Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems (P. M. Pardalos, Editor), pp. 1-2
©1999 Kluwer Academic Publishers

An Approximate Algorithm For A Weapon Target Assignment Stochastic Program

Robert A. Murphey (Murphey@eglin.af.mil)
Air Force Research Laboratory, Munitions Directorate, 101 W. Eglin Blvd., Ste 144 Eglin AFB, FL 32542-6810

Abstract

The Weapon Target Assignment (WTA) problem considers optimally assigning M weapons to N targets so that the total expected damage to the targets is maximized. If at some time t the numbers and locations of weapons and targets are known with certainty, then a single assignment may be made at time t such that all of the weapons are committed. This formulation is denoted *static* WTA. In its most general form, static WTA is known to be $\mathcal{NP}$-complete. A more difficult problem results when the numbers and locations of targets are not known a priori. Typically, constraints on the weapons maneuverability and range will require a sequence of partial assignments at times $t_1, t_2, \ldots, t_k$, where at each t_i, a subset of the N targets are known with certainty and the remainder are either not known or known only stochastically. This *dynamic* WTA formulation may be modeled as a stochastic program (SP). In general, stochastic programs may be solved by decomposing the SP into a sequence of deterministic problems. However, for dynamic WTA, the integrality and non-linearity of the problem makes it difficult to obtain a solution by decomposition. This paper studies an algorithm that finds an optimal solution for a similar problem which is close to optimal for the original problem but is amenable to on-line execution.

Keywords: Weapon target assignment, Stochastic programming.

1 Introduction

The Weapon Target Assignment (WTA) problem considers the issue of optimally assigning M weapons to N targets so as to maximize the total expected damage to the targets. The WTA problem was first described by Flood [5] and since that time has been primarily developed for application to battlefield strategy planning in ballistic missile defenses, most notably, the Strategic Defense Initiative (SDI) [6].

Prior to the SDI research, WTA was assumed to be *static*; that is, the assignment is made at a fixed point in time and M and N are completely known. In the SDI research of Athens, Hosein, and Walton [1] a *dynamic* model was introduced whereby assignments can be made at several points or epochs in time. An inherent difficulty in dynamic WTA models involves accounting for the uncertainty of future events when making an assignment at the current time. In other words, the uncertainty of future events in concert with the assignments at the current epoch constrain assignability for future epochs. In most WTA models, both static and dynamic, the destruction of a target given that a weapon has been assigned to it is considered to be stochastic. In the dynamic WTA model of [1], this uncertainty in the outcome of the current assignment is taken into consideration when making assignments for future epochs. Further, the authors allow for damage assessment observations of the targets after each engagement to determine if the target was actually destroyed; information that is used in making subsequent assignments.

While this assumption may be reasonable for strategic scenarios where the targets are cities and easily observed, the scenario studied in this paper involves more conventional targets which are assumed to be much smaller, such as buildings and vehicles. Consequently, it is assumed that damage assessments are not possible, so that ensuring, with unity probability, the destruction of each target engaged is also not possible and not the objective. The scenario studied in this paper instead assumes that the target set evolves over time. As a consequence, the total number of targets is unknown and the number of targets that may be assigned to is a stochastic function of time.

This paper begins with an explanation of the static WTA in Section 2. An algorithm is presented for the case when the weapons are all equivalent (uniform). This algorithm will be modified for use in our dynamic problem. In Section 3 a dynamic model is introduced that defines the number of targets to be a function of time. Section 4 draws on this model to define a 2-stage non-linear integer stochastic programming formulation. The general method of solution by decomposition of stochastic programs is described in Section 5. The method of decomposition relies on the ability to successively constrain the stage 1 problem with supporting hyperplanes that represent the stage 2 problem. Since convexity is required for construction of the hyperplanes, a relaxation problem is described that can be solved by decomposition and provides a lower bound on the non-linear integer problem. An algorithm implementing this decomposition is described in Section 6 and is shown to provide

solutions that are pertinent to the original problem.

2 Static Assignment Models

Consider that there are M weapons and N independent targets. If the weapons are all assumed to be *uniform* that is, they are all the same then we may define the decision variable $x_j,\ j = 1, 2, \ldots, N$ as

$$x_j = \text{number of weapons assigned to target } j.$$

Given that a single weapon engages target j, that is $x_j = 1$, the outcome is random and assumes a Bernoulli distribution:

$$P(\text{target } j \text{ is destroyed by weapon}) = p_j$$
$$P(\text{target } j \text{ is not destroyed by weapon}) = 1 - p_j \triangleq q_j$$

Assume that each weapon engagement is independent of every other engagement. Then the outcomes of the engagements are independent Bernoulli distributed.

Let each target be assigned a positive real number V_j to indicate preference between targets. Assume that targets may be partitioned into classes and each class has a unique value to the decision maker. Let the set of all class values be contained in $\mathbb{V}$ such that $\mathbb{V}$ is of cardinality K. Our objective is to maximize the expected damage to the targets which is equivalent to minimizing the expected target value. The resulting integer programming formulation is:

SWTA

$$\begin{aligned}
&\text{minimize} && \sum_{j=1}^{N} V_j q_j^{x_j} \\
&\text{subject to} \\
&&& \sum_{j=1}^{N} x_j = M, \\
&&& V_j \in \mathbb{V} \in \mathbb{R}_+^K,\ \ j = 1, 2, \ldots, N \\
&&& x \in \mathbb{Z}^N
\end{aligned}$$

Throughout this paper $\mathbb{R}$ denotes the real number space, and $\mathbb{Z}$ and $\mathbb{B}$ denote the integer and binary subspaces respectively. The equality constraint ensures that all of the weapons are used in the assignment. Notice that nothing prevents all weapons from being assigned to a single target. This problem formulation was first suggested by Manne [9].

Step 0. For each $j = 1, \ldots N$, let $x_j = 0$, and denote probability of survival of target j by $S_j = V_j q_j^{x_j}$. Initialize weapon index $i = 1$.

While $i \leq M$ DO

Step 1. Find target k for which munition i has greatest effect:

$$k = \arg\max_j \{S_j(1 - q_j)\}.$$

Step 2. Add munition i to target k: $x_k = x_k + 1$ and revise probability of survival of target k: $S_k = S_k q_k$.

$i = i + 1$.

Figure 1: Minimum Marginal Return (MMR)

den Broeder *et al* [4] presented an algorithm for solving SWTA optimally in $O(N + M \log(N))$ time termed the minimum marginal return (MMR). Essentially, MMR assigns weapons one at a time to the target which realizes the greatest decrease in value, where target value is defined to be $q_j^{x_j}$. den Broeder showed that this greedy strategy is optimal for the uniform weapon assumption. The algorithm is shown in Figure 1.

3 The Dynamic Assignment Model

In the static WTA formulations, the decision maker knows at one instant in time the total number of targets N and their locations. Consider another WTA problem, where at some time t, only a subset of the total targets are known to the decision maker. Let the number of targets known at time t be denoted by $n(t)$ and $n(t) \leq N$ for any t. As time progresses, additional targets are discovered, hence $n(t)$ is nondecreasing with time. Since N is unknown, an assignment of weapons can only be made to the $n(t)$ weapons that are known or else reserved for targets that are expected in the future. If the targets are ranked according to their value, as in the static WTA problems, then it is possible that at time t only low value targets have been found, in which case it is desirable to wait before making an assignment. If, after waiting time τ, all of the targets are discovered, that is $n(\tau) = N$, then a static

WTA problem results. Assume however that there is a cost associated with waiting to make an assignment. This cost is based on two facts. First, quite often the weapons themselves are responsible for detecting and tracking targets. A weapon has finite fuel so once it is deployed, it has a finite time to detect and engage targets. If a weapon passes over a mobile target, the target may pass out of the weapon's field of view, meaning that the weapon can no longer track the target. As a result, it may be difficult to find the target again before the weapon has exhausted its fuel. In fact, the target may move out of a weapon's engagement range altogether. This effect will be termed *missed reacquisition*. Second, weapons are targets themselves and their probability of survival decreases with time, an effect termed *attrition*.

Define T to be the latest possible time to assign a weapon to a target. The *dynamic WTA problem* then attempts to find an assignment $x(t)$, which minimizes the value of targets over all $t \in [0, T]$. The cost function monotonically increases with time to account for the cost of waiting and, as in the Static WTA problems, is weighted by target values $V_j \in \mathbb{V}$. Assume that $\mathbb{V}$ does not change over time.

To make the problem easier to discuss, time will be discretized. A deterministic formulation for the dynamic WTA problem with uniform weapon assumption is

DWTA

$$\text{minimize} \quad \sum_{i=1}^{T} c(i) \sum_{j=1}^{n(i)} V_j q_j^{x_j(i)}$$

subject to

$$\sum_{i=1}^{T} \sum_{j=1}^{n(i)} x_j(i) = M,$$

$$V_j \in \mathbb{V} \in \mathbb{R}_+^K, j = 1, 2, \ldots, n(T)$$

$$x(i) \in \mathbb{Z}^{n(i)} i = 1, 2, \ldots, T$$

where i is the discrete time index and $c(i)$ is a nondecreasing function that represents the cost of waiting. To model attrition, the number of munitions M may be a function of i. If so, $c(i)$ need only represent the cost of missed reacquisitions, which can be formulated as a linear scalar function of i.

4 The Stochastic Programming Formulation

On-line implementation of the dynamic formulation of DWTA is unrealistic since it determines the optimal decision after the final time T has passed, yet assignments for any time interval $i < T$ must be made during that interval. A realistic model must account for the fact that at interval I, $n(i)$ are stochastic for $i > I$.

There are two approaches for a stochastic model; two-stage and multistage.

Suppose we have some stochastic intuition regarding the locations of targets in time. If, in addition, the trajectories of the munitions are known, then a random vector $\xi_i \in \Xi \in \mathbb{Z}^{K\times s}$ may be defined to denote the number of targets at stage i in each class $k,\ k = 1,\ldots,K$. However, more often than not, it is difficult to determine approximately when an unknown target might appear. Conversely, it is fairly straightforward to guess how many targets of each class remain undetected within a finite geographic region. This is possible by making a guess at the number of targets in each class and then updating that number recursively based on search results and the amount of territory covered. In such a case, the random vector $\xi \in \Xi$ denotes the number of targets in each class $k,\ k = 1,\ldots,K$ that have yet to be detected. In a stochastic model, the former approach is referred to as a *multistage stochastic program* and the later as a *two-stage stochastic program*. For the reasons mentioned, the two-stage formulation will be pursued.

Suppose we are given the probabilities of survival and the target values for each target class. Then for any instance of random vector ξ, the second stage values of target value V_2, probability of survival q_2 and of course the total number of yet to be detected (stage 2) targets n_2 are easily determined. The 2-stage stochastic programming formulation is as follows:

SP

$$Z_1(x) = \min_x\ f_1(x) + E_{\xi\in\Xi}[Z_2(x,\xi^j)] \tag{1}$$

subject to

$$\sum_{i=1}^{n_1} x_i \le M, \tag{2}$$

$$x \le b \tag{3}$$

$$x \in \mathbb{Z}^{n_1} \tag{4}$$

where $f_1(x)$ is the stage one cost function of the first stage assignment x and is integer-convex, that is, the continuous relaxation of $f_1(x)$ is convex. $E_{\xi\in\Xi}[Z_2(x,\xi^j)]$ is the expectation operator with respect to ξ. The constraint in (2) is the resource capacity constraint as it appears in the static problem only now with inequality to permit assignment of resources in the second stage. The constraint in (3) is used to limit the number of weapons assigned to any single target, preventing the assignment of large numbers of weapons to high value targets. The right hand side of (3) is determined by setting a threshold Φ_j on the reduced value of each target as in

$$q_i^{x_i} \le \Phi_j$$

which can always be written in the form of (3) by applying a logarithm. $Z_2(x,\xi^j)$ is the solution to a second stage problem that clearly shows its dependence on the

stochastic parameter $\xi \in \Xi$ and the first stage decision x as captured in the following program:

S2P

$$Z_2(x, \xi^j) = \min_y \ f_2^j(y) \tag{5}$$

subject to

$$\sum_{i=1}^{n_1} x_i + \sum_{i=1}^{n_2(\xi^j)} y_i = M, \tag{6}$$

$$y \in \mathbb{Z}^{n_2} \tag{7}$$

where $f_2^j(y)$ is the stage two cost function of the number of weapons y assigned in the second stage. Since y depends on the outcome of ξ, $f_2^j(y)$ depends on ξ and furthermore is integer-convex. Specifically, $f_1(x)$ and $f_2^j(y)$ are the following:

$$f_1(x) = \sum_{i=1}^{n_1} V_1^i (q_1^i)^{x_i} \tag{8}$$

$$f_2^j(y) = \sum_{i=1}^{n_2(\xi)} V_2^i(\xi) q_2^i(\xi)^{y_i} \tag{9}$$

Due to the discrete support Ξ of ξ which naturally occurs for discrete targets, the formulation in SP may be replaced with the following so called *deterministic equivalent* program:

DESP

$$Z_1(x) = \min_x \ f_1(x) + \sum_{j=1}^{s} p^j Z_2(x, \xi^j) \tag{10}$$

subject to

$$\sum_{i=1}^{n_1} x_i \leq M, \tag{11}$$

$$x \leq b \tag{12}$$

$$x \in \mathbb{Z}^{n_1} \tag{13}$$

where s is the total number of scenarios (outcomes) for the random vector ξ.

5 Solution By Decomposition

Let us examine the stochastic program DESP/S2P more closely. In each stage this program has an integer-convex objective and integer affine constraints. Integrality already makes this a difficult problem to approach in the classic sense due to our inability to obtain supporting hyperplanes for the stage 2 problem (see for example [3], [7], [10]) and the overall complexity of solving non-linear stochastic programs [2]. The nature of the dependency on Ξ is also of particular interest. Consider the following "conventional" stochastic program with *dual block angular* form

$$\begin{aligned} &\min f_1(x) + \sum_{j=1}^{s} p^j f_2^j(y) \\ &\text{subject to} \\ &\qquad Ax \leq b \\ &\qquad T(\xi)x + Wy \leq h(\xi) \\ &\qquad x \in X,\ y \in Y \end{aligned}$$

Unlike the conventional stochastic programs where a stochastic technology matrix $T(\xi)$ links the stage 1 and stage 2 variables in the stage two constraints, the stochastic dependency in our program is primarily found in the second stage objective, not the constraint.

The solution approach presented is based on the *decomposition method* that is widely used for solving stochastic programs. Decomposition methods decouple the stage 1 and stage 2 problems by first solving a variant of the stage 1 problem, often called the *current problem, e g.*

CP

$$\begin{aligned} &\min_{x,\theta} \quad f_1(x) + \theta \\ &\text{subject to} \\ &\qquad Ax \leq b, \\ &\qquad x \in X \end{aligned}$$

with a scalar variable θ taking the place of the stage 2 solution so that

$$\theta \leq \sum_{j=1}^{s} p^j Z_2(x, \xi^j).$$

The stage 2 problem is solved for each $\xi^j, j = 1, \ldots s$. If the current problem solution is infeasible for any of the stage 2 problem constraints, then a valid inequality

or separating hyperplane (*feasibility cut*) is added to CP which effectively removes the infeasible solution while preserving all feasible solutions. The current problem is then solved again and the stage 2 problems solved again using the latest iterate until all the stage 2 solutions are feasible. Information from the stage 2 problems is then used to develop a constraint which bounds the first stage resource to θ (*optimality cut*). This constraint is added to CP which is solved once again. Iterations between the CP and stage 2 problems continue until the new solution is ϵ-close to the latest optimality cut.

Now consider the stochastic program DESP/S2P. The stage 2 constraint in (6) is a simplex constraint without a stochastic technology matrix. Furthermore, the right hand side is identical to the right hand side of the stage 1 simplex constraint (11). Consequently, any solution to the current problem will be feasible for the second stage constraints. Clearly, feasibility cuts are not necessary for decomposing DESP.

Optimality cuts are very useful in the decomposition of DESP as they represent the value of second stage assignments in the first stage constraints. Some basic results will now be presented.

Define the convex relaxation of S2P:

CS2P

$$Z_2(x, \xi^j) = \min_y \ f_2^j(y) \tag{14}$$

subject to

$$\sum_{i=1}^{n_1} x_i + \sum_{i=1}^{n_2(\xi^j)} y_i \leq M, \tag{15}$$

$$y \geq 0, y \in \mathbb{R}^{n_2} \tag{16}$$

where the inequality in (15) replaces the equality of (6) without loss of generality and is done to ensure that a strictly interior solution exists, hence the Slater condition is satisfied. The reasoning for this will become clear in Lemma 2.

We will now explore some properties of CS2P.

Lemma 1. *Consider the CS2P program. $Z_2(x, \xi)$ is convex in ξ.*

Proof. Clearly $\sum_{i=1}^{n_1} x_i + \sum_{i=1}^{n_2(\xi^j)} y_i = M$ is affine so it remains to show that the objective is convex.

Since $f_2(y)$ is the sum of convex functions it is in turn convex. Define two solutions y_1 and y_2 which correspond to random vectors ξ^1 and ξ^2 respectively. Further define a scalar λ, such that $0 \leq \lambda \leq 1$. Then

$$\begin{aligned} Z_2(x, \lambda\xi^1 + (1-\lambda)\xi^2) &\leq f_2(\lambda y^1 + (1-\lambda)y^2) \\ &\leq \lambda f_2(y^1) + (1-\lambda) f_2(y^2) \\ &\leq \lambda Z_2(x, \xi^1) + (1-\lambda) Z_2(x, \xi^2) \end{aligned}$$

□

The following lemma is based on a proof by Birge [2].

Lemma 2. *A lower bound on DESP solution Z_1, denoted by $\bar{Z}_1$ may be obtained by solving the following program:*

$$
\begin{aligned}
\bar{Z}_1(x) = min_{x,\theta}\ & f_1(x) + \theta \\
& \textit{subject to} \\
& \sum_{i=1}^{n_1} x_i \leq M, \\
& x \leq b \\
& A^l x + d^l \leq \theta \qquad\qquad l = 1, \ldots, L \\
& x \geq 0, \quad x \in \mathbb{Z}^{n_1}
\end{aligned}
$$

Proof. Clearly the program above without the constraints

$$A^l x + d^l \leq \theta \quad l = 1, \ldots, L$$

has no lower bound on θ thus yielding a lower bound on Z_1. Of course a lower bound on θ would yield a tighter bound on Z_1. Consider for some scenario ξ^j the program CS2P and its dual:

DCS2P

$\max_{\lambda^j}\ \phi(\lambda^j)$

with

$$\phi(\lambda^j) = \min_{y \geq 0}\ f_2^j(y) + \lambda^j \left(\sum_{i=1}^{n_1} x_i + \sum_{i=1}^{n_2(\xi^j)} y_i - M \right)$$

where λ^j is the dual variable associated with scenario j. By the convexity result of Lemma 1 and *strong duality* there is no duality gap, that is

$$\max_{\lambda^j} \phi(\lambda^j) = Z_2(x, \xi^j)$$

Suppose an optimal solution $Z_2(x, \xi^j)$ to CS2P and hence DCS2P is found. Since, as stated previously, the Slater condition is satisfied by CS2P, then we may construct a lower supporting hyperplane at $Z_2(x, \xi^j)$ which is

$$Z_2(x, \xi^j) \geq f_2^j(y) + \lambda^j \Big(\sum_{i=1}^{n_1} x_i + \sum_{i=1}^{n_2(\xi^j)} y_i - M \Big)$$

which must hold for all y, λ.

Furthermore, by observing DESP and the Lemma 2 program we conclude that

$$\sum_{j=1}^{s} p^j Z_2(x, \xi^j) = \theta.$$

Combining these results, we obtain

$$\sum_{j=1}^{s} p^j \Big[f_2^j(y) + \lambda^j \Big(\sum_{i=1}^{n_1} x_i + \sum_{i=1}^{n_2(\xi^j)} y_i - M \Big) \Big] \leq \theta.$$

Rearranging terms yields

$$\sum_{j=1}^{s} p^j \lambda^j e^T x + \sum_{j=1}^{s} p^j \Big[f_2^j(y) + \lambda^j \Big(\sum_{i=1}^{n_2(\xi^j)} y_i - M \Big) \Big] \leq \theta$$

where e is a vector of ones of size comparable to x. Equivalently

$$Ax + b \leq \theta.$$

□

The constraint $Ax + b \leq \theta$ of Lemma 2 is called an *optimality cut* for DESP.

6 The Main Algorithm

The algorithm presented here provides an approximate solution for DESP/S2P by solving a similar problem to ϵ-optimality. The problem solved is simply DESP/CS2P that is, $y \in \mathbb{R}^{n_2(\xi)}$ replaces the integrality condition of the second stage decision.

We present the algorithm for DESP/CS2P then show that it converges in a finite number of steps to an approximate solution for DESP/S2P.

`Decomposition for Dynamic WTA Stochastic Program (DDWSP).`

Step 0. Set $\theta = 0$, $L = 0$, $k = 1$.

Step 1. Solve the current problem

Current problem

$$\bar{Z}_1(x) = \min_{x,\theta} f_1(x) + \theta \tag{17}$$

$$\text{subject to} \sum_{i=1}^{n_1} x_i \leq M, \tag{18}$$

$$x \leq b \tag{19}$$

$$A^k x + d^k \leq \theta \quad k = 1, \ldots, L \tag{20}$$

$$x \geq 0, \quad x \in \mathbb{Z}^{n_1} \tag{21}$$

If infeasible, STOP: problem is infeasible.
Else denote solution by (x^k, θ^k).

Step 2. For $j = 1, \ldots, s$ solve second stage primal:

$$Z_2(x^k, \xi^j) = \min_y \; f_2^j(y)$$

$$\text{subject to} \sum_{i=1}^{n_1} x_i^k + \sum_{i=1}^{n_2(\xi^j)} y_i = M,$$

$$y \geq 0, \quad y \in \mathbb{R}^{n_2}$$

and its dual:

$$\max_\lambda \min_{y \geq 0} \; f_2^j(y) + \lambda^j \Big(\sum_{i=1}^{n_1} x_i^k + \sum_{i=1}^{n_2(\xi^j)} y_i - M \Big)$$

Let the solutions to these problems be (y^{jk}, λ^{jk}).

Step 3. Define

$$A = \sum_{j=1}^{s} p^j \lambda^{jk}$$

$$d = \sum_{j=1}^{s} p^j \big(f_2^j(y^k) + \lambda^{jk} (e^T y^{jk} - M) \big)$$

where e is a vector of ones of comparable size to y^{jk}. If $\theta^k < Ae^T x^k + d + \epsilon$, for $\epsilon > 0$ where here e is a vector of ones of comparable size to x, then add cut to current problem: $A^k e^T x + d^k \leq \theta$. Let $k = k+1$. Return to step 1.

Else STOP: (x^k, θ^k) is ϵ-optimal solution for DESP/CS2P.

Proposition 1. *DDWSP terminates in a finite number of iterations with a solution that is ϵ-optimal for DESP/CS2P.*

Proof. We first show that at each iteration for which the algorithm does not terminate, the lower bound $\bar{Z}_1$ increases. This is true since if at iteration k the solution (x^k, θ^k) satisfies $\theta^k < Ae^T x^k + d + \epsilon$ for $\epsilon > 0$ then the constraint $A^k e^T x + d^k \leq \theta$ is added to the current problem and (x^k, θ^k) is no longer feasible. Hence the solution at step $k+1$ must satisfy this new constraint and $\bar{Z_1}^{k+1} > \bar{Z_1}^k$.

Now to prove finite termination we need only observe that if at step k the solution (x^k, θ^k) satisfies $\theta^k \geq A^l e^T x^l + d^l + \epsilon$ for $l = 1, \ldots L$, then the algorithm terminates with an objective value ϵ-close to the lower bound $\bar{Z_1}^k$. If, on the other hand, the solution satisfies $\theta^k < A^k e^T x^k + d^k + \epsilon$, then as stated above, a constraint is added and the lower bound increases and so eventually must terminate. □

Proposition 2. *DDWSP yields a lower bound solution for DESP/S2P.*

Proof. Denote the solution to DESP/S2P, assuming it can be obtained, by $\hat{Z}_1$. In the solution to DESP/CS2P, the second stage decision y represents assignment to conjectured targets, consequently the actual value of y, including the fact that it may be non-integral, is unimportant; only the effect it has on the value of x is important. At step k this effect is conveyed via θ^k as the value of

$$\theta^k = \sum_{j=1}^{s} p^j Z_2(x^k, \xi^j)$$

which is obtained by s applications of CS2P. Since CS2P is a convex relaxation of S2P, any solution of the former is a lower bound on the later. Consequently, the exact solution Z_1 of DESP/CS2P is a lower bound on the solution $\hat{Z}_1$ of DESP/S2P. From Proposition 1 we know that Z_1 is in turn ϵ-close to the solution obtained by DDWSP hence, $\bar{Z_1}^k \leq Z_1 \leq \hat{Z}_1$. □

At this point the reader may wonder why only an approximate solution for DESP/S2P is obtained. This question naturally arises since the second stage problems are seen to be equivalent to SWTA which is easily solved to integral optimality by MMR. However, the construction of the optimality cut in DDWSP is only allowed due to the convexity proved in Lemma 1 which is a consequence of the continuous relaxation of S2P. In graphical terms, an integral solution optimal solution to S2P that is not optimal to CS2P cannot be guaranteed to support a hyperplane such that no feasible solutions are cut [3]. However, MMR could certainly be used to provide an excellent starting feasible point for CS2P.

6.1 Algorithm For The Current Problem

Solving the convex programming problems in step 2 of DDWSP is straightforward. Consequently we now focus on an algorithm for the current problem in step 1 where non-Boolean integrality and non-linearity are the complicating issues. As discussed in relation to the SWTA problem, the MMR algorithm provides an optimal solution to a similar integer program very quickly. Whereas the SWTA constrains the sum of the decision variables to equality with M, the current problem has inequality constraints on both the sum of the decisions as in equations (18) and (20) and each decision itself as in equation (19). MMR is easily adapted to handle constraints of type (19) by simply moving to the target which realizes the next greatest decrease in value if the target with greatest decrease in value has satisfied its limit b. A natural approach for handling the constraints of type (18) and (20) is to iteratively apply MMR to the current problem where the constraints (18) and (20) are replaced with the constraint

$$\sum_{j=1}^{N} x_j = m$$

where

$$m = \frac{\theta - d}{A}$$

and m is successively iterated from $0, 1, \ldots M$. Then for each value of m, the value of θ is uniquely determined and hence fixed in the objective. This approach is illustrated in Figure 2.

7 Conclusions

An algorithm was presented in this paper which approximately solves the dynamic weapon target assignment problem with uniform weapon assumption. The algorithm is based upon a 2-stage convex stochastic program and uses a continuous relaxation of the integrality constraints in stage 2 to enforce the convexity. It is shown that the solution to the relaxed problem is a lower bound on the integral solution. The stage 1 assignment is to known targets and remains integral in the relaxed program. Furthermore, the stage 2 assignment is only useful in the influence it has on the stage 1 assignment; this is because the stage 2 assignment is for targets that are only conjectured to appear in the future and thus simply serves to limit the number of weapons used in the stage 1 assignment. As a result, the relaxed solution is always pertinent to the original problem and is usually close to the true solution. Further work should be done to explicitly define the the nearness of the bound. Generalization of these results to the non-uniform weapon case would also be very useful.

Let $m = 0$ and $\bar{c} = +\infty$

While $m \leq M$ DO

Step 0. For each $j = 1, \ldots n_1$, let $x_j = 0$, and denote probability of survival of target j by $S_j = V_j q_j^{x_j}$. Initialize weapon index $i = 1$. Initialize target index list $L = \{1, 2, \ldots, n_1\}$.

While $i \leq m$ and $L \neq \{\emptyset\}$ DO

Step 1. Find target k for which munition i has greatest effect:

$$k = \arg\max_{j \in L} \{S_j(1 - q_j)\}.$$

Step 2. If $S_k q_k \leq b_k$:

Add munition i to target k: $x_k = x_k + 1$,
revise probability of survival of target k: $S_k = S_k q_k$
and let $i = i + 1$.

Else remove k from L.

OD

Step 3. Let $\theta_m = mA + d$ and $\bar{c} = \min\{\bar{c}, \ \theta_m + \sum_{j=1}^{n_1} S_j\}$

$m = m + 1$.

Figure 2: Successive Minimum Marginal Return (SMMR) Algorithm

References

[1] M. Athans, P. Hosein and J. Walton, Dynamic weapon target assignment problems with vulnerable $C3$ nodes, *Proceedings of the Command and Control Symposium, Monterey, CA*, (1988).

[2] J.R. Birge (1999), "Parallel decomposition of large-scale stochastic nonlinear programs," *working paper*.

[3] C.C. Carøe, and J. Tind (1998), "L-shaped decomposition of two-stage stochastic programs with integer recourse," *Mathematical Programming*, vol. 83, pages 451-464.

[4] G.G. denBroeder, R.E. Ellison, and L. Emerling (1959), "On optimum target assignments," *Operations Research*, vol. 7, pages 322-326.

[5] M. Flood (1957), Verbal communication at The Princeton University Conference on Linear Programming.

[6] J.B. Hofman, W.A. Metler and F.L. Preston (1990), "A suite of weapon assignment algorithms for an SDI midcourse battle manager," *Naval Research Laboratory Report* **6713**.

[7] G. Laporte and F.V. Louveaux (1993), "The integer L-shaped method for stochastic integer programs with complete recourse," *Operations Research Letters*, vol. 13, pages 133-142.

[8] S.P. Lloyd and H.S. Witsenhausen (1986), "Weapons allocation is $\mathcal{NP}$-complete," *IEEE Summer Simulation Conference*.

[9] A.S. Manne (1958), "A target assignment problem," *Operations Research*, vol. 6, pages 346-351.

[10] R. Shultz, L. Stougie, and M.H. van der Vlerk (1998), "Solving stochastic programs with integer recourse by enumeration: a framework using Gröbner basis reductions," *Mathematical Programming*, vol. 83, pages 229-252.