# Application of a COTS Resource Optimization Framework to the SSN Sensor Tasking Domain – Part I: Problem Definition

**Triet Tran**

*Cornerstone Consulting LLC, 835 Tyco Ct, #1A, Colorado Springs, CO 80906*

*triettran75@gmail.com, 1 (719) 360-9878 (m).*

**ABSTRACT**

With the onset of the SmallSat era, the Resident Space Object (RSO) catalog is expected to see continuing growth in the near future. This presents a significant challenge to the current sensor tasking of the Space Surveillance Network (SSN). The Air Force is in need of a sensor tasking system that is robust, efficient, scalable, and able to respond in real-time to interruptive events that can change the tracking requirements of the RSOs. Furthermore, the system must be capable of using processed data from heterogeneous sensors to improve tasking efficiency.

The SSN sensor tasking can be regarded as an economic problem of supply and demand: the amount of tracking data needed by each RSO represents the demand side while the SSN sensor tasking represents the supply side. As the number of RSOs to be tracked grows, demand exceeds supply. The decision-maker is faced with the problem of how to allocate resources in the most efficient manner.

Recently we developed a framework called Multi-Objective Resource Optimization using Genetic Algorithm (MOROUGA) within one of Braxton's modern COTS software products, ACE Premier™ Intelligent Resource Optimizer (AceIRO). This optimization framework takes advantage of the maturing technology of evolutionary computation in the last 15 years. This framework was applied successfully to address the resource allocation of an AFSCN-like problem. In any resource allocation problem, there are five key elements: (1) the resource pool, (2) the tasks using the resources, (3) a set of constraints on the tasks and the resources, (4) the objective functions to be optimized, and (5) the demand levied on the resources. In this paper we explain in detail how the design features of this optimization framework are directly applicable to address the SSN sensor tasking domain. We also discuss our validation effort using two- and three-objective test problems as well as present the result of the AFSCN resource allocation domain using a prototype based on this optimization framework.

## 1. INTRODUCTION

With the onset of the SmallSat era, the Resident Space Object (RSO) catalog is expected to see continuing growth in the near future [1, 2]. This presents a significant challenge to the current sensor tasking of the Space Surveillance Network (SSN). The complexity of coordinating the heterogeneous sensors tasking of similar systems is well-known [3]. In addition, the current space-based space surveillance (SBSS) soon will be part of the overall SSN asset that will routinely be tasked for space situational awareness (SSA) usage [4]. One of the unique challenges of the sensor tasking for the SSN is the large number of RSOs that must be tracked in such a way as to produce acceptable orbital state accuracy (including mean orbital state and associated covariance information) to support conjunction analysis and collision avoidance (CA) maneuver planning [5].

Recently we developed a framework called Multi-Objective Resource Optimization Using Genetic Algorithm (MOROUGA) hosted by Braxton's 4th generation Commercial Off-the-Shelf (COTS) product ACE Premier™ Intelligent Resource Scheduler (AceIRO). This framework was successfully demonstrated in an Air Force Satellite Control Network (AFSCN)-like scheduling domain [6].

We took advantage of the rapid growth in multiobjective genetic algorithm (GA) research in the past decade [7, 8, 9] to implement the MOROUGA framework. MOROUGA generalizes the specific problem of task scheduling into an economic problem of resource allocation. In any resource allocation problem, (resource) demands always exceed supplies: the decision maker is faced with the problem of how to allocate the resource in the most optimal way. In this paper we will show how this framework can be extended to address the optimization of the SSN sensors tasking. Applying GA to solve the AFSCN scheduling problem has been attempted in the past with some success, but the specific domains also presented many challenges [10, 11]. This is the reason GA has not seen widespread application in the aerospace community, despite its popularity in many other fields. In many practical optimization problems, often there are many different and conflicting objectives to consider. For example, in manufacturing, one would like to maximize production while minimizing its cost. Or, in logistics, one would like to deliver as many packages as possible while minimizing delivery time. It is no different with the SSN domain. As will be shown, there are advantages in adopting multiple objectives over single objective formulation of an optimization framework.

There are a number of solution techniques that have been proposed to address the optimization of SSN sensor tasking. Reference [12] work is well known and is based on marginal analysis of the energy dissipation rate of orbital objects. OrbitLogic in [13] recently proposed an optimization algorithm inherent in its COTS product with a proprietary optimization search. Combinations of multiple algorithms were also proposed including the algorithm for bottleneck avoidance [14]. Covariance-based scheduling also has been proposed [15]. All these efforts have focused on optimizing an aggregate of multiple parameters into a single cost function (objective). To the author's knowledge, there has not been any attempt of using multiobjective optimization for resource allocation in the SSN community. Our work represents the first attempt at proposing a multiobjective formulation for the SSN domain.

The rest of the paper is organized as follows: Section 2 outlines the mathematical framework of a multiobjective optimization problem, Section 3 briefly reviews the genetic algorithm, Section 4 deals with how we transform the resource allocation into a multiobjective optimization problem, Section 5 describes the key features of the MOROUGA framework, Section 6 presents the result of MOROUGA, Section 7 highlights the challenges of applying MOROUGA to the SSN sensor tasking domain, and Section 8 provides a summary and conclusion. Appendix A discusses our validation approach of the optimization engine using two- and three-objective test optimization problems, and Appendix B summarizes the verification of MOROUGA with previous work using publicly available data.

## 2. MATHEMATICAL FRAMEWORK

This section provides a brief review of mathematical framework for the general multiobjective optimization problem.

Find a vector $\mathbf{x}^* = \left[ x_1^*, \quad x_2^*, \quad \dots, \quad x_n^* \right]$ which will optimize the vector function (aka, objective function)

$$\mathbf{f(x)} = \left[ f_1(\mathbf{x}), \quad f_2(\mathbf{x}), \quad \dots, \quad f_k(\mathbf{x}) \right]^T$$

subject to the *m* inequality constraints

$$g_i(\mathbf{x}) \geq 0 \qquad i = 1, 2, \dots, m$$

and the *p* equality constraints:

$$h_i(\mathbf{x}) = 0 \qquad i = 1, 2, \dots, p$$

where $\mathbf{x} = \left[ x_1, \quad x_2, \quad \dots, \quad x_n \right]$ represents any sample of the solution space $\mathbf{X}$, i.e., $\mathbf{x} \in \mathbf{X}$.

In multiobjective optimization, the term "optimal" must be understood in the sense of Pareto Optimality [9]. In this sense, the optimal solution contains multiple solutions that form a Pareto set. This set represents "trade-offs" in the objective space.

The Pareto Optimality is best illustrated with a diagram, as seen in Fig. 1.

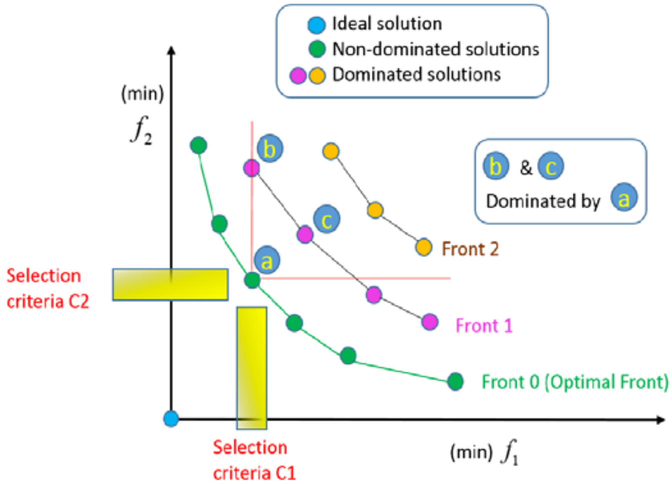


**Fig. 1. Illustration of Pareto Optimality and Pareto Dominance**

In the above figure, there are two objective functions that are to be minimized simultaneously. The ideal solution is assumed to be situated at the origin (0,0). In a problem with conflicting objectives, the ideal solution will never be reached. Each circle in the figure represents a solution, **x**, to the optimization problem. Solution **a** is said to dominate solutions **b** and **c** because solution **a** has at least one objective value that is equal to or better than those of **b** and **c**. Note also that solution **a** dominates all solutions of Front 2. Solutions represented by the green circles are said to be Pareto Optimal (Front 0) since, as a set, they dominate other solutions (Front 1 and Front 2). Any solution belonging to the Front 0 can be selected as a unique optimal solution. The Front 0 therefore represents the best compromising choice in the objective space. Higher level information, such as criteria for the objective value, must be invoked to decide on a particular solution.

To apply this mathematical framework to the resource allocation problem, we introduce a new set of variables that represent the resource item set $\mathbf{q} = \left[ q_1, \quad q_2, \quad \ldots, \quad q_r \right]$, and the corresponding constraints on the resources, namely:

$$\theta_i(\mathbf{q}) \geq 0, \qquad i = 1, 2, \ldots, s$$
$$\phi_i(\mathbf{q}) = 0, \qquad i = 1, 2, \ldots, t$$

For the particular resource allocation problem, we ascribe a particular meaning to the solution **x**; namely it is a set of all tasks (or activities) that use a subset of **q** that must be carried out within some specified window of time. We therefore denote $\mathbf{x} = \mathbf{x}(\mathbf{q})$ as a set of all tasks that will be assigned the required resources within **q**.

## 3. BRIEF REVIEW OF GENETIC ALGORITHM

Analytical solutions exist for the simplest of scheduling problems first addressed in the mid-1950s [16]. In general, the scheduling problem is an NP-hard problem: a solution to the optimal problem cannot be obtained in finite polynomial time [17]. For this reason, researchers have been driven to find approximate solutions that can be obtained within reasonable execution time, such as GA, Particle Swarm Optimization (PSO) [18], and Ant Colony Optimization [19]. In many practical applications approximate solutions are very adequate in addressing the need of the decision makers. Advances in evolutionary computing in recent decades gave rise to many heuristic solution techniques that take advantage of computing technologies: distributed computing as well as graphical processing unit (GPU)-based computing [8]. We also adopted the heuristic method of solution in using GA in the MOROUGA framework as it holds many promising features for the SSN domain, as will be shown in this paper.

Genetic algorithm is a modern population-based stochastic optimization search. It is modeled after the Theory of Evolution based on the concept of "survival of the fittest" [9]. Individuals within a population that are a better fit with respect to the pressure of the environment will have a better chance of survival than others. Those that survive are mated to one another to produce children that bear the "genes" from both parents. The children may or may not be better than their parents. The selection process, often known as "Elitism," favors those with better fitness scores. Individuals with lower fitness scores are discarded to keep the number of individuals within the population constant. The reproduction cycle repeats until some convergence criterion is reached, for example, the maximum number of generational iteration or no more significant gain in fitness scores for all objectives.

In our MOROUGA implementation, we adopt the multiobjective optimization algorithm of [20] which is summarized in the flowchart in Fig. 2. The specific feature of this algorithm is in the fast non-dominated sorting technique that can sort individuals across all objective scores simultaneously. This sorting algorithm is used in ranking the individuals within the population based on their combined fitness score (boxes 2 and 7 in Fig. 2).

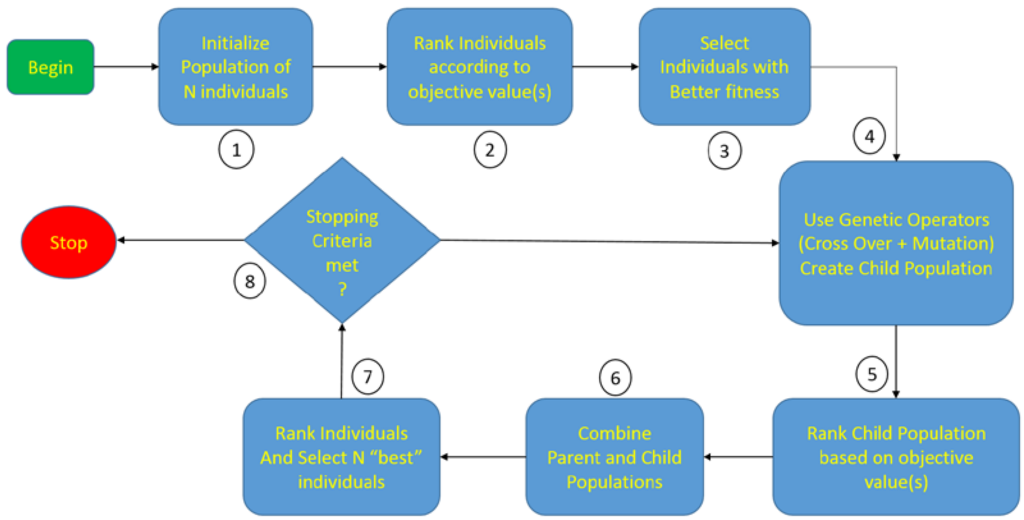


**Fig. 2. Logic Flow Diagram of Genetic Algorithm**

In heuristic searches such as GA, there is no "proof" of optimality. However, in most practical implementations of GA, convergence criteria are used to indicate when "optimal objective values" have been reached and is often used as a tuning parameter for a particular domain of application. Other tuning parameters include the number of individuals in a population, the genetic operator of crossover, and mutation.

There are a number of significant features that exist within most evolutionary algorithms that do not exist in classical optimization such as gradient-based nonlinear programming. First, GA does not require knowledge of a priori (starting) solution to set the iteration loop in motion. Second, the objective functions need not be continuous as there is no gradient needed to be computed in GA. Third, with the proper tuning, global optimization is possible but cannot be proved. The global optimal solution is related to the "diversity" of the individuals within the population and there are techniques to ensure diversity is attained. Fourth, in multi-objective optimization, constraints on the solution can be separated from the cost (objective) functions [25], unlike single-objective optimization where constraints are often added to the cost function with some arbitrary weighting factors. For an in-depth treatment of evolution computing, the reader is referred to the work by Deb in [9].

## 4. RESOURCE ALLOCATION AS A MULTIOBJECTIVE OPTIMIZATION PROBLEM

As mentioned in Section 2, the resource allocation problem can be mapped into a multiobjective optimization problem using an added dimension for the resource items. Section 3 points to the solution method of GA as a viable candidate to consider for getting to an approximate optimal solution which, in many practical applications, is all that is needed to make an informed decision.

In any resource allocation, we assert that there are five key components to the problem:

1. The resource pool – a set of all resource items that can be assigned to a task or activities;
2. The task (or activity) definition – tasks to be defined as different types having a start and end time and needing different combinations of resources to execute;
3. The constraints on the resources and tasks;
4. The objective functions to be optimized; and
5. The demand placed on the resource.

In our MOROUGA framework, the above elements are implemented in a generic sense to allow MOROUGA to address different domains. From a resource allocation perspective, task scheduling can be viewed as an economic problem of supply and demand where demand is "insatiable" and supply (of resource) is "scarce." In such situations resource conflicts naturally arise. The SSN sensor tasking domain is no different where the growing number of RSOs required to be tracked will easily exceed the capacity of the SSN sensors. In this situation, the decision maker is faced with the problem of how to allocate available resources to achieve some metric of "optimality" such as most efficient, most fair, least cost, maximum information return, etc., while satisfying all constraints levied on the resources as well as the tasks. When there are unresolvable conflicts, the decision maker must accept certain tasks and reject others. In many cases the resource availabilities become a direct constraint on the problem, for example, non-dedicated sensors of the SSN (contributing and collateral sensors) [21].

It is obvious that in any scheduling problem the common dimension is time. The quantity that will most often be used is the time interval as a task must be defined over an interval, visibility between a tracker and a target is an interval, downtime of a resource is an interval, availability of contributing and collateral sensors is a set of intervals, etc. The Gantt chart is most useful in representing intervals associated with tasks or resources. In what will follow, we illustrate the general problem of resource conflict and, later, show how the mutation operator of the genetic algorithm is implemented in MOROUGA to resolve the conflict.

Fig. 3 illustrates a Gantt chart depicting the mapping of resource to task indicating which resource is needed by which task over which intervals. In general, a task may require multiple resources to execute, such as how a knee operation may require various doctors, nurses, and assistants in a medical domain. Different task types will require different combinations of resources. From the figure, conflict is seen as the same resource being requested by different tasks over an overlap interval. Five overlap intervals are seen in the figure but only four are unique conflicting task pairs. The concept of conflict resolution is fairly straightforward: if there exists a different interval over which a conflicted task can be "moved," then the action is to assign that task to that interval. The second option is to find a resource of the same type, if it exists, that has not been requested in the same interval. Conflict resolution is either a change in time interval, a change in resource assignment, or both.

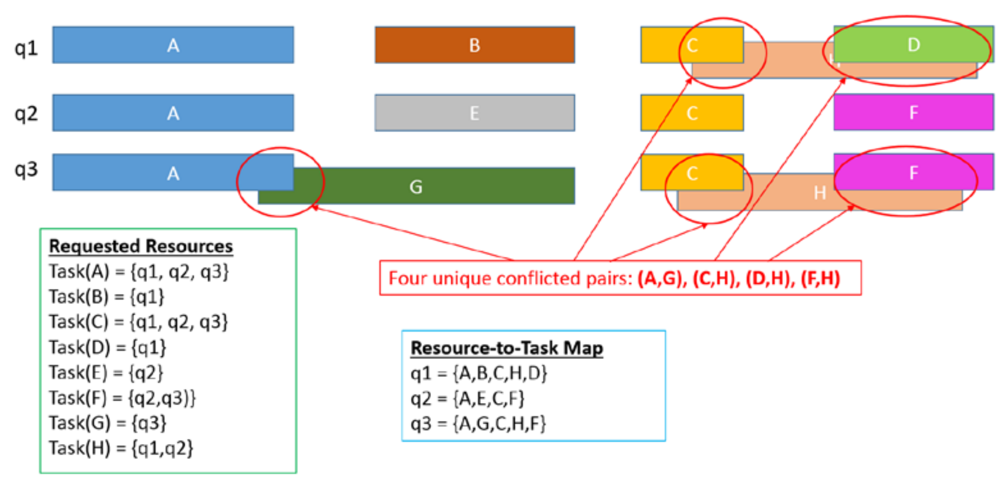


**Fig. 3. Illustration of Resource Assignment and Conflict**

Since conflicts naturally arise in a scheduling problem, one of the objectives used most often is the minimization of the number of conflicts during the scheduling period. This particular objective is also applicable to the SSN sensors tasking. Another objective that we adopted for the MOROUGA framework is to minimize the overlap between any unresolvable conflicting tasks. This objective, when optimized, allows the operator to further resolve the conflict by shortening the time duration of each task to remove the overlap.

## 5. DESIGN FEATURES OF MOROUGA

The MOROUGA has been implemented in software into two distinct layers: the Resource Domain Translator (RDT) (patent pending) and the Multi-Objective Optimization Engine (MOOE), as shown in Fig. 4. The RDT main function is to translate and map the domain specific entities ("physics") into the genetic algorithm entities of the MOOE (pure "math") and perform the reverse mapping. The MOOE is implemented after the work of Deb [20] with improvement in non-dominated sorting using the algorithm of [22]. The separation of distinct layers allows for validation of the MOOE using analytical functions with known Pareto optimal solutions of two and three objectives (Appendix A). The RDT has been verified using publicly available data and compared against the single-objective GA work of [10,11] (Appendix B). In the MOROUGA framework we leverage extensively polymorphism of the objected-oriented design in implementing the RDT layer. In this implementation, each task to be scheduled is mapped as a gene of an individual who represents a complete schedule. The rich feature of the attributes of a task makes possible the implementation of a purposeful mutation for conflict resolution. The de-confliction algorithm implemented within MOROUGA is similar to repairing a defective gene: a conflicted task which is rejected is marked as a "defective" gene which must be repaired. The repair mechanism is the mutation operator which either moves the rejected task to a different time interval or assigns a different resource or both. This is done in a stochastic manner and does not guarantee that the repair will be successful. However, due to "Elitism" of the GA, the healthier individuals will survive the selection by the pressure of the environment and pass on the better genes to their children, grandchildren and so on. Over generational iterations, individuals will become healthier and thus minimization of the number of conflicts is achieved.

The complete separation of the physics from the math affords the flexibility of using MOROUGA for different physical domains. Since this is a data-driven platform, many of the data structures are defined in the database. For the SSN sensors tasking domain, the effort then is concentrated at the RDT so that a proper translation is achieved to allow the pure algorithm to obtain the approximate Pareto-optimal solution set. The design of the user-specified objective function is possible via a simple API that allows the math level to perform the processing without knowing how the objective functions are computed. The advantage is that an external process can provide the objective functions without disturbing the fundamental architecture of MOROUGA.

An important feature of the COTS product AceIRO in hosting the MOROUGA framework is AceIRO's distributed architecture where a central scheduling authority approves the final schedule in which task requests are submitted by any number of clients at remote locations. The scheduling authority and remote clients all have the same computational power to perform what-if scenarios, but only the scheduling authority can approve and distribute the approved schedule to sites that will execute the scheduled tasks.

Another key function of the RDT is the construction of the feasible region. This construction is based on the constraints levied on the resources as well as the tasks. Fig. 5 illustrates an example of constructing the feasible region. This region is simply a set of intervals free of constraint. We are confident that this is the most appropriate method of handling constraints for resource optimization as it guarantees that constraints are not violated. As illustrated, Fig. 5 is an example of a task of using a designated resource to track a specific target which involves a set of visibility intervals between tracker and target (e.g., due to orbital geometry) and the set of intervals over which the resource is not available (e.g., optical sensors that can only operate in darkness on clear nights). The task therefore can only be placed within the feasible intervals.

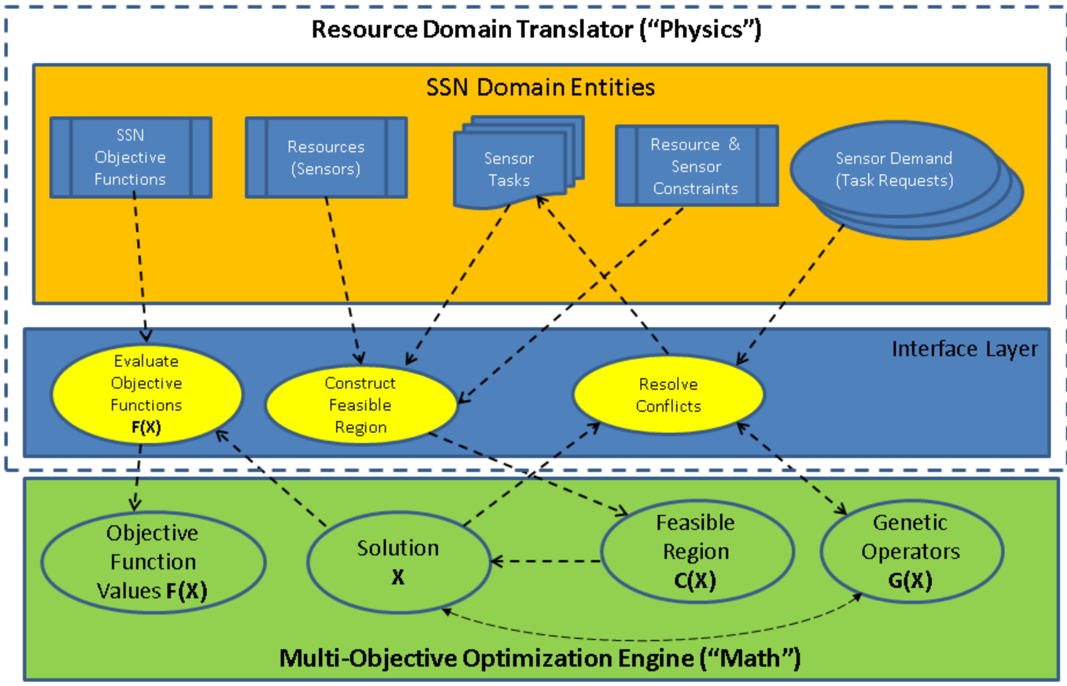


**Fig. 4. The Resource Domain Translator – Separate the "Physics" from the "Math"**

$Task\ a_k$ : Use resource $q_i$ to track object $s_j$

where

$A = \{a_k\}_{k=1,2,\dots K}$ set of all tasks

$Q = \{q_i\}_{i=1,2,\dots I}$ set of all resource (trackers)

$S = \{s_j\}_{j=1,2,\dots J}$ set of all RSO's

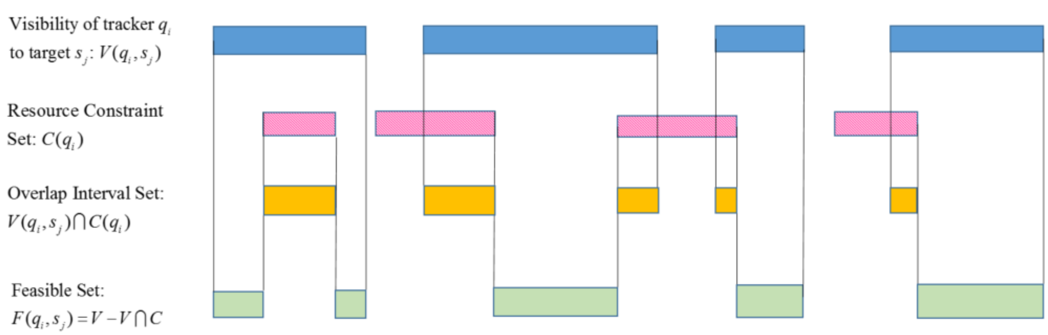


**Fig. 5. Illustration of Feasible Region Construction Process**

## 6. SAMPLE RESULTS FROM RUNNING MOROUGA

As mentioned previously, we have used publicly available data to verify our implementation of MOROUGA. Fig. 6 and Fig. 7 illustrate the before and after optimization of a test case in which about 300 contact tasks are to be scheduled using a set of 16 antennas over a 24-hour period. This is an example of an AFSCN scheduling problem. Initially there are about 70+ conflicts; after the optimization run, there remain three conflicts that cannot be resolved. The optimization was run with two simultaneous objectives of minimizing the number of conflicts and minimizing the overlap of conflicted tasks. However, due to this second objective, it is demonstrated that the conflict task pairs have very small overlap intervals which could further be resolved if the task durations are allowed to be shortened.

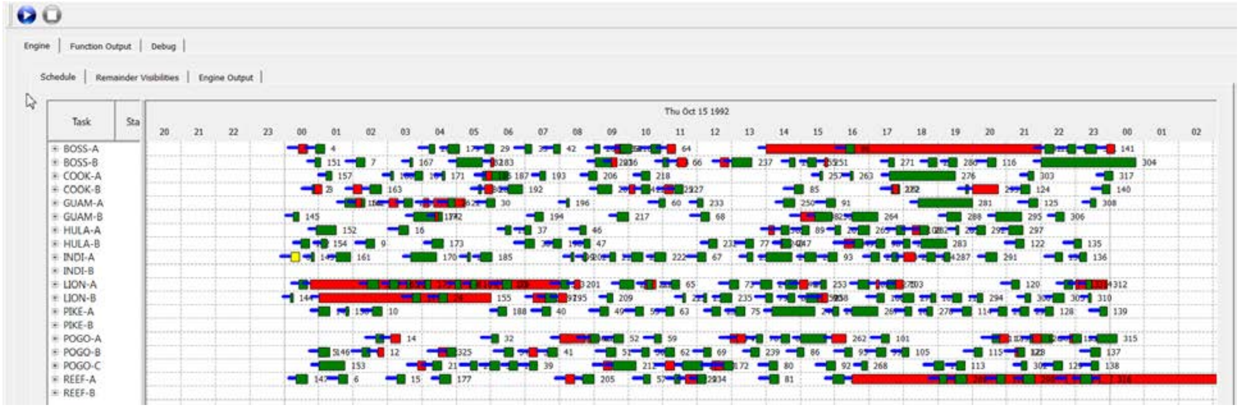

**Fig. 6. AFSCN Schedule before Optimization – Red indicates rejected tasks (70+ conflicts)**

**Fig. 7. AFSCN Schedule after Optimization (3 unresolved conflicts)**

In another recent study [24], we used MOROUGA to investigate the capacity of a proposed network of ground antennae to determine the minimum number of antennae that can support upward of 450 contact tasks during a 24-hour period with zero conflict. Fig. 8 and Fig. 9 show the before and after optimization using the same two objectives as in the AFSCN case. The execution time of the run is also indicated in Fig. 9.

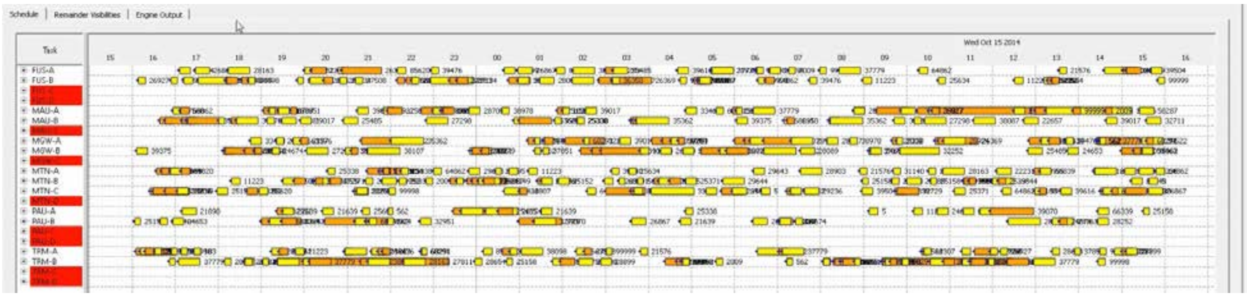

**Fig. 8. Initial Conflicts Before Optimization (170 conflicts)**

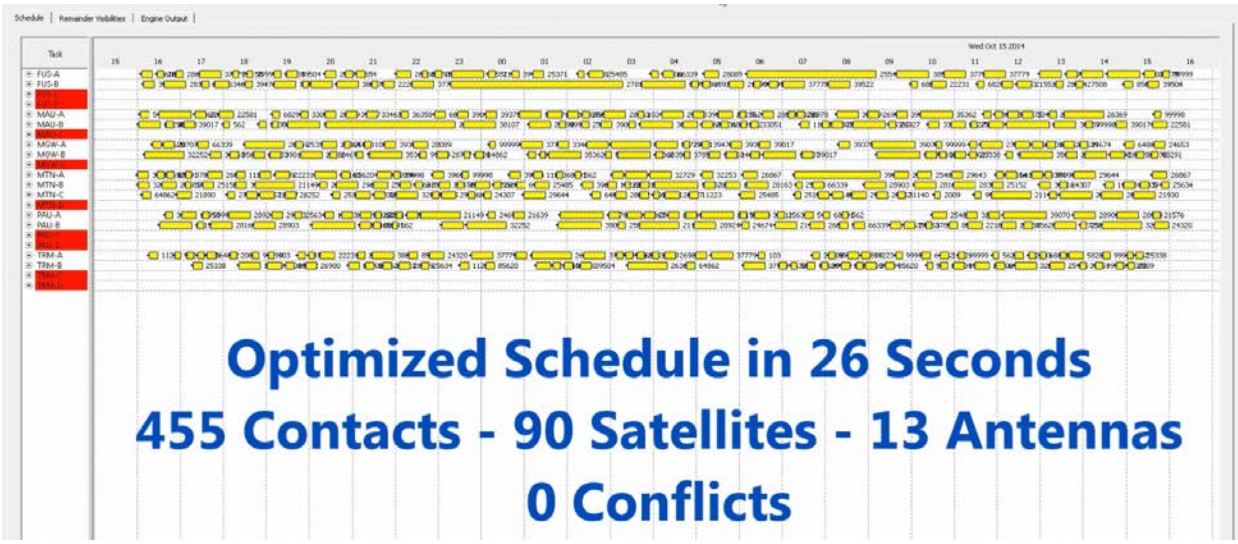


**Fig 9. Schedule After Optimization Run (antennas painted red are excluded from usage)**

## 7. CHALLENGES OF SSN SENSOR TASKING FOR MOROUGA

From the foregoing discussion, it is obvious that MOROUGA as hosted by the distributed architecture of AceIRO can be extended to map the entities of SSN sensor tasking domain into a multiobjective resource optimization problem. We identify all the SSN sensors as a resource set that can be tasked to track any RSO. Different task types can be defined for optical sensors, phase-array radars, and space-based sensors. Each sensor has its particular operational modes and constraints depending whether the sensors are dedicated, contributing, or collateral. Each sensor can be modeled to track each RSO according to its orbital characteristics and radar cross section or visual magnitude. The demand on the SSN sensors is represented by a set of selected RSOs, for example, those with highest priority. This set can change on a daily basis to accommodate the feedback information from each sensor as well as the accuracy knowledge required for the RSOs. Multiple optimization objectives can be defined to address information gained of the RSO set, efficient usage of all resources, minimization of the number of resource conflicts, or any other objectives that are deemed important to the community. It is worth noting again that the results of multiobjective optimization provide the “best” trade-offs in the objective space for the decision maker. Evaluation of neighboring solutions provides context for the selection of a particular solution.

The recent results of MOROUGA demonstrate the potential of using multiobjective genetic algorithm to address the SSN sensor tasking. Certainly there are a number of significant challenges in applying MOROUGA to the SSN domain. One of these is due to a large and growing number of RSOs that must be addressed as well as the complex task of modeling the different ground and space-based sensors with different characteristics and constraints. With tens of thousands of tracking tasks to be scheduled, a population-based algorithm such as MOROUGA will be overwhelmed using a single CPU machine. Therefore, to show that MOROUGA can be scaled with the growing number of RSOs, we must demonstrate that MOROUGA can be implemented on a distributed computing or GPU-based platform. The feasibility of implementing GA on a GPU has been demonstrated by recent works [8,23], taking advantage of the fact that individuals within a population work independently from one another. Implementing MOROUGA to run on a GPU-based platform is the obvious next step.

An additional challenge is implementing the capability of MOROUGA to use the output of the SSN sensor as a factor in the evaluation of the objective functions. For example, a missed opportunity will result in the growth of the uncertainty (covariance) of the orbit state of the RSO. Such a missed opportunity can be made to correlate with an increase in priority of the RSO which will impact the output schedule. This shows the need for any robust SSN scheduling system to react to interruptive events that can change the resulting optimized schedule. During our testing of MOROUGA, we have also demonstrated the ability to re-optimize the schedule where a site was taken down unexpectedly. The host AceIRO is built with a monitoring function that triggers a re-optimization run whenever such an event is detected.

## 8. SUMMARY AND CONCLUSION

We have described the implementation and discussed the features of MOROUGA, a framework that is implemented with a multiobjective optimization engine using genetic algorithm applied to a resource optimization problem. We pointed out the advantage of using multiple objectives formulation over single objective, as well as the advantage of using genetic algorithm for optimization search versus traditional gradient-based optimization. We presented results of using MOROUGA for an AFSCN-like scheduling problem to demonstrate the merits of multiobjective genetic algorithm. We showed how the MOROUGA framework can be extended to address the SSN sensor tasking domain. However, the most challenging aspect of applying this framework is the ability of a population-based optimization search to address tens of thousands of RSOs in a reasonable execution time. This performance metric is important to assess whether such algorithm can be relied upon to address real time re-optimization due to interruptive events.

The application of multiobjective optimization search using genetic algorithm has not seen widespread use in the aerospace community. Our present work with MOROUGA shows many promises and applicability of this framework to a variety of resource allocation domains. Work with MOROUGA will continue to demonstrate its applicability to the SSN sensor tasking domain.

## 9. ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Braxton Technologies, LLC, for the financial support of this work. He would also like to thank the dedicated engineering team, led by Robert Johnston, who provided excellent technical support in putting together the COTS product AceIRO which hosts the MOROUGA framework.

## 10. Appendix A – Validation of the Algorithm

This section briefly shows our approach to validating the "math" component of the MOROUGA, also called the "engine" (Multi-Objective Optimization Engine – MOOE). The heart of the MOOE is the non-dominated sorting algorithm called NSGA-II as detailed in [20]. This reference also provides a number of analytical and empirical test problems for verification of multi-objective evolutionary algorithms. Additionally, the reference describes two methods of measuring the results: a convergence metric which measures the convergence of the solutions onto the Pareto front, and a diversity metric which measures the diversity of solutions along the Pareto front. As a companion to the reference paper, the Kanpur Genetic Algorithms Laboratory (KanGAL, http://www.iitk.ac.in/kangal/codes.shtml) has available the software which demonstrates the NSGA-II algorithm against a set of standard multi-objective problems. The validation of the MOOE uses the NSGA-II software as a comparison truth source.

The problems listed in Table A-1were selected to test the implementation of the MOOE. These problems came from a combination of problems listed in [20] and the NSGA-II sample software. In this table *n* indicates the number of genes and *m* indicates the number of objectives being optimized.

**Table A-1: MOOE Validation Test Problems**

| Problem | n | m | Variable Bounds | Objective Functions |
|---|---|---|---|---|
| FON | 3 | 2 | $[-4, 4]$ | $f_1(x) = 1 - e^{-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2}$ $f_2(x) = 1 - e^{-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2}$ |
| KUR | 3 | 2 | $[-5, 5]$ | $f_1(x) = \sum_{i=1}^{2} -10e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}}$ $f_2(x) = \sum_{i=1}^{3}\left(\lvert x_i\rvert^{0.8} + 5\sin x_i^3\right)$ |
| SCH1 | 1 | 2 | $[-10, 10]$ | $f_1(x) = x^2$ $f_2(x) = (x-2)^2$ |

| Problem | n | m | Variable Bounds | Objective Functions |
|---|---|---|---|---|
| SCH2 | 1 | 2 | $[-5, 10]$ | $f_1(x) = \begin{cases} -x & x \le 1 \\ x-2 & 1 < x \le 3 \\ 4-x & 3 < x \le 4 \\ x-4 & 4 < x \end{cases}$ $f_2(x) = (x-5)^2$ |
| VNT | 2 | 3 | $[-3, 3]$ | $f_1(x) = \frac{1}{2}\left(x_1^2 + x_2^2\right)\sin\left(x_1^2 + x_2^2\right)$ <br> $f_2(x) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15$ <br> $f_3(x) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1e^{-\left(x_1^2 + x_2^2\right)}$ |

In some cases, the optimal solution is given analytically in the reference. In these cases, the measurement of the convergence metric is measured from the known optimal Pareto front. For all other cases, the convergence metric is measured empirically against a sample of NSGA-II truth data. The NSGA-II truth data was generated using a population size of 500 individuals and ran for a total of 1,000 generations. Table A-2 lists the analytical optimal solution for a test problem or indicates if the test data were compared against the NSGA-II truth data.

**Table A-2: Known Optimal Solutions for Test Problems**

| Problem | Optimal Solution |
|---|---|
| FON | $x_1 = x_2 = x_3 \in \left[-1/\sqrt{3}, \quad -1/\sqrt{3}\right]$ |
| KUR | Compared against NSGA-II |
| SCH1 | $x \in [0, 2]$ |
| SCH2 | Compared against NSGA-II |
| VNT | Compared against NSGA-II |

For each of the test problems, the MOOE software and the NSGA-II software were each run for a total of 20 test runs. Each test run was set up to run with the same population size as well as the same number of generations. The implementation of the MOOE slightly modified the NSGA-II algorithm in two ways: the mutation operation is different from NSGA-II, and the selection of parents to create new children has been modified. The resulting differences between MOOE and NSGA-II are small but inconsequential.

**Table A-3: MOOE Test Problem Setup**

| Problem | Pop Size | # Generations |
|---|---|---|
| FON | 100 | 100 |
| KUR | 100 | 100 |
| SCH1 | 48 | 500 |
| SCH2 | 48 | 50 |

| Problem | Pop Size | # Generations |
|---|---|---|
| VNT | 500 | 300 |

After all the runs were complete, the solutions were analyzed to determine the Euclidean distance of each individual in the final generation to the Pareto front. In addition, plots have been made showing sample initial and final iteration. Table A-3 lists the population size and number of generations used for each test case.

Table A-4 lists the comparison between the MOOE and NSGA-II of the mean and standard deviation of the Euclidean distances for all the test runs.

**Table A-4: Validation Results**

| Algorithm | FON | KUR | SCH1 | SCH2 | VNT |
|---|---|---|---|---|---|
| MOOE Mean | 0.004364 | 0.029142 | 0.006536 | 0.007176 | 0.008786 |
| MOOE std dev | 0.000497 | 0.011269 | 0.001651 | 0.002034 | 0.000320 |
| NSGA-II Mean | 0.002654 | 0.014322 | 0.008047 | 0.005751 | 0.008752 |
| NSGA-II std dev | 0.000199 | 0.001543 | 0.001665 | 0.000559 | 0.000338 |

Despite the small differences in the converged solution in all test problems seen in Table A-4, the overall agreement with the original results of NSGA-II is excellent, as shown in Figs. A-1 to A-5. In all figures, the blue circles represent the result from NSGA-II and the red triangles represent that from the MOOE. The results presented indicate that the MOOE implemented within the MOROUGA framework produces correct results with respect to the known optimization solutions for two- and three-objective optimization test problems.

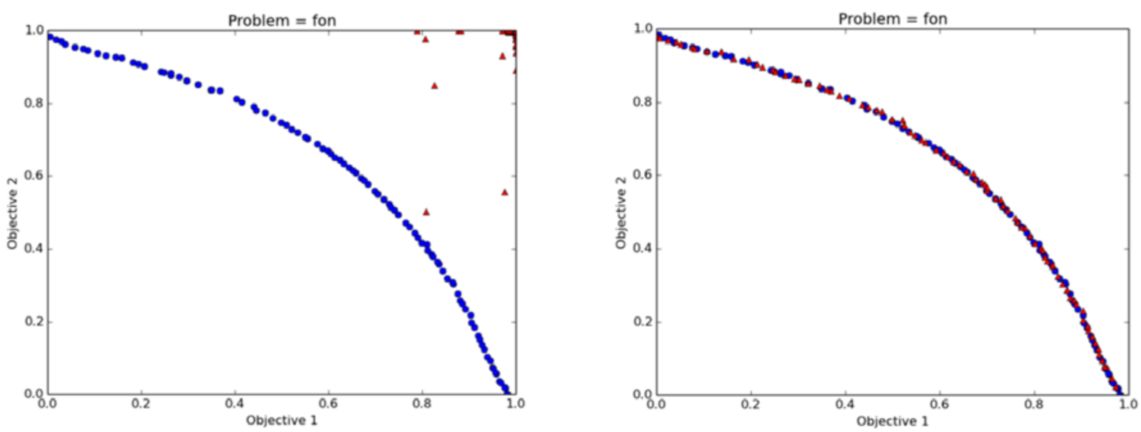


**Fig. A1. Two-objective validation result with FON, left plot: initial iteration, right plot: converged**

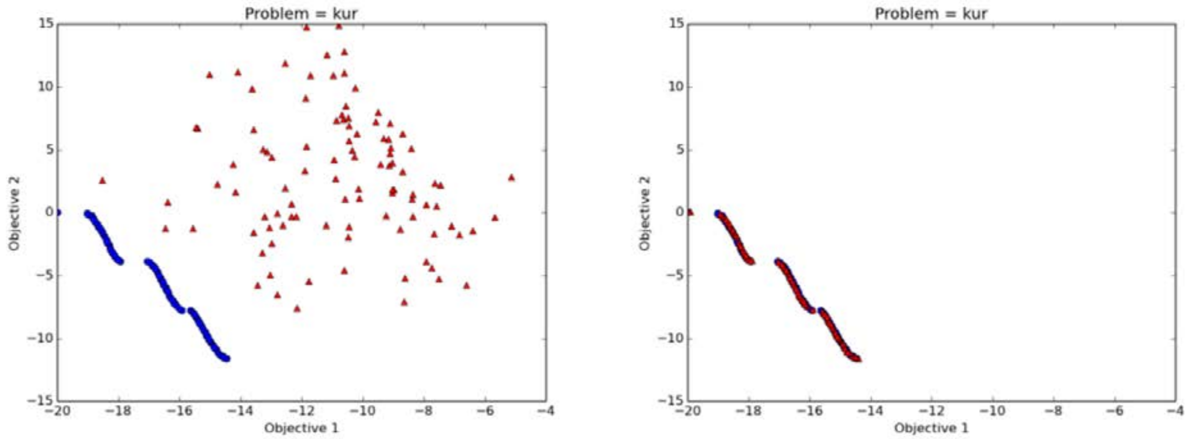


**Fig. A2. Two-objective validation with kur, left plot (initial iteration), right plot (converged).**

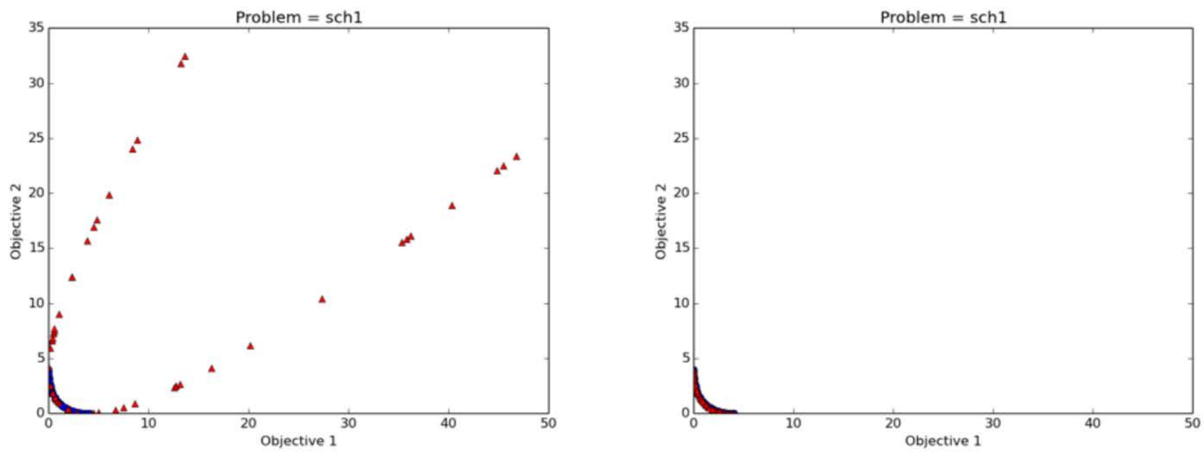


**Fig. A3. Two-objective validation with sch1, left plot (initial iteration), right plot (converged).**

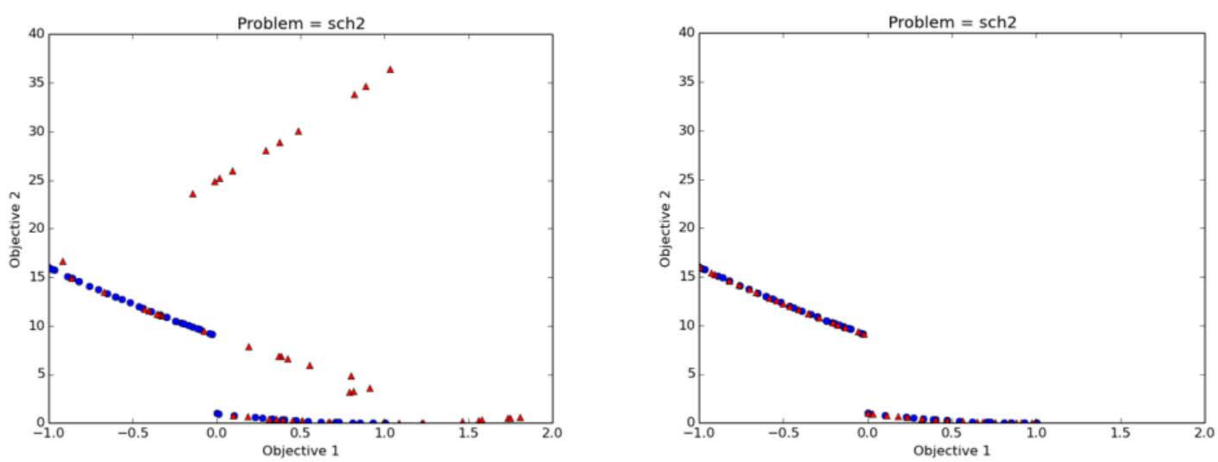


**Fig. A4. Two-objective validation with sch2, left plot (initial iteration), right plot (converged).**

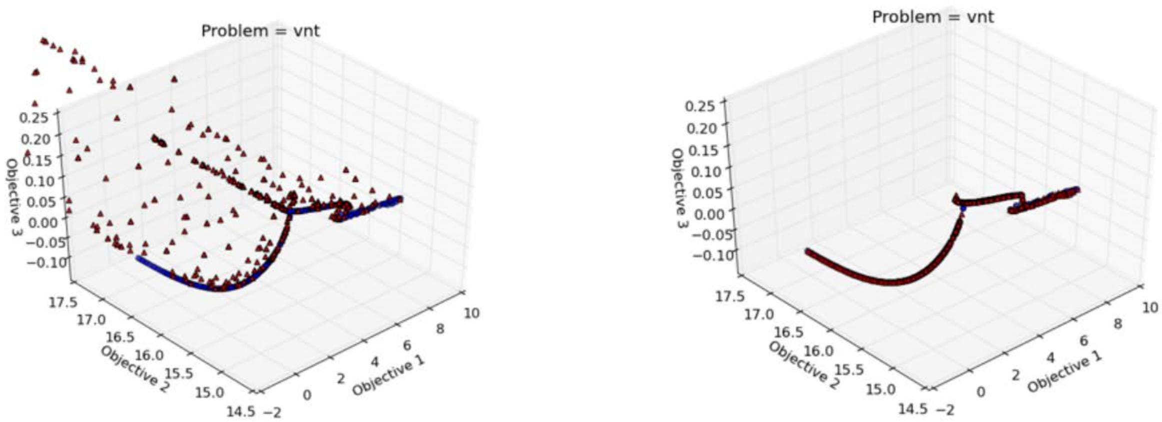


**Fig. A5. Three-objective validation with vnt, left plot (initial iteration), right plot (converged).**

## Appendix B – Verification of MOROUGA

We mentioned the AFSCN scheduling using an implementation of GA in [11]. In this work, there is a set of data that is made publicly available by Colorado State University and can be downloaded (link: http://www.cs.colostate.edu/sched/data.html). This data were originally generated by the Air Force Institute of Technology (AFIT) [10]. There are seven independent data sets for the AFIT data, each covering a 24 hour scheduling period consisting of typically about 300 tasks to be scheduled using 16 AFSCN antennae. The data sets consist of the input tasks (i.e., requests) and all of the associated visibility intervals based on the AFSCN antennae.

The AFIT data files were processed by the MOROUGA prototype and the results are presented in Table B-1 for comparison. Two objective functions selected for simultaneous minimization were the total overlap duration and the total rejected tasks duration. The engine parameters included a population size of 300 and a max iteration of 1000.

**Table B-1: Comparison between MOROUGA and published data of [11]**

| | | # Tasks rejected | | Total Overlap (minutes) | |
|---|---|---|---|---|---|
| **Data Set** | **Total tasks requested** | **Genitor (*)** | **AceIRO** | **Genitor (*)** | **AceIRO** |
| 10/12/1992 | 322 | 8 | 8 | 104 | 107(**) |
| 10/13/1992 | 302 | 4 | 4 | 13 | 13 |
| 10/14/1992 | 311 | 3 | 3 | 28 | 28 |
| 10/15/1992 | 318 | 2 | 2 | 9 | 9 |
| 10/16/1992 | 305 | 4 | 4 | 30 | 30 |
| 10/17/1992 | 299 | 6 | 6 | 45 | 45 |
| 10/18/1992 | 297 | 6 | 6 | 46 | 46 |

(*) Data from [11]

It is demonstrated that the MOROUGA agrees exactly with the Genitor result of [11], except for the case of 10/12/1992 where there is a discrepancy (**) of three minutes in the total overlap between MOROUGA and published data. It is suspected that there might be some discrepancy with the input data for this case rather than an error in the algorithm. Because the detailed schedule was not published in the reference, it is difficult to perform further analysis. It should be noted that the implementation of Genitor solved for each objective one at a time while

MOROUGA optimized both objectives simultaneously. MOROUGA does not make any distinction regarding different orbital regimes for “low fliers” and “high fliers” as the case with Genitor. This feature points to the general nature of MOROUGA in solving resource allocation problems.

## 11. REFERENCES


1. Svitak, A., SpaceX, OneWeb Unveil Rival Broadband Constellation Plans, Aviation Week & Space Technology, online article, Jan 21, 2015. Retrieved September 10, 2015 from: http://aviationweek.com/space/spacex-oneweb-unveil-rival-broadband-constellation-plans.
2. Richardson, G. et al, Small Satellite Trends 2009-2013, Proceedings Small Satellite Conference 2015, Logan, UT, 2015. Retrieved September 10, 2015 from: http://digitalcommons.usu.edu/smallsat/2015/all2015/48/.
3. Weeden, B. and Kelso, T. S., Analysis of the Technical Feasibility of Building an International Civil Space Situational Awareness System, International Astronautical Congress, Daejeong, South Korea, 2009.
4. Air Force Space Command Fact Sheet, Space Based Space Surveillance (SBSS), online brochure, US Air Force Command Public Affairs, Petersen AFB, 2015. Retrieved September 10, 2015 from: http://www.afspc.af.mil/library/factsheets/factsheet.asp?id=20523.
5. Duncan, M., Operational Collision Risk Management – Evaluating and Mitigating High-Risk Conjunction Events, poster presentation, AMOS Conference 2011, Maui, HI, 2011.
6. Johnston, R. and Tran, T., “Enabling Automated Real-Time AFSCN Scheduling,” Braxton Technologies, LLC, SBIR Phase I Final Report to the Air Force Research Laboratory, Kirkland AFB, NM, 2014.
7. Lucken, C. V., et al, Survey on multi-objective evolutionary algorithms for many-objective problems, *Computational Optimization and Applications*, Vol. 58, Issue 3, pp 707-756, 2014.
8. Gong, Y. et al, Distributed evolutionary algorithms and their models: A survey of the state-of-the-art, *Applied Soft Computing*, Vol. 34, pp 286-300, 2015.
9. Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd. New York, NY, 2002.
10. Barbulescu, L. et al, Trading Places: How to Schedule More in a Multi-Resource Oversubscribed Scheduling Problem, *ICAPS-04 Proceedings*, pp. 227-234; American Association for Artificial Intelligence, 2004.
11. Barbulescu, L. et al, Understanding Algorithm Performance on an Oversubscribed Scheduling Application, *Journal of Artificial Intelligence Research*, Vol. 27**,** pp. 577-615, 2006.
12. Miller, J. G., A New Sensor Allocation Algorithm for the Space Surveillance Network, *Military Operations Research*, Vol. 12, No. 1, pp. 57–70, 2007.
13. Herz, A. et al, SSA Sensor Tasking Approach for Improved Orbit Determination Accuracies and More Efficient Use of Ground Assets, AMOS Conference, 2013.
14. Stottler, R., Intelligent Space Surveillance Network (SSN) Scheduling, Proceedings of the AIAA@Infotech, Vol. 1, no. 2433, pp 387-395, 2012.
15. Hill, K. et al, Covariance-Based Network Tasking of Optical Sensors, in Proc. AAS/AIAA Space Flight Mechanics Meeting, Feb, 2010.
16. Jain, A. S. and Meeran, S., Deterministic job-shop scheduling: Past, present and future, *European Journal of Operational Research*, No. 113, pp 390-434, 1999.
17. Wall, M. B., A Genetic Algorithm for Resource-Constrained Scheduling, PhD Thesis, Sloan School of Management, Massachusetts Institute of Technology, 1991.
18. Venter, G., and Sobiesczanski-Sobieski, J., Particle Swarm Optimization, 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2002.
19. Dorigo, M. and Stutzle, T., *Ant Colony Optimization*, MIT Press, 2004.
20. Deb, K. et al, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002.
21. HQ AFSPC/XOCS, Space Surveillance Network (SSN) Site Information Handbook, May 2003.
22. Jensen, M. T., Reducing the run-time complexity of multiobjective EAs: the NSGA-II and other algorithms, , *IEEE Transaction on Evolutionary Computation*, Vol. 7, no. 5, pp 503-515, 2003.
23. Wong, M. and Cui, G., Data Mining Using Parallel Multi-Objective Evolutionary Algorithms on Graphics Hardware, IEEE World Congress on Computation Intelligence, Barcelona, Spain, 2010.
24. Gressel, S. et al, AFSCN Commercialization Study, Technical Performance Briefing, CDRL-A003, 2014.
25. Ray T. et al, An evolutionary algorithm for constrained optimization, in Proc. Genetic and Evolutionary Computation Conf. (GECCO’2000), D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, Eds., pp. 771–777, San Francisco, CA, 2000.