Working Paper

A homotopy method for equilibrium programming under uncertainty

Charles H. Rosa

WP-96-044
April 1996

IIASA International Institute for Applied Systems Analysis ◻ A-2361 Laxenburg ◻ Austria
Telephone: +43 2236 807 ◻ Fax: +43 2236 71313 ◻ E-Mail: info@iiasa.ac.at

A homotopy method for equilibrium programming under uncertainty

Charles H. Rosa

WP-96-044
April 1996

International Institute for Applied Systems Analysis ◘ A-2361 Laxenburg ◘ Austria

Telephone: +43 2236 807 ◘ Fax: +43 2236 71313 ◘ E-Mail: info@iiasa.ac.at

Abstract

We consider a homotopy method for solving stochastic Nash equilibrium models. The algorithm works by following, via a predictor-corrector method, the one-dimensional manifold of the homotopy constructed to connect the systems of equations describing the solution set of the scenario equilibrium model (no nonanticipativity constraints) and the stochastic equilibrium model. The predictor and corrector phases of this homotopy method require the usual solutions of large linear systems, a computationally expensive task, which we render less difficult through our use of Jacobi techniques designed to take advantage of the problem's near separability across scenarios.

Key words: equilibrium programming, decomposition, homotopy methods, stochastic programming

A homotopy method for equilibrium programming under uncertainty

Charles H. Rosa[**]

1 Introduction

We consider a class of equilibrium programming problems that incorporate a special form of uncertainty. The special form of the problem, along with a potential solution technique, has already been introduced in [Ros96]. In this paper, we will review the problem description, and then consider another solution methodology. The solution methodology will be based on the powerful path-following idea implicit in homotopy methods [Dav53], [Eav72], [EaS76], [GaG78], [GaZ79b], [GaZ79a], [GaZ79a], [GaZ81], [OrR70], [Sca67]. These methods can be used to globally solve smooth nonlinear systems of equations of considerable size and complexity. We will find that our problem is a natural candidate for these methods because there is a homotopy that connects the solution sets of the separable scenario equilibrium problems and the stochastic equilibrium problem. Furthermore, we will find that we can take advantage of the "almost" separability of the scenario components of the homotopy to follow the path (using tangential predictor steps) and stay within an arbitrary tube around the path (using Newton corrector steps) in an efficient manner that uses Jacobi iterative techniques to more efficiently solve the large linear systems (i.e., separably across scenarios).

The remainder of this paper will be organized as follows. In §2 we will discuss the stochastic multistage equilibrium programming problem. In §3 we will review homotopy methods and how they are used to solve systems of equations. Finally, in §4 we will discuss how homotopy methods, and, in particular, the solution of the large linear systems of equations, specialize and become easier when applied to our problem.

2 Problem statement

We consider the structure of multistage stochastic equilibrium programming problems.

[*]This research was supported in part by an appointment to the Global Change Distinguished Postdoctoral Fellowship Program sponsored by the U.S. Department of Energy, Office of Health and Environmental Research, and administered by the Oak Ridge Institute for Science and Education.

[**]The author is a postdoctoral fellow at Argonne National Laboratory, 9700 S. Cass Ave., Argonne, IL 60439, USA (rosa@dis.anl.gov).

Let the set of all agents in the problem be denoted as $\Omega = \{1, 2, \ldots, A\}$, the time parameter t be an element of $\{1, \ldots, T\}$, and the set of all *scenarios* $S = \{1, \ldots, N\}$. We can then associate with each scenario path j the following objects from the a^{th} agent's decision problem: the decision subvector

$$x_a^j = (x_a^j(1), x_a^j(2), \ldots, x_a^j(T)) \in \mathbb{R}^{q_a^1} \times \cdots \times \mathbb{R}^{q_a^T},$$

the closed concave utility function $u_a^j(x_a^j, x_{\Omega_{-a}}^j) : \mathbb{R}^{q_a^1} \times \cdots \times \mathbb{R}^{q_a^T} \times \prod_{b \in \Omega_{-a}} (\mathbb{R}^{q_b^1} \times \cdots \times \mathbb{R}^{q_b^T}) \rightarrow (-\infty, +\infty]$ where Ω_{-a} represents the set of all agents other than a, and the probability p_j. We set $q_a = q_a^1 + \ldots + q_a^T$, $n_a = N q_a$ and $n = n_1 + \ldots + n_A$. To efficiently model the uncertainty in our problem, we find it useful to use the method described in [RoW91]. In particular, the agent's entire decision vector $x_a = (x_a^1, \ldots, x_a^N) \in \mathbb{R}^{n_a}$ must satisfy the *nonanticipativity* constraint: for all $t = 1, \ldots, T-1$ and for all pairs (i, j) of scenarios indistinguishable through the first t time stages, one must have

$$x_a^i(\tau) - x_a^j(\tau) = 0 \quad \text{for } \tau = 1, \ldots, t. \tag{2.1}$$

The set described by equation 2.1 is more efficiently summarized as follows. Suppose that for each $\tau \in \{1, \ldots, T\}$ we denote as $C = \{C_1(\tau), \ldots, C_{M(\tau)}(\tau)\}$ the set of $M(\tau)$ sets that partitions $\{1, \ldots, N\}$ into groups of scenarios that are as yet indistinguishable. Suppose also that we impose an ordering on each of the member sets of C. Then equation 2.1 can be rewritten as

$$x_a^{\text{ord}(k, C_j(\tau))}(\tau) - x_a^{\text{ord}(k+1, C_j(\tau))}(\tau) = 0 \tag{2.2}$$

for $\tau = 1, \ldots, T, j = 1, \ldots, M(\tau), k = 1, \ldots, \text{card}(C_j(\tau)) - 1$, where $\text{ord}(k, C_j(\tau))$ represents the k^{th} element of $C_j(\tau)$, and $\text{card}(C_j(\tau))$ is the size of $C_j(\tau)$. We now will denote the set of all 3-tuples, (i, j, τ), for which there exists an equation 2.2 as U[1]. Thus, the stochastic equilibrium programming problem can be formed as follows:

$$\text{Find that set } \{x_1^*, x_2^*, \ldots, x_A^*\} \in \mathbb{R}^n \text{ where} \tag{2.3}$$

$$x_1^* = \arg \max_{x_1 \in \mathcal{L}_1} \sum_{j=1}^{N} p_j u_1^j(x_1^j, (x_{\Omega_{-1}}^j)^*)$$

$$\text{and } \mathcal{L}_1 = \{(x_1^1, \ldots, x_1^N) | x_1^i(\tau) - x_1^j(\tau) = 0 \quad \text{for } (i, j, \tau) \in U\}$$

$$\vdots$$

$$x_A^* = \arg \max_{x_A \in \mathcal{L}_A} \sum_{j=1}^{N} p_j u_A^j(x_A^j, (x_{\Omega_{-A}}^j)^*)$$

$$\text{and } \mathcal{L}_A = \{(x_A^1, \ldots, x_A^N) | x_A^i(\tau) - x_A^j(\tau) = 0 \quad \text{for } (i, j, \tau) \in U.\}$$

To ensure that duality holds for this problem, we assume, in addition to the convexity assumption made above, the appropriate constraint qualification that for each $a \in \{1, \ldots, A\}$, the set $\{d_a | d_a \perp \mathcal{L}_a\}$ is not empty.

[1] We assume that each agent faces the same structural uncertainty which makes U identical for all agents.

This notion of equilibrium is special from both an interpretative and structural perspective. It is special from a structural perspective because each scenario component of each agent's problem depends only on decisions made in that scenario. This means that there is the potential for separability across the scenarios. We will take advantage of this in our solution procedure. Additionally, it is special from an interpretative perspective because it models competitive behavior under uncertainty by assuming that the contribution to an agent's utility from a certain scenario is a function only of decisions from that scenario. This means that agents, when thinking about a certain scenario's contribution to their overall utility, will think only about the decisions that they and their opponents are making in *that scenario*.

3 Homotopy Methods for Solving Systems of Equations

Homotopy methods, as first proposed by [Sca67] and [EaS76], and further developed by [GaG78], [GaZ79b], [GaZ79a], and [GaZ79a], are powerful ways of determining solutions to complex systems of equations. Assuming that we wish to find a solution $x \in D \subset R^n$ to

$$F(x) = 0 \tag{3.1}$$

that we know to exist, where $F : R^n \rightarrow R^n$, and D is compact, we might first solve an easier system, also having a solution in D,

$$G(x) = 0 \tag{3.2}$$

where $G : R^n \rightarrow R^n$. From these, we can construct the linear homotopy

$$H(x,t) = tF(x) + (1-t)G(x) \tag{3.3}$$

where $H : R^{n+1} \rightarrow R^n$, and then, given that the function H exhibits the appropriate properties, follow the differentiable path of solutions of $H(x,t) = 0$ that leads from the point where $t = 0$ (and $G(x) = 0$) to the point where $t = 1$ (and x is such that $F(x) = 0$). The appropriate properties that H must have regard the rank of it's Jacobian at various points of the set $H^{-1} = \{(x,t) \in D \times [0,1] | H(x,t) = 0\}$. In particular, H^{-1} is composed of a finite number of disjoint continuously differentiable paths [Mil69] if:

- For all $(x,t) \in H^{-1}$, the Jacobian H' has rank n.
- For all $x \in \{x \in D | H(x,0) = 0 \text{ or } H(x,1) = 0\}$, the matrix $H'_{-(n+1)}$ has rank n, where $H'_{-(n+1)}$ is the Jacobian of H without the column corresponding to differentiation with respect to t.
- $H : R^{n+1} \rightarrow R^n$ is a C^2-map.

These continuously differentiable paths may be loops in $D \times (0,1)^2$, or may connect two boundary points of $D \times [0,1]$ (note that the path will not necessary connect a point with $t = 0$ to a point with $t = 1$). Most importantly, though, these paths will never have bifurcations and will never be infinite in length. Thus, in principle, they can be followed from one end to another, or, in the case of loops, from an arbitrary starting point back to the same point.

We use this method to solve the system 3.1 by traversing the path that begins at $(x_g, 0)$ (x_g, a solution of 3.2), and ends at $(x_f, 1)$ (x_f, a solution to 3.1). We remark that such a path is only guaranteed to exist if we assume that

$$(x,t) \in H^{-1} \rightarrow \ x \text{ is not in } \partial D, \tag{3.4}$$

where ∂D is the boundary of D, and that x_g is the unique solution of 3.2. These assumptions remove the possibility that the path leaving $(x_g, 0)$ might curl back and intersect $\{(x,0)|x \in D\}$, or that the path leaving $(x_g, 0)$ might intersect $\{(x,t)|x \in \partial D, t < 1\}$. The only possibility left is that the path must reach $(x_f, 1)$.

We can traverse this path using a predictor-corrector continuation method. One variation on this idea is suggested by [AlG90]. To trace $(x,t) \in H^{-1}$ starting at $(x_g, 0)$ we parameterize the curve according to arc length, θ. We note that the exact curve we seek to trace is described by the following system:

$$H(x(\theta), t(\theta)) = 0, \tag{3.5}$$

$$(x(0), t(0)) = (x_g, 0).$$

At any point, θ, along the curve described by 3.5, the following related differential system:

$$H_x^{'}(x(\theta), t(\theta))\frac{\partial x(\theta)}{\partial \theta} + H_t^{'}(x(\theta), t(\theta))\frac{\partial t(\theta)}{\partial \theta} = 0. \tag{3.6}$$

$$||(\frac{\partial x(\theta)}{\partial \theta}, \frac{\partial t(\theta)}{\partial \theta})|| = 1$$

$$det \left\{ \begin{array}{c} H^{'} \\ \frac{\partial x(\theta)}{\partial \theta} \\ \frac{\partial t(\theta)}{\partial \theta} \end{array} \right\} > 0$$

provides the tangent, $(\frac{\partial x(\theta)}{\partial \theta}, \frac{\partial t(\theta)}{\partial \theta})$, to the curve. Thus, at any point along the curve, θ, we can calculate this tangent, move in its direction, and expect to stay close to the curve, at least for a small step size, τ. To insure that the move to $(x^{'}, t^{'}) = (x(\theta), t(\theta)) + \tau(\frac{\partial x(\theta)}{\partial \theta}, \frac{\partial t(\theta)}{\partial \theta})$ doesn't lead us too far from the curve we are seeking to follow, we follow-up our "predictor" step in the direction of the tangent with a correcting sequence of Newton steps to bring us back to a new point on the curve. The new point, $(\hat{x}, \hat{t})$, we find using the Newton steps solves the following system of equations:

$$H(\hat{x}, \hat{t}) = 0, \tag{3.7}$$

[2]There can be no points of tangency at (x,t) where $H(x,t) = 0$ and $t = 1$ or $t = 0$ because of our assumption about the rank of $H^{'}_{-(n+1)}$ at such points.

$$\left\{ \begin{array}{c} \hat{x} - x' \\ \hat{t} - t' \end{array} \right\}^T \left\{ \begin{array}{c} \frac{\partial x(\theta)}{\partial \theta} \\ \frac{\partial t(\theta)}{\partial \theta} \end{array} \right\} = 0$$

and is, thus, on the curve, and, in addition, further along the curve by virtue of the orthogonality requirement which ensures a positive inner product between the original predictor step $(\frac{\partial x(\theta)}{\partial \theta}, \frac{\partial t(\theta)}{\partial \theta})$ and $(\hat{x}, \hat{t}) - (x(\theta), t(\theta))$. It should be noted that the system 3.7 may not have a solution if the path is highly nonlinear and τ is too large. In this case, τ will have to be reduced. The above steps are combined into the following algorithm:

Algorithm 3.1.
Step 0: Select (x_1, t_1) such that $G(x_1) = 0$ and $t_1 = 0$. Choose μ,ϵ,$\hat{\tau}$. Set $k = 1$.

Step 1: Set $\tau_k = \hat{\tau}$. Find $(\frac{\partial x_k}{\partial \theta}, \frac{\partial t_k}{\partial \theta})$ that solves 3.6 with θ replaced everywhere by k.

Step 2: Set $(x_k', t_k') = (x_k, t_k) + \tau_k(\frac{\partial x_k}{\partial \theta}, \frac{\partial t_k}{\partial \theta})$.

Step 3: Find $(\hat{x}_k, \hat{t}_k)$, via a sequence of Newton steps, that is a μ-solution to 3.7:

Set i=0 and let $(\hat{x}^i, \hat{t}^i) = (x_k', t_k')$.
Repeat
Find $(\hat{x}^{i+1}, \hat{t}^{i+1})$ such that

$$H(\hat{x}^i, \hat{t}^i) + H'(\hat{x}^i, \hat{t}^i)((\hat{x}^{i+1}, \hat{t}^{i+1}) - (\hat{x}^i, \hat{t}^i)) = 0, \tag{3.8}$$

$$\left\{ \begin{array}{c} \hat{x}^{i+1} - x_k' \\ \hat{t}^{i+1} - t_k' \end{array} \right\}^T \left\{ \begin{array}{c} \frac{\partial x_k}{\partial \theta} \\ \frac{\partial t_k}{\partial \theta} \end{array} \right\} = 0$$

until $(\hat{x}^{i+1}, \hat{t}^{i+1})$ solves 3.7 to within μ^3 or the method diverges.

If $(\hat{x}^{i+1}, \hat{t}^{i+1})$ is a μ-solution of 3.7, $(\hat{x}_k, \hat{t}_k) = (\hat{x}^{i+1}, \hat{t}^{i+1})$
If the Newton steps diverge, set $\tau_k = \tau_k/3$ and return to step 2.
Set $(x_{k+1}, t_{k+1}) = (\hat{x}_k, \hat{t}_k)$.
If $||t_{k+1} - 1|| < \epsilon$, done.
Otherwise, if $t_{k+1} < 1$, go to step 1.
Otherwise, if $t_{k+1} > 1$ and $t_k < 1$, then apply Newton's method to 3.1 using x_{k+1} as a starting point.

It is clear that the procedures that make this a potentially expensive algorithm, especially in the case of large systems, are the solution of system 3.6 and the repeated solution of the Newton system 3.8. We will see that our problem possesses an "almost separable across scenarios" structure that will allow us to intelligently solve these systems of equations at a reasonable cost.

[3] where the error of the solution is measured as the norm of system 3.7 evaluated at $(\hat{x}^{i+1}, \hat{t}^{i+1})$.

4 A Homotopy Solution Method for Stochastic Equilibrium Problems

4.1 Homotopy Formulation

We wish to solve problem 2.3 by writing out the following system of equations that describes the solution set of the problem[4]:

$$\left\{\begin{array}{c} p_1\partial_{x_1^1}u_1^1(x_1^1,x_{\Omega_{-1}}^1)-B_1^{1,1}y_1^{1,1}-B_1^{1,2}y_1^{1,2} \\ \vdots \\ p_1\partial_{x_A^1}u_A^1(x_A^1,x_{\Omega_{-A}}^1)-B_A^{1,1}y_A^{1,1}-B_A^{1,2}y_A^{1,2} \\ \vdots \\ p_N\partial_{x_1^N}u_1^N(x_1^N,x_{\Omega_{-1}}^N)-B_1^{N,1}y_1^{N,1}-B_1^{N,2}y_1^{N,2} \\ \vdots \\ p_N\partial_{x_A^N}u_A^N(x_A^N,x_{\Omega_{-A}}^N)-B_A^{N,1}y_A^{N,1}-B_A^{N,2}y_A^{N,2} \\ x_1^i(\tau)-x_1^j(\tau) \quad \text{for } (i,j,\tau)\in U \\ \vdots \\ x_A^i(\tau)-x_A^j(\tau) \quad \text{for } (i,j,\tau)\in U \\ y_1^{i,1}(\tau)+y_1^{j,2}(\tau) \quad \text{for } (i,j,\tau)\in U \\ \vdots \\ y_A^{i,1}(\tau)+y_A^{j,2}(\tau) \quad \text{for } (i,j,\tau)\in U \end{array}\right\} = 0 \tag{4.1}$$

where $B_a^{i,1}$ and $B_a^{i,2}$ are $q_a \times q_a$ diagonal matrices whose diagonal entries corresponding to the dual variables $y_a^{i,1}(\tau)$ and $y_a^{i,2}(\tau)$ are 1 if there exists a j such that $(i,j,\tau)\in U$ and $(j,i,\tau)\in U$, respectively, and 0 otherwise. We refer to the system of equations in 4.1 as $F(x)$ and it's size is $m \times m$ where $m = (n + 2 * (\text{card}(U)))$ and $\text{card}(U)$ represents the cardinality of set U. We assume that the functions $\{u_1^1(\cdot),\ldots,u_1^N(\cdot),\ldots,u_A^1(\cdot),\ldots,u_A^N(\cdot)\}$ have a structure appropriate to make the nonlinear operator 4.1 uniformly monotone, and it's Jacobian diagonally dominant. These properties will make possible the decomposition strategy that we'll use to solve the large linear systems associated with algorithm 3.1. From a modeling perspective, we note that we actually include in our formulation of the problem more dual variables, $y_a^i(\tau)$, than the one per nonanticipativity constraint that is required. Specifically, we include two per constraint. To model things properly, then, we insist that the sum of the two dual variables be equal to zero so that the dual variables have the same absolute values but opposite signs, as is correct. These redundant dual variables give the system more flexibility when we apply our solution methodology. Since we desire to find a solution to this system via the Homotopy methodology reviewed in section §3, we must construct another system, $G(x)$, with a solution easier to find than $F(x)$. We choose for $G(x)$ the system of equations describing the solution set of our stochastic equilibrium problem with the nonanticipativity requirement relaxed. That is,

$$\text{Find that set } \{x_1^*, x_2^*, \ldots, x_A^*\} \in \mathbb{R}^n \text{ where} \tag{4.2}$$

[4] We can do this because of the constraint qualification and convexity assumption made in §2.

$$x_1^* = \arg\max_{x_1} \sum_{j=1}^{N} p_j u_1^j(x_1^j, (x_{\Omega_{-1}}^j)^*)$$

$$\vdots$$

$$x_A^* = \arg\max_{x_A} \sum_{j=1}^{N} p_j u_A^j(x_A^j, (x_{\Omega_{-A}}^j)^*).$$

It has the associated system of equations:

$$\left\{ \begin{array}{c} p_1 \partial_{x_1^1} u_1^1(x_1^1, x_{\Omega_{-1}}^1) - B_1^{1,1} y_1^{1,1} - B_1^{1,2} y_1^{1,2} \\ \vdots \\ p_1 \partial_{x_A^1} u_A^1(x_A^1, x_{\Omega_{-A}}^1) - B_A^{1,1} y_A^{1,1} - B_A^{1,2} y_A^{1,2} \\ \vdots \\ p_N \partial_{x_1^N} u_1^N(x_1^N, x_{\Omega_{-1}}^N) - B_1^{N,1} y_1^{N,1} - B_1^{N,2} y_1^{N,2} \\ \vdots \\ p_N \partial_{x_A^N} u_A^N(x_A^N, x_{\Omega_{-A}}^N) - B_A^{N,1} y_A^{N,1} - B_A^{N,2} y_A^{N,2} \\ y_1^{i,1}(\tau) \quad \text{for } (i,j,\tau) \in U \\ \vdots \\ y_A^{i,1}(\tau) \quad \text{for } (i,j,\tau) \in U \\ y_1^{j,2}(\tau) \quad \text{for } (i,j,\tau) \in U \\ \vdots \\ y_A^{j,2}(\tau) \quad \text{for } (i,j,\tau) \in U \end{array} \right\} = 0. \tag{4.3}$$

This system of equations[5] is much easier to solve because it is completely separable across the scenarios. That is, each of the N scenario equilibrium problems can be solved separately, and, hence, with much less effort than for the original system.

Since we assume that both systems, F and G, have solutions, and bounded solutions at that, we can certainly perform all our computation within some arbitrary compact set D that contains these points. We also will assume that the functions of our problem have a suitable degree of differentiability and regularity so that all the points mentioned in [Mil69] that are necessary for paths to be continuously differentiable are present. Finally, we note that the path can never turn back on itself[6], and that the strong "no boundaries assumption" that ensures our path will never hit a boundary where $t < 1$ holds[7]. Thus,

[5] System 4.3 can be solved without the variables, $(y_1^{1,1}, y_1^{1,2}, \ldots, y_A^{1,1}, y_A^{1,2}, \ldots, y_1^{N,1}, y_1^{N,2}, \ldots, y_A^{N,1}, y_A^{N,2})$. They are included, though, to give this system and system 4.1 the same dimension.

[6] Uniform monotonicity of the operator in equation 4.1 implies that the Jacobian of this system is invertible, which implies that the last component of the solution, $(\frac{\partial x(\theta)}{\partial \theta}, \frac{\partial t(\theta)}{\partial \theta})$, of equation 3.6 is always nonzero. Since it starts out positive, it can never become negative.

[7] Uniform monotonicity assures us that the "no boundaries assumption" is true for a large enough compact set D. Why? Because monotonicity assures us that the Jacobian of operator 4.1 is invertible and this means that $\frac{\partial x(\theta)}{\partial \theta}$ is bounded.

we have all the components necessary to construct the homotopy of equation 3.3:

$$\left\{\begin{array}{c} p_1\partial_{x_1^1}u_1^1(x_1^1,x_{\Omega_{-1}}^1)-B_1^{1,1}y_1^{1,1}-B_1^{1,2}y_1^{1,2} \\ \vdots \\ p_1\partial_{x_A^1}u_A^1(x_A^1,x_{\Omega_{-A}}^1)-B_A^{1,1}y_A^{1,1}-B_A^{1,2}y_A^{1,2} \\ \vdots \\ p_N\partial_{x_1^N}u_1^N(x_1^N,x_{\Omega_{-1}}^N)-B_1^{N,1}y_1^{N,1}-B_1^{N,2}y_1^{N,2} \\ \vdots \\ p_N\partial_{x_A^N}u_A^N(x_A^N,x_{\Omega_{-A}}^N)-B_A^{N,1}y_A^{N,1}-B_A^{N,2}y_A^{N,2} \\ tx_1^i(\tau)-tx_1^j(\tau)+(1-t)y_1^{i,1}(\tau) \quad \text{for } (i,j,\tau)\in U \\ \vdots \\ tx_A^i(\tau)-tx_A^j(\tau)+(1-t)y_A^{i,1}(\tau) \quad \text{for } (i,j,\tau)\in U \\ ty_1^{i,1}(\tau)+y_1^{j,2}(\tau) \quad \text{for } (i,j,\tau)\in U \\ \vdots \\ ty_A^{i,1}(\tau)+y_A^{j,2}(\tau) \quad \text{for } (i,j,\tau)\in U \end{array}\right\} = 0. \tag{4.4}$$

and solve it via algorithm 3.1. The issue that still needs to be dealt with, though, regards the manner in which the systems 3.6 and 3.8 should be solved.

4.2 Linear Equation Solving

The success of algorithm 3.1 depends on the ease with which the two linear systems, 3.6 and 3.8, can be solved. The solutions of these systems are important because they determine, respectively, the direction of search, and the route back to the safety of the manifold we follow to the solution. Both systems can most easily be solved using an iterative Jacobi technique that, first fixes all but the first scenario's variables and finds that configuration of the first scenario's variables that solves a subset of the equations, then fixes all but the second scenario's variables and solves a different subset of the equations, and so on, repeating until the entire system of equations is solved. Of course, since this is a Jacobi iterative technique, each of these steps can be performed in parallel because all updates are based on "old" data. The reason why such a technique can be used effectively is best understood if we look more closely at the structure of the Jacobian associated with 4.4. Consider an example problem in which 3 agents compete against one another over a horizon lasting 3 periods and faced with uncertainty described by 3 scenarios. We will assume that the tree appears as in figure 4.1 when written in explicit nonanticipative form. This tree gives us $U = \{(1,2,1),(2,3,1),(1,2,2)\}$ which means that all decisions made by each of the three agents in the first period must be identical across the three scenarios. In the second period only the decisions made in the first and second scenarios need be identical, while in the last period all scenario decisions are independent and potentially different. Let's first consider an example of the $m \times m+1$ linear equation involving the Jacobian that we need to solve in order to compute the tangent. If we order the columns, first according to the scenario, and then, within scenario groupings, according to the period, we find that our linear system has the following structure pictured

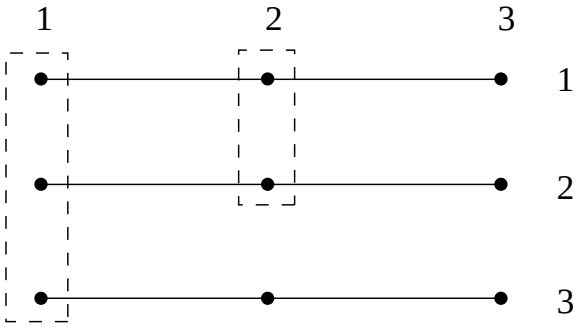

Figure 4.1: 3 agent, 3 period, 3 scenario tree

in equation 4.5[8]

$$\left\{\begin{array}{|ccc|}\hline c_1^1 & & \\ c_1^2 & & \\ c_1^3 & & \\ \hline & c_2^1 & \\ & c_2^2 & \\ & c_2^3 & \\ \hline & & c_3^1 \\ & & c_3^2 \\ & & c_3^3 \\ \hline T & -T & \\ T & -T & -T \\ T & -T & \\ \hline \end{array}\right.$$

$$\left.\begin{array}{cccccccccc} -I & & & & & & & & & \\ & -I & -I & & & & & & & \\ & & & & -I & & & & & \\ & & & & & & -I & & & \\ & & & & & & & & & \\ I-T & & & & & & & & & \\ & & I-T & & & & & & & \\ T & I & & & & & & & & \\ & & T & I & & & & & & \\ & & & & I-T & & & & & \\ & & & & T & I & & & & \\ & & & & & & & & & \end{array}\right\} \left\{\begin{array}{c} x_a^1(1) \\ x_a^1(2) \\ x_a^1(3) \\ x_a^2(1) \\ x_a^2(2) \\ x_a^2(3) \\ x_a^3(1) \\ x_a^3(2) \\ x_a^3(3) \\ y_a^{1,1}(1) \\ y_a^{2,2}(1) \\ y_a^{2,1}(1) \\ y_a^{3,2}(1) \\ y_a^{1,1}(2) \\ y_a^{2,2}(2) \\ t \end{array}\right\} = 0. \tag{4.5}$$

This structure is common to all stochastic equilibrium problems of the type defined in section §2. Scenario blocks of columns and constraints describing separable multi-agent equilibrium problems, connected to ajacent blocks via nonanticipativity constraints and their associated duals. This structure suggests some kind of decomposition procedure for finding the element in the kernel of the Jacobian, J. For example, upon squaring the system out with an $(m+1)^{st}$ row, e^T, "close" to the null space of J (to assure a unique solution to the rank m system), a series of Jacobi iterations might be used. Such a procedure is assured of converging by virtue of our assumption that the problem is uniformly monotone (which implies that the Jacobian matrix is positive definite [OrR70], which along with the assumed diagonal dominance, makes any associated linear system amenable to such a procedure [GoL89].) The actual element of the Null space that we seek would then be [AlG90]:

$$(\frac{\partial x_k}{\partial \theta}, \frac{\partial t_k}{\partial \theta}) = \frac{e - x}{||e - x||} \tag{4.6}$$

where

$$Jx = Je$$
$$e^T x = 0.$$

To actually implement a Jacobi procedure, though, we need to decide how to locate such an $(m+1)^{st}$ row, e^T, and how to order the Jacobi procedure. The simplest choice of

[8] Let $c_i^j = p_i \partial_{x_a^i(j)} u_a^i(\cdot)$ and T be the diagonal matrix $I * t$. Note that $\cdot$ means a non-zero entry.

e^T is the previous iterations tangent (as the system and it's null space will not have changed that much). The order of the Jacobi procedure should be along scenario lines as suggested earlier. In particular, for each $i = 1, \ldots, N$, where N is the number of scenarios, the square subsystem made up of the rows containing the i^{th} scenario equilibrium block, and the rows corresponding to the i^{th} scenario duals should be solved in terms of the i^{th} scenario's variables and duals, holding all other scenario variables and duals constant. Finally, the last row of the system should be solved in terms of t. This operation is possible because the last row (assuming it is close enough to the null space of J) will always contain a nonzero entry in the column associated with t because of the assumed uniformly monotone structure of the problem[9]. This same property assures us that each of the separable scenario operations in the Jacobi procedure are possible as well. Finally, each iteration of the correcting step involves the solution of an $(m+1) \times (m+1)$ system of equations that can be handled in the same fashion as that described above.

The natural separability of scenarios in our stochastic equilibrium problem, together with the assumption we've made regarding it's structure (uniform monotonicity and diagonal dominance), ensure that a solution to the problem can be located by following the smooth path that lies between the solution set of the scenario problem and the full blown stochastic model with its attendant nonanticipativity constraints. Best of all, the natural separability across scenarios can be harnassed to ease the work involved in following this path.

References

[AlG90] E. L. Allgower and K. Georg, *Numerically stable homotopy methods*, in Computational Solution of Nonlinear Systems of Equations, E. L. Allgower and K. Georg, eds., American Mathematical Society, Providence, 1990, pp. 1–14.

[Dav53] D. Davidenko, *On the approximate solution of a system of nonlinear equations*, Ukrain. Mat. Z. **5** (1953) 196–206.

[EaS76] B. C. Eaves and H. Scarf, *The solution of systems of piecewise linear equations*, Mathematics of Operations Research **1** (1976) 1–27.

[Eav72] B. C. Eaves, *Homotopies for computation of fixed points*, Mathematical Programming **3** (1972) 1–22.

[GaG78] C. B. Garcia and F. J. Gould, *A theorem on homotopy paths*, Mathematics of Operations Research **3** (1978) 282–289.

[GaZ79a] C. B. Garcia and W. I. Zangwill, *An approach to homotopy and degree theory*, Mathematics of Operations Research **4** (1979) 390–405.

[GaZ79b] ———, *Determining all solutions to certain systems of nonlinear equations*, Mathematics of Operations Research **4** (1979) 1–14.

[GaZ81] ———, *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice-Hall, Englewood Cliffs, N.J., 1981.

[GoL89] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1989.

[9] Uniform monotonicity of the operator in equation 4.1 implies that the Jacobian of this system is invertible, which implies that the $\frac{\partial t(\theta)}{\partial \theta}$ component of the solution, $(\frac{\partial x(\theta)}{\partial \theta}, \frac{\partial t(\theta)}{\partial \theta})$, of equation 3.6 is always nonzero.

[Mil69] J. W. Milnor, *Topology from the differential viewpoint*, The University Press of Virginia, Charlottesville, Virginia, 1969.

[OrR70] J. M. Ortega and W. C. Rheinboldt, *Iterative Solutions of Nonlinear Equations in Several Variables*, Academic Press, 1970.

[Ros96] C. H. Rosa, *A decomposition technique for equilibrium programming under uncertainty*, WP-96-013, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1996.

[RoW91] R. T. Rockafellar and R. J.-B. Wets, *Scenarios and policy aggregation in optimization under uncertainty*, Math. Oper. Res. **16** (1991) 119–147.

[Sca67] H. E. Scarf, *The approximation of fixed points of continuous mappings*, SIAM Journal of Applied Mathematics **15** (1967) 1328–1343.