MANAGING CYBER SECURITY RESOURCES VIA SIMULATION-BASED OPTIMIZATION

Pasquale Legato[a], Rina Mary Mazza[a]

[a]Department of Informatics, Modeling, Electronics and System Engineering
University of Calabria
Via P. Bucci 42C
87036 Rende (CS), ITALY

[a]legato@dimes.unical.it, [a]rmazza@dimes.unical.it

ABSTRACT

Simulation-based optimization (SO) has been applied in many different application areas with the objective of searching for the settings of controllable decision variables that yield the minimum (maximum) expected performance of a stochastic system. Here we propose an SO method to deal with computer/network security related to systems for conditional access. The basic idea consists in designing and developing a simulation-based optimization tool to evaluate cyber attack tolerance along with the related performance degradation. In particular, we optimize training-based recovery actions aimed at restoring the target quality of service level for the services under attack while enhancing the knowledge of the human resources (i.e. analysts) engaged in defending cyber security assets. An illustrative example is presented to show how system performance varies according to whether the analysts in a cyber defense team (i.e. the controllable decision variables) are called to work alone or in consultation with other analysts.

Keywords: simulation optimization, cyber security, team formation and collaboration

1. INTRODUCTION

Simulation-based optimization (SO) is the practice of searching for the settings of controllable decision variables that yield the minimum (maximum) expected performance of a stochastic system that is represented by a simulation model (Fu and Nelson 2003). In an SO procedure, a structured iterative approach calls an optimization algorithm to decide how to change the values for the set of input parameters (i.e. configuration) and then uses the responses generated by the simulation runs to guide the selection of the next set. The logic of this approach is shown in Figure 1.

SO methods have been applied to applications with a single objective, applications that require the optimization of multiple criteria, and applications with non-parametric objectives. (Carson and Maria 1997) review the area of simulation optimization by providing a critical review of the methods employed and presenting applications developed in the area. A similar work is somewhat proposed in (Wang and Shi 2013), but without major differences in content. Fields of SO application include, but are not limited to, energy, environment, economics, health, manufacturing, high tech, education, government, and defense.

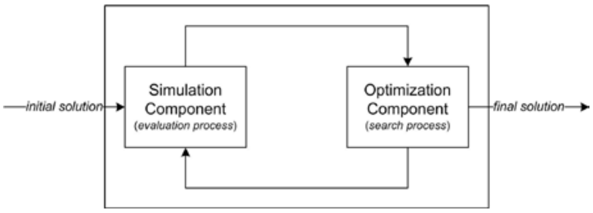

Figure 1: Logic of an SO Procedure

Whatever be the application field of interest, the appeal of SO is that it allows one to work with essentially arbitrarily complex simulation models, freeing the modeler from the tyranny of restricting model complexity to tractable forms (Pasupathy and Henderson 2011). In (cit. op.), the authors develop and promote a library (www.simopt.org) of over 50 simulation optimization problems intended to spur development and comparison of simulation-optimization methods and algorithms with respect to practical guarantees of performance.

To our knowledge, SO applied to cyber security has not received great attention in the past literature. (Fischer et al. 2010) present an effective simulation methodology called optimal splitting technique for rare events (OSTRE) with applications to cyber security. On one side, a splitting methodology is used to create separate copies of the simulation whenever it gets close to the rare event in order to multiply the promising runs that are "near" the rare event and, thus, improve the efficiency of the simulation. On the other, the notion of optimal computing budget allocation is applied to determine a good allocation of simulation runs at the intermediate levels (i.e. levels measure proximity to the rare event). The overall methodology is applied to simulate the link performance of an Internet Protocol (IP) network under a worm attack where the worm propagation creates a denial of service in many parts of the Internet and, thus, changes the traffic loading on the network. (Masi et al. 2011) extends (Fischer et al. 2010) by studying the sensitivity of the benefit of splitting to the number and location of the levels and also examining equal-allocation splitting.

(Zhang et al. 2012) present a learning environment to provide users with a unique way of improving their understanding of cyber intelligence with respect to the identification, tracking, analysis and countering of security threats in the cyberspace. In their system, a simulation engine drives the environment dynamics and changes; an optimization engine based on a multi-objective genetic algorithm implements decision making mechanisms; and data mining techniques provide for adaptation.

(Kiesling et al. 2013) introduce an approach that is based on an adversary-centric view and combines modeling and simulation-optimization techniques to detect ongoing attacks and prevent their successful execution. On the simulation side, human adversaries are represented as agents that make active decisions in attacking a system by means of attack patterns (e.g. brute force, SQL injection, social attack, spearfish attack, keylogger, and backdoor installation), deliberately exploiting dynamic interactions of vulnerabilities. On the optimization side, a Multi-objective genetic algorithm metaheuristic is introduced to optimize information systems and enable decision-makers to study how its security may be improved (e.g., by adding physical, technical, operational, and organizational security controls) while trading off multiple objectives (e.g., effectiveness against different types of adversaries, cost, risk, awareness of attacks). The overall model returns non-dominated efficient portfolios, i.e., there is no other portfolio with equal or better values for all objectives and a strictly better value in one of the objectives.

In our work we focus on computer/network security related to *systems for conditional access* by which we refer to digital systems that administer certain rights of their users pertaining to the access of documents, confidential data or, even more importantly, digital payment systems. Our ongoing research activity currently consists in designing and developing a qualitative and quantitative simulation-based optimization tool to evaluate attack tolerance, along with the related performance degradation. In particular, we optimize training-based recovery actions aimed at restoring the target quality of service level for the digital services under attack while enhancing the knowledge of the human resources (i.e. analysts) engaged in defending, alone or in cooperation with others, the cyber security assets to which they are assigned.

The rest of the paper is organized as follows: the statement of the problem is presented in section 2. In section 3, we present the meta-heuristic technique for the optimization of the controllable decision variables sets. Section 4 illustrates the practical usefulness of the tool by means of optimizations for a sample scenario. Conclusions and directions for further research investigations are presented in Section 5.

2. PROBLEM STATEMENT

Digital service systems are a fast-growing IT market area in which data exchange, transactions and payments are increasingly implemented by using advanced technologies, devices and network architectures (e.g. cloud computing, mobile devices, etc.). Within this context, cyber security has become an “enabling factor” for the use of digital systems, due to the fact that the comprehension and control of risk scenarios is assuming a particularly critical role, especially because of the latest emerging risk characteristics. These can be summarized as large-scale network attacks, associated with either fraudulent or denial-of-services activities, that exploit the increased vulnerability associated with new technologies (such as smartphones), growing availability of cloud services and infrastructure virtualization.

In the above market scenario, a major response can certainly come from deploying cyber defense security analysts. The main job of a cyber defense security analyst entails auditing computer networks and information systems for vulnerabilities, developing solutions for security issues and investigating security breaches. These analysts are also often responsible for training both (junior) co-workers and clients with respect to the best computer security practices.

In fulfilling the major of the above purposes, besides taking appropriate off-the-shelf preventive measures by installing firewalls and anti-virus software, a security analyst monitors server logs and network traffic for unusual or suspicious activity or data flow, often with the support of automation software and applications designed to detect and filter intrusion. Obviously, in a corporate-based perspective the contribution to providing an efficient protection of the cyber assets cannot be delivered regardless of the types of skills (e.g. the ability to carry out triage analysis, escalation analysis, correlations analysis, forensic analysis as described in (D’Amico and Whitley (2007)) and levels of skills (e.g. expert, average and novice) possessed by the analysts. As a result, both company activity scheduling and knowledge-sharing policies among analysts must undergo a systematic approach. To begin with, these activities require the formalization of skill types and levels. If different types of skills are defined with letters (e.g. 5 different types of skills are labeled from “A” to “E”) and different levels of skills are defined with numbers (e.g. expert level is 3, average level is 2 and novice level is 1), then the cyber defense skill map can be formalized by means of a 2-entry table such as the one depicted by Table 1 in which the cyber defense security staff is supposed to consists of 10 units.

Table 1: Example of Skill Types and Levels of the Cyber Defense Security Staff

Analyst/Skill	A	B	C	D	E
analyst 1		3		3	3
analyst 2	1	2	2		
analyst 3				1	1
analyst 4	3	2			
analyst 5		2	1		3
analyst 6		2	2	3	
analyst 7		1	3	2	
analyst 8			3		1
analyst 9	3		1	2	
analyst 10	3	2	1		

Besides "who knows how to do what", from the above table one may derive additional information. For instance, since analyst 1 is a senior worker in three skills (i.e. B, D and E), while analyst 3 is a junior worker bearing two of the same skills held by analyst 1 (i.e. D and E), if analyst 3 is meant to improve on skills D and E or start training on skill B, then he/she is likely to be engaged by senior management in teamwork with analyst 1. This and similar options lead to considering different knowledge-sharing policies and practices that may be addressed by the company.

In general, depending on their roles and/or tasks performed within a specific workflow, cyber defense analysts in a company may be called to *i*) work alone or *ii*) in consultation with other analysts who are committed to a common mission and are willing to share the knowledge that is necessary to fulfill that mission (Kvan and Candy 2000). The former case may apply in small companies that operate a non-collaborative policy because analyst staff is limited in number and each unit is dedicated to monitoring a specific cyber asset (see left-hand side of Figure 2). The latter case may apply in more complex organizations in which a team of analysts, each bearing specific knowledge and behavioral characteristics, exploit a set of rules to study macro-level patterns emerging from micro-level interactions among team members (see right-hand side of Figure 2).

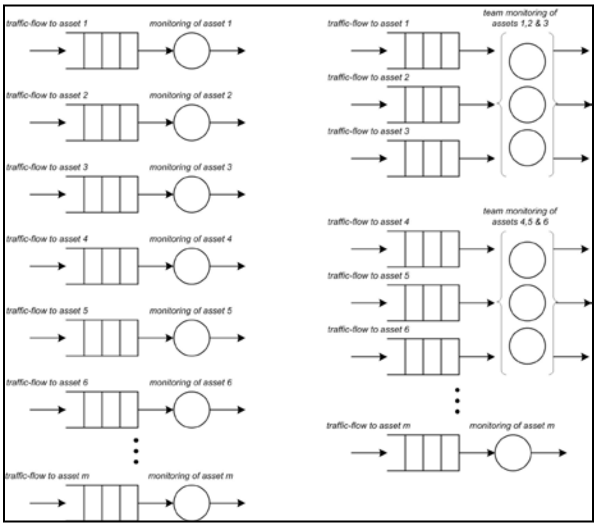

Figure 2: No Knowledge-sharing vs Knowledge-sharing Working Procedure

Whatever be the working procedure, let us consider the case in which a company operates an evaluation program according to which an analyst generates a certain number of credits for every attack mitigated. A credit is a measure of security performance ranking from 1 to 4 according to the type of attack. If the analyst works alone, every type of threat for which he/she is skilled will be detected and mitigated according to the proper service time - the service time depends on the type of attack and the analyst's (expert, average or novice) level of skill. As a result of attack mitigation, the analyst will be "rewarded" with the entire credit. If no such expertise is held by the analyst, the lack of ability to mitigate the malicious attack will have a negative impact on the entire system and likely cause a "loss" of overall performance. On the other hand, if the analyst works in consultation with other analysts, two situations may occur depending on whether or not the analyst holds the appropriate skill to detect and manage an attack. If he/she is properly skilled, then the attack will be dealt with according to the expertise of the analyst who, in turn, will be rewarded the entire credit at the end of the attack management process. Vice versa, if the analyst is obliged to consult with his/her team members in order to acquire the necessary know-how to manage the attack, then as a result of the ongoing interaction process: the service times of all the interacting team members will be inflated; the status of the skill level of the "enquiring" analyst will change thanks to the learning process he/she is undergoing; and the final credit will be shared among the team members that took part in the knowledge-sharing process. Of course, if none of the team members hold the appropriate skill to manage the attack, in the same way as the work alone *modus operandi*, this lack will have a negative impact on the entire system and cause a loss of overall performance.

In the present study we consider both of the above policies, but individually and propose for each configuration a qualitative/quantitative simulation-based optimization methodology to evaluate attack tolerance, along with the related performance degradation. Specifically, under a given attack scenario by which we mean different types and rates of attacks targeting a predefined set of cyber assets, the resulting model is aimed to estimate the following (average) performance metrics:

- percentage of attacks mitigated;
- resource (analyst) utilization;
- number of credits gained

and in addition for the knowledge-sharing policy among team members:

- number of cyber defense security analysts per team;

- cyber defense security team composition in terms of skill types and levels held by every single analyst assigned to every single team;
- knowledge gain.

3. SO METHODOLOGY

It should be clear at this point that the problem at hand is a problem of cost-performance-security evaluation of security services, where both (scarce and costly) human resources have to be allocated to important assets in a rational way in order to face the execution of activities and transactions by several actors under different policies. In this section we show how simulation-based optimization takes over whenever optimal resource allocation is more properly modeled in stochastic environment, due to the exclusive capability of discrete-event simulation to reproduce attacker activities against vulnerable assets and defender responses under security controls and policies. We expect to achieve a rational cost-effective organization of security analysts devoted to activity and resource monitoring, along with a rational cooperation and training of skilled personnel.

To show that this is the case here, we present an integer programming based mathematical formulation of the simplified decision problem of allocating analysts from several sources to a given asset to face multiple types of possible threats. It is inspired from the classical multi-choice multi-knapsack problem (Hifi, Michrafy and Sbihi 2004). Let's assume that:

- $i = 1,...,I$ analysts are available, each with a certain number of skills measured in terms of capability of covering to some extent a fixed set of threats;
- $j = 1,...,J$ assets are given as subjected to $k = 1,...,K$ threats;
- r_{ij}^{k} is the expected reward achieved (amount of risk covered) with respect to threat k by allocating analyst i to asset j;
- analyst i could be allocated to asset j (through the binary x_{ij} variable) at an expected cost c_{ij}.

We want to minimize the total allocation cost under the constraint of guaranteeing a fixed level of coverage (R_k) against each threat.

The formulation follows.

$$\min \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} x_{ij} \qquad (1)$$

$$\sum_{i=1}^{I} \sum_{j=1}^{J} r_{ij}^{k} x_{ij} \geq R_k \qquad k = 1,...,K \qquad (2)$$

$$\sum_{j=1}^{J} x_{ij} = 1 \qquad i = 1,...,I \qquad (3)$$

$$x_{ij} \in \{0,1\} \qquad i = 1,...,I; \quad j = 1,...,J \qquad (4)$$

In principle, a mathematical programming formulation as the one just presented can be embodied in a simulation-based optimization algorithm (see, for example, the context-specific chart illustrated in Figure 3). Once both the objective and the set of constraints have been formulated, i.e. formalized under some linear or non linear functions and inequalities, and an initial feasible solution is available, then statistical analysis of the simulation output data gathered from one sufficiently-long run or multiple simulation runs allows to estimate the expected values of the predefined security rewards and costs corresponding to the current feasible solution. An iterative search process is then applied with the aim of exploiting the neighborhood of the current solution and sometimes suitably escaping from local minima to explore the whole feasible set of solutions.

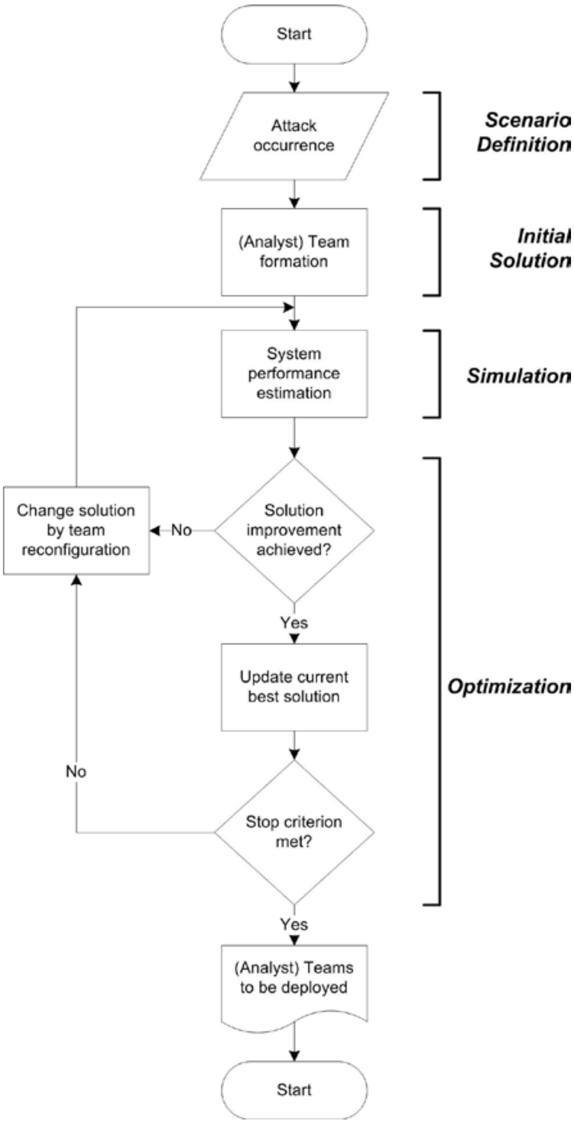
Start
Attack occurrence
Scenario Definition
(Analyst) Team formation
Initial Solution
System performance estimation
Simulation
Change solution by team reconfiguration
No
Solution improvement achieved?
Yes
Update current best solution
Optimization
No
Stop criterion met?
Yes
(Analyst) Teams to be deployed
Start

Figure 3: The Context-Specific SO Procedure

In a "complete" cyber environment, one could accept the deterministic measure of the c_{ij} parameter as a pure (monetary) cost. However, the inadequacy of such a formulation becomes evident by recalling the more complex policies and objectives underlying the complex cooperation and training settings under the randomly occurring events described in the previous section. So, we focus on a simulation-based optimization approach and simply put aside the IP formulation. As for the search process, here two different simulation optimization algorithms are tailored to the problem: *simulated annealing* and *ant colony optimization*.

3.1. Heuristic Methods

Heuristic methods represent the latest developments in the field of direct search methods that are frequently used for simulation optimization. Many of these techniques balance the global search for promising solutions within the entire feasible region (*exploration*) with the local search of promising sub-regions (*exploitation*), thereby resulting in efficient global search strategies. In the following we tailor two different types of meta-heuristics to our team formation problem: *simulated annealing* and *ant colony optimization*. The reason for choosing to illustrate these two meta-heuristics is due to the fact that, given a fixed time budget, they represent two different philosophies in the implementation of a computationally expensive activity such as simulation optimization. In the former, more computational time is devoted to estimate via simulation (*evaluation process*) the solutions found because only one neighbor solution is sampled and more time is left for the simulation process. In the latter, more time is devoted to find improved solutions on the optimization side (*search process*) because of the wider sampling of the neighborhood. In general, for any problem it is impossible to determine *a priori* which option carries a better pay-off.

3.1.1. Simulated Annealing

Originally introduced by (Kirkpatrick, Gelatt and Vecchi 1983), simulated annealing (SA) was developed on the similarities between combinatorial optimization problems and statistical mechanics. In the field of metal sciences, the annealing process is used to eliminate the reticular defects from crystals by heating and then gradually cooling the metal. In our case, a reticular defect could be seen as grouping analysts in teams that are not able to "properly" cover cyber assets and, thus, guarantee a given quality of service level when the above assets undergo an attack.

Technically speaking, the annealing process is aimed to generate feasible teams of analysts, explore them in a more or less restricted amount and, finally, stop at a satisfactory solution. To avoid getting caught in local minima, during the exploration process a transition to a worse feasible solution (higher-energy state) can occur with probability

$$p = e^{\Delta / T} \tag{5}$$

where Δ is the difference between the values of the objective function (energy) of the current solution (state) θ and the candidate solution θ_t and T is the process temperature. A prefixed value of T determines the stop of the entire process and it usually decreases according to a so-called *cooling schema*. Unfortunately, in the literature there is no algorithm that can determine "correct" values for the initial temperature and cooling schema, but, as suggested by empirical knowledge simple cooling schemas seem to work well (Ingber 1993).

In the following, some pseudo-code is given for the original SA algorithm for a minimization problem.

```
Algorithm 1: Simulated Annealing
$\theta \leftarrow$ initial solution
for time = 1 to time-budget do
    $T \leftarrow$ cooling-schema[time]
    if $T = 0$ then
        Present current solution as the estimate of the optimal solution and stop
    Generate a random neighbor $\theta_t$ of the current solution $\theta$ by performing a move.
    $\Delta = f(\theta) - f(\theta_t)$
    if $\Delta > 0$ then
        $\theta \leftarrow \theta_t$
    else
        $\theta \leftarrow \theta_t$ (with probability $p = e^{\Delta/T}$)
end for
```

When customizing the SA algorithm to our problem, some choices need to be made.

To begin with, choosing the proper cooling schema has great impact on reaching a global minimum. In particular, it affects the number and which analysts are assigned to a team (solutions) that will be evaluated by running the SA algorithm. To this end, the so-called simple mathematical cooling schema $T_{i+1} = \alpha \cdot T_i$ has been tested, and the best results are returned for an initial temperature $T_0 = 100$ and a decreasing rate $\alpha \approx 0.9$.

The "move" definition for neighborhood generation is very context-sensitive. For our problem, a move must be defined with respect to the feasibility (or lack thereof) of a team by taking into account the analysts' skills, as well as the constraint that limits the number of analysts that can communicate and, thus, be assigned to the same team. Some examples of moves are:

- move analyst l from team i to team j ($i \neq j$);
- swap analyst l and analyst k ($l \neq k$), originally assigned to team i and team j ($i \neq j$), respectively.

As far as the stopping criteria are concerned, designers can choose among the following possibilities:

- stop when the algorithm has reached a fixed number of iterations n or an upper bound on the available time-budget;
- stop when the current solution has not been updated in the last m iterations;
- stop when the cooling schema has reached a *lower bound* on the temperature.

3.1.2. Ant Colony Optimization

Ant colony optimization (ACO) is a population-based metaheuristic for combinatorial optimization problems which was inspired by the capability of real ants to find the shortest path between their nest and a food source. Dorigo (1992) developed the first ACO algorithm called ant system and applied it to solve the traveling salesman problem (TSP). In this problem, an ant builds a tour by moving from one city to another until all cities have been visited and the objective is to find the tour that minimizes the distance traveled in visiting all cities.

The probability that ant *k* in city *i* chooses to go to city *j* is given by the following rule

$$p_k(i,j)=\begin{cases}\dfrac{[\tau(i,j)]\cdot[\eta(i,j)]^{\beta}}{\sum_{g\in J_k(i)}[\tau(i,g)]\cdot[\eta(i,g)]^{\beta}} & \text{if } j\in J_k(i)\\ 0 & \text{otherwise}\end{cases}\tag{6}$$

where $\tau(i,j)$ is the pheromone associated to the connection from city *i* to city *j*, $\eta(i,j)$ is a simple heuristic guiding the ant, for example $\eta(i,j)=1/d(i,j)$ where $d(i,j)$ is the distance between the two cities, and β is used to define the importance of the heuristic information as opposed to the pheromone information.

Once ant *k* has built a tour, the pheromone trail is updated according to

$$\tau(i,j)=\rho\cdot\tau(i,j)+\sum_{k=1}^{m}\Delta\tau_k(i,j)\tag{7}$$

where ρ is the evaporation rate and

$$\Delta\tau_k(i,j)=\begin{cases}\dfrac{1}{L_k} & \text{if } (\text{i, j})\in \text{tour of ant } k\\ 0 & \text{otherwise}\end{cases}\tag{8}$$

is the pheromone increase on all the edges visited by the all the ants (the more the ants visit an edge, the greater the pheromone they leave).

In the following, some pseudo-code is given for a basic ACO algorithm for a minimization problem.

Algorithm 2: Ant Colony Optimization

```
P ← pheromone initialization
$\theta_{gb}$ ← global best solution is null
**for** *time = 1* **to** *time-budget* **do**
    $\Theta_{iteration}$ ← {}
    **for** *j=1* **to** *n° of ants*
        $\theta$ ← build a solution based on P
        **if** $\theta$ is feasible **then**
            **if** $(f(\theta)< f(\theta_{gb}))$ or $\theta_{gb}$ is null **then**
                $\theta_{gb} \leftarrow \theta$
            $\Theta_{iteration}$ ← $\Theta_{iteration} \cup \{\theta\}$
    **end for**
    P ← pheromone update
**end for**
**return** $\theta_{gb}$
```

Some variants to the original algorithm have been proposed such as:

- update the pheromone trail by allowing only the best ant to place pheromone after an iteration of the algorithm $\tau(i,j)=\rho\cdot\tau(i,j)+\Delta\tau_{ij}(\text{best iteration ant})$;
- update the pheromone trail every γ iterations by allowing only the best global ant to place pheromone $\tau(i,j)=\rho\cdot\tau(i,j)+\Delta\tau_{ij}(\text{best global ant})$.

Once again we must think of customization: the TSP is an ordering problem, while in our case we face a grouping problem. AS has been applied to solve other grouping problems such as bin packing, cutting stock (see, for example, Levine and Lucatelle 2004) and graph coloring (see, for example, Costa and Hertz 1997).

Rather than visiting cities, in our problem an ant moves to connect analysts with different skills and, thus, form teams to defend a given set of cyber assets. In doing so, the ant leaves a pheromone trail between analysts *i* and *j* which may be seen as the global goodness of teaming *i* and *j*. So the probability of an ant *k* connecting analyst *i* with *j* is still given by (6), where $\eta(i,j)$ is the number of different skills obtained when teaming the analysts. Once ant *k* has teamed all the analysts the pheromone trail is updated according to (7) where ρ is the evaporation rate and

$$\Delta\tau_k(i,j)=\begin{cases}\dfrac{\sum_{j\in K_k(i))}\tau(i,j)}{\eta(i,j)} & \text{if } I\neq\{\,\}\\ 1 & \text{otherwise}\end{cases}\tag{9}$$

is the pheromone value given by the sum of all the pheromone values between analyst *i* and all the analysts connected to *i* (including *j* obviously)

As for the stopping criteria, designers can chose among the same options given for the customization of the SA algorithm.

4. ILLUSTRATIVE EXAMPLE

4.1. Preliminary Verification

The SO model has been implemented in Microsoft Visual Basic 6.0 Professional and experiments have all been run on a personal computer equipped with a 2.26Hz Inter Core™2 duo processor and 3 GB of RAM. Input data and SO parameters are specified in the proper sections of a simple GUI panel, as the one illustrated in Figure 4. In particular, one must first define the attack scenario by specifying attack interarrivals (in time units) and composition (in percentage) with respect to different types of attacks. Skills are then defined by specifying for every analyst which skills he/she features and the level of competence for each skill (0=no skill, 1=novice, 2=average, 3=expert). After inserting the number of teammates (here ranging between 1 and 10), the input stage is then completed by providing the settings for the simulated annealing-based SO scheme and the simulation settings which are, respectively, the initial temperature along with the cooling rate of the SA procedure, and the time horizon, the number of simulation runs to be performed and the simulation seed.

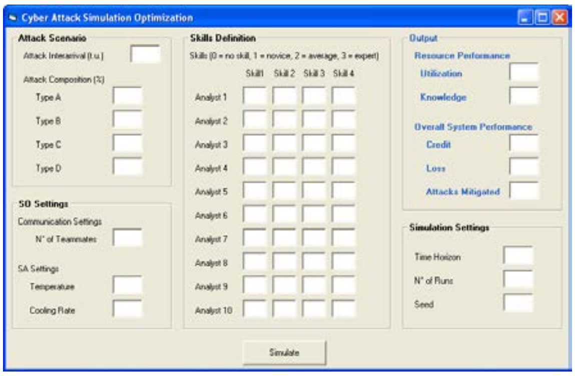

Figure 4: Snapshot of the SO Tool

The design and implementation of the SO tool depicted in Figure 4 has been carried-out in compliance with all the conventional steps used to guide a thorough and sound simulation study (Banks et al. 2001). For the time being, due to the unavailability of real-world input data as driving force for the SO model, here we will focus on illustrating the "predictive" capability of the model by explicitly reporting on the verification, rather than validation step of the study.

Verification has been performed with respect to the input parameters and logical structure of the SO model by combining three classes of techniques: common-sense techniques, thorough documentation and traces. A set of *ad hoc* instances have been used for verification by running the model in boundary cognitive conditions which allow expecting a predetermined system behavior. For instance, lets us consider the case in which every analyst *i* has an expert competence in every skill *j* (i.e. *SkillMap*(*i*,*j*)=3 for every (*i*,*j*)). Now, since every single analyst is skilled at the highest level for any type of attack, whatever be the number of teammates in the given scenario, the SO model is likely to return a fairly stable *utilization* of analysts (U) along with high accomplishments in both *credit* (C) and *number of attacks mitigated* (AM), while no *loss* (L) or whatsoever gain in *knowledge* (K) should be recorded. As a result, as one may see from Table 1, in which all of the above indices are averaged over (suitable) multiple runs and plotted as the number of teammates grows from 1 to 10, there are no significant changes in system performance whatever be the level of collaboration between analysts inserted by the user.

Table 2: Model Verification in a "Boundary" Scenario

Teammates	U	K	C	L	AM
1	0.66	0	14349	0	5736
2	0.66	0	14066	0	5708
3	0.67	0	14389	0	5751
4	0.66	0	14254	0	5705
5	0.66	0	14358	0	5736
6	0.67	0	14397	0	5756
7	0.67	0	14388	0	5758
8	0.66	0	14294	0	5720
9	0.66	0	14314	0	5702
10	0.67	0	14443	0	5768

The results of this and other similar experiments allow us to be confident in the correctness of the SO model.

4.2. Problem Set-up

In this subsection we present the problem set-up involved in the second set of experiments designed to estimate performance under a given attack scenario.

As far as system features are concerned, we consider 10 cyber assets which are attacked, according to an exponential renewal process, on average every 200 time units (λ , the average interarrival rate, is thus equal to 1/200) from a combination of 4 different types of attacks (i.e. 60% type A, 25% type B, 10% type C and 5% type D). Defense is provided by a team of 10 cyber analysts, where each analyst is skilled according to the data reported in Table 3. In the given scenario, analysts respond to attacks by working alone (n° of teammates=1) or in cooperation with other analysts (n° of teammates>1). The rate (μ) of the attack mitigation activity depends on the type of attack, the skill level held by the analyst and if mitigation occurs alone or in cooperation with other analysts. In the later case, mitigation times are inflated by 25%.

Table 3: Skill Types and Levels of the Cyber Defense Security Staff

Analyst/Skill	A	B	C	D
analyst 1	1	0	1	0
analyst 2	3	0	0	0
analyst 3	0	2	0	0
analyst 4	1	0	3	0
analyst 5	2	0	0	0
analyst 6	1	3	0	0
analyst 7	0	0	1	3

analyst 8	2	0	0	0
analyst 9	0	3	0	0
analyst 10	3	0	1	0

The initial temperature and the cooling rate of the SA scheme are set equal to 100 and at least 0.948, respectively, so that at least 100 different team-formation and assignment configurations are considered for the given scenario. The time horizon is fixed at 14400 time units (i.e. one four-hour labor shift) and, based on system variance, from 10 to 30 runs are performed for each experiment in order to obtain point estimates and/or construct 95% confidence intervals for estimating resource (i.e. analyst) *utilization* and *knowledge* gain and system *credit*, *loss* and *number of attacks mitigated*. Here, for clarity of illustration, we prefer using (stable) point estimates to show the numerical results in the next subsection.

4.3. Numerical Experiments

In this subsection, the illustration and related discussion of the numerical results returned by the previously defined scenario of the SO model is quite clear. In summary, except for the case in which an analyst is called to work alone, cooperation in small teams seems to return better performances.

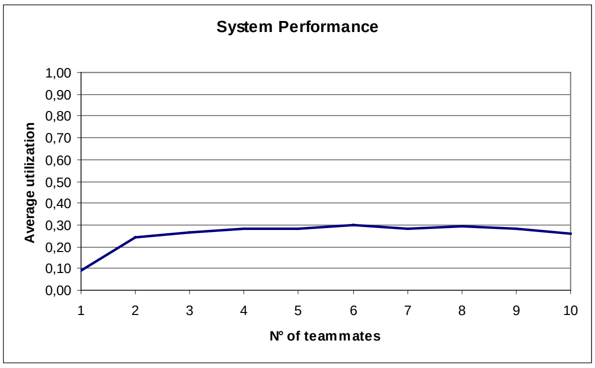

Figure 5: Trend of Average Analyst Utilization

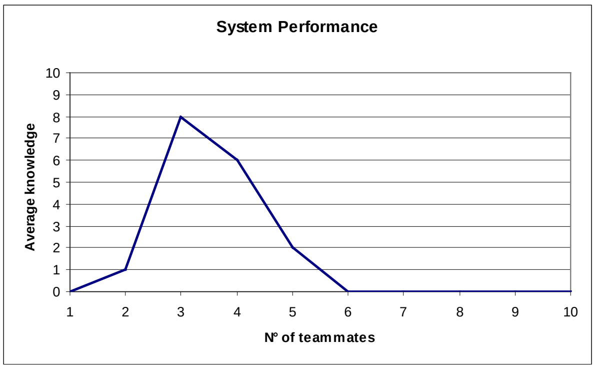

Figure 6: Trend of Average Knowledge Gain

To begin with, let us consider the (average) values of what we believe to be two measures of resource performance: analyst utilization and knowledge gain. Figures 5 and 6, respectively, show how utilization reaches its peak when the number of *teammates* is equal to 3, while utilization stays rather stable for $teammates > 3$. As one may see from Table 4, this happens in conjunction with a specific asset-analyst assignment and subsequent team formation in which analysts with complementary skills have been teamed together.

Table 4: Details of Best Asset-Analyst Assignment and Team Formation when Number of Teammates=3

Analyst	Asset	Teammates	Skills Covered
1	10	6 & 9	1, 2 & 3
2	5	4 & 10	1 & 3
3	8	5 & 7	1, 2, 3 & 4
4	3	2 & 8	1 & 3
5	6	3 & 6	1 & 2
6	7	1 & 5	1, 2 & 3
7	9	3 & 10	1, 3 & 4
8	1	4 & 9	1, 2 & 3
9	2	1 & 8	1, 2 & 3
10	4	2 & 7	1, 3 & 4

As for the so-called system performance measures, system loss is totally overcome when $teammates \geq 4$, while the credit acquired and the number of attacks mitigated do not seem to experience significant changes beyond the above value of *teammates* (see Figures 7, 8 and 9, respectively). This is even more evident if one plots the point-by-point difference between system credit and loss over the number of teammates: for $teammates \geq 4$ this trend is stable on a value approximately equal to 1200 units of credit.

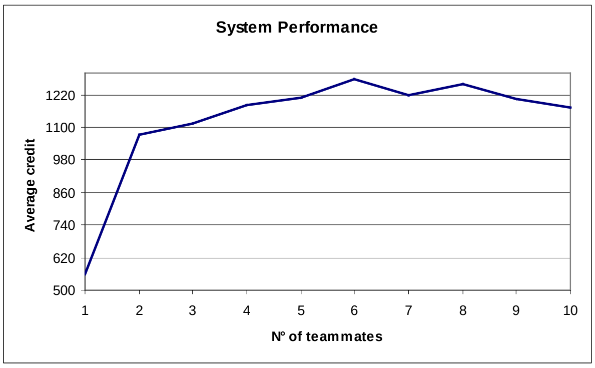

Figure 7: Trend of Average System Credit

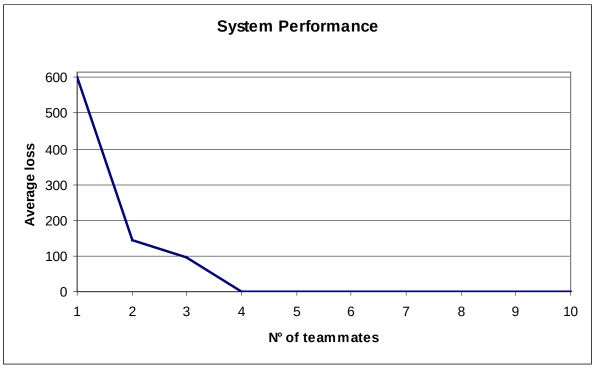

Figure 8: Trend of Average System Loss

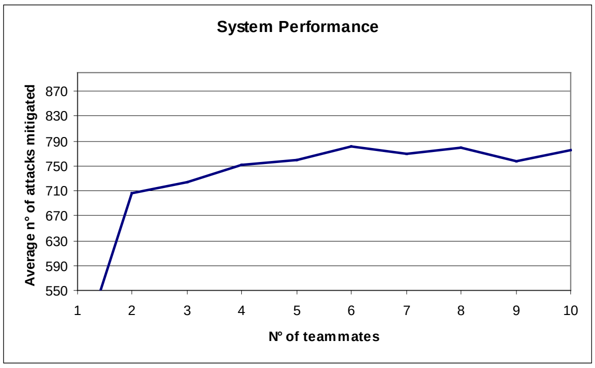

Figure 9: Trend of Average Number of Attacks Mitigated

So now the final question should be whether or not there are changes in system behavior when the average interarrival rate (λ) of the attacks grows significantly, for example by a factor of four (e.g. average interarrival times go from 200 to 50 time units), while clearly keeping satisfied the classical stability condition $\lambda/\mu < 1$ where we recall μ being the service rate (i.e. attack mitigation rate). Regardless of the growth of the interarrival rate, numerical experiments not reported here for lack of space state that only minor changes occur in system behavior such as system loss that goes to zero at a lower speed and specifically when $teammates = 10$, rather than $teammates \geq 4$.

5. CONCLUSIONS

In this paper we presented a simulation-based optimization model to assess attack tolerance in cyber systems when man-monitored assets are targeted by different types of attacks and different attack rates. The opportunity of teaming cyber defense analysts to work together in attack mitigation, rather than implementing a non-cooperative working policy seems to suitably fit the twofold purpose of cyber security defense unit: protection and learning. Numerical evidence shows that resource (analyst) utilization and knowledge, as well as system loss, credit and number of attacks mitigated benefit from the formation of small, yet well-assorted teams as far as the skills held by analysts belonging to the same team are concerned. In the proposed set of sample experiments, the best overall performance is recorded for a number of teammates ranging between 3 and 4.

Future research effort will focus on input modeling pertaining to both data and organizational matters. In the former case, a finer representation of the stochastic cyber attacks process will be provided via phase-type distribution-based models to match, for instance, the dynamics of attack graphs. In the latter, new policies will be considered. In particular, the current version of the SO model may be seen as the in vitro lab to assess the proficiency of proactive policies within the wider underlying attacker-defender game logic in cyber security.

ACKNOWLEDGMENTS

This work was partly supported by the Cyber Security Technological District funded within the Italian National Operational Programme for Research and Competitiveness 2007-2013 under grant number PON03PE_00032_2.

REFERENCES

Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2000. *Discrete-Event System Simulation*. 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Carson, Y. and Maria, A., 1997. Simulation optimization: methods and applications. In *Proceedings of the 1997 Winter Simulation Conference*, Andradóttir, S., Healy, K.J., Withers, D.H., and Nelson, B.L. (eds.), 118-126. December 7-10, 1997, Atlanta (Georgia, USA).

Costa, D. and Hertz, A., 1997. Ants can colour graphs. *Journal of the Operational Research Society*, 48:295–305, 1997.

D'Amico A. and Whitley, K., 2007. The Real Work of Computer Network Defense Analysts. In *VizSEC 2007. Proceedings of the Workshop on Visualization for Computer Security*. J.R. Goodall, G. Conti, K.-L. Ma (eds.), Springer-Verlag Berlin Heidelberg, 19-37.

Dorigo, M., 1992. Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale. Thesis (PhD), Politecnico di Milano, Italy.

Fischer, M.J., Masi, D.M.B., Chen, C.-H., and Shortle, J.F., 2010. Simulating non-stationary congestion systems using splitting with applications to cyber security. In In *Proceedings of the 2010 Winter Simulation Conference*, Johansson, B., Jain, S., Montoya-Torres, J., Hugan, J., and Yücesan, E. (eds.), 2865-2875, December 5-8, 2010, Baltimore (Maryland, USA).

Fu, M. and Nelson, B., 2003. Guest Editorial. *ACM Transactions on Modeling and Computer Simulation* 13(2), 105–107.

Hifi, M., Michrafy, M. and Sbihi, A., 2004. Heuristic algorithms for the multiple-choice multidimensional knapsack problem. *The Journal of the Operational Research Society*, 55(12):1323-1332.

Ingber, L., 1993. Simulated annealing: Practice versus theory. *Mathematical Computer Modelling*, 18(11):29-57.

Kiesling, E., Ekelhart, A., Grill, B., Strauss, C. and Stummer, C., 2013. Simulation-based optimization of information security controls: an adversary-centric approach. In *Proceedings of the 2013 Winter Simulation Conference*, Pasupathy, R., Kim, S.-H., Tolk, A., Hill, R., and Kuhl, M.E. (eds.), 2054-2065, December 8-11, 2013, Washington DC (USA).

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M. P., 1983. Optimization by Simulated Annealing. *Science*, New Series, 220(4598):671-680.

Kvan, T. and Candy, L., 2000. Designing Collaborative Environments for Strategic Knowledge in Design. *Knowledge-Based Systems* 13(6), 429-438.

Levine, J. and Lucatelle, D., 2004. Ant colony optimisation and local search for bin packing and cutting stock problems *Journal of the Operational Research Society*, 55(7):705–716.

Masi, D.M.B., Fischer, M.J., Shortle, J.F., and Chen, C.-H., 2010. Simulating network cyber attacks using splitting techniques. In In *Proceedings of the 2011 Winter Simulation Conference*, Jain, S., Creasey, R.R., Himmelspach, J., White, K.P., and Fu, M. (eds.), 3217-3228, December 5-8, 2010, Baltimore (Maryland, USA).

Pasupathy, R. and Henderson, S.G., 2011. SimOpt: a Library of Simulation Optimization Problems. In *Proceedings of the 2011 Winter Simulation Conference*. Jain, S., Creasy, R.R., Himmelspach, J., Whote, K.P. and Fu, M. (eds.), 4080-4090, December 11-14, 2011, Phoenix (Arizona, USA).

Wang, L.-F. and Shi, L.-Y., 2013. Simulation optimization: a review on theory and applications. *Acta Automatica Sinica*, **39**(11): 1957−1968.

Zhang, B., Shafi, K., and Abbass, H.A., 2012. Robo-teacher: a computational simulation based educational system to improve cyber security. *Robot Intelligence Technology and Applications*, Kim, J.-H., Matson, E.T., Myung, H., and Xu, P. (eds.) AISC 208 179-186.

AUTHORS BIOGRAPHY

Pasquale LEGATO is an Associate Professor of Operations Research in the Department of Informatics, Modeling, Electronics and System Engineering at the University of Calabria, Rende (CS, Italy), where he teaches courses on simulation for system performance evaluation. He has published on queuing network models for job shop and logistic systems, as well as on integer programming models. He has been involved in several national and international applied research projects and is serving as a reviewer for many international journals. His current research activities focus on the development and analysis of queuing network models for logistic systems, discrete-event simulation and the integration of simulation output analysis techniques with combinatorial optimization algorithms for real-life applications. His email is legato@deis.unical.it and his web page is www.deis.unical.it/legato.

Rina Mary MAZZA is the Research Manager in the Department of Informatics, Modeling, Electronics and System Engineering at the University of Calabria, Rende (CS, Italy). She received a Ph.D. degree in Operations Research from the above university. She has a seven-year working experience on knowledge management and quality assurance in research centers. She is also a consultant for operations modeling and simulation in terminal containers. Her current research interests include discrete-event simulation and optimum-seeking by simulation in complex systems. Her e-mail address is rmazza@deis.unical.it.