Pattern Recognition*, Wiley, 2001.