MAAP: the Military Aircraft Allocation Planner

P. Abrahams, R. Balart, J.S. Byrnes, D. Cochran, M.J. Larkin,
W. Moran, G. Ostheimer, and A. Pollington

Abstract— **We present an application of Genetic Algorithms to the field of large-scale allocation problems in which a collection of resources (assets) must be mapped in an optimal or near-optimal manner to a number of objectives (targets), as measured by an objective function. Such problems are complex due to their requirements for integer solutions, non-linear objective functions and linear asset constraints. Genetic Algorithms have turned out to be a natural fit for this application. In this paper, we summarize the scope of the MAAP tool prototype as delivered to the U.S. Air Force and indicate our plans for ongoing and future research.**

I. Introduction

We have addressed the problem of allocation of air assets among a list of targets (e.g., as given by an Air Tasking Order) in an optimal or near-optimal manner. This paper reviews results obtained thus far, discusses the current prototype, and looks at extensions into a software tool suitable for integration into the emerging automated air campaign planning infrastructure.

As part of the work accomplished to date, mathematical mechanisms were developed to address the large scale optimization problems involved in the allocation of a hierarchically organized collection of assets to a hierarchically organized set of objectives. These include the Extended Dependency Model (EDM) for evaluation of costs and benefits within such structures and a Genetic Algorithm (GA) for efficiently finding near-optimal solutions to combinatorially large problems. These mechanisms form the foundation for a prototype automated decision aid.

Our approach has been tested on examples involving tasking of aircraft assets in tactical scenarios. Among the many other real-world decision problems that involve allocation of elementary assets among elementary objectives in order to achieve a high-level goal, we have also investigated the application of our technology to drug traffic interdiction, and to financial investment allocation.

It was agreed that the assignment of military aircraft to targets in a manner that best supports an overall campaign goal should be the prime focus for our allocation approach. In the remainder of this paper, we discuss the fundamental aspects of our methodology and the design of our prototype Military Aircraft Allocation Planner (MAAP). We continue by presenting an example of our prototype's operation in a specific air asset allocation scenario and conclude with our view of the next phase of development of the technology in line with perceived needs of the Air Force.

This work was supported by the U.S. Air Force under SBIR contract number F30602-94-C-0038. All authors with Prometheus Inc., 21 Arnold Ave., Newport RI 02840. Byrnes is also at the University of Massachusetts at Boston. Cochran also at Arizona State University. Larkin also at the Naval Undersea Warfare Center. Moran also at the Flinders University of South Australia. Pollington also at Brigham Young U.

II. Summary of Work Accomplished

The primary thrust of our program addressed *static* asset allocation problems. In such problems, a fixed collection of assets is to be allocated among a fixed collection of objectives in a scenario where crucial parameters such as target value and weapon effectiveness are independent of time.

Our approach is distinguished by a sound mathematical formulation of the static asset allocation problem and the incorporation of two powerful techniques (EDM and GA) into the optimization methodology. As part of our work, we analyzed the additional capabilities that would be needed to achieve operational viability, basing our analysis on information provided by the Air Force.

This section describes the mathematical structure developed for framing static asset allocation problems, the nature of the data necessary to formulate and solve such a problem, the tools used in the solution, and issues encountered in implementing all of these in a computer-based decision aid.

A. *Mathematical framework for static allocation problems*

Our approach to static asset-to-objective allocation problems is based on optimization of a payoff function. In such problems, elementary objectives often contribute to high-level goals in a manner that can be captured by a hierarchical dependency model. EDM is used to compute the importance of low-level objectives to achieving higher-level objectives within the payoff function [3]. A GA is used to find asset allocation strategies that provide near-optimal values of the EDM-based payoff function.

A.1 Mathematical problem formulation

In mathematical terms, the static asset-to-objective allocation problem begins with a collection of elementary assets $\{B_1, ..., B_m\}$ and a set of elementary objectives $\{T_1, ..., T_n\}$. Each asset B_i has an importance r_i associated with it, and each objective T_j has an importance ρ_j associated with it. These importances will generally be evaluated indirectly in terms of dependency models on hierarchical structures of assets and objectives, as discussed in [3]. The goal is to determine optimal or nearly optimal *allocation strategies* in which each B_i is assigned to some T_j (or possibly left unassigned), where optimality is with respect to a payoff function which incorporates the importances of the assets and objectives as well as additional information about costs and effectivenesses associated with the strategy. Examples of such information, which may include both "hard" data and figures derived from "expert opinion," are:

- The *efficiency* $p_{i,j}$ of asset B_i in achieving objective T_j.
- The *value* v_i of asset B_i.
- The *risk* $\gamma_{i,j}$ associated with allocating B_i to T_j.
- The *joint efficiency* $\xi_{i_1,i_2,j}$ added by simultaneously assigning assets B_{i_1} and B_{i_2} to objective T_j.

A.2 Representation of strategy

An allocation strategy can be represented as a binary $m \times n$ matrix X whose $(i,j)^{\text{th}}$ entry is one if B_i is allocated to T_j and is zero otherwise. In many static allocation problems, an individual asset can be assigned to at most one objective. This constraint is reflected in X by the row sums being less than or equal to one:

$$\sum_{j=1}^{n} x_{i,j} \leq 1 \tag{1}$$

Although this constraint is part of our current MAAP prototype, we recognize that dynamic allocation of air assets will require multiple assignments per asset.

A.3 The Measure of Effectiveness

The nature of the payoff function to be optimized in an asset allocation problem depends substantially on the particular application. In the context of MAAP, the payoff function $J(X)$ is called the "measure of effectiveness" (MOE) of the allocation strategy X. The role of MAAP is to assist the user in quantifying the relative importances of numerous assets and objectives, the effectiveness of each asset against every objective, and the potential risks and costs involved in using any asset against any objective. MAAP then employs this quantification to provide the user with a list of candidate strategies having optimal or nearly optimal MOE's. It also allows the user to evaluate the MOE of any proposed strategy.

If the efficiency $p_{i,j}$ is interpreted as the probability of the event "elementary asset i will achieve elementary objective j if so allocated" and these events are statistically independent, then the probability of elementary objective j being achieved by allocation strategy X is

$$\Pi_j = 1 - \prod_{i=1}^{m}(1 - x_{i,j}p_{i,j}). \tag{2}$$

Thus the MOE contains a term of the form

$$U(X) = \sum_{j=1}^{n} \rho_j \Pi_j \tag{3}$$

which weights the probability each elementary objective is achieved by its importance.

The degree to which the events fail to be independent are captured by a term $V(X)$ which quantifies the additional "benefit" of allocating multiple assets to the same objective. In the current MAAP prototype, this term incorporates data $\xi_{i_1,i_2,j}$ describing the marginal benefit of allocating both B_{i_1} and B_{i_2} to T_j:

$$V(X) = \sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\sum_{j=1}^{n} \xi_{i_1,i_2,j} x_{i_1,j} x_{i_2,j}$$

We note that the $\xi's$ can be negative as well as positive (e.g., it is undesirable to pair a stealth bomber with a more detectable aircraft for a surprise attack).

The final term in the MOE has the form

$$Y(X) = \sum_i v_i \sum_j \gamma_{i,j} x_{i,j} \tag{4}$$

and measures the cost of the strategy X. This cost includes deterministic costs, such as fuel and ordnance, as well as statistical costs, such as risk of asset damage or loss.

Combining these terms, the MOE used in the MAAP prototype is

$$J(X) = AU(X) + BV(X) - \Gamma Y(X). \tag{5}$$

The weights A, B, and Γ allow the importance of each term to be adjusted in the overall MOE.

B. Optimization

The combinatorics of the problem of finding the optimal allocation strategy are such that exhaustive search for an optimal strategy becomes inviable for even modest collections of assets and objectives and computationally simple cost functions. Assuming that each asset can be allocated to at most one objective, a collection of m assets and n objectives yields $(n+1)^m$ possible allocation strategies. A problem involving 15 assets and 10 objectives in which the MOE requires 1 microsecond to compute would require over 130 years to optimize by exhaustive search, for example.

Of several efficient search methods known for addressing combinatorially large optimization problems (e.g., gradient methods, simulated annealing, etc.), GA's offer some features that are particularly attractive for problems involving complicated non-convex payoff functions. They are faster than simulated annealing and less likely to stagnate in local extrema than gradient methods. In addition, under assumption (1), allocation strategies are conveniently represented as "genes" in a way that will be described below. The principles of GA's, which seek the "fittest" genes (i.e., best strategies in terms of J) by a Darwinian paradigm, are presented in [8]. In our context the change in population from one generation to the next involves first random mutations at randomly chosen sites on each gene, followed by a pairing of the genes. The pairs are randomly divided to produce progeny, thus generating a population of children and parents of twice the original size. This is culled by selection of the fittest, though some of the best parents are also retained. The GA was found to work better with this modification. Several different values for the various probabilities were tested to find a suitable one.

C. Air asset allocation example

A simple example illustrating the function of the decision aid follows. A set of seven elementary aircraft assets is available:

- B_1 = B2 stealth bombers
- B_2 = B52 heavy bombers

- B_3 = F111 fighter-bombers
- B_4 = F16 multi-mission fighters
- B_5 = F16 multi-mission fighters
- B_6 = F4-G Wild Weasels
- B_7 = F4-G Wild Weasels

These assets are to be employed in a conflict where the overall objective "Degrade the opponent's war sustaining capabilities" is supported by numerous sub-objectives in an objective hierarchy. One cluster of elementary objectives is:

- T_1 = Port, AAA protection, relative importance 0.20
- T_2 = Railyard, no protection, relative importance 0.10
- T_3 = Large petroleum reserve, SAM protection, relative importance 0.35
- T_4 = Warehouses, AAA protection, relative importance 0.10
- T_5 = Small petroleum reserve, SAM protection, relative importance 0.20
- T_6 = Bridges, SAM protection, relative importance 0.05

Using a substantial database describing the individual and pairwise effectivenesses and costs of each elementary asset against each elementary objective, a GA optimizer produced several candidate solution genes. A gene is a vector of length m (number of assets) in which the i^{th} element is the index j if B_i is to be allocated to T_j and zero if the asset is not used. Under assumption (1), every strategy X can be represented in this way.

The strategies with the highest MOE's were (1,2,4,3,5,3,3) and (3,2,1,4,5,5,0), which are depicted in figure 1. Readers familiar with the nature of these assets and objectives can judge the reasonableness of the solutions obtained (bearing in mind that they reflect the underlying data used). It is interesting to note that the second-best strategy obtained does not use one of the assets.

D. Discussion

The problem of optimal allocation of assets to objectives requires that an MOE be assigned to each possible allocation strategy. We believe this is not only desirable, but necessary, since it is with respect to such a measure that "optimality" of the strategy is defined.

We first used EDM as a framework for computing the MOE of allocation strategies that assign assets at the lowest level of an asset hierarchy to objectives at the lowest level of an objective hierarchy. Optimal asset-allocation assignment problems are thus a natural setting for EDM application.

Next we developed the MOE (payoff function) $J(X)$ (5), which combines EDM output with parameters such as the efficiency of particular assets when used against specific objectives. Continuing, we chose GA's to attack our complex optimization problem, and tuned the GA parameters to fit the current magnitude of the required calculations.

Finally, we have tested our approach on numerous examples, one of which is described in section II-C. At the present time the type and scope of examples that we are able to consider are limited by certain assumptions required under the current level of effort. An outline of our plan for removing these limitations, completing our development of a software tool suitable for integration into the emerging air campaign planning infrastructure, and performing this integration, is given in the next section.

III. Ongoing and Future Research

The effort described above has produced an automated decision aid suitable for use in static air allocation problems. During consultation with Air Force personnel regarding asset allocation in real-world operational scenarios, several considerations beyond the scope of the prototype framework emerged as crucial:

- **Dynamic information.** MAAP currently handles evolving scenarios by allowing the user to enter new information as it becomes available and then re-optimizing the asset allocation strategy in light of the new information. In actual air allocation problems, the effectiveness of a strategy typically depends on a large number of time-varying conditions such as weather, actions taken by the adversary, availability of EW and tanker support, damage to or loss of assets, and the degree of success of earlier missions. In many cases, an essential part of the problem is choosing the best time to deploy each asset.
- **Mission sequencing.** Realistic allocation strategies must include time information (e.g., times of takeoff, on target, and rendezvous with tankers) and location information (e.g., positions of assets, targets, and opposing air defenses) to ensure proper sequencing of attacks, coordination of EW and tanker support, and scheduling of multiple sorties per aircraft. Inclusion of such information in the strategy is also essential to support re-allocation of assets in response to evolving information about targets, defenses, and available assets.
- **Multiple targets.** The current requirement that an asset be assigned to only one objective (e.g, target) per mission is too restrictive. If an asset is to attack more than one target on a mission, then the weapon loadout must be chosen to support all attacks.
- **Availability of armaments.** The effectiveness of an aircraft against a particular target type depends on its weapon loadout. In evaluating the effectiveness of a multi-mission aircraft against a target, MAAP currently assumes the availability of the most suitable loadout. In practice, the best weapons may not be immediately available (e.g., if an aircraft is re-allocated to a new target after takeoff).
- **ATO Perturbations.** When recommending changes to today's war (e.g., if a new high-priority target is identified), perturbations to the existing Air Tasking Order (ATO) must be minimized. Deviations from the Commander's guidance should occur only in uncommon circumstances.

These considerations require that the current MAAP model be extended to address *dynamic* asset allocation problems. Such problems, we believe, can be addressed by generalizing the above approach. The ongoing effort is therefore building upon the results of this earlier work to develop the needed extensions of the mathematical framework and database structures.

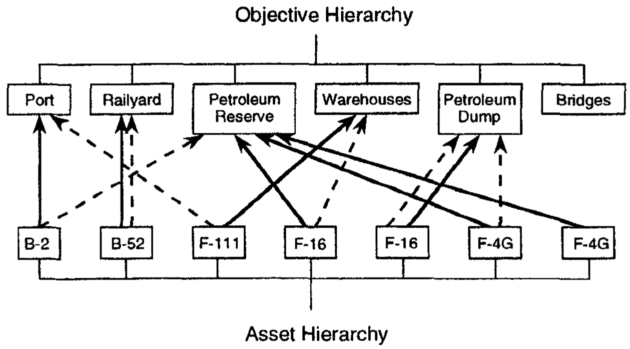

Fig. 1. Two near-optimal asset-to-objective allocation strategies identified for a tactical air example. The solid arrows represent gene (1 2 4 3 5 3 3) with MOE 0.584; the dashed arrows represent gene (3 2 1 4 5 5 0) with MOE 0.577. Note that the second solution leaves one asset unallocated.

The necessary generalizations will entail additional complexity in both computation (particularly in GA optimization) and database design and maintenance. The remainder of this section describes capabilities that are being developed and outlines how they will be realized by extensions of the existing mathematical framework, database structures and tools, and computational paradigms.

A. A Dynamic MAAP

We perceive the most critical step in extending our MAAP tool prototype to be incorporation of temporal and spatial information into allocation strategies. The same framework necessary to support this extension will also allow inclusion of time-varying data such as weather conditions.

With the addition of such information, MAAP will evaluate and optimize asset allocation strategies in which the following are taken into account:

- Target priorities that vary with time and depend on the status of other targets (e.g., destruction of a certain bridge before the opponent's tank column reaches it is crucial unless another bridge along the route has already been destroyed).
- Temporal and spatial coordination of assets in a strategy (e.g., the bombers should arrive at the refueling point before the tanker leaves and arrive at the target after the Wild Weasels attack its SAM defenses).
- Asset lists that change as aircraft arrive in the theater, are damaged or lost, require reloading/refueling, etc.
- Asset effectiveness and vulnerability that depend on weather, degree of darkness, distance and route to the target (e.g., need for additional fuel may allow fewer weapons to be carried), and on how other assets are allocated.

B. Extensions of the mathematical framework

In the static framework we have developed for our current prototype, an allocation strategy consists of an assignment of each asset B_i to a single target T_j or to no target. To accommodate multiple targeting and time-varying parameters in MAAP, the notion of an allocation strategy will be extended so that each B_i is allocated to a *route*. A route consists of a starting time and an ordered sequence of objectives. An objective will generally be a target, but may also be a tanker rendezvous, an air superiority or reconnaissance mission, or another critical milestone in the route.

Mathematically, the route for asset B_i is a pair (τ_i, S_i) where S_i is a sequence of objectives $T_{j_1}, ..., T_{j_{n_i}}$ and τ_i is the initial departure time for the asset. The alternative is to treat a route as a sequence of pairs (T_{j_k}, τ_{j_k}). Our selection of the former model is based on the realistic assumption that the time between objectives under given environmental conditions is fixed. This assumption dramatically reduces the search space.

Since a route implicitly specifies the time at which an asset is at a given target, time-varying parameters may be accommodated. Moreover, the MOE for an asset-to-route allocation strategy can depend on whether two or more assets are allocated to routes resulting in a particular sequence of events (e.g., whether some Wild Weasel has a particular target on its route before a bomber arrives at that target).

The size of the space of routes depends on the number of objectives, the resolution of the time indexing, and whether an asset is allowed to visit a given target multiple times in a route. Even under conservative circumstances, a small number of targets will yield a collection of routes so large that even a GA cannot produce near-optimal asset allocations in a timely way. To address this issue, we note that most of the routes defined in a given problem are not viable. B_i cannot attack T_{j_1} and T_{j_2} within a ten-minute period if they are hundreds of miles apart, for example. Limits on radius of operation, speed, payload, etc. lead to rules that can be used to prune the space of routes to a substantially smaller space of *feasible routes*. It is important to note that, while the space of all routes depends only on the T_j, the space of feasible routes depends on both the B_i and T_j.

When information in a scenario changes, such as weather conditions or available assets, we can exploit the previous solution by performing an incremental analysis — re-optimizing only over a severely pruned route space. For example, the search could be limited by allowing the route for B_i to change only by addition of a single target and only if all other targets and times in its previous route are unchanged. Using new information in this way allows it to be exploited while forcing "on-the-fly" changes to each individual aircraft's itinerary to be minimal (thus curtailing incremental changes to the ATO and adhering to the Commander's instructions).

We anticipate that this approach will allow a GA to identify near-optimal allocation strategies for time-varying problems in a computationally tractable manner — particularly when parallel implementation (see section III-C.2) is employed.

C. Implementation Issues

Achieving the extensions discussed in the previous section will require major augmentations to our current MAAP software architecture. These will affect the database contents and management approach, the GA optimization engine, and interfaces to external data.

C.1 Database

The current MAAP database structure consists of a master database (MDB) and scenario specific databases (SSDB's). For each asset type, the MDB contains cost information and loadout descriptors. It also contains a list of valid *objective node* types, each consisting of a target type and air defense configuration descriptor. The MDB stores the effectiveness and vulnerability of each asset/loadout configuration against objective node type and the joint effectiveness of each asset/loadout configuration paired with every other asset/loadout configuration against each objective node type.

The SSDB is based on a list of assets and objective nodes involved in a specific scenario together with the hierarchical structures used by EDM in computing the MOE. The SSDB extracts data from the MDB specific to the assets and objectives involved in the given scenario.

The MDB is stored in a formatted ASCII text file which is created and managed using a text editor. Building of SSDB's is supported by a small library of software tools we developed. This approach to data management has suited development of the MAAP prototype well. The MDB can be examined on a computer display, modified with an editor, and printed on a printer. Both databases are sufficiently small that disk storage and lookup speed limitations have not been significant factors in prototype development and testing. Moreover, simplicity of the SSDB structure and tools has expedited saving and retrieving scenarios through the GUI.

In the extended framework, MAAP will retain the MDB/SSDB concept, but both will require significant upgrades to support the mathematical approach described above:

- More asset characteristics will be stored in the MDB. These include expanded loadout descriptors, operational radius as a function of loadout and speed, and effectiveness and vulnerability statistics that depend on environmental conditions, presence of supporting aircraft, and target status.
- The SSDB will incorporate asset and objective location, the locations of non-targeted air defenses (which could influence routing, for example), and the joint importance of combinations of targets in addition to their individual importances.
- The database files will be binary to support efficient disk storage and fast retrieval.
- A database editor will be needed to allow accurate and efficient management of the MDB.
- A database merger will also be provided to create new SSDB's and MDB's by combining existing files.
- Highly efficient database tools for real-time operation will be developed.

In addition to these features, we anticipate that developing and using utilities to import data from existing Air Force software and export data for their use will be essential for making MAAP useful in actual campaign planning.

C.2 Optimization Engine

The GA optimization approach, which has proven successful so far, is being retained and refined for the dynamic MAAP tool. As discussed above, however, introducing time and other dependencies into the asset allocation problem leads to a combinatorial explosion in the search for near-optimal allocation strategies. We anticipate that some enhancements to the MAAP optimization engine will be needed to deal with this issue:

- Heuristic rules will be developed to reduce the size of the space of asset-to-route allocations by removing inviable routes. The GA will be employed only on the remaining "feasible region."
- The possibility of developing "heuristic metrics" which can predict that the MOE's of two strategies will be similar without evaluating them explicitly will be investigated. If such exist, they can be used to avoid searching extensively within regions of the space where all strategies have nearly the same MOE.
- The GA software will be implemented to allow parallel execution. Exploiting several computers in the optimization process will allow a more thorough search of the space within a fixed time limit. Alternatively, it will reduce the time needed to run a GA search with fixed parameters.

Various parameters govern the behavior of the GA itself, and we are currently investigating variants of the basic algorithm to increase stability and improve convergence. This investigation is being continued because of the criticality of this component in extending MAAP to handle dynamic asset allocation problems.

C.3 Data Interfaces with Air Force Software

We recognize the need to ensure compatibility between MAAP and other components of the Air Force's automated campaign planning structure. Our current MAAP prototype has been constructed with a view toward importing data from and exporting data for use by other software systems. The dynamic MAAP tool will build on this by allowing users to incorporate external data directly into scenario descriptions and supporting display of external data in formats used by existing Air Force software tools. We currently have specific ideas for integrating MAAP with ACPT (Air Campaign Planning Tool), APS (Advanced Planning System), and FLEX (Force Level Execution), and their components CTAPS (Contingency Theater Automated Planning System), CTEM (Conventional Targeting Effectiveness Model), EADSIM (Enemy Air Defense Simulation), and RAAP (Rapid Application of Air Power).

IV. Conclusion

The MAAP framework described above has demonstrated the power and utility of the EDM and GA approach. We are now proceeding to meet the challenges described in section III. The future implementation of MAAP will provide the Air Force with a unique software tool that has the potential to become a significant component of the automated air campaign planning infrastructure.

References

[1] R. Balart, "Graphical User Interfaces for Automatic Decision Aids Software," *Proceedings of the 5th Annual IEEE Dual-Use Technologies Conference*, Utica NY, May 1995.

[2] R. Balart, J.S. Byrnes, D. Cochran, M.J. Larkin, and W. Moran, "Decision Aids for Asset-to-Objective Allocation," *Proceedings of the 29th Asilomar Conference on Signals, Systems, and Computers*, October 1995.

[3] J.S. Byrnes and R.C. Angell, "The Dependency Model: A tool for calculating system effectiveness," *IEEE Transactions on Reliability*, vol. R-34(1), pp. 17-23, April 1985.

[4] T.A. Clark, *Force Level Execution Program Briefing.* 1994.

[5] T.A. Clark, G.H. deWalder, E.C. LaBatt, Jr. and G.C. Ruigrok, *Force Level Execution (FLEX): Developing and Demonstrating a Monitoring and Execution Capability for Combat Operations.* 1992.

[6] L. Davis, ed. *Genetic Algorithms and Simulated Annealing.* Morgan Kaufmann Publishers, 1987.

[7] L. Ekberg, *Background on Rapid Application of Air Power (RAAP).* 1994.

[8] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[9] E.C. LaBatt, Jr., *Marquee: A Graphical Mission Monitoring Tool.*

[10] M.J. Larkin, D.S. Chambers and J.S. Byrnes, "Using the Genetic Algorithm Dependency Model as a tool for financial asset allocation," Prometheus Inc. Technical Report, 1995.

[11] M.J. Larkin and J.S. Byrnes, "Using the Genetic Algorithm Dependency Model as a tool for drug interdiction," Prometheus Inc. Technical Report, 1995.

[12] M.J. Larkin, W. Moran and J.S. Byrnes, "Evolutionary computing, neural networks and tactical decision aids," *Presented at the 5th Annual IEEE Dual-Use Technologies Conference*, Utica NY, May 1995.

[13] W. Moran, *Genetic Algorithms and the Dependency Model.* Prometheus Inc. Technical Report, 1994.