12-21-2015

Sensor Resource Management: Intelligent Multi-objective Modularized Optimization Methodology and Models

Boris Kovalerchuk

Leonid Perlovsky

Sensor Resource Management: Intelligent Multi-objective Modularized Optimization Methodology and Models

Boris Kovalerchuk[1], Leonid Perlovsky[2]
[1] Dept. Of Compter Science, Central Washington University, USA,
[2] LP Information Technology and Harvard University, USA

Abstract. The importance of the optimal Sensor Resource Management (SRM) problem is growing. The number of Radar, EO/IR, Overhead Persistent Infra-Red (OPIR), and other sensors with best capabilities, is limited in the stressing tasking environment relative to sensing needs. Sensor assets differ significantly in number, location, and capability over time. To determine on which object a sensor should collect measurements during the next observation period k, the known algorithms favor the object with the expected measurements that would result in the largest gain in relative information. We propose a new tasking paradigm OPTIMA for sensors that goes beyond information gain. It includes Sensor Resource Analyzer, and the Sensor Tasking Algorithm (Tasker), which maintains timing constraints, resolution and geometric differences between sensors relative to tasking requirements on track quality and the measurements of object characterization quality, based on the computational intelligence approach of multi-objective optimization, which involves evolutionary methods.

Keywords: Sensor Resource Management (SRM), Multi-objective optimization, adaptive models, integer linear programming, evolutional computing, dynamic logic, optimization under uncertainty.

1 Introduction

The configurations of sensor platforms can include dozens of global radars and EO/IR sensors (ship-based, sea-based, ground-based, space-based, and air-based) and thousands of local sensors with different bands and capabilities. The challenge and opportunity of the Sensor Resource Management (SRM) is related to a large difference in resolutions, errors and uncertainties of the sensors. There are situations where none of the sensors individually can improve certainty/resolution of the object to the required level. However, it is possible with assigning a pair of sensors with optimal lines of sights to the object. For instance, the range resolution of radars could be about 1 cm, but angular resolution could be only 1 km, thus two orthogonal radars could increase resolution significantly.

This leads to the tasks to optimize sensor resource use *across assets* in real time under the operational *constraints* of each sensor type, and to use for *planning.* Among the common SRM goals are: maximizing available sensor resources for *search*, optimizing sensor resources for *tracking,* and, *defending* better high priority assets in a raid environment.

The SRM goals often contradict each other. Consider typical goals: (i) decreasing the overall sensor resource utilization, (ii) increasing the probability that all threat objects in a raid are tracked, and (iii) decreasing potential overload of sensors at individual platforms/units. The chances that all these goals will not contradict each other and all will be satisfied by a particular solution (assignment of sensors) are low. This consideration leads to the necessity of **multi-objective optimization** approach, which is pursued in this work using the computational intelligence approach.

It is possible that the most resource utilization is a full 100% load of N sensors at unit A without any room for handling more areas of interests and objects, while only 10% of N sensors at the unit B are used. A more even use/allocation/tasking of sensors at units A and B, that decreases potential *overload* of sensors, may require more sensors, say 60% at unit A and 60% at unit B used at time t. In the first scenario (100:10) at some moment unit A may have not enough sensors to defend itself not only to track objects of interest, while in the second scenario (60:60) there is a plenty room for extra load, but more sensors are used. This is another reason for mlti-objective optimization,

There is extensive literature on sensor resource management [Borndörer, 2012; Hall, Llinas, 1997; Hero, Cochran, 2012; Hero. Keucher, Blatt, 2008; Hero, Castanon, Cochran, Kastella, 2007; Castanon, 1997; Junji et al, 2011; Kreucher et al, 2007, 2004-2005, Lambert, Sinno , 2011; Liggins , Hall, 2008; Patsikas, 2007; Perillo, Heinzelman, 2004; Smith, Nguyen, 2005-2006; Tian, Bar-Shalom, 2009; Weir, Sokol, 2009].

Information gain is one of most actively used approaches in SRM. The relative information gain is a *scalar* measure between the *prior and posterior probability density functions* $p(x_k|z_{1:k-1})$ and $p(x_k|z_{1:k})$ based on the Renyi o-divergence [Hero, Keucher, Blatt, 2008]

$$\mathcal{D}_\alpha\left(\mathbf{z}_{1:k} \,\|\, \mathbf{z}_{1:k-1}\right) = \frac{1}{\alpha-1}\log\int\left[\frac{p\left(\mathbf{x}_k \mid \mathbf{z}_{1:k}\right)}{p\left(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}\right)}\right]^\alpha p\left(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}\right)\, d\mathbf{x}_k.$$

To determine on which object a sensor should collect measurements during the next observation period k the algorithm in [Lambert, Sinno, 2011] adopts the strategy from [Manyika, Durrant-Whyte, 1994; Hero, Keucher. Blatt, 2008] that favor the object whose expected measurements would result in the *largest gain in relative information*:

$$\mu_\nu = \arg\max_\nu \left({}^\alpha\hat{\Delta}_k^{\mu\nu}\right),$$

where

$${}^\alpha\hat{\Delta}_k^{\mu\nu} = \int p\left(\mathbf{z}_k^{\mu\nu} \mid \mathbf{z}_{1:k-1}^{\nu}\right)\mathcal{D}_\alpha\left(\mathbf{z}_k^{\mu\nu}, \mathbf{z}_{1:k-1}^{\nu} \,\|\, \mathbf{z}_{1:k-1}^{\nu}\right)\, d\mathbf{z}_k^{\mu\nu}$$

is the expected local information.

This formulation that selects the object with the largest gain in relative information has an important *weakness of locality*: it computes the increase of certainty of the measurement (at time k) of a *given object* A by applying itself to a *single sensor* S relative to certainty of measurement at time k-1 recorded in the system track [Kreucher et al, 2007].

We may have a situation where none of the sensors individually will improve certainty/resolution of object to the required level as shown in Fig. 1 (a)-(b). However, reaching the required certainty is possible with assigning a pair of sensors (radars) with Lines of Sights (LOS) to the object that are close to be orthogonal as Fig. 1 (c) shows. In Fig. 1 the blue ellipse shows the original uncertainty of the location of the object at time k-1. The ellipses of uncertainty of the radars R_1 and R_2 are narrow ellipses that have a long intersections with the blue ellipse of the original uncertainty of the object location as shown in Fig. 1(a) and (b). Fig. 1(c) shows that the area of uncertainty due to overlap of uncertainty areas for R_1 and R_2 is much smaller with dramatic information gain. Note that individually both radars R_1 and R_2 equally and ch less improve relative information gain used in [Lambert, Sinno, 2011].

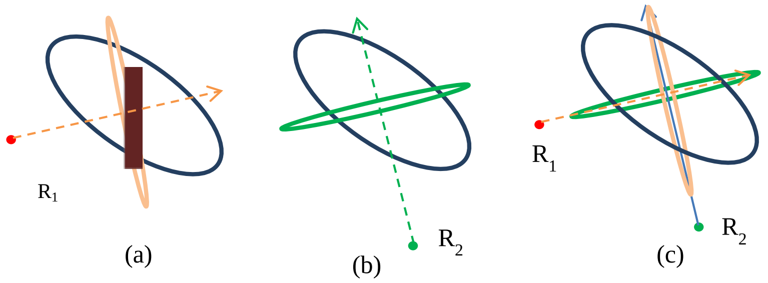


Fig. 1. Information gain with allocation of single radar vs. allocation of two radars to the object

The proposed approach goes beyond the state of the art described in [Lambert, Sinno, 2011] including the powerful idea of learning the parameter *a* in Renyi o-divergence. It also overcomes a potential *conflict between assignments* of the sensors to the objects in the *independent assignment* of "best" sensors for each object. In the independent sensor assignment a particular sensor can get conflicting "best" assignments to two or more objects that are at different locations that the sensor cannot cover at the same time.

In the information gain SRM formulation [Lambert, Sinno, 2011], the state estimation and SRM are *tightly coupled*. This leads to the complex models and computational challenges. In addition it is difficult to change one of the components in a tightly coupled formulation. For instance, it is difficult to add more optimization criteria beyond information gain and to incorporate the effect of a new *interaction* between the sensors. In general the progress in tracking, discrimination, fusion, and SRM technologies is not synchronous. Therefore the tightly coupled formulation makes it difficult to integrate these technologies into a more advanced system. Therefore we consider the state estimation itself as an *external* but connected task to SRM. The proposed new sensor tasking system OPTIMA includes a *Sensor Resource Analyzer* and the *Sensor Tasking models and algorithms* (Tasker).

The **requirements** for the SRM solution are:

1. Minimize the number of sensors for a given coverage, and maximize the effectiveness of each sensor given its performance and resource constraints.
2. Provide dynamic tasking of sensors where multiple sensors cooperate in search, detection, tracking, and identification.
3. Maximize the probability of successfully covering all threat objects.

The fusion center can provide requirements for SRM: desired resolution, angle, band, level of decrease of uncertain-ties in object characterization, and others. The challenge is that requirements (1)-(3) likely contradict each other which requires using a multi-objective optimization approach. The types of **questions** that the *Sensor Resource Analyzer* of the OPTIMA system intends to answer are:

- Will particular configuration C of platforms provide a full coverage of some areas of interests **A** with required capabilities **R** for tracking and discrimination?
- What is a *minimal configuration* **C** of platforms to provide a full coverage of some areas of interests **A** with required capabilities **R** for tracking and discrimination?
- What part of areas of interest **A** will not be covered at required capabilities **R** for tracking and discrimination if configuration **C** of platforms will be used?

Sensor coverage can be degraded due to multiple reasons at any time. Natural environments, engagement conditions, high noise background can impact radars and EO/IR sensors. As a result **degradation** can take multiple forms: inadequate signal to noise ratio, degraded specific range cells and azimuthal directions in the Field of View (FoV), too much energy on the focal plane of IR sensor, aspects of degradations compensated by the sensor itself.

Reconstructing the scene degradation from this varying information is a challenge. As a result inadequate input information can corrupt tracking and discrimination of objects. Information collected from widely distributed sensors can be used to determine areas where and how sensor coverage is degraded to allocate alternate resources to compensate. While individual sensors (EO/IR, Radar) can generally determine when a particular portion of the scene is degraded the challenge is in an effective use of this information for efficient sensor tasking. We consider tasking sensors in the degraded environment as a **generalization** of the sensor resource management (SRM) task **for a degraded environment**. This means that input **data messages** to the SRM describe not only normal, but also degraded sensor capabilities and degraded environment.

This paper is organized as follows. Section 2 presents the SRM optimization models. Section 3 presents translation of tracking and discrimination requirements to flags and solution of optimization models using Computational Intelligence techniques. The paper concludes with the description of the related and future work.

2 Approach and Optimization Models

2.1 OPTIMA system architecture with Computational Intelligence Solution

This work proposes a **new sensor tasking method** for both long-time planning and for real-time SRM based on Intelligent Multi-objective Modularized Optimization Model.

The SRM system OPTIMA and its context are illustrated in Fig. 2. The OPTIMA maintains timing constraints, resolution and geometric differences between the sensors relative to the tasking requirements on track quality and the measurements of object characterization quality. The solution is based on the computational intelligence approach that involves evolutionary methods, dynamic logic, and multi-objective optimization.

The system design allows a user to select the version of an objective function of the minimal configuration such as minimal number of platforms, minimal cost/value/capabilities of sensor platforms. In the version of the model presented below, it is assumed that all motions of sensor platforms are known, as well as the capabilities (possible degraded) and status of the sensors onboard the platforms.

The OPTIMA Model involves:

- multiple sensors of different types and with varying resolution and capabilities.
- sensor locations with respect to the object complex,
- timing constraints
- requirements for track quality, and
- requirements for measurements of object characterization (discrimination) quality

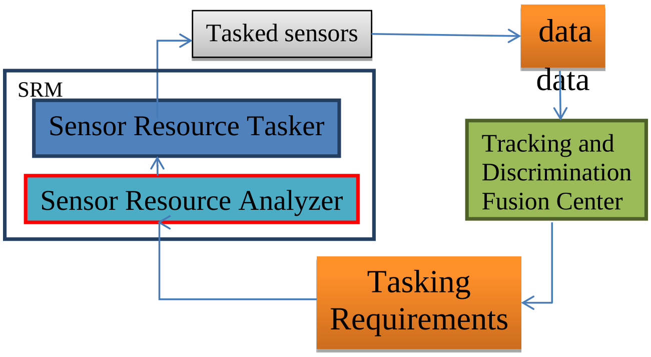


Fig. 2. Context of Sensor Resource Management

Tracking and discrimination may require different sensors and these requirements can change dynamically. Thus the model is updated with such new input, and the new output is produced as it is illustrated in Fig. 2.

The *uniqueness* of this approach is in use of *intelligent multi-objective optimization* of SRM Model with adaptable Integer Linear Programming (ILP) models, Cramer-Rao Bounds (CRBs) and algorithms accounting for association part of tracking and fusion problem. These CRBs able to *evaluate object characterization* (classification features) and therefore object values. Another *uniqueness* of the approach is in using *flags* within SRM, which encompass all the information external to the main goals/task (such as information from tracking algorithms). These flags are readily computed from available information or information adaptively estimated in real time. These *benefits* surpass existing state of the art and permit more accurate overall sensor coordination.

2.2 Modularized design

This work follows the *modularized design paradigm* where tracking, discrimination, fusion, and SRM as separate, but communicating modules to allow the SRM algorithm to work when tracking, discrimination and fusion algorithms are changed/upgraded.

To implement the modularized design, we build a set of integer linear programming (ILP) models described below with both continuous and binary variables that extensively use the concept of the *flags*. One of the flags is a binary flag, $f(a_i,s_j,t,r)$. If sensor s_j is *capable* of *covering/observing* area of interest a_i at the time interval t with the required resolution r, then $f(a_i,s_j,t,r)=1$, else $f(a,s,t,r)=0$. Another flag is a *stochastic flag*, which is a probability that sensor s_j is capable of covering/observing area of interest a_i at the time interval t with the required resolution r. Flags serve as a mechanism to *link* tracking, discrimination, fusion modules, models and algorithms with SRM optimization models and algorithms.

The advantage of this approach is that it allows separating: (i) rigorous formulation of the SRM optimization models (objective functions and constraints), (ii) multi-objective models that combine them, and (iii) feasible and fast computations for solving these models.

The OPTIMA system model tasking sensors in the both normal and degraded environment that is input data messages represent not only normal, but also degraded sensor capabilities and degraded environment description. For instance, instead of entering to the SRM systems the normal resolution r of the sensor S another value $r\text{-}\Delta$ is entered. Such generalization means that a set of additional input module/modules with appropriate algorithms need to be developed to generate degradations Δ. Such a module is shown in Fig.3 as a block with a red frame. Fig. 3 shows also the whole modularized ***architecture*** of the OPTIMA system.

This architecture assumes two types of algorithms: (A1) optimal sensor assignment for reducing sensor resource utilization using estimated performance and (A2) adaptive algorithm for working with the first algorithm to support input of sensor metadata including sensor “health” data, tracking, track correlation data, and data fusion data. The first algorithm exploits sensor geometry including a possibility that two sensors observing the same object (at approximately orthogonal geometry) could produce more accurate results in a shorter time.

The algorithm of the first type solves the problem of optimal sensor assignment. The algorithm of the second type provides the input data for the first one allowing using tracking and fusion algorithms that are at the level of the Cramer-Rao Bound (CRM) which sets up the best possible performance accounting for associations between sensors, objects, and tracks [Perlovsky 1997]. These algorithms exploit a novel technique [Deming, Perlovsky 2007; Perlovsky, Deming, 2013] developed for air-ground radars. In the case of the degraded environment CRM is computed for the degraded sensor environment by using appropriate sensor models in CRB computation.

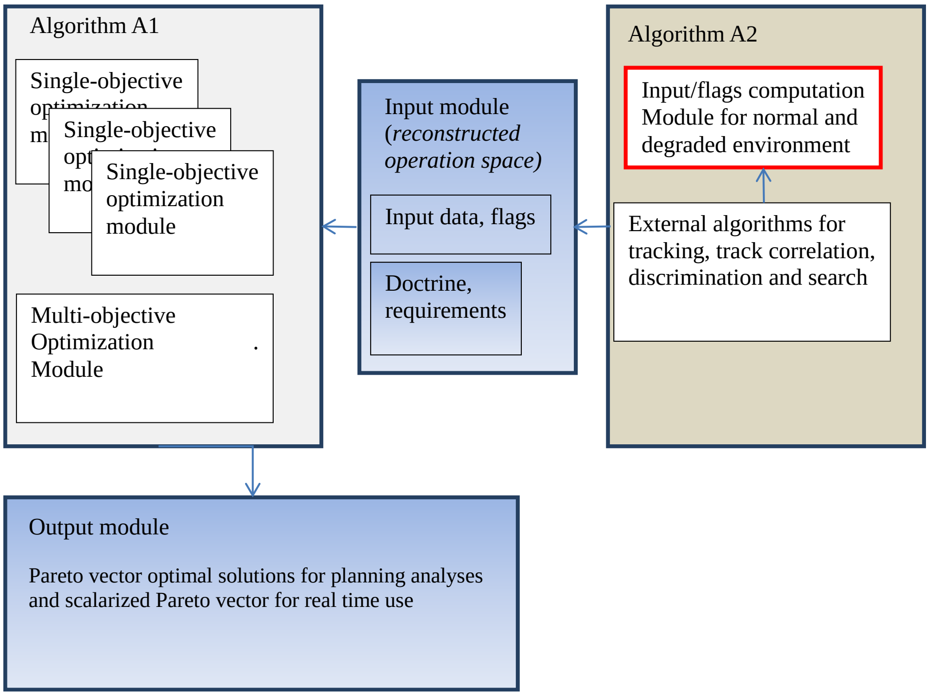

Algorithm A1
Single-objective optimization module
Single-objective optimization module
Single-objective optimization module
Multi-objective Optimization Module
Input module (*reconstructed operation space)*
Input data, flags
Doctrine, requirements
Algorithm A2
Input/flags computation Module for normal and degraded environment
External algorithms for tracking, track correlation, discrimination and search
Output module
Pareto vector optimal solutions for planning analyses and scalarized Pareto vector for real time use

Fig. 3. Modularized architecture of the SRM OPTIMA system with operation space characterization

This architecture combines mathematical techniques (multi-objective optimization with analysis of the Pareto border, Integer Liner Programming, Adaptive methods) and physical considerations (sensor phenomenology and geometry of locations relative to objects). It allows addressing contradictory requirements that cannot be addressed by the classical optimization methods without setting up a tradeoff between them in advance. However in the dynamic environment the tradeoff must be dynamic as well. Thus the optimization algorithm must adapt to such dynamic environment in real time which is proposed in this paper via dynamic updates of input data, looping optimization cycles and flexible selection of objective functions.

The ***main mathematical advantage*** of this architecture in comparison with the state of the art is that it *decouples* tracking and track estimation algorithms/filters from the optimization and environment estimation. The other advantages of a new architecture are that it allows:

(1) a *variety* of *external tracking and discrimination algorithms* by computing "flags" representing external algorithms,

(2) *multiple optimization criteria* by selecting/changing modules adapting for a particular scenario,

(3) *multiple tradeoffs between multiplicity of optimization criteria* in multi-objective setting providing a mathematically rigorous solutions.

These advantages make this modular architecture *more adaptable* and universal than other architectures, which is important for practical applications.

The important aspect of modularized design is that it computes flags (parameters of optimization models) outside of the optimization module and outside of tracking module as well. This is a *buffer* idea that has been very successful in many other areas including computer architecture design with cache memory as a buffer between the primary memory and CPU. Another example is separation/decoupling data from computational modules by database management systems. A user can change/fix the computational module without changing database (DB). Similar separation was made between a Knowledge Base (KB) and Computational modules (Reasoning Engines) in Artificial Intelligence.

The modular architecture separates the operation space characterization from both types of computations: sensor tasking optimization and object tracking and discrimination. Another advantage of this separation is clearer mathematical formulation of the SRM as an optimization task and use of the powerful techniques developed in this mathematical area for decades.

Many other popular SRM approaches are tightly integrated with Information Gain (IG) maximization that gives this single criterion dominance over the other criteria. Such *monocentric* approach has fundamental weaknesses. Maximization of the IG can contradict to minimization of the number of sensors and to the need to provide relatively even load to different sensors.

The new approach is free from tight integration while allowing using information gain approach too. Information gain criterion is very sensitive to the accuracy of estimates of probability distributions such as covariance matrixes, accuracy of tracking data association/correlation and so on. The proposed approach is free from such unrealistic assumption too.

2.3 Computational Intelligence Methodologies

Biologically-inspired computational intelligence approaches are promising for SRM modeling. One of the biological inspirations is coming from similarity between ultimate goals of SRM and the foraging/hunting models in mathematical ecology that have natural analogy with "hunting" targets. Foraging had already thousands of years evolution time that it much longer than SRM evolution.

Especially interesting here are tradeoffs between time and gain observed in natural foraging systems and respective models and objective functions that combines them. In one of the models [Verlinden, Wiley 1989; Van Gils, 2005] the predator attempts to maximize E/(h+s), where s is the search time involved. For a range of prey, the predators average intake rate is

$$\mathbf{E}_{\text{average}}/(\mathbf{h}_{\text{average}}+\mathbf{s}_{\text{average}}),$$

where $\mathbf{E}_{\text{average}}$ is the average energy of all prey items in the diet, $\mathbf{h}_{\text{average}}$ is the average handling time and $\mathbf{s}_{\text{average}}$ is the average search time.

In terms of SRM models the energy E is in line with a class of gain functions. The most interesting part of E is the tradeoff between handling time and search time. The weak part of many bio-inspired methods that they do not go further than shallow inspiration ending up with algorithms with little interpretation of tradeoffs borrowed from other fields like formula $\mathbf{E}_{\text{average}}/(\mathbf{h}_{\text{average}}+\mathbf{s}_{\text{average}})$ above.

Beyond the weakness of *justification* of the optimization criteria the search of the optimal solution is also *highly heuristic* in bio-inspired algorithms. Therefore we focus on the mathematically *rigorous* solution as a *benchmark* for the heuristic solutions that may have some benefits of faster computations and simplicity.

The key idea of the proposed approach is to combine computational intelligence techniques (multi-objective optimization based on the Pareto border, Integer Liner Programming under uncertainty, and Adaptive Learning methods) with physical considerations (sensor phenomenology and geometry of locations relative to targets).

The Computational Intelligence methodologies that are applicable to solve SRM OPTIMA models are evolutionary computing methods including adaptive multi-objective optimization that exploit genetic algorithms, colony optimization, particle swarm optimization, interval, stochastic and fuzzy optimization, and adaptive dynamic logic of phenomena.

Contradictory goals of SRM represented by multiple objective functions cannot be reached by the classical optimization methods without setting up a tradeoff between them and/or constrains in advance. However in the dynamic environment the tradeoff must be dynamic as well. Thus the optimization algorithm must learn and adapt to such dynamic environment in real time. This is challenging task with the growing number of assets, closely spaced objects and real time constrains for the algorithm runtime.

The main idea of the approach is borrowed from the **nature -- modularized optimization**. Such bio-inspired approach mimics cooperative team hunting/foraging in nature with abilities of dynamic learning, adapting and self-tasking.

The existing models in mathematical ecology [Pastor, 2008] such as Optimal Foraging Theory(OFT) and its Digestive Rate Model (DRM) of foraging that are now linked to the cumulative prospect theory (Tversky, Kahneman, Nobel Prize 2002) of human decision making under uncertainty are valuable sources of novel ideas to improve SRM models under conflicting objective functions. In particular, OFT and DRM deal with optimizing the tradeoff between foraging times.

The OPTIMA architecture assumes two types of algorithms:

- Algorithms (A1) that provide optimal sensor assignment (including deliberately reduced sensor utilization to be able to sensor more objects later) using estimated performance including degraded one and
- Algorithms (A2) that support input of sensor data and metadata to A1 including sensor “health” data, tracking, track correlation data, and data fusion data.

The first class of algorithms exploits sensor geometry including a possibility that two sensors observing the same object (at approximately orthogonal geometry) can produce more accurate results in a shorter time.

The algorithm of the first type solves the problem of optimal sensor assignment. The algorithm of the second type provides the input data for the first one allowing using tracking and fusion algorithms that are at the level of the Cramer-Rao Bound (CRM) which sets up the best possible performance accounting for associations between sensors, objects, and tracks [Perlovsky 1997]. These algorithms exploit a novel computational intelligence technique [Deming, Perlovsky 2007; Perlovsky, Deming, 2013] developed for air-ground radars, which have not been applied to the degraded scenarios.

The proposed SRM methodology for sensor planning includes:

- simulating different input data/scenarios (including different locations of assets and levels of degradation of sensors),
- observing and analyzing values of all objective functions of interest for best Pareto solutions.

Pareto solutions can be very different, e.g., with maximization of objective functions F_1 and F_2 the Pareto border may include pairs of their values (0.9, 03) and (08, 04). In summary the approach consists of: (1) Intelligent SRM models and algorithms, (2) Translation of tracking and discrimination requirements to flags for sensors in SRM model, (3) SRM model and algorithm for paired sensors.

2.4 Notation

Below we introduce the notations:

- T is a *track.*
- G is a *object.*
- U_T is a *track ambiguity descriptor* (*message*) at time t in the form of a model U_T (t) = $\langle A_T, \Omega_T \rangle$, where A_T is a set of *ambiguity characteristics of the track* T and Ω_T is a set of relations on A_T at time t.
- U_G(t) is a *object ambiguity descriptor* (*message*) *of the object* G at time t in the form of a model $U_G = \langle A_G, \Omega_G \rangle$, where A_G is a set of *ambiguity characteristics of the object* G and Ω_G is a set of relations on A_G at time t.
- The triple $E(t)=\langle U_T(t), U_G(t), C(t) \rangle$ is called a *sensing environment* at time t, where C(t) is a sensor model (a set characteristics of the sensors C such as locations, orientations, FOV, resolution, and others).
- M(E(t)) is a *vector of measures of environment degradation* E(t).

- $V(C_K)$ is an *environment* ***operator*** (*algorithm*) that produces a new environment $E_K(t)$, $V(C_K) = E_K$.
- K(E(t) is a ***tasking operator for sensors*** (*algorithm*) at environment E(t), that is K assigns a new set characteristics to the sensors C_K (sensor model). In these mathematical terms we need to design a tasking operator K to ***decrease the environment degradation*** that is $M(E(t))>M(E_K(t+1))$. Below the design of such operator is provided via a set of single-objective Integer Linear Programming (ILP) models and multi-objective optimization models on the top of ILP models as outlined in Figure 1.
- $\{a\}_j$ is a set of *Areas of Interests* (AOI) that sensor s_j can observe at time interval t. Each area of interest may contain a *Object Complex* (TC). For a search/scanning sensor only the area can be known, for a tracking sensor a object complex can be known. $a_{ij} \in \{a\}_j$ is a *marked area of interest* in $\{a\}_j$. Only *one* area is marked for each sensor. This marking can have different interpretations depending on version of sensor tasking. The examples of interpretations of marked areas are: (1) a marked area a_{1j} is the main focus area of sensor s_j that is assigned by the requirements and tasking objectives, (2) a marked area a_{2j} is an area that has the best viewing geometry from s_j, (3) a marked area a_{3j} is an area that has the best resolution from s_j at time interval t, (4) a marked area a_{4j} is an area that has the worst resolution from s_j at time interval t.
- r is a resolution of the sensor. Cramer-Rao Bound (CRB) is one of the ways to assign value to r, because CRB is the error of localization and tracking of a object with the given sensor. CRB gives the best possibly achievable resolution. Also CRB is useful for assigning resolution to discriminate RV and decoy. CRB can include coordinates and velocities, and in addition, classification features related to the classification/discrimination probability.
- v is a Field of View (FOV) of a sensor at a particular time.
- $f(a_i,s_j,t,r)$ is a binary flag, $f(a_i,s_j,t,r)=1$, if sensor s_j is *capable* to *cover/observe* area of interest a_i at the time interval t with required resolution r, else $f(a,s,t,r)=0$.
- $f_s(a_i,s_j,t,r)$ is a stochastic flag, $f(a_i,s_j,t,r) \in[0,1]$. It is a *probability* that sensor s_j is *capable* to *cover/observe* area of interest a_i at the time interval t with required resolution r.
- $p(s_j,t,v)$ is a *time to point* sensor s_j to get FOV v by time t. This requires changing its line of sight (LOS).
- $f^*(a_i,s_j,t,r,v)$ is a binary flag, $f^*(a_i,s_j,t,r,v)=1$, if $f(a_i,s_j,t,r)=1$ and a_i is a marked area a_{ij} of sensor s_j at the time interval t within FOV v, else $f^*(a_i,s_j,t,r,v)=0$. In other words, If sensor s_j observes several AOIs within FOV v then only the marked AOI will get flag $f^*(a_i,s_j,t,r,v)=1$, other AOI a_k within the same FOV v of sensor s_j will get $f^*(a_k,s_j,t,r,v)=0$. This flag will be used in the optimization task formulation to minimize the number and cost of required sensors.
- $f^*_s(a_i,s_j,t,r,v)$ is a *stochastic* flag (*probability*), $f^*_s(a_i,s_j,t,r,v) = f_s(a_i,s,t,r)$ if and a_i is a marked area a_{ij} of sensor s_j at the time interval t within FOV v, else $f^*(a_i,s_j,t,r,v)=0$.
- $\{s\}_i$ is a set of sensors that can observe AOI a_i at time interval t.
- $s_{ij} \in\{s\}_i$ is a *marked sensor* in $\{s\}$. More than *one* sensor can be marked for each area. This sensor can have different interpretations depending on a version of

sensor tasking interest. The examples of interpretations of marked sensors are: (1) sensors s_j that are assigned to AOI a_i by the requirements and tasking objectives, (2) sensors that have the best viewing geometry to AOI a_i, (3) sensors that have the best resolution of AOI a_i at time interval t, (4) sensors that have the worst resolution from s_j at time interval t, (5) Aegis radars, and others.

- $\{g_l(s_j)\}$ is a set of binary flags, $g_k(s_j)=1$ indicates that sensor s_j is of *type/category* g_k, else $g_k(s_j)=0$. The tasking requirements may include specific types of sensors, e.g., $g_1(s_j)=1$ can indicate that s_j is a staring sensor, $g_2(s_j)=1$ indicates that s_j is a scanning sensor (or scanning mode of the sensor), and $g_3(s_j)=1$ indicates that s_j is a sensor on a specific platform.
- $\{h_k(s_j,s_q)\}$ is a set of binary flags, $h_k(s_j,s_q)=1$ indicates that sensors s_j and s_q have a specific relationship, e.g., $h_1(s_j,s_q)=1$ can indicate that the angle between their LOSs at time t is in the interval $[75^0,105^0]$ that is close to orthogonality.
- x_{ijt} is a *binary variable*, x_{ijt} =1 indicates that sensor s_j is *tasked* to observe area a_i for the time interval t, else x_{ijt} =0. Finding values of x_{ijt} is the goal of optimal sensor tasking.
- $\{c_j\}$ is a set of *objective function coefficients*. The interpretation of these coefficients depends on the specification of the sensor tasking objectives. The examples of interpretation of $\{c_j\}$ are: (1) costs of the sensors $\{s_j\}$ and their platforms (2) pointing time, (4) errors of sensors (e.g., covariance matrixes of LOS), (5) capability characteristics of sensors such as resolution or sensitivity of the sensors (6) information gain that the sensors add relative to the current knowledge of the object complexes, and others.

Coefficients of types (1)-(4) lead to minimization models (min of number of sensors, min of cost, min of pointing time, min of errors). Coefficients of types (5)-(6) lead to maximization models (max of capabilities, max of information gain). In essence we have two opposite categories of objective functions: ***cost*** (1-4) and ***gain*** (5-6) with wide interpretations of costs and gains. It is not required to interpret them literally. The multi-objective approach will allow seeing the optimal value of each objective function in the context of the values of other objective functions before a tradeoff between objective functions is made. In contrast the popular weighting approach combines such objective functions to their weighted sum “in the dark” without such analysis

The **Pareto multi-objective SRM model** that combines models with these objective functions allows the analysis of the *optimal value* of each objective function, in the context of the values of other objective functions, *before a tradeoff* between objective functions is made. In contrast, the popular weighting approach combines objective functions into their weighted sum “*in the dark*” without such an analysis. The detailed elaboration of multi-objective SRM model is a topic of a separate paper.

2.5 ILP Models for time t

Consider optimization *objective functions for a fixed time*:

$$ext\sum_{i=1}^{n}\sum_{j=1}^{m} c_j \cdot f^*(a_i, s_j, t, r, v) \cdot x_{ijt}$$

where *ext* (extremum) stands for max or min with the constraints presented below.

Coverage constraints (all areas of interest {a$_i$} must be covered at least by one sensor):

$$\sum_{j=1}^{m} f(a_i, s_j, t, r) \cdot x_{ijt} \geq 1, \quad i = 1,2, \dots, N$$

Constrains (all variables that assign sensors to areas at time t must be binary; sensor either assigned or not to the AOI):

$$x_{ijt} \in \{0,1\}, \qquad i = 1,2, \dots, N; j = 1,2, \dots, M$$

Sensor types constraints (sensors of all required types g$_1$,g$_2$,...,g$_K$ must be used):

$$\textstyle\sum_{j:\, g_k(s_j)=1} x_{ijt} \geq 1,\ i = 1,2, \dots, N,\ k = 1,2, \dots, K$$

Sensor relationship constraints (sensors with all required relations h$_1$,h$_2$,...,h$_L$ must be used in required quantities H$_{il}$ for each AOI a$_i$),

$$\textstyle\sum_{j:\, h_l(s_j, s_q)=1} x_{ijt} \geq H_{il},\ i = 1,2, \dots, N,\ l = 1,2, \dots, L$$

If {c$_j$} are *costs* of observing all areas a$_i$ with object complexes, then we use the first objective function and minimize the total cost of observation under constraints. This will provide the least expensive solution under constraints. If costs of all sensors are considered to be equal then in fact this objective function minimizes the *total number of sensors*. Similarly if {c$_j$} are *capabilities* of the sensors, then we minimize total capabilities under constraints.

If {c$_j$} are *information gains* then we use the second objective function and maximize the total information gain. Flags $f^*(a_i, s_j, t, r, v)$ in the objective function allow to optimize the use of sensors for the situations when some individual sensors observe several object complexes within a single FOV avoiding double counting such sensors.

The coverage constraints require that at least one sensor will be *tasked* to observe/*cover* each area a$_i$ with the required resolution of observation. The sensor type constraints require that at least one sensor of each required type will be used. These sensor type constraints can be generalized by substituting 1 on the right side of the inequality with another required number of sensors of type g$_k$.

The relationship constraints allow the incorporation into the optimization model of multiple desired geometric relationships between the locations of sensors and objects, such as orthogonality relative to the object. If there are no sufficient resources to track each objects with two sensors, then we can choose which objects require two sensors, and which will be tracked with one sensor. The models also allow dedicating only a small amount of time of the second sensor for tracking the same object, which significantly increases the resolution in terms of Cramer-Rao Bounds (CRB) as a function of two parameters: time on object for a sensor from platform 1, and time on object for a sensor from platform 2. This optimization framework allows highly modular SRM where some set of modules is responsible for computing all flags and updating them.

2.6 ILP Model for larger time interval

The optimization model described above assumes a *short time interval* where all flags do not change their values significantly. For the larger time intervals this assumption is not true, the values of flags change dynamically within larger time intervals. This leads to the model modification with additional summation for all time moments from t=1 to T:

Objective functions for a longer time interval:

$$ext \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{j=1}^{m} c_j \cdot f^*\left(a_i, s_j, t, r, v\right) \cdot x_{ijt}$$

with the constraints presented below. If $\{c_j\}$ are *costs* of observing all areas a_i with object complexes, then we use the first objective function and minimize the total cost of observation under constraints. This provides the least expensive solution under constraints. If costs of all sensors are considered to be equal then in fact this objective function minimizes the *total number of sensors*. Similarly if $\{c_j\}$ are *capabilities* of the sensors, then we minimize total capabilities under constraints.

If $\{c_j\}$ are *information gains* then we use the second objective function and maximize the total information gain. Flags $f^*(a_i, s_j, t, r, v)$ in the objective function allow optimizing the use of sensors for the situations when some individual sensors observe several object complexes within a single FOV avoiding double counting such sensors.

Coverage constraints (all areas of interest $\{a_i\}$ must be covered at least by one sensor):

$$\sum_{j=1}^{m} f(a_i, s_j, t, r) \cdot x_{ijt} \geq 1, \quad i = 1,2, \dots, N, t = 1.2.,,,T$$

Binary constrains (all variables that assign sensors to areas at time t must be binary; sensor either assigned or not to the AOI):

$$x_{ijt} \in \{0,1\}, \quad i = 1,2, \dots, N; j = 1,2, \dots, M, \quad t = 1.2.,,,T$$

Sensor types constraints (sensors of all required types g_1, g_2,...,g_K must be used):

$$\sum_{j:\, g_k(s_j)=1} x_{ijt} \geq 1, \qquad i = 1,2,\dots,N, \qquad k = 1,2,\dots,K, \qquad t = 1.2.,,,T$$

Sensor relationship constraints (sensors with all required relations h_1,h_2,...,h_L must be used in required quantities H_{il} for each AOI a_i),

$$\sum_{j:\, h_l(s_j,s_q)=1} x_{ijt} \geq H_{il}, \qquad i = 1,2,\dots,N, \qquad l = 1,2,\dots,L, \qquad t = 1,\dots,T$$

The coverage constraints require that at least one sensor will be *tasked* to observe/*cover* each area $a_{i,}$ with the required resolution of observation. The sensor type constraints require that at least one sensor of each required type will be used. These sensor type constraints can be generalized by substituting 1 on the right side of the inequality by another required number of sensors of type g_k.

The relationship constraints allow incorporating into the optimization model the multiple desired geometric relationships between locations of sensors and objects, such as orthogonality. If there are no sufficient resources to track each of the objects with two radars, then the algorithm selects objects requiring two sensors, using attained accuracy and object classifications. The model also allows dedicating only a small amount of time of the second radar for tracking the same object, which will also significantly increase track resolution and object characterization.

This optimization framework allows the highly modular SRM using modules that compute flags.

2.7 Stochastic ILP Model for larger time interval

The optimization models described above assume that deterministic flags in the objective functions. Below we present stochastic versions of objective function that can explicitly capture *uncertainty* of the operation space situation. It requires changing deterministic flags f* to stochastic flags f^*_s in objective functions and constraints:

$$ext \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{M} c_j \cdot f_s^*(a_i, s_j, t, r, v) \cdot x_{ijt}$$

with the constraints presented below.

Coverage constraints (all areas of interest $\{a_i\}$ must be covered with confidence F_i or greater:

$$\sum_{j=1}^{m} f_s^*\big(a_i, s_j, t, r, v\big) \cdot x_{ijt} \geq F_i, \quad i = 1,2,\dots,N, \qquad t = 1.2.,,,T$$

Binary constrains (all variables that assign sensors to areas at time t must be binary; sensor either assigned or not to the AOI):

$$x_{ijt} \in \{0,1\}, \qquad i = 1,2,\dots,N; j = 1,2,\dots,M, \;\; t = 1.2.,,,T$$

Sensor relationship constraints (sensors with all required relations $h_1,h_2,\dots,h_L$ must be used in required quantities H_{il} for each AOI a_i),

$$\sum_{j:\, h_l(s_{j,}s_q)=1} x_{ijt} \geq H_{il},$$
$$i = 1,2,\dots,N, \;\; l = 1,2,\dots,L\,, t = 1,\dots,T$$

Sensor relationship constraints (sensors with all required relations $h_1,h_2,\dots,h_L$ must be used):

$$\sum_{j:\, h_l(s_{j,}s_q)=1} x_{ijt} \geq 1,$$

$i = 1,2,\dots,N, \;\; l = 1,2,\dots,L,$ t=1,2,…,T

2.8 Sensor message constructs and operation space reconstruction

The algorithm for sensor message constructs is as follows:

1. Select the sensor composition (e.g., two radars of one type, two radars of another type, and a constellation of 12 EO/IR sensors).
2. Select a representative operation scenario with sensors listed in (1) above.

Describe these sensors in terms of:

a. U_T (*track ambiguity descriptor/ message*) at time t in the form of a model U_T(t) = ⟨A_T, Ω_T⟩, where A_T is a set of *ambiguity characteristics of the track* T and Ω_T is a set of relations on A_T at time t;

b. U_G(t) (*object ambiguity descriptor/message*) *of the object* G at time t in the form of a model U_G = ⟨A_G, Ω_G⟩, where A_G is a set of *ambiguity characteristics of object* G and Ω_G is a set of relations on A_G at time t;

c. E(t)=⟨U_T(t), U_G(t), C(t)⟩ triple (*sensing environment* at time t), where C(t) is a sensor model (a set characteristics of the sensors C such as locations, orientations, FOV, resolution, “health”, and others);

d. M(E(t)) (*vector of measures of environment degradation* E(t));

e. V(C_K) (*environment **operator**/algorithm*) that produces a new environment E_K(t), V(C_K) = E_K.

3. Describe the items listed in step 2 in the form of messages with a specific format.

The example of the messages going to the *sensing environment* at time t, E(t)=⟨U_T(t), U_G(t), C(t)⟩ is a binary or numeric flag f_{Energy} that is too much energy are on the focal plane. It is accompanied by additional flags that indicate the consequences of this degradation such as f_{SNR} that indicates the decreased SNR, $f_{detection}$ that indicates decreased detection sensitivity, f_{range} that indicates that the range is uncertain, f_{angle} that indicates that the angle and pointing vector are is uncertain.

In the notation section several flags have been introduced. These flags are used to define sensor messages and are parts of the messages.

The binary flag $f(a_i,s_j,t,r)$ indicates whether sensor s_j is *capable* to *cover/observe* area a_i at the time interval t with required resolution/capability r. This flag is computed directly for each value of its variables: the identified areas a_i, sensor s_j, and required resolution/capability r at time t. LOS and FOV with their errors for sensor s_j at time t, location of the AOI a_i are used to check if a required resolution/capability can be reached. If the AOI a_i contains a detected moving object then the known dynamic properties of the trajectory of the object are used to identify the next location of the AIO a_i and to compute flag f for time t+1. These properties are derived from the external tracking, track correlation algorithms. Similarly each sensor resolution relative to objects are derived, other capabilities and their adequacy relative to required ones are computed.

The flag $f^*(a_i,s_j,t,r,v)$ is directly computed by the algorithm from $f(a_i,s_j,t,r)$, when an area a is marked for a sensor s_j. The marking of the area is identified from tasking requirements and tasking doctrine. Flags $\{gk(s_j)\}$ that indicate types of sensors are computed from the database of specifications of sensors. Relations flags are computed based on the definitions of the relations. In some cases all flags are specified in advance. In other cases these flags are computed to achieve the required resolution of the sensor system. This includes Monte-Carlo Simulation and Cramer Rao Bounds (CRB). The CRBs are computed for each object and sensor, as well as for each object and a pair of good and degraded sensors as required. If required it is done by using tracking and fusion algorithms outlined below.

These flags can tell a story: ‘we were unable to track and/or characterize the object without excess ambiguity’. This information is used to guide SRM to find a sensor which can view the object complex, from a less impacted viewing angle, or in a wavelength that is less susceptible to the degrading agent in the SRM models.

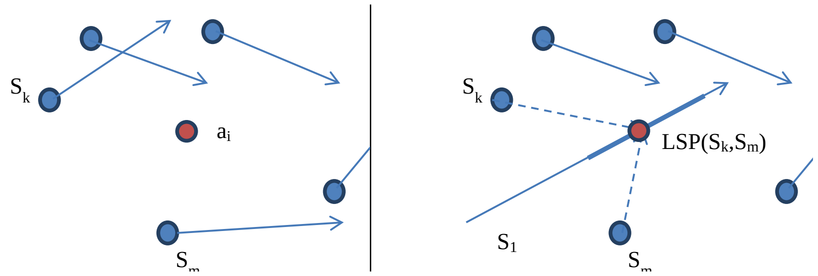
2.9 SRM Model with Orthogonalization

Consider sensors $S_1, S_2,\ldots, S_n$. Assume that for each sensor we know the sensor model M which includes its location, orientation, and capabilities (FOV, resolution, band, response time, and others). Thus we have models $M(S_1), M(S_2),\ldots,M(S_n)$.

Assume that there is no single sensor that provides required resolution for the area of interest a_i. In this situation we search for a pair of sensors (S_k, S_m) that will have the angle between their LOSs closer to 90^0 than any other pair of sensors when pointed to the center of a_i (See Fig. 4). In other words, we order all pairs of sensors according to the dot products of vectors of LOSs and search for the OPIR sensors with max of dot products,

$$\text{Arg max}_{\,i=1,n,\ k=1:n}\,(LOS(S_i)\cdot LOS(S_k))$$

After all these pairs of sensors are found for all areas of interests the algorithm checks their consistency that is no sensor assigned to two or more areas. If such in-consistency is found it is resolved by removing a conflicting sensor from the pair with the least value of dot product and assign the best sensor to this pair that is not used yet for any object. If the set of such sensors is empty then wait until the busy sensor will be released from the area ai, or use the round robin method.



Sensors with current LOSs

Orthogonality test

Fig. 4. Orthogonality test for pairs of sensors

If the center of the area a_i is not known or this area is very large then the algorithm searches a pair of sensors (S_k, S_m) that has "*better*" angles between their *possible* LOSs relative to several subareas within area of a_i than any other pair of sensors. The angles are "better" if they correspond to the *largest sets of points***,** LSP(S_k, S_m) in the subarea where the angle between LOS(S_k) and LOS (S_m) is closer to 90° than for any other pair of sensors ("*orthogonality*" *test*).

This base algorithm can be enhanced to deal with motion of platforms and areas of interests. Specifically the subarea can be selected based on the tracking of the object in the area a_i as shown in Fig. 4. Tracking by nearly orthogonal sensors such as radars and EO/IR brings significant accuracy improvement.

2.10 Multi-objective SRM optimization

To resolve the ***contradictory goals*** such as decreasing the overall sensor resource *utilization*, increasing the *probability* that all threat objects are tracked, and decreasing potential *overload* of sensors at individual platforms/units the ***multi-objective optimization approach*** is used.

The multi-objective optimization model is built on a set of S*ingle-Objective Tasks* (SOT) as shown on Fig. 1. Let F_1,F_2,…,F_n are objective functions of respective single-objective SRM tasks.

The optimal solution of SOT_1 provides the value f_1 of F_1. We also can compute values $f_{i1}$of F_2,F_3,…, F_n for this solution for SOT_1. This produces a vector (f_1, f_{21}, f_{31},…f_{n1}). Similarly values f_i and associated values f_{ij} are produced for all other SOT_i with objective functions F_i. Each of these vectors constitutes a *vector solution*. The Pareto border is a set of all vector solutions that cannot be improved.

Consider an example with two vector solutions {(0.6; 0.2), (0.2; 0.8)} for two objective functions, F_1 and F_2 that are maximized. Here the best solution relative to objective function F_1 is 0.6, but it is very weak (0.2) relative to F_2. Similarly, the best solution for the F_2 is 0.8, but it is very weak (0.2) for $F_{1.}$ A solution (0.5; 0.5) may exist and be a good tradeoff between these two solutions. These vectors are a part of the Pareto border.

The advantage of the Pareto approach is that we analyze a much wider set of possible solutions than a set of "optimal" solutions provided by scalar cost functions such as based on information gain [Kreucher et al., 2007]. In essence, it is *better to introduce cost functions within the Pareto set than without it*. The same Pareto approach we use for the discrimination task and for the combination of the tracking and discrimination.

An innovative approach based on the analysis of T-norms [Kovalerchuk, 2010] allows coming to the optimal solution by combining several objective functions. To move from the Pareto boundary to a ***trade-off solution***, multiple fusion (aggregation) operators have been proposed. The class of fusion operators used in fuzzy logic for membership functions is known as T-norms. T-norms can distort the Pareto order property dissolving the important difference between the nodes [Kovalerchuk, 2010]. For example, suppose that W= { (0.0; 0.5) , (0.2; 0.8) , (0.6; 0.2) } then the best points (Pareto points) are P = { (0.2; 0.8) , (0.6; 0.2) }. First, we need to know that T-norms do not contradict the Pareto optimum. In fact, a T-norm such as popular in fuzzy logic min can add new 'best" points that do not belong to the Pareto optimum.

Consider W= { (0.0;0.5) , (0.2;0.8) , (0.6;0.2), (1.0;0.8) , (0.9;0.8)}. Here Pareto optimum includes only (1.0; 0.8), but the T-norm as minimum gives also (0.9; 0.8) as a best point which is wrong. Now we see how the lack of interpretability is translated into a lower accuracy of the solution.

Setting up a trade-off preference relation H between alternative vectors (nodes) in the Pareto set must be consistent with meaningful preference of assigning sensor to objects. Unfortunately relation H rarely is known completely. Each T-norm serves as a compact approximation of H. However a T-norm can be far away from modeling H satisfactory. It is desirable that T-norms preserve the strict order for all pairs (x,y) that is

$$(x,y) < (v,u) \Rightarrow T(x,y) < T(v,u).$$

However this is true only for some pairs, e.g.,

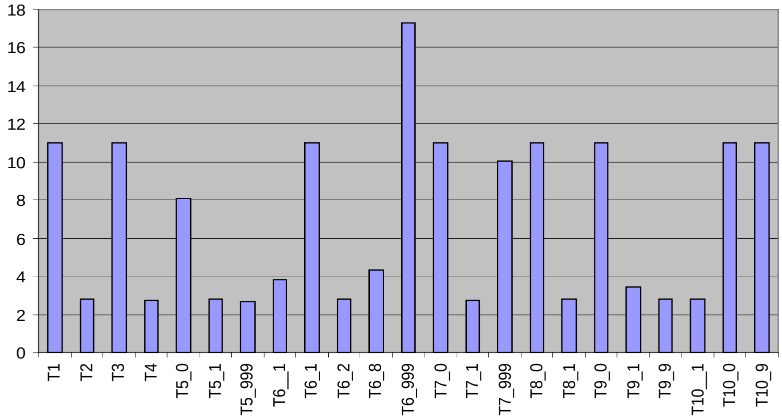
$$(0.3;0.5) < (0.4;0.7) \Rightarrow \min(0.3;\ 0.5) < \min(0.4;\ 0.7),$$

but it is not true for (0.3; 0.5) < (0.4; 0.7), where min(0.3; 0.5) = min(0.3; 0.7). Thus, T-norms can distort the order property dissolving the important difference between the nodes.

We measure this distortion by introducing a *Pareto set distortion factor* k_2 and use **least distorted** T-norms. Factor k_2 computes a ratio of two numbers m and h = r(r-1)/2: where m is the *number of unequal pairs* of nodes of the lattice that have equal T-norm values and (2) h= r(r-1)/2 is the *total number of different pairs of nodes* of lattice L that has r nodes. Next we use the *quantified Pareto set distortion factor* k_3 that is modified factor k_2 to address the requirement (3) of sufficient power of scale given by T-norm [Kovalerchuk, 2010],

$$k_2 = \frac{2m}{r(r-1)} \qquad k_3 = \frac{\sum_{i=1}^{11} \binom{2}{m_i}}{\binom{2}{r}}$$

where r is the same as for k2 and mi is the number of nodes in the i-th subinterval of the lattice L. The i-th component of the sum in k_3 gives the number of "glued" pairs nodes of the lattice in the subinterval i. Fig. 5 shows lattice distortion factor k_1 by different T-norms.



.

Fig. 5. Distortion factor k_1 for considered T-norms

For speeding up computations of the Pareto border, we use the theory of Monotone Boolean Functions [Kovalerchuk et al, 1996]. The main idea of cutting the computation time is finding attributes that are relatively independent and as such they can be processed relatively independently in parallel.

2.11 Degraded sensors and environment

Helping degraded IR sensor with a single IR sensor. Fig. 6 shows a case when the first EOIR sensor is degraded and the second sensor can be used instead of the first one. The second EOIR sensor has a better viewing geometry. However, the first degraded EOIR sensor S_1 cannot guide the second sensor S_2 how to change its FOV, because the first sensor has no range information to the object complex. It has only 2-D directional information of the line of sight (LOS).

Consider a scenario where at time t sensor S_1 observes an Area of interests (AOI) a_i that contains an object complex. The 3-D center of the a_i is already identified. For instance, it can be done jointly by this sensor S_1 and another sensor S_2 that was available at time t. At the next time t+1 sensor S_1 is downgraded to the level that another sensor S_3 should substitute it to continue accurately tracking the object complex in a_j.

The second sensor needs to scan the FOV of the first sensor to find the Area of Interest (AOI) that contains objects.

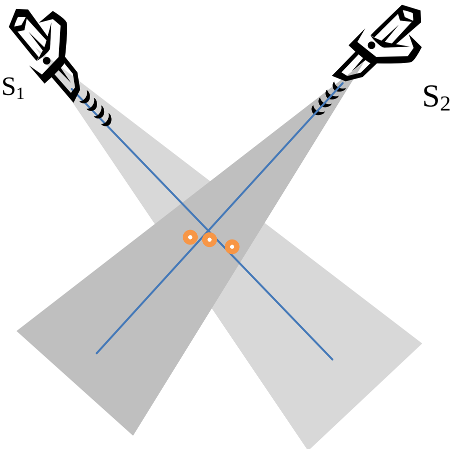


Fig. 6. Situation with a better viewing geometry for the second sensor and lack of guidance from the degraded first sensor on the location of the object complex

Algorithm outline:

Step 1: Search for a substitute sensor S_3 that has the same or very similar capabilities as s_1 and similar geometry relative to the AOI a_j. It is identified by the thresholds on differences in distances and FOVs,

$$\text{Capab}((s_1) \approx \text{Capab}(s_3), \quad \text{Dist}(s_1,s_3) < T_{\text{distance}}\,, \quad \text{diffFOV}(s_1,s_3) < T_{\text{diffFOV}}$$

Step 2: If Step 1 did not find a sensor S_3 that satisfy the requirements of Step 1 then the search is conducted under modified relaxed requirements. Sensor S_3 should satisfy only specific capabilities requirements such as resolution. Let R_1 be a resolution that sensor S_1 provides for area a_j. Another sensor can be a more powerful sensor located further from a_j than S_1, but it still can provide resolution R_1,

$$\text{Resolution}(S_3,a_j) \geq \text{Resolution}(S_1,a_j) = R_1.$$

Additional requirements can be imposed on SNR and on differences in FOV: $\text{diffFOV}(S_1,S_3) < T_{\text{diffFOV}}$. The dynamic adjusting requirements can be modeled in accordance with the Dynamic logic process [Kovalerchuk, Perlovsky, Wheeler, 2012].

Helping degraded IR sensor by selecting two IR sensors. Consider a degraded sensor S_1 and needs to be substituted by other sensors and there is no simple solution, that is there is no other sensor with very similar or better capabilities that can be quickly reoriented to the same area as S_1. The algorithm searches for a pair of sensors (S_k, S_m) that are able to substitute sensor S_1. The idea of the search algorithm is to find (S_k, S_m) with Lines of Sight (LOS) closest to LOS of S_1 within 90^o limits.

$$\forall\ S_i \neq S_k\ ,\ S_i \neq S_m\quad 90^o \geq \|LOS(S_1)\text{-}LOS(S_i)\| \geq\ \|LOS(S_1)\text{-}LOS(S_k)\|\quad \&$$
$$90^o\ \geq\ \|LOS(S_1)\text{-}LOS(S_i)\| \geq\ \|LOS(S_1)\text{-}LOS(S_m)\|$$

In other words we order all sensors according to the dot products of vectors of LOS and search for the sensors with max of dot products,

$$\text{Arg max}_{\ i=2,n}\,(LOS(S_1)\cdot LOS(S_i))$$

In addition a pair of sensors (S_k, S_m) should have "*better*" angles between their *possible* LOSs relative to LOS(S_1) than any other pair of sensors. The angles are "better" if they correspond to the *largest sets of points***,** LSP(S_k, S_m), on the LOS(S_1) where the angle between LOS(S_k) and LOS (S_m) is closer to 90^o than for any other pair of sensors (**"orthogonality" test**).

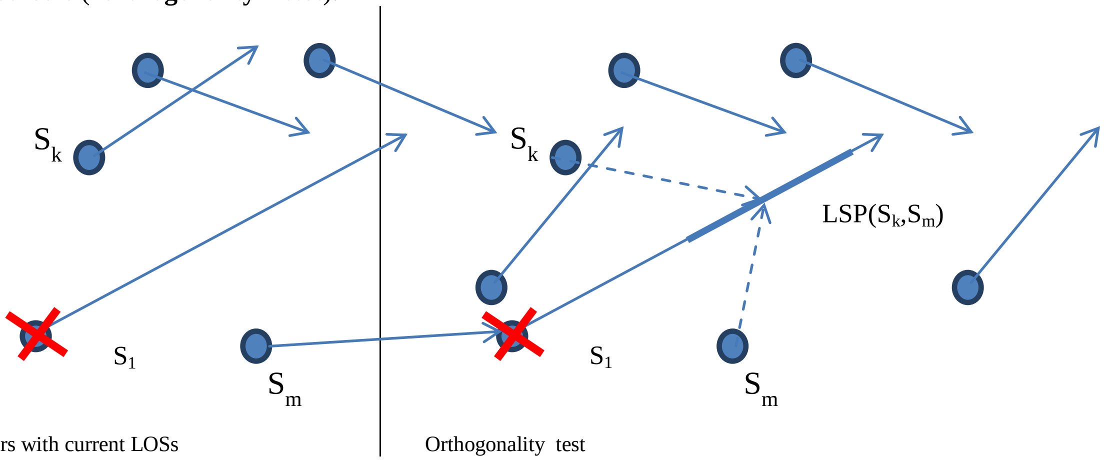


Fig. 7. Orthogonality test in degraded environment

Now assume that sensor S_1 is *degraded partially***,** that is some its information is useful not only its LOS. We have already assumed that LOS of S_1 carries useful information, e.g., there are some object detections in its FOV with the given LOS. This means that it makes sense to continue to observe the environment in the area captured by FOV of S_1. Let the additional uncorrupted information from S_1 be the *Direction to the Cluster of Objects* (DCT), that is we have vector DCT(S_1) in addition to LOS(S_1). Now we can search for the pair of sensors (S_k, S_m) that have *largest sets of points***,** LSP(S_k, S_m) on DCT(S_1), not only on the LOS(S_1). This increases the accuracy of the information that (S_k, S_m) provides (see Fig.7).

If we have an uncorrupted DCT(S_1) only in the part of the FOV of $S_{1,}$ then a solution based on DCT is used in this part of the FOV and the solution based on the LOS(S_1) is used in the corrupted part of FOV. In the case of multiple directions to the Cluster of Objects for sensor S_1 the optimal pairs of sensors are computed for each direction.

Helping degraded radar by selecting two IRs. Now consider a situation when S_1 is a degraded radar, or a radar that observes a too-dense scene and sensors $S_2,S_3,\ldots,S_n$ are EOIR sensors that can be used to help or substitute S_1, depending on the level of degradation or complexity of the scene. Assume that radar gives 3-D location L of the object with acceptable accuracy to point an EOIR sensor. The algorithm searches for the EOIR sensor with LOS that is most close to 90^0 to the radar LOS, and closest to the location L to provide a better object resolution,

$$\text{Arg min}_{\,i=2,n}\,(\text{LOS}(S_1)\cdot\text{LOS}(S_i)),\ \ \text{Arg min}_{\,i=2,n}\,(\text{L(T)},\text{L}(S_i)).$$

For two IRs that we consider here the formulation is similar to Task 3.2 relative to orthogonality of IRs.

Selecting two Radars to help degraded IR. Of the two radars, choose the one the LOS of which is closest to 90^0 to the LOS of IR. If location of the object is too uncertain (from IR data), divide the IR-LOS into two pieces, one - best for the radar 1, and part two - best for the radar 2.

The worst case is when a degraded sensor S_1 does not provide any useful information. In this case LOS (S_1) is degraded too. This task is equivalent to a general task of tasking n-1 sensors and has very little specifics relative to the general sensor management task to be exploited efficiently. This task is out of the scope of this paper.

3 Computational Intelligence Solution for SRM Model

3.1 Exact and heuristic algorithms

The proposed above ILP SRM models require efficient algorithms to solve them. The classical Linear Programming models can be solved by the simplex method for relatively large N and M. For instance, 15 OPIR sensors (N=15) and 15 areas of interests (M=15) lead to 225 variables for each time interval t.

The proposed ILP SRM tasks are NP-hard problems that cannot be solved exactly when the number of parameters is large. For smaller number of parameters multiple computational methods solve it exactly [Junger, et al., 2010]. Thus, depending of the size of the SRM model and time constraints to solve it (planning or real-time tasking) exact, approximate or heuristic methods are needed.

Multiple generic heuristic can solve ILP SRM tasks: (1) Tabu search, (2) Hill climbing, (3) Simulated annealing, (4) Reactive search optimization, (5) Ant colony optimization, (6) Hopfield neural networks, (7) Genetic algorithms and others. The last three classes of methods are effective Computational Intelligence methods that have been successful in SRM [Burgess, Levins, 2003; Hanlon, Cohen, Kivelevitch, 2015; Severson, Paley 2014; Shea, Kirk, Welchons, 2009]

The development of specialized algorithm is also advantageous for the SRM to get a better performance by taking into account specifics of the IL SRM models. This includes using bio-inspired methodologies of Dynamic Logic [Kovalerchuk, Perlovsky, Wheeler, 2012; Kovalerchuk, Perlovsky, 2008-2009; Perlovsky, Deming, Ilin, 2011]

The exact algorithms to solve SRM task include: (1) cutting plane methods (solving the LP relaxation and adding linear constraints that drive the solution towards being integer without excluding any integer feasible points), (2) variants of the branch and bound method (the branch and cut method combines both branch and bound and cutting plane methods). The solutions of the LP relaxations give a worst-case estimate of how far from optimality the returned solution is.

The relaxation method to solve this SRM task consists of converting this discrete Linear Programming (LP) task to the LP task with continuous variables by substituting the binary constrains to the constraints where x_{ijt} are non-negative numbers limited by 1 (discrete constraint relaxation),

$$0 \leq x_{ijt} \leq 1, \qquad i = 1,2,\dots,N; j = 1,2,\dots,M$$

This classical LP task can be solved by the simplex method for large N and M. The next step is exploring the vicinities of vertices produced by the simplex method. This exploration includes finding feasible binary points in the vicinity, computing the value of the objective function on them and selecting the best ones. The size of the vicinities and the number of simplex vertices to be explored can be adjusted to minimize the computations.

A simple suboptimal version of this approach is to interpret non-integer component of the solution, x_{ijt} as a confidence measures that sensor sj should be assigned to the AOI a_i at time t. If such a confidence measure x_{ijt} is, say, above 0.8 then we can use rounding of x_{ijt} to 1 to get an integer solution. While such use of rounding is commonly criticized that it does not lead to the optimal solution, but its deviation from the optimal non-integer solution can be estimated and a suboptimal reasonable solution can be produced for large datasets using classical LP techniques.

3.2 Generalization of ILP SRM models to uncertain numbers

The natural generalization is coming from the fact that some coefficients and flags are uncertain in ILP SRM models. This uncertainty can be modeled by defining coefficients/flags as *uncertain numbers* given as: (1) intervals, (2) probability distributions, or (3) fuzzy sets. Respectively it leads to different classes of models and algorithms of optimization under uncertainty: *interval, stochastic,* or fuzzy optimization models and algorithms {Kovalerchuk, 1994].

In the optimization under uncertainty the key issue is defining and justifying a way to sum up uncertain numbers (summands in the ILP formulation). The definition of the sum depends on assumptions and goals. Below we consider three categories them.

Case 1: (classical interval math): All points in [a, b] interval are equally belong to this interval. We are interested only in the low and upper limits of the sum:

$$[a, b]+ [c, d] = [a+c, b+d].$$

Case 2: (discrete pdf math): All discrete points are distributed in [a, b] interval, that is for all x and y in [a,b] probabilities p(x) and p(y) are given and independent from the discrete pdf on [c, d]. We are interested in the distribution p(w) of sum points in the sum interval [a+c, b+d] including most likely sums:

$$p(w) = \sum_{x+y=w} p_{[a,b]}(x)p_{[c,d]}(y) \quad \text{for all } x+y=w$$

Case 3: (fuzzy math): All points in [a, b] are given with fuzzy logic membership function $m_{[a,b]}(x)$ in [a, b] interval and all points in [c, d] are given with fuzzy logic membership function $m_{[c,d]}(x)$. We are interested in getting a membership function of sum points in [a+c, b+d] interval including a most possible sum.

Zadeh [Berkeley Initiative on Soft Computing/BISC group, 04/07/2014] asserts that case 3 must be solved by applying his *extension principle*, described in his 1965 paper, because *fuzzy math* is based on this principle:

$$m_{[a,b]+[c,d]}(w) = \min \max (m_{[a,b]}(x), m_{[c,d]}(y))$$

for all x, y such that x+y=w, where x and y are from [a,b] and [c,d], respectively. The discussion at BISC [Zadeh, Kovalerchuk, Piegat, Tschantz and others, 2014] revealed disagreement on case 3 because the reference to the *extension principle* is not sufficient to justify the minmax formula above. This formula has a status of the hypothesis in general and in ILP SRM models with uncertain numbers given my membership functions in particular. Our approach for the case 3 is an adaptation of the case 2 [Kovalerchuk, BISC, April 2014].

3.3 Computation of flags - parameters of ILP Problem

Computing flags requires translating *requirements for tracking* in the form of track accuracy and uncertainty to the sensor capabilities in terms of flags. The similar translation is needed for the *discrimination requirements*. This *translation* allows a user flexibility to use both measurement information and tacit expert knowledge.

The binary flag $f(a_i,s_j,t,r)$ indicates whether sensor s_j is *capable* to *cover/observe* area a_i at the time interval t with required resolution/capability r. This flag is computed directly for each value of its variables: the identified areas a_i, sensor s_j, and required resolution/capability r at time t. LOS and FOV with their errors for sensor s_j at time t, location of the AOI a_i are used to check if a required resolution/capability can be reached. If the AOI a_i contains a detected moving object then the known dynamic properties of the trajectory of the object are used to identify the next location of the AIO a_i and to compute flag f for time t+1. These properties are derived from the external tracking, track correlation algorithms as outlined in Figure 1. Similarly each sensor resolution relative to objects is derived along other required for the model characteristics.

The flag $f^*(a_i,s_j,t,r,v)$ is directly computed from $f(a_i,s_j,t,r)$, when an area a is marked for a sensor s_j. The marking of the area is identified from tasking requirements and tasking doctrine. Flags $\{g_k(s_j)\}$ that indicate types of sensors are computed from the database of specifications of sensors. Relations flags are computed based on the definitions of the relations.

Flags can be assumed known or computed to achieve the required resolution of the sensor system. This includes Monte-Carlo Simulation and CRBs. The CRBs is computed for each object and sensor, as well as for each object and a pair of sensors as required. It can also be done by using tracking and fusion algorithms [Kovalerchuk, 2007] and algorithms outlined below.

3.4 Solutions with CRB and Dynamic Logic

Lambert and Sinno [2011] discuss in details that significant part of errors in tracking and fusion might originate from incorrect associations between sensors and objects; and therefore their results using CRBs that do not account for associations are only approximate. The proposed OPTIMA system uses CRBs with algorithms for tracking and fusion accounting for the association part of these problems. These types of the CRBs were obtained in [Perlovsky 1997]. Similarly algorithms optimally accounting for the association part of tracking and fusion problems [Perlovsky, Deming, & Ilin 2011; Perlovsky, Deming 2007, 2013] are used.

It is well known that the best algorithms currently used for tracking and fusion in difficult conditions cannot attain the best theoretically possible performance as specified by the Cramer-Rao Bounds for given difficult conditions [Perlovsky 1997]. This deficiency is due to high *computational complexity* of current tracking and fusion algorithms [Perlovsky 1997]. This limits sensor resource utilization. This fundamental difficulty of algorithms currently in use has been overcome by a computational intelligence *dynamic logic* approach [Perlovsky, Deming, & Ilin 2011; Perlovsky & Deming, 2013, Kovalerchuk, Perlovsky, 2013].

Dynamic logic starts not from the actual LP model M, but modifies both the objective function and constraints of M to produce a model M_1, and then solving M_1 as a solution for M. The full dynamic logic methodology process has multiple stages that *generate dynamically a sequence of models* M_1, M_2, …, M_n, where only model M_n is a solution of M. Models M_1,M_2,…,M_{n-1} provide only intermediate solutions.

Some algorithms for the ILP problem exploit the general idea, which is a core of dynamic logic approach described above. The LP relaxation algorithm is in this category, when some constraints are removed and the objective function is modified by adding a penalty summand. LP decomposition methods also modify LP models [Vanderbeck, Wolsey, 2010].

The cutting plane algorithm [Cornuejols, 2008] creates model M_2 by adding linear constraints (called cuts) to the relaxed model M_1 to drive the solution to be integer. The relaxed model M_1 removes constraints that variables are integers. The cut removes the current non-integer solution from a set of feasible solutions to the relaxation. This process of constructing new models M_i is repeated until an optimal integer solution is found in model M_n. However, these methods *do not change the dimensionality* of the search space and variables beyond converting discrete variables into continuous ones. The Dynamic Logic approach expands this by pointing out this underused opportunity. This is a fundamentally biologically inspired approach within the Computational Intelligence paradigm.

One of the drawbacks of the heuristic computational intelligence approaches is in the difficulties of estimating how far its solution is from the optimal one. Another drawback is in the uncertainty of the situation, when this algorithm does not find any solution. In this case we do not know whether the optimal solution does not exist or just was not found. This is a motivation for developing specialized computational intelligence algorithms for SRM integer linear programming models.

The dynamic logic idea for solving the SRM ILP model is finding some preliminary candidate solutions then adding more constrains and getting more accurate solutions. In this process we change an optimization criterion to find all feasible solutions that are under specified constraints and then solving the task again for the feasible solutions under new constraints.

Within the dynamic logic methodology the original objective function F is not used at the beginning of the process as an objective function but as a source to construct a new objective function F_1. Similarly, the original constraints C are used to build new constraints C_1. For instance, we can solve classical LP problem by removing a constraints that all variable are binary and then search for the binary solution using the classical solution as guidance.

Similarly the objective function F can be modeled much "rougher" by substituting coefficients by their interval estimates, e.g., a=5 is substituted by the interval [4,6] in function F_1. For instance, if F_1 is computed in way where positive coefficients are taken with their min value from their interval while negative coefficients are computed with their max value, we will get a lower estimate for both F and F_1. Similarly we can get upper estimates for these functions. In extreme cases if interval of the coefficient is [0,10] then low limits means removing this coefficient from the constraint, which simplifies computations.

4 Applications, Related and Future Work

An idea of distributed network coordination is discussed in [Smith Nguen, 2005, Tharmarasa at al., 2011] as a way to minimize communications and increase system robustness. The argument is that in the traditional approach sensors accept tasking orders from networked tracking elements that may have only *uncertain knowledge about sensor's capability*, limitations and other tasks such as self-defense. In particular, a real-time fuzzy control algorithm in [Smith, Nguyen, 2005, 2006] running on each UAV gives the UAV limited autonomy allowing it to change course immediately without consulting with any other entity. In a similar development a resource manager based on fuzzy logic is optimized by evolutionary algorithms.

In [Smith, Nguyen 2005-2006] a fuzzy logic resource allocation algorithm enables UAVs to automatically cooperate. The algorithm determines the trajectory and points each UAV for measurements. This fuzzy logic model takes into account the UAVs' risk, risk tolerance, reliability, mission priority for sampling, fuel limitations, mission cost, and other uncertainties. While the scope of this work differs from our task, expanding our optimization design to accommodate such factors as mission cost and related uncertainties is one of the topics of the future work. It is also an important direction in further development of fuzzy optimization [Kovalerchuk, 1994].

The discrete optimization formulation for large-scale sensor selection in decentralized networks is proposed in [Tharmarasa, 2011]. It considers a situation without central fusion center. Each Fusion Center (FC) communicates only with the neighboring FCs. Our model can be expanded for this situation too.

The methodology and OPTIMA class of SRM optimization models are quite universal. Therefore, the development and application of multi-sensor fusion systems based on these models opens significant opportunities for detection and classification in a wide range of areas from bio-surveillance, monitoring, fault diagnostics, medical diagnosis, to cargo inspection, inspection of infrastructure, and others.

Optimization of the wireless sensors and phone communications is one of them to increase the efficiency of communications. Others include environment monitoring, and management of any business resources: mapping of fires, detecting and mapping pollutions, air-quality, water-quality by a network of distributed sensors. Recent accidents with high speed trains motivate development SRM in this area [Junji et al, 2011]. Maturing and integrating SRM models and algorithms will make solution of the above outlined problems more efficient.

The resource saving is turned to more accurate monitoring to save electricity, water, paper, etc. Networked, temperature sensors can automatically map insulation leaks in buildings and reduce energy waste~~.~~ Temperature sensors, just like radars have coverage areas, sensitivity diagrams, etc. The ease of installation leads for the sensors to discover each other and communicate their measurements. This requires intelligent and adaptive algorithms such as OPTIMA. The same applies to irrigation, humidity, insect, soil chemical composition, etc. sensors in agriculture. Large number of low-cost, solar-powered, mesh-networked humidity and other sensors placed on a plantation can be optimized to help farmers save water, increase crop yields and lower cost.

5 Conclusion

Optimal SRM opens the opportunity: (1) to maximize the available sensor resources for search, (2) to optimize sensor resources for tracking, and, (3) to defend better the high priority assets. The models and algorithms proposed in this work allow the decreasing of the overall sensor resource usage, while increasing the probability that all threat objects in a raid are tracked, in addition, target characterization is optimized. Our unique approach is in multi-objective SRM optimization model and algorithms, as well as in the use of Cramer-Rao Bounds (CRBs), and the algorithms accounting for the association part of the tracking and fusion problem. These CRBs allow to evaluate target characterization (classification features), and therefore target values. Another uniqueness of our approach is in using flags within the SRM, which encompass all of the information external to the main goals of the program (such as information from tracking algorithms). These flags are readily computed from the available information or information adaptively estimated in real time. These benefits surpass existing state of the art and permit efficient sensor coordination.

6 References

1. Burgess, D., Levins, C.: Intelligent sensor resource management using evolutionary computing techniques. In: International Conference on Integration of Knowledge Intensive Multi-Agent Systems/KIMAS (2003) 325 – 329
2. Borndörer, R.; M.: Designing telecommunication networks by integer programming (2012)
3. Bar-Shalom Y., Li. X, Rong, Kirubarajan, T.: Estimation with applications to Tracking and Navigation. John Wiley & Sons (2001)
4. Castanon, D.A.: Approximate Dynamic Programming for Sensor Management. Proceedings of the 36th IEEE Conference on Decision and Control 1997
5. Cornuejols. G. Valid Inequalities for Mixed Integer Linear Programs, Mathematical Programming Ser. B 112 (2008) 3-44
6. Deming, R., Perlovsky, L.: Concurrent multi-object localization, data association, and navigation for a swarm of flying sensors, Information Fusion, 8 (2007) 316-330
7. Junger, M., et al.(Eds): 50 Years of Integer Programming 1958–2008, Springer (2010)
8. Junji, K. Makoto, S., Akihiko, S., Kentarou, H., Yutaka, S., Koichi, N., Takashi, M., Tetsuo, K.: Development of 100Mbps-Ethernet-based Train Communication Network. In: 9th World Congress on Railway Research, (5) (2011) 1-12
9. Hall, D, Llinas J.:An introduction to multisensor resource management, Proc.IEEE, 85(1) (1997) 6-23
10. Hanlon, N. Cohen, K, Kivelevitch, E.: Sensor Resource Management to Support UAS Integration into the National Airspace System, AIAA (2015) http://dx.doi.org/10.2514/6.2015-0360
11. Hero, A., Cochran, D.: Sensor Management: Past, Present and Future. IEEE Sensors Journal 11(12) (2011) 3064-3075
12. Hero. A., Keucher. C., Blatt, D.: Information theoretic approaches to sensor management. In: Foundations and Applications of Sensor Management, A. Hero. D. Castanon, D. Cochran, and K. Kastella. (Eds.) New York: Springer (2008) 33-57
13. Hero, A., Castanon, D., Cochran, D., and Kastella K., (Eds.) editors: Foundations and Applications of Sensor Management. Chapter 3. Springer (2007) 33-58
14. Kovalerchuk, B., Perlovsky, L., Wheeler, G.: Modeling of Phenomena and Dynamic Logic of Phenomena. Journal of Applied Non-classical Logics, 22(1) (2012) 51-82
15. Kovalerchuk B.: Interpretable Fuzzy Systems: Analysis of T-norm interpretability. 2010 IEEE World Congress on Computational Intelligence, Barcelona (2010) DOI: 10.1109/ FUZZY.2010.5584837
16. Kovalerchuk, B.: Modeling ATR processes prior to experimentation to predict ATR system performance. The 12th International Conference on Information Fusion, Seattle, USA (2009) 1-12.
17. Kovalerchuk. B., Perlovsky, L.: Fusion and Mining Spatial Data in Cyber-physical space with Phenomena Dynamic Logic. In: Proc. Intern. Joint Conf. on Neural Networks Atlanta, (2009) 440-2447
18. Kovalerchuk. B. Perlovsky, L., Dynamic Logic of Phenomena and Cognition, In: The 2008 IEEE World Congress on Computational Intelligence (WCCI 2008). Hong Kong (2008) 3529-3536
19. Kovalerchuk, B.: Structural and me tric correlation of electro-optical and radar generated tracks, Defense and Security Symposium. SPIE Proc. Vol. 6567-05 (2007)
20. Kovalerchuk, B.: Affine invariant and robust image registration/conflation algorithm, SPIE Defense and Security Symposium, Orlando, FL, SPIE Proc, Vol. 6568, 6568-08 (2007)
21. Kovalerchuk, B.: Current situation in foundations of fuzzy optimization. In: M. Delgado, J. Kacprzyk, J.-L. Verdegay and M.A. Vila (Eds.): Fuzzy Optimization: Recent Advances. Springe, (1994) 45-60
22. Kovalerchuk, B.: Optimal distribution of n jobs to m partially changeable processors. In: Application of optimization methods in planning and control. (Ed. L.V. Kantorovich), Moscow (1980) 245-253
23. Kreucher, C., Hero, A., Kastella K. and Morelande M., An Information-Based Approach to Sensor Management in Large Dynamic Networks. In: Proceedings of the IEEE 95(5) (2007) 978-999

24. Kreucher, C., Hero, A., and Kastella. K.: A Comparison of Task Driven and Information Driven Sensor Management for Target Tracking. 44th IEEE Conf. on Decision and Control (2005) 4004-4009
25. Kreucher C., Hero A., Kastella K., Chang D.: Efficient Method of non-Myopic Sensor Management for Multi-target tracking. 43rd IEEE Conference on Decision and Control (2004) 722-727
26. Lambert, H.C., Sinno, D.: Bioinspired Resource Management for Multi-Sensor object Tracking Systems, MIT Lincoln Laboratory Project Report MD-26 (2011)
27. Liggins, M. E., Hall, D. L., Llinas J. (eds): Handbook of Multisensor Resource Management. CRC (2008)
28. Manyika, L., Durrant-Whyte, H.: Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach. New York: Ellis Ilorwood (1994)
29. Pastor, J.: Mathematical Ecology of Populations and Ecosystems. Wiley-Blackwell (2008)
30. Patsikas, D.: Track Score Processing of Multiple Dissimilar Sensors, NPS, Monterey CA (2007)
31. Perillo,M., Heinzelman,W.:Sensor Management. Wireless Sensor Networks.Springer (2004) 351-372
32. Perlovsky, L.I., Deming, R. W.: Maximum likelihood joint tracking and association in a strong clutter without combinatorial complexity. International Journal of Advanced Robotic System, 9 (2013) 1-9
33. Perlovsky L., Deming, R., Ilin R.: Emotional Cognitive Neural Algorithms with Engineering Applications Dynamic Logic: From Vague to Crisp. Springer (2011)
34. Perlovsky, L.I. and Deming, R.W.: Neural Networks for Improved Tracking. IEEE Trans. Neural Networks, 18(6) 2007 1854-1857
35. Perlovsky, L.I.: Concurrent multi-target localization, data association, and navigation for a swarm of flying sensors, Information Fusion 8 (2007) 316-330
36. Perlovsky, L.: Cramer-Rao Bound for Tracking in Clutter and Tracking Multiple Objects". Pattern Recogn. Letters 18(3) (1997) 283-288
37. Perlovsky, L.I.: Conundrum of Combinatorial Complexity. IEEE Trans. PAMI, 20(6) (1998) 666-670
38. Perlovsky, L.I., Schoendorf, W.H., Tye, D.M., Chang, W.: Concurrent Classification and Tracking Using Maximum Likelihood Adaptive Neural System. J. Underwater Acoustics 45(2), (1995) 399-414
39. Severson, T., Paley, D.: Optimal sensor coordination for multiobject search and track assignment. IEEE Transactions on Aerospace and Electronic Systems 50(3) (2014) 2313-2320,
40. Shea, P., Kirk, J., Welchons, D.: Adaptive sensor management for multiple missions, Proc. SPIE 7330, Sensors and Systems for Space Applications III, 73300M, doi:10.1117/12.818892 (2009)
41. Smith III, J. F., Nguyen, T.Vu H.: Distributed autonomous systems: resource management, planning, and control algorithms. Proc. SPIE 5809, 65 (2005)
42. Smith, III, J.F., Nguyen, T.Vu H.: Fuzzy Logic Based UAV Allocation and Coordination. Naval Research Lab Washington DC, DTIC (2006) 11 p.
43. Tharmarasa, R., Kirubarajan, T., Sinha, A., and Lang, T.: Decentralized Sensor Selection for Large-Scale Multisensor-Multiobject Tracking. IEEE Transactions on Aerospace and Electronic Systems, 47(2) (2011) 1307-1324
44. Tian, X., Bar-Shalom, Y.: Exact Algorithms for Four Track-to-Track resource management Configurations: All you wanted to know but were afraid to ask. In: 12th International Conference on Information Fusion, Seattle, USA (2009) 537-554
45. Van Gils, J.: Digestive bottleneck affects foraging decisions in red knots Calidris canutus. I. Prey choice. Journal of Animal Ecology 74(1) (2005) 120-130
46. Vanderbeck, F., Wolsey,L.: Reformulation and Decomposition of Integer Programs, In: 50 Years of Integer Programming 1958–2008 M.Junger, T. Liebling, et al.(Eds), Springer (2010) 431-504
47. Verlinden, C. , Wiley, R.H.: The constraints of digestive rate: An alternative model of diet selection. Evolutionary Ecology 3(3) (1989) 264–273
48. Weir, B. S., Sokol, T. M.: Radar coordination and resource management in a distributed sensor network using emergent control. Proc. SPIE 7350, 73500I (2009)