MUTLI-STAGE RESOURCE ALLOCATION UNDER UNCERTAINTY

by

G. Calafiore, L. El Ghaoui and A. Nilim

Memorandum No. UCB/ERL M03/45

2 November 2003

MUTLI-STAGE RESOURCE ALLOCATION UNDER UNCERTAINTY

by

G. Calafiore, L. El Ghaoui and A. Nilim

Memorandum No. UCB/ERL M03/45

2 November 2003

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Multi-Stage Resource Allocation Under Uncertainty

G. Calafiore, L. El Ghaoui, A. Nilim

November 2, 2003

Abstract

In this report, we discuss a strategic planning problem of allocating resources to groups of tasks organized in successive stages. Each stage is characterized by a set of survival rates whose value is imprecisely known. The goal is to allocate the resources to the tasks (i.e. to form 'teams') by dynamically re-organizing the teams at each stage, while minimizing a cost objective over the whole stage horizon. A modelling framework is proposed, based on linear programming with adjustable variables. The resulting 'uncertain linear program' is subsequently solved using the sampled scenarios randomized technique.

1 Problem Statement

We start by describing the basic model under study. Consider Figure 1, and suppose that a total amount C of a single type of resource is available at an initial stage. These resources should be committed to a series of tasks, which are organized in successive stages, $s = 1, \ldots, N$. For instance, at the initial stage, a team $x(1) = [x_1(1) \cdots x_m(1)]^T$ is formed, where $x_i(1)$ denotes the amount of resource allocated for the i-th task in the first stage. In general, we denote with $x(s) = [x_1(s) \cdots x_m(s)]^T$ the composition of the team that is allocated for the s-th stage, and we assume (basically without loss of generality) that each stage is composed of a fixed number m of tasks.

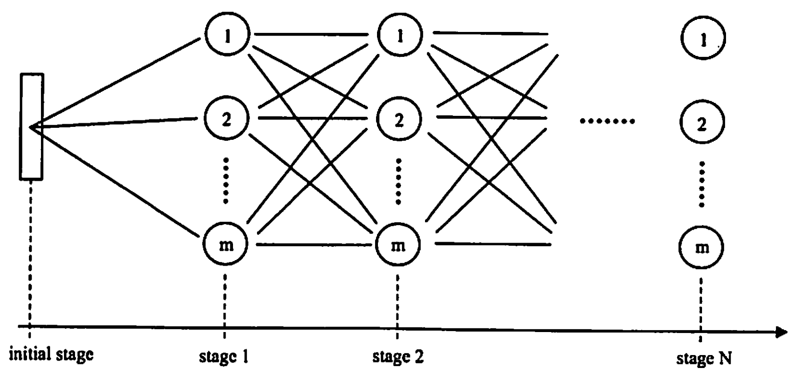

Figure 1: Multi-stage resource allocation model.

The composition of the team $x(s+1)$ (i.e. the team that should go to stage $s+1$) is decided just after the team $x(s)$ has engaged and completed the s-th stage. In our basic model, when a team engages a stage, it incurs some losses, which are described by a matrix $R(s) = \text{diag}(r(s))$ of *survival rates* $r(s) = [r_1(s) \cdots r_m(s)]^T$. If we denote with $x(s_+)$ the composition of the team $x(s)$ just after it completed the engagement with stage s, then we have

$$x(s_+) = R(s)x(s).$$

Based on the outcome of stage s, at 'time' s_+ we have the opportunity of re-adjusting the composition of team, i.e. we can decide to re-allocate resources from one task to another, before attacking stage $s+1$. This means that the composition of the team attacking stage $s+1$ is given by

$$x(s+1) = R(s)x(s) + u(s) \qquad (1)$$

where $u(s) = [u_1(s) \cdots u_m(s)]^T$ is the decision vector of resource re-allocation at stages $s = 0, \ldots, N-1$, and we set $x(0) \doteq 0$. Notice that $u_i(s) > 0$ means that more resources are committed for the i-th task, while $u_i(s) < 0$ means that the resources are withdrawn from this task.

Our goal is to determine the allocations $u(s)$, $s = 0, \ldots, N-1$, such that a certain cost objective is minimized over the entire stages horizon, and suitable constraints are satisfied. The problem constraints and objective are specified in the next section.

1.1 Constraints and optimization objective

Assume first that the stage survival rates $r(s)$, $s = 1, \ldots, N$ are exactly known in advance, and consider the dynamics of the team composition (1),

$$x(s+1) = R(s)x(s) + u(s), \quad x(0) = 0 \tag{2}$$

where $u(s) \in \mathbb{R}^m$, $s = 0, \ldots, N-1$ are the decision variables. We must impose the following physical constraints on the problem.

1. Total resources constraint.

$$\mathbf{1}^T u(0) \leq C \tag{3}$$

   where $\mathbf{1}$ denotes a vector of ones. The initial assignment should not exceed the total availability of resources, C.

2. Conservation constraints.

$$\mathbf{1}^T u(s) = 0, \quad s = 1, \ldots, N-1. \tag{4}$$

   At each stage (except for the initial one $s = 0$) the net sum of the exchanged resources must be zero.

3. Team composition constraints.

$$x_L(s) \leq x(s) \leq x_U(s), \quad s = 1, \ldots, N. \tag{5}$$

   At each stage, the resources assigned to each task should remain between a-priori fixed lower limit $x_L(s)$ (for instance $x_L(s) = 0$), and upper limit $x_U(s)$. Notice that (5) are linear constraints on the decision variable $U = [u^T(0), \ldots, u^T(N-1)]^T$, that can be explicitly expressed in the form

$$x_L(s) \le \Phi(s)U \le x_U(s), \quad s = 1, \ldots, N \tag{6}$$

where we defined, for $s = 1, \ldots, N$

$$\Phi(s) \doteq \left[R^{(s-1,1)} \; R^{(s-1,2)} \; \cdots \; R^{(s-1,s-1)} \; I_m \mid 0_{m,m} \; \cdots \; 0_{m,m}\right] \tag{7}$$

$$R^{(s-1,i)} \doteq R(s-1)R(s-2)\cdots R(i), \quad \text{for } i = 1, \ldots, s-1 \tag{8}$$

(notice that, when forming $\Phi(s)U$, the left part of $\Phi(s)$ multiplies the decision variables $u(0), \ldots, u(s-1)$, while the zero part of $\Phi(s)$ multiplies $u(s), \ldots, u(N-1)$).

Optimization objective. When transferring resources from one task to another we incur a 'transition cost,' that we assume to be proportional to the amount of the transferred resources, regardless of the sign. The optimization objective is to minimize the total transition cost accumulated over the stages horizon.

We assume that $W(s) \in \mathbb{R}^{m,m} \ge 0$, $s = 0, \ldots, N-1$, are given diagonal matrices that weight the transition costs for the different tasks at the different stages, and therefore the total cost is expressed as

$$J = \sum_{s=0}^{N-1} \|W(s)u(s)\|_1 = \|\tilde{W}U\|_1 \tag{9}$$

where $\tilde{W} \doteq \text{diag}(W(0), \ldots, W(N-1))$, and the above norm is the usual ℓ_1 vector norm, $\|x\|_1 = \sum |x_1|$.

Notice that minimizing J subject to certain constraints is equivalent to minimizing a slack variable γ subject to the original constraints, plus the constraint $J \le \gamma$ (epigraphic form). In turn, this latter constraint can be expressed as a set of linear inequalities in the decision variable U, introducing a vector $z \in \mathbb{R}^{Nm}$ of additional slack variables:

$$\|\tilde{W}U\|_1 \le \gamma \quad \Leftrightarrow \quad \begin{cases} -z \le \tilde{W}U \le z \\ \sum_{i=1}^{Nm} z_i \le \gamma. \end{cases} \tag{10}$$

Remark 1 In the above approach, transitions are penalized irrespective of the source-destination task pair, meaning that the cost is sensitive to the net resource reallocation to task i, $u_i(s)$, but does not take into account from which of the other tasks these resources are drawn. It could instead be of interest to attribute different transition costs to different source-destination pairs. This could be taken into account as follows. Denote with $d_{ij}(s)$ the resource amount (positive or negative) to be transferred from task j to task i, $i \neq j = 1, \ldots, m$, before engaging stage s. Then, we write $u_i(s) = \sum_{j \neq i} d_{ij}(s)$, or in vector notation

$$u(s) = D(s)\mathbf{1}, \tag{11}$$

where $[D(s)]_{ij} = d_{ij}(s)$ is a skew-symmetric matrix. Equation (11) represents a breakdown of the total transferred amounts $u(s)$ into the individual components $d_{ij}(s)$. We can therefore add the variables $d_{ij}(s)$ to the problem, and enforce the equality constraints (11), for $s = 1, \ldots, N-1$. Subsequently, the $d_{ij}(s)$ variables are inserted in the cost, by substituting to each term proportional to $|u_i(s)|$ in (9), a term proportional to a positive linear combination of $|d_{i1}(s)|, \ldots, |d_{im}(s)|$. $\Diamond$

From the discussion in this section, we conclude that the basic resource allocation problem is expressed as a standard linear program in the variables U, ξ, γ which can be solved with great efficiency:

$$\min_{U,z,\gamma} \gamma \text{ subject to: } (3), (4), (6), (10). \tag{12}$$

Remark 2 (Integer solutions) Although in some applications it can be reasonable to allocate fractional resources to tasks (consider for instance money as a resource, and different assets as the tasks), in some other applications the resources to be allocated must be integer multiples of a some type of unit. This is for instance the case when the resources are mobile agents such as robots, UAVs, etc. In this situation, the correct problem formulation would be in the form of an integer linear program. However, due to computational difficulties in dealing with integer programs, in this report we do not use this formulation. Instead, when we know in advance that the resulting optimal solution will need to be approximated by an integer one, we introduce an 'immunization' technique that guarantees the satisfaction of constraints against all possible approximation errors. This technique is discussed in Section 1.3. $\Diamond$

1.2 Multiple resources allocation

We next briefly describe how the basic allocation model previously discussed can be extended to deal with multiple types of resources. We hence assume hereafter that at stage $s = 0$ we have n different types of resources that should be allocated to the m tasks at stage $s = 1$, and subsequently re-organized dynamically. We denote with C_k, $k = 1, \dots, n$ the total availability of the k-th resource at the initial stage, and we let $x(s) \in \mathbb{R}^{mn}$ be the vector describing the composition of the team that is sent to stage s, $s = 1, \dots, N$. In particular, $x(s)$ is now divided into m blocks

$$x(s) \doteq \begin{bmatrix} x_1(s) \\ \vdots \\ x_m(s) \end{bmatrix}$$

where each block $x_i(s) \in \mathbb{R}^n$, $i = 1, \dots, m$ is of the form

$$x_i(s) \doteq \begin{bmatrix} x_i^{(1)}(s) \\ \vdots \\ x_i^{(n)}(s) \end{bmatrix}$$

where $x_i^{(k)}(s)$ denotes the amount of resource of type k that is allocated to the i-th task of stage s. The decision vector of resource re-allocations is partitioned similarly as

$$u(s) \doteq \begin{bmatrix} u_1(s) \\ \vdots \\ u_m(s) \end{bmatrix}$$

where

$$u_i(s) \doteq \begin{bmatrix} u_i^{(1)}(s) \\ \vdots \\ u_i^{(n)}(s) \end{bmatrix}$$

$u_i^{(k)}(s)$ denoting the amount of resource of type k that we decide (upon completion of stage s) to add or subtract to the i-th task. With this notation, the team dynamics retain the structure (1)

$$x(s+1) = R(s)x(s) + u(s)$$

where the survival rate matrix $R(s) \in \mathbb{R}^{mn,mn}$ is block-partitioned conformably to $x(s), u(s)$. The total resources constraint (3) now writes

$$(\mathbf{1}_m^T \otimes e^T(n,k))u(0) \leq C_k, \quad k = 1, \ldots, n$$

where $e^T(n,k)$ denotes a vector in $\mathbb{R}^n$ with all zeros, except for the k-th component, which is set to one. Similarly, the conservation constraints (4) are now expressed as

$$(\mathbf{1}_m^T \otimes e^T(n,k))u(s) = 0, \quad k = 1, \ldots, n; \quad s = 1, \ldots, N-1.$$

Basically, all the rest of the problem model and solution goes through in the same way as described for the basic problem with a single resource.

1.3 Dealing with integer approximations

As discussed in Remark 2, in some applications we need to deal with integer quantities in problem (12). In this situation, both the team composition $x(s)$ and the re-allocations $u(s)$ must be integers. This, however, is in contrast with the dynamic model (2), since $R(s)$ is real (its elements are in fact probabilities of survival), and therefore $x(s+1)$ will result to be real, even if $x(s), u(s)$ are integer vectors. One idea is to take into account into the dynamic model the presence of integer approximation errors. In particular, we assume that a first error $\zeta(s)$ is introduced when $R(s)x(s)$ is replaced by its integer approximation, and a second error $\varrho(s)$ is due to the integer approximation of $u(s)$. The dynamic model now becomes

$$x(s+1) = R(s)x(s) + \zeta(s) + (u(s) + \varrho(s)) \tag{13}$$

where $\|\zeta(s)\|_\infty \leq 0.5$, $\|\varrho(s)\|_\infty \leq 0.5$. We now review the problem constraints, considering the presence of these errors.

The constraint (3) should now be 'immunized' against approximation errors, i.e. it becomes

$$\mathbf{1}^T u(0) + \mathbf{1}^T \varrho(0) \leq C, \quad \forall \varrho(0) : \|\varrho(0)\|_\infty \leq 0.5$$

which, since $\varrho(0) \in \mathbb{R}^m$, simply writes

$$\mathbf{1}^T u(0) + \frac{m}{2} \leq C. \tag{14}$$

The conservation constraints (4) are equality constraints, and therefore impose a restriction on the allowable approximation errors $\varrho(s)$, $s = 1, \dots, N-1$, which must hence be assumed to belong to the set $\Xi = \{z \in \mathbb{R}^m : \|z\|_\infty \leq 0.5,\ \mathbf{1}^T z = 0\}$. With this position, (4) remain unchanged.

For the team composition constraints (5), notice that setting $\varrho = [\varrho^T(0), \dots, \varrho^T(N-1)]^T$ and $\zeta = [0_{1,m}, \zeta^T(1), \dots, \zeta^T(N-1)]^T$, we have

$$x(s) = \Phi(s)\,(U + \varrho + \zeta)$$

and hence the constraints

$$x_L(s) \leq \Phi(s)\,(U + \varrho + \zeta) \leq x_U(s), \quad \forall \varrho \in \Xi^{(N)}, \zeta \in Z^{(N)}; \quad \text{for } s = 1, \dots, N \tag{15}$$

where

$$\begin{aligned} \Xi^{(N)} &\doteq \{[z_1^T, z_2^T]^T : \|z_1\|_\infty \leq 0.5, z_2 \in \{\Xi \times \Xi \times \cdots \times \Xi\} \\ Z^{(N)} &\doteq \{[0_{1,m}, z_1^T, \cdots, z_{N-1}]^T : z_i \in \mathbb{R}^m, \|z_i\|_\infty \leq 0.5, i = 1, \dots, N-1\}. \end{aligned}$$

In turn, the constraints (15) are equivalent to

$$\begin{aligned} \Phi(s)U + \sup_{\varrho \in \Xi^{(N)}, \zeta \in Z^{(N)}} \Phi(s)(\varrho + \zeta) &\leq x_U(s) \\ \Phi(s)U + \inf_{\varrho \in \Xi^{(N)}, \zeta \in Z^{(N)}} \Phi(s)(\varrho + \zeta) &\geq x_L(s). \end{aligned}$$

The values of ϱ, ζ attaining the previous sup (say $\bar{\varrho}(s), \bar{\zeta}(s)$), and inf (say $\underline{\varrho}(s), \underline{\zeta}(s)$) are determined solving two linear programs, and therefore the composition constraints finally write

$$\Phi(s)(U + \bar{\varrho}(s) + \bar{\zeta}(s)) \leq x_U(s) \tag{16}$$

$$\Phi(s)(U + \underline{\varrho}(s) + \underline{\zeta}(s)) \geq x_L(s), \tag{17}$$

for $s = 1, \dots, N$.

Finally, we notice that the constraints related to the objective can be treated similarly to the previous case. Specifically, the inequalities (10) now write

$$\begin{aligned} &\tilde{W}U + \sup_{\varrho \in \Xi^{(N)}, \zeta \in Z^{(N)}} \tilde{W}(\varrho + \zeta) \leq z \\ &\tilde{W}U + \inf_{\varrho \in \Xi^{(N)}, \zeta \in Z^{(N)}} \tilde{W}(\varrho + \zeta) \geq -z \\ &\sum_{i=1}^{Nm} z_i \leq \gamma, \end{aligned} \tag{18}$$

where the values of ϱ, ζ attaining the extrema can again be computed solving two linear programs.

2 Resource Allocation under Uncertainty

The formulation introduced in the previous section hinges on the very unrealistic hypothesis that the values of the survival rates $r(s)$ at the various stages are exactly known. In the following, we relax this assumption and consider the problem of resource allocation under uncertainty. Specifically, we assume that the survival rate vectors $r(s)$ are of the form

$$r(s) = \bar{r}(s) + \delta(s), \quad s = 1, \ldots, N-1$$

where $\bar{r}(s)$ is the known nominal value of the rate, and $\delta(s) \in \Delta(s)$ represent unknown 'fluctuations' or uncertainties around the nominal value, with $\Delta(s) \subseteq \mathbb{R}^m$ representing the allowable range of variation of the uncertainties.

A first idea in this respect would be to apply a 'robust optimization' methodology (see e.g. [2, 4]), and solve a version of problem (12) where the constraints are enforced for all admissible values of the uncertainty. This approach is however likely to be very conservative, since it neglects an important feature of the problem at hand, that is, there exist a stage schedule according to which the decisions have to be taken. To clarify the concept, we observe that not all the adjustments $u(s)$ need to be computed in advance (i.e. at the initial stage $s = 0$). Instead, only the decision $u(0)$ need to be taken at $s = 0$ (here-and-now decision), while before deciding for $u(1)$, we can wait and see what happens to the teams as they complete stage $s = 1$. In other words, the decision at $u(1)$ can benefit from the knowledge of the realization of the 'uncertainty' at $s = 1$. More generally, we observe that each decision $u(s)$, $s = 1, \ldots, N-1$ can benefit from a 'basis of knowledge' of what happened from the initial stage up to s.

To exploit this information in a manageable way, we here assume that each decision vector $u(s)$ can be adjusted in function of the realization of $r(s)$, and we explicitly set up an affine dependence of the form

$$u(s) = \bar{u}(s) + H(s)\delta(s) \tag{19}$$

where now $\bar{u}(s) \in \mathbb{R}^m$ and $H(s) \in \mathbb{R}^{m,m}$, $s = 0, \ldots, N-1$ (with $H(0) \equiv 0_{m,m}$) are the new optimization variables. In more compact matrix form, we have that

$$U = \bar{U} + \bar{H}\bar{\delta}$$

where $\bar{U} \doteq [\bar{u}^T(0) \cdots \bar{u}^T(N-1)]^T$ and $\bar{H} \doteq \mathrm{diag}(0_{m,m}, H(1), \ldots, H(s-1))$ contain optimization variables, and $\bar{\delta} \doteq [0_{1,m}\, \delta^T(1) \cdots \delta^T(N-1)]^T \in \mathcal{D}$

contains the uncertainty terms, where $\mathcal{D} \doteq \{[0_{1,m}\, q] \,:\, q \in \Delta(1) \times \cdots \times \Delta(N-1)\}$.

With these positions, we now write the 'adjustable robust' version (see e.g. [1]) of our optimal allocation problem as

$$\min_{\gamma, z, \bar{U}, \bar{H}} \gamma \quad \text{subject to:} \tag{20}$$

$$\mathbf{1}^T \bar{u}(0) \le C \tag{21}$$

$$\mathbf{1}^T \bar{u}(s) = 0; \quad s = 1, \ldots, N-1 \tag{22}$$

$$\mathbf{1}^T H(s) = 0_{1,m}; \quad s = 1, \ldots, N-1 \tag{23}$$

$$x_L(s) \le \Phi(s, \bar{\delta}) \left(\bar{U} + \bar{H}\bar{\delta}\right) \le x_U(s), \tag{24}$$

$$\forall \bar{\delta} \in \mathcal{D}; \quad s = 1, \ldots, N \tag{25}$$

$$-z \le \tilde{W}(\bar{U} + \bar{H}\bar{\delta}) \le z, \quad \forall \bar{\delta} \in \mathcal{D} \tag{26}$$

$$\sum_{i=1}^{Nm} z_i \le \gamma. \tag{27}$$

In the above problem, we used the notation $\Phi(s, \bar{\delta})$ to underline the fact that the matrix $\Phi(s)$ defined in (7) depends on the survival rates $R(s) = \text{diag}(r(s))$, $s = 1, \ldots, N-1$, and hence on the uncertainty $\bar{\delta}$.

Problem (20)–(27) is a robust linear program, i.e. a linear program having a continuous infinity of constraints, see [1, 2]. In the mentioned papers, the authors show that in several 'tractable' cases the robust linear program can be converted *exactly* into a standard convex program having a finite number of constraints, and hence solved efficiently via interior point methods. Problem (20)–(27), however, does not fall among the tractable cases, since the uncertainty is affecting the problem data in a non-linear way, and the 'recourse matrix' (i.e. the matrix $\Phi(s)$ that multiplies the adjustable variables, see [1]) is itself dependent on the uncertainty. Besides this technical difficulty, another motivation for not pursuing the worst-case approach is that this approach places evenly the importance among the possible uncertainty outcomes. In practical applications, one instead typically knows that some outcomes are 'more likely' than others, and may wish to exploit this knowledge when computing a solution.

We next describe a recently developed methodology for solving a probabilistic relaxation of problem (20)–(27).

3 Scenario-based Optimization

The idea behind scenario-based solutions of robust linear programs is very simple: instead of considering the whole infinity of constraints of the problem, we consider only a finite number M of these constraints, selected at random according to a given probability distribution. Specifically, the constraints in (20)–(27) are parameterized by $\bar{\delta} \in \mathcal{D}$. Therefore, assuming a probability measure Π over $\mathcal{D}$, we first extract M (we shall discuss later 'how large' M should be) independent and identically distributed samples of $\bar{\delta}$: $\bar{\delta}^{(1)}, \ldots, \bar{\delta}^{(M)}$, which constitute our uncertainty scenarios upon which we base our design. We remark that the choice of the probability measure Π now reflects our additional knowledge on which outcomes of the uncertainty are more likely than others. Subsequently, we solve the 'scenario counterpart' of the robust problem (20)–(27), which is defined below.

$$\min_{\gamma,\xi,\bar{U},\bar{H}} \gamma \quad \text{subject to:} \tag{28}$$

$$\mathbf{1}^T \bar{u}(0) \leq C \tag{29}$$

$$\mathbf{1}^T \bar{u}(s) = 0; \quad s = 1, \ldots, N-1 \tag{30}$$

$$\mathbf{1}^T H(s) = 0_{1,m}; \quad s = 1, \ldots, N-1 \tag{31}$$

$$x_L(s) \leq \Phi(s, \bar{\delta}^{(i)}) \left(\bar{U} + \bar{H}\bar{\delta}^{(i)}\right) \leq x_U(s), \tag{32}$$

$$i = 1, \ldots, M; \quad s = 1, \ldots, N \tag{33}$$

$$-\xi \leq \tilde{\underline{W}}(\bar{U} + \bar{H}\bar{\delta}^{(i)}) \leq \xi, \quad i = 1, \ldots, M \tag{34}$$

$$\sum_{i=1}^{Nm} \xi_i \leq \gamma. \tag{35}$$

A first immediate consideration about (28)–(35) is that it is a standard linear program (with a possibly large, but finite number of constraints), which is easily solvable by LP numerical codes. A fundamental question is however related to what kind of guarantees of robustness can be provided by a solution that a-priori satisfies only a finite number M of selected constraints. A good news in this respect is that, if we sample a sufficiently large number of constraints, then the scenario solution will be 'approximately feasible' for the robust problem (20)–(27), i.e. the probability measure of the set of uncertainties such that the corresponding constraints are violated by the scenario solution goes to zero rapidly as M increases. This result has been recently derived in [3], and it is next contextualized to the problem at hand.

3.1 Approximate feasibility of scenario solutions

Consider a generic robust LP in the form

$$\min_x c^T x \text{ subject to } A(\xi)x \leq b, \quad \forall \xi \in \mathcal{X} \tag{36}$$

where $x \in \mathbb{R}^n$ and $\mathcal{X} \subseteq \mathbb{R}^\ell$ is a closed set, and no restrictions are imposed on the dependence of the data matrix A on ξ. Assume that (36) is feasible, and suppose that a probability measure Π is imposed on $\mathcal{X}$. Then, the scenario counterpart of (36) is the linear program

$$\min_x c^T x \text{ subject to } A(\xi^{(i)})x \leq b, \quad i = 1, \ldots, M \tag{37}$$

where $\xi^{(i)}$, $i = 1 \ldots, M$ are iid samples of $\xi \in \mathcal{X}$, extracted according to probability Π. Assume further that (37) has a unique optimal solution x^* (this uniqueness assumption is technical and could be removed, see [3]). Clearly, the scenario solution x^* depends on the random sample $\xi^{(i)}$, $i = 1 \ldots, M$, and it is hence a random variable. The following theorem highlights the 'approximate feasibility' property of this solution.

Theorem 1 *Fix a probabilistic risk level $\epsilon \in (0, 1)$ and a confidence level $\beta \in (0, 1)$, and let x^* be the optimal solution of the scenario problem (37), computed with*

$$M \geq \frac{n}{\epsilon\beta} - 1. \tag{38}$$

Then, it holds with probability at least $1 - \beta$ that

$$\text{Prob}\{\xi \in \mathcal{X} : A(\xi)x^* \not\leq b\} \leq \epsilon. \tag{39}$$

In other words, this theorem states that the measure of the set of uncertainties that could possibly violate the inequality $A(\xi)x^* \leq b$ can be made arbitrarily small by sampling a sufficient number of scenarios, and therefore we say that the scenario solution is (with high probability $1 - \beta$) approximately feasible for the robust problem (36), i.e. it satisfies all but a small set of the original constraints.

3.2 A-posteriori analysis

It is worth noticing that a distinction should be made between the a-priori and a-posteriori assessments that one can make regarding the probability of constraint violation for the scenario solution. Indeed, *before* running the optimization, it is guaranteed by Theorem 1 that if $M \geq n/\epsilon\beta - 1$ samples are considered, the solution of the scenario problem will be ϵ-approximately feasible, with probability no smaller than $1-\beta$. However, the a-priori parameters ϵ, β are generally chosen not too small, due to technological limitations on the number of constraints that one specific optimization software can deal with.

On the other hand, once a scenario solution has been computed (and hence $x = x^*$ is fixed), one can make an a-posteriori assessment of the level of feasibility using Monte-Carlo techniques. In this case, a new batch of $\tilde{M}$ independent random samples of $\xi \in \mathcal{X}$ is generated, and the *empirical probability* of constraint violation, say $\hat{V}_{\tilde{M}}(x^*)$, is computed according to the formula $\hat{V}_{\tilde{M}}(x^*) = \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} 1(A(\xi^{(i)})x^* \leq b)$, where $1(\cdot)$ is the indicator function. If $V(x^*) \doteq \text{Prob}\{\xi \in \mathcal{X} : A(\xi)x^* \not\leq b\}$ denotes the true violation probability, then the classical Hoeffding's inequality, [5], states that

$$\text{Prob}\{|\hat{V}_{\tilde{M}}(x^*) - V(x^*)| \leq \tilde{\epsilon}\} \geq 1 - 2\exp(-2\tilde{\epsilon}^2\tilde{M}),$$

from which it follows that $|\hat{V}_{\tilde{M}}(x^*) - V(x^*)| \leq \tilde{\epsilon}$ holds with confidence greater than $1 - \tilde{\beta}$, provided that

$$\tilde{N} \geq \frac{\log 2/\tilde{\beta}}{2\tilde{\epsilon}^2} \tag{40}$$

test samples are drawn. This latter a-posteriori test can be easily performed using a very large sample size $\tilde{N}$ because no optimization problem is involved in such an evaluation.

Returning to our resource allocation problem, the solution procedure that we propose is the following one.

1. Select the a-priori probabilistic risk level ϵ and confidence β, and compute the number of necessary scenarios according to (38). We remark that experimental numerical experience showed that the actual probabilistic levels achieved by the scenario solution are usually *much* better than the ones established a-priory by means of Theorem 1. This fact suggest in practice not to insist on too small a-priori levels.

2. Solve the scenario LP (28)–(35), obtaining the optimal variables $\gamma^*, z^*, \bar{U}^*, \bar{H}^*$.

3. Test a-posteriori the obtained solution via Monte-Carlo, using a large sample size, so to determine a very reliable estimate of the actual probability of violation of the scenario solution. If this level of probability

is acceptable for the application at hand, we are finished, otherwise we may try another scenario design step, taking into account a larger set of sampled scenarios, and iterate the procedure.

3.3 Interaction models

In this section, we propose an iterative heuristic for the solution of a modified allocation problem. Consider the generic robust LP problem formulated in (36). In deriving the scenario counterpart of this problem, we assumed that a *fixed* probability distribution was assigned on the uncertain parameter ξ. In terms of the actual resource allocation problem, this assumption means that the survival rates are random variables, and that we know a-priori their probability distributions. However, a more realistic model of the problem should be able to take into account *interaction* effects between the decision variables and the uncertainties. By interaction we here mean that the probability distribution of the survival rates of a certain stage could be itself dependent on the amount of resources that we commit for that stage. For instance, the overall chance of surviving a given stage may increase if we send more resources to that stage.

A way of modelling this interaction in our generic framework (36) is to assume that the probability distribution on $\xi \in \mathcal{X}$ depends on x, that is, we assign a *conditional* distribution $\Pi(\xi|x)$ on ξ. Clearly, if interaction is present, we can no longer apply directly the scenario approach, since the correct distribution according to which we need to sample the scenarios is unknown. We hence propose the following iterative heuristic to solve the problem in presence of interaction.

1. Let an initial solution $x^{(k)}$, $k = 0$ be available;

2. Draw random scenarios $\xi^{(1)}, \ldots, \xi^{(M)}$ according to probability $\Pi(\xi|x^{(k)})$, and solve the resulting scenario problem. Let $k \leftarrow k + 1$, and denote with $x^{(k)}$ the optimal scenario solution;

3. Repeat 2., until some suitable convergence condition is reached.

The effectiveness of this heuristic needs to be tested on numerical examples.

4 Numerical Examples

In this section, we address the problem of allocating UAVs (Unmanned Aerial Vehicles) optimally and dynamically in order to perform various sequential tasks where risk is present due to hostile opponents. In practice, it is often required to allocate UAVs in teams in order to perform various sequential tasks in a hostile environment where their survival rates are uncertain. This makes this particular application a good illustration of our method.

4.1 The nominal problem

In this problem, we consider that our opponent has 5 different types of equipments, namely small SAM (surface to air missile), medium SAM, large SAM and the Early Warning Radars (EWRs). All kinds of SAMs have destructive capability. However, the EWRs and the Long SAM-fcs(fire control sensor) work as tracking and sensing tools and don't have any destructive capability. Their destructive ranges are given in Table 1. The enemy equipments with higher destructive ranges are riskier to destroy than that with lower destructive ranges.

	Small SAM	Medium SAM	Large SAM	Long Sam-fcs	EWR
Range (km)	25	50	100	0	0

Table 1: Ranges of opponent's equipments

The controller needs to assign UAVs to teams in order to perform six main tasks, which are destroying 6 enemy EWRs, namely EWR1, EWR2, EWR3, EWR4, EWR5 and EWR6. However, due to the presence of other enemy SAMs, it is not possible to destroy all the 6 targets with an acceptable risk level, unless some other enemy SAMs are destroyed. As a result, in order to perform the main tasks under an acceptable risk level, we need to destroy other targets first. We define primary targets as the targets which are originally assigned to be destroyed and secondary targets as the targets that need to be destroyed in order to reduce the risk inherent to the mission to the primary targets. Hence, the assignment problem becomes a sequential and a dynamic one. We perform the assignment in 'waves' (stages): we start the assignment process by forming teams in the first wave in a way that they destroy some assigned targets that are under a threshold risk level. Once the targets are destroyed at the first stage, the risk for the targets to be destroyed in the second wave is reduced under the threshold level. At the end of the first wave, we reassign the team composition among the survived UAVs in order to destroy the targets assigned for the second wave. We keep reassigning till we destroy all the assigned targets in all the stages. In this experiment, the targets to be destroyed in various stages are input data, as described in Table 2.

Threat Name	Objective Classification	wave
Medium SAM 27	Secondary	1
EWR3	Primary	1
EWR 1	Primary	1
Long SAM-fcs 3	Secondary	1
Medium SAM 5	Secondary	1
Medium SAM 3	Secondary	1
Medium SAM 28	Secondary	1
Long SAM 14	Secondary	2
EWR 2	Primary	2
Long SAM 2	Secondary	2
Medium SAM 9	Secondary	2
Medium SAM 2	Secondary	2
Medium SAM 30	Secondary	2
Long SAM-fcs 4	Secondary	2
Medium SAM 10	Secondary	3
Long SAM 8	Secondary	3
EWR 4	Primary	3
Medium SAM 12	Secondary	3
Long SAM 5	Secondary	4
Medium SAM 13	Secondary	4
Long SAM 7	Secondary	4
Medium SAM 14	Secondary	4
EWR 5	Primary	4
EWR 6	Primary	4

Table 2: Tasks

According to the input data, we define $m_p = [7\ 6\ 4\ 6]^T \in \mathbb{R}^N$, where N denotes the total number of stages, and $m_p(s)$ denotes the number of tasks at the stage s. In order to be consistent with notations described earlier in the paper, we define $m := \max_i(m_p(i))$ and therefore we assign $m = 7$ tasks at each stage. However, if the number of tasks is $l < m$ in stage s, we add extra slack $(m-l)$ tasks in that stage, such that $0 \leq x_i(s) \leq 0$ for all $l < i \leq m$. We also assume that the cost for UAV allocation at the first wave is 1 unit and the transaction cost in the later waves is TC units,

where C is an experimental variable. Total number of available UAV is 40. The nominal survival rate of an UAV when assigned to destroy medium SAM, long SAM, and EWRs are 0.65, 0.5, and 1 respectively. Moreover, there is an additional constraint imposed in this resource allocation problem: at any stage, at least one UAV is required to be assigned to each target.

We ran the experiment with different transition costs $TC = 0, 0.9, 1, 100, 000$ and the resulting team constitutions are as shown in Table 3-6. In the later part of the paper, when we mention $TC = \infty$, we actually mean that the experimental run was performed with $tc = 100,000$. Total cost and the total number of UAVs required for the various assignments are summarized in Table 7.

	Stage 1	Stage 2	Stage 3	Stage 4
Team 1	Medium SAM27 (1)	Long SAM14 (1)	Medium SAM10 (1)	Long SAM5(1)
Team 2	EWR3 (3)	EWR2 (4)	Long SAM8 (1)	Medium SAM13(1)
Team 3	EWR1 (3)	Long SAM2 (1)	WEWR4(4)	Long SAM7(1)
Team 4	Long SAM-fcs3 (1)	Medium SAM9(1)	Medium SAM12 (1)	Medium SAM14(1)
Team 5	Medium SAM5 (1)	Medium SAM30(1)	(0)	EWR5(1)
Team 6	Medium SAM3 (1)	Long SAM-fcs4(1)	(0)	EWR6(1)
Team 7	Medium SAM29 (1)	(0)	(0)	(0)

Table 3: Task Assignment with $TC = 0$

	Stage 1	Stage 2	Stage 3	Stage 4
Team 1	Medium SAM27 (1)	Long SAM14 (1)	Medium SAM10 (1)	Long SAM5(1)
Team 2	EWR3 (5)	EWR2 (5)	Long SAM8 (1)	Medium SAM13(1)
Team 3	EWR1 (2)	Long SAM2 (1)	WEWR4(4)	Long SAM7(1)
Team 4	Long SAM-fcs3 (1)	Medium SAM9(1)	Medium SAM12 (1)	Medium SAM14(1)
Team 5	Medium SAM5 (1)	Medium SAM30(1)	(0)	EWR5(1)
Team 6	Medium SAM3 (1)	Long SAM-fcs4(1)	(0)	EWR6(1)
Team 7	Medium SAM29 (1)	(0)	(0)	(0)

Table 4: Task Assignment with $TC = 0.2$

It is clear from Table 7 that the total cost and the total number of UAVs required increase with the increase of transition cost. When $TC = \infty$, the

	Stage 1	Stage 2	Stage 3	Stage 4
Team 1	Medium SAM27 (2)	Long SAM14 (1)	Medium SAM10 (1)	Long SAM5(1)
Team 2	EWR3 (4)	EWR2 (4)	Long SAM8 (4)	Medium SAM13(1)
Team 3	EWR1 (5)	Long SAM2 (5)	EWR4(3)	Long SAM7(1)
Team 4	Long SAM-fcs3 (1)	Medium SAM9(1)	Medium SAM12 (1)	Medium SAM14(1)
Team 5	Medium SAM5 (2)	Medium SAM30(1)	(0)	EWR5(1)
Team 6	Medium SAM3 (2)	Long SAM-fcs4(1)	(0)	EWR6(1)
Team 7	Medium SAM29 (1)	(0)	(0)	(0)

Table 5: Task Assignment with $TC = 1$

	Stage 1	Stage 2	Stage 3	Stage 4
Team 1	Medium SAM27 (8)	Long SAM14 (5)	Medium SAM10 (2)	Long SAM5(1)
Team 2	EWR3 (2)	EWR2 (2)	Long SAM8 (2)	Medium SAM13(1)
Team 3	EWR1 (2)	Long SAM2 (2)	WEWR4(2)	Long SAM7(1)
Team 4	Long SAM-fcs3 (8)	Medium SAM9(4)	Medium SAM12 (2)	Medium SAM14(1)
Team 5	Medium SAM5 (4)	Medium SAM30(2)	(1)	EWR5(1)
Team 6	Medium SAM3 (4)	Long SAM-fcs4(2)	(1)	EWR6(1)
Team 7	Medium SAM29 (2)	(1)	(1)	(1)

Table 6: Task Assignment with $TC = \infty$

team assignment becomes static and produces higher total cost.

If the survival rates are certain and accurate, the assignments obtained by using our algorithms produce the minimum cost, while using the minimum number of UAVs.

4.2 The Robust Counterpart

In the experiment, we assume that the survival rates of UAVs while encountering SAMs are not certain. Instead, the survival rates are stochastic. We suppose for the purpose of this example that the survival rates while encountering medium SAMs and long SAMs follow uniform distributions $\mathbf{U}[0.45\ ,0.55]$ and $\mathbf{U}[0.6\ ,0.7]$ respectively. Using the algorithm discussed in Section 3, we ran the experiment with varying TC. In each case, we randomly picked 15 sample points. Though the algorithm doesn't always guarantee a team assignment that satisfies all the constraints, it does in most of the tested cases. We ran each of the experiment 20 times . In each of the experiment with a fixed TC, our algorithms produce assignments that yield a total cost

	Total Cost	Total Number of UAVs
$TC = 0$	11	11
$TC = 0.2$	15.02	13
$TC = 1$	23.98	17
$TC = \infty$	30	30

Table 7: Total cost incurred and total number of UAVs required

and the total number of UAVs required to complete all the tasks. We compute the averages of these two quantities over all the runs. The summary is shown in Table 8.

	Average Total Cost	Average Total Number of UAVs	% Successful run
$TC = 0$	12.06	12	85%
$TC = 0.2$	15.56	13	90%
$TC = 1$	26.20	17	80%
$TC = \infty$	37	36.96	85%

Table 8: Total cost incurred and total number of UAVs required

We observe that even with the presence of uncertainty in the survival rate, our algorithm performs well. In most of the runs, the stochastic algorithm is able to satisfy all the constraints.

Moreover, we ran an experiment where the survival rate is stochastic but the nominal team assignments are used. We record the percentage of time the nominal assignment produces successful run (satisfies all the constraints) and compare the results with the robust counterpart. The comparison is shown in Figure 2. We observe that robust algorithm provides successful assignment significantly more often than the nominal counterpart under uncertainty.

We conclude that the robust assignment algorithm based on robust linear program performs very well even if the survival rate is uncertain.

5 Conclusion

We have proposed a strategic planning scheme that allocates resources to groups of tasks organized in successive stages. Our algorithms allocate the resources to the tasks (i.e. form 'teams') by dynamically re-organizing the teams at each stage, while minimizing a cost objective over the whole stages

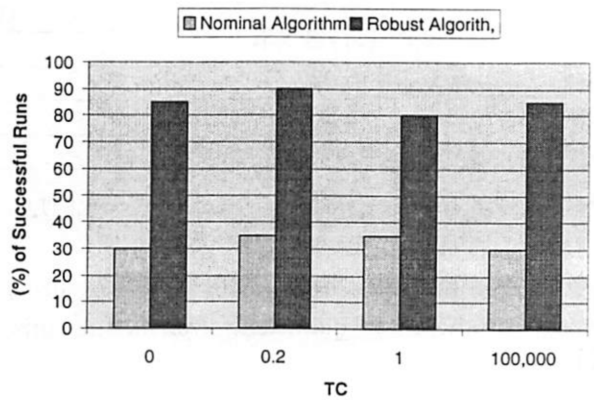

Figure 2: Percentage of the runs when all the constraints are satisfied.

horizon. Furthermore, we have proposed an algorithm based on 'linear programming with adjustable variables,' that can solve uncertain linear program by means of the sampled scenarios randomized technique. We have applied our algorithm to a problem of UAVs allocation in an uncertain and risky environment. We have shown that our model provides an optimal solution to the problem while satisfying all the constraints in most of the runs.

In the specific context of UAV allocation, many further issues remain open for numerical investigation. First, we have here considered a fixed statistical model for the survival rates. However, a model that takes into account 'interactions' (see Section 3.3) or at least a saturation on the survival rates seems better suited for the application at hand. Also, we would like to add origin-destination dependent transaction costs at each stage in our model, as discussed in Remark 1, as well as different types of UAVs.

6 Ackwoledgement

The authors would like to thank Pravin Varaiya for valuable suggestions and constant guidance in every stage of this research.

References

[1] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Minerva Report, http://iew3.technion.ac.il/Labs/Opt/*, 2003.

[2] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

[3] G. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *To appear in Mathematical Programming*, 2004.

[4] L. El Ghaoui and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM J. Optim.*, 9(1):33–52, 1998.

[5] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.