

Fundação Getúlio Vargas
Escola de Pós-Graduação em Economia - EPGE

Dinâmica Monetária Eficiente sob Encontros Aleatórios: uma Classe de Métodos Numéricos que Exploram Concavidade

Dissertação submetida à Escola de Pós-Graduação em Economia
da Fundação Getúlio Vargas como requisito de obtenção do título de
Mestre em Economia

Aluno: Jefferson Donizeti Pereira Bertolai
Orientador: Ricardo de Oliveira Cavalcanti

Rio de Janeiro
Dezembro de 2009

Fundação Getúlio Vargas
Escola de Pós-Graduação em Economia - EPGE

Dinâmica Monetária Eficiente sob Encontros Aleatórios: uma Classe de Métodos Numéricos que Exploram Concavidade

Dissertação submetida à Escola de Pós-Graduação em Economia
da Fundação Getúlio Vargas como requisito de obtenção do título de
Mestre em Economia

Aluno: Jefferson Donizeti Pereira Bertolai

Banca Examinadora:

Ricardo de Oliveira Cavalcanti (EPGE-FGV/RJ)

Luis Henrique Braido (EPGE-FGV/RJ)

Heleno Martins Pioner (USP)

Rio de Janeiro
Dezembro de 2009

Agradecimentos

Primeiramente gostaria de agradecer à minha família por apoiar minhas escolhas profissionais e pessoais que fizeram de mim um candidato ao título de Mestre em Economia por uma escola como a EPGE - FGV, mesmo que isso a privasse de minha presença pessoal, como o foi nestes últimos três anos. Obrigado pelo apoio e carinho, com vocês divido esta conquista.

Agradeço a minha mulher e amiga de todas as horas, Karina Justino, com quem divido meus sonhos e futuro. Obrigado pelo apoio incondicional a minha opção pela carreira acadêmica e pelos sacrifícios que esta escolha nos impõe. Sua companhia torna tudo mais fácil e prazeroso.

Ao meu orientador, Ricardo Cavalcanti, agradeço o apoio e o incentivo. As longas e frequentes discussões têm sido fundamentais para meu amadurecimento como economista e pesquisador. Seus ensinamentos e pontos de vista, nem sempre usuais, catalisam meu interesse pela Ciência Econômica.

Agradeço aos amigos de curso de Mestrado pelo companheirismo nestes anos de intensa dedicação a formação acadêmica e pelas constantes e frutíferas discussões sobre Economia. Em especial agradeço aos amigos Marinho Bertanha, João Ayres, Leonardo Gomes, Marcus Studart, Vitor Farinha e Leonardo Silva.

Finalmente agradeço a EPGE-FGV a formação proporcionada, o ambiente de incentivo a pesquisa acadêmica do mais alto nível e o apoio logístico. Minha gratidão aos professores e funcionários que com sua dedicação garantem e engrandecem a qualidade da instituição.

Resumo

A dificuldade em se caracterizar alocações ou equilíbrios não estacionários é uma das principais explicações para a utilização de conceitos e hipóteses que trivializam a dinâmica da economia. Tal dificuldade é especialmente crítica em Teoria Monetária, em que a dimensionalidade do problema é alta mesmo para modelos muito simples. Neste contexto, o presente trabalho relata a estratégia computacional de implementação do método recursivo proposto por Monteiro e Cavalcanti (2006), o qual permite calcular a sequência ótima (possivelmente não estacionária) de distribuições de moeda em uma extensão do modelo proposto por Kiyotaki e Wright (1989). Três aspectos deste cálculo são enfatizados: (i) a implementação computacional do problema do planejador envolve a escolha de variáveis contínuas e discretas que maximizem uma função não linear e satisfaçam restrições não lineares; (ii) a função objetivo deste problema não é côncava e as restrições não são convexas; e (iii) o conjunto de escolhas admissíveis não é conhecido a priori. O objetivo é documentar as dificuldades envolvidas, as soluções propostas e os métodos e recursos disponíveis para a implementação numérica da caracterização da dinâmica monetária eficiente sob a hipótese de encontros aleatórios.

Palavras-Chave: Alocações Monetária não-Estacionárias, Economia Computacional, Métodos Recursivos, MINLP.

Abstract

The difficulty in characterizing non-stationary allocations or equilibria is one of the main explanations for the use of concepts and assumptions that trivialize the dynamics of the economy. This difficulty is especially critical in Monetary Theory, in which the dimensionality of the problem is high even for very simple models. In this context, this paper reports the computational strategy for implementing the recursive method proposed by Monteiro and Cavalcanti (2006), which allows you to calculate the optimal sequence (possibly non-stationary) of distributions of money in an extension of the model proposed by Kiyotaki and Wright (1989). Three aspects of this calculation are emphasized: (i) the computational implementation of the planner's problem involves the choice of continuous and discrete variables that maximize a nonlinear function and satisfies nonlinear constraints; (ii) the objective function of this problem is not concave and constraints are not convex, and (iii) the set of admissible choices is not known a priori. The goal is to document the difficulties involved, the proposed solutions and available methods and resources to implement the numerical characterization of efficient monetary dynamics under the assumption of random matching.

Keywords: Non-Stationary Monetary Allocations, Computational Economics, Recursive Methods, MINLP.

Sumário

1	Introdução	6
2	Modelo	7
2.1	Meio Físico	7
2.2	Problema Recursivo	8
3	Otimização Numérica	12
3.1	Técnica de Linearização	15
4	Métodos e Pacotes de Otimização	17
4.1	Método Branch and Bound (BB)	17
4.2	Método Outer Approximation (OA)	23
4.3	Pacotes de Otimização	27
4.3.1	BONMIN	28
4.3.2	TOMLAB	29
5	Iteração do Conjunto de Estados Admissíveis	29
5.1	O Método	29
5.2	Implementação	32
6	Considerações Finais	35
7	Bibliografia	37

1 Introdução

Monteiro e Cavalcanti (2006), doravante denominados MC, demonstram a existência de uma sequência ótima de distribuição de moeda em uma extensão do modelo proposto por Kiyotaki e Wright (1989), na qual não há restrições aos encaixes monetários individuais e existe choque agregado de preferências. O mérito de tal resultado reside na possibilidade de esta sequência ótima não ser estacionária. Adicionalmente, é apresentado um método recursivo não tradicional para o cálculo da função política ótima do planejador. Tal método é inovador pela utilização de uma variável de estado não usual e pela iteração não só da função valor do problema, mas também no conjunto de escolhas do planejador. A variável de estado proposta é a utilidade indireta proporcionada pela moeda e a iteração se dá no conjunto de promessas sustentáveis desta variável.

A extensão imediata desta contribuição é a implementação numérica do método proposto, a fim de obter características da alocação ótima do modelo. No entanto, a alta dimensionalidade e a existência de variáveis contínuas e discretas, de não linearidades e de não concavidade torna a solução numérica do problema uma tarefa desafiadora. O problema é classificado pela literatura em otimização computacional como Mixed-Integer NonLinear Problem (MINLP) não convexo¹ de larga escala, sendo um dos mais difíceis de se tratar. Talvez por esta razão os economistas, em geral, contornem tal dificuldade utilizando hipóteses que simplificam a fonte de não linearidade ou que reduzem a dimensão discreta do problema. Lagos e Wright (2005), por exemplo, usam a hipótese de preferências quase-lineares, enquanto Kiyotaki e Wright (1989) restringe os encaixes monetários individuais a um conjunto binário. Tais hipóteses permitem solucionar os modelos não só numericamente, mas também analiticamente. Apesar de terem possibilitado as relevantes contribuições destes trabalhos, tais hipóteses impossibilitam o tratamento de outras perguntas importantes, como a caracterização da sequência ótima de distribuições de moeda.

Motivados em responder outra pergunta não tratável em modelos simplificados como os destacados acima, Deviatov e Wallace (2009) encaram o desafio de trabalhar com MINLP's não convexos. A descrição do problema computacional, no entanto, é relegada a uma nota de rodapé, reproduzida a seguir:

"The optimum problem is solved using the General Algebraic Modeling System (GAMS), which is designed for the solution of large linear, nonlinear, and mixed

¹É tradicional discutir algoritmos de otimização para problemas de minimização, o que explica a preocupação com a convexidade do problema e não com sua concavidade, como era de se esperar em problemas de maximização. Problemas de maximização, no entanto, são trivialmente equivalentes à minimização do negativo da função objetivo.

integer optimization problems. It consists of a language compiler and a large menu of stable integrated high-performance solvers. The solvers are divided into two groups: local solvers (which are fast, but do not guarantee that the global solution is located) and global solvers (which are slow, but are very likely to find the global optimum). The global solver used is a Branch-And-Reduce Optimization Navigator (BARON) solver. BARON implements a deterministic algorithm of the branch-and-bound type, which is guaranteed to find the global optimum under very general conditions. These conditions include bounds on variables and the functions of them that appear in the nonlinear programming problem to be solved."

Claramente esta nota não apresenta detalhadamente a solução (e o problema) computacional utilizada, o que já era esperado de um trabalho cujo objetivo principal é responder uma pergunta teórica. Neste sentido, este relatório complementa trabalhos como Deviatov & Wallace (2009) na medida em que relata à academia em Economia o problema e a solução computacional do MINLP associado à implementação do método recursivo proposto por MC.

A extensão de MC ao modelo de Kiyotaki e Wright (1989) e o resultado de equivalência entre as formulações sequencial e recursiva do problema do planejador são apresentados na seção 2. A seção 3 descreve a formulação computacional do problema de otimização, com destaque para a técnica de linearização que torna o problema côncavo. Na seção 4 são apresentados os métodos Branch and Bound (BB) e Outer Approximation (OA), dois dos principais algoritmos de solução de MINLP's, e as ferramentas BONMIN e TOMLAB, as quais disponibilizam implementações numéricas destes métodos e/ou variantes destes. A seção 5 apresenta o método de iteração do conjunto de estados admissíveis e sua implementação. A seção 6 conclui com algumas considerações finais e futuras linhas de trabalho.

2 Modelo

2.1 Meio Físico

O meio físico considerado por MC é uma generalização daquele proposto por Kiyotaki e Wright (1989). O tempo é discreto, o horizonte é infinito e existe na economia uma população contínua de medida unitária e N bens de consumo perecíveis e divisíveis, em que $N \geq 3$. Os agentes são heterogeneos e se diferenciam pelo seu tipo $i \in I \equiv \{1, 2, \dots, N\}$, de forma que agentes do tipo i auferem utilidade somente do consumo do bem i e são capazes de produzir somente o bem $i + 1$, módulo² N . Há $1/N$ pessoas de cada tipo i . As pessoas se encontram uma vez

²Agentes do tipo N , consomem bem N e produzem bem 1.

por período segundo pareamento aleatórios. Tais pareamento podem ser de coincidência de interesses simples, quando uma pessoa do tipo i encontra com uma pessoa do tipo $i + 1$ ou do tipo $i - 1$, ou sem coincidência³.

Em um encontro de coincidência simples o agente do tipo i é um potencial produtor para o tipo $i + 1$, se o encontro é entre tipo i e tipo $i + 1$, e um potencial consumidor do bem o qual o tipo $i - 1$ é capaz de produzir, se o encontro é entre tipo i e tipo $i - 1$. Se $y \in \mathbb{R}_+$ unidades são produzidas e consumidas neste encontro, o consumidor auferir utilidade instantânea $u_z(y)$ e o produtor auferir $-y$, independentemente dos tipos envolvidos no encontro. O subscrito z indica o choque de preferências agregado, o qual possui suporte $\mathcal{Z} = \{z_1, z_2, \dots, z_k\}$ e distribuição de probabilidade *iid* tal que $\Pr(z = z_i) = p_i > 0$. Para cada $z \in \mathcal{Z}$, a função $u_z : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ é crescente, contínua e tal que $u_z(0) = 0$. O produto y de cada encontro é limitado por $y_{\max} \in \mathbb{R}_+$ tal que $u_z(y_{\max}) \leq y_{\max}$, para todo $z \in \mathcal{Z}$.

Moeda é indivisível e é o único ativo desta economia. Não há restrições sobre os encaixes monetários individuais, mas a quantidade agregada é limitada em M unidades. A distribuição de moeda nesta economia será, então, uma sequência $\mu = \{\mu_m\}_{m=0}^\infty$, em que $\mu_m = \mu(m)$ é a massa de agentes com m unidades monetárias. Dessa forma, o espaço de possíveis distribuições de moeda é dado pelo espaço \mathcal{M} de sequências μ tais que $\sum_{m=0}^\infty \mu_m = 1$, $\sum_{m=0}^\infty m\mu_m = M$ e $\mu_m \geq 0$ para todo $m \in \mathbb{Z}_+$.

Os agentes não são capazes de se comprometer com ações futuras e suas histórias são privadas. Tipos e quantidades monetárias são observáveis durante o encontro.

2.2 Problema Recursivo

Primeiramente são necessárias algumas definições:

Definição 1 *Uma história σ^t é uma lista $((\mu_0, z_0), (\mu_1, z_1), \dots, (\mu_t, z_t))$ de distribuições de moeda e realizações de choques agregados até o período t .*

Encontros de coincidência simples são indexados por (m, n) , em que m é a quantidade de moeda do potencial consumidor e n é a quantidade de moeda do potencial produtor. A ausência de indicação aos tipos dos agentes envolvidos no encontro decorre do interesse em alocações simétricas em relação aos tipos.

³A hipótese de que $N \geq 3$ exclui a possibilidade de pareamentos de coincidência dupla.

Definição 2 Uma recomendação é um par de funções (y, g) que associam a cada $(m, n) \in \mathbb{Z}_+^2$ um par ordenado $(y(m, n), g(m, n)) \in \mathbb{R}_+ \times \mathbb{Z}_+$ que descreve para encontros do tipo (m, n) o produto transacionado, $y(m, n)$, e a quantidade de moeda em poder do consumidor após a troca, $g(m, n)$.

A quantidade de moeda em poder do produtor após a troca é descrita pela função $h : \mathbb{Z}_+^2 \rightarrow \mathbb{Z}_+$ tal que⁴ $h(m, n) = m + n - g(m, n)$ para todo $(m, n) \in \mathbb{Z}_+^2$.

Definição 3 Um plano contingente, ou alocação, é (Y, G, μ_0) , em que (Y, G) são funções definidas para cada lista (σ, m, n) tais que $Y(\sigma, m, n) = y(m, n)$ e $G(\sigma, m, n) = g(m, n)$, para alguma recomendação (y, g) .

Observe que, dado μ_0 , G gera uma sequência $(\mu_1, \mu_2, \dots, \mu_t)$ para cada (z_0, z_1, \dots, z_t) .

Definição 4 Uma alocação (Y, G, μ_0) é factível se, dada qualquer história gerada por G e μ_0 , as recomendações a ela associada são factíveis.

Uma recomendação (y, g) é dita factível se $0 \leq y \leq y_{\max}$ e $0 \leq g(m, \cdot) \leq m$.

Definição 5 Dada uma função limitada V e uma história σ , o plano (Y, G) é dito implementável relativamente a V em σ se a recomendação a ela associada, (y, g) , é factível e satisfaz

$$u_z(y(m, n)) + \beta E[V(\sigma', g(m, n))|\sigma] \geq \beta E[V(\sigma', m)|\sigma] \quad (1)$$

$$-y(n, m) + \beta E[V(\sigma', h(n, m))|\sigma] \geq \beta E[V(\sigma', n)|\sigma] \quad (2)$$

$$\begin{aligned} V(\sigma, m) &= \frac{1}{N} \sum_{n=0}^{\infty} \{u_z(y(m, n)) + \beta E[V(\sigma', g(m, n))|\sigma]\} \mu(dn) + \\ &+ \frac{1}{N} \sum_{n=0}^{\infty} \{-y(n, m) + \beta E[V(\sigma', h(n, m))|\sigma]\} \mu(dn) + \\ &+ \left(1 - \frac{2}{N}\right) \beta E[V(\sigma', m)|\sigma] \end{aligned} \quad (3)$$

para todo $(m, n) \in \mathbb{Z}_+^2$.

⁴Tal restrição antecipa que, sob a alocação ótima, os agentes não desejarão se desfazer de moeda.

Em que (1) e (2) são as restrições de participação do consumidor e do produtor, respectivamente, e (3) é denominada Identidade de Markov (*IM*). As duas primeiras restrições garantem que os agentes não possuem incentivo a desviar unilateralmente da recomendação (y, g) , dado o retorno futuro da moeda definido pela função $V(\sigma', \cdot)$. A restrição (3) exige que o planejador honre a promessa de retorno da moeda $V(\sigma, \cdot)$, seja por meio de trocas $(u_z(y))$, seja por meio de mais promessas $V(\sigma', \cdot)$.

Dado uma alocação (Y, G, μ_0) , o problema do indivíduo é escolher aceitar ou não a recomendação do planejador em cada período $t + i$, cada possível história σ^{t+i} , cada possível quantidade de moeda m_{t+i} e cada possível pareamento (m_{t+i}, n) ou (n, m_{t+i}) , ou seja, dada uma história σ^t gerada por (Y, G, μ_0) , um agente com m_t unidades monetárias resolve

$$\sup_{\alpha} \left\{ E_{\sigma} \sum_{i=0}^{\infty} \beta^i \sum_{n=0}^{\infty} \frac{1}{N} [\alpha_{t+i}^c u_z(Y(\sigma^{t+i}, m_{t+i}, n)) - \alpha_{t+i}^p Y(\sigma^{t+i}, n, m_{t+i})] \mu_{t+i}(n) \right\} \quad (4)$$

o que lhe proporciona utilidade $V^*(\sigma^t, m_t)$. Em (4), $\alpha_{t+i}^c = \alpha_{t+i}^c(\sigma^{t+i}, m_{t+i}, n)$ é 1 se o agente aceita a recomendação para o encontro (m_{t+i}, n) , sob a história σ^{t+i} , sendo ele o consumidor com m_{t+i} unidades monetárias. Se rejeitar a recomendação, então $\alpha_{t+i}^c = 0$. A variável $\alpha_{t+i}^p = \alpha_{t+i}^p(\sigma^{t+i}, n, m_{t+i})$ é definida de maneira análoga. Dessa forma, a função V^* associa a cada história σ^t gerada por (Y, G, μ_0) e cada possível montante de moeda m_t , a máxima utilidade auferida sob as recomendações associadas a (Y, G) .

Definição 6 *A alocação (Y, G, μ_0) é implementável se é factível e se o plano (Y, G) é implementável em relação à V^* em todas as possíveis histórias geradas por G e μ_0 .*

Definição 7 *Uma alocação é ótima se é implementável e se as recomendações a ela associadas resolvem*

$$\sup E_{\sigma_0} \sum_{t=0}^{\infty} \beta^t U(\mu_t, z_t, y_t) \quad (\text{SP})$$

em que $\sigma_0 = (\mu_0, z_0)$ é dada e $U(\cdot)$ é a função de payoff social.

O problema (SP) define $W^*(\mu_0, z_0)$ para cada história $\sigma_0 = (\mu_0, z_0)$. W^* é denominada função supremo do problema do planejador. Um exemplo para U , e que será utilizado no problema computacional a seguir, é a utilidade média

$$U_{rep}(\mu, z, y) = \frac{1}{N} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} [u_z(y(m, n)) - y(m, n)] \mu_m \mu_n \quad (5)$$

MC apresentam uma formulação recursiva do problema e demonstram sua equivalência com o problema sequencial (*SP*). A formulação é dividida em dois passos: primeiramente uma recomendação ótima é encontrada resolvendo-se a equação de Bellman

$$\begin{aligned} w(\mu, v, z) &= \sup_{(\mu', v', y, g)} \{U(\mu, z, y) + \beta E[w(\mu', v', z')]\} \\ &\quad s.t. (\mu', v', y, g) \in \Gamma(\mu, v, z) \\ &\quad 0 \leq \|v'\|_\infty \leq K \end{aligned} \tag{FE}$$

em que Γ é a correspondência que associa a cada $(\mu, v, z) \in A$ o conjunto $\Gamma(\mu, v, z)$ de elementos (μ', v', y, g) tais que $\forall (m, n) \in \mathbb{Z}_+^2$ tem-se

$$0 \leq g(m, n) \leq m \quad h(m, n) = m + n - g(m, n)$$

$$u_z(y(m, n)) \geq \beta E[v'(m) - v'(g(m, n))]$$

$$y(m, n) \leq \beta E[v'(h(m, n)) - v'(n)]$$

$$\begin{aligned} v_z(m) &= \beta \left(1 - \frac{2}{N}\right) E[v'(m)] + \frac{1}{N} \sum_{n=0}^{\infty} \{u_z(y(m, n)) - y(n, m)\} \mu_n + \\ &\quad + \frac{\beta}{N} \sum_{n=0}^{\infty} \{E[v'(g(m, n)) + v'(m + n - g(n, m))]\} \mu_n \end{aligned}$$

$$\mu'([0, \lambda]) = \left(1 - \frac{2}{N}\right) \mu([0, \lambda]) + \frac{1}{N} (\mu \times \mu([g \leq \lambda]) + \mu \times \mu([h \leq \lambda])) \quad \forall \lambda \geq 0 \tag{6}$$

e K é um limite superior uniforme para v dado por

$$\max \{u_z(y_{\max}); z \in \mathcal{Z}\}$$

Em seguida, uma alocação ótima é encontrada resolvendo-se

$$\begin{aligned} W^*(\mu_0, z_0) &= \sup_x w^*(\mu, x, z) \\ &\quad s.t. (\mu_0, x, z_0) \in A \end{aligned}$$

em que A é o conjunto de estados admissíveis e w^* é a função valor obtida no primeiro passo.

Um estado (μ, v, z) é dito admissível se existe (y, g) e (μ', v') tais que

- (i) $(\mu', v', y, g) \in \Gamma(\mu, v, z)$;
- (ii) $(\mu', v', z') \in A$ para todo $z' \in Z$; e
- (iii) $0 \leq \|v'\|_\infty \leq K$

ou seja, são estados para os quais existe recomendação que induz estados admissíveis e limitados para qualquer possível choque agregado futuro. A definição de A como o maior conjunto de estados admissíveis garante que a alocação (Y, G, μ_0) obtida no segundo passo da formulação recursiva é implementável.

3 Otimização Numérica

Na solução numérica de (FE) são calculadas a função valor w^* , ponto fixo da equação funcional, e a função política do planejador, a qual associa a cada possível valor de (μ, v, z) a escolha (μ', v', y, g) que resolve (FE) para $w = w^*$. Para calcular w^* , escolhe-se uma função w_0 candidata a ponto fixo da equação funcional, resolve-se (FE) para $w = w_0$ e a função valor obtida é usada como nova candidata. Tal procedimento é repetido até a convergência da sequência de candidatas $\{w_i\}_{i=0}^\infty$. Cada solução de (FE) para uma dada candidata w_i define uma iteração da função valor e envolve um problema de otimização para cada possível valor de (μ, v, z) . Esta seção apresenta este problema de otimização computacional, ou seja, a otimização numérica a ser executada a cada iteração da função valor e para cada possível realização das variáveis de estado.

Desta forma, considere como dados valores para as variáveis de estado, (μ, v, z) . O problema é escolher $(\mu', v', y, g) \in \Gamma(\mu, v, z)$, com $0 \leq \|v'\|_\infty \leq K$, que resolva

$$\max \{U_{rep}(\mu, z, y) + \beta E[w(\mu', v', z')]\}$$

MC demonstram que neste caso especial, em que U assume a forma funcional U_{rep} definida em (5), tem-se que a função valor que satisfaz (FE) é w^* tal que

$$w^*(\mu, v, z) = \sum_{m=0}^{\infty} v_z(m) \mu(m)$$

A fim de simplificar a exposição do problema numérico de otimização, será utilizada esta solução analítica.

Visto que a implementação numérica requer dimensões finitas, suponha que os encaixes monetários individuais não podem ser maiores do que uma constante \overline{M} , $\overline{M} < \infty$. Como consequência, distribuições de moeda podem ser descritas por vetores do $\mathbb{R}_+^{(1+\overline{M})}$, ou seja, o espaço de possíveis distribuições de moeda é

$$\mathcal{M} = \left\{ \mu \in \mathbb{R}_+^{M_1}; \sum_{m=0}^{\overline{M}} \mu_m = 1 \text{ e } \sum_{m=0}^{\overline{M}} m\mu_m = M \right\}$$

em que $M_1 \equiv (1 + \overline{M})$. Adicionalmente, a variável de estado v deixa de ser uma lista de funções (uma para cada z) com domínio \mathbb{N} e passa a ser uma lista de funções com domínio $\mathbb{M} = \{0, 1, \dots, \overline{M}\}$, ou seja, v_z é elemento de

$$\mathbb{V} = \{v : \mathbb{M} \rightarrow \mathbb{R}_+; v \text{ é crescente}\}$$

As funções y e g passam a ser elementos de \mathbb{Y}^b e \mathbb{Y} , respectivamente, em que

$$\begin{aligned} \mathbb{Y}^b &= \{y : \mathbb{M}^2 \rightarrow [0, y_{\max}]\} \\ \mathbb{Y} &= \{g : \mathbb{M}^2 \rightarrow \mathbb{M}\} \end{aligned}$$

Sob tais hipóteses o problema passa a ser

$$\max \left\{ \frac{1}{N} \sum_{m=0}^{\overline{M}} \sum_{n=0}^{\overline{M}} [u_z(y(m, n)) - y(m, n)] \mu_m \mu_n + \beta \sum_{i=1}^k p_i \left[\sum_{m=0}^{\overline{M}} v'_{z_i}(m) \mu'_m \right] \right\} \quad (7)$$

sujeito à $(\mu', v', y, g) \in \Gamma(\mu, v, z)$, com $\Gamma : \mathbb{A} \rightarrow \mathcal{M} \times \mathbb{V}^{\mathbb{Z}} \times \mathbb{Y}^b \times \mathbb{Y}$, e $0 \leq \max_{(z, m)} [v'] \leq K$. Observe que a esperança de w^* é tomada somente em relação a z' , visto que μ' é determinada por g , pela distribuição de moeda atual μ e pela dinâmica de transição definida em (6). Além disso, tal relação permite reescrever a função objetivo em (7) diretamente em função de g e h .

Para ver isto note que (6) implica que $\forall m \in \mathbb{N}$

$$\begin{aligned}
\sum_{m=0}^{\overline{M}} v'(m) \mu'(\{m\}) &= \sum_{m=0}^{\overline{M}} \left(1 - \frac{2}{N}\right) v'(m) \mu_m + \\
&\quad + \sum_{m=0}^{\overline{M}} \frac{1}{N} v'(m) (\mu \times \mu([g=m]) + \mu \times \mu([h=m])) \\
&= \left(1 - \frac{2}{N}\right) \sum_{m=0}^{\overline{M}} v'(m) \mu_m + \\
&\quad + \frac{1}{N} \sum_{m=0}^{\overline{M}} \sum_{i=0}^{\overline{M}} \sum_{j=0}^{\overline{M}} v'(m) (\chi_{[g(i,j)=m]} + \chi_{[h(i,j)=m]}) \mu_i \mu_j
\end{aligned}$$

em que

$$\chi_{[g(i,j)=m]} = \begin{cases} 1 & \text{se } g(i,j) = m \\ 0 & \text{se } g(i,j) \neq m \end{cases} \quad \chi_{[h(i,j)=m]} = \begin{cases} 1 & \text{se } h(i,j) = m \\ 0 & \text{se } h(i,j) \neq m \end{cases}$$

Como a função v'_z pode ser representada por um vetor do \mathbb{R}^{M_1} , a função y por um vetor do $[0, y_{\max}]^{M_1^2}$, e g por um vetor do $\mathbb{M}^{M_1^2}$, as variáveis de escolha são as entradas destes vetores. Ou seja, v'_z será determinada por meio da escolha das M_1 variáveis $v'_z(m) \in [0, K]$, $m \in \mathbb{M}$, y será determinada pela escolha de $M_2 \equiv (1 + \overline{M})^2$ variáveis $y(m, n) \in [0, y_{\max}]$, $(m, n) \in \mathbb{M}^2$ e g será determinada pela escolha de M_2 variáveis $g(m, n) \in \mathbb{M}$, $(m, n) \in \mathbb{M}^2$. A função h é obtida da determinação de g . O número de variáveis de escolha é

$$\begin{aligned}
&kM_1 \text{ variáveis } v'_z(m) \\
&M_2 \text{ variáveis } y(m, n) \\
&M_2 \text{ variáveis } g(m, n) \\
&\underline{M_2 \text{ variáveis } h(m, n)} \\
&kM_1 + 3M_2 \text{ variáveis}
\end{aligned}$$

No entanto, as funções g e h aparecem como argumento da função v' nas restrições de participação e na IM . Uma forma de tratar tal dificuldade é escrever

$$v'_z[g(m, n)] = \sum_{j=0}^{\overline{M}} v'_z(j) \chi_{[g(m,n)=j]} \quad \text{e} \quad v'_z[h(m, n)] = \sum_{j=0}^{\overline{M}} v'_z(j) \chi_{[h(m,n)=j]}$$

Com isso, $g(m, n)$ passa a ser determinada pela escolha de M_1 variáveis binárias $\chi_{[g(m,n)=j]}$ tais

que $\sum_{j=0}^{\overline{M}} \chi_{[g(m,n)=j]} = 1$, visto que $g(m,n) = \sum_{j=0}^{\overline{M}} j \chi_{[g(m,n)=j]}$. Similarmente, $h(m,n)$ é determinada por M_1 variáveis binárias $\chi_{[h(m,n)=j]}$ tais que $\sum_{j=0}^{\overline{M}} \chi_{[h(m,n)=j]} = 1$, visto que $h(m,n) = \sum_{j=0}^{\overline{M}} j \chi_{[h(m,n)=j]}$. Ou seja, cada uma das M_2 variáveis $g(m,n)$ e cada uma das M_2 variáveis $h(m,n)$ é agora determinada por M_1 variáveis binárias e uma restrição linear. Consequentemente, o número de variáveis aumenta para $kM_1 + M_2 + 2M_3$, com $M_3 \equiv (1 + \overline{M})^3$, e adiciona-se $2M_2$ restrições lineares.

3.1 Técnica de Linearização

Na forma que foi apresentado na subseção anterior, o problema do planejador para cada valor de variável de estado é um MINLP não côncavo. As fontes de não concavidade são os termos bilineares entre $v'_z(j)$ e $\chi_{[g(m,n)=j]}$ presentes na IM , nas restrições de participação do produtor e do consumidor e na função objetivo. Problemas de otimização não côncavos podem possuir mais de um máximo local, o que exige a utilização de algoritmos que explorem todos os máximos locais a fim de garantir a globalidade da solução. Esses algoritmos utilizam técnicas que os tornam computacionalmente custosos se comparados aos algoritmos de busca local. Custo este ainda maior quando há variáveis de escolha contínuas e discretas.

Felizmente, no presente caso, o problema pode ser reescrito como um problema côncavo utilizando a linearização proposta por Glover (1975). Essa reformulação garante que a solução encontrada por algoritmos de busca local é global e permite (a princípio) explorar versões mais ricas (maior dimensão) do modelo.

A técnica de linearização propõe substituir os termos bilineares $v_z(j)\chi_{[f(m,n)=j]}$ por

$$s_{zj}^f(m,n) \equiv s(f,m,n,z,j) = v_z(j)\chi_{[f(m,n)=j]} \quad (8)$$

e adicionar as seguintes restrições

$$\begin{aligned} 0 &\leq s_{zj}^f(m,n) \leq K\chi_{[f(m,n)=j]} \\ v_z(j) - K(1 - \chi_{[f(m,n)=j]}) &\leq s_{zj}^f(m,n) \leq v_z(j) \end{aligned}$$

em que K é o limite superior uniforme para v . Observe que se $\chi_{[f(m,n)=j]} = 0$, a primeira

restrição exige que $s_{zj}^f(m, n) = 0$ e a segunda restrição, $v_z(j) - K \leq s_{zj}^f(m, n) \leq v_z(j)$, é redundante. Se $\chi_{[f(m,n)=j]} = 1$ então a segunda restrição exige que $s_{zj}^f(m, n) = v_z(j)$ e a primeira restrição, $0 \leq s_{zj}^f(m, n) \leq K$, é redundante. Dessa forma, as restrições garantem que (8) seja satisfeita. Com essa substituição as restrições de participação do consumidor e do produtor, $\forall(m, n) \in \mathbb{M}^2$, tornam-se

$$u_z(y(m, n)) \geq \beta \sum_{i=1}^k p_i \left[v'(m) - \sum_{j=0}^{\overline{M}} s_{zj}^g(m, n) \right] \quad (9)$$

$$y(m, n) \leq \beta \sum_{i=1}^k p_i \left[\sum_{j=0}^{\overline{M}} s_{zj}^h(m, n) - v'(n) \right] \quad (10)$$

enquanto a IM , para todo $m \in \mathbb{M}$

$$\begin{aligned} v_z(m) = & \beta \left(1 - \frac{2}{N} \right) \sum_{i=1}^k p_i v'_{z_i}(m) + \frac{1}{N} \sum_{n=0}^{\overline{M}} \{ u_z(y(m, n)) - y(n, m) \} \mu_n + \\ & + \frac{\beta}{N} \sum_{n=0}^{\infty} \left\{ \sum_{i=1}^k p_i \left[\sum_{j=0}^{\overline{M}} s_{zj}^g(m, n) + \sum_{j=0}^{\overline{M}} s_{zj}^h(n, m) \right] \right\} \mu_n \end{aligned} \quad (11)$$

e a função objetivo

$$U_{rep}(\mu, z, y) + \sum_{m=0}^{\overline{M}} \left\{ \left(1 - \frac{2}{N} \right) \mu_m v'_{z_l}(m) + \frac{1}{N} \sum_{i=0}^{\overline{M}} \sum_{j=0}^{\overline{M}} \mu_i \mu_j [s_{z_l m}^g(i, j) + s_{z_l m}^h(i, j)] \right\} \quad (12)$$

Como consequência, o número de variáveis de escolha aumenta em $2kM_1M_2 = 2kM_3$ e o número de restrições em $8kM_3$, quatro restrições para cada nova variável. Apesar do aumento no número de variáveis de escolha e de restrições, a linearização de Glover elimina todas as fontes de não concavidade e reduz as fontes de não linearidades do problema: a restrição de participação do produtor é agora linear e a única fonte de não linearidade na restrição de participação do consumidor, na IM e na função objetivo é a função u_z . Além disso, as novas variáveis são contínuas e as restrições são lineares, o que não representa um custo elevado, visto que o esforço computacional de solução de MINLP's depende predominantemente do número de variáveis discretas e do grau de não linearidade do problema.

Assim reformulado, o problema computacional é a escolha de $kM_1 + M_2 + 2M_3 + 2kM_3$

variáveis para solucionar um MINLP côncavo com $1 + M_2 + 2M_2 + 8kM_3$ restrições lineares⁵ e $M_1 + M_2$ restrições não lineares⁶. Por exemplo, se o modelo de Kiyotaki & Wrigth (1989), no qual $\overline{M} = 1$ e $k = 1$, for modificado para $k = 2$, haverá 56 variáveis e 141 restrições. Tal problema é definido pela maximização de (12), sujeito às restrições (9), (10), (11), $0 \leq \max_{(z,m)}[v'] \leq K$ e

$$\begin{aligned} \max \{0, m + n - \overline{M}\} &\leq g(m, n) \leq m \\ h(m, n) &= m + n - g(m, n) \end{aligned}$$

Ou seja, trata-se de um MINLP de larga escala, mas cuja concavidade permite a utilização de algoritmos de busca local, os quais são eficientes e garantem a globalidade da solução obtida.

4 Métodos e Pacotes de Otimização

Conforme destacado por Bonami et al. (2008), o desenvolvimento de algoritmos de solução de MINLP's tornou-se recentemente uma área de pesquisa muito ativa. De forma geral, estes algoritmos combinam métodos de solução de problemas lineares com variáveis contínuas e discretas (MILP's) com aqueles desenvolvidos para problemas não lineares com somente variáveis contínuas (NLP's). A razão para tal é o avançado estágio de desenvolvimento destes métodos e a existência de consolidadas rotinas de implementação. Dentre os algoritmos de solução de MINLP's, os métodos Branch and Bound (BB) e Outer Approximation (OA) são os mais conhecidos e formam a base para a maioria dos demais métodos. Ambos serão detalhados a seguir.

Cabe destacar que os métodos aqui apresentados são adequados a MINLP's côncavos e servem apenas como abordagens heurísticas aos MINLP's não côncavos.

4.1 Método Branch and Bound (BB)

Embora o interesse final resida em MINLP's, MILP's são mais simples e suficientes para a apresentação do método BB. Além disso, MILP's são úteis como subproblemas de problemas MINLP's, conforme evidenciado pelo método OA.

⁵Uma relação entre h e g , M_2 restrições de participação do produtor, $2M_2$ restrições sobre as variáveis binárias e $8kM_3$ restrições geradas pela linearização de Glover.

⁶ M_1 Identidades de Markov e M_2 restrições de participação do consumir.

O problema a seguir é a formulação geral^{7,8} de problemas MILP's.

$$\begin{aligned}
& \min_{x,y} \{c'x + d'r\} \\
& s.a \quad Ax + Br \leq b \\
& \quad x \geq 0, x \in X \subseteq R^n \\
& \quad r \in \{0,1\}^q
\end{aligned} \tag{13}$$

em que x é um vetor de n variáveis contínuas, r é um vetor de q variáveis binárias, c e d são vetores de parâmetros de dimensão n e q , respectivamente. As matrizes A e B possuem dimensão $p \times n$ e $p \times q$, respectivamente, e b é um vetor de dimensão p .

Note que, dado um vetor $\bar{r} \in \{0,1\}^q$, (13) torna-se um problema de programação linear (LP). Uma possível estratégia de solução é enumerar todos os possíveis valores de \bar{r} , para cada um deles resolver o LP associado e escolher a solução de menor magnitude. Apesar de garantir a obtenção da solução, essa alternativa é muito ineficiente e muitas vezes impraticável, visto que haverá 2^q LP's associados a (13). A figura 1 ilustra a árvore de enumeração para o caso em que há três variáveis binárias ($r_1 = y$, $r_2 = w$ e $r_3 = s$) e, portanto, $2^3 = 8$ LP's associados. O nó correspondente ao nível 0 ($l = 0$) é a raiz da árvore e representa o problema (13). Os nós do nível 1 ($l = 1$) correspondem aos subproblemas de (13) gerados pela adição de uma restrição de igualdade sobre a variável y . Os nós do nível 2 ($l = 2$) correspondem aos subproblemas dos subproblemas do nível 1 ($l = 1$) gerados pela adição de uma restrição de igualdade sobre a variável w . Os nós do nível 3 ($l = 3$) correspondem aos subproblemas dos subproblemas do nível 2 ($l = 2$) gerados pela adição de uma restrição de igualdade sobre a variável s .

Em geral, os algoritmos buscam solucionar MILP's sem recorrer a completa enumeração das combinações das variáveis binárias. Dentre estes, o método BB é o algoritmo mais comumente utilizado, conforme observado por Floudas (1995).

A idéia básica do método BB é o cálculo de uma sequência de cotas superiores⁹ que converge para a solução de (13) em um número finito de iterações. Tais cotas superiores são soluções dos LP's associados ao problema considerado. A atratividade do método é que ele não requer

⁷Observe que o caso de variáveis discretas (mas não-binárias) também é contemplado nesta formulação. Suponha que exista uma variável s que assume valores em $S = \{s_1, s_2, \dots, s_n\}$. Tal variável pode ser reescrita como $s = \sum_{i=1}^n s_i \cdot u_i$, em que u_i 's são variáveis binárias tais que $\sum_{i=1}^n u_i = 1$.

⁸Lembre-se que os métodos e algoritmos tradicional são desenvolvidos para problemas de minimização.

⁹Lembrando (mais uma vez) que os algoritmos são tradicionalmente desenvolvidos para problemas de minimização.

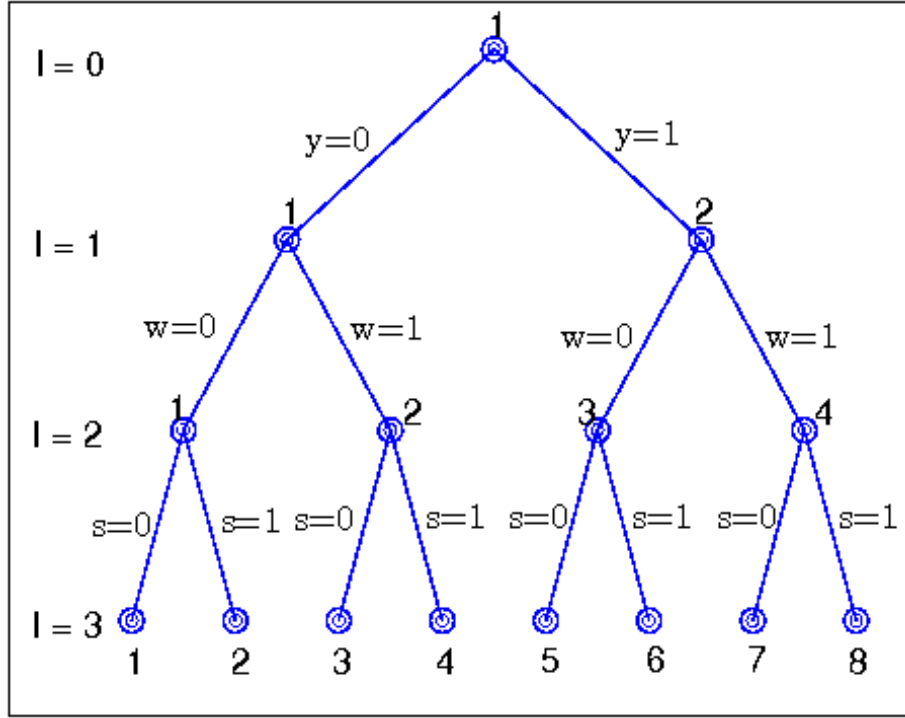


Figura 1: Árvore com três variáveis binárias.

a exploração de todos os LP's para concluir que a sequência já convergiu. O detalhamento do método, no entanto, requer a definição de alguns conceitos e objetos.

Definição 8 Seja $SF(Q)$ o conjunto de soluções factíveis de um problema Q . Um conjunto de subproblemas q_1, q_2, \dots, q_n de Q é definido como uma separação de Q se:

- (i) $SF(q_i) \subseteq SF(Q) \quad i = 1, 2, \dots, n$
- (ii) $SF(q_i) \cap SF(q_j) = \phi \quad i \neq j$

Definição 9 Um problema de otimização é definido como um relaxamento do problema Q e denotado RQ , se $SF(Q) \subseteq SF(RQ)$.

O relaxamento é obtido descartando-se restrições, por exemplo. No caso de (13) é possível obter um relaxamento substituindo-se a restrição $r \in \{0, 1\}^q$ pela restrição $r \in [0, 1]^q$. A vantagem de se trabalhar com relaxamentos é o seu menor custo computacional de solução. Note que a definição acima implica que:

- (i) se $SF(RQ) = \phi$, então $SF(Q) = \phi$;
- (ii) se z_Q é solução ótima de Q e z_{RQ} é solução ótima de RQ , então $z_{RQ} \leq z_Q$; e

(iii) se (x^*, y^*) soluciona RQ e $(x^*, y^*) \in SF(Q)$, então (x^*, y^*) soluciona Q .

Seja Q_i um subproblema de Q candidato a solucioná-lo, ou seja, $SF(Q_i)$ é candidato a conter a solução de Q . Q_i será considerado **explorado/entendido** (*fathomed*) se for possível afirmar que $SF(Q_i)$ não contém solução menor que a menor solução já conhecida (*incumbent*) ou se a solução ótima de Q_i for encontrada. O objetivo do método BB é explorar/entender todos os subproblemas de Q para encontrar sua solução, sem necessariamente resolvê-los. Uma forma atrativa de explorar/entender o subproblema Q_i é resolver seu relaxamento., RQ_i , e testar os três critérios de entendimento (CE) a seguir:

(CE_1) se $SF(RQ_i) = \phi$, então $SF(Q_i) = \phi$;

(CE_2) se $z_{RQ_i} \geq z^*$, em que z_{RQ_i} é solução ótima de RQ_i e z^* é a solução *incumbent*.

(CE_3) se (x^*, y^*) resolve RQ_i e $(x^*, y^*) \in SF(Q_i)$, então $z_{Q_i} = z_{RQ_i}$.

Se CE_1 é satisfeito, então Q_i não é factível e, portanto, deve ser descartado como candidato à solucionar Q . Além disso, todos os subproblemas de Q_i , denotados Q_{ij} , podem ser descartados (os galhos da árvore gerados a partir de Q_i podem ser podados), visto que $SF(Q_{ij}) \subseteq SF(Q_i)$. Se CE_2 é verificado, então Q_i não contém solução menor que a menor solução já conhecida e, portanto, deve ser descartado como candidato a solucionar Q_i . Novamente os galhos da árvore gerados a partir de Q_i podem ser podados, visto que $SF(Q_{ij}) \subseteq SF(Q_i)$. Se, no entanto, CE_3 é satisfeito, então obtém-se a solução de Q_i , a qual pode ser comparada com a menor solução conhecida, z^* , para descobrir se z_{Q_i} é a nova solução *incumbent* ou se Q_i e todos os seus sub-problemas podem ser descartados. O descarte de subproblemas e os galhos a partir dele gerados define a exploração seletiva empreendida pelo método BB. O algoritmo formaliza a seguir formaliza os argumentos aqui apresentados:

Algoritmo 10 Método BB

- p0.* Defina \mathbb{L} como uma lista de subproblemas e inicialize-a com o próprio (13) e defina $z^* = +\infty$;
- p1.* Se $\mathbb{L} = \phi$, o algoritmo termina e a solução ótima é o atual *incumbent*. Se não há *incumbent* finito, então (13) é não factível.
- p2.* Selecione um dos subproblemas de \mathbb{L} e o defina como o atual Q_i .
- p3.* Escolha uma forma de relaxamento de Q_i para definir RQ_i . Encontre, se possível, a solução ótima associada z_{RQ_i} .

(a) note z_{RQ_i} que é uma cota inferior (lower bound) para a solução de Q_i .

p4. Teste os critérios de exploração/entendimento (fathoming):

CE_1 : se $SF(RQ_i) = \phi$, então $SF(Q_i) = \phi$. Exclua Q_i de \mathbb{L} e retorne a p1.

CE_2 : se $z_{RQ_i} \geq z^*$, então $z_{Q_i} \geq z_{RQ_i} \geq z^*$. Exclua Q_i de \mathbb{L} e retorne a p1.

CE_3 : se (x^*, y^*) resolve RQ_i e $(x^*, y^*) \in SF(Q_i)$, então $z_{Q_i} = z_{RQ_i}$. Caso $z_{RSC} \leq z^*$, defina $z^* = z_{RSC}$, exclua Q_i de \mathbb{L} e vá para p5. Caso contrário, exclua Q_i de \mathbb{L} e retorne a p1.

p5. Separe¹⁰ o atual subproblema Q_i , adicione seus subproblemas à \mathbb{L} e exclua Q_i . Retorne a p1.

O exemplo a seguir ilustra o algoritmo.

Exemplo 11 Considere o problema a seguir, em que há três variáveis binárias e uma contínua e cuja árvore de enumeração dos $2^3 = 8$ LP's é apresentada na Figura 1. Observe que os oito nós do nível 3 correspondem aos LP's associados a (14), os quais compõem uma separação de (14). Dessa forma, um desses 8 LP's tem a mesma solução de (14). O objetivo do método BB é encontrar este LP e resolvê-lo sem precisar explorar todos os 8 LP's.

$$\begin{aligned} \min \{ & 2x - 3y - 2w - 3s \} & (14) \\ \text{s.a. } & x + y + w + s \geq 2 \\ & 10x + 5y + 3w + 4s \leq 10 \\ & x \geq 0 \quad y, w, s \in \{0, 1\} \end{aligned}$$

Note que a escolha ótima da variável x é $x = 0$, visto que a função objetivo é crescente em x . Das variáveis binárias, y e s são as que mais contribuem para a redução da função objetivo. Portanto, a escolha ótima deste problema é $(x, y, w, s) = (0, 1, 0, 1)$, a qual implica em $z^* = -6$.

Inicialmente a lista \mathbb{L} é inicializada com o próprio (14). Se SC_n^l denota o candidato correspondente ao nó n do nível l da árvore, então $\mathbb{L} = \{SC_1^0\}$, em p1. O valor inicial para a solução incumbent é infinito, $z^* = +\infty$. Em seguida, observe que p1 é irrelevante, pois $\mathbb{L} = \{SC_1^0\} \neq \phi$. De acordo com p2, defina $Q_i = SC_1^0$. Neste estágio o algoritmo começa a explorar o nó 1 do

¹⁰Lembre-se da definição de separação.

nível 0. A substituição da restrição $y, w, s \in \{0, 1\}$ por $y, w, s \in [0, 1]$ define o relaxamento RQ_i de Q_i , em p3. Resolvendo RQ_i utilizando programação linear, obtém-se $z_{RQ_i} = -6.8$ e $(x, y, w, s) = (0, 0.6, 1, 1)$. Nenhum dos critérios de entendimento é satisfeito em p4. Em p5, adicione à Q_i a restrição $y = 0$ para definir SC_1^1 , a restrição $y = 1$ para definir SC_2^1 , exclua SC_1^0 de \mathbb{L} , adicione SC_1^1 e SC_2^1 a \mathbb{L} e retorne a p1.

Novamente p1 é irrelevante, pois $\mathbb{L} = \{SC_1^1, SC_2^1\} \neq \phi$. Em p2 selecione $Q_i = SC_2^{111}$. Neste estágio o algoritmo começa a explorar o nó 2 do nível 1. A substituição da restrição $w, s \in \{0, 1\}$ por $w, s \in [0, 1]$ define o novo RQ_i , em p3. Resolvendo RQ_i obtém-se $z_{RQ_i} = -6.667$ e $(y, w, s) = (1, 0.333, 1)$. Novamente nenhum dos critérios de entendimento é satisfeito em p4. Em p5 adicione à Q_i a restrição $w = 0$ para definir SC_3^2 , adicione à Q_i a restrição $w = 1$ para definir SC_4^2 , exclua SC_2^1 de \mathbb{L} , adicione SC_1^2 e SC_2^2 a \mathbb{L} e retorne a p1.

Mais uma vez p1 é irrelevante, pois $\mathbb{L} = \{SC_1^1, SC_3^2, SC_4^2\} \neq \phi$. Em p2 obtém-se $Q_i = SC_4^2$. Neste estágio o algoritmo começa a explorar o nó 4 do nível 2. A substituição da restrição $s \in \{0, 1\}$ por $s \in [0, 1]$ define RQ_i , em p3. Resolvendo RQ_i obtém-se $z_{RQ_i} = -6.65$ e $(y, w, s) = (1, 1, 0.5)$. Nenhum dos critérios de entendimento é satisfeito em p4. Em p5 adicione à Q_i a restrição $s = 0$ para definir SC_7^3 , adicione à Q_i a restrição $s = 1$ para definir SC_8^3 , exclua SC_4^2 de \mathbb{L} , adicione SC_7^3 e SC_8^3 a \mathbb{L} e retorne a p1.

Novamente p1 é irrelevante, pois $\mathbb{L} = \{SC_1^1, SC_3^2, SC_7^3, SC_8^3\} \neq \phi$. Em p2 obtém-se $Q_i = SC_8^3$. Neste estágio o algoritmo começa a explorar o nó 8 do nível 3. Em p3 não há relaxamento e SC_8^3 é não factível. Em p4, CE_1 é satisfeito. Exclua SC_8^3 de \mathbb{L} e retorne a p1. Até aqui sabe-se que o subproblema SC_8^3 não possui a mesma solução que (14)

Como $\mathbb{L} = \{SC_1^1, SC_3^2, SC_7^3\} \neq \phi$, p1 é irrelevante. Em p2, $Q_i = SC_7^3$. Neste estágio o algoritmo começa a explorar o nó 7 do nível 3. Resolvendo RQ_i em p3 obtém-se $z_{RQ_i} = -5$ e $(y, w, s) = (1, 1, 0)$. Em p4, CE_3 é satisfeito. Atualize $z^* = z_{RSC} = -5$, exclua SC_7^3 de \mathbb{L} e retorne a p1. Até aqui o subproblema SC_7^3 é o melhor candidato a solucionar (14).

Passo p1 é irrelevante, pois $\mathbb{L} = \{SC_1^1, SC_3^2\} \neq \phi$. Em p2, $Q_i = SC_3^2$. Neste estágio o algoritmo começa a explorar o nó 3 do nível 2. Resolvendo RQ_i em p3 obtém-se $z_{RQ_i} = -6$ e $(y, w, s) = (1, 0, 1)$. Em p4, CE_3 é satisfeito. Atualize $z^* = z_{RSC} = -6$, exclua SC_3^2 de \mathbb{L} e retorne a p1. Até aqui o subproblema SC_6^3 é o melhor candidato a solucionar (14)

Como $\mathbb{L} = \{SC_1^1\} \neq \phi$, então p1 é irrelevante. Em p2, $Q_i = SC_1^1$. Neste estágio o algoritmo começa a explorar o nó 1 do nível 1. Resolvendo RSC em p3 obtém-se $z_{RSC} = -5$ e

¹¹Outras alternativas de seleção são: *breadth-first search* e *best bound search*. A regra *depth first search with backtracking* seleciona o último subproblema adicionado a lista de candidatos \mathbb{L} .

$(y, w, s) = (0, 1, 1)$. Em $p4$, CE_2 é satisfeito. Exclua SC_1^1 de \mathbb{L} e retorne a $p1$.

Em $p1$, como $\mathbb{L} = \phi$, conclui-se que a solução de (14) é $z^* = -6$ e $(x, y, w, s) = (0, 1, 0, 1)$, ou seja, o LP SC_6^3 resolve (14).

Apesar de simples, o exemplo destaca a atratividade do método: foi necessário resolver somente 3 dos 8 LP's associados ao problema. Isso é possibilitado pela obtenção de cotas inferiores para os demais LP's via o relaxamento de subproblemas de (14) que os antecedem na árvore de enumeração. Em problemas de larga escala, esta capacidade de descartar LP's sem precisar resolvê-los é extremamente importante.

Quando aplicado a MINLP's, os relaxamentos podem ser construídos não só considerando contínuas as variáveis binárias, mas também linearizando a função objetivo e/ou as restrições. A segunda técnica é utilizada também no método *OA* a seguir.

4.2 Método Outer Approximation (OA)

A apresentação do método "*Outer Approximation*" (*OA*) requer a utilização de MINLP's. O problema a seguir é a formulação geral¹² de MINLP's.

$$\begin{aligned} \min_{x,y} \quad & f(x, y) \\ \text{s.a} \quad & g(x, y) \leq 0 \\ & x \in X \subseteq R^n \\ & y \in \{0, 1\}^q \subseteq Y = [0, 1]^q \end{aligned} \tag{15}$$

em que X é um conjunto não-vazio, compacto e convexo, $f : X \times Y \rightarrow \mathbb{R}$ e $g : X \times Y \rightarrow \mathbb{R}^m$ são continuamente diferenciáveis. Se f e g são funções convexas, então (15) é um MINLP convexo.

Note que, dado um vetor $y \in \{0, 1\}^q$, (15) torna-se um problema de programação não-linear (NLP) com restrições. Uma possível estratégia de solução é enumerar todos os 2^q possíveis valores de y , para cada um deles resolver o NLP associado e escolher a solução de menor magnitude. Essa alternativa é ainda mais ineficiente e impraticável para o caso MINLP que para o caso MILP, visto que a solução de cada um desses subproblemas não lineares é computacionalmente muito custosa. Similarmente ao caso linear, os algoritmos propostos para o caso

¹²Observe que, novamente, o caso de variáveis discretas (mas não-binárias) também é contemplado nesta formulação.

não-linear procuram explorar seletivamente estes subproblemas associados.

A idéia básica do método de *OA* é a geração duas sequências (uma de cotas inferiores e a outra de cotas superiores) que convergem para a solução de (15) em um número finito de iterações. As cotas superiores são obtidas resolvendo-se (15) para um dado valor de y e as cotas inferiores são obtidas resolvendo-se um relaxamento de (15), no qual as funções não-lineares são aproximadas por linearizações. O detalhamento do método, novamente, requer a definição de alguns conceitos e objetos.

Definição 12 *Seja y^k um vetor tal que $y^k \in \{0, 1\}^q$ e P um MINLP definido como em (15). O NLP associado a P gerado pela fixação de $y = y^k$ é dado por*

$$\begin{aligned} \min_x & f(x, y^k) & (P(y^k)) \\ \text{s.a} & g(x, y^k) \leq 0 \\ & x \in X \subseteq \mathbb{R}^n \end{aligned}$$

se o problema for factível, e por

$$\begin{aligned} \min \sum_{i=1}^m u_i & & (P^F(y^k)) \\ \text{s.a} & g(x, y^k) - u \leq 0 \\ & u \geq 0 \\ & x \in X \quad u \in \mathbb{R}^m \end{aligned}$$

se $P(y^k)$ não for factível.

Definição 13 *Seja $\mathbb{T} = \{(x^1, y^1), (x^2, y^2), \dots, (x^K, y^K)\}$ um conjunto de pontos. A aproximação externa (outer approximation) de (15) segundo o conjunto \mathbb{T} é dada por*

$$\begin{aligned} \min_{\alpha, x, y} & \alpha & (P^{OA}(T)) \\ \text{s.a} & y \in \{0, 1\}^q \quad x \in X \quad \alpha \in \mathbb{R} \\ & \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + f(x^k, y^k) \leq \alpha \quad \forall (x^k, y^k) \in \mathbb{T} \\ & \nabla g(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + g(x^k, y^k) \leq 0 \quad \forall (x^k, y^k) \in \mathbb{T} \end{aligned}$$

Note que $(P^{OA}(T))$ é um MILP. A seguinte proposição é um corolário do teorema 1 de Bonami et al (2008).

Proposição 14 *Seja P um MINLP definido como em (15). Suponha que P seja factível e tenha um ótimo finito. Para todo $\bar{y} \in \{0,1\}^q$, se o problema $P(\bar{y})$ é factível, defina \bar{x} sua solução ótima. Se $P(\bar{y})$ não é factível, defina \bar{x} como sendo a solução ótima de $P^F(\bar{y})$. Defina \hat{T} como o conjunto de todos estes pares (\bar{x}, \bar{y}) . Se as condições de KKT são satisfeitas em todo ótimo de $P(\bar{y})$ (ou de $P^F(\bar{y})$, quando aplicável), então P e $P^{OA}(\hat{T})$ tem a mesma solução ótima.*

Com base neste resultado propõe-se o seguinte algoritmo:

Algoritmo 15 *Método OA*

- p0. Defina $k = 1$, escolha $\bar{y} \in \{0,1\}^q$, defina $y^1 = \bar{y}$, $c^u = \infty$, $c_l = -\infty$ e escolha o grau de tolerância ε .*
- p1. Resolva $P(y^k)$ (ou $P^F(y^k)$ se $P(y^k)$ não for factível). Seja x^k a solução encontrada. Adicione (x^k, y^k) a T .*
- p2. Se $P(y^k)$ é factível com valor ótimo z^k e $z^k \leq c^u$, atualize $c^u = z^k$ e $(x^*, y^*) = (x^k, y^k)$.*
- p3. Resolva $P^{OA}(T)$ para obter $(\alpha^k, \tilde{x}, y^{k+1})$. Atualize $c_l = \alpha^k$.*
- p4. Se $c^u - c_l > \varepsilon$, aumente k em uma unidade e retorne ao passo 1. Caso contrário, vá para o passo 5.*
- p5. Se $P^{OA}(T)$ é factível, então a solução ótima de P é dada por (x^*, y^*) , com valor ótimo c^u . Se $P^{OA}(T)$ é não factível, então P não é factível.*

O exemplo a seguir ilustra a operacionalização do algoritmo:

Exemplo 16 *Considere o seguinte MINLP, no qual há duas variáveis binárias e uma contínua e, portanto, $2^2 = 4$ NLP's associados. Note que o argumento minimizador é dado por $(x^*, y_1^*, y_2^*) = \left(\frac{-7\sqrt{2}}{10}, 0, 1\right)$ e o valor ótimo é $z^* = \frac{-7\sqrt{2}}{2}$.*

$$\begin{aligned}
 & \min_{x, y_1, y_2} \{5x + y_1\} \\
 \text{s.a} \quad & \left(y_1 - \frac{1}{10}\right)^2 + \left(y_2 - \frac{9}{10}\right)^2 + x^2 \leq 1 \\
 & (y_1, y_2) \in \{0, 1\}^2 \quad x \in \mathbb{R}
 \end{aligned}$$

De acordo com o algoritmo OA, em p_0 , defina $k = 1$, $\varepsilon = 10^{-2}$, escolha $(\bar{y}_1, \bar{y}_2) = (0, 0)$ e defina $y^1 = (y_1^1, y_2^1) = (\bar{y}_1, \bar{y}_2)$. Resolvendo $P(y^1)$ em p_1 , obtém-se $x^1 = \frac{-3\sqrt{2}}{10}$ e $z^1 = \frac{-3\sqrt{2}}{2}$. Atualize $T = \{(x^1, y^1)\}$. Em p_2 , atualize $c^u = z^1$ e $(x^*, y^*) = (x^1, y^1)$, já que $z^1 \leq c^u = \infty$. Em p_3 , resolva o seguinte problema:

$$\begin{aligned} \min_{\alpha, x, y} \alpha & \quad (P^{OA}(y^1)) \\ \text{s.a} \quad & 5x + y_1 - \alpha \leq 0 \\ & -3\sqrt{2}x - y_1 - 9y_2 - 1,8 \leq 0 \end{aligned}$$

em que a primeira restrição corresponde à linearização da função objetivo e a segunda à linearização da restrição não-linear, ambas em torno do ponto (x^1, y^1) . Na verdade, visto que a função objetivo é linear, a primeira restrição é a linearização da função objetivo para qualquer ponto (x^k, y^k) . Com isso, nos passos a seguir não serão adicionadas as linearizações da função objetivo. A solução deste problema é $(\alpha^1, \tilde{x}, y_1^2, y_2^2) = \left(\frac{6-59\sqrt{2}}{6}, \frac{-59\sqrt{2}}{30}, 1, 1\right)$. Atualize $c_l = \alpha^1$. Em p_4 verifique que $c^u - c_l = \frac{25\sqrt{2}-3}{3} \simeq 10,78 > \varepsilon$, atualize $k = 2$ e retorne a p_1 .

Resolvendo $P(y^2)$ em p_1 , obtém-se $x^2 = \frac{-3\sqrt{2}}{10}$ e $z^2 = 1 - \frac{3\sqrt{2}}{2}$. Atualize $T = \{(x^1, y^1), (x^2, y^2)\}$. Como $z^2 > c^u = -\frac{3\sqrt{2}}{2}$, passe para p_3 . Em p_3 , resolva o seguinte problema:

$$\begin{aligned} \min_{\alpha, x, y} \alpha & \quad (P^{OA}(y^2)) \\ \text{s.a} \quad & 5x + y_1 - \alpha \leq 0 \\ & -3\sqrt{2}x - y_1 - 9y_2 - 1,8 \leq 0 \\ & -3\sqrt{2}x + 9y_1 + y_2 - 11,8 \leq 0 \end{aligned}$$

em que a terceira restrição corresponde à linearização restrição não-linear, em torno de (x^2, y^2) . A solução deste problema é $(\alpha^2, \tilde{x}, y_1^3, y_2^3) = \left(-9\sqrt{2}, \frac{-9\sqrt{2}}{5}, 0, 1\right)$. Atualize $c_l = \alpha^2$. Em p_4 verifique que $c^u - c_l = \frac{15\sqrt{2}}{2} \simeq 10,6 > \varepsilon$, atualize $k = 3$ e retorne a p_1 .

Resolvendo $P(y^3)$ em p_1 , obtém-se $x^3 = \frac{-7\sqrt{2}}{10}$ e $z^3 = \frac{-7\sqrt{2}}{2}$. Atualize T para $\{(x^1, y^1), (x^2, y^2), (y_1^3, y_2^3)\}$. Em p_2 , atualize $c^u = z^3$ e $(x^*, y^*) = (x^3, y^3)$, já que $z^3 \leq c^u = -\frac{3\sqrt{2}}{2}$. Em

p3, resolva o seguinte problema:

$$\begin{aligned}
& \min_{\alpha, x, y} \alpha && (P^{OA}(y^3)) \\
& s.a \quad 5x + y_1 - \alpha \leq 0 \\
& \quad -3\sqrt{2}x - y_1 - 9y_2 - 1,8 \leq 0 \\
& \quad -3\sqrt{2}x + 9y_1 + y_2 - 11,8 \leq 0 \\
& \quad -7\sqrt{2}x - y_1 + y_2 - 10,8 \leq 0
\end{aligned}$$

em que a terceira restrição corresponde à linearização restrição não-linear, em torno de (x^2, y^2) . A solução deste problema é $(\alpha^3, \tilde{x}, y_1^4, y_2^4) = \left(\frac{-7\sqrt{2}}{2}, \frac{-7\sqrt{2}}{10}, 0, 1\right)$. Atualize $c_l = \alpha^3$. Em p4 verifique que $c^u - c_l = 0 < \varepsilon$, siga para p5. Em p5, conclui-se que o argumento minimizador é dado por $(x^*, y_1^*, y_2^*,) = \left(\frac{-7\sqrt{2}}{10}, 0, 1\right)$ e o valor ótimo é $z^* = \frac{-7\sqrt{2}}{2}$, exatamente a solução encontrada sem o auxílio do algoritmo.

Novamente o exemplo mostra de forma clara a atratividade do método: foi suficiente resolver somente 3 dos 4 NLP's associados ao MINLP. Não é necessário resolver o quarto NLP, pois da convergência das sequências de cotas inferiores e superiores pode-se concluir que a solução deste NLP não é a solução do MINLP. O descarte de NLP's é uma capacidade ainda mais importante do que o descarte de LP's, tendo em vista o maior custo computacional de solução de problemas não lineares.

4.3 Pacotes de Otimização

A implementação dos métodos numéricos de solução de MINLP's pode ser via a utilização de pacotes de otimização. Um pacote de otimização é um conjunto de rotinas que recebem como insumo os dados do problema e geram como produto a solução do problema (se esta existir) e informações a respeito da otimização, como o número de iterações, número de avaliações da função objetivo, etc. Tais pacotes podem ser de código aberto ou fechado (*black-box*). Pacotes disponibilizados via código aberto são em geral livres de pagamento de licenças, permitem que os usuários leiam as rotinas utilizadas e, em certos casos, edite-as como lhe convier. Adicionalmente, há o incentivo à participação dos usuários no desenvolvimento e aprimoramento do pacote. Já os pacotes de código fechado são em geral acessíveis somente via o pagamento de licenças de uso e não permitem o acesso ao código fonte das rotinas utilizadas. No entanto, estes

pacotes costumam utilizar inovações que ainda não foram disponibilizadas em código aberto (como subrotinas mais eficientes) e possuir interface com o usuário mais acessível.

4.3.1 BONMIN

Desenvolvido por pesquisadores da IBM e da Carnegie Mellon University, BONMIN (Basic Open-source Non-linear Mixed INteger programming) é uma rotina de otimização em código aberto para resolver MINLP's. É distribuído pela COIN-OR¹³ sob a Commom Public License (CLP), uma licença aprovada pela Open-Source Initiative (OSI). Ou seja, BONMIN é um pacote de otimização de código aberto.

Valendo-se do grande progresso obtido no desenvolvimento e implementação de algoritmos de solução de MILP's e NLP's, o BONMIN utiliza métodos que resolvem separadamente os subproblemas não lineares e os subproblemas discretos. Há quatro opções de algoritmo: (i) um método BB em que se resolve um NLP em cada nó da árvore explorado; (ii) um método OA baseado no algoritmo de decomposição; (iii) o método de branch-and-cut de Quesada & Grossmann (1992); e (iv) um algoritmo híbrido entre o método OA e o branch-and-cut.

Escrito em linguagem de programação C++, o pacote possui quatro interfaces com o usuário. A primeira delas é a linguagem de modelagem Ampl¹⁴, a qual converte uma descrição do problema próxima a linguagem matemática em linguagem de programação, criando um arquivo do tipo .nl a ser utilizado pelo pacote. Alternativamente, é possível invocar o pacote via linha de comando e e um arquivo .nl manualmente criado, conforme detalhado por Gay (2005). A linguagem de modelagem GAMS¹⁵ é outra opção que exige do usuário pouco conhecimento de linguagem de programação. Assim como o Ampl, ela também converte uma descrição relativamente abstrata do problema em linguagem computacional acessível ao BONMIN. A última alternativa é invocar o pacote via um programa escrito na própria linguagem C++. Esta última opção se destaca pela disponibilidade de formas de implementação livres de pagamento de licenças de uso¹⁶ e pela grande flexibilidade e poder computacional proporcionados pela linguagem C++.

¹³Computational Infrastructure for Operational Research (www.coin-or.org).

¹⁴www.ampl.com.

¹⁵www.gams.com.

¹⁶Programas em C++ podem ser implementados utilizando-se compiladores gratuitos.

4.3.2 TOMLAB

Conforme destacado por Holmstrom, Goran & Edvall (2009), TOMLAB é um ambiente de desenvolvimento, modelagem e controle ótimo implementado em Matlab visando o ensino, a pesquisa e soluções práticas de problemas de otimização. Ele disponibiliza uma coleção de algoritmos de otimização (*Solvers*) dos mais variados tipos de problemas, dentre eles MINLP's.

Por meio do Matlab, descreve-se o problema a ser resolvido utilizando-se *M-Files* e invoca-se um dos *solvers* disponibilizados pelo TOMLAB. Isto permite que o usuário se valha dos poderosos recursos matriciais disponibilizados pelo Matlab.

O *solver* disponibilizado pelo TOMLAB para a solução de MINLP's é o minlpBB. Diferentemente da estratégia adotada no desenvolvimento do BONMIN, o minlpBB não se vale das rotinas desenvolvidas para MILP's e NLP's. O faz em favor de abordagem específica ao MINLP, a qual explica sua maior eficiência, segundo Holmstrom, Goran & Edvall (2009). Tal abordagem se baseia no método BB, mas não resolve o NLP correspondente a cada nó da árvore considerado. Propõe-se proceder a exploração da árvore (*branching*) antes da convergência à solução ótima do NLP correspondente ao nó atual. Esta opção reduz o custo computacional, mas inviabiliza a utilização do critério de entendimento 3 (CE_3) descrito anteriormente. A solução é substituir tal critério por um teste de factibilidade dos problemas quadráticos gerados pelo *SQP solver*, um algoritmo desenvolvido para resolver NLP's por meio da solução de uma sequência de problemas quadráticos^{17,18}.

A principal vantagem deste pacote é a utilização do Matlab como interface com o usuário e a prometida eficiência do algoritmo utilizado.

5 Iteração do Conjunto de Estados Admissíveis

5.1 O Método

Os métodos descritos na seção anterior são suficientes para resolver o problema de otimização do planejador definido para um valor arbitrário das variáveis de estado (μ, v, z) . No entanto, não há conhecimento prévio do conjunto de valores admissíveis para estas variáveis, A . Isso, a princípio, inviabilizaria a solução do problema (FE). Felizmente, MC propõem um método

¹⁷Tal algoritmo também pode ser implementado via BONMIN, mas requer o pagamento de licença de uso aos seus desenvolvedores.

¹⁸Mais detalhes podem ser encontrados em Holmstrom, Goran & Edvall (2009).

capaz de calcular o conjunto A concomitantemente ao cálculo de w^* . A idéia básica é definir a correspondência Γ em um conjunto conhecido A_0 tal que $A \subset A_0$ e resolver (FE) para cada $(\mu, v, z) \in A_0$. Em seguida, o conjunto A_0 é atualizado excluindo-se os valores de (μ, v, z) tais que $\Gamma(\mu, v, z) = \phi$. O mesmo procedimento é repetido até a convergência da sequência de conjuntos gerada. O limite desta sequência é o conjunto A .

Vejamos os detalhes do método. Considere as correspondências $\Omega^b : \mathcal{H} \rightarrow \mathcal{H}$, $\Omega : \Omega^b \rightarrow \mathcal{H}$ e $X : \mathcal{H} \times \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{Y}^b$ tais que

$$\Omega^b(\mu, v, z) = \{(\mu', v') \in \mathcal{H}; \exists(y, g) \in \mathbb{Y}^b \times \mathbb{Y} \text{ com } (\mu', v', y, g) \in \Gamma(\mu, v, z)\} \quad (16)$$

$$\Omega(\mu, v, z) = \{(\mu', v') \in \Omega^b; \|v'\|_\infty \leq K\} \quad (17)$$

$$X[(\mu, v), (\mu', v'), z] = \{y \in \mathbb{Y}^b; \exists g \in \mathbb{Y} \text{ com } (\mu', v', y, g) \in \Gamma(\mu, v, z)\}$$

com as quais é possível definir a função

$$F[(\mu, v), (\mu', v'), z] = \begin{cases} \sup \{U(\mu, z, y); y \in X[(\mu, v), (\mu', v'), z]\} & \text{se } (\mu', v') \in \Omega(\mu, v, z) \\ -\infty & \text{se } (\mu', v') \notin \Omega(\mu, v, z) \end{cases}$$

Lembre-se que A é o conjunto de valores $(\mu, v, z) \in \mathbb{H}$ para os quais existe (y, g) e (μ', v') tais que $(\mu', v', y, g) \in \Gamma(\mu, v, z)$, $(\mu', v', z') \in A$ para todo $z' \in \mathcal{Z}$, e $\|v'\|_\infty \leq K$. Com a aplicação recursiva desta definição e a utilização de (17) e (16), A pode ser rescrito como¹⁹

$$A \equiv \{(\mu^1, v^1, z^1) \in \mathbb{H}; \forall \{z^t\}_{t=2}^\infty \text{ em } \mathcal{Z}, \exists \{(\mu^t, v^t)\}_{t=2}^\infty \text{ com } (\mu^t, v^t) \in \Omega(\mu^{t-1}, v^{t-1}, z^{t-1})\}$$

ou seja, o conjunto de estados admissíveis é o conjunto de valores para (μ, v, z) tais que, para toda possível história de choques agregados, existe sequência de distribuições e de retornos limitados de moeda que possui continuação (μ', v', y, g) em $\Gamma(\mu, v, z)$.

MC demonstram existe uma única função $w^* : \mathbb{H} \rightarrow \mathbb{R} \cup \{-\infty\}$ tal que

- (i) $w^*(\mu, v, z) = \sup \{F[(\mu, v), (\mu', v'), z] + \beta E[w^*(\mu', v', \cdot)]; (\mu', v') \in \mathcal{H}\}$
- (ii) $w^*(\mu, v, z) > -\infty$ se $(\mu, v, z) \in A$

Observe que w^* é o ponto fixo que resolve a equação funcional (FE) . A demonstração desta proposição é a base para o método de iteração do conjunto de estados admissíveis e será

¹⁹Aqui a relação de A com alocações implementáveis fica evidente.

explorada a seguir. Defina $\mathbb{U}_1 = \{(\mu, v, z) \in \mathbb{H}; \Omega(\mu, v, z) = \phi\}$ e para todo $k \in \{1, 2, \dots\}$

$$\mathbb{U}_{k+1} = \{(\mu, v, z) \in \mathbb{H}; \exists z' \in \mathcal{Z} \text{ com } (\mu', v', z') \in \mathbb{U}_k \ \forall (\mu', v') \in \mathcal{H}\}$$

O conjunto \mathbb{U}_1 é composto pelos valores (μ, v, z) que não são sustentáveis no período atual, no sentido de que não existe recomendação (y, g) e promessa v' tais que, dada a atual distribuição de moeda μ e o atual choque agregado z , satisfaçam as restrições de participação dos consumidores e dos produtores e honrem a promessa de retorno da moeda v . Os elementos de \mathbb{U}_2 são valores (μ, v, z) que não são sustentáveis em um período, no sentido de que, com probabilidade positiva, induz um estado (μ', v', z') não sustentável no próximo período. Por indução, conclui-se \mathbb{U}_{k+1} é o conjunto de valores (μ, v, z) que não são sustentáveis em k períodos, o que pode ser visto claramente se notarmos que \mathbb{U}_{k+1} pode ser reescrito como o conjunto

$$\{(x, z) \in \mathbb{H}; \exists (z^l)_{l=1}^k \text{ em } \mathcal{Z} \text{ tal que } \Omega(x^k, z^k) = \phi, \ \forall \{x^l\}_{l=1}^k \text{ com } x^l \in \Omega(x^{l-1}, z^{l-1})\}$$

em que $x = (\mu, v)$.

Seja T o operator que associa a cada função $w : \mathbb{H} \rightarrow \mathbb{R} \cup \{-\infty\}$ a função $Tw : \mathbb{H} \rightarrow \mathbb{R} \cup \{-\infty\}$ tal que

$$[Tw](\mu, v, z) = \sup \{F[(\mu, v), (\mu', v'), z] + \beta E[w(\mu', v', \cdot)]; (\mu', v') \in \mathcal{H}\}$$

e $\{w_n\}_{n=0}^\infty$ a sequência definida por uma função $w_0 : \mathbb{H} \rightarrow \mathbb{R}$ qualquer e a relação $w_{n+1} = Tw_n$ para todo $n \in \{0, 1, 2, \dots\}$. Para cada $n \in \{1, 2, \dots\}$, denote $[w_n = -\infty]$ o conjunto $\{(\mu, v, z) \in \mathbb{H}; w_n(\mu, v, z) = -\infty\}$. Observe que $[w_1 = -\infty] = \mathbb{U}_1$. Com efeito, se $(\mu, v, z) \in [w_1 = -\infty]$, então $F[(\mu, v), (\mu', v'), z] = -\infty$ para todo $(\mu', v') \in \mathcal{H}$ e, portanto, $\Omega(\mu, v, z) = \phi$. Por outro lado, se $(\mu, v, z) \in \mathbb{U}_1$, então $F[(\mu, v), (\mu', v'), z] = -\infty$ para todo $(\mu', v') \in \mathcal{H}$ e, portanto, $w_1(\mu, v, z) = -\infty$. Note também que $[w_2 = -\infty] = \mathbb{U}_1 \cup \mathbb{U}_2$. Para ver isto, suponha que $(\mu, v, z) \in [w_2 = -\infty]$, então $F[(\mu, v), (\mu', v'), z] = -\infty$ ou $E[w(\mu', v', \cdot)] = -\infty$ para todo $(\mu', v') \in \mathcal{H}$. No primeiro caso $(\mu, v, z) \in \mathbb{U}_1$ e no segundo $(\mu, v, z) \in \mathbb{U}_2$. Se $(\mu, v, z) \in \mathbb{U}_1 \cup \mathbb{U}_2$, então $F[(\mu, v), (\mu', v'), z] = -\infty$ ou $E[w(\mu', v', \cdot)] = -\infty$ para todo $(\mu', v') \in \mathcal{H}$, ambos implicando $w_2 = -\infty$. Por indução, conclui-se que $[w_n = -\infty] = \cup_{k=1}^n \mathbb{U}_k$. Com base nestas observações, MC mostram que $w^* = \lim_{n \rightarrow \infty} w_n$ e que $A = [\lim_{n \rightarrow \infty} \cup_{k=1}^n \mathbb{U}_k]^c$.

5.2 Implementação

A cada iteração da função valor e do conjunto de estados admissíveis, é necessário resolver o problema de otimização descrito na seção 4 para cada possível valor de $(\mu, v, z) \in \mathbb{H}$. Embora exista somente $k < \infty$ elementos em \mathcal{Z} , existe uma quantidade não enumerável de possíveis valores para μ e v . Com isso, \mathbb{H} é um conjunto não enumerável. A exigência computacional de dimensões finitas requer uma estratégia para contornar tal dificuldade. Descreve-se a seguir tal estratégia.

O modelo foi apresentado sob a hipótese de que a população era composta por um contínuo de pessoas. Este artifício, combinado com a hipótese de pareamentos aleatórios *iid*, possibilita que a incerteza individual em relação ao padrão de encontros não se traduza em incerteza agregada em relação a quantidade de cada tipo de encontro, já que se supõe válida a Lei dos Grandes Números. Adicionalmente, garante-se anonimato aos indivíduos, uma vez que o agente não pode ser identificado pelo seu tipo ou encaixe individual (há sempre um contínuo de agentes para cada tipo e nível de moeda) e a probabilidade de reencontro entre agentes é nula. O anonimato impede o monitoramento das histórias individuais e com isso confere essencialidade à moeda, conforme discutido em Kocherlakota (1998). Por outro lado, a hipótese de população contínua faz de \mathcal{M} um conjunto não enumerável. A alternativa óbvia para garantir que o número de possíveis distribuições de moeda seja finito é assumir que a população é finita. No entanto, não há na literatura em Teoria Monetária um modelo com população finita que garanta essencialidade à moeda. Seria necessário garantir anonimato aos indivíduos ou garantir que o monitoramento existente não seja elevado a ponto de a existência da moeda não proporcionar uma melhoria de Pareto à sociedade (o que confere essencialidade à moeda).

Para entender melhor as dificuldades associadas à hipótese de população finita, considere a noção de implementabilidade proposta por MC. As restrições de participação do consumidor e do produtor exigem que as recomendações sejam pelo menos tão atrativas quanto sua rejeição. Sob a hipótese de população contínua, o *payoff* de desvio unilateral de um agente com m unidades monetárias é dado por $\beta E[V(\mu', m, \cdot)]$, a utilidade esperada de possuir m unidades monetárias no próximo período sob a distribuição de moeda μ' . Ou seja, *payoff* de desvio (fora do equilíbrio) é determinado pelo *payoff* de se possuir m unidades de moeda no equilíbrio. Esta propriedade se deve ao fato de o planejador não ser capaz de identificar tais desvios, pois o desvio unilateral de um agente (o qual possui medida nula) não provoca qualquer alteração na distribuição futura de moeda. Como não identifica desvios, não pode puni-los. Além disso,

esta propriedade simplifica enormemente o espaço de estratégias do planejador, visto que não é necessário descrever as ações que seriam tomadas fora do equilíbrio. Sob a hipótese de população finita, esta propriedade não se verifica, já que agora os agentes possuem medida positiva e seus desvios mudam a distribuição de moeda futura, certas vezes revelando não só que houve um desvio, mas também os participantes do encontro no qual ele ocorreu. Isso possibilita que o planejador manipule o *payoff* de desvio a fim de implementar recomendações que não seriam implementáveis sob anonimidade. Existe a possibilidade de a capacidade de monitoramento de desvios do planejador ser tão grande que ele nem precise mais de moeda para implementar a produção e o consumo desejados para cada encontro. Dessa forma, a utilização da hipótese de população finita requer a formulação de uma noção de implementabilidade que não proporcione tamanha capacidade de monitoramento ao planejador.

Como o objetivo aqui não é conciliar população finita e essencialidade da moeda, é apresentar a implementação computacional da iteração do conjunto de estados admissíveis, suponha que os agentes não podem possuir mais de uma unidade monetária, ou seja, $\overline{M} = 1$. Esta hipótese trivializa a representação das distribuições de moeda, visto que basta descrever a proporção de agentes com moeda e a proporção de agentes sem moeda. Além disso, a restrição de que todas as distribuições possuem a mesma média faz de \mathcal{M} um conjunto unitário.

Mesmo com uma única distribuição de moeda possível, o conjunto \mathbb{H} ainda é não enumerável. Felizmente, isso não mais é uma fonte de problemas. É possível implementar a iteração do conjunto de estados admissíveis sem a necessidade de um número finito de elementos em $\mathbb{V}^{\mathbb{Z}}$. A estratégia aqui se baseia no fato de que, dado $\mu \in \mathcal{M}$ e $z \in \mathbb{Z}$, se $\tilde{v}(m) \leq \bar{v}(m)$ para todo $m \in \mathbb{M}$, então

$$\Gamma(\mu, \bar{v}, z) \subseteq \Gamma(\mu, \tilde{v}, z) \quad (18)$$

Esta propriedade permite que a solução do problema de otimização (apresentado na seção 4) para finitos valores de v seja suficiente para atualizar o conjunto de estados admissíveis, conforme descrito a seguir.

Lembre-se que o método de iteração do conjunto de estados admissíveis determina a escolha de um conjunto A_0 tal que $A \subseteq A_0$ e para a solução de (FE) para cada $(\mu, v, z) \in A_0$. Como μ é única no caso de $\overline{M} = 1$, a iteração se restringe ao conjunto de promessas v sustentáveis.

1. É de conhecimento que $v(m) \in [0, K]$ para todo²⁰ m em \mathbb{M} . Então defina $A_0 = \mathcal{M} \times$

²⁰Como v é sempre crescente em m , é possível restringir a otimização as combinações de (v_0, v_1) tais que $v_0 \leq v_1$, de forma a reduzir o custo computacional.

$[0, K]^2 \times \mathcal{Z}$.

2. Aproxime o conjunto $[0, K]^2$ utilizando um conjunto finito de pontos $(v_0, v_1) \in [0, K]^2$, denotado GR . Escolha $(\mu, V, z) \in \mathcal{M} \times GR \times \mathcal{Z}$.
3. Maximize (12) sujeito a $\Gamma(\mu, V, z)$ e $v'_z \in [0, K]^2$ para todo $z \in \mathcal{Z}$.
 - Se o problema é factível, então $\Gamma(\mu, V, z) \neq \phi$. De (18) tem-se que $\Gamma(\mu, v, z) \neq \phi$ para todo $v \in [0, K]^2$ tal que $v \leq V$.
 - Se o problema não é factível, então $\Gamma(\mu, V, z) = \phi$. De (18) tem-se que $\Gamma(\mu, v, z) = \phi$ para todo $v \in [0, K]^2$ tal que $v \geq V$.
4. Selecione $(\mu, V, z) \in GR$ tal que o status²¹ de $\Gamma(\mu, V, z)$ ainda não seja conhecido e retorne ao passo 3. Se todos os pontos (μ, V, z) de GR tem $\Gamma(\mu, V, z)$ com status conhecido, então termine²² a varredura do conjunto $[0, K]^2$.

Após os quatro passos destacados, obtém-se uma aproximação do conjunto de promessas sustentáveis no período atual sob choque z , denotada $PS_0(z)$, a qual será tão boa quanto a aproximação de GR a $[0, K]^2$. Tal aproximação, se não vazia, será uma união de retângulos contida em $[0, K]^2$ que possui $(0, 0)$ como elemento. O mesmo procedimento pode ser repetido para cada possível valor de $z \in \mathcal{Z}$. Essa repetição encerra a primeira iteração, $i = 0$, do conjunto de estados admissíveis. Cada uma das demais iterações, $i \geq 1$, é definida por:

1. Defina o conjunto A_i como o produto cartesiano de \mathcal{M} com o conjunto de pontos (v, z) tais que $v \in PS_{i-1}(z)$ e Γ_i como a correspondência $\Gamma : A_i \rightarrow A_i$, em que, além das restrições de participação, (9) e (10), e da IM , (11), exige-se que $v'_z \in PS_{i-1}(\bar{z})$ para todo $\bar{z} \in \mathcal{Z}$.
2. Para cada $z \in \mathcal{Z}$, execute os passos 3, 4 e 5.
3. Aproxime $PS_{i-1}(z)$ utilizando um conjunto finito de pontos $(v_0, v_1) \in PS_{i-1}(z)$, denotado GR_z^i .
4. Escolha $V \in GR_z^i$ tal que o status de $\Gamma_i(\mu, V, z)$ ainda não seja conhecido. Se todos os pontos V de GR_z^i tem $\Gamma_i(\mu, V, z)$ com status conhecido, então termine²³ a varredura do conjunto $PS_{i-1}(z)$ e defina $PS_i(z)$ o conjunto de pontos $V \in PS_{i-1}(z)$ tais que $\Gamma_i(\mu, v, z) \neq \phi$.

²¹Vazio ou não.

²²Para que a varredura seja completa é necessário do GR contenha os pontos $(0, 0)$ e (K, K) .

²³Para que a varredura seja completa é necessário do GR_z^i contenha os pontos da fronteira de $PS_{i-1}(z)$.

5. Maximize (12) sujeito a $\Gamma_i(\mu, V, z)$.

- Se o problema é factível, então $\Gamma_i(\mu, V, z) \neq \phi$. De (18) tem-se que $\Gamma_i(\mu, v, z) \neq \phi$ para todo $v \in [0, K]^2$ tal que $v \leq V$.
- Se o problema não é factível, então $\Gamma_i(\mu, V^0, z) = \phi$. De (18) tem-se que $\Gamma_i(\mu, v, z) = \phi$ para todo $v \in [0, K]^2$ tal que $v \geq V$.
- Retorne ao passo 4.

Após cada iteração i , obtém-se uma aproximação do conjunto de promessas sustentáveis em i períodos. O procedimento de iteração do conjunto de estados admissíveis é executado até a convergência da sequência A_i . Este limite é o conjunto A .

A generalização desse método de iteração para o caso em que $\overline{M} > 1$ é direta, basta definir e iterar o conjunto de promessas sustentáveis PS não só para todos os valores de $z \in \mathcal{Z}$, mas para toda combinação de z e μ . A questão em aberto é a existência de um método que não exija \mathcal{M} finito, similar ao método que não exige que o conjunto \mathbb{V}^Z seja finito, de forma que seja possível manter a hipótese de população contínua ou de uma noção apropriada de implementabilidade que permita supor uma população finita sem eliminar a essencialidade da moeda.

6 Considerações Finais

Este trabalho mostra que é possível implementar computacionalmente o método recursivo proposto por Monteiro e Cavalcanti (2006) e com isso, caracterizar a sequência ótima de distribuições de moeda em uma economia em que as trocas se dão em encontros aleatórios. Em linhas gerais, escolhe-se um palpite inicial qualquer para a função valor do problema do planejador apresentado na seção 2. Em cada iteração da função valor e do conjunto de estados admissíveis, resolve-se um MINLP para cada possível valor de variável de estado utilizando os métodos e pacotes de otimização apresentados na seção 4. A aplicabilidade destes métodos requer que o MINLP seja côncavo, o que é garantido pela técnica de linearização descrita na seção 3. Os conjuntos de promessas sustentáveis, um para cada combinação de z e μ , são então atualizados segundo o método discutido na seção 5. Os resultados de convergência da função valor e do conjunto de estados admissíveis sob este método são demonstrados em Monteiro e Cavalcanti (2006) e foram informalmente discutidos aqui.

A implementação de fato do método recursivo é extensão óbvia deste trabalho. A utilização da linguagem C++, utilizando o pacote de otimização BONMIN, surge como a alternativa mais atrativa. A experiência até aqui adquirida com a utilização do pacote TOMLAB, mostra que o problema de otimização é eficientemente computável²⁴ no caso de $\overline{M} = 1$. Uma pergunta relevante a ser respondida é a dimensão máxima (\overline{M} máximo) passível de solução computacional, o que requer a solução das questões deixadas em aberto na seção 5.

²⁴Sendo necessários, em média, dois a três segundos por maximização.

7 Bibliografia

Bonami, P.; Biegler, L.; Conn, A.; Cornuéjols, G.; Grossmann, I.; Laird, C.; Lee, J.; Lodi, A.; Margot, F.; Sawaya, N.; Wachter, A. (2008). An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs. *Discrete Optimization*, vol. 5(2): 186-204.

Deviatov; Wallace, N. (2009). A Model in Which Monetary Policy is about Money, *Journal of Monetary Economics*, vol. 56(3): 283-288.

Floudas, C. (1995) *Nonlinear and Mixed-Integer Optimization : Fundamentals and Applications*, Oxford University Press.

Gay, D. (2005) Writing .nl Files. Sandia National Laboratories, Technical Report No. 2005-7907P

Glover, F. (1975). Improved Linear Integer Programming Formulations of Nonlinear Integer Problems, *Management Science*, vol. 22(4):455-460.

Holmstrom, K.; Goran, A.; Edvall, M. (2009) User's guide for TOMLAB/MINLP. Technical report, Tomlab Optimization Inc., San Diego, CA.

Kiyotaki, N.; Wright, R. (1989) On Money as a Medium of Exchange. *The Journal of Political Economy*, vol. 97(4): 927-954.

Kocherlakota, N. (1998). Money is Memory, *Journal of Economic Theory*, vol. 81(2): 232-251.

Lagos, R.; Wright, R. (2005) A Unified Framework for Monetary Theory and Policy Analysis. *Journal of Political Economy*, vol. 113(3): 463–484.

Monteiro, P.; Cavalcanti, R. (2006). On the Optimum Distribution of Money, manuscript.

Quesada, I.; Grossmann (1992). An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems. *Computers & Chemical Engineering*, vol. 16: 937-947.