Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2009-016 April 22, 2009

Risk Allocation for Multi-agent Systems using TˆTtonnement

Masahiro Ono and Brian C. Williams

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Risk Allocation for Multi-agent Systems using Tâtonnement *

Masahiro Ono† Brian C. Williams‡

April 22, 2009

Abstract

This paper proposes a new market-based distributed planning algorithm for multi-agent systems under uncertainty, called MIRA (Market-based Iterative Risk Allocation). In large coordination problems, from power grid management to multi-vehicle missions, multiple agents act collectively in order to optimize the performance of the system, while satisfying mission constraints. These optimal plans are particularly susceptible to risk when uncertainty is introduced. We present a distributed planning algorithm that minimizes the system cost while ensuring that the probability of violating mission constraints is below a user-specified level.

We build upon the paradigm of *risk allocation* [13], in which the planner optimizes not only the sequence of actions, but also its allocation of risk among each constraint at each time step. We extend the concept of risk allocation to multi-agent systems by highlighting risk as a good that is traded in a computational market. The equilibrium price of risk that balances the supply and demand is found by an iterative price adjustment process called *tâtonnement* (also known as *Walrasian auction*). The simulation results demonstrate the efficiency and optimality of the proposed distributed planner.

1 Introduction

1.1 Motivation

There is an increasing need for multi-agent systems that perform optimal planning under uncertainty in multi-agent system. An example is planning and control of power grid systems [3][15][21]. A power grid consists of a numbers of generators and electric transformers whose control should be carefully planned in order to maximize efficiency. A significant issue in power grid planning is the uncertainty in demand for energy by consumers. As the use of renewable energy, such as solar and wind power, become more popular, uncertainty in supply increases due to weather conditions.

Another example is the Autonomous Ocean Sampling Network (AOSN) [7][17], which consists of multiple automated underwater vehicles (AUVs), robotic buoys, and

*This research is funded by The Boeing Company grant MIT-BA-GTA-1

†Masahiro Ono is a PhD student in CSAIL, MIT. hiro_ono@mit.edu

‡Brian C. Williams is a professor in CSAIL, MIT. williams@mit.edu

aerial vehicles. AOSN should maximize science gain while being exposed to external disturbances, such as tides and currents.

To deal with such problems, we developed *Market-based Iterative Risk Allocation* (MIRA), a multi-agent optimal planning algorithm that operates within user-specified risk bounds.

1.2 Scope of this Paper

We address the problem of planning in continuous state space with time-evolved goals, which are expressed as Qualitative State Plans [10][8]. This paper considers the problem in which effects have stochastic distribution such as Gaussian, goals are convex regions on state space, and sequence of actions are planned cooperatively by multiple agents.

1.3 Overview

Planning under uncertainty, and risk allocation When planning actions under uncertainty, there is always a risk of failure. However, in many cases, performance can be improved only by taking extra risk. We can reach a destination faster by driving at a faster speed and accepting a higher risk of an accident. Hannibal, a Carthaginian military commander in the third century B.C., was able to frustrate the Roman army by taking the great risk of crossing the Alps with 50,000 troops. As seen in these examples, risk and performance are in a trade-off relationship. In other words, risk is a resource that can be spent to improve the performance of the system.

Without taking any risk, nothing can be done; however, no one dares to take unlimited risk. Although the sensitivity for risk varies from person to person, everyone somehow balances risk and performance to find the optimal plan.

There are three main ways to formulate the trade-off problem of risk and performance; the first is to set a negative utility for failure (i.e. penalty), and maximize the expected total utility (the utilitarian approach, such as embodied in MDP); the second is to set upper bound on risk and maximize performance within this bound; the third is to set lower bound on performance and minimize risk. It is up to the system operator to choose which formulation to use according to her needs and requirements.

Our focus is on the second approach: performance maximization with an upper-bound on risk. An example problem is to drive a car as fast as possible while limiting the probability of a crash to 0.01%. This formulation is particularly useful for planning problems that involve high-impact low-probability risk such as loss of life.

With this formulation, [13] showed that the planner should plan not only the sequence of actions but also the *risk allocation* in order to maximize the performance under a risk bound .

The example shown in Figure 1 illustrates the concept of risk allocation. A race car driver wants to plan a path to get to the goal as fast as possible. However, crashing into the wall leads to a fatal accident, so he wants to limit the probability of a crash to 0.01%. An intelligent driver would plan a path as shown in Figure 1, which runs mostly in the middle of the straightaway, but gets close to the wall at the corner. Why? It is because taking a risk (i.e. approaching the wall) at the corner results in a greater time saving than taking the same risk along the straightaway; in other words, the utility of taking risk is greater at the corner than the straightaway. Therefore the optimal plan allocates a large portion of risk to the corner, while allocating little to the straightaway. As illustrated by this example, *risk allocation* needs to be optimized across the constraints, in order to maximize the performance.

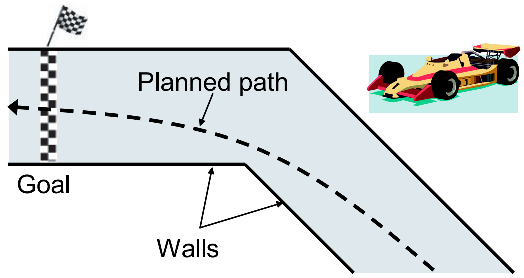

Figure 1: Risk allocation in a race car path planning scenario. A large portion of risk is allocated to the corner, since taking a risk (approaching the wall) at corner results in greater time saving than taking the same risk along straightaway.

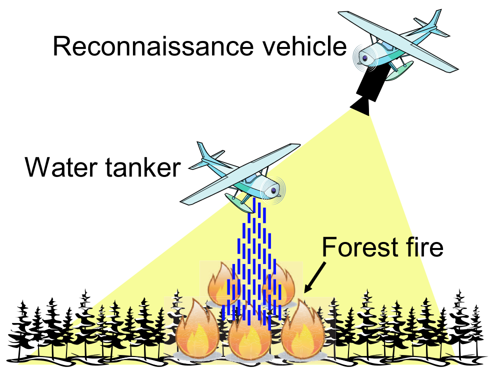

Figure 2: Risk allocation for multi-UAV fire-fighting system. The water tanker is allowed to fly low since it is allocated larger risk than the reconnaissance vehicle.

The planner then needs to generate an optimal action sequence that abides to the allocated risk at each constraint.

Distributed risk allocation for multi-agent system The concept of risk allocation can be naturally extended to multi-agent systems. Figure 2 shows an example of a multi-agent system with two unmanned air vehicles (UAVs), whose mission is to extinguish a forest fire. A water tanker drops water while a reconnaissance vehicle monitors the fire with its sensors. The loss of either vehicle results in a failure of the mission. Two vehicles are required to extinguish the fire as efficiently as possible, while limiting the probability of mission failure to a given risk bound, say, 0.1%. The water tanker can improve efficiency by flying at a lower altitude, but it involves risk. The reconnaissance vehicle can also improve the data resolution by flying low, but the improvement of efficiency is not as great as the water tanker. In such a case a plausible plan is to allow the water tanker to take a large portion of risk by flying low, while keeping the reconnaissance vehicle at a high altitude to avoid risk. This is because the utility of taking risk (i.e. flying low) is greater for the water vehicle than for the reconnaissance vehicle.

Then, the question is how to find the optimal risk allocation between multiple vehicles in a distributed manner.

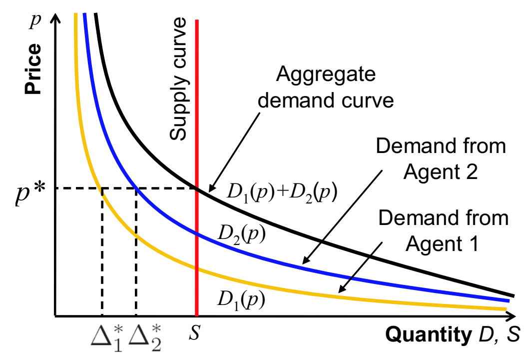

Figure 3: Market-based risk allocation in a system with two agents. Note that we followed the economics convention of placing the price on the vertical axis. The equilibrium price is $p^\star$, and the optimal risk allocation is $\Delta_1^\star = D_1(p^\star)$ for Agent 1 and $\Delta_2^\star = D_2(p^\star)$ for Agent 2.

Tâtonnement: market-based risk allocation Our approach is to use the market-based mechanism to optimize the risk allocation between agents. In a computational market each agent demands for risk in order to improve its own performance. However, it cannot take risk for free; it has to purchase it from the market at a given price. It demands more risk when the price is low. On the other hand, the supply of risk is constant, since the upper-bound of total risk is given.

Agents are price takers. Therefore the demand is a function of the price of risk (*demand curve*). Each agent has a different demand curve according to its sensitivity to risk. This setting corresponds to the perfectly competitive economy, where no agent has monopoly power.

The price must be adjusted so that the total demand (*aggregate demand*) becomes equal to the supply. The equilibrium price can be found by a simple iterative process as follows:

- Increase the price if the aggregate demand exceeds the supply,
- Decrease the price if the supply exceeds the aggregate demand,
- Repeat until the demand and the supply are balanced.

This iterative price adjustment process is called *tâtonnement* or *Walrasian auction* [16].

Figure 3 gives the graphical interpretation of the market-based risk allocation in a system with two agents. In this example Agent 2 has higher demand for risk than Agent 1, since Agent 2 has more utility of taking risk than Agent 1. The aggregate demand curve is obtained by adding the two demand curves horizontally. The supply curve is a vertical line since it is constant. The equilibrium price $p^\star$ lies at the intersection of the aggregate demand curve and the supply curve. The optimal risk allocation for the two agents are their demands for risk at the equilibrium price ($\Delta_1^\star$ and $\Delta_2^\star$ in Figure 3).

It is proven in a later section that the performance of the entire system is maximized at the equilibrium price, although each agent only maximizes its own utility. The only information that needs to be exchanged between agents is price and demand. These are desirable features for distributed systems.

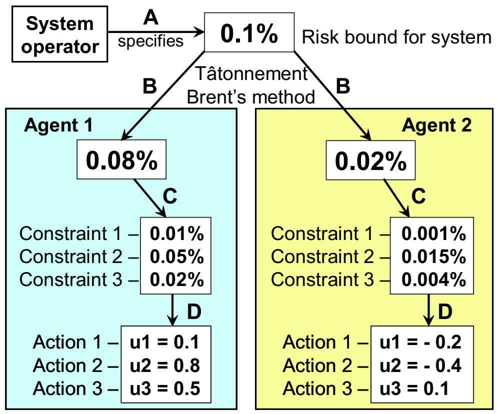

Figure 4: Structure of Market-based Iterative Risk Allocation (MIRA) algorithm. **A**: System operator specifies the risk bound for system. **B**: It is broken down to the risk bounds for each agent. **C**: Each agent optimizes internal risk allocation. **D**: Each agent optimizes the action plan according to the risk allocation.

MIRA Our proposed algorithm, MIRA (Market-based Iterative Risk Allocation), optimizes risk allocation between agents, internal risk allocation of each agent, and action sequences of each agent concurrently in a distributed manner. Figure 4 shows the structure of MIRA. The risk bound is given by the system operator (A). At the top level of the algorithm it optimizes the risk allocation between agents by tâtonnement (B). At the middle level each agent optimizes internal risk allocation (C). At the bottom level the action sequence is optimized according to the risk allocation (D).

The next section presents the related work. The following three sections explains MIRA algorithm in bottom-up order.

1.4 Related Work

The basis of this work is the model-based plan executive called *Sulu* [10]. It takes a high-level plan (*Qualitative State Plan* or *QSP*) as an input, and outputs a low-level continous action sequence. Two limitations of Sulu are that it does not consider uncertainty, and it is a centralized planner. Our final goal is to create a distributed model-based plan executive for multi-agent systems under uncertainty. The work in this paper is an important stepping stone to the goal.

Planning under uncertainty with risk bound (called *chance constraint*) is intensively researched in the robust model predictive control (RMPC) community [2]. Due to the difficulty of handling the chance constraint analytically, past work used either a very conservative bound that resulted in large suboptimality [18][6][12], or a sample-based method [4] that is computationally inefficient. We introduced the concept of risk allocation that decomposes the chance constraint into multiple atomic chance constraints [13], and developed Iterative Risk Allocation (IRA) algorithm that can optimize risk allocation efficiently, with substantially smaller suboptimality than the past work [14].

Market-based approach has recently been recognized as an effective tool for decentralized multi-agent systems in AI community [20][11]. Although tâtonnement has drawn less attention than auctions, it has been successfully applied to various problems such as the distribution of heating energy in an office building [19], and resource allocation in communication networks [9]. The convergence of tâtonnement has been an issue in economics for a long time; with a simple linear price update rule, it can only be guaranteed under a quite restrictive condition [16]. Fortunately, since we are working in a computational economy, we do not have to stick to the simple price update rule that is natural in a real-world economy; instead, we use Brent's method, which is guaranteed to converge with superlinear convergence rate [1].

2 Risk Allocation for a Single-agent System

We will first briefly review the mathematical formulation of risk allocation. Our focus on this paper is a problem with continuous state space, although the concept of risk allocation can be used for discrete/hybrid systems [13]. This section corresponds to the bottom and middle level optimization of MIRA (Figure 4).

2.1 Formulation

Action planning under uncertainty Time evolved goals in Qualitative State Plan can be encoded into a set of linear constraints, and the optimal sequence of actions are found by solving a mixed integer linear programming [10]. Based on this work, we formulate the probabilistic planning problems with chance constraints as constrained convex optimization problems as follows:

$$\begin{aligned} \min_{\boldsymbol{u}_{1:T}} \quad & J(\boldsymbol{u}_{1:T}) && (1) \\ \text{s.t.} \quad & \forall_t \quad \boldsymbol{x}_{t+1} = \boldsymbol{A}x_t + \boldsymbol{B}u_t + \boldsymbol{w}_t && (2) \\ & \forall_t \quad \boldsymbol{u}_{\min} \leq \boldsymbol{u}_t \leq \boldsymbol{u}_{\max} && (3) \\ & \Pr\left[\bigwedge_{t=0}^{T} \bigwedge_{n=1}^{N_t} g_{t,n}(\boldsymbol{x}_t) \leq 0 \right] \geq 1 - \Delta && (4) \end{aligned}$$

where $\boldsymbol{x}_t$, $\boldsymbol{u}_t$, and $\boldsymbol{w}_t$ are the state vector, action (control input) vector, and disturbance respectively. The subscript indicates the time step. The probabilistic distribution of the disturbance $\boldsymbol{w}_t$ is known, and the distribution of the initial state $\boldsymbol{x}_0$ is also given.

We assume a discrete-time linear dynamic system defined by Eq.(2) with constraints on actions Eq.(3). We consider a fixed planning window $1 \leq t \leq T$. In each time step there are N_t constraints $g_{t,n}(\boldsymbol{x}_t) \leq 0$. Eq.(4) is the chance constraint. Since violation of any constraint at any time step is regarded as a mission failure, the probability of satisfying all constraints at all time steps must be $1 - \Delta$, where Δ is the upper bound of the probability of failure (risk bound).

The problem is to find the optimal sequence of actions $\boldsymbol{u}_{1:T} := [\boldsymbol{u}_1 \cdots \boldsymbol{u}_T]^T$ that minimizes the cost J(Eq.(1)).

The risk bound Δ is a constant specified by the system operator. The system minimizes the cost (i.e. maximizes the performance) within this given risk bound. In other words, the system operator can adjust the risk averseness of the system by tuning the risk bound Δ.

We assume that $J(\cdot)$ and $g(\cdot)$ are convex functions. A problem with non-convex feasible state space can also be formulated in this framework by introducing integer variables, and solved by branch-and-bound algorithm.

Decomposition of chance constraint The chance constraint Eq.(4) is hard to be evaluated since it involves a probability defined on a multi-dimensional distribution. It can be decomposed into multiple atomic chance constraints that only involve single-dimensional distribution using the following Boole's inequality or union bound:

$$\Pr\left[\bigcup_i A_i\right] \leq \sum_i \Pr\left[A_i\right] \tag{5}$$

Observe that, using Boole's inequality Eq.(5), following condition Eq.(6), together with Eq.(7), implies the original chance constraint Eq.(4).

$$\forall_{(t,n)} \Pr\left[g_{t,n}(\boldsymbol{x}_t) \leq 0\right] \geq 1 - \delta_t^n \tag{6}$$

$$\sum_{t=1}^{T} \sum_{n=1}^{N_t} \delta_t^n \leq \Delta \tag{7}$$

Therefore, the original chance constraint Eq.(4) can be replaced with Eq.(6) and Eq.(7). Since we have introduced new variables δ_t^n, the cost J in Eq.(1) needs to be optimized over δ_t^n as well as the sequence of actions $\boldsymbol{u}_{1:T}$:

$$\min_{\boldsymbol{\delta}, \boldsymbol{u}_{1:T}} J(\boldsymbol{u}_{1:T}) \tag{8}$$

where

$$\boldsymbol{\delta} = \left[\delta_0^1 \; \delta_0^2 \; \cdots \delta_T^{N_T - 1} \; \delta_T^{N_T}\right]^T.$$

We now have the revised constrained optimization problem defined by Eq.(8) with constraints Eq.(2)(3)(6)(7).

2.2 Risk allocation

What is the meaning behind these mathematical manipulations? Here is the answer: the newly introduced variable $\boldsymbol{\delta}$ is the mathematical representation of *risk allocation*. Now each single constraint at each time step has its own risk bound δ_t^n (Eq.(6)); in other words, δ_t^n is the amount of risk allocated to nth constraint at tth time step. Eq.(7) states that the total amount of risk is upper-bounded by the original risk bound Δ; therefore risk is regarded as a resource with total amount Δ. In order to obtain the maximum performance (minimum cost), the risk allocation needs to be optimized(Eq.(8)), just like the resource allocation problem.

The risk allocation optimization problem can be solved efficiently by a two-stage optimization algorithm called Iterative Risk Allocation [14]. Alternatively it can also be solved by single shot optimization [5]. We employ the latter approach in this work.

3 Distributed Risk Allocation for Multi-agent Systems

This section formulates the top level optimization in MIRA, shown in Figure 4.

We will first formulate the planning problem under uncertainty for multi-agent systems in a centralized manner, and then derive the distributed formulation from there using KKT conditions. We will then observe that the economical concepts such as price, demand, and supply naturally show up through the mathematical manipulation.

3.1 Formulation

Planning under risk for multi-agent system In a multi-agent system such as the UAV fire fighting system illustrated in Figure 2, a failure of one agent leads to a failure of entire system. In a manned system, loss of one crew member is regarded as a failure of mission. Therefore the system operator wants to set an upper-bound on the probability of having at least one agent fail.

With the same discussion as in the previous section, the following bound is obtained by using Boole's inequality Eq.(5):

$$\sum_{i=1}^{I} \Delta_i \leq S \tag{9}$$

where Δ_i is the upper bound on the probability of failure of the ith agent, I is the number of agents in the system, and S is the upper bound on the probability of failure of the entire system (i.e. total amount of risk the entire system can take). Note that Δ was a given constant in the single-agent case (Eq.(4) or (7)), but now each Δ_i is a decision variable, while S is the new given constant, which is specified by the system operator.

Then how is the performance of the entire system defined? We consider a simple case where the performance (cost) of the system is the sum of the performance (cost) of all agents:

$$J_{sys} = \sum_i J_i(\boldsymbol{u}_{i,1:T}) \tag{10}$$

Therefore the planning problem under risk for multi-agent systems is formulated as the constrained optimization problem to minimize Eq.(10), subject to the constraints Eq.(2)(3)(6)(7), and (9).

This formulation is a little messy. To clean it up, we define a function $J_i^\star(\Delta_i)$, which is equal to the minimized cost for the ith agent obtained by solving the constrained optimization problem for a single agent Eq.(8)(2)(3)(6)(7) given Δ_i:

$$J_i^\star(\Delta_i) = J(\boldsymbol{u}^\star_{i,1:T}) \tag{11}$$

where $\boldsymbol{u}^\star_{1:T}$ is the solution to the single agent optimization problem) given Δ_i.

Using $J_i^\star(\Delta_i)$, the optimization problem can be rewritten in a simple form as follows:

$$\min_{\Delta_{1:I}} \quad \sum_{i=1}^{I} J_i^\star(\Delta_i) \tag{12}$$

$$\text{s.t.} \quad \sum_i \Delta_i \leq S \tag{13}$$

An important fact is that function $J_i^\star(\Delta_i)$ is a convex function if the original cost function $J(\boldsymbol{u}_{1:T})$ is convex (which is our assumption) and the distribution of the disturbance $\boldsymbol{w}_t$ is quasi-concave with its maximum at the mean, such as Gaussian distribution. See Appendix for the proof.

This formulation describes a centralized planning problem since the action sequences and risk allocations of all agents are planned in one optimization problem. We will next derive the distributed formulation using KKT conditions.

Distributed planning The KKT conditions of the optimization problem Eq.(12)(13) are [1]

$$\left.\frac{dJ_i^\star}{d\Delta_i}\right|_{\Delta_i^\star} + p = 0 \tag{14}$$

$$\sum_i \Delta_i^\star \leq S \tag{13}$$

$$p \geq 0 \tag{15}$$

$$p\left(\sum_i \Delta_i^\star - S\right) = 0 \tag{16}$$

where p is the Lagrange multiplier corresponding to the constraint Eq.(13). This is the necessary and sufficient condition for optimality since $J_i^\star(\Delta_i)$ is convex.

Observe that Eq.(14) is also the optimality condition for the following unconstrained optimization problem:

$$\min_{\Delta_i} J_i^\star(\Delta_i) + p\Delta_i \tag{17}$$

Therefore solving the optimization problem Eq.(12) and (13) is equivalent to solving I independent optimization problems Eq.(17) with common parameter p, which is determined by Eq.(13), (15), and (16). Since Eq.(17) contains only the variables related to ith agent, it can be solved by each agent in a distributed manner.

Note: It is often not possible, and not necessary as well, to obtain the function $J_i^\star(\Delta_i)$ in a closed form; in practice $J_i^\star(\Delta_i)$ is evaluated simply by solving the optimization problem Eq.(8)(2)(3)(6)(7), with an extra term $p\Delta_i$ added to the objective function Eq.(8).

3.2 Economic Interpretation

The interpretation of these mathematical manipulations becomes clear by regarding the Lagrange multiplier p as the *price of risk*. Each agent can reduce the cost (i.e. improve the performance) by taking more risk Δ_i, but not for free. Note that a new term $p\Delta_i$ is added to the cost function Eq.(17). This is what the agent has to pay to take the amount of risk Δ_i. The agent must find the optimal amount of risk $D_i(p)$ to minimize the cost plus payment, by solving the optimization problem Eq.(17) with a given price p:

$$D_i(p) = \arg\min_{\Delta_i} J_i^\star(\Delta_i) + p\Delta_i.$$

In other words, $D_i(p)$ is the amount of risk the ith agent wants to take given the price of risk p. Therefore $D_i(p)$ can be interpreted as the ith agent's *demand for risk*. On the other hand, the total amount of risk S can be interpreted as the *supply of risk*.

In order to obtain the optimal plan, we need to find the optimal price $p^\star$ that satisfies the KKT conditions Eq.(13), (15), and (16), with the optimal demands at the price $\Delta_i^\star = D_i(p^\star)$. Such price $p^\star$ is also called as the equilibrium price.

The condition Eq.(16) illustrates the relation between the optimal price, demand, and supply; in the usual case where the optimal price is positive $p^\star > 0$, the aggregate

[1] We assume the differentiability of $J_i^\star(\Delta_i)$ here; in fact, since $J_i^\star(\Delta_i)$ is a convex function, it is continuous on C and differentiable at all but at most countably many points; we can obtain the same result for the point where it is not differentiable by using extended KKT condition with subgradient.

demand $\sum_i \Delta_i^\star$ must be equal to the supply S; in a special case where the supply always exceeds the demand for all $p \geq 0$, the optimal price is zero $p^\star = 0$. If the aggregate demand always exceeds the supply for all $p \geq 0$, there is no solution that satisfies the primal feasibility condition (13), and hence the problem is infeasible. See Figure 3 for the graphical interpretation.

The next section discusses how to find such equilibrium price $p^\star$.

4 Tâtonnement: Price adjustment mechanism

This section provides the algorithm that solves the top level optimization in MIRA, shown in Figure 4.

In the real world economy the demand for a good is a monotonically decreasing function of price (people want more if price is less). This is also the case in our computational economy. By differentiating Eq.(14),

$$\frac{dp}{dD_i} = -\left.\frac{d^2 J_i^\star}{d\Delta_i^2}\right|_{\Delta_i = D_i} \leq 0 \tag{18}$$

We used the fact that $J_i^\star(\Delta_i)$ is a convex function. (See Appendix for the proof.)

This decreasing monotonicity of demand curve justifies the use of tâtonnement; iteratively increase the price if the aggregate demand exceeds the supply and decrease the price if the supply exceeds the aggregate demand, until the demand and supply are balanced.

Algorithm 1 shows the flow of tâtonnement. At the begging of each iteration, the price is announced by an auctioneer (Line 4). Then each agent bids the quantity of risk they would like to purchase (i.e. demand for risk) at the price. Auctioneer adjusts the price according to the excess demand (supply) (Line 9 and 11), and announces it at the beginning of the next iteration. This process is repeated until the demand and supply are balanced (Line 7). Mathematically tâtonnement can be seen as a root-finding process.

Then what should the specific price update rule be? In the following subsections we will investigate two update rules: linear increment and Brent's method [1].

Algorithm 1 Tâtonnement (fixed supply)

1: Fix S; //Total supply of risk
2: Initialize p; //Price of risk
3: **loop**
4:   Auctioneer announces p;
5:   Each agent submits its demand for risk $D_i(p)$;
6:   **if** $|\sum_i D_i^\star(p) - S| < \epsilon$ **then**
7:     **break;**
8:   **else if** $\sum_i D_i^\star(p) - S > 0$ **then**
9:     Increase p;
10:   **else if** $\sum_i D_i^\star(p) - S < 0$ **then**
11:     Decrease p;
12:   **end if**
13: **end loop**

4.1 Linear Price Increment

The following simple price update rule is most intensively researched in economics context:

$$p_{k+1} = \max\left\{p_k + \lambda\left(\sum_i D_i(p_k) - S\right), 0\right\} \tag{19}$$

The price is guaranteed to converge for *sufficiently small* $\lambda > 0$, if a condition called gross substitutability is satisfied [16]. In our case where there are only two goods exchanged in the market (risk and money), gross substitutability is implied by the decreasing monotonicity of the demand function. However, practically, the bound for λ is very hard to obtain a priori. Its slow convergence is also a serious issue for our purpose.

4.2 Brent's method

As far as we know there is no research that applies standard root-finding algorithms such as Newton's method or Brent's method for tâtonnement. This is probably because the price adjustment with such complex methods is not a natural model of the real-world economy. Nonetheless, we do not have to care about such limitation in our computational economy, since our objective is to obtain the optimal plan, not to model the real-world economy!

We use Brent's method for its quick and guaranteed convergence [1]. Another important feature of Brent's method is that it does not require the derivative $D_i(p)$, which is very hard to obtain.

4.3 MIRA algorithm

We have finished explaining all pieces of the MIRA algorithm shown in Figure 4. At the top level the risk allocation between agents is optimized through tâtonnement. We use Brent's method to update the price in each iteration of tâtonnement. Each agent submits its demand at a given price in each iteration; the demand is computed by solving optimization problem Eq.(17), or equivalently, Eq.(8)(2)(3)(6)(7), with an extra term $p\Delta_i$ added to the objective function Eq.(8). The solution of the optimization problem gives the risk allocation in each agent and the action sequence (middle and bottom level of Figure 4).

5 Simulation

5.1 Validity

To check the validity of the proposed algorithm, we tested it on the multi-UAV fire-fighting scenario (Figure 2). Figure 5 shows the simulation result. Two vehicles fly at the constant horizontal speed, starting from $d = 0$ at altitude 0.5. The mission is to extinguish the fire at $d = 6, 7$. Both vehicles minimize the flight altitude above the fire, although the water tanker is given 100 times more penalty (cost) of flying at high altitude than the reconnaissance vehicle. Both have uncertainty in altitude, so flying at lower altitude involves more risk. The total risk must be less than 0.1%.

The optimal plan allocates 99.2% of the total risk to the water tanker while only 0.8% to the reconnaissance vehicle. This is because the utility of taking risk (i.e. flying

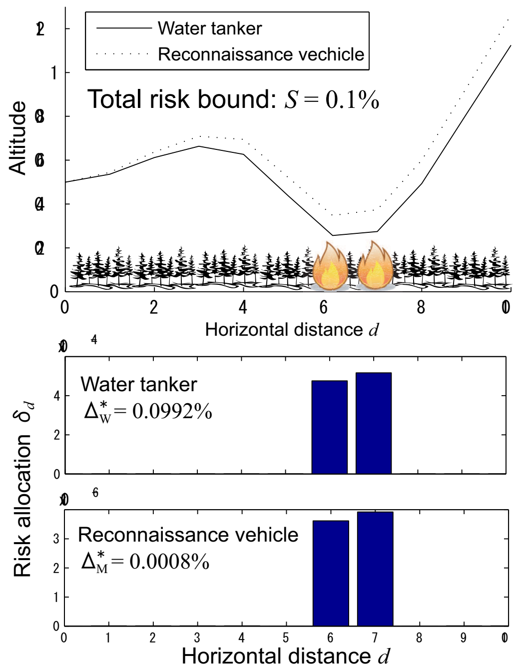

Figure 5: Simulation result of flight altitude planning problem for multi-UAV fire-fighting scenario (see Figure 2). **Upper graph:** The water tanker is allocated 99.2% of the total risk; as a result, it is allowed to fly lower than the reconnaissance vehicle. **Lower graphs:** Both vehicles take most of the allocated risk above the fire ($d = 6, 7$); they fly as high as possible before and after the fire, since there is no benefit of taking risk at those places .

low) is larger for the water tanker than for the reconnaissance vehicle. As a result, the water tanker flies at lower altitude.

Both vehicles optimize the internal risk allocation as well. For example, the water tanker takes 99.9% of the allocated risk above the fire, at $d = 6$ and 7 (the middle graph in Figure 5).

The optimal action sequence is planned according to the risk allocation; both vehicles dive before the fire, and climb as fast as possible after they pass the fire (the top graph in Figure 5). This is because there is no merit of conducting risky low-altitude flight before and after the fire.

These results conform with the intuition. The resulted total cost is 53.77, which is within 0.01% difference compared to the cost computed by the centralized algorithm.

5.2 Efficiency

In order to evaluate the efficiency of the MIRA algorithm, the computation time of the following three algorithms are compared:

1. Centralized optimization

2. Distributed optimization (tâtonnement) with linear price increment

Table 1: Comparison of the computation time of three optimization algorithms. Values are the average of 10 runs with randomly generated constraints.

Number of agents	Computation time [sec]		
	Centralized	Distributed (linear increment)	MIRA
2	13.9	80.6	6.4
4	63.8	540.5	18.1
8	318.5	797.8	37.5

3. MIRA: distributed optimization (tâtonnement) with Brent's method

Table 1 shows the result. The three algorithms are tested with different problem sizes - two, four, and eight agents. Each algorithm is run 10 times for each problem size with randomly generated constraints. The average running time is shown in the table.

The computation time of the centralized optimization algorithm quickly grows as the problem size increases. Distributed optimization with linear price increment is even slower than the centralized algorithm, although the growth rate of computation time is slower.

MIRA, the proposed algorithm, outperforms the other two for all problem sizes. The advantage of MIRA becomes clearer as the problem size increases.

A counterintuitive phenomenon observed in the result is that the distributed algorithms (MIRA and distributed optimization with linear increment) also slow down for large problems, although not as significantly as the centralized optimization. This is mainly because the iterations of tâtonnement must be synchronized among all agents. When each agent computes its demand for risk by solving the non-linear optimization problem, the computation time diverges from agent to agent, and from situation to situation. In each iteration of tâtonnement, all agents must wait until the slowest agent finish computing its demand. As a result, tâtonnement process slows down for large problems as the expected computation time of the slowest agent grows.

5.3 Used Parameters

The horizontal speed of the vehicles is 1 per time step. Hence, $d = t$. The planning window is $1 \le t \le 10$. Other parameters are set as follows:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, \boldsymbol{x}_0 = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix},$$
$$u_{\min} = -0.2, u_{\max} = 0.2, \quad g_t(\boldsymbol{x}_t) = -\begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_t + l_t$$

$\boldsymbol{w}_t$ is sampled from zero-mean Gaussian distribution with variance

$$\Sigma_w = \begin{bmatrix} 0.001 & 0 \\ 0 & 0 \end{bmatrix}.$$

l_t is the ground level at t. It is set at zero in the fire-fighter UAV scenario, and randomly generated for the evaluation of computation time. The cost functions are

$$\begin{aligned} J_W &= E\left[\begin{bmatrix} 100 & 0 \end{bmatrix} (\boldsymbol{x}_{6,W} + x_{7,W})\right] \\ J_R &= E\left[\begin{bmatrix} 1 & 0 \end{bmatrix} (\boldsymbol{x}_{6,R} + x_{7,R})\right] \end{aligned}$$

for the fire-fighter UAV scenario (subscript W and R indicate the water tanker and the reconnaissance vehicle respectively), and

$$J_i = E\left[\left[\begin{array}{cc} 1 & 0 \end{array}\right](\sum_{t=1}^{10} \boldsymbol{x}_{t,i})\right].$$

for the evaluation of computation time. Note that the expectation of $\boldsymbol{x}_t$ is a function of $\boldsymbol{u}_{1:t-1}$. Therefore J is a function of $\boldsymbol{u}_{1:T}$.

5.4 Computation and Programming Environment

The program was written in Matlab, and the middle and bottom level optimization problems were solved by SNOPT. The top level optimization (Brent's method) is solved by Matlab fzero function. Simulations were conducted on a machine with Intel(R) Core(TM) i7 CPU clocked at 2.67 GHz and 8GB RAM.

6 Conclusion

We have developed Market-based Iterative Risk Allocation (MIRA), a multi-agent optimal planning algorithm that operates within user-specified risk bounds. It was built upon the concept of risk allocation. The three key innovations that enabled MIRA were:

1. Extension of the concept of risk allocation to multi-agent system
2. Derivation of distributed optimization method for multi-agent risk allocation
3. Introduction of Brent's method to tâtonnement as a price update rule.

The simulation result showed that MIRA can optimize the action sequence of the multi-agent system by optimally distributing risk. It achieved substantial speed-up compared to centralized optimization approach, particularly in a large problem.

Appendix

A Proof of the Convexity of Cost as a Function of Risk

Robust model-predictive control (RMPC) with joint chance constraint is a cost minimization problem that applies to dynamic systems under stochastic uncertainty, and includes a risk constraint specified as an upper bound on the probability of failure (risk bound). The optimized cost ($J_i^\star(\Delta_i)$, defined in Eq.11) can be viewed as the function over the risk bound. This section proves that the optimized cost is a convex function of the risk bound.

A.1 Problem Statement

We review the original problem formulation Eq.(1)-(4), which is a RMPC with a joint chance constraint:

$$\begin{aligned}
\min_{\boldsymbol{U}} \quad & J(\boldsymbol{u}_{1:T}) \\
\text{s.t.} \quad & \forall_t \quad \boldsymbol{x}_{t+1} = \boldsymbol{A}x_t + \boldsymbol{B}u_t + \boldsymbol{w}_t \\
& \forall_t \quad \boldsymbol{u}_{\min} \le \boldsymbol{u}_t \le \boldsymbol{u}_{\max} \\
& \Pr\left[\bigwedge_{t=0}^{T}\bigwedge_{n=1}^{N_t} g_{t,n}(\boldsymbol{x}_t) \le 0\right] \ge 1-\Delta
\end{aligned}$$

where $\boldsymbol{x}_t$, $\boldsymbol{u}_t$, and $\boldsymbol{w}_t$ are the state vector, control input, and disturbance, respectively, and subscripts indicate the time step. The disturbance $\boldsymbol{w}_t$ and the the initial state $\boldsymbol{x}_0$ has Gaussian distributions with known mean and variance. We assume that J and $g_{t,n}$ are convex functions.

Our past work[14] showed that this problem can be reformulated as follows:

$$\min_{\boldsymbol{U},\boldsymbol{\delta}} \quad J(\boldsymbol{u}_{1:T}) \tag{20}$$

$$\text{s.t.} \quad \forall_t \quad \bar{\boldsymbol{x}}_{t+1} = \boldsymbol{A}\bar{\boldsymbol{x}}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{w}_t \tag{21}$$

$$\forall_t \quad \boldsymbol{u}_{\min} \le \boldsymbol{u}_t \le \boldsymbol{u}_{\max} \tag{22}$$

$$\forall_{t,n} \quad g'_{t,n}(\bar{\boldsymbol{x}}_t, \delta_t^n) \le 0 \tag{23}$$

$$\sum_{t,n} \delta_t^n \le \Delta \tag{24}$$

where g' is a convex function of $\bar{\boldsymbol{x}}_t$ and δ_t^n[14]. The following notation is used throughout this report for convenience:

$$\boldsymbol{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_T]^T, \qquad \bar{\boldsymbol{X}} = [\bar{\boldsymbol{x}}_0 \cdots \bar{\boldsymbol{x}}_T]^T \qquad \boldsymbol{\delta} = \left[\delta_1^1 \cdots \delta_T^{N_t}\right]^T.$$

Let $J^\star(\Delta)$ be the optimal value of the objective function, given the parameter Δ;

$$J^\star(\Delta) = J(\boldsymbol{U}^\star; \Delta) \tag{25}$$

where $\boldsymbol{U}^\star$ is the optimal solution for Δ.

The problem to solve is to show that $J^\star(\Delta)$ is a convex function of Δ.

A.2 Proof of Convexity

Assume that the constrained optimization problem (20)-(24) is feasible for Δ_1 and Δ_2. Let $(\boldsymbol{U}_1^\star, \bar{\boldsymbol{X}}_1^\star, \boldsymbol{\delta}_1^\star)$ and $(\boldsymbol{U}_2^\star, \bar{\boldsymbol{X}}_2^\star, \boldsymbol{\delta}_2^\star)$ be the optimal solution for Δ_1 and Δ_2 respectively. Since (21) and (24) are linear constraints and $g'_{t,n}$ in (23) is a convex function, $(\lambda \boldsymbol{U}_1^\star + (1-\lambda)\boldsymbol{U}_2^\star, \lambda \bar{\boldsymbol{X}}_1^\star + (1-\lambda)\bar{\boldsymbol{X}}_2^\star, \lambda \boldsymbol{\delta}_1^\star + (1-\lambda)\boldsymbol{\delta}_2^\star)$ satisfies the constraints (21)-(24) for $\Delta = \lambda\Delta_1 + (1-\lambda)\Delta_2$ for all $0 \le \lambda \le 1$.

Then let $(\boldsymbol{U}_\lambda^\star, \bar{\boldsymbol{X}}_\lambda^\star, \boldsymbol{\delta}_\lambda^\star)$ be the optimal solution for $\Delta = \lambda\Delta_1 + (1-\lambda)\Delta_2$. Since this is the optimal solution, it follows that,

$$\begin{aligned} J^\star(\lambda\Delta_1 + (1-\lambda)\Delta_2) &= J(\boldsymbol{U}_\lambda^\star) \\ &\le J(\lambda \boldsymbol{U}_1^\star + (1-\lambda)\boldsymbol{U}_2^\star). \end{aligned}$$

Since J is a convex function,

$$\begin{aligned} J(\lambda \boldsymbol{U}_1^\star + (1-\lambda)\boldsymbol{U}_2^\star) &\le \lambda J(\boldsymbol{U}_1^\star) + (1-\lambda)J(\boldsymbol{U}_2^\star) \\ &= \lambda J^\star(\Delta_1) + (1-\lambda)J^\star(\Delta_2). \end{aligned}$$

Therefore, for all $0 \le \lambda \le 1$,

$$J^\star(\lambda\Delta_1 + (1-\lambda)\Delta_2) \le \lambda J^\star(\Delta_1) + (1-\lambda)J^\star(\Delta_2),$$

and thus $J^\star$ is a convex function. Q.E.D.

Acknowledgments

Thanks to Michael Kerstetter, Scott Smith and the Boeing Company for their support. Hirokazu Ishise and Yuichi Yamamoto provided valuable comments on economics.

References

[1] Kendall E. Atkinson. *An Introduction to Numerical Analysis, Second Edition*. John Wiley & Sons, 1989.

[2] V. Tsachouridis B. Kouvaritakis, M. Cannon. Recent developments in stochastic mpc and sustainable development. *Annual Reviews in Control*, 28:23–35, 2004.

[3] K. Bell, A. I. Coles, M. Fox, D. Long, and A. J. Smith. The application of planning to power substation voltage control. In *Proceedings of ICAPS Workshop on Scheduling and Planning Applications*, 2008.

[4] Lars Blackmore. A probabilistic particle control approach to optimal, robust predictive control. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2006.

[5] Lars Blackmore. Convex chance constrained predictive control without sampling. Technical report, NASA JPL Document D-44784, 2008.

[6] Lars Blackmore, Hui Li, and Brian C. Williams. A probabilistic approach to optimal robust path planning with obstacles. In *Proceedings of American Control Conference*, 2006.

[7] E. Fiorelli, N.E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D.M. Fratantoni. Multi-auv control and adaptive sampling in monterey bay. In *Proceedings of IEEE/OES Autonomous Underwater Vehicles 2004: Workshop on Multiple AUV Operations (AUV04)*, 2004.

[8] Andreas G. Hofmann and Brian C. Williams. Robust execution of temporally flexible plans for bipedal walking devices. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-06)*, 2006.

[9] F.P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[10] Thomas Léauté and B. C. Williams. Coordinating agile systems through the model-based execution of temporal plans. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, 2005.

[11] Jeffrey K. MacKie-Mason, Anna Osepayshvili, Daniel M. Reeves, and Michael P. Wellman. Price prediction strategies for market-based scheduling. In *Proceedings of 14th International Conference on Automated Planning and Scheduling (ICAPS-04)*, 2004.

[12] Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17:969–996, 2006.

[13] Masahiro Ono and Brian C. Williams. An efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-08)*, 2008.

[14] Masahiro Ono and Brian C. Williams. Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint. In *Proceedings of 47th IEEE Conference on Decision and Control (CDC-08)*, 2008.

[15] Hairong Qi, Wenjuan Zhang, and L. M. Tolbert. A resilient real-time agent-based system for a reconfigurable power grid. In *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, 2005.

[16] Jan Tuinstra, editor. *Price Dynamics in Equilibrium Models: The Search for Equilibrium and the Emergence of Endogenous Fluctuations*. Kluwer Academic Publishers, Norwell, MA, 2000.

[17] Roy M. Turner and Elise H. Turner. A two-level, protocol-based approach to controlling autonomousoceanographic sampling networks. *IEEE Journal of Oceanic Engineering*, 26:654–666, 2001.

[18] Dennis Harald van Hessem. *Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation*. PhD thesis, Delft University of Technology, 2004.

[19] Holger Voos. Agent-based distributed resource allocation in technical dynamic systems. In *Proceedings of IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS 2006)*, 2006.

[20] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.

[21] Pei Zhang, Liang Min, L. Hopkins, and B. Fardanesh. Utility experience performing probabilistic risk assessment for operational planning. In *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, 2007.