A GENETIC GAME OF TRADE, GROWTH AND EXTERNALITIES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ECONOMICS
AND THE INSTITUTE OF ECONOMICS AND SOCIAL
SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULLFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Süheyla Özyıldırım
February 1997

A GENETIC GAME OF TRADE, GROWTH AND EXTERNALITIES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ECONOMICS
AND THE INSTITUTE OF ECONOMICS AND SOCIAL
SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULLFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY.

By

Süheyla Özyıldırım
February 1997

I certify that I have read this dissertation and in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy in Economics.

Professor A. Bülent Özgüler

I certify that I have read this dissertation and in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy in Economics.

Professor Asad Zaman

I certify that I have read this dissertation and in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy in Economics.

Associate Professor Gönül Turhan Sayan

I certify that I have read this dissertation and in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy in Economics.

Associate Professor Erinç Yeldan

I certify that I have read this dissertation and in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy in Economics.

Assistant Professor Nedim M. Alemdar (Supervisor)

I certify that this dissertation conforms the formal standards of the Institute of Economics and Social Sciences.

Professor Ali L. Karaosmanoğlu

Director of the Institute of Economics
and Social Sciences

Abstract

A GENETIC GAME OF TRADE, GROWTH AND EXTERNALITIES

Süheyla Özyıldırım
Ph.D. Thesis in Economics
Supervisor: Professor Nedim M. Alemdar
February 1997

This dissertation introduces a new adaptive search algorithm, *Genetic Algorithm (GA)*, for dynamic game applications. Since *GAs* require little knowledge of the problem itself, computations based on these algorithms are very attractive for optimizing complex dynamic structures. Part one discusses *GA* in general, and dynamic game applications in particular. Part two is comprised of three essays on computational economics. In Chapter one, a genetic algorithm is developed to approximate open-loop Nash equilibria in non-linear difference games of fixed duration. Two sample problems are provided to verify the success of the algorithm. Chapter two covers discrete-time dynamic games with more than two conflicting parties. In games with more than two players, there arises the possibility of coalitions among groups of players. A three-country, two-bloc trade model analyzes the impact of coalition formation on optimal policies. Chapter three extends *GA* further to solve open-loop differential games of infinite duration. In a dynamic North/South trade game with transboundary knowledge spillover and local pollution optimal policies are searched. Cooperative and noncooperative modes of behavior are considered to address the welfare effects of pollution and knowledge externalities.

Keywords: Genetic Algorithm, Dynamic Games, North/South Trade, Externalities.

Öz

TİCARET, BÜYÜME VE DIŞSALLIK ÜZERİNE GENETİK BİR OYUN

Süheyla Özyıldırım
Ekonomi Doktora Tezi
Tez yöneticisi: Profesör Nedim M. Alemdar
Şubat 1997

Bu doktora tezi dinamik oyun uygulamaları için uyum kabiliyetli yeni bir arama algoritmasını, *Genetik Algoritmayı (GA)* tanıtmaktadır. Genetik algoritmalar problemin kendisi hakkında az bilgiye ihtiyaç duyduğundan, bu algoritmalara dayanan çözümler karmaşık dinamik yapıların optimizasyonu için çok caziptirler. İlk bölüm genel olarak *GA*'yı, özel olarak da dinamik oyun uygulamalarını tartışmaktadır. Bölüm iki, sayısal çözüm uygulamalı ekonomi üzerine üç makaleden oluşur. İlk makalede doğrusal olmayan fasılalı sabit süreli oyunların takribi açık-döngü Nash dengelerini bulan genetik algoritma geliştirilmiştir. Algoritmanın doğruluğunu ispatlamak için iki örnek problem de verilmiştir. İkinci makale, birbirleri ile çatışan ikiden fazla grubun fasılalı dinamik oyunlarını kapsar. İkiden fazla oyunculu oyunlarda, oyuncular arasındaki koalisyon olasılığı ortaya çıkmaktadır. Üç-ülke, iki-bloklu ticaret modeli ile koalisyon teşkilinin optimal politikalar üzerindeki etkileri de bu makalede incelenmiştir. Üçüncü makalede sonsuz süreli açık-döngü diferensiyal oyunları çözmek için genetik algoritmanın kapsamı daha da genişletilmiştir. Burada sınırları aşan bilgi akışı ve yerel hava kirliliği olan dinamik Kuzey/Güney ticaret oyununda optimal politikalar aranmıştır. Hava kirliliği ve bilgi dışsallığının, refah üzerindeki etkilerine, işbirliği yapılmayan ve yapılan davranış biçimlerinde incelenmiştir.

Anahtar Sözcükler: Genetik Algoritma, Dinamik Oyunlar, Kuzey/Güney ticareti, Dışsallık.

Acknowledgements

First of all, I would like to thank to my thesis supervisor Professor Nedim M. Alemdar. He is an excellent economist, I am indebted very much to him for his invaluable suggestions and guidance. I learned much from him how an economist approaches to the problem and why economic insight is the main priority. I am also indebted to Professor Bülent Özgüler for his careful reading and insightful suggestions. I thank Professor Asad Zaman, Professor Gönül Turhan and Professor Erinç Yeldan for going over the text and for valuable comments and suggestions. I would like to thank also Professor Jean Mercenier who gave a seminar at Bilkent University by the invitation of Professor Yeldan. I learned much from Prof. Mercenier and his works at the crucial stages of my dissertation. I would like to express my graditude to Professors Serdar Sayan, Faruk Selçuk, Sübidey Togan and Bahri Yılmaz for their encouragements.

I am thankful to Kadriye Göksel, İsmail Sağlam, Jülide Yıldırım and Murat Yülek. Their support helped to maintain my enthusiasm.

Most of all, I am grateful to my family and friends who provided me with constant support during my studies.

Contents

Part I
Introduction

The unifying theme in this dissertation is the development of *Genetic Algorithm* for dynamic game applications. Ever since the study of differential games was launched by Isaacs (1954), a plethora of literature emerged many studies on the subject have appeared, but very little attention has been given to the development of computational techniques to solve such problems. Prior to the 1975 work by Pau, all known references have investigated open-loop controls based upon necessary Nash equilibrium conditions, and the literature on the computation of Nash equilibriums in non-linear non-zero sum differential games contains even fewer studies.

There are basically three main ingredients in an optimization theory, whether static or dynamic: the criterion function, the controller (player), and the information available to the controller. For traditional control theory, all ingredients are singular in the sense that there is only one criterion, one controller coordinating all actions and one information set available to controller. But it is possible to conceive, situations in which there are more than one performance measure, more than one controller (player) operating with or without coordination from others, and finally two or more controllers who may or may not have the same information set available to them. The cases of more than one players with conflicting purposes are characterized as games. Thus, dynamic game theory provides a framework for quantitative modeling and analysis of the interactions among economic agents over time.

The solution concepts directly suggest the mathematical or numerical tools under various information structures. All these involve optimization of functionals either over time or stagewise at each point in time. The former requires direct application of optimal control theory (specifically the minimum principle of Pontryagin) which amounts to derivation of conditions for open-loop Nash equilibria in the context of dynamic game theory. The discrete counter-part of the minimum principle is likewise applicable to open-loop Nash equilibria of multi-stage (discrete-time) games. In both cases, each player faces a standard optimal control problem, which are arrived at by fixing the other players' policies as some arbitrary functions. In principle, the necessary and/or sufficient conditions for open-loop Nash equilibria can be obtained by listing down the conditions required by each optimal control problem and then requiring that all these be satisfied *simultaneously*. Because of the couplings that exist between these various conditions, each one corresponding to the optimal control problem faced by one player, to solve the corresponding equilibria analytically or numerically is several orders of magnitude more difficult than to solve optimal control problems. Very few closed-form solutions exist for these game problems, one of which pertains to the case when performance criteria are quadratic and the state equation is linear (Başar 1986).

The need to solve optimization problems arises in one form or other in almost all fields. As a consequence, enormous amount of effort has gone into developing both analytical and numerical optimization techniques. So, these kind of assumptions on the functional representations of the equations are now unnecessary and unrealistic. There is a large class of interesting problems for which no reasonably fast algorithms have been developed. As a consequence, there is a continuing search for new and more robust optimization techniques capable of handling such problems. For some complex optimization problems, we have seen increasing interest in probabilistic algorithms including *Genetic Algorithm (GA)*. *GAs* are stochastic algorithms whose search methods model some natural phenomena: genetic inheritance and Darwinian strife for survival.

Genetic Algorithms were developed by Holland in 1975 as a way of studying adaptation, optimization, and learning. They are modeled on the processes of evolutionary genetics. A basic *GA* manipulates a set of structures, called *population*. Structures are usually coded as strings of characters drawn from some finite alphabet often binary. Whatever the interpretation, each string is assigned a measure of performance called its *fitness*, based on the performance of the corresponding structure in its environment. The *GA* manipulates this population in order to produce a new population that is better adapted to the environment. It is proved that *GAs* are powerful techniques for locating improvements in complicated high-dimensional spaces. They exploit the mutual information inherent in the population, rather than simply trying to exploit the best individual in the population. Thus, Holland's schema theorem puts the theory of genetic algorithms on rigorous footing by calculating a bound on the growth of useful similarities or *building blocks.* The fundamental principle of *GAs* is to make good use of these templates.

There are currently other so-called artificial intelligence (*AI*) techniques used for the design and optimization of control systems. Most widely used *AI* techniques are neural networks, knowledge-based systems, fuzzy logic systems ans simulated annealing. Knowledge-based and fuzzy logic systems rely on *a priori* knowledge of the problem to be solved and can therefore be classified under techniques that learn from past performance. On the other hand, even though artificial neural networks learn through repeated exposure to desired input-output relationship, *GAs* provide a technique for optimizing a given control structure whereas the artificial neural networks can only be an integral part of the structure. In other words, *GAs* can be used to design artificial neural networks but not vice versa. Finally, simulated annealing is a combinatorial optimization that recently has attracted attention for optimization problems of large scale. The difference between simulated annealing and the genetic algorithm lies in the fact that the *GA* focuses on the importance of recombination and other operators found in the nature, is better understood (Krishnakumar and Golberg 1992).

In this dissertation, we design and implement a numerical method *called* shared-memory algorithm using *GAs* for the solution of multiple criterion dynamic optimization problems. In chapter two, the shared-memory algorithm has been introduced to optimize a complex system such as a dynamic game. This algorithm was designed to be used for the approximation of the problems which are beyond analytical methods and which present significant difficulties for numerical techniques.

GAs require little knowledge of the problem itself, computations based on these algorithms are very attractive to the complex dynamic optimization problems. Optimal control problems which are one-player dynamic optimization problems, are still quite difficult to deal with numerically. Recently, Michalewicz (1992) used *GAs* to solve optimal control problems. The application of *GAs* to optimal control problems is the natural extension of genetic algorithms as a function optimizer. Besides, the use of this algorithm allows to solve a number of classes of control problems which require special techniques: One such case is that of systems in which have objective functionals that are linear in the control. The solvability of these problems is possible as long as control vector is constrained by the existence of upper and lower bounds on the control. There is another special situation which occurs when the objective functional is not continuous over some interval of the time ($t_1 \leq t \leq t_2$). Since this causes the control to vanish in that interval, it is not clear how we can determine the optimal value of u_t over $t_1 \leq t \leq t_2$. The control over this interval is called a *singular arc* and these specially difficult problems are called *singular control problems.* These problems must be exercised with special caution in traditional approaches. However, as an adaptive search algorithm, *GA* is quite suitable for such problems.

Actually, as mentioned by De Jong (1993), *GAs* are not only function optimizers but they have *learning* ability which we use to solve open-loop difference games. Thus, in the game-theoretic framework, we used both the optimization and the learning property of *GAs.* Since *GAs* are numerical algorithms, in order to test the success of the algorithm, in chapter two, we applied the *shared-memory algorithm* to the previously solved open-loop Nash difference game problems. First example is the numerical example from Kydland (1975) and the second one is the typical example of policy optimization in a policy game given by Brandsma and Hughes-Hallett (1984) using the formulation by L.R. Klein to describe the mechanism of the American economy in the interwar period, 1933-1936. This chapter has been published in the *Journal of Evolutionary Economics.*

In chapter three, we extend the solution procedure for N-player games. The general differential/difference game has N players, each controlling a different set of inputs to a single nonlinear dynamic system and each trying to minimize a different performance criterion. Although one naturally expects that methods for computing solutions to these problems can be obtained by generalizing well-known methods, several difficulties arise which are absent in control problems ($N = 1$) and two person games (Starr and Ho 1969). On the Nash trajectory, each player's cost is minimized with respect to his own control but not with respect to the other player's controls. Because his cost is not minimized with respect to the jth player's control, the ith player is very sensitive to the changes in his rival's controls. The importance of sensitivity increases with the number of players in the game. This is the causes of considerable difficulties in developing algorithms for computing Nash controls for nonlinear problems. Thus, when $N > 2$, as is usually the case in the nonlinear problems, the complexity increases more than twice. Also, the most important phenomenon arising in the extension from two players to N players is the possibility of *coalitions* among groups of players. Very little can be said unless strict rules governing coalition formation are postulated. One special case, a single coalition of all players (the so-called *pareto optimal* solution) is studied in the fourth chapter. In this chapter, we extend the shared memory algorithm for the open-loop solutions for $N \geq 2$ games. The algorithm is applied to a three-country, two-bloc trade model in the same vein as Galor (1986). This chapter has been published in the *Computers and Mathematics with Applications.*

In an increasingly complex and integrated world economy, international and interregional economic interactions have been an issue of public policy for a long time. Of growing interest is a particular set of problems characterized by *externalities* between countries and regions arising through trade relations. One set of issues involves the local pollution generated while extracting tradable raw materials. Another problem, which will become increasingly important as knowledge spillover, involves the recovery of the effect of pollution. A common feature of these issues is that the welfare of one country depends upon the economic behavior of a foreign country. Thus, one purpose of chapter four is to develop an economic model that incorporates the principal features of the various types of international externality problems outlined above. Specifically, the chapter deals with a simple pollution-knowledge model as a *vehicle for coordination* between nations. The model is constructed as an infinite horizon continuous problem which is initially discretized as a finite horizon dicrete-time problem and solved again using shared-memory algorithm. Mercenier and Michel (1994) propose time aggregation to discretize continuous time infinite horizon optimal control problems, and we extend their results to open-loop solutions with multi-controllers. In adressing trade and environment concerns, we used the North/South debate where the North has the technology and the South has the polluting raw material. In the dynamic game framework, both regions are allowed to interact noncooperatively and cooperatively. This chapter concludes that a unilateral act by the North which lifts barriers to disseminate knowledge related to pollution abatement would be welfare improving for both regions.

Part II

Computing Open-Loop Noncooperative Solution In Discrete Dynamic Games

1.1 Introduction

Economics and other social sciences are concerned with the dynamics arising from the interaction among different decision makers. Their interactions do not always coincide; thus game theoretic considerations become important. Game theory involves multi-person decision making; it is dynamic if the order in which the decisions are made is important and it is noncooperative if each person involved pursues his or her own interests which are partly conflicting with others (Başar and Oldser 1982).

One might argue that ideally all economic problems should be modeled as dynamic games since each individual interacts constantly with others in a society. Thus, many authors who think in this way have sought explicit solutions to dynamic and noncooperative games (eg. Kydland 1975, Pindyck 1977, De Bruyne 1979, Van der Ploeg 1982, Miller and Salmon 1985). However, a closer investigation of the available solutions reveal that most are actually only optimal subject to certain simplifying restriction which permit the derivation of *analytically tractable* decision rules. Because of their mathematical tractability and the possibility of obtaining an analytical solution, the linear quadratic games are well suited for the purpose of deriving necessary and sufficient conditions for a noncooperative equilibrium.

In the last few decades, remarkable progress has been made in adapting control theory to economic problems; and the methods for computing solutions to the multi-person (N-person) problem are obtained by generalizing well known methods of optimal control theory (Starr and Ho 1969). In the traditional control theory, all the ingredients are singular in the sense that there is only one criterion, one central controller coordinating all control actions, and one information set available to the controller (Ho 1970). On the other hand, one can also argue that this viewpoint is unnecessarily narrow. Surely, we can all visualize situations or problems in which there are more than one criterion or performance measure, more than one intelligent controller operating with or without coordination from others and finally all the controllers may or may not have the same information set available to them. Generally, a problem of game theory is not merely a bit harder than its traditional one player counterpart but many times as hard. In the simplest terms, the former has a full, at least two-dimensional matrix; the latter, a single-row matrix. Besides there is no single satisfactory definition of optimality for these N-person 3problems. Depending on the applications, various types of solutions are relevant (eg. open-loop or closed-loop solutions).

In situations where analytical resources fail to cast light, computational simulations of a model can provide much needed *clues* to what constitutes the true behavior of the system in question. This approach has had considerable recent success in many areas of the physical and biological sciences and in mathematics itself. Indeed, whole areas of inquiry owe their existence to the careful examination of well conceived numerical computations (Bona and Santos 1994). The numerical methods were originally developed by control theorists and their chief interest has been testing their capability of solving moderate sized economic problems (Kendrick and Taylor 1970).

Recently, economists are increasingly turning to numerical techniques for analyzing dynamic economic models (Judd 1992). While the progress has been substantial, the numerical techniques have tended to be, or at least have appeared to be, *problem specific*. Hence, this chapter presents a new optimization algorithm as the numerical solution of noncooperative N-person nonzero sum difference games. Difference games are dynamic games in discrete time.

The task of optimizing a complex system such as a dynamic game presents at least two levels of problems. First, a class of optimizing algorithms that are suitable for application to the system must be chosen. Second, various parameters of the optimization algorithm need to be tuned for efficiency (Grefenstette 1986). In this study, a class of adaptive search procedures called *Genetic Algorithm (GA)* has been designed to optimize complex systems such as noncooperative difference games. We will study open-loop Nash equilibrium solution and leave problems of stability, uniqueness etc. aside. The challenge in here is to introduce a general and efficient purpose algorithm to analyze more complex economic models.

The remainder of this chapter is organized as follows: Open-loop noncooperative notations and solutions are described in the Section 1.2. The Genetic Algorithm is introduced in Section 1.3 and the main numerical approximation algorithm is sketched out in Section 1.4. Numerical examples and their results are given in Section 1.5. Finally, a brief conclusion is given in Section 1.6.

1.2 Open-Loop Noncooperative Solution In Discrete Dynamic Games

A solution concept from game theory that has been used frequently in economic applications is the noncooperative solution. Noncooperation implies that each player in the game maximizes his self interest subject to his perception of the constraints on his decision variables. Noncooperative equilibrium solutions to nonzero-sum discrete games were discussed in detail in Kydland (1975), Pindyck (1977), De Bruyne (1979), Brandsma and Hallett (1984) Karp and Calla (1983) and De Zeeuw and Van der Ploeg (1991) and also several references to literature can be found there.

The open-loop solution is a sequence of decisions for each time period and these decisions all depend on the initial state. The open-loop noncooperative solution presumes that at time 0, each player can make binding commitments about the actions he or she announces to undertake in the entire planning period.[1] Each player in the game designs its optimal policy based on its own objectives at the beginning of the period and sticks to that policy throughout the entire period.

In the general N-player, nonzero-sum difference game, the i^{th} player chooses $u_{i1}, u_{i2}, ..., u_{iT}$ trying to maximize (or minimize)

$$J_i = \sum_{t=0}^{T} L_i(u, w, t)$$

subject to

$$x_t^i = f_t^i(x_0, w_1^i, ..., w_t^i, u_{i1}, ..., u_{it}), \quad t = 1, ..., T$$

[1] Here, of course, the term "open-loop noncooperative" should be interpreted in a different context than in standard game theory. In the latter, noncooperative is used to imply the absence of binding commitments whereas in this study, we used the same term to describe that the agents have conflicting interests. The term "open-loop noncooperative" used in this study, however, implies only that the players with conflicting interests have the ability to make binding commitments.

$$x_0, w_1^i, ..., w_T^i \text{ given}$$

where

$$w_t^i = (w_{1t}, ..., w_{i-1,t}, w_{i+1,t}, ..., w_{Nt})$$

is the player i's expectation of the other players' decisions. The possible inequality or equality constraints on the state and the control variables are omitted for simplicity.

Player i has control over u_i only, but he has to consider what the other players do in the game. Thus, the open-loop solution of the optimization problem for player i in period t is the solution of the above problem and solutions for each time period are N mappings

$$x_0, w_1^i, ..., w_T^i \rightarrow u_{it} \quad i = 1, ..., N; \quad t = 1, ..., T$$

derived from the first-order conditions for a minimum. The assumption of noncooperative solution implies that

$$u_t = w_t = g_t^*(x_0), \qquad t = 1, ..., T$$

The quadratic approximation of the objective function with linear constraints has been extensively studied by Kydland (1975) and Pindyck (1977). In previous solutions of open-loop discrete dynamic games, the solution methodologies depend on the assumptions that make the problem's mathematics reasonably tractable. Hence, in almost all studies in this area, each player arrives at its decision using the *same* econometric model (i.e., each has the same view of the way world works), but has a *different set of objectives*, since the possibility of two players having the same set of objectives but each exercising control based on decisions arrived at using different econometric models, is not amenable to solution.

The general linear-quadratic dynamic games use the following procedure to solve such games: There are N players, each with control vector $\mathbf{u_{it}}$ where $i = 1, 2, ..., N$ and $t = 1, ..., T$. The evolution of the state $\mathbf{x_t}$ is given by the linear difference equation

$$\mathbf{x_t = Ax_{t-1} + Bu_t + C}$$

where $\mathbf{x_t}$ is K dimensional and $\mathbf{u_t}$ is N dimensional. The objective of player i is to maximize (or minimize)

$$\mathbf{L_i(x_t, u_t) = \sum_{t=1}^{T}(x_t' Q x_t + u_t' R u_t)}$$

for all $i = 1, 2, ..., N$. The vectors and matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Q}$, and $\mathbf{R}$, are given and at an appropriate dimensions (($\mathbf{Q}$) and ($\mathbf{R}$) are positive symmetric matrices). They indicate the effect on the current state of the previous state ($\mathbf{A}$), current controls ($\mathbf{B}$) and the exogenous change ($\mathbf{C}$); ($\mathbf{Q}$) and ($\mathbf{R}$) give the effect of the current state and the control on player i's single period payoff respectively. The inclusion of the controls in the state vector allows the function $\mathbf{L_i}$ to depend on both the controls and the state.

Since in general $\mathbf{L_i} \neq \mathbf{L_j}$, the players have conflicting objectives. We seek a noncooperative Nash solution to this game by finding a set of N strategies from which no player can unilaterally deviate without decreasing his payoff. Open-loop controls require that at the beginning of the game, each player determines his entire trajectory of controls as a function of time. It is well known that when the objective function is quadratic and the equation of motion linear, optimal controls can be expressed as a linear function of state. Thus, it is not surprising that in the difference game, the equilibrium reaction functions are *linear* in the initial state:

$$\mathbf{u_t} = \mathbf{d_t} + \mathbf{E_t x_0}$$

As in the control problems, $\mathbf{d_t}$ and $\mathbf{E_t}$ are independent of the state, but depend on the parameters of the problem. Very little attention has been given to the development of *computational* techniques to solve N-person games and especially to find open-loop and closed-loop equilibrium controls (Pau 1975). Hence, the chief interest here is to describe a new technique for solving more general problems without making too much simplifications on the environment. The numerical algorithm described is used for the approximation of open-loop Nash equilibrium controls in a difference game of fixed duration and initial state.

1.3 Genetic Algorithm

GAs are search algorithms based on the mechanics of natural selection and natural genetics. *GA* was developed by Holland (1975) in such a way that even in large and complicated search spaces, given certain conditions on the problem domain, *GAs* would tend to converge on solutions that are globally optimal or nearly so (Goldberg 1989). A number of experimental studies have shown that *GAs* exhibit impressive efficiency in practice. While classical gradient search techniques are more efficient for problems which satisfy *tight* constraints (e.g, continuity, low dimensionality, unimodility etc.) Genetic Algorithms consistently outperform both gradient techniques and various forms of random search on more difficult (and more common) problems, such as optimization involving discountinuous, noisy, high dimensional and multimodal objective functions. A *GA* performs a multi-directional search maintaining a population of potential solutions. This population undergoes a simulated evolution: at each generation the relatively "good" solutions reproduce, while relatively "bad" solutions die. Hence, it is an iterative procedure which maintains a constant size population of candidate solutions or structures. During each iteration step, called generation, the structures in the current population are evaluated and on the basis of those evaluations, a new population of candidate solutions are formed. At first glance, it seems strange that such simple mechanisms should motivate anything useful but *GAs* combine partial string to form new solutions that are possibly better than their predecessor. So, a general sketch of the algorithm appears as:

```
procedure GA
begin
t = 0;
  initialize P(t);
  evaluate structures in P(t);
  while termination condition not satisfied do
   begin
   t = t + 1;
     select P(t) from P(t-1);
     recombine structures in P(t);
     evaluate structures in P(t);
   end
end
```

Fig 1: A General Sketch of the Genetic Algorithm

Structures are usually coded as strings of characters drawn from some finite alphabet (often the binary alphabet; 0,1). For example, if we represent structures or solutions with finite lenght of $l = 3$, then we have *eight* possible choice of binary strings with the following interpretations:

String	Interpretation
000	$0x2^0+0x2^1+0x2^2 = 0$
001	$1x2^0+0x2^1+0x2^2 = 1$
010	$0x2^0+1x2^1+0x2^2 = 2$
011	$1x2^0+1x2^1+0x2^2 = 3$
100	$0x2^0+0x2^1+1x2^2 = 4$
101	$1x2^0+0x2^1+1x2^2 = 5$
110	$0x2^0+1x2^1+1x2^2 = 6$
111	$1x2^0+1x2^1+1x2^2 = 7$

GAs use vocabulary borrowed from natural genetics, thus the structures or decision rules in a population are called *chromosomes* or *strings*. In our game context, a set of strings would be interpreted as a set of strategies or optimal plans. The performance of the strategies or the decision rules in a given environment is evaluated through their *fitness* functions. In economic modeling, the fitness function measures the value of profit or utility resulting from the behavior prescribed by a given rule or rules. The rules are updated using a set of genetic operators which include *reproduction, crossover, mutation,* and *election.*

Reproduction makes copies of individual chromosomes. The criterion used in copying is the value of the fitness function. This operator is an artificial version of natural selection, a survival of the fittest by Spencer among string creatures. In natural population, fitness is determined by a creature's ability for survival and subsequent reproduction.

The primary genetic operator for *GA* is the crossover operator. The crossover operator is executed in three steps: (1) a pair of strings is chosen from the set of copies; (2) the strings are placed side by side and a point is randomly chosen somewhere along the length of the strings; (3) the segments to the left of the point are exchanged between the strings. For example, a crossover of 111000 and 010101 after the second position produces the offsprings **01**1000 and **11**0101. Crossover, working with reproduction according to performance, turns out to be a powerful way of biasing the system towards certain patterns.

Mutation is a secondary search operator which increases the variability of the population. After selection of one individual, each bit position (allele in the chromosome) in the new population undergoes a random change with a probability equal to the mutation rate. For example, after mutation individual 111000 becomes 101000 since the second bit position undergoes a change. A high level of mutation yields an essentially random search.

Election tests the newly generated offsprings before they are permitted to become members of a new population. The *potential* fitness of an offspring is compared to the *actual* fitness values of its parents. Parents are the pairs of strings that are taken from mating pool for the crossover application. A pair is randomly matched and mated; thus the parents and children or offsprings form the new population after election.

The algorithm starts by selecting a random sample of M strings (M decision rules), and then applying four operators sequentially. After a new population is created via mating operator, the algorithm applies the same four operators again, continuing either for a prespecified number of rounds or until a stable population of string values or decision rules emerges. As the "solution" of the original, select the fittest member from the final population.

The reproduction operator increases the representation of relatively fit individuals in the population, but does nothing to find a *fitter* individual. The mutation and mating operator (crossover operator) can add new elements to the population, while destroying old ones. If mutation is applied too frequently (mutation probability is too high), it slows or prevents convergence and degrades the performance of the algorithm because it destroys the fit individuals along with the unfit. The mating operator seems to be a very good device for probabilistically injecting diversity, while giving structures that have proved their fitness a shot at surviving.

This algorithm has proved its value in a variety of applications (Sargent 1993). It has some features of a parallel algorithm, both in obvious sense that it simultaneously processes a sample distribution of elements, and in the subtler sense that instead of processing individuals, it is really processing equivalence classes of individuals. These equivalence classes, which Holland calls *schemata*, are defined by the lengths of common segments of bit strings. The algorithm is evidently a random search algorithm, one that does not confine its searches locally.[2]

[2] For application of *GA* to economic problems, see Marimon *et al.* 1990, Marks 1992, Arifovic 1994a,1994b

1.4 Algorithm For Searching Open-Loop Noncooperative Solutions

GAs aim at complex problems. They belong to the class of probabilistic algorithms, yet they are different from random algorithms as they combine elements of stochastic search. The advantage of such genetic-based search methods is that they maintain a population of potential solutions whereas all other methods process a single point of search space. Thus a *GA* performs a multi-directional search by maintaining a population of potential solutions and encouraging information formation and exchanges between these directions. In the game theoretic framework, we use both the optimization and the learning property of the *GA*. At each generation or step, players play the whole game and the scores are rated.

For presentational simplicity, we will restrict attention to two-player games here, but the algorithm can be generalized to N players, each with a different objective function. Thus, in this environment, there are two artificially intelligent players who update their strategies through *GA* and a fictive player who has full knowledge of both players' actions. The term "fictive player", or "referee", is not essential but we need an intermediary for the exchange of best responses of each player to the action of other player in each generation and this intermediary is called fictive player in the study[3]. This player has no decisive role but provides the best strategies in each iteration to the requested parties synchronously.

In this environment, we have two separate *GAs*. Each *GA* is used to play one side of the problem and each side has its own evaluation function (utility function, profit function or cost function) and population. The evaluation functions of each player have different parameters and functional form depending on the problem of that player. The crucial part in this algorithm is that the problems of each player are solved *synchronously*. Since we have two different players with different objectives, the problem complexity of each player varies. Thus, based on the complexity of the fitness function, one-player might evaluate the performance of his or her strategies faster than the other player. However, in order to learn the action of the other player against his or her strategies, each player waits for the other player's action in each

[3]In the *UNIX* system, in order to reach and distribute the information available we used ***shared memory*** and called this memory as a fictive player.

generation. Thus the game must be played synchronously and genetic operators must be applied sequentially to each generation (see Figure 2).

Another crucial part is that the best response of each player is available immediately and each player decides his own strategies according to the best strategies of the other player. This is the learning process provided by *GA*. The information of the best strategies are kept in the *shared memory* which is controlled by the fictive player. Each player solves his or her problem and sends the information about the best decision rules or solutions to the shared memory. Then each player copies the result of the other player's solution and solves his problem again through *GA*.

Initial decision rules are generated randomly for the entire planning period and in order to measure the performance of these randomly generated rules, each player needs the actions of the other player. Hence, each player sends the very first decision rules generated randomly to the shared memory and the performance of the randomly generated rules are evaluated. Since *GA* maintains constant size population of candidate solutions, we have initially M solutions for each time period decision rule. For example, for the two-period problem, we have two decision rules to find, hence initially 2 times M solutions are generated for each player to start the game.

1.5 Numerical Examples

1.5.1 A Simple Example

Usually, numerical optimizations and simulations of a model constitute an experiment and consequently, they should be performed and evaluated with some sort of critical eye that is appropriate to a laboratory or field experiment. Hence we will offer technical information to the interested researchers for the application of the algorithm for other discrete dynamic games.[4].

A simple numeric example is taken from Kydland (1975). Assume that the problems of two noncooperative players are

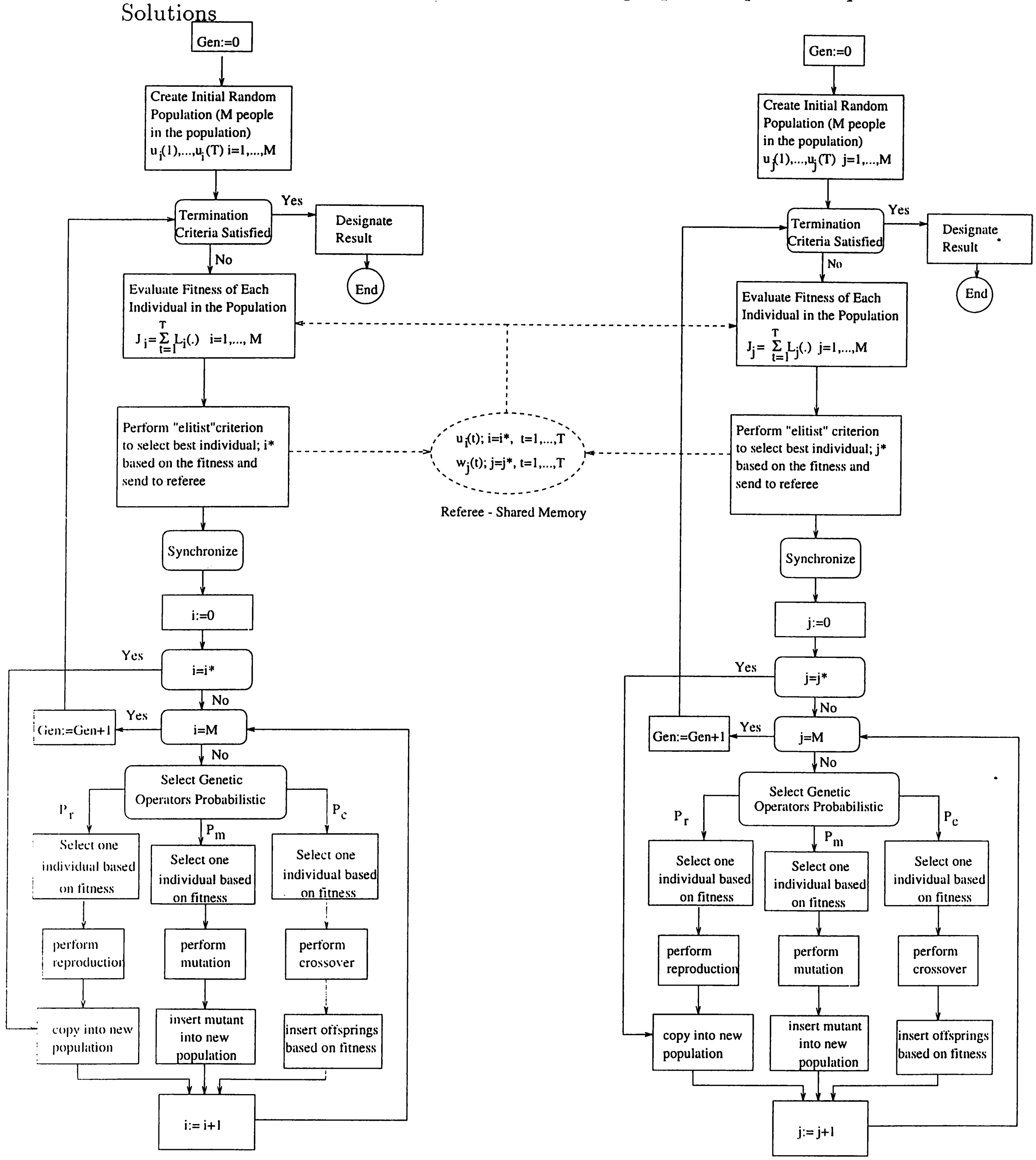
$$max \sum_{t=1}^{2}(1 - x_{1t} - x_{2t})x_{it} - \frac{1}{2}u_{it}^2$$

subject to

$$x_{it} = x_{i,t-1} + u_{it}, \quad x_0 \text{ given}, \quad i = 1,2$$

[4]Upon request, I will provide the C code of the algorithm by e-mail. My e-mail address is suheyla@bilkent.edu.tr

Figure 2: Flowchart of Algorithm for Searching Open-Loop Noncooperative Solutions



The maximization is over decision rules; u_{i1} and u_{i2} for $i = 1, 2$. The analytical solution of open-loop decision rules for player i $(i = 1, 2)$[5] are

$$u_{i1} = -0.6947x_{10} - 0.0947x_{20} + 0.2632$$

$$u_{i2} = -0.1789x_{10} + 0.0211x_{20} + 0.0526$$

If the initial state variables were $x_{10} = x_{20} = 0.1$, then the open-loop solutions for player 1 would be $u_{11} = 0.1842$ and $u_{12} = 0.0368$.

The algorithm developed in this study starts with the determination of the *GA* parameters which affect the convergence property of the algorithm. However, since the aim of this study is to obtain a general purpose algorithm for N-person difference game, we tried to use parameters which are generalized in the most *GA* experiments.

1.5.1.1 The Space of Genetic Algorithm

Holland (Grefenstette 1986) describes a fairly general framework for the class of *GAs*. There are many possible elaborations of *GAs* involving variations such as other genetic operators, variable sized population etc. This study is limited to a particular subclass of *GAs* characterized by the following parameters; *population size M, crossover rate p_c, mutation rate p_m, generation gap G, selection strategy S*. Population size affects both performance and efficiency of *GAs*, and a large population is more likely to contain representatives from a large number of hyperplanes. Crossover rate controls the frequency with which the crossover operation is applied. Mutation, which is the secondary search operator, increases the variability of the search spaces. The generation gap controls the percentage of population to be replaced during each generation. The selection strategy is the *elitist strategy, E* which stipulates that the structure with the best performance always survives intact into the next generation. In the absence of such a strategy, it is possible for the best structure to disappear due to sampling error, crossover and mutation.

In this study, we use parameters as $GA = GA(50, 0.6, 0.03, 1.0, E)$. The parameters are population size, crossover rate, mutation rate, generation gap and selection strategy respectively. For having an initial population which is generated randomly, player 1 uses random seed 123456789 and player 2 uses 987654321.

[5] Since the game is symmetric, solutions are same for each player.

1.5.1.2 Computational Details

The example discussed above needed little computer power; but, as the aim in this chapter is to study more complex problems, we worked in the *UNIX* operating system. All the programs are written in the *C* programming language. There are various versions of *UNIX* in use today and *AT&T Unix System V* which was merged under *Sun* Microsystems *SunOS* (late 1989) is one of the predominant version. Here, we will give running time under *SunOS*.

Furthermore, we used publicly documented *Genetic Search Implementation System; GENESIS* (see Appendix) developed by Grefenstette for the application of genetic operators. Although the *Sun* microcomputers are used in this study, it is always possible to run all programs in *IBM PC's* using *Turbo C*.

1.5.1.3 Experiment and Results

The example given in Kydland is two-period, two-player noncooperative game. Hence, we have to find two optimal strategies; u_{i1} and u_{i2} $(i = 1, 2)$ for each player which maximize the fitness functions of the two players. The fitness function f_i which measures the performance of each individual strategy in the population, is obtained by substituting constraints into the objective function:

Player 1:

$$\begin{aligned} f_1 &= (1 - x_{10} - u_{11} - x_{20} - u_{21})(x_{10} + u_{11}) - \frac{1}{2}u_{11}^2 + \\ &\quad (1 - x_{10} - u_{11} - u_{12} - x_{20} - u_{21} - u_{22})(x_{10} + u_{11} + u_{12}) - \frac{1}{2}u_{12}^2 \end{aligned}$$

Player 2:

$$\begin{aligned} f_2 &= (1 - x_{10} - u_{11} - x_{20} - u_{21})(x_{20} + u_{21}) - \frac{1}{2}u_{21}^2 + \\ &\quad (1 - x_{10} - u_{11} - u_{12} - x_{20} - u_{21} - u_{22})(x_{20} + u_{21} + u_{22}) - \frac{1}{2}u_{22}^2 \end{aligned}$$

It is worth noting that *GA* uses the original problem *not* the first-order conditions for decision making. As mentioned before, the game starts with the generation of 50 decision rules; u_{i1}, u_{i2} for each player $(i = 1, 2)$. For the evaluation of these decisions, each player needs to know the other player's decisions for two periods. Hence, when the random numbers are generated as the potential decision rules of each player, the very first individual decision set in the population is immediately sent to the shared memory by each player and is used as the best decisions of that player for that generation:

Gen	**Individual**	u_{11}	u_{12}	u_{21}	u_{22}
0	1	0.341924	0.400504	0.285466	0.057359

The fitness functions for generation 1 are calculated for the whole population by using these values as the best decision rules of each player. From Table 1.1, we can follow the generation of new individuals and their convergence to the optimal decisions for each player:

Table 1.1: Convergence to the Optimal Decisions

Gen	u_{11}	u_{12}	f_1	u_{21}	u_{22}	f_2
0	0.055772	0.094682	0.225860	0.105699	0.015076	0.242384
1	0.177234	0.044419	0.221428	0.194690	0.012314	0.221126
5	0.177234	0.044419	0.218435	0.188846	0.032502	0.221697
10	0.177127	0.043458	0.218440	0.188846	0.032502	0.222071
15	0.179187	0.044419	0.214624	0.189212	0.043656	0.220319
20	0.179187	0.044312	0.214872	0.188846	0.043580	0.220370
25	0.179080	0.040299	0.218371	0.188891	0.032761	0.221896
30	0.179080	0.040299	0.218371	0.188891	0.032761	0.221896
35	0.183627	0.041459	0.220636	0.184390	0.034211	0.218788
40	0.183627	0.039750	0.220649	0.184390	0.034211	0.219333
45	0.183627	0.039750	0.220358	0.184390	0.035111	0.219336
50	0.183627	0.039750	0.220518	0.184085	0.035187	0.219336
200	0.184115	0.037003	0.219864	0.184375	0.036698	0.219923
400	0.184207	0.036835	0.219916	0.184207	0.036851	0.219921
600	0.184207	0.036835	0.219912	0.184222	0.036835	0.219921
800	0.184207	0.036851	0.219912	0.184222	0.036835	0.219916
1000	0.184207	0.036851	0.219921	0.184207	0.036835	0.219916
1200	0.184207	0.036851	0.219916	0.184207	0.036851	0.219916
1400	0.184207	0.036851	0.219916	0.184207	0.036851	0.219916
1495	0.184207	0.036851	0.219916	0.184207	0.036851	0.219916

At trial 50000, the open-loop decision rules for player 1 and player 2 has almost been reached by GA. However, we ran 25000 more trials which takes less than 2 more minutes, to decide that GA converged to the optimal rules. When compared to the exact solutions, it is apparent that GA works. The theory behind why GA works depends on the idea that an optimally intelligent system should process currently available information about payoff from the unknown environment so as to find the optimal tradeoff between cost of *exploration* of new points in the search space and the cost of *exploitation* of already evaluated points in the search space.

1.5.2 A Wage Bargaining Game

A typical example of policy optimization in a policy game is given by Brandsma and Hughes Hallett (1984) using the formulation by L.R. Klein to describe the mechanism of the American economy in the interwar period, 1933-1936. Theil (1964) reduced Klein's highly aggregative equation system into three instruments and three noncontrolled (target) variables. The targets (consumption C, investment I, and income distribution D) are linked to three instruments (public sector wages W_2, the indirect tax yield T, and government expenditures G). The decision problem of Roosevelt is to resolve the depression and to return economic activity per head to at least its previous peak (1929) level. Brandsma and Hughes Hallett partition the available instruments as W_2 for one player (organized labor or union) and T and G for second player (government or budgetary authority). The idea is to examine the optimal policies in a noncooperative game between the Roosevelt administration and organized labor as the economic strategy of the New Deal was introduced to counter the great recession of that period. Organized labor was one of the chief groups in the coalition which brought Roosevelt to power. It had suffered particularly from the recession and unemployment after 1929. So, labor could have been expected to attempt to extract a price for its continued support. Its instruments would be power to influence wage demands and hence income distribution as well as the industrial relations in public and private sector (Hughes Hallett and Rees, 1983). In Theil's exercise, current welfare is loosely represented by consumption; future welfare by investment, and the distribution of income between capital and labor reflects the New Deal commitment to organized labor.

By the above, desired consumption in 1936 is $C_{36}^d = C_{29}(1+\alpha)^7$ where $\alpha = 0.1$ is the observed population growth rate and the subscripts denote the year ($C_{29} = 57.8$ in billions of dollars of 1934, is the realized level of total consumption in 1929). As to investment, I, it appears that this variable was of the order of 10 percent of consumption during 1920s, and hence, desired investment in 1939, averaged $I^d_{36} = 0.1C^d_{36}$. Theil took $D = W_1 - 2\pi$ as the income distribution where W_1 is private wage bill and π is profit, and put its desired level in 1936 equal to zero; $D^d_{36} = 0$. To specify desired values for the intermediate years; 1933-1935, simple linear interpolation is done between the actual values in 1932 (the last year before President Roosevelt administration) and the corresponding desired values in 1936. Hence, $C^d_{32+t} = C_{32} + \frac{t}{4}(C^d_{36} - C_{32})$, $I^d_{32+t} = I_{32} + \frac{t}{4}(C^d_{36} - C_{32})$ and $D^d_{32+t} = D_{32} + \frac{t}{4}(D^d_{36} - D_{32})$ for $t = 1, ..., 4$. A game naturally arises here through conflicting private interests of the government (or employer) and organized labor for rebuilding consumption and investment. Organized labor would press for faster increases in the wage bill W_1 in order to restore their last earnings and employment levels. But employers would demand a restoration of profits (by 1932, D was large and positive) in order to boost investment and the government would have to admit in order to secure the welfare. That implies the employees would have ideal values for D which remain above Theil's smooth restoration of status quo ante; while the employers and government would set values which imply a faster return to zero. This shows that administration was fully aware of this conflict and the need to resolve it.

The desired level of instruments is also handled in an analogous manner. Following the trend argument, the desired values are projections of the observed trends in the associated variables over 1920-32. The numerical specification of these values are given in Table 1.2 (Brandsma and Hughes Hallett, 1984).

Table 1.2: Desired Values of Instruments and Target Variables

Policy	1933	1934	1935	1936
Labor				
C	49.49	53.78	57.88	61.97
I	-3.10	0.00	3.10	6.20
D	12.50	10.00	7.50	5.00
W_2	5.04	5.25	5.47	5.69
Government				
C	49.49	53.78	57.88	61.97
I	-3.10	0.00	3.10	6.20
D	0.00	0.00	0.00	0.00
T	7.40	7.64	7.87	8.11
G	10.44	10.87	11.30	11.73

The objective function was formulated as the minimization of the sum of squares of the deviations between actual and desired values of variables which are interests of each player. Thus, the private interests of each player i, can be represented by the quadratic loss function,

$$w^{(i)} = \frac{1}{2}(\tilde{y}^{(i)\prime} B^{(i)} \tilde{y}^{(i)} + \tilde{x}^{(i)\prime} A^{(i)} \tilde{x}^{(i)}), i = 1, 2$$

where $\tilde{y}^{(i)} = y^{(i)} - y^{(i)^d}$ is the vector of target variables, $\tilde{x}^{(i)} = x^{(i)} - x^{(i)^d}$ is the vector of instruments, and $B^{(i)}$ and $A^{(i)}$ are positive definite symmetric matrices. These matrices; $B^{(i)}, A^{(i)}$ denote penalties which imply that accelerating unit penalties accrue to persistent or cumulating failures in electorally significant variables. The private objective functions were specified by picking out the relevant penalties from those revealed in the historical policy decisions (Ancot et al. 1982). The objective function is summarized in Table 1.3.

Table 1.3: The Preference Structure: penalties on squared failures

Policy	1933	1934	1935	1936	Preference Matrix
Labor					
C	1.0	1.0	1.0	1.0	B
I	0.01	0.3	1.0	0.5	B
D	0.5	0.5	2.0	5.0	B
W_2	1.0	2.0	1.0	5.0	A
Government					
C	1.0	2.0	5.0	4.0	B
I	0.1	0.3	1.0	0.5	B
D	0.5	0.4	2.0	0.25	B
T	1.0	0.1	0.1	0.8	A
G	2.0	1.0	1.0	1.0	A
T and G	0.25	0.25	0.0	-0.4	A
C and I	0.05	0.0	1.0	0.0	B
C and D	0.3	0.0	-1.0	-1.0	B

Intertemporal Penalties:		
T_{35} and T_{36}:-0.1	T_{35} and G_{36}:-0.1	G_{33} and T_{36}:-0.5
G_{33} and G_{36}:0.5	G_{35} and T_{36}:-0.2	G_{35} and $G36$:0.3
C_{33} and C_{35}:0.5	C_{33} and C_{36}:0.5	C_{35} and C_{36}:2.0
D_{35} and C_{36}:-1.0	C_{33} and G_{36}:0.5	

Player 1 has one instrument $x^{(1)} = W_2$ while player 2 has two instruments $x^{(2)} = T, G$ to play. Also, each decision maker has ideal values $y^{(i)^d}, x^{(i)^d}$ for his own decision variables, so that $\tilde{y}^{(i)} = y^{(i)} - y^{(i)^d}$ and $\tilde{x}^{(i)} = x^{(i)} - x^{(i)^d}$ define his policy failures. Thus each player's aim is to minimize his failures subject to constraints, implied by the econometric model:

$$y^i = R^{i,1}x^1 + R^{i,2}x^2 + s^i, i = 1,2$$

where $R^{i,j}, j = 1,2$ are known matrices containing the dynamic multipliers $R^{i,j}_{tk} = \partial y^i_t / \partial x^j_k$ if $t \geq k$ and zeros otherwise and where s^i is the subvector of s associated with y^i and denotes structural disturbances. Thus $R^{i,i}$ describes the response of player i's targets to his own instruments and $R^{i,j}$ their responses to player j's decisions. Since each player has the same noncontrollable variables, the constraint equation can be written in aggregate form:

$$y = Rx + s$$

or more exclusively, as

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} R_1 & & & \\ R_2 & R_1 & & \\ R_3 & R_2 & R_1 & \\ R_4 & R_3 & R_2 & R_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

where $x' = (x^{(1)'}, x^{(2)'})$ is the vector of instrument variables and the coefficient matrices R_1 measure the effectiveness of instruments with respect to current uncontrolled variables, hence they are equal to submatrices $R^{(i)}_{11}, R^{(i)}_{22}, R^{(i)}_{33}, R^{(i)}_{44}$. In the same way, the matrix R_2 measures the effectiveness of the instruments with respect to uncontrolled variables one year later, hence it is equal to submatrices $R^{(i)}_{21}, R^{(i)}_{32}, R^{(i)}_{43}$. And so on. Thus, the submatrices of the multiplicative and the additive structure of the constraint can be summarized as in Table 1.4 (Theil, 1964).

Evidently, each player's optimal strategy depends on and must be determined simultaneously with the optimal decisions to be expected from the other player. In the absence of cooperation, the optimal decisions $(x^{(1)^*}, x^{(2)^*})$ will satisfy

$$w^{(i)}(x^{(i)^*}, x^{(j)^*}) \leq w^{(i)}(x^{(i)}, x^{(j)^*})$$

for $i = 1,2; i \neq j$ for all feasible $x^{(i)} \neq x^{(i)^*}$.

Table 1.4: The Multiplicative and Additive Structure of the Constraints

	W_2	T	G	
		R_1		s_1
C	0.666	-0.188	0.671	37.55
I	-0.052	-0.296	0.259	-5.62
D	0.285	2.358	1.427	5.04
		R_2		s_2
C	-0.234	-1.014	1.170	35.18
I	-0.152	-0.894	0.759	-4.64
D	0.095	1.172	-0.475	0.08
		R_3		s_3
C	-0.172	-1.006	0.859	36.55
I	-0.076	-0.518	0.382	-2.49
D	-0.007	0.186	0.033	-3.61
		R_4		s_4
C	-0.079	-0.543	0.396	42.84
I	-0.005	-0.088	0.024	2.11
D	-0.060	-0.285	0.301	-11.91

1.5.2.1 Optimal Strategies

The numerical result of this wage-setting game by *GA* is summarized in Table 1.5.

Table 1.5: Open-Loop Nash Strategies in 1934 billions dollar

Policy	1933	1934	1935	1936
Instruments:				
W_2	5.621	6.591	11.505	7.621
T	5.139	6.099	4.805	8.122
G	11.690	10.929	10.001	11.365
Target Strategies:				
C	48.172	52.908	58.985	61.666
I	-4.406	-0.533	1.297	4.115
D	2.077	1.748	0.612	-1.665
$w^{(1)} = 237.187$				
$w^{(2)} = 10.765$				

Before examining the result, the technical points can be summarized as follows; population size is 50, crossover rate is 0.6, mutation rate is 0.03 and finally, number of trials is 2 millions. The optimal strategies are tested whether or not they satisfy necessary conditions, first and second order conditions. Since *GA* does not use first-order conditions for decision making, we have to test whether the results of GA satisfy these conditions derived in the original paper. Then the necessary conditions for an optimal strategy to hold is the following:

$$\partial w^{(i)}/\partial x^{(i)} + (\partial y^{(i)}/\partial x^{(i)})'\partial w^{(i)}/\partial y^{(i)} = 0$$

for all the estimated decision rules $x^{(i)}$'s. We found that the first-order conditions hold with 0.003 error. Since this is a numeric study, the results are all near optimal and quite convincing.

The results show rather stable values for the government expenditure on goods and services. The values are consistently above their desired level which is, of course, in accordance with generally accepted ideas about anti-depression policies. Regarding the uncontrollable variables; C and I, we observe that strategy values are below the desired levels, which is also in accordance with a depression situation.

1.6 Why Does GA Work?

One method of obtaining the open-loop Nash equilibrium solutions of the class of discrete time games is to view them as static infinite games, and directly apply the methodology that minimizes cost functional over control sets and then determines the intersection points of the resulting reaction curves. Such an approach can sometimes lead to quite unwieldy expressions, especially if the number of stages in the game is large (Başar and Oldser 1982). An alternative derivation which partly removes this difficulty is the one that utilizes techniques of optimal control theory, by making explicit use of the stage additive nature of the cost functionals and the specific structure of the extensive-form description of the game (For a more general treatment see Başar and Oldser (chapter 6) 1982). There is, in fact, a close relationship between the derivation of open-loop Nash equilibria and the problem of solving (jointly) N optimal control problems since each N-tuple of strategies constituting a Nash equilibrium solution describes an optimal control problem whose *structure* is not affected by the remaining players' control vectors. This structure of dynamic or differential games enables GA to work in deriving optimal open-loop strategies.

Amenability to parallelization is an appealing feature of the conventional genetic algorithm. In genetic methods, the genetic operations themselves are very simple and not very time consuming; whereas the measurement of fitness of the individuals in the population is typically complicated and time consuming. In considering approaches for parallelizing genetic methods, it is important to note that for all but the most trivial problems, the majority of the computational effort is consumed by the calculation of fitness. The execution of the genetic operation can be parallelized in the manner described by Robertson (1987); however, we focus here on ways of parallelizing genetic programming that distribute the computational effort needed to compute fitness. There are some basic approaches to parallelization. In one of the approaches called *distributed genetic approach*, the population for a given run is divided into subpopulations (Koza 1993). Each subpopulation is assigned to a processor, and the genetic algorithm operates on each subpopulation separately. Upon completion of a certain designated number of generations, a certain percentage of the individuals in each population are selected for emigration, and there is a partial exchange of members between the subpopulations.

The algorithm developed in this chpater follows a similar idea while exchanging the best results in each generation. We used the property of problem independence of *GA* and solved open-loop strategies as an N joint optimal control problem. The main goal in this study is to show the success of the *algorithm* developed here in solving nonlinear dynamic games with some empirical evidence. Başar and Oldser proved that an N-person linear quadratic dynamic game with appropriate dimensional matrices, has a unique open-loop Nash equilibrium solution. For problems that violate the appropriate dimensionals, we apply numerical methods. *GA* offered here is an efficient technique to obtain open-loop Nash-equilibrium solutions of the finite-horizon discrete-time dynamic games.

Future research might profitably examine the closed-loop noncooperative solutions. For closed-loop solution procedure, we need to work on the objective function of each player separately to make selection. So, in this case, the most important part will be to devise a more efficient *selection* procedure. In our open-loop procedure, we were selecting new population of each player at the same time but in the closed-loop, we might offer intersection or union of the selections done separately to form new population and apply other genetic operators. Each individual's gene will be the one attached to another (*concatenation*) of the genes from what we have in the two population shared memory algorithm. So, to use *GA* to solve closed-loop we have to write a new selection procedure such as: Initially using the first players objective function, we make selection and form set of solutions of that player and then doing the same procedure, we form the other player's set of solutions. By either intersection or union of these sets, we form a new set of solutions but of course, both have some difficulties such as having an empty set in the intersection. However, the critical in the solution procedure for the closed-loop games in GA will be selecting the new population set. We have to further study whether intersection or union will be the suitable selection procedure.

Appendix

A.1 *Setup Program of GENESIS*

The GENEtic Search Implementation System describes a system for function optimization based on genetic search techniques. The aim of this program is to promote the study of genetic algorithms for function minimization *only*. Only genetic operators as a procedure for various purposes are done automatically. Then the researchers design their program using *C* language.

Before running the GA, execute the "setup" program, which prompts you for a number of input parameters for genetic operators. All of this information is stored in files for future use, so you may only need to run "setup" once. A <return> response to any prompt gets the default value shown in brackets. The prompts are as follows:

– the suffix for file names :

If a string is entered, say "foo", then the files for this run will have names like "in.foo", "out.foo", "log.foo", etc. Otherwise, the file names are "in", "out", "log", etc.

– Floating point representation [y]:

Unless this is declined, the user will be asked the specify the

– number of genes:

Each gene will take on a range of floating point values, with a user-defined granularity and output format. The user will be asked to specify for each gene: its minimum value; its maximum value; the number of values (must be a positive power of 2); the desired output format for this gene (using printf format, e.g., also specify a repetition count, meaning that there a number of genes with the same range, granularity, and output format. When all genes have been specified, the information is stored in the "template" file, and Setup prompts for:

– the number of experiments [1]:

This is the number of independent optimizations of the same function.

– the number of trials per experiment [1000]:

– the population size [50]:

– the length of the structures in bits [30]:

If the "f" (floating point representation) option is selected, this number will computed automatically from the information collected above.

– the crossover rate [0.60]:

– the mutation rate [0.001]:

– the generation gap [1.0]:

The generation gap indicates the fraction of the population which is replaced in each generation.

– the scaling window [5]:

When minimizing a numerical function with a GA, it is common to define the performance value u(x) of a structure x as u(x) = fmax - f(x), where fmax is the maximum value that f(x) can assume in the given search space. This transformation guarantees that the performance u(x) is positive, regardless of the characteristics of f(x). Often, fmax is not available a priori, in which case we may define u(x) = f(xmax) - f(x), where f(xmax) is the maximum value of any structure evaluated so far. Either definition of u(x) has the unfortunate effect of making good values of x hard to distinguish. For example, suppose fmax = 100. After several generations, the current population might contain only structures x for which 5 < f(x) < 10. At this point, no structure in the population has a performance which deviates much from the average. This reduces the selection pressure toward the better structures, and the search stagnates. One solution is to define a new parameter Fmax with a value of, say, 15, and rate each structure against this standard. For example, if f(xi) = 5 and f(xj) = 10, then u(xi) = Fmax - f(xi) = 10, and u(xj) = Fmax - f(xj) = 5; the performance of xi now appears to be twice as good as the performance of xj. The scaling window W allows the user to control how often the baseline performance is updated. If W > 0 then the system sets Fmax to the greatest value of f(x) which has occurred in the last W generations. A value of W = 0 indicates an infinite window (i.e. u(x) = f(xmax) - f(x)).

– the number of trials between data collections [100]:

– how many of the best structures should be saved [10]:

– how many consecutive generations are permitted without any evaluations occurring [2]:

– the number of generations between dumps [0]:

0 indicates no dumps will occur.

– the number of dumps that should be saved [0]:

– the options (see below) [cefgl]:

The option "c" collects statistics concerning the convergence of the algorithm, "e" uses the "elitist" selection strategy, "f" uses the floating point representation, "g" uses Gray code. A Gray code is sometimes useful in representing integers in *GAs*, "l" log activity (starts and restarts) in the "log" file. Some error messages also end up in the "log" file. integers in genetic algorithms. Other used options in this experiment are "M" maximizes the evaluation function. The default is to minimize the evaluation function and "a" evaluates all structures in each generation.

– the seed for the random number generator [123456789]:

– RankMin [0.75]:

This is the minimum expected number of offspring for ranking (used only if option "R" is set). The Ranking selection algorithm used here is a linear mapping under which the worst structure is assigned RankMin offspring and the best is assigned (2 - RankMin).

Setup then echoes the input file, and exits. The setup program should be run at least once for each new evaluation function, but after that, it may be more convenient to simply edit the input file to make minor changes to the parameters.

Appendix

A.1. *C codes of the main program for the game introduced by Kydland*

Player 1:[6]

```
#include "extern.h"
#include "semop.h"

#define SHMKEY_P ((key_t) 9894)
#define SHMKEY_S ((key_t) 9895)
#define SEMKEY1 ((key_t) 8393)
#define SEMKEY2 ((key_t) 8394)
#define SEMKEY3 ((key_t) 8395)

#define NORTH          1

#define TMAX  2
#define Y10   0.1
#define Y20   0.1

typedef double Mesg[TMAX];


int shmid;
int shmsem,syncsem1,syncsem2;
Mesg *x2;
Mesg *x1;


getshared()
{
     if((shmid=shmget(SHMKEY_P,sizeof(Mesg),0666 | IPC_CREAT))<0)
          {
          printf("server can't get shared memory");
          exit(1);
          }
     if((x2=(Mesg *) shmat(shmid,(char*)0,0)) == (Mesg *) -1)
          {
          printf("server can't attach shared memory");
          exit(1);
          }
     if((shmid=shmget(SHMKEY_S,sizeof(Mesg),0666 | IPC_CREAT))<0)
          {
          printf("server can't get shared memory");
          exit(1);
          }
     if((x1=(Mesg *) shmat(shmid,(char*)0,0)) == (Mesg *) -1)
          {
```

[6] Since the original problem is symmetric, we do not need to put the program for Player 2

```
            printf("server can't attach shared memory");
            exit(1);
            }
}

relshared()
{
      if(shmdt(x2)<0)
            {
            printf("server can't detach shared memory");
            exit(1);
            }
      if(shmdt(x1)<0)
            {
            printf("server can't detach shared memory");
            exit(1);
            }
          sem_signal(syncsem1);
          sem_signal(syncsem2);
          sem_close(syncsem1);
      sem_close(syncsem2);
      sem_close(shmsem);
}

initshared()
{
#if NORTH
      if ( (syncsem1 = sem_create(SEMKEY1, 0)) < 0)
            err_sys("north: can't create sync semaphore");

          if ( (syncsem2 = sem_create(SEMKEY3, 0)) < 0)
            err_sys("north: can't create sync semaphore");

      if ( (shmsem = sem_create(SEMKEY2, 1)) < 0)
            err_sys("north: can't create shm semaphore");
#else
      if ( (syncsem1 = sem_open(SEMKEY1)) < 0)
            err_sys("south: can't create sync semaphore");

      if ( (syncsem2 = sem_open(SEMKEY3)) < 0)
            err_sys("south: can't create sync semaphore");

      if ( (shmsem = sem_open(SEMKEY2)) < 0)
            err_sys("south: can't create shm semaphore");
#endif
```

```
        sem_wait(shmsem);
        Unpack(New[0].Gene, Bitstring, Length);
        if (Floatflag)
                {
                FloatRep(Bitstring, Vector, Genes);
#if NORTH
                memcpy(x1,Vector,sizeof(double)*TMAX);
#else
                memcpy(x2,Vector,sizeof(double)*TMAX);
#endif
                }
        sem_signal(shmsem);
           synchronize();
}


synchronize()
{
#if NORTH
           sem_signal(syncsem1);
           sem_wait(syncsem2);
#else
       sem_wait(syncsem1);
       sem_signal(syncsem2);
#endif
}


movebesttoshared()
{
           sem_wait(shmsem);
       Unpack(New[Best_guy].Gene, Bitstring, Length);
       if (Floatflag)
               {
               FloatRep(Bitstring, Vector, Genes);
#if NORTH
               memcpy(x1,Vector,sizeof(double)*TMAX);
#else
               memcpy(x2,Vector,sizeof(double)*TMAX);
#endif
               }
      sem_signal(shmsem);
}
```

```
double eval(str, length, vect, genes)
char str[];        /* string representation            */
int length;        /* length of bit string             */
double vect[];     /* floating point representation    */
int genes;         /* number of elements in vect       */

{
        register int t;
        double ans=0;
        double y1[TMAX];
        double y2[TMAX];

        sem_wait(shmsem);

        for (t = 0; t<TMAX; t++)
                {
                if (t==0)
                        {
                        y1[t]=Y10+vect[t];
                        y2[t]=Y20+(*x2)[t];
                        }
                else
                        {
                        y1[t]=y1[t-1]+vect[t];
                        y2[t]=y2[t-1]+(*x2)[t];
                        }
                ans+=(1-y1[t]-y2[t])*y1[t]-0.5*vect[t]*vect[t];
                }
        sem_signal(shmsem);
        return(ans);
}
```

Three-Country Trade Relations: A Discrete Dynamic Game Approach

2.1 Introduction

In recent years, countries have grouped for international economic cooperation with less than full success. Relapses into conflicting policies have been frequent. Progress appears to be better achieved in certain regions (the North) than in others such as poor regions (the South). This may be due to closer and more frequent contacts that, at the international level, are institutionally possible among advanced nations. In the Southern countries cooperation is limited by two factors- institutional deficiences and trade barriers. As in the North, as well as a dismantling of trade barriers, the South needs institutions to facilitate cooperative trade. Specialization and trade within an industry across national frontiers is difficult to organize without institutions which operate easily across nations. A number of common markets and regional trading agreements have tried to provide the required trading infrastructure and to reduce trade restriction within the South.

In an interdependent world, rational policymakers in one country may be expected to condition their actions on policies pursued in other countries; policymaking has unavoidable game aspects. In the absence of direct cooperation, it is well known that the outcome of such games are socially inefficient. In this chapter, we describe a game that may be played by agents in three countries. The analysis has three objectives. The first is to investigate the nature of optimal noncooperative strategies played between more than two players. The second is to explore the impact of cooperative actions and outcomes between some of the players within the three-country world. The third is to introduce a new solution procedure for numerical optimization of the discrete dynamic games using *Genetic Algorithm*

The three-country, two-commodity model is developed to illustrate the dynamics among the North and the South. This study presents a simple model of international trade and growth between the industrial region and the non-industrial primary exporting region. There are two Southern economies that perceive themselves as being in competition with each other for profitable international trade. The model is described as a dynamic game between three countries in which the North determines the rate of investment each period, whereas countries in the South determine their terms of trade. The goal is to compare the noncooperative solution in which each country optimizes while taking as given the strategies abroad, with the cooperative (coalition) equilibria in the South in which binding commitments can be made between the Southern countries.

In two-person games, players had to share the control of their own fate with a partner but they had control over their partner's fate, which they could use as a threat. In n-person (three-person) games, even this threat is generally denied by the players, they must form coalitions with others and consider what inducements they must offer and accept. Hence, we consider a world economy in which two resource producing nations (the South) and one resourceless nation (the North) are involved. For the economic situation containing only one country in the North, we regard the North as player 1 and resource extracter countries as player 2 and player 3. Thus, the situation enables us to model both a noncooperative and a cooperative three-person game where two players in the South make a coalition.

Dynamic games based on dynamic models almost inevitably lead to solutions which are analytically intractable (Levine and Brociner 1994). It is true that with considerable ingenuity, simplifying assumptions can be made which enable tractable solutions to emerge. But solutions for the dynamic game equilibria concepts set out in this chapter require numerical solutions given particular sets of parameter values. The optimal control problems are quite difficult to deal with numerically. The task of designing and implementing algorithms for the solution of optimal control problems is the difficult part. However, genetic algorithms (GA) require little knowledge of the problem itself, and therefore, computations based on these algorithms are very attractive to dynamic optimization problems, particularly the discrete dynamic game used in this study.

The plan of this chapter is as follows. Section 2.2 sets out the two-bloc, two good model. Section 2.3 describes the solution procedure and methodology for numerical analysis in the three person game framework. The optimum solutions for various cases in noncooperative and cooperative strategies are analyzed in Section 2.4. Section 2.5 provides conclusions and suggestions for future research.

2.2 The Model

Three country model to be discussed is as follows. The analysis is conducted within a similar dynamic North-South model of Galor (1986) where this time there are two Southern countries and the model is discrete. The Southern countries produce an essential raw material using a single factor (labor) and sell the raw material to the North. The production functions for raw material R in the two countries are

$$R_{1t} = b_1 L_{1t} \qquad \text{and} \qquad R_{2t} = b_2 L_{2t}$$

where L_{it} is the amount of labor used in the production of raw material R_{it} in the country i of the South. We adapt the assumption that small countries, like the Southern countries, with small markets would specialize in constant returns products (Dreze, 1960). On the other hand, the North produces a single composite commodity which can be used either for consumption or for investment. The production function for good Y_t is governed by fixed proportions production function

$$Y_t = min[aK_t, nN_t, rR_t]$$

where K_t, N_t and R_t are the amount of capital, labor and raw material used respectively in the output production at time t. This production function is used merely for simplicity of exposition in the three-country world. There is nothing intrinsic about it. One could utilize Cobb-Douglas or CES production functions equally well and obtain similar results, though at the cost of considerably more time and algebra.

The North's labor force is fixed over time at given level $\bar{N}$; however, unlimited guest workers are available in the three-country world at a given real wage $\bar{w}$, causing the supply of labor faced by the North to be perfectly elastic at this wage (Lewis, 1954). The raw material cannot be produced at the North, but bought from any of the countries in the South that offers the minimum fixed price per unit, p_{it} in terms of consumption good ($i = 1, 2$). Assuming no foreign investment in the North, the production function depends on the proportion of capital available

$$Y_t = aK_t$$

where K_t is determined by the given initial capital stock of the North; K_0.

By the specification of the production function in the North, we can derive the labor and raw material requirements of the North as follows

$$N_t = aK_t/n \qquad \text{and} \qquad R_t = aK_t/r$$

Full employment of the North's labor force is assumed at the initial time i.e. $K_0 > n\bar{N}/a$.

The North invests a proportion s_t of the return to its capital at time t while the rest is devoted to consumption. Its entire wage income is consumed, assuming $\bar{w}$ is the subsistence level in the North. Hence, the problem faced by the North is to choose rate of investment s_t to maximize the discounted value of its consumption stream over a given time horizon T[7]:

<u>**North:**</u>

$$max \sum_{t=0}^{T} \rho^t u(C_t^N) \qquad 0 < \rho < 1 \tag{1}$$

subject to

$$K_{t+1} = [Y_t - \bar{w}(N_t - \bar{N}) - min(p_{1t}, p_{2t})R_t]s_t + (1-\delta)K_t$$

$$K_0 \text{ given}$$

$$0 \leq s_t \leq 1$$

where ρ is the subjective discount factor. The North's consumption at time t is

$$C_t^N = [Y_t - \bar{w}(N_t - \bar{N}) - min(p_{1t}, p_{2t})R_t](1 - s_t)$$

and the capital stock evolves according to

$$K_{t+1} = [Y_t - \bar{w}(N_t - \bar{N}) - min(p_{1t}, p_{2t})R_t]s_t + (1-\delta)K_t$$

where $0 < \delta < 1$ is the depreciation rate and the portion s_t of Northern income so that earned will be saved and invested. Also, the selection criterion for the price of

[7]The terminal condition in this study is rather arbitrary. The game is played for certain periods chosen initially. We assume that there will be no game after T periods, however, in general terminal conditions are chosen where the stable equilibria of the economy are satisfied. Since the terminal conditions are not important part of the aim of this study, we disregard the analysis of the terminal conditions.

raw material p_{it} offered by the country i in the South, being minimum, is added in both of the consumption and investment equations of the North.

Finally, in order to derive the optimal rate of investment stream and estimate the model, the constant risk aversion (CARA) utility function is adapted:

$$u(C_t^N) = \frac{(C_t^N)^{1-\sigma}}{1-\sigma}$$

where the degree of risk aversion $\sigma > 0$ and $\sigma \neq 1$ [8] .

On the other hand, the problem of the country i ($i = 1, 2$) in the South is to choose the terms of trade p_{it} that maximizes the discounted value of its consumption stream.

Southern production is used for consumption only

$$C_t^{S_i} = p_{it} R_{it} \quad i = 1, 2$$

From the production function of the North, the price of raw material has no direct effect on the output produced in the North but through the accumulation of capital, the price affects the current investment and consumption in the North.

The South is characterized by the existence of surplus labor. The supply of labor is perfectly elastic at a fixed real wage $\bar{w}$ in terms of the consumption good. The South trades the raw material for the consumption good produced in the North. The terms of trade determined by the South at any point in time are assumed to be greater than the price which enables the South to consume at least at the subsistence level and smaller than the price which enables the North to consume strictly more than its subsistence level.

In this world economy, the demand for the raw material by the North is determined according to the production technology in the North. The primary product is demanded for investment and consumption purposes in the North and the division of any amount demanded from each of the ith country in the South depends on the price offered by the South ($i = 1, 2$).

It is assumed that in addition to the production cost in the South, there is also the cost of carrying, holding or destroying for the amount unsold (cost of overage). Thus,

[8]The constant elasticity of substitution utility has the economic property that elasticity of substitution between consumptions in any two points in time is constant and equals to $1/\sigma$. This instantenous utility function is frequently used in intertemporal optimizing model and has no relevant effect on the conclusion of the study.

in the three-country world where Southern countries are same type, the amount of primary product demand from ith country in the South is *randomly* determined by the North when both of the countries offer the same minimum price ($p_{1t} = p_{2t}$). Hence, under risk, the terms of trade decisions of the South will cover the cost of overage denoted by d.

Thus, the problems of the Southern countries are as follows:

South 1:

$$max \sum_{t=0}^{T} \rho^t u(C_t^{S_1}) \tag{2}$$

subject to

$$\begin{aligned}
K_{t+1} &= [Y_t - \bar{w}(N_t - \bar{N}) - min(p_{1t}, p_2)R_t]s_t + (1-\delta)K_t \\
C_t^{S_1} &= p_{1t}R_{1t} - J_{1t} \\
R_{1t} &= \begin{cases} R_t & \text{if } p_{1t} = min(p_{1t}, p_{2t}), p_{1t} \neq p_{2t} \\ \alpha_t R_t & \text{if } p_{1t} = min(p_{1t}, p_{2t}), p_{1t} = p_{2t} \\ 0 & \text{if } p_{1t} \neq min(p_{1t}, p_{2t}) \end{cases} \\
J_{1t} &= \begin{cases} 0 & \text{if } R_{1t} = R_t \\ d(R_t - R_{1t}) & \text{if } R_{1t} \neq R_t \end{cases} \\
K_0 & \quad \text{given}
\end{aligned}$$

South 2:

$$max \sum_{t=0}^{T} \rho^t u(C_t^{S_2}) \tag{3}$$

subject to

$$\begin{aligned}
K_{t+1} &= [Y_t - \bar{w}(N_t - \bar{N}) - min(p_{1t}, p_2)R_t]s_t + (1-\delta)K_t \\
C_t^{S_2} &= p_{2t}R_{2t} - J_{2t} \\
R_{2t} &= \begin{cases} R_t & \text{if } p_{2t} = min(p_{1t}, p_{2t}), p_{1t} \neq p_{2t} \\ (1-\alpha_t) R_t & \text{if } p_{2t} = min(p_{1t}, p_{2t}), p_{1t} = p_{2t} \\ 0 & \text{if } p_{2t} \neq min(p_{1t}, p_{2t}) \end{cases}
\end{aligned}$$

$$J_{2t} = \begin{cases} 0 & \text{if } R_{2t} = R_t \\ d(R_t - R_{2t}) & \text{if } R_{2t} \neq R_t \end{cases}$$

$$K_0 \quad \text{given}$$

where α_t is a random variable which determines the amount of raw material bought by the North from the country i of the South. We can consider α_t as demand shock. It is assumed that there is no transaction cost in the world, so, the North is indifferent to buying raw material from any of the South countries if both offer the same minimum price; hence, according to the number generated between and including zero and one, the amount sold by each South country will be determined. Even when, any one of the Southern countries offers the minimum price, there is always the possibility of selling no raw material to the North. The world is uncertain for the South.

Since it is assumed that the consumption good and the raw material are not storable goods, in the cases of similar price offers, each Southern country will suffer positive amount of cost of getting rid of the excess production if $\alpha_t < 1$. Then the South trades the raw material for the consumption good produced in the North accordingly

$$C_t^{S_1} = \begin{cases} -dR_t & \text{if } \alpha_t = 0 \\ (p_{1t}\alpha_t - d(1-\alpha_t))R_t & \text{if } 0 < \alpha_t < 1 \\ p_{1t}R_t & \text{if } \alpha_t = 1 \end{cases}$$

$$C_t^{S_2} = \begin{cases} p_{2t}R_t & \text{if } \alpha_t = 0 \\ (p_{2t}(1-\alpha_t) - d\alpha_t)R_t & \text{if } 0 < \alpha_t < 1 \\ p_{1t}R_t & \text{if } \alpha_t = 1 \end{cases}$$

Under certainty, the terms of trade determined by the South, p_{it} at any point in time, would be greater than the subsistence level $\bar{w}/b_i$ ($i = 1,2$); however, in the three-country world with constant demand of raw material by the North, the South should consider the cost of unsold units of their production. Because of the existence of the risk of not selling all of the raw material produced in the country i, each Southern country takes destroying cost (cost of overage) of excess production into the derivation of its minimum price offer.

Since the amount of raw material demanded from the ith country is determined randomly ($0 \leq \alpha_t \leq 1$) by the North, Southern countries will calculate expected value of this random variable and determine their minimum offers

$$p_{1t} \geq \frac{d(1-\alpha^e)}{\alpha^e} + \frac{\bar{w}}{b_1\alpha^e}$$

$$p_{2t} \geq \frac{d\alpha^e}{(1-\alpha^e)} + \frac{\bar{w}}{b_2\alpha^e}$$

where α^e is the expected value of the random variable α. The expected value of the random variable α_t with uniform distribution over [0 1] is 0.5, thus, the minimum price offered by the country i in the South is

$$p_{it} \geq \frac{2\bar{w}}{b_i} + d \qquad i = 1, 2$$

In order to obtain concrete results, we adapt the assumption that Southern countries also have identical and homothetic tastes as in the North:

$$u(C_t^{S_i}) = \frac{(C_t^{S_i})^{1-\sigma}}{1-\sigma} \qquad i = 1, 2$$

where $\sigma > 0$ and $\sigma \neq 1$.

2.3 Dynamic Equilibria

The equilibria will be determined by the simultaneous solution of the three countries' problem. The solution of the North's maximization problem determines the optimal time path of s_t given the South's prices p_{1t} and p_{2t}, whereas the solution of the country i of the South's maximization problem determines the optimal path of p_{it} given the paths of s_t and p_{jt}, $j \neq i$. The dynamic equilibria is given by the triplet solution $[s_t^*, p_{1t}^*, p_{2t}^*]$.

2.3.1 Solution Procedure

In this three-country game, players move or act simultaneously within each stage or period of the game and know the actions that were chosen in all past strategies. This three-country game is a dynamic game that concerns itself with determining how policymakers, or agents, within each economy, acting over time, choose optimally among some given set of actions. A crucial point is that even within a deterministic context, the choice of plan and the nature of the underlying information pattern is critical to the equilibrium outcome of the game. This is in contrast to a single country dynamic optimization context where, under the assumptions of uncertainty, such a choice is unimportant.

In a dynamic game, a precise delineation of the information pattern, such as which agent knows what, how the information pattern available to each agent evolves over time, how much of this is common information shared by all players and what part of it constitutes private information for each player, is of paramount importance. An information set is *open-loop* if only the priori raw data set is available at all points in time and in this case the policy variables that depend only upon time are called *open-loop* policies.

The players are assumed to never observe any history other than their moves and time; at the beginning of the game they must choose time paths of actions that depend only on calendar time, hence the dynamic equlibrium in open-loop strategies found in this experiment is an *open-loop* equilibrium.

If the players can condition their strategies on other variables in addition to calendar time, they may prefer not to use open-loop strategies in order to react to mixed strategies and the possible deviations by their rivals from the equilibrium strategies. Such strategies are called *closed-loop* strategies which is valid when the player's can observe and respond to their opponent's action at the end of each period. However, in this study, open-loop strategies are preferred; firstly, they are analytically tractable in the three country game because the closed-loop strategy space is so much larger, secondly, it is assumed that the players in the South are small in the sense that unexpected deviations by the opponent would have little influence on the player's optimal play.

The solutions of dynamic games with multiperiod even for twoplayer games, are hard to handle analytically, a three-player game would immediately increase the strategy space to search. We have to first set the rule of the game and adapt the shared memory algorithm developed by Özyıldırım (1997) for the numerical solution of dynamic games.

Finally, in this three country world, there are two countries which are allowed to be identical or different in production technology which enables us to analyze varies experiments over dynamic North-South game.

2.3.2 Shared Memory Algorithm For 3-Players

Many techniques are used today for optimizing control systems. Most of these techniques can be broadly classified under two main classes: calculus-based techniques, and, enumerative schemes. The calculus-based techniques, although extensively used, have the following drawbacks: they are local in slope, i.e., the extrema they seek are the ones closer to the current point, and they depend on the existence of either derivatives or some function evaluation scheme. Thus, calculus-based methods lack robustness over the broad spectrum of optimization functions. Many enumerative schemes have been proposed to overcome the shortcomings of calculus-based methods. These schemes lack efficiency because many practical search spaces are too large to search. Another type of algorithm that has gained popularity is the random search technique. This algorithm lacks efficiency and in the long run, can be expected to do no better than enumerative schemes.

One technique that is global and robust over a broad spectrum of problems is the *genetic algorithm (GA)*. Genetic algorithms are search procedures based on the mechanics of natural genetics. Genetic algorithms were originally developed by Holland in 1975. The approach is very different from classical search methods, where movement is from one point in the search space to another point based on some transition rule. Another important difference between *GAs* and the classical approaches is in the selection of the transition rule. In classical methods of optimization, the transition rule is deterministic. In contrast, *GAs* use probabilistic operators to guide their search (Goldberg 1989).

A simple genetic algorithm is composed of three operators:

1. reproduction
2. crossover
3. mutation

Reproduction is a process where old strings is carried through into a new population depending on the performance index (i.e. fitness or utility) values. Due to this move, strings with better fitness values get large number of copies in the next generation. Selecting good strings for the reproduction operation can be implemented in many different ways. A simple crossover follows reproduction in three steps. First, the newly reproduced strings are paired together at random. Secondly, an integer position n along every pair of strings is selected uniformly at random. Finally, based on a probability of crossover, the paired strings undergo crossing over at the integer position n along the strings. This results in new pairs of strings that are created by swapping all of the characters between 1 and n inclusively. Although the crossover operator is a randomized event, when combined with reproduction it becomes an effective means of exchanging information and combining portions of good quality solutions. Reproduction and crossover give *GAs* most of their search power. The third operator, mutation, is simply an occasional random alteration of a string position (based on the probability of mutation). In a binary code, this involves changing a **1** to a **0** and vice versa. The mutation operator helps in avoiding the possibility of mistaking a local minimum for a global minimum. When mutation is used sparingly with reproduction and crossover, it improves the global nature of the genetic algorithm search.

At the first glance, it seems strange, or at least interesting, that such a simple mechanism should motivate anything useful; however, genetic algorithm is strictly inductive when compared with other search methods, which are ploddingly deductive. However, induction for its own sake is not a compelling argument to use for any method, unless it can be shown how and when the method is likely to converge. Holland's schema theorem places the theory of genetic algorithms on rigorous footing by calculating a bound on the growth of useful similarities or building blocks. The fundamental principle of *GAs* is to make good use of these similarity templates (Krishnakumar and Goldberg 1992).

The genetic algorithm we described is mostly applied to optimal control theory which involves the calculation of time paths for one or more variables in order to minimize or maximize some functional. However, for the problems where there are more than one player or controller, different algorithms need to be developed and/or used that consider the dynamics arising from the interactions among different decision makers. Since the interests do not coincide, game-theoretic considerations become important. A solution concept from game theory which has been used a lot in economic applications is the noncooperative solution, or Nash equilibrium (Kydland 1975). The open-loop noncooperative solution is a sequence of decisions for each time period, and these decisions all depend on the *initial* state and in the presence of *uncertainty* on observed disturbances.

To solve n-persons (3-persons) optimization problem in dynamic games, a Nash equilibrium is solved jointly for $\{s_t, p_{1t}, p_{2t}\}$, $t = 0, 1, ..., T$ (Sargent 1987). Thus, we developed an algorithm in which GA is used to solve dynamic games for n-players.

Briefly, we have parallelly implemented n-separate genetic algorithms where each GA is used to solve the discrete optimal problem of the player in the game. Each player has its own evaluation function derived by substituting the constraints of that player to its own objective function. Naturally, in each evaluation function, the choice variables (strategies) of the other players causing conflicts are also included. In the open-loop Nash equilibria, each player takes the entire future path of the others' $(n-1$ players) controls as given and choose its vector of optimal strategies over time. The North, in choosing $\{s_t\}_{t=0}$ to maximize $u(C_t^N)$ subject to the dynamics of the system described in the model, takes K_0 and $p_{it}, i = 1, 2$ for $t \geq 0$ as given, while player i in the South in choosing $\{p_{it}\}_{t=0}$ to maximize $u(C_t^{S_i})$, takes the whole time paths of s_t and $p_{jt,j\neq i}$ as given. So, we used the solution procedure of both open-loop solutions and genetic algorithm in order to develope the algorithm described below[9].

A genetic algorithm to solve a problem must have 5 components:

1. bit string (0's and 1's) representation of solutions of the problem,
2. a way to create an initial population of solutions,
3. an evaluation function that rate the solutions in terms of fitness,
4. genetic operators that generate new solutions and
5. values of the parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators, etc.).

Initialization routines vary. For research purposes, a good deal can be learned by initializing a population *randomly*. Moving from a randomly created population to a well-adapted population is a good test of the algorithm, since the critical features of the final solution will have been produced by the search and recombination mechanisms of the algorithm, rather than the initialization procedures. Hence, each player's GA begins with the randomly generated policies of all parties.

In our experiment, the North calculates its initial evaluation function using ran-

[9] For details, see Özyıldırım (1997).

domly generated sequence of $\{p_{1t}, p_{2t}\}_{t=0}^{T}$

$$U_N^1 = \sum_{t=0}^{T} \rho^t u(C_t^N)$$

where U_N^1 is the North's evaluation function at the $1st$ iteration which contains the entire pricing policies of the Southern countries. The same initial procedure is applied to the Southern countries, where $U_{S_1}^1$ and $U_{S_2}^1$ are calculated using randomly generated $\{s_t, p_{2t}\}_{t=0}^{T}$ and $\{s_t, p_{1t}\}_{t=0}^{T}$ respectively. Thus, in our three-country world, the initial best (b) results $\{s_t^b, p_{1t}^b, p_{2t}^b\}_{t=0}^{T}$ are obtained from randomly generated policies. The informations about the best strategies of the other players at each iteration are kept in the *shared memory* where each player sends its best results and in exchange learns the best results of the other players. Hence, each player uses the best strategies of the other players $(n-1)$ in each generation (or iteration) while solving its own problem. Here, still, the best does not mean the optimum for a particular functional. Thus, each side solves its problem and writes the best solutions to the shared memory and waits other sides to do the same thing. The waiting procedure is very important since each iteration or generation has to be evaluated synchronically. After copying the results of the other players, the problem of each player is to find the optimum time path of the variable(s) under investigation.

Using the close relation between the derivation of open-loop Nash equilibria and the problem of solving (jointly) n optimal control problems (Başar and Oldser, 1982), we used GA to optimize the control system of each player. GA is a probabilistic algorithm which maintains a population of individuals, $P(t) = \{x_1, ..., x_n\}$ where $x_i = \{x_{i1}, ..., x_{iT}\}$. Each individual, x_i represents a potential solution vector to the problem at hand. Each solution vector is evaluated to give some measure of its fitness (utility value). Then, a new population is formed by selecting the more fit solutions (individuals). Some members of the new population undergo transformations by means of genetic operators (crossover, mutation) to form the new solution set. This procedure is repeated until the global optimum is converged for the problem under investigation.

It is crucial to understand that the evaluation functions are derived by substituting constraints into the objective functions of the particular problem, hence GA does not use first-order conditions to derive the optimal strategies for each player.

Even, if the whole solution algorithm seems strange, since GA is a highly parallel mathematical algorithm, it is very successful in solving n-person discrete dynamic games. Most computer programs consist of a control sequence (the instructions) and a collection of data elements. Large programs have tens of thousands, or even millions of data elements. There are opportunities for parallelism in both the control sequence and the collection of data elements. In the control sequence, it is possible to identify threads of control that could operate independently, and thus on different processors. This is the method used for programming most multiprocessor computers. The primary problems with this approach are the difficulty of identifying and synchronizing these independent threads of control (Robertson 1987).

We used a similar idea for this particular Nash equilibria, where we have to solve the problem jointly as systems of equations. Hence, the whole system is divided into n (3, in this study) parallel systems and solved using the proved schema theory behind GA. Since both the theory and the findings satisfying the optimality conditions such as first order and second order conditions, we can immediately say GA works in the open-loop equilibria of the discrete dynamic games.

2.4 Optimal Strategies and Equilibrium Time Paths

The North's maximization problem (1) will be solved for the rate of investment and the country i's problem ($i = 1, 2$) (2) and (3) respectively in the South are solved for the term of trade using numerical analysis. For the numerical results, we have to specify some of the benchmark parameter values used:

$$\begin{array}{lllll} a = 5 & n = 4 & r = 4 & \rho = 0.95 & \bar{N} = 20 \\ \bar{w} = 1 & d = 0 & \sigma = 0.5 & \delta = 0.05 & K_0 = 100 \end{array}$$

In the benchmark parameter, unit cost of destroying excess production (cost of overage) in the Southern economies is set to $d = 0$ for simplicity. Thus, the subsistence level, or minimum level of price, will be $p_{it} \geq 2/b_i$, and according to the value of productivity parameter in the ith country of the South, the minimum offers will be determined.

The parameters necessary for the genetic algorithm, the crossover and the mutation rates are 0.60 and 0.03 respectively. These rates are default rates in most of the genetic algorithms. All of the experiments are done for 500000 trials and for

$T = 12$[10]. Even the choice of planning horizon is arbitrary, the trade relations and commitments between three countries would be less informative for longer periods.

In our analysis, we consider three representative cases:

2.4.1 Case 1: Identical Technologies in the South ($b_1 = b_2$)

Certain features of Southern economies are of particular relevance to the applicability and implications of the various theories (Stewart 1984). These features all stem from lower levels of development. One of these features is that Southern economies are characterized by very substantial elements of inefficiency in terms of underemployment of some resources and poor productivity of resources in use. As far as the South is concerned, the criteria for assessing international trade are generally broader than in much gains from trade literature, but the employment of resources is one obvious gain in any case.

We start our analysis assuming that two of the Southern economies are characterized with the same production technology and productivity for producing raw material:

$$b_1 = b_2 = 1$$

Thus, the minimum price for any of the Southern countries will be $p_i \geq 2, i = 1, 2$ With the benchmark parameters and the productivity parameter of the Southern technology, the optimal strategies for noncooperative three country game are summarized in Table 2.1.

Both of the Southern economies offer minimum price according to the expected selection criterion for the North. Within symmetric technology in the South, no transaction and no transportation world, the optimal strategies of the noncooperatively acting Southern economies will offer minimum price even at the end of the world ($t = T$). Since the amount sold will be determined exogenously and randomly, the welfare u^* of each Southern country would be different even though both offered the same price. Thus, in this experiment, the second Southern country trades and gains more, compared to the first country. However, if this experiment is repeated again, in this symmetric countries in the South, none of the Southern nations has guarantee of gaining trade with the North over other Southern nation.

[10]The period can be taken as months or years

Table 2.1: Optimal Strategies

t	s_t	p_{1t}	p_{2t}
0	1.000	2.816	2.000
1	1.000	2.000	2.247
2	1.000	2.000	2.345
3	1.000	2.000	2.000
4	1.000	2.000	2.000
5	1.000	2.651	2.000
6	0.990	2.000	2.000
7	0.996	2.000	2.000
8	0.988	2.000	2.000
9	0.961	2.000	2.000
10	0.874	2.000	2.000
11	0.576	2.000	2.000
12	0.000	2.000	2.000
u^*	25868	25481	49325

Over the 12 periods, the North saves all in the early periods and grows then consumes fifteen percent of their output after ten periods and consumes all at the end of planning horizon. In this numeric study, we didn't specify any end value for the state variables but specified only the end of period, and the optimal strategy at the end of the period is determined within the model.

The stationary values for this game are reached within less than ten periods:

$$p_{1t}^* = p_{2t}^* = 2, \quad s_t^* = 1$$

2.4.2 Case 2: Different Technologies in the South ($b_1 \neq b_2$)

As producers become increasingly dependent on the South markets, developments in the South will have a commensurately larger effect on output in rich countries. Thus improved terms of trade or rising productivity in the South will reduce the cost of the North's imports, giving consumers a boost in real income. In this section, we allow one of the Southern economies to be more productive:

$$b_1 = 2 > b_2 = 1$$

Thus, the minimum prices will be $p_{1t} \geq 1$ and $p_{2t} \geq 2$ for all t. Then, the optimal strategies for asymmetric South case are summarized in Table 2.2. The impact of improved terms of trade in one of the Southern economies, boosts the annual income

Table 2.2: Optimal Strategies

t	s_t	p_{1t}	p_{2t}
0	1.000	1.016	
1	1.000	1.008	
2	1.000	1.055	
3	1.000	1.008	
4	1.000	1.061	
5	1.000	1.047	
6	0.996	1.016	
7	0.988	1.016	
8	0.949	1.016	
9	0.937	1.008	
10	0.874	1.031	
11	0.498	1.596	
12	0.000	1.988	
u^*	140910	231290	0

of the trading countries. Hence, even though the saving behavior of the North is similar to the previous case, in terms of aggregate income, the trade is strongly beneficial for the North and the productive South. The welfare of the both nations are increased, while the less productive Southern country is not able to sell and earns zero utility. So, the best policy for that country will be either to increase its productivity or to specialize in the production of another competitive good.

The pricing of extractive resources has traditionally been the source of North-South conflict, with the exporting South trying for better prices and the North resisting the South. Changes in the price of some resources such as oil, however, have forgoed a strong interdependence between the North and the South, both in real and in financial markets. There is now common interest between exporters and importers in keeping prices within a reasonable range neither too high nor too low (Chichilnisky and Heal 1986). In this experiment, we observed that the pricing policy of the productive Southern country is to offer low price for the periods when the North saves and grows rapidly and then to increase the price slightly less than the minimum price that can be offered by the other Southern country. The result can be taken as supportive to the neoclassical view since we clearly observed welfare gain from the trade of the two nations in the three-country world.

2.4.3 Case 3: Cooperation in the South

Suppose now that two players in the South agree to cooperate for minimizing the risk and maximizing their intertemporal utility. In three-players game, a subset of the player set $(2 \subset 3)$ is called a coalition. In the world described in this study, the same type Southern countries under risk act cooperatively before determining their pricing policies. By coalition, they will get rid of excess usage of their resources in the production of one good. Also, by the elimination of possible cost of overage.

$$p_t \geq \bar{w}/b, \qquad b_1 = b_2 = b$$

and the problem becomes

<u>**North:**</u>

$$max \sum_{t=0}^{T} \rho^t u(C_t^N) \tag{4}$$

subject to

$$\begin{aligned} K_{t+1} &= [Y_t - \bar{w}(N_t - \bar{N}) - p_t R_t] s_t + (1-\delta) K_t \\ C_t^N &= [Y_t - \bar{w}(N_t - \bar{N}) - p_t R_t](1 - s_t) \\ K_0 & \quad \text{given and } \; 0 \leq s_t \leq 1 \end{aligned}$$

<u>**South :**</u>

$$max \sum_{t=0}^{T} \rho^t u(C_t^S) \tag{5}$$

subject to

$$\begin{aligned} K_{t+1} &= [Y_t - \bar{w}(N_t - \bar{N}) - p_t R_t] s_t + (1-\delta) K_t \\ C_t^S &= p_t R_t \\ K_0 & \quad \text{given} \end{aligned}$$

In our experiment, using the technology where $b = 1$, the minimum prices are $p_t \geq 1$ for all t. Hence, with the reduction in the resource prices the trade between North and South becomes beneficial for all parties (Table 2.3).

Table 2.3: Optimal Strategies

t	s_t	p_t
0	1.000	1.023
1	1.000	1.047
2	1.000	1.016
3	1.000	1.031
4	1.000	1.008
5	1.000	1.016
6	1.000	1.008
7	1.000	1.000
8	0.918	1.031
9	0.839	1.016
10	0.435	1.196
11	0.071	2.451
12	0.000	2.976
u^*	70628	183400

Assuming that both of the Southern countries share equally the welfare gain from the trade, it is found that the welfare of each country is $u^* = 91700, i = 1, 2$ which is higher than the one in noncooperative symmetric case. Hence, as compared to the Case 1 where technologies are same in the Southern countries, all of the three players gain from cooperative strategies. Thus, the elimination of random shock in demand lowered the prices but increased the trade and welfare of the parties in the game.

2.5 Conclusion

The sharp pricing shocks of the 1970s raised the attention to the possible conflicts of interests between resource exporters and importers. As well as having common interests in certain types of price movements, exporters and importers shared influence over the price movements. They should design their economic policies so as to use their joint influences to pursue common interest.

The dynamic aspects of economic interdependence have invited the application of dynamic game theory. To characterize the relations between the players in the North-South interactions, we examined the symmetric equilibria of Nash differential games, open-loop, using *GA*. The noncooperative equilibria have been compared with the cooperative equilibrium in non-coordinated resource pricing and investment strategies.

In the model presented here, international coalition leads to dynamically efficient trade relation within the three-country world. The welfare impacts of the cooperation are obvious.

Although the solution algorithm studied in this chapter is designed for non-convex dynamic games, the model itself is simple; extensions in the direction of generating a rich model structure will be desirable. And, finally, future research might profitably examine the terminal condition where all the economies reach stability. Instead of solving the game where the world will end at the end of T periods, we have to solve the game that the terminal periods for each country are chosen where the economies reach their stable equilibria. However, since the aim of this study is to introduce a new technique for solving dynamic games where the terminal conditions are rather certain by either targets of decision variables or periods, we disregard the choice of terminal conditions in this study. Nevertheless if the equilibrium states and when the economies reach to these states are known, the adjustment is straightforward.

A Genetic Game of Trade, Growth and Externalities

3.1 Introduction

It is by now a standard practice in economic modeling to apply results from optimal control theory to economic dynamics. When dynamics involve optimization of controls over time by a single controller, the application is immediate. If , however, there are multiple controllers then dynamic game theory provides a framework for modeling and sets the appropriate tools for *optimal* decisions under various environment. (Başar and Olsder 1982) The solution concepts involve optimization of functionals, either over time or stage wise at each point in time. The former requires results from optimal control theory to be used.

With the recent advances in computer technology, there has also been a growing awareness and interest in the computational aspects of complex dynamic structures which can not be easily handled with the traditional analytical methods (Judd 1992). Most widely used numerical techniques to optimize control problems today are either calculus-based or enumerative. Both have short-comings: calculus-based techniques require derivatives and they are only local, i.e., the extrema they seek are the ones closer to the current point. Enumerative schemes that have been developed to overcome these drawbacks lack efficiency because many practical search spaces are too large to search one at a time. Random search techniques that narrow the search space have been shown to fare no better than the enumerative schemes in the long run (Krishnakumar and Goldberg 1992).

One search technique that is both global and robust over a broad spectrum of problems is the genetic algorithm. Genetic algorithm search combines a Darwinian survival-of-the-fittest strategy to eliminate unfit characteristics and uses random information exchange, with exploitation of the knowledge contained in the old solutions, to affect the search mechanism with surprising power and speed.

Many search techniques require much auxiliary information in order to work properly. For example, gradient techniques, as mentioned above, need derivatives (calculated both analytically and numerically) in order to be able to climb the current peak. Other local search procedures like the greedy techniques of combinatorial optimization require access to most if not all tabular parameters. By contrast, Genetic algorithms have no need for all this auxiliary information: *GAs* are *blind.* To perform an effective search for better and better structures, they only require payoff functions (measures of fitness) associated with decision variables (Krishnakumar and Goldberg 1992).

Michalewicz (1992) devised a GA to optimize control problems with a single controller. Özyıldırım (1996) extented GA to solve open-loop difference games of finite horizon. In this chapter a general purpose GA is developed to solve open-loop differential games of infinite duration. The lack of attention paid to the development of computational techniques to solve such problems was first addressed by Pau (1975a, 1975b).

Numerical solutions using genetic algorithm, necessarily require reformulating the problem into a discrete finite horizon approximation. In general, this transformation involves two types of decisions: the length of the finite planning horizon (the transient phase) and the treatment of post terminal behavior (the stationary phase). Errors on the optimal trajectory that will result from discrete-time approximation may be reduced by increasing the length of the decision horizon with a resulting *boost* in computational costs. Mercenier and Michel (1994), propose time aggregation to transform continous time infinite horizon optimal control problems to discrete-time approximation with the *same* steady-state. This property imposes consistency constraints on the joint formulation of preferences and accumulation equations. It is shown that this consistency is achieved by simple restriction on the choice of discount factor. We show that their results extend to control problems with multiple controllers. Then we exploit the inherent parallelism in GA to solve the time-aggregated finite-time difference games.

The second objective of the chapter is to lay out a model of a two-region, North/South, dynamic trade game in the same vein as Galor (1986). North specializes in the production of a single good which is consumed, invested (only in the North) and traded (at the terms of trade decided by the South) for an essential resource produced only by the South. As different from Galor, however, the game extends indefinetely into future so that there is no arbitrariness as to the ending of the game. Also knowledge, broad capital, accumulated in the North diffuses, albeit at a diminishing rate, to check the damage done to the Southern environment from resource extraction. Thus, patterns of trade and growth are further complicated by the presence of local and transboundary externalities. In a dynamic game setup with transboundary pollution, Dockner and Long (1993) found that Pareto-efficient steady-state pollution level can be sustained with non-linear Markow-Perfect strategies if discount rates are sufficiently small. Baç (1996) analyzes incentives to free-ride on transboundary abatements when there are informational asymmetries. Thus, the chapter aims to contribute to the recent literature exploring the linkages between international trade, environmental degradation and growth by bringing to the fore the dynamic gaming aspects of these issues (Markusen 1975, Clemhout and Wan 1985, Conrad and Clark 1987, Ludema and Wooton 1994, Lopez 1994).

As different from the studies which analyze the said externalities in a static framework (Milliman and Prince 1994, Diwan and Safadi 1992, Low and Shafik 1992, Alpay and Saglam 1996), the knowledge accumulation brings about additional intertemporal tradeoffs thus pointing to additional sources of inefficiencies. Today's price and investment policies via their effects on the rate of knowledge accumulation determine the feasibility of tomorrow's prices and investment. Consequently, optimality here is in reference to the whole time profile of the policies, i.e., the policies must be optimal in their time sequence as well.

In the search for optimal policies two modes of behavior are considered: non-cooperative and cooperative. In the non-cooperative Nash search, each region is represented by an artificially intelligent GA to adopt policies taking the rival's as given. Choices are evaluated in terms of their impact on the respective fitness functions, ignoring the *side effects* on the rival's fitness. Policies are then iteratively improved upon using a synchronous Darwinian search mechanism. Fittest policies are found if no improvement in "life time" fitnesses is possible.

South chooses resource prices with a view to maximize her own fitness. Resource prices also affect the rate of knowledge accumulation and hence the Northern welfare which however does not enter into South's calculus of price determination. Likewise in its search for optimal resource/knowledge mix, the Northern intertemporal calculus ignores the fact that there are detrimental side effects of resource use to the Southern environment and also that knowledge accumulation reduces these effects. As such, Northern policies will exacerbate pollution in the South, and Southern policies will retard growth in the North. It is worth noting here that though environmental pollution is local in nature, it has global ramifications calling for an international approach to appropriately internalize it.

In the cooperative search, the world fitness is represented as a weighted sum of each region's respective fitnesses. Consequently, all externalities are internalized. The resulting price and resource/knowledge paths are efficient as they are optimal relative to a global fitness.

Cooperative solutions may be difficult to achieve for various reasons e.g., lack of enforcement mechanisms, high monitoring costs, etc. We show that with a sufficiently strong knowledge difussion, even if the parties act noncooperatively, there are substantial welfare gains to be obtained by both parties over cooperation with restricted knowledge spillover. This indicates that the extent of knowledge spillover is crucial in determining the long-run fortunes of the regions in question. Hence a unilateral act by the North which lifts barriers to dissemination of knowledge related to pollution abatement would be welfare improving for both regions.

In the model we consider, environmental pollution and knowledge stock appear in the Southern fitness, but external to the North. Hence the terms of trade chosen by the South incorporate the local costs and benefits of these externalities as well as a monopoly rent due to sole ownership of resources. Consequently, inefficiencies stemming from noncooperative Northern behavior are partly reduced, though not completely eliminated, leaving South's exercise of monopoly power over resource pricing as the major source of dynamic inefficiency. Monopoly pricing results in a lower resource/knowledge mix than warranted by a global efficiency measure. It inhibits growth as well as the volume of trade between the regions.

The balance of the chapter is organized as follows: The *Genetic Algorithm* to solve open loop dynamic games is described in Section 3.2. Section 3.3 discusses the dynamic trade game between North and South. Section 3.4 contains the numerical results and Section 3.5 concludes.

3.2 Genetic Algorithm

Genetic Algorithm initiated by Holland and further extended by De Jong is best viewed in terms of optimizing a sequential decision process involving uncertainty in the form of lack of *a priori* knowledge, noisy feedback and time varying payoff function (De Jong 1993).

The genetic algorithm is a highly parallel mathematical algorithm that transforms a set of (population) individual mathematical objects (typically fixed length character strings patterned after chromosome strings), each with an associated fitness value, into a new population (i.e., the next generation) using operations patterned after Darwinian principles of reproduction and survival of the fittest after naturally occurring genetic operations.

A *GA* performs a multi-directional search by maintaining a population of individuals, $P(t) = \{x_1, ..., x_n\}$ where $x_i = \{x_{i1}, ..., x_{iT}\}$; each individual, x_i represents a potential solution vector to the problem at hand. An objective function (fitness) plays the role of an environment to discriminate between "fit" and "unfit" solutions. The population undergoes a simulated evolution: at each generation the relatively "fit" solutions *reproduce* while the relatively "unfit" solutions die.

The structure of a simple genetic algorithm is as follows: During iteration t, a genetic algorithm maintains a population of potential solutions; (chromosomes, vectors). Each solution x_i^t is evaluated to give some measure of its *fitness*. Then a new population (iteration $t+1$) is formed by selecting the more fit individuals. Some members of this new population undergo reproduction by means of *crossover* and *mutation*, to form new potential solutions.

Crossover combines the features of two parent chromosomes to form two similar offspring by swapping corresponding segments of the parents. The intuition behind the applicability of the crossover operator is the information exchange between different potential solutions. A simple crossover follows reproduction in three steps. First, the newly reproduced strings are paired together at random. Second, an integer position k along every pair of strings is selected uniformly at random. Finally, based on a probability of crossover, the paired strings undergo crossing over at the integer position k along the string. This results in the new pairs of strings that are created by swapping all of the characters between characters between characters 1 and k inclusively. As an example, consider two strings X and Y of length 5, mated at random. if the random draw chooses position 3, the resulting crossover yields two new strings, X^* and Y^*, after the crossover:

	Before Crossover						After Crossover				
X	0	0	0	1	1	X^*	**1**	**1**	**1**	1	1
Y	1	1	1	0	0	Y^*	**0**	**0**	**0**	0	0

Mutation arbitrarily alters one or more genes of a selected chromosome by a random change with a probability equal to the mutation rate *pmut*. The intuition behind the mutation operator is the introduction of some extra variability into the population. After some number of generations, the program converges- the best individual hopefully represents the optimum solutions[11].

3.2.1 Genetic Algorithm For Noncooperative Open-Loop Dynamic Games

A prime example of a dynamic game equilibrium where optimal control directly applies is the open-loop Nash equilibrium in differential games. Each player faces a standard optimal control problem, which is arrived at by fixing the other players' policies at some arbitrary functions. Hence, each such optimal control problem is parameterized in terms of some open-loop control policies which, however, do not alter the structure of the underlying optimization problems because of their open-loop character. Therefore, in principle, the necessary and/or sufficient conditions for open-loop Nash equilibria can be obtained by listing down the conditions required by each optimal control problem (via minimum principle) and then requiring that these all be satisfied simultaneously (Starr and Ho 1969). Because of the couplings that exist between these various conditions, each one corresponding to the optimal control problem faced by one player, solution of a Nash differential game equilibria, analytically or numerically, is several orders of magnitude more difficult than solving single controller optimal control problems (Başar 1986).

In general, the task of designing and implementing algorithms for the solution of optimal control problems is a difficult one. However, *GA* is recently applied to the traditional optimal control theorems by Michalewicz (1992) where everything is singular in the sense that there is one criterion to be optimized, one controller

[11]For further details, see Goldberg (1989), Michalewicz (1992), Arifovic (1994)

coordinating all control actions, and one information set available to the controller. Considering the fact that *GA* is a highly parallel mathematical algorithm, we offer a new solution procedure using *GA* to visualize situations or problems in which there are more than one performance measure and more than one intelligent controller (player) operating with or without coordination from others. Thus, we use both the optimization and the learning property of the *GA* to solve the problems of multiple criteria optimization (game theory) for the players receiving no information during the play (open-loop control) in a dynamic environment. The type of equilibrium controls studied here are conflicting Nash-Cournot equilibrium where the Nash control is the best control for each player if it is assumed that all the other players (n-1) are holding firm to their own Nash controls.

The necessary conditions for the existence of equilibria in generalized dynamic games under simultaneous play can be found in Başar and Oldser (1982), Starr and Ho (1969), and Pindyck (1977) (see Appendix). Here we discuss the numerical approximation of the open loop equilibrium solutions as developed by Özyıldırım (1996, 1997).

Since the open-loop n-person Nash equilibria can be obtained as the joint solution to n optimal control problems (Başar and Oldser 1982), then we can use n*GA*s to optimize the control system of each player.

In this environment, there are n artifically intelligent players (controller) who update their strategies through *GA* and a referee, or a fictive player, who administers the parallel implementation of the algorithm and acts as an intermediary for the exchange of best response of each player to the actions of the players in each generation. This fictive player (*shared memory*) has no decisive role but provides the best strategies in each iteration to the requested parties *synchronously.* In making his decisions, each player has certain expectations as to what the other players will do. These expectations are shaped through the information received from the shared memory in each iteration.

We parallelly implement n-separate genetic algorithms where each *GA* is used to solve the discrete optimal control problem of one player in the game. Each player has her own fitness (evaluation function) which, naturally, incorporates the (possibly conflicting) controls of the other players. In the open-loop Nash equilibria, each player takes the entire future path of the others' ($n - 1$ players) controls as given and chooses a vector of optimal strategies over time. After initialization with random choices, the information about the best strategies of the other players for later iterations are kept in the *shared memory* where each player sends its best results and in exchange learns the best results of the other players. Hence, each player uses the best strategies of the other players $(n-1)$ in each generation (or iteration) while solving her own problem. Here, still, the best does not mean the optimum for a particular functional. Thus, each side solves its problem and writes the best solutions to the shared memory and waits other sides to do the same thing. Waiting procedure is very important since each iteration or generation has to be evaluated synchronuously. After copying the results of the other players, the problem of each player is to find the optimum time path of the variable(s) under investigation. The most crucial part in this algorithm is that the problems of each player are solved *synchronously.* Since we have n different players with different objectives, the problem of each player varies. Thus, based on the complexity of the fitness function, one player might evaluate the performance of his strategies faster than the other players. However, in order to learn the action of the other players against his strategies, each player waits for the other player's action in each generation. Thus the game must be played synchronously and genetic operators must be applied sequentially to each generation.

Since *GA* is a highly parallel mathematical algorithm, it is very successful in solving n-person discrete dynamic games. Most computer programs consist of a control sequence (the instructions) and a collection of data elements. Large programs have tens of thousands, or even millions of data elements. There are opportunities for parallelism in both the control sequence and the collection of data elements. In the control sequence, it is possible to identify threads of control that could operate independently, and thus on different processors. This is the method used for programming most multiprocessor computers. The primary problem with this approach is the difficulty of identifying and synchronizing these independent threads of control (Robertson 1987).

We use a similar idea for the approximation of dynamic Nash equilibria where we have to solve the problem jointly as systems of equations. Hence, the whole system is divided into n parallel systems and solved using the proved schema theory behind *GA*.

The following figure shows the general outline of the algorithm we use for the two-region dynamic trade game:

```
procedure North GA;                 procedure SouthGA;
begin                                  begin
 initialize PN(0)                       initialize PS(0);
 randomly initialize                    randomly initialize
  shared memory;                        shared memory;
 synchronize;                           synchronize;
 evaluate PN(0);                        evaluate PS(0);
 t = 1;                                 t = 1;
 repeat                                 repeat
  select PN(t) from PN(t-1);             select PS(t) from PS(t-1);
  copy best to shared memory;            copy best to shared memory;
  synchronize;                           synchronize;
  crossover and mutate PN(t);            crossover and mutate PS(t);
  evaluate PS(t);                        evaluate PS(t);
  t=t+1;                                 t=t+1;
 until(termination condition);        until(termination condition);
end;                                  end;
```

In the above algorithm, each side waits for the presence of the previous best structure of the other side in the synchronize statement.

Both players share their results while solving their own problems. This is done by using a shared memory that contains the best solution of each side. Each side solves her own problem and writes the best solution to the shared memory and waits for the other side to do the same. Then it copies the result of the other side's solution and continues solving its problem.

In each step of this algorithm, two *GAs* are solved. Its time complexity is very high. In order to reduce this time complexity, these two *GAs* are solved for one generation. Thus they continuously share the best results while solving their problem. This approach has two advantages. First it reduces the time complexity; second, it guarantees the convergence to a global extreme.

3.2.2 Genetic Algorithm For Cooperative Games

In a cooperative game, the strategic rivalry that exists in noncooperative games is eliminated by invoking a fictive social planner who maximizes the "total fitness" which is the weighted sum of each player's respective fitness. Thus, we can solve typical control problems by standard *GA* techniques (Krishnakumar and Goldberg 1992 and Michalewicz 1992).

In general, controls may involve constraints so that, either penalty functions or substitution may be used to transform the original problem to an unconstrained optimization problem. Thus, we have evaluation (performance index) or fitness function for *GA* implementation. A *GA* performs the following steps to optimize a control problem: (1) Randomly generate an initial potential solution set, (2) Suppose we have n control variables, T periods and m potential solutions for each nT variables in each generation, evaluate the fitness value for a solution set of nTm, (3) Apply selection, crossover, and mutation operations to each set of solutions to reproduce a new population, (4) Repeat steps (1), (2) and (3) until computation is terminated according to a convergence criterion, (5) Choose the solution set nT based on the best fitness value from the current generations as the optimal solution set.

3.3 Description of the Model

3.3.1 Non-Cooperative Model of Behavior

Consider a global economy comprised of two regions namely, North and South. Employing a concave production technology $Y = F(K, R, u)$, North produces manufactured goods which are either consumed and invested in the North or exported to the South at a fixed world price of unity. K stands for ***broad capital*** measuring the current state of technical knowledge in the North (Griliches, 1979), R is the raw material imported at a monopoly price determined by the South and u captures all other uncounted determinants of output.

The state of knowledge accumulates in pace with the rate of investment,

$$\dot{K}_t = Y_t - p_t R_t - \delta K_t - C_t^N \tag{6}$$

where p_t is the relative price of resources (Southern terms of trade), $0 < \delta < 1$ is the rate of depreciation of the broad capital[12]. Henceforth, a dot over a variable denotes its time derivative while superscripts $\mathcal{N}$ and $\mathcal{S}$ stand for North and South respectively. Equation (1) shows that the pace of knowledge accumulation will be determined not only by the desired consumption profile in the North, but also in the South. Although no investment takes place in the South so that proceeds from the resource sale are totally consumed, South nonetheless indirectly contributes to knowledge accumulation via its choice of resource price.

Northern optimal consumption plan maximizes the discounted Northern lifelong welfare (fitness)

$$\max_{C_t^{\mathcal{N}}, R_t} \int_0^\infty U(C_t^{\mathcal{N}}) e^{-\rho_{\mathcal{N}} t} dt \qquad 0 \leq \rho_{\mathcal{N}} \leq 1$$

subject to Equation (1) and $K(0) = K_0$ given, $C_t^{\mathcal{N}} \geq 0$ for all t.
$U(C_t^{\mathcal{N}})$ is the instantaneous felicity satisfying the usual Inada conditions and $\rho_{\mathcal{N}}$ denotes the Northern time preference rate.

We assume endowment asymmetry in that primary resource is only produced in the South. The output of raw material is postulated by a constant returns to scale production function and is assumed, for simplicity, to be a fixed coefficient type $R_t = bL_t,\ b > 0$ where L_t is the labor employed at time t.

Resource extraction causes pollution, $\mathcal{P}$ at the rate

$$\mathcal{P} = \frac{1}{\gamma} \frac{R_t^\gamma}{K_t^\phi} \tag{7}$$

where $\gamma > 1$, $0 < \phi < 1$. γ measures the exponential order of environmental damage due to extraction and ϕ is a knowledge diffusion (spillover) parameter, signifing the degree of applicability of knowledge to pollution reduction.

The type of pollution we are considering has a high natural decay rate so that cumulative effects are underplayed. With a special case of this sort, the magnitude of the stock becomes proportional to the size of the flow defined as in Keeler, Spence and Zeckhauser (1972) and Markusen (1975)[13].

[12]The question of depreciation of the knowledge capital at the national level is discussed extensively by Griliches (1979) and interpreted as either that only a fraction of current technology flow is to be thought as a net addition to the social stock of knowledge capital or that some fraction of the preexisting stock of this capital is replaced (depreciated) annually.

[13]It is assumed that the supply of labor in the South is perfectly elastic at a fixed real wage w in terms of the

$\mathcal{P}$ enters into the Southern utility as a flow with a negative marginal utility. Given the Northern demand for resources, South chooses the terms of trade to maximize lifetime utility

$$\max_{p_t} \int_0^\infty [U(C_t^S, \mathcal{P}_t)] e^{-\rho_S t} dt \qquad 0 \le \rho_S \le 1$$

subject to Equations (1), (2) and

$$\begin{aligned} C_t^S &= p_t R_t \\ K(0) &= K_0 \quad \text{given}, \quad C_t^S \ge 0 \quad \text{for all t.} \end{aligned}$$

ρ_S is the Southern time preference. Instantaneous utility is assumed separable in consumption, C_t^S and pollution $\mathcal{P}_t$ so that $U(C_t^S, \mathcal{P}_t) = U(C_t^S) - D(\mathcal{P}_t)$. $U(C_t^S)$ is strictly concave in C_t^S and $D(\mathcal{P}_t)$ is strictly increasing in R_t and decreasing in K_t.

3.3.2 Cooperative Behavior

The problem in designing cooperative strategies is to establish the potential gains the participants to stand to gain and how an acceptable distribution of those gains will be achieved. This distribution depends on the weights that are put on the respective fittnesses, ω. The value of ω most likely to prevail in a cooperative agreement requires some bargaining which recognizes the relative power of the participants. This is outside the scope of this study.

Pareto-efficient solution is found by maximizing

$$\max_{C_t^{\mathcal{N}}, R_t, p_t} \int_0^\infty \omega U(C_t^{\mathcal{N}}) + (1-\omega)[U(C_t^S) - D(\mathcal{P}_t)] e^{-\rho t} dt \qquad 0 \le \rho \le 1$$

subject to

$$\begin{aligned} \dot{K}_t &= Y_t - p_t R_t - \delta K_t - C_t^{\mathcal{N}} \qquad (8) \\ K(0) &= K_0 \quad \text{given} \quad C_t^{\mathcal{N}}, C_t^S \ge 0 \end{aligned}$$

$\rho = \omega \rho_{\mathcal{N}} + (1-\omega)\rho_S$ is the weighted time preference term.

---

manufacturing goods. The nature of the labor force coupled with the CRTS production function determines the labor income per unit of raw material produced at a fixed level w/b. Competitive firms in the South at any point in time would then have charged a price equal to the private marginal cost of resource extraction w/b. The assumed social planner in the South levies an export tax, not only to internalize the social cost of pollution, but also to extract monopoly profit from the North.

Cooperative search takes place on the premise that South and North can enter into a binding commitment. Precommitment is difficult in the absence of suitable institutions which can enforce global decisions. Nonetheless, cooperative solutions, though they lack credibility, are important so far as they establish an efficient benchmark against which other solutions can be compared.

3.4 Numerical Experiment

Numerical solutions using genetic algorithm, necessarily require reformulating the problem into a discrete finite horizon approximation. We generalize the result by Mercenier and Michel (1994) to transform continuous time infinite horizon control problems to discrete-time approximation for multi-player control problems:

3.4.1 Discrete-Time Approximation of the Model with Steady State Invariance

In the original paper by Mercenier and Michel (1994), a general continuous-time multi-dimensional infinite horizon inter temporal problem with state vector $x(t) \in \Re^s$ and control vector $u(t) \in \Re^m$

$$\max \int_0^\infty e^{-\rho t} g(x(t), u(t)) dt$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t), u(t)), \qquad x(0) = x_0 \text{ given}$$

in which they assume existing interior stationary solution $(\hat{x}, \hat{u}, \hat{q})$ characterized by

$$f(\hat{x}, \hat{u}) = 0, \quad \rho\hat{q} = \nabla_x H(\hat{x}, \hat{u}), \quad \text{and} \quad \nabla_u H(\hat{x}, \hat{u}) = 0 \tag{9}$$

where $H(x, u) = g(x, u) + q^T f(x, u)$[14] is the Hamiltonian, $q(t) \in \Re^s$ is the vector of undiscounted shadow prices associated with the dynamic constraints, and, ∇_x and ∇_u are the gradients with respect to x and u respectively. The objective of the paper is to study the relationship that exists between $(\hat{x}, \hat{u}, \hat{q})$ and the stationary equilibrium of the discrete version of the above problem. The discrete-time problem with same stationary solution will be the following:

$$\max \sum_{n=0}^{\infty} \theta_n \Delta_n g(x(t_n), u(t_n))$$

[14] T denotes transpose of a vector

s.t. $x(t_{n+1}) - x(t_n) = \triangle_n f(x(t_n), u(t_n)), \quad x(t_0) = x_0$ given

where $\triangle_n$ is a scalar factor that converts the continuous flow in a stock increment and would be the length of time interval ($\triangle_n = t_{n+1} - t_n$). Hence, the discount factors θ_n for which the stationary solution of the continuous-time problem is a stationary solution of the discrete-time problem has a *recurrence relation* derived from the the optimality conditions of the Lagrangian of the discretized problem:

$$\triangle_n \nabla_u \{g(x(t_n), u(t_n)) + q(t_n)^T f(x(t_n), u(t_n))\} = 0$$

$$-\theta_{n-1} q(t_{n-1}) + \theta_n q(t_n) + \theta_n \triangle_n \{\nabla_x \{g(x(t_n), u(t_n)) + q(t_n)^T f(x(t_n), u(t_n))\} = 0$$

such that from (9), the first condition is satisfied by $(\hat{x}, \hat{u}, \hat{q})$ and the second may be equivalently written as

$$\theta_{n-1}\hat{q} = \theta_n \hat{q} + \theta_n \triangle_n \rho \hat{q}$$

Thus, the necessary and sufficient restrictions on the discount factor are derived so that both finite horizon discrete-time approximation and an infinite horizon continuous-time optimization problem have the same steady state. These conditions apply regardless of the dimension of the *state space* and of the *time grid.*

In our study, we applied the same discretization and the same recurrence relation for the solutions of the differential games introduced. The general representation for two player game will be as follows:

Player 1:

$$\max \int_0^\infty e^{-\rho_1 t} g_1(x(t), u_1(t), u_2(t)) dt \tag{10}$$

s.t. $\dot{x}(t) = f(x(t), u_1(t), u_2(t)), \quad x(0) = x_0$ given

Player 2:

$$\max \int_0^\infty e^{-\rho_2 t} g_2(x(t), u_1(t), u_2(t)) dt \tag{11}$$

s.t. $\dot{x}(t) = f(x(t), u_1(t), u_2(t)), \quad x(0) = x_0$ given

For *noncooperative* solution, the stationary equilibrium $(\hat{x}, \hat{u}_1, \hat{u}_2, \hat{q}_1, \hat{q}_2)$ of the problem (5) and (6) will be characterized by

$$\begin{aligned} &f(\hat{x}, \hat{u}_1, \hat{u}_2) = 0 \quad \rho_1 \hat{q}_1 = \nabla_x H^1(\hat{x}, \hat{u}_1, \hat{u}_2) \quad \nabla_{u_1} H^1(\hat{x}, \hat{u}_1, \hat{u}_2) = 0 \\ &f(\hat{x}, \hat{u}_1, \hat{u}_2) = 0 \quad \rho_2 \hat{q}_2 = \nabla_x H^2(\hat{x}, \hat{u}_1, \hat{u}_2) \quad \nabla_{u_2} H^2(\hat{x}, \hat{u}_1, \hat{u}_2) = 0 \end{aligned} \tag{12}$$

where $H^1(x,u_1,u_2) = g_1(x,u_1,u_2) + q_1^T f(x,u_1,u_2)$ and $H^2(x,u_1,u_2) = g_2(x,u_1,u_2) + q_2^T f(x,u_1,u_2)$ are the Hamiltonian for the first and second player respectively.

The discretization of the above problem which would have the same stationary equilibrium of the continuous one, is the following:

Player 1:

$$\max \sum_{n=0}^{\infty} \theta_n^1 \triangle_n g_1(x(t_n), u_1(t_n), u_2(t_n))$$

$$\text{s.t.} \quad x(t_{n+1}) - x(t_n) = \triangle_n f(x(t_n), u_1(t_n), u_2(t_n)), \qquad x(t_0) = x_0 \ \text{ given}$$

Player 2:

$$\max \sum_{n=0}^{\infty} \theta_n^2 \triangle_n g_2(x(t_n), u_1(t_n), u_2(t_n))$$

$$\text{s.t.} \quad x(t_{n+1}) - x(t_n) = \triangle_n f(x(t_n), u_1(t_n), u_2(t_n)), \qquad x(t_0) = x_0 \ \text{ given}$$

From the Lagrangians of the discretized problems:

$$\sum_{n=0}^{\infty} \theta_n^1 \{\triangle_n g_1(x(t_n), u_1(t_n), u_2(t_n))$$
$$+ q_1(t_n)^T [\triangle_n f(x(t_n), u_1(t_n), u_2(t_n)) + x(t_n) - x(t_{n+1})]\}$$

and

$$\sum_{n=0}^{\infty} \theta_n^2 \{\triangle_n g_2(x(t_n), u_1(t_n), u_2(t_n))$$
$$+ q_2(t_n)^T [\triangle_n f(x(t_n), u_1(t_n), u_2(t_n)) + x(t_n) - x(t_{n+1})]\}$$

the optimality conditions for an interior solution are:

$$\triangle_n \nabla_{u_1} \{g_1(x(t_n), u_1(t_n), u_2(t_n)) + q_1(t_n)^T f(x(t_n), u_1(t_n), u_2(t_n))\} = 0$$

$$-\theta_{n-1}^1 q_1(t_{n-1}) + \theta_n^1 q_1(t_n)$$

$$+\theta_n^1 \triangle_n \{\nabla_x \{g_1(x(t_n), u_1(t_n), u_2(t_n))) + q_1(t_n)^T f(x(t_n), u_1(t_n), u_2(t_n))\} = 0$$

and

$$\triangle_n \nabla_{u_1} \{g_2(x(t_n), u_1(t_n), u_2(t_n)) + q_2(t_n)^T f(x(t_n), u_1(t_n), u_2(t_n))\} = 0$$

$$-\theta_{n-1}^2 q_2(t_{n-1}) + \theta_n^2 q_2(t_n)$$

$$+\theta_n^2 \Delta_n \nabla_x \{g_2(x(t_n), u_1(t_n), u_2(t_n))) + q_2(t_n)^T f(x(t_n), u_1(t_n), u_2(t_n))\} = 0$$

From (12) the first and third conditions are satisfied by $(\hat{x}, \hat{u_1}, \hat{u_2}, \hat{q}_1, \hat{q}_2)$ and the second and the fourth may be equivalently written as

$$\theta_{n-1}^1 \hat{q}_1 = \theta_n^1 \hat{q}_1 + \theta_n^1 \Delta_n \rho_1 \hat{q}_1$$

$$\theta_{n-1}^2 \hat{q}_2 = \theta_n^2 \hat{q}_2 + \theta_n^2 \Delta_n \rho_2 \hat{q}_2$$

from which the recurrence relations for θ_n^1 and θ_n^2 are

$$\theta_n^1 = \frac{\theta_{n-1}^1}{1 + \rho_1 \Delta_n} \quad n > 0 \quad \text{and} \quad \theta_n^2 = \frac{\theta_{n-1}^2}{1 + \rho_2 \Delta_n} \quad n > 0$$

for any θ_0^1, $\theta_0^2 > 0$.

In the *cooperative* analysis, the application of the recurrence relation will be the same as the one introduced by Mercenier and Michel.

For computational purposes, the discretization of problem (10) and (11) will involve the arbitrary choice of a time horizon of finite length. The finite horizon discrete-time problem is

$$\max \sum_{n=0}^{N-1} \theta_n^i \Delta_n g_i(x(t_n), u_1(t_n), u_2(t_n)) + \beta_N G^i(K(t_N)) \qquad i = 1, 2$$

$$\text{s.t.} \quad x(t_{n+1}) - x(t_n) = \Delta_n f(x(t_n), u_1(t_n), u_2(t_n)), \quad 0 \le n \le N-1, \quad x_0 \text{ given}$$

The steady-state invariance property imposes specific restrictions on the choice of functions $G^i(.)$. In terms of economic interpretation, the value of terminal state is the discounted sum of the constant flows that it generates

$$G^i(x) = \int_0^\infty e^{-\rho_i t} g_i(x, u_1(x), u_2(x)) dt = \frac{1}{\rho_i} g_i(x, u_1(x), u_2(x))$$

and $\beta_N = \theta_{N-1}$..

For our model, the discrete-time approximation of *finite horizon* two-region model with steady state invariance will be as follows for each region $i = \mathcal{N}, \mathcal{S}$:

$$\max \sum_{n=0}^{N-1} \theta_n^i \Delta_n U^i(t_n) + \theta_{N-1} G^m(K(t_N)) \tag{13}$$

subject to

$$\begin{aligned} K(t_{n+1}) - K(t_n) &= \Delta_n[Y(t_n) - p(t_n)R(t_n) - C^{\mathcal{N}}(t_n) - \delta K(t_n)] \\ K(t_0) &= K_0 \quad \text{given} \quad C^i \geq 0 \end{aligned} \tag{14}$$

where θ^i_n is the discount factor of the i^{th} region and the sequence of discount factors θ^i_n for which the stationary solution of the continuous-time optimization problem is a stationary solution of the discrete-time problem. The sequences are unique within the choice of $\theta^{\mathcal{N}}_0$ and $\theta^{\mathcal{S}}_0$ and they are defined by the following recurrence:

$$\theta^{\mathcal{N}}_n = \frac{\theta^{\mathcal{N}}_{n-1}}{1+\rho_{\mathcal{N}}\Delta_n} \quad \theta^{\mathcal{N}}_0 > 0 \quad \text{and} \quad \theta^{\mathcal{S}}_n = \frac{\theta^{\mathcal{S}}_{n-1}}{1+\rho_{\mathcal{S}}\Delta_n} \quad \theta^{\mathcal{S}}_0 > 0$$

Δ_n is a scalar factor that converts the continuous flow into a stock increment. Δ may be the length of the time interval, $\Delta_n = t_{n+1} - t_n$, but, the specific choice of Δ_n plays no role in the sequel and the developments that follow are independent of the chosen grid. Finally, the functions $G^i(.)$ denote the value of terminal state where steady-state is already reached. Thus, we used the time aggregated approximation to handle the differential game with moderate size but constrained by computational capacities that limit the number of time intervals until steady-state is reached.

3.4.2 Terminal Condition

The role of terminal or transversality conditions in the solution of the economic models has been subject of controversy in the literature (Schiller 1977). The main reasons for this revolve around questions of uniqueness and stability. Any model should have the characteristic that for any given inputs only one set of outputs, only one unique solution, should exist.

If we choose to represent economic activity by means of differential or difference equations and a set of boundary conditions, the representation can be described as well posed if two criteria are satisfied. First the solution should be unique since experience suggests that a given set of economic circumstances leads to just one outcome. Secondly, the solution should be stable in the sense that an arbitrary small change in the boundary conditions should not result in explosive behavior. If the economic process under examination were dynamically unstable, then some trivial disturbances would cause the process to explode (Holy and Hughes-Hallett 1989). If the mathematical representation is non-unique or unstable then we must conclude that the problem is not well posed and could not be associated with economic process. The use of terminal conditions will not ensure a stable solution.

There still remains the problem of how the terminal conditions are actually chosen. A number of proposals have been made. One particular proposal which is used in this study is that the terminal conditions should be set at the equilibrium solution (steady state) of the model. The justification for this is that nonconvergent behavior would provide a response on the part of economic agents who would then respond so as to eliminate the non-convergence. One of the difficulties of using the equilibrium solution of the model as terminal conditions is the problem of determining it, especially when the model is large and non-linear. The extent to which the assumed terminal conditions corrupt the solution will depend both upon the stability (eigenvalues of the roots) and upon the size T.

3.4.3 Handling Constraints

In our model, we have linear constraints both as equalities (capital accumulation equation) and as inequalities (non-negativity constraints, $C_t^{\mathcal{N}} \geq 0$ and $C_t^{\mathcal{S}} \geq 0$ for all t). The equalities are eliminated at the start by substitution. Thus, we first substituted equation (14) into (12) for each region, North and South, and then, the inequalities are handled by *penalty functions*. The constrained problem is transformed to an unconstrained problem by associating a penalty with all constraint violations and the penalties are included in the evaluation function. The penalty term will be an arbitrary negative big number such as -10^8 since we are solving a maximization problem. Thus, if consumptions at time t are negative, the utility at that period becomes $-10^8 C^i(t)^2, \quad i = \mathcal{N}, \mathcal{S}$. Hence, the terms of trade and investment plans which make objective functions negative are immediately disregarded from the search space. In this study, we are using *GA* algorithm as a solution technique, we ignored the various new approaches to solve numeric optimization problems with linear constraints based on *GA* (Michalewicz and Janikow 1991) but rather used penalty functions in accordance with the general knowledge.

3.4.4 Simulation Results

For numerical analysis, we adopt the following particular functional forms:

$$U(C_t^i) = \begin{cases} \frac{C_t^{i\,1-\sigma}}{1-\sigma} & \text{for } \sigma > 0, \ \sigma \neq 0 \quad i = \mathcal{N}, \mathcal{S} \\ log\ C_t^i & \text{for } \sigma = 1 \end{cases}$$

and

$$D(\mathcal{P}_t) = \frac{d}{\gamma}\frac{R_t^\gamma}{K_t^\phi} \quad d > 0$$

where d converts pollution to disutility. Also,

$$Y_t = aK_t^\alpha R_t^\beta, \quad \alpha + \beta < 1 \text{ and } a > 0$$

All uncounted inputs u were normalized to one for simplicity.

The following set of parameter values are assumed:

$\alpha = 0.80$ $\beta = 0.15$ $\gamma = 2$ $a = 1$ $b = 1$ $d = 1e-5$

$\sigma = 1.50$ $\delta = 0.08$ $\rho_{\mathcal{N}} = 0.02$ $\rho_{\mathcal{S}} = 0.02$ $\omega = 0.50$ $\phi = 0.15$

These parameter values are assumed for the purposes of illustration, nevertheless they are not totally unjustified. Similar values of α, σ, δ, ρ_i and a are used by Auerbach and Kotlikoff (1987) in a different context. d is so chosen to conform with the assumed utility function. ϕ parameterizes the importance of the effetcs of knowledge spillovers in the North/South trade game. To highlight the significance of knowledge spillover, we run the experiment with $\phi = 0.30$ as well. β, γ, and b are chosen to satisfy parameter restrictions and are inconsequential to our arguments about knowledge spillovers[15].

In the time-aggregate model, we assume 21 periods (M=20) with a dense *equally* spaced gridding of the time horizon T ($t(M)$ =200) which is also quite sufficient to capture the convergence over time.

[15]The genetic operators in this paper were done using the public domain *GENESIS* package (Grefenstette, 1990) on a *SUN SPAC-1000* running *Solaris 2.4*. A typical run uses *population size*, j 50, runs 15 million generations for noncooperative game and 30 million generations for cooperative cases, *crossover rate* is 0.60 and *mutation rate* is 0.03. None of the results depends on the values of genetic operators other than run time by the choice of number of generations. For each parameter set, we have to implement three separate *GAs*. Hence, we are limited by the increased computational costs in our scope for a complete sensitivity analysis.

As mentioned earlier, we simultaneously run two separate genetic programs, $GA^{\mathcal{N}}$ and $GA^{\mathcal{S}}$, to solve the *noncooperative game.* $GA^{\mathcal{N}}$ generates a population of candidate solutions (chromosomes) $K(t)$ representing the Northern accumulated knowledge. The population of chromosomes $p(t)$ denotes theset of Southern price strategies and is produced by $GA^{\mathcal{S}}$. Structures, K_j, p_j in each population $(j = 1, 2, ..., 50)$ are represented as a binary string ($\{0\ 1\}$) of length l. For string j of length $l(= 10)$, decoding works as follows:

$$K_j(t) = \sum_{h=1}^{l} a_j^h(t) 2^{h-1} \qquad p_j(t) = \sum_{h=1}^{l} a_j^h(t) 2^{h-1}$$

where $a_j^h(t)$is the value $\{0\ 1\}$ taken at the hth position in the string. After strings are decoded, integers are normalized in order to obtain a real number value.

Since K_0 is given while p_0 is free, in every iteration (generation), $GA^{\mathcal{N}}$ computes M and $GA^{\mathcal{S}}$ $M+1$ structures each with a domain, $D_i = [\underline{d}, \bar{d}] \subseteq \Re$; $i = p, K$. D_i is cut into $(\bar{d} - \underline{d})2^{10}$ equal size ranges. Thus the noncooperative game has the minimal search domain of $2^{410} = 2.64423e + 123$.

Cooperative solutions are computationally much more complex than noncooperative ones. In the latter case, the search for the optimum consists of two one-dimensional problems, while the former represents one two-dimensional problem. In the cooperative experiment, three chromosomes, p_t, K_t and R_t, (62 structure) are searched in the minimal domain of 2^{620}.

Regional decisions are updated using genetic operators, selection, crossover, and mutation. The selection strategy is *elitist* so that the best performing strategy in the population of survivors is kept. This selection rule is a natural candidate in noncooperative Nash games. Therefore, it is especially crucial for the dynamic noncooperative game as it requires best responses be mutually exchanged. Were it not for the elitist selection, the best structures may disappear making for a nonconvergence.

The termination conditions are specified beforehand as certain number of iterations. We gradually increase the number of iterations until no further improvements are observed.

Tables 1 to 4 and Figures 1 to 3 summarize our numerical findings based on the assumed parameter values.

First, we note that as knowledge spills over more freely, the optimal long-run resource/capital ratio rises. This is so, since higher knowledge spillover lowers the long-run pollution cost thereby reducing the supply price and making increased use of resources for any given level of knowledge optimal.

Secondly, the higher degree of knowledge diffusion increases stationary capital stock and lowers the long-run pollution level irrespective of the mode of behavior. As we note above with higher ϕ, R/K rises. The marginal reduction in pollution due to higher K outweighs the incremental increase in pollution because of higher R so that the overall long-run pollution falls.

Figure 3 compares some results from the cooperative behavior with $\phi = 0.15$ and the noncooperative model with $\phi = 0.30$. Clearly, dissemination of knowledge strongly influences the long-run behavior. The cost of setting up institutions to monitor and enforce the international cooperation may be prohibitive. To the extent that knowledge diffusion can be enhanced at relatively less cost, regions may opt to cooperate on sharing of information. This indicates that even if the parties may fail, say due to enforcement problems, to realize the first best solution, they may still achieve significant improvement in welfare by strengthening the knowledge flow from North to South which lowers the cost of pollution to the South thereby lowering the supply price of the resources:

Total discounted welfares of each region under different mode of behavior

Rate of	North		South	
Diffusion (ϕ)	Noncooperative	Cooperative	Noncooperative	Cooperative
0.15	-0.058138	**-0.054725**	-0.072729	**-0.065882**
0.30	**-0.047346**	-0.045009	**-0.060418**	-0.054318

Both regions' welfares improve with the sufficient dissemination of knowledge.

From Figures 1 and 2, though cooperation yields significant increases in the levels of long-run knowledge stock and resource use, the optimal resource/knowledge mix is almost the same under both regimes. This is so because the long-run resource prices are almost the same. Also, notice that cooperation results in a higher long-run pollution level. Included in the noncooperative resource price is a monopoly rent which restricts growth and resource use. Consequently, the resulting level of pollution is less than what is warranted by a global welfare.

Table 1: The result of the noncooperative game with diffusion rate of 15 percent

t	K_t	p_t	R_t	$\mathcal{P}_t$	C_t^N	C_t^S	R_t/K_t	$\mathcal{P}_t/K_t$
0	1000000.000	77.967	283.125	5045.758	33064.556	22074.356	0.000283	0.005046
1	1120234.604	86.442	279.039	4818.407	35627.665	24120.597	0.000249	0.004301
2	1234604.106	94.252	276.186	4652.048	38184.480	26031.175	0.000224	0.003768
3	1340175.953	101.564	273.253	4498.064	40373.921	27752.721	0.000204	0.003356
4	1436950.147	108.045	271.306	4388.060	42354.554	29313.215	0.000189	0.003054
5	1524926.686	114.027	269.286	4284.599	44088.316	30705.942	0.000177	0.002810
6	1604105.572	119.345	267.680	4201.621	45662.203	31946.254	0.000167	0.002619
7	1674486.804	124.164	266.037	4123.555	46772.060	33032.227	0.000159	0.002463
8	1739002.933	128.485	264.800	4062.196	48103.356	34022.724	0.000152	0.002336
9	1794721.408	132.141	263.921	4016.228	49353.218	34874.649	0.000147	0.002238
10	1841642.229	135.298	263.001	3972.867	50203.064	35583.532	0.000143	0..002157
11	1882697.947	137.957	262.438	3942.815	51027.299	36205.093	0.000139	0.002094
12	1917888.563	140.283	261.849	3914.248	51790.614	36733.103	0.000137	0.002041
13	1947214.076	142.278	261.241	3887.233	52206.763	37168.797	0.000134	0.001996
14	1973607.038	143.939	260.983	3871.732	52931.005	37565.710	0.000132	0.001962
15	1994134.897	145.269	260.702	3857.408	53316.818	37871.848	0.000131	0.001934
16	2011730.205	146.432	260.411	3843.733	53386.591	38132.566	0.000129	0.001911
17	2029325.513	147.595	260.122	3830.200	53746.547	38392.741	0.000128	0.001887
18	2043988.270	148.592	259.824	3817.304	54085.868	38607.874	0.000127	0.001868
19	2055718.475	149.423	259.519	3805.080	54405.853	38778.193	0.000126	0.001851
20	2064516.129	149.922	259.545	3803.405	55337.091	38911.488	0.000126	0.001842

Table 2: The result of the cooperative game with diffusion rate of 15 percent

t	K_t	p_t	R_t	$\mathcal{P}_t$	C_t^N	C_t^S	R_t/K_t	$\mathcal{P}_t/K_t$
0	1000000.000	51.818	598.651	22558.876	31053.778	31021.008	0.000599	0.022559
1	1225806.448	60.909	586.921	21031.282	35933.968	35748.814	0.000479	0.017157
2	1460410.552	69.853	584.233	20298.802	40712.731	40810.621	0.000400	0.013899
3	1697947.216	79.384	574.457	19186.562	45916.300	45602.827	0.000338	0.011300
4	1923753.664	87.889	571.281	18622.863	50150.338	50209.026	0.000297	0.009680
5	2140762.468	96.979	564.927	17921.275	54982.562	54786.292	0.000264	0.008371
6	2331378.304	106.657	550.020	16771.980	57228.799	58663.375	0.000236	0.007194
7	2507331.376	112.815	548.553	16501.593	61906.551	61885.174	0.000219	0.006581
8	2653958.944	117.361	543.666	16071.253	64431.200	63804.988	0.000205	0.006056
9	2791788.856	123.079	541.711	15835.157	66743.358	66673.302	0.000194	0.005672
10	2912023.456	126.745	542.199	15763.725	69246.953	68720.993	0.000186	0.005413
11	3017595.304	132.023	540.733	15595.052	70757.598	71389.460	0.000179	0.005168
12	3105571.852	135.689	542.444	15626.375	72579.693	73603.737	0.000175	0.005032
13	3175953.076	138.622	541.711	15531.866	74402.654	75092.852	0.000171	0.004890
14	3228739.000	139.648	542.688	15549.477	75613.083	75785.369	0.000168	0.004816
15	3275659.828	143.460	542.199	15487.926	76374.164	77784.150	0.000166	0.004728
16	3304985.332	143.607	541.222	15411.513	78440.272	77723.273	0.000164	0.004663
17	3319648.096	143.607	540.000	15331.820	74690.809	77547.801	0.000163	0.004619
18	3375366.568	145.806	541.711	15390.638	77901.150	78984.908	0.000160	0.004560
19	3398826.976	137.302	547.087	15681.350	76759.806	75116.168	0.000161	0.004614
20	3483870.964	148.739	540.000	15221.176	81028.038	80319.062	0.000155	0.004369

Table 3: The result of the noncooperative game with diffusion rate of 30 percent

t	K_t	p_t	R_t	$\mathcal{P}_t$	C_t^N	C_t^S	R_t/K_t	$\mathcal{P}_t/K_t$
0	1000000.000	33.343	769.100	4687.433	40390.275	25644.170	0.000769	0.004687
1	1249266.862	40.205	760.908	4291.783	47658.693	30592.521	0.000609	0.003435
2	1506842.620	47.067	754.121	3985.032	54830.827	35494.555	0.000500	0.002645
3	1764418.377	53.803	747.500	3734.328	60987.681	40217.422	0.000424	0.002116
4	2021994.135	60.283	743.355	3545.101	66417.730	44812.030	0.000368	0.001753
5	2279569.892	66.637	739.617	3385.541	72826.629	49286.121	0.000324	0.001485
6	2520527.859	72.610	734.888	3243.138	78298.376	53360.227	0.000292	0.001287
7	2744868.035	77.947	732.542	3141.080	83202.928	57099.637	0.000267	0.001144
8	2952590.420	82.903	729.719	3049.443	87493.390	60496.074	0.000247	0.001033
9	3143695.015	87.478	726.686	2967.777	91280.228	63569.031	0.000231	0.000944
10	3318181.818	91.417	725.973	2914.347	94835.891	66366.589	0.000219	0.000878
11	3476050.831	95.103	723.980	2858.231	97954.462	68852.409	0.000208	0.000822
12	3617302.053	98.280	723.134	2817.681	100047.483	71069.273	0.000200	0.000779
13	3750244.379	101.329	721.700	2776.297	102748.345	73129.473	0.000192	0.000740
14	3866568.915	103.998	720.379	2740.904	106069.849	74917.967	0.000186	0.000709
15	3957966.764	106.031	719.810	2717.463	107546.961	76322.329	0.000182	0.000687
16	4041055.718	108.192	716.810	2678.120	108703.692	77552.851	0.000177	0.000663
17	4115835.777	109.589	718.356	2674.931	111021.151	78724.278	0.000175	0.000650
18	4173998.045	110.860	718.100	2661.795	112211.082	79608.752	0.000172	0.000638
19	4223851.417	112.004	717.454	2647.559	113297.510	80357.660	0.000170	0.000627
20	4265395.894	113.021	716.437	2632.319	117610.283	80972.110	0.000168	0.000617

Table 4: The results of the cooperative game with diffusion rate of 30 percent

t	K_t	p_t	R_t	$\mathcal{P}_t$	C_t^N	C_t^S	R_t/K_t	$\mathcal{P}_t/K_t$
0	1000000.000	23.072	1597.290	20217.970	37433.968	36852.467	0.001597	0.020218
1	1364824.609	29.335	1591.687	18287.835	46345.522	46692.902	0.001166	0.013399
2	1787925.607	35.152	1608.048	17213.262	56806.064	56525.621	0.000899	0.009628
3	2263924.062	42.422	1591.687	15711.831	67700.291	67522.498	0.000703	0.006940
4	2766351.580	48.686	1593.480	14828.288	77747.379	77579.514	0.000576	0.005360
5	3304057.180	54.726	1604.014	14245.315	87214.859	87780.484	0.000485	0.004311
6	3877001.532	61.548	1604.462	13585.639	99967.284	98752.037	0.000414	0.003504
7	4432326.508	66.582	1627.547	13429.120	108431.263	108364.748	0.000367	0.003030
8	5014080.547	73.964	1618.358	12795.634	117888.573	119699.843	0.000323	0.002552
9	5569405.523	79.109	1615.893	12360.970	126898.505	127831.485	0.000290	0.002219
10	6115920.811	85.820	1611.634	11955.405	140779.303	138310.269	0.000264	0.001955
11	6565450.873	91.524	1602.221	11567.393	146735.396	146642.091	0.000244	0.001762
12	6979742.183	98.123	1591.015	11198.678	152487.541	156115.849	0.000228	0.001604
13	7332326.348	101.815	1572.188	10774.731	159340.646	160071.535	0.000214	0.001469
14	7640822.743	102.933	1595.273	10957.160	164925.320	164206.236	0.000209	0.001434
15	7940548.78	108.190	1586.308	10710.010	169193.976	171622.663	0.000200	0.001349
16	8178528.343	109.532	1581.602	10552.658	178158.237	173236.347	0.000193	0.001290
17	8354800.761	108.637	1622.392	11033.182	173812.724	176252.449	0.000194	0.001321
18	8619248.717	115.125	1595.049	10565.188	184793.166	183629.538	0.000185	0.001226
19	8725043.631	117.585	1581.378	10346.920	186627.283	185946.965	0.000181	0.001186
20	8795560.464	120.270	1561.206	10060.310	191094.069	187765.933	0.000177	0.001144

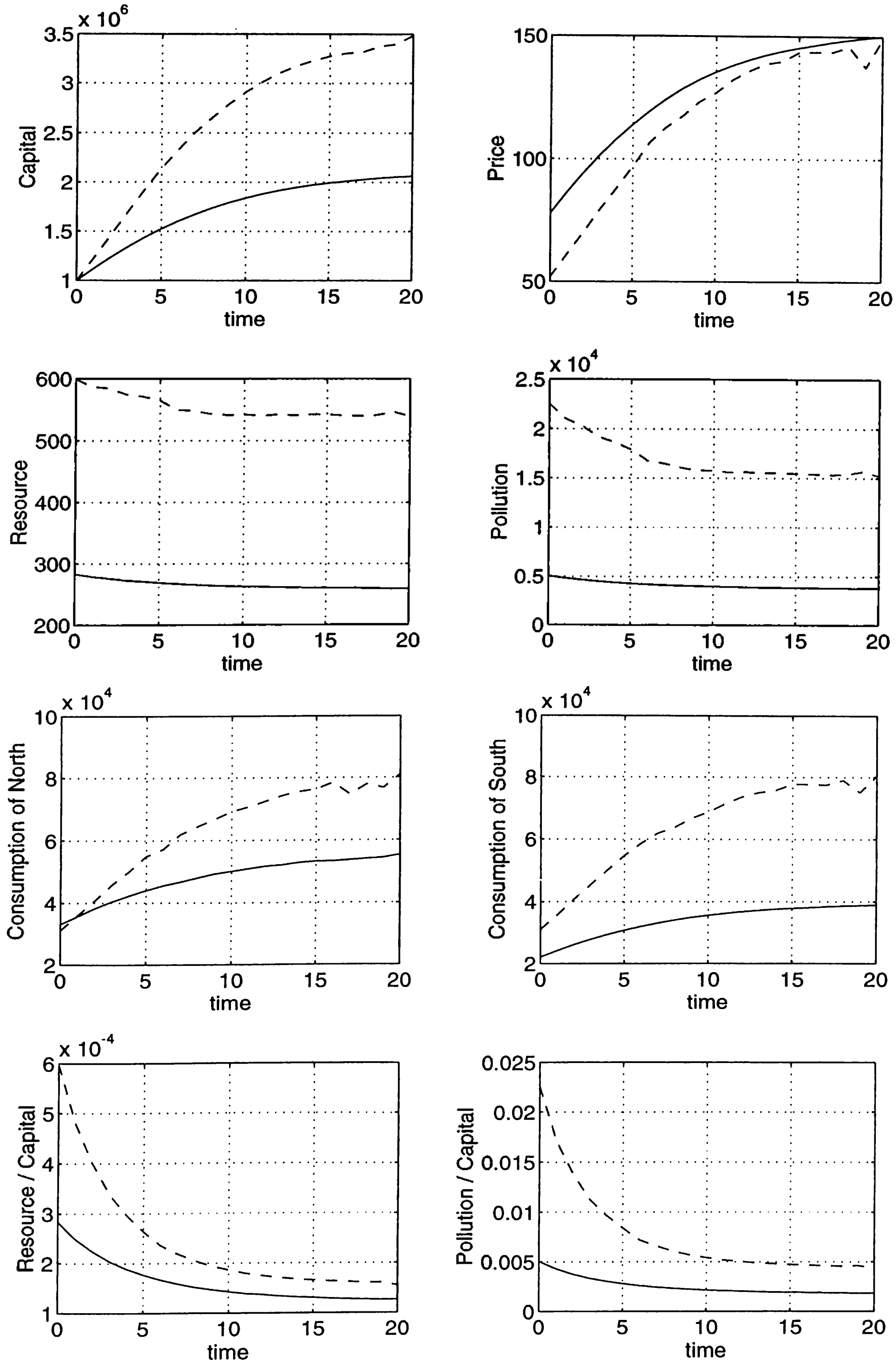


Figure 1: The figures related to diffusion rate of 15 percent. Solid lines are used to denote noncooperative optimal strategies.

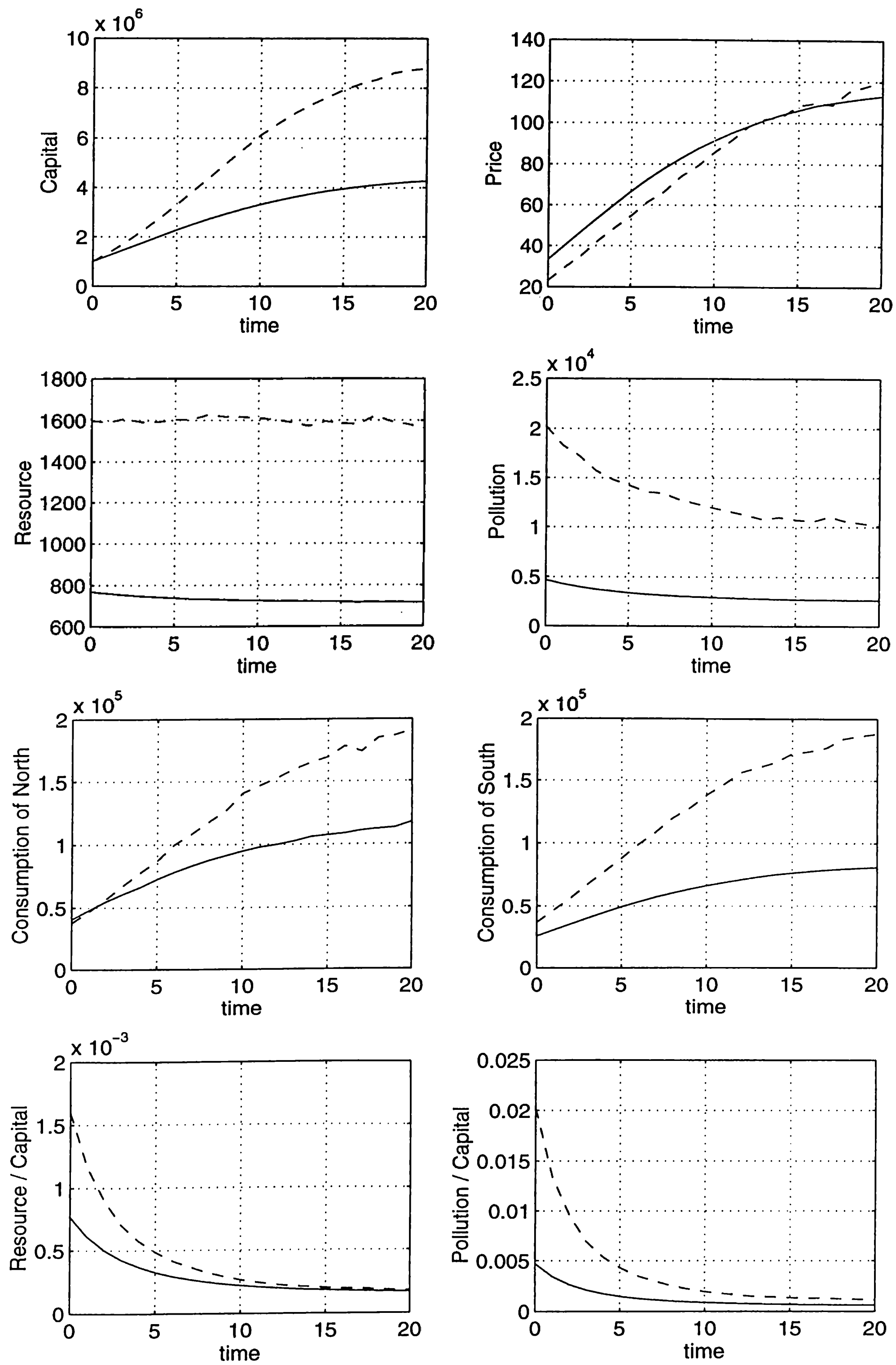


Figure 2: The figures related to diffusion rate of 30 percent. Solid lines are used to denote noncooperative optimal strategies.

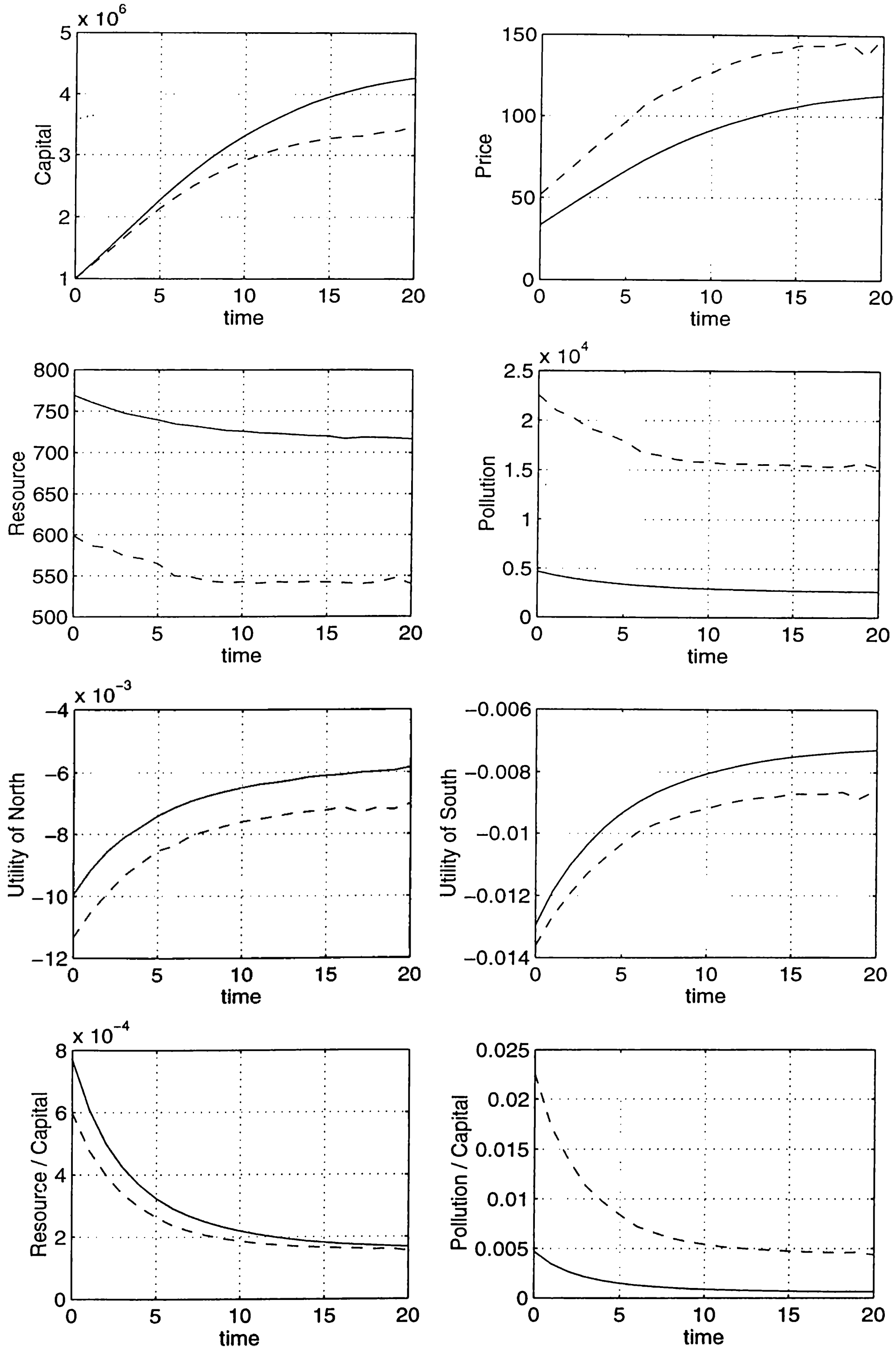


Figure 3: Solid lines denote result of noncooperative game with rate of diffusion of 30 percent and dashed lines denote result of cooperative game with rate of diffusion of 15 percent.

As far as the transition paths are concerned, the most striking difference is that along the noncooperative path, resource prices are initially too high and therefore the rate of investment is too low relative to the cooperative case. Moreover, the transition to the stationary equilibrium is more rapid along the cooperative path. The resource price starts lower but ultimately catches up with the noncooperative resource price.

In the noncooperative mode the shadow value (the marginal benefit) of the knowledge stock differs for the two regions as the regions have different preferences (fitnesses) leading to conflicting policies and harmful "policy externalities." Moreover, when policies are chosen with a view to maximize own fitness taking the rival's as given, the "incentive" effects of the policies are ignored. The South chooses resource prices for any "given investment policy" of the North, thus ignoring the fact that a lower price today (lower consumption) may "induce" the North to invest more today which then leads to higher prices (higher Southern consumption) as the higher knowledge stock shifts the demand for resources tomorrow. The North on the other hand, ignores the fact that, an initially higher investment profile (lower consumption) may induce South to ask for lower resource prices today in return for higher prices tomorrow (as the demand for resources will shift) and also to higher Northern consumption in the future as the amount to be invested will be lower in the future (higher Northern consumption). Parties ignore the incentive effect for the fact that promises are not credible. If South were to offer prices along the cooperative path, it is not optimal for the North to invest as much as promised along the cooperative path. North will consume more and invest less. Likewise if North commits itself to the investment plan along the cooperative path, then it will not be optimal for the South to ask for prices along the cooperative path. South will ask higher prices and consume more. Hence, parties will employ Nash strategies to maximize respective fitnesses. Note that these effects are independent of the presence of monopoly power and the pollution and knowledge externalities, but obviously will further compound the ineffiencies that stem from them.

3.5 Conclusion

A genetic algorithm is introduced to search for optimal policies in the presence of knowledge spillover and local pollution in a dynamic North/South trade game. Noncooperative trade compounds inefficiencies stemming from externalities. Competitive resource production in the South would overpollute whereas "local" internalization of pollution together with resource monopoly limit growth and trade and result in underpollution.

Because of the spillovers, the stock of knowledge is partially a common property (Grossman and Helpman 1991). For the fact that benefits from investment in knowledge can not be totally captured by the North, there is underinvestment. In the presence of knowledge spillovers, there is an added incentive on the part of the North to cooperate as the benefit from knowledge spillovers are internalized in the form of reduced waste which is then reflected to the North as initial lower resource prices.

Cooperative trade policies are efficient and yet not credible. Nonetheless, even if parties trade noncooperatively, dynamic gains still materialize if the South's access to the North's stock of accumulated knowledge is enhanced. Short of a joint maximization of the global welfare, transfer of knowledge remains as the viable route to improve global welfare.

Appendix

A.1. *First Order Condition for Noncooperative Game:*

We begin by looking at the North's problem. The current value Hamiltonian is

$$H^{\mathcal{N}}(C^{\mathcal{N}}, R, K, \mu_{\mathcal{N}}) = U(C^{\mathcal{N}}) + \mu_{\mathcal{N}}[F(K,R) - pR - \delta K - C^{\mathcal{N}}]$$

where $\mu_{\mathcal{N}}$ is the costate variable for the North associated with the capital accumulation equation (2). $\mu_{\mathcal{N}}$ may be interpreted as the shadow value of a marginal increase in the capital accumulation of all kinds. The standard optimal control procedure yields the following first order necessary conditions:

$$\frac{\partial H}{\partial C^{\mathcal{N}}} = \frac{\partial U}{\partial C^{\mathcal{N}}} - \mu_{\mathcal{N}} = 0 \qquad (i)$$

$$\frac{\partial H}{\partial R} = \mu_{\mathcal{N}}[F_R - p] = 0 \qquad (ii)$$

$$-\frac{\partial H}{\partial K} = \dot{\mu}_{\mathcal{N}} = \mu_{\mathcal{N}}[\rho_{\mathcal{N}} + \delta - F_K] \qquad (iii)$$

and equation (2) as well as the usual transversality condition. F_i denotes $\partial F/\partial i$.

Equation (i) gives us the usual consumption smoothing condition that the imputed price of consumption at each period equals to its marginal utility, (ii) implies that resource is demanded upto its marginal benefit of resource (marginal productivity) equals its marginal cost (price of the resource) and equation (iii) implies that the rate of growth in the imputed price of investment indicating capital gains is equal to the difference between marginal productivity of capital and the marginal cost of the capital, $\rho_{\mathcal{N}} + \delta$.

A similar method applies to the South's maximization problem where p is the control variable, μ_S is the costate variable and K is the state variable

$$H^S(p, K, \mu_S) = U(C(p,R)) - U(\mathcal{P}(R(p), K)) + \mu_S[F(K, R(p)) - pR(p) - \delta K - C^{\mathcal{N}}]$$

and the first order conditions are:

$$\frac{\partial H}{\partial p} = \underbrace{\frac{\partial U}{\partial C^S}\frac{\partial C^S}{\partial p}}_{R} + \underbrace{\frac{\partial U}{\partial C^S}\frac{\partial C^S}{\partial R}}_{p}\frac{\partial R}{\partial p} - \underbrace{\frac{\partial U}{\partial \mathcal{P}}\frac{\partial \mathcal{P}}{\partial R}}_{\text{pollution externality}}\frac{\partial R}{\partial p}$$

$$+\mu_S[\frac{\partial F}{\partial R}\frac{\partial R}{\partial p} - R - p\frac{\partial R}{\partial p}] = 0 \qquad (iv)$$

$$-\frac{\partial H}{\partial K} = \dot{\mu}_S = \underbrace{\frac{\partial U}{\partial \mathcal{P}}\frac{\partial \mathcal{P}}{\partial K}}_{\substack{\text{knowledge}\\ \text{externality}}} + \mu_S[\rho_S + \delta - F_K] \qquad (v)$$

and equation (2) and the usual transversality condition. The equation (iv) is rearranged and written as

$$R + \frac{\partial R}{\partial p}\Big[p - \frac{\partial U}{\partial \mathcal{P}}\frac{\partial \mathcal{P}}{\partial R}\Big] = \mu_S[R - F_R + p]$$

where the first term is the marginal benefit of price increase on the consumption and utility of the South and the term before the bracket is the price elasticity of resource and inside the bracket net welfare benefit of resource sales (utility minus disutility). This equation says that the net welfare benefit from price changes (directlty or indirectly through resource demand changes) equals to the shadow price of an increase in investment in the North times the difference between resource demanded and the net benefit of resource bought by the North. The interpretation of eq. (v) is different than the eq. (iii) since there is the term of knowledge externality. Thus, marginal valuation of Norht's investment in the South includes the welfare effect of knowledge spillover as well. To solve such problems with traditional methods, we have to construct the state space representation of these *F.O.C.'s* and solve the model for $K_t, p_t, C_t^{\mathcal{N}}, \mu_{\mathcal{N}}$ and μ_S.

First Order Condition for Cooperative Game:

Now, let's define the current Hamiltonian as

$$\begin{aligned} H(C^i, p, R, K, \mu) &= \omega U(C^{\mathcal{N}}) + (1-\omega)[U(C^{\mathcal{S}}(p,R)) - U(\mathcal{P})] \\ &\quad + \mu[F(K,R,u) - pR - \delta K - C^{\mathcal{N}}] \end{aligned}$$

The necessary conditions for maximum are:

$$\frac{\partial H}{\partial C^{\mathcal{N}}} = \omega\frac{\partial U}{\partial C^{\mathcal{N}}} - \mu = 0 \qquad (vi)$$

$$\frac{\partial H}{\partial p} = (1-\omega)\underbrace{\frac{\partial U}{\partial C^{\mathcal{S}}}\frac{\partial C^{\mathcal{S}}}{\partial p}}_{R} - \mu R = 0 \qquad (vii)$$

$$\frac{\partial H}{\partial R} = (1-\omega)[\underbrace{\frac{\partial U}{\partial C^{\mathcal{S}}}\frac{\partial C^{\mathcal{S}}}{\partial R}}_{p} - \underbrace{\frac{\partial U}{\partial \mathcal{P}}\frac{\partial \mathcal{P}}{\partial R}}_{\text{pollution externality}}] + \mu[F_R - p] = 0 \qquad (viii)$$

$$-\frac{\partial H}{\partial K} = \dot{\mu} = \mu[\rho + \delta - F_K] + (1-\omega)\underbrace{\frac{\partial U}{\partial \mathcal{P}}\frac{\partial \mathcal{P}}{\partial K}}_{\text{knowledge externality}} \qquad (ix)$$

and equation (2) and the transversality conditions. The equations (vi) and (vii) might be written as the ratio

$$\frac{\partial U/\partial C^{\mathcal{N}}}{\partial U/\partial C^{\mathcal{S}}} = \frac{(1-\omega)}{\omega}$$

which says that the ratio of the marginal utilities of the North's consumption to the South's consumption is equal to the ratio of the cooperative agreements (weights). Finally, if we rearrange eq. (viii)

$$\frac{(1-\omega)}{\omega}\left[\frac{U_{C^{\mathcal{S}}}}{U_{C^{\mathcal{N}}}}C_R + \frac{U_{\mathcal{P}}}{U_{C^{\mathcal{N}}}}\mathcal{P}_R\right] = F_R - p$$

it gives weighted marginal net utility (utility minus disutility) of resource usage on South normalized to the marginal benefit of this on consumption of North equals the difference between marginal productivity of resource and price of that resource. Finally equation (ix) is interpreted as the rate of growth in the imputed price of the investment indicating capital gain which is now difference between marginal benefit and marginal cost of capital plus positive effect of knowledge spillover on the growth of capital gains

Thus, as long as appropriate simplifications on functional forms, we can solve the simulatenous systems of equations $R, K, p, C^{\mathcal{N}}$, and μ.

A.2.1. *C codes of the main program for the noncooperative game between North and South*

North:

```
#include "extern.h"
#include "semop.h"

#define SHMKEY_P ((key_t) 2890)
#define SHMKEY_S ((key_t) 2891)
#define SEMKEY1 ((key_t) 3893)
#define SEMKEY2 ((key_t) 3894)
#define SEMKEY3 ((key_t) 3895)

#define NORTH  1

#define TMAX 21
#define (double)A    1
#define (double)B    0.15
#define (double)D    0.0000001
#define (double)ALPHA 0.80
#define (double)GAMMA 0.15
#define (double)RHON  0.02
#define (double)RHOS  0.02
#define (double)SIGMA 1.5
#define (double)DELTA 0.08
#define (double)K0   1000000.0
typedef double Mesg[TMAX];


int shmid;
int shmsem,syncsem1,syncsem2;
Mesg *p;
Mesg *s;


getshared()
{
        if((shmid=shmget(SHMKEY_P,sizeof(Mesg),0666 | IPC_CREAT))<0)
                {
                printf("server can't get shared memory");
                exit(1);
                }
        if((p=(Mesg *) shmat(shmid,(char*)0,0)) == (Mesg *) -1)
                {
                printf("server can't attach shared memory");
                exit(1);
                }
        if((shmid=shmget(SHMKEY_S,sizeof(Mesg),0666 | IPC_CREAT))<0)
                {
```

```
                printf("server can't get shared memory");
                exit(1);
                }
        if((s=(Mesg *) shmat(shmid,(char*)0,0)) == (Mesg *) -1)
                {
                printf("server can't attach shared memory");
                exit(1);
                }
}

relshared()
{
        if(shmdt(p)<0)
                {
                printf("server can't detach shared memory");
                exit(1);
                }
        if(shmdt(s)<0)
                {
                printf("server can't detach shared memory");
                exit(1);
                }
        sem_signal(syncsem1);
        sem_signal(syncsem2);
/*    sem_close(syncsem1);
  sem_close(syncsem2);
  sem_close(shmsem);*/
}

initshared()
{
#if NORTH
        if ( (syncsem1 = sem_create(SEMKEY1, 0)) < 0)
                err_sys("north: can't create sync semaphore");

        if ( (syncsem2 = sem_create(SEMKEY3, 0)) < 0)
                err_sys("north: can't create sync semaphore");

        if ( (shmsem = sem_create(SEMKEY2, 1)) < 0)
                err_sys("north: can't create shm semaphore");
#else
        if ( (syncsem1 = sem_open(SEMKEY1)) < 0)
                err_sys("south: can't create sync semaphore");

        if ( (syncsem2 = sem_open(SEMKEY3)) < 0)
                err_sys("south: can't create sync semaphore")
        if ( (shmsem = sem_open(SEMKEY2)) < 0)
                err_sys("south: can't create shm semaphore");
#endif
        sem_wait(shmsem);
        Unpack(New[0].Gene, Bitstring, Length);
        if (Floatflag)
                {
                FloatRep(Bitstring, Vector, Genes);
```

```
#if NORTH
                    memcpy(s,Vector,sizeof(double)*TMAX);
#else
                    memcpy(p,Vector,sizeof(double)*TMAX);
#endif
                    }
  sem_signal(shmsem);
          synchronize();
}

synchronize()
{
#if NORTH
          sem_signal(syncsem1);
          sem_wait(syncsem2);
#else
  sem_signal(syncsem2);
  sem_wait(syncsem1);
#endif
}

movebesttoshared()
{
          sem_wait(shmsem);
          Unpack(New[Best_guy].Gene, Bitstring, Length);
          if (Floatflag)
                    {
                    FloatRep(Bitstring, Vector, Genes);
#if NORTH
                    memcpy(s,Vector,sizeof(double)*TMAX);
#else
                    memcpy(p,Vector,sizeof(double)*TMAX);
#endif
                    }
          sem_signal(shmsem);
}

double eval(str, length, vect, genes)
char str[];       /* string representation          */
int length;       /* length of bit string           */
double vect[];    /* floating point representation  */
int genes;        /* number of elements in vect     */

{
register int t;
double ans=0;
double k[TMAX];
double n[TMAX];
double r[TMAX];
double y[TMAX];
double c[TMAX];
double rhocoef=1;
double interv=10;
static unsigned char first = 1;
```

```
static double c1,c2,c3;

     if (first)
      {
        c1=pow(A, (1/(1GAMMA))) *pow(GAMMA,(GAMMA/(1-GAMMA));
        c2=ALPHA/(1-GAMMA);
        c3=-GAMMA/(1-GAMMA);
        first=0;
      }
        sem_wait(shmsem);

        for (t = 0; t <(TMAX-1); t++)
         {
          if (t==0)
            {
             k[0]=K0;
             y[t]=c1*pow((*p)[t],c3)*pow(k[t],c2);
             r[t]=(GAMMA/(*p)[t])*y[t];
             c[t]=y[t]-((*p)[t]*r[t])-(k[t]*DELTA)-((double)(1/interv)*(vect[t]-k[t]));
              if (c[t]<0)
                 ans+=-100000000*c[t]*c[t];
              else
                  ans+=rhocoef*interv*(pow(c[t],(1-SIGMA))/(1-SIGMA));
               }
               else
               {
                k[t]=vect[t-1];
                y[t]=c1*pow((*p)[t],c3)*pow(k[t],c2);
                r[t]=(GAMMA/(*p)[t])*y[t];
                c[t]=y[t]-((*p)[t]*r[t])-(k[t]*DELTA)-((double)(1/interv)*(vect[t]-k[t]));
                if (c[t]<0)
                  ans+=-100000000*c[t]*c[t];
                 else
                  {
                  rhocoef*=(double)(1/(1+(interv*(double)RHON)));
                  ans+=rhocoef*interv*(pow(c[t],(1-SIGMA))/(1-SIGMA));
                  }
                  }
        }
       k[t]=vect[t-1];
       y[t]=c1*pow((*p)[t],c3)*pow(k[t],c2);
       r[t]=((double)GAMMA/(*p)[t])*y[t];
       c[t]=y[t]-(*p)[t]*r[t]-(k[t]*DELTA);
       if (c[t]<0)
          ans+=-100000000*c[t]*c[t];
       else
         {
          ans+=(rhocoef/RHON)*(pow(c[t],(1-SIGMA))/(1-SIGMA));
         }
        sem_signal(shmsem);
        return(ans);
}
```

South:

```
#include "extern.h"
#include "semop.h"

#define SHMKEY_P ((key_t) 2890)
#define SHMKEY_S ((key_t) 2891)
#define SEMKEY1 ((key_t) 3893)
#define SEMKEY2 ((key_t) 3894)
#define SEMKEY3 ((key_t) 3895)

#define TMAX   21
#define (double)A      1
#define (double)B      0.15
#define (double)D      0.0000001
#define (double)ALPHA   0.80
#define (double)GAMMA   0.15
#define (double)RHON   0.02
#define (double)RHOS   0.02
#define (double)SIGMA   1.5
#define (double)DELTA   0.08
#define (double)K0      1000000.0
typedef double Mesg[TMAX];

int shmid;
int shmsem,syncsem1,syncsem2;
Mesg *p;
Mesg *s;

getshared()
{
        if((shmid=shmget(SHMKEY_P,sizeof(Mesg),0666 | IPC_CREAT))<0)
                {
                printf("server can't get shared memory");
                exit(1);
                }
        if((p=(Mesg *) shmat(shmid,(char*)0,0)) == (Mesg *) -1)
                {
                printf("server can't attach shared memory");
                exit(1);
                }
        if((shmid=shmget(SHMKEY_S,sizeof(Mesg),0666 | IPC_CREAT))<0)
                {
                printf("server can't get shared memory");
                exit(1);
                }
        if((s=(Mesg *) shmat(shmid,(char*)0,0)) == (Mesg *) -1)
                {
                printf("server can't attach shared memory");
                exit(1);
```

```
                }
}

relshared()
{
        if(shmdt(p)<0)
                {
                printf("server can't detach shared memory");
                exit(1);
                }
        if(shmdt(s)<0)
                {
                printf("server can't detach shared memory");
                exit(1);
                }
        sem_signal(syncsem1);
        sem_signal(syncsem2);
/*      sem_close(syncsem1);
        sem_close(syncsem2);
        sem_close(shmsem);*/
}


initshared()
{

#if NORTH
        if ( (syncsem1 = sem_create(SEMKEY1, 0)) < 0)
                err_sys("north: can't create sync semaphore");

        if ( (syncsem2 = sem_create(SEMKEY3, 0)) < 0)
                err_sys("north: can't create sync semaphore");

        if ( (shmsem = sem_create(SEMKEY2, 1)) < 0)
                err_sys("north: can't create shm semaphore");
#else
        if ( (syncsem1 = sem_open(SEMKEY1)) < 0)
                err_sys("south: can't create sync semaphore");

        if ( (syncsem2 = sem_open(SEMKEY3)) < 0)
                err_sys("south: can't create sync semaphore");

        if ( (shmsem = sem_open(SEMKEY2)) < 0)
                err_sys("south: can't create shm semaphore");
#endif

        sem_wait(shmsem);
        Unpack(New[0].Gene, Bitstring, Length);
        if (Floatflag)
                {
                FloatRep(Bitstring, Vector, Genes);
#if NORTH
                memcpy(s,Vector,sizeof(double)*TMAX);
```

```
#else
                memcpy(p,Vector,sizeof(double)*TMAX);
#endif
                }
        sem_signal(shmsem);
        synchronize();
}

synchronize()
{
int semval;
#if NORTH
        sem_signal(syncsem1);
        sem_wait(syncsem2);
#else
        sem_signal(syncsem2);
        sem_wait(syncsem1);
#endif
/*semval = semctl(shmsem, 0, GETVAL, 0);
printf("%d ",semval);
semval = semctl(syncsem1, 0, GETVAL, 0);
printf("%d ",semval);
semval = semctl(syncsem2, 0, GETVAL, 0);
printf("%d ",semval);
printf("\n");*/
}

movebesttoshared()
{
        sem_wait(shmsem);
        Unpack(New[Best_guy].Gene, Bitstring, Length);
        if (Floatflag)
                {
                FloatRep(Bitstring, Vector, Genes);
#if NORTH
                memcpy(s,Vector,sizeof(double)*TMAX);
#else
                memcpy(p,Vector,sizeof(double)*TMAX);
#endif
                }
        sem_signal(shmsem);
}

double eval(str, length, vect, genes)
char str[];     /* string representation          */
int length;     /* length of bit string           */
double vect[];  /* floating point representation  */
int genes;      /* number of elements in vect     */



{
register int t;
double ans=0;
```

```
double k[TMAX];
double y[TMAX];
double n[TMAX];
double r[TMAX];
double c[TMAX];
double u[TMAX];
double rhocoef=1;
double interv1=10;
static unsigned char first = 1;
static double c1,c2,c3;

    if (first)
     {
       c1=pow(A,(double)(1/(1-GAMMA)))*pow(GAMMA,(double)(GAMMA/(1-GAMMA)));
       c2=ALPHA/(1-GAMMA);
       c3=-GAMMA/(1-GAMMA);
       first=0;
     }

       sem_wait(shmsem);

       for (t = 0; t < (TMAX-1) ; t++)
        {
         if (t==0)
           {
            k[0]=K0;
            y[t]=c1*pow(vect[t],c3)*pow(k[t],c2);
            r[t]=((double)GAMMA/vect[t])*y[t];
            u[t]=r[t]*r[t]*0.5*D*pow(k[t],(double)(-B));
            c[t]=(pow(vect[t]*r[t],(1-SIGMA)))/(1-SIGMA);
             ans+=rhocoef*interv*(c[t]-u[t]);
            }
            else
           {
            k[t]=(*s)[t-1];
            y[t]=c1*pow(vect[t],c3)*pow(k[t],c2);
            r[t]=((double)GAMMA/vect[t])*y[t];
            u[t]=r[t]*r[t]*0.5*D*pow(k[t],(double)(-B));
            c[t]=(pow(vect[t]*r[t],(1-SIGMA)))/(1-SIGMA);
            rhocoef*=(double)(1/(1+interv*(double)RHOS));
            ans+=rhocoef*interv*(c[t]-u[t]);
           }
           }
      k[t]=(*s)[t-1];
      y[t]=c1*pow(vect[t],c3)*pow(k[t],c2);
      r[t]=((double)GAMMA/vect[t])*y[t];
      u[t]=r[t]*r[t]*0.5*D*pow(k[t],(double)(-B));
      c[t]=(pow(vect[t]*r[t],(1-SIGMA)))/(1-SIGMA);
      ans+=(rhocoef/((double)RHOS))*(c[t]-u[t]);
      sem_signal(shmsem);
      return(ans);}
```

A.2.2 *C codes of the main program for the cooperative game*

```
#include "extern.h"
#include "math.h"

#define NORTH  1

#define TMAX   21
#define (double)A      1
#define (double)D      0.0000001
#define (double)ALPHA  0.80
#define (double)GAMMA  0.15
#define (double)BETA   0.30
#define (double)RHON   0.02
#define (double)RHOS   0.02
#define (double)SIGMA  1.5
#define (double)DELTA  0.08
#define (double)K0     1000000.0

double eval(str, length, vect, genes)
char str[];        /* string representation          */
int length;        /* length of bit string           */
double vect[];  /* floating point representation     */
int genes;         /* number of elements in vect     */

{
register int t;
double ans=0;
double k[TMAX];
double y[TMAX];
double d[TMAX];
double c[TMAX];
double u[TMAX];
double h[TMAX];
double m[TMAX];
double v[TMAX];
double rhocoefn=1;
double interv1=10;
static unsigned char first = 1;

        for (t = 0; t < (TMAX-1); t++)
          {
          if (t==0)
            {
            k[0]=K0;
            m[t]=vect[t+20]*vect[t+41];
            y[t]=A*pow(vect[t+20],GAMMA)*pow(k[t],ALPHA);
            d[t]=(.5*D*vect[t+20]*vect[t+20])/pow(k[t],BETA);
            c[t]=y[t]-(k[t]*DELTA)-m[t]-((double)(1/interv1)*(vect[t]-g[t]));
            u[t]=(pow(c[t],(double)(1-SIGMA)))/(1-SIGMA);
```

```
            v[t]=(pow(m[t],(double)(1-SIGMA)))/(1-SIGMA);
            if (c[t]<0)
              ans+=-100000000*c[t]*c[t];
            else
              ans+=rhocoefn*interv1*((u[t]+v[t])-d[t]);
            }
            else
            {
            k[t]=vect[t-1];
            m[t]=vect[t+20]*vect[t+41];
            y[t]=A*pow(vect[t+20],GAMMA)*pow(k[t],ALPHA);
            d[t]=(.5*D*vect[t+20]*vect[t+20])/pow(k[t],BETA);
            c[t]=y[t]-(k[t]*DELTA)-m[t]-((double)(1/interv1)*(vect[t]-g[t]));
            u[t]=(pow(c[t],(double)(1-SIGMA)))/(1-SIGMA);
            v[t]=(pow(m[t],(double)(1-SIGMA)))/(1-SIGMA);
            if (c[t]<0)
              ans+=-100000000*c[t]*c[t];
            else
              {
              rhocoefn*=(double)(1/(1+(interv1*(double)RHON)));
              ans+=rhocoefn*interv1*((u[t]+v[t])-d[t]);
              }
            }
          }
        k[t]=vect[t-1];
        m[t]=vect[t+20]*vect[t+41];
        y[t]=(double)A*pow(vect[t+20],(double)GAMMA)*pow(k[t],(double)ALPHA);
        d[t]=(.5*(double)D*vect[t+20]*vect[t+20])/pow(k[t],(double)BETA);
        c[t]=y[t]-(k[t]*DELTA)-m[t];
        u[t]=(pow(c[t],(double)(1-SIGMA)))/(1-SIGMA);
        v[t]=(pow(m[t],(double)(1-SIGMA)))/(1-SIGMA);
        if (c[t]<0)
          ans+=-1000000000*c[t]*c[t];
        else
          {
          ans+=(rhocoefn/((double)RHOS))*((u[t]+v[t])-d[t]);
          }
      return(ans);
}
```

Bibliography

Alpay, S. and İ. Sağlam, 1996, Transnational pollution, regulation and technological transfers, *manuscript*.

Ancot, J. P., A. J. Hughes Hallett and J.H.P. Paelinck, 1982, The determination of implicit preferences: Two possible approaches compared,*European Economic Review*, 18, 267-289.

Arifovic, J., 1994, Genetic algorithm learning and the cobweb model, *Journal of Economic Dynamics and Control*, 18, 3-28.

Arifovic, J., 1994, The behavior of the exchange rate in genetic algorithm and experimental economics, *Simon Fraser University working papers.*

Audretsch, D. B. and M. D. Feldman, 1996, R&D spillovers and the geography of innovation and production, *American Economic Review*, 86, 630-640.

Baç, M. 1996, Incomplete information and incentives to free ride on international environmental resources, *Journal of Environmental Economics and Management*, 30, 301-315.

Başar, T., and G.J. Oldser, 1982, *Dynamic noncooperative game theory*, Academic Press: New York.

Başar, T., 1986, A tutorial on dynamic and differential games, Başar, T. (ed.), *Dynamic Games and Applications in Economics*, Springer Verlag, Berlin, Heidelberg.

Başar, T. and V. D'Orey, 1990, Dynamic strategic monetary policies and coordination in interdependent economies, S.J. Turnovsky, ed., *International Macroeconomic Stabilization Policy*, Basil Blackwell Ltd, Oxford.

Baumol W. and W. Oates, 1988, *The Theory and Environment*, Cambridge Univ. Press, New York.

Bona, J. L. and M. S. Santos, 1994, On the role of computation in economic theory, *Universidad Carlos III working paper.*

Brandsma, A. S. and A. J. Hughes Hallett, 1984, Economic conflict and the solution of dynamic games, *European Economic Review*,26,13-32.

Chichilnisky, G. and G. Heal, 1986, *The Evolving International Economy*, Cambridge University Press, Cambridge.

Clark, C. W., 1976, *Mathematical Bioeconomics: The Optimal Management of Renewable Resources*, John Wiley, Canada.

Clemhout S. and H. Wan Jr., 1985, Dynamic common property resource and environmental problems, *Journal of Optimization, Theory and Applications*, 46, 471-481.

Conrad J. M. and C. W. Clark, 1987, *Natural Resource Economics: Notes and Problems*, Cambridge Univ. Press, New York.

Copeland, B. R. and M. S. Taylor, 1994, North-South trade and the environment, *Quarterly Journal of Economics*, 109, 755-787.

D'Arge, R. C. and K. C. Kogiku, 1972, Economic growth and the environment, *Review of Economic Studies*, 40, 61-78.

De Bruyne, G., 1979, Pareto optimality of noncooperative equilibrium in a time dependent multiperiod game, *European Economic Review*, 12, 243-260.

De Jong, K. A., 1993, Genetic algorithms are NOT function optimizers, L.D. Whitley (ed), *Foundations of Genetic Algorithms 2*, Morgan Kaufmann Publishers Inc., San Mateo, CA.

De Zeeuw, A. J. and F. Van der Ploeg, 1991, Difference games and policy evaluation: A conceptual framework, *Oxford Economic Papers*, 43, 612-636.

Diwan, I. and N. Shafik, 1992, Investment, technology and global environment toward international agreement in a world of disparities, P. Low, ed., *International Trade of Environment*, World Bank, Washington D.C..

Diwan, I. and N. Shafik, 1992, Investment, technology and global environment: toward international agreement in a world of disparities, P. Low (ed), *International Trade of Environment*, World Bank, Washington D.C..

Dockner, E. J. and N. V. Long, 1993, International pollution control: cooperative versus noncooperative strategies, *Journal of Environmental Economics and Management*, 24, 13-29.

Dreze, 1960, J., Quelques reflexions sereines sur l'adaptation de l'industrie belge au

Marche Commun, *Competes Rendus des Travaux de la Société Royale d'Economie Politique de Belgique*, 275.

Fudenberg, D. and D. K. Levine, 1988, Open-loop and closed-loop equilibria in dynamic games with many players, *Journal of Economic Theory*, 44, 1-8.

Galor, O., 1986, Global dynamic inefficiency in the absence of international policy coordination: A North-South case, *Journal of International Economics*, 21, 137-149.

Goldberg, D. E., 1989, *Genetic algorithm in search, optimization, and machine learning*, Addison-Wesley, Reading, Massachusetts.

Grefenstette, J. J., 1986, Optimization of control parameters for genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, 16, 122-128.

Grefenstette, J. J., 1990, A user's guide to GENESIS Version 5.0, *manuscript.*

Griliches, Z., 1979, Issues in assessing the contribution of R&D to productivity growth, *The Bell Journal of Economics*, 10, 92-116.

Grossman, G. M., and E. Helpman, 1990, Trade, innovation and growth, *Americal Economic Review Papers and Proceedings*, 80, 86-71.

Grossman, G. M., and E. Helpman, 1991, Trade, knowledge spillovers, and growth, *European Economic Review*, 35, 517-526.

Hämäläinen, R. P. and H. K. Ehtama (eds), 1990, *Dynamic Games in Economic Analysis*, Springer Verlag, Berlin, Heidelberg.

Ho, Y. C., 1970, Differential games, dynamic optimization, and generalized control theory, *Journal of Optimization Theory and Applications*, 6, 179-209.

Hoel, M., 1991, Global environmental problems: the effects of unilateral actions taken by one country, *Journal of Environmental Economics and Management*, 20, 55-70.

Holland, J.H., 1975, *Adaptation in Natural and Artificial Systems*, Ann Arbor, University of Michigan Press.

Holly, S., and A. Hughes-Hallett, 1989, *Optimal Control, Expectations and Uncertainty*, Cambridge University Press, Cambridge.

Hughes Hallett, A. J. and H. J. B. Rees, 1983, *Quantitative economic policies and interactive planning*, Cambridge University Press, Cambridge, New York.

Isaacs, R., 1965, *Differential Games*, John Wiley, New York.

Isaacs, R., 1969, Differential games: their scope, nature and future, *Journal of Optimization Theory and Applications*, 3, 283-295.

Jaffe, A. B., M. Tanjtenberg and R. Henderson, 1993, geographic localization of knowledge spillovers as evidence by patent citations, *Quarterly Journal of Economics*, 63, 577-598.

Judd, K. L., 1992, Projection methods for solving aggregate growth models *Journal of Economic Theory*, 58, 410-452.

Kahn., M. A. and P. Lin, 1985, Two way capital flows in a asymmetric world economy, *The Manchaster School*, 2, 219-224.

Karp, L. S. and A. F. McCalla, 1983, Dynamic games and international trade: An application to the world corn market, *American Journal of Agricutural Economics*, 65, 641-650.

Keeler, E., M. Spence, and R. Zeckhauser, 1971, The optimal control of pollution, *Journal of Economic Theory*, 4, 19-34.

Kendrick, D. and L. Taylor, 1970, Numerical solution on nonlinear planning models, *Econometrica*, 38, 453-467.

Kennedy, P. 1994, Equilibrium pollution taxes in open economies with imperfect competition, *Journal of Environmental Economics and Management*, 26, 49-63.

Koza, J. R., 1992, *Genetic programming*, The MIT Press, Cambridge, Massachusetts.

Krishnakumar, K. and Goldberg, D. E., 1992, Control system optimization using genetic algorithm, *Journal of Guidance, Control, and Dynamics*, 15, 735-738.

Kydland, F., 1975, Noncooperative and dominant player solutions in discrete dynamic games, *International Economic Review*, 16, 321-335.

Lancaster, K., 1973, The dynamic inefficiency of capitalism, *Journal of Political Economy*, 81, 1092-1109.

Levine, P. and Brochirer, A., 1994, Fiscal policy coordination and EMU: a dynamic game approach, *Journal of Economic Dynamics and Control*, 18, 699-729.

Lewis, A. W., 1954, Economic development with unlimited supplies of labor, *The Manchaster School*, 22, 139-191.

Lopez, R., 1994, The environment as a factor of production: the effects of economic growth and trade liberalization, *Journal of Environmental Economics and Management*, 26, 163-184.

Low, P. and R. Shafik, 1992, Trade and Pollution, P. Low (ed), *International Trade of Environment*, World Bank, Washington D.C..

Ludema R. D. and I. Wooton, 1994, Croos-border externalities and trade liberalization: the strategic control of pollution, *Canadian Journal of Economics*, 27, 49-63.

Marimon, R., E. McGrattan and T. Sargent, 1990, Money as a medium of exchange in an economy with artificially intelligent agents, *Journal of Economic Dynamics and Control*, 14, 329-373.

Marks, R. E., 1992, Breeding hybrid strategies:optimal behaviour for oligopolists, *Journal of Evolutionary Economics*, 2, 17-38.

Martin, E. W., R. H. Patrick and B. Tolwinski, 1993, A dynamic game of transboundary pollutant with asymmetric players, *Journal of Environmental Economics and Management*, 24, 1-12.

Markusen, J. R., 1975, Cooperative control of intenational pollution and common property resources, *Quartely Journal of Economics*, 90, 618-632.

Matsuyama, K., 1992, Agricultural productivity, comparative advantage and economic growth, *Journal of Economic Theory*, 58, 317-334.

Mercenier, J., and P. Michel, 1994, Discrete-time finite horizon approximation of infinite horizon optimization problems with steady-state invariance, *Econometrica*, 62, 635-656.

Michalewicz, Z., and C. Janikow, 1991, Handling constraints in Genetic Algorithms, Belew, R. and Booker, L. (eds.) *Proceedings of the fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Los Altos, CA.

Michalewicz, Z., 1992, *Generic Algorithm + Data Structures = Evolution Program*, Springer Verlag, Berlin Heidelberg, New York.

Miller, M. and M. Salmon, 1983, Policy coordination and dynamic games, W.H. Buiter and R.C. Marston eds., *International Economic Policy Coordination*, Cambridge University Press, Cambridge.

Milliman, S. R. and R. Prince, 1989, Firm incentives to promote technological change in pollution control, *Journal of Environmental Economics and Management*, 17, 247-265.

Özyıldırım, S., 1996, Three-country trade relations: a discrete dynamic game approach, *Computers and Mathematics with Applications*, 32, 43-56.

Özyıldırım, S., 1997, Computing open-loop noncooperative solution in discrete dynamic games, forthcoming, *Journal of Evolutionary Economics*,

Pau, L. F., 1975a, Differential games and a Nash equilibrium searching algorithm, *SIAM J Control*, 13, 835-842.

Pau, L. F., 1975b, A differential game among sectors in a macroeconomy, *Automatica*, 11, 473-485.

Pindyck, R. S., 1977, Optimal economic stabilization under decentralized control and conflicting objectives, *IEEE Transactions and Automatic Control*, AC-22, 517-530.

Plourde, C. G., 1972, A model of waste accumulation and disposal, *Canadian Journal of Economics*, 5, 119-125.

Robertson, G., 1987, Parallel implementation of genetic algorithms in a classifier systems, L. Davis ed., *Genetic Algorithms and Simulated Annealing*, Pittman,London.

Sargent, T. J., 1987, *Dynamic Macroeconomic Theory*, Harvard University Press, Cambridge, Massachusetts.

Sargent, T. J., 1993, *Bounded Ratinality in Macroeconomics*, Oxford University Press, Oxford.

Shiller, R. J., 1977, Rational expectations and dynamic structure of macroeconomic models with rational expectations, *Econometrica*, 45, 6-15.

Stewart, F., 1984, Recent theories of international trade: some implications for the

South, H. Kierzkowski, ed., *Monopolistic Competition and International Trade*, Oxford University Press, Oxford.

Starr, A. W. and Y. C. Ho, 1969, Nonzero-sum Differential Games, *Journal of Optimization Theory and Application*, 3, 184-206.

Starr, A. W. and Y. C. Ho, 1969, Further Properties of nonzero sum differential games, *Journal of Optimization Theory and Applications*, 3, 207-219.

Theil, H., 1964, *Optimal decision rules for government and industry*, North-Holland, Amsterdam.

Ulph, A. and D. Ulph, 1994, The optimal time of a carbon tax, *Oxford Economic Papers*, 46, 857-868.

Van der Ploeg, F., 1982, Government policy, real wage resistance and the resolution of conflict, *European Economic Review*, 19, 181-212.

Whalley, J., 1991, The interface betweeen environmental and trade policies, *Economic Journal*, 101, 180-189.

Vita

SÜHEYLA ÖZYILDIRIM

Home: Büklüm Sokak 40/20, 06660, Kavaklıdere, Ankara, Turkey • Phone:+312-428 28 77 • E-mail: suheyla@bilkent.edu.tr

Office: Dept.of Economics, Bilkent University, Ankara, Turkey • Phone: (90-312) 266 40 40-1895 • Fax: (90-312) 266 49 60

EDUCATION

BILKENT UNIVERSITY, ANKARA
Ph.D., Economics, expected April 1997

JOHNS HOPKINS UNIVERSITY, BALTIMORE, MD
M.A. in Economics, 1993

BILKENT UNIVERSITY, ANKARA
M.A. in Economics, 1991

BILKENT UNIVERSITY, ANKARA
M.B.A. in Finance, 1989

MIDDLE EAST TECHNICAL UNIVERSITY, ANKARA
B.A. in Management, 1986

RESEARCH PAPERS

A Note on Price Volume Dynamics in an Emerging Stock Markets, jointly with Kürşat Aydoğan and Erdem Başçı, **Journal of Banking and Finance**, March 1996.

Optimal Monetary and Fiscal Policies:A Survey, **METU Studies in Development**, March 1996.

Three-Country Trade Relations: A Discrete Dynamic Game Approach, **Computers and Mathematics with Applications**,June 1996.

Computing Open-Loop Noncooperative Solution in Discrete Dynamic Games, **Journal of Evolutionary Economics**, February, 1997.

A Genetic Game, Trade, Growth and Externality, jointly with Nedim M. Alemdar, submitted to *Journal of Economics Dynamics and Control*, 1997.

Price Volume Dynamics in a Stock Market with Heterogeneous Expectations and Restricted Shortsales, Discussion Paper, Bilkent University, 1993.

INTERESTS

- TEACHING INTERESTS:
  Macroeconomics, International Trade.
- RESEARCH INTERESTS:
  Dynamic Games, Optimization, Genetic Algorithms

DISSERTATION

- A GENETIC GAME OF TRADE, GROWTH AND EXTERNALITIES
  Developed a numeric algorithm to solve complex games between conflicting agents, countries or regions

TEACHING EXPERIENCE

- BILKENT UNIVERSITY, *Instructor (Fall 1992-Spring 1995)*
  Taught "Topics in Macroeconomics", ECON 563-564

PROFESSIONAL EXPERIENCE

- BILKENT UNIVERSITY, Dept. of Economics, Ankara, Turkey
  Research Assistant, Fall 1990 through 1996
- STATE INSTITUTE OF STATISTICS, Dept. of E.S.E., Ankara, Turkey
- BILKENT UNIVERSITY, Institute of Economic Research and Analysis
  Research Assistant, Summer 1988 through Summer 1989

PARTICIPATIONS

- Workshop on Learning and the emergence of markets and other organizations, Florence 1995
- Presented *Genetic Algorithm, Terms of Trade and Growth in an Interdependent World*, Eastern Economic Association 22nd Annual Convention, Boston 1996
- Presented *Computing Open-Loop Noncooperative Solution in Discrete Dynamic Games*, European Economic Association 11th Annual Conference, Istanbul 1996

PERSONAL

Born in 1965 in Ankara, Turkey. Turkish citizen. Fluent in Turkish (mother tongue) and English. Intermediate knowledge of French.