

EDNANDO BATISTA VIEIRA

**EMERGÊNCIA DE COOPERAÇÃO EM SISTEMAS SÓCIO-ECOLÓGICOS:
UM MODELO BASEADO EM AGENTES**

Dissertação apresentada à
Universidade Federal de Viçosa,
como parte das exigências do
Programa de Pós-Graduação em
Economia, para obtenção do título de
Magister Scientiae.

VIÇOSA
MINAS GERAIS – BRASIL
2011

EDNANDO BATISTA VIEIRA

**EMERGÊNCIA DE COOPERAÇÃO EM SISTEMAS SÓCIO-ECOLÓGICOS:
UM MODELO BASEADO EM AGENTES**

Dissertação apresentada à
Universidade Federal de Viçosa,
como parte das exigências do
Programa de Pós-Graduação em
Economia, para obtenção do título de
Magister Scientiae.

Aprovada: 14 de Março de 2010.

Prof. Jader Fernandes Cirino

Prof. José Norberto Muniz

Prof. Newton Paulo Bueno
(Orientador)

Dedico este trabalho a professora
Luciene Garcia de Almeida.

“Não confunda derrotas com fracasso nem vitórias com sucesso. Na vida de um campeão sempre haverá algumas derrotas, assim como na vida de um perdedor sempre haverá vitórias. A diferença é que, enquanto os campeões crescem nas derrotas, os perdedores se acomodam nas vitórias.”

Roberto Shinyashiki

AGRADECIMENTOS

Agradeço inicialmente a Deus, por sempre estar comigo em minha jornada e me dar condições de seguir em frente, vencendo barreiras e crescendo como profissional e como pessoa. A toda a minha família que sempre me apoiou, em especial à minha mãe que sempre acreditou em mim, mesmo nos momentos em que tudo parecia um sonho distante.

Aos institutos financiadores, em especial a CAPES que por meio do REUNI, me forneceu auxílio financeiro para custear meus estudos no mestrado. À Universidade Federal de Viçosa que me acolheu e me fez passar bons e inesquecíveis momentos. Aos professores que me ensinaram tanto, o professor Adriano, a professora Silvia, o professor Jader e a professora Elaine que coordenou as minhas atividades como bolsista do REUNI no meu primeiro ano de curso. E um agradecimento muito especial ao professor Newton que coordenou minhas atividades como bolsista REUNI durante o segundo ano de meu curso além de ser um orientador excepcional, me dando total apoio e insights para o desenvolvimento deste trabalho.

Agradeço aos amigos que conquistei em Viçosa, ao Paulo Nei, Daiana, Mirian, Marcelo, Jefferson, Aline, Juliana, Bruno e sobretudo ao Rodrigo, estes que me aturaram e me ajudaram em todos os momentos.

Não posso esquecer-me das minhas companheiras de projeto, Raissa, Lorena e Amanda que ralaram junto comigo para desenvolver um bom trabalho. E muito menos dos colegas de república Manuela, Rafel, Carlos, Denis, Daniel, Magnus, Juan Filipe, Antonio Carlos, Vitor e Luca.

Enfim, agradeço a todos, que direta ou indiretamente me ajudaram a alcançar meus objetivos.

A todos, meu mais sincero obrigado.

SUMÁRIO

LISTA DE TABELAS	vii
LISTA DE FIGURAS	viii
RESUMO	ix
ABSTRACT	x
1. INTRODUÇÃO	1
1.1 PROBLEMA E SUA IMPORTÂNCIA.....	1
1.2 OBJETIVOS DO TRABALHO	6
2. REFERENCIAL TEÓRICO	7
2.1 TEORIA DA AÇÃO COLETIVA.....	7
2.2 CAPITAL SOCIAL.....	13
2.3 MODELO BASEADO EM AGENTE.....	15
2.3.1 Ambiente computacional	17
3. METODOLOGIA	20
3.1 O MODELO BASE	20
3.2 CARACTERIZAÇÃO DO MODELO – PRIMEIRA PARTE	21
3.3 SEGUNDA PARTE - MODELO DE IRRIGAÇÃO	25
4. RESULTADOS E DISCUSÕES	28
4.1 RESULTADOS PARA A PRIMEIRA PARTE.....	28
4.1.1 Efeito da heterogeneidade	28
4.1.2 Efeito do tamanho	33
4.2 RESULTADO PARA A SEGUNDA PARTE - MODELO DE IRRIGAÇÃO - E COMPARAÇÃO ENTRE O MODELO VIRTUAL E O CASO REAL.....	35
5. CONCLUSÃO	40
6. REFERENCIAS BIBLIOGRÁFICAS	42
7. APÊNDICE 1	45
7.1 MODELO DE COOPERAÇÃO.....	45

7.1.1	PROGRAMAÇÃO DO MODELO DE COOPERAÇÃO.....	45
7.2	DETALHANDO OS PROCEDIMENTOS DA PROGRAMAÇÃO.....	47
8.	APÊNDICE 2.....	52
8.1	MODELO DE IRRIGAÇÃO	52
8.1.1	Criação da população 1.....	57
8.1.2	Criação da população 2.....	59
8.2	CRIANDO O <i>TO SETUP</i> E O <i>TO GO</i>	65

LISTA DE TABELAS

Tabela 1 – Estatísticas dos testes de cooperação com $T = 8$, $T = 10$ e $T = 10,2$	31
Tabela 2 – Estatísticas dos testes de cooperação para diferentes quantidades iniciais da população	33

LISTA DE FIGURAS

Figura 1 – Estrutura de vizinhança de autômatos celulares	18
Figura 2 – Ambiente virtual	26
Figura 3 – Evolução dos níveis de cooperação para 70% das repetições para $T = 6$. 30	
Figura 4 – Evolução dos níveis de cooperação para 30% das repetições com $T = 6$. 30	
Figura 5 – Evolução das quantidades de agentes cooperativos com o aumento da heterogeneidade	32
Figura 6 – Evolução dos níveis de cooperação para população inicial de 100, 200 e 400 indivíduos	34
Figura 7 – Mapa da Região do Vale do Rio Doce, onde estão localizadas as duas comunidades (Primeiro de Julho e Cachoeirinha).....	35
Figura 8 - Visualização das rodadas de interação no modelo de Irrigação	38
Figura A2 – Ambiente virtual	56

RESUMO

VIEIRA, Ednando Batista, M. Sc., Universidade Federal de Viçosa, março de 2011. **Emergência de cooperação em sistemas sócio-ecológicos: um modelo baseado em agentes.** Orientador: Newton Paulo Bueno. Co-orientadores: Elaine Aparecida Fernandes e Adriano Provezano Gomes.

No campo da Nova Economia Institucional (NEI), os estudos de como os indivíduos se organizam para criar mecanismos capazes de proporcionar a cooperação em pequenos grupos vem ganhando grande atenção recentemente. Nesse sentido, algumas variáveis como heterogeneidade e tamanho do grupo são tidas como importantes para a cooperação, contudo existe uma indefinição dos efeitos que estas causam, e se elas são fortes o suficiente para levar uma comunidade ao colapso. Para buscar responder essa questão, este trabalho cria um modelo baseado em agentes, com o intuito de testar quais os efeitos que essas duas variáveis (tamanho e heterogeneidade) têm sobre a cooperação em assentamentos rurais, e comparar os resultados modelados com os resultados reais de uma pesquisa de campo realizada em 2005 nas comunidades Primeiro de Julho e Cachoeirinha, localizadas na região do Vale do Rio Doce, no norte do Estado de Minas Gerais. O modelo baseado em agentes indica que aumentos no tamanho do grupo levam ao aumento da não cooperação, como indicado pela Teoria da Ação Coletiva, mas não é suficiente para levar a extinção do grupo; já a heterogeneidade leva tanto a queda dos níveis de cooperação quanto à extinção da comunidade, uma vez que quanto mais diferentes e divergentes forem os interesses das pessoas, mais difícil será para eles se organizarem a fim de criarem instituições capazes de proporcionar o desenvolvimento dessa comunidade.

ABSTRACT

VIEIRA, Ednando Batista, M. Sc., Universidade Federal de Viçosa, March, 2011. **Emergence of cooperation in Social-Ecological System: an agent-based model.** Adviser: Newton Paulo Bueno. Co-Advisers: Elaine Aparecida Fernandes and Adriano Provezano Gomes.

In the field of New Institutional Economics (NIE), studies of how individuals are organized to create mechanisms to provide cooperation in small groups have gained much attention recently. Therefore, some variables such as heterogeneity and group size are considered important for cooperation, yet there is a blurring of the effects that they cause, and if they are strong enough to lead a community to collapse. To seek to answer this question, this work creates an agent-based model, with the aim of testing the effects that these two variables (size and heterogeneity) have on cooperation in rural settlements, and compare the modeled results with actual results of a field study conducted in 2005 in communities Primeiro de Julho and Cachoeirinha, located in the Vale do Rio Doce, in the north of Minas Gerais. The agent-based model indicated that increases in group size leads to increased defection, as indicated by the Theory of Collective Action, but not enough to lead to extinction of the group. Yet, the heterogeneity leads to decreased levels of cooperation and the extinction of the community, since the more different and divergent are the interests of people, is harder for them to organize themselves in order to create institutions capable of providing the development of this community.

1. INTRODUÇÃO

1.1 PROBLEMA E SUA IMPORTÂNCIA

Na literatura sobre Ação Coletiva há questionamentos de quais tipos de modelos de comportamento humano são eficientes para explicar observações reais obtidas em estudos de campo, uma vez que o comportamento de participantes em experimentos de ação coletiva não é consistente com as previsões de modelos de escolha racional, onde indivíduos maximizam seus *payoffs*.

Poteete, Janssen e Ostrom (2010) relatam que estudos baseados em pesquisas de campo, experimentos em campo e em laboratório, teoria dos jogos e modelos baseados em agentes tem demonstrado que é possível que indivíduos ajam coletivamente gerindo recursos naturais em uma base sustentável. Sobretudo, mostram que desenvolvimentos recentes apontam modelos baseados em agentes como uma ferramenta eficaz para testar modelos alternativos que replicam os padrões observados de participantes, de uma base comum de recursos, em experimentos de laboratório.

Estes autores argumentam que modelos baseados em agentes é um tipo de modelagem formal que possibilita identificar as condições sob as quais a cooperação pode evoluir. O objetivo principal deste tipo de modelo é identificar o conjunto de mecanismos em um nível micro em que padrões mais amplos evoluem. Estes modelos podem ser usados para comparar explicações alternativas que são definidas em estudos de campo e experimentais.

Contudo esse tipo de modelagem possui grade liberdade para a especificação de agentes e suas interações, o que é uma vantagem que possibilita a criação de inúmeras representações da realidade, mas que também pode ser um problema que descaracteriza a validade dos resultados obtidos. Para não incorrer em tais erros são necessárias hipóteses consistentes e uma fundamentação teórica para que não se tornar um modelo *ad hoc*. Assim sendo, como podem ser utilizados modelos baseados em agentes de forma eficaz e correta, capazes de explicar o comportamento humano, dado o leque de possibilidades de criação que a metodologia disponibiliza e a complexidade existente em sistemas sócio-ecológicos?

Entende-se por sistemas Sócio-ecológicos como sendo ambientes onde indivíduos inseridos em comunidades interagem com outros membros e também com o meio natural, ou mais especificamente, essas pessoas dividem uma base comum de recursos. Ostrom (2005) define base comum de recursos como sendo recursos, naturais ou criados pelo homem (como lagos, oceano, sistemas de irrigação, locais de pesca, florestas, internet, estratosfera), onde é difícil excluir ou limitar seu uso, além do fato de que o consumo de uma unidade de tal recurso por uma pessoa exclui seu uso por outra pessoa.

Recursos comuns envolvem sistemas complexos e não sistemas simples. Complexos não pelo fato de existir uma grande quantidade de variáveis, mas sim pela presença de complexidade sistêmica, que diz que as interações entre os elementos existentes no sistema estão ligados em ciclos de retroalimentação de forma positiva ou negativa, com efeitos distantes no tempo e espaço. Assim a ação de um indivíduo afeta outras pessoas além do ambiente em que vive, tal ação gera reações que voltarão à afeta-lo de forma imediata e/ou após algum período de tempo.

A emergência de resultados não intencionais, ocorridos através da interação entre indivíduos maximizadores agindo racionalmente é uma característica de sistemas complexos. Estes sistemas apresentam, de acordo com Bueno (2009a), complexas redes de relacionamento que tornam difícil entender sua dinâmica sem a construção de modelos baseados em conhecimentos especializados. A propriedade de

emergência em sistemas complexos, segundo Bueno (2010), manifesta-se porque as decisões individuais estão interligadas por ciclos de retroalimentação.

Na construção de tais modelos é necessário coletar informações dos agentes dentro da comunidade, como eles agem, as instituições que eles criam, a forma com que vivem. Com relação ao comportamento dos agentes na criação de regras de uma comunidade, para Poteete, Janssen e Ostrom (2010), os usuários de um sistema de recursos comparam os benefícios líquidos da produção ao se utilizar as regras antigas com o resultado esperado com o novo conjunto de regras. E cada indivíduo verifica se os incentivos para a mudança no padrão são positivos ou não. Caso o resultado esperado seja negativo, o produtor não terá incentivos a adotar as novas regras e não cooperará.

Os usuários destes recursos, caso os benefícios sejam positivos, avaliam três tipos de custos: primeiro, os custos iniciais de tempo e esforço despendidos na elaboração e acordo de novas regras; os custos de curto prazo da implementação de tais regras; e, custos de longo prazo do monitoramento e manutenção de um sistema auto-regulado ao longo do tempo. Sendo o somatório dos custos esperado superior ao incentivo a mudança para cada usuário, este não se despende a investir tempo e recursos para a criação de novas instituições.

Porém, cada indivíduo tem uma percepção diferente com relação às expectativas de custos e benefícios. Para que as expectativas se aproximem da realidade é necessário um grande conjunto de informações. Se alguém pudesse obter meios válidos e confiáveis de perceber os custos e benefícios da ação coletiva em que está envolvido, estes dados seriam a informação central necessária para prever quando a ação coletiva ocorreria para mudar as regras. Assim obter informação sobre os atributos do sistema de recursos e seus usuários é, de acordo com Poteete, Janssen e Ostrom (2010), um passo essencial para aumentar a capacidade de diagnosticar por que alguns grupos superam os desafios de ação coletiva e outros não.

Estes autores também argumentam que em muitos ambientes onde os usuários utilizam uma base comum de recursos, é irrealístico assumir que todos os indivíduos

tomam suas decisões com informações completas da estrutura onde estão inseridos. Estudiosos, como Elinor Ostrom, que pesquisam os fatores envolvidos no surgimento da cooperação e resolução de dilemas de ação coletiva, conhecem as dificuldades envolvidas com as informações utilizadas pelos indivíduos e quais as principais variáveis que afetam a cooperação, contudo não explicam de forma inteiramente convincente como a cooperação surge.

É difícil de acreditar que indivíduos busquem a todo tempo soluções complexas de custos e benefícios, além de possuírem informações de todas as ações potenciais que poderiam ser tomadas, todos os produtos que poderiam ser obtidos, e todas as estratégias que os outros membros poderiam tomar. Contudo, é plausível assumir que indivíduos busquem sempre maximizar sua utilidade e tenham certo nível de informação sobre o ambiente em que estão inseridos. Mas essa maximização se realiza em um nível sub-ótimo, dada a restrição informacional. É mais plausível assumir que a tomada de decisões e a criação de instituições que resultam na solução de dilemas de ação coletiva e levam a emergência da cooperação se dê de uma forma bem mais simples do que a proposta pela teoria convencional, através de indivíduos agindo localmente, tentando maximizar sua utilidade com base no seu conjunto de informações, verificando apenas seus interesses. Se os incentivos a agirem de forma egoísta forem satisfatórios, o indivíduo assim agirá, e quanto mais heterogêneo e maior for o grupo, maior será essa tentação e menor o nível de cooperação.

No estado atual do conhecimento sobre instituições permanece uma indefinição em relação ao efeito de algumas variáveis importantes sobre a cooperação entre indivíduos de um grupo. Em Ostrom (2005) condições microsituacionais como o tamanho do grupo e a heterogeneidade existente entre seus membros, e macrosituacionais como pressões de mercado, direitos de propriedade e políticas governamentais são algumas das principais peças deste quebra-cabeça teórico. O desafio então é entender como as combinações ao longo do tempo destas variáveis, de estratégias, características e fatores espaciais criam situações onde os usuários de recursos se auto-organizam (e a cooperação é sustentada) e em outras não.

É fato que, para alcançar a cooperação os indivíduos devem solucionar dilemas sociais de ordem superior. Um dilema social, de acordo com Bates (1995) surge quando escolhas feitas por indivíduos racionais produzem resultados que são socialmente irracionais. Um dilema social de primeira ordem ocorre quando indivíduos não são capazes de desenvolver instituições para superar os custos de transação. Dilemas sociais de segunda e terceira ordens são relacionados, respectivamente, à falta de instituições para prevenir comportamentos do tipo *free-riding* e *rent-seeking* e para assegurar eficientemente o monitoramento e aplicação das regras sociais - tais dilemas de ordem superior geralmente ocorrem em ambientes de dilema do prisioneiro.

Pela abordagem de North (1991), as instituições surgem para reduzir a incerteza e os custos nas transações, ou seja, elas surgem de ações intencionais dos indivíduos para superarem dilemas sociais de primeira ordem, e assim problemas de custos de transação. Instituições são restrições feitas pelos homens sobre as interações políticas, econômicas e sociais, sendo elas informais: sanções, tabus, costumes, tradições, regras de conduta; ou formais: constituições, leis e direito de propriedade.

Na linha trabalhada por Elinor Ostrom, os indivíduos que utilizam uma base comum de recursos se organizam para delimitar regras e restrições, visando à sustentabilidade da comunidade. Nessa corrente, as instituições criadas pela comunidade destinam-se especialmente a resolver problemas de ação coletiva, como comportamentos *free-riding* e *rent-seeking*. Em suas pesquisas de campo a autora identificou oito padrões de comportamento existentes em grupos que conseguiam solucionar seus problemas de ação coletiva.

Ostrom (1998) formalizou um modelo baseado em agentes para explicar o comportamento observado em experimentos de pescadores e catadores de caranguejo na ilha de Providence na Colômbia.

A literatura internacional tem buscado o uso de modelos baseados em agentes para responder a questões de porque indivíduos cooperam em dadas situações e em outras não. Por exemplo, Peter Deadman foi o pioneiro no uso dessa metodologia para explorar dados experimentais. Jager e Janssen (2002) elaborou um modelo baseado

em agentes em estrutura meta-teórica de teorias psicológicas e testam seu modelo com experimentos de base comum de recursos. Janssen e Ahn (2006) fazem uma análise quantitativa rigorosa que compara a performance empírica dos modelos de tomada de decisão para explicar os resultados de experimentos em ambientes de bens públicos sem comunicação. Contudo, existe uma carência na literatura nacional no uso deste tipo de abordagem metodológica. Assim, este trabalho pretende contribuir para o desenvolvimento nacional de estudos de dilemas de ação coletiva com o uso de modelos baseados em agentes.

1.2 OBJETIVOS DO TRABALHO

Este trabalho tem como objetivo geral criar um Modelo Baseado no Agente consistente, capaz de identificar como algumas variáveis como heterogeneidade e tamanho do grupo, que são algumas das variáveis importantes citadas na literatura sobre ação coletiva, influenciam na emergência de cooperação, e assim, gerar informações para o desenho de instituições capazes de criar cooperação em sistemas sócio-ecológicos.

Com o intuito de cumprir este objetivo, pretende-se de forma específica:

- Elaborar um modelo computacional com agentes representativos capazes de obedecer a regras de comportamento e de interagirem mutuamente e com seu ambiente;
- Testar o efeito da heterogeneidade entre os indivíduos e do tamanho do grupo no surgimento da cooperação e auto-sustentabilidade da comunidade;
- Comparar os resultados virtuais com situações reais, utilizando para tal comparação duas comunidades vizinhas, porém com padrões comportamentais distintos, localizadas na região de Governador Valadares na região Norte do estado de Minas Gerais: a comunidade Primeiro de Julho e a comunidade Cachoeirinha.

2. REFERENCIAL TEÓRICO

A Nova Economia Institucional (NEI), de acordo com Williamson (2000), possui duas pressuposições básicas que são: i) instituições têm importância para explicar o grau de eficiência com que a economia opera; e ii) os determinantes de instituições são passíveis de estudo e suscetíveis a análises por ferramentas da teoria econômica. A segunda proposição é a que distingue a NEI das antigas correntes que estudavam instituições.

Instituições, pela definição de North (1991) são restrições que os homens impõem às suas ações, restringindo sua liberdade com movimentos estratégicos para que seja possível a cooperação entre eles.

Williamson (2000) coloca as instituições em quatro níveis de análise: o primeiro nível referente aos costumes, religiões, normas; o segundo nível referente ao ambiente institucional, regras formais, direito de propriedade; o terceiro nível de análise corresponde a estruturas de governança e a economia dos custos de transação; e o quarto nível refere-se ao ambiente de estudos neoclássico - o ambiente da firma. Porém, não existe espaço na definição de Williamson para a Teoria da Ação Coletiva - pelos princípios metodológicos e suposições feitas por essa linha, ela estaria localizada em um nível intermediário entre os níveis dois e três, e assim, a Teoria da Ação Coletiva é parte integrante da Nova Economia Institucional.

2.1 TEORIA DA AÇÃO COLETIVA

A obra de Mancur Olson “*The Logic of Collective Action*” de 1965 deu início aos estudos no campo de ação coletiva. Nesta obra o autor inicia a discussão do por que indivíduos racionais não agem de forma cooperativa, mesmo esta ação sendo benéfica para o grupo.

Em Olson (1982) é discutido o paradoxo que acontece no comportamento dos grupos. O paradoxo é que mesmo que exista um grupo de pessoas com interesses em comum e que queiram buscar esse interesse, quando agem racionalmente para isto, acabam não atingindo aquele interesse em comum do grupo. Isso acontece devido ao

comportamento *free-rider* que surgirá, pois, como os ganhos do grupo irão para todos mesmo que um agente não se esforce, então, as pessoas terão incentivos para não se esforçarem e deixarão que os outros façam. Isso implica que não haverá nenhuma ação em conjunto, e o interesse não será atingido. E esse efeito se agrava com o tamanho do grupo, quanto maior ele for, mais difícil será a provisão deste bem público.

Esse problema ocorre também nas associações como sindicatos, cartéis, *lobbies*, que se assemelham com os serviços básicos do Estado, em que os direitos conseguidos pelos participantes vão para toda a classe de trabalhadores e cidadãos, por exemplo, com ou sem a contribuição de todos. No entanto, surgem os “incentivos seletivos”, que criam uma matriz de incentivos para o indivíduo em contribuir ou não para a provisão do bem público. Estes incentivos podem ser positivos ou negativos, e geram uma situação na qual o agente *free-rider* é punido caso não contribua. Alguns exemplos são o pagamento de multas e taxas caso uma pessoa seja pega sonegando imposto e no caso de sindicatos, quando estão em greve, trabalhadores que tentam furar a greve são altamente punidos pelos outros.

Ainda de acordo com Olson (1982), existe uma situação na qual ocorre a ação coletiva mesmo sem a existência de incentivos seletivos, no entanto, acontece somente quando poucos irão se beneficiar com a ação. Quando existem poucos participantes do grupo, pode ocorrer uma barganha entre eles para que se beneficiem mais com a ação coletiva, agindo estrategicamente entre si. Como em qualquer barganha, uma das partes pode ter incentivos para se apropriar de uma parte maior dos ganhos e enfraquecer a ação coletiva. Mas a conclusão principal é que quanto maior o número de indivíduos que se beneficiarão da ação coletiva, menores serão os benefícios individuais que cada um terá da ação. Assim, sem os incentivos seletivos, os incentivos para ações coletivas diminuem com o aumento do tamanho do grupo.

Desta forma, mostra-se que grupos que se deparam com incentivos seletivos têm mais probabilidade de contribuir para a provisão de bens coletivos que os demais.

Já com relação ao estudo das regras efetivamente utilizadas na regulamentação e na utilização de recursos comuns em muitos países do mundo, leva a conclusão de

que a situação é mais complexa, recursos comuns envolvem sistemas complexos e não sistemas simples. Assim, é necessária uma política coerente, aninhada a um conjunto de conceitos que possam ser usados na análise, pesquisa ou assessoria política, de forma cumulativa.

Por meio de testes, principalmente sobre o desempenho de sistemas auto-organizados que são consistentes com os “princípios” de decisão, verifica-se que existem formas de governança que aumentam as possibilidades de adaptação e aprendizagem de uma sociedade num mundo em mudança e incerto com contínuos avanços no conhecimento e tecnologias.

Ostrom (1990) identificou oito desses princípios de design que caracterizam as robustas instituições de recursos comuns que perduram no tempo. Estes princípios não são obrigatórios, eles apenas descrevem as grandes semelhanças estruturais entre os sistemas auto-organizados que foram capazes de se adaptar e aprender de modo a serem suficientemente robustos perante os distúrbios de muitos fatores sociais, econômicos e ecológicos que ocorreram ao longo do tempo.

a) Fronteiras bem definidas: se um grupo de usuários pode determinar seus próprios membros, incluindo aqueles que concordam em utilizar o recurso de acordo com suas regras acordadas e excluir aqueles que não concordam com essas regras, o grupo deu um primeiro passo importante para limitar o acesso de invasores e promover o desenvolvimento e o sentimento de maior confiança e reciprocidade. Em alguns casos, o possível comportamento caronístico de usuários infiltrados no meio social pode levar o regime ao colapso. Assim, se o recurso em si e os usuários dos recursos estão claramente definidos, os membros da sociedade são capazes de defender os recursos de invasores.

b) Equivalência entre custos e benefícios: as regras de uso devem atribuir benefícios proporcionais aos insumos que são necessários. Se um grupo de usuários está indo para a colheita de um recurso em longo prazo, eles devem definir regras relacionadas ao quanto, quando e quantos produtos diferentes serão colhidos. Quando as regras relativas à distribuição dos benefícios forem largamente consistentes com a

distribuição dos custos, os participantes estarão mais dispostos a arregaçar as mangas para manter um recurso bem conservado de forma sustentável.

c) Regime descentralizado de escolha coletiva: a escolha das regras deve ser democrática. Um regime tende a ser mais eficiente onde a maioria dos indivíduos envolvidos é autorizada a participar na tomada de decisão e modificar as suas regras. Sociedades que utilizam este princípio são mais capazes de adaptar as regras para as circunstâncias locais e elaborar regras que são consideradas justas pelos participantes.

d) Monitoramento: as regras devem ser aplicadas de maneira eficaz com o intuito de assegurar uma estrutura de governança robusta. A questão de como as regras serão realmente executadas é frequentemente ignorada quando mudanças institucionais são analisadas e uma reforma é proposta. A aplicação da regra é necessária para a criação de um recurso sustentável ao longo do tempo, e o monitoramento deve ser realizado pelos próprios usuários do recurso, ou seja, efetivamente os mais interessados pela manutenção e pela qualidade futura do bem. São eles que possuem o conhecimento e a vivência necessária para lidar com possíveis desvios de conduta dos membros da sociedade. Sendo que, mesmo técnicos de institutos governamentais, que imponham regras gerais “de cima para baixo” não se adequarão bem ao regime, e o dilema social provavelmente não será superado.

e) Sanções gradativas: em muitos sistemas auto-organizados, a primeira sanção imposta por um monitor local é tão baixa que não têm nenhum impacto sobre a razão custo-benefício de quebrar as regras locais. A grande ameaça ao sistema se dá quando o participante quebra as regras repetidamente. Assim, as penalidades devem ser aumentadas até que culminem na expulsão do indivíduo do meio local.

f) Mecanismos de resolução de conflitos: O funcionamento dos princípios acima é amparado por este princípio, que aponta para a importância do acesso rápido e de baixo custo à arenas locais para a resolução de conflitos entre usuários ou entre usuários e Oficiais. Regras, ao contrário das limitações físicas, têm de ser compreendidas, a fim de serem eficazes. Há sempre situações em que os participantes possam interpretar a regra de maneiras diferentes. Através da simples criação de locais onde se possam expor e discutir os conflitos imediatamente e verificar as resoluções

que norteiam a comunidade, o número de conflitos que reduzem a confiança pode se reduzir significativamente.

g) Reconhecimento de direitos mínimos para se organizar: a capacidade dos usuários locais para desenvolver um regime cada vez mais eficaz ao longo do tempo é afetada pelo fato de terem pelo menos um mínimo de reconhecimento do direito de se organizar por um governo nacional ou local. Os indivíduos devem possuir o direito de dizer não! Caso contrário, qualquer participante temporariamente descontente que for contra uma imposição na mudança das regras poderia ir para sociedades externas, em busca de regras mais adequadas a custos relativamente baixos, a ameaçar o regime em si.

h) Organizações aninhadas: Quando os recursos do fundo comum são grandes, um princípio tende a caracterizar os sistemas robustos - a presença de atividades de governança organizadas em múltiplas camadas de níveis aninhados. Neste cenário, surge a possibilidade de usuários se agruparem de forma a aproveitar o tamanho da estrutura com o intuito de aumentar o “produto”, no caso, buscar maximizar o uso do recurso.

Ao analisar as ameaças as quais os sistemas de recursos de uso comum estão sujeitos, Ostrom (2005) ressalta que até mesmo regimes onde as características listadas acima estão presentes podem entrar em decadência por sofrer com determinadas influências exógenas.

Além de citar o problema da imigração, que desperta a desconfiança das pessoas já instaladas e o da emigração que deixa a comunidade carente de usuários leais às leis locais, Ostrom ressalta cinco principais problemas comumente encontrados que devem ser contornados:

a) Rápidas mudanças exógenas.

b) Falhas de transmissão de uma geração para a próxima dos princípios operacionais em que a comunidade é baseada.

c) Programas baseados em manuais além de existir fácil acesso a financiamento externo.

d) Corrupção e outras formas de comportamento oportunístico.

e) Falta de mecanismos institucionais de grande escala relacionadas com a coleta, agregação e de divulgação de informações confiáveis; mecanismos de resolução de conflitos justos e de baixo custo; facilidades educacionais e de extensão; e facilidades para ajudar em caso de catástrofes naturais ou quando outros problemas graves ocorrem a nível local.

Pequenos sistemas policêntricos podem se desenvolver de forma sustentável e até mesmo perto da fronteira de eficiência de Pareto. Por governo policêntrico, deve-se entender um sistema onde cidadãos estão aptos a organizar não apenas um, mas múltiplos sistemas de governança em diferentes escalas. Cada unidade exercita uma independência considerável para fazer e aplicar regras dentro de um domínio circunscrito de autoridade para uma área geográfica delimitada.

Aprender habilidades para resolver problemas em um contexto local gera cidadãos com mais competência para resolução de um problema geral, além de permitir alcançar e analisar de forma mais eficaz os problemas que afetam sistemas sócio-ecológicos. Ostrom (2005) afirma que os recursos são melhor gerenciados de forma policêntrica, em que o próprio grupo consegue criar regras que promovam a sustentabilidade do sistema e que a criação de regras para o uso de recursos comuns é de grande complexidade.

Aumentar a autoridade dos indivíduos para elaborar as suas próprias regras pode resultar em processos que permitem envolver normas sociais e, assim, aumentar a probabilidade dos indivíduos empregarem esforços com o intuito de resolver problemas de ação coletiva. Sendo que regimes de auto-organização permitem as pessoas a aprender mais sobre suas necessidades e a necessidade de outros. Aprender habilidades para resolver problemas em um contexto local gera cidadãos com mais competência para resolução de um problema geral, além de permitir alcançar e analisar de forma mais eficaz os problemas que afetam sistemas sócio-ecológicos.

Indivíduos inseridos em pequenos grupos geralmente enfrentam dilemas sociais em ambiente do tipo dilema do prisioneiro. Segundo Epstein (1998), mesmo os indivíduos jogando um jogo do tipo dilema do prisioneiro, onde o resultado esperado seria a não cooperação, a interação repetida e a memória de interações anteriores fazem surgir zonas de cooperação.

Lansing (2006) faz um estudo de caso da emergência de cooperação entre produtores de arroz em terraços em Bali. Foi observado que com a interação durante décadas entre esses produtores emergiu um processo de cooperação, mesmo os agricultores não agindo diretamente para este fim, ou seja, obedecendo a regras locais de maximizar a produção diante dos problemas de disponibilidade de água, ataques de pragas e competição com a produção dos vizinhos imediatos, emerge o resultado global de sincronização da produção e redução de pragas.

Para buscar compreender o mecanismo de cooperação, Poteete, Janssen e Ostrom (2010) apontam um caminho que vai além da Teoria da Ação Coletiva, e essa construção teórica é baseada em três níveis de análise: comportamento humano individual; microsituações; e contextos sócio-ecológicos amplos.

2.2 CAPITAL SOCIAL

Para se explicar o desenvolvimento econômico, cada vez mais tem-se focado em estudos teóricos o capital social como uma importante ferramenta, como expõe Matos (2002).

Em comunidades onde a presença de capital social é maior, torna-se mais provável o desenvolvimento espontâneo de instituições capazes de solucionar dilemas sociais. O termo capital social é utilizado para caracterizar o conjunto das relações sociais em que um indivíduo se encontra inserido e que o ajudam a atingir objetivos que, segundo Coleman (1990), sem tais relações seriam inalcançáveis ou somente alcançáveis a um custo de transação mais elevado. O capital social é formado basicamente através da existência de grupos e redes formais e informais, de relações de confiança mútua, da presença de normas e sanções, de relações que envolvem a imposição de autoridade, além de relações que permitam aos indivíduos minimizar os custos de se obter informação.

Gomes (2005) afirma que o capital social é um fator adicional na função de produção, assim como o capital físico, gerando um fluxo de rendimentos para os indivíduos ou sociedades que a ele obtêm acesso. Essa adoção do capital social como fator de produção está, segundo Gomes e Bueno (2008), em sintonia com a evolução da teoria do crescimento e desenvolvimento econômico, que passa a aceitar também como fatores de produção o capital humano e a tecnologia endógena.

Fatores institucionais e as relações sociais são importantes para se explicar o desenvolvimento econômico de uma região. A observação de que não são apenas os fatores de produção tradicionais que afetam o desempenho econômico é realizada por Arraes e Barreto (2002) ao verificarem as implicações do capital social no desenvolvimento econômico analisando os efeitos de variáveis institucionais e do capital social na elevação do nível do PIB per capita e na promoção do desenvolvimento.

Quanto mais desenvolvidos os indicadores de capital social em uma comunidade, de acordo com Putnam (1996), maior a probabilidade de que os indivíduos agem de forma cooperativa em benefício coletivo.

Gomes e Bueno (2008) sugerem que a acumulação de capital social pode ajudar a superar dilemas de ação coletiva, porém é necessário que isto ocorra aliado a outros fatores, como o pequeno tamanho da comunidade e a participação efetiva dos agentes no processo, como proposto por Ostrom (2005).

Da revisão até este ponto emerge a conclusão de que, embora extremamente rica, a literatura da NEI ainda contém algumas lacunas sobre os fatores que propiciam ou desestimulam a emergência da cooperação em comunidades. Textos recentes, como Arthur (2005) e Poteete, Janssen e Ostrom (2010), tem sugerido que uma das formas mais promissoras de avançar nessa direção é através do uso de modelos baseados em agentes, como o modelo que é detalhado na próxima seção.

2.3 MODELO BASEADO EM AGENTE

Através da modelagem computacional baseada no agente, segundo Rauch (2002), é possível aprender a antecipar eventos futuros e saber onde procurar por intervenções que podem funcionar. Esse tipo de modelagem tende a mostrar que padrões sociais complicados podem emergir e se transformar a partir da adoção de regras simples pelos indivíduos.

O pioneiro nesse tipo de modelagem foi o economista Thomas C. Schelling, que elaborou um modelo para estudar a segregação de bairros americanos. Este modelo mostra que a segregação racial pode surgir mesmo em ambientes onde os agentes individualmente não sejam racistas.

O modelo de Schelling implica, de acordo com Rauch (2002), que mesmo sociedades simples podem produzir resultados ordenados e não intencionais: resultados que não foram em nenhum sentido acidentais e tão pouco intencionais. Sendo assim as interações das escolhas individuais é um sistema complexo, com resultados coletivos que não possuem uma relação estreita com as intenções individuais.

Modelos baseados em agente são representações computacionais de indivíduos autônomos que interagem com outros indivíduos em um nível local e produzem um padrão de comportamento global. Este tipo de modelo utiliza algoritmos para processar as informações baseadas em suposições sobre a habilidade cognitiva dos indivíduos e a topologia de suas interações. Para Poteete, Janssen e Ostrom (2010) como há muita liberdade para especificar os agentes e suas interações, é importante que as hipóteses utilizadas para montar o modelo estejam bem explícitas.

Este tipo de modelo é uma das ferramentas mais poderosas para estudar o surgimento da cooperação em ambientes sociais complexos. Eles mostram que a história de como os indivíduos tomam decisões, as informações que eles têm, e como eles trocam informações pode ser bastante diferente da visão que se supõe geralmente.

Agentes, de acordo com Arthur (2005), são adaptativos e heterogêneos, isto é, eles possuem diferentes experiências, diferentes histórias e diferentes psicologias.

Sendo os agentes heterogêneos, eles ajustam continuamente às situações globais que eles criam, e então se adaptam a esse novo ambiente. E ao se adaptarem eles mudam o ambiente. Assim modelos baseados em agentes podem ser entendidos como abordagens evolucionárias.

Tais modelos permitem representar, em múltiplas escalas de análise, a emergência de estruturas em níveis hierárquicos mais elevados a partir da ação individual. Tais modelos consistem de agentes, os quais em geral são constituídos por partes distintas de um programa computacional usadas para representar atores sociais. Os agentes interagem dentro de um ambiente também modelado em computador, de modo a representar o ambiente real em que os agentes operam.

As premissas chave de um modelo baseado no agente típico, como apresentam Macy e Willer (2002), são que os agentes:

- a) São autônomos, isto é, padrões sistêmicos emergem de baixo para cima, das interações locais entre tomadores de decisões autônomos;
- b) São interdependentes, sendo que esta interdependência pode se manifestar tanto através do contato interpessoal, como os agentes imitando o comportamento dos demais, ou através de modificações no ambiente produzidas por outros agentes e que afetam o comportamento dos demais;
- c) Seguem regras simples dadas, por exemplo, por instituições formais ou informais, como as normas e convenções sociais. Tais regras embora possam ser bastante simples podem produzir resultados não intuitivos e difíceis de entender. Modelos baseados em agente tentam identificar o conjunto mais simples possível de regras capazes de gerar um macro-padrão de interesse;
- d) São adaptativos e ajustam suas expectativas com base em observações e não em cálculos sofisticados sobre estratégias futuras.

Essas pressuposições podem ser sintetizadas, de acordo com Gilbert (2008), nos seguintes princípios práticos para construir um modelo baseado no agente onde os agentes devem:

- a) Ser capazes de perceber seu ambiente, isto é, de detectar quais objetos e agentes estão localizados em sua vizinhança;
- b) Poder se mover em um ambiente;
- c) Ser hábeis de mandar e receber mensagens de outros agentes;
- d) Conseguir interagir com seu ambiente;
- e) Poder armazenar informações e percepções de estados anteriores;
- f) Dispor de um conjunto de regras, heurísticas ou estratégicas que determinam seu comportamento a partir das informações disponíveis;
- g) O conjunto de informações, instituições, estratégias e características dos agentes e características físicas do ambiente (como a disponibilidade de áreas para agricultura na vizinhança relevante dos agentes), constituem o ambiente em que eles se movem.

2.3.1 Ambiente computacional

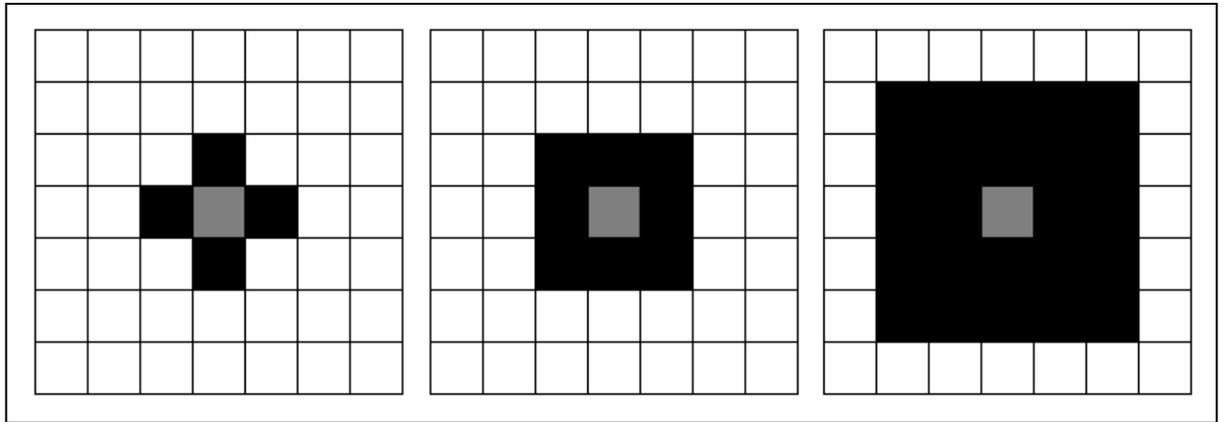
O ambiente em um modelo baseado em agentes é formado por autômatos¹ celulares (AC) que funcionam como reconhecedores de uma determinada linguagem e servem para modelar uma máquina ou um computador simples. Um AC é um sistema com um ambiente de células que podem ter um número limitado de estados e o tempo transcorre em intervalos discretos, um AC tem as seguintes características:

- a) Latitude n-dimencional, onde cada célula desta latitude tem um estado discreto;
- b) Comportamento dinâmico, correspondente à estratégia atribuída a ele. Estas estratégias descrevem o estado da célula em cada intervalo de tempo.

Cada célula possui um estado binário, que representa presença ou ausência de alguma característica. Em simulações complexas, cada célula pode ter múltiplos estados, além de poderem mudar o estado de acordo com as estratégias de transição,

¹ Um autômato funciona como um reconhecedor de uma determinada linguagem e serve para modelar uma máquina ou, se quiserem, um computador simples. É usado, por exemplo, em editores de texto para reconhecer padrões. Um conceito fundamental nos autômatos é o conceito de estado.

que determinam o estado da célula no próximo intervalo de tempo. Uma estratégia define o estado da célula com uma função de seus vizinhos. Dependendo da estrutura da vizinhança os resultados da simulação são diferentes. A vizinhança pode ser de três tipos como mostrado na figura 1:



Fonte: Poteete, Janssen e Ostrom (2010)

Figura 1 – Estrutura de vizinhança de autômatos celulares

- a) Zona de Von Neumann: onde os vizinhos estão dispostos nas quatro latitudes, Leste, Oeste, Norte e Sul;
- b) Zona de Moore: engloba a Zona de Von Neumann e acrescenta as células da diagonal, Noroeste, Nordeste, Sudoeste e Sudeste, e;
- c) Zona de Moore Estendida: engloba a Zona de Moore além das dezesseis células imediatamente ao seu redor.

Outro elemento importante em um modelo deste tipo é a figura do agente. Um agente é definido como um algoritmo autônomo tomador de decisões. Eles podem ser objetivos: agindo com o intuito de maximizar sua utilidade; reativos: respondendo automaticamente a mudanças no ambiente; e capazes de interagir com outros agentes. Em um modelo baseado em agente podem existir duas figuras:

- a) As *turtles*: são os indivíduos do modelo. As *turtles* possuem a habilidade de se mover ao longo do território. Elas também podem variar de tamanho, cor e tipo ou raça.

b) E os *patches*: que são as células que compõem o território, elas não se movem e podem ser ocupadas por uma ou mais *turtles*.

Essas duas figuras podem assumir n características e conseguem interagir entre si, ou seja, uma *turtle* pode interagir tanto com outra *turtles* quanto com um *patch*. Com relação as suas quantidades dentro do modelo, essas duas figuras diferem bastante. As *turtles* podem aumentar ou diminuir sua quantidade durante as interações, já os *patches* tem quantidade fixa, sendo o seu valor determinado antes de se iniciar as interações do modelo. Já as demais características atribuídas aos *patches* podem sofrer alteração do longo do tempo simulado.

3. METODOLOGIA

O modelo elaborado neste trabalho busca responder alguns questionamentos feitos pela Teoria de Ação Coletiva. Através do uso da modelagem baseada em agentes, entender como a heterogeneidade dos indivíduos e o tamanho da comunidade afeta o nível de cooperação dentro de uma comunidade. O modelo foi construído em duas etapas, na primeira parte o modelo é construído apenas com os indivíduos da comunidade, ou seja, se têm agentes que interagem uns com os outros, e nenhum efeito externo influencia suas ações. E assim é testado como os agentes reagem a variações no tamanho do grupo e diferentes níveis de heterogeneidade entre os membros da comunidade. Na segunda parte do trabalho é inserido o ambiente, um sistema de irrigação onde diferentes quantidades de recursos estão disponíveis no território, de acordo com a proximidade com o rio. Nessa segunda parte os agentes são influenciados pelas condições do ambiente, interagindo com ele, e o ambiente respondendo às ações dos agentes, se modificando e influenciando novamente os agentes. As condições dessas interações são descritas abaixo. Os agentes interagem em um jogo do tipo dilema do prisioneiro buscando maximizar sua obtenção de recursos neste sistema de irrigação.

3.1 O MODELO BASE

O modelo é baseado em um trabalho de Joshua Epstein, *Zones of Cooperation in Demographic Prisoner's Dilemma*. Neste artigo o autor testa a emergência de zonas de cooperação em um ambiente onde os indivíduos interagem seguindo um jogo do tipo dilema do prisioneiro. Neste modelo o ambiente é composto de um território dividido em 900 *patches*, e inicialmente por 100 indivíduos, cada *patch* pode ser ocupado por um único agente em cada interação.

Existem no modelo dois tipos de indivíduos, cooperativos e não-cooperativos, que são escolhidos e distribuídos de forma aleatória pelo território. Cada indivíduo possui um valor de recurso, que é iniciado com 10 unidades. Os agentes obedecem às seguintes regras:

- Regra de movimento: em cada rodada os agentes irão andar para um *patch* vizinho que esteja desocupado.
- Regra de interação: cada agente vai, em cada rodada, interagir com todos os seus vizinhos e dependendo do tipo do vizinhos o seu recurso vai variar positiva ou negativamente. Se o indivíduo tiver o recurso acumulado menor ou igual a zero ele morre e sai da simulação. O valor dos *payoffs* é dado pela seguinte matriz:

$$\begin{array}{c} C \\ D \end{array} \begin{array}{cc} C & D \\ (R, R) & (S, T) \\ (T, S) & (P, P) \end{array}$$

Onde R é a recompensa pela cooperação mútua, S é a perda por cooperar quando o outro não coopera, T é a tentação para não cooperar, e P é a punição pena não cooperação mútua. E sendo um jogo do tipo dilema do prisioneiro, $T > R > P > S$. Em um jogo sem repetição de dilema do prisioneiro a estratégia dominante é D para ambos os jogadores, sendo C indicativo de cooperação e D de não cooperação.

- Regra de reprodução: se o valor do recurso for igual ou maior que 10 e existir algum *patch* vizinho vazio o indivíduo vai reproduzir e criar outro agente com as mesmas características dele e com um recurso de 6 unidades.

3.2 CARACTERIZAÇÃO DO MODELO – PRIMEIRA PARTE

No modelo deste trabalho existem tanto as *turtles* quanto os *patches*. O território é formado, assim como no artigo de Epstein, por 900 patches. Estes patches não possuem nenhuma característica e não realizam nenhum tipo de interação. Inicialmente existem 100 *turtles* que estão dispostas aleatoriamente pelo território, sendo que cada *patch* pode ser ocupado por apenas uma única *turtle*. As turtles possuem duas características:

- a) *Resource*: que é o recurso necessário para manter o indivíduo vivo. No instante inicial todos os agentes possuem 10 unidades de *resource*, que pode

ser acumulado ao longo das interações, contudo se o indivíduo em determinada rodada obtiver um valor menor ou igual a 0 de *resource* ele morre e sai da simulação.

- b) *Cooperation*: esta característica possui um valor binário, 0 ou 1. Se o indivíduo apresentar 0 neste atributo, ele será um agente cooperativo, em caso contrario, se ele apresentar o valor 1, o agente será não cooperativo.

Diferentemente do modelo de Epstein (1998) a cooperação dos indivíduos não é fixa, ou seja, um indivíduo pode cooperar em determinada situação e não cooperar em outra. Assim a cooperação obedece a uma probabilidade, que vai depender de uma propensão a não cooperação λ (quanto menor o seu valor, maior será a nível de cooperação), e de uma função distância dos *payoffs*:

$$\frac{T - R}{R} \quad (1)$$

Essa função distância mostra a porcentagem em que a tentação a não cooperação é maior que a recompensa por cooperar.

Segundo Ledyard (1995), em comunidades geralmente em média 50% dos indivíduos agem exatamente como previsto pela teoria dos jogos, onde a estratégia de equilíbrio de Nash é a não cooperação. Outros 40% respondem aos seus próprios interesses, assim como um jogador Nash se os incentivos forem altos o suficiente, porém também cometem erros e respondem a decisões de custos, a lealdade e ao altruísmo. E os outros 10% tem um padrão totalmente aleatório, se comportando de maneira inexplicável.

Desta forma, para que a probabilidade de cooperação do modelo represente o padrão encontrado por Ledyard (1995), o parâmetro λ para 50% da população será exatamente igual ao inverso da função distância, para 40% não existe um valor definido, e será utilizado um valor arbitrário de 0,75, e para os outros 10% varia entre zero e o inverso da função distância. Assim a probabilidade de um indivíduo ser cooperativo é:

$$p_{coop} = 1 - \min\left(1, \lambda \frac{T - R}{R}\right) \quad (2)$$

Com essa probabilidade e tendo o λ como descrito acima, 50% da população será necessariamente egoísta, ou seja, agindo de forma não cooperativa, enquanto os outros 50% podem ser egoístas, mas também podem ser altruístas, e agirem cooperativamente.

Assim, tendo uma matriz de *payoffs* com $T = 6$ e $R = 5$, como a proposta por Epstein, a probabilidade de cooperação ficará:

$$p_{coop} = 1 - \min\left(1, \lambda \frac{6 - 5}{5}\right) \quad (3)$$

Para 50% da população a probabilidade de cooperação será:

$$p_{coop} = 1 - \min\left(1, \frac{5}{6 - 5} \frac{6 - 5}{5}\right) \quad (4)$$

$$p_{coop} = 1 - \min(1, 1) \quad (5)$$

$$p_{coop} = 0 \quad (6)$$

Assim, essa parcela da população será sempre não cooperativa. Para 40% da população, tendo $\lambda = 0,75$ a probabilidade de cooperação será:

$$p_{coop} = 1 - \min\left(1, 0,75 \frac{6 - 5}{5}\right) \quad (7)$$

$$p_{coop} = 1 - \min(1, 0,15) \quad (8)$$

$$p_{coop} = 0,85 \quad (9)$$

Neste caso, com $\lambda = 0,75$ e a referida matriz de *payoffs*, esta parcela da população terá 85% de chance de ser cooperativa. Para os 10% restante da população o valor de λ varia entre 0 e o inverso da função distância, como mencionado anteriormente, assim sendo $\lambda = 0$:

$$p_{coop} = 1 - \min\left(1, 0 \frac{6-5}{5}\right) \quad (10)$$

$$p_{coop} = 1 - \min(1, 0) \quad (11)$$

$$p_{coop} = 1 \quad (12)$$

Assim, esses 10% da população varia entre totalmente cooperativa, e totalmente não cooperativa, assim como observou Ledyard.

Os agentes que forem cooperativos serão representados como um círculo, enquanto os agentes não cooperativos serão triângulos. No decorrer da simulação os agentes obedecem três regras, movimento, interação e reprodução, nesta ordem. Após realizado o movimento, inicia a interação e quando este terminar, começa a reprodução:

- a) Regra de Movimento: a *turtle* vai a cada interação se dirigir para um *patch* vizinho. Se este *patch* estiver ocupado por outra *turtle*, ela vai se mover mais uma vez. O movimento vai cessar quando o indivíduo encontrar um local desocupado.
- b) Regra de Interação: o agente vai interagir com todos os seus vizinhos, sendo que os vizinhos são aqueles que estão na Zona de Von Neumann, e o seu recurso resultante será igual a somatória dos *payoffs* para cada vizinho mais o recurso do período anterior, obedecendo a matriz de *payoffs* pré-estabelecida. Por exemplo, se o agente for cooperativo e tiver três vizinhos na Zona de Von Neumann, sendo dois deles cooperativos e um não cooperativo, o seu recurso atual será:

$$\text{Recurso atual} = \text{Recurso anterior} + R + R + S \quad (3)$$

Sendo o R recebido ao interagir com cada vizinho cooperativo e S com o não cooperativo.

- c) Regra de Reprodução: Se o valor do recurso do agente for maior ou igual a 10 unidades e existir algum *patch* vazio ao seu redor, ele irá reproduzir, e criará outro agente neste *patch* vazio selecionado. Este novo agente terá as mesmas características do agente base e seu recurso inicial será de 6 unidades.

A programação completa desta primeira parte do modelo, que foi executada através do software *NetLogo*, está detalhada no Apêndice 1 deste trabalho.

3.3 SEGUNDA PARTE - MODELO DE IRRIGAÇÃO

Na segunda parte do modelo existem tanto as *turtles* quanto os *patches*. O território é formado por 2500 *patches*. Cada *patch* possui uma quantidade máxima de recurso e realizam interação com os agentes que o ocupa. Inicialmente existem 300 *turtles* que estão divididas em dois grupos, um disposto na parte superior e outra na parte inferior da simulação, e cada *patch* pode ser ocupado por apenas uma única *turtle*. As *turtles* possuem três características:

- a) *Resource*: que é o recurso necessário para manter o indivíduo vivo. No instante inicial cada agente possui de 1 a 20 unidades de *resource*, escolhido de forma aleatória, e assim como na primeira parte do modelo, pode ser acumulado ao longo das interações, contudo se o indivíduo em determinada rodada obtiver um valor menor ou igual a 0 de *resource* ele morre e sai da simulação.
- b) *Consumption*: que é a quantidade de recurso que o agente retira do ambiente a cada rodada. Cada comunidade possui níveis distintos de consumo, caracterizando o padrão de heterogeneidade entre os seus membros, uma comunidade menos heterogênea possui membros que consomem menos recursos, possibilitado que mais indivíduos utilizem de tal recurso. A comunidade localizada na parte superior da simulação é menos heterogênea, tendo seus membros consumindo de 1 a 2 unidades de recurso por rodada. Já os agentes na comunidade a montante consome de 3 a 6 unidade de recurso por rodada de interação.

- c) *Vision*: que indica qual a distancia que cada indivíduo consegue enxergar. Não há diferenças entre as duas comunidades, cada indivíduo consegue enxergar e se mover de 1 a 3 patches.

Os patches também possuem a característica de recurso. Existem quatro faixas de recursos no ambiente, as áreas mais próximas do rio e dos canais de irrigação possuem uma quantidade máxima de 4 unidades de recursos. À medida que o território está mais distante da fonte de água, menos recurso está disponível, sendo 3, 2 e 0 a quantidade máxima de recurso disponível, como indicado na figura 2 abaixo:

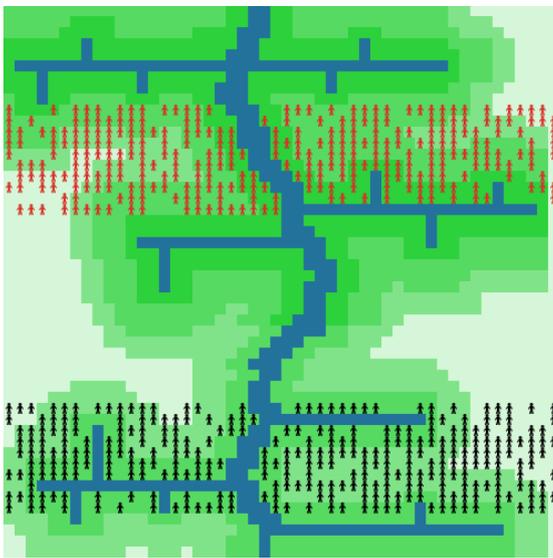


Figura 2 – Ambiente virtual

Fonte: Resultado da pesquisa

Os *patches* e as *turtles* também obedecem a regras. Os *patches* possuem a regra de regeneração do recurso, na qual cada *patch* regenera seu recurso a uma taxa de 30%, até o atingir o seu valor máximo. As *turtles* obedecem a duas regras:

- a) *Regra de Movimento*: a *turtle* vai a cada interação se dirigir para o *patch* que tiver a maior quantidade de recurso dentro de seu campo de visão. Se este *patch* estiver ocupado por outra *turtle*, ela vai se mover mais uma vez. O movimento vai cessar quando o indivíduo encontrar um local desocupado.

b) *Regra de Consumo*: cada *turtles* a cada rodada vai retirar do *patch* em que está localizado o valor de recurso de seu consumo e este será acumulado. Se o valor de recurso do *patch* for inferior ao valor de consumo o agente vai retirar todo o valor de recurso do *patch* e para completar suas necessidades vai retirar de sua reserva a quantidade que estiver faltando para seu consumo. Se o valor acumulado de recurso da *turtle* for menor ou igual a zero, a *turtle* morrerá e será retirado da simulação.

A programação completa da segunda parte do modelo, que também foi executada através do software *NetLogo*, está detalhada no Apêndice 2 deste trabalho.

4. RESULTADOS E DISCUSÕES

4.1 RESULTADOS PARA A PRIMEIRA PARTE

Este trabalho se propôs a investigar os efeitos de variáveis microsituacionais sobre a cooperação. Para tanto foi construído um modelo baseado em agente, onde as regras que os indivíduos seguem foram apresentadas na metodologia. Os testes foram realizados com os agentes interagindo durante 1000 rodadas, e repetidos trinta vezes para cada matriz de payoffs. O modelo é iniciado com uma população de 100 indivíduos dispersos aleatoriamente pelo ambiente composto de 900 sítios (*patches*) e, como já foi mencionado anteriormente, cada *patch* pode ser ocupado por apenas um agente (*turtle*). Os passos para construir o modelo estão descritos no apêndice 1.

4.1.1 Efeito da heterogeneidade

A heterogeneidade foi medida através da diferenciação da distância dos resultados R e T da matriz de *payoffs*, ou seja, quanto maior for a distância, mais heterogêneo será o grupo.

As pessoas possuem habilidades próprias, quanto maiores e mais distintas forem as habilidades de um indivíduo em relação ao restante do grupo em que está inserido, maior será o esforço que esta pessoa fará para melhorar os seus resultados e assim se tornar individualmente mais lucrativo. Portanto, são as diferenças entre as habilidades dos agentes e os resultados de suas ações individuais que criam as tentações para agirem de forma não cooperativa.

Dessa forma, quanto maior a diferença existente entre os membros de um grupo maior será a tentação a agir de forma *free-riding* e *rent-seeking*. Assim, foram realizados testes aumentando a diferença entre R e T, utilizando as seguintes matrizes:

$$\begin{array}{cc} & C & D \\ \begin{array}{c} C \\ D \end{array} & \begin{pmatrix} (5,5) & (-6,6) \\ (6,-6) & (-5,-5) \end{pmatrix} & (a) \end{array}$$

$$\begin{array}{cc} & C & D \\ \begin{array}{c} C \\ D \end{array} & \begin{pmatrix} (5,5) & (-8,8) \\ (8,-8) & (-5,-5) \end{pmatrix} & (b) \end{array}$$

$$\begin{array}{c}
 C \\
 D
 \end{array}
 \left(
 \begin{array}{cc}
 C & D \\
 (5,5) & (-10,10) \\
 (10,-10) & (-5,-5)
 \end{array}
 \right) \quad (c)$$

$$\begin{array}{c}
 C \\
 D
 \end{array}
 \left(
 \begin{array}{cc}
 C & D \\
 (5,5) & (-10.2,10.2) \\
 (10.2,-10.2) & (-5,-5)
 \end{array}
 \right) \quad (d)$$

Em um jogo do tipo dilema do prisioneiro, como o que é representado nessas matrizes, quando há apenas uma interação entre os agentes, o resultado de equilíbrio é a não cooperação mútua entre os jogadores. Contudo, quando o jogo é repetido inúmeras vezes, o resultado é alterado. Cada agente será cooperativo ou não cooperativo em cada rodada de acordo com sua probabilidade de cooperação, que segue a função descrita na metodologia.

Os resultados em um modelo de agentes são extremamente sensíveis às condições iniciais do modelo. Durante as trinta repetições, para cada matriz de payoffs, nenhuma variável é alterada. Contudo, nos testes da matriz (a) foram identificados dois padrões de resultados distintos, o caso predominante que ocorre em cerca de 70% das repetições, que apresenta estabilidade nas quantidades de agentes cooperativos, que ficaram em torno de 650 indivíduos, e não cooperativos, entre 200 e 300 agentes, como mostrado na figura 3 abaixo:

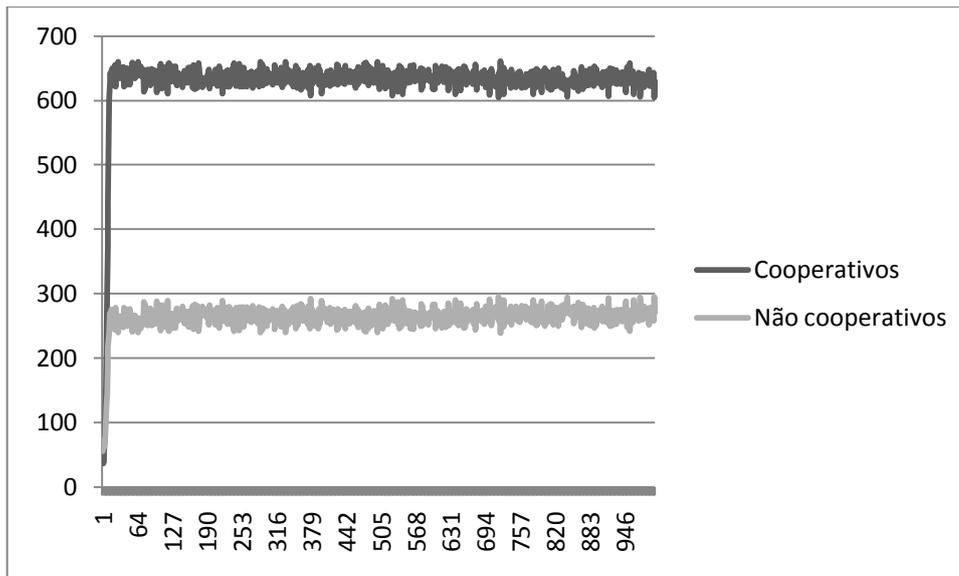


Figura 3 – Evolução dos níveis de cooperação para 70% das repetições para $T = 6$
 Fonte: Resultado da pesquisa

Nas repetições restantes, os outros 30%, apresentaram certa instabilidade, ou seja, há uma queda do número de agentes cooperativos de equilíbrio e uma predominância de agentes não cooperativos, como mostra a figura 4 abaixo:

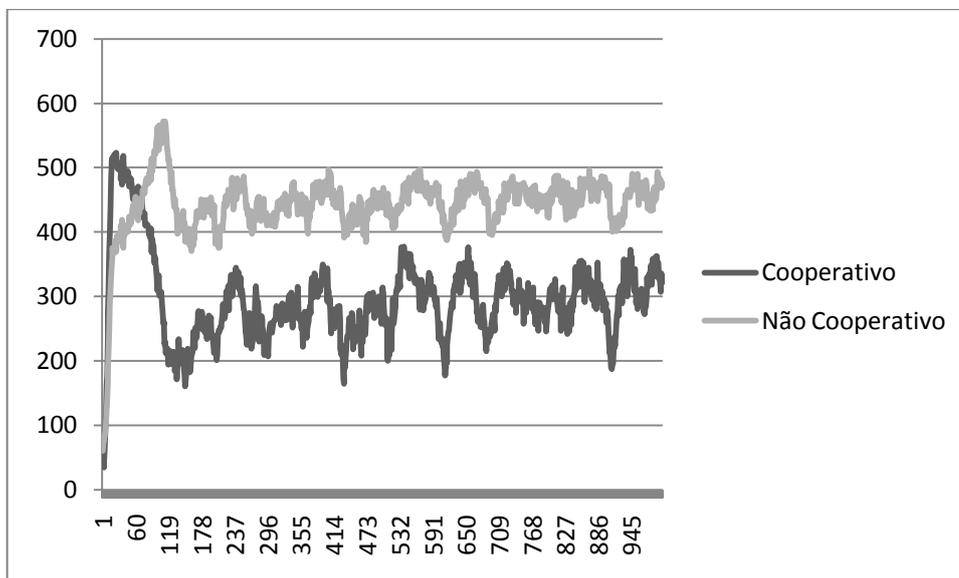


Figura 4 – Evolução dos níveis de cooperação para 30% das repetições com $T = 6$
 Fonte: Resultado da pesquisa

Esses 30% dos testes apresentam uma população cooperativa com um nível mais baixo, variando entre 200 e 400 indivíduos. Porém para esse nível de tentação a

não cooperar, mesmo apresentando dois padrões diferentes, a comunidade consegue se equilibrar e coexistir indefinidamente.

Espera-se ao realizar testes com distancias cada vez maiores, que se tenham cada vez menos indivíduos cooperativos e mais não cooperativos. Bem como foi observado por Ledyard (1995) que diz que mesmo agentes altruístas agiriam como Nash players quando a tentação a não cooperar for grande o suficiente. Os testes foram realizados para um T de 8, 10 e 10,2, como mostrado pela tabela 1 abaixo:

	T	Cooperativos					Não Cooperativos				
		Mínimo	Máximo	Média	Desvio	CV	Mínimo	Máximo	Média	Desvio	CV
População inicial	6	36	45	39,44	2,23	5,65	62	79	60,55	2,21	3,65
	8	21	38	28,26	4,22	14,93	62	79	71,73	4,22	5,88
	10	11	28	16,73	3,83	22,89	72	86	83,26	3,83	4,60
	10,2	7	23	13,8	3,4	24,63	77	93	86,3	3,4	3,94
População Final	6	229	630	451,13	131,51	29,15	203	401	392,72	73,19	18,66
	8	67	187	130,53	31,35	24,02	203	401	315	51,3	16,28
	10	0	116	36,06	31,59	87,60	2	361	136,21	111,51	81,86
	10,2	0	1	0,2	0,4	200	2	10	3,5	1,83	52,28

Tabela 1 – Estatísticas dos testes de cooperação com T = 8, T = 10 e T =10,2

Fonte: Resultado da pesquisa

Com o aumento da tentação a agir de forma não cooperativa T, há uma queda tanto na quantidade inicial quanto na final da população cooperativa, além de um aumento em sua variabilidade, medido pelo coeficiente de variação. Enquanto os agentes não cooperativos inicialmente tem sua população inicial aumentada com o aumento da heterogeneidade, contudo o resultado final é uma queda significativa nos níveis de cooperação.

A comunidade consegue se manter até a tentação da não cooperar ser o dobro do valor da recompensa pela cooperação mútua. Quando o T vale 10 unidades o grupo atinge seu mais baixo nível de cooperação de forma sustentável, a partir deste ponto qualquer ampliação no incentivo a não cooperar leva ao colapso, como mostra a figura 5 a seguir:

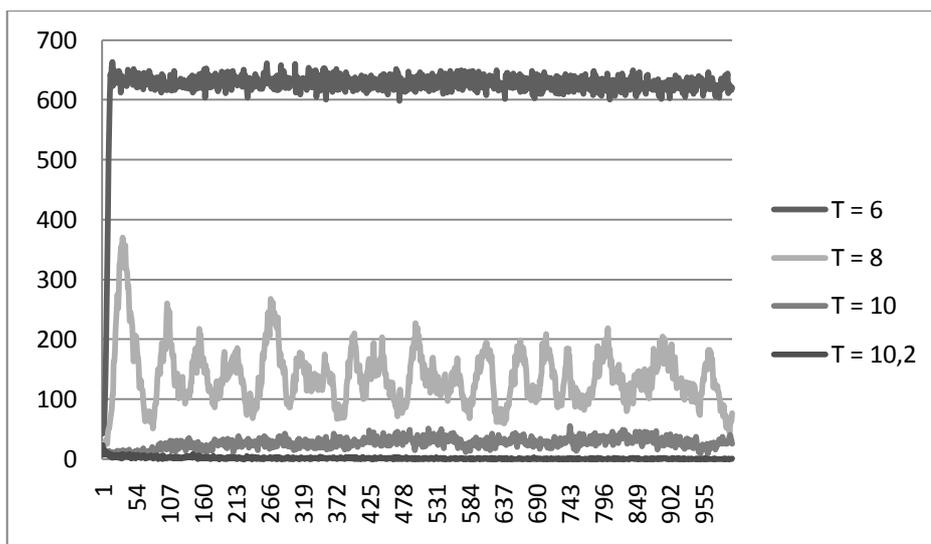


Figura 5 – Evolução das quantidades de agentes cooperativos com o aumento da heterogeneidade

Fonte: Resultado da pesquisa

Quando os payoffs de $R = 5$ e $T = 6$, 70% dos casos a população cooperativa se estabiliza acima de 600 agentes. À medida que essa diferença é aumentada, com um T igual a 8, 10 e 10,2 respectivamente, a quantidade de agentes cooperativos é diminuída drasticamente. O caso limite onde a comunidade consegue existir mesmo com um nível muito baixo de cooperação é quando a tentação a cooperar é o dobro da recompensa pela cooperação. A partir deste ponto o assentamento entra em um ciclo de colapso, quanto maior for a tentação a não cooperar, mais rápido se dará o declínio do grupo.

Assim é possível afirmar que a heterogeneidade é uma causa tanto de instabilidade quanto de colapso para uma comunidade. Quanto mais diferentes forem os membros de um grupo, mais distintos serão seus interesses, e maior será a tentação a agir de forma egoísta e mais difícil será fazer com que seus membros se unam com o intuito de criar as instituições necessárias para a redução dos custos de produção e fornecimento de bens públicos. Sendo que quando a tentação a não cooperar é o dobro da recompensa pela cooperação a comunidade consegue coexistir em seu nível populacional mais baixo, a partir deste ponto a comunidade entra em declínio.

4.1.2 Efeito do tamanho

Já para se verificar o efeito do tamanho do grupo sobre o nível de cooperação, foram realizados testes com populações de 100, 200 e 400 indivíduos, os resultados das simulações são mostrados a seguir na tabela 2:

	Qtd	Cooperativos					Não Cooperativos				
		Mínimo	Máximo	Média	Desvio	CV	Mínimo	Máximo	Média	Desvio	CV
População inicial	100	36	45	39,44	2,23	5,65	62	79	60,55	2,21	3,65
	200	66	86	78,66	4,46	5,67	114	134	121,33	4,46	3,67
	400	147	167	159,23	5,48	3,44	233	253	240,76	5,48	2,27
População Final	100	229	630	451,13	131,51	29,15	203	401	392,72	73,19	18,66
	200	236	483	299,32	51,86	17,29	366	520	446,78	30,07	6,73
	400	180	345	278,23	44,14	15,86	372	501	439,56	31,82	7,24

Tabela 2 – Estatísticas dos testes de cooperação para diferentes quantidades iniciais da população

Fonte: Resultado da pesquisa

Com o aumento do tamanho do grupo, a proporção inicial entre agentes cooperativos e não cooperativos não se altera. E no decorrer das simulações ocorre uma pequena queda na população cooperativa, entre os testes com 100 e 200 agentes, ao se dobrar novamente o tamanho, passado para 400, o resultado não sofre grandes alterações. O aumento do tamanho do grupo não gera redução significativa dos seus níveis de equilíbrio. A figura 6 abaixo mostra a evolução dos níveis de cooperação com o aumento do tamanho do grupo:

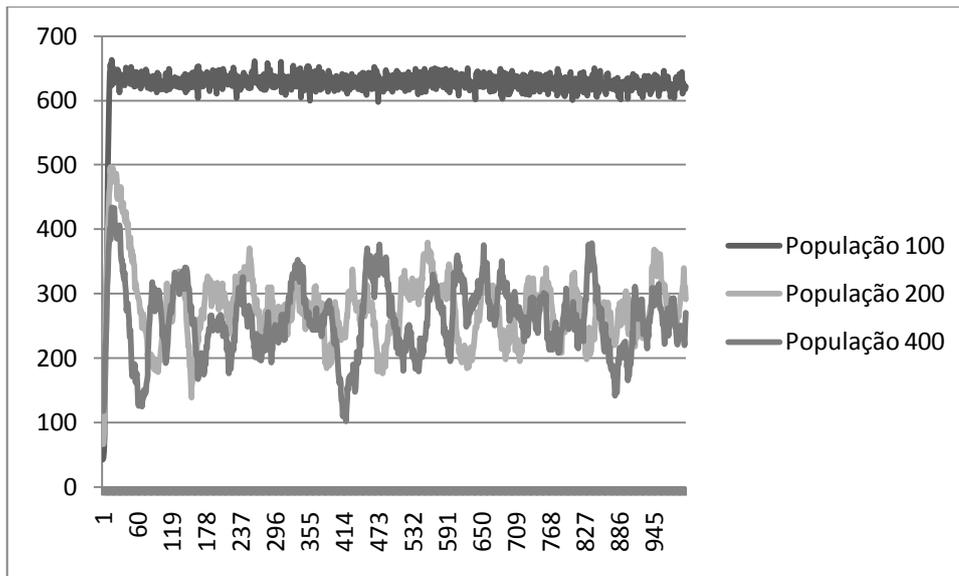


Figura 6 – Evolução dos níveis de cooperação para população inicial de 100, 200 e 400 indivíduos

Fonte: Resultado da pesquisa

Quando o grupo é pequeno o nível de cooperação é mantido um nível bastante alto, ocupando cerca de 66% do território, uma vez que o tamanho limite é de 900 agentes. Ao se aumentar a concentração inicial desta população, aumentando o tamanho da comunidade, o nível de cooperação é diminuído, porém a população cooperativa fica em torno de 250 indivíduos mesmo com uma concentração maior.

Ao se aumentar ainda mais o tamanho do grupo, o esperado seria um aumento da instabilidade e indícios de tendência a ciclos de colapso. Todavia, ao se dobrar novamente o tamanho inicial da comunidade. Ocorre realmente um aumento da instabilidade em alguns instantes das interações, contudo a população cooperativa ainda se concentra entre 200 e 400 indivíduos. E à medida que as interações evoluem os grupos ficam menos instáveis e mais homogêneos.

Como o defendido por Olson e a Teoria da Ação Coletiva, onde grupos maiores tentem a apresentar menores níveis de cooperação pela maior presença de comportamentos oportunistas, os testes aqui realizados, confirmam essa tendência. Assim, com grupos maiores ocorre menos cooperação. Contudo, o tamanho o grupo provoca apenas menores níveis de cooperação, não sendo um fator suficiente para causar o colapso de uma comunidade.

Por outro lado, a heterogeneidade tem o poder de gerar ciclos de colapso. Assim quanto mais heterogêneo o grupo, mais instável será a evolução da população e menor será o grupo, tendendo ao desaparecimento da comunidade, uma vez que se torna mais difícil a união de seus para a criação de mecanismos que sustentem a cooperação. Quanto maior a heterogeneidade, mais rápido ocorrerá o desaparecimento do grupo.

4.2 RESULTADO PARA A SEGUNDA PARTE - MODELO DE IRRIGAÇÃO - E COMPARAÇÃO ENTRE O MODELO VIRTUAL E O CASO REAL

O caso real utilizado para comparar os resultados encontrados no modelo deste trabalho é um estudo realizado por Bueno (2005). Neste trabalho o autor estuda duas comunidades rurais localizadas na região do Vale do Rio Doce próximo a cidade de Tumiritinga, no norte do estado de Minas Gerais, as comunidades Primeiro de Julho e Cachoeirinha. A localização é indicada pela Figura 7 abaixo:



Figura 7 – Mapa da Região do Vale do Rio Doce, onde estão localizadas as duas comunidades (Primeiro de Julho e Cachoeirinha).

Fonte: Google Earth

Segundo Bueno (2005) o Assentamento Primeiro de Junho foi implantado em 1993, após a ocupação da então Fazenda Califórnia por integrantes do Movimento dos Sem Terra (MST). Esta comunidade está localizada na região de Governador Valadares, às margens do Rio Doce e é composto por 83 famílias. Na época, criou-se na mesma área o Assentamento Cachoeirinha, com 33 famílias oriundas da própria região e que se organizaram na fase final da ocupação da fazenda para reivindicar parte das terras. Isto criou uma situação inicial de conflito, visto que os ocupantes do MST consideraram que as famílias da região não faziam jus à terra por não haverem participado efetivamente da longa luta que se iniciou em 1988, além do fato das famílias da região não aceitarem as regras impostas pelos integrantes do assentamento Primeiro de Julho. Desde o início, todas as famílias do Assentamento Cachoeirinha organizaram seu trabalho e dividiram a posse da terra sob o regime individual.

Assim é possível estabelecer algumas características para as duas comunidades, o assentamento Primeiro de Julho é duas vezes e meia maior que o assentamento Cachoeirinha, o que tende a torná-lo mais instável e, embasados pela teoria de Olson, torná-lo também mais propenso a observarem comportamentos oportunistas. O fato das famílias integrantes da comunidade Primeiro de Julho terem trajetórias de vidas distintas e pertencerem a regiões diferentes, a torna mais heterogênea que a comunidade Cachoeirinha, em que seus habitantes pertencem a mesma região e possuem uma proximidade maior.

Com base em dados reais, obtidos através de pesquisa de campo, é possível construir um modelo baseado no agente que represente as condições de vida da comunidade e a partir daí identificar quais variáveis são mais importantes para a sustentabilidade da cooperação. E assim identificar os meios de criar instituições que possam melhorar o desempenho econômico da comunidade.

O modelo construído para simular a dinâmica dessas comunidades reproduz o rio, que percorre o ambiente e possui ramificações, e sendo que quanto mais próximo a terra está desse rio mais fértil ela será. A fertilidade é dada pela quantidade de recursos existentes na terra. Outra característica do ambiente é que a região a montante é mais fértil, e conforme se caminha para baixo, a quantidade de recursos é

reduzida. Este ambiente tem uma taxa de regeneração que não é total, ou seja, a cada rodada o ambiente vai se regenerando menos, dependendo da absorção de recursos por parte do agente ali presente. Os passos necessários para construir um ambiente virtual a partir de um real estão descritos no apêndice 2.

Dois grupos distintos de indivíduos foram introduzidos no ambiente, representando duas comunidades diferentes. O grupo que se encontra na região jusante da simulação é mais heterogêneo e seu ambiente possui menos recursos, enquanto que o grupo localizado a montante do rio é menos heterogêneo e possui uma quantidade maior de recursos naturais.

Cada grupo tem um nível de visão e um ritmo metabólico. É possível realizar simulações com essas variáveis, variando seus valores e ver o resultado das interações. As outras características que fazem parte da criação dos indivíduos são iguais para os dois grupos, como a regra de movimento e a quantidade de recurso.

Para que os indivíduos pudessem interagir uns com os outros introduziu-se a produção, que será maior quando esta for realizada juntamente com um vizinho. Neste caso, o agente analisará se é ou não vantajoso interagir.

A diferença dos valores de recursos, tanto dos indivíduos como do ambiente, foi necessária para verificar a sobrevivência e a preferência de lugar na disposição dos indivíduos de cada grupo no ambiente.

O modelo tem como finalidade verificar o comportamento de duas comunidades reais com diferentes necessidades de consumo e grau de degradação ambiental, advinda de fatores passados da comunidade (como por exemplo, a cultura passada entre as gerações), sem construir leis, regras ou normas quaisquer conseguem adaptar-se ao ambiente. O resultado é a extinção do grupo mais heterogêneo e que degrada mais o ambiente. Enquanto o grupo mais homogêneo permanece.

A figura 8 abaixo mostra o início das simulações e a evolução das duas comunidades ao longo do tempo, além o efeito que estas causam no ambiente.

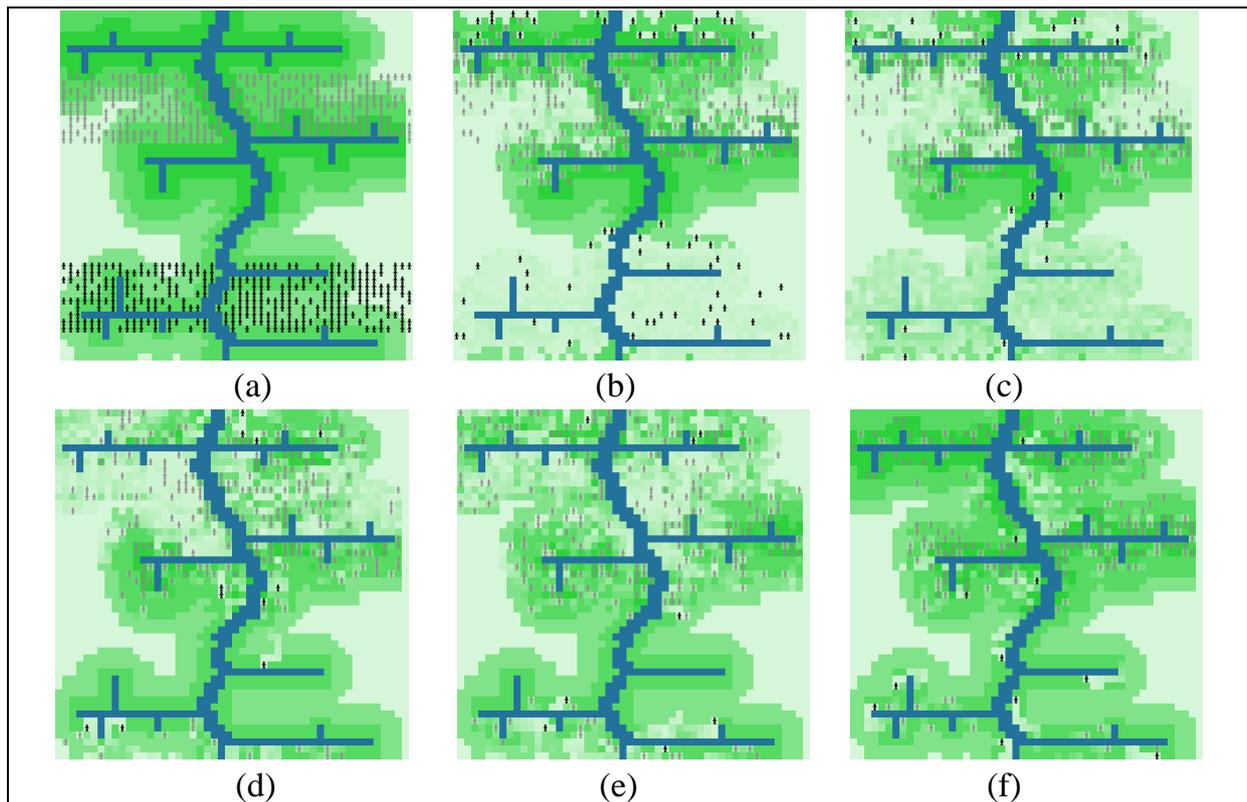


Figura 8 - Visualização das rodadas de interação no modelo de Irrigação
Fonte: Resultado da pesquisa

A imagem (a) da Figura 8 acima mostra o momento inicial da simulação, onde o ambiente ainda está inalterado e as duas comunidades estão dispostas a montante, com membros mais homogêneos, e a jusante, com indivíduos mais heterogêneos. Ao iniciar a simulação os agentes buscam recursos para sua sobrevivência, e retira este do ambiente. O grupo a jusante que utiliza uma quantidade maior de recursos naturais, degrada mais o ambiente, como mostra a imagem (b). A medida que não existe recursos suficientes para todos os indivíduos, alguns deixam a comunidade e são retirados da simulação.

O grupo a jusante, mais heterogêneo, degrada seu ambiente de forma bem mais rápida que o outro grupo a montante, mais homogêneo, como indica a imagem (b). A piora das condições do ambiente obra a população mais heterogênea a saírem da simulação. A população mais homogênea se espalha pelo território, mas continua a montante, onde existe uma quantidade maior de recursos.

O grupo a montante, que consegue utilizar menos recursos naturais, também degrada o ambiente, mas leva mais tempo para isso, mesmo assim alguns de seus moradores não conseguem se adaptar e saem da simulação. O resultado é uma adaptação da comunidade ao ambiente e este conseguindo se regenerar de forma satisfatória, como indica as mudanças entre as imagens (e) e (f) da simulação. No fim a comunidade mais heterogênea acaba por desaparecer, enquanto a comunidade mais homogênea consegue se ajustar ao ambiente e sobreviver.

Os resultados obtidos neste trabalho corroboram os fatos observados por Bueno (2005). Foi identificado pelo autor um alto grau de capital social entre os integrantes da comunidade Primeiro de Julho, o que para o autor, é resultante da experiência comum de lutas pela terra e de convivência, contudo a comunidade não havia conseguido até então se desenvolver economicamente. Além disso, mesmo situando-se às margens de um grande rio, o que torna a água um fator relativamente abundante, Bueno (2005) observou durante suas pesquisas de campo, um uso apenas incipiente de irrigação. Contrariamente, no Assentamento Cachoeirinha, uma parcela expressiva dos produtores adotou a irrigação, mesmo que os resultados econômicos não tenham sido, até o momento da pesquisa, ainda muito significativos.

Uma informação importante, e que merece destaque, é que atualmente o Assentamento Primeiro de Julho não existe mais, enquanto a comunidade Cachoeirinha, permanece no local. Fato que foi encontrado nos resultados este trabalho, em que grupos mais heterogêneos tendem a desaparecer com o tempo. Uma das causas que levaram a extinção pode ser a maior heterogeneidade desta comunidade.

5. CONCLUSÃO

O objetivo deste trabalho foi contribuir para preencher duas lacunas existentes na literatura, o efeito que as variáveis microsituacionais, heterogeneidade e do tamanho do grupo, causam sobre a cooperação entre membros de assentamentos rurais. Utilizando modelos baseados em agentes conclui-se que:

- Comunidades maiores geram mais agentes não cooperativos, assim como defendido pela teoria de Olson. Contudo, o tamanho do grupo diferentemente do que era esperado, a presença de mais agentes agindo de forma oportunísticas, não é um fator suficiente para extinguir uma comunidade.
- Já a heterogeneidade é uma variável importante, quanto mais heterogêneo o grupo, mais instável é ele. Sendo a tentação a não cooperar o dobro do valor da recompensa pela cooperação mútua, a cooperação ainda é sustentada, mesmo com altos níveis de agentes não cooperativos. A partir deste ponto, qualquer adição à tentação a não cooperação leva a comunidade à extinção, quanto maior for o valor, mais rápido ocorrerá.

Ao se comparar os resultados modelados com os observados em pesquisas de campo, percebe-se que o modelo computacional consegue simular bem a realidade. E pode ser utilizado para identificar quais fatores afetam a sobrevivência de um grupo, e a partir daí antecipar possíveis problemas para tentar evitá-los. Auxiliando na criação de mecanismos capazes de aumentar a cooperação e que possam desenvolver a comunidade.

Por exemplo, em ambientes mais heterogêneos, a criação de novas oportunidades econômicas, como acontece em regiões que estão se modernizando rapidamente, faz com que aumente o payoff das atividades não cooperativas e tal fato pode levar a extinção da comunidade. Assim mecanismos devem ser criados para restringir esses payoffs, tornando a ação isolada de um indivíduo mais penosa, mais cara em se comparado com a adoção de trabalhos coletivos, que visem a criação de bens públicos.

Assim, das variáveis microsituacionais, a que realmente tem impacto sobre a sustentabilidade de um assentamento é o nível de heterogeneidade e não tanto o tamanho desta comunidade.

Esses resultados são ainda muito preliminares, mas apontam no sentido de que a modelagem baseada em agentes pode ser uma metodologia útil para estudos em economia institucional.

6. REFERENCIAS BIBLIOGRÁFICAS

ARRAES, R. A. BARRETO, R. C. S. Implicações do capital social e do capital político no desenvolvimento econômico. In: **VII Encontro Regional de Economia**, 2002, Fortaleza-CE. Disponível em:

<http://www.bnb.com.br/content/aplicacao/ETENE/Anais/docs/implicacoes.pdf>.

ARTHUR, W. B. Out-of-equilibrium economics and agent-based modeling. **Handbook of Computational Economics**. Vol. 2, 2005

BATES, R. Social dilemmas and rational individuals: an assessment of the new institutionalism. In: Harriss, J. et al., **The new institutional economics and third world development**. London: Routledge, 1995.

BUENO, N. P. **Arquétipos de problemas institucionais sistêmicos aplicados a projetos de irrigação** – adaptação e simulação de modelos de desenvolvimento sustentável para regiões subdesenvolvidas do estado de Minas Gerais. CNPq EDITAL 32004 HUM/SOC/AP/CIÊNCIAS HUMANAS, Relatório final de pesquisa. 2005

COASE, R. The New Institutional Economics. **The American Economic Review**. Vol. 88, n. 1, 1998, pp.72-74.

COLEMAN, J. S. Foundation of social theory. Cambridge, Mass: **Harvard University Press**, 1990.

DIXIT, A. NALEBUFF, B. **Thinking Strategically**: the competitive edge in business, politics, and everyday life. New York: W.W. Norton, 1991.

EPSTEIN, J. M. Zones of Cooperation in Demographic Prisoner's Dilemma. **Complexity**, 1998, 4: 36-48.

GILBERT, N. Agent-based models. Thousand Oaks, Cal.: **Sage Publications**, 2008.

GOMES, A. P. W. **Capital Social e ação coletiva**: um estudo sob a ótica da Nova Economia Institucional no Assentamento Primeiro de Junho – MG. 2005. 93f. Dissertação (Magister Scientiae em Extensão Rural). Departamento de Economia Rural, Universidade Federal de Viçosa, Viçosa. 2005.

GOMES, A. P. W. BUENO, N. P. Capital social e dilemas de ação coletiva: um estudo de caso em um pequeno assentamento rural de Minas Gerais. *Ruris*. Vol. 2, N. 2, 2008.

LANSING, J. S. **Perfect Order**: recognizing complexity in Bali. New Jersey, Princeton University Press. 2006.

MACY, M. WILLER, R. From factors to actors: computational sociology and agent-based modeling. **Annual review of Sociology**, vol. 28, 2002.

MATOS, A. G. Capital social e autonomia. NEAD, **Ministério do desenvolvimento Agrário**. 2002 Disponível em:

<http://www.nead.gov.br/index.php?acao=biblioteca&publicacaoID=227>

NORTH, D. C. Institutions. **The Journal of Economic Perspectives**. Vol. 5, n. 1, 1991.

OLSON, M. **The Rise and Decline of Nations**. New Haven: Yale University Press. 1982.

OSTROM, E. Collective Action and the Evolution of Social Norms. **The Journal of Economic Perspectives**. Vol. 14, n. 3, 2000. pp. 137-158.

_____. **Understanding Institutional Diversity**. New Jersey: Princeton University Press, 2005.

_____. The New Institutional Economics: Taking Stock, Looking Ahead. **Journal of Economic Literature**. Vol. 38, n. 3, 2000. pp. 595-613.

POTEETE, A. R. JANSSEN, M. A. OSTROM E. **Working Together**: Collective Action, the Commons, and Multiple Methods in Practice. New Jersey: Princeton University Press, 2010.

PUTNAM, R. D. Comunidade e democracia. Rio de Janeiro: Editora Fundação Getúlio Vargas, 1996. p. 257.

RAUCH, J. Seeing around corners. **The Atlantic Monthly**. 2002

WILLIAMSON, O. E. The New Institutional Economics: Taking Stock, Looking Ahead. **Journal of Economic Literature**. Vol. 38, n. 3, 2000. pp. 595-613.

7. APÊNDICE 1

7.1 MODELO DE COOPERAÇÃO

Para a elaboração desta programação foi utilizado o software *NetLogo*®. Neste programa existem três guias, Interface: onde é possível colocar botões para iniciar e parar o modelo, monitores para acompanhar o progresso de determinada variável, gráficos, além da visualização do andamento do modelo; Information: nessa guia é possível detalhar o funcionamento do modelo, para auxiliar quem for utilizar posteriormente o modelo, tais informações não interferem no funcionamento do modelo; e Procedure: nesta guia está contida toda a programação do modelo, é neste local que estão contidas a programação para a criação e execução do modelo.

Toda programação apresentada aqui se refere à seguinte matriz de *payoffs*, que é a matriz para o caso padrão:

$$\begin{array}{c} C \\ D \end{array} \begin{array}{cc} C & D \\ \left(\begin{array}{cc} (5,5) & (-6,6) \\ (6,-6) & (-5,-5) \end{array} \right) \end{array}$$

Para outras matrizes de *payoffs* a única alteração na programação será o ajuste de R, S, T e P no procedimento *update-globals*, para os novos valores da matriz.

7.1.1 PROGRAMAÇÃO DO MODELO DE COOPERAÇÃO

```
globals [ R T S P ]      (indica quais variáveis são de uso geral do modelo)

turtles-own [             (indica quais variáveis podem ser usadas apenas pelos agentes)
  resource
  cooperation ;; se cooperativo= 0, se nao cooperativo=1
  λ
]

to setup
  ca                      (cria os agentes do modelo)
  update-variables
  ask n-of (number * 0.5) patches
  [ sprout 1
    [ set λ ((R / (T - R)))
      setup-turtles ]
  ]
  ask n-of (number * 0.4) patches
  [ sprout 1
    [ set λ 0.75
      setup-turtles ]
  ]
end
```

```

ask n-of (number * 0.1) patches
  [ sprout 1
    [ set  $\lambda$  random-float ((R / (T - R)))
      setup-turtles]
  ]
do-plots          (cria os gráficos do modelo de agente)
end

to update-variables  (atualiza a cada rodada, os valores das variáveis)
  update-globals
  update-turtles
end

to update-globals  (ajusta o valor das variáveis globais)
  set R 5
  set T 6
  set S -6
  set P -5
end

to update-turtles  (torna os agentes cooperativos ou não cooperats)
  ask turtles [ ifelse random 101 <= (100 - (min (list 1 ( $\lambda$  * ((T - R) / R))) * 100))
    [set cooperation 0]
    [set cooperation 1] ]
end

to setup-turtles  (torna os agentes cooperativos e não cooperativos)
  move-to one-of patches with [ not any? other turtles-here ]  quando de sua criação
  set resource 6  (atribui um valor de recurso inicial ao agente)
  ifelse random 101 <= (100 - (min (list 1 ( $\lambda$  * ((T - R) / R))) * 100))
    [set cooperation 0]
    [set cooperation 1]
  recolor
end

to recolor  (torna os agentes cooperativos vermelhos, e não cooperativos amarelo)
  ifelse (cooperation = 0)
    [set color red]
    [set color yellow]
end

to go  (inicia os interações do modelo)
  update-variables
  ask turtles [
    move
    interaction
    reproduction
    recolor
    if resource <= 0 [
      die
    ]
  ]
  tick
  do-plots

```

```

end

to move          (movimenta o agente para um local desocupado)
  rt random-float 360
  fd random-float 1
  if any? other turtles-here
    [ move ]
  move-to patch-here
end

to interaction   (faz os agentes se interagirem)
  ifelse ( cooperation = 0 )
  [ let coop ((count (turtles-on neighbors4) with [ cooperation = 0 ] ) * R)
    let ncoop ((count (turtles-on neighbors4) with [ cooperation = 1 ] ) * S)
    set resource (resource + coop + ncoop) ]
  [ let coop1 ((count (turtles-on neighbors4) with [ cooperation = 0 ] ) * T)
    let ncoop1 ((count (turtles-on neighbors4) with [ cooperation = 1 ] ) * P)
    set resource (resource + coop1 + ncoop1) ]
end

to reproduction (faz os agentes se reproduzirem)
  let seedpatch one-of neighbors with [ not any? turtles-here ]
  if ( seedpatch != nobody ) and (resource >= 10)
    [ hatch 1 [ move
              set resource 6]]
end

to do-plots     (cria os gráficos)
  set-current-plot-pen "Cooperativo"
  plot count turtles with [color = red]
  set-current-plot-pen "Não Cooperativo"
  plot count turtles with [color = yellow]
end

```

7.2 DETALHANDO OS PROCEDIMENTOS DA PROGRAMAÇÃO

Cada procedimento é iniciado com o comando *to* e finalizado com o comando *end*. Existem dois procedimentos básicos no modelo, *setup* e *go*. O procedimento *setup* cria a estrutura inicial do modelo, construindo o ambiente e as *turtles*, atribuindo as características a cada um deles, e posicionando cada *turtle* em um determinado *patch*. É no procedimento *go* que ocorrem as interações e mudanças do modelo ao longo do tempo.

Existem dois tipos de variáveis em uma programação de um modelo baseado em agente, as variáveis globais e as locais. As variáveis globais podem ser chamadas em qualquer procedimento, o seu valor pode ser ajustado em um determinado procedimento e esta variável pode ser usada em qualquer outro local. Já as variáveis

locais tem seu valor ajustado em um procedimento e esta variável só pode ser utilizada dentro deste mesmo procedimento.

No *NetLogo*® para ajustar os valores das variáveis globais é utilizado o comando *set*, enquanto para atribuir os valores para as variáveis locais é utilizado o comando *let*.

O primeiro passo que deve ser seguido para a construção do modelo é a determinação das variáveis globais:

- `globals [R T S P]`

No caso, as variáveis globais são os valores da matriz de payoffs. Em seguida são identificadas as características dos agentes:

- `turtles-own [resource cooperation ;; se cooperativo= 0, se nao coopertivo=1 λ]`

No item acima, a palavra *cooperation* é seguida de dois ponto e vírgula, durante a programação é possível explicar ou realizar algum comentário e para isso é utilizado esses dois ponto vírgula. Toda a informação contida após este símbolo não é desconsiderada pelo programa.

Se os *patches* tivessem alguma característica seria colocado também o procedimento *patches-own*. Estes procedimentos, *globals*, *turtles-own* e *patches-own*, são exceção à regra de se iniciar os procedimentos com *to* e finalizar com *end*. Além disso as variáveis destes três procedimentos são todas globais, a diferença é que *globals* são para variáveis gerais do modelo, *turtles-own* são para variáveis referentes a *turtles* e *patches-own* são para as variáveis globais referentes aos *patches*.

O próximo procedimento (*setup*) é onde o modelo é criado, o procedimento *setup* é iniciado com o comando *ca*, que é responsável por limpar todo o modelo toda vez que este for reiniciado. Em seguida vem o comando *update-variables*, que chama outro procedimento:

- `to setup`

```
ca
update-variables
```

Posteriormente, ainda dentro do *setup*, vêm os comandos que criam os agentes, o primeiro comando *ask* escolhe metade de *number* (que é o número total de *turtles* que o modelo será iniciado) para serem os *patches* que serão ocupados pelas *turtles*. Então em cada um dos *patches* selecionados será criada uma *turtle* (através do comando *sprout 1*). Logo após são ajustadas as características das *turtles*. É ajustado o valor de λ e é chamado o comando *setup-turtles* que chama outro procedimento que ajusta as outras características das *turtles*:

- ask n-of (number * 0.5) patches
 [sprout 1
 [set λ ((R / (T - R)))
 setup-turtles]
]
ask n-of (number * 0.4) patches
 [sprout 1
 [set λ 0.75
 setup-turtles]
]
ask n-of (number * 0.1) patches
 [sprout 1
 [set λ random-float ((R / (T - R)))
 setup-turtles]
]
do-plots
end

Os outros dois *asks* selecionam mais patches, 40% e 10% do valor de *number* respectivamente, criam as *turtles* nos *patches* selecionados e ajustam suas características, assim como realizado pelo primeiro comando *ask*. E o *setup* é finalizado com o comando *do-plot*, que chama outro procedimento que será detalhado posteriormente.

O próximo procedimento é o *update-variable*, chamado pelo *setup* e em outros pontos da programação também:

- to update-variables
 update-globals
 update-turtles
end

Este procedimento chama mais dois comandos, *update-globals* e *update-turtles*. O procedimento *update-globals* é dado como se segue:

- to update-globals
set R 5
set T 6
set S -6
set P -5
end

Neste procedimento são ajustados os valores da matriz de *payoffs*. Já o procedimento *update-turtles* vai dizer se o agente é cooperativo ou não em cada roda do jogo:

- to update-turtles
ask turtles [ifelse random 101 <= (100 - (min (list 1 (1 * ((T - R) / R))) * 100))
[set cooperation 0]
[set cooperation 1]]
end

A cooperação do agente vai depender de uma probabilidade. Um número vai ser escolhido aleatoriamente entre 0 e 100, através do comando *random 101* (é colocado 101 uma vez que o comando *random* seleciona aleatoriamente um número inteiro positivo menor que o indicado, no caso, menor que 101), se a função “ $(100 - (\min (\text{list } 1 (1 * ((T - R) / R))) * 100))$ ” para este agente for menor ou igual ao valor aleatório escolhido nesta dada rodada, o agente será cooperativo, caso contrario o agente será não cooperativo.

Um outro procedimento que foi chamando no setup foi o *setup-turtles*:

- to setup-turtles
move-to one-of patches with [not any? other turtles-here]
set resource 6
ifelse random 101 <= (100 - (min (list 1 (1 * ((T - R) / R))) * 100))
[set cooperation 0]
[set cooperation 1]
recolor
end

A primeira linha deste comando faz com que as *turtles* ao serem criadas sejam alocadas em um *patch* vazio. A segunda linha atribui o valor inicial de 6 unidades de recurso para as *turtles*. A terceira, quarta e quinta linha atribui a cooperação ou não ao agente. E a sexta linha chama um procedimento, *recolor*.

O procedimento *recolor* é dado como se segue:

- to recolor
 ifelse (cooperation = 0)
 [set color red]
 [set color yellow]
 end

Neste procedimento são atribuídas cores aos agentes de acordo com o valor da cooperação, se esta for 0, ou seja, o agente for cooperativo, o indivíduo será vermelho, caso contrário, será amarelo.

O próximo procedimento é o *go*:

- to go
 update-variables
 ask turtles [
 move
 interaction
 reproduction
 recolor
 if resource <= 0 [
 die
]
]
 tick
 do-plots
 end

No *go* a primeira ordem que aparece é o comando que chama o *update-variable*, isto se faz necessário para que em cada rodada de simulação sejam atualizadas as variáveis da matriz de *payoffs*, que no caso são fixas, e também sejam atualizados os valores para a cooperação. Com essa atualização a cada rodada, e dada a probabilidade atribuída a cada agente, um dado indivíduo que na rodada anterior foi cooperativo, na rodada atual pode ser não cooperativo.

O comando seguinte do procedimento *go* é o *ask turtles*, neste comando existe mais quatro comandos, *move*, *interaction*, *reproduction* e *recolor*, além disso, esse comando é finalizado impondo que se o valor de recurso para o agente naquela rodada for menor ou igual a zero ele morre. Os outros dois comandos do *go* são *tick*: que é um contador de rodadas; e *do-plots*, que chama outro procedimento à ser detalhado posteriormente.

Os procedimentos *move*, *interaction* e *reproduction*, chamados por *go*, seguem a Regra de Movimento, a Regra de Interação e a Regra de Reprodução, respectivamente, todas estas descritas na metodologia deste trabalho.

O último procedimento do modelo é o do-plot, chamado tanto por go quanto por setup.

8. APÊNDICE 2

8.1 MODELO DE IRRIGAÇÃO

O modelo aqui apresentado é uma representação de como modelar através de agent-based modeling um caso real, utilizando informações coletadas em pesquisas de campo.

Com base na imagem do local é possível a criação de um ambiente virtual similar ao real. Para tanto, para captar as diferentes características do ambiente, será diferenciado as zonas pela quantidade de recursos existente. Com a utilização do programa Microsoft Excel é possível alocar uma quantidade de recurso para cada célula, sabendo o tamanho desejado do ambiente monta-se uma tabela onde cada célula representará um *patch* do ambiente no *NetLogo*. No modelo, o número que está em cada célula representa a quantidade de recursos existentes em cada *patch*, sendo este número associado com a tonalidade que se deseja colocar no *patch*.

Desse modo, programa-se o procedimento *to setup-patches*, contendo o caminho para chegar ao arquivo feito no Excel com as informações do ambiente. Este arquivo está em formato *.txt*, significando que este é separado por tabulações, que é o modo em que o programa *NetLogo* reconhece os arquivos utilizados para a programação, no caso, para a formação do ambiente. Isso foi feito para dar ao ambiente diferentes níveis de recursos nos *patches*. Assim, coloca-se o nível inicial de recursos em cada *patch* desse modelo, tendo o arquivo o nome *Rio.txt*. Deve inserir também dentro desse procedimento o comando *patch-recolor*, que dará ao ambiente a cor que foi definida no procedimento *to patch-recolor* (definido posteriormente).

O *to setup-patches* conterà o caminho para se chegar ao arquivo que representa o ambiente e, o abrirá. Esse arquivo é a tabela que contém a quantidade inicial de recursos em cada *patch*, sendo que esta quantidade é máxima. O comando a lerá como a quantidade máxima e dará a ela o nome de *presource*, que significa a quantidade de

recursos no *patch*. Como o ambiente não é homogêneo devido à diferenciação da quantidade de recursos nos *patches* é necessário fazer com que essas diferenças sejam visíveis. Para isso, serão dadas cores distintas aos *patches* de acordo com a quantidade de recursos em cada um deles, ou seja, *patches* com quantidades de recursos iguais terão a mesma cor, ao contrário dos que não tem a mesma quantidade de recursos, que terão a cor diferente.

```
to setup-patches
  set-current-directory "E:\\ "
  file-open "Rio.txt"
  foreach sort patches [
    ask ? [
      set max-presource file-read
      set presource max-presource
      patch-recolor
    ]
  ]
  file-close
end
```

- to setup-patches: procedimento para a criação dos *patches*
- set-current-directory "E:\\ ": comando para indicar o caminho para a localização do arquivo que está no Excel
- file-open "Rio.txt": abrir o arquivo desejado
- foreach sort patches : lista de *patches* a serem escolhidos
- ask ? [: pergunta
- set max-presource file-read: ler colocando o máximo de recurso de cada *patch*
- set presource max-presource: chamar o comando anterior de *presource*, quantidade de recursos no *patch*
- patch-recolor: colorir o *patch*
- file-close: fechar o arquivo que foi aberto anteriormente
- end: fim do procedimento

Agora o procedimento para colorir o ambiente será criado. De acordo com o modelo, o ambiente terá a cor verde com gradações na tonalidade, de acordo com a quantidade de recursos existentes em cada *patch*, e o rio será da cor azul. O *NetLogo*

disponibiliza uma caixa com várias cores e suas tonalidades, cada uma com um número correspondente, para assim criar o comando e definir as cores desejadas através desse número.

Este procedimento será o *to patch-recolor*. A cor de cada *patch* será definida a partir da quantidade de recursos existentes nele. A cor base será um verde cuja tonalidade é representada pelo número 69. A cor de cada *patch* será definida como esse número (69) menos a quantidade de recursos que tem no *patch*. Por exemplo, se o *patch* tem a quantidade de recursos igual a 2, sua cor será $69-2 = 67$, ou seja, será colorido pela cor correspondente ao número 67. Os *patches* de tiverem o número correspondente à cor menor que 65, passarão a ter a cor cujo número é 94. Essa nova cor é um azul que representará o rio.

```
to patch-recolor
  set pcolor (69 - presource)
  ask patches with [ pcolor < 65 ] [ set pcolor 94 ]
end
```

- to patch-recolor: procedimento para colorir o ambiente
- set pcolor (69 - presource): colocar a cor nos *patches* de acordo com a fórmula. O número 69 representa a tonalidade mais clara da cor verde. Para colorir cada *patch* de acordo com os seus recursos disponíveis faz-se essa subtração, onde a cor do *patch* será 69 menos a quantidade de recurso que esse *patch* têm.
- ask patches with [pcolor < 65] [set pcolor 94]: *patches* com valor da cor menor que 65 serão coloridos da cor 94, que é azul, pois é o rio.
- end: fim do procedimento

O ambiente não tem uma taxa de regeneração completa, ou seja, ele não volta ao que era antes, não será 100% igual. Ele tem uma taxa de regeneração que é descrita pelo procedimento *to patch-growback*, definindo a quantidade de recursos no *patch* a cada *tick*, ou seja, o quanto há de regeneração no ambiente por período. O comando pega uma lista com a quantidade máxima de recursos que é calculada da seguinte maneira: será uma taxa fixa de 0,05 mais a quantidade de recursos que tem no *patch* multiplicada por 1,3, ou seja, a taxa de regeneração do ambiente é de 30%. A

quantidade mínima de recursos que tem no *patch* será essa quantidade que foi calculada. O comando *min* é utilizado para limitar o valor de recurso do em seu valor máximo, assim se $(0,05 + (presource*1,3))$ ultrapassar o valor máximo de recurso especificado para o *patch* (*Max-presource*), então o valor o recurso será o estabelecido pelo comando *Max-presource*.

```
patch-growback
  set presource min (list max-presource ( 0.05 + ( presource * 1.3)))
end
```

- to patch-growback: a regeneração do ambiente
- set presource min (list max-presource (0.05 + (presource * 1.3))): a nova quantidade de recursos no *patch* é dada por uma quantidade fixa de 0,05 mais a quantidade de recursos do *patch* vezes 1,3, ou seja, a taxa de regeneração do ambiente é de 30%
- end: fim do procedimento

O ambiente tem suas características e essas devem ser especificadas em um procedimento chamado *patch-own*, que conterà todos os comandos que serão realizados no *patch*, definindo as suas variáveis. As características do *patch* são a quantidade de recursos existentes nele, a sua quantidade máxima de recursos e a quantidade de recursos que é retirada dele pelo indivíduo que o ocupa, retirada essa que é feita devido ao metabolismo do indivíduo, sendo equivalente ao valor do metabolismo. Este procedimento deve necessariamente estar no começo da programação, como forma descritiva.

```
patches-own [
  max-presource
  presource
  pconsumptio
]
```

Assim, com base nessas informações é criado o ambiente virtual onde o modelo será simulado, e é indicado pela figura A2 abaixo:

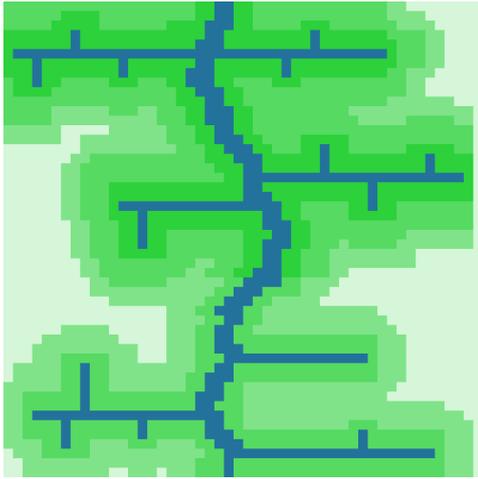


Figura A2 – Ambiente virtual

Fonte: Resultado da pesquisa

Com o ambiente criado, o passo seguinte é a criação dos indivíduos representativos às duas comunidades.

O comando *breed* é utilizado para que o programa possa diferenciar as raças. No modelo há somente uma raça, porém dividida em dois grupos. Esse criará a raça humana, criando cada indivíduo e a população.

```
breed [person people]
```

- `breed [person people]`: criação das pessoas (`person`) e da população (`people-conjunto de pessoas`)

Após a criação dos indivíduos são definidas suas características, assim como foi feito com o ambiente. Essas características são chamadas de variáveis globais, pois envolvem todos os indivíduos do modelo. As características dos indivíduos são quantidade de recursos, metabolismo, nível de visão, pontos de visão, produção e consumo. Sendo que dessas características somente a quantidade de recursos e a produção variam, enquanto as outras são fixas.

```
turtles-own [  
  resource  
  metabolism  
  vision  
  vision-points  
  production  
  consumption
```

-]
- turtles-own : características dos indivíduos
 - resource: quantidade de recursos
 - metabolism : metabolismo
 - vision: nível de visão
 - vision-points: alcance da visão
 - production: produção
 - consumption: consumo

Como se trata de duas populações distintas é criado um comando para cada uma, para assim, ser possível diferenciá-las. A diferença entre as duas populações está na cor (sendo uma *black* e a outra *red*) e no metabolismo (que no nosso modelo é entre 2 e 6 para a população *black* e entre 1 e 4 para a população *red*). O nível de visão foi considerado o mesmo para ambas, mas se quiser pode alterar, assim como também o metabolismo pode ser alterado.

8.1.1 Criação da população 1

Nesse procedimento são definidas todas as características dos indivíduos desse grupo, como também as atividades que eles exercerão. Como foi dito anteriormente, os indivíduos terão a mesma raça, tendo a forma de pessoas, sendo diferenciados pela cor. Essa primeira população terá a cor preta. Cada grupo se localizará inicialmente em uma região limitada do ambiente. Para que eles sejam dispostos corretamente devemos indicar as coordenadas que correspondem a essa localização. No nosso modelo os indivíduos desse grupo serão fixados inicialmente em locais vazios, exceto o rio, entre as coordenadas 3 e 14 do eixo Y, sendo esse limite vertical (horizontalmente eles ficam em qualquer lugar). A quantidade inicial de recursos será dada aleatoriamente variando entre 1 e 20, assim como o metabolismo entre 2 e 6 e o nível de visão entre 1 e 3. Eles se caracterizarão pela ausência de visão diagonal, ou seja, a visão será limitada aos quatro pontos principais, norte, sul, leste e oeste. Os possíveis pontos de visão serão dados por uma lista com as coordenadas dos *patches*

e, o indivíduo escolherá o melhor *patch* dessa lista para se mover. Uma forma de o indivíduo adicionar riqueza na forma de recursos é através da produção, que é possível pela prática da agricultura. Essa produção será função da quantidade de recursos do indivíduo e da presença ou não de um vizinho para interagir. Quando o indivíduo interage com seu vizinho a produção dele é dez vezes maior que quando ele está só. Para cada uma dessas características há um comando específico.

```

to setup-person
  set color black
  set shape "person"
  move-to one-of patches with [ (not any? other person-here) and (pcolor != 94) and ( pycor >
3 and pycor < 14 ) ]
  set resource random-in-range 1 20
  set metabolism random-in-range 2 6
  set vision random-in-range 1 3
  set vision-points []
  foreach n-values vision [? + 1]
  [
    set vision-points sentence vision-points (list (list 0 ?) (list ? 0) (list 0 (- ?)) (list (- ?) 0))
  ]
  agriculture
end

```

- to setup-person
- set color black: a cor dessa população será preto
- set shape "person": o formato da população será “person” (pode-se ter vários formatos diferentes, para achá-los vai em Tools – Turtle Shapes Editor)
- move-to one-of patches with [(not any? other person-here) and (pcolor != 94) and (pycor > 3 and pycor < 14)]: comando para informar o movimento da *turtle*. Nesse caso ela vai se mover para um *patch* onde não haja outra *turtle* ali, exceto para os *patches* da cor 94, que é o rio. Além disso, irão se mover entre as coordenadas 3 e 14 do eixo Y.
- set resource random-in-range 1 20: distribuir aleatoriamente os recursos para cada indivíduo, sendo que a quantidade desses recursos vai variar entre 1 e 20.
- set metabolism random-in-range 2 6: distribuir aleatoriamente o metabolismo para cada indivíduo, com este variando entre 2 e 6

- set vision random-in-range 1 3: distribuir aleatoriamente a visão para cada indivíduo, com esta variando entre 1 e 3
- set vision-points []: comando que indica que o ponto de visão é limitado à horizontal e vertical, ou seja, a turtle não pode ver as diagonais
- foreach n-values vision [? + 1]: comando que gera uma lista de possíveis pontos de visão
- set vision-points sentence vision-points (list (list 0 ?) (list ? 0) (list 0 (- ?)) (list (- ?) 0)): os pontos de visão são uma lista, que indica coordenadas
- agriculture: característica da população que será descrita mais a frente
- end: fim do procedimento

8.1.2 Criação da população 2

A criação da segunda população segue os mesmos passos e tem as mesmas características do primeiro grupo, com algumas diferenças. Os dois grupos se distinguem pela cor, sendo a deste grupo a cor vermelha. A localização inicial também difere, com estes se localizando entre as coordenadas 31 e 40 do eixo Y, mas mantendo a movimentação para um lugar vazio, exceto o rio. A quantidade de recursos, o nível de visão, os pontos de visão e a prática da agricultura seguem o mesmo procedimento. O que também difere é o metabolismo, que deste grupo varia entre 1 e 4.

```

to setup-person2
  set color red
  set shape "person"
  move-to one-of patches with [ (not any? other person-here) and (pcolor != 94) and ( pycor >
30 and pycor < 41 ) ]
  set resource random-in-range 1 20
  set metabolism random-in-range 1 4
  set vision random-in-range 1 3
  set vision-points []
  foreach n-values vision [? + 1]
  [
    set vision-points sentence vision-points (list (list 0 ?) (list ? 0) (list 0 (- ?)) (list (- ?) 0))
  ]

```

```
agriculture
end
```

O procedimento seguinte, *to person-move*, descreve a regra de movimento dos agentes. Como já foi mencionado desde o início, eles irão se moverão para o *patch* que esteja vazio (exceto o rio) e que contenha a maior quantidade de recursos, tendo que sua visão é limitada, ou seja, o *patch* obrigatoriamente estará dentro do seu campo de visão. Terão os *patches* que serão os possíveis vencedores, o vencedor será aquele que tiver o máximo de recursos e que estiver a uma distância mínima do indivíduo.

```
to person-move
  let move-candidates (patch-set patch-here (patches at-points vision-points) with [ (not any?
person-here) and (pcolor != 94)])
  let possible-winners move-candidates with-max [presource]
  if any? possible-winners [
    move-to min-one-of possible-winners [distance myself]
  ]
end
```

- *to person-move*: procedimento da movimentação do indivíduo
- *let move-candidates (patch-set patch-here (patches at-points vision-points) with [(not any? person-here) and (pcolor != 94)])*: uma lista com os possíveis *patches* será dada para esse *patch*, e a *turtle* se moverá para um *patch* que estiver vazio e que não seja o rio, isso de acordo com o seu ponto de visão.
- *let possible-winners move-candidates with-max [presource]*: dos possíveis vencedores (dados na lista), a *turtle* se moverá para o *patch* que tiver o máximo de recursos
- *if any? possible-winners*: se forem verdadeiros os possíveis vencedores
- *move-to min-one-of possible-winners [distance myself]*: a *turtle* moverá para um *patch* (desses possíveis) que tiver o mínimo de distância dela mesma
- *end*: fim do procedimento

Para sobreviver o indivíduo necessita consumir recursos. Este seu consumo é equivalente ao valor do seu metabolismo. Inicialmente é dada ao indivíduo uma quantidade de recursos aleatoriamente. É possível a acumulação de recursos por parte do indivíduo, pois além de a este ser dada uma quantidade inicialmente, ele é capacitado da habilidade de produzir que lhe acresce recursos, apesar da quantidade que o seu metabolismo consome.

O procedimento *to person-consumption* diz que o consumo do indivíduo será igual ao seu metabolismo, ou seja, se o indivíduo tem metabolismo 3, a quantidade de recursos consumida por ele será igual a 3 também. A nova quantidade de recursos do indivíduo dependerá do valor de recursos do *patch* e do consumo do indivíduo. Se a quantidade de recursos no *patch* for maior que o consumo do indivíduo, a nova quantidade de recursos dele será sua quantidade anterior menos o seu metabolismo mais o seu consumo, que é igual ao metabolismo. Porém, se a quantidade de recursos no *patch* for menor que o consumo do indivíduo, ele consumirá tudo o que tem no *patch*, sendo a nova quantidade de recursos do indivíduo igual à quantidade anterior menos o seu metabolismo mais a quantidade de recursos do *patch*, que foi totalmente consumida por ele, por ser menor que seu metabolismo. A quantidade de recursos retirada do *patch* (*pconsumption*) é igual ao consumo do indivíduo, que é o seu metabolismo. O máximo de recursos existentes em um *patch* é calculado pela diferença da quantidade que tinha de recursos (*presource*) e a quantidade que foi retirada (*pconsumption*) pelo consumo do indivíduo. Quando essas quantidades forem equivalentes, os recursos do *patch* são esgotados e, espera-se a próxima rodada para a regeneração do *patch*.

```
to person-consumption
  set consumption metabolism
  set resource ifelse-value ( presource > consumption )
  [ resource - metabolism + consumption ]
  [ resource - metabolism + presource ]
  set pconsumption consumption
  set presource max (list 0 ( presource - pconsumption ) )
end
```

- to person-consumption

- set consumption metabolism: o consumo do indivíduo é o seu próprio metabolismo
- set resource ifelse-value (presource > consumption): indica o novo valor de recursos do indivíduo que será dado por uma condição que, no caso, refere-se ao recurso do *patch* ser maior que o consume do indivíduo. Caso isso seja verdadeiro, o valor do recurso do indivíduo será o recurso que ele já tinha (*resource*) menos o seu metabolismo (*metabolism*) mais o que ele consumiu do *patch* (*consumption*, que é o seu metabolismo). Caso contrário, ou seja, caso o valor de recursos existente no *patch* seja menor do que o consumo que o indivíduo necessita (que é seu metabolismo), ele irá consumir todo o recurso do *patch*.
- [resource - metabolism + consumption]: valor do recurso do indivíduo caso a condição seja verdadeira
- [resource - metabolism + presource]: valor do recurso do indivíduo caso a condição seja falsa
- set pconsumption consumption: o que foi retirado do *patch* (*pconsumtion*) é igual ao consumo do indivíduo
- set presource max (list 0 (presource - pconsumption)): a quantidade de recursos máxima do *patch* será a quantidade de recursos que ele tinha antes do consumo do indivíduo menos o que foi consumido. Esta será zero quando ambos forem iguais.
- End: fim do procedimento

Uma importante característica do indivíduo é a capacidade de praticar a agricultura, possibilitando assim a produção. Essa produção é função da interação com um vizinho e da quantidade de recursos do indivíduo. A quantidade de recursos é diretamente relacionada com a produção, ou seja, quanto mais recursos têm o indivíduo maior a sua produção, o inverso também é válido. A relação entre a produção e a interação com um vizinho é que, quando o indivíduo interage com seu vizinho e ambos produzem, essa produção é maior que quando o indivíduo a pratica sozinho. Consideramos que quando a produção é entre o indivíduo e seu vizinho ela

equivale à quantidade de recursos do indivíduo multiplicada por 100. Já quando ele está sozinho essa produção é dez vezes menor sendo, a quantidade de recursos do indivíduo multiplicada por dez.

```
to agriculture
  if any? other neighbors [
    set production (resource * 100) ]
  set production (resource * 10)
end
```

- to agriculture: procedimento para a produção
- if any? other neighbors: indica uma condição que, no caso, é a existência de outros vizinhos. Se a condição for verdadeira, ou seja, se existirem outros vizinhos, a produção do indivíduo será dada pela primeira forma (*resource * 100*). Caso contrário, a produção será dada pela outra forma (*resource * 10*).
- set production (resource * 100)]: primeira forma, que indicará a produção caso a condição seja verdadeira. A produção será a quantidade de recursos do indivíduo vezes 100.
- set production (resource * 10): segunda forma, que indicará a produção caso a condição seja falsa. A produção será a quantidade de recursos do indivíduo vezes 10.
- end: fim do procedimento

Durante a programação foi usado várias vezes o comando *random-in-range*. Esse comando é utilizado quando queremos que uma variável seja escolhida aleatoriamente dentro de uma faixa, que indica o valor mínimo e máximo. Por exemplo, no caso da quantidade de recursos do indivíduo, sabemos que essa quantidade varia entre 1 e 20, sendo essa a faixa. Deseja-se que a quantidade de recursos seja dada ao indivíduo aleatoriamente e, obrigatoriamente essa quantidade será um valor dentro da faixa, ou seja, variará entre 1 e 20. O comando consegue captar o valor mais baixo e o mais alto, então é necessário somar mais um para que os limites da variação realmente

sejam os valores que colocamos no comando. Será reportado o valor mais baixo somando-o com o resultado aleatório da diferença entre o mais alto e o mais baixo mais um. Sempre que o comando *random-in-range* for utilizado, necessariamente o procedimento a seguir deve ser programado.

```
to-report random-in-range [low high]
  report low + random (high - low + 1)
end
```

Para ter uma visualização melhor do que está acontecendo há o auxílio de gráficos e monitores. Na programação de gráficos todos são feitos da mesma forma, o que muda são as variáveis consideradas. No caso, foi criado um gráfico para a população, para poder enxergar melhor a variação na quantidade de indivíduos dos dois grupos. Para isso é dado um nome ao gráfico e define-se as variáveis. Como são dois grupos distintos define-se uma linha com a cor diferente para cada um desses grupos.

```
to update-plots
  set-current-plot "Population"
  set-current-plot-pen "red"
  plotxy ticks count person with [color = red]
  set-current-plot-pen "black"
  plotxy ticks count person with [color = black]
end
```

- to update-plots: atualização gráfica
- set-current-plot "Population": nome do gráfico
- set-current-plot-pen "red": uma das linhas do gráfico é vermelha
- plotxy ticks count person with [color = red]: colocar no gráfico a quantidade de indivíduos vermelhos à cada rodada (tick)
- set-current-plot-pen "black": a outra linha do gráfico é preta
- plotxy ticks count person with [color = black]: colocar no gráfico a quantidade de indivíduos pretos à cada rodada (tick).
- end: fim do procedimento

8.2 CRIANDO O *TO SETUP* E O *TO GO*

Assim como no modelo de cooperação descrito no Apêndice 1, aqui também existem dois procedimentos fundamentais, o *to setup* e o *to go*, já que é neles que colocamos as características iniciais e os comandos que serão acionados a cada rodada do modelo.

O *to setup* descreve a configuração do modelo, ativando os comandos que criam o ambiente, a população e suas características iniciais. Quando utiliza-se gráficos é necessário escrever o comando neste procedimento, para que desde a rodada inicial possa ser visualizado. Um importante comando é o *ca*, ou seja, o *clear all*. Este comando faz com que ao final de cada *tick* tudo seja zerado, levando todas as variáveis para zero.

No modelo programou-se o *to setup* colocando primeiramente o comando *clear all*, que limpa todas as informações do modelo. Como esse procedimento configura o modelo acrescentam-se os comandos referentes à criação do ambiente (*setup-patches*) e dos indivíduos, que são dois grupos (*setup-person* e *setup-person2*), assim como o comando para a criação do gráfico.

Para este procedimento foi criado um botão chamado **SETUP**. Quando clicar no botão **SETUP** tudo o que foi descrito no procedimento será ativado.

```
to setup
  ca
  setup-patches
  create-person initial-population [ setup-person ]
  create-person initial-population [ setup-person2 ]
  update-plots
end
```

- *to setup*: nome do procedimento
- *ca*: a cada rodada ele zera tudo

- `setup-patches`: criação dos patches
- `create-person initial-population [setup-person]`: criação da população 1
- `create-person initial-population [setup-person2]`: criação da população 2
- `update-plots`: criação de gráficos
- `end`

Enquanto o procedimento *to setup* dá as configurações iniciais do modelo, o procedimento *to go* dá as configurações e alterações que ocorrerão durante cada rodada (*tick*). Como já foi descrito os agentes movimentam, consomem e produzem durante a vida. Essas atividades serão descritas aqui com o comando referente a cada uma delas. O movimento dos indivíduos é explicado pelo comando *person-move*, onde eles se moverão para um *patch* vazio, que não seja o rio e, que tenha a maior quantidade de recursos, sendo que o *patch* deve ser o mais próximo do indivíduo. O consumo é explicado pelo comando *person-consumption*, que indica que o consumo equivale ao metabolismo e nos dá a nova quantidade de recursos do indivíduo que depende da quantidade de recursos existentes no *patch*. A produção é definida pelo comando *agriculture*, onde a produção é função da quantidade de recursos do indivíduo e da interação ou não com os vizinhos. Como é sabido, se a quantidade de recursos do indivíduo for menor ou igual a zero este indivíduo morre, sendo essa a única causa de morte no modelo.

Assim como há alterações nos indivíduos há também no ambiente. O ambiente tem uma taxa de regeneração que é descrita pelo comando *patch-growback*, aonde vimos que ele regenera a uma taxa fixa de 0,05 unidades mais 30% da quantidade de recursos existentes no *patch*. O ambiente sofre alterações na sua cor, devido a mudanças na quantidade de recursos nos *patch*, já que a cada quantidade de recursos há uma tonalidade diferente associada.

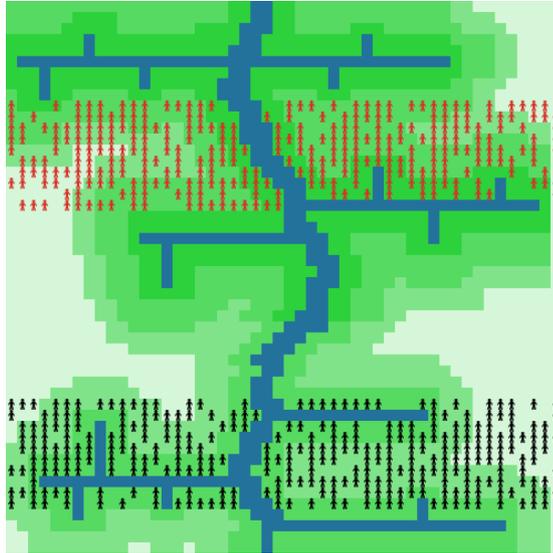
Esses comandos serão acionados a cada rodada e, para isso, é necessário escrever o comando *tick* nesse procedimento, indicando que deverão ser repetidos a cada *tick*. Os gráficos devem ser atualizados a cada rodada, por isso, não pode deixar de escrever o comando *update-plots*.

Para este procedimento criamos um botão chamado GO e, após clicar nele, este procedimento será acionado e se repetirá em cada uma das rodadas (*ticks*).

```
to go
  ask person [
    person-move
    person-consumption
    agriculture
    if resource <= 0 [ die ]
  ]
  diffuse pconsumption 0.5
  ask patches [
    patch-growback
    patch-recolor ]
  tick
  update-plots
end
```

- to go: procedimento que depois de ativado se repedirá em cada uma das rodadas (*ticks*).
- ask person: procedimentos que serão acionados e realizados pelos indivíduos
- person-move: movimentação do indivíduo
- person-consumption: consumo do indivíduo
- agriculture: produção do indivíduo
- if resource <= 0 [die]: indica que se o valor do recurso do indivíduo for menor ou igual a zero ele morrerá
- diffuse pconsumption 0.5: o que foi consumido do *patch* será difundido em uma taxa de 0,5
- ask patches: procedimentos que serão acionados e realizados no *patch*
- patch-growback: regeneração do ambiente
- patch-recolor: colorir o ambiente
- tick: rodadas
- update-plots: atualização dos gráficos a cada *tick* (rodada)
- end: fim do procedimento

Após o fim da programação clica-se no botão SETUP e tem-se a seguinte visualização:



Depois de clicar no botão SETUP, clica-se no botão GO e o modelo começa a rodar, sendo que cada período de tempo é representado por um *tick*. Quando se desejar que o modelo pare de rodar basta clicar novamente no botão GO.