Resource Allocation Optimization in Alien Migration Interdiction Operations Using an Evolutionary Algorithm Technique

Dr. Neal Wagner, Assistant Professor of Information Systems[1], Dr. Joe DiRenzo III, Chief, Operations Analysis Division[2], and LCDR Ben Maule, Operations Analyst[2]

[1] Fayetteville State University, Fayetteville, NC USA
[2] US Coast Guard Atlantic Area Command, Portsmouth, VA USA

Summary. Alien Migration Interdiction Operations (AMIO) refers to the US Coast Guard mission to interdict undocumented migrants traveling by sea prior to landfall in the US. Resource allocation in AMIO refers to the scheduling of Coast Guard resources to patrol common sea routes used by migrant smuggling vessels. The AMIO resource allocation optimization problem involves multiple objectives and multiple constraints including resource limitations, minimization of operating and transportation costs, and maximization of functional capabilities of scheduled resources. The problem is a form of combinatorial optimization and is known to be NP-hard. Such problems are characterized by solution search spaces that grow exponentially with problem size and are often intractable when classical, deterministic algorithms are applied to real-world problem instances.

Evolutionary Algorithms (EA) is a class of computational intelligence techniques that use the biological process of evolution as a model for designing computer programs that evolve solutions to a given problem. EA techniques are non-deterministic and have been shown to be effective for many real-world problem domains which are characterized by large solution spaces and non-linear dynamics.

This paper proposes a hybrid EA-heuristic model for AMIO resource allocation optimization that seeks to combine domain-specific heuristics with the evolutionary search process. The goal is to build a robust optimization model that leverages the strengths of evolution-based search with the domain expertise of US Coast Guard Operations.

Keywords: Resource Allocation; Optimization; Evolutionary Algorithm; Modeling and Simulation; Operations Research.

1 Introduction

Protecting national borders has become an increasingly important concern in the US in recent years. The threat of terrorism and the abuses of human trafficking, among others, are significant factors that spur efforts to improve the efficiency and effectiveness of US border protection operations. The US Coast Guard is a unique branch of the US Armed Forces in that it is charged with maritime law enforcement [11]. One of the Coast Guard's primary missions is Alien Migration Interdiction Operations (AMIO), which is concerned with the prevention and interception of undocumented migrants traveling by sea prior to landfall in the US [1]. The mission includes prevention of migrant smuggling attempts, detection of migrant smuggling vessels, and apprehension, detention, and repatriation of undocumented migrants.

Of paramount importance to the AMIO mission is the optimal allocation of Coast Guard resources to accomplish multiple objectives and subject to multiple constraints. The gravity of the resource allocation problem stems from the fact that sub-optimal allocations result not only in larger breaches of national borders but also in significantly increased risk for loss of migrant life. Among the aimed-for objectives are the minimization of costs due to operation and transportation of resources and the maximization of functional capabilities that are available in various Coast Guard sea-going and air-going vessels. Several allocation scheduling constraints exist pertaining to the availability of various resources and including various domain-specific logical rules that are part Coast Guard AMIO mission protocol. The AMIO resource allocation problem is a version of the assignment problem from the field of combinatorial optimization and is known to be NP-hard [3]. NP-hard problems exhibit solution search spaces that grow exponentially with problem size. Such problems are often intractable when classic deterministic algorithms are applied to real-world problem instances [4].

Evolutionary Algorithms (EA) is a class of techniques from the field of computational intelligence that use the biological process of evolution and natural selection as a model for designing a computer program that evolves solutions to a given problem [10]. EA is a fitness-based stochastic search process and is widely known for its effectiveness in generating optimal or near-optimal solutions to NP-hard problems in general and to combinatorial optimization problems in particular [7]. This paper proposes a hybrid EA-heuristic model for the AMIO resource allocation problem that seeks to combine the effectiveness of evolution-based search with domain-specific heuristic techniques. The idea is to leverage the domain expertise and current best practices of US Coast Guard Operations along with the strengths of the evolutionary search process to build a robust model that can provide optimal resource allocations to real-world problem instances.

The rest of this paper is organized as follows. Section 2 provides a description of the AMIO resource allocation problem, section 3 details the proposed EA-heuristic optimization model, and section 4 discusses future work necessary transfer this technology to a research prototype and, ultimately, to an enterprise-grade decision support software system.

2 AMIO RESOURCE ALLOCATION PROBLEM

The AMIO mission is to prevent and interdict undocumented migrants traveling by sea before they achieve landfall in the US. This mission has many facets and includes measures aimed at prevention/reduction of migrant smuggling attempts, surveillance and detection of migrant smuggling vessels, apprehension of smuggling vessels, and migrant holding and servicing during repatriation procedures [1]. Inherent in all of these mission activities is the effort to protect and safeguard human life [11, pp. 75-85]. Undocumented sea-going migrant vessels are often extremely unsafe due to severe overcrowding and disrepair. Further compounding the problem are the facts that migrants oftentimes have no protective accessories such as lifejackets and may not be able to swim.

Resource allocation refers to the scheduling of Coast Guard resources to patrol common sea routes used by migrant smuggling vessels. Like many businesses and organizations, the US Coast Guard (USCG) is concerned with optimal resource allocation in order to maximize efficiency and mission effectiveness and minimize costs. Unique to the USCG's and other police and security organizations' missions is the element of risk to human life. Therefore, it is critical to the AMIO mission that resource allocation be optimized as improper (sub-optimal) allocation can have disastrous consequences.

Figure 1 displays a map containing example areas that USCG is concerned with. In the figure a portion of the Caribbean is shown including Cuba, Haiti, The Dominican Republic, and other Caribbean islands along with southern Florida. The yellow shaded areas represent example operational areas that USCG must monitor and patrol while the dark blue shaded dots represent example bases at which USCG resources are located.[1] Typically, USCG divides geographical areas into various sections each with its own resources and base locations and its own operational areas for which it is responsible.

Each section has a number of sea-going and air-going vessels available for mission activities and each type of vessel has varying functional capabilities and operating constraints. Figure 2 displays examples of USCG sea-going and air-going vessels: subfigure (a) depicts a 270 foot USCG "Cutter" ship, subfigure (b) depicts a fixed wing USCG "Super Hercules" aircraft, and subfigure (c) depicts a rotary wing (i.e. helicopter) USCG "Dolphin" aircraft.[2]

Resource functional capabilities include speed, migrant holding and servicing capacity, surveillance ability, and deterrence ability. Migrant holding and servicing capacity refers to the number of migrants that can be interdicted by a resource during a single event. This functional capability takes into account space and facilities available on the resource, the number of crew members that it has, and the time it takes to detain and repatriate interdicted

[1] The map shown in Figure 1 is only used for illustrative purposes and is not meant to accurately or exhaustively depict actual USCG operational areas and resource bases.

[2] For more information on USCG sea-going and air-going vessels, please see http://www.uscg.mil/datasheet/ (2012).

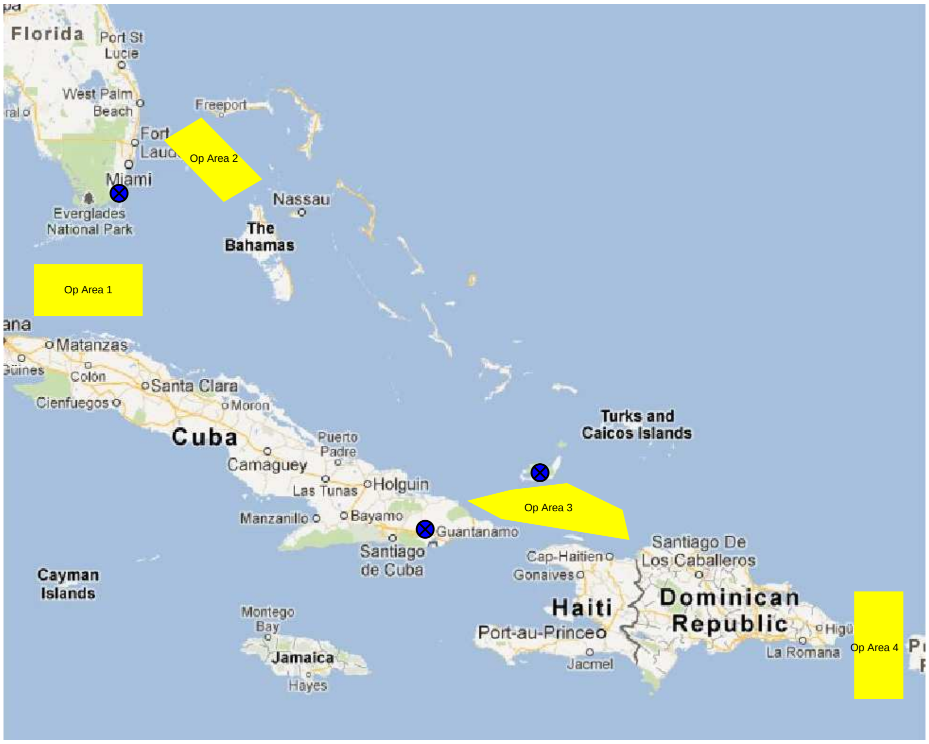

Fig. 1. Example operational map

migrants. Surveillance refers to a resource's monitoring ability and takes into account accuracy and range of detection as well as resource speed. Deterrence refers to the perceived USCG presence generated by the resource when deployed from the perspective of potential migrants. This functional capability seeks to reduce interdiction events by projecting a vigilant USCG presence and, thereby deterring potential migrants from attempting migration.

It is important to note that while speed and migrant holding and servicing capacity are functional capabilities that are easily quantified, surveillance ability and deterrence ability are not. Thus, it is necessary for planners to use their judgment and domain knowledge to assess the surveillance and deterrence abilities of a resource before including it in a resource allocation plan.

Resource operating constraints include the maximum number of days a sea-going vessel can operate before it must return to its base and the maximum number of flights an air-going vessel can make per day and its maximum range of flight from its base.

In addition to functional capabilities and resource operating constraints, there are a number of domain-specific logical constraints that must be considered. For example, air-going resources may have high surveillance abilities

(a) USCG Cutter ship

(b) USCG aircraft (fixed wing)

(c) USCG aircraft (rotary)

Fig. 2. USCG sea-going and air-going vessels

due to their speed but low migrant holding and servicing abilities. Conversely, sea-going resources such as large ships may have high migrant holding and servicing abilities but low surveillance abilities due to their slow speed. Thus, it is necessary for planners to allocate resources in such a way as to balance the varying functional capability levels so that operational areas are effectively monitored and patrolled. It would be ineffective for an area to be patrolled by, for example, only fast air-going vessels with high surveillance capabilities as there would not be sufficient migrant holding capabilities to successfully handle interdiction events.

And there is also the issue of deterrence. Some operational areas have particularly high rates of migrant smuggling and, for such areas resources with high deterrence abilities are utilized to prevent or reduce the number of migration attempts. From an operations standpoint it may be more cost effective to deploy resources with higher deterrence abilities (and lower other abilities) for such areas than to deploy other types of resources as it is cheaper to dissuade potential migrants from attempting migration than it is to interdict them during an attempt.

Other inputs to the AMIO resource allocation problem include migrant flow predictions per operational area per time period, individual resource operating and transportation costs, the size of each operational area, and the distance from each operational area to each base location. A migrant flow prediction is a prediction of the total number of migrants that will attempt migration through a particular operational area during a specified time period. Operating and transportation costs are the monetary costs per hour to deploy a resource to an operational area and to transport a resource to and from the operational area.

Figure 3 gives a logical diagram of the AMIO resource allocation problem. In the figure problem inputs are shown to the left and the problem output (solution) is shown to the right. As shown in the figure the solution is a resource to operational area mapping for a specific time period. As mentioned in the previous section, the AMIO problem is a form of combinatorial optimization and is classified as NP-hard [3]. The solution search space for such problems grows exponentially with problem size and often causes real-world problem instances to be intractable when classical deterministic algorithms are applied [4]. The AMIO problem is also highly constrained and may contain both linear and non-linear constraints. Such constraints complicate the solution search process by rendering areas of the search space infeasible[3] and, in many cases, more difficult to explore [8]. This combination of factors makes the problem quite challenging for USCG operations planners to solve optimally.

3 EA-HEURISTIC OPTIMIZATION MODEL

This section describes the proposed evolutionary optimization model for the AMIO resource allocation problem. It is important to note that the model detailed below represents an initial conceptual model designed before any preliminary implementation or testing has taken place, and thus cannot be considered as either exhaustive or complete. The following sections provide a brief overview of the EA paradigm and its benefits, discuss AMIO solution representation, and detail important model features.

[3] Problem solutions located in infeasible search space areas are solutions that violate one or more problem constraints.

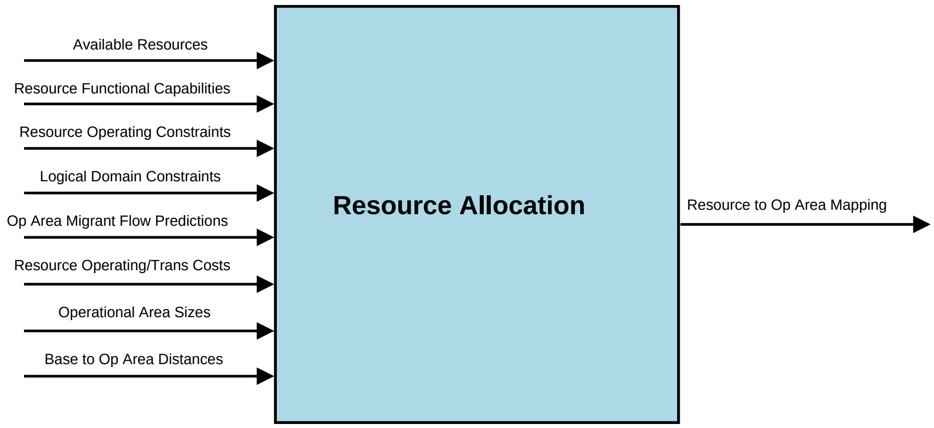

Fig. 3. AMIO resource allocation logical diagram

3.1 The EA Paradigm

EA is a biologically-inspired computational model that uses the Darwinian theory of evolution and natural selection as a method of designing computer programs that automatically evolve solutions to a given problem. The benefits of such an approach are the following.

1. In many applications EA and other evolution-based techniques have been able to find solutions that are better than or equal to the best solutions found by humans [6].
2. EA can find solutions to a problem without having to be explicitly told what the optimal solution looks like or how to go about finding it [5].
3. EA has been shown to provide optimal or near-optimal solutions for NP-hard class optimization problems with large problem instances such as the AMIO problem where classical deterministic algorithms often fail [7].

In addition to the above benefits, there are also the concerns of time and labor. Many real-world optimization problems require team(s) of people significant time to analyze and solve. When an unexpected event occurs that alters problem parameters, for example a resource malfunction, the problem must be revisited as the previously found solution may no longer be optimal. In an environment with quickly changing conditions, there may not be enough time for human analysts to re-analyze and solve the problem optimally. Computer systems based on EA have the potential to solve and re-solve problem instances as changing conditions dictate with quicker solution turnaround and significantly reduced labor. The general EA paradigm includes the following steps.[4]

[4] For more information on the EA paradigm please see [10].

1. A population of candidate solutions is created.
2. Each solution is measured and ranked based on its quality, i.e. its proximity to the optimal.
3. Higher ranking solutions are selected to produce offspring solutions.
4. Offspring are generated by genetic operators such as mutation and crossover. Mutation usually involves one parent solution while crossover usually involves two.
5. A new population of candidate solutions with sufficient population size is produced by iterative application of steps 3 and 4.
6. Steps 2 through 5 are applied iteratively to create successive populations of candidate solutions until some run termination condition has been reached. Upon termination the best solution of the run is designated the result. Example termination conditions include discovery of the optimal solution, maximum runtime being reached, or multiple successive populations having no improvement of solution quality.

3.2 AMIO Solution Representation

As mentioned in the previous section, the proposed model seeks to produce a resource to operational area mapping that minimizes costs due to operation and transportation of resources and maximizes resource functional capabilities subject to multiple constraints. Consider the following example AMIO problem instance. Suppose the four operational areas shown in figure 1 are to be covered for a given time period and there exist 50 sea-going and air-going resources each with its own constraints, costs, and functional capabilities spread across the three base locations shown in figure 1. An example solution for such an instance is given in figure 4.

In the figure operational areas are represented by the acronym OAi where $i = 1, \ldots, 4$ and correspond to operational areas 1 through 4 from figure 1, respectively, and resources are represented by the acronym Rj where $j = 1, \ldots, 50$ and correspond to the 50 sea-going and air-going resources of the example problem instance.[5] Allocated sea-going resources in the example solution mapping refer to boats/ships that are to be deployed continuously in an operational area from a group of sea-going resources. For example, for operational area 1 (OA1 in the figure) the mapping specifies 2 sea-going resources from a group of 3 sea-going resources including resources numbered 22 through 24. This means that 2 of these resources are to be deployed continuously in the corresponding operational area with one extra resource waiting and being maintained at its base location to supply relief when needed.

Allocated air-going resources in the example mapping refer to the number of flights that should be made to patrol the operational area per day during the given time period. For example, for operational area 2 (OA2 in the

[5] Please note that the numbers used to represent operational areas and available resources are arbitrary and are presented here for illustrative purposes only.

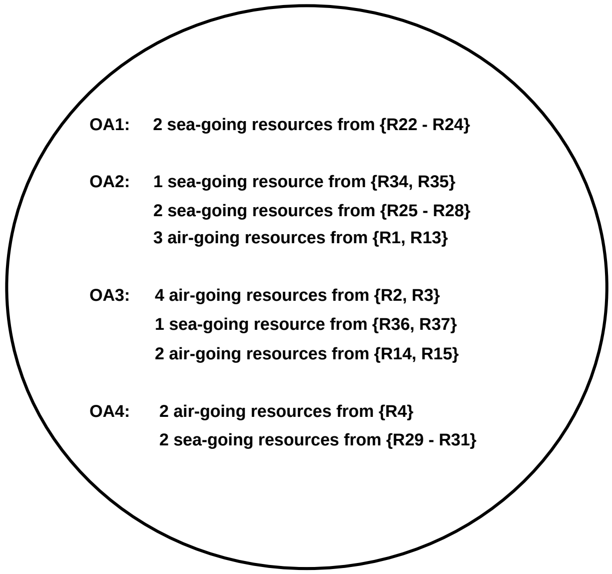

Fig. 4. Example resource to operational area mapping solution

figure) the mapping specifies 3 daily flights to be made from a group of 2 air-going resources including resources numbered 1 and 13. The following section discusses candidate solution creation and evaluation for the proposed model.

3.3 Solution Creation and Evaluation

For the proposed evolution-based optimization model, initially a population of candidate solutions of the format shown in figure 4 is created via a combination of fully-random and partially-random assignment procedures. Partially-random assignment refers to the creation of an initial candidate solution by randomly mapping available resources to operational areas in such a way as to not violate any operational or domain-specific constraints. Fully-random assignment refers to creation of a candidate solution in the same way as for partially-random assignment except that constraints are not considered. Thus, a portion of the initial population produced contains only feasible solutions (i.e. solutions that do not violate any constraint) and a portion of the population may contain a mix of both feasible and infeasible solutions (i.e. solutions that violate one or more constraints).

Intuitively it may appear that allowing infeasible candidate solutions is harmful to the search for the optimal. However, it has been noted that for highly-constrained problems the optimal often resides at the boundary between a feasible and infeasible region of the search space [12]. Consequently it has been shown that including infeasible solutions can be helpful to the search because they push solution exploration toward these boundaries [13].

Once the initial population is generated, the evolutionary process requires that solutions be measured and ranked according to their quality. This entails the specification of a mathematical expression, called a fitness function, which will be used to accomplish this. The function must handle the dual objectives of minimization of operational and transportation costs and the maximization of deployed resource functional capabilities. The operational and transportation costs incurred when a particular resource is deployed to a particular operational area are straightforward calculations and easily included in the fitness function. The migrant holding and servicing capability of a resource is also readily quantifiable, and thus is easily included as well. As discussed in the previous section, resource functional capabilities concerning surveillance and deterrence are not as readily quantified. Therefore some method must be constructed to quantify these attributes so they may be included in the fitness calculation.

Here, the proposed model takes into account the domain knowledge of USCG operations to quantify these resource capabilities in the following way. Operations personnel assign an ability level (e.g. High, Medium, or Low) to a resource pertaining to the capability being measured. These levels are initially assigned numeric values by operations personnel and later these level-quantifying values are tuned during preliminary experiments on several historical problem instances examples.

Deterrence ability, as detailed in the previous section, is concerned with reducing the number of smuggling attempts. With this in mind, the fitness function includes a component that seeks to mimic this dynamic. As shown in figure 3, one of the inputs to the AMIO problem are predictions of migrant flow through operational areas for a given time period. When a resource rated as having a high level of deterrence ability is deployed to an operational area, the migrant flow prediction for that area is reduced by an amount that represents a percentage of the whole prediction. This reduction percentage is specified as a constant parameter of the fitness function. When a resource is rated as having low deterrence ability, the corresponding migrant flow prediction is increased by a similar percentage amount (also specified as a parameter to the function). Predictions for resources with medium deterrence ability remain unchanged. In this way the behavior of potential migrants in reaction to resources with varying deterrence abilities is captured.

Surveillance ability, as detailed in the previous section, is concerned with detection of smuggling attempts. Thus, the evaluation of candidate solutions also includes a surveillance component that seeks to capture this effect. This component is executed in conjunction with another evaluation component to be described below.

Migrant flow through an operational area does not occur at a uniform rate but rather in individual bursts of smuggling attempts termed “events” by USCG operations personnel. Smuggling events can take place at random times during a given time period and with random numbers of migrants involved. In order to evaluate the quality of a candidate mapping solution it is necessary to simulate such events and calculate the number of events/migrants detected and interdicted as well as the number of migrants that escaped detection. The surveillance component mentioned above interacts with this component in a similar way as described above for the deterrence component, that is by reducing or increasing the number of migrants detected depending on deployed resources' surveillance capability levels. For deployed resources with high levels of surveillance ability, the number of smuggling event detections is increased by an amount that represents a percentage of the number of smuggling events that would have been detected with normal (medium level) surveillance ability. Similarly, the number of event detections is decreased by a percentage amount for deployed resources with low surveillance ability. For deployed resources with medium levels of surveillance ability, the number of event detections remains unchanged.

It is important to note that the number of migrants interdicted in simulated events is also affected by the migrant holding and servicing capacity of deployed resources in an operational area. For example an event may be detected by a deployed resource but if there is not sufficient migrant holding and servicing capacity deployed to the operational area, then interdiction is not achieved.

Another significant concern is the varying types of constraints involved in the AMIO problem. Some problem constraints are considered "hard". Hard constraints are those constraints that if violated render a solution completely unviable. Other constraints are "soft", that is they may be violated by a solution without completely invalidating the solution. In order to capture such kinds of constraints, the proposed model includes both hard and soft constraint components in the fitness function. Equation 1 gives the full fitness function of the proposed model.

$$fit(s) = \sum_{i=1}^{n} \sum_{j=1}^{OpA} \begin{pmatrix} -1 * (esc(s,i,j) + sc(s,i,j) + (hcp * hc(s,i,j))) + \\ \sum_{k=1}^{Res}(-c(s,i,j,k) + mig_k + dt_k + sv_k) \end{pmatrix} \quad (1)$$

In equation 1 s is the candidate solution being evaluated by function fit; n is the number of simulated event runs; OpA is the number of operational areas being mapped; $esc(s,i,j)$ is a function that calculates the number of escaped migrants (i.e., migrants who achieved US landfall) for solution s, simulated event run i, and operational area j; $sc(s,i,j)$ is a function that calculates the number of soft constraints violated by solution s during simulated event run i and operational area j; hcp is a constant term that represents a penalty weight for violation of a hard constraint; $hc(s,i,j)$ is similar to $sc(s,i,j)$ except that it is concerned with hard constraint violations; Res is the number of available resources; $c(s,i,j,k)$ is a function that calculates the operational and transportation cost of solution s for simulated event run i, operational area j, and resource k; and mig_k, dt_k, and sv_k represent the migrant holding capacity, deterrence level, and surveillance level of resource k, respectively.

The goal of the AMIO allocation problem is to minimize resource costs and maximize resource functionality. The fitness function given by equation 1 includes components designed to model and capture the numerous aspects of both of these objectives. During the evolutionary process candidate solutions are measured and ranked and higher ranking solutions are selected to produce offspring. For the proposed model, higher ranking solutions are given by greater fitness values. The following section details parent solution selection and offspring production for the proposed model.

3.4 Selection and Offspring Production

The EA paradigm specifies that candidate solutions are to be selected for offspring production on a probabilistic basis with higher ranking solutions given higher probability of selection. There exist many algorithms for accomplishing this, but one well known effective and efficient method is tournament selection [2]. In tournament selection a group of solutions are selected randomly from a population and the one with the highest fitness is chosen to produce offspring. The size of the initial selection group depends on the size of the population. Because tournament selection has been vetted in numerous EA applications, it is the selection method proposed here. Additionally, a technique known as *elitism* has been also been used to preserve the quality of the best candidate solution [2]. Elitism refers to the practice of taking the best candidate solution from a previous population and copying it directly into the next population unchanged. This method has also been well vetted in the literature and is included in the model proposed here.

After selection takes place, genetic operators of mutation and crossover are used to transform parent solutions into offspring solutions which are then inserted into a new population of candidate solutions. Mutation usually involves one parent solution while crossover usually involves two. For the AMIO problem it is necessary to construct mutation and crossover operators that can act upon candidate solutions with the solution representation as shown in figure 4. For the proposed model solution mutation is executed on a single solution using the following steps.

1. Randomly select one of three possible mutation operations: removal, replacement, or addition.
2. Randomly select to allow the created offspring to violate constraints or not.
3. Randomly select an operational area from the solution's list of operational areas.
4. If removal or replacement is selected, randomly select a deployed resource from the selected operational area and either remove it or replace it with another non-deployed resource of the same type (i.e. replace air-going resources with air-going resources and sea-going resources with sea-going resources).

5. If addition is selected, randomly select a non-deployed resource to be deployed to the selected operational area.
6. If non-violation of constraints has been selected, check the offspring for violated constraints and discard offspring if any exist.

Crossover is the algorithmic equivalent of sexual reproduction in biology and involves two parent solutions creating offspring solution(s). For the proposed model crossover of two solutions is executed to produce two offspring solutions using the following steps.

1. Randomly select to allow the created offspring to violate constraints or not.
2. Randomly select an operational area from one of the parent solutions and swap its deployed resource set with the deployed resource set of the other parent solution.
3. To correct any duplication of deployed resources in a single offspring, randomly perform removal or replacement mutation on the duplicated resources of the offspring until no more duplicates exist.
4. If non-violation of constraints has been selected, check each produced offspring for violated constraints and discard offspring if any exist.

Figures 5 and 6 provide an example of the crossover genetic operation. In figure 5 the initial steps of crossover are displayed. Here, two parent solutions produce two offspring solutions by swapping resources sets corresponding to operational area 2 (OA2 in the figure). Notice that the swap causes duplicate resources to be present in offspring 1 of the figure (these duplications are highlighted yellow). Figure 6 depicts the duplicate correction step for offspring 1. The three duplicated resources (highlighted yellow in the offspring to the left of the diagram) are corrected by performing point mutations on three resources (highlighted violet in the offspring to the right of the diagram). Notice that in the corrected offspring (to the right of the diagram) resource 22 (R22) previously deployed to operational area 1 (OA1) is replaced by resource 25 (R25), resource 36 (R36) previously deployed to operational area 3 (OA3) is replaced by resource 38 (R38), and resource 2 (R2) previously deployed to operational area 3 (OA3) is removed thereby reducing the number of air-going resources deployed to OA3.

It is important to note that when an offspring is discarded due to constraint violation, it is not inserted into the new population. Instead the selection and transformation algorithm is executed iteratively until a new population of suitable size is generated. Intuitively it may seem that repairing solutions that violate one or more constraints may be a more efficient procedure than discarding them and re-executing the transformation algorithm. However, it has been recognized that reparation of infeasible solutions for optimization problems that contain non-linear constraints can be quite difficult especially when the feasible regions of the search are non-convex [9]. Because the AMIO problem may exhibit such characteristics for real-life problem instances, reparation of infeasible solutions is not prescribed in the proposed model.

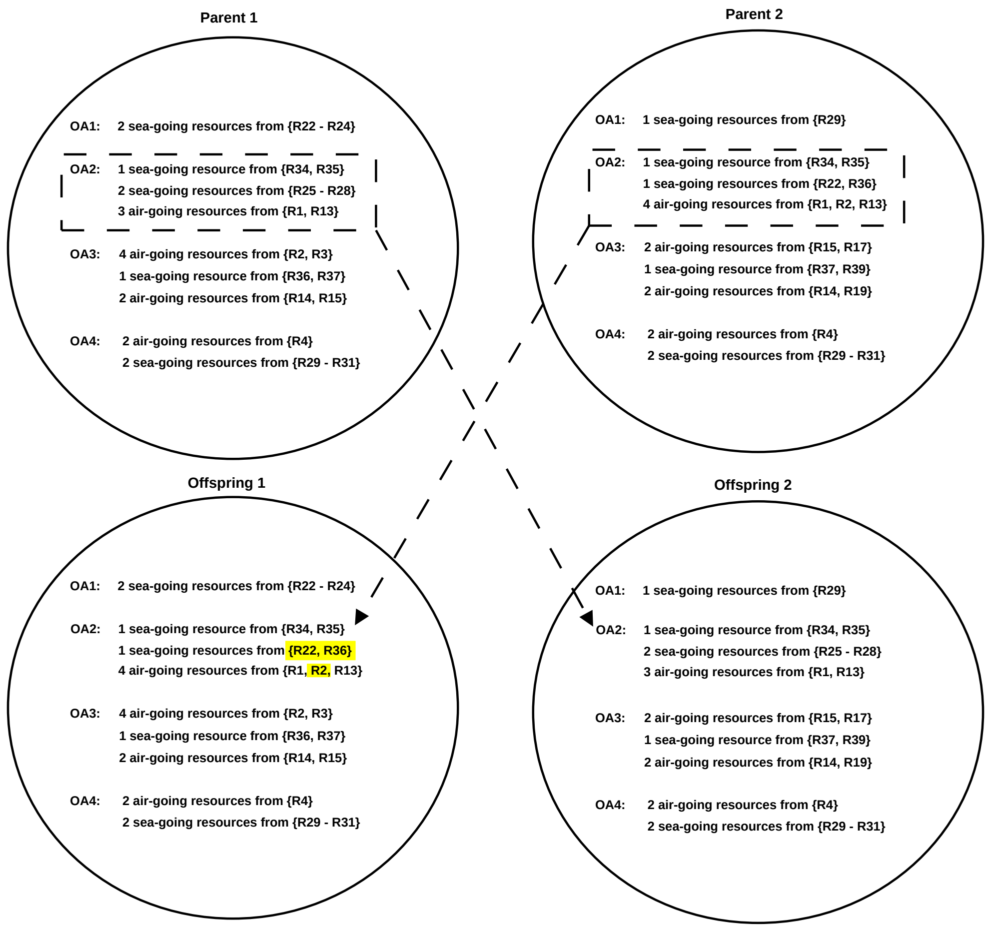

Fig. 5. Crossover genetic operator - initial steps

As discussed in section 3.1, the evolutionary search process involves an iterative execution of evaluation, selection, and transformation of candidate solutions until some termination condition is satisfied. For the proposed model, termination occurs when a maximum computation time has been reached or if the best evolved candidate solution has not been improved for a maximum number of iterations. The following section discusses informational metrics and the proposed model's potential for what-if analysis.

3.5 Informational Metrics and What-if Analysis

The primary purpose of the model described in this paper is to produce optimal resource allocations for AMIO problem instances. As defined in section 2,

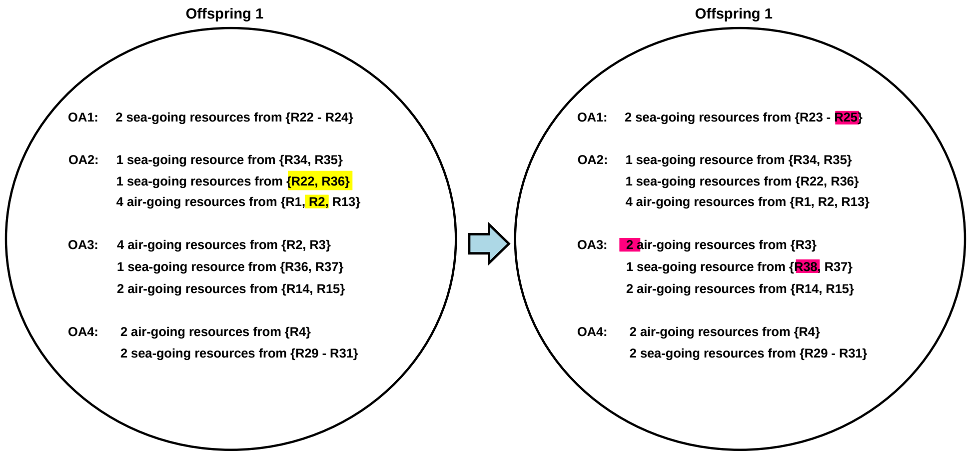

Fig. 6. Crossover genetic operator - duplicate resource correction

the AMIO problem seeks allocations that minimize multiple costs, maximize multiple functionalities, and are subject to multiple constraints. Because the problem has numerous aspects it is useful to gain insights into what aspects are well handled by a particular solution and what aspects are badly handled. It also may be useful to for the model to provide "what-if" analyses, that is allow the running of alternative scenarios to gather insight into what improvements in mission execution and mission cost can be gained or lost by using alternative resource pools and/or under alternative environmental conditions.

The model proposed here seeks to provide support for such activities by generating for each optimization scenario a set of informational metrics that allow insights into the multiple aspects of the AMIO problem. Potential metrics to be generated by the model include smuggling detection and interdiction rates, migrant absorption per resource, number of escaped/interdicted migrants, number of deterred migrants, among others. These metrics combined with multiple scenario execution potentially provide the user with powerful tools to better understand a complex problem. Figure 7 gives a logical diagram of the proposed EA optimization model for the AMIO problem. The diagram is similar to that of figure 3 with two outputs instead of one: the optimized resource to operational area mapping and optimization scenario informational metrics.

4 ROADMAP FOR TECHNOLOGY TRANSFER

This paper describes the AMIO resource allocation problem and proposes a customized optimization model based on the EA paradigm. The high-level design of the proposed model is discussed and its features detailed. The end

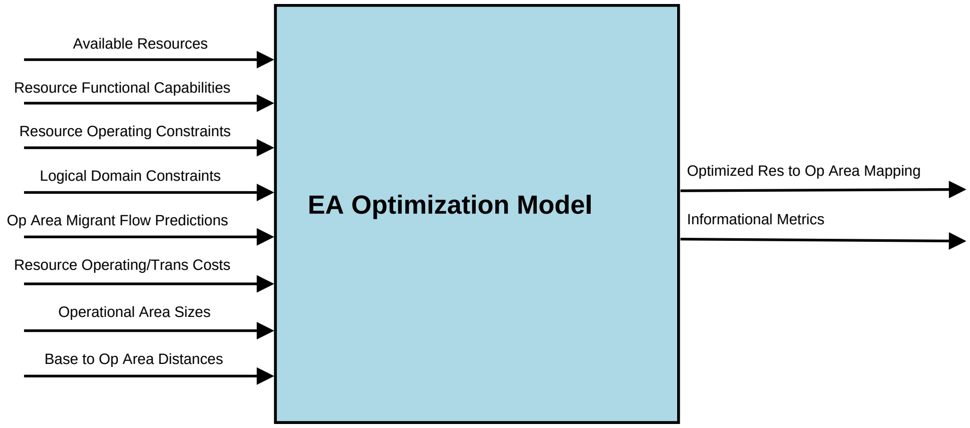

Fig. 7. EA optimization model logical diagram

goal is to build an enterprise-grade system that can provide AMIO decision support to USCG operations personnel at the district level for the Atlantic region.

In order to realize this goal there are a number of necessary steps. The first point that must be clear is that the proposed system is fundamentally different than many commercial software systems because it is to be customized to USCG needs and because it is being asked not simply to provide speed and automation to a well-understood process but instead to solve a very complex problem *better than human solvers alone can*. Such a system is not built with a single design, implement, test cycle of software development. It is necessary to execute the software development cycle iteratively to first produce a functional system and then build on that system, enhance, and refine it until it is developed to a performance level that allows for frequent and meaningful use.

Another point that should be recognized is that the building of such a system requires a team effort and involves algorithmic scientists, software engineers, project managers, and operations analysts. Personnel fulfilling the above-listed roles must be involved at *every* step in order to ensure a successful result.

This paper is a "conceptual" study meant to define high-level system requirements, propose a solution, and provide the high-level design of the proposed solution. The next step following this is the development of a research-grade first prototype. The goal of this step is to build the system's core functionality and perform rigorous testing and experimentation to verify the model. This step represents the "heavy lifting" step of the project as creating the core of a system based on cutting-edge optimization science is a significant undertaking. In addition to the human project resources listed above there may also be a need for computational/IT resources above what currently exists at Fayetteville State University.

Once a research-grade prototype has been built, tested, and documented, the next step is to transfer the technology to an enterprise-grade system. This requires the development of appropriate user interfaces, input and output interfaces, data storage, distributed network access/communication functionalities, and information security funtionalities. This also requires the development of a deployment environment, documentation and user training support, and system support and maintainence. For this step in addition to the human resources mentioned previously it is also necessary to involve a software vendor who is capable of executing this step of the technology transfer and providing long-term system support.

Once an enterprise-grade system is in place, future steps may include enhancements to the system as well as the development of similar optimization systems for different districts/geographical regions under USCG control. With appropriate resources and planning this project may become the starting point for significant improvements to and transparency of USCG operations.

ACKNOWLEDGEMENTS

This work presented in this paper was sponsored by Fayetteville State University's Center for Defense and Homeland Security, http://www.uncfsu.edu/cdhs/.

References

1. US Executive Order 12807, 1992.
2. Bäck T., Fogel D., Michalewicz Z. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., 1997.
3. Burkard R., Dell'Amico M., Martello S. *Assignment Problems*. SIAM, 2012.
4. Garey M.R. and Johnson D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.
5. Koza J. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
6. Koza J., Keane M., Streeter M., Mydlowec W., Yu J., and Lanza G. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers, 2003.
7. Michalewicz Z. and Fogel D. *How to Solve it: Modern Heuristics*. Springer-Verlag, 2004.
8. Michalewicz Z. and Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4:1–32, 1996.
9. Michalewicz Z., Deb K., Schmidt M., and Stidsen T. Test-case generator for constrained parameter optimization techniques. *IEEE Transactions on Evolutionary Computation*, 4:197–215, 2000.
10. Mitchell M. *An Introduction to Genetic Algorithms*. MIT Press, 1996.
11. Noble D. *The US Coast Guard's War on Human Smuggling*. University Press of Florida, 2011.

12. Schoenauer M. and Michalewicz Z. Evolutionary computation at the edge of feasibility. In *Proceedings of the 4th Parallel Problem Solving from Nature*, pages 245–254, 1996.
13. Schoenauer M. and Michalewicz Z. Boundary operators for constrained parameter optimization problems. In *Proceedings of the 7th International Conference on Genetic Algorithms*, pages 320–329, 1997.