A Lagrangian Heuristic for Missile Defence Location Problems

Marc Poelstra

1 Introduction

This paper proposes a different solution method for solving the Missile Defence Location Problem described by Bloemen et al. [1]. They applied two solution approaches to this problem: simulated annealing (a heuristic method) and an exact solution method. Simulated annealing produces results within a short amount of computation time, however simulated annealing does not say how good its results are. The exact solution method uses an integer programming formulation which can be solved to optimality using a standard solver. The downside of this approach is that the computation time is significantly higher. Due to the exact method, it was shown that the results of simulated annealing were actually quite good.

Combining time efficiency and having a confidence level for the solution, we approach the problem this time via a heuristic algorithm based upon Lagrangian relaxation and subgradient optimization. Such an algorithm is described by Beasley [2].

In section 2 we explain the Missile Defence Location Problem. Before presenting our Lagrangian heuristic in section 4, we will show the original model and also explain the used notation in section 3. In section 5 we give a summary about the data used in the model. The results obtained by our heuristic are analysed in section 6. Finally some conclusions are drawn in section 7.

2 Problem description

The Missile Defence Location Problem (MDLP) is the problem of determining the minimum number of interceptor systems and their corresponding locations, such that a given area will be defended against ballistic missiles. Ballistic Missiles (BMs) follow a ballistic trajectory, can have long range and can carry explosive, nuclear, biological or chemical warheads. As such they provide the capability to launch an attack from a distance and enable the projection of power both in a regional and strategic context.

Two of several more stadia of missile defence are active defence and passive defence. Active defence consists of physical interception and neutralisation of ballistic missiles after being launched. The latter can be achieved by launching an interceptor from a Ballistic Missile Defence (BMD) system that will try to intercept and destroy the BM. If the BM has not been destroyed during its flight, then passive defence measures are required like the use of air-raid shelters or gas masks.

The MDLP is illustrated in Figure 1. In this example, which is a copy of the example described in the paper of Bloemen, a rogue country has two ballistic missile launch locations (L1 and L2). It can attack four assets (e.g. major cities: A1, A2, A3 and A4) from these launch locations.

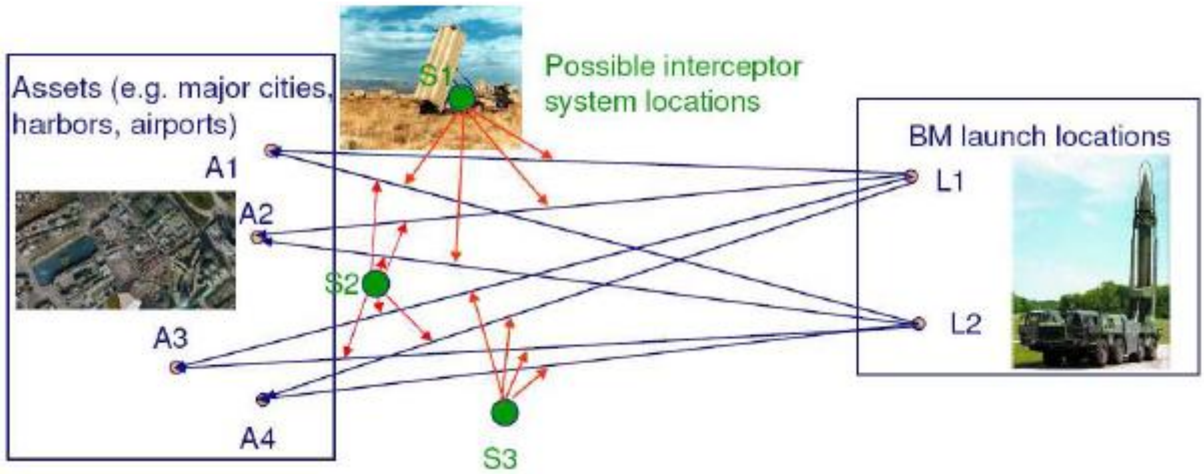


Figure 1: Example

Shot lines are combinations of a single ballistic missile launch location and a single asset. In this example eight shot lines exist. There are three possible locations for BMD interceptor systems: locations S1, S2 and S3. An interceptor system at location S1 can engage the four shot lines to A1 and A2 (represented by red arrows in Figure 1). A system at location S2 can engage the four shot lines to A2 and A3, plus the shot lines from L2 to A1 and from L1 to A4. A system at location S3 can engage the four shot lines to A3 or A4. The minimum number of systems needed to engage all shot lines is two: a system at location S1 and a system at location S3. On the other hand, if the number of available interceptor systems is fixed to one, the selection criterion could for example be to maximise the number of assets to which all shot lines can be engaged. In that case, S2 will be the best location: for two assets all shot lines can be engaged (A2 and A3) and for the other two assets we can engage part of the shot lines (A1 and A4).

Clearly, these solutions are the best if one would be certain that the assumed situation portraits the future accurately. However, in general, deep uncertainty exists about the motives of current and future rogue countries and therefore it would be very short-sighted to blindly solve such a problem based on one possible projection of the future. Consequently, we consider multiple future scenarios, wherein a scenario is defined by the rogue country (including its launch locations), the type of ballistic missile and the assets that have to be defended. Obviously, these scenarios are the result of intelligence gathering and intelligence analysis. The situation portrayed in the example could be regarded as one possible scenario. An additional scenario could for example involve all possible shot lines from another rogue country. Solving MDLP then results in selecting the interceptor systems that meet certain requirements for both scenarios simultaneously.

In order to formally describe the two variants of the MDLP, we first need to define when an asset or area is considered ‘defended’. Given a particular selection of interceptor systems, we say that an asset is defended in a particular scenario s if the proportion of shot lines to that asset that can be engaged by at least one interceptor system, is at least equal to a specific threshold t^s_{asset} . Similarly, an area is defended in a particular scenario s if the proportion of assets that are defended in that scenario, is at least equal to a specific threshold t^s_{area} . Both these thresholds can be different for different scenarios. For example, we could require higher levels for both these thresholds for scenarios in which major cities are to be defended, than for scenarios that only consider minor cities. Note that by selecting S1 and S3 in the example, all shot lines to all assets can be engaged. Thus, these locations assure $t^s_{asset} = t^s_{area} = 1$. We will refer to selecting interceptor systems such that all shot lines to all assets can be engaged, as obtaining the 'maximum defence level'.

3 Original model

In this section we will provide a summarized formal description of the original model. First, we will introduce the notation, and after that the IP-formulation.

3.1 Notation

Let S denote the set of scenarios, L^s the set of launch locations in scenario $s \in S$, A^s the set of assets (cities) in scenario s, and I the set of possible locations for the interceptor systems. A BM shot from a specific $l \in L^s$ to a specific $a \in A^s$ in scenario s, is called a shot line. To keep it simple all possible shot line are feasible. We can now define:

$r^s_{l,a,i} = 1$, if a BM shot from $l \in L^s$ to $a \in A^s$ can be engaged by an interceptor launched from location in $i \in I$ scenario s,

$r^s_{l,a,i} = 0$, otherwise.

For selecting the interceptor site $i \in I$, we define the binary decision variable x_i. When x_i = 1, interceptor site i is chosen and x_i = 0 otherwise. We also define a cost variable k_i, which is needed for the heuristic. Initially we set $k_i = 1\ \forall i \in I$, because all interceptor sites cost the same. For a given solution, a shot line will be referred to as 'covered' if the solution contains at least one interceptor site i that is able to engage the shot line. An asset a is said to be defended, when at most $b_s = \lfloor (1 - t^s_{asset})|L^s| \rfloor$ shot lines to the asset a may remain uncovered. An area is defended when at most $c_s = \lfloor (1 - t^s_{area})|A^s| \rfloor$ assets remain undefended.

Considering these definitions, a shot line is allowed not to be covered and for assets it is allowed not to be defended. In the IP-formulation slack variables need to be introduced for each shot line and each asset. For every shot line from $l \in L^s$ to $a \in A^s$ in scenario $s \in S$, we introduce the slack variable $u^s_{l,a}$. For every asset $a \in A^s$ in scenario $s \in S$, we introduce the slack variable v^s_a. If asset a is undefended in scenario s in a given solution, then the solution has $v^s_a = 1$. When v^s_a = 0, the uncovered shot lines to asset a in scenarios have $u^s_{l,a} = 1$.

We are interested in the variables x_i, $u^s_{l,a}$ and v^s_a, which we can determine with the help of the parameters $r^s_{l,a,i}$, b_s and c_s.

3.2 IP-formulation of the original model (MDLP)

Minimize

$$z^* = \sum_{i \in I} k_i x_i$$

Subject to

$$\sum_{i \in I} x_i r^s_{l,a,i} + u^s_{l,a} + v^s_a \geq \quad 1 \qquad \forall l \in L^s, \forall a \in A^s, \forall s \in S$$

$$\sum_{l\in L^s} u_{l,a}^s \leq b_s(1-v_a^s) \quad \forall a \in A^s, \forall s \in S$$

$$\sum_{a\in A^s} v_a^s \leq \quad c_s \qquad \forall s \in S$$

$$x_i \in \{0,1\} \qquad \forall i \in I$$

$$u_{l,a}^s \in \{0,1\} \qquad \forall l \in L^s, \forall a \in A^s, \forall s \in S$$

$$v_a^s \in \{0,1\} \qquad \forall a \in A^s, \forall s \in S$$

The first constraint is the set covering constraint. It entails that a shot line from $l \in L^s$ to $a \in A^s$ in scenario $s \in S$ satisfies at least one of the following properties:

1) it is covered, in which case $\sum_{i\in I} x_i r_{l,a,i}^s \geq 1$,
2) it is part of the uncovered shot lines to a defended asset, in which case $u_{l,a}^s = 1$ and $v_a^s = 0$, or
3) it is part of any of the shot lines to an undefended asset, in which case $v_a^s = 1$.

For assets that have $v_a^s = 0$, the second constraint ensures that the number of uncovered shot lines to asset a does not exceed the amount b_s, and by definition in that case the asset is defended. The third constraint ensures that the area contains at most c_s undefended assets.

4 Lagrangian heuristic

First in this section, we will show a Lagrangian relaxation of the original model, followed by a description of our heuristic.

4.1 Lagrangian relaxation

To be able to relax the first two constraint of the original model into the objective function, we need to introduce two set of Lagrange multipliers. $\lambda_{l,a,s}$ $(\geq 0;\ \ \forall l \in L^s, \forall a \in A^s, \forall s \in S)$ for the first constraint and $\tau_{a,s}$ $(\geq 0;\ \ \forall a \in A^s, \forall s \in S)$ for the second constraint. Now we define the Lagrangian lower-bound program (LLBP) as follows:

Minimize

$$z = \sum_{i\in I}\left(k_i - \sum_{s\in S}\sum_{a\in A^s}\sum_{l\in L^s} \lambda_{l,a,s}\, r_{l,a,i}^s\right) x_i + \sum_{s\in S}\sum_{a\in A^s}\sum_{l\in L^s}(\tau_{a,s} - \lambda_{l,a,s}) u_{l,a}^s$$
$$- \sum_{s\in S}\sum_{a\in A^s}\left(\sum_{l\in L^s} \lambda_{l,a,s} - \tau_{a,s} b_s\right) v_a^s + \sum_{s\in S}\sum_{a\in A^s}\sum_{l\in L^s} \lambda_{l,a,s} - \sum_{s\in S}\sum_{a\in A^s} \tau_{a,s} b_s$$

Subject to

$$\sum_{a\in A^s} v_a^s \leq c_s \qquad \forall s \in S$$

$$x_i \in \{0,1\} \qquad \forall i \in I$$

$$u_{l,a}^s \in \{0,1\} \qquad \forall l \in L^s, \forall a \in A^s, \forall s \in S$$

$$v_a^s \in \{0,1\} \qquad \forall a \in A^s, \forall s \in S$$

For given Lagrange multipliers, let C_i represent the coefficient of x_i in the objective function above, then it is optimal to take $X_i = 1$ if $C_i \leq 0$ ($X_i = 0$ otherwise). Likewise for $D_{l,a,s}$ as the coefficient of $u_{l,a}^s$ and $U_{l,a}^s = 1$ if $D_{l,a,s} \leq 0$ ($U_{l,a}^s = 0$ otherwise). $E_{a,s}$ is the coefficient of v_a^s and $V_a^s = 1$ for the c_s most negative value of $E_{a,s}$ ($V_a^s = 0$ otherwise). A valid lower bound (Z_{LB}) on the optimal solution to the original SCP is given by:

$$Z_{LB} = \sum_{i \in I} C_i X_i + \sum_{l \in L^s} \sum_{a \in A^s} \sum_{s \in S} D_{l,a,s} U_{l,a}^s - \sum_{a \in A^s} \sum_{s \in S} E_{a,s} V_a^s + \sum_{l \in L^s} \sum_{a \in A^s} \sum_{s \in S} \lambda_{l,a,s} - \sum_{a \in A^s} \sum_{s \in S} \tau_{a,s} b_s$$

4.2 Lagrangian heuristic

Like Beasley, we use subgradient optimization in an attempt to maximize the lower bound obtained from LLBP.

Let Z_{max} represent the maximum lower bound found, Z_{UB} the best feasible solution found and P_k the lower bound when column k $(k = 1, \dots, n)$ is forced to be in the solution.

1) Initialize $Z_{max} = -\infty$, $Z_{UB} = \infty$, $P_k = k_k$ $(k = 1, \dots, n)$, $\lambda_{l,a,s} = 1$ $(\forall l \in L^s, \forall a \in A^s, \forall s \in S)$ and $\tau_{a,s} = 1$ $(\forall a \in A^s, \forall s \in S)$. According to Beasley it does not really matter how these multipliers are initialised, because the quality of the final Z_{max} is relatively insensitive to the initial choice of multipliers.

2) Solve LLBP with the current set of multipliers ($\lambda_{l,a,s}$ and $\tau_{a,s}$) and, as stated above, let the corresponding value be Z_{LB}. Then update Z_{max} by $Z_{max} = \max(Z_{max}, Z_{LB})$.

3) Construct a feasible solution to the original MDLP in the following way:

    1. Defining sets:
       $F = [i \in I \mid X_i = 1]$
       $F_{temp} = empty\ set$

    2. If $\sum_{l \in L^s} \sum_{i \in F} X_i r_{l,a,i}^s < \lfloor (t_{asset}^s) L^s \rfloor$
       then $v_a^s = 1; else\ v_a^s = 0$

    3. Does the current set of v_a^s satisfy $\sum_{a \in A^s} v_a^s \leq c_s \;\; \forall s \in S$?

       NO) $a^* = min_{a \in A^s}\{\sum_{l \in L^s} \sum_{i \in F} X_i r_{l,a,i}^s\}$
       if $\sum_{i \in F} X_i r_{l,a^*,i}^s = 0$
       then $min\{i | x_i r_{l,a^*,i}^s = 1, \forall i \in I\}$ and add this i to F_{temp}
       Add i to F that corresponds to $min\{i | i \in F_{temp}\}$

       Then go to 2., start over with the new set of F and $F_{temp} = empty\ set$ again.

YES) Consider each site $i \in F$ in descending index (i) order and if $F - [i]$ is a feasible solution to the original model (with new sets for $u_{l,a}^{s}$ and v_{a}^{s}) set $F = F - [i]$.

4. Update Z_{UB} by $Z_{UB} = min(Z_{UB}, |F|)$.

4) Stop if $Z_{max} = Z_{UB}$ (this is the optimal solution).

5) Removing interceptor sites from the problem:

$$P_k = max(P_k, Z_{LB} + C_k), \quad \text{if } X_k = 0, \; k = 1, \dots, n$$

$$P_k = max(P_k, Z_{LB}), \quad \text{if } X_k = 1, \; k = 1, \dots, n$$

We can now remove a site by:

$$k_k = \infty, \quad \text{if } P_k > Z_{LB}, \; k = 1, \dots, n$$

since a site can then never be in an improved feasible solution.

6) Calculate the subgradients $G_{\mathrm{l,a,s}}$ and $H_{a,s}$ using the following formulas

$$G_{\mathrm{l,a,s}} = \left(1 - \sum_{i \in I} X_i r_{l,a,i}^{s} - U_{l,a}^{s} - V_a^{s}\right) \quad (\forall l \in L^s, \forall a \in A^s, \forall s \in S)$$

$$H_{a,s} = \left(\sum_{l \in L^s} U_{l,a}^{s} - b_s(1 - V_a^{s})\right) \quad (\forall a \in A^s, \forall s \in S)$$

It is helpful to adjust the subgradients before calculating the step size T, for improving the multipliers, using:

$G_{\mathrm{l,a,s}} = 0, \quad$ if $\lambda_{l,a,s} = 0$ and $G_{\mathrm{l,a,s}} \leq 0$.

$H_{a,s} = 0, \quad$ if $\tau_{a,s} = 0$ and $H_{a,s} \leq 0$.

The reason this can be helpful is that when a subgradient is negative, it implies that the corresponding multiplier needs to be lowered. However if that multiplier is already zero it cannot be lowered any further and therefore that subgradient can be set equal to zero.

7) If $\sum_{l \in L^s} \sum_{a \in A^s} \sum_{s \in S} (G_{\mathrm{l,a,s}})^2 = 0$ the multipliers $\lambda_{l,a,s}$ remain the same.

If $\sum_{a \in A^s} \sum_{s \in S} (H_{\mathrm{a,s}})^2 = 0$ the multipliers $\tau_{a,s}$ do not change.

When both hold, then the heuristic stops, since in this case we cannot define a suitable T (step 8 below).

8) Define a step size T_G and T_H:

$$T_G = f(1.05 Z_{UB} - Z_{LB}) / \left(\sum_{l \in L^s} \sum_{a \in A^s} \sum_{s \in S} \left(G_{l,a,s}\right)^2\right)$$

$$T_H = f(1.05Z_{UB} - Z_{LB}) / \left(\sum_{a \in A^s} \sum_{s \in S} (H_{a,s})^2 \right)$$

where $f = 2$ initially. In both papers of Beasley ([2] and [3]) he says that, if Z_{max} has not improved in the last 30 subgradient iterations with the current value of f then halve f.

9) Stop if $f \leq 0.005$ (this is an arbitrarily chosen stopping criteria).

10) Updating the Lagrange multipliers using:

$$\lambda_{l,a,s} = \max\left(0, \lambda_{l,a,s} + T_G G_{l,a,s}\right), \quad \forall l \in L^s, \forall a \in A^s, \forall s \in S$$

$$\tau_{a,s} = \max\left(0, \tau_{a,s} + T_H H_{a,s}\right), \quad \forall a \in A^s, \forall s \in S$$

After updating the multipliers go to step 2 and resolve the problem.

5 Data

The data gives information about all the possible interceptor sites. We use the same sets as Bloemen. In a short the data sets consist of different scenarios for each site. Scenarios 1 up till 4 are about 64 minor assets (cities) and the others are about 36 major cities. Every scenario consists of 25 possible launch locations. An overview is given in Table 1.

Set	Size
S	8
I	100
L^{1-8}	25
A^{1-4}	64
A^{5-8}	36

Table 1: Overview of the sets and sizes

Without loss of generality we have ordered the interceptor sites in descending order of the number of shot lines that they cover. In essence this means that for any shot line (l, a, s) the interceptor site $min[i|x_i r_{l,a,i}^s = 1, i = 1, \ldots ,100]$ is the "best" site to use in covering shot line (l, a, s), because besides shot line (l, a, s) it covers the highest number of additional shot lines.

The set of thresholds used in the basic problem variant of the paper of Bloemen is given in Table 2.

	Major cities	Minor cities
t_{asset}	0.95	0.9
t_{area}	0.7	0.5

Table 2: Setting of the threshold values

With these threshold values we can now calculated the values for b_s and c_s in Table 3.

	Major cities	Minor cities
b_s	1	2
c_s	10	32

Table 3: Values of b_s and c_s

6 Results

This section contains the results obtained for the data sets described in the previous section. The computations were performed on an Intel(R) Core(TM)2 CPU, 2.13 GHz, 2.00 GB of RAM.

Table 4 shows the results of the initial settings. As is shown, there is a gap between Z_{UB} and Z_{max} . This gives an indication of the quality of the solution (Z_{UB}) obtained. For this indication the following formula is used: $(Z_{UB} - Z_{max})/Z_{max}$. We wrote a model in MATLAB to solve our heuristic. The heuristic always stops in *Step 9*. The computation time in seconds and the number of iterations are also contained in Table 4.

Table 4 contains a column with information about the best found average defence level of the area over the eight scenarios (y^*). It can be determined by

$$1 - y^* = min\left\{\frac{1}{|S|}\sum_{s\in S}\frac{\sum_{a\in A^s} v_a^s}{|A^s|}\right\}$$

The corresponding chosen sites of the best defence level are shown in Table 5.

Initial settings		Number of interceptor sites		Indication of quality	Computation time	Iterations	Average area defence level
$\lambda_{l,a,s}$	$\tau_{a,s}$	Z_{UB}	Z_{max}				
1	1	4	0.4285	8.3348	138 sec	349	90,8%

Table 4: Result of the initial settings

$\lambda_{l,a,s}$	$\tau_{a,s}$	Chosen sites: $(i\|X_i = 1)$			
1	1	5	6	15	26

Table 5: Corresponding sites chosen

The CPLEX method and simulated annealing both showed that the required number of site for these threshold values was 4. So we know that the upper bound gives the right number of sites and this states that we somehow need to improve the maximum lower bound. The lower bound is calculated by mainly using the Lagrange multipliers. Therefore we changed the initial values of the multipliers to see if this matters, even though Beasley states in both papers that this would not have a large effect. Tables 6 and 7 show the results and the corresponding interceptor sites.

Multiplier settings		Number of interceptor sites		Indication of quality	Computation time	Iterations	Average area defence level
$\lambda_{l,a,s}$	$\tau_{a,s}$	Z_{UB}	Z_{max}				
1	2	4	0.3491	10.4550	136 sec	329	91,0%
2	1	4	1.4296	1.7979	207 sec	521	91,0%
1	0	4	1.4031	1.8509	156 sec	389	91,0%
0	1	4	1.3526	1.9572	133 sec	341	91,0%
2	2	4	1.4296	1.7979	217 sec	523	91,0%
0	0	4	1.3560	1.9498	205 sec	542	90,8%

Table 6: Result of the different multiplier settings

$\lambda_{l,a,s}$	$\tau_{a,s}$	Chosen sites: $(i \mid X_i = 1)$			
1	2	5	15	16	25
2	1	5	15	16	25
1	0	5	15	16	25
0	1	5	15	16	25
2	2	5	15	16	25
0	0	5	6	15	26

Table 7: Corresponding sites chosen

As can be seen in Table 6, the settings of $\lambda_{l,a,s} = 2$ & $\tau_{a,s} = 1$ and $\lambda_{l,a,s} = 2$ & $\tau_{a,s} = 2$ provides the best quality of the solution and also has the highest average area defence level. This improvement of the lower bound comes at the price of more or less one minute more computation time.

We find only two different sets of chosen sites. They both have almost the same average area defence level. In Table 8 we give the defence levels for each scenario individual of our two defence level and compare them with the solution of Bloemen.

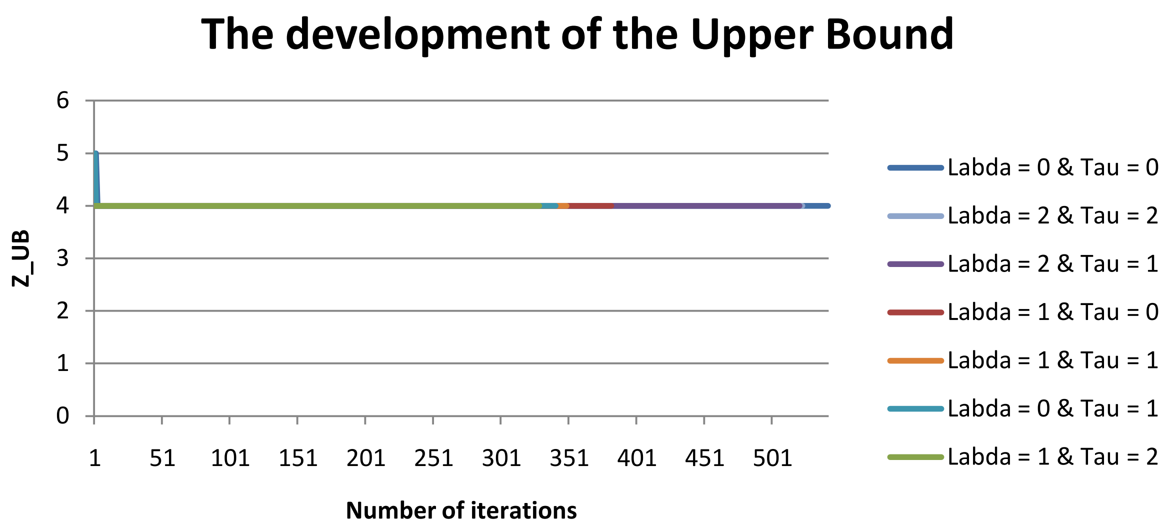
	Poelstra		Bloemen
Scenario	(5,15,16,25)	(5,6,15,26)	
1	98.44	**100.00**	98.44
2	**79.69**	**90.63**	78.13
3	**98.44**	**98.44**	100.00
4	**87.50**	**84.38**	96.88
5	100.00	100.00	100.00
6	**75.00**	**83.33**	77.78
7	100.00	**97.22**	100.00
8	88.89	**72.22**	88.89
Average	90.99	90.78	92.51

Table 8: Our solution versus solution of Bloemen, using four interceptor sites

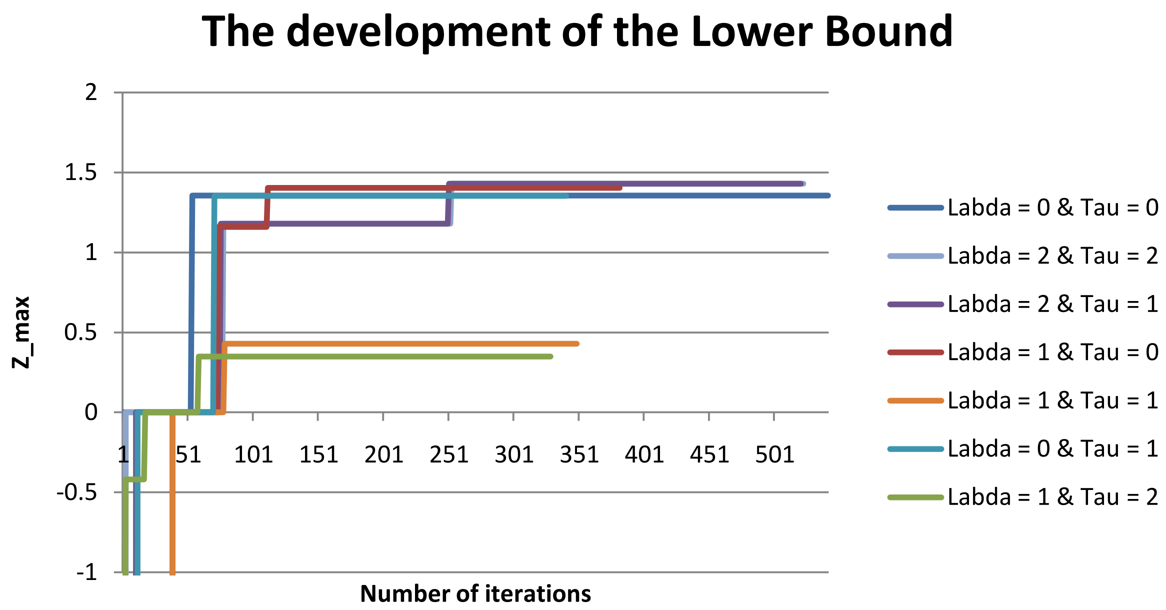
The bold numbers are the defence levels which differ from the solution found by Bloemen. The solution with the chosen sites (5,6,15,26) has in almost all scenarios a different defence level. When we compare the other solution with Bloemen, then we can see that our solution only differs in four scenarios and the average of scenarios 2 and 3 is the same in both solutions.

The development of the upper bound and lower bound is shown in respectively Graph 1 and 2. The legend is ordered descending to the number of iterations. All settings find the upper bound of four interceptor sites, only if the setting contains $\lambda_{l,a,s} = 0$ this upper bound is found after one or two iterations.

The development of the lower bound is somewhat different for each setting. According to Beasley the final outcome would not vary a lot, but as you can in see in Graph 2 and Table 6 this does not hold here. Settings of $\lambda_{l,a,s} = 1$ & $\tau_{a,s} = 1$ and $\lambda_{l,a,s} = 1$ & $\tau_{a,s} = 2$ both give significantly worse values for the best lower bound, than the other settings. For the other settings we find three different values. A thing to be noticed is that when the best lower bound gets better, we need iteration steps to find this value. Settings of $\lambda_{l,a,s} = 0$ & $\tau_{a,s} = 0$ and $\lambda_{l,a,s} = 0$ & $\tau_{a,s} = 1$ find their best lower bound within 75 iteration steps. Due to very minor (so minor that Excel cannot find them) improvements setting $\lambda_{l,a,s} = 0$ & $\tau_{a,s} = 0$ takes so many iterations steps to stop calculating. Settings of $\lambda_{l,a,s} = 2$ & $\tau_{a,s} = 1$ and $\lambda_{l,a,s} = 2$ & $\tau_{a,s} = 2$ find the best lower bound, but both need more than 250 iteration steps to find this value.



Graph 1: The development of the upper bound



Graph 2: The development of the lower bound

In the next part we see what kind of results the heuristic gives if we solve the problem with different threshold levels. Table 9 contains the results for the maximum defence level; this is when all threshold levels are equal to one. We did this also for two different multiplier settings. Two interesting facts are shown in this table. The first fact is that the values of Z_{max} do not differ must, just as Beasley says, and the second fact is that the heuristic finds as upper bound ten interceptor sites. Bloemen found an optimal solution of nine. The reason we found ten interceptor sites is, because of the way we chose the required interceptor sites (*Step 3* of the heuristic).

Multiplier settings		Number of interceptor sites		Indication of quality	Computation time	Iterations	Average area defence level
$\lambda_{l,a,s}$	$\tau_{a,s}$	Z_{UB}	Z_{max}				
1	1	10	3.3336	1.9998	1675 sec	4436	100%
2	1	10	3.1947	2.1302	1827 sec	4456	100%

Table 9: Result of the different multiplier settings for the maximum defence level

In the following experiments, we did not make a distinction between the threshold values of major and minor cities and used the same thresholds for all the scenarios: $t_{asset}^{s} = t_{asset}$ and $t_{area}^{s} = t_{area}$ $\forall s \in S$. Table 10 shows the impact of both threshold values on the upper bound and lower bound. The table also shows the best bounds given multiplier setting $\lambda_{l,a,s} = 1$ & $\tau_{a,s} = 1$ or $\lambda_{l,a,s} = 2$ & $\tau_{a,s} = 1$.

$t_{asset} = 0.9$				$t_{area} = 1$			
t_{area}	Z_{UB}	Z_{max}	Indication of quality	t_{asset}	Z_{UB}	Z_{max}	Indication of quality
0.5	3	1.4716	1.0386	0.9	6	1.3864	3.2775
0.6	3	1.4716	1.0386	0.92	7	1.3696	4.1108
0.7	4	1.3184	2.0340	0.94	7	1.3696	4.1108
0.8	4	1.3184	2.0340	0.96	7	1.3696	4.1108
0.9	5	1.2509	2.9971	0.98	10	3.3336	1.9998
1	6	1.3864	3.2775	1	10	3.3336	1.9998

Table 10: Minimum required sites and best lower bound for different threshold value

Only for the maximum defence level is Z_{max} really different from all other settings, but due to the decreasing of Z_{UB} the setting of $t_{asset} = 0.9$ & $t_{area} = 0.5$ gives the best quality solution.

7 Conclusion

We know from the paper of Bloemen that the best defence level, when using four interceptor sites, is 92.5% and the best level we found with this heuristic is 91%. Maybe if we use a different method to improve the lower bound we can solve this problem to optimality and get as defence level 92.5%. Perhaps with that method we also find an optimal value of nine for the upper bound of the maximum defence level, because the value of the upper bound really depends on the X_i determined when solving the lower bound.

References

[1] A. Bloemen, L. Evers, A.I. Barros, H. Monsuur, and A. Wagelmans. A Robust Approach to the Missile Defence Location Problem. 2010

[2] J.E. Beasley, A Lagrangian Heuristic for Set-Covering Problems, *Naval Research Logistics,* 37 – 151-164, 1990.

[3] J.E. Beasley, Lagrangean Relaxation, 1992