Department of Mathematics

Effect Oriented Planning

Nils-Hassan Quttineh, Kristian Lundberg
Kaj Holmberg, Torbjörn Larsson

LiTH-MAT-R–2012/06–SE

Linköpings universitet

Linköpings universitet
581 83 Linköping

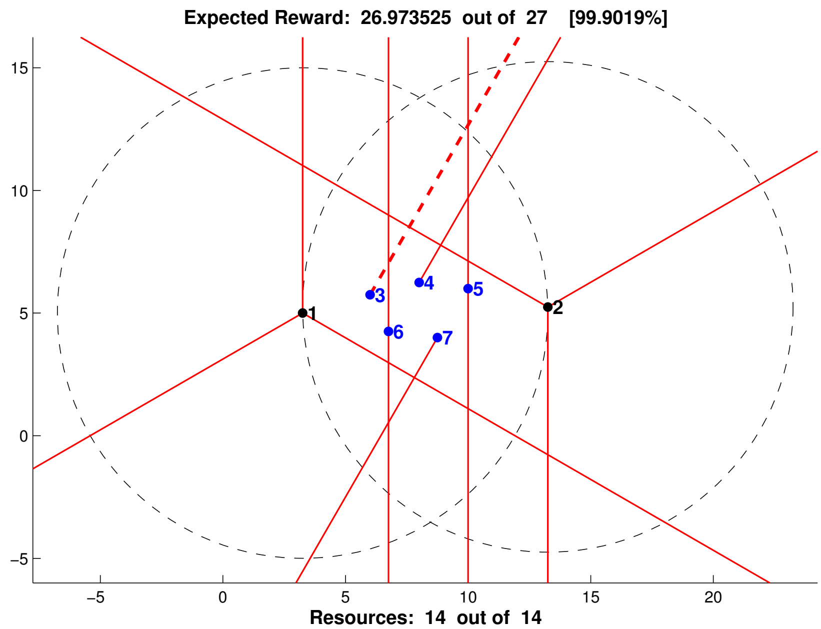
Linköping University
SE-581 83 Linköping, Sweden

Effect Oriented Planning

Nils-Hassan Quttineh[a,1], **Kristian Lundberg**[a]
Kaj Holmberg[a], **Torbjörn Larsson**[a]

Linköping University

January 2012



[a] Department of Mathematics, Linköping University, SE-581 83 Linköping, Sweden
[1] Corresponding author. Tel: +46 13 282185. E-mail address: nils-hassan.quttineh@liu.se

This work is funded by the Swedish funding agency Vinnova.

Effect Oriented Planning

Typeset by the authors in LaTeX2e documentation system.

Contents

Effect Oriented Planning

Nils-Hassan Quttineh, Kristian Lundberg
Kaj Holmberg, Torbjörn Larsson

January 24, 2012

Abstract The problem setting concerns the tactical planning of a military operation. Imagine a big wide open area where a number of interesting targets are positioned. It could be radar stations or other surveillance equipment, with or without defensive capabilities, which the attacker wishes to destroy. Moreover, the targets are possibly guarded by defending units, like Surface-to-Air Missile (SAM) units. The positions of all units, targets and defenders, are known. We consider the problem of the attacker, where the objective is to maximize the expected outcome of a joint attack against the enemy, subject to a limited amount of resources (i.e. aircraft, tanks). We present a mathematical model for this problem, together with alternative model versions which provide optimistic and a pessimistic approximations. The model is not efficient for large problem instances, hence we also provide heuristic solution approaches and successfully provide solutions to a number of scenarios.

Keywords: Targeting Problem, Weapon-Target Assignment, Military Operations Research, Decision Support.

1 Introduction

Effect Based Operations (EBO) is a military concept which emerged during the 1991 Gulf war for the planning and conduct of operations combining military and non-military methods to achieve a particular effect. The doctrine was developed to take advantage of advancements in weaponry and tactics, from an emerging understanding that attacking a second-order target may have first order consequences for a variety of objectives. The Commander's intent can be satisfied with a minimum of collateral damage or risk to his own forces, it follows that EBO embrace political factors as well as economic which makes the whole problem complex and hard to solve.

Despite its complexity, this not an impossible task. We have been dealing with these challenges on an ad hoc basis throughout history. The good news is that we now can use modern technologies and process thinking to provide all ingredients of successful effect based operations.

A network-centric system is a system-of-systems concept where a number of actors are attached to each other in a network sharing information in an adaptable and interoperable manner. Obviously networking enables an enormous rise in accessible information and the intrinsic challenge is the development of systems and functions to shape this information into guidance and control of a variety of operations with multiple objectives. For example, an optimization methodology is presented in [5] for finding the correct balance between weapons and attack damage assessment sensors.

The above mentioned pinpoints the trend in military operational planning, also at the Swedish military arena. In our case we can use this paradigm shift to put functional and algorithmic requirements on planning of air to ground missions. This leads to adaptation to new doctrines of command and control and to a tool that contains the most of planning experience implemented by planning specialist personnel in cooperation with algorithm experts. Mission performance can be driven to its limits with a model based planning which simultaneously keeps control of both objective and system performance, which is probably the most cost effective way to gain performance.

1.1 Network Centric Framework

In a network centric framework, a resource is not an entity tightly coupled to a sluggish hierarchical organization but a resource with own intelligence to offer specific effects to a variety of effect customers. Our work is entitled *Effect Oriented Planning* which indicates that it does not embrace the full meaning of EBO but is guided by quantifying and responding to effect requests and hence becoming a true entity of a network centric system. In order to understand the paradigm shift in EBO planning or network centric planning, Figure 1 shows the principles of future effect based operations.

Initially an effect must be achieved in order to answer what to do. Thereafter possible systems are considered and how theses systems could manage to do it. The last issue of the effect chain is to decide the resources allocation. As can be noticed, resource owners are considered in the later planning stages which is quite a change from traditional planning. Obviously there are two dimensions in the Effect chain, the mission-conduction and the resource owner dimension. The resource owner dimension keeps and conducts resources supply chain as well as allocation schemes and schedules. The mission-conduction states individual missions and how they shall be implemented.

Actually if a future EBO planning system shall apply to the above picture some extra requirements must be considered.

- Weapon systems must estimate associated effects and time duration of a mission
- Estimations must be performed rapidly
- Weapon systems must communicate actual state parameters such as position, health, endurance e.t.c.

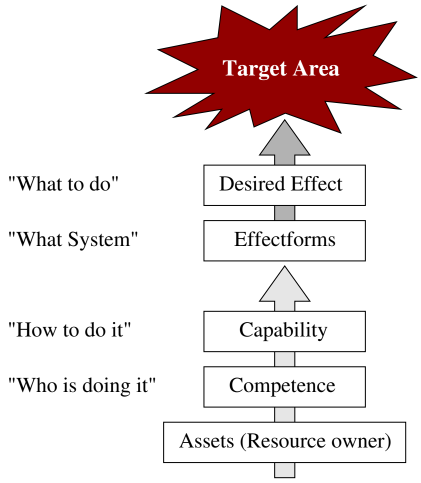


Figure 1: The effect chain including an EBO principle of a split up of the planning process into stages from the target to allocation of individual platforms

In order to fulfill these requirements on demand, effort must be put on scalable model-based algorithms which promotes an easy workflow and a high speed planning performance. Each scenario shall be individually stated by the set of input data, but planning shall always be performed via implemented tactics and knowledge of actual resource performance and mission pattern.

1.2 Mission Planning

An Air to Ground mission planning system is modular and contains a planning system and weapon systems, hosted by a variety of carriers such as UAV's or fighter aircraft. In order to perform effect oriented planning in line with Figure 1 we transform the planning process according to Figure 2 where each platform is separated into carrier and weapon performance and tactics producing a certain effect which can be matched with the effect customers needs.

Initially we maximize system effect in the target area by optimally allocating the number of weapons to suppress enemy defense and destroy vital targets. A target area can consist of different ground based targets and sheltering air defense units. Each target has a specific value which indicates the importance of the target. The effect oriented weapon allocation of the target area is followed by a search for appropriate platforms, where platform location and other scheduling parameters are considered. Further each platform

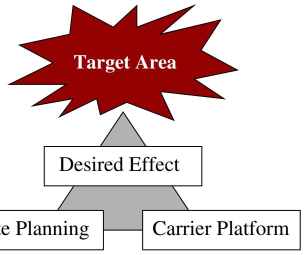


Figure 2: A resource, a fighter with weapon system, has a relationship between route planning, type of weapon and a set up of tactics which forms the final effect.

must have a route to the firing position including tactical features such as hiding and a limited exposure of radar cross section during the flight phase.

These planning aspects are coupled but with an acceptable loss of generality the effect planning task can be separated from the platform in order to start an overall planning process. The following paper address a model based approach to rapidly calculate weapon allocation to optimize system effect in an hostile ground based target area. Early work on a similar problem was done by Miercort and Soland in [4], but they consider a less complicated model without intricate dependencies. In a more recent paper by Kwon et. al [3], a new weapon-target allocation problem is presented together with a branch-and-price algorithm for solving it. In contrast, Kaminer and Ben-Asher present a model in [2] for maximizing the effectiveness of a defense.

These key issues in effect based planning shall of course be followed by other planning tasks but shall be seen as potential future work.

1.3 Paper Overview

Here follows a guide for the reader. In Section 2 we define and describe the problem at hand, basically a weapon-targeting problem, together with some basic concepts that will be used throughout the paper. The section ends with a generic mathematical model for the problem, found on page 12. The model is straight forward with only two linear constraints, but comes with a nasty non-convex and non-linear objective. In Sections 2.5 and 2.6 follows optimistic and pessimistic models that can be used to find upper and lower bounds on the true optimal objective value. These models are Linear Binary Problems and easy to solve.

In order to actually use the generic model and to solve real scenarios, it is necessary to specify in detail how to evaluate a given situation. One possible way to do this is presented in Section 3, where we state how the defenders act in different situations. A graphical example is found on page 17.

In Section 4 we present a Non-Linear Integer Programming model adapted to the rules defined in Section 3, followed by a Mixed Integer Linear Programming model in Section 5 where the non-linear objective function is approximated using piece-wise linear functions. In Section 6 we present some results on small and midsized scenarios, involving 2-6 units.

To solve bigger scenarios in practice, involving 5 to 20 units, Section 7 looks into different heuristic approaches who cannot guarantee optimality but find high quality solutions within a reasonable time frame. Sections 8 and 9 contain tests and results for the different heuristics considered in Section 7. Finally, in Section 10, we present some remarks and conclusions together with suggestions on future work.

2 The problem

Imagine a big wide open area, like a desert, where a number of interesting targets are positioned. It could be radar stations or other surveillance equipment, with or without defensive capabilities, which the attacker wishes to destroy. Moreover, the targets are possibly guarded by defending units, like Surface-to-Air Missile (SAM) units.

The positions of all units, targets and defenders, are known. The set of all units is denoted $\mathcal{S}$, and the subset $\bar{\mathcal{S}}$ denotes those units with defensive capabilities defined by radius of defense and armament. Each unit is given a specified reward r_s, where important units get high values and the other units are given low values.

This is the problem of the attacker, where the objective is to maximize the expected outcome of a joint attack against the enemy, subject to a limited amount of resources R, for example tanks and aircraft.

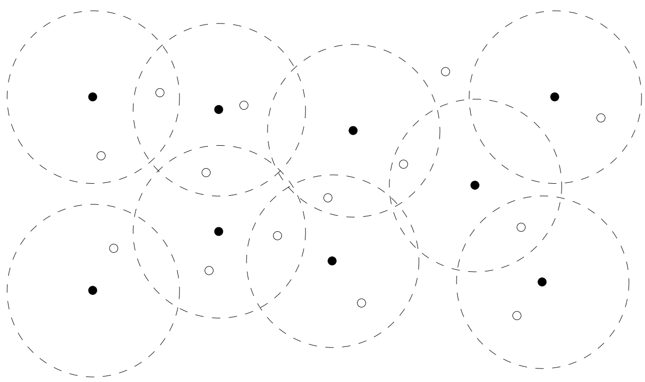


Figure 3: A possible attack scenario. Some units, here shown in black, are air defense units. The other units are radar stations or similar surveillance units who are valuable to destroy.

Each unit $s \in \mathcal{S}$ should be assigned an attack plan which specifies the number of resources to be used against it, and also from which direction.

As seen in Figure 3, some units does not have a defensive system of their own, but depends on the defense of other units. Also, the radius of defense for different units might overlap. A unit will always protect itself primarily, and then engage resources passing by inside its radius of defense towards other units.

2.1 Tactics

Each unit s, if attacked, is done so by a predefined tactic t, chosen from a set of tactics $\mathcal{T}$. Each such tactic have its own features, such as the number resources needed (n_t) and the number of attacking directions involved (V_t).

More important, each tactic t gives rise to a probability of success, for each of the n_t resources, against an isolated unit s. This probability is denoted p_{st} and might vary between each unit $s \in \mathcal{S}$, depending on their respective defensive capabilities.

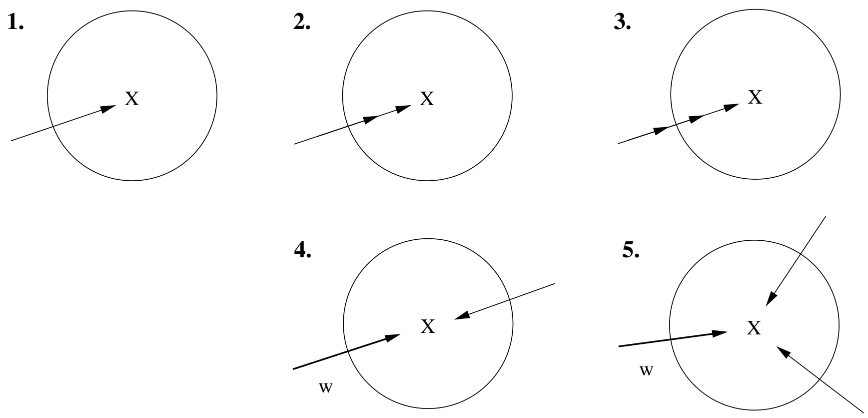


Figure 4: A graphical description of the 5 tactics considered.

At the moment, we limit ourselves to the set of tactics $\mathcal{T}$ described graphically in Figure 4. The idea behind these tactics is to overload the defensive system of a single unit. This is done by either sending multiple resources from the same direction (tactics 1-3), or by attacking simultaneously from multiple, evenly spread, angles (tactics 4-5).

In real life there is obviously an unlimited number of ways to perform an attack. The reason for limiting the attacks to predefined tactics is related to the difficulty of finding input data to the problem, in this case the probability of success for different attacks.

Each tactic $t \in \mathcal{T}$ is associated with a reference angle of attack, w, which defines from which direction the attack is launched. Since we only consider evenly spread angles of attack, one reference angle is enough.

2.2 Angle of attack

For tactics which involve more than one angle of attack, $V_t > 1$, multiple angles w might give rise to exactly the same attack since we consider evenly spread angles. To avoid such symmetries, we introduce a subset $\mathcal{W}_{st}$ which contains all valid reference angles w to be used together with tactic t against unit s.

In this paper, we consider a coarse angle discretization, consisting of evenly spread angles v defined by the set $\mathcal{V}$. For tactics involving multiple angles, we define

$$w_j = w + (j-1) \cdot \frac{2\pi}{V_t} \ , \quad j = 1, \ldots, V_t$$

We also introduce the concept of an engagement path (s, v), which is the line emanating from unit s at angle v. In total, there are $|\mathcal{S}| \cdot |\mathcal{V}|$ different engagement paths. For a certain tactic and angle, though, only a few of these paths will be used. If there is at least one resource on the path, we call it an *active path*.

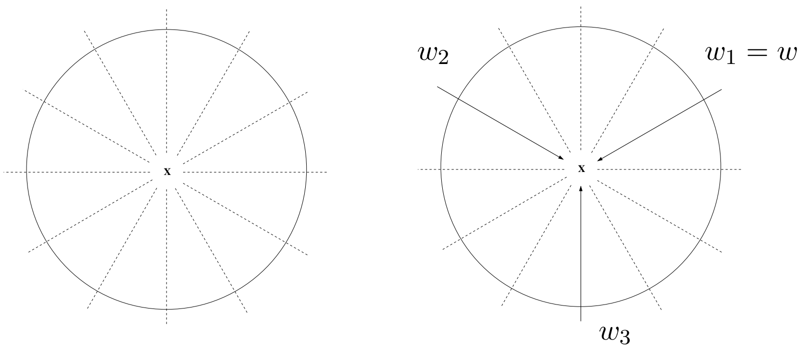


Figure 5: To the left, all possible engagement paths (s, v) toward a unit for a given angle discretization, here 12 evenly spread angles. To the right, the unit is attacked using tactic number 5 and reference angle of attack w. This gives rise to three active engagement paths toward the unit.

For the given angle discretization $\mathcal{V}$ shown in Figure 5, the set $\mathcal{W}_{st}$ for tactic $t = 5$ would consist of only the first four angles in order to avoid redundant attacks (symmetry). Although $\mathcal{W}_{s5} = \{1, 2, 3, 4\}$, engagement paths along all angles $v \in \mathcal{V}$ become active as multiple angles are involved for this tactic.

Throughout this paper, a reference angle of attack is always denoted w and defined by the set $\mathcal{W}_{st}$, whereas an angle v refers to an individual angle in $\mathcal{V}$ used for general discussions involving engagement paths (s, v).

2.3 The objective

The essence of this problem is to decide, for each unit s, which tactic t that should be used (if any) and specify a reference angle of attack w. We introduce the binary variable z_{stw} to be one if unit s should be attacked by tactic t from reference angle w.

$$z_{stw} = \begin{cases} 1 & \text{if unit } s \text{ is attacked using tactic } t \text{ from angle } w. \\ 0 & \text{otherwise} \end{cases}$$

A collection of such decisions, at most one for each unit s, is defined as an attack plan $\mathbf{z}$. Let $p^{\text{kill}}_{stw}(\mathbf{z})$ be the probability of successfully incapacitate unit s when attacked by tactic t from reference angle w. As will be clear from the upcoming analysis, this probability depends on the overall attack plan $\mathbf{z}$, a fact which complicate things.

The objective is to maximize the expected reward of the attack, found by multiplying the probability of success of an attack against a unit with its reward. Since we want to optimize the whole attack, these expected values should be added. The objective becomes

$$\max \sum_{s \in S} \left[\sum_{t \in T} \sum_{w \in \mathcal{W}_{st}} p^{\text{kill}}_{stw}(\mathbf{z}) \cdot z_{stw} \right] \cdot r_s$$

where r_s is the associated reward (or price) for each unit $s \in \mathcal{S}$.

2.3.1 The probability of success

In order to define the expression for $p^{\text{kill}}_{stw}(\mathbf{z})$, let us analyze the situation in Figure 6. Unit 2 is attacked with tactic $t = 2$, i.e. two resources from the same angle of attack w_2. This engagement path (s_2, w_2) does not intersect the area of defense for unit 1, hence the only threat for our resources is the defense of unit 2. For a single resource, $p^{\text{kill}}_{stw}(\mathbf{z})$ is the given probability p_{st}.

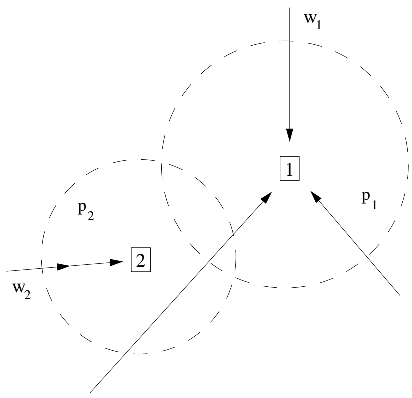


Figure 6: A possible attack situation. Unit 2 is attacked by two resources from the same angle, i.e. tactic 2. Unit 1 is attacked using tactic 5, i.e. one resource from three different angles.

For two resources, the probability of incapacitating the unit is equal to 1 minus the probability that neither resource survives the defense of unit 2. That is,

$$p^{\text{kill}} = 1 - (1 - p_{st})^2 \qquad \text{for } s = 2,\ t = 2$$

The case for unit 1 is more complicated since one of the active engagement paths intersects the area of defense for unit 2. The probability for our resource to survive this defense will vary depending on how busy unit 2 is defending itself, since we assume that a unit always defend itself primarily.

The probability for a resource to survive the defense of a unit i which it passes by on its way towards the target s on path (s, v) depends on the attack plan $\mathbf{z}$, i.e. what tactics are used against the surrounding units. We denote this dependence by $p_{isv}(\mathbf{z})$.

In all, the probability of success using the tactic in Figure 6 against unit 1 is equal to 1 minus the probability that none of the three resources survive. That is,

$$p^{\text{kill}} = 1 - (1 - p_{st})^2 \cdot (1 - p_{st} \cdot p_{isv})$$

Two of the engagement paths only intersect the area of defense for unit 1 and the third path intersects also the area of defense for unit 2, which is reflected in the expression above.

We now generalize this way of calculating the probability of success for a certain tactic and reference angle of attack. For a given unit s, tactic t and angle of attack w the probability of successfully eliminating unit s is:

$$p^{\text{kill}}_{stw}(\mathbf{z}) \quad = \quad 1 - \prod_{j=1}^{V_t} \left[1 - p_{st} \prod_{\substack{i \in \mathcal{S} \\ i \neq s}} p_{isw_j}(\mathbf{z}) \right]^{m_t} \tag{1}$$

For the given tactic t, the number of resources that is launched from each of the angles V_t is denoted by m_t. Naturally, whenever an engagement path (s, v) doesn't intersect the area of defense for unit i, we set $p_{isv}(\mathbf{z}) = 1$.

The expression for $p^{\text{kill}}_{stw}(\mathbf{z})$ is very complex, since it needs to incorporate many things. The success of an attack against a certain unit depends on

1. the number of resources used against the unit ($n_t = V_t \cdot m_t$).
2. the units ability to defend itself against incoming resources (p_{st}).
3. the probability of successfully survive the defense of every other unit which the resource pass by on its way towards the target (p_{isw_j}).

Both n_t and p_{st} are given data for the problem, so they pose no problem. It is the third one, p_{isw_j}, which makes this problem very complicated.

That is, everything is connected since the probability of success for a tactic t and angle w against a unit s depends on which tactics are applied against every other unit. This dependence is the core of the problem and is very troublesome.

2.4 A generic model

Let $p^{\text{kill}}_{stw}(\mathbf{z})$ be the probability of successfully incapacitate unit s when attacked by tactic t from reference angle w, defined as in (1). The probability for a resource to survive as it passes by unit i towards unit s on path (s, v), denoted $p_{isv}(\mathbf{z})$, depends on the attack plan $\mathbf{z}$, but for the generic model we make no assumptions on the exact nature of this dependence.

$$\max \quad \sum_{s \in S} \left[\sum_{t \in T} \sum_{w \in \mathcal{W}_{st}} p^{\text{kill}}_{stw}(\mathbf{z}) \cdot z_{stw} \right] \cdot r_s \qquad [GENERIC]$$

$$\begin{aligned}
s.t. \quad & \sum_s \sum_t \sum_{w \in \mathcal{W}_{st}} n_t \cdot z_{stw} \leq R && (1)\\
& \sum_t \sum_{w \in \mathcal{W}_{st}} z_{stw} \leq 1 \quad \forall\, s \in \mathcal{S} && (2)\\
& z_{stw} \in \{0, 1\} \quad \forall\, s \in \mathcal{S}, t \in \mathcal{T}, w \in \mathcal{W}_{st}
\end{aligned}$$

Notice that it is not necessary to attack all units. Depending on the rewards specified for each unit, this might not be optimal. Constraint (1) states that we cannot use more resources than we have. Constraint (2) makes sure that each unit is attacked at most once. Both constraints are linear, but the objective is definitely not. In Section 5, we present a way to approximate it using some clever rewriting and piecewise linear approximations. In all, we manage to present a linear model which approximates the nasty objective arbitrarily well.

2.5 Optimistic model

It is possible to construct two auxiliary problems, providing an upper and lower bound to the generic problem. Let us analyze (1), the expression for $p^{\text{kill}}_{stw}(\mathbf{z})$, under two specific assumptions.

Assume that no unit will shoot against resources passing by towards other units, just against resources towards themselves. This means that $p_{isv}(\mathbf{z}) \equiv 1$ for all units $s \in \mathcal{S}$, and $p^{\text{kill}}_{stw}(\mathbf{z})$ collapses to

$$p^{\text{kill}}_{stw}(\mathbf{z}) \;=\; 1 - \prod_{j=1}^{V_t} \Big[1 - p_{st} \prod_{\substack{i \in \bar{\mathcal{S}} \\ i \neq s}} 1 \Big]^{m_t} \;=\; 1 - (1 - p_{st})^{n_t}$$

This expression does not depend on the angle w anymore, hence we only have to decide which tactic t to use against each unit s, if any tactic at all. The most important thing is that the probabilities of success no longer depend on the overall attack plan $\mathbf{z}$. Since $p^{\text{kill}}_{stw}(\mathbf{z})$ only depends on the given probabilities p_{st}, they are also (implicitly) given as input data. The probabilities are now separable in s.

We define $P_{st} = 1 - (1 - p_{st})^{n_t}$, and the optimization model becomes

$$\begin{array}{llllll} \max & \sum_s \sum_t & r_s \cdot P_{st} \cdot z_{st} & & & [OPTIMISTIC] \\ \\ s.t. & \sum_s \sum_t & n_t \cdot z_{st} & \leq & R & & (1) \\ & \sum_t & z_{st} & \leq & 1 & \forall\, s \in \mathcal{S} & (2) \\ & & z_{st} & \in & \{0,1\} & \forall\, s \in \mathcal{S}, t \in \mathcal{T} & \end{array}$$

This linear binary problem is easily solved using CPLEX. Solutions to this optimistic model are valid upper bounds for the original problem since the values of all coefficients in the objective are systematically increased. Even more, this is a valid upper bound for all choices of discretization $\mathcal{V}$.

The solution found is also a feasible solution to the original problem, if complemented with an arbitrary reference angle of attack for each tactic used. It means that we can easily calculate the "true" objective value and also get a lower bound. This bound is only valid for the considered discretization $\mathcal{V}$ though.

2.6 Pessimistic model

It is also possible to find a pessimistic model, generating lower bounds to the original problem. By assuming that each unit will shoot against all resources passing by towards other units with full force, one can find a pessimistic value for the probability of surviving the defense of other units, $p_{isv}(\mathbf{z}) \equiv \tilde{p}_{isv}$. The expression for $p^{\text{kill}}_{stw}(\mathbf{z})$ thus become

$$p^{\text{kill}}_{stw}(\mathbf{z}) \;=\; 1 - \prod_{j=1}^{V_t} \left[1 - p_{st} \prod_{\substack{i \in \bar{\mathcal{S}} \\ i \neq s}} \tilde{p}_{isw_j} \right]^{m_t}$$

Using $P_{stw} = p^{\text{kill}}_{stw}(\mathbf{z})$ as above, the optimization model becomes

$$\begin{array}{llllll} \max & \sum_s \sum_t \sum_{w \in W_{st}} & r_s \cdot P_{stw} \cdot z_{stw} & & & [PESSIMISTIC] \\ \\ s.t. & \sum_s \sum_t \sum_{w \in W_{st}} & n_t \cdot z_{stw} & \leq & R & & (1) \\ & \sum_t \sum_{w \in W_{st}} & z_{stw} & \leq & 1 & \forall\, s \in \mathcal{S} & (2) \\ & & z_{stw} & \in & \{0,1\} & \forall\, s \in \mathcal{S}, t \in \mathcal{T}, w \in \mathcal{W}_{st} & \end{array}$$

and is easily solved using CPLEX.

The values of $\tilde{p}_{isv}$ might become unrealistically pessimistic, since the assumption is extreme, and hence the solution will probably provide bad lower bounds on the optimal objective value.

Hopefully, though, the structure of the solution (the attack plan $\mathbf{z}$) is close to the optimal one, and by evaluation in the real objective one can find a better pessimistic bound.

3 Simulation details

In order to actually solve the generic model presented in Section 2.4, one needs to specify how the probability $p_{isv}(\mathbf{z})$ depends on the attack plan $\mathbf{z}$. It is obviously an impossible task to model a fully realistic case, and not very meaningful in practice due to the amount of input data needed for such a model.

We will analyze the different factors that affects $p_{isv}(\mathbf{z})$, the probability for a resource to survive the defense of a unit as it passes by toward its target, and how it depends on $\mathbf{z}$. To do this, we look into the details of the defensive systems of the units and define their rules of engagement.

Some factors are the amount of defensive capacity, distance between the unit and the engagement path as well as limitations in the defensive systems.

3.1 Residual Defensive Capacity

To start with, each unit s has a specified number of defensive channels, i.e cannons and anti-missile systems, denoted C_s. We define the set $\bar{\mathcal{S}}$ as the set of units with $C_s > 0$, i.e. units with any defensive capability. Units in $\bar{\mathcal{S}}$ will be indexed by i from now on, and their radius of defense is denoted ρ_i.

For each unit $i \in \bar{\mathcal{S}}$ and defined tactics $t \in \mathcal{T}$, the parameter d_{it} states how many of these defensive channels will be occupied when unit i takes on the incoming resources defined by tactic t.

Since a unit always defend itself primarily, we define the residual defensive capacity D_i for each unit $i \in \bar{\mathcal{S}}$ as the number of defensive channels available after the allocation of d_{it} channels required to handle the incoming attack.

These residual channels should be used to defend the other units, by engaging resources passing by inside their area of defense. So for each unit $i \in \bar{\mathcal{S}}$, we introduce integer variables u_{isv} to decide how many defensive channels that should be allocated against the resources on each active path (s, v) passing by.

3.2 Active Paths and Distance Measure

The number of active engagement paths passing by a unit i is denoted B_i and defined as the number of active paths that intersect the area of defense for each unit $i \in \bar{\mathcal{S}}$, i.e. possible targets for the unused defensive channels. The area of defense for each unit i is the circle defined by its defense radius ρ_i.

The number of resources on each path, denoted N_{sv}, also affects the probability of success. We define $K = \max_{t \in \mathcal{T}} \{n_t \ : \ V_t = 1\}$ to be the maximum number of resources traveling on a single engagement path. Hence, N_{sv} is in the range $k = 0, \ldots, K$.

We define parameter d_{isv} to be the orthogonal distance between unit i and the engagement path (s, v). For units with positions inside the area of defense of unit i, the distance to the mid-point of this path is used. This is illustrated in Figure 7.

3.3 Ranking

Each active path is given a rank number, where the path closest to unit i gets rank 1, second closest path is ranked 2 and so on. Closest path refers to the smallest distance d_{isv} and is thus relative to each unit i. This ranking will be used when the defense units need to prioritize, i.e. when they cannot engage all paths passing by.

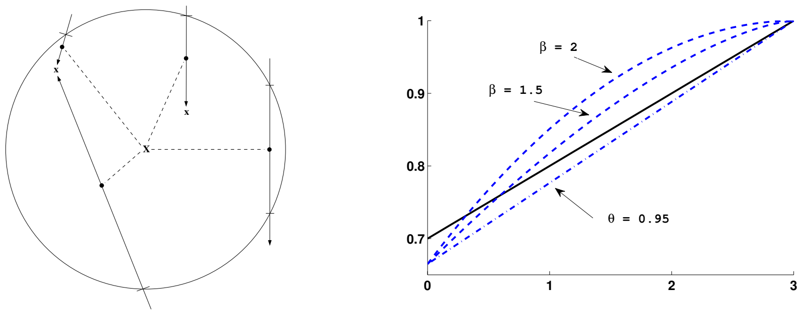


Figure 7: To the left, an illustration on how the distance between a unit and the active engagement path is measured. To the right, an example on how the design parameters β_{ik} and θ_{ik} affects the probability p^k_{isv}.

3.4 Introducing p^k_{isv}

The probability p_{isv} is a function of the distance d_{isv} and the number of resources on the path N_{sv}, who are both a direct consequence of the attack plan $\mathbf{z}$. The obvious way to model this dependence would be to demand values for all such combinations as input data, but this is not possible in practice. We introduce an analytical function instead, based on both d_{isv} and N_{sv}.

We define p_{isv}^k to be the probability for a resource to successfully pass by one defensive channel of unit i, which also depends on $k = N_{sv}$, the number of resources that are traveling on the active path (s, v).

These probabilities are derived from the values of p_{st}, for tactics $t \in \mathcal{T}$ where all $k = n_t$ resources are sent from the same angle ($V_t = 1$). Since this is only relevant for units in $\bar{\mathcal{S}}$, we denote this p_{ik} for all $i \in \bar{\mathcal{S}}$, $k = 1, \ldots, K$.

$$p_{isv}^k \quad = \quad 1 - \left(1 - \frac{d_{isv}}{\rho_i}\right)^{\beta_{ik}} \cdot (1 - \theta_{ik} \cdot p_{ik}) \qquad (2)$$

Here, β_{ik} and θ_{ik} are design parameters used to model the defensive capacities of each unit i against different number of resources k. In Figure 7, one can see in the left picture how the distance between a unit and an active engagement path is defined. This distance, denoted d_{isv}, is then used to derive a value for p_{isv}^k. The rightmost plot in Figure 7 shows the probability p_{isv}^k on the y-axis as a function of the distance d_{isv} on the x-axis.

For this example, probability $p_{ik} = 0.7$ is used, and the black line corresponds to parameter values $\beta_{ik} = 1$ and $\theta_{ik} = 1$. The dash-dotted line (blue) illustrates the effect of parameter θ_{ik}, as its value is set to 0.95. The two dashed curves (blue) correspond to the values of 1.5 and 2 respectively for parameter β_{ik}.

In all, this analytical function shows a natural behaviour. For $d_{isv} = 0$, its value becomes $\theta_{ik} \cdot p_{ik}$ and for $d_{isv} = \rho_i$ the probability becomes 1. For distances in between, the parameter β_{ik} is used to control the effectiveness of the defensive system of unit i.

3.5 Specifications of the defense system

Since we consider the problem of the attacker, we need to assume and specify a set of deterministic engagement rules for the defenders. We make the following assumptions for each unit $i \in \bar{\mathcal{S}}$:

1. A unit i will primarily defend itself.
2. If there are some residual defensive channels, $D_i > 0$, they will be evenly
   allocated against the active engagement paths that pass by the unit.
3. At most F_i channels might be used against a single engagement path.
4. At most G_i different engagement paths might be engaged.
5. All defensive channels should be used if there is something to shoot at.
6. If there are more active paths than defensive channels, one defensive channel is allocated to each path as long as possible with respect to their ranking, i.e. starting with the path closest to the unit.

To make sure that variables u_{isv} follow these rules, we need to introduce auxiliary variables and numerous logical constraints. These are found in Section 4 where we present a mathematical model for this problem.

3.6 Defining $p_{isv}(\mathbf{z})$

Now finally, we are able to define p_{isv}, the probability for a resource to survive as it passes by unit $i \in \bar{\mathcal{S}}$ towards unit $s \in \mathcal{S}$ on path (s, v):

$$p_{isv} = \prod_{k=1}^{K} \left(p_{isv}^{k}\right)^{u_{isv}^{k}} \tag{3}$$

Variable u_{isv}^k is an auxiliary integer variable, equal to u_{isv} if $k = N_{sv}$ and zero otherwise. This construction is necessary in order to keep all constraints linear. A mathematical model is presented in the upcoming Section 4.

Since u_{isv}, and thus also u_{isv}^k, might be greater than one the probability of success decreases with the number of defensive channels assigned to the engagement path. This is realistic as the defensive channels can be seen as independent, and the probability for a resource to survive two channels should be equal to the probability of surviving them both.

3.7 Illustrative Example

In this example, we focus on a unit i with some given defensive radius. Assume that we name all paths (s, v) intersecting the area of defense in accordance with their rank, i.e. the path with rank 1 is also named path 1 (and so on).

Furthermore, we assume defensive parameters $G_i = 4$ and $F_i = 3$, and that the number of residual defensive channels $D_i = 5$.

The situation is illustrated in Figure 8. Notice that one of the engagement paths never intersect the area of defense. This path is never considered when the residual defensive channels are assigned.

Case I, all paths are active

Consider the case where $B_i = 4$, i.e. all four paths passing by unit i are active (at least one resource travelling on the path). Since $B_i \leq G_i$ all paths should be engaged. Also, since $D_i > B_i$, each path get at least one defensive channel locked against them. The remaining one is assigned to the path closest to unit i, i.e. path 1 with rank 1. The variables u_{isv} for unit i becomes $u_{i1} = 2$, $u_{i2} = 1$, $u_{i3} = 1$ and $u_{i4} = 1$.

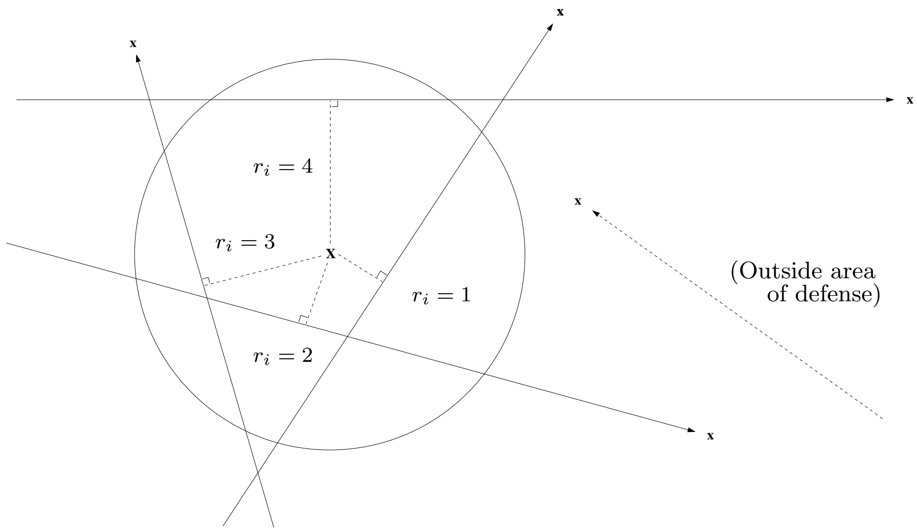


Figure 8: A situation where multiple engagement paths intersect the area of defense for a unit i. There is also one path that does not affect unit i.

Case II, three active paths

In the case where either B_i or G_i (or both) decreases to 3, unit i can only engage 3 engagement paths.

For $B_i = 4$ and $G_i = 3$, the path with highest rank (most far away) will no longer be engaged. The residual defensive channels are then distributed as follows: $u_{i1} = 2$, $u_{i2} = 2$, $u_{i3} = 1$ and $u_{i4} = 0$.

If $B_i = 3$ and $G_i = 3$ (or 4), it means that only three engagement paths are active. Depending on which path that is not active, the other paths are assigned defensive channels like before, with respect to rank. Assume that for example path 2 (with rank 2) is not active, then we get: $u_{i1} = 2$, $u_{i2} = 0$, $u_{i3} = 2$ and $u_{i4} = 1$.

Case III, only one active path

Finally, if $B_i < 2$, all defensive channels cannot be assigned to an engagement path since $F_i = 3$. With only one (or none) active path, at most $B_i \cdot F_i \leq 1 \cdot 3 = 3$ channels could be assigned. For example, if only path 3 is active, we get: $u_{i1} = 0$, $u_{i2} = 0$, $u_{i3} = 3$ and $u_{i4} = 0$.

In the mathematical model, a slack variable is introduced to handle these cases.

4 Mathematical Model

In this section, we introduce the mathematical model and describe the constraints whom are divided into five groups. This should improve the readability of the model, and also ease the comparison of the different model versions that will be described. In most cases some groups are unchanged or just slightly modified, whereas other groups are changed drastically.

It should be stated that if one aims at solving the problem using some meta-heuristic, it suffices to consider constraints (1) and (2), i.e. the generic problem described on page 12. These are the only real constraints of the problem, all the rest are used to model the behaviour of the units with defensive capabilities and are uniquely defined for a given attack plan $\mathbf{z}$.

4.1 Notation

Variables

integer

B_i	number of active engagement paths (s, v) passing by unit i.
D_i	residual defensive capacity for unit i.
S_i	slack variable for the residual defense quota for unit i.
N_i	help variable, $N_i = \min\{B_i, G_i\}$.
u_{isv}	number of defensive channels that unit i will use against resources on path (s, v).
u^k_{isv}	equal to u_{isv} if $n_{svk} = 1$, zero otherwise.
N_{sv}	number of resources on path (s, v).

binary

z_{stw}	1 if unit s is attacked using tactic t and angle w, where $w \in \mathcal{W}_{st}$.
x_{sv}	1 if any resource travels toward unit s on path (s, v).
n_{svk}	1 if $N_{sv} = k$, zero otherwise.
y_i	1 if $B_i \geq D_i$ for each unit i.
q_i	1 if $D_i \leq F_i \cdot \min\{G_i, B_i\}$ for each unit i.
z_i	1 if $B_i \geq G_i$ for each unit i.
U_{isv}	1 if unit i will use any defensive channel against path (s, v).

Definitions

engagement path	(s, v)	$:=$	the line emanating from unit s at angle v.
attack plan	$\mathbf{z}$	$:=$	a collection of z-variables, one for each unit s, which defines a tactic t and angle w.

Sets

given

$\mathcal{S}$, $\mathcal{T}$, $\mathcal{V}$	set of units $\mathcal{S}$, tactics $\mathcal{T}$ and angle discretization $\mathcal{V}$.
$\bar{\mathcal{S}}$	set of units with defensive capabilities. Subset of $\mathcal{S}$.

pre-processed

$\mathcal{W}_{st}$	set of feasible angles w for tactic t against unit s.
Δ_i	set of paths (s, v) that passes by unit $i \in \bar{\mathcal{S}}$.
$\mathcal{R}_i$	set of triplets $(i, sv, \bar{s}\bar{v})$ where $r_{isv} < r_{i\bar{s}\bar{v}}$.

Parameters

given as input data

R	total resource available, the amount of resources.
r_s	reward (value/price) of unit s.
n_t	# resources used by tactic t.
V_t	# angles used by tactic t.
m_t	# resources/angle used by tactic t.
C_i	number of defensive channels for unit $i \in \bar{\mathcal{S}}$.
F_i	maximum number of defensive channels against a path.
G_i	maximum number of paths that unit i can engage.
ρ_i	radius of defense for unit i.
d_{it}	# defensive channels used by unit i when attacked by tactic t.
p_{st}	probability that a resource survives the defense of unit s when part of tactic t.

pre-processed, i.e. derived from the given parameters

K	maximum number of resources/angle, i.e. $\max\{\, m_t,\ t = 1, \dots \lvert\mathcal{T}\rvert\}$.
M_i	maximum number of engagement paths that passes by unit i.
A_{vtw}	1 if path (s, v) is active when the combination of tactic t and angle w is used against some unit s, where $w \in \mathcal{W}_{st}$.
d_{isv}	distance from unit i to center point of path (s, v) inside ρ_i.
δ_{isv}	1 if $d_{isv} < \rho_i$. Indicates which paths unit i might engage. Used to define set Δ_i.
r_{isv}	ranking of paths (s, v) passing by each unit i, where $\delta_{isv} = 1$. The shorter distance d_{isv}, the closer to the unit and lower ranking.
p_{ik}	probability p_{st}, defined only for units $i \in \bar{\mathcal{S}}$, where $V_t = 1$ and $m_t = k$, for $k = 1, \dots, K$.
p^k_{isv}	probability of surviving the defense of unit i for a resource on path (s, v) who is part of an attack t where $m_t = k$.

4.2 The Non-Linear Integer Programming Model

In this general model we assume limiting values for both parameter F_i and G_i. Although, since it is possible to find out in advance if $M_i \leq G_i$ or $C_i \leq F_i$, or both, for each unit $i \in \bar{\mathcal{S}}$, many constraints are redundant and some variables are unnecessary. In these cases, it is possible to reformulate some of the constraints in order to avoid unnecessary work.

All constraints are linear, it is the nasty objective which makes the problem non-linear. Constraints (1) and (2), found already in the generic model, make sure we use no more resources than available and that each unit s is attacked at most once.

$$\max \sum_{s\in S} \left[\sum_{t\in T} \sum_{w\in \mathcal{W}_{st}} p^{\text{kill}}_{stw}(\mathbf{z}) \cdot z_{stw} \right] \cdot r_s \qquad [NLIP]$$

$$\begin{aligned} s.t. \quad & \sum_s \sum_t \sum_{w\in\mathcal{W}_{st}} n_t \cdot z_{stw} && \leq R && (1) \\ & \sum_t \sum_{w\in\mathcal{W}_{st}} z_{stw} && \leq 1 \quad \forall\, s \in \mathcal{S} && (2) \end{aligned}$$

The rest of the constraints are divided into five groups and presented in the following subsections. The full NLIP model can be found on page 56.

4.2.1 The Objective

The non-linear function $p^{\text{kill}}_{stw}(\mathbf{z})$ is part of the objective, and defined below. Since it is only units $i \in \bar{\mathcal{S}}$ that can defend other units, the general formula from Section 2.3 is now altered somewhat.

$$p^{\text{kill}}_{stw}(\mathbf{z}) = 1 - \prod_{j=1}^{V_t} \left[1 - p_{st} \prod_{\substack{i\in\bar{\mathcal{S}} \\ i\neq s}} \prod_{k=1}^{K} \left(p^k_{isw_j} \right)^{u^k_{isw_j}} \right]^{m_t}$$

The values of variables u^k_{isv} are dependent on the entire attack plan $\mathbf{z}$, which makes the problem very difficult. Once their values are known, it is straight forward to evaluate the objective.

4.2.2 Group I

The constraints in this first group are mostly constraints used to define auxiliary variables and might actually be removed, but are kept for the sake of readability.

$$
\begin{array}{rlcll}
\sum_t \sum_{w \in \mathcal{W}_{st}} & A_{vtw} \cdot z_{stw} & = & x_{sv} & \forall\, s \in \mathcal{S},\ v \in \mathcal{V} \quad (3) \\
C_i - \sum_t \sum_{w \in \mathcal{W}_{it}} & d_{it} \cdot z_{itw} & = & D_i & \forall\, i \in \bar{\mathcal{S}} \quad (4) \\
\sum_{(s,v) \in \Delta_i} & x_{sv} & = & B_i & \forall\, i \in \bar{\mathcal{S}} \quad (5) \\
\sum_s \sum_v & u_{isv} \ + \ S_i & = & D_i & \forall\, i \in \bar{\mathcal{S}} \quad (6)
\end{array}
$$

Constraints (3) – (5) defines help variables x_{sv}, D_i and B_i. Variable D_i defines the number of residual defensive channels for each unit i. Variable x_{sv} keeps track of which engagement paths that are active for the current attack plan $\mathbf{z}$. Variable B_i defines the number of active paths (s, v) which pass by unit i. Constraint (6) forces each unit i to allocate the correct number of residual defensive channels D_i. Also, in order to handle the case where no engagement paths (s, v) are active, and hence all u_{isv} will become 0, a slack variable S_i is introduced. Further constraints are introduced in order to make sure that variable S_i is used only when this is the case.

4.2.3 Group II

The second group of constraints defines all logical variables and implications.

$$
\begin{array}{lcll}
D_i - F_i \cdot N_i + (F_i M_i) \cdot q_i & \geq & S_i & \forall\, i \in \bar{\mathcal{S}} \quad (7a) \\
D_i - F_i \cdot N_i & \leq & S_i & \forall\, i \in \bar{\mathcal{S}} \quad (7b) \\
C_i \cdot (1 - q_i) & \geq & S_i & \forall\, i \in \bar{\mathcal{S}} \quad (7c) \\
F_i \cdot N_i + C_i \cdot (1 - q_i) & \geq & D_i & \forall\, i \in \bar{\mathcal{S}} \quad (8a) \\
D_i - 1 + (F_i M_i + 1) \cdot q_i & \geq & F_i \cdot N_i & \forall\, i \in \bar{\mathcal{S}} \quad (8b) \\
D_i + M_i \cdot y_i & \geq & B_i & \forall\, i \in \bar{\mathcal{S}} \quad (9a) \\
B_i + C_i \cdot (1 - y_i) & \geq & D_i & \forall\, i \in \bar{\mathcal{S}} \quad (9b) \\
G_i & \geq & N_i & \forall\, i \in \bar{\mathcal{S}} \quad (10a) \\
B_i & \geq & N_i & \forall\, i \in \bar{\mathcal{S}} \quad (10b) \\
G_i - M_i \cdot (1 - z_i) & \leq & N_i & \forall\, i \in \bar{\mathcal{S}} \quad (10c) \\
B_i - M_i \cdot z_i & \leq & N_i & \forall\, i \in \bar{\mathcal{S}} \quad (10d)
\end{array}
$$

Constraints $(7a)$ – $(7c)$ are used to assign the correct value to the slack variable S_i. If variable $q_i = 0$, the slack becomes $S_i = D_i - F_i \cdot N_i$ which is exactly the number of channels that cannot be assigned. Otherwise, it means that no slack is necessary and forces $S_i = 0$.

Constraints $(8a) - (8b)$ defines the value of q_i. Either $D_i \leq F_i \cdot N_i$ which means that it is possible to assign all channels to active engagement paths, and hence the slack should be zero. Otherwise it is not possible to assign all residual defensive channels and the slack variable should become active.

In a similar way, constraints $(9a) - (9b)$ defines the value of y_i. If $B_i \geq D_i$ variable y_i is forced to become 1, which in upcoming constraints might restrict variable u_{isv} to either 0 or 1. Constraints $(10a) - (10d)$ defines the value of N_i, which is a linear way of rewriting the constraint $N_i = \min\{B_i, G_i\}$.

4.2.4 Group III

Constraints $(11) - (17)$ make sure that each unit i allocate their residual defensive channels D_i according to the set of rules defined and motivated in Section 3.5.

$$F_i \cdot U_{isv} \geq u_{isv} \quad \forall\, i,\ (s,v) \in \Delta_i \tag{11a}$$

$$F_i \cdot x_{sv} \geq u_{isv} \quad \forall\, i,\ (s,v) \in \Delta_i \tag{11b}$$

$$1 + (F_i - 1) \cdot (1 + z_i - y_i) \geq u_{isv} \quad \forall\, i,\ (s,v) \in \Delta_i \tag{12}$$

$$u_{isv} + F_i \cdot (1 - x_{sv}) \geq u_{i\bar{s}\bar{v}} \quad \forall\, i,\ (sv, \bar{s}\bar{v}) \in \mathcal{R}_i \tag{13a}$$

$$U_{isv} + (1 - x_{sv}) \geq U_{i\bar{s}\bar{v}} \quad \forall\, i,\ (sv, \bar{s}\bar{v}) \in \mathcal{R}_i \tag{13b}$$

$$u_{i\bar{s}\bar{v}} + 1 + (F_i - 1) \cdot (1 - U_{i\bar{s}\bar{v}}) \geq u_{isv} \quad \forall\, i,\ (sv, \bar{s}\bar{v}) \in \mathcal{R}_i \tag{14}$$

$$F_i \cdot (U_{isv} - q_i) \leq u_{isv} \quad \forall\, i,\ (s,v) \in \Delta_i \tag{15}$$

$$U_{isv} \leq x_{sv} \quad \forall\, i,\ (s,v) \in \Delta_i \tag{16}$$

$$\sum_{(s,v) \in \Delta_i} U_{isv} = N_i \quad \forall\, i \in \bar{\mathcal{S}} \tag{17}$$

Since we assume $M_i > G_i$ it means that we might need to limit the number of engagement paths to which we assign defensive channels. This is taken care of by constraint (17) where we limit the number of U_{isv} that can be set to one.

Constraint (16) states that if a path is not active, we cannot assign defensive channels to it. Constraint $(11a)$ forces u_{isv} to be zero if the corresponding U_{isv} is zero, i.e. no defensive channels will be assigned. Otherwise, u_{isv} is limited by F_i. Constraint $(11b)$ is actually redundant and should probably be removed.

Constraint (15) forces the value of u_{isv} to F_i if all paths that we decide to engage should be engaged fully. For this case to happen, $U_{isv} = 1$ and $q_i = 0$, which means that the path should be engaged and there is more residual defensive capacity than could be used. This is a redundant constraint.

Constraint (12) limits the value of u_{isv} to zero or one, depending on the values of y_i and z_i. In the case where $y_i = 1$ and $z_i = 0$, there are more active engagement paths than defensive channels, but not more than G_i, and hence each path should get at most one defensive channel assigned to it.

Finally, constraints (13) – (14) are precedence constraints which forces the assignment of resources to be done according to the rules from Section 3.5. That is, shoot against engagement paths in decreasing order with respect to distance. Hence, the further away from unit i, the less likely to be shot at.

4.2.5 Group IV

Constraints (18) – (22) are used to model how p_{isv}, the probability for a resource to survive as it passes by unit i on path (s, v), is affected by the number of resources on the active path.

$$\sum_t \sum_{w \in \mathcal{W}_{st}} A_{vtw} \cdot m_t \cdot z_{stw} = N_{sv} \qquad \forall\, s \in \mathcal{S},\ v \in \mathcal{V} \tag{18}$$

$$\sum_{k=1}^{K} k \cdot n_{svk} = N_{sv} \qquad \forall\, s \in \mathcal{S},\ v \in \mathcal{V} \tag{19}$$

$$\sum_{k=1}^{K} n_{svk} \leq 1 \qquad \forall\, s \in \mathcal{S},\ v \in \mathcal{V} \tag{20}$$

$$\sum_{k=1}^{K} u_{isv}^{k} = u_{isv} \qquad \forall\, i,\ (s,v) \in \Delta_i \tag{21}$$

$$F_i \cdot n_{svk} \geq u_{isv}^{k} \qquad \forall\, i,\ (s,v) \in \Delta_i,\quad k = 1, \ldots, K \tag{22}$$

Constraint (18) defines N_{sv}, the number of resources on path (s, v), and in constraint (19) the appropriate n_{svk} is set to 1. These two constraints can of course be combined, and hence eliminating variable N_{sv}. Constraint (20) makes sure that for example $N_{sv} = 3$ is not represented as $1 + 2$.

Constraint (21) in combination with constraint (22) assigns the correct number of defensive channels to the correct variable u_{isv}^{k}, i.e. the one corresponding to $n_{svk} = 1$.

4.2.6 Group X

Defines all variables. In an upcoming section, there will be a fifth group of constraints, used to linearize the objective. This is the reason for the strange numbering of constraints in Group X.

$$B_i,\ D_i,\ N_i,\ u_{isv},\ u^k_{isv},\ N_{sv},\ S_i \quad \in \quad \mathbb{Z}^+ \qquad \forall\ i,s,v,k \tag{30}$$

$$z_{stw},\ x_{sv},\ y_i,\ q_i,\ z_i,\ n_{svk},\ U_{isv} \quad \in \quad \{0,1\} \qquad \forall\ i,s,t,v,w,k \tag{31}$$

5 A Mixed Integer Linear Model

In this section, we present a way to linearize the objective, which results in an overall linear model. It is possible to get arbitrarily good approximations, both optimistic and pessimistic, but at the cost of more variables in the model. The reason and justification for doing this is to be able to find optimal solutions for small and mid-size problems, needed to evaluate and validate the meta-heuristic approaches proposed in Section 7.

5.1 Linearization of the objective

In order to linearize the objective, we dissect it into smaller components and approximate them separately. Recall the nasty structure of the objective

$$\max \sum_{s\in S} \left[\sum_{t\in T} \sum_{w\in W_{st}} p^{\text{kill}}_{stw}(\mathbf{z}) \cdot z_{stw} \right] \cdot r_s$$

where $p^{\text{kill}}_{stw}(\mathbf{z})$ is found by combining (1) and (3).

$$p^{\text{kill}}_{stw}(\mathbf{z}) \;=\; 1 - \prod_{j=1}^{V_t} \left[1 - p_{st} \prod_{\substack{i\in\bar{S} \\ i\neq s}} \prod_{k=1}^{K} \left(p^k_{isw_j} \right)^{u^k_{isw_j}} \right]^{m_t}$$

We note that if a certain z_{stw} is zero then obviously $p^{\text{kill}}_{stw}(\mathbf{z})$ should also be zero, independent of what happens to other units. By introducing a new variable P_{stw} and two extra constraints, we present an equivalent formulation

$$\begin{aligned} \max \sum_{s\in S} & \left[\sum_{t\in T} \sum_{w\in W_{st}} P_{stw} \right] \cdot r_s \\ s.t. \quad P_{stw} &\leq z_{stw} \\ P_{stw} &\leq p^{\text{kill}}_{stw}(\mathbf{z}) \end{aligned}$$

This is valid since P_{stw} represents a probability and hence limited to values between zero and one. The objective is now linear, but $p^{\text{kill}}_{stw}(\mathbf{z})$ is still not, now part of the new constraint instead.

To linearize an expression like $p^{\text{kill}}_{stw}(\mathbf{z})$, which includes products, the key observation is that one needs to use logarithms in order to transform them into sums (i.e. linear).

Here follows a mathematically equivalent expression of $p^{\text{kill}}_{stw}(\mathbf{z})$. First we define the following auxiliary variables:

$$\begin{aligned}
Y_{stw} &= \log\left(\prod_{j=1}^{V_t}\left[1 - p_{st}\prod_{\substack{i\in\bar{\mathcal{S}}\\ i\neq s}}\prod_{k=1}^{K}\left(p^k_{isw_j}\right)^{u^k_{isw_j}}\right]^{m_t}\right)\\
&= m_t \cdot \sum_{j=1}^{V_t}\log\left(1 - p_{st}\prod_{\substack{i\in\bar{\mathcal{S}}\\ i\neq s}}\prod_{k=1}^{K}\left(p^k_{isw_j}\right)^{u^k_{isw_j}}\right)
\end{aligned}$$

Each angle w_j corresponds to an angle $v \in \mathcal{V}$, and we define:

$$\begin{aligned}
X_{stv} &= \log\left(p_{st}\prod_{\substack{i\in\bar{\mathcal{S}}\\ i\neq s}}\prod_{k=1}^{K}\left(p^k_{isv}\right)^{u^k_{isv}}\right)\\
&= \underbrace{\log(p_{st})}_{\text{LP}_{st}} + \sum_{\substack{i\in\bar{\mathcal{S}}\\ i\neq s}}\sum_{k=1}^{K} u^k_{isv}\cdot\underbrace{\log(p^k_{isv})}_{\text{LP}^k_{isv}}
\end{aligned}$$

We note that since p_{st} and p^k_{isv} are parameters to the model, hence known in advance, we can save the logarithm of these values in the new LP_{st} and LP^k_{isv} parameters. The expression for $p^{\text{kill}}_{stw}(\mathbf{z})$ can now be written as:

$$\begin{aligned}
p^{\text{kill}}_{stw}(\mathbf{z}) &= 1 - e^{Y_{stw}} && (I)\\
Y_{stw} &= m_t \cdot \sum_{j=1}^{V_t}\overline{Y}_{stw_j}\\
\overline{Y}_{stv} &= \log(1 - e^{X_{stv}}) && (II)\\
X_{stv} &= LP_{st} + \sum_{\substack{i\in\bar{\mathcal{S}}\\ i\neq s}}\sum_{k=1}^{K} LP^k_{isv}\cdot u^k_{isv}
\end{aligned}$$

Apart from equations (I) and (II), who define non-linear functions, the expression for $p^{\text{kill}}_{stw}(\mathbf{z})$ consists of linear relations. To get an overall linear expression, we need to approximate (I) and (II) somehow, using only linear relations. Lets analyze these equations in detail.

5.2 Function (I)

Equation (I) defines a relationship between $p_{stw}^{\text{kill}}(\mathbf{z})$ and the new variable Y_{stw}. It is straightforward to show that all feasible values of Y_{stw} must be non-positive, hence we are interested in an approximation of the concave but non-linear function $p(y) = 1 - e^y,\ y \leq 0$.

The idea is to approximate $p(y)$ using a piecewise linear function, composed of L_Y pieces. With an increasing number of linear pieces, this approximation can be made arbitrarily good. There exist standard methods for modeling piecewise linear approximations, and we will utilize SOS-constraints.

As seen in Figure 9, function $p(y)$ is concave with function values between 0 and 1 for non-positive arguments. Using the fact that our objective is to maximize the probability of success, and that $p(y)$ is concave, it is sufficient to use so called SOS1-constraints. Even if not forced, the approximation will be tight. The approximation in Figure 9 is an outer approximation, hence giving optimistic estimates of the function.

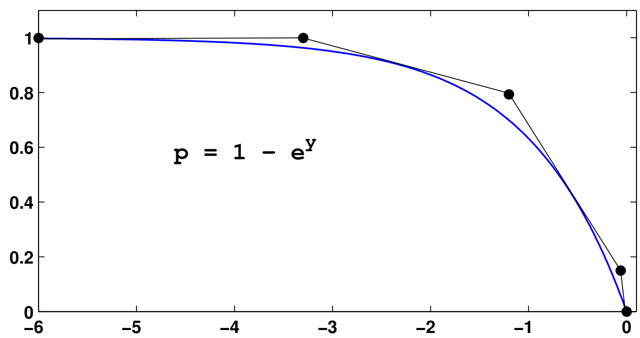


Figure 9: The concave function $p(y) = 1 - e^y$. The piecewise linear approximation is seen in black, based on the sample points represented by the black circles. This coarse approximation uses only 4 line segments. Using more segments, it is possible to get arbitrarily good approximations.

Let $\tilde{y}$ and $\tilde{p}$ denote the sample points and function values respectively. The new variables $0 \leq \lambda_{stwl} \leq 1$ are defined by the first equation, since Y_{stw} is the given argument, and used to find the linear approximation of $p_{stw}^{\text{kill}}(\mathbf{z})$.

$$Y_{stw} = \tilde{y}_0 + \sum_{l=1}^{L_Y} (\tilde{y}_l - \tilde{y}_{l-1}) \cdot \lambda_{stwl} = \bar{y}_0 + \sum_{l=1}^{L_Y} \bar{y}_l \cdot \lambda_{stwl} \tag{4}$$

$$P_{stw} = \tilde{p}_0 + \sum_{l=1}^{L_Y} (\tilde{p}_l - \tilde{p}_{l-1}) \cdot \lambda_{stwl} = \bar{p}_0 + \sum_{l=1}^{L_Y} \bar{p}_l \cdot \lambda_{stwl} \tag{5}$$

Note that the same constants $\bar{y}_l$ and $\bar{p}_l,\ l = 0, 1, \ldots, L_Y$ can be used for all (s, t, w) combinations since it is the same function that is approximated, only evaluated with different arguments.

How to choose the points $\tilde{y}_l$ where the function is sampled is itself a difficult optimization problem, where the objective is (for example) to minimize the maximum error. But once done, we only need the points and corresponding function values in order to do this approximation.

5.3 Function (II)

Equation (II) defines a relationship between variable $\overline{Y}_{stv}$ and the argument X_{stv}, which is an auxiliary variable. It is straightforward to show that for a feasible solutions, variable X_{stv} will be non-positive. Hence, as seen in Figure 10, we are interested in an approximation of the concave function $y(x) = \log(1 - e^x),\ x < 0$.

The objective is to maximize the probability of success, which depends on the auxiliary variable Y_{stw}. We see from Figure 9 that a smaller value of Y_{stw} is preferable, hence we want to minimize the concave function $y(x)$ which is not a convex problem.

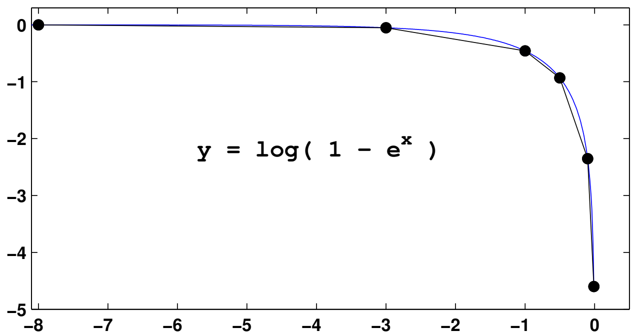


Figure 10: The concave function $y(x) = \log(1-e^x)$. The piecewise linear approximation is seen in black, based on the sample points represented by the black circles. This is an inner approximation, resulting in slightly pessimistic function values, i.e. more negative than they should.

Once again, we wish to approximate $y(x)$ using a piecewise linear function (L_X pieces). This time, we need to use so called SOS2-constraints and add some extra constraints in order to force the approximation to be tight.

$$X_{stv} = \tilde{x}_0 + \sum_{l=1}^{L_X} (\tilde{x}_l - \tilde{x}_{l-1}) \cdot \alpha_{stvl} = \hat{x}_0 + \sum_{l=1}^{L_X} \hat{x}_l \cdot \alpha_{stvl} \tag{6}$$

$$\overline{Y}_{stv} = \tilde{y}_0 + \sum_{l=1}^{L_X} (\tilde{y}_l - \tilde{y}_{l-1}) \cdot \alpha_{stvl} = \hat{y}_0 + \sum_{l=1}^{L_X} \hat{y}_l \cdot \alpha_{stvl} \tag{7}$$

$$\begin{aligned} \alpha_{stvl} &\leq \pi_{stvl} & l \in \mathcal{L}_X &= \{1, \ldots, L_X\} \\ \pi_{stv(l+1)} &\leq \alpha_{stvl} & l \in \mathcal{L}_X^{-1} &= \{1, \ldots, L_X - 1\} \end{aligned}$$

Like before, $\hat{x}$ and $\hat{y}$ denote sample points and function values of $y(x)$. The new variables $0 \leq \alpha_{stvl} \leq 1$ and $\pi_{stvl} \in \{0, 1\}$ are defined by the first equation, since X_{stv} is the given argument, and then used to find the linear approximation of $\overline{Y}_{stv}$. Again, the same constants $\hat{x}_l$ and $\hat{y}_l$, $l = 0, 1, \ldots, L_X$ can be used for all (s, t, v) combinations.

5.4 Optimistic and pessimistic estimates

It is possible to construct both an optimistic estimate and a pessimistic estimate of the objective function, depending on the combination of inner and outer approximations of the functions (I) and (II). We use the following terminology:

P	=	Function (I)	I	=	Inner approximation
Y	=	Function (II)	O	=	Outer approximation

The following combinations yield valid estimates:

$$\begin{aligned} P_O + Y_I &\implies \text{Optimistic estimate} \\ P_I + Y_O &\implies \text{Pessimistic estimate} \end{aligned}$$

All these approximations can be done arbitrarily accurate, at the cost of more line segments, i.e. more variables in the model.

5.5 A MILP model

With the general model presented in Section 4 in mind, where the only non-linear part is the objective, we are now ready to present a mixed integer linear model for the problem.

Using the approximation steps described in the previous section, together with the fact that all constraints are linear, the result is a linear model which approximates the real problem arbitrarily well. Also, using inner and outer approximations in the suggested combinations, the linear approximation can be either optimistic or pessimistic.

Notation, such as parameter names, are kept from Section 4, we just extend the model with the new auxiliary parameters and variables introduced for the linearization process. These constraints are found in the new Group V.

The full MILP model can be found on page 58.

5.5.1 Objective function

The new objective includes the auxiliary variables P_{stw}, the approximated contribution of each attack.

$$\max \sum_{s}\sum_{t}\sum_{w\in W_{st}} r_s \cdot P_{stw} \qquad [MILP]$$

The objective does not involve variable z_{stw} anymore, which is instead taken care of by constraints (23) and (24) in the new Group V. This is a valid reformulation since we want to maximize P_{stw}.

5.5.2 Group V

Constraints $(1)-(22)$ are the same as before. The NLIP model is augmented with Group V, where the approximation of the non-linear objective is taken care of in constraints $(24)-(27)$.

V

$$
\begin{array}{lcllr}
z_{stw} & \geq & P_{stw} & \forall\, s,t,w & (23)\\
\sum_{l=1}^{L_Y} \bar{p}_l \cdot \lambda_{stwl} + \bar{p}_0 & \geq & P_{stw} & \forall\, s,t,w & (24a)\\
\sum_{l=1}^{L_Y} \bar{y}_l \cdot \lambda_{stwl} + \bar{y}_0 & = & Y_{stw} & \forall\, s,t,w & (24b)\\
\sum_{j=1}^{V_t} m_t \cdot \overline{Y}_{stw_j} & = & Y_{stw} & \forall\, s,t,w & (25)\\
\sum_{l=1}^{L_X} \hat{y}_l \cdot \alpha_{stvl} + \hat{y}_0 & = & \overline{Y}_{stv} & \forall\, s,t,v & (26a)\\
\sum_{l=1}^{L_X} \hat{x}_l \cdot \alpha_{stvl} + \hat{x}_0 & = & X_{stv} & \forall\, s,t,v & (26b)\\
\alpha_{stvl} & \leq & \pi_{stvl} & \forall\, s,t,v,l \in \mathcal{L}_X & (26c)\\
\pi_{stv(l+1)} & \leq & \alpha_{stvl} & \forall\, s,t,v,l \in \mathcal{L}_X^{-1} & (26d)\\
\sum_{\substack{i\in\bar{\mathcal{S}}\\ i\neq s}}\sum_{k=1}^{K} \mathrm{LP}_{isv}^k \cdot u_{isv}^k + \mathrm{LP}_{st} & = & X_{stv} & \forall\, s,t,v & (27)
\end{array}
$$

Constraint (23) makes sure we get no contribution to the objective as soon as the corresponding z_{stw} is zero. Otherwise, variable P_{stw} is limited above by the linear approximation.

Finally, constraints (32) − (34) are added to Group X, which defines the variables needed for the linearization.

$$
\begin{aligned}
P_{stw},\ \lambda_{stwl},\ \alpha_{stvl} &\in [0,1] && \forall\ s,t,v,w,l && (32)\\
Y_{stw},\ \overline{Y}_{stv},\ X_{stv} &\leq 0 && \forall\ s,t,v,w && (33)\\
\pi_{stvl} &\in \{0,1\} && \forall\ s,t,v,l \in \mathcal{L}_X && (34)
\end{aligned}
$$

5.5.3 Group VI, strengthening the model

In order for computer software to solve the mathematical models efficiently, one often need to add redundant constraints to strengthen the model. This means that although the extra constraints are redundant, when considering the LP relaxation of the MILP problem, these constraints will push solutions towards integer solutions.

Our first observation is that as soon as any z_{stw} is zero, the corresponding probability P_{stw} will also become zero. In this case, all the extra equations that linearizes the objective are unnecessary, and we wish to speed up this process.

In our linearization of the function $y(x) = \log(1 - e^x)$, independent of the number of line segments we wish to use, we add the extreme sample point $(x = -10, y = 0)$. Although this is not true strictly mathematical, $y(-10)$ is in practice equal to zero.

We do the following changes and additions to our model:

$$
\sum_{\substack{i\in\bar{\mathcal{S}}\\ i\neq s}} \sum_{k=1}^{K} \mathrm{LP}_{isv}^{k} \cdot u_{isv}^{k} + \mathrm{LP}_{st} \geq X_{stv} \qquad \forall\ s,t,v \qquad (27)
$$

VI

$$
\begin{aligned}
(1 - z_{stw}) &\leq \lambda_{stwl} && \forall\ s,t,w,l \in \mathcal{L}_Y && (28)\\
-10 \cdot (1 - z_{stw}) &\geq X_{stw_j} && \forall\ s,t,w,j \in V_t && (29)
\end{aligned}
$$

Constraint (27) is changed from an equality into an inequality. This is valid since we wish to maximize the objective, and the larger X_{stw_j} is the greater becomes the probability P_{stw}. Hence, the objective makes sure that our X variables attains the correct value. This change is necessary in order to add constraints (28) and (29), which forces all $\lambda_{stwl} = 1$ and $X_{stw_j} = -10$ as soon as z_{stw} is zero.

6 Model validation

To validate the MILP model and test its efficiency, a small benchmark with 6 cases were set up and solved. The smallest test case include 2 defender units and 2 target units, and the largest test case include 4 defender units and 3 target units. Rewards are set to 1 for defender units and 2 for target units. All test cases are shown in Figures 11–12 on page 32, and one unit in the pictures correspond to 1000 meters.

In the tests, a pessimistic estimation is used, that is an inner approximation for Function I and an outer approximation for Function II. Hence all objective values found by the model are somewhat lower than the true objective values.

To get accurate and sound linear approximations, 15 line segments were used for Function I and 25 line segments for Function II. The reason for using so many segments is due to the nature of probabilities, they are bounded between zero and one, hence in order to get trustworthy solutions one need to use a good approximation.

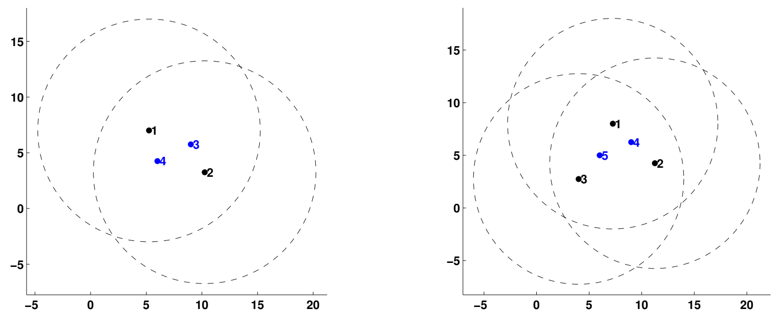


Figure 11: To the left: Test case 1 with 2 defender units and 2 target units.
To the right: Test case 2 with 3 defender units and 2 target units.

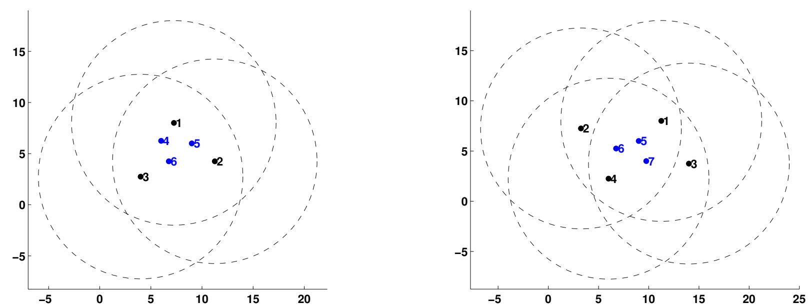


Figure 12: To the left: Test case 3 with 4 defender units and 3 target units.
To the right: Test case 4 with 4 defender units and 3 target units.

The optimal solutions found by the MILP model are evaluated using the true objective, and these objective values are compared with the ones from the MILP model. The tests were performed on a HP DL160 server with two 6-core Intel Xeon CPUs and 72 GB of RAM memory, running Linux, and the MIP solver used were CPLEX/12.2 for 64 bit environment.

The results of the benchmark are presented in Table 1, and one can see that the model is correct and finds optimal solutions. The MILP objective value is correct up to the second digit, compared to the real objective value. The number of constraints (rows), variables (columns) and binary variables (bins) reported are taken from the CPLEX output, which is the size of the reduced model after various presolve steps from both CPLEX and AMPL.

Table 1: Results for the evaluation of the MILP model. Number of constraints (rows) and variables (cols), as well as number of binary variables, are stated together with solution time and objective value. The units column state number of defender units and target units respectively.

Case	units	rows	cols	bins	objective	real obj.	time
1	2/2	13145	8344	2579	5.60474	5.6249	28 min
2	3/2	23571	12031	3928	6.55110	6.5854	11 h
3	3/3	32629	15024	5014	8.83264	8.8454	19 h
4	4/3	48025	18541	6240	-	-	-

The size of the model do increase drastically for each additional unit and the time required for solving the model grows drastically, hence this approach is not tractable for cases involving even as few as 5–6 units. For the largest test instance, including 7 units, CPLEX ran out of memory (72 GB) after 20 hours. A good feasible solution was available, but the duality gap were still as big as 2%.

Further tests were done, using less line segments for Function I and Function II, 8 and 17 respectively, at the loss of good approximation of the objective function. The number of constraints, variables and binary variables did decrease, but without any major effect on solution times.

It is possible for AMPL to represent these linear approximations in such a way that CPLEX will treat them explicitly as SOS constraints. It seems reasonable that it would increase the efficiency of CPLEX, as many auxiliary variables can be skipped, but the result turned out to be the opposite. The upper bound became worse, making it even tougher to close the duality gap, hence prolonging the solution times. The explanation is probably due to the extra constraints (28) and (29), found in Group VI, as they disappear together with the auxiliary variables. These constraint strengthen the model, and without them the LP relaxation is much weaker, hence affecting the upper bound.

7 Meta-Heuristics

A problem like this, with only a few natural constraints (one attack per unit and shared resources) and a nasty non-linear objective function (non-costly though), is well suited for meta-heuristics. Throughout this section, we base our work on the following assumptions:

1. The number of available resources is limited, i.e. it is not possible to use the maximum number of resources against all unit.
2. It is optimal to use all available resources.

The first assumption is reasonable, otherwise the problem is reduced to choose between tactics 3 and 5, either assigning all resources on the same path or split them on 3 different paths. One would still need to figure out the optimal combination of tactics and angle of attack for each unit, so it would still be a non-trivial problem though.

The second assumption is very reasonable and simplifies the work of defining neighbourhoods and setting up heuristic schemes.

7.1 Neighbourhoods

Regardless of the meta-heuristic one wishes to use, we need to define a neighbourhood for a solution $\mathbf{z}$. Under the assumptions stated above, all we need is to work with feasible attack plans $\mathbf{z}$ that uses all available resources. Hence we define five neighbourhoods of an attack plan $\mathbf{z}$, denoted $\mathcal{N}_k(\mathbf{z})$, in the following ways:

$\mathcal{N}_1$. The angle of attack w is changed for one unit s and tactic t in the attack plan, that is $z_{stw} \rightarrow z_{st\bar{w}}$.

$\mathcal{N}_2$. The tactic against one unit is changed by switching between one angle and multiple angles, that is $z_{stw} \rightarrow z_{s\bar{t}\tilde{w}}$. If necessary, the reference angle w is adjusted. For example, instead of two resources attacking from the same angle, they attack from different angles. Notice that the number of resources involved in the attack is still the same though.

$\mathcal{N}_3$. Pick two units at random and switch their tactics and angle of attack. For example, variables $z_{s_1t_1w_1}$ and $z_{s_2t_2w_2}$ become $z_{s_1t_2w_2}$ and $z_{s_2t_1w_1}$ instead.

$\mathcal{N}_4$. Pick two units at random and exchange their angle of attack. For example, variables $z_{s_1t_1w_1}$ and $z_{s_2t_2w_2}$ become $z_{s_1t_1w_2}$ and $z_{s_2t_2w_1}$ instead.

$\mathcal{N}_5$. Pick two units at random, which does not use the same number of resources, and change to new tactics which increase/decrease the number of resources used respectively. For example, one unit is changed to be attacked by two resources instead of one, while another unit is attacked by two resources instead of three.

By continuously changing between these neighbourhoods, all feasible solutions can be reached. Notice that neighbourhood $\mathcal{N}_5$ is crucial, since without it the number of resources allocated against each unit would remain fixed to that of the initial solution throughout the search.

7.2 Simulated Annealing

One popular meta-heuristic, which is easy to implement, is Simulated Annealing (SA). The basic idea, which makes it a meta-heuristic and not a local search method, is to accept solutions which are non-improving in order to escape local optima. This is done by chance, where the probability to accept the non-improving value is connected to the difference in objective values between the new solution and the current one.

Also, in order to assure a local optimum, the probability of accepting worse solutions decrease over time. This is done by parameter T, the so called temperature, which decreases as the iterations goes by. A Simulated Annealing approach is successfully used for a Weapon Target Allocation Problem in [1], further motivating this section.

7.2.1 SA algorithm

In Algorithm 1 we describe the implemented SA heuristic to be used later on in the Benchmark. Iteration counter t represents the outer iteration, and at the end of this loop the temperature T is updated (decreased). Inside each outer iteration loop, we cycle through the different neighbourhoods, defined in the user-given NBH sequence.

For each neighbourhood, we generate 100 new solutions who are evaluated and possibly saved as the current solution. The use of multiple neighbourhoods provide diversity to the search, and we also keep track of the overall best found solution.

7.3 Post Processing

For a problem like this, it is natural to consider some sort of post processing. Given a solution $\mathbf{z}$, found by some heuristic scheme, one should definitely try to improve it locally, i.e. perform a local search.

For this problem, where a solution $\mathbf{z}$ states which tactic t and angle w to be used for each unit s, it is straightforward to test all feasible angles $w \in \mathcal{W}_{st}$ for the assigned tactic t, one unit at a time, and save the best improvement (if any). Then, if there were some improvement, one could repeat the same process again (since one unit is now attacked from a different angle, and further improvements might be possible) until the process converges.

Algorithm 1 Simulated Annealing (SA)

Define Neighbourhood sequence: $NBH := \{5, 2, 1, 3, 4, 1, 5, 2, 1\}$.
Define Initial Temperature $T = 0.9$. Cooling factor $COOL = 0.7$.
Generate Initial solution $\mathbf{z}$ (randomly) with objective value f.
Initialize Best found solution $\mathbf{z}_b = \mathbf{z}$ with objective value $f_b = f$.

for $t = 1 \ldots 8$ **do**
  for $k = 1 \ldots 9$ **do**
    for $n = 1 \ldots 100$ **do**
      Get new solution: $\mathbf{z}_n = \mathcal{N}_k(\mathbf{z})$ with objective value f_n.
      If $f_n > f_b$, update best found solution.
      Calculate diff: $\Delta = f_n - f$. Generate random number: $r \in [0, 1]$.
      if $r < \exp(\Delta/T)$ **then**
        $f = f_n$
        $\mathbf{z} = \mathbf{z}_n$
      end if
    end for
  end for
  Update temperature: $T = T \cdot COOL$.
end for

At the same time as one tries all angles, one could also test to switch between all tactics using the same number of resources in total, hence conserving the overall usage of resources (assumed to be at its upper limit). We present the local search procedure in Algorithm 2.

Algorithm 2 Post Processing (PP)

Given Starting solution $\mathbf{z}_0$ with objective value f_0.
Initialize Best found solution $\mathbf{z}_b = \mathbf{z}_0$ with objective value $f_b = f_0$.

for $s \in \mathcal{S}$ **do**
  Get $\tilde{t} =$ Tactic currently used against unit s.
  Find $T_s := \{t \in \mathcal{T} : n_t = n_{\tilde{t}}\}$.
  for $t \in T_s$ **do**
    for $w \in \mathcal{W}_{st}$ **do**
      Set $\mathbf{z}_n = \mathbf{z}_0$, but with tactic t and angle w used against unit s.
      Evaluate $\mathbf{z}_n$. If $f_n > f_b$, update best found solution ($z_b = z_n$).
    end for
  end for
end for

if $f_b > f_0$ **then** {Check if any improvement has been found}
  Set $z_0 = z_b$ and call the PP algorithm again.
end if

This has proven to be a very powerful tool, often providing good solutions for almost any starting solution with an allocation of resources close to the optimal one. This is crucial since the number of resources are never shifted throughout the post processing.

7.4 Augment Solution

Another intuitive strategy is to iteratively augment a previous solution, adding one extra resource in each iteration. It seems plausible that the optimal solution using, lets say, 8 resources is close to the optimal solution for 7 resources.

Provided a solution using k resources, denoted $\mathbf{z}_k$, we seek a solution $\mathbf{z}_{k+1}$. This is done by considering one unit at a time in the solution $\mathbf{z}_k$, adding one resource if not $K = 3$ resources are already in use, and then performing a local search. The best such augmentation is saved and returned as the new solution $\mathbf{z}_{k+1}$. The Augmented Solution procedure is described in Algorithm 3.

As a bonus, this approach will generate a Pareto-like solution, stating the expected outcome of the attack for a given number of resources. Such a solution is very useful as a decision support for choosing the number of resources to use for an attack. As will be seen in the forthcoming results, the gain in expected outcome when augmenting an additional resource decreases as a function of the number of resources already in use.

Algorithm 3 Augmented Solution (AS)

Given Starting solution $\mathbf{z}_k$, using k resources, with objective value f_k.
Initialize Best found solution $\mathbf{z}_b = \mathbf{z}_k$ with objective value $f_b = f_k$.

```
for s ∈ S do
    Find M_s, the number of resources used against unit s in z_k.
    if M_s < K then
        Let z_{k+1} = z_k but add one resource against unit s.
        Perform a Post Processing step. z_PP = PP(z_{k+1}).
        If f_PP > f_b, update best found solution (z_b = z_PP).
    end if
end for
```

This procedure can be applied to any feasible solution $\mathbf{z}_k$ using k resources and will produce a locally optimal solution $\mathbf{z}_{k+1}$ using $k+1$ resources. It is thus possible to find good solutions in the following manner. Initially find a really good solution $\mathbf{z}_k$, which is efficient for very small k, and then use Algorithm 3 to find $\mathbf{z}_{k+1}$. Repeat the process to find a solution $\mathbf{z}_{k+2}$, and so on until a requested upper limit.

The algorithm performs $|S|$ post processing searches, one for each unit. The time needed will therefore increase with respect to the number of enemy units. Even so, Algorithm 3 should spend less time than the Simulated Annealing.

8 Benchmark

In order to test the SA approach, we define a set of problems to be part of a benchmark. In these tests we like to cover some different characteristics like the number of defending units and targets units, as well as some different reward settings for these two kinds of units. A coarse angle discretization of 12 angles, as in Figure 5, is used for all problems in the benchmark.

It is also interesting to perform a sensitivity analysis with respect to the number of available resources. Hence all problem instances are solved having from 1 up to 30 resources available. In all, a total of 1080 problem instances are solved.

We also test some other methods to compare with the Simulated Annealing heuristics. All these features of the benchmark are described in more detail in the upcoming sections.

8.1 Scenarios and Cases

A total of three different scenarios are considered, where the number and positions of the defender units differ. Also, for each of these scenarios, four different versions of positions and number of target units are solved. In all cases, one unit step in the pictures corresponds to 1 km.

8.1.1 Scenario 1

The first scenario includes two defender units positioned 10 km apart, each with a defensive radius of 10 km. In the different cases, we consider 5, 7, 8 and 14 target units respectively, positioned in a fashion similar to Figure 13. The distance between the target units is 300-500 meters.

8.1.2 Scenario 2

The second scenario includes five defender units positioned in an x-shaped formation as in Figure 14. In the different cases, we consider 5, 8, 12 and 16 target units respectively. The distance between the target units is 300-500 meters.

8.1.3 Scenario 3

The third scenario includes four defender units positioned as in Figure 15. In the different cases, we consider 5, 8, 12 and 16 target units respectively, and like before the distance between them is 300-500 meters.

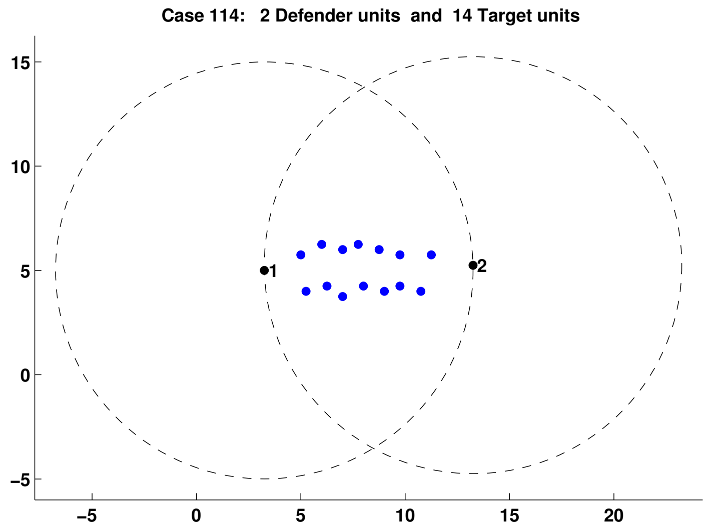


Figure 13: Test case 114. 2 defender units and 14 target units.

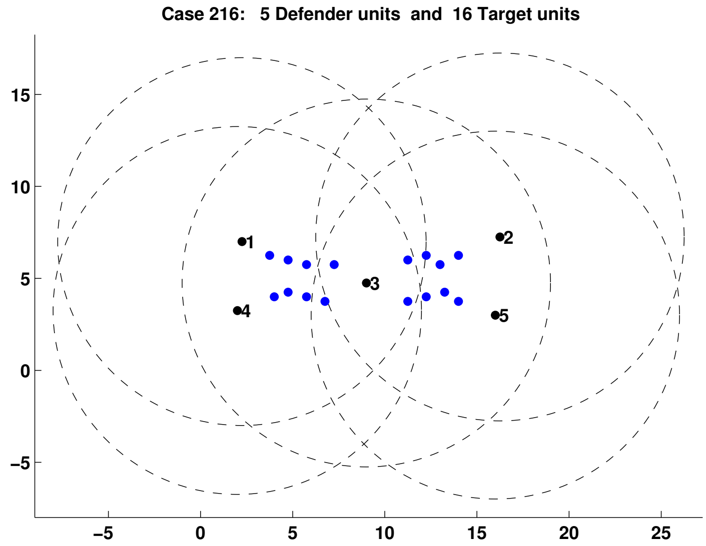


Figure 14: Test case 216. 5 defender units and 16 target units.

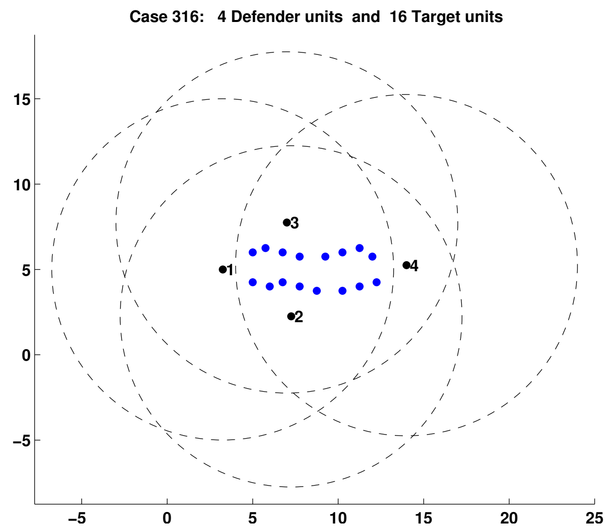


Figure 15: Test case 316. 4 defender units and 16 target units.

8.2 Rewards

To make sure our model responds to the reward of each unit, we define three different reward settings. As seen in Table 2, the first setting only premiere target units. It might still be optimal to attack the defender units though, in order to reduce their defensive capabilities and thus increase the overall profit.

Table 2: The different reward settings.

	Reward	
Setting	Defender units	Target units
1	0	1
2	1	2
3	1	5

In the second setting, defender units are now also considered valuable but only second to the target units. The third setting is just a different version where the two groups of units are differentiated even more.

8.3 Tests

Here follows a short description of all tests performed in the benchmark.

8.3.1 Optimistic and pessimistic model

The Optimistic and Pessimistic models presented in Sections 2.5 and 2.6 are easily solved using some linear IP solver, in our case CPLEX. These solutions provide an upper and lower bound on the objective value and are found in fractions of a second. In order to improve the lower bound, the pessimistic solution provided by the solver is simply evaluated using the real objective. This simple action improves the bound significantly and is also done instantly.

Moreover, if a local search is performed from the pessimistic solution, an even better (at least as good) solution is found. This comes with the cost of one local search, which might take up to half a minute depending on the size of the instance and depending on how many improvements that are done. The Post Processing (PP) procedure described in Algorithm 2 is used. In all, this is fairly inexpensive and improves the bound even more in most cases.

8.3.2 Random solutions with and without local search

These experiments generate feasible solutions in a random fashion. Each solution is evaluated and the best found solution is returned. In the first test, 5000 solutions are generated and evaluated. This is fairly inexpensive timewise, which can be seen in the results later on.

The second test generates 100 feasible solutions and a local search (Post processing) is performed from each solution. Compared to first test, this is a quite expensive approach. The time used to perform 100 local searches widely exceeds the time for generating and evaluating 5000 feasible solutions, but superior solutions are expected.

8.3.3 Simulated Annealing

The SA heuristic has already been described in detail in Section 7.2 and Algorithm 1. This is a fairly expensive method, but should generate the best quality solutions of all methods.

8.3.4 Augmented Solution

Using the locally improved pessimistic solution for $k = 2$ resources, this test simply applies Algorithm 3 to augment one resource at a time, finally solving the problem with the maximum number of resources. The procedure should generate near-optimal solutions to the cost of a Post processing step for each new resource.

9 Results

In this section, we present some results and conclusions for the different approaches discussed in Section 8. A limited but representative number of graphs are shown.

9.1 Case 105

To analyse the results of the benchmark, we take a closer look at Case 105 where we have 2 defender units and 5 target units, positioned as in Figure 16.

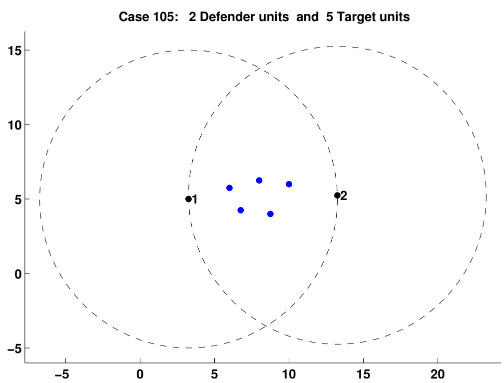


Figure 16: Test case 105. 2 defender units and 5 target units.

In the upcoming sections, we present and analyze the result for the different reward settings.

9.1.1 A typical solution

In Figure 17 we see a graphical representation of the best found solution for Case 105 with reward setting 3 and 14 resources available. Both defender units, numbers 1 and 2, are attacked by tactic 5 which means 3 resources from different directions. Target units 5 and 6 are attacked using tactic 4, were 2 resources attack from opposite directions. Target unit 3 is attacked using tactic 2, i.e. 2 resources from the same direction, indicated by the dashed line. Finally, target units 4 and 7 are attacked by a single resource respectively.

The solutions are not always intuitive at first glance. For example, one of the attack paths toward unit 1 intersects the defensive area of unit 2 for a long distance, and vice versa. Is it not better to attack with all 3 resources from the same angle and avoid the defense of the other defender unit?

The explanation is logical. Consider the resource attacking defender unit 1. By traveling inside the defensive area of defender unit 2, some of the defensive capabilities of unit 2 will be allocated against this resource. As one of three resources taking part of the attack against unit 1, the total expected probability of success will be quite high even though this specific resource face great danger. In this way, the defensive capabilities available for unit 1 to use against other resources are reduced, and the overall objective will gain from it.

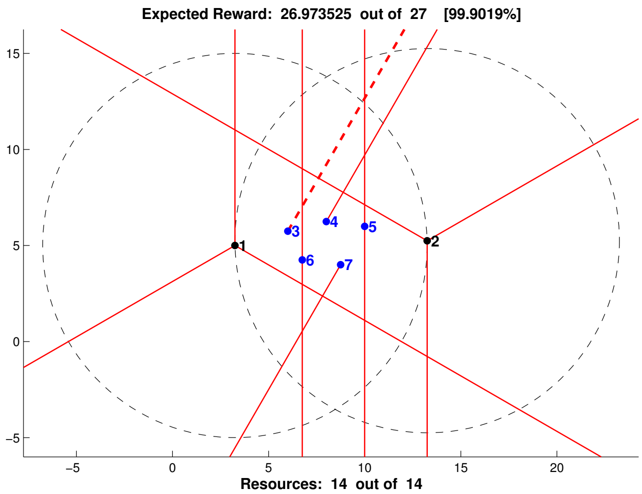


Figure 17: Test case 105. 2 defender units and 5 target units. The solution requires 14 resources, who are used in different ways, according to the tactics used.

Figure 18 shows a graphical representation of the best found solution for the same case but with 17 resources available. The objective is improved somewhat, but not much.

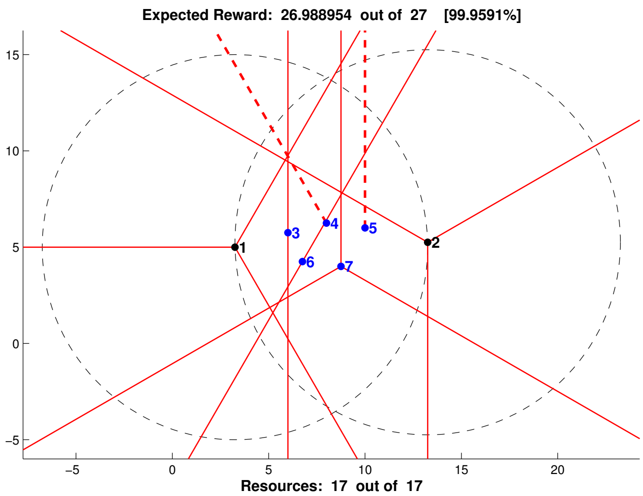


Figure 18: Test case 105. 2 defender units and 5 target units. The solution requires 17 resources, who are used in different ways, according to the tactics used.

9.1.2 Reward setting 1

The use of reward setting 1, i.e. reward 0 for all defender units and reward 1 for all target units, render the result seen in Figure 19. The x-axis represents the number of resources available and the y-axis the corresponding objective values.

The blue lines represent the upper and lower bounds found by CPLEX, where the pessimistic solutions have been evaluated using the real objective function. The green dots represent the pessimistic value given by CPLEX.

The black dots, found in between the blue lines, are the locally improved pessimistic solutions. We can see that for most of the instances, this improvement is substantial. The black line with crosses indicate the best found solutions from the SA heuristic.

The red line with dots show the result of the Augmented Solution approach. The solutions are in general the best ones found for all instances, sometimes coinciding with the Simulated Annealing solutions. The magenta colored lines show the range for the 100 local searches performed, together with a cross indicating the best objective of the 5000 solution generated in the other test.

9.1.3 Reward settings 2 and 3

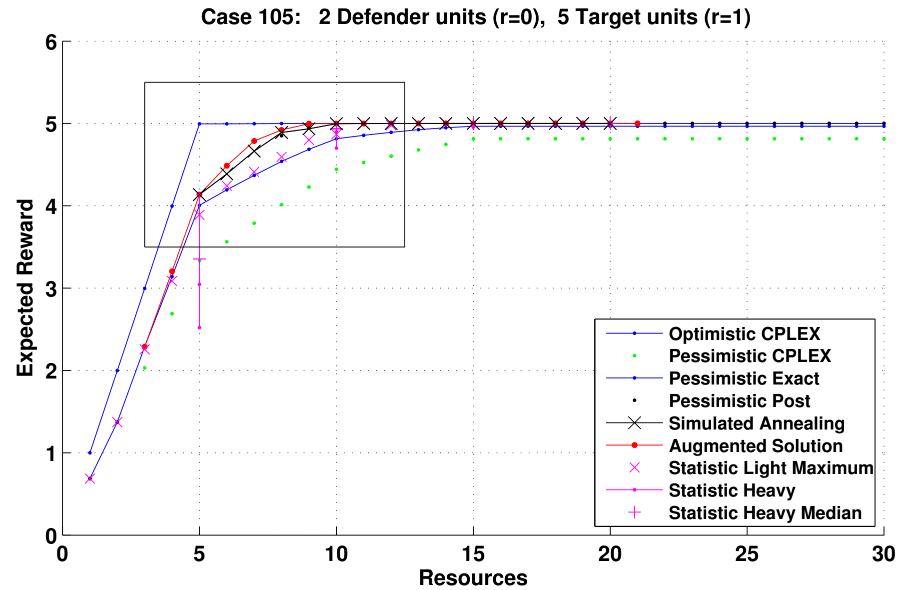
Similar results compared with reward setting 1. The same behaviour can be observed in Figure 20 and Figure 21 respectively. Obviously, the objective value itself differs due to the different reward settings, but the overall trend is the same.

9.1.4 Time Case 105

In Figure 22, we find the solution times in minutes for the different tests. The results are the mean over the three different reward settings, as they were all very similar.

The time needed for CPLEX to find all optimistic and pessimistic solutions are less than a second, even though each problem is solved 30 times. The local search for each pessimistic solution do take some time, but varies from a few seconds as for this case up to at most 30 seconds for big instances where 12-16 units are involved. This is found in Figure 22 as the small red bars, not even visible for some of the instances.

To generate and evaluate 5000 feasible solutions take about 5-15 seconds, seen as the blue bars. Even though the solution time is relatively quick, the results are not very promising as seen in Figures 19-21. For some of the instances (5,10,15 and 20 resources), 100 feasible solutions have been generated and evaluated, followed by a local search. Solution times are



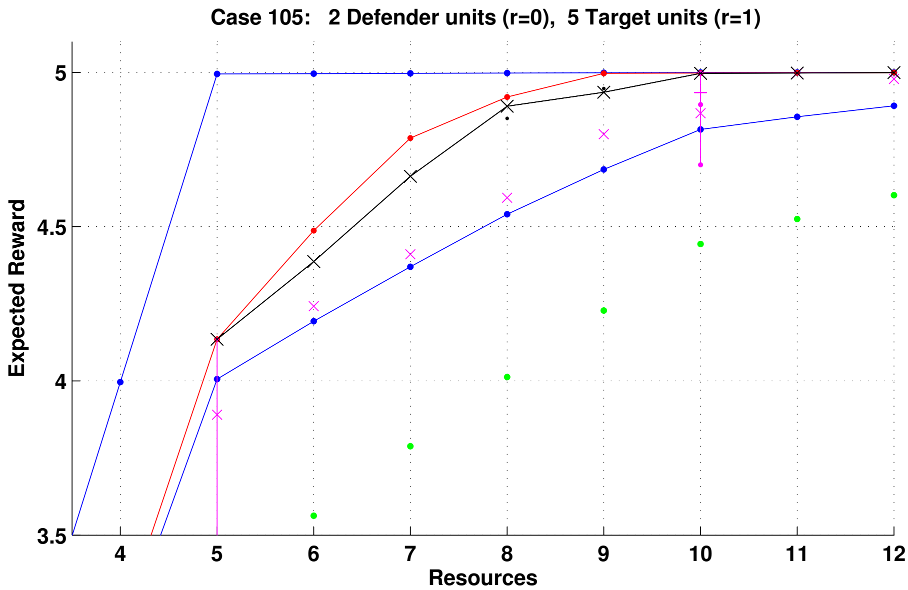


Figure 19: Test case 105. 2 defender units and 5 target units. To get a better view, the lower picture provides a zoom of the black box from the upper picture, the area of interest.

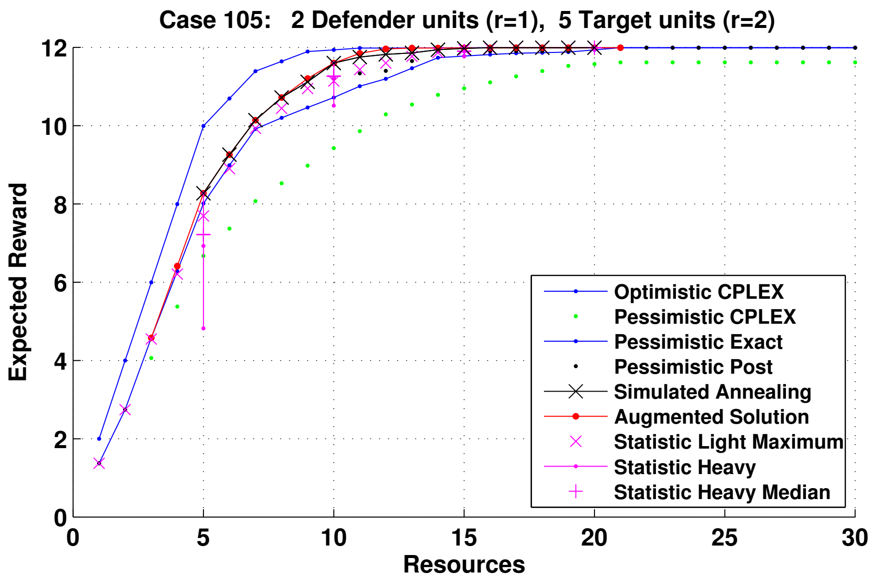


Figure 20: Test case 105. 2 defender units and 5 target units.

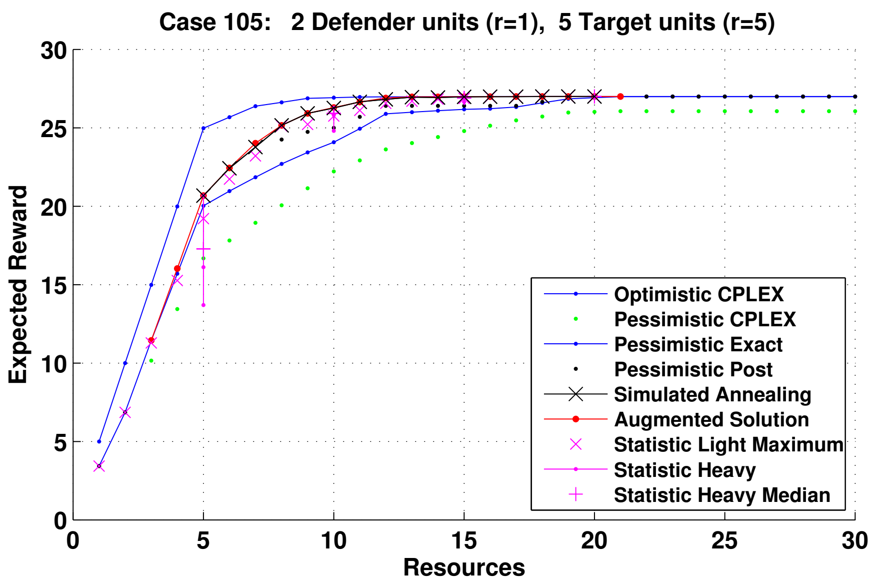


Figure 21: Test case 105. 2 defender units and 5 target units.

Figure 22: Test case 105. 2 defender units and 5 target units.

represented by the yellow bars, and as seen, this approach is quite time consuming.

The pink (magenta) bars represent solution times for the SA heuristic, and are around 1.5 minutes. Although considerably more expensive than the first two approaches, SA provides in general very good solutions. The difference in objective value is sometimes significant compared with the other methods, and the additional time might be well spent. It is also possible to stop the SA heuristic at any time, opening up for the alternative of running it as long as time permits.

Finally, the green bars show the solution time for the Augmented Solution algorithm which provides the best solutions. Since this method builds a solution recursively, the solution times are accumulated over the number of resources. So even though each step is time efficient, the total time to find a solution for many resources, say R, do get expensive, but at the same time *all* solutions using up to R resources are generated.

All implementations are done in MATLAB at the moment. If implemented in a more efficient language, the computation times would most certainly decrease significantly.

9.2 Remarks Case 105

The behaviour is very similar for the different reward settings. The upper and lower bounds found by CPLEX are not extremely tight for 1-15 resources, but improves after the local search. For this small case, involving a total of 7 units, using more than 15 resources is not very interesting. As seen in the graphs, the optimistic and pessimistic bounds are tight.

For larger instances, with 10-20 units involved, the situation is somewhat different. We do suspect that the lower bound is near optimal for up to 10 resources, and that the strength of the upper bound improves with an increasing number of resources. For instances where 10-20 resources are available, none of the bounds seems to be tight. This can be seen in the results for Case 212, found in the Appendix on page 80.

The random generation of 5000 feasible solutions is not very successful, most of the time far worse than the locally improved pessimistic solution. The best found solutions of 100 local searches are on the other hand quite impressive, ranging from comparable with the lower bound up to the overall best found solution, even better than the solution provided by Simulated Annealing.

The Simulated Annealing algorithm performs very well, and provide solutions comparable with the Augmented Solution approach, but requires comparable long time for moderate number of resources. The Augmented Solution approach do find the best found solutions most of the time, only beaten by the SA method on single occasions, but requires even more time than the SA algorithm when considering many resources.

9.3 Summary of Benchmark Results

The analysis of Case 105 is representative for all scenarios and cases. To compare results for the 12 different scenarios, the objective values are normalized with respect to the optimistic value found for each instance. The mean objective for each method is then presented in Figure 23, and also in Table 3 for some specific number of resources.

Table 3: Mean objective values for each method. Best values are boldfaced and second best values are emphasized.

	Resources					
Method	5	10	15	20	25	30
Opt. CPLEX	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Pess. Exact	0.6702	0.7691	0.8399	0.8894	0.9209	0.9463
Pess. Post	*0.6893*	0.8300	0.9023	0.9522	0.9737	0.9834
Sim. Ann.	0.6845	*0.8545*	*0.9369*	*0.9840*	*0.9918*	*0.9940*
Aug. Sol.	**0.6999**	**0.8599**	**0.9382**	**0.9852**	**0.9941**	**0.9988**
Stat. Light	0.6185	0.7786	0.8771	0.9472	0.9506	0.9696
Stat. Heavy	0.6793	0.8514	0.9338	0.9824	0.9882	0.9926

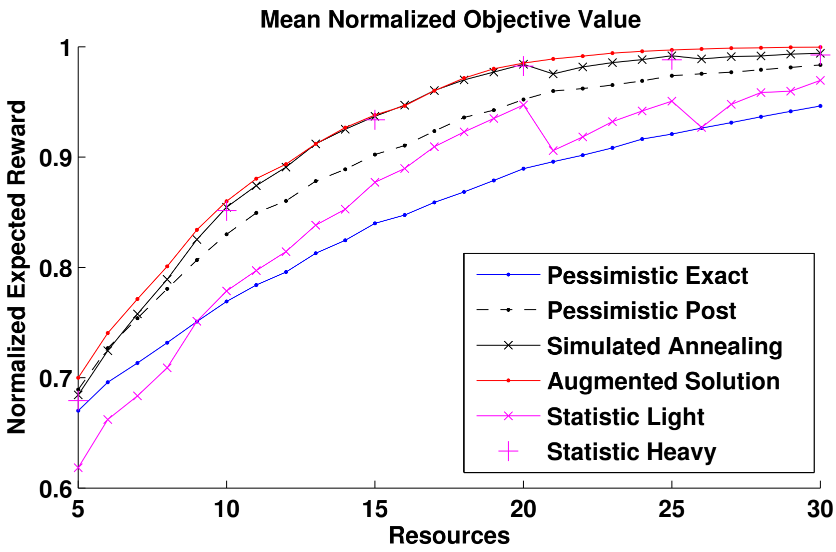


Figure 23: Mean of normalized objective values for all methods.

The Augmented Solution approach is the most stable of all solution methods, providing top quality solutions for all different scenarios and reward settings. The Simulated Annealing method is also very successful, with a clear second place after the Augmented Solution approach.

The random generation of a large number of feasible solutions is not successful at all, even beaten by the locally improved pessimistic solutions when using few resources. The 100 local search approach, Heavy Statistic, do provide excellent results but is very time consuming. This approach is 100% parallelizable though, so if computer power is no issue Heavy Statistic is a good and straightforward alternative.

Because of the long calculation times required for a single run of the SA method, it is only competitive with the Augmented Solution approach when seeking a single solution for *one* specific number of resources, which also needs to be quite large. Otherwise, the Augmented Solution approach provides both better calculation times and solution quality, with the extra feature of providing a whole set of solutions. In all, the Augmented Solution is the clear winner.

For a comprehensive list of results, we refer to Section 12 on page 61, found in the Appendix.

10 Conclusions and Future Work

In this paper, we have introduced and defined a mission planning problem. The mathematical model of the problem is presented, and the complex objective function is analyzed in detail. This results in a generic model, later used to derive optimistic and pessimistic models. Such models are an important tool since they provide upper and lower bounds on the objective value, hence limiting the uncertainty of the quality of solutions.

However, in order to solve real life problem sizes, it is necessary to use heuristic methods. We have proposed a Simulated Annealing heuristic and an Augmented Solution method to solve this difficult problem. The methods were tested on a benchmark of problems, along with some other methods, and the results are very promising.

Solution times are quite slow for the SA algorithm, but a more professional implementation of the heuristic will surely improve them substantially. The Augmented Solution method has good solution times as it is, but they can surely be improved as well. All methods are generic and can handle different objective functions. It is actually sufficient to provide a black-box function to call whenever the objective needs to be evaluated. Hence, if the assumptions in Section 3 are inadequate, or needs to be modified in any way, the given framework will still be applicable.

Future Work

Consequently, this paper has focused on the development of a planning system only considering target scene parameters such as unit location and defense system description, and how they react upon attack. Resource performance is certainly included in the analysis but just in the sense of a static set up of effect-on-target as a function of tactics, and the ability to survive in a surface-to-air defense system environment.

This approach is carefully chosen to comply with future command and control doctrines which promote a separation of effect planning and resource allocation planning.

To extend the mission scope we can include planning aspects of the platform. Route planning can be conducted in a flexible way with its own objectives to conclude the overall mission success. Obvious aspects are minimizing radar cross section exposure during route phase, and minimize time to target, i.e. explore hiding possibilities or by clever surveillance tactics during the cruise phase.

Also, since there is a strong separation, firing platforms must not be given in advance, instead maximizing the effect of the target area can be the driver to find the best platforms from a larger set.

Based on this fact, future work could address at least two obvious scenarios:

- The first case is when the target scene is known and there are a predefined number of platforms where route planning is included in the overall mission.

- A second case consider when several platforms are available. In this case we must allocate good firing units from a set of platforms but also decide firing position and route planning.

An obvious continuation from our work within this paper, is to investigate the coupling between route and effect planning. If this is solved properly a large step is taken to control and comprise vital aspects of ground attack planning.

References

[1] C. Ciobanu, and G. Marin: On Heuristic Optimization. *An. St. Univ. Ovidius Constanta* **9** (2), 17–30 (2001).

[2] B. I Kaminer, and J. Z. Ben-Asher: A Methodology for Estimating and Optimizing Effectiveness of Non-Independent Layered Defense. *Systems Engineering* **13** (2), 119–129 (2010).

[3] O. Kwon, K. Lee, D. Kang, and S. Park: A Branch-and-Price Algorithm for a Targeting Problem. *Naval Research Logistics* **54**, 732–741 (2007).

[4] F. A. Miercort and R. M. Soland: Optimal Allocation of Missiles against Area and Point Defenses. *INFORMS*, 605–617 (1971).

[5] K. A. Yost, and A. R. Washburn: Optimizing Assignment of Air-to-Ground Assets and BDA Sensors. *Military Operations Research* **V5 N2**, 77–91 (2000).

Appendix

The appendix consists of two parts. In the first section, the full NLIP and MILP mathematical models are given, and in the second part a comprehensive collection of benchmark results are presented.

11 Mathematical Models

In this section, we introduce the full NLIP and MILP mathematical models.

It should be stated that if one aims at solving the problem using some meta-heuristic, it suffices to consider constraints (1) and (2), i.e. the generic problem described on page 12. These are the only real constraints of the problem, all the rest are used to model the behaviour of the units with defensive capabilities and are uniquely defined for a given attack plan $\mathbf{z}$.

11.1 The Non-Linear Integer Programming Model

For the general model presented here we assume limiting values for both parameter F_i and parameter G_i. Although, since the parameters M_i, G_i, C_i and F_i are known in advance, it is possible to check whether $M_i \leq G_i$ or $C_i \leq F_i$, or both, for each unit $i \in \bar{\mathcal{S}}$. If so, many constraints are redundant and some variables are unnecessary, and it is possible to reformulate some constraints in order to avoid extra work.

All constraints are linear, it is only the nasty objective which makes the problem extremely non-linear. Constraints (1) and (2), found already in the generic model, make sure we use no more resources than available and that each unit s is attacked at most once. All other constraints are divided into five groups, and a thorough description of each group can be found in Section 4.2.

11.2 The Mixed-Integer Linear Programming Model

Using the linear approximations derived in Section 5, the nasty objective is reformulated and a Mixed Integer Linear Programming (MILP) problem is presented. All constraints from the NILP model is kept, but the objective function is reformulated and additional constraints and variables for the piecewise linear approximations are added.

11.3 Notation

Parameters

given	
R	total resource available, the amount of resources.
r_s	reward (value/price) of unit s.
n_t	# resources used by tactic t.
V_t	# angles used by tactic t.
m_t	# resources/angle used by tactic t.
C_i	number of defensive channels for unit $i \in \bar{\mathcal{S}}$.
F_i	maximum number of defensive channels against a path.
G_i	maximum number of paths that unit i can engage.
ρ_i	radius of defense for unit i.
d_{it}	# defensive channels used by unit i when attacked by tactic t.
p_{st}	probability that a resource survives the defense of unit s when part of tactic t.
L_Y	number of linear pieces in the approximation of Function (I).
L_X	number of linear pieces in the approximation of Function (II).
pre-processed	
K	maximum number of resources/angle, i.e. $\max\{\, m_t,\ t = 1, \ldots \lvert\mathcal{T}\rvert\}$.
M_i	maximum number of engagement paths that passes by unit i.
A_{vtw}	1 if path (s, v) is active when the combination of tactic t and angle w is used against some unit s, where $w \in \mathcal{W}_{st}$.
d_{isv}	distance from unit i to center point of path (s, v) inside ρ_i.
δ_{isv}	1 if $d_{isv} < \rho_i$. Indicates which paths unit i might engage. Used to define set Δ_i.
r_{isv}	ranking of paths (s, v) passing by each unit i, where $\delta_{isv} = 1$. The shorter distance d_{isv}, the closer to the unit and lower ranking.
p_{ik}	probability p_{st}, defined only for units $i \in \bar{\mathcal{S}}$, where $V_t = 1$ and $m_t = k$, for $k = 1, \ldots, K$.
p^k_{isv}	probability of surviving the defense of unit i for a resource on path (s, v) who is part of an attack t where $m_t = k$.

Definitions

engagement path	(s, v)	:=	the line emanating from unit s at angle v.
attack plan	$\mathbf{z}$	:=	a collection of z-variables, one for each unit s, which defines a tactic t and angle w.

Sets

given

$\mathcal{S}$, $\mathcal{T}$, $\mathcal{V}$	set of units $\mathcal{S}$, tactics $\mathcal{T}$ and angle discretization $\mathcal{V}$.
$\bar{\mathcal{S}}$	set of units with defensive capabilities. Subset of $\mathcal{S}$.

pre-processed

$\mathcal{W}_{st}$	set of feasible angles w for tactic t against unit s.
Δ_i	set of paths (s, v) that passes by unit $i \in \bar{\mathcal{S}}$.
$\mathcal{R}_i$	set of triplets $(i, sv, \bar{s}\bar{v})$ where $r_{isv} < r_{i\bar{s}\bar{v}}$.

linearization

$\mathcal{L}_X$	the set $\{1, \ldots, L_X\}$.
$\mathcal{L}_X^{-1}$	the set $\{1, \ldots, L_X - 1\}$.

Variables

integer

B_i	number of active engagement paths (s, v) passing by unit i.
D_i	residual defensive capacity for unit i.
S_i	slack variable for the residual defense quota for unit i.
N_i	help variable, $N_i = \min\{B_i, G_i\}$.
u_{isv}	number of defensive channels that unit i will use against resources on path (s, v).
u_{isv}^k	equal to u_{isv} if $n_{svk} = 1$, zero otherwise.
N_{sv}	number of resources on path (s, v).

binary

z_{stw}	1 if unit s is attacked using tactic t and angle w, where $w \in \mathcal{W}_{st}$.
x_{sv}	1 if any resource travels toward unit s on path (s, v).
n_{svk}	1 if $N_{sv} = k$, zero otherwise.
y_i	1 if $B_i \ \geq \ D_i$ for each unit i.
q_i	1 if $D_i \ \leq \ F_i \cdot \min\{G_i, B_i\}$ for each unit i.
z_i	1 if $B_i \ \geq \ G_i$ for each unit i.
U_{isv}	1 if unit i will use any defensive channel against path (s, v).

linearization

P_{stw}	linearized approximation of $p_{stw}^{\text{kill}}(\mathbf{z})$.
Y_{stw}	input argument to the approximation of Function (I).
$\overline{Y}_{stv}$	function value of the approximation of Function (II).
X_{stv}	input argument to the approximation of Function (II).
λ_{stwl}	weight of each linear piece for Function (I).
α_{stvl}	weight of each linear piece for Function (II).
π_{stvl}	binary variable used to force certain α_{stvl} to zero or one .

The Non-Linear Integer Programming Model

$$\max \quad \sum_{s\in S}\left[\sum_{t\in T}\sum_{w\in\mathcal{W}_{st}} p^{\text{kill}}_{stw}(\mathbf{z})\cdot z_{stw}\right]\cdot r_s \qquad [NLIP]$$

$$\begin{array}{lrcll}
s.t. & \sum_s\sum_t\sum_{w\in\mathcal{W}_{st}} n_t\cdot z_{stw} & \leq & R & & (1)\\
& \sum_t\sum_{w\in\mathcal{W}_{st}} z_{stw} & \leq & 1 & \forall\, s\in\mathcal{S} & (2)
\end{array}$$

I

$$\begin{array}{rcll}
\sum_t\sum_{w\in\mathcal{W}_{st}} A_{vtw}\cdot z_{stw} & = & x_{sv} & \forall\, s,v & (3)\\
C_i-\sum_t\sum_{w\in\mathcal{W}_{it}} d_{it}\cdot z_{itw} & = & D_i & \forall\, i\in\bar{\mathcal{S}} & (4)\\
\sum_{(s,v)\in\Delta_i} x_{sv} & = & B_i & \forall\, i\in\bar{\mathcal{S}} & (5)\\
\sum_s\sum_v u_{isv}\ +\ S_i & = & D_i & \forall\, i\in\bar{\mathcal{S}} & (6)
\end{array}$$

II

$$\begin{array}{rcll}
D_i-F_i\cdot N_i+(F_iM_i)\cdot q_i & \geq & S_i & \forall\, i\in\bar{\mathcal{S}} & (7a)\\
D_i-F_i\cdot N_i & \leq & S_i & \forall\, i\in\bar{\mathcal{S}} & (7b)\\
C_i\cdot(1-q_i) & \geq & S_i & \forall\, i\in\bar{\mathcal{S}} & (7c)\\
F_i\cdot N_i+C_i\cdot(1-q_i) & \geq & D_i & \forall\, i\in\bar{\mathcal{S}} & (8a)\\
D_i-1+(F_iM_i+1)\cdot q_i & \geq & F_i\cdot N_i & \forall\, i\in\bar{\mathcal{S}} & (8b)\\
D_i+M_i\cdot y_i & \geq & B_i & \forall\, i\in\bar{\mathcal{S}} & (9a)\\
B_i+C_i\cdot(1-y_i) & \geq & D_i & \forall\, i\in\bar{\mathcal{S}} & (9b)\\
G_i & \geq & N_i & \forall\, i\in\bar{\mathcal{S}} & (10a)\\
B_i & \geq & N_i & \forall\, i\in\bar{\mathcal{S}} & (10b)\\
G_i-M_i\cdot(1-z_i) & \leq & N_i & \forall\, i\in\bar{\mathcal{S}} & (10c)\\
B_i-M_i\cdot z_i & \leq & N_i & \forall\, i\in\bar{\mathcal{S}} & (10d)
\end{array}$$

III

$$
\begin{array}{rcll}
F_i \cdot U_{isv} & \geq & u_{isv} & \forall\ i,\ (s,v) \in \Delta_i \quad (11a) \\
F_i \cdot x_{sv} & \geq & u_{isv} & \forall\ i,\ (s,v) \in \Delta_i \quad (11b) \\
1 + (F_i - 1) \cdot (1 + z_i - y_i) & \geq & u_{isv} & \forall\ i,\ (s,v) \in \Delta_i \quad (12) \\
u_{isv} + F_i \cdot (1 - x_{sv}) & \geq & u_{i\bar{s}\bar{v}} & \forall\ i,\ (sv, \bar{s}\bar{v}) \in \mathcal{R}_i \quad (13a) \\
U_{isv} + (1 - x_{sv}) & \geq & U_{i\bar{s}\bar{v}} & \forall\ i,\ (sv, \bar{s}\bar{v}) \in \mathcal{R}_i \quad (13b) \\
u_{i\bar{s}\bar{v}} + 1 + (F_i - 1) \cdot (1 - U_{i\bar{s}\bar{v}}) & \geq & u_{isv} & \forall\ i,\ (sv, \bar{s}\bar{v}) \in \mathcal{R}_i \quad (14) \\
F_i \cdot (U_{isv} - q_i) & \leq & u_{isv} & \forall\ i,\ (s,v) \in \Delta_i \quad (15) \\
U_{isv} & \leq & x_{sv} & \forall\ i,\ (s,v) \in \Delta_i \quad (16) \\
\sum_{(s,v) \in \Delta_i} U_{isv} & = & N_i & \forall\ i \in \bar{\mathcal{S}} \quad (17)
\end{array}
$$

IV

$$
\begin{array}{rcll}
\sum_t \sum_{w \in \mathcal{W}_{st}} A_{vtw} \cdot m_t \cdot z_{stw} & = & N_{sv} & \forall\ s \in \mathcal{S},\ v \in \mathcal{V} \quad (18) \\
\sum_{k=1}^{K} k \cdot n_{svk} & = & N_{sv} & \forall\ s \in \mathcal{S},\ v \in \mathcal{V} \quad (19) \\
\sum_{k=1}^{K} n_{svk} & \leq & 1 & \forall\ s \in \mathcal{S},\ v \in \mathcal{V} \quad (20) \\
\sum_{k=1}^{K} u^k_{isv} & = & u_{isv} & \forall\ i,\ (s,v) \in \Delta_i \quad (21) \\
F_i \cdot n_{svk} & \geq & u^k_{isv} & \forall\ i,\ (s,v) \in \Delta_i \\
 & & & k = 1, \ldots, K \quad (22)
\end{array}
$$

X

$$
\begin{array}{rcll}
B_i,\ D_i,\ N_i,\ u_{isv},\ u^k_{isv},\ N_{sv},\ S_i & \in & \mathbb{Z}^+ & \forall\ i, s, v, k \quad (30) \\
z_{stw},\ x_{sv},\ y_i,\ q_i,\ z_i,\ n_{svk},\ U_{isv} & \in & \{0,1\} & \forall\ i, s, t, v, w, k \quad (31)
\end{array}
$$

The Mixed Integer Linear Programming Model

$$\max \quad \sum_{s} \sum_{t} \sum_{w \in W_{st}} r_s \cdot P_{stw} \qquad [MILP]$$

$$s.t. \quad \sum_{s} \sum_{t} \sum_{w \in \mathcal{W}_{st}} n_t \cdot z_{stw} \leq R \tag{1}$$

$$\sum_{t} \sum_{w \in \mathcal{W}_{st}} z_{stw} \leq 1 \qquad \forall\, s \in \mathcal{S} \tag{2}$$

I

$$\sum_{t} \sum_{w \in \mathcal{W}_{st}} A_{vtw} \cdot z_{stw} = x_{sv} \qquad \forall\, s, v \tag{3}$$

$$C_i - \sum_{t} \sum_{w \in \mathcal{W}_{it}} d_{it} \cdot z_{itw} = D_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{4}$$

$$\sum_{(s,v) \in \Delta_i} x_{sv} = B_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{5}$$

$$\sum_{s} \sum_{v} u_{isv} + S_i = D_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{6}$$

II

$$D_i - F_i \cdot N_i + (F_i M_i) \cdot q_i \geq S_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{7a}$$

$$D_i - F_i \cdot N_i \leq S_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{7b}$$

$$C_i \cdot (1 - q_i) \geq S_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{7c}$$

$$F_i \cdot N_i + C_i \cdot (1 - q_i) \geq D_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{8a}$$

$$D_i - 1 + (F_i M_i + 1) \cdot q_i \geq F_i \cdot N_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{8b}$$

$$D_i + M_i \cdot y_i \geq B_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{9a}$$

$$B_i + C_i \cdot (1 - y_i) \geq D_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{9b}$$

$$G_i \geq N_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{10a}$$

$$B_i \geq N_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{10b}$$

$$G_i - M_i \cdot (1 - z_i) \leq N_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{10c}$$

$$B_i - M_i \cdot z_i \leq N_i \qquad \forall\, i \in \bar{\mathcal{S}} \tag{10d}$$

III

$$F_i \cdot U_{isv} \geq u_{isv} \quad \forall\, i,\ (s,v) \in \Delta_i \quad (11a)$$

$$F_i \cdot x_{sv} \geq u_{isv} \quad \forall\, i,\ (s,v) \in \Delta_i \quad (11b)$$

$$1 + (F_i - 1) \cdot (1 + z_i - y_i) \geq u_{isv} \quad \forall\, i,\ (s,v) \in \Delta_i \quad (12)$$

$$u_{isv} + F_i \cdot (1 - x_{sv}) \geq u_{i\bar{s}\bar{v}} \quad \forall\, r_{isv} < r_{i\bar{s}\bar{v}} \quad (13a)$$

$$U_{isv} + (1 - x_{sv}) \geq U_{i\bar{s}\bar{v}} \quad \forall\, r_{isv} < r_{i\bar{s}\bar{v}} \quad (13b)$$

$$u_{i\bar{s}\bar{v}} + 1 + (F_i - 1) \cdot (1 - U_{i\bar{s}\bar{v}}) \geq u_{isv} \quad \forall\, r_{isv} < r_{i\bar{s}\bar{v}} \quad (14)$$

$$F_i \cdot (U_{isv} - q_i) \leq u_{isv} \quad \forall\, i,\ (s,v) \in \Delta_i \quad (15)$$

$$U_{isv} \leq x_{sv} \quad \forall\, i,\ (s,v) \in \Delta_i \quad (16)$$

$$\sum_{(s,v) \in \Delta_i} U_{isv} = N_i \quad \forall\, i \in \bar{\mathcal{S}} \quad (17)$$

IV

$$\sum_t \sum_{w \in \mathcal{W}_{st}} A_{vtw} \cdot m_t \cdot z_{stw} = N_{sv} \quad \forall\, s \in \mathcal{S},\ v \in \mathcal{V} \quad (18)$$

$$\sum_{k=1}^{K} k \cdot n_{svk} = N_{sv} \quad \forall\, s \in \mathcal{S},\ v \in \mathcal{V} \quad (19)$$

$$\sum_{k=1}^{K} n_{svk} \leq 1 \quad \forall\, s \in \mathcal{S},\ v \in \mathcal{V} \quad (20)$$

$$\sum_{k=1}^{K} u_{isv}^{k} = u_{isv} \quad \forall\, i,\ (s,v) \in \Delta_i \quad (21)$$

$$F_i \cdot n_{svk} \geq u_{isv}^{k} \quad \forall\, i,\ (s,v) \in \Delta_i,\ k = 1, \ldots, K \quad (22)$$

V

$$z_{stw} \geq P_{stw} \qquad \forall\, s,t,w \qquad (23)$$

$$\sum_{l=1}^{L_Y} \bar{p}_l \cdot \lambda_{stwl} + \bar{p}_0 \geq P_{stw} \qquad \forall\, s,t,w \qquad (24a)$$

$$\sum_{l=1}^{L_Y} \bar{y}_l \cdot \lambda_{stwl} + \bar{y}_0 = Y_{stw} \qquad \forall\, s,t,w \qquad (24b)$$

$$\sum_{j=1}^{V_t} m_t \cdot \overline{Y}_{stw_j} = Y_{stw} \qquad \forall\, s,t,w \qquad (25)$$

$$\sum_{l=1}^{L_X} \hat{y}_l \cdot \alpha_{stvl} + \hat{y}_0 = \overline{Y}_{stv} \qquad \forall\, s,t,v \qquad (26a)$$

$$\sum_{l=1}^{L_X} \hat{x}_l \cdot \alpha_{stvl} + \hat{x}_0 = X_{stv} \qquad \forall\, s,t,v \qquad (26b)$$

$$\alpha_{stvl} \leq \pi_{stvl} \qquad \forall\, s,t,v,l \in \mathcal{L}_X \qquad (26c)$$

$$\pi_{stv(l+1)} \leq \alpha_{stvl} \qquad \forall\, s,t,v,l \in \mathcal{L}_X^{-1} \qquad (26d)$$

$$\sum_{\substack{i \in \bar{S} \\ i \neq s}} \sum_{k=1}^{K} \mathrm{LP}_{isv}^k \cdot u_{isv}^k + \mathrm{LP}_{st} \geq X_{stv} \qquad \forall\, s,t,v \qquad (27)$$

VI

$$(1 - z_{stw}) \leq \lambda_{stwl} \qquad \forall\, s,t,w,l \in \mathcal{L}_Y \qquad (28)$$

$$-10 \cdot (1 - z_{stw}) \geq X_{stw_j} \qquad \forall\, s,t,w,j \in V_t \qquad (29)$$

X

$$B_i,\ D_i,\ N_i,\ u_{isv},\ u_{isv}^k,\ N_{sv},\ S_i \in \mathbb{N}^+ \qquad \forall\, i,s,v,k \in K \qquad (30)$$

$$z_{stw},\ x_{sv},\ y_i,\ q_i,\ z_i,\ n_{svk},\ U_{isv} \in \{0,1\} \qquad \forall\, i,s,t,v,w,k \qquad (31)$$

$$P_{stw},\ \lambda_{stwl},\ \alpha_{stvl} \in [0,1] \qquad \forall\, s,t,v,w,l \qquad (32)$$

$$Y_{stw},\ \overline{Y}_{stv},\ X_{stv} \leq 0 \qquad \forall\, s,t,v,w \qquad (33)$$

$$\pi_{stvl} \in \{0,1\} \qquad \forall\, s,t,v,l \in \mathcal{L}_X \qquad (34)$$

12 Comprehensive Results

Here follows a complete set of graphs for all Scenarios and Cases considered in the benchmark. In all result graphs, the x-axis represents the number of resources available, while the y-axis correspond to objective values or solution times in minutes.

Defender Unit Settings

For this benchmark, all defender units have identical defensive capabilities. They have a 10 km defensive radius in which they can defend themselves, and other units, using one or more of their 8 defensive channels. Table 4 defines the probability of success for each resource part of the considered tactics, along with the number of defensive channels occupied.

Table 4: Full characteristics of all considered tactics.

	Defenders		Targets	Tactics		
t	p_{it}	d_{it}	p_{st}	n_t	V_t	m_t
1	0.700	2	0.999	1	1	1
2	0.736	3	0.999	2	1	2
3	0.753	4	0.999	3	1	3
4	0.776	3	0.999	2	2	1
5	0.830	6	0.999	3	3	1

At most $G_i = 3$ number of active paths might be engaged and no more than $F_i = 4$ defensive channels can be used against a single engagement path. Further, the defensive parameters $\theta_{ik} = 0.7$ and $\beta_{ik} = 2$ for all units $i \in \bar{\mathcal{S}}$ and $k = 1, \ldots, K$.

Objective value graphs

Starting with result graphs for the objective, the upper and lower blue lines represent the optimistic and pessimistic bounds found by CPLEX, where the pessimistic solutions have been evaluated using the real objective function. The green dots represent the pessimistic value given by CPLEX.

The black dots, found in between the blue lines, are the local search improved solutions. We can see that for most of the instances, this improvement is substantial. The black line with crosses indicate the best found solutions from the SA heuristic. The red line with dots show the result of the Augmented Solution approach.

The magenta colored lines show the range for the 100 local searches performed, together with a cross indicating the best objective of the 5000 solution generated in the other test.

Solution Time graphs

Solution times for CPLEX to find all the optimistic and pessimistic solutions are instant. The local search for each pessimistic solution do take some time, and the solution times are represented by the small red bars, not even visible for some of the instances.

The blue bars represent the time needed to generate and evaluate 5000 feasible solutions. Solution times for generation and evaluation of 100 feasible solutions, followed by a local search, are represented by the yellow bars. Since this is extremely time consuming for most instances, they are only found for 5, 10, 15 and so on up to 30 resources for the biggest scenarios.

The magenta bars represent solution times for the SA heuristic. Finally, the green bars show the solution time for the Augmented Solution algorithm.

The MILP formulation

The MILP model, presented in the previous section, is not very helpful in practice. Even for a small sized problem including 5 units, it takes CPLEX considerable time (hours) to solve the model to optimality. Although good solutions are found quickly, the duality gap is closed slowly, and memory becomes an issue as well. Even if one had several hours or days at hand, the memory requirements would be substantial.

In order for the linearization to be accurate enough to be helpful, at least 10 to 15 line segments are needed for each of the approximations. This yields a lot of extra variables to the model, where some of them are binary, which introduces even more branching possibilities than before.

It would be interesting to tailor a restricted model, for situations where there are few resources compared to the number of units with positive reward. For such cases, it is most probable that only tactic 1 will be used in the optimal solution. And even if this is not optimal, such a model still provides valid pessimistic solutions.

Scenarios

A total of three different scenarios are considered, where the number and positions of the defender units differ. Also, for each of these scenarios, four different versions of positions and number of target units are solved.

Each of these 12 cases are solved for many different number of resources, ranging from 1 to 30 available resources. The distance between the target units is between 300-500 meters.

Scenario 1

The first scenario includes two defender units positioned 10 km apart, each with a defensive radius of 10 km. One unit in the picture corresponds to 1 km. In the different cases, we consider 5, 7, 8 and 14 target units respectively.

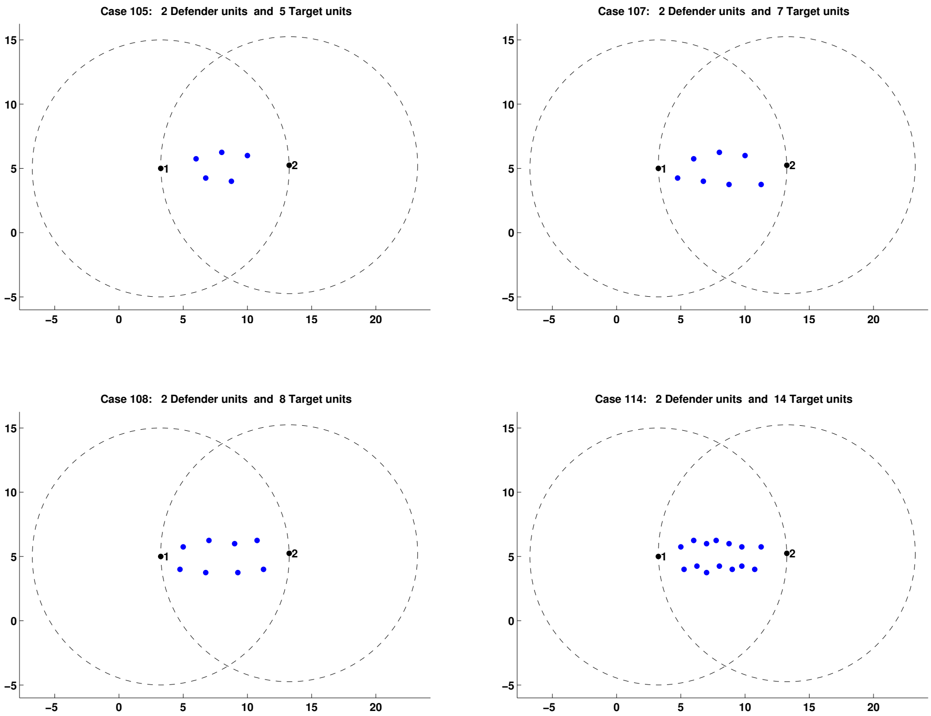


Figure 24: Scenario 1 and the 4 different cases. The top left figure presents Case 105 with 5 target units, represented by the blue dots, followed by Case 107 to the right. The bottom left figure is Case 108 and to the right Case 114.

Scenario 2

The second scenario includes five defender units positioned in an x-shaped formation, each with a defensive radius of 10 km. One unit in the picture corresponds to 1 km. In the different cases, we consider 5, 8, 12 and 16 target units respectively.

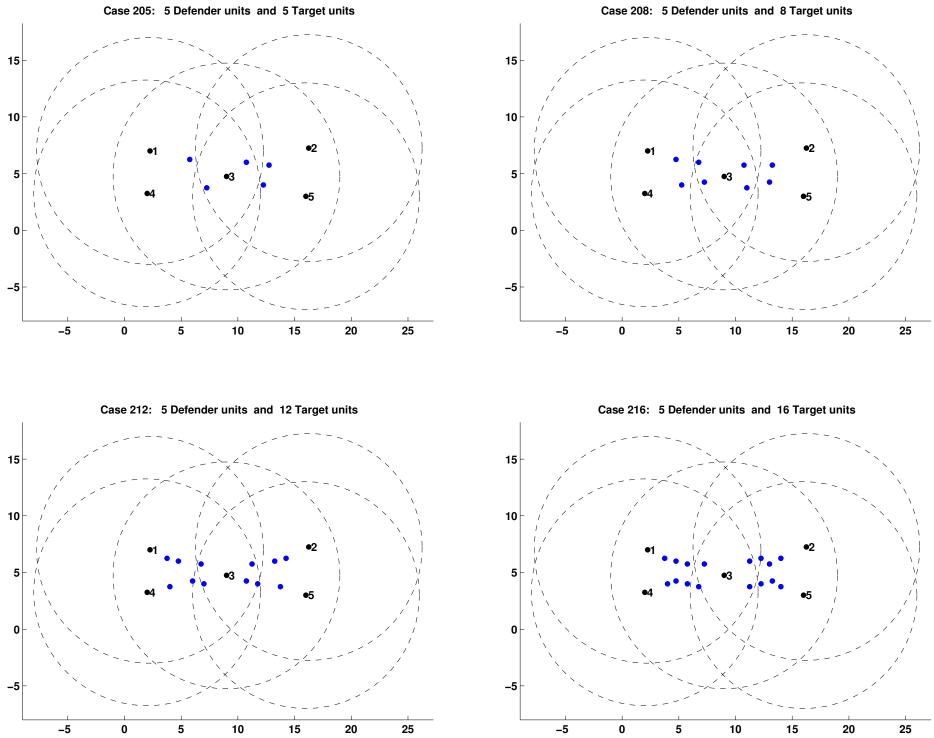


Figure 25: Scenario 2 and the 4 different cases. The top left figure presents Case 205 with 5 target units, represented by the blue dots, followed by Case 208 to the right. The bottom left figure is Case 212 and to the right Case 216.

Scenario 3

The third scenario includes four defender units, each with a defensive radius of 10 km. One unit in the picture corresponds to 1 km. In the different cases, we consider 5, 8, 12 and 16 target units respectively.

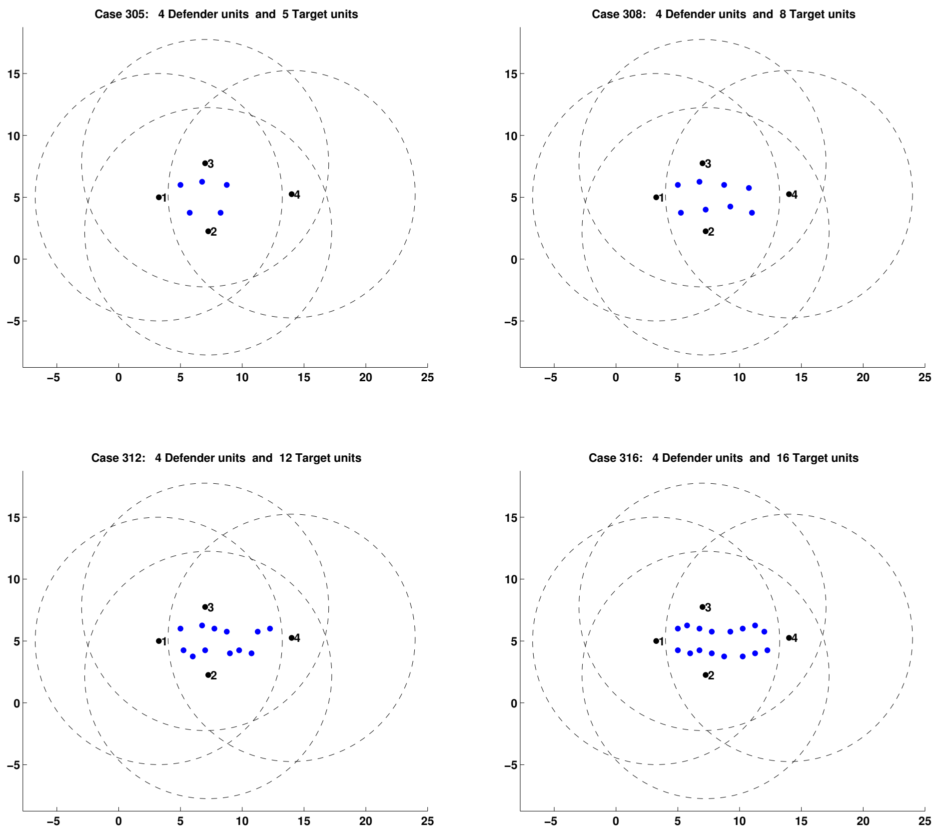


Figure 26: Scenario 3 and the 4 different cases. The top left figure presents Case 305 with 5 target units, represented by the blue dots, followed by Case 308 to the right. The bottom left figure is Case 312 and to the right Case 316.

12.1 Case 105

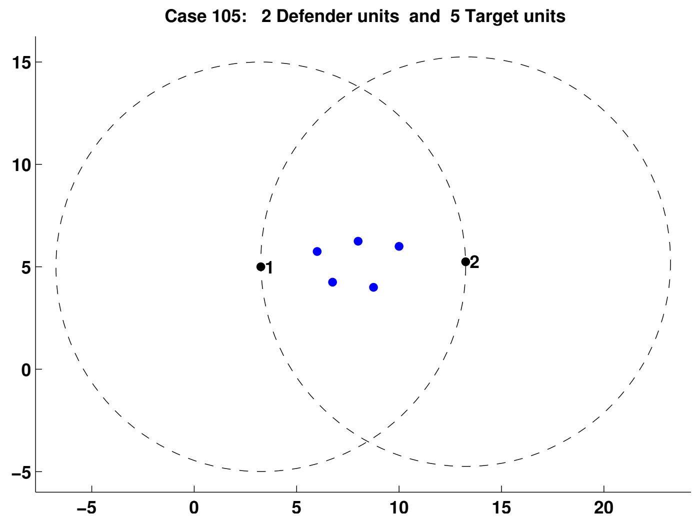


Figure 27: Test case 105. 2 defenders and 5 targets.

12.1.1 Reward setting 1

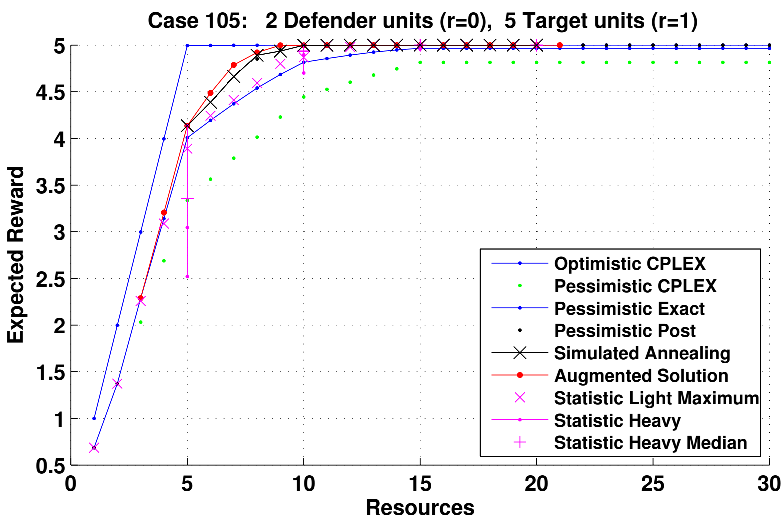


Figure 28: Test case 105. 2 defenders and 5 targets.

12.1.2 Reward setting 2

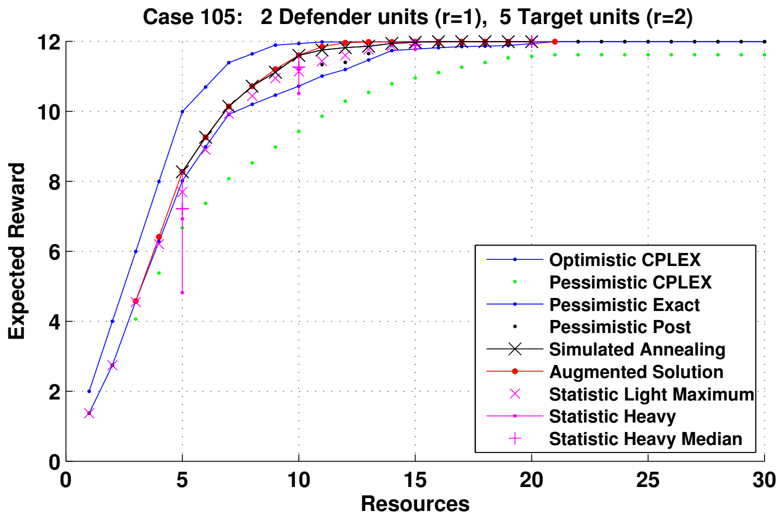


Figure 29: Test case 105. 2 defenders and 5 targets.

12.1.3 Reward setting 3

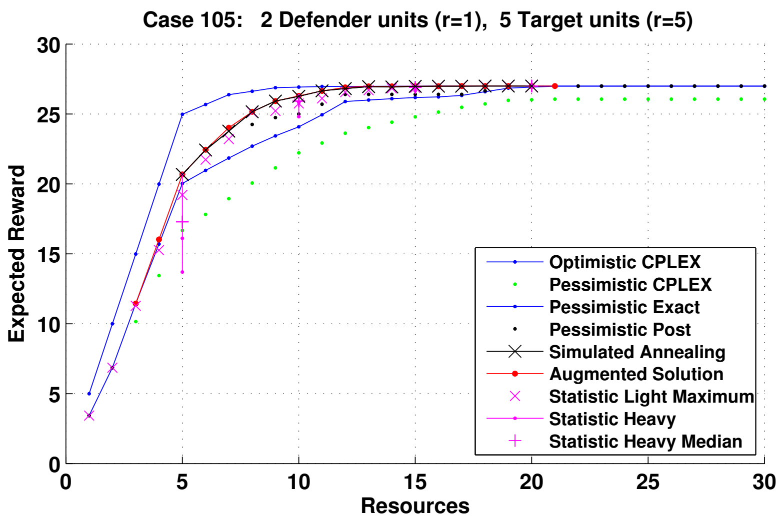


Figure 30: Test case 105. 2 defenders and 5 targets.

12.2 Case 107

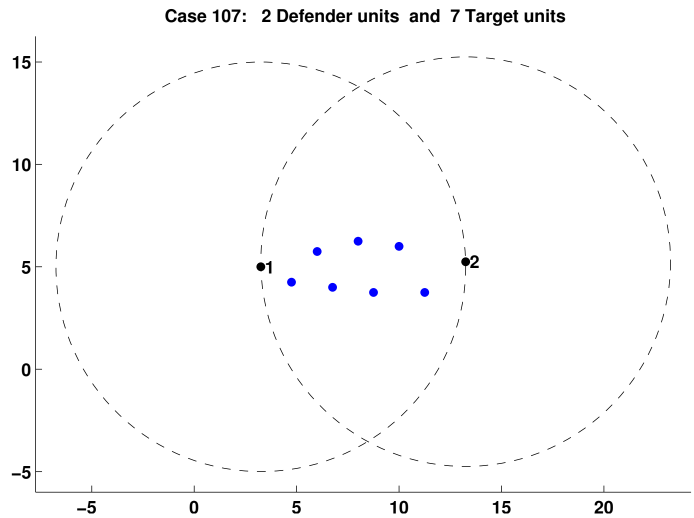


Figure 31: Test case 107. 2 defenders and 7 targets.

12.2.1 Reward setting 1

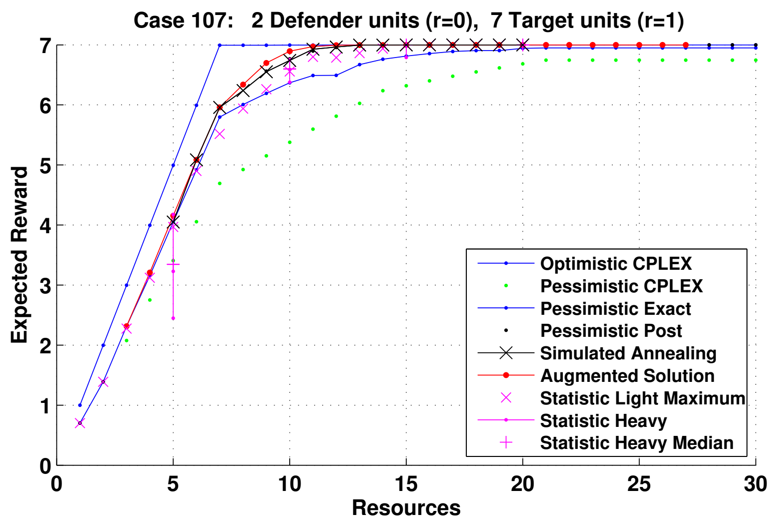


Figure 32: Test case 107. 2 defenders and 7 targets.

12.2.2 Reward setting 2

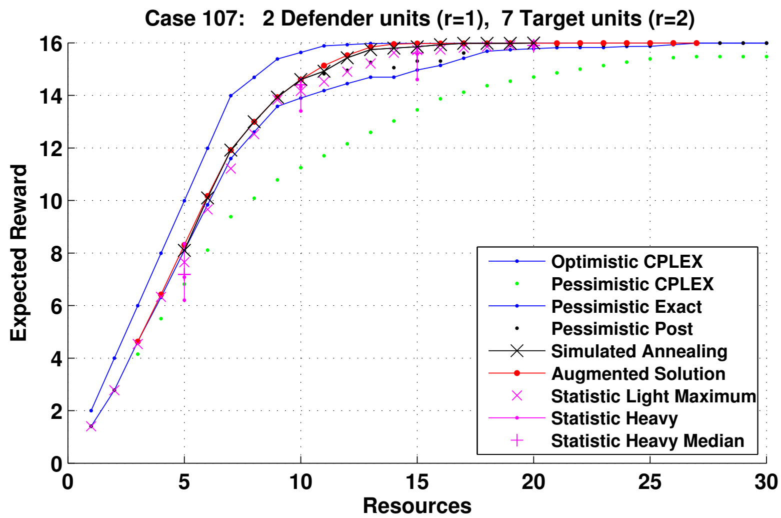


Figure 33: Test case 107. 2 defenders and 7 targets.

12.2.3 Reward setting 3

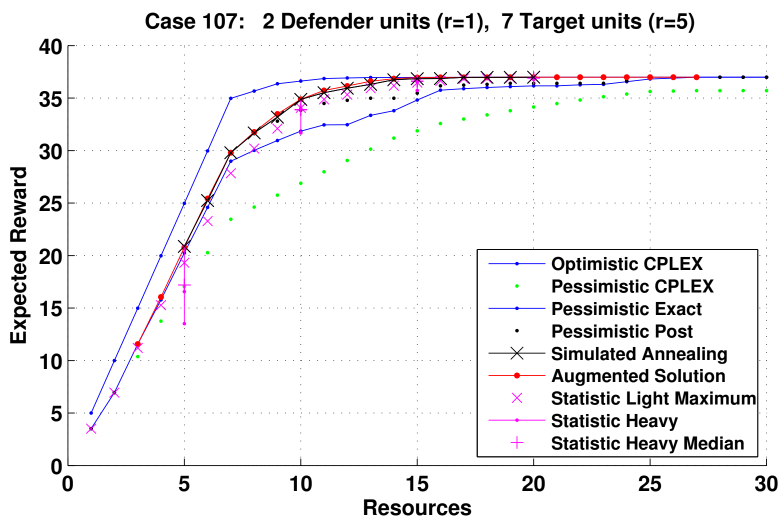


Figure 34: Test case 107. 2 defenders and 7 targets.

12.3 Case 108

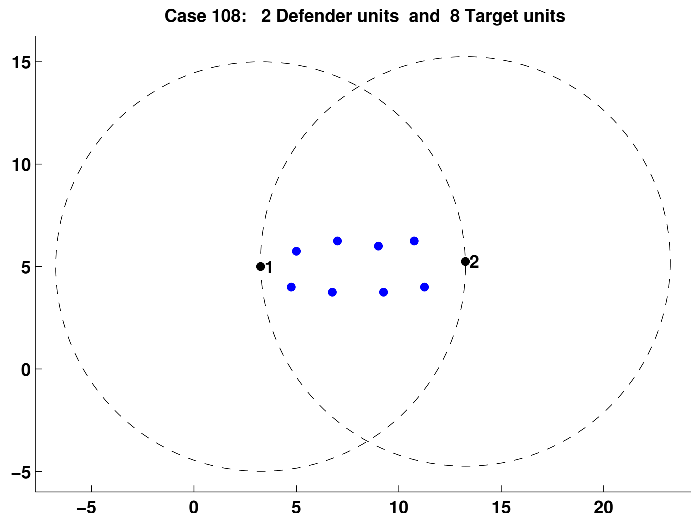


Figure 35: Test case 108. 2 defenders and 8 targets.

12.3.1 Reward setting 1

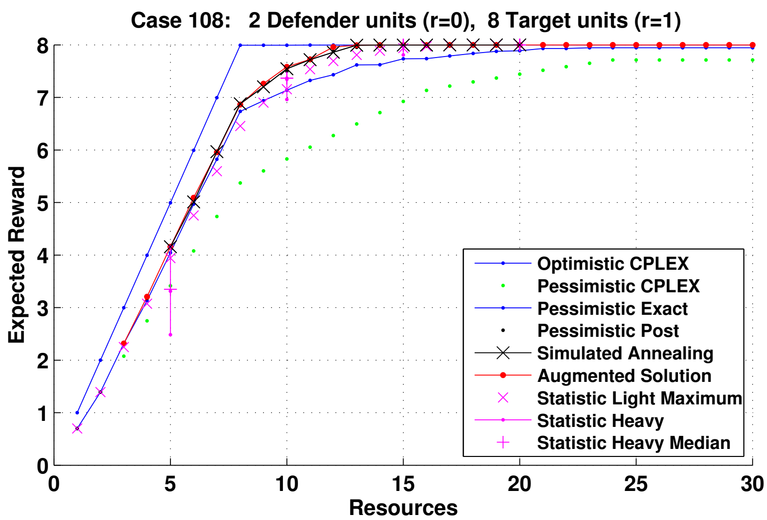


Figure 36: Test case 108. 2 defenders and 8 targets.

12.3.2 Reward setting 2

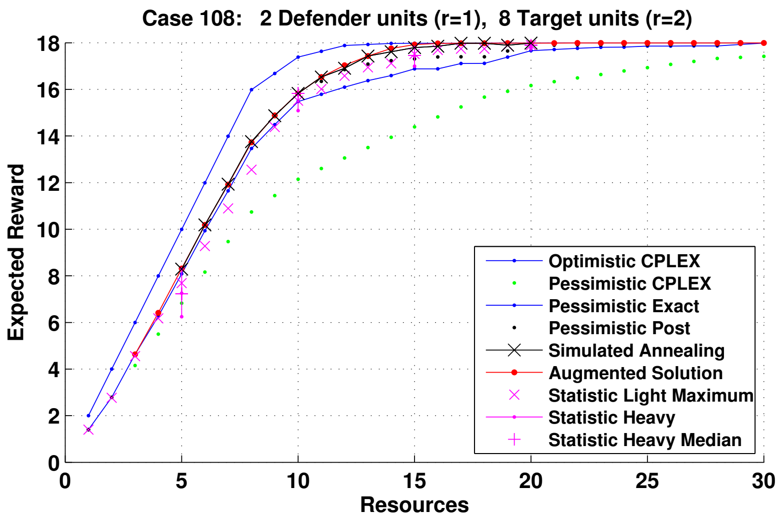


Figure 37: Test case 108. 2 defenders and 8 targets.

12.3.3 Reward setting 3

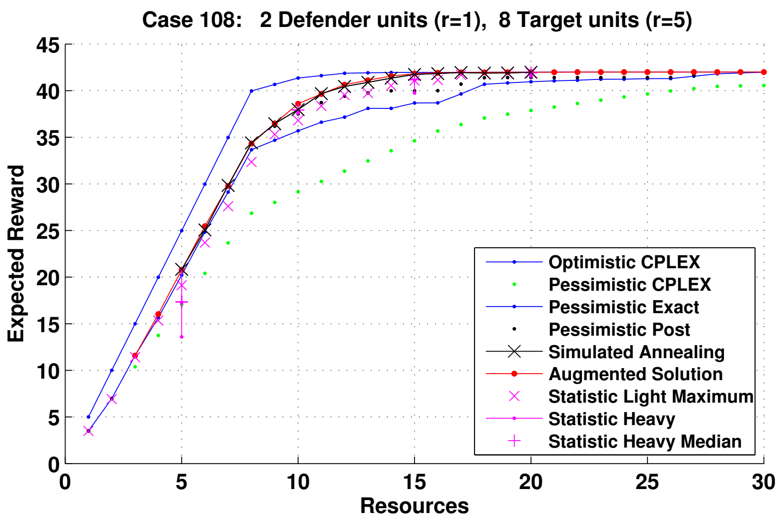


Figure 38: Test case 108. 2 defenders and 8 targets.

12.4 Case 114

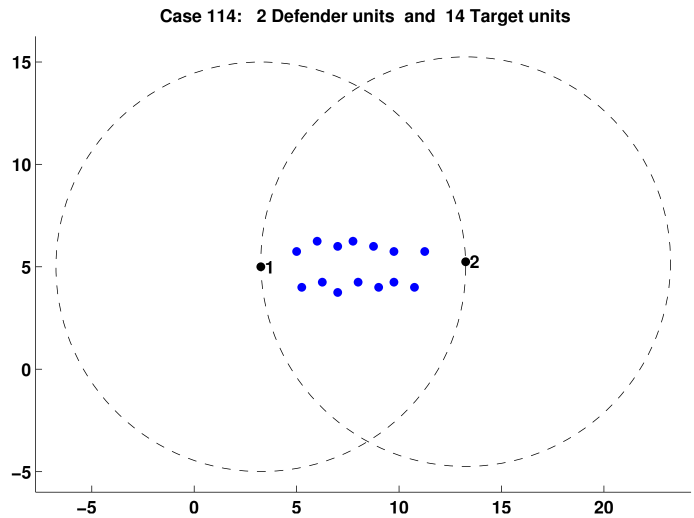


Figure 39: Test case 114. 2 defenders and 14 targets.

12.4.1 Reward setting 1

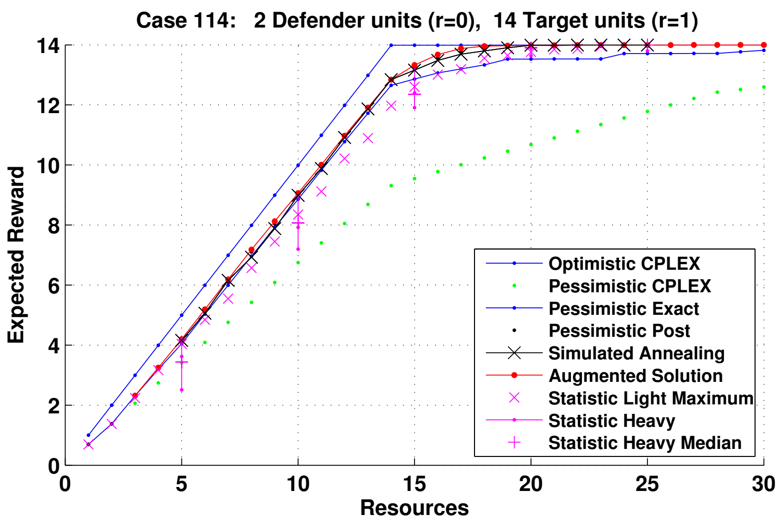


Figure 40: Test case 114. 2 defenders and 14 targets.

12.4.2 Reward setting 2

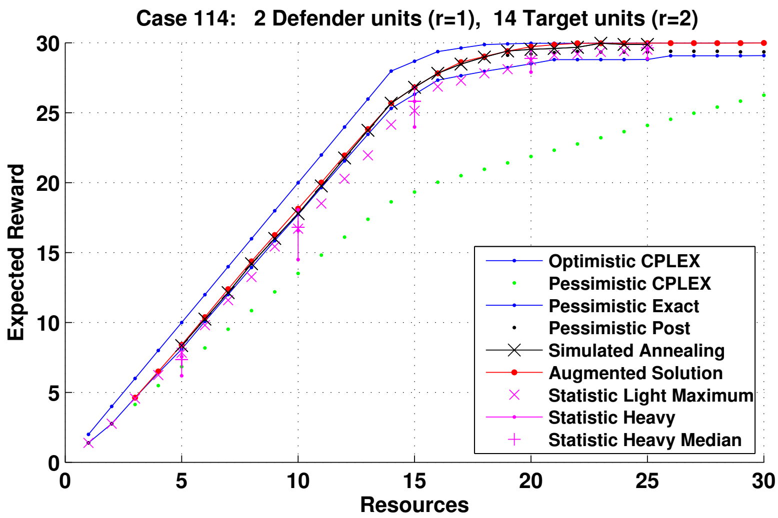


Figure 41: Test case 114. 2 defenders and 14 targets.

12.4.3 Reward setting 3

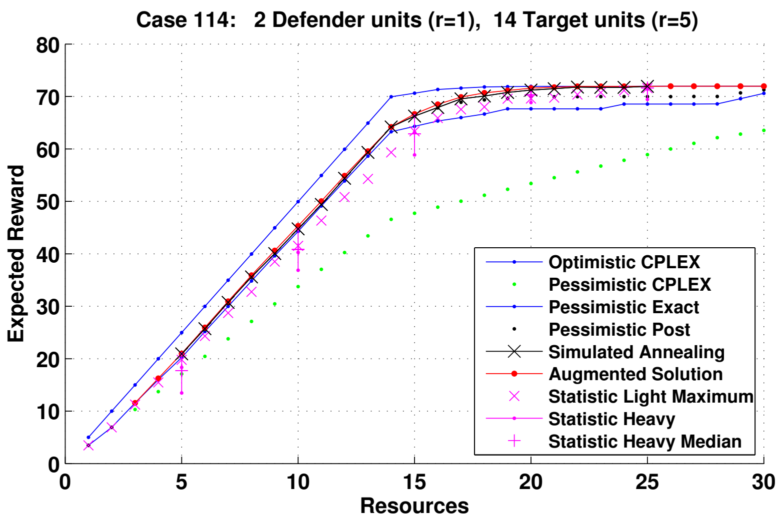


Figure 42: Test case 114. 2 defenders and 14 targets.

12.5 Scenario 1, Solution Times

12.5.1 Time Case 105

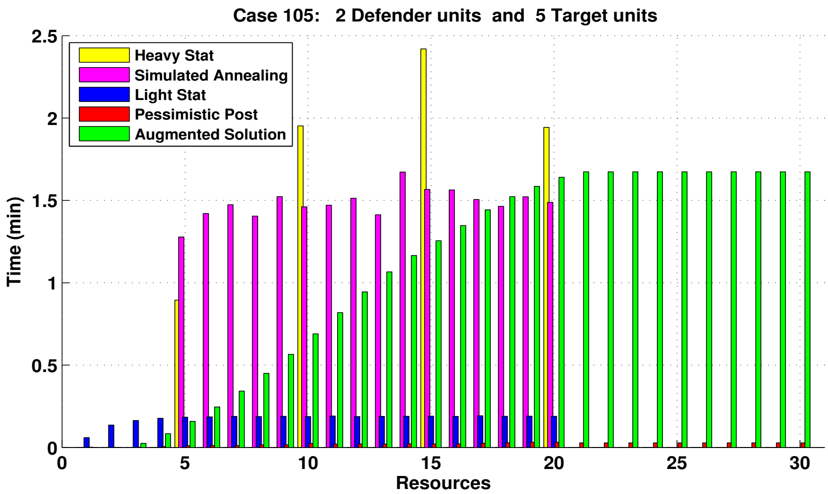


Figure 43: Test case 105. 2 defenders and 5 targets.

12.5.2 Time Case 107

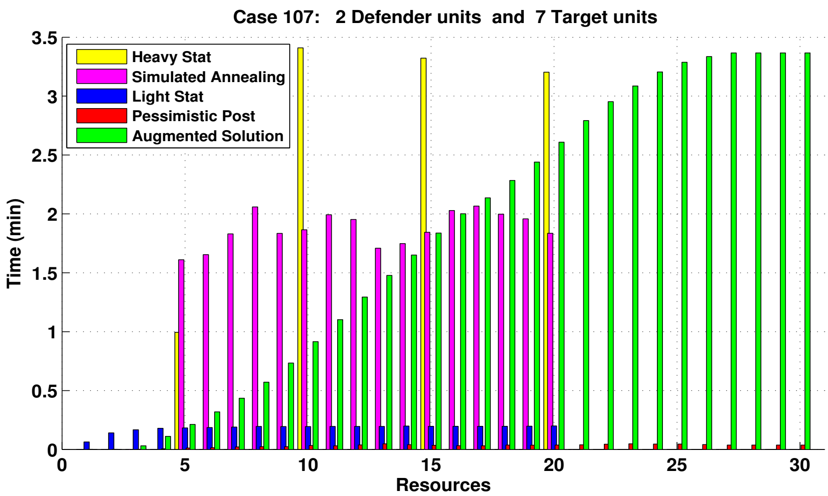


Figure 44: Test case 107. 2 defenders and 7 targets.

12.5.3 Time Case 108

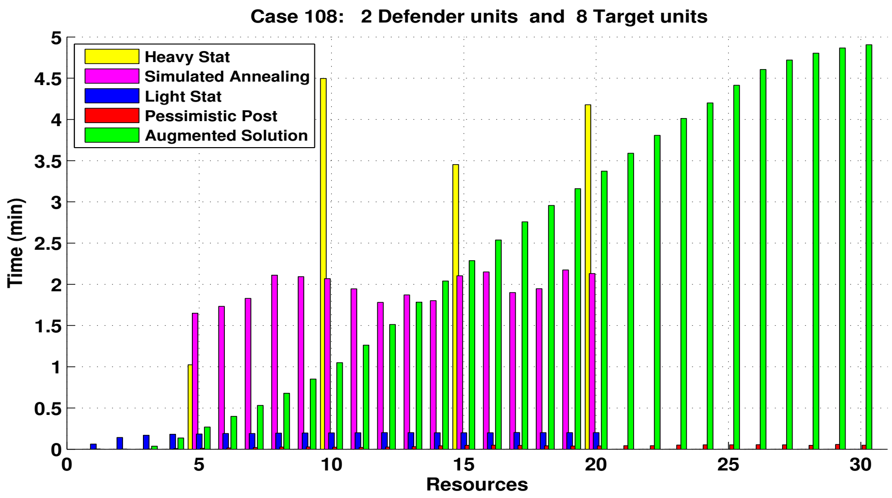


Figure 45: Test case 108. 2 defenders and 8 targets.

12.5.4 Time Case 114

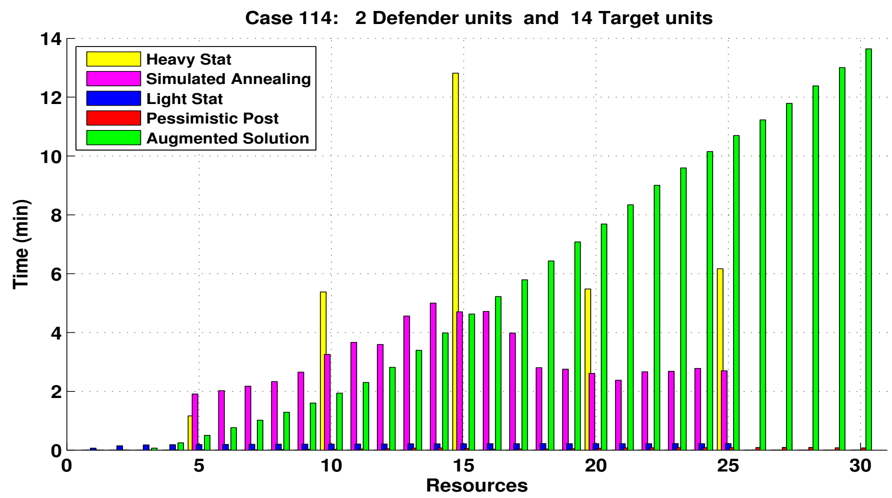


Figure 46: Test case 114. 2 defenders and 14 targets.

12.6 Case 205

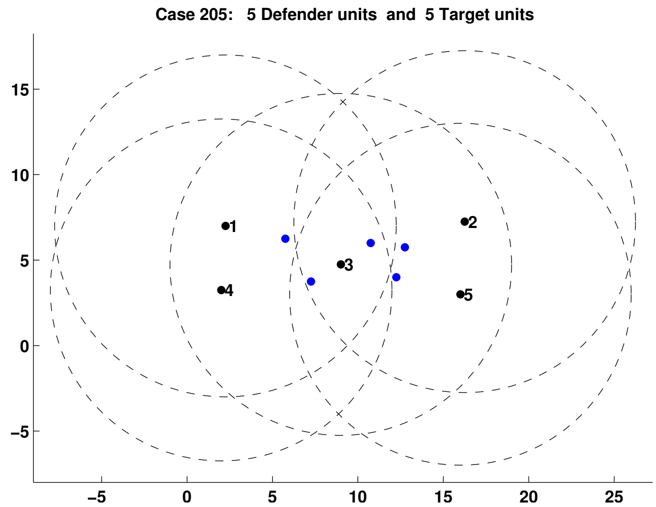


Figure 47: Test case 205. 5 defenders and 5 targets.

12.6.1 Reward setting 1

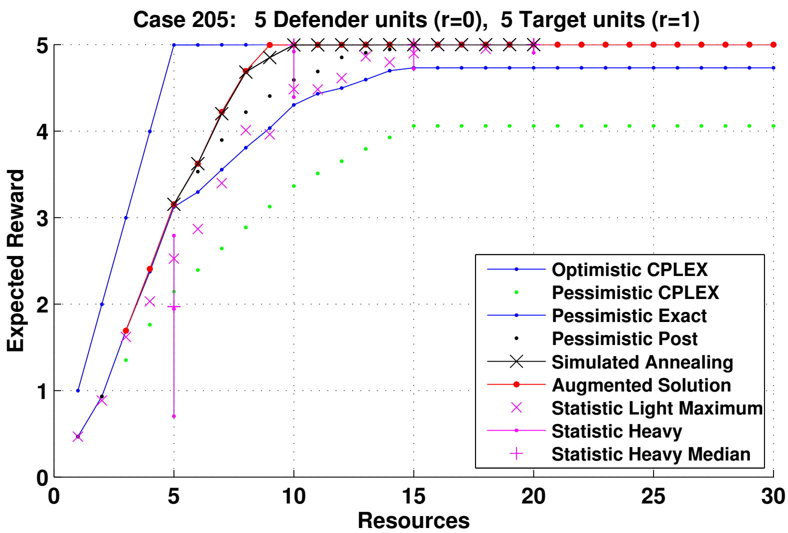


Figure 48: Test case 205. 5 defenders and 5 targets.

12.6.2 Reward setting 2

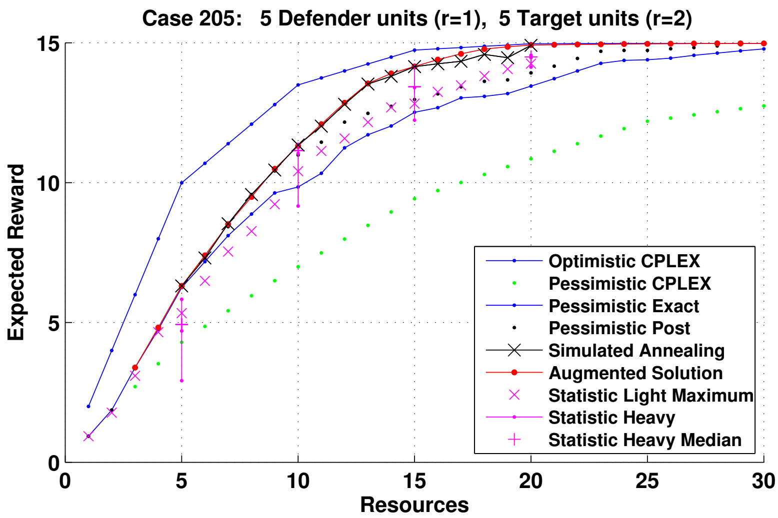


Figure 49: Test case 205. 5 defenders and 5 targets.

12.6.3 Reward setting 3

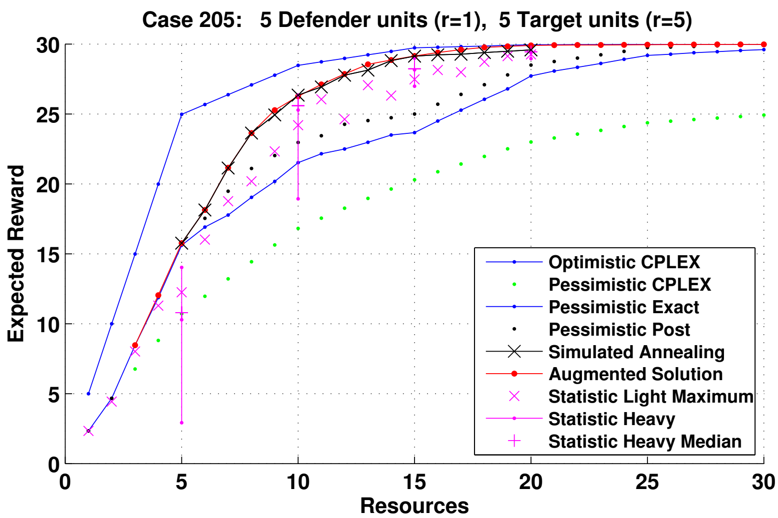


Figure 50: Test case 205. 5 defenders and 5 targets.

12.7 Case 208

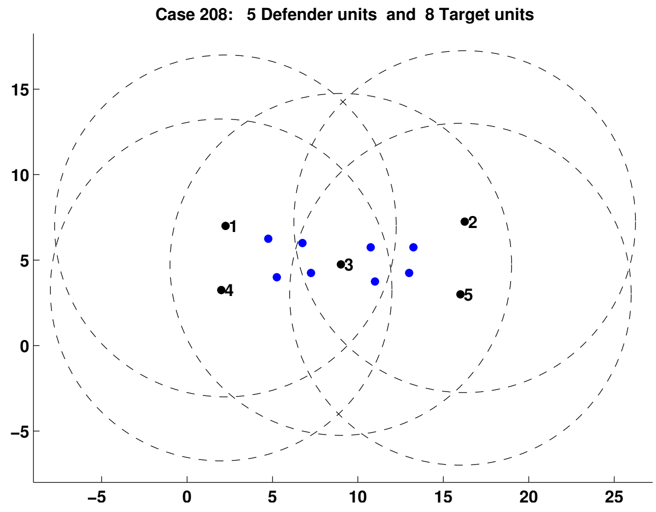


Figure 51: Test case 208. 5 defenders and 8 targets.

12.7.1 Reward setting 1

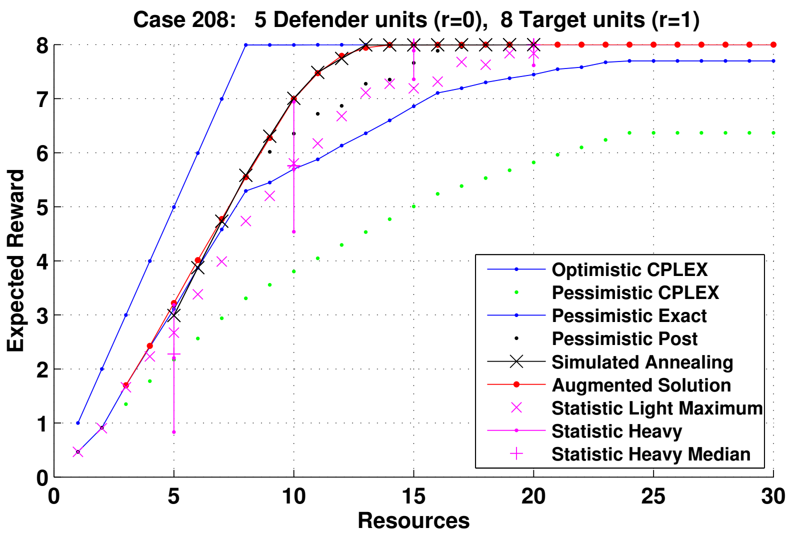


Figure 52: Test case 208. 5 defenders and 8 targets.

12.7.2 Reward setting 2

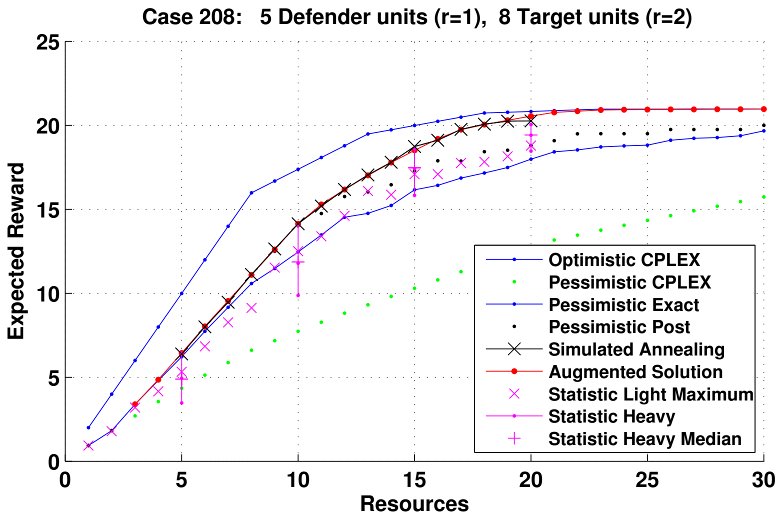


Figure 53: Test case 208. 5 defenders and 8 targets.

12.7.3 Reward setting 3

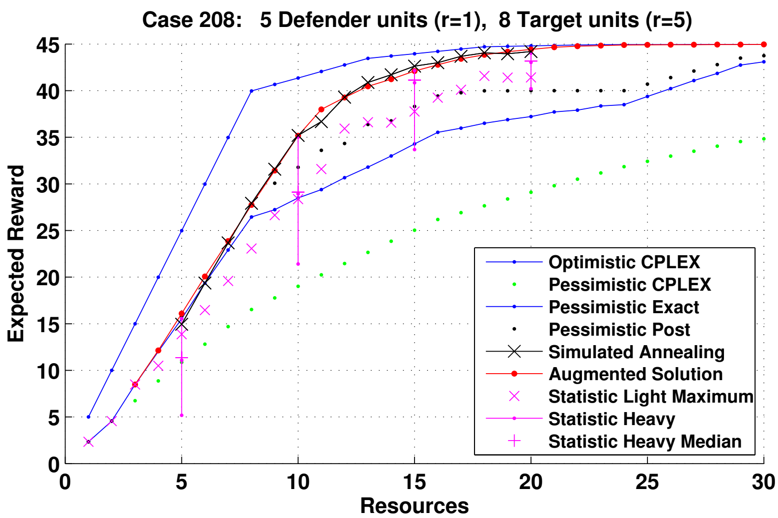


Figure 54: Test case 208. 5 defenders and 8 targets.

12.8 Case 212

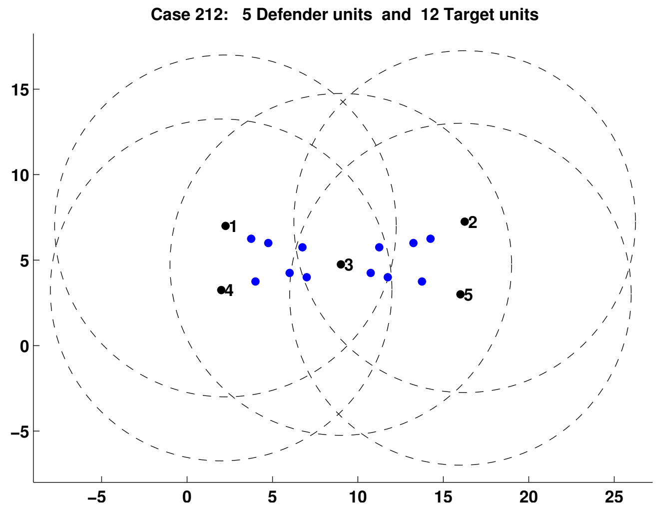


Figure 55: Test case 212. 5 defenders and 12 targets.

12.8.1 Reward setting 1

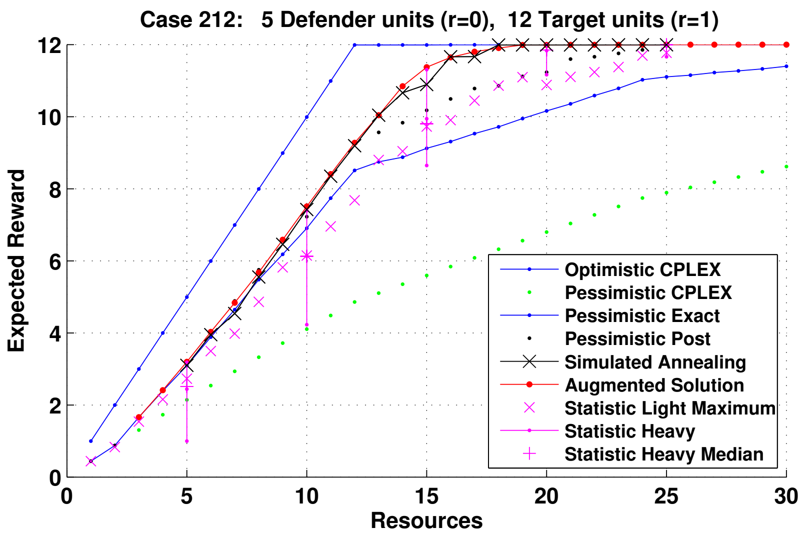


Figure 56: Test case 212. 5 defenders and 12 targets.

12.8.2 Reward setting 2

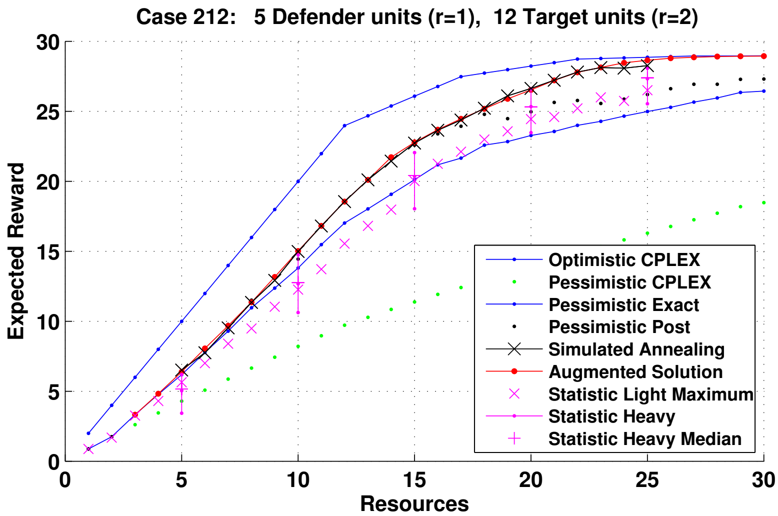


Figure 57: Test case 212. 5 defenders and 12 targets.

12.8.3 Reward setting 3

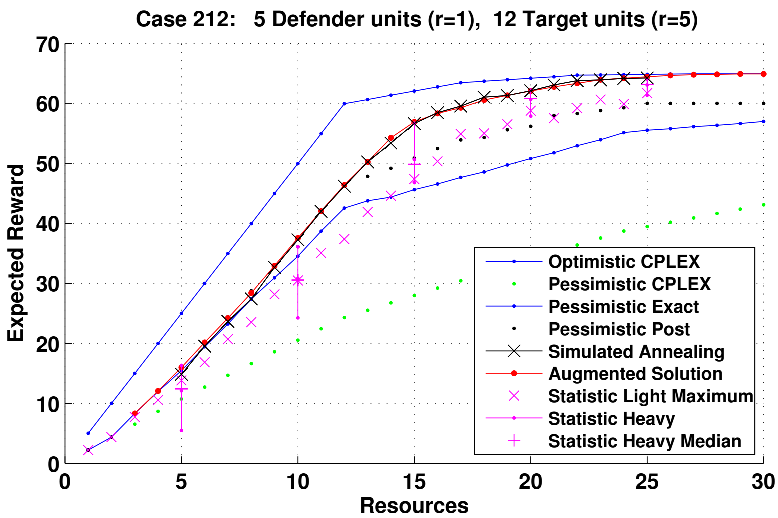


Figure 58: Test case 212. 5 defenders and 12 targets.

12.9 Case 216

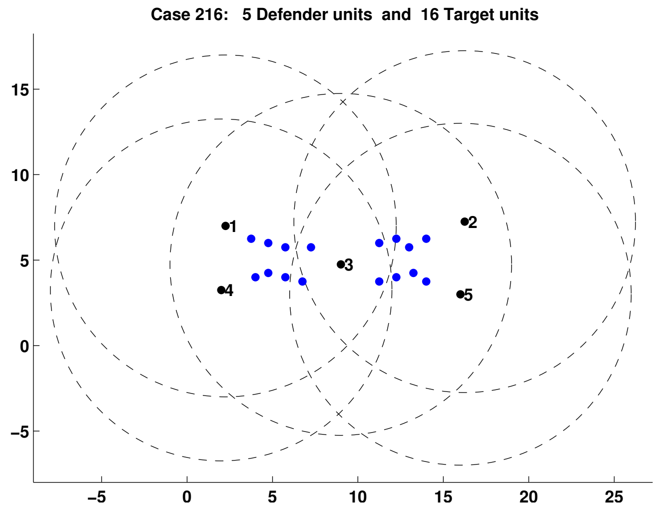


Figure 59: Test case 216. 5 defenders and 16 targets.

12.9.1 Reward setting 1

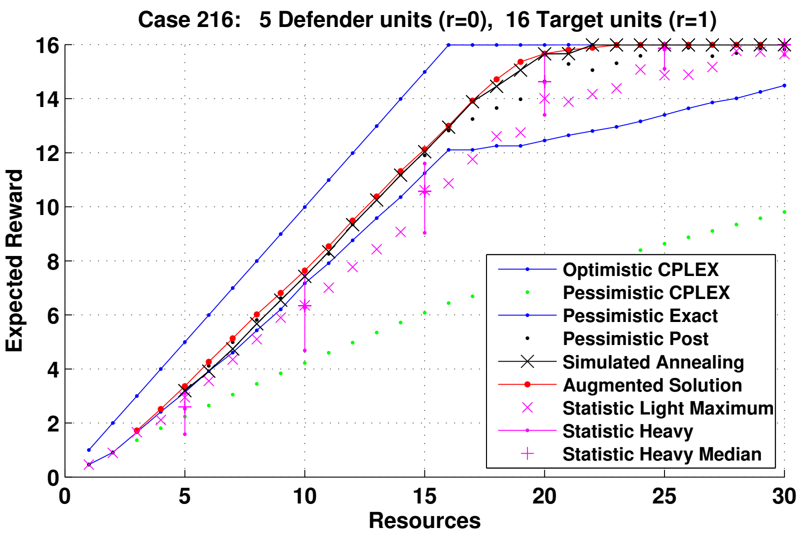


Figure 60: Test case 216. 5 defenders and 16 targets.

12.9.2 Reward setting 2

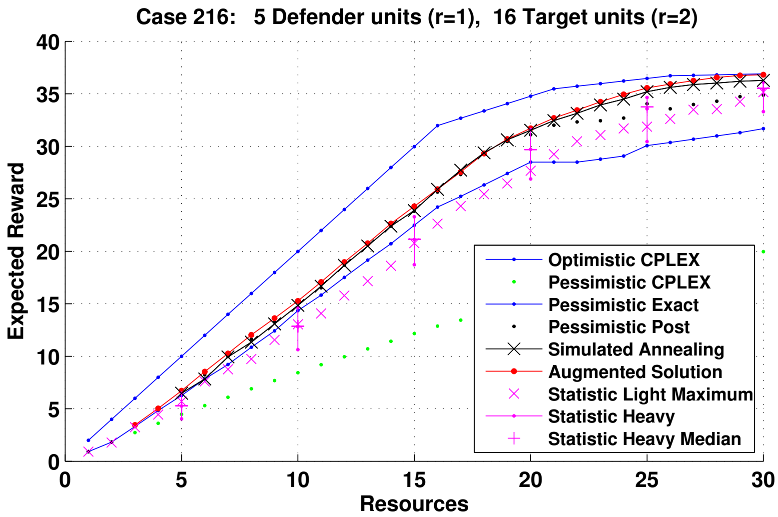


Figure 61: Test case 216. 5 defenders and 16 targets.

12.9.3 Reward setting 3

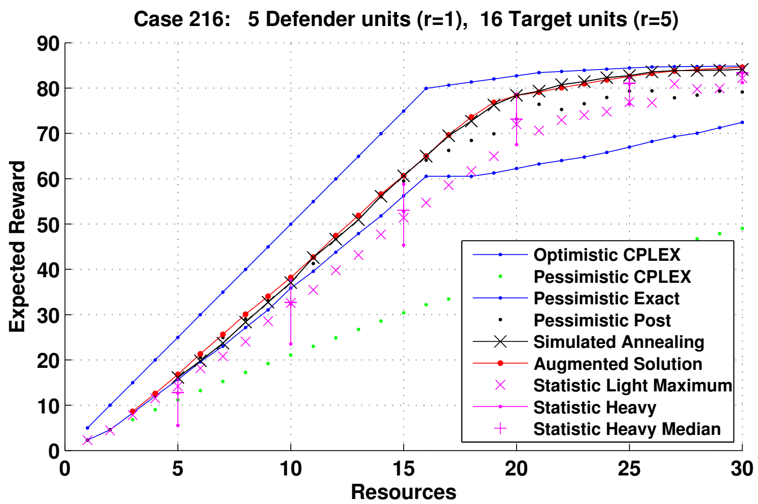


Figure 62: Test case 216. 5 defenders and 16 targets.

12.10 Scenario 2, Solution Times

12.10.1 Time Case 205

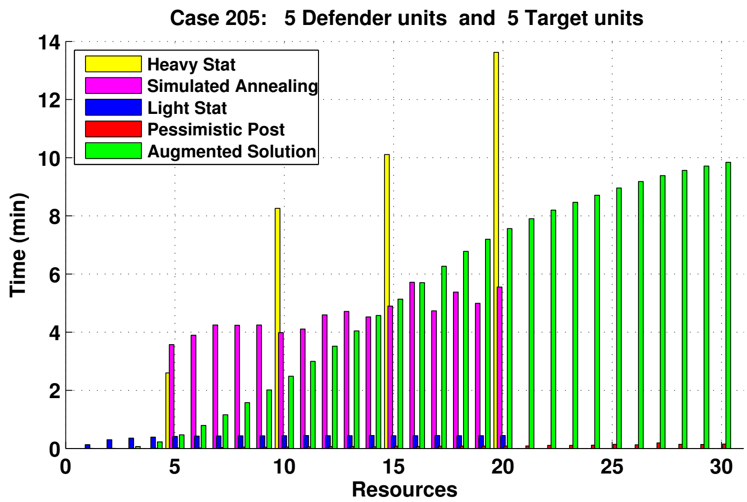


Figure 63: Test case 205. 5 defenders and 5 targets.

12.10.2 Time Case 208

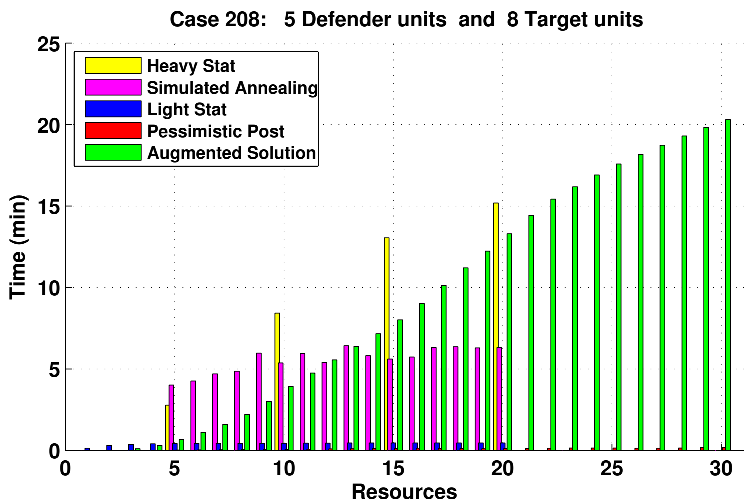


Figure 64: Test case 208. 5 defenders and 8 targets.

12.10.3 Time Case 212

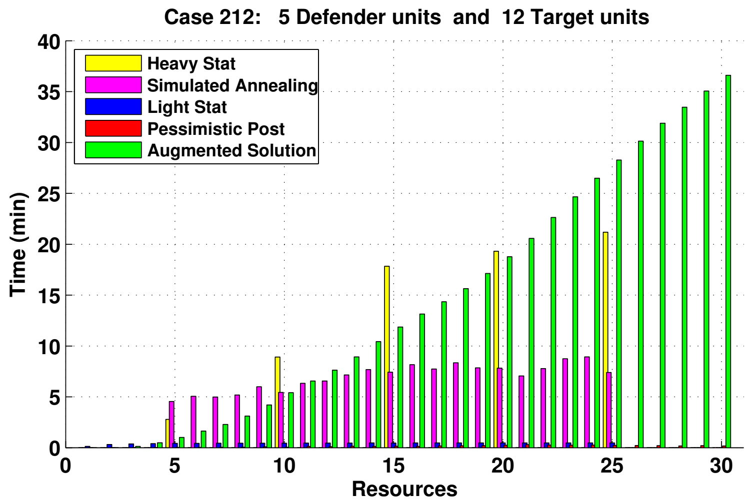


Figure 65: Test case 212. 5 defenders and 12 targets.

12.10.4 Time Case 216

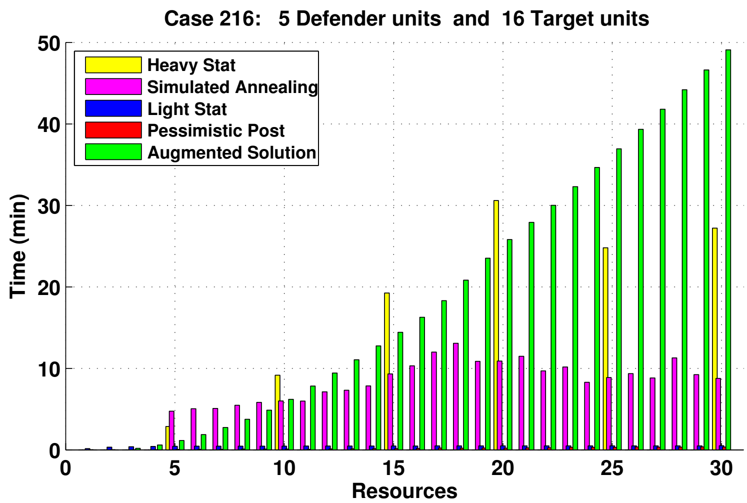


Figure 66: Test case 216. 5 defenders and 16 targets.

12.11 Case 305

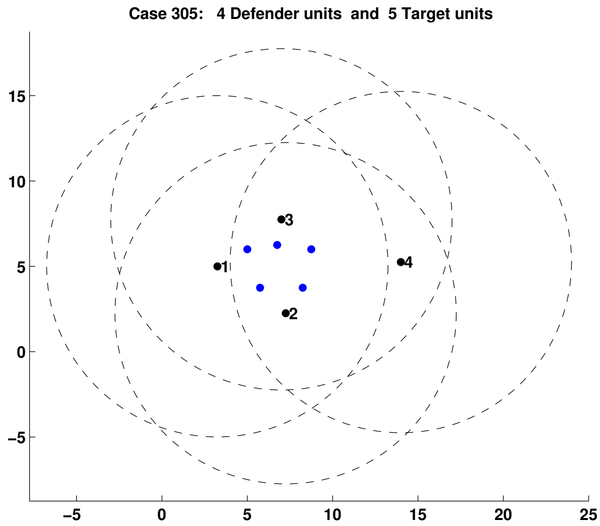


Figure 67: Test case 305. 4 defenders and 5 targets.

12.11.1 Reward setting 1

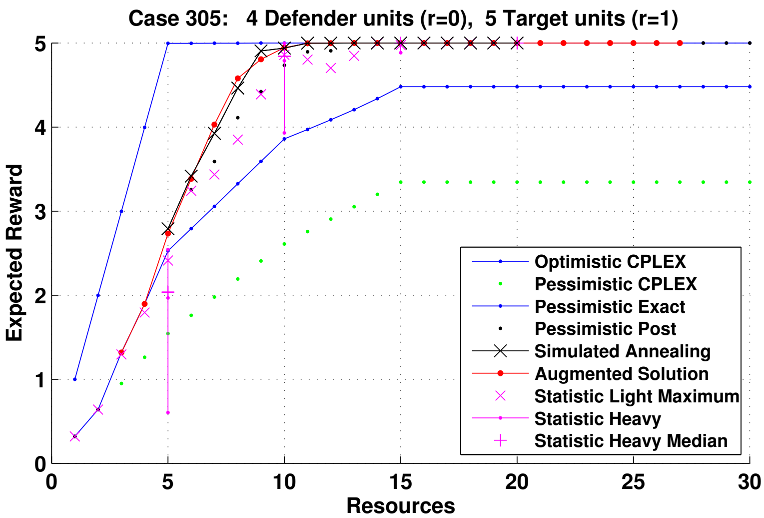


Figure 68: Test case 305. 4 defenders and 5 targets.

12.11.2 Reward setting 2

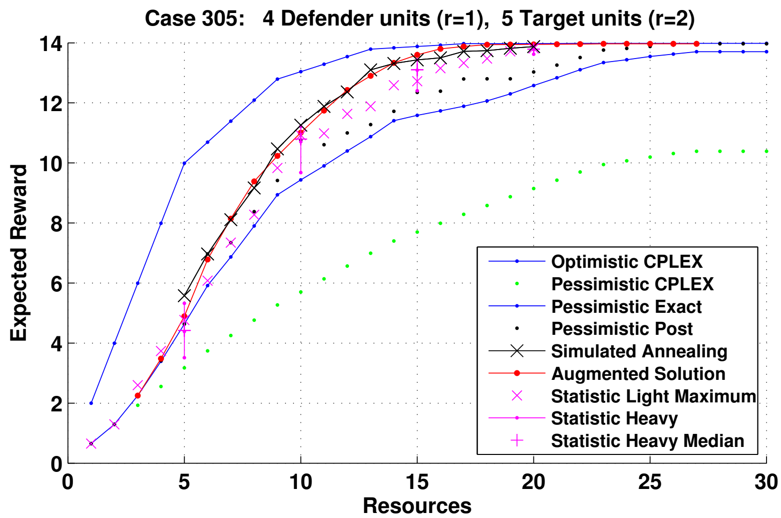


Figure 69: Test case 305. 4 defenders and 5 targets.

12.11.3 Reward setting 3

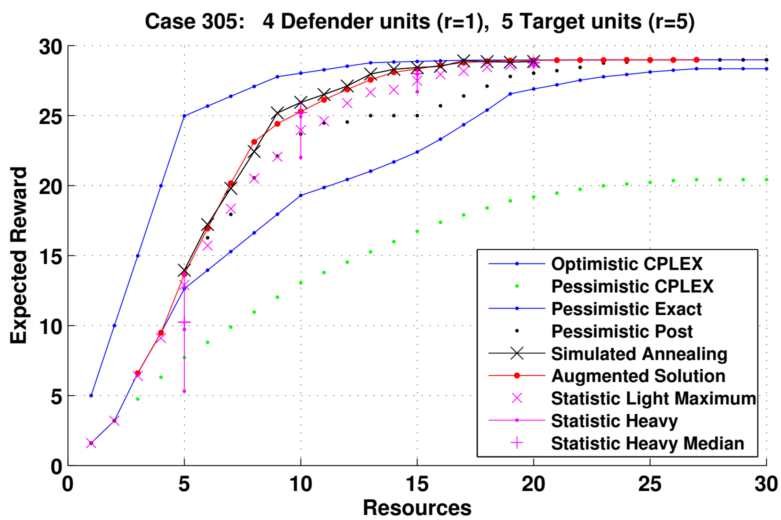


Figure 70: Test case 305. 4 defenders and 5 targets.

12.12 Case 308

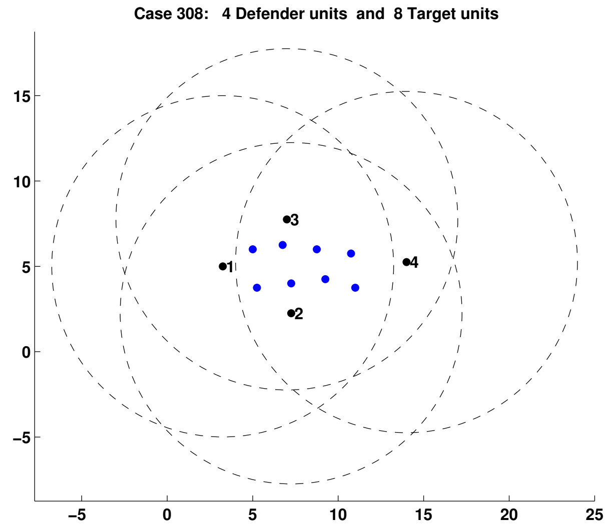


Figure 71: Test case 308. 4 defenders and 8 targets.

12.12.1 Reward setting 1

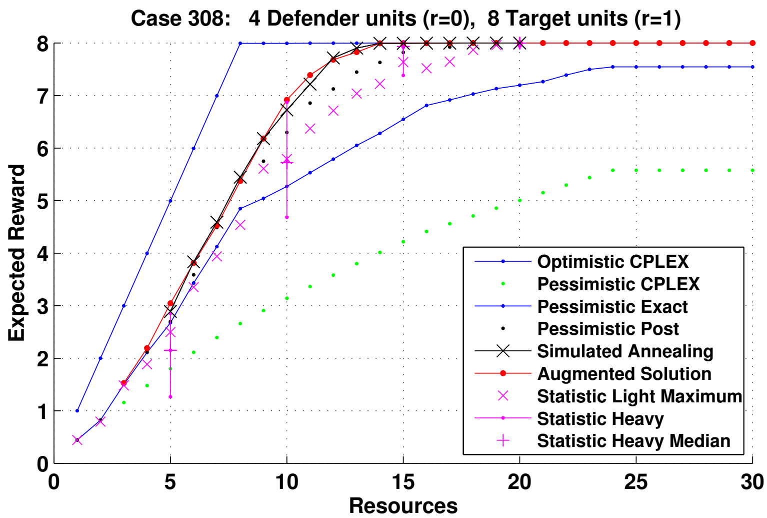


Figure 72: Test case 308. 4 defenders and 8 targets.

12.12.2 Reward setting 2

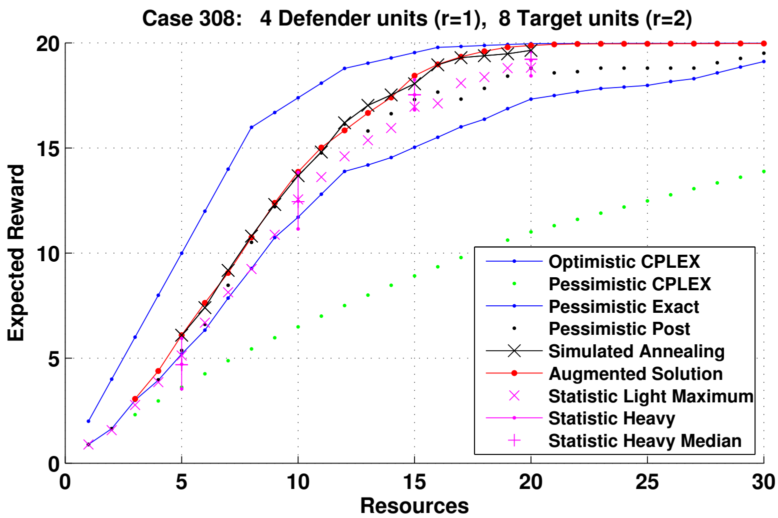


Figure 73: Test case 308. 4 defenders and 8 targets.

12.12.3 Reward setting 3

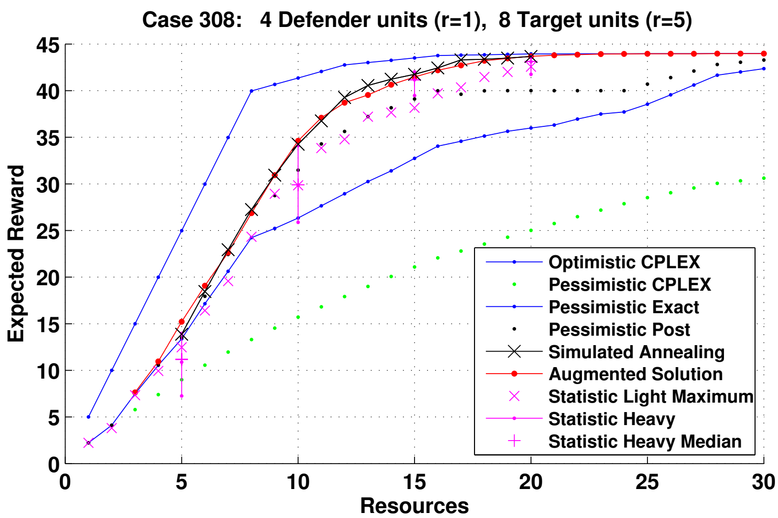


Figure 74: Test case 308. 4 defenders and 8 targets.

12.13 Case 312

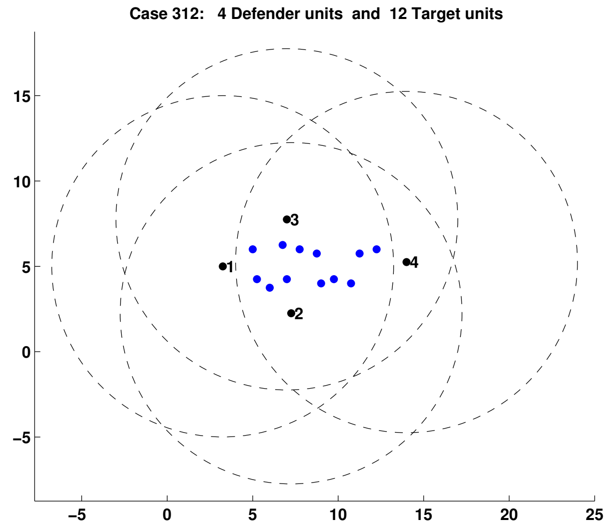


Figure 75: Test case 312. 4 defenders and 12 targets.

12.13.1 Reward setting 1

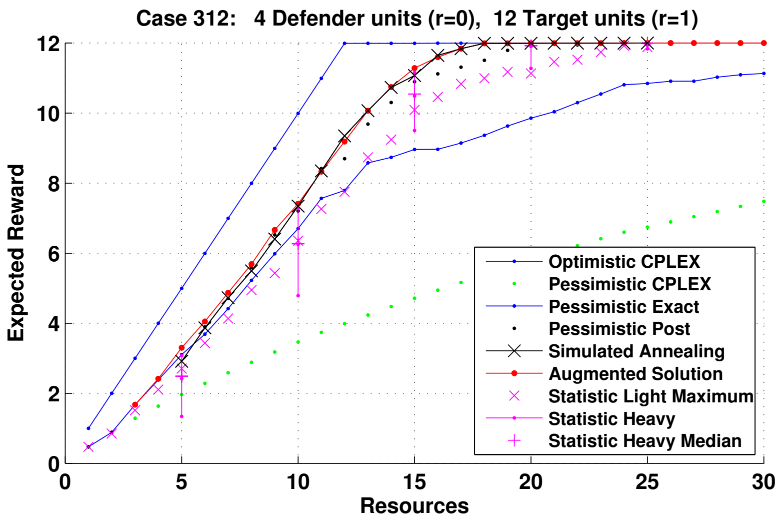


Figure 76: Test case 312. 4 defenders and 12 targets.

12.13.2 Reward setting 2

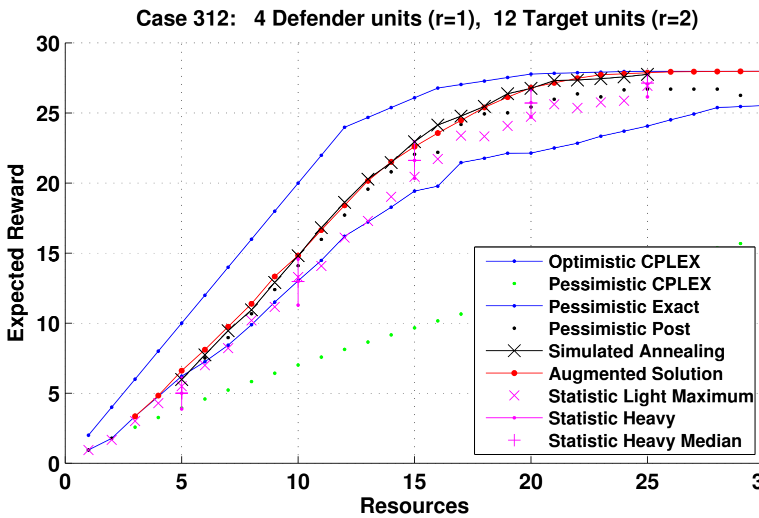


Figure 77: Test case 312. 4 defenders and 12 targets.

12.13.3 Reward setting 3

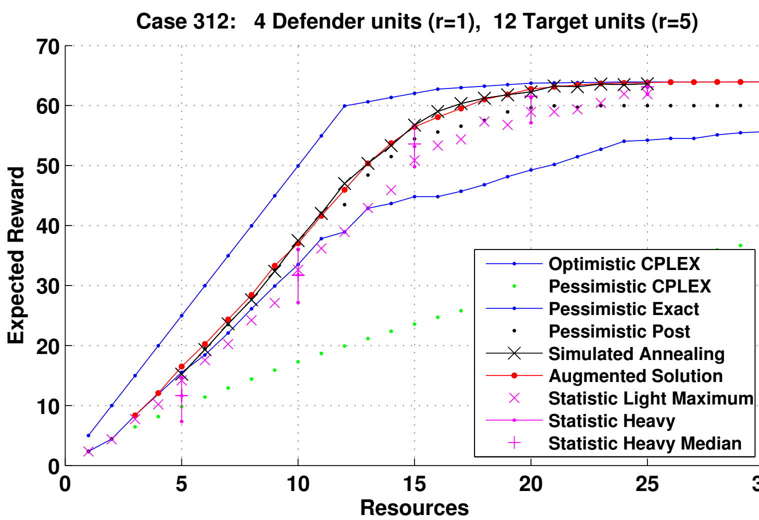


Figure 78: Test case 312. 4 defenders and 12 targets.

12.14 Case 316

Figure 79: Test case 316. 4 defenders and 16 targets.

12.14.1 Reward setting 1

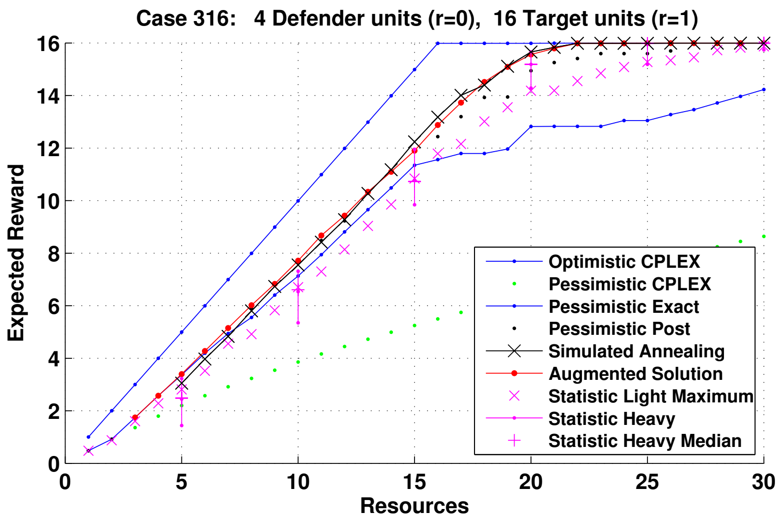


Figure 80: Test case 316. 4 defenders and 16 targets.

12.14.2 Reward setting 2

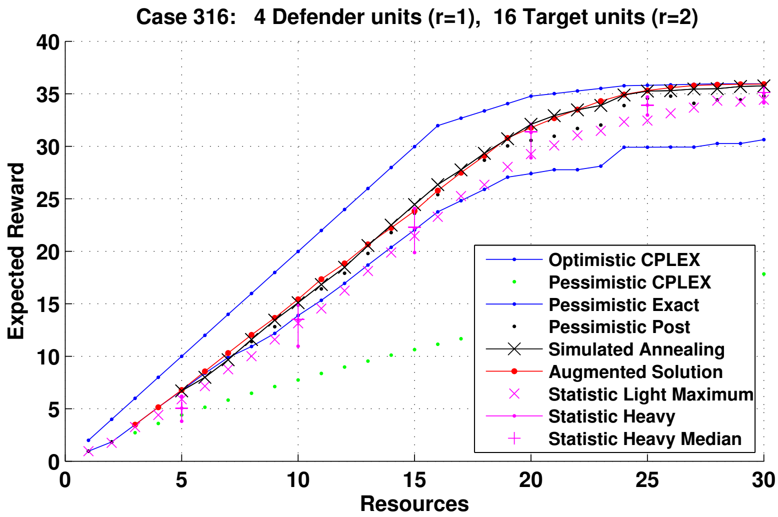


Figure 81: Test case 316. 4 defenders and 16 targets.

12.14.3 Reward setting 3

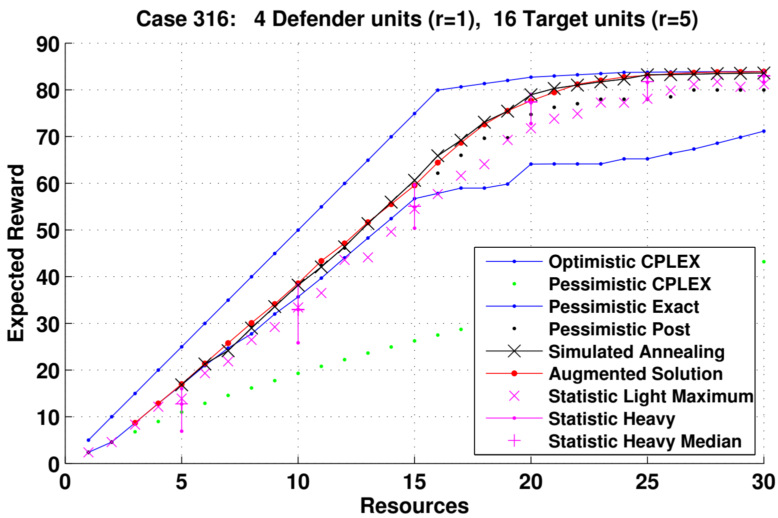


Figure 82: Test case 316. 4 defenders and 16 targets.

12.15 Scenario 3, Solution Times

12.15.1 Time Case 305

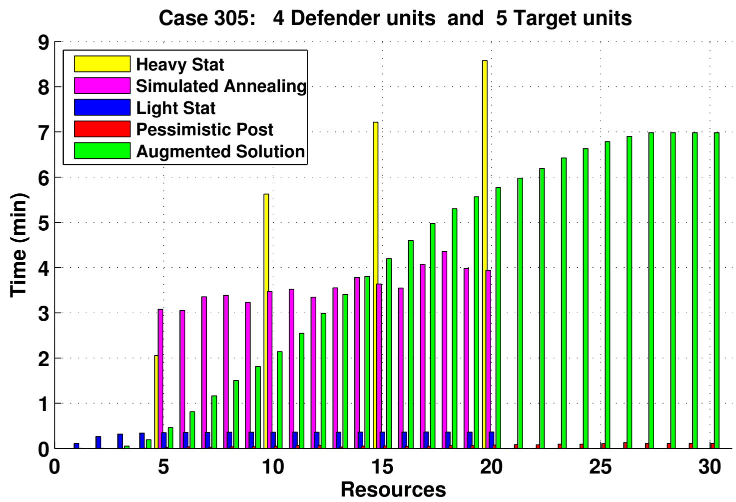


Figure 83: Test case 305. 4 defenders and 5 targets.

12.15.2 Time Case 308

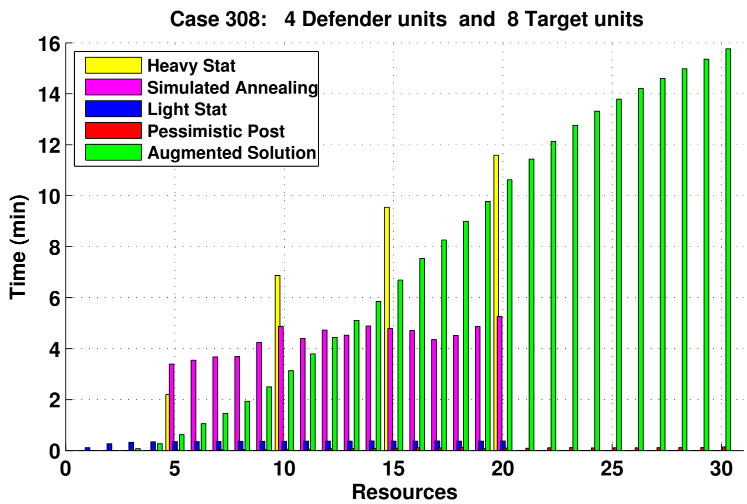


Figure 84: Test case 308. 4 defenders and 8 targets.

12.15.3 Time Case 312

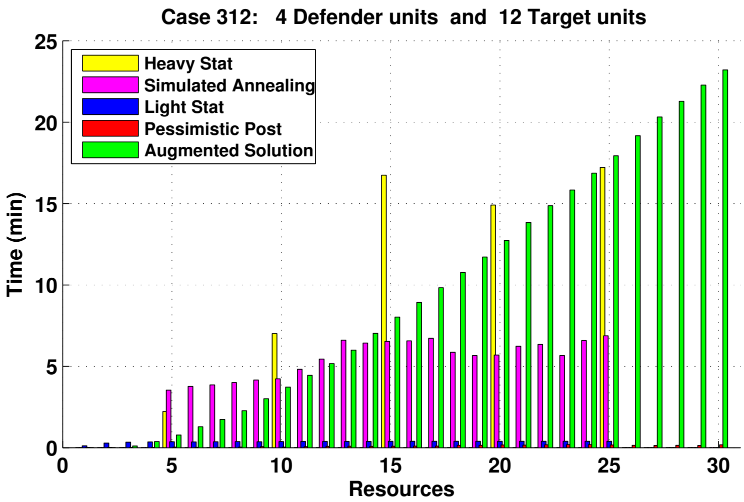


Figure 85: Test case 312. 4 defenders and 12 targets.

12.15.4 Time Case 316

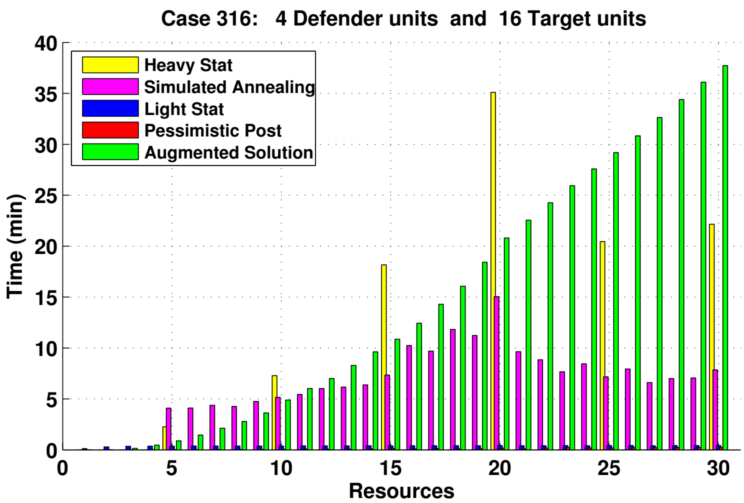


Figure 86: Test case 316. 4 defenders and 16 targets.