13th ICCRTS
"C2 for Complex Endeavors"

Planning for Task Assignment Agility in Large Heterogeneous Teams

Paper Number 087

James C. Kilian
Lockheed Martin
77 S. Bedford St.
Burlington, MA 01803
781-565-1059
james.c.kilian@lmco.com

Planning for Task Assignment Agility in Large Heterogeneous Teams

James C. Kilian
Lockheed Martin

Abstract

There are many circumstances in which a mission revolves around preventing the deployment of an adversary's assets. Ballistic/tactical missile defense, blockades, force protection, checkpoints for controlled area access and air defense applications all fit this paradigm. In these sorts of problems, uncertainty around the expected behavior of the adversary can be substantial. In their recent book, Alberts and Hayes distinguish between complicated and complex endeavors. In a complex situation, causal relations may no longer be well-understood and directly available for analysis. They may also be characterized by instability in the sense that small perturbations in initial conditions or temporal behaviors result in large changes in the resulting mission outcomes. In response, agility of planning and decision-making systems becomes paramount over notions of static optimality. A cooperative co-evolutionary computation approach to planning is derived here that is directed toward achieving agility in the task assignment configurations of large heterogeneous teams. Architectural context is also discussed in terms of fitting co-evolutionary computation strategies engaged in complex endeavors into Service-Oriented Architectures (SOA) and Event-Driven Architectures (EDA).

1 Background

1.1 Mission Planning

As planning is the focus of this analysis, and given that planning may have different interpretations in different fields, the following textbook definition is offered as a starting point [15, p1]:

> "Planning is the reasoning side of acting. It is an abstract, explicit, deliberative process that chooses and organizes actions by anticipating their expected outcomes. This deliberation aims at achieving, as best as possible, some prestated objectives."

Planning problems are often categorized by whether complete or partial information on the state of the world is available to the planner, or on whether or not the consquences of actions on the state of the world are deterministic, or on whether time-dependence both of a dynamic world and of plan validity are important, or on whether finite resources must be

managed. Plus, many applications require simultaneous planning for multiple agents, which introduces a distinction between individual and overall objectives. Multi-agent planning consists both of simultaneous planning for multiple agents and planning by multiple agents based on coordination and/or negotiation among those agents [11, 12]. Of interest here is the subset of multi-agent planning in which a multi-agent plan is a distribution of action sequences across the set of agents to achieve a common goal within sets of local and global constraints.

Every planning domain raises its own issues and benefits from suitably matched planning strategies. Mission planning is an umbrella term that adds, to the standard planning definition given above, the selection of capabilities to apply, the deployment of platforms carrying those capabilities, and the formation and dissolution of organizational units charged to achieve the mission [18]. For the purposes of the current analysis, three necessary and distinct stages in the mission planning process are defined

1. Asset selection, deployment and initial tasking (Pre-mission planning)
2. Post-deployment plan refinement (Pre-engagement planning)
3. Reactive adaptation to real-time events

These three phases of this flavor of mission planning are shown in Figure 1. Each stage represents a different level of commitment of resources and a different decision-making timescale. Cases of interest here are those in which assets are committed in the first stage prior to any direct engagement. Decisions in this case are presumed to be made based on acquired intelligence which may be incomplete or uncertain. For example, in a ballistic missile defense scenario, the assets may not be very mobile relative to the timescales of the engagement part of a mission, and so, once deployed, they can't be significantly re-deployed. Once an engagement scenario begins, the relevant timescale is on the order of minutes [19, p 17] rather than the days or weeks that might be available in other mission scenarios. This type of situation, where assets must be selected and coarsely organized prior to engagement, is the most salient characteristic of the mission planning domain treated in the present analysis.

1.2 Planning and Causality

There are a variety of specific representation schemes for planning problems. Most however, have some notions of action, observation, world state, information state, and constraint representations. Examples include state-variable, plan-space, and task network representations [15]. There are, in some treatments, provisions for autonomous processes, which may influence plans but are not influenced by plans. An adversary's actions or dynamic environmental elements like weather are examples of autonomous processes [22]. Autonomous processes are significant not only in that they influence the planning process, but partial observability into autonomous processes leads to the need to address uncertainty in the planning process.

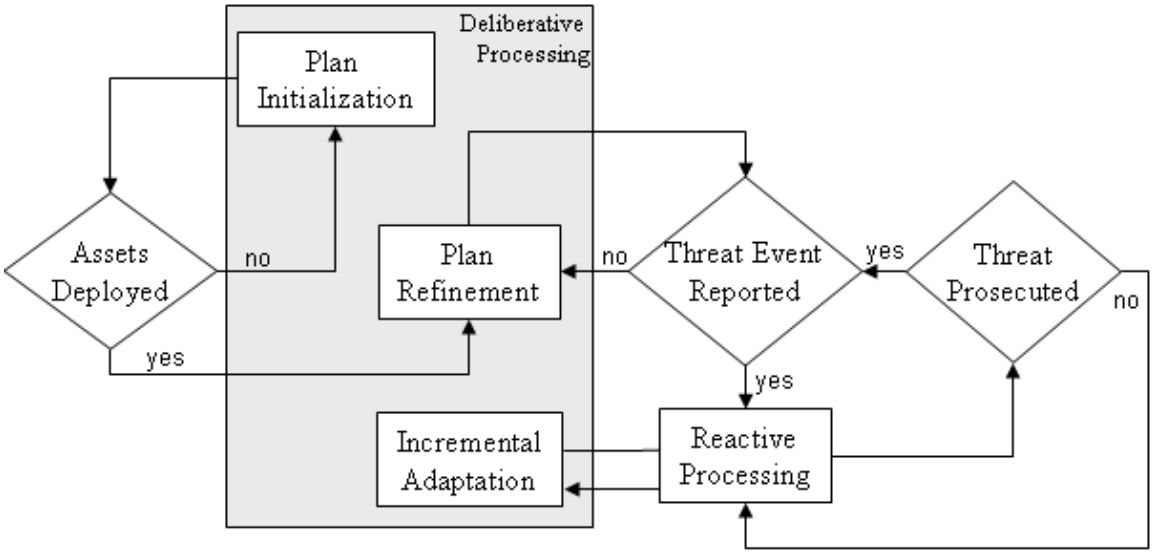


Figure 1: A three-stage model of deliberative processing provides the context for the mission planning problem definition treated here.

Planning is grounded in the notion of causality. In order for an agent to predict or evaluate a future step, a causal relation must exist between actions and state variables, between the autonomous processes and state variables and between state variables, autonomous processes and observations by an agent. Causal knowledge then either exists *a priori* or is inferred from observation. A corpus of causal assertions define optimality or sufficiency conditions for a plan. Invalidity of one or more of these causal assertions may render a plan invalid. Issues of plan robustness, therefore, are tied directly to the inviolability of the corpus of causal assertions that form the foundation of that plan. Classically, causality represents necessity but not sufficiency. Modern notions of causality do incorporate concepts of uncertainty and incompleteness and so causality can be modeled probabilistically [28]. Causal with unit probability generally recovers the classic deterministic notion of causality.

In their recent book, Alberts and Hayes [2] distinguish between complicated and complex endeavors. The former is defined as a situation in which causal relations are well-defined and understood. Complication, therefore, can usually be mitigated by direct application of additional resources to reduce individual task loading. In a complex situation, however, changes to the situation are uncertain, interdependencies between events may not be well-understood, and situations may also be characterized by instability in the sense that small perturbations in initial conditions or temporal behaviors result in large changes in mission outcomes. Robust planning was a first step to cope with complexity in the sense used here. Decision-theoretic planning techniques capture uncertainty in the planning domain by modeling dynamic variables as stochastic processes and planning itself as a Markov decision process (MDP) [6, 15], as partially observable Markov decision processes (POMDP) and their approximations and variants [21, 31, 33, 37]. These techniques are nonetheless still concerned with planning to some mean state variable response with a probability of success driving the selection of actions. Thus, those techniques still assume that the world is well-behaved and that causality is mostly intact. The concept of planning for agility in response to unexpected circumstances is not really captured in a straightforward implementation of decision-theoretic planning. Alberts and Hayes go on to assert that to maintain plan agility actions should be pursued that

1. Minimize the resources which are irretrievably committed
2. Improve the information position relative to the adversary's
3. Shape the adversary's information position
4. Monitor the impacts of actions taken
5. Place the adversary in a reactive mode

The authors also note that for agility, sets of actions must insure [2, p 131]

> " ... that their approaches are flexible (provide more than one way to achieve success), robust (are effective across a variety of circumstances in order to allow for changes in the situation), resilient (permit recovery from missteps or adversity

arising from a lack of full understanding), responsive (able to act within windows of opportunity), innovative (able to do new things or old things in new ways so that they have a greater chance of success than taking actions adversaries can anticipate), and adaptive (permitting changes in both processes and organization as information is gained about the complex situation)."

The probabilistic representations in decision-theoretic planning approaches can be combined with dynamic planning techniques like conditional planning (contingencies), plan monitoring and repair, and continual planning [11] in order to achieve agility in a by providing a planner mechanisms to deliberatively respond to a dynamic and uncertain situation. In a networked system of assets, task assignments can be made almost instantaneously relative to mission timescales and within the capability constraints of the set of assets. An agile system would be able to take advantage of tasking plasticity and make optimal task assignments in response to unfolding circumstances in any of these three planning paradigms. However, the spatial distribution of assets committed to the mission along with the capabilities of individual assets must be such that choices in tasking configuration are supported.

There is also some relationship between a pursuit of task assignment agility and the set covering problem (SCP) applied to distributing nodes in a sensor network. The set covering problem is a well-established optimization problem that models many resource selection situations. The classic set covering problems can be stated abstractly by first defining a universe U a family, S, of subsets of U. A cover is a subfamily $C \subseteq S$ of sets such that $\bigcup c_j = U$. For the set covering decision problem, one asks, given (U, S) and an integer k, if there is a set covering C of size k or less. The set cover optimization problem asks, given (U, S), for the smallest set cover [8, Sec 35.3]. Cardei [7] has extended the classic SCP to accommodate heterogeneous elements and applied that extension to the problem of an energy-constrained optimization of target coverage by a network of heterogeneous sensors. Dambreville [10] used cross-entropy to perform a similar analysis and maximize the (space-time) area of tracks under observation. Others have extended these concepts to minimize the closest distance to any sensor at any point along the object trajectory or to minimized the integral of the sensor intensity along the trajectory [23, 24]. The idea then is to distribute resources such that the system will detect arbitrary incursions. Assumptions on the spatial distribution of an adversary are relaxed relative to traditional optimal resource allocation problems. The difference between the set covering problems and planning for task assignment agility lies in the fact that the distribution of assets in the latter case is based on multiple task capabilities rather than a single task capability (detection), and so what are being maximized are the number of viable task assignment configurations in this multiple task scenario.

The goal of this analysis is to capitalize on these concepts for mission planning situations in which the three-stage model of planning, shown in Figure 1, is relevant. In essence, a loose coupling between (1) the selection and distribution of assets and (2) the task assignments, given those assets, drives the approach. These situations tend to be large scale with multiple timelines. Naval missile defense problems fall into this category as do incursion prevention

in some homeland defense situations and force protection in asymmetric conflicts. Rather than optimizing the sequence of actions needed to achieve the mission, the agility of the set of assets to respond to unexpected as well as expected situations will be optimized.

2 Approach

The basic approach to achieving task assignment agility for mission planning is to define an optimization technique for defining and deploying teams of assets so that the greatest number of possible task assignments to that distribution of assets are preserved. Teams of assets may either be homogeneous, where all members are generalists - able to perform all tasks in the mission, or heterogeneous teams blending generalists and specialists. Specialization tends to reduce agility for a fixed number of assets. In cases where the number of assets is limited and in situations with high uncertainty, generalists may be preferred over specialists in agile tasking systems.

Evolutionary computation is a family of popular techniques for discrete optimization and learning problems, especially those characterized by high-dimensional search spaces [14]. Much of the terminology associated with evolutionary computation is derived from its biological inspiration, but in essence the techniques focus on generating and updating a sampling distribution over the objective (fitness) function. Basic evolutionary approaches to optimization involve only three operators on a population of candidate solutions: Selection, cross-over, and mutation. Given a population of candidate solutions, n-ary sequences, evolutionary computation techniques encourage promising sequences to exchange partial sequences, cross-over, then select the next population generation. Occasionally, an element in a sequence is randomly altered, mutation, in order to expand the coverage of the search space and prevent premature convergence. This procedure is iterated through multiple generations until conditions are met that stop the process. That description is, in a nutshell, an account of a genetic algorithm, one of the variants of evolutionary computation. There are techniques, though, that explicitly create, and sample from, the joint sampling distribution over the fitness function, which are called probabilistic model-building genetic algorithms (PMBGA). It has been shown that PMBGA have convergence rates that scale as $o(n \ln n)$ or as $o(n^2 \ln n)$ compared to standard cross-over techniques that can scale exponentially with the dimension of the search space [30]. For this study, we've primarily relied on a simple PMBGA called Probabilistic Incremental Learning (PBIL) [4], which treats genes as statistically independent. We've also done some additional work with a slightly more complicated model, Mutual Information Maximizing Input Clustering (MIMIC) [5], which models pairwise ordering of variables and maximizes the mutual information between adjacent positions in a chain. There are still other models that can learn even more complex dependencies between variables [29], but the generally sequential dependencies between part-tasks in many missions is a good match to pairwise dependencies modeled in the MIMIC technique. Furthermore, the trade-off between expressibility and complexity reinforces that choice.

Mission planning often targets the actions of multiple heterogeneous assets working toward a common goal. Historically, these organizations tend to be strongly hierarchical with a centralized approach to planning, command and control. More recently, netcentric operations have become part and parcel of the re-envisioning of many organizations. Distribution of a heterogeneous set of asset capabilities provides much more flexibility for the planner than concentrating that same capability within one or just a few assets. Redundancy provides failover capability increasing the overall reliability of a given system. Distributed processing can also be inherently parallel, and so computational efficiencies and scaling advantages may be gained through distributivity. The planning operation itself can also reap these benefits when implemented in a distributed fashion. As the number of assets increases or, in an adversarial situation, the number of threats that must be countered increases, the benefits of distributing the planning process across the agents themselves become more apparent. Thus, dense asset distributions in target rich environments favor a decentralized distributed approach to mission planning. Mali and Kambhampati [20] review a taxonomy for direct distributed planning systems:

1. Local planning and merging. A supervisor process resolves conflicts then merges indivdidual plans
2. Hierarchical planning. High-level goals are successively decomposed into finer-grained objectives until a concrete executable plan results
3. Partial global planning. Agents communicate local plans, including interactions, to build plans that are partially global.
4. Meta-level control-based planning. Local planning constrained by a high-level strategic plan

Except for the partial global planning strategy, each of these involves some degree of hierarchical control with a subset of agents specializing in either merging lower level plans or flowing down higher level plans. There are also indirect approaches to distributed planning but these are not considered here.

Evolutionary computation techniques are often applied to locally optimize individual plans within either a local planning and merging approach or within a hierarchical planning approach. An additional twist to the evolutionary computation approach normally used is to implement a decentralized distributed version in the same vein as partial global planning techniques mentioned above. Co-evolutionary computation techniques are a concrete instantiation of co-evolutionary concepts that drive netcentric concepts of operation [1]. Co-evolutionary computation can be divided into cooperative and competitive co-evolution techniques. In competitive co-evolution fitness is determined by direct contests between agents. The more "victories" the greater the likelihood that that agent's genetic material will contribute to future generations [3, Ch 4]. In cooperative co-evolution, the agents share the consequences of successes or losses. Starting with the early architectures [32] for a cooperative co-evolutionary genetic algorithm (CCGA), multiple species are evolved, then individuals are evaluated in a central domain model which then informs the individual species.

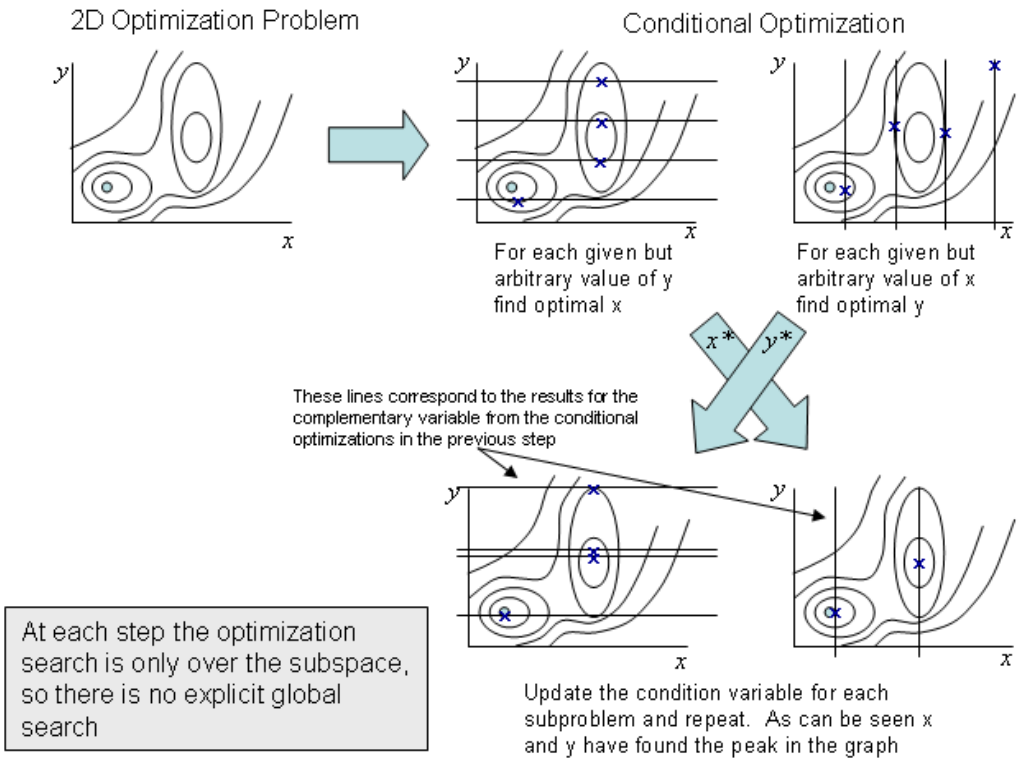


Figure 2: Iterative factored optimization can be the foundation of a partial global planning technique.

Building on this early work, both Orgeta [27] and Subbu and Sanderson [34] developed an approach to CCGA that is completely decentralized. The idea is to factor the search space into subspaces then distribute the search problems for the set of subspaces to the distributed set of agents. Periodic communication among all assets updates the current partial solutions produced by each asset and the process repeats until all searches converge. Each asset then possesses an optimal global plan without need for a separate supervisory process to merge plans or control the process. Thus, the process is self-synchronizing. Figure 2 illustrates this procedure. This approach to iterative conditional optimization is the approach adopted here, in which to embed a PMBGA, with the intent of producing agile distributions of assets for mission planning.

3 A Fitness Landscape for Tasking Agility

So, finding optimal plans, at least in a static sense is not of primary concern. The environment is dynamic. The adversary may do something unexpected. The details of any potential situation are uncertain. The primary question, given the basic approach already described, addresses the nature of fitness when optimizing for task assignment agility. In describing a fitness function both gains and costs associated with actions taken by a set of assets must be captured. The form of the fitness function is not based on any specific assumptions at this level of description as to the nature of the mission or of the particular scenarios to which it may be applied.

Taking advantage of the phased nature of mission planning (Figure 1), the goal is to address two subproblems. The first is to find a distribution of assets that preserves the maximum number of different possible distinct tasking assignments. The second subproblem is, given a distribution of assets, find the tasking assignments that maximize performance as uncertainty is resolved as the mission plays out.

3.1 Subproblem 1 Fitness: Gains

Given a high-level task sequence and a set of assets, one can define an assignment matrix, A, of binary values that associate an asset to zero or more tasks. For example, for three assets and three tasks one might have an assignment matrix

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \tag{1}$$

where an asset (row) is assigned tasks (columns) as indicated by unit values in the matrix. If an assignment is executed, then there is some notion of success for each task that might be written as

$$g_A(a_j, \tau_m) = \begin{cases} 0 & \text{If the assignment failed at the jth task} \\ 1 & \text{If the assignment succeeded at the jth task} \end{cases}$$

given a world state trajectory τ_m. A successful assignment for a given set of assets is one for which every task is successfully achieved. From this abstract representation of success for an assignment, we can also define

$$s_A(\tau_m) = \min_j g_A(a_j, \tau_m)$$

so that if $s_A = 0$, then there is at least one task at which the assignment fails and so the assignment itself is not successful. It also follows then that for multiple assignments, A_k, if

$$S_m = \max_k s_{A_k}(\tau_m)$$

then $S_m = 0$ implies that no assignment is successful (*i.e.*, succeeds at all tasks), and if $S_m = 1$, then at least one assignment is successful. To support agility in task assignment, then it is desirable to minimize the likelihood that there are any world states for which no assignment is successful. If

$$Q \equiv \min_m S_m,$$

then $Q = 0$ implies that there is at least one world state trajectory for which no assignment succeeds, and hence the condition we seek can be written as

$$\min_{\mathbf{X}} p(Q = 0 \mid \mathbf{X})$$

where $\mathbf{X}$ denotes the locations of the assets under consideration. This formulation treats all world state trajectories equally, so it could be the case that rare events could dominate the minimization process. To accommodate this dependence an alternate formulation using S_m can replace that using Q. In this case,

$$\max_{\mathbf{X}} \sum_m p(S_m = 1 \mid \tau_m, \mathbf{X}) p(\tau_m) \tag{2}$$

marginalizes, over the set of world state trajectories, the probability that at least one assignment succeeds in a given world state trajectory weighted by the likelihood of that trajectory.

This formalism does accommodate the uncertainty inherent in this staged approach to planning. For example, the number of adversaries may not be known *a priori*, but each world state trajectory may have different tasking possibilities due to a variable number of adversaries for a fixed number of assets. The procedure remains the same, to maximize the likelihood that at least one task configuration for a fixed set of assets succeeds over all world state trajectories.

3.2 Subproblem 1 Fitness: Costs

Thus far, only the performance of a set of assets has been considered for the first phase mission planning problem. The role of resource costs is an essential piece leading to a description of a fitness function for asset distribution. Costs do depend on the distribution of resources across part-tasks. They also depend on the relations between adversaries and on the timings of adversary actions. In some domains costs are measured in the same units as gains. When this is the case a simple ratio of gain to loss is a typical quantity to be maximized to optimize some process. In the case of mission planning, there may be multiple costs to consider. For example, a ratio of successful engagements to unsuccessful engagements fits the gain to loss ratio model. Factoring in actual costs of some notion of the resources (energy) expended is also desirable. The problem here is that the units of cost are irreconcilable. How many units of energy are worth one less unsuccessful engagement?

The simplest representation of resource cost is as an inequality constraint. This way there is no need to try to reconcile different cost units. However, evolutionary computation

techniques must be adapted in problem-specific ways to handle constrained optimization problems. Techniques for guaranteeing that only feasible solutions can be obtained by constructing specialized operators or data structures. Infeasible solutions can be eliminated as they arise, but this approach is only effective if a small fraction of the search space is infeasible. Repair methods essentially transform infeasible solutions into the feasible part of the search space. Probably the most popular approach makes use of penalty or barrier functions [26]. A number of authors have surveyed different variants of penalty functions for optimization by evolutionary computational techniques [9, 17, 26, 36]. In choosing a penalty function for the purposes here, it is noted that the penalties of interest are static in the sense that they do not depend on the iterations of the evolutionary process. Only inequality constraints with no equality constraints are assumed here, which simplifies the representation of the fitness function. A main issue with static penalty functions is that there may be several parameters that must be set, and the performance of the algorithm can be very sensitive to those set parameter values [25]. Furthermore, there still seems to be some dispute, for static penalties, whether those techniques based on number of constraint violations perform better than penalties based on the magnitude of violations. Based on a statistical analysis of penalty functions [17], a simple static penalty function that is based on the number of constraint variables [16] has been selected. The fitness function for deployment of assets which leads to agile tasking thus becomes

$$F(\mathbf{X}) = f(\mathbf{X}) + \pi(\mathbf{X}) \tag{3}$$

where $f(\mathbf{X})$ is defined in Equation 2 and the penalty term is defined as

$$\pi(\mathbf{X}) = \begin{cases} K(1 - \frac{s}{p}) - f(\mathbf{X}) & s \neq p \\ 0 & otherwise \end{cases} \tag{4}$$

with p equal to the number of constraints and s is the number of these which have been satisfied. The constant K only has the requirement that it be large enough that any non-feasible individual received a lower fitness than any feasible individual. Kuri-Morales [17] used a value of 10^9 in their analysis. It can be seen that if no constraints are satisfied, $s = 0$, then the penalty is a maximum.

All the machinery for finding the deployments leading to maximally agile tasking has now been defined. Specific probabilities of success and corresponding constraint functions are problem-specific.

3.3 Subproblem 2 Fitness

Given a set of committed assets and their deployment distribution, the second subproblem is to find the optimal tasking given data on the current situation in which those assets are embedded. The agility in tasking obtained by an optimal deployment of assets is revealed in the ability to respond to an unexpected situation.

$$f(\mathbf{x}, \mathbf{y}) = \max_j p\left(\mathbf{s} = \mathbf{1} \mid \mathbf{T}_j, x_1, x_2, x_3, y_1, y_2, y_3, \tau_{obs}\right) \tag{5}$$

where $\mathbf{T}_j$ is a vector representing a task configuration (a column in the matrix in Equation 1). The quantity τ_{obs} is an observed trajectory, which in some sense defines the particular situation in which the set of assets find themselves. This second subproblem is amenable to the same decentralized distributed co-evolutionary techniques used to find the optimal deployment.

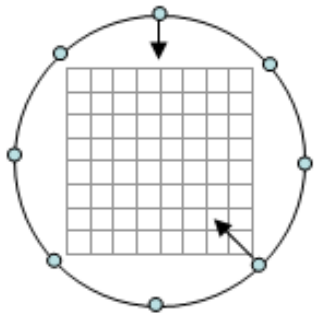
4 An Adversary Exclusion Problem

4.1 Problem Statement

There are many circumstances in which a mission revolves around preventing the deployment of an adversary's assets. Ballistic/tactical missile defense, force protection, blockades, checkpoints for controlled area access and air defense applications all fit this paradigm. In these sorts of problems, the staged approach to mission planning shown in Figure 1 is appropriate. In these sorts of problems, uncertainty around the expected behavior of the adversary can be substantial. An idealized problem is offered here to illustrate cooperative co-evolutionary computation approach to mission planning in order to maximize task assignment agility. This idealization is also a small problem so it does not treat scalability issues for which, it has been asserted, the technique is indeed quite suited, but a small-scale problem is easier to visualize and interpret. So performance considerations are sacrificed for insight into the workings of the decentralized co-evolutionary computation technique to planning for agility in task assignment configurations of a team of assets.

In this idealization, the world is two-dimensional. An adversary has the ability to launch an assault utilizing at most 3 incursions from any combination of 8 locations originating from the cardinal points on a circle. Each incursion crosses the interior of the circle and ends at any non-adjacent cardinal point from its starting point. The choice of starting point configurations for three incursions is uniformly distributed across all possible combinations (there is no time-ordering in this problem).

Each defensive asset can perform self-locate, search, and intercept actions. Figure 3a illustrates the set up. Asset locations can be changed in discrete steps on an 8×8 planar grid wholly contained within the circle. A single search sector constitutes any pair of cells in the 8-neighborhood of an asset's location (thus, there are twenty-seven possible search sector positions that can be selected). The intercept action places a barrier at the location at which a threat was detected. Each asset can place at most two barriers. For a problem given three assets, the trio must cooperate in order to contain the maximum number of possible incursions.



a

X_1	Y_1	S_{11}	S_{12}	B_{11}	B_{12}
X_2	Y_2	S_{21}	S_{22}	B_{12}	B_{22}
X_3	Y_3	S_{31}	S_{32}	B_{31}	B_{32}

b

Figure 3: The set up for the simplified problem is shown in (a). A sequence of discrete variables defines the genotype for the set of assets is shown in (b).

4.2 Genotypes, Species, and Fitness

The genetic structure for the pair of assets is shown in Figure 3b and takes the form of a matrix (because the set of assets is homogeneous). This matrix corresponds to the position, search sector selection and intercept assignment actions that can be taken by the set of assets. Each row is the genotype of a given asset (species). Within the coevolutionary process asset-1 will search over the first row of the structure for fixed values of the second row. Asset-2 searches the second row and asset-3 the third. The fitness evaluation for each, however, encompasses all rows of the matrix.

The gains in fitness are calculated by replacing the probabilities in Equation 2 by simple counts. The denominator in that equation plays no role in this problem simplifying the assessment of fitness even further. Some costs have already been captured by limiting the genotype. Additional costs are upper and lower distances from an asset at which a barrier can be placed. A requirement that search sector not overlap is another cost that can be put in a form of Equation 4.

4.3 Results

The deployment and tasking subproblems are separated as per the earlier discussion. The primary item of interest here is in preserving the maximum number of choices for tasking when there is great uncertainty predicting an adversary's tactics. The cooperative co-evolutionary genetic algorithm is executed on three networked laptop computers corresponding to the three assets in the problem statement. Local evolution proceeds in parallel, interspersed with periodic cross-asset updates as conditional optima are achieved.

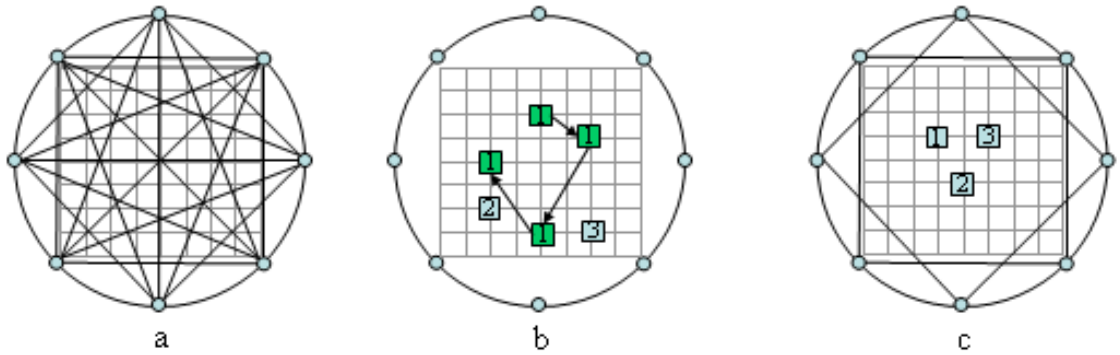


Figure 4: The number of possible incursion trajectories is shown in (a). The conditional optimization from the perspective of asset-1 samples positions for asset-1 with assets-2 and 3 fixed is illustrated in (b). In (c), is a final spatial distribution of the three assets. Fifteen out of twenty-three trajectories can be intercepted by several available task assignment configurations.

Without the costs given above, detection of incursions drives the deployment so that assets are forced to the center to insure that search sectors can be set so that all possible incursions can be detected. Unique search sectors force the assets apart so that there is minimal cost for coverage of possible incursion trajectories near the center of the circle. This is the set covering problem. If there are range bounds for the placement of a barrier, then the size of the asset deployment increases again. However, the symmetry of the problem forces symmetry in the deployment of assets, which is reflected in the outcome of the co-evolutionary process as shown in Figure 4.

5 Summary

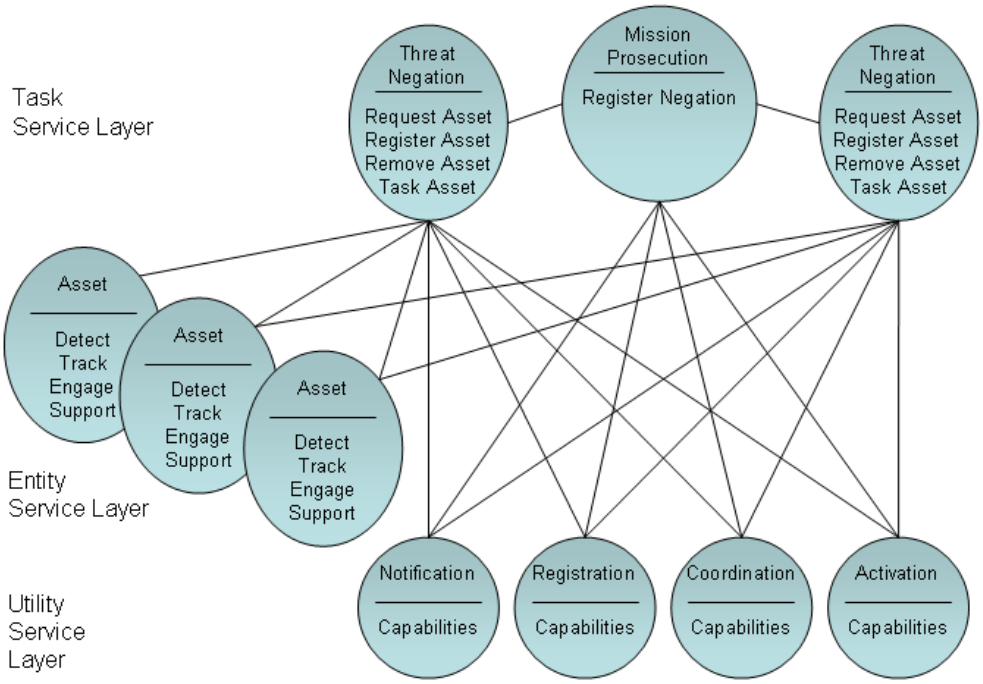
An evolutionary computation approach to the problem of planning for agility in the task assignment configurations of large heterogeneous teams has been derived. A key point in achieving self-synchronization as well as tractability for large-scale mission planning problems is obtained from the decentralized distributed co-evolutionary computing architecture that is employed. In particular, cooperative co-evolutionary computing algorithms are used to realize a decentralized distributed approach to planning. Task assignment agility for large teams is obtained through appropriately defined fitness functions for each of two separated subproblems:

1. Asset selection and deployment distribution

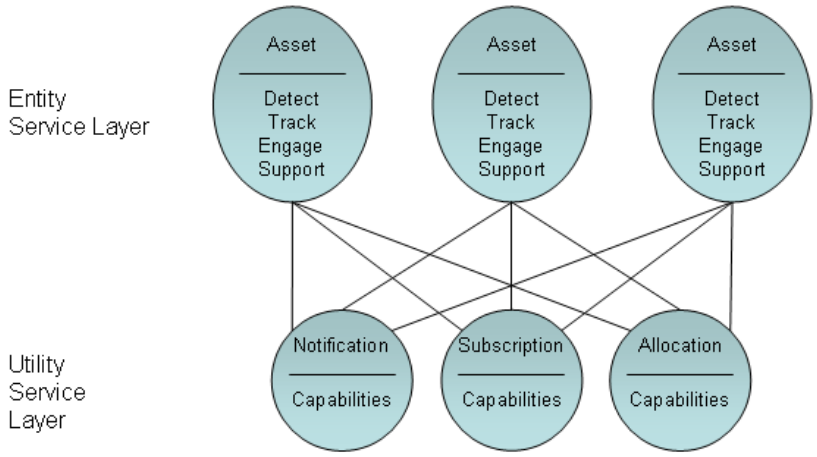
2. Task assignment

The fitness function for the first subproblem seeks to maximize the number of successful tasking choices available for the task assignment part of the problem. The benefit of this approach increases with increasing team size and under high uncertainty in scenario specifics. For more certain situations, results from this approach converges to more conventional mission planning techniques.

A final point can be made as to the compatibility of the cooperative co-evolutionary genetic algorithm with emerging netcentric computing architectures. In particular, services-oriented architectures (SOA) and event-driven architectures (EDA) are becoming established as *de facto* standard paradigms for networked organizations [13]. The difference between these two approaches to netcentric computing lies in the underlying communication pattern each employs [35]. SOA rely on a query-response pattern underlying communication between services in the network. This pattern, while very efficient, nonetheless requires two-way communication traffic to execute a transaction, which for a densely connected network may lead to bottlenecks. The publish-subscribe pattern in the EDA, on the other hand, just broadcasts information based on local subscriptions, so network traffic is less than the query-response pattern of communications. For mission planning using the decentralized distributed co-evolutionary genetic algorithm, communication between all assets is essential, therefore, the EDA's publish-subscribe is a better fit. The corresponding architecture is also simpler in the EDA paradigm. Figure 5 shows a pair of high level service models that compares, at least qualitatively, SOA and EDA architectures supporting cooperative co-volutionary planning algorithms.



Co-evolving ensemble configuration and tasking can be centrally coordinated in an SOA



Co-evolving ensemble configuration and tasking better fits an EDA service model

Figure 5: Mission planning with decentralized distributed co-evolutionary algorithms is compatible with netcentric computing architectures, but seems a better fit to the event-driven paradigm.

References

[1] D.S. Alberts, J.J. Garstka, and F.P. Stein. *Network Centric Warfare*. DoD Command & Control Research Program (CCRP), www.dodccrp.org, 2nd edition, 1999.

[2] D.S. Alberts and R.E. Hayes. *Planning: Complex Endeavors*. DoD Command & Control Research Program (CCRP), www.dodccrp.org, 2007.

[3] D. Ashlock. *Evolutionary Computing for Modeling and Optimization*. Springer, New York, 2006.

[4] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, 1994.

[5] J.S. De Bonet, C.L. Isbell, and P. Viola. MIMIC: Finding optima by estimating probability distributions. In *Advances in Neural Information Processing Systems (NIPS-96)*, pages 424–431, 1996.

[6] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[7] I. Cardei. Energy efficient target coverage in heterogeneous wireless sensor networks. In *Proceedings of the Third International Conference on Mobile ad hoc and Sensor Systems*, pages 397–406, 2006.

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, New York, 2nd edition, 2001.

[9] W.A. Crossley and E.A. Williams. A study of adaptive penalty functions for constrained genetic algorithm-based optimization. In *Thirty-Fifth Aerospace Sciences Meeting and Exhibit*. AIAA, 1997.

[10] F. Dambreville. Optimal area covering by sensors for planning a tracks collection. In *Proceedings of the Ninth International Conference on Information Fusion*, pages 1–7, 2006.

[11] M.E. desJardins, E.H. Durfee, C.L. Ortiz Jr., and M.J. Wolverton. A survey of research in distributed, continual planning. *AI Magazine*, Winter:13–22, 1999.

[12] M. deWeerdt and C. Witteveen. Planning in multi-agent systems. In *Proceedings of the Ninth European Agent Systems Summer School*, 2007.

[13] T. Erl. *Principles of Service Design*. Prentice Hall, Upper Saddle River, 2008.

[14] D.B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Wiley, New York, 3rd edition, 2005.

[15] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, San Francisco, 2004.

[16] A. Kuri. A universal eclectic genetic algorithm for constrained optimization. In *Proceedings of the Sixth European Congress on Intelligent Techniques and Soft Computing*, pages 518–522, 1998.

[17] A. Kuri-Morales and J. Gutiérrez-García. Penalty function methods for constrained optimization with genetic algorithms: A statistical analysis. In *MICAI 2002: Advances in Artificial Intelligence: Second Mexican International Conference Artificial Intelligence*, volume 2313/2002 of *Lecture Notes in Computer Science*. Springer, Berlin, 2004.

[18] G.M. Levchuk, Y.N. Levchuk, J. Luo, K.R. Pattapati, and D.L. Kleinman. Normative design of organizations - part I: Mission planning. *IEEE Transactions on Systems, Man and Cyternetics*, 32:346–359, 2002.

[19] R.H.M. Macfadzean. *Surface-Based Air Defense System Analysis*. R.H.M Macfadzean, macfac@sirinet.net, 1992.

[20] A.D. Mali and S. Kambhampati. Distributed planning. http://rakaposhi.eas.asu.edu/papers.html.

[21] Mausam, P. Bertoli, and D.S. Weld. A hybridized planner for stochastic domains. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 1972–1978, 2007.

[22] Drew V. McDermott. Reasoning about autonomous processes in an estimated-regression planner. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 2003), June 9-13, 2003, Trento, Italy*, pages 143–152. AAAI, 2003.

[23] S. Meguierdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, pages 1380–1387. IEEE Press, 2001.

[24] S. Meguierdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in wireless ad-hoc sensor networks. In *Proceedings of the Seventh International Conference on Mobile Computing and Networking*, pages 139–150. ACM Press, 2001.

[25] Z. Michalewicz. Genetic algorithms, numerical optimizaton and constraints. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 151–158. Morgan Kaufmann, 1995.

[26] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, 3rd edition, 1996.

[27] J. Ortega, J.L. Bernier, A.F. Díaz, I. Rojas, M. Salmerón, and A. Prieto. Parellel combinatorial optimization with evolutionary cooperation between processors. In *Proceedings of the Congress on Evolutionary Computation*, pages 1051–1058. IEEE Press, 1999.

[28] J. Pearl. *Causality*. Cambridge University Press, Cambridge, 2002.

[29] M. Pelikan. *Hierarchical Bayesian Optimization Algorithms*. Springer-Verlag, Berlin, 2005.

[30] M. Pelikan, D.E. Goldberg, and F.G. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[31] K.L. Pollock. Plans and decisions. *Theory and Decision*, 57:79–107, 2005.

[32] M.A. Potter and K.A. DeJong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.

[33] S. Sanner and C. Boutilier. Approximate solution techniques for factored first-order MDPs. In *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling (ICAPS 2007), September 22-26, 2007, Providence, Rhode Island, USA*. AAAI, 2007.

[34] R. Subbu and A.C. Sanderson. *Network-Based Distributed Planning Using Coevolutionary Algorithms*. World Scientific, River Edge, 2004.

[35] J. van Hoof. SOA and EDA: Using events to bridge decoupled service boundaries. *SOA Magazine*, IV, 2007. http://www.soamag.com/I4/0207-2.asp.

[36] O. Yeniay. Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Applications*, 10(1):45–56, 2005.

[37] S. Yoon, A. Fern, and R. Givan. FF-Replan: A baseline for probabilistic planning. In *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling (ICAPS 2007), September 22-26, 2007, Providence, Rhode Island, USA*. AAAI, 2007.